УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У НОВОМ САДУ

# Релативна експресивност процесних рачуна који поседују могућност адаптације и динамичког ажурирања током извршавања

ДОКТОРСКА ДИСЕРТАЦИЈА

# Relative Expressiveness of Process Calculi with Dynamic Update and Runtime Adaptation

DOCTORAL DISSERTATION

Ментори:

Проф. др Јованка Пантовић

Prof. dr Jorge Andres Pérez Parra

Кандидат:

Јована Дедеић

Нови Сад, 2021. године

**УНИВЕРЗИТЕТ У НОВОМ САДУ** **ОБРАЗАЦ – 5а**
**ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА**

## КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА[1]

| | |
|---|---|
| Врста рада: | Докторска дисертација |
| Име и презиме аутора: | Јована Дедеић |
| Ментор (титула, име, презиме, звање, институција) | др Јованка Пантовић, редовни професор, Факултет техничких наука, Универзитет у Новом Саду,<br>др Хорхе Андрес Перез Пара, ванредни професор, Универзитет у Гронингену |
| Наслов рада: | Релативна експресивност процесних рачуна који поседују могућност адаптације и динамичког ажурирања током извршавања |
| Језик публикације (писмо): | Енглески |
| Физички опис рада: | Унети број:<br>Страница 205<br>Поглавља 7<br>Референци 56<br>Табела 8<br>Слика 41<br>Графикона 0<br>Прилога 0 |
| Научна област: | Примењена математика |
| Ужа научна област (научна дисциплина): | Формални модели у рачунарству |
| Кључне речи / предметна одредница: | Конкурентни системи, дистрибуирани системи, семантика програмских језика, процесни рачуни, руковање компензацијом, динамичко ажурирање, експресивност |
| Резиме на језику рада: | У тези су разматрани проблеми програмских конструката који подржавају управљање грешкама у центру механизама који откривају грешке и враћају систем у конзистентно стање. Теза формално повезује програмске апстракције за руковање компензацијама и динамичког ажурирања током извршавања. Анализира се релативна експресивност поменутих рачуна. Развијено је дванаест кодирања, шест процесних рачуна за руковање компензацијама у два рачуна за адаптивне процесе. |
| Датум прихватања теме од стране надлежног већа: | 24.09.2020. |
| Датум одбране: (Попуњава одговарајућа служба) | |
| Чланови комисије: (титула, име, презиме, звање, институција) | Председник:<br>др Јелена Иветић, ванредни професор,<br>Факултет техничких наука, Универзитет у Новом Саду |

[1] Аутор докторске дисертације потписао је и приложио следеће Обрасце:
5б – Изјава о ауторству;
5в – Изјава о истоветности штампане и електронске верзије и о личним подацима;
5г – Изјава о коришћењу.
Ове Изјаве се чувају на факултету у штампаном и електронском облику и не кориче се са тезом.

| | |
|---|---|
| | Члан:<br>др Душан Гајић, ванредни професор,<br>Факултет техничких наука, Универзитет у Новом Саду<br><br>Члан:<br>др Иван Прокић, доцент,<br>Факултет техничких наука, Универзитет у Новом Саду<br><br>Члан:<br>др Хуго Филипе Мендес Торес Виеира, виши научни сарадник,<br>Универзитет Беира Интериор Ковиља<br><br>Члан, ментор:<br>др Хорхе Андрес Перез Пара, ванредни професор,<br>Универзитет у Гронингену<br><br>Члан, ментор:<br>др Јованка Пантовић, редовни професор,<br>Факултет техничких наука, Универзитет у Новом Саду |
| Напомена: | |

**UNIVERSITY OF NOVI SAD**
**FACULTY OF TECHNICAL SCIENCES**

**KEY WORD DOCUMENTATION**[2]

| | |
|---|---|
| Document type: | Doctoral dissertation |
| Author: | Jovana Dedeić |
| Supervisor (title, first name, last name, position, institution) | Dr. Jovanka Pantović, full professor, Faculty of Technical Sciences, University of Novi Sad,<br>Dr. Jorge Andres Pérez Parra, associate professor, University of Groningen |
| Thesis title: | Relative Expressiveness of Process Calculi with Dynamic Update and Runtime Adaptation |
| Language of text (script): | English |
| Physical description: | Number of:<br>Pages 205<br>Chapters 7<br>References 56<br>Tables 8<br>Illustrations 41<br>Graphs 0<br>Appendices 0 |
| Scientific field: | Applied mathematics |
| Scientific subfield (scientific discipline): | Formal Models in Computer Science |
| Subject, Key words: | Concurrent systems, distributed systems, semantics of programming languages, process calculi, compensation handling, dynamic update, expressiveness |
| Abstract in English language: | The thesis considers problems of programming constructs that support failure handling at the heart of mechanisms that detect failures and bring the system back to a consistent state. We formally connect programming abstractions for compensation handling and runtime adaptation and analyzes the relative expressiveness of these calculi. More concrete, we develop twelve encodings of six process calculi with compensation handling into two calculi of adaptable processes. |
| Accepted on Scientific Board on: | 24.09.2020. |
| Defended:<br>(Filled by the faculty service) | |
| Thesis Defend Board:<br>(title, first name, last name, position, institution) | President:<br>dr. Jelena Ivetić, associate professor,<br>Faculty of Technical Sciences, University of Novi Sad |

[2] The author of doctoral dissertation has signed the following Statements:
5б – Statement on the authority,
5в – Statement that the printed and e-version of doctoral dissertation are identical and about personal data,
5г – Statement on copyright licenses.
The paper and e-versions of Statements are held at he faculty and are not included into the printed thesis.

| | |
|---|---|
| | Member:<br>dr. Dušan Gajić, associate professor,<br>Faculty of Technical Sciences, University of Novi Sad<br><br>Member:<br>dr. Ivan Prokić, assistant professor,<br>Faculty of Technical Sciences, University of Novi Sad<br><br>Member:<br>dr. Hugo Filipe Mendes Torres Vieira, senior researcher,<br>C4, Universidade de Beira Interior, Covilhã<br><br>Member, Mentor:<br>dr. Jorge Andres Pérez Parra, associate professor,<br>University of Groningen<br><br>Member, Mentor:<br>dr. Jovanka Pantović, full professor,<br>Faculty of Technical Sciences, University of Novi Sad |
| Note: | |

# Acknowledgements

Jovanka Pantović and Jorge A. Pérez deserve my deepest gratitude. It has been truly inspiring to have them as supervisors. Their close supervision has had a significant impact on the way I conduct and approach research work. I especially appreciate that Jovanka always made time for me, not only for scientific discussions but also for resolving mundane difficulties. I was fortunate enough to meet Jorge at the beginning of my research career. I express my gratitude to Jorge, especially because he was a wonderful host during my visits to Groningen, where most of the results of this dissertation have been obtained.

In addition, I am grateful to the following members of my defense committee: Dr. Jelena Ivetić, Dr. Dušan Gajić, Dr. Ivan Prokić, and Dr. Hugo Filipe Mendes Torres Vieira for their effectiveness, constructive comments, and suggestions.

I also express my gratitude to all my dear colleagues at the Chair of Mathematics, Faculty of Technical Sciences. They welcomed me into the collective and give me support during my professional career.

Thank you so much to all of my friends for their unwavering support and love over the years. Marija, Boško, and Sanja deserve special recognition for being my professional partners and sincere friends throughout my entire career and life.

My family members, without a doubt, are the persons to whom I owe the most. Thank you, my parents, Veselinka and Danilo, for your unwavering love and support throughout life. One of the most crucial sources of motivation in my life has been their love and kindness. I am grateful for all of their advice and care. Thanks to them, I can persevere in all life and professional challenges. Thank you for being perfect siblings, Jelena, Mira, and Milan. Throughout my life, you take the time to be in the right places at the right times. I would not be the same without you, and I am lucky to have grown up with you.

Milena, Mirčeta, and Sanda, my “in-law” family, have been tremendously supportive.

Last but not least, an immense thanks to my husband Strahinja and our sons Danilo and Luka for their unquestionable love, support, and patience. This thesis would not have been finished without their constant support and motivation. Thank you boys, I am happy to have you in my life.

# Rezime

## 0.1 Motivacija

Razvoj informacionih tehnologija (IT): računarstvo u oblaku, računarstvo orijentisano na usluge, veštačka inteligencija, analiza podataka, praćenje i predviđanje itd. podržani su od strane velike računarske infrastrukture, kao što su centri podataka. Takođe, IT često koriste bežične i mobilne mreže, paralelne i distribuirane sisteme. Distribuirani sistem je sistem čije se komponente nalaze na različitim računarima, koji su umreženi, komuniciraju i koordinišu svoje procese prenoseći poruke među sobom. Stoga, možemo zaključiti da su distribuirani sistemi pogodniji za rešavanje problema u poređenju sa sekvencijalnim sistemima. Ipak, treba imati na umu da prilikom modeliranja i implementacije sistema, distribuirani sistemi nameću određene prepreke. U svim softverskim sistemima *komunikacija* i *interakcija* postali su centralne karakteristike sistema. Takođe, analiza i verifikacija softvera je složen i izazovan zadatak jer postoji opšta pretpostavka da softverski sistemi moraju da rade neograničeno i bez neočekivanog prekida. Tokom prethodnih nekoliko decenija, infrastruktura poput one koja podržava računarstvo visokih performansi porasla je u obimu i složenosti. Njihova snaga, fleksibilnost i pogodnost idu zajedno sa potrebom za efikasnom potrošnjom energije.

Veliki računarski sistemi sve češće doživljavaju prekide ili greške, a mehanizmi/tehnike za njihovo prevazilaženje su od presudne važnosti. *Formalne metode* su tehnike koje omogućavaju formalnu specifikaciju i verifikaciju složenih (softverskih i hardverskih) sistema, zasnovane na *matematici* i *logici*. Postoji veliki broj formalnih metoda koje se koriste za poboljšanje razvoja softvera, koji radi na velikim računarskim infrastrukturama. Formalne metode imaju primarni zadatak, da predvide mogućnost pojave greške u aplikacijama i obezbede pravovremenu reakciju. Takođe, one treba da obezbede da ne dođe do nepotrebnog rasipanja resursa (kao što je, na primer, energija). Jedna od najčešćih formalnih metoda koji se koristi za analizu složenih sistema je *proceni račun* (ili procesna algebra). Procesni računi su raznovrsna porodica povezanih pristupa za formalno modeliranje složenih sistema. Stoga procesni računi se mogu koristiti za izražavanje različitih koncepata, na primer, nedeterminizma, paralelizma, distribucije, problema u realnom vremenu, stohastičkih fenomena, itd. Kako je navedeno u [38], procesne algebre dolaze sa preciznim matematičkim okvirom koji ima dobro definisanu *sintaksu* i *operacionu semantiku*. Operaciona semantika opisuje i verifikuje svojstava konkurentnih komunikacionih sistema. Dakle, procesna algebra se fokusira na specifikaciju i manipulaciju procesnim termima baziranih na kolekciji simbola operatora [18], koji se koriste za konstrukciju: konačnih procesa, paralelno izvršavanje, komunikaciju i oblik rekurzije za izražavanje beskonačnog ponašanja. Primarna komponenta procesne algebre je sintaksa. Kada je jezik sintaksno definisan, onda je ključno obezbediti način da se opiše ponašanje sistema koji se modelira, što se postiže uvođenjem operacione semantike. Operaciona semantika treba da opiše način na koji se proces realizuje/redukuje. Tokom proteklih četrdeset godina, istraživački rad na procesnim algebrama je veoma intenzivan i objavljen je zančajan broj rezultata. Kratak istorijski pregled razvoja procesne algebre predstavljen je u radu [3]. Milnerov $\pi$-račun [33] u novije vreme je postao značajan kao procesni račun za razmišljanje o mobilnim sistemima. Postoji značajan broj računa za teoriju konkurentnih sistema u kojima se $\pi$-račun koristi kao osnova [1, 12, 20, 24, 23, 28, 41, 47, 29, 30, 49].

Analizom softverskih aplikacija ustanovljeno je da je veliki broj istih zasnovan na *dugoročnim transakcijama* (eng. long-running transactions) kao osnovnom gradivnom elementu. Dugoročne transakcije se često primenjuju u servisno orijentisanim sistemima [17], i opisuju vremenski opsežne aktivnosti koje uključuju nekoliko distribuiranih komponenti slabo povezanih resursa. U računarskim naukama: *atomičnost, doslednost, izolovanost i trajnost* (eng. atomicity, consistency, isolation, and durability – ACID) je skup svojstava transakcija čiji je cilj da garantuje validnost čak i u slučaju da se pojave greške, dođe do nestanka struje ili neke druge nepredviđene situacije. Dugoročne transakcije ipak ne zadovoljavaju sva navedena svojstva. Tačnije, one ne zadovoljavaju *izolovanost* jer izvođenje jedne dugoročne transakcije nema za cilj blokiranje celog sistema. Odnosno, zbog prirode ovih sistema i vremenskog trajanja aktivnosti, nije moguće zaključati (nelokalne) resurse. Za upravljanje dugoročnim transakcijama, otkazivanje upravljanja je osetljiv aspekt: potrebno je eksplicitno programirati mehanizme za otkrivanje grešaka i vraćanje dugoročne transakcije u konzistentno stanje. Budući da je projektovanje i potvrđivanje ispravnosti ovih mehanizama sklono greškama, specijalizovani konstrukti, kao što su *izuzeci* i *kompenzacije*, predloženi su da ponude direktnu programsku podršku.

Literatura nudi različite konstrukte. U Javi, na primer, nalazimo konstrukciju „obrada izuzetka”, eng. try $P$ catch $e$ $Q$, gde je proces $Q$ zadužen za upravljanje *izuzetkom* $e$ koji je podignut unutar procesa $P$; u WS-BPEL [2] nalazimo napredne mehanizme koji koriste grešku, prekid i kompenzaciju za rešavanje greške u kodu.

U ovoj disertaciji fokus je na istraživanju programskih konstrukata koji podržavaju upravljanje greškama u centru mehanizama koji otkrivaju kvarove i vraćaju sistem u konzistentno stanje. Kao što im ime sugeriše, *mehanizmi za kompenzacije* imaju za cilj da kompenzuju činjenicu da je dugoročna transakcija naišla ne grešku ili je otkazana (tj. nije uspela da se realizuje). Po prijemu signala greške, od mehanizama za kompenzaciju se očekuje da instaliraju i aktiviraju alternativna ponašanja za oporavak doslednosti sistema. Takvo kompenzaciono ponašanje može se razlikovati od početnog ponašanja dugoročne transakcije. Široko proučavani u servisno orijentisanim sistemima, mehanizmi za upravljanje kompenzacijama takođe nalaze primenu u kolektivno adaptivnim sistemima (bar konceptualno), posebno zato što se autonomni uređaji počinju koristiti u tradicionalnim transakcionim aktivnostima, poput distribucije i isporuke, na primer, Amazon Prime Air i DHL-ov Parcelcopter.

U literaturi su predloženi različiti računi za konkurentne sisteme sa konstruktima za upravljanje kompenzacijama (na primer, [5, 10, 31, 11, 17]). Nadovezujući se na procesne račune kao što su Milnerov račun komunikacionih sistema (eng. Calculus of Communicating Systems — CCS) [32], Hoareov račun komunikacionih sekvencijalnih procesa (eng. Communicating Sequential Processes — CSP) [27] i Milnerov $\pi$-račun [33], oni obuhvataju različite oblike oporavka sistema usled greške i nude tehnike rezonovanja (npr. bihevioralnu ekvivalenciju) o komuniciranju procesa koji sadrže konstrukte za kompenzacije. Sličnosti između mnogobrojih različitih predloga (računa) nisu uvek sasvim jasni, a u literaturi postoje radovi koji imaju za cilj formalno upoređivanje ekspresivnosti predloženih mehanizama. Konkretno, ekspresivna moć takvih procesnih računa proučavana je u radovima [11, 8, 29, 30]. Lanese sa koautorima u radu [29] bavi se ovim pitanjem razvijajući formalno poređenje različitih pristupa dugoročnim transakcijama u konkuretnom i mobilnom okruženju. U [29] autori razmatraju procesni jezik koji sadrži različite mehanizme za rukovanje greškama.

Detaljnije, Lanese sa koautorima u [29] definiše osnovni račun sa kompenzacijama, koji proširuje Milnerov $\pi$-račun sa sledećim procesima:

- *transakcija* $t[P,Q]$, gde procesi $P$ i $Q$ predstavljaju *podrazumevanu* i *kompenzacionu* aktivnost, respektivno,
- *zaštićen blok* $\langle Q \rangle$ i
- *ažuriranje kompenzacije* $\mathtt{inst}\lfloor \lambda X.Q \rfloor.P$, koji ponovo konfiguriše kompenzacionu aktivnost $Q$.

| **Kompenzabilni procesi (Statički oporavak) – $\mathcal{C}$** | **Oznaka** |
|---|---|
| Račun za kompenzabilne procese sa semantikom odbacivanja | $\mathcal{C}_{\mathtt{D}}$ |
| Račun za kompenzabilne procese sa semantikom očuvanja | $\mathcal{C}_{\mathtt{P}}$ |
| Račun za kompenzabilne procese sa semantikom prekida | $\mathcal{C}_{\mathtt{A}}$ |
| **Kompenzabilni procesi (Dinamički oporavak) – $\mathcal{C}^{\lambda}$** | **Oznaka** |
| Račun za kompenzabilne procese sa semantikom odbacivanja | $\mathcal{C}^{\lambda}_{\mathtt{D}}$ |
| Račun za kompenzabilne procese sa semantikom očuvanja | $\mathcal{C}^{\lambda}_{\mathtt{P}}$ |
| Račun za kompenzabilne procese sa semantikom prekida | $\mathcal{C}^{\lambda}_{\mathtt{A}}$ |
| **Adaptivni procesi** | **Oznaka** |
| Račun za adaptivne procese sa subjektivnim ažuriranjem | $\mathcal{S}$ |
| Račun za adaptivne procese sa objektivnim ažuriranjem | $\mathcal{O}$ |

Slika 1: Oznake za procesne račune koji se koriste u tezi.

Procesni račun sa kompenzacijama sačinjen je od *statičkog* i *dinamičkog oporavka*. Ukoliko nije dopušteno ažuriranje kompenzacione aktivnosti onda za procesni račun sa kompenzacijama kažemo da je sa statičkim oporavkom, u suprotnom je sa dinamičkim oporavkom. U ovom procesnom računu odgovor na greške može se realizovati uz pomoć tri semantike:

- *semantike odbacivanje* (eng. discarding semantics),
- *semantike očuvanja* (eng. preserving semantics),
- *semantike prekida* (eng. aborting semantics).

Jezik u [29], shodno gore navedenom, ima šest različitih formalnih računa koji sadrže osnovne elemente za kompenzacije, Slika 1.

Blisko povezan sa mehanizmima za rukovanje kompenzacijama, ali na drugačiji način, predložen je procesni račun za *adaptivne procese*. Procesni račun za adaptivne procese je predložen za specifikaciju *dinamičkog ažuriranja* u komunikacionim sistemima [7]. Adaptivni procesi određuju oblike dinamičke rekonfiguracije koji su pokrenuti usled nekog neočekivanog događaja, ali ne nužno katastrofalnog. Jednostavan primer je rekonfiguracija specifičnih jedinica rojeva robota (eng. robot swarm), što je obično teško predvideti i podrazumeva promenu ponašanja uređaja. Adaptivni procesi mogu se primeniti na *lokacijama*, koje služe kao graničnici za dinamička ažuriranja. Proces $P$ koji se nalazi na lokaciji $l$, označen sa $l[P]$, može ponovo da se rekonfiguriše pomoću *prefiksa za ažuriranje* $l\{(X).Q\}.R$, gde proces $Q$ označava rutinu prilagođavanja za lokaciju $l$, parametrizovanu procesnom promenljivom $X$. Sa ova dva konstrukta, dinamičko ažuriranje se ostvaruje prema sledećem redukcionom pravilu, u kojem $C_1$ i $C_2$ označavaju kontekste proizvoljno ugneždenih lokacija:

$$C_1\big[l[P]\big] \mid C_2\big[l\{(X).Q\}.R\big] \longrightarrow C_1\big[Q\{^P/X\}\big] \mid C_2\big[R\big] \tag{1}$$

Ovaj tip ažuriranja nazivamo *objektivnim ažuriranjem*: locirani proces se ponovo konfiguriše u svom kontekstu koristeći prefiks za ažuriranje, koji se nalazi u drugom kontekstu.
Zaista, prefiks ažuriranja $l\{(X).Q\}$ komunicira sa procesom ažuriranja $l[P]$ i *premešta* proces $Q$ iz konteksta $C_2$ u kontekst $C_1$, tako da je rekonfigurisano ponašanje $Q\{^P/X\}$ ostavljeno u

kontekstu $C_1$. Procesna promenljiva $X$ se može pojaviti nula ili više puta u procesu $Q$. Napominjemo, ako $Q$ ne sadrži $X$, kao rezultat ažuriranja trenutno ponašanje $P$ će biti obrisano. Na ovaj način, dinamičko ažuriranje je oblik mobilnosti procesa, implementirano korišćenjem *komunikacije procesa višeg reda* (eng. higher-order process communication). Ovakav oblik komunikacije nalazi se u jezicima kao što su, na primer, $\pi$-račun višeg reda [51], Kelov račun [52] i Homer [25].

Alternativa objektivnom ažuriranju je *subjektivno ažuriranje* u kojem se rekonfiguracija procesa odvija u suprotnom pravcu: proces $P$ na lokaciji $l$ se pomera iz svog konteksta u kontekst u kojem je smešten prefiks za ažuriranje:

$$C_1\big[l[P] \mid R_1\big] \mid C_2\big[l\{(X).Q\}.R\big] \longrightarrow C_1\big[\mathbf{0} \mid R_1\big] \mid C_2\big[Q\{^P/X\} \mid R\big] \tag{2}$$

Kao i objektivno ažuriranje, subjektivno ažuriranje se oslanja na mobilnost procesa. Međutim, kao što je već napomenuto, pravac pomeranja procesa se razlikuje. U (2) proces $P$ se premešta iz konteksta $C_1$ u kontekst $C_2$, a rekonfigurisno ponašanje $Q\{^P/X\}$ ostaje u svom kontekstu $C_2$. Primer koji sledi, ilustruje objektivno i subjektivno ažuriranje.

**Primer 1.** Poredimo subjektivno i objektivno ažuriranje pomoću primera koji smo pruzeli iz [7] i adekvatno prilagodili. Posmatramo *operator prekida* koji započinje izvršavanje procesa $P$, ali može napustiti njegovo izvršavanje radi izvršavanja procesa $Q$. Kada $Q$ emituje signal za prekid $t_Q$, operator se vraća da izvrši ono što je preostalo od procesa $P$. Koristeći adaptivne procese, ova vrsta ponašanja moše se opisati kao što sledi u nastavku:

$$Sys = l_1\big[l[P] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big]$$

gde su $l$, $l_1$ i $l_2$ različite lokacije, a ime $t_Q$ je poznato samo procesu $Q$. Proces $Q$ ne sadrži procesnu promenljivu $X$. Ako proces $P$ evoluira u proces $P'$ neposredno pre nego što bude prekinut, koristeći sematiku sa objektivnim ažuriranjem imamo sledeći scenario:

$$\begin{aligned} Sys &\longrightarrow^* l_1\big[l[P'] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big] \\ &\longrightarrow\ \ l_1\big[Q \mid t_Q.P' \mid R_1\big] \mid l_2\big[R_2\big] \\ &\longrightarrow^* l_1\big[P' \mid R_1\big] \mid l_2\big[R_2\big] \end{aligned}$$

Na ovaj način dobijeno je da proces $P$ i njegov derivat $P'$ ostaju na lokaciji $l_1$. Treba uočiti da bi izvršavanje $Sys$ upotrebom semantike sa subjektivnim ažuriranjem dovelo do drugačijeg ponašanja, jer bi proces $P'$ (kao i proces $Q$) pogrešno bio premešten na lokaciju $l_2$:

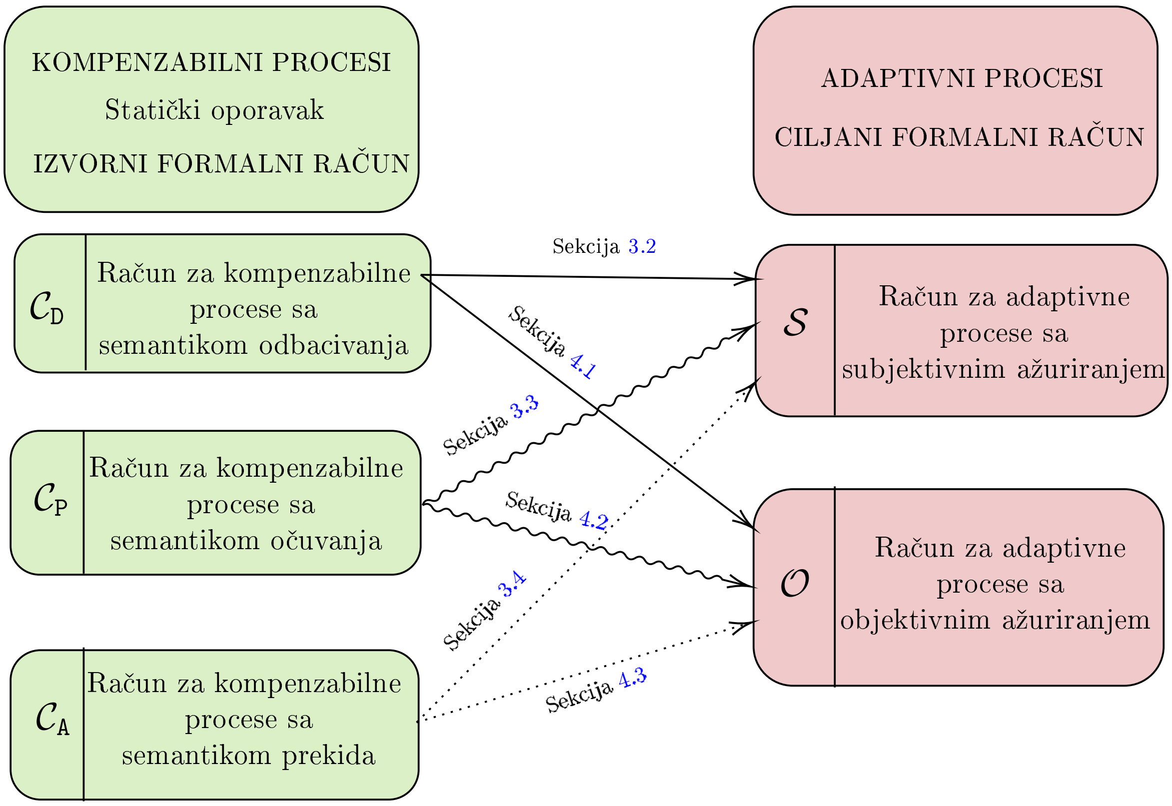
$$\begin{aligned} Sys &\longrightarrow^* l_1\big[l[P'] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big] \\ &\longrightarrow\ \ l_1\big[R_1\big] \mid l_2\big[Q \mid t_Q.P' \mid R_2\big] \\ &\longrightarrow^* l_1\big[R_1\big] \mid l_2\big[P' \mid R_2\big] \end{aligned}$$

Ovo pokazuje da bi se za postizanje planiranog ponašanja prekida u subjektivnom okruženju $Sys$ trebao izmeniti, kako bi se proces $P'$ na kraju vratio na lokaciju $l_1$. Sledeća varijacija $Sys$ to postiže:

$$Sys' = l_1\big[l[P] \mid l'\{(X).X\}.R_1\big] \mid l_2\big[l\{(X).l'[Q \mid t_Q.X]\}.R_2\big]$$

gde se koristi $l'$ kao pomoćna lokacija koja treba da posluži da se proces $P'$ iz lokacije $l_2$ vrati na lokaciju $l_1$.

Na osnovu prethodnog pregleda procesnih računa sa kompenzacijama i adaptacijama, važno je da primetimo da su kompenzacije i ažuriranje intuitivno slični. Sličnost ovih računa ogleda se u tome što oba određuju kako se osobina konkurentnog sistema menja u vremenu kao odgovor na neočekivani događaj. Sa druge strane, treba naglasiti da su ovi računi tehnički veoma različiti.




Slika 2: Kodiranje $\mathcal{C}_{\mathtt{D}}, \mathcal{C}_{\mathtt{P}}, \mathcal{C}_{\mathtt{A}}$ u $\mathcal{S}$ i $\mathcal{O}$. Strelica označava kodiranje.

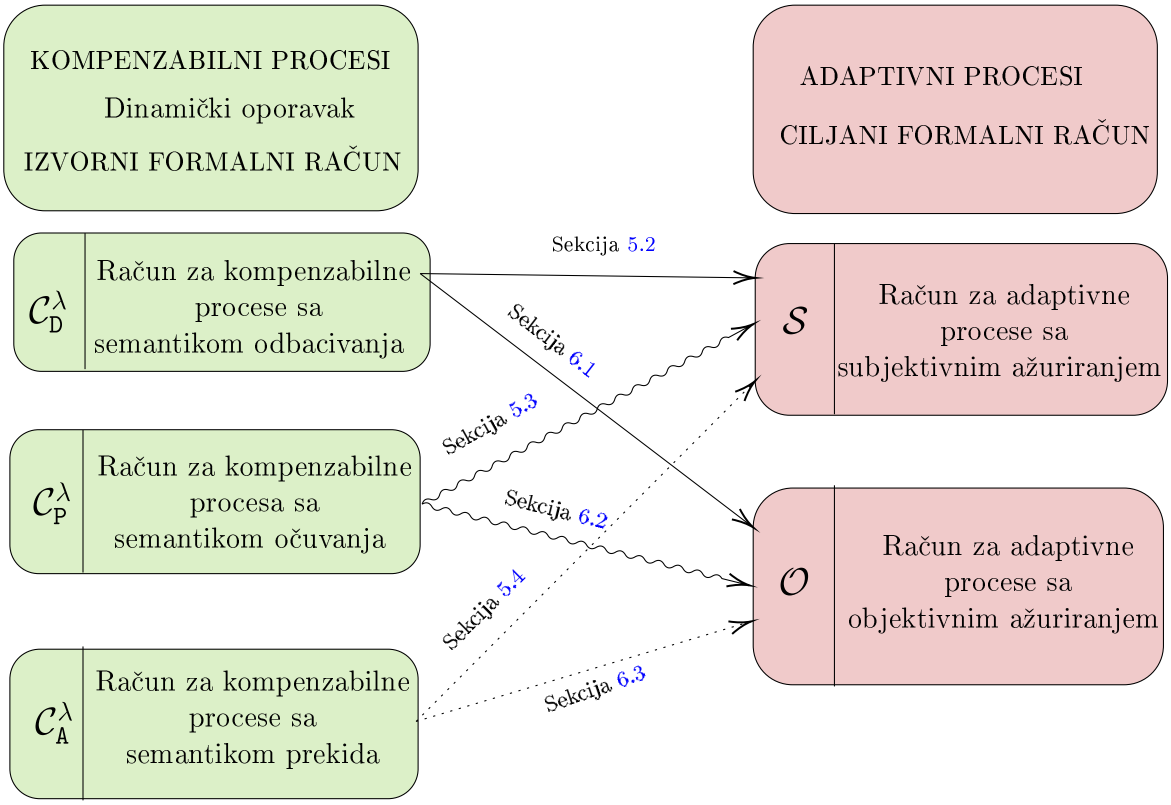
Cilj ove teze je da formalno poveže programske apstrakcije za rukovanje kompenzacijama (tipične za modele namenjene za usluge i dugoročne transakcije) i dinamičkog ažuriranja tokom izvršavanja. U disertaciji, porede se mehanizmi za rukovanje kompenzacijama i dinamičkim ažuriranjem u računima za konkurentne sisteme. Analizirala se relativna ekspresivnost pomenutih računa. Konkretno, razvijeno je ***dvanaest kodiranja***: šest procesnih računa za rukovanje kompenzacijama u dva računa za adaptivne procese. Pregled rezultata ilustrovan je pomoću Slike 2 i Slike 3.

Takođe, kodiranja računa sa kompenzacijama u račun sa adaptacijama zadovoljavaju (sve ili odabrane) dobro poznate kriterijume kodiranja, koje je predstavio Gorla u radu [22]:

(1) *kompozicionalnost*
(2) *invarijantnost kodiranja u odnosu na izbor imena*
} ***strukturni*** kriterijumi

(3) *operaciona korespodencija (kompletnost i valjanost)*
(4) *refleksija divergencije*
(5) *osetljivost na uspeh*
} ***semantički*** kriterijumi

Studije o ekspresivnosti procesnih računa imaju dugu istoriju i predstavljaju veoma aktivnu oblast istraživanja. Nedavni prikaz savremenih pristupa o formalnim poređenjima različitih procesnih računa predstavljen je u [44]. Analizirati kvalitet kodiranja i isključiti trivijalna ili besmislena kodiranja je zadatak kriterijuma za kodiranje ([45, 39, 37, 40, 22]). Kao što je već prethodno navedeno, za formalizaciju kodiranja u ovoj disertaciji koriste se kriterijumi koje je predstavio Gorla u [22]. Izabrani kriterijumi kvaliteta omogućiće da dobijeni rezultati kodiranja budu razumni i uporedivi. U odnosu na kriterijume iz [22], definicija *validnog kodiranja* (Definition 2.3.5) sadrži sledeće razlike:




Slika 3: Kodiranje $\mathcal{C}^{\lambda}_{\mathtt{D}}, \mathcal{C}^{\lambda}_{\mathtt{P}}, \mathcal{C}^{\lambda}_{\mathtt{A}}$ u $\mathcal{S}$ i $\mathcal{O}$. Strelica označava kodiranje.

(1) kako bi se uzele u obzir putanje $\rho$ u kojima se nalaze transakcije, razmatra se pojam kompozicionalnost koji je manje fleksibilan od Gorlinog,

(2) oslanjamo se na oblik operacione korespodencije – kompletnost koja, za razliku od Gorline, eksplicitno opisuje broj koraka u ciljanom formalnom računu potrebnih za oponašanje koraka u izvornom formalnom računu i

(3) razmatra se novi kriterijum kodiranja, nazvan *efikasnost*, koji omogućava da se precizno uporede kodiranja.

U vezi sa tačkom (3) treba napomenuti da se ne zna za radove koji koriste kriterijum koji je sličan kriterijumu efikasnosti. Najbližu povezanost možemo naći sa radom koji su predstavili Lanese, Vaz i Fereira [29] kao i sa radom Lanese i Zavataro [30]. Rad u [29] analizira ekspresivnu moć računa sa kompenzacijama fokusirajući se na tri različita mehanizma specifikacije za kompenzacije: statički oporavak, paralelni oporavak i dinamički oporavak. Autori pokazuju da se paralelni oporavak (gde se kompenzacija dinamički gradi kao paralelni sastav kompenzacionih elemenata) može kompoziciono kodirati pomoću statičkog oporavka. Takođe, autori u [29] pokazuju nemogućnost kodiranja dinamičkog oporavka pomoću statičkog oporavka. Rad u [30] predstavlja fundamentalne razlike između statičkog i dinamičkog oporavka: pokazano je da je prekid (tj. odsustvo beskonačne putanje računanja koja počinje od datog procesa) odlučivo svojstvo za procese sa statičkim oporavkom, ali neodlučivo za procese sa dinamičkim oporavkom.

Rezultati o ekspresivnosti, predstavljeni u disertaciji, dopunjuju rezultate predstavljene u [29, 30] implementacijom statičkog i dinamičkog oporavka u kompenzabilnim procesima koristeći različite okvire procesa definisane za adaptivne procese. U istom kontekstu, ali manje povezano, Vaz i Fereira [29] proučavaju kriterijume kada su kompenzabilni procesi ispravni i uspostavljaju da je „self-healing” kompenzacija korektna. Kriterijumi predstavljeni u [29] su različiti od dobro formiranih kompenzabilnih procesa koje smo mi razvili kako bismo formalizovali kodiranje, za koje je notifikacija greške od krucijalne važnosti.

Braveti i Zavataro [8] porede ekspresivnost računa koji predstavlja varijantu Milnerovog CCS-a proširenog sa operatorom prekida iz CSP: „obrada izuzetka” (eng. „try-catch”) operator za rukovanje izuzetkom, operatori replikacije i rekurzije. Njihovo poređenje zasnovano je na (ne)odlučivosti egzistencijalnih i univerzalnih problema prestanka: prvi se tiče postojanja jednog završnog računanja, dok se drugi pita da li se sva računanja završavaju. Autori dokazuju da u CCS-u sa replikacijom nema razlike između prekida i „obrade izuzetka”: univerzalni prekid je odlučiv, dok egzistencijalni nije. Nasuprot tome, u CCS-u sa rekurzijom i „obradom izuzetka”, univerzalni problem završetka postaje neodlučiv, otkrivajući tako jaz u ekspresivnosti u odnosu na jezik sa rekurzijom i prekidom.

U nastavku su predstavljeni konkretni doprinosi ove teze.

## 0.2 Doprinosi disertacije

Disertacija doprinosi teoriji konkurentnih sistema sa originalnim rezultatima o relativnoj ekspresivnosti procesnih računa koji poseduju mogućnost adaptacije i dinamičkog ažuriranja tokom izvršavanja. Rezultati predstavljeni u disertaciji su jedinstveni u literaturi. Takođe, predstavljeni rezultati produbljuju i poboljšavaju razumevanje teorije konkurentnih sistema u celosti.
Preciznije, glavna tema disertacije je upoređivanje procesnih računa za rukovanje kompenzacijama sa dinamičkim ažuriranjem, sa stanovišta *relativne ekspresivnosti.*

Postoje opravdani razlozi za proučavanje računa za rukovanje kompenzacijama, formalizovanom u [29] i za dinamičko ažuriranje, formalizovanom u [7]. S jedne strane, račun za kompenzacije u [29] je dovoljno ekspresivan da obuhvati nekoliko različitih jezika koji su predloženi u literaturi. Analize izražajnosti u [29] su prilično iscrpne i donose istovetnost u proučavanju formalnih modela za dugoročne transakcije. Zbog svoje izražajnosti, ovaj račun predstavlja odgovarajuću polaznu tačku za dalja istraživanja. S druge strane, račun za adaptivne procese predstavljen u [7] je jednostavan procesni model dinamičke adaptacije i rekonfiguracije, zasnovan na nekoliko procesnih terma i operacionoj semantici, koja podržava dve vrste ažuriranja: objektivno i subjektivno ažuriranje. Nasuprot tome, kao što ćemo prikazati, račun za kompenzacije oslanja se na zamršen označeni tranzicioni sistem. Kao takvi, adaptivni procesi obezbeđuju fleksibilan okvir koji koristimo za razjašnjavanje osnova mehanizama za rukovanje kompenzacijama, iz nove perspektive.
Preciznije opisan doprinos disertacije predstavljen je u nastavku:

(1) Pružamo objedinjenu, sveobuhvatnu prezentaciju dvanaest *preslikavanja* između računa za kompenzacije u račun za adaptivne procese, uzimajući u obzir objektivna i subjektivna ažuriranja. Pratimo i poboljšavamo rezultate o ekspresivnosti koje smo predstavili u [16] i [14], respektivno.

(2) Utvrđujemo ispravnost posmatranih dvanaest preslikavnja. Tačnije, utvrđujemo da su prevodi $\mathcal{C}_{\mathtt{D}}$ u $\mathcal{S}$ i $\mathcal{O}$ *validna kodiranja* — zadovoljavaju *kompozicionalnost, invarijantnost kodiranja u odnosu na izbor imena, operacionu korespondenciju, refleksiju divergencije* i *osetljivost na uspeh* – svojstva koja svedoče o robusnosti preslikavnja. Za prevođenja $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{A}}$, $\mathcal{C}^{\lambda}_{\mathtt{A}}$ u $\mathcal{S}$ i $\mathcal{O}$ utvrđujemo da zadovoljavaju osobine: *kompozicionalnost, invarijantnost kodiranja u odnosu na izbor imena* i *operacionu korespondenciju.* Analize ostalih kriterijuma ostavljene su za budući rad. Utvrđujemo i da prevodi $\mathcal{C}_{\mathtt{P}}$ i $\mathcal{C}^{\lambda}_{\mathtt{P}}$ u $\mathcal{S}$ i $\mathcal{O}$ zadovoljavaju kriterijume *invarijantnosti kodiranja u odnosu na izbor imena* i *operacionu korespondenciju,* dok su analize ostalih kriterijuma ostavljene za buduća istraživanja.

(3) Koristimo uvedenih **dvanaest preslikavanja** kako bismo jasno razlikovali *subjektivno* i *objektivno ažuriranje* u računima za konkurentne sisteme. Razmatra se novi kriterijum

kodiranja, nazvan *efikasnost*, koji omogućava da se kodiranja precizno uporede. Efikasnost se definiše apstraktno, uzimajući u obzir broj koraka redukcije koji su potrebni ciljnom jeziku da bi imitirali ponašanje izvornog jezika. U disertaciji je dokazano sa su subjektivna ažuriranja bolje prilagođena za kodiranje kompenzabilnih procesa jer ostvaruju bolje rezultate operacione korespondencije.

(4) Razvili smo klasu *dobro formiranih* kompenzabilnih procesa za formalizovanje kodiranja. Preciznije, ova klasa procesa onemogućava određene nedeterminističke interakcije koje uključuju ugnežđene transakcije i obaveštenja o greškama.

Tačku (3) je potrebno dodatno objasniti. Čvrsto verujemo da postoji opravdana potreba za konstruisanjem i dokazivanjem dvanaest preslikavanja. Glavni razlog je što tri različite semantike sa statičkom kompenzacijom: odbacivanje, očuvanje i prekidanje implementiraju različite nivoe zaštite. Intuitivno:

- kompenzabilni procesi sa semantikom odbacivanja vode računa samo o kompenzacionoj aktivnosti u transakciji i zašatićenom bloku,
- kompenzabilni procesi sa semantikom očuvanja pored zaštićenih blokova takođe štite i ugnežđene transakcije. Svi procesi koji nisu zatvoreni u zaštićenom bloku se odbacuju,
- kompenzabilni procesi sa semantikom prekida zadržavaju sve zaštićene blokove i kompenzacijske aktivnosti u podrazumevanoj aktivnosti, uključujući one u ugnežđenim transakcijama.

Kao ilustraciju posmatrajmo proces $P = t\big[t_1[P_1,Q_1] \mid t_2[\langle P_2\rangle,Q_2] \mid R \mid \langle P_3\rangle,Q_5\big]$. Dakle, u zavisnosti od izbora semantike dobijamo sledeće:

$$\begin{aligned}
\mathcal{C}_{\mathtt{D}}: \quad \bar{t} \mid P &\xrightarrow{\tau}_{\mathtt{D}} \langle P_3\rangle \mid \langle Q_5\rangle;\\
\mathcal{C}_{\mathtt{P}}: \quad \bar{t} \mid P &\xrightarrow{\tau}_{\mathtt{P}} \langle P_3\rangle \mid \langle Q_5\rangle \mid t_1[P_1,Q_1] \mid t_2[\langle P_2\rangle,Q_2];\\
\mathcal{C}_{\mathtt{A}}: \quad \bar{t} \mid P &\xrightarrow{\tau}_{\mathtt{A}} \langle P_3\rangle \mid \langle Q_5\rangle \mid \langle P_2\rangle \mid \langle Q_1\rangle \mid \langle Q_2\rangle.
\end{aligned}$$

U semantici *odbacivanja* sačuvani su samo zaštićeni blokovi na najvišem nivou. Stoga se vodi računa samo o kompenzacionoj aktivnosti za transakciju $t$ i zaštićenom bloku $\langle P_3\rangle$. Semantika očuvanja štiti i ugnežđene transakcije $t_1$ i $t_2$. Proces kao što je $R$, bez zaštićenog bloka, se odbacuje. Konačno, semantika prekida zadržava sve zaštićene blokove i aktivnosti kompenzacije u podrazumevanoj aktivnosti za $t$, uključujući i one u ugnežđenim transakcijama, kao što je proces $\langle P_2\rangle$.

Takođe, u disertaciji razmatramo kompenzabilne procese sa dinamičkim oporavkom. Glavna razlika u poređenju sa kompenzabilnim procesima sa statičkim oporavkom je u tome što proces $P$ iz transakcije $t[P,Q]$ može da ažurira kompenzacionu aktivnost $Q$. Ažuriranje kompenzacione aktivnosti zapravo vrši novi operator $\mathtt{inst}\lfloor \lambda Y.R\rfloor.P'$, gde je funkcija $\lambda Y.R$ ažuriranje kompenzacije ($Y$ se može pojaviti unutar $R$). Primena takvog ažuriranja kompenzacije na kompenzacionu aktivnost $Q$ proizvodi novu kompenzacionu aktivnost $R\{^{Q}\!/_{Y}\}$, nakon internog prelaska. Treba imati na umu da se proces $R$ možda neće pojaviti u nastaloj kompenzacionoj aktivnosti, a može se pojaviti i više puta. Na primer, $\lambda Y.\mathbf{0}$ briše trenutnu kompenzacionu aktivnost.

Na osnovu prethodno predstavljene intuicije za različite semantike, koristimo sledeći primer da još jednom ilustrujemo kompenzabilne procese sa semantikom odbacivanja.

**Primer 2.** Posmatrajmo jednostavan scenario rezervacije hotela, u kojem hotel i klijent stupaju u interakciju kako bi klijent rezervisao i platio sobu, a potom razmenjuju fakturu. Ovaj scenario može biti predstavljen korišćenjem kompenzabilnih procesa na sledeći način:

$$Reservation \stackrel{def}{=} Hotel \mid Client$$

$$
\begin{aligned}
Client &\stackrel{def}{=} \overline{book}.\overline{pay}.(invoice + \overline{t}.refund)\\
Hotel &\stackrel{def}{=} t[book.pay.\overline{invoice},\overline{refund}]
\end{aligned}
$$

Ponašanje hotela predstavljeno je kao transakcija $t$ koja omogućava klijentima da rezervišu sobu i da je plate. Ako je klijent zadovoljan rezervacijom, hotel će mu poslati račun. U suprotnom, klijent može otkazati transakciju; u tom slučaju hotel nudi klijentu povrat novca. Pretpostavimo da klijent odluči da otkaže svoju rezervaciju. Kao što ćemo videti, postoje četiri koraka prelaska za proces *Reservation*:

$$
\begin{aligned}
Reservation &\xrightarrow{\tau}_{\mathtt{D}} t[pay.\overline{invoice},\overline{refund}] \mid \overline{pay}.(invoice + \overline{t}.refund)\\
&\xrightarrow{\tau}_{\mathtt{D}} t[\overline{invoice},\overline{refund}] \mid invoice + \overline{t}.refund\\
&\xrightarrow{\tau}_{\mathtt{D}} \langle\overline{refund}\rangle \mid refund\\
&\xrightarrow{\tau}_{\mathtt{D}} \langle\mathbf{0}\rangle \mid \mathbf{0}.
\end{aligned}
$$

Kako bismo još jednom uputili čitaoca na sličnosti i razlike između kompenzabilnih i adaptivnih procesa, prethodni primer analiziraćemo i u kontekstu adaptacija.

**Primer 3.** Razmotrimo ponovo scenario rezervacije hotela iz Primera 2, ovaj put izražen koristeći račun za adaptivne procese:

$$
\begin{aligned}
Reservation &\stackrel{def}{=} Hotel \mid Client\\
Client &\stackrel{def}{=} \overline{book}.\overline{pay}.(\overline{t}.refund + invoice)\\
Hotel &\stackrel{def}{=} t[book.pay.\overline{invoice}] \mid t.t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle \mid p_t[\overline{refund}]
\end{aligned}
$$

Koristimo CCS procese sa lokacijama i (subjektivnim) prefiksima ažuriranja. Ponašanje klijenta uključuje slanje zahteva za rezervaciju i plaćanje sobe, nakon čega sledi prijem računa od strane hotela ili greška na $t$ koja označava kraj transakcije i zahtev za povrat novca. Očekivano ponašanje hotela nalazi se na lokaciji $t$: hotel omogućava klijentu da rezerviše sobu i plati je. Ako je klijent zadovoljan rezervacijom, hotel će mu/joj poslati račun. Specifikacija hotela uključuje:

(i) subjektivni prefiks ažuriranja $t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle$ (na isti način može se koristiti objektivno ažuriranje $t\{(Y).\mathbf{0}\}$), koji briše lokaciju $t$ sa njenim sadržajem u slučaju da klijent nije zadovoljan rezervacijom i

(ii) jednostavnu proceduru povrata novca koja se nalazi na lokaciji $p_t$, koja upravlja interakcijom sa klijentom u tom scenariju.

Ako klijent odluči da otkaže rezervaciju, koraci redukcije za proces *Reservation* bi bili sledeći:

$$
\begin{aligned}
Reservation &\longrightarrow t[pay.\overline{invoice}] \mid t.t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle \mid p_t[\overline{refund}] \mid \overline{pay}.(\overline{t}.refund + invoice)\\
&\longrightarrow t[\overline{invoice}] \mid t.t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle \mid p_t[\overline{refund}] \mid \overline{t}.refund + invoice\\
&\longrightarrow t[\overline{invoice}] \mid t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle \mid p_t[\overline{refund}] \mid refund\\
&\longrightarrow p_t[\overline{refund}] \mid refund\\
&\longrightarrow p_t[\mathbf{0}].
\end{aligned}
$$

U ovom primeru mogli smo da upotrebimo i objektivno ažuriranje $t\{(Y).\mathbf{0}\}$ umesto subjektivnog ažuriranja $t\langle\!\langle(Y).\mathbf{0}\rangle\!\rangle$. Upotrebom objektivnog ažuriranja, ponašanje procesa *Reservation* je veoma slično prikazanom.

Razvili smo kodiranje za različite semantike kompenzabilnih procesa u adaptibilne procese u dva slučaja, tj. za subjektivno i objektivno ažuriranje. Kodiranje u adaptibilne procese sa objektivnim ažuriranjima otkriva izvesna ograničenja: predstavljajući „relokaciju” zaštićenih blokova raspoređenih unutar ugnežđenih transakcija, objektivna ažuriranja ostavljaju procese na „pogrešnoj" lokaciji. Ova situacija podseća na razlike prikazane u Primeru 1. Da bi se to ispravilo, kodiranje koristi dodatne sinhronizacije za dovođenje procesa na odgovarajuće lokacije. Ovo se značajno odražava na *cenu* oponašanja koraka izračunavanja izvornog računa, mereno brojem povezanih koraka računanja ciljanog računa (koji su navedeni u tvrđenjima o operacionoj korespodenciji). Kodiranje u račun sa subjektivnim ažuriranjima nema ovo ograničenje, pa je u skladu sa tim *efikasnije* od kodiranja koje koristi objektivno ažuriranje.

Rezultati kodiranja pokazali su da je kodiranje kompenzabilnim procesa sa semantikom prekida (sa statičkim i dinamičkim oporavkom) u adaptibilne procese (sa subjektivnim i objektivnim ažuriranjem) najsloženije. Ovakvi dobijeni rezultati su očekivani, jer je semantika prekida pokazala da ima najviši nivo zaštite.

## 0.3 Publikacije i struktura disertacije

**Publikacije** Disertacija objedinjuje, revidira i dopunjuje rezultate iz rada predstavljenog na međunarodnoj konferenciji:

- J. Dedeić, J. Pantović, and J. A. Pérez. On compensation primitives as adaptable processes. In S. Crafa and D. Gebler, editors, Proceedings of the Combined 22nd International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, and 12th Workshop on Structural Operational Semantics, EXPRESS/SOS 2015, Madrid, Spain, 31st August 2015., volume 190 of EPTCS, pages 16 – 30, 2015 ([16]).

U ovom radu predstavljamo translacije iz računa za kompenzabilne procese u račun za adaptibilne procese sa *objektivnim ažuriranjem*. Nastavak istraživačkog rada inspirisan je istom temom. Kao rezultat daljeg istraživanja objavljen je sledeći rad:

- J. Dedeić, J. Pantović, and J. A. Pérez. Efficient compensation handling via subjective updates. In Proceedings of the Symposium on Applied Computing, SAC'17, pages 51 – 58, Marrakesh, Morocco, April 3 – 7, 2017. ACM ([14]).

U ovom radu predstavili smo translacije iz računa za kompenzabilne procese u račun za adaptivne procese sa *subjektivnim ažuriranjem* i uporedili ih sa rezultatima iz [16]. Izveli smo zaključak da su translacije/kodiranja sa subjektivnim ažuriranjem bolja, u smislu efikasnosti, od kodiranja sa objektivnim ažuriranjima, jer produkuju bolje rezultate operacione korespondencije.

Istraživanje smo nastavili, jer smo uočili da prethodni rezultati koje smo objavili zahtevaju dodatnu analizu, verifikaciju i proširenja. Novodobijeni rezultati objavljeni su u:

- J. Dedeić, J. Pantović, and J. A. Pérez. On primitives for compensation handling as adaptable processes. Journal of Logical and Algebraic Methods in Programming, page 100675, 2021 ([15]).

Ovaj rad prikuplja i poboljšava preliminarne rezultate iz radova [16] i [14]. Dok smo u [16] proučavali kodiranje u adaptibilne procese sa objektivnim ažuriranjem, u [14] smo proučavali kodiranje u adaptibilne procese sa subjektivnim ažuriranjem i upoređivali ih sa rezultatima iz [16]. Glavna razlika između radova [16, 14] i rada koji je objavljen u časopisu ([15]) je u tome što se u konferencijskim radovima koncentrišemo na određeni izvorni račun, naime na račun u [29] sa *statičkim oporavkom* i *semantikom odbacivanja* (Figure 2, kodiranje $\mathcal{C}_{\mathsf{D}}$ u $\mathcal{S}$ i $\mathcal{O}$). Istraživanje u radovima [16, 14] uzima u obzir i izvorne račune sa *dinamičkim oporavkom* i/ili semantikom *očuvanja* i *prekida*. Račun sa statičkim oporavkom i semantikom odbacivanja nedvosmisleno definiše najjednostavnije podešavanje za oba kodiranja, u kojem se ključne razlike između kompenzacionih i adaptibilnih procesa mogu preciznije predstaviti. Takođe, fokus na semantici odbacivanja nam omogućava sažetu prezentaciju rezultata. U radu iz časopisa proširujemo analizu (efikasnog) kodiranja izvornih računa sa semantikom koju smo razmatrali u [16] i [14].

U nastavku navodimo ključna proširenja rezultata koji su razvijeni u radu iz časopisa:

(1) Razvijamo klasu *dobro formiranih* kompenzabilnih procesa za formalizovanje kodiranja, za koje su obaveštenja o greškama ključna. Preciznije, ova klasa procesa onemogućava određene nedeterminističke interakcije koje uključuju ugnežđene transakcije i obaveštenja o greškama.

(2) Proširujemo kriterijume uključene u definiciju validnog kodiranja. Dodati su sledeći kriterijumi: *invarijantnost kodiranja u odnosu na izbor imena*, *refleksija divergencije* i *osetljivost na uspeh*. Stoga smo proširili rad sa dodatnim definicijama i teoremama koje imaju za cilj da verifikuju da kodiranje zadovoljava novouvedene kriterijume.

(3) Razvijamo dodatne definicije i teoreme potrebne za dokaze operacione korespodencije (potpunost i valjanost).

**Struktura disertacije** Disertacija je organizovana u sedam poglavlja.

**Prvo poglavlje** prikazuje osnovne motive za razvoj translacije procesnog računa za kompenzacije predstavljenog u [29] u procesni račun sa adaptacijama predstavljenog u [7]. Takođe, ovo poglavlje daje pregled literature koja je povezana sa temom istraživanja predstavljenoj u disertaciji.

**Drugo poglavlje** pruža pregled teorijskih osnova za disertaciju, uvodi osnovnu terminologiju i pojmove koji se koriste u disertaciji. Takođe, pruža fundamentalni pregled računa za kompenzabilne procese i računa za adaptivne procese. Najpre na neformalan način, kroz primere, uvodimo procesne račune, a zatim sledi njihov formalan prikaz kroz detaljnu analizu sintakse i operacione semantike. Ovo poglavlje sadrži i definiciju dobro formiranih kompenzabilnih procesa, klase procesa koja onemogućava određene nedeterminističke interakcije. Dobro formirani kompenzabilni procesi predstavljaju, između ostalog, originali naučni doprinos ove disertacije. Pored navedenog, drugo poglavlje predstavlja glavna pitanja analize ekspresivnosti procesa u konkurentnim sistemima i daje pregled literature o studijama ekspresivnosti i tehnikama koje su korišćene.

**Treće poglavlje** ima za glavnu temu, da predstavi translaciju/kodiranje računa za kompenzabilne procese u račun za adaptivne procese sa subjektivnim ažuriranjem. Dakle, uvodimo osnovne pojmove i oznake, zatim predstavljamo kodiranja $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ i $\mathcal{C}_{\mathtt{A}}$ u $\mathcal{S}$. Dokazujemo da kodiranja zadovoljavaju sve ili određene osobine definisane za *validno kodiranje*.

**Četvrto poglavlje** predstavlja translaciju/kodiranje računa za kompenzabilne procese u račun za adaptivne procese sa objektivnim ažuriranjem. Dakle, uvodimo osnovne pojmove i oznake, zatim predstavljamo kodiranja $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ i $\mathcal{C}_{\mathtt{A}}$ u $\mathcal{O}$. Dokazujemo da kodiranja zadovoljavaju sve ili određene osobine definisane za *validno* kodiranje. Takođe, u ovom poglavlju bavimo se pitanjem efikasnosti kodiranja. S obzirom da su u disertaciji razvijene dve vrste kodiranja kompenzabilnih procesa u adaptibilne procese, u smislu: kodiranje sa subjektivnim i objektivnim ažuriranjem, u ovom poglavlju upoređujemo njihovu *efikasnost*. Efikasnost se definiše apstraktno, uzimajući u obzir broj koraka redukcije koji su potrebni ciljnom jeziku da bi imitirali ponašanje izvornog jezika. Preciznije, dokazujemo da su kodiranja koja koriste subjektivno ažuriranje efikasnija od kodiranja koja koriste objektivno ažuriranje.

**Peto poglavlje** proširuje sintaksu za kompenzabilne procese za dinamičko ažuriranje kompenzacione aktivnosti. Takođe, diskutuju se odgovarajuća proširenja dobro formiranih kompenzabilnih procesa. Međutim, glavna tema ovog poglavlja je predstavljanje prevođenja kompenzabilnih procesa sa dinamičkim oporavkom, $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}^{\lambda}_{\mathtt{P}}$ i $\mathcal{C}^{\lambda}_{\mathtt{A}}$ u adaptivne procese sa subjektivnim ažuriranjem, $\mathcal{S}$. Zatim se dokazuje da definisana kodiranja zadovoljavaju odabrane osobine iz definicije *validnog* kodiranja.

**Šesto poglavlje** predstavlja translacije/kodiranja kompenzabilnih procesa sa dinamičkim oporavkom, $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}^{\lambda}_{\mathtt{P}}$ i $\mathcal{C}^{\lambda}_{\mathtt{A}}$ u adaptivne procese sa objektivnim ažuriranjem, $\mathcal{O}$. Nakon definisanja kodiranja sledi postupak dokazivanja da kodiranja zadovoljavaju odabrane osobine iz definicije *validnog* kodiranja. Ovo poglavlje se takođe bavi pitanjem efikasnosti kodiranja kompenzabilnih procesa sa dinamičkim ažuriranjem. Tačnije, ponovo je dokazano da su kodiranja koja koriste subjektivno ažuriranje efikasnija od kodiranja koja koriste objektivno ažuriranje.

**Sedmo poglavlje** sadrži zaključak disertacije. Takođe, prikazuje diskusiju o aktuelnim i budućim pravcima istraživanja kandidata.

# Abstract

Identifying uniform and rigorous ways of comparing different models of computation from the point of view of their expressiveness is a longstanding and important research theme in concurrency theory. In the case of process calculi, these comparisons aim at clarifying to what extent the process constructs in already existing calculi relate to each other. We see this as an essential prerequisite step towards the definition of sensible, widely applicable programming abstractions. This dissertation contributes to foundational studies of the relative expressiveness of process calculi. We concentrate on calculi with constructs for compensation handling and dynamic update, which are increasingly relevant in the rigorous specification of reliable computing systems. Compensations and updates are intuitively similar: both specify how the behavior of a computing system changes at runtime in response to an exceptional event. Process calculi with these constructs, however, are technically quite different. We study the relative expressiveness of these calculi by developing encodings: language translations that enjoy precise correctness properties, which bear witness to the quality of the translation. Encodings can be seen as formal compilers that correctly translate process terms from a source language (in our case, a calculus with compensation handling) into a target language (in our case, a calculus of adaptable processes, which implements dynamic update). We consider two different target languages, which account for complementary forms of process mobility: the first consists of adaptable processes with subjective updates, in which, intuitively, a process reconfigures itself; the second target language considers objective updates, in which a process is reconfigured by another process in its context. Our main technical contributions are encodings that preserve well-known correctness properties, namely compositionality, name invariance, operational correspondence, divergence reflection and success sensitiveness. Our encodings not only represent a non-trivial application of process mobility as present in adaptable processes; they also shed light on the intricate semantics of processes with compensation.
**Key Words:** concurrency, semantics of programming languages, process calculi, compensation handling, dynamic update, expressiveness.

# Contents

# CHAPTER 1

# Introduction

## 1.1 Motivation

Recent information technology (IT) advances: cloud computing, service-oriented computing, artificial intelligence, data analytics, monitoring and predicting, etc., are supported by large computing infrastructures such as data-centers. Also, IT frequently uses wireless and mobile networking, parallel and distributed systems. A distributed system is one in which the components are spread across multiple networked computers, which communicate and coordinate their actions by sending messages to one another. Therefore, distributed systems are more convenient than sequential systems for solving a problem. In sequential computing, each step is carried out one at a time, while in distributed computing, a problem is distributed across multiple computing devices. Sequential computing has drawbacks: takes a long time and is very expensive in programs with large number of steps. Nonetheless, when modeling and implementing correct systems, distributed systems impose greater obstacles such as: unreliable network, slow processes, unexpected load, etc. In all software systems, *communication* and *interaction* have become central features. Also, analysis and verification of software are complex and challenging assignments since there is a general assumption that software systems have to operate indefinitely and without unexpected termination. Over the last few decades, infrastructures such as those supporting high-performance computing have grown in scale and complexity. Their power, flexibility, and convenience go hand-by-hand with the need for efficient energy consumption. Large systems may experience a variety of faults or errors with increasing frequency, and mechanisms/techniques for overcoming them are of crucial importance. *Formal methods* are techniques suitable for the formal specification and verification of complex (software and hardware) systems based on mathematics and logic. There are plenty of formal methods for improving software development that runs on large computing infrastructures. Formal methods have the primary task of anticipating the possibility of errors in applications and ensure a timely response. They should also avoid unnecessary waste of resources (such as energy, for example).

Building on what has been said before, many software applications are based on *long-running transactions* (henceforth LRTs) as a fundamental building block. Frequently found in service–oriented systems, [17], LRTs describe time-extensive activities that involve several distributed components, and loosely coupled resources. In computer science, ACID (atomicity, consistency, isolation, and durability) is a set of attributes for transactions that ensures their validity even in the face of errors, power outages, and other problems. LRTs do not satisfy isolation since the execution of a single LRT is not intended to block the whole system, i.e., due to the nature of these systems and the time duration of the activities, it is not feasible to lock (non-local) resources. For LRTs management, handling failures is one sensitive aspect: mechanisms for detecting failures and bringing the LRT back to a consistent state need to be explicitly programmed. As designing and certifying the correctness of such mechanisms are error prone, specialized constructs, such as *exceptions* and *compensations*, have been put forward to offer direct programming support. The literature offers a variety of constructs. In Java, for instance, we find the construct try $P$ catch $e$ $Q$, where $Q$ is in charge of managing *exceptions* $e$ raised inside $P$; in WS-BPEL [2] we find advanced mechanisms exploiting fault, termination, and compensation handlers to handle errors.

We are interested in researching programming constructs that support failure handling at the heart of mechanisms that detect failures and bring the system back to a consistent state. As their name suggests, *compensation mechanisms* are meant to compensate the fact that an LRT has received a failure signal. Upon receiving a failure signal, compensation mechanisms are expected to install and activate alternative behaviors for recovering system consistency. Such compensation behavior may be different from the LRT's initial behavior. Widely studied in service-oriented settings, forms of compensation handling also find an application in Collective Adaptive Systems (at least conceptually), especially as self-autonomous devices begin to be used in traditional transactional activities, such as distribution and delivery — consider, e.g., Amazon's Prime Air and DHL's Parcelcopter.

A variety of calculi for concurrency with constructs for compensation handling has been proposed (see, e.g., [5, 10, 31, 11, 17]). Building upon process calculi such as CCS [32], CSP [27], and the $\pi$-calculus [33], they capture different forms of error recovery and offer reasoning techniques (e.g., behavioral equivalences) on communicating processes with compensation constructs. The relationships between the different proposals are not clear, and there has been work aimed to formally compare the expressiveness of the proposed mechanisms. The expressive power of such proposals has also been studied [11, 8, 29, 30]. Lanese et al. [29] address this question by developing a formal comparison of different approaches to LRTs in a concurrent and mobile setting. They consider a process language on top of which different primitives for error handling are uniformly considered.

More in detail, Lanese et al. [29] define a core calculus of compensable processes, which extends the $\pi$-calculus with *transactions* $t[P,Q]$ (where processes $P$ and $Q$ represent default and compensation activities, respectively), *protected blocks* $\langle Q \rangle$, and *compensation updates* $\mathtt{inst}\lfloor \lambda X.Q \rfloor.P$, which reconfigure a compensation activity. To this end, compensations may admit *static* or *dynamic* recovery (depending on whether compensation updates are allowed) and the response to failures can be specified via *preserving*, *discarding*, and *aborting* semantics. The language in [29] thus leads to six distinct calculi with compensation primitives.

Related to compensation handling, but on a somewhat different vein, a process calculus of *adaptable processes* was proposed to specify the *dynamic update* in communicating systems [7]. Adaptable processes specify forms of dynamic reconfiguration that are triggered by exceptional events, not necessarily catastrophic. A simple example is the reconfiguration of specific units of a robot swarm, which is usually hard to predict and entails modifying the device's behavior; still, it is certainly not a failure. Adaptable processes can be deployed in *locations*, which serve as delimiters for dynamic updates. A process $P$ *located at* $l$, denoted $l[P]$, can be reconfigured by an *update prefix* $l\{(X).Q\}.R$, where $Q$ denotes an adaptation routine for $l$, parameterized by variable $X$. With these two constructs, dynamic update is realized by the following reduction rule, in which $C_1$ and $C_2$ denote contexts of arbitrarily nested locations:

$$C_1\big[l[P]\big] \mid C_2\big[l\{(X).Q\}.R\big] \longrightarrow C_1\big[Q\{^{P}/X\}\big] \mid C_2\big[R\big] \qquad (1.1)$$

We call this an ***objective update***: a located process is reconfigured in its own context by an update prefix at a different context. Indeed, the update prefix $l\{(X).Q\}$ interacts with update process $l[P]$ and *moves* process $Q$ from $C_2$ to $C_1$, such that the reconfigured behavior $Q\{^{P}/X\}$ is left in $C_1$. After the located process $l[P]$ synchronizes with the appropriate update prefix, the location name $l$ is deleted. As expected, $X$ may occur zero or many times in $Q$; if $Q$ does not contain $X$ then the process $P$ will be erased as a result of the update. This way, dynamic update is then a form of process mobility, implemented using *higher-order process communication* as found in languages such as, e.g., the higher-order $\pi$-calculus [51], the Kell calculus [52], and Homer [25].

An alternative to objective update is ***subjective update***, in which process reconfiguration flows in the opposite direction: it is the located process that moves its process (e.g., $P$) to a (remote) context with an update prefix:

$$C_1\big[l[P] \mid R_1\big] \mid C_2\big[l\{(X).Q\}.R\big] \longrightarrow C_1\big[\mathbf{0} \mid R_1\big] \mid C_2\big[Q\{^{P}\!/X\} \mid R\big] \tag{1.2}$$

Same as an objective update, subjective update relies on process mobility; however, the direction of movement is different: above, process $P$ moves from $C_1$ to $C_2$, and the reconfigured behavior $Q\{^{P}\!/X\}$ is left in $C_2$, not in $C_1$. Thus, in an objective update, the located process "reconfigures itself", which makes for a more autonomous semantics for adaptation than subjective updates.[1]

**Example 1.1.1.** We contrast objective and subjective update by means of an example, adapted from [7]. Consider an *interrupt* operator that starts executing process $P$ but may abandon its execution to execute $Q$ instead; once $Q$ emits a termination signal $t_Q$, the operator returns to execute what is left of $P$. Using adaptable processes, this kind of behavior can be expressed as follows:

$$Sys = l_1\big[l[P] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big]$$

where $l$, $l_1$, and $l_2$ are different locations and name $t_Q$ is only known to $Q$. Process $Q$ does not contain process variable $X$. If $P$ evolves into $P'$ right before being interrupted, under a semantics with *objective update* we have

$$\begin{aligned} Sys &\longrightarrow^* l_1\big[l[P'] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big] \\ &\longrightarrow\ l_1\big[Q \mid t_Q.P' \mid R_1\big] \mid l_2\big[R_2\big] \\ &\longrightarrow^* l_1\big[P' \mid R_1\big] \mid l_2\big[R_2\big] \end{aligned}$$

This way, $P$ and its derivative $P'$ reside at location $l_1$. Notice that executing $Sys$ under a semantics with subjective update would yield a different behavior, because $P'$ (and $Q$) would be moved to $l_2$:

$$\begin{aligned} Sys &\longrightarrow^* l_1\big[l[P'] \mid R_1\big] \mid l_2\big[l\{(X).Q \mid t_Q.X\}.R_2\big] \\ &\longrightarrow\ l_1\big[R_1\big] \mid l_2\big[Q \mid t_Q.P' \mid R_2\big] \\ &\longrightarrow^* l_1\big[R_1\big] \mid l_2\big[P' \mid R_2\big] \end{aligned}$$

This shows that to achieve the intended interrupt behavior in a subjective setting, $Sys$ should be modified in order to eventually bring process $P'$ back to $l_1$. The following variation of $Sys$ achieves this:

$$Sys' = l_1\big[l[P] \mid l'\{(X).X\}.R_1\big] \mid l_2\big[l\{(X).l'[Q \mid t_Q.X]\}.R_2\big]$$

where we use $l'$ as an auxiliary location that "pulls back" $P'$ from $l_2$ into $l_1$.

Based on the previous overview of compensable processes and adaptable processes, it is important to keep in mind that compensations and updates are intuitively similar. The similarity is that both specify how the behavior of a concurrent system changes at runtime in response to an exceptional event. On the other side, these calculi are technically very different.

The context of this thesis is to formal connect programming abstractions for *compensation handling* (typical of models for services and LRTs) and for *runtime adaptation*. In Figure 1.1, we present notations, which will be used in the thesis, to denote different calculus of compensable processes and different calculus of adaptable processes.

[1] We use adjectives 'subjective' and 'objective' for updates following the distinction between subjective and objective *mobility*, as in calculi such as Ambients [12] and Seal [13]. As explained in [13], Ambients use subjective mobility (an agent moves itself), while Seal uses objective mobility (an agent is moved by its context).

| Compensable processes (Static recovery) – $\mathcal{C}$ | Notation |
|---|---|
| Calculus of compensable processes with discarding semantics | $\mathcal{C}_{\mathtt{D}}$ |
| Calculus of compensable processes with preserving semantics | $\mathcal{C}_{\mathtt{P}}$ |
| Calculus of compensable processes with aborting semantics | $\mathcal{C}_{\mathtt{A}}$ |
| **Compensable processes (Dynamic recovery) – $\mathcal{C}^{\lambda}$** | **Notation** |
| Calculus of compensable processes with discarding semantics | $\mathcal{C}^{\lambda}_{\mathtt{D}}$ |
| Calculus of compensable processes with preserving semantics | $\mathcal{C}^{\lambda}_{\mathtt{P}}$ |
| Calculus of compensable processes with aborting semantics | $\mathcal{C}^{\lambda}_{\mathtt{A}}$ |
| **Adaptable processes** | **Notation** |
| Calculus of adaptable processes with subjective update | $\mathcal{S}$ |
| Calculus of adaptable processes with objective update | $\mathcal{O}$ |

Figure 1.1: Notation of process calculus.

In particular, we compare in a systematic way mechanism for *compensation handling* and *dynamic update* in calculi for concurrency. We have analyzed the *relative expressiveness* of these calculi. More concrete, we develop ***twelve encodings*** of ***six*** process calculi with compensation handling into ***two*** calculi of adaptable processes. These results are illustrated in Figure 1.2 and Figure 1.3. The encodings preserve all or some of the following well-known correctness properties, namely: *compositionality, name invariance, operational correspondence, divergence reflection* and *success sensitiveness*. The encodings not only represent a non-trivial application of two sensible types of mobility for adaptable processes, they also provide new insight into the complex semantics of compensable processes.

Studies on the expressiveness of process calculi have a long history and constitute a vibrant research area. A recent account of modern approaches to formal comparisons between different process calculi is presented in [44]. To analyze the quality of encodings and to rule out trivial or meaningless encodings, they are augmented with encodability criteria ([45, 39, 37, 40, 22]). In this dissertation, we have followed Gorla's framework for formalizing encodability and separation results [22]. With respect to the criteria in [22], our definition of *valid encoding* (cf. Definition 2.3.5) presents the following differences:

- first, to account for the paths $\rho$ in which transactions reside, we consider a notion of compositionality that is slightly less flexible than Gorla's. More precisely, we consider compositional contexts that depend on an arbitrary list $\rho$ of external transaction names. Nevertheless, encoding still preserves the main principles of the notion of compositionality. We can translate compensable terms by translating their operator without need to analyze the structure of the subterms,

- second, we rely on a form of operational completeness that, unlike Gorla's, explicitly describes the number of steps required to mimic a step in the source language, and

- finally, we consider a new criterion, called *efficiency*, which allows us to precisely compare encodings (Definition 4.1.7).

We do not know of prior works using criteria similar to efficiency. The efficiency clarifies fundamental differences between subjective and objective updates. Since subjective updates induce

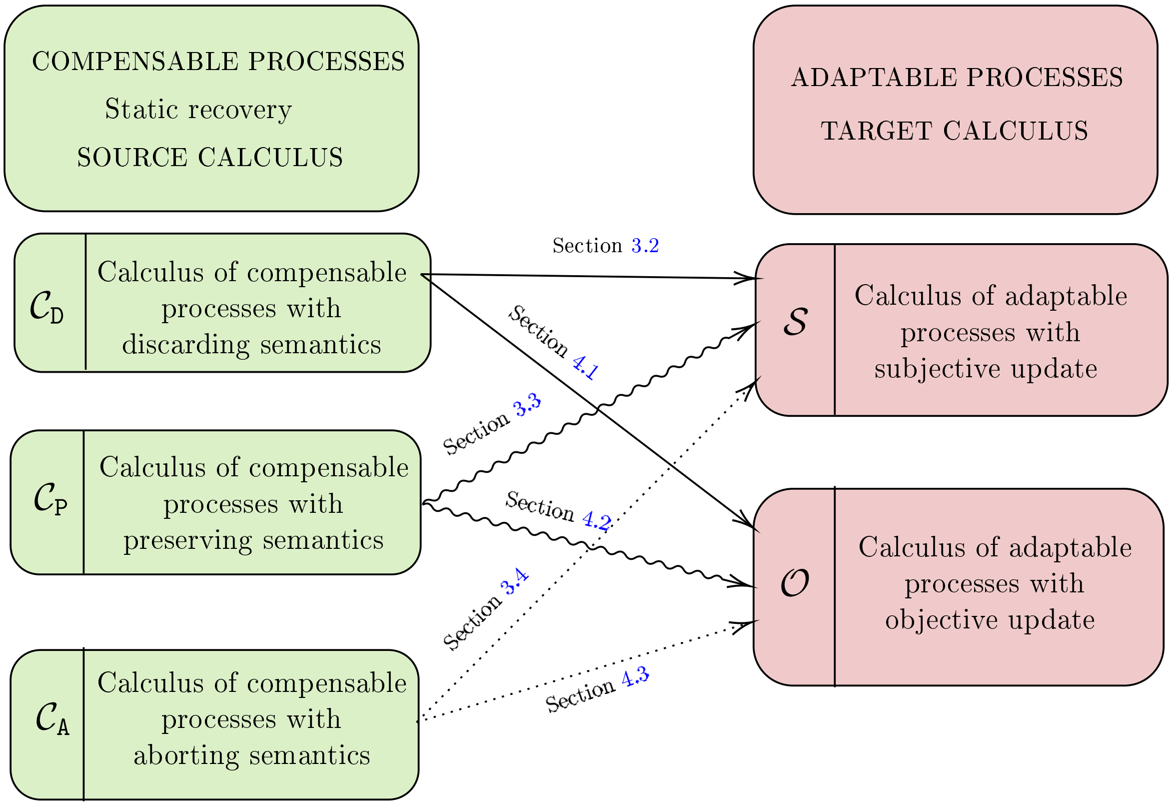



Figure 1.2: Encoding $\mathcal{C}_{\mathtt{D}}, \mathcal{C}_{\mathtt{P}}, \mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$. An arrow indicates encoding.

tighter operational correspondences, we can formally declare that subjective updates are more suited to encode compensation handling than objective updates. The closest related works are by Lanese, Vaz, and Ferreira [29] and by Lanese and Zavattaro [30]. The work in [29] analyzes the expressive power of the compensation calculus focusing on three different specification mechanisms for compensations: static recovery, parallel recovery, and dynamic recovery. The authors show that parallel recovery (where the compensation is dynamically built as the parallel composition of compensation elements) can be compositionally encoded using static recovery; they also show the impossibility of encoding dynamic recovery using static recovery. The work in [30] sheds further light on the fundamental differences between static and dynamic recovery: it is shown that termination (i.e., the absence of an infinite computation path starting from a given process) is a decidable property for processes with static recovery but undecidable for processes with dynamic recovery.

Our expressiveness results complement the findings in [29, 30] by implementing static and dynamic recovery in compensable processes using the different process framework defined by adaptable processes. In the same line, although slightly less related, Vaz and Ferreira [54] study criteria for determining when a compensable process is *correct* and establish that self-healing compensations are correct. The criteria in [54] are different from the notion of well-formed compensable processes that we developed to formalize encodings, for which error notifications are crucial.

Bravetti and Zavattaro [8] compare the expressiveness of variants of Milner's CCS extended with the interrupt operator of CSP, the try-catch operator for exception handling, and operators for replication and recursion. Their comparison is based on the (un)decidability of existential and universal termination problems: the former concerns the existence of one terminating computation, whereas the latter asks whether all computations terminate. They prove that in CCS with replication there is no difference between interrupt and try-catch: universal termination is

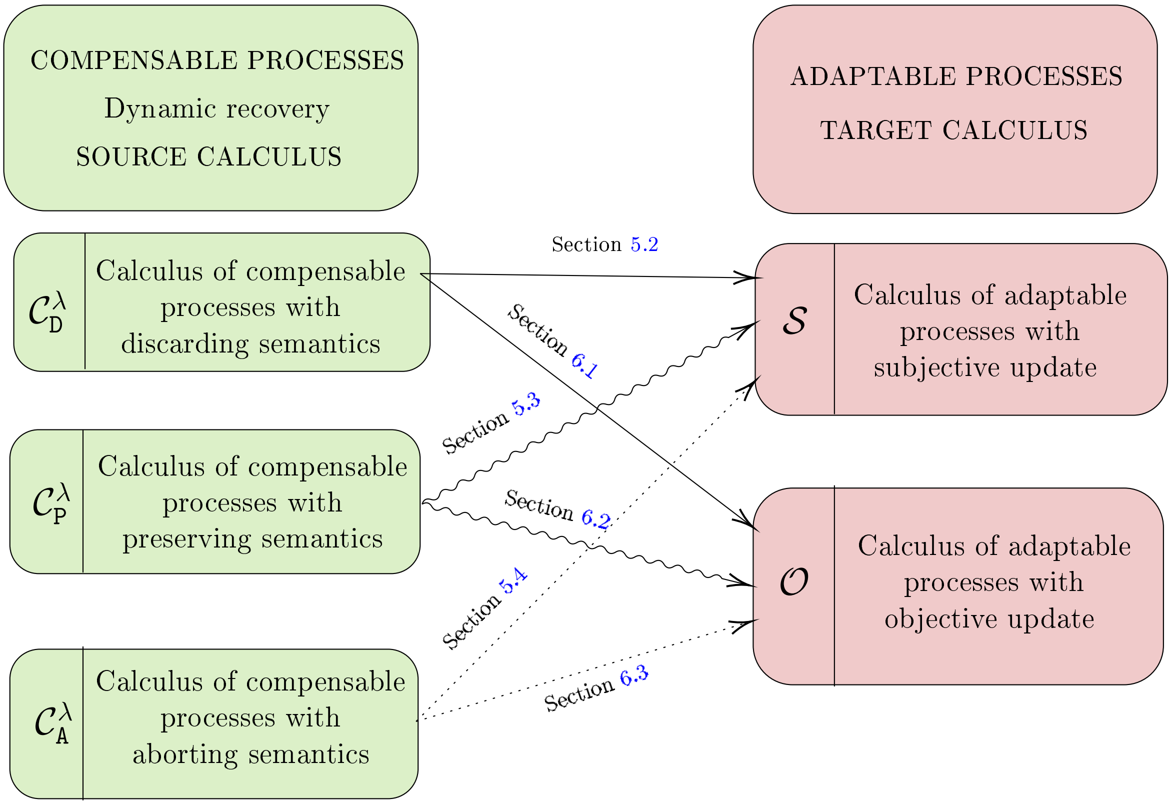



Figure 1.3: Encoding $\mathcal{C}^{\lambda}_{\mathtt{D}}, \mathcal{C}^{\lambda}_{\mathtt{P}}, \mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$. An arrow indicates encoding.

decidable while existential termination is not. In contrast, in CCS with recursion and try-catch, the universal termination problem becomes undecidable, thus revealing an expressiveness gap with respect to the language with recursion and interrupt.

In the following section, we present the original contribution of this thesis.

## 1.2 Contributions

The dissertation presents novel and unique results on the expressiveness for process calculi with dynamic update and runtime adaptation. As a result, this dissertation contributes to the theory of concurrency. More precisely, the purpose of this thesis is to compare process calculi with compensation handling (as formalized in [29]) and with dynamic update (as formalized in [7]), from the point of view of *relative expressiveness*.

There are good reasons for focusing on compensation handling as formalized in [29] and on dynamic update as formalized in [7]. On the one hand, the calculus of compensable processes in [29] is expressive enough to capture several different languages proposed in the literature; the analyses of expressiveness in [29] are rather exhaustive, and bring uniformity to the study of formal models for LRTs. Because of its expressiveness, this calculus provides an appropriate starting point for further investigations. On the other hand the calculus of adaptable processes in [7] is a simple process model of dynamic adaptation and reconfiguration, based on a few process constructs and endowed with operational (reduction) semantics, which can support both objective and subjective updates. In contrast, as we will see, the calculus of compensable processes relies on an intricate labeled transition system. As such, adaptable processes provide a flexible framework to elucidate the underpinnings of compensation handling, from a fresh perspective.

We present our contributions as follows:

(1) We provide a unified, comprehensive presentation of twelve *processes translations* between the calculus of compensation handling into the calculus of adaptable processes, considering both objective and subjective updates. We follow and refine the expressiveness results in [16] and [14], respectively.

(2) We establish the correctness of the twelve process calculus translations. Precisely, we prove that translations of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ and $\mathcal{O}$ are *valid encodings* — they satisfy *compositionality, name invariance, operational correspondence, divergence reflection* and *success sensitiveness* properties that bear witness to the robustness of translations. For translations of $\mathcal{C}_{\mathtt{D}}^{\lambda}$, $\mathcal{C}_{\mathtt{A}}$, $\mathcal{C}_{\mathtt{A}}^{\lambda}$ into $\mathcal{S}$ and $\mathcal{O}$ we prove that they satisfy: *compositionality, name invariance* and *operational correspondence.* We establish that translations of $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{S}$ and $\mathcal{O}$ satisfy *name invariance* and *operational correspondence*, the analysis of the other criteria are left for future work.

(3) We exploit our **twelve translations** to clearly distinguish between *subjective* and *objective updates* in calculi for concurrency. We state it in terms of efficiency: subjective updates are better suited to encode compensation handling because they induce solid results of operational correspondence.

Point (3) deserves further explanations. We strongly believe that there is a justified need to construct and prove twelve translations. The main reason is that the three different semantics with static compensation: discarding, preserving, and aborting implement different levels of protection. Intuitively, the discarding semantics only concerns the compensation activity for the transaction and the protected block. The preserving semantics also protects the nested transactions, all processes without an enclosing protected block are discarded. Finally, the aborting semantics preserves all protected blocks and compensation activities in the default activity, including those in nested transactions. We also consider dynamic compensations where using compensation updates one may specify an update for the compensation behavior in default activity.

We developed encodings for all different semantics of compensable into adaptable processes in two cases, i.e., for subjective and objective update. The encoding into adaptable processes with objective updates reveals a limitation: in the representation of the "recollection" of protected blocks scattered within nested transactions, objective updates leave behind processes in the "wrong" location. The situation is reminiscent of the differences shown in Example 1.1.1 for the "interrupt" behavior. To remedy this, the encoding uses additional synchronizations to bring processes into the appropriate locations. This reflects prominently in the *cost* of mimicking a source computation step, as measured by the number of its associated target computation steps (which are spelled out by statements of operational correspondence). The encoding into the calculus with subjective updates does not have this limitation, and so it is more efficient than the encoding that uses objective update.

Ultimately, results of encoding presented in the dissertation have shown that encoding of aborting semantics (with static and dynamic compensation) into adaptable processes (with subjective and objective update) is the most complex. These are expected results since aborting semantics provide the highest level of protection.

## 1.3 Publications and Structure

**Publications** This thesis distills and brings together results from the workshop paper:

- J. Dedeić, J. Pantović, and J. A. Pérez. On compensation primitives as adaptable processes. In S. Crafa and D. Gebler, editors, Proceedings of the Combined 22nd International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, and 12th Workshop on Structural Operational

Semantics, EXPRESS/SOS 2015, Madrid, Spain, 31st August 2015., volume 190 of EPTCS, pages 16 – 30, 2015 ([16]).

Particularly, in this paper we present translations from the calculus of compensable processes into the calculus of adaptable processes with *objective updates*. The continuation of our research work has been inspired by the same topic. Therefore, we got the next conference paper as a result:

- J. Dedeić, J. Pantović, and J. A. Pérez. Efficient compensation handling via subjective updates. In Proceedings of the 32nd ACM SIGAPP Symposium On Applied Computing, SAC'17, pages 51 – 58, Marrakesh, Morocco, April 3 – 7, 2017. ACM ([14]).

In this conference paper we presented translations from the calculus of compensable processes into the calculus of adaptable processes with *subjective updates*, and compared against those in [16]. We state that encodings with subjective update are better than encodings with objective updates in terms of efficiency.

The previous results required additional analysis, verification, and extensions that are published in:

- J. Dedeić, J. Pantović, and J. A. Pérez. On primitives for compensation handling as adaptable processes. Journal of Logical and Algebraic Methods in Programming, page 100675, 2021 ([15]).

This journal paper distills, improves, and collects preliminary results from our papers [16] and [14]. A main difference between [16, 14] and the journal paper ([15]) is that here we concentrate on a specific source calculus, namely the calculus in [29] with *static recovery* and *discarding semantics* (cf. Figure 1.2, encoding $\mathcal{C}_{\mathsf{D}}$ into $\mathcal{S}$ and $\mathcal{O}$). Indeed, the developments in [16, 14] consider also source calculi with *dynamic recovery* and/or *preserving* and *aborting semantics*. The calculus with static recovery and discarding semantics arguably defines the simplest setting for both encodings, one in which the key differences between compensable and adaptable processes can be more sharply presented. Also, this focus allows us to have a concise presentation. In this thesis, we extend the analysis of the (efficient) encoding to source calculi with the semantics that we considered in [16] and [14].

Below we list the key extensions of the results that are developed with respect to the journal paper [15]:

1. We develop the class of *well-formed* compensable processes to formalize encodings, for which error notifications are crucial. More precisely, this class of processes disallows certain non-deterministic interactions that involve nested transactions and error notifications.
2. We extend the criteria included in the definition of valid encoding. The following criteria have been added: *name invariance*, *divergence reflection* and *success sensitiveness*. Therefore, we included additional definitions and theorems that establish that encoding satisfies these new criteria.
3. We develop additional definitions and theorems necessary to complete the proof of operational correspondence (completeness and soundness).

The work presented in the thesis builds on the work presented in the journal paper ([15]) by providing a more pedagogical introduction to the model and incorporating all additional results. Results presented in the thesis extend results of other source calculi by following the insights in [16, 14]. More precisely, the extension of results was done through a detailed analysis of the remaining semantics of compensable processes: preserving, aborting, and dynamic recovery. In the thesis, we also prove that translations from the calculus of compensable processes into the calculus of adaptable processes with the subjective and objective update are valid encodings. In such a way, twelve different encodings of the calculus with *compensation* into calculus with the *dynamic update* have been obtained and proved.

We point out that in the dissertation, there are: definitions, theorems, lemmas, proofs, examples with their relevant explanations, and notation conventions, taken in the original from the papers [14, 15, 16]. The other materials in the dissertation which came from another source are cited adequately.

**Structure of the thesis.** The thesis consists of seven chapters.

**Chapter 1 Introduction.** Describes the main subject and goals of the research and provides motivation for the development of the encoding presented in Chapter 3 to Chapter 6. Also, this chapter provides an overview of the literature related to the research topic.

**Chapter 2 Preliminaries.** This chapter introduces the theoretical foundation for the dissertation. It provides a fundamental overview of the calculus of compensable processes and the calculus of adaptable processes. First, we introduce process calculi informally, through examples, and then present their formal presentation follows through a detailed analysis of syntax and operational semantics. This chapter also contains a definition of well-formed compensable processes. Well-formed compensable processes represent a class of processes. As noted, this class of processes disables certain non-deterministic interactions. Well-formed compensatory processes represent, among other things, the original scientific contribution of this dissertation. Also, it provides a general overview of the expressiveness of concurrent languages, an overview of the literature, and the techniques used in them.

**Chapter 3 Encoding compensable into adaptable processes with subjective update.** In this chapter, we study the expressive power of the encoding calculus for compensable processes into the calculus of adaptable processes with the subjective update. Precisely, we present translations of calculus for compensable processes with *static* recovery, $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, $\mathcal{C}_{\mathtt{A}}$ into calculus of adaptable processes with *subjective* update, $\mathcal{S}$. We also prove that translations satisfy all or selected properties defined for a valid encoding.

**Chapter 4 Encoding compensable into adaptable processes with objective update.** This chapter studies the expressive power of the encoding of calculus for compensable processes into calculus for adaptable processes with *objective* update. First, it introduces the basic concepts and notions, and then we define the translations, $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$. Afterward, it proves that translations satisfy all or selected properties defined for a valid encoding. Also, in this chapter, we deal with the efficiency of encoding, i.e., we introduce a new criterion. Since in the dissertation we develop two kinds of encodings of compensable processes into adaptable processes: encoding with the subjective and objective update, in this chapter we compare their *efficiency*. We define efficiency in abstract terms, considering the number of reduction steps that a target language requires to mimic the behavior of a source language. We prove that encodings that use subjective updates are more efficient than encodings that use objective updates.

**Chapter 5 Encoding dynamic compensation processes into adaptable processes with subjective update.** This chapter introduces preliminaries for *encodings of* $\mathcal{C}^{\lambda}$ *into* $\mathcal{A}$. Also, it studies the expressive power by the encoding of a calculus for compensable processes with *dynamic recovery*, $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}^{\lambda}_{\mathtt{P}}$, $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into calculus for adaptable processes with subjective update, $\mathcal{S}$. We prove that translations satisfy selected properties defined for a valid encoding.

**Chapter 6 Encoding dynamic compensation processes into adaptable processes with objective update.** This chapter studies the expressive power of the encoding of calculus of compensable processes with dynamic recovery, $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}^{\lambda}_{\mathtt{P}}$, $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into calculus of adaptable processes with the objective update, $\mathcal{O}$. We also analyze the question of the efficiency of encodings. Specifically, we have again proved that encodings that use a subjective update are more efficient than encodings that use an objective update.

**Chapter 7 Conclusions and perspectives.** We conclude with an overview of the contributions of the thesis. We additionally provide suggestions/ideas on how the work presented in the dissertation can be enhanced and extended. We state several open questions that we plan to consider as a part of further research work.

# CHAPTER 2

# Preliminaries

In this chapter, we introduce the theoretical background for the dissertation. The chapter is formed of the following three sections, in which we give a brief introduction to the most important concepts related to the development of the thesis:

**Section 2.1** introduces the basic terminology and concepts used in the dissertation. More precisely, it gives a brief introduction to process calculi — compensable processes and adaptable processes.

**Section 2.2** intuitively, by using examples, introduces core calculi for compensable processes ([29]) and adaptable processes ([7]). In the continuation of the section, we formally present the corresponding calculi through their syntax and operational semantics. Also, this section contains the definition of well-formed compensable processes. Well-formed compensable processes present a class of processes that disallows certain non-deterministic interactions involving nested transactions and error notification names. Well-formed compensable processes present, among others, an original contribution of our work.

**Section 2.3** presents the most significant issues of the analysis of the expressiveness of concurrent languages. We provide an overview of expressiveness studies as well as the techniques utilized to conduct them.

## 2.1 Process Calculi

The complexity of programs increases, and naturally, this affects the complexity of the models required to reason about them. As stated in [48], *formal methods* are used for the analysis of properties of complex systems. The design and verification of software systems should be mathematically based since ensuring the reliability and correctness of software systems is very difficult. Formal methods are techniques based on mathematical and logical frameworks, and they are used for the *specification* and *verification* of complicated systems. In terms of the formal specification, a system is defined with a modeling language. A modeling language employs accurate mathematical *syntax* and *semantics*. Also, when formal specification is created, one can demonstrate a set of properties of the system. Mathematical proofs are used to verify the theorems. In the following, we list some formal methods for concurrency, as examples of formal approaches that one may use to specify and verify application behavior: Petri nets [46, 50], communicating state machines [6], and process calculi [26, 32, 33, 34].

One of the most used formal methods for the analysis of complex systems is *process calculi* (or process algebra). The process calculi are a diverse family of related approaches for the formal modeling of complex systems. Therefore, they can be used to express different concepts, e.g., nondeterminism, parallelism, distribution, real-time, stochastic phenomena, etc.

As stated in [38], process algebras come with a precise mathematical framework that has well-defined syntax and semantics. Semantics permit describing and verifying properties of communicating systems. Therefore, process algebra focuses on the specification and manipulation of process terms as induced by a collection of operator symbols [18]. The operator symbols are used to build finite processes, parallel execution, communication, and some form of recursion to express infinite behavior. The first component of process algebra is syntax. It contains all the necessary rules to build terms from the operators and other language constructors. When the language is syntactically defined, then it is crucial to provide a way to describe the behavior of the system being modeled. The prior goal will be achieved if we introduce semantics. Semantics should describe the way a process reduces.

Over the past forty years, research work on process algebras has been very intensive, and a large number of results have been published that started with the introduction of CCS [32], CSP [26], and ACP [4]. A brief historical overview of process algebra is presented in the paper [3]. The $\pi$-calculus [33] has become more recently prominent as a process calculus to reason about mobile systems. There is a large number of calculi for the concurrency theory in which the $\pi$-calculus is used as a core: [1, 12, 20, 24, 23, 28, 41, 47, 29, 30, 49]. Our interest is in CCS and $\pi$-calculus.

## 2.2 The Calculi

In this section, we introduce formally the calculus of compensable processes and the calculus of adaptable processes. To focus on their essentials, both calculi are defined as extensions of CCS [32].

We especially emphasize subsection § 2.2.2 where we identify/develop a class of *well-formed* compensable processes. This class of compensable processes is very useful in our developments. We start by defining some relevant base sets for names.

**Definition 2.2.1** (Base Sets)**.** We assume the following countable sets of names:

- $\mathcal{N}_t$ is a finite set of *transaction names*, ranged over by $t, t', s, s', \ldots$, also used as *error notification names*;
- $\mathcal{N}_l$ is a set of *location names*, ranged over by $l, l', t, t', s, s', \ldots$, also used as *input names*;
- $\mathcal{N}_s$ is the set that collects all other (input/output) names, ranged over by $a, b, c, \ldots$.

For compensable processes, we shall use the set $\mathcal{N}_c = \mathcal{N}_t \cup \mathcal{N}_s$; for adaptable processes, we shall use the set $\mathcal{N}_a = \mathcal{N}_l \cup \mathcal{N}_s$. Some assumptions on these sets are in order. First, $\mathcal{N}_l \cap \mathcal{N}_s = \emptyset$ and $\mathcal{N}_t \cap \mathcal{N}_s = \emptyset$. Also, $\mathcal{N}_t \subseteq \mathcal{N}_l$: our encoding will map each transaction into a process residing at a location with the same name. Finally, we shall use $x, y, w, x', y', w', \ldots$ to denote elements of the three sets when there is no need to distinguish them. For adaptable processes, we shall use $X, Y, Z, \ldots$ to denote *process variables.*

Figure 2.1 illustrates base sets of names that are given in Definition 2.2.1.

### 2.2.1 Compensable Processes

In the field of concurrent and mobile systems, the concept of a long-running transaction is utilized to solve challenges caused by unexpected events that commonly occur during application execution (such as the Internet or wireless networks). As stated in [9], compensable programs offer an appropriate paradigm to carry out long-running transactions. They provide a structured and modular approach to the composition of distributed transactional activities. The main idea is that a particular activity has its compensation and that the compensable program fixes the order of execution of such activities.

Throughout the following subsections, we assume the following notation conventions:

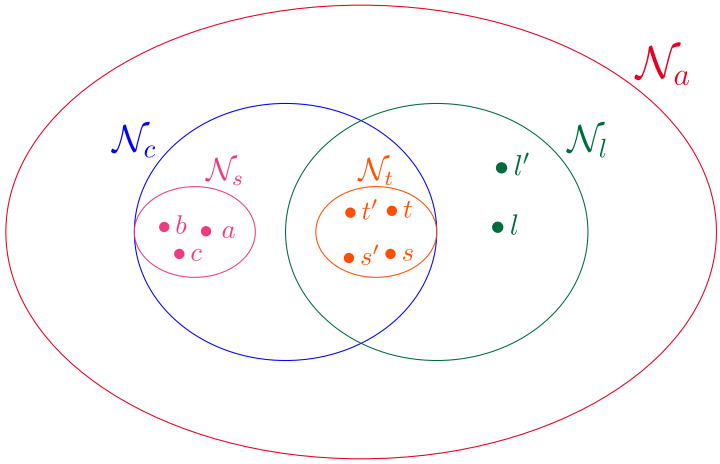



Figure 2.1: Base sets of names.

- $\mathcal{C}_{\mathtt{D}}$ denotes the calculus of compensable processes with discarding semantics and static recovery;
- $\mathcal{C}_{\mathtt{P}}$ denotes the calculus of compensable processes with preserving semantics and static recovery;
- $\mathcal{C}_{\mathtt{A}}$ denotes the calculus of compensable processes with aborting semantics and static recovery.

#### 2.2.1.1 Compensable Processes, by Example

The process language with compensations that we consider here is based on the calculus in [30] (which is, in turn, a variant of the calculus in [29]). The calculi in [30, 29] were introduced as extensions of the $\pi$-calculus with primitives for *static* and *dynamic recovery*. We consider a variant without name mobility and with static recovery; this allows us to focus on the fundamental aspects of compensations. Later, in the thesis, we will extend our focus and analyze dynamic recovery in detail (Chapter 5). The languages in [30, 29] feature two salient constructs:

1. *Transactions* $t[P,Q]$, where $t$ is a transaction name and $P, Q$ are processes;
2. *Protected blocks* $\langle Q \rangle$, where $Q$ is a process.

A transaction $t[P,Q]$ is a process that consists of a *default activity* $P$ and a *compensation activity* $Q$. Transactions can be nested: process $P$ in $t[P,Q]$ may contain other transactions. Also, they can be cancelled: process $t[P,Q]$ behaves as $P$ until an *error notification* (failure signal) arrives along name $t$. Error notifications are output messages coming from inside or outside the transaction. To illustrate the simplest manifestation of compensations, consider the following transitions:

$$t[P,Q] \mid \bar{t}.R \xrightarrow{\tau} Q \mid R \qquad\qquad t[\bar{t}.P_1 \mid P_2,Q] \xrightarrow{\tau} Q \tag{2.1}$$

The left (resp. right) transition shows how $t$ can be canceled by an external (resp. internal) signal. Failure discards the default behavior; the compensation activity is executed instead. In both cases, the default activity (i.e., $P$ and both $P_1, P_2$) are entirely discarded. This may not be desirable in all cases; after the compensation is enabled, we may like to preserve (some of) the behavior in the default activity. When compensation is executed, then the system must be returned to a consistent state. This consistent state may be different from the state in which the transaction started. To this end, one can use *protected blocks* to shield a process from failure signals. These blocks are transparent: $Q$ and $\langle Q \rangle$ have the same behavior, but $\langle Q \rangle$ is not affected by failure signals. This way, the transition

$$t_2[P_2,Q_2] \mid \overline{t_2} \xrightarrow{\tau} \langle Q_2 \rangle,$$

says that the compensation behavior $Q_2$ will be immune to failures. Now consider process

$$P = t_1\big[t_2[P_2,Q_2] \mid \overline{t_2}.R_1,Q_1\big],$$

in which transaction $t_2$ occurs nested inside $t_1$ and $P_2$ does not contain protected blocks. The semantics in [30, 29] refines (2.1) by providing ways to (partially) preserve behavior after a compensation step. This is realized by the *extraction function* on processes, denoted $\mathsf{extr}(\cdot)$. For process $P$, the semantics in [30, 29] decree:

$$t_1\big[t_2[P_2,Q_2] \mid \overline{t_2}.R_1,Q_1\big] \xrightarrow{\tau} t_1\big[\langle Q_2 \rangle \mid \mathsf{extr}(P_2) \mid R_1,Q_1\big].$$

There are different choices for this extraction function: in the *discarding* semantics that we consider here, only top-level protected blocks are preserved (cf. Figure 2.2); hence, in the example above, $\mathsf{extr}(P_2) = \mathbf{0}$.

We consider *discarding*, *preserving*, and *aborting* variants for $\mathsf{extr}(\cdot)$. They define three different semantics for compensations (cf. Figure 2.2). Noted $\mathsf{extr}_{\mathtt{D}}(\cdot)$, $\mathsf{extr}_{\mathtt{P}}(\cdot)$, and $\mathsf{extr}_{\mathtt{A}}(\cdot)$, respectively, these functions concern mostly protected blocks and transactions. Given a process $P$, we would have:

- $\mathsf{extr}_{\mathtt{D}}(P)$ keeps only protected blocks in $P$. Other processes (including transactions) are discarded.
- $\mathsf{extr}_{\mathtt{P}}(P)$ keeps protected blocks and transactions at the top-level in $P$. Other processes are discarded.
- $\mathsf{extr}_{\mathtt{A}}(P)$ keeps protected blocks in nested transactions in $P$, including their respective compensation activities. Other processes are discarded.

As an example, consider the process $P = t\big[t_1[P_1,Q_1] \mid t_2[\langle P_2 \rangle,Q_2] \mid R \mid \langle P_3 \rangle,Q_5\big]$. We then have:

$$\begin{aligned}
\mathcal{C}_{\mathtt{D}} :\ & \overline{t} \mid P \xrightarrow{\tau}_{\mathtt{D}} \langle P_3 \rangle \mid \langle Q_5 \rangle; \\
\mathcal{C}_{\mathtt{P}} :\ & \overline{t} \mid P \xrightarrow{\tau}_{\mathtt{P}} \langle P_3 \rangle \mid \langle Q_5 \rangle \mid t_1[P_1,Q_1] \mid t_2[\langle P_2 \rangle,Q_2]; \\
\mathcal{C}_{\mathtt{A}} :\ & \overline{t} \mid P \xrightarrow{\tau}_{\mathtt{A}} \langle P_3 \rangle \mid \langle Q_5 \rangle \mid \langle P_2 \rangle \mid \langle Q_1 \rangle \mid \langle Q_2 \rangle.
\end{aligned}$$

Notice that the three different semantics provide varying levels of protection. In the *discarding* semantics only top-level protected blocks are preserved. Therefore, it only concerns the compensation activity for transaction $t$ and the protected block $\langle P_3 \rangle$. The *preserving semantics* protects also the nested transactions $t_1$ and $t_2$; a process such as $R$, without an enclosing protected block, is discarded. Last but not least, the *aborting semantics* preserves all protected blocks and compensation activities in the default activity for $t$, also including those in nested transactions, such as $\langle P_2 \rangle$.

With these intuitions in place, we illustrate compensable processes with discarding semantics, by means of an example.

**Example 2.2.1.** Consider a simple hotel booking scenario in which a hotel and a client interact to book and pay a room, and to exchange an invoice. This scenario may be represented using compensable processes as follows (below we omit trailing $\mathbf{0}$s):

$$\mathit{Reservation} \stackrel{def}{=} \mathit{Hotel} \mid \mathit{Client}$$

$$\begin{array}{rcl} Client & \stackrel{def}{=} & \overline{book}.\overline{pay}.(invoice + \overline{t}.refund) \\ Hotel & \stackrel{def}{=} & t[book.pay.\overline{invoice},\overline{refund}] \end{array}$$

Here we represent the hotel's behavior as a transaction $t$ that allows clients to book a room and pay for it. If the client is satisfied with the reservation, then the hotel will send his/her an invoice. Otherwise, the client may cancel the transaction; in that case, the hotel offers the client a refund. Suppose that the client decides to cancel his/her reservation; as we will see, there are four transition steps for process *Reservation*:

$$\begin{array}{rcl} Reservation & \xrightarrow{\tau}_{\mathtt{D}} & t[pay.\overline{invoice},\overline{refund}] \mid \overline{pay}.(invoice + \overline{t}.refund) \\ & \xrightarrow{\tau}_{\mathtt{D}} & t[\overline{invoice},\overline{refund}] \mid invoice + \overline{t}.refund \\ & \xrightarrow{\tau}_{\mathtt{D}} & \langle \overline{refund} \rangle \mid refund \\ & \xrightarrow{\tau}_{\mathtt{D}} & \langle \mathbf{0} \rangle \mid \mathbf{0}. \end{array}$$

#### 2.2.1.2 Compensable Processes, Formal Description

This section formally introduces the semantics for the language of compensable processes with static recovery. Likewise, compensable processes with dynamic recovery will be discussed in Chapter 5.

##### 2.2.1.2.1 Syntax - Static recovery processes

The calculus of *compensable processes* considers prefixes $\pi$ and processes $P, Q, \ldots$ defined as:

$$\pi \quad ::= \quad a \;\big|\; \overline{x}$$

$$P,Q \quad ::= \quad \underbrace{\mathbf{0} \;\big|\; \pi.P \;\big|\; !\pi.P \;\big|\; (\nu x)P \;\big|\; P \mid Q}_{\text{CCS processes}} \;\big|\; \underbrace{t[P,Q] \;\big|\; \langle Q \rangle}_{\text{extension}}$$

Prefixes $\pi$ include input actions ($a$), output actions ($\overline{a}$) and error notifications ($\overline{t}$). Processes for inaction ($\mathbf{0}$), action prefix ($\pi.P$), guarded replication ($!\pi.P$), restriction ($(\nu x)P$) and parallel composition ($P \mid Q$) are standard. Protected blocks $\langle Q \rangle$ and transactions $t[P,Q]$ have already been motivated. Name $x$ is bound in $(\nu x)P$, i.e., name $x$ is known only to the process $P$.

##### 2.2.1.2.2 Operational Semantics

Following [29, 30], the semantics of compensable processes is given in terms of a labeled transition system (LTS). Ranged over by $\alpha, \alpha'$, the set of labels includes $a$, $\overline{a}$, $t$, $\overline{t}$ and $\tau$. As in CCS, $a$ denotes an input action, $\overline{a}$ denotes an output action, $\overline{t}$ denotes an error notification and $\tau$ denotes synchronization (internal action). As explained in § 2.2.1.1, this LTS is parametric in an extraction function, which is defined in Figure 2.2. Error notifications can be *internal* or *external* to the transaction: if the error notification is generated from the default activity then we call it internal; otherwise, the error is external.

Formally, we have three different LTSs, corresponding to processes under discarding, preserving, and aborting semantics. Therefore, for each $\kappa \in \{\mathtt{D}, \mathtt{P}, \mathtt{A}\}$, we will have an extraction function $\mathsf{extr}_\kappa(\cdot)$ and a transition relation $\xrightarrow{\alpha}_\kappa$. The rules of the LTSs are given in Figure 2.3. As a convention, whenever a notion coincides for the three semantics, we shall avoid decorations $\mathtt{D}$, $\mathtt{P}$, and $\mathtt{A}$. This way, e.g., by writing $\mathsf{extr}(\langle P \rangle) = \langle P \rangle$ we mean that the extraction function for protected blocks is the same for all three semantics. Figure 2.3 gives the rules of the LTS; we comment briefly on each of them:

- Axioms (L-In) and (L-Out) execute input and output prefixes, respectively.

$$\begin{array}{ll}
\mathsf{extr_D}(t[P,Q]) = \mathbf{0} & \mathsf{extr_P}(t[P,Q]) = t[P,Q] \\
\mathsf{extr_A}(t[P,Q]) = \mathsf{extr_A}(P) \mid \langle Q\rangle & \mathsf{extr}(P \mid Q) = \mathsf{extr}(P) \mid \mathsf{extr}(Q) \\
\mathsf{extr}(\langle P\rangle) = \langle P\rangle & \mathsf{extr}((\nu a)P) = (\nu a)\mathsf{extr}(P) \\
\mathsf{extr}(\pi.P) = \mathbf{0} & \mathsf{extr}(!\pi.P) = \mathbf{0}
\end{array}$$

Figure 2.2: Extraction function for static recovery.

$$\text{(L-In)}\ \ a.P \xrightarrow{a} P \qquad \text{(L-Out)}\ \ \overline{x}.P \xrightarrow{\overline{x}} P \qquad \text{(L-Rep)}\ \ \frac{\pi.P \xrightarrow{\alpha} P'}{!\pi.P \xrightarrow{\alpha} P' \mid !\pi.P} \qquad \text{(L-Par1)}\ \ \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \qquad \text{(L-Par2)}\ \ \frac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$$

$$\text{(L-Res)}\ \ \frac{P \xrightarrow{\alpha} P' \quad \alpha \notin \{x,\overline{x}\}}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'} \qquad \text{(L-Comm1)}\ \ \frac{P \xrightarrow{x} P' \quad Q \xrightarrow{\overline{x}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \qquad \text{(L-Comm2)}\ \ \frac{P \xrightarrow{\overline{x}} P' \quad Q \xrightarrow{x} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \qquad \text{(L-Block)}\ \ \frac{P \xrightarrow{\alpha} P'}{\langle P\rangle \xrightarrow{\alpha} \langle P'\rangle}$$

$$\text{(L-Rec-Out)}\ \ \frac{}{t[P,Q] \xrightarrow{t} \mathsf{extr}(P) \mid \langle Q\rangle} \qquad \text{(L-Scope-Out)}\ \ \frac{P \xrightarrow{\alpha} P' \quad \alpha \neq \overline{t}}{t[P,Q] \xrightarrow{\alpha} t[P',Q]} \qquad \text{(L-Rec-In)}\ \ \frac{P \xrightarrow{\overline{t}} P'}{t[P,Q] \xrightarrow{\tau} \mathsf{extr}(P') \mid \langle Q\rangle}$$

Figure 2.3: An LTS for compensable processes.

- Rule (L-Rep) deals with guarded replication, i.e., the replicated process executes an action while simultaneously activating a copy of the original process.
- Rules (L-Par1) and (L-Par2) allow one parallel component to progress independently.
- Rule (L-Res) is the standard rule for restriction. A transition of process $P$ determines a transition of process $(\nu x)P$, where the side condition provides that the restricted name $x$ does not occur in $\alpha$.
- Rules (L-Comm1) and (L-Comm2) define synchronization on $x$, i.e., performs communication between output and input action.
- Rule (L-Block) specifies that protected blocks are transparent units of behavior.
- Rule (L-Rec-Out) allows an external process to abort a transaction via an output action $\overline{t}$. The resulting process contains two parts: the first is obtained from the default activity of the transaction via the extraction function (cf. Figure 2.2); the second corresponds to the compensation activity, executed in a protected block.
- Rule (L-Scope-Out) allows the default activity of a transaction to progress in case there is no internal error notification.
- Rule (L-Rec-In) handles failure when the error notification is internal to the transaction.

To define the semantics and to capture the fundamental properties of language constructs, we use structural congruence. In the following, we define structural congruence ($\equiv$) and evaluation contexts for compensable processes.

**Definition 2.2.2** (Structural congruence)**.** Structural congruence is the smallest congruence relation on processes that is generated by the following rules:

| | |
|---|---|
| $P \mid Q \equiv Q \mid P$ | $(\nu x)\mathbf{0} \equiv \mathbf{0}$ |
| $P \mid (Q \mid R) \equiv (P \mid Q) \mid R$ | $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$ |
| $P \mid \mathbf{0} \equiv P$ | $Q \mid (\nu x)P \equiv (\nu x)(P \mid Q)$ if $x \notin \mathtt{fn}(Q)$ |
| $!\pi.P \equiv \pi.P \mid !\pi.P$ | $t[(\nu x)P,Q] \equiv (\nu x)t[P,Q]$ if $t \neq x,\ \ x \notin \mathtt{fn}(Q)$ |
| $P \equiv Q$ if $P \equiv_\alpha Q$ | $\langle(\nu x)P\rangle \equiv (\nu x)\langle P\rangle$ |

The first column in Definition 2.2.2 contains standard rules: commutativity, associativity and neutral element for parallel composition. We rely on usual notions of $\alpha$-conversion (noted $\equiv_\alpha$). The second column contains garbage collection of useless restrictions, swapping of restrictions, and scope extrusion for parallel composition. Rules for transaction scope and protected blocks are in red because these are rules created due to the extension of CCS syntax (cf. Paragraph 2.2.1.2.1).

**Definition 2.2.3** (Evaluation Contexts)**.** The syntax of evaluation contexts of compensable processes is given by the following grammar:

$$C[\bullet] ::= [\bullet] \;\big|\; \langle C[\bullet]\rangle \;\big|\; t[C[\bullet],P] \;\big|\; C[\bullet] \mid P \;\big|\; (\nu x)C[\bullet],$$

where $P$ is a compensable process.

We write $C[Q]$ to denote the process obtained by replacing the hole $[\bullet]$ in context $C[\bullet]$ with $Q$. In the following we will use $E[\bullet], D[\bullet]$ to denote contexts. Before the proof of Proposition 2.2.3, we present the following auxiliary lemma:

**Lemma 2.2.2.** Let $P$ be a compensable process.

(a) If $P \xrightarrow{a} P'$ then $P \equiv C[a.P_1]$ and $P' = C[P_1]$;

(b) If $P \xrightarrow{t} P'$ then $P \equiv C[t[P_1,Q_1]]$ and $P' \equiv C[\mathsf{extr}(P_1) \mid \langle Q_1\rangle]$;

(c) If $P \xrightarrow{\bar{x}} P'$ then $P \equiv C[\bar{x}.P_1]$ and $P' \equiv C[P_1]$,

for some context $C[\bullet]$, names $a, t, x$ and processes $P_1, Q_1$.

*Proof.* The proof is by induction on the derivation of $P \xrightarrow{\alpha} P'$, where $\alpha \in \{a, t, \overline{x}\}$. ∎

The following proposition is key to operational correspondence statements.

**Proposition 2.2.3.** Let $P$ be a compensable process. If $P \xrightarrow{\tau} P'$ then one of the following holds:

(a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$,

(b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$,

(c) $P \equiv C[t[D[\overline{t}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}(D[P_1]) \mid \langle Q\rangle]$,

for some contexts $C[\bullet], D[\bullet], E[\bullet]$, processes $P_1, P_2, Q$ and names $a, t$.

*Proof.* The proof proceeds by induction on the inference of $P \xrightarrow{\tau} P'$. We will show that the proposition is true for the base cases, whereas the inductive step follows directly.

By LTS (cf. Figure 2.3), in accordance with Rules (L-COMM1), (L-COMM2) and (L-REC-IN) we have two possible cases, for $E[\bullet] = [\bullet]$ and some contexts $C[\bullet], D[\bullet]$, process $P_1, P_2, P_1', P_2', P_1'', P_2'', Q, Q_1$ and names $a, t$:

- *Case* (L-COMM1)*:* By Rule (L-COMM1) we have: $P \equiv P_1' \mid P_2'$, $P_1' \xrightarrow{x} P_1''$, $P_2' \xrightarrow{\bar{x}} P_2''$, $P' \equiv P_1'' \mid P_2''$ and by Lemma 2.2.2, we conclude that:
  (a) $P_2' \equiv D[a.P_2]$, $P_2'' \equiv D[P_2]$, $P_1' \equiv C[\bar{a}.P_1]$, and $P_1'' \equiv C[P_1]$, or
  (b) $P_2' \equiv D[\bar{t}.P_2]$, $P_2'' \equiv D[P_2]$, $P_1' \equiv C[t[P_1, Q_1]]$, and $P_1'' \equiv C[\mathsf{extr}(P_1) \mid \langle Q_1 \rangle]$.
- *Case* (L-REC-IN)*:* By Rule (L-REC-IN) we have: $P \equiv t[P_1', Q]$, $P_1' \xrightarrow{\bar{t}} R$, $P' \equiv \mathsf{extr}(R) \mid \langle Q \rangle$ and by Lemma 2.2.2, we conclude that: $P_1' \equiv D[\bar{t}.P_1]$ and $R \equiv D[P_1]$.

■

**Remark 2.2.4** (Reductions)**.** We define a reduction semantics for compensable processes by exploiting the LTS just introduced: we shall write $P \longrightarrow P'$ whenever $P \xrightarrow{\tau} P''$ and $P'' \equiv P'$, for some $P''$. As customary, we write $\longrightarrow^*$ to denote the reflexive and transitive closure of $\longrightarrow$.

### 2.2.2 Well-formed Compensable Processes

We shall focus on *well-formed* compensable processes: a class of processes that disallows certain non-deterministic interactions involving nested transaction and error notification names. Concise examples of processes that are not well-formed are the following:

$$\begin{aligned} P &= t_1\big[a \mid t_2[b,\bar{b}],\bar{a}\big] \mid \overline{t_1} \mid \overline{t_2} \;\; \times \\ P_1 &= t_1[a,b] \mid t_2[\overline{t_1},d] \mid \overline{t_2} \;\; \times \\ P_2 &= t_1[\overline{t_2},a] \mid t_2[\overline{t_1},b] \;\; \times \end{aligned} \tag{2.2}$$

Processes $P$, $P_1$ and $P_2$ feature concurrent error notifications (on $t_1$ and $t_2$), which induce a form of non-determinism that is hard to capture properly in the (lower level) representation that we shall give in terms of adaptable processes. Indeed, $P$ features an *interference* between the failure of $t_1$ and $t_2$; it is hard to imagine patterns where this kind of interfering concurrency may come in handy. For the same reason, we will assume that all transaction names in a well-formed process are different. In contrast, we would like to consider as well-formed the following processes:

$$P' = t_1\big[a \mid t_2[b,\bar{b}],\bar{a}\big] \mid \overline{t_2}.\overline{t_1} \;\; \checkmark \qquad\qquad P'' = t_1[a,\bar{a}] \mid t_2[b,\bar{b}] \mid \overline{t_1} \mid \overline{t_2} \;\; \checkmark \tag{2.3}$$

In what follows, we formally introduce well-formed compensable processes. We require some notations: (a) sets of pairs $\Gamma, \Delta \subseteq \mathcal{N}_t \times \mathcal{N}_t$, (b) sets $\gamma, \delta \in \mathcal{N}_t$, and (c) boolean $p \in \{\top, \bot\}$. These elements have the following reading:

- $\Gamma$ is the *set of (potential) pairs of parallel failure signals* in $P$;
- $\Delta$ is the *set of (potential) pairs of nested transaction names* in $P$ (with form (parent,child));
- $\gamma$ is the *set of failure signals* in $P$;
- $\delta$ is the *set of top-level transactions* in $P$;
- $p$ is $\top$ iff $P$ contains protected blocks.

$$\text{(W-NIL)}\quad \emptyset;\emptyset \vdash_{\emptyset;\emptyset;\bot} \mathbf{0} \qquad \text{(W-OUT1)}\ \frac{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} P}{\Gamma;\emptyset \vdash_{\gamma\cup\{t\};\emptyset;\bot} \overline{t}.P} \qquad \text{(W-OUT2)}\ \frac{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} P}{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} \overline{a}.P} \qquad \text{(W-IN)}\ \frac{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} P}{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} a.P}$$

$$\text{(W-RES)}\ \frac{\Gamma;\Delta \vdash_{\gamma;\delta;p} P}{\Gamma,\Delta \vdash_{\gamma;\delta;p} (\nu x)P} \qquad \text{(W-BLOCK)}\ \frac{\Gamma;\Delta \vdash_{\gamma;\delta;p} P}{\Gamma;\Delta \vdash_{\gamma;\delta;\top} \langle P\rangle} \qquad \text{(W-REP)}\ \frac{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} \pi.P}{\Gamma;\emptyset \vdash_{\gamma;\emptyset;\bot} !\pi.P}$$

$$\text{(W-TRANS)}\ \frac{\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P \qquad \Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q \qquad f_t(\mathcal{P}(P),\mathcal{P}(Q)) = (\Gamma,\Delta) \qquad \Gamma^s \cap \Delta^t = \emptyset}{\Gamma,\Delta \vdash_{\gamma_1\cup\gamma_2;\{t\};p_1\vee p_2} t[P,Q]}$$

$$\text{(W-PAR)}\ \frac{\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P \qquad \Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q \qquad f(\mathcal{P}(P),\mathcal{P}(Q)) = (\Gamma,\Delta) \qquad \Gamma^s \cap \Delta^t = \emptyset}{\Gamma,\Delta \vdash_{\gamma_1\cup\gamma_2;\delta_1\cup\delta_2;p_1\vee p_2} P \mid Q}$$

Figure 2.4: Auxiliary relation for well-formed compensable processes.

We say $P$ is well-formed if $\Gamma;\Delta \vdash_{\gamma;\delta;p} P$, can be derived by means of the rules in Figure 2.4. We write $\mathcal{P}(P)$ to denote the parameters $\Gamma$, $\Delta$, $\gamma$, and $\delta$ associated to $P$, i.e., $\mathcal{P}(P) = (\Gamma,\Delta,\gamma,\delta)$. We briefly comment on the rules in Figure 2.4:

- Rule (W-NIL) states that the inactive process has neither parallel failure signal nor nested transactions; it also does not contain protected blocks.
- Rules (W-OUT1), (W-OUT2), and (W-IN) enforce that protected blocks or transactions do not appear behind prefixes (i.e., $p = \bot$, $\delta = \emptyset$). Rule (W-OUT1) says that if the name of the prefix is the failure signal then it will be collected by $\gamma$. Rule (W-OUT2) says that if the name of the prefix is not the failure signal then the set of the failure signals will be as in the process that appears after the prefix. For example, by (W-NIL) and two successive applications of (W-OUT1), we can infer

$$\emptyset;\emptyset \vdash_{\{t_1,t_2\};\emptyset;\bot} \overline{t_2}.\overline{t_1}.$$

- Rule (W-RES) says that if $P$ satisfies the predicate for some parameters, then $(\nu x)P$ satisfies the predicate with the same parameters.
- Rule (W-BLOCK) specifies that if $P$ satisfies the predicate for some parameters, then $\langle P\rangle$ satisfies the predicate with the same $\Gamma,\Delta,\gamma$ and $\delta$. The fifth parameter for $\langle P\rangle$ specifies that it contains protected blocks ($p = \top$ in the conclusion). This way, for example, we have

$$\emptyset;\emptyset \vdash_{\{t_1,t_2\};\emptyset;\top} \langle\overline{t_2}.\overline{t_1}\rangle.$$

Rules (W-REP), (W-TRANS) and (W-PAR) rely on the following auxiliary notations. First, given sets $\gamma_1,\gamma_2$, $\delta$ and a name $t$, we introduce the following sets:

$$\gamma_1 \times \gamma_2 = \{(t',t'') : t' \in \gamma_1 \wedge t'' \in \gamma_2\} \qquad \{t\} \times \delta = \{(t,t') : t' \in \delta\}. \tag{2.4}$$

Also, we write $\Gamma^s$ and $\Delta^t$ to denote the symmetric closure of $\Gamma$ and the transitive closure of $\Delta$, respectively. We will use, respectively the following functions $f_t$ and $f$ for conditions in Rules (W-Trans) and (W-Par):

$$f_t(\mathcal{P}(P), \mathcal{P}(Q)) = (\Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2), \Delta_1 \cup \Delta_2 \cup (\{t\} \times (\delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))) \tag{2.5}$$
$$f(\mathcal{P}(P), \mathcal{P}(Q)) = (\Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2), \Delta_1 \cup \Delta_2) \tag{2.6}$$

where $\mathcal{P}(P) = (\Gamma_1, \Delta_1, \gamma_1, \delta_1)$ and $\mathcal{P}(Q) = (\Gamma_2, \Delta_2, \gamma_2, \delta_2)$.

We may now discuss Rules (W-Rep), (W-Trans), and (W-Par):

- Rule (W-Rep) says that the set of pairs of parallel failure signals in $!\pi.P$ is $\gamma \times \gamma$, where $\gamma$ is the set of failure signals in $\pi.P$. This is directly related to the transition rule (L-Rep) in Figure 2.3. All other parameters of the predicate satisfied by $!\pi.P$ are the same as for $\pi.P$. For example, we can derive:

$$\{(t_1,t_1),(t_1,t_2),(t_2,t_1),(t_2,t_2)\}; \emptyset \vdash_{\{t_1,t_2\};\emptyset;\bot} !\,\overline{t_2}.\overline{t_1}$$

- Rule (W-Trans) specifies the conditions for $t[P,Q]$ to be well-formed. First, $\delta = \{t\}$. The set of pairs of parallel failure signals is the union of the respective sets for $P$ and $Q$ and set whose elements are pairs of failure signals; in the pair, with the first component belonging to the set of failure signals of $P$ and the second component belonging to the set of failure signals of $Q$. This extension with $\gamma_1 \times \gamma_2$ is necessary for $t[P,Q]$, because $P$ may contain protected blocks which will be composed in parallel with $\langle Q \rangle$ in case of an error. The set of pairs of nested transaction names is obtained from those for $P$ and $Q$, also considering further pairs as specified by $\Delta(t, \delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2)$ (cf. (2.4)). The rule additionally enforces:

$$(\Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2))^s \cap (\Delta_1 \cup \Delta_2 \cup \Delta(t, \delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))^t = \emptyset$$

  For example, we can derive:

$$\emptyset; \{(t_1,t_2)\} \vdash_{\emptyset;\{t_1\};\bot} t_1[a \mid t_2[b,\bar{b}],\bar{a}].$$

- Rule (W-Par) specifies the cases in which process $P \mid Q$ satisfies the predicate provided that $P$ and $Q$ individually satisfy it. The set of pairs of parallel failure signals is obtained as in Rule (W-Trans). The set of pairs of nested transactions is obtained as the union of sets of pairs of nested transactions for $P$ and $Q$. Also, it must hold that:

$$(\Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2))^s \cap (\Delta_1 \cup \Delta_2)^t = \emptyset.$$

  For example, for processes $P'$ and $P''$ in (2.3) we have

$$\emptyset; \{(t_1,t_2)\} \vdash_{\{t_1,t_2\};\{t_1\};\bot} t_1\big[a \mid t_2[b,\bar{b}],\bar{a}\big] \mid \overline{t_2}.\overline{t_1} \quad \text{and}$$

$$\{(t_1,t_2)\}; \emptyset \vdash_{\{t_1,t_2\};\{t_1,t_2\};\bot} t_1\big[a,\bar{a}\big] \mid t_2[b,\bar{b}] \mid \overline{t_1} \mid \overline{t_2}.$$

  One should notice that processes from (2.2) do not satisfy the predicate, since their sets of pairs of parallel failure signals and nested transaction names are not disjoint: they are both equal to $\{(t_1,t_2)\}$.

We then have the following definition:

**Definition 2.2.4** (Well-formedness)**.** A compensable process $P$ is *well-formed* if

(i) transaction names in $P$ are mutually different, and

(ii) $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$ holds for some $\Gamma, \Delta, \gamma, \delta, p$.

The following theorem captures the main properties of well-formed processes: they do not contain subterms with protected blocks or transactions behind prefixes; also, they do not contain potential parallel a failure signals for nested transaction names. Since the former is required to hold also for compensations within transactions, we introduce specific evaluation contexts (cf. Definition 2.2.3) as follows:

$$C^{wf}[\bullet] ::= [\bullet] \;\big|\; \langle C^{wf}[\bullet] \rangle \;\big|\; t[C^{wf}[\bullet], P] \;\big|\; C^{wf}[\bullet] \mid P \;\big|\; (\nu x) C^{wf}[\bullet] \;\big|\; t[P, C^{wf}[\bullet]].$$

**Proposition 2.2.5.** Let $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$, for some $\Gamma, \Delta, \gamma, \delta$ and $p$. Then the following holds:

(i) if $P \equiv C^{wf}[\pi.P_1]$ then $\Gamma'; \emptyset \vdash_{\gamma';\emptyset;\bot} P_1$, for some $\Gamma'$ and $\gamma'$, and

(ii) $\Gamma^s \cap \Delta^t = \emptyset$.

*Proof.* (i) By induction on the structure of $C^{wf}[\bullet]$.

(ii) By induction on the derivation $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$.

■

In the following, we are going to prove that well-formed processes always evolve into well-formed processes. Before giving the statement we present several auxiliary results.

**Lemma 2.2.6** (Inversion Lemma). For some $\Gamma_1, \Delta_1, \gamma_1, \delta_1, p_1, \Gamma_2, \Delta_2, \gamma_2, \delta_2, p_2$, the following holds:

1) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} \mathbf{0}$ then $\Gamma, \Delta, \gamma, \delta$ are empty sets and $p = \bot$;

2) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} \bar{t}.P$ then there is $\gamma'$ such that $\gamma = \gamma' \cup \{t\}$ and $\Gamma; \emptyset \vdash_{\gamma';\emptyset;\bot} P$ and $\Delta = \emptyset$.

3) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} \bar{a}.P$ then $\Gamma; \emptyset \vdash_{\gamma;\emptyset;\bot} P$ and $\Delta = \emptyset$ and $p = \bot$;

4) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} a.P$ then $\Gamma; \emptyset \vdash_{\gamma;\emptyset;\bot} P$ and $\Delta = \emptyset, \delta = \emptyset$ and $\gamma = \emptyset$;

5) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} (\nu x)P$ then $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$;

6) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} \langle P \rangle$ then $p = \top$ and $\Gamma; \Delta \vdash_{\gamma;\delta;p'} P$ for some $p' \in \{\top, \bot\}$;

7) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} !\pi.P$ then $\Gamma'; \emptyset \vdash_{\gamma;\emptyset;\bot} \pi.P$ and $\Gamma = \gamma \times \gamma$ and $\Delta = \emptyset, \delta = \emptyset$ and $p = \bot$;

8) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} P \mid Q$ then $\Gamma_1; \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P$ and $\Gamma_2; \Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2)$ and $\Delta = \Delta_1 \cup \Delta_2$ and $\gamma = \gamma_1 \cup \gamma_2$ and $\delta = \delta_1 \cup \delta_2$ and $p = p_2 \vee p_2$ and $\Gamma^s \cap \Delta^t = \emptyset$;

9) If $\Gamma; \Delta \vdash_{\gamma;\delta;p} t[P, Q]$ then $\Gamma_1; \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P$ and $\Gamma_2; \Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2)$ and $\Delta = \Delta_1 \cup \Delta_2 \cup (\{t\} \times (\delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))$ and $\gamma = \gamma_1 \cup \gamma_2$ and $\delta = \delta_1 \cup \delta_2$ and $p = p_2 \vee p_2$ and $\Gamma^s \cap \Delta^t = \emptyset$;

*Proof.* The proof follows directly from the auxiliary relation for well-formed compensable processes, cf. Figure 2.4. ■

In the following we introduce auxiliary statements that are needed for proving that well-formedness of compensable process is preserved by the rules in Figure 2.3.

**Lemma 2.2.7.** If $\Gamma; \Delta \vdash_{\gamma;\delta;p} t[P, Q]$ then there are $\Delta_1, \delta_1$ such that $\Gamma, \Delta_1 \vdash_{\gamma;\delta_1;\top} P \mid \langle Q \rangle$ and $\Delta_1 \subseteq \Delta$.

*Proof.* Let $\Gamma;\Delta \vdash_{\gamma;\delta;p} t[P,Q]$.

- By Lemma 2.2.6 it follows $\Gamma_1';\Delta_1' \vdash_{\gamma_1';\delta_1';p_1'} P$ and $\Gamma_2';\Delta_2' \vdash_{\gamma_2';\delta_2';p_2'} Q$ where $\Gamma = \Gamma_1' \cup \Gamma_2' \cup (\gamma_1' \times \gamma_2')$ and $\Delta = \Delta_1' \cup \Delta_2' \cup (\{t\} \times (\delta_1' \cup \delta_2' \cup \gamma_1' \cup \gamma_2'))$ and $\gamma = \gamma_1' \cup \gamma_2'$ and $p = p_1' \vee p_2'$ and $\Gamma^s \cap \Delta^t = \emptyset$.
- By formation on Rule (W-Block) we get that $\Gamma_2';\Delta_2' \vdash_{\gamma_2';\delta_2';\top} \langle Q \rangle$.
- By formation on Rule (W-Par) we get that $\Gamma,\Delta_1 \vdash_{\gamma;\delta_1;\top} P \mid \langle Q \rangle$ where $\Delta_1 = \Delta_1' \cup \Delta_2'$ and it is clear that $\Delta_1 \subseteq \Delta$.
- It should be noted that for $\Delta_1 \subseteq \Delta$, based on basic properties of set operations from $\Gamma^s \cap \Delta^t = \emptyset$, that it follows $\Gamma^s \cap \Delta_1^t = \emptyset$.

■

**Lemma 2.2.8.** If $\Gamma;\Delta \vdash_{\gamma;\delta;p} P$ then there are $\Gamma_1, \Delta_1, \gamma_1, \delta_1$ and $p_1$ such that $\Gamma_1, \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} \mathsf{extr}(P)$ and $\Gamma_1 \subseteq \Gamma, \Delta_1 \subseteq \Delta$ and $\gamma_1 \subseteq \gamma$.

*Proof.* The proof proceeds by induction on the derivation $\Gamma;\Delta \vdash_{\gamma;\delta;p} P$.

***Base case:*** If $\Gamma;\Delta \vdash_{\gamma;\delta;p} P$ was derived by (W-Nil) then $P = \mathbf{0}$, $\mathsf{extr}(\mathbf{0}) = \mathbf{0}$. $\Gamma = \Gamma_1 = \emptyset$, $\Delta = \Delta_1 = \emptyset$, $\gamma = \gamma_1 = \emptyset$, $\delta = \delta_1 = \emptyset$, and $p_1 = \bot$.

***Induction step:*** There are six cases, depending on the last rule applied in the derivation. Below we present three cases since the proof follows directly for the other cases.

- *Case* 1 (W-Par): Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;p} P' \mid Q'$ follows from $\Gamma_1';\Delta_1' \vdash_{\gamma_1';\delta_1';p_1'} P'$ and $\Gamma_2';\Delta_2' \vdash_{\gamma_2';\delta_2';p_2'} Q'$,
  - By (W-Par) it follows that $\Gamma = \Gamma_1' \cup \Gamma_2' \cup (\gamma_1' \times \gamma_2')$ (cf. (2.4)), $\Delta = \Delta_1' \cup \Delta_2'$, $\delta = \delta_1' \cup \delta_2'$ and $p = p_1' \vee p_2'$, and holds that $\Gamma^s \cap \Delta^t = \emptyset$.
  - By definition of extraction function (cf. Figure 2.2) we get: $\mathsf{extr}(P' \mid Q') = \mathsf{extr}(P') \mid \mathsf{extr}(Q')$.
  - For process $P'$ by induction hypothesis there are $\Gamma_1'' \subseteq \Gamma_1'$ and $\Delta_1'' \subseteq \Delta_1'$ such that: $\Gamma_1'';\Delta_1'' \vdash_{\gamma_1'';\delta_1'';p_1''} \mathsf{extr}(P')$.
  - Similarly, for process $Q'$ by induction hypothesis there are $\Gamma_2'' \subseteq \Gamma_2'$ and $\Delta_2'' \subseteq \Delta_2'$ such that: $\Gamma_2'';\Delta_2'' \vdash_{\gamma_2'';\delta_2'';p_2''} \mathsf{extr}(Q')$.
  - By formation on Rule (W-Par) we get: $\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} \mathsf{extr}(P') \mid \mathsf{extr}(Q')$ where $\Gamma_1 = \Gamma_1'' \cup \Gamma_2'' \cup (\gamma_1'' \times \gamma_2'')$, $\Delta_1 = \Delta_1'' \cup \Delta_2''$, $\delta_1 = \delta_1'' \cup \delta_2''$ and $p_1 = p_1'' \vee p_2''$, also holds $\Gamma_1^s \cap \Delta_1^t = \emptyset$.
  - It is easy to conclude: $\Gamma_1 \subseteq \Gamma, \Delta_1 \subseteq \Delta$ and $\gamma_1 \subseteq \gamma$.
- *Case* 2 (W-Block): Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;\top} \langle P' \rangle$ follows from $\Gamma;\Delta \vdash_{\gamma;\delta;p} P'$. By definition of extraction function (cf. Figure 2.2) we get: $\mathsf{extr}(\langle P' \rangle) = \langle P' \rangle$. Therefore, it is easy to be concluded that the statement holds.
- *Case* 3 (W-Trans)*:* Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;p} t[P',Q']$ follows from $\Gamma_1';\Delta_1' \vdash_{\gamma_1';\delta_1';p_1'} P'$ and $\Gamma_2';\Delta_2' \vdash_{\gamma_2';\delta_2';p_2'} Q'$.
  - By Rule (W-Trans) we get: $\Gamma = \Gamma_1' \cup \Gamma_2' \cup (\gamma_1' \times \gamma_2')$, $\Delta = \Delta_1' \cup \Delta_2' \cup (\{t\} \times (\delta_1' \cup \delta_2' \cup \gamma_1' \cup \gamma_2'))$, $\delta = \delta_1' \cup \delta_2'$ and $p = p_1' \vee p_2'$, also condition $\Gamma^s \cap \Delta^t = \emptyset$ holds.

- By definition of extraction function (cf. Figure 2.2) we get $\mathsf{extr}(P) = \mathbf{0}$. Therefore, statement holds directly.

For all other cases for the proof follows directly, because of definition of extraction function. ∎

**Lemma 2.2.9.** If $\Gamma; \Delta \vdash_{\gamma;\delta;p} P \mid Q$ then there are $\Gamma_1, \Delta_1, \gamma_1, \delta_1$ and $p_1$ such that $\Gamma_1, \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} \mathsf{extr}(P) \mid Q$ and $\Gamma_1 \subseteq \Gamma, \Delta_1 \subseteq \Delta$ and $\gamma_1 \subseteq \gamma$.

*Proof.* Let $\Gamma; \Delta \vdash_{\gamma;\delta;p} P \mid Q$ .

- By Lemma 2.2.6 we get $\Gamma'_1; \Delta'_1 \vdash_{\gamma'_1;\delta'_1;p'_1} P$ and $\Gamma'_2; \Delta'_2 \vdash_{\gamma'_2;\delta'_2;p'_2} Q$ where $\Gamma = \Gamma'_1 \cup \Gamma'_2 \cup (\gamma'_1 \times \gamma'_2)$ $(cf.(2.4))$, $\Delta = \Delta'_1 \cup \Delta'_2, \gamma = \gamma'_1 \cup \gamma'_2, \delta = \delta'_1 \cup \delta'$ and $p = p'_1 \vee p'_2$ and condition $\Gamma^s \cap \Delta^t = \emptyset$ holds.
- For process $P$ by Lemma 2.2.8 there are $\Gamma''_1 \subseteq \Gamma'_1$, $\Delta''_1 \subseteq \Delta'_1$, and $\gamma''_1 \subseteq \gamma'_1$ such that: $\Gamma''_1; \Delta''_1 \vdash_{\gamma''_1;\delta''_1;p''_1} \mathsf{extr}(P)$.
- For processes $\mathsf{extr}(P)$ and $Q$ by formation on Rule (W-Par) the following holds: $\Gamma_1; \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} \mathsf{extr}(P) \mid Q$ where $\Gamma_1 = \Gamma''_1 \cup \Gamma'_2 \cup (\gamma''_1 \times \gamma'_2), \Delta_1 = \Delta''_1 \cup \Delta'_2, \gamma_1 = \gamma''_1 \cup \gamma'_2, \delta_1 = \delta''_1 \cup \delta'_2$ and $p_1 = p''_1 \vee p'_2$ and $\Gamma^s_1 \cap \Delta^t_1 = \emptyset$ holds by basic properties of set operations.

∎

We may now prove a soundness result, which ensures that well-formedness is preserved under LTS rules.

**Theorem 2.2.10.** If $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$ and $P \xrightarrow{\alpha} P'$ then there are $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$ such that $\Gamma'; \Delta' \vdash_{\gamma';\delta';p'} P'$.

*Proof.* The proof proceeds by induction on the depth of the derivation $P \xrightarrow{\alpha} P'$.

***<u>Base cases:</u>*** In the following we consider four base cases.

- *Case* (L-Out1): Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;p} \bar{t}.P$ and $\bar{t}.P \xrightarrow{\bar{t}} P$. By Lemma 2.2.6 Case 2) we get $\Gamma; \emptyset \vdash_{\gamma \cup \{t\};\delta;\bot} P$.
- *Case* (L-Out2)*:* Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;p} \bar{a}.P$ and $\bar{a}.P \xrightarrow{\bar{a}} P$. By Lemma 2.2.6 Case 3) it holds that $\Gamma; \emptyset \vdash_{\gamma;\delta;\bot} P$.
- *Case* (L-In): Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;p} a.P$ and $a.P \xrightarrow{a} P$. By Lemma 2.2.6 Case 4) we have that $\Gamma; \emptyset \vdash_{\gamma;\delta;\bot} P$.
- *Case* (L-Rec-Out): Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;p} t[P, Q]$, and $t[P, Q] \xrightarrow{t} \mathsf{extr}(P) \mid \langle Q \rangle$.
  - By Lemma 2.2.7 there are $\Gamma', \Delta', \gamma', \delta'$ and $p'$ such that $\Delta' \subseteq \Delta$ and the following holds: $\Gamma'; \Delta' \vdash_{\gamma';\delta';\top} P \mid \langle Q \rangle$.
  - By Lemma 2.2.9 we finally get: $\Gamma'_1; \Delta'_1 \vdash_{\gamma'_1;\delta'_1;p'_1} \mathsf{extr}(P) \mid \langle Q \rangle$ where $\Gamma'_1 \subseteq \Gamma, \Delta'_1 \subseteq \Delta$ and $\gamma'_1 \subseteq \gamma$.

***<u>Induction step</u>***: In the following we consider seven cases. We omit symmetric cases (L-Par2) and (L-Comm2).

- *Case* (L-Par1): Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;p} P_1 \mid Q$ and if the last applied rule is (L-Par1) then $P_1 \mid Q \xrightarrow{\alpha} P_1' \mid Q$ is derived from $P_1 \xrightarrow{\alpha} P_1'$.
  - By Lemma 2.2.6 Case 2) the following holds: $\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P_1$ and $\Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ such that $\Gamma = \Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2)$ (cf. (2.4)), $\Delta = \Delta_1 \cup \Delta_2, \gamma = \gamma_1 \cup \gamma_2, \delta = \delta_1 \cup \delta_2$ and $p = p_1 \vee p_2$ and condition $\Gamma^s \cap \Delta^t = \emptyset$ holds.
  - By induction hypothesis there are $\Gamma_1' \subseteq \Gamma_1$ and $\Delta_1' \subseteq \Delta_1$, such that $\Gamma_1';\Delta_1' \vdash_{\gamma_1';\delta_1';p_1'} P'$.
  - We get that condition $(\Delta_1' \cup \Delta_2)^t \cap (\Gamma_1' \cup \Gamma_2 \cup (\gamma_1' \times \gamma_2))^s = \emptyset$ holds based on basic properties of set operations.
  - For processes $P_1'$ and $Q$ by formation on Rule (W-Par) we get: $\Gamma';\Delta' \vdash_{\gamma';\delta';p'} P_1' \mid Q$ where $\Gamma' = \Gamma_1' \cup \Gamma_2 \cup (\gamma_1' \times \gamma_2), \Delta' = \Delta_1' \cup \Delta_2, \gamma = \gamma_1' \cup \gamma_2, \delta = \delta_1' \cup \delta_2$ and $p = p_1' \vee p_2$.

  Therefore, we obtain that the conditions of the theorem are satisfied.

- *Case* (L-Comm1): Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;p} P \mid Q$ and if the last applied rule is (L-Comm1) then $P \mid Q \xrightarrow{\tau} P' \mid Q'$ is derived from $P \xrightarrow{x} P'$ and $Q \xrightarrow{\overline{x}} Q'$.
  - By Lemma 2.2.6 Case 2) the following holds: $\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P$ and $\Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ such that $\Gamma = \Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2), \Delta = \Delta_1 \cup \Delta_2, \gamma = \gamma_1 \cup \gamma_2, \delta = \delta_1 \cup \delta_2$ and $p = p_1 \vee p_2$ and condition $\Gamma^s \cap \Delta^t = \emptyset$ holds.
  - For process $P'$ by induction hypothesis there are $\Gamma_1' \subseteq \Gamma_1$ and $\Delta_1' \subset \Delta_1$, such that $\Gamma_1';\Delta_1' \vdash_{\gamma_1';\delta_1';p_1'} P'$.
  - For process $Q'$ by induction hypothesis there are $\Gamma_2' \subseteq \Gamma_2$ and $\Delta_2' \subset \Delta_2$, such that $\Gamma_2';\Delta_2' \vdash_{\gamma_2';\delta_2';p_2'} Q'$.
  - We get that condition $(\Delta_1' \cup \Delta_2')^t \cap (\Gamma_1' \cup \Gamma_2' \cup (\gamma_1' \times \gamma_2'))^s = \emptyset$ holds based on basic properties of set operations.
  - For processes $P'$ and $Q'$ by formation on Rule (W-Par) we get: $\Gamma';\Delta' \vdash_{\gamma';\delta';p'} P' \mid Q'$ where $\Gamma' = \Gamma_1' \cup \Gamma_2' \cup (\gamma_1' \times \gamma_2'), \Delta = \Delta_1' \cup \Delta_2', \gamma = \gamma_1' \cup \gamma_2', \delta = \delta_1' \cup \delta_2'$ and $p = p_1' \vee p_2'$.

  Therefore, we obtain that the conditions of the theorem are satisfied.

- *Case* (L-Rep): Assume that $\gamma \times \gamma;\Delta \vdash_{\gamma;\delta;p} !\pi.P$ and if the last applied rule is (L-Rep) then $!\pi.P \xrightarrow{\alpha} P' \mid !\pi.P$ is derived from $\pi.P \xrightarrow{\alpha} P'$.
  - By Lemma 2.2.6 Case 7) we get $\Gamma';\emptyset \vdash_{\gamma;\emptyset;\bot} \pi.P$ and $\Delta = \emptyset$.
  - By induction hypothesis there is $\Gamma_1' \subseteq \Gamma'$ such that $\Gamma_1';\emptyset \vdash_{\gamma';\delta';p'} P'$.
  - For processes $!\pi.P$ and $P'$ by formation on Rule (W-Par) we get: $\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P' \mid !\pi.P$ such that $\Gamma_1 = \Gamma_1' \cup (\gamma \times \gamma_1), \Delta_1 = \emptyset, \gamma_1 = \gamma \cup \gamma', \delta_1 = \delta \cup \delta'$ and $p_1 = p \vee p'$. Condition $\Gamma_1^s \cap \Delta_1^t = \emptyset$ holds by basic properties of set operations.

  Therefore, we obtain that the conditions of the theorem are satisfied.

- *Case* (L-Rec-In): Assume that $\Gamma;\Delta \vdash_{\gamma;\delta;p} t[P,Q]$ and if the last applied rule is (L-Rec-In) then $t[P,Q] \xrightarrow{\tau} \mathsf{extr}(P') \mid \langle Q \rangle$ is derived from $P \xrightarrow{\overline{t}} P'$.
  - Using the assumption and by Lemma 2.2.6 Case 9) we get that for process $P$ the following holds: $\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P$ for some $\Gamma_1, \Delta_1, \gamma_1, \delta_1, p_1$, and for process $Q$ the following holds: $\Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ for some $\Gamma_2, \Delta_2, \gamma_2, \delta_2, p_2$.

- By induction hypothesis there are $\Gamma'_1 \subseteq \Gamma_1, \Delta'_1 \subseteq \Delta_1$ such that $\Gamma'_1; \Delta'_1 \vdash_{\gamma'_1;\delta'_1;p'_1} P'$.
- For transaction $t[P,Q]$ by Lemma 2.2.7 there are $\Delta', \delta'$ such that $\Gamma_1; \Delta' \vdash_{\gamma;\delta';\top} P \mid \langle Q \rangle$ and $\Delta' \subseteq \Delta$.
- For process $P'$ by formation on Rule (W-PAR) and by set operations' properties the following holds: $\Gamma'_2; \Delta'_2 \vdash_{\gamma'_2;\delta'_2;\top} P' \mid \langle Q \rangle$, where $\Gamma'_2 = \Gamma'_1 \cup \Gamma_2 \cup (\gamma'_1 \times \gamma_2)$, $\Delta'_2 = \Delta'_1 \cup \Delta_2$, $\gamma'_2 = \gamma'_1 \cup \gamma_2$, $\delta'_2 = \delta'_1 \cup \delta_2$.
- Applying Lemma 2.2.8 on process $\mathsf{extr}(P')$ and Applying Lemma 2.2.9 we finally get: $\Gamma''_2; \Delta''_2 \vdash_{\gamma''_2;\delta''_2;\top} \mathsf{extr}(P') \mid \langle Q \rangle$ where $\Gamma''_2 \subseteq \Gamma'_2, \Delta''_2 \subseteq \Delta'_2$, $\delta''_2 \subseteq \delta'_2$ and $\gamma''_2 \subseteq \gamma'_2$.

- *Case* (L-SCOPE-OUT): Assume that $\Gamma; \Delta \vdash_{\gamma;\{t\};p} t[P,Q]$ and if the last applied rule is (L-SCOPE-OUT) then $t[P,Q] \xrightarrow{\alpha} t[P',Q]$ is derived from $P \xrightarrow{\alpha} P'$.

  - By Lemma 2.2.6 Case 9) we get: $\Gamma_1; \Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P$ and $\Gamma_2; \Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q$ and $\Gamma = \Gamma_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2)$, $\Delta = \Delta_1 \cup \Delta_2 \cup (\{t\} \times (\delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))$, $\gamma = \gamma_1 \cup \gamma_2$, $\delta = \{t\}$ and $p = p_1 \vee p_2$, also condition $\Gamma^s \cap \Delta^t = \emptyset$ holds.
  - By inductive hypothesis there are $\Gamma'_1 \subseteq \Gamma_1, \Delta'_1 \subseteq \Delta_1$ and $\delta'_1, \gamma'_1, p'_1$ such that $\Gamma'_1; \Delta'_1 \vdash_{\gamma'_1;\delta'_1;p'_1} P'$.
  - Based on set operations' properties for $\Gamma = \Gamma'_1 \cup \Gamma_2 \cup (\gamma_1 \times \gamma_2)$ and $\Delta = \Delta'_1 \cup \Delta_2 \cup (\{t\} \times (\delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))$ the condition $\Gamma^s \cap \Delta^t = \emptyset$ holds too.
  - For process $Q$ and obtained process $P'$ by formation on Rule (W-TRANS) we get: $\Gamma'; \Delta' \vdash_{\gamma';\{t\};p'} t[P',Q]$, where $\Gamma' = \Gamma'_1 \cup \Gamma_2 \cup (\gamma'_1 \times \gamma_2)$, $\Delta' = \Delta'_1 \cup \Delta_2 \cup (\{t\} \times (\delta_1 \cup \delta_2 \cup \gamma_1 \cup \gamma_2))$, $\gamma = \gamma'_1 \cup \gamma_2$, $\delta = \{t\}$ and $p = p'_1 \vee p_2$.

  Therefore, we obtain that the conditions of the theorem are satisfied.

- *Case* (L-RES): Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;p} (\nu x)P$ and if the last applied rule is (L-RES) then $(\nu x)P \xrightarrow{\alpha} (\nu x)P'$ is derived from $P \xrightarrow{\alpha} P'$.

  - By Lemma 2.2.6 we get: $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$.
  - By inductive hypothesis there are $\Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta, \gamma', \delta', p'$ such that $\Gamma'; \Delta' \vdash_{\gamma';\delta';p'} P'$.
  - For process $P'$ by formating on Rule (W-RES) the following holds: $\Gamma'; \Delta' \vdash_{\gamma';\delta';p'} (\nu x)P'$.

- *Case* (L-BLOCK): Assume that $\Gamma; \Delta \vdash_{\gamma;\delta;\top} \langle P \rangle$ and if the last applied rule is (L-BLOCK) then $\langle P \rangle \xrightarrow{\alpha} \langle P' \rangle$ is derived from $P \xrightarrow{\alpha} P'$.

  - By Lemma 2.2.6 we get: $\Gamma; \Delta \vdash_{\gamma;\delta;p} P$.
  - By inductive hypothesis there are $\Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta, \gamma', \delta', p'$ such that $\Gamma'; \Delta' \vdash_{\gamma';\delta';p'} P'$.
  - For process $P'$ by formating on Rule (W-BLOCK) the following holds: $\Gamma'; \Delta' \vdash_{\gamma';\delta';p'} \langle P \rangle'$.

■

The following is immediate from Definition 2.2.4 and Theorem 2.2.10:

**Corollary 2.2.11.** If $P$ is a well-formed compensable process and $P \longrightarrow^* P'$ then $P'$ is also well-formed.

### 2.2.3 Adaptable Processes

Process calculi have certain limitations for the description of the pattern of dynamic evolution and adaptable processes are introduced as a way of overcoming these limitations. As stated in [7], such patterns rely on direct methods of controlling the behavior and location of running processes. Therefore, adaptable processes are at the heart of the adaptation capabilities present in many modern concurrent systems.

This section is divided into two parts:

**Section 2.2.3.1** illustrates most salient features by means of a simple example for better understanding the concept of adaptable processes introduced by Bravetti et al. [7].

**Section 2.2.3.2** presents formal description of adaptable processes, through their syntax and operational semantics.

#### 2.2.3.1 Adaptable Processes, by Example

The calculus of adaptable processes was introduced as a variant of Milner's CCS [32] (without restriction and relabeling), extended with the following two constructs, aimed at representing the dynamic reconfiguration (or update) of communicating processes:

1. A *located process*, denoted $l[P]$, represents a process $P$ which resides in a location called $l$. Locations can be arbitrarily *nested*, which allows to organize process descriptions into meaningful hierarchical structures.

2. *Update prefixes* specify an adaptation mechanism for processes at location $l$. We write $l\langle\langle(X).Q\rangle\rangle$ and $l\{(X).Q\}$ to denote subjective and objective update prefixes; in both cases, $X$ is a process variable that occurs zero or more times in $Q$.

This way, in the calculus of adaptable processes an update prefix for location $l$ can interact with a located process at $l$ to update its current behavior. Depending on the kind of prefix (objective or subjective), this interaction is realized by a reduction rule ((1.1) or (1.2), see also below).

We illustrate adaptable processes by revisiting example in Section 2.2.1.1:

**Example 2.2.12.** Consider again the hotel booking scenario in Example 2.2.1 , this time expressed using the calculus of adaptable processes (below we omit trailing $\mathbf{0}$s):

$$\begin{aligned} Reservation &\stackrel{def}{=} Hotel \mid Client \\ Client &\stackrel{def}{=} \overline{book}.\overline{pay}.(\overline{t}.refund + invoice) \\ Hotel &\stackrel{def}{=} t[book.pay.\overline{invoice}] \mid t.t\langle\langle(Y).\mathbf{0}\rangle\rangle \mid p_t[\overline{refund}] \end{aligned}$$

We use CCS processes with the located processes and (subjective) update prefixes. The client's behavior involves sending requests for booking and paying for a room, which are followed by either the reception of an invoice or an output on $t$. If the client sends output on $t$, then it follows the request for a refund. The expected behavior of the hotel is located at location $t$: the hotel allows the client to book a room and pay for it; if the client is satisfied with the reservation, the hotel will send him/her an invoice. The hotel specification includes also (i) a subjective update prefix $t\langle\langle(Y).\mathbf{0}\rangle\rangle$ (in the same way, can be used objective update $t\{(Y).\mathbf{0}\}$), which deletes the location $t$ with its content if the client is not satisfied with the reservation, and (ii) a simple refund procedure located at $p_t$, which handles the interaction with the client in that case. If the client decides to cancel his reservation, the reduction steps for process *Reservation* can be as follows:

$$Reservation \longrightarrow t[pay.\overline{invoice}] \mid t.t\langle\langle(Y).\mathbf{0}\rangle\rangle \mid p_t[\overline{refund}] \mid \overline{pay}.(\overline{t}.refund + invoice)$$

$$\begin{aligned}
&\longrightarrow\ t[\overline{invoice}] \mid t.t\langle\langle (Y).\mathbf{0}\rangle\rangle \mid p_t[\overline{refund}] \mid \overline{t}.refund + invoice\\
&\longrightarrow\ t[\overline{invoice}] \mid t\langle\langle (Y).\mathbf{0}\rangle\rangle \mid p_t[\overline{refund}] \mid refund\\
&\longrightarrow\ p_t[\overline{refund}] \mid refund\\
&\longrightarrow\ p_t[\mathbf{0}].
\end{aligned}$$

In this example, we could have used objective update $t\{(Y).\mathbf{0}\}$ instead of subjective update $t\langle\langle (Y).\mathbf{0}\rangle\rangle$; with the objective update, the behavior of process *Reservation* is quite similar. Once we formally define our translations, a detailed derivation and explanation for this scenario will be provided later on.

#### 2.2.3.2 Adaptable Processes, Formal Description

Below is a formal and detailed description of the core calculus for adaptable processes.

##### 2.2.3.2.1 Syntax

We consider *prefixes* $\pi$ and *processes* $P, Q, \ldots$ defined as:

$$\pi \quad ::= \quad \overbrace{x \mid \overline{x}}^{\text{CCS prefixes}} \quad \mid \quad \overbrace{l\langle\langle (X).Q\rangle\rangle \mid l\{(X).Q\}}^{\text{extension}}$$

$$P, Q \quad ::= \quad \underbrace{\mathbf{0} \mid \pi.P \mid\ !\pi.P \mid (\nu x)P \mid P \mid Q}_{\text{CCS processes}} \quad \mid \quad \underbrace{l[P] \mid X}_{\text{extension}}$$

We consider input and output prefixes (denoted $x$ and $\overline{x}$, respectively) as well as the update prefixes $l\langle\langle (X).Q\rangle\rangle$ and $l\{(X).Q\}$ for subjective and objective update, respectively. We assume that $Q$ may contain zero or more occurrences of the process variable $X$. Although here we consider a process model with both update prefixes, we shall consider target calculi with only one of them: the calculus of adaptable processes with subjective and objective update will be denoted $\mathcal{S}$ and $\mathcal{O}$, respectively. The syntax includes constructs for inaction ($\mathbf{0}$); action prefix ($\pi.P$); guarded replication ($!\pi.P$), i.e. infinitely many occurrences of $P$ in parallel, each of them are triggered by prefix $\pi$; restriction ($(\nu x)P$); parallel composition ($P \mid Q$); located processes ($l[P]$); and process variables ($X$). We omit $\mathbf{0}$ whenever possible; we write, e.g., $l\langle\langle (X).P\rangle\rangle$ instead of $l\langle\langle (X).P\rangle\rangle.\mathbf{0}$.

Name $x$ is bound in $(\nu x)P$ and process variable $X$ is bound in $l\langle\langle (X).Q\rangle\rangle$. Given this, the sets of free and bound names for a process $P$ — denoted by $\mathtt{fn}(P)$ and $\mathtt{bn}(P)$ — are as expected (and similarly for process variables). We rely on expected notions of $\alpha$-conversion (noted $\equiv_\alpha$) and process substitution: we denote by $P\{Q/X\}$ the process obtained by the (capture-avoiding) substitution of $Q$ for $X$ in $P$.

##### 2.2.3.2.2 Operational Semantics

Adaptable processes are governed by a reduction semantics, denoted $P \longrightarrow P'$, a relation on processes that relies on *structural congruence* (denoted $\equiv$) and *contexts* (denoted $C, D, E$).

**Definition 2.2.5** (Structural congruence)**.** Structural congruence is the smallest congruence relation on processes that is generated by the following rules, which extend standard rules for

(R-In-Out)
$$E\Big[C[\overline{x}.P] \mid D[x.Q]\Big] \longrightarrow E\Big[C[P] \mid D[Q]\Big]$$

(R-Ob-Upd)
$$E\Big[C\big[l[P]\big] \mid D\big[l\{(X).Q\}.R\big]\Big] \longrightarrow E\Big[C\big[Q\{^{P}\!/X\}\big] \mid D\big[R\big]\Big]$$

(R-Sub-Upd)
$$E\Big[C\big[l[P]\big] \mid D\big[l\langle\!\langle(X).Q\rangle\!\rangle.R\big]\Big] \longrightarrow E\Big[C\big[\mathbf{0}\big] \mid D\big[Q\{^{P}\!/X\} \mid R\big]\Big]$$

(R-Str)
$$\frac{P \equiv P' \quad P' \longrightarrow Q' \quad Q' \equiv Q}{P \longrightarrow Q}$$

Figure 2.5: Reduction semantics for adaptable processes.

the CCS calculus with scope extrusion for locations:

$$\begin{array}{ll}
P \mid Q \equiv Q \mid P & (\nu x)\mathbf{0} \equiv \mathbf{0} \\
P \mid (Q \mid R) \equiv (P \mid Q) \mid R & (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P \\
P \mid \mathbf{0} \equiv P & Q \mid (\nu x)P \equiv (\nu x)(Q \mid P) \text{ if } x \notin \mathtt{fn}(Q) \\
!\pi.P \equiv \pi.P \mid !\pi.P & (\nu x)l[P] \equiv l[(\nu x)P] \text{ if } l \neq x \\
P \equiv Q \text{ if } P \equiv_\alpha Q &
\end{array}$$

Contexts are processes with a *hole* $[\bullet]$; their syntax is defined as follows:

**Definition 2.2.6** (Evaluation Contexts)**.** The syntax of contexts is given by the following grammar:

$$C[\bullet] ::= [\bullet] \;\Big|\; l\big[C[\bullet]\big] \;\Big|\; C[\bullet] \mid P \;\Big|\; (\nu x)C[\bullet].$$

We write $C[Q]$ to denote the process resulting from filling in the hole $[\bullet]$ in context $C$ with process $Q$. Reduction $\longrightarrow$ is the smallest relation on processes induced by the rules in Figure 2.5, which we now briefly discuss:

- Rule (R-In-Out) formalizes synchronization between processes $\overline{x}.P$ and $x.Q$, enclosed in contexts $C$ and $D$, respectively.
- Rules (R-Sub-Upd) and (R-Ob-Upd) formalize the equations (1.1) and (1.2) given in the Introduction. They implement subjective and objective update of a process located at location $l$ that resides in contexts $C$ and $E$. In general, we shall use one of these two rules, not both.
- Rule (R-Str) closes the reduction relation under structural congruence.

**Remark 2.2.13.** We write $\longrightarrow^*$ to denote the reflexive and transitive closure of $\longrightarrow$.

We close this section with the following example because we want to emphasize the difference between subjective and objective updates. We note once again that the difference in the movement through updates is crucial for encodings.

**Example 2.2.14.** Let $s, t, l_1$, and $l_2$ be location names. Let us consider reduction steps for the following processes $P_1$ and $P_2$ for subjective and objective update, respectively:

- For *subjective update* we have the following two reduction steps:

$$\begin{aligned} P_1 &= t\big[l_1[a] \mid l_1[b] \mid c\big] \mid \mathtt{outd}^{\mathtt{s}}(l_1, l_2, 2, Q) \\ &= t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\langle\!\langle (X_1).l_1\langle\!\langle (X_2).(l_2[X_1] \mid l_2[X_2] \mid Q)\rangle\!\rangle\rangle\!\rangle \\ &\longrightarrow t\big[l_1[b] \mid c\big] \mid l_1\langle\!\langle (X_2).(l_2[a] \mid l_2[X_2] \mid Q)\rangle\!\rangle \\ &\longrightarrow t\big[c\big] \mid l_2[a] \mid l_2[b] \mid Q \end{aligned}$$

- For *objective update* we have the following two reduction steps:

$$\begin{aligned} P_2 &= t\big[l_1[a] \mid l_1[b] \mid c\big] \mid \mathtt{outd}^{\mathtt{o}}(l_1, l_2, 2, Q) \\ &= t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\{(X_1).l_1\{(X_2).(l_2[X_1] \mid l_2[X_2] \mid Q)\}\} \\ &\longrightarrow t\big[l_1\{(X_2).(l_2[a] \mid l_2[X_2] \mid Q)\} \mid \ \mid l_1[b] \mid c\big] \\ &\longrightarrow t\big[l_2[a] \mid l_2[b] \mid Q \mid c\big] \end{aligned}$$

The difference between subjective and objective update:

- by using subjective update, after two reduction steps, processes on location $l_1$ that are nested in location $t$, have been relocated to location $l_2$ and moved out of location $t$;
- by using objective update, after two reduction steps, we got that processes on location $l_1$, nested in location $t$, have been relocated to location $l_2$. They are still nested in location $t$. Therefore, if we want to relocate processes on location $l_2$ out of the location $t$, we need to perform at least one more reduction step.

## 2.3 Expressiveness of Concurrent Calculi

In this part of the dissertation, we provide a general survey of the most significant approaches to the expressiveness of concurrent languages. Throughout the following subsections we assume the following notation conventions. We use $\mathcal{L}_1, \mathcal{L}_2, \ldots$ to range over languages and $\mathcal{L}_{\mathbf{s}}, \mathcal{L}_{\mathbf{t}}$ to range over source and target languages, respectively.

### 2.3.1 Generalities

Calculi are subject to evaluation and an important criterion for assessing their significance is their expressiveness. The theory of concurrency lacks a distinctive argument on a formal definition of the expressive power of a language. Although a unified theory would be ideal, a large number and variety of concurrency models suggests that there is no single theory for language comparison that incorporates all of them [42].

As explained in [43], for the study of expressiveness, the notion of encoding is of crucial importance. The encoding is a function $[\![\cdot]\!]$ from the terms of a *source calculus* into the terms of a *target calculus* that satisfies correctness criteria which cover *structural* and *semantic* criteria of function $[\![\cdot]\!]$. Because many criteria arise from different practical needs, defining these criteria is the main difficulty in defining a unique theory for language comparison. Several times in the literature (e.g., in [39, 37, 40, 22]), it is stated that there is no consensus on what set of criteria constitutes a reasonable and meaningful encoding. It is common to create criteria based on the requirements of the current analysis. Also, in [40] author states that with the significant increase in the number of process models, their comparison in a systematic way represents an important aspect of research in the field.

The question of the purpose of the existence of expressiveness can be posed. In this regard, expressiveness studies usually consider: *encodability* and *non-encodability* results. Naturally, encodability is studying the existence of an encoding. The non-encodability, on the other hand, deals with the opposite problem. Let us consider languages $\mathcal{L}_1$ and $\mathcal{L}_2$, if we want to confirm that $\mathcal{L}_1$ is more expressive than $\mathcal{L}_2$, we have to provide the existence of both results: we should present/prove an encoding $[\![\cdot]\!] : \mathcal{L}_2 \longrightarrow \mathcal{L}_1$ and, simultaneously, we should provide a formal argument demonstrating that an encoding $[\![\cdot]\!] : \mathcal{L}_1 \longrightarrow \mathcal{L}_2$ does not exist.

The literature also has another classification of expressiveness. As explained in [42, 43], there are two approaches to evaluate the expressive power of a language that consider *absolute* or *relative expressiveness*:

- If we consider the expressive power of a single process calculus, we yield an *absolute result*. We usually get a *positive absolute result* by proving the ability to solve some kind of problem. Otherwise, we obtain a *negative absolute result* by demonstrating the inability to solve a problem ([40], [22]). In [40] has been explained that this question entails determining exactly the transition systems that are expressible in a given language. The term absolute expressiveness is used to describe the expressive power of a language and its semantics. As a result, the emphasis is on the expressiveness of the terms of the language, as well as the kind of operators that can be expressed in it. The answers to these questions are contingent on the use of appropriate denotations of LTS. Hence, only basic process calculi with simple labels have been reported to have absolute expressiveness ([40]).

- *Relative expressiveness*, which is studied in this dissertation, measures the expressive power of a language $\mathcal{L}_1$ by taking some language $\mathcal{L}_2$ as a reference. When we want to show that $\mathcal{L}_1$ and $\mathcal{L}_2$ have the same expressive power, we use this type of expressiveness. The goal is to get two encodability results, one for each direction. Other situation is to determine the impact of a specific operator or construct on the expressiveness of a language $\mathcal{L}_1$. The language $\mathcal{L}_2$ is the fragment of $\mathcal{L}_1$ without the operator(s) of interest. Precisely, one goal can be that we want to show that $\mathcal{L}_1$ cannot be encoded into $\mathcal{L}_2$. If this is the case then the difference in the expressive power between the two languages exists. The difference is in the operators that $\mathcal{L}_1$ has but that $\mathcal{L}_2$ lacks, and in the literature, this scenario is known as a *separation result*. It examines a construct that separates the world with it from the world without it ([56, 55]).

A more detailed historical overview of the evolution of the definition of *encoding* starting from proposal, within programming language at large and concluding with the most relevant proposals for concurrent languages can be found in [42].

### 2.3.2 The Notation of Encoding

As we previously stated, an encoding function is a function from the set of processes of the *source calculus* into the set of processes of the *target calculus*. The existence of an encoding shows that the target language is at least as expressive as the source language. Conversely, proving the non-existence of such an encoding shows that the source language can express some behavior not expressible in the target language. By combining these *encodability results* (positive and negative), the differences in expressivity between languages can be established.

Also, it is common to interpret the translation as a mapping of the syntax of the language, $\mathcal{L}_\mathbf{s}$ into the language $\mathcal{L}_\mathbf{t}$ and $\mathcal{P}_\mathbf{s}$ and $\mathcal{P}_\mathbf{t}$ are set of process terms of the source and target language, i.e., $[\![\cdot]\!] : \mathcal{P}_\mathbf{s} \longrightarrow \mathcal{P}_\mathbf{t}$. Such defined functions can be obtained by doing some trivial mappings. One trivial example can be that we define the encoding that translates every process to the inaction. This is a reasonable assumption because the inaction is a part of every process calculus. It is important to remember that such encoding says nothing about the expressive power of the considered calculi. Therefore, if we want to analyze the quality of encodings and also to rule out trivial or meaningless encodings, encodings are augmented with a set of *correctness criteria*, which attest to their quality. Overview and discussion of commonly used correctness criteria can be found in [40, 22, 35, 36, 55, 21, 43].

Our objective is to relate the calculus of compensable and the calculus of adaptable processes through *valid encodings* (simply *encodings* in the following). Here we define a basic abstract framework that will help us formalize these relations.

To define valid encodings, we adopt five correctness criteria formulated by Gorla [22]:

$$\left.\begin{array}{l}(1)\ \textit{compositionality}\\ (2)\ \textit{name invariance}\end{array}\right\}\ \textbf{\textit{structural}}\text{ criteria}$$

$$\left.\begin{array}{l}(3)\ \textit{operational correspondence}\\ (4)\ \textit{divergence reflection}\\ (5)\ \textit{success sensitiveness}\end{array}\right\}\ \textbf{\textit{semantic}}\text{ criteria}$$

Structural criteria describe the static structure of the encoding, whereas the semantic criteria describe its dynamics — how the behavior of encoded terms relates to that of source terms, and vice versa. As stated in [40], structural criteria are needed in order to measure the expressiveness of operators in contrast to expressiveness of terms. As for semantic criteria, operational correspondence is divided into *completeness* and *soundness* properties: the former ensures that the behavior of a source process is preserved by the translation in the target calculus; the latter ensures that the behavior of a translated (target) process corresponds to that of some source process. Divergence reflection ensures that a translation does not introduce spurious infinite computations, whereas success sensitiveness requires that source and translated terms behave in the same way with respect to some notion of *success*.

Following [22], we start by defining an abstract notion of calculus, which we will later instantiate with the three calculi of interest here:

**Definition 2.3.1** (Calculus)**.** We define a *calculus* as a triple $(\mathcal{P}, \longrightarrow, \approx)$, where:

- $\mathcal{P}$ is a set of processes;
- $\longrightarrow$ is its associated reduction semantics, which specifies how a process computes on its own;
- $\approx$ is an equality on processes, useful to describe the abstract behavior of a process, which is a congruence at least with respect to parallel composition.

We will further assume that a calculus uses a countably infinite set of names, usually denoted $\mathcal{N}$. Accordingly, the abstract definition of encoding refers to those names.

**Definition 2.3.2** (Encoding)**.** Let $\mathcal{N}_\mathbf{s}$ and $\mathcal{N}_\mathbf{t}$ be countably infinite sets of source and target names, respectively. An *encoding* of the source calculus $(\mathcal{P}_\mathbf{s}, \longrightarrow_\mathbf{s}, \approx_\mathbf{s})$ into the target calculus $(\mathcal{P}_\mathbf{t}, \longrightarrow_\mathbf{t}, \approx_\mathbf{t})$ is a tuple $([\![\cdot]\!], \varphi_{[\![\cdot]\!]})$ where $[\![\cdot]\!] : \mathcal{P}_\mathbf{s} \longrightarrow \mathcal{P}_\mathbf{t}$ denotes a *translation* and $\varphi_{[\![\cdot]\!]} : \mathcal{N}_\mathbf{s} \longrightarrow \mathcal{N}_\mathbf{t}$ denotes a *renaming policy* for $[\![\cdot]\!]$.

The renaming policy defines the way names from the source calculus are translated into the target calculus. A valid encoding cannot depend on the particular names involved in source processes.

We shall use the following notations. We write $\longrightarrow^*$ to denote the reflexive, transitive closure of $\longrightarrow$. Also, given $k \geq 1$, we will write $P \longrightarrow^k P'$ to denote $k$ consecutive reduction steps leading from $P$ to $P'$. That is, $P_1 \longrightarrow^k P_{k+1}$ holds whenever there exist $P_2, \ldots, P_k$ such that $P_1 \longrightarrow P_2 \longrightarrow \cdots \longrightarrow P_k \longrightarrow P_{k+1}$.

For compositionality, we use a context to combine the translated subterms, which depends on the source operator that combines the subterms. This context is parametrized on a finite set of names, noted $\mathtt{N}$ below, which contains the set of free names of the respective source term. In a slight departure from usual definitions of compositionality, the set $\mathtt{N}$ may contain transaction names that do not occur free in the term. As we will see, we have an initially empty parameter on the encoding function that is accumulated while translating a source term.

For the operational correspondence we follow more strict criteria than Gorla in [22]. We rely on a form of operational completeness that, unlike Gorla's, explicitly describes the number of steps required to mimic a step in the source language.

For divergence reflection we will use the following definition:

**Definition 2.3.3** (Divergence)**.** A process $P$ diverges, written $P \longrightarrow^{\omega}$, if there exists an infinite sequence of processes $\{P_i\}_{i\geq 0}$ such that $P = P_0$ and for any $i$, $P_i \longrightarrow P_{i+1}$.

To formulate success sensitiveness, we assume that both source and target calculi contain the same success process $\checkmark$; also, we assume that $\Downarrow$ is a predicate that asserts reducibility (in a "may" modality) to a process containing an unguarded occurrence of $\checkmark$. This process operator does not affect the operational semantics and behavioral equivalence of the calculi: $\checkmark$ cannot reduce and $\mathtt{n}(\checkmark) = \mathtt{fn}(\checkmark) = \mathtt{bn}(\checkmark) = \emptyset$. Therefore, this language extension does not affect the validity of the encodability criteria.

**Definition 2.3.4** (Success)**.** Let $(\mathcal{P}, \longrightarrow, \approx)$ be a calculus. A process $P \in \mathcal{P}$ (may)-succeeds, denoted $P \Downarrow$, if it is reducible to a process containing an unguarded occurrence of $\checkmark$, i.e., if $P \longrightarrow^{*} P'$ and $P' = C[\checkmark]$ for some $P'$ and context $C[\bullet]$.

In the following definition, we formally present the five criteria for valid encoding where n-adic context $C[\bullet_1, \ldots, \bullet_n]$ is a term such that $n$ occurrences of $\mathbf{0}$ are replaced by the holes $[\bullet_1], \ldots, [\bullet_n]$:

**Definition 2.3.5** (Valid Encoding)**.** Let $\mathcal{L}_{\mathbf{s}} = (\mathcal{P}_{\mathbf{s}}, \longrightarrow_{\mathbf{s}}, \approx_{\mathbf{s}})$ and $\mathcal{L}_{\mathbf{t}} = (\mathcal{P}_{\mathbf{t}}, \longrightarrow_{\mathbf{t}}, \approx_{\mathbf{t}})$ be source and target calculi, respectively, each with countably infinite sets of names $\mathcal{N}_{\mathbf{s}}$ and $\mathcal{N}_{\mathbf{t}}$. An encoding $(\llbracket\cdot\rrbracket, \varphi_{\llbracket\cdot\rrbracket})$, where $\llbracket\cdot\rrbracket : \mathcal{P}_{\mathbf{s}} \longrightarrow \mathcal{P}_{\mathbf{t}}$ and $\varphi_{\llbracket\cdot\rrbracket} : \mathcal{N}_{\mathbf{s}} \longrightarrow \mathcal{N}_{\mathbf{t}}$, is a *valid encoding* if it satisfies the following criteria:

(1) **Compositionality**: $\llbracket\cdot\rrbracket$ is *compositional* if for every $n$-ary ($n \geq 1$) operator $\mathtt{op}$ on $\mathcal{P}_s$ and for every set of names $\mathtt{N}$ there is an $n$-adic context $C^{\mathtt{N}}_{\mathtt{op}}[\bullet_1, \ldots, \bullet_n]$ such that, for all $P_1, \ldots, P_n$ with $\mathtt{fn}(P_1, \ldots, P_n) \subseteq \mathtt{N}$ it holds that $\llbracket \mathtt{op}(P_1, \ldots, P_n) \rrbracket = C^{\mathtt{N}}_{\mathtt{op}}\left[\llbracket P_1 \rrbracket, \ldots, \llbracket P_n \rrbracket\right]$.

(2) **Name invariance**: $\llbracket\cdot\rrbracket$ is *name invariant* if for every substitution $\sigma : \mathcal{N}_{\mathbf{s}} \longrightarrow \mathcal{N}_{\mathbf{s}}$ there is a substitution $\sigma' : \mathcal{N}_{\mathbf{t}} \longrightarrow \mathcal{N}_{\mathbf{t}}$ such that (i) for every $a \in \mathcal{N}_{\mathbf{s}} : \varphi_{\llbracket\cdot\rrbracket}(\sigma(a)) = \sigma'(\varphi_{\llbracket\cdot\rrbracket}(a))$ and (ii) $\llbracket \sigma(P) \rrbracket = \sigma'(\llbracket P \rrbracket)$.

(3) **Operational correspondence**: $\llbracket\cdot\rrbracket$ is *operational corresponding* if it satisfies the two requirements:

   a) **Completeness**: If $P \longrightarrow_{\mathbf{s}} Q$ then there exists $k$ such that $\llbracket P \rrbracket \longrightarrow^{k}_{\mathbf{t}} \approx_{\mathbf{t}} \llbracket Q \rrbracket$.

   b) **Soundness**: If $\llbracket P \rrbracket \longrightarrow^{*}_{\mathbf{t}} R$ then there exists $P'$ such that $P \longrightarrow^{*}_{\mathbf{s}} P'$ and $R \longrightarrow^{*}_{\mathbf{t}} \approx_{\mathbf{t}} \llbracket P' \rrbracket$.

(4) **Divergence reflection**: $\llbracket\cdot\rrbracket$ *reflects divergence* if, for every $P$ such that $\llbracket P \rrbracket \longrightarrow^{\omega}_{\mathbf{t}}$, it holds that $P \longrightarrow^{\omega}_{\mathbf{s}}$.

(5) **Success sensitiveness**: $\llbracket\cdot\rrbracket$ is *success sensitive* if, for every $P \in \mathcal{P}_{\mathbf{s}}$, it holds that $P \Downarrow$ if and only if $\llbracket P \rrbracket \Downarrow$.

##### 2.3.2.0.1 Concrete Instances

We now instantiate Definition 2.3.1 with the source and target calculi of interest:

**Source Calculus: $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$** The source calculus will be the calculus of compensable processes defined in § 2.2.1. The set of processes, which we denote $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$ will contain only *well-formed* compensable processes (cf. Section 2.2.2). We shall consider the reduction relation $\longrightarrow$ defined at the end of Section 2.2.1. We shall use structural congruence (Definition 2.2.2) as behavioral equivalence.

**Target Calculi: $\mathcal{S}$ and $\mathcal{O}$** There will be two target calculi, both based on the calculus of adaptable processes defined in Section 2.2.3. The first one, with set of processes denoted $\mathcal{S}$, uses subjective updates only; its reduction semantics is as given in Figure 2.5, with updates governed by Rule (R-Sub-Upd). Similarly, the second calculus, with set of processes denoted $\mathcal{O}$, uses objective updates only; its reduction semantics is governed by Rule (R-Ob-Upd) instead. In both cases, the structural congruence of Definition 2.2.5 will be used as behavioral equivalence.

In the definition of operational correspondence (Definition 2.3.5 (3)), the purpose of $\approx_{\mathtt{t}}$ is to abstract away from "junk" processes, i.e., processes left behind as a result of the translation that do not add any meaningful behavior to translated processes. As we will see, in our translations the inactive process $\mathbf{0}$ will be the only possible junk process; as such, it is *inactive junk* in the sense that it does not perform further reductions on its own nor interact with the surrounding target terms. This is why it suffices to use structural congruences on source and target processes as behavioral equalities.

**Brief summary of the chapter**:
This chapter introduced: (i) basic concepts and terminology; (ii) core calculi for compensable and adaptable processes; (iii) a class of well-formed compensable processes that disallows non-deterministic interactions involving nested transactions and error notification names; (iv) an overview of expressiveness studies and the techniques utilized to conduct them.

In the next chapter, we introduce all preliminaries for encodings $\mathcal{C}$ into $\mathcal{A}$. It also presents the encoding compensable into adaptable processes with subjective update (encodings $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$).

# CHAPTER 3

# Encoding Compensable into Adaptable Processes with Subjective Update

The main purpose of this chapter is to present *valid encodings* of calculus for compensable processes into calculus of adaptable processes with subjective update. In the following, a brief structure of the chapter is given:

**Section 3.1** introduces all preliminaries for *encodings* $\mathcal{C}$ *into* $\mathcal{A}$.

**Section 3.2** at the beginning introduces the translation of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ in an informal way to acquaint the reader with the basic intuition. Then it follows the formal definition of encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$. The main contribution is a *valid encoding* $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$.

**Section 3.3** presents the translation of $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$ informally and formally. Then it follows the formal definition of encoding $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$. We prove that the encoding satisfies *name invariance* and *operational correspondence (completeness and soundness)*.

**Section 3.4** presents the translation of $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ informally, then goes on to give a formal definition of the encoding. We prove that encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ satisfies the following criteria: *compositionality*, *name invariance*, and *operational correspondence (completeness and soundness)*.

## 3.1 Preliminaries

Recall the base sets defined in Definition 2.2.1; in particular, $\mathcal{N}_t$ denotes the base set of transaction names. Encodings of $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ rely on the following notion of *path*, a sequence of transaction names:

**Definition 3.1.1** (Paths)**.** Let $\mathcal{N}_t^k$ (with $k \in \mathbb{N}$) be the set of sequences/tuples of length $k$ of names in $\mathcal{N}_t$. These sequences will be denoted by $\mu, \mu', \ldots$; we assume they have pairwise distinct elements. We write $\varepsilon$ to denote the empty path. The concatenation of such sequences with $\varepsilon$ at the end will be referred to as *paths* and denoted by $\rho, \rho', \ldots$ (i.e., $\rho = \mu\varepsilon, \rho' = \mu'\varepsilon, \ldots$). We will sometimes omit writing the tail $\varepsilon$ in $\rho$ (i.e., $\mu\varepsilon = \mu, \mu'\varepsilon = \mu', \ldots$). By a slight abuse of notation, given a transaction name $t$ and a path $\rho$, we will write $t \in \rho$ if $t$ occurs in $\rho$.

We also require sets of *reserved names*. We have the following definition:

**Definition 3.1.2** (Reserved Names)**.** The sets of *reserved names* $\mathcal{N}_s^r$ and $\mathcal{N}_l^r$ are defined as follows:

- $\mathcal{N}_s^r = \{h_x, j_x, r_x, k_x \mid x \in \mathcal{N}_t\}$ is the set of *reserved synchronization names*, and

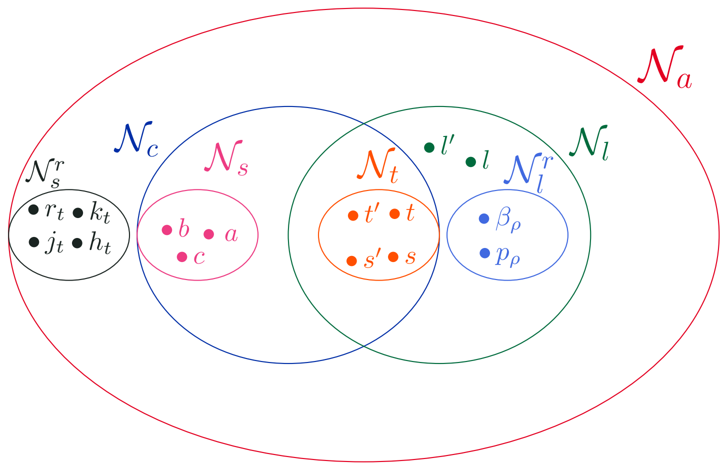



Figure 3.1: Base sets of names and reserved names.

- $\mathcal{N}_l^r = \{p_\rho, \beta_\rho \mid \rho \text{ is a path}\}$ is the set of *reserved location names.*

If $t_1, t_2 \in \mathcal{N}_t$ such that $t_1 \neq t_2$ then $h_{t_1} \neq h_{t_2}$, $p_{t_1} \neq p_{t_2}$ and $\beta_{t_1} \neq \beta_{t_2}$. We let $\mathcal{N}_l^r \subseteq \mathcal{N}_l \setminus \mathcal{N}_t$ and $\mathcal{N}_s \cap \mathcal{N}_s^r = \emptyset$. In what follows we shall use the set $\mathcal{N}_a = \mathcal{N}_l \cup (\mathcal{N}_s \cup \mathcal{N}_s^r)$ for adaptable processes.

Figure 3.1 illustrates base sets of names that are given in Definition 3.1.2.
We will find it convenient to adopt the following abbreviations for adaptable processes.

**Convention 3.1.1.** Recall that $l\langle\!\langle (X).Q \rangle\!\rangle$ and $l\{(X).Q\}$ denote subjective and objective update prefixes, respectively.

- We write $\prod_{i=1}^{n} l[X_i]$ to abbreviate the process $l[X_1] \mid \ldots \mid l[X_n]$.

- We write $t\langle\!\langle \dagger \rangle\!\rangle$ to denote the subjective update prefix $t\langle\!\langle (Y).\mathbf{0} \rangle\!\rangle$, which "kills" both location $t$ and the process it hosts. This way, for instance:

$$s[t[c]] \mid t\langle\!\langle \dagger \rangle\!\rangle \longrightarrow s[\mathbf{0}] \tag{3.1}$$

  Similarly, we write $t\{\dagger\}$ to stand for the objective update prefix that "kills" $t$ and its content.

- We write $t\langle\!\langle (Y_1, Y_2, \ldots, Y_n).R \rangle\!\rangle$ to abbreviate the *nested update prefix*:

$$t\langle\!\langle (Y_1).t\langle\!\langle (Y_2).\cdots.t\langle\!\langle (Y_n).R \rangle\!\rangle \cdots \rangle\!\rangle \rangle\!\rangle.$$

  For instance:

$$s\Big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\langle\!\langle (X_1, X_2).(l_2[X_1] \mid l_2[X_2]) \rangle\!\rangle\Big] \longrightarrow^* s\Big[t\big[c\big] \mid l_2[a] \mid l_2[b]\Big].$$

  Similarly, $t\{(Y_1, Y_2, \ldots, Y_n).R\}$ will stand for the objective prefix:

$$t\{(Y_1).t\{(Y_2).\cdots.t\{(Y_n).R\}\cdots\}\}.$$

Below, this chapter contains three sections. They present the encoding source calculus presented in [29] with *static recovery* and *discarding* (Section 3.2), *preserving* (Section 3.3) and *aborting* (Section 3.4) semantics into adaptable processes.

## 3.2 Translating $\mathcal{C}_D$ into $\mathcal{S}$

In this section we concentrate on a specific source calculus, namely the calculus in [29] with *static recovery* and *discarding semantics*. Before giving a formal presentation of the encoding we introduce some useful conventions and intuitions.

### 3.2.1 The Translation, Informally

In compensable processes: transactions, protected blocks, and the extraction function (cf. Figure 2.2) represent the most interesting process terms to be addressed in encodings.

We shall use paths (cf. Definition 3.1.1) to represent the hierarchical structure induced by nested transactions. That way, we can represent and trace the location of the transactions and protected blocks in a process. Our translation of $\mathcal{C}_D$ into $\mathcal{S}$ will be indexed by a path $\rho$: it will be denoted $[\![\cdot]\!]_\rho$ (cf. Definition 3.2.3 below). This way, e.g., the encoding of a protected block found at path $\rho$ will be defined as:

$$[\![\langle P\rangle]\!]_\rho = p_\rho\big[[\![P]\!]_\varepsilon\big]$$

where $p_\rho$ is a reserved name in $\mathcal{N}_l^r$ (cf. Definition 3.1.2).

A key aspect in our translation is the representation of the extraction function. As we have seen, this function is part of the operational semantics and formalizes the protection of transactions/protected blocks. Our translation explicitly specifies the extraction function by means of update prefixes. We use the auxiliary process $\texttt{outd}^{\texttt{s}}(l_1, l_2, n, Q)$, which moves $n$ processes from location $l_1$ to location $l_2$, and composes $Q$ in parallel. Using the notations from Convention 3.1.1, this auxiliary process can be defined as follows:

$$\texttt{outd}^{\texttt{s}}(l_1, l_2, n, Q) = \begin{cases} Q & \text{if } n = 0 \\ l_1\langle\!\langle (X_1, \ldots, X_n).\big(\prod_{i=1}^{n} l_2[X_i] \mid Q\big)\rangle\!\rangle & \text{if } n > 0 \end{cases} \tag{3.2}$$

**Example 3.2.1.** Consider the process:

$$s\big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid \texttt{outd}^{\texttt{s}}(l_1\,,\, l_2, 2, Q)\big]$$

We have two reductions:

$$\begin{aligned} & s\big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\langle\!\langle (X_1, X_2).(l_2[X_1] \mid l_2[X_2] \mid Q)\rangle\!\rangle\big] \\ \longrightarrow\ & s\big[t\big[l_1[b] \mid c\big] \mid l_1\langle\!\langle (X_2).(l_2[a] \mid l_2[X_2] \mid Q)\rangle\!\rangle\big] \\ \longrightarrow\ & s\big[t\big[c\big] \mid l_2[a] \mid l_2[b] \mid Q\big]. \end{aligned}$$

The first reduction corresponds to the synchronization between $l_1[a]$ and $l_1\langle\!\langle (X_1, X_2).(l_2[X_1] \mid l_2[X_2] \mid Q)\rangle\!\rangle$, while the second is the synchronization between $l_1[b]$ and $l_1\langle\!\langle (X_2).(l_2[a] \mid l_2[X_2] \mid Q)\rangle\!\rangle$. Figure 3.2 depicts these interactions using boxes to denote nested locations.

The auxiliary process $\texttt{outd}^{\texttt{s}}(l_1, l_2, n, Q)$ will be used in Definition 3.2.2 — see next.

### 3.2.2 The Translation, Formally

Our translation of compensable processes into adaptable processes relies on an auxiliary process, denoted $\texttt{extrd}\langle\!\langle t, l_1, l_2\rangle\!\rangle$, that explicitly represents the extraction function. Its definition uses some additional functions, which we present below:

**Definition 3.2.1.** Let $P$ be a closed adaptable process.

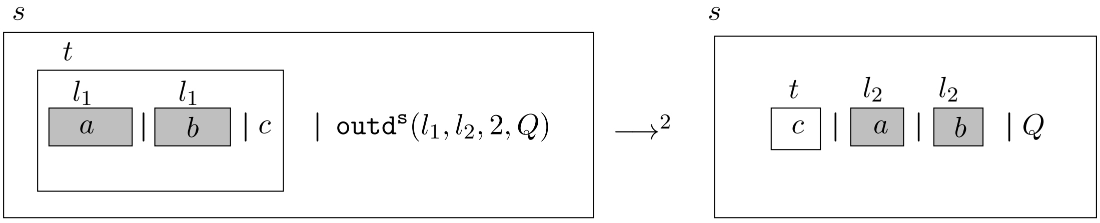



Figure 3.2: Illustrating $\mathtt{outd^s}(l_1, l_2, 2, Q)$ in Example 3.2.1.

1. Function $\mathtt{nl}(l, P)$ denotes the number of locations $l$ in process $P$. It is defined as follows:

$$\mathtt{nl}(l, l[P]) = \mathtt{nl}(l, P) + 1 \qquad \mathtt{nl}(l_1, l_2[P]) = \mathtt{nl}(l_1, P) \text{ if } l_1 \neq l_2$$
$$\mathtt{nl}(l, (\nu x)\, P) = \mathtt{nl}(l, P) \qquad \mathtt{nl}(l, P \mid Q) = \mathtt{nl}(l, P) + \mathtt{nl}(l, Q)$$
$$\mathtt{nl}(l, \mathbf{0}) = \mathtt{nl}(l, X) = 0 \qquad \mathtt{nl}(l, \pi.P) = \mathtt{nl}(l, !\pi.P) = 0$$

2. For a name $t$ and a process $P$, function $\mathsf{ch}(t, P)$ returns $h_t.\mathbf{0}$ if $P$ equals to an evaluation context with the hole replaced by $h_t.P'$ up to structural congruence (for some $P'$), where the hole is not located within $p_{t,\rho}$, and returns $\mathbf{0}$ otherwise. It is defined as follows:

$$\mathsf{ch}(t, h_t.P) = h_t.\mathbf{0} \qquad \mathsf{ch}(s, h_t.P) = \mathbf{0}$$
$$\mathsf{ch}(t, l[P]) = \begin{cases} \mathbf{0} & \text{if } l = p_{t,\rho} \\ \mathsf{ch}(t, P) & \text{otherwise} \end{cases} \qquad \mathsf{ch}(t, P \mid Q) = \mathsf{ch}(t, P) \mid \mathsf{ch}(t, Q)$$
$$\mathsf{ch}(t, !\pi.P) = \mathbf{0} \qquad \mathsf{ch}(t, \mathbf{0}) = \mathsf{ch}(t, X) = \mathbf{0}$$
$$\mathsf{ch}(t, \pi.P) = \mathbf{0} \text{ if } \pi \neq h_t \qquad \mathsf{ch}(t, (\nu x)P) = \mathsf{ch}(t, P)$$

We assume that functions $\mathtt{nl}(\cdot,\cdot)$ and $\mathsf{ch}(\cdot,\cdot)$ operate only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that they are applied once they are provided with an argument. We are now ready to define process $\mathtt{extrd}\langle\langle t, l_1, l_2\rangle\rangle$:

**Definition 3.2.2** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extrd}\langle\langle t, l_1, l_2\rangle\rangle$ to stand for the following (subjective) update prefix:

$$\mathtt{extrd}\langle\langle t, l_1, l_2\rangle\rangle = t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t, Y) \mid \mathtt{outd^s}(l_1, l_2, \mathtt{nl}(l_1, Y), t\langle\langle \dagger\rangle\rangle.\overline{h_t})\rangle\rangle. \tag{3.3}$$

Intuitively, process $\mathtt{extrd}\langle\langle t, l_1, l_2\rangle\rangle$ serves to "prepare the ground" for the use of $\mathtt{outd^s}(l_1, l_2, n, Q)$ which is the one that actually extracts processes from one location and relocates them into another one. Once that occurs, location $t$ is destroyed, which is signaled using name $h_t$.

We are now ready to formally define the translation of $\mathcal{C}_D$ into $\mathcal{S}$.

**Definition 3.2.3** (Translating $\mathcal{C}_D$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes into subjective adaptable processes as a tuple $(\llbracket\cdot\rrbracket_\rho, \varphi_{\llbracket\cdot\rrbracket_\rho})$ where:

(a) The renaming policy $\varphi_{\llbracket\cdot\rrbracket_\rho} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined as:

$$\varphi_{\llbracket\cdot\rrbracket_\rho}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{3.4}$$

$$
\begin{aligned}
[\![\langle P\rangle]\!]_\rho &= p_\rho[[\![P]\!]_\varepsilon] \\
[\![t[P,Q]]\!]_\rho &= t\Big[[\![P]\!]_{t,\rho}\Big] \mid t.\,(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}\,,\, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]) \\
[\![a.P]\!]_\rho &= a.[\![P]\!]_\rho \\
[\![\overline{a}.P]\!]_\rho &= \overline{a}.[\![P]\!]_\rho \\
[\![\,\overline{t}.P\,]\!]_\rho &= \overline{t}.h_t.[\![P]\!]_\rho \\
[\![\mathbf{0}]\!]_\rho &= \mathbf{0} \\
[\![(\nu x)P]\!]_\rho &= (\nu x)[\![P]\!]_\rho \\
[\![P_1 \mid P_2]\!]_\rho &= [\![P_1]\!]_\rho \mid [\![P_2]\!]_\rho \\
[\![!\pi.P]\!]_\rho &= ![\![\pi.P]\!]_\rho
\end{aligned}
$$

Figure 3.3: Translating $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$.

(b) The translation $[\![\cdot]\!]_\rho : \mathcal{C}_{\mathtt{D}} \to \mathcal{S}$ is as in Figure 3.3.

Some intuitions are in order. The renaming policy focuses on transaction names: if $x$ is a transaction name, then it is mapped into the set of all (reserved) names that depend on it, including reserved names whose indexed path mentions $x$. Otherwise, $x$ is mapped into the singleton set $\{x\}$.

We now explain the process mapping in Figure 3.3, which is parametric into a path $\rho$ that records the hierarchical structure induced by nested transactions. This way, top level process $P \in \mathcal{C}_{\mathtt{D}}$ is translated as $[\![P]\!]_\varepsilon$.

Unsurprisingly, the main challenge in the translation is in representing transactions and protected blocks as adaptable processes. More in details:

- The translation of a protected block found at path $\rho$ will be enclosed in the location $p_\rho$.
- In the translation of $t[P,Q]$ we represent processes $P$ and $Q$ independently, using processes in separate locations. More in details:
  - The default activity $P$ is enclosed in a location $t$ while the compensation activity $Q$ is enclosed in a location $p_\rho$. That is, $Q$ is immediately treated as a protected block.
  - The translation of $P$ is obtained with respect to path $t, \rho$, thus denoting that $t$ occurs nested within the transactions described by $\rho$.
  - In case of a failure signal $\bar{t}$, our translation activates process $\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle$ (cf. Definition 3.2.2): it extracts all processes located at $p_{t,\rho}$ (which correspond to translations of protected blocks) and moves them to their parent location $p_\rho$.
  - The structure of a transaction and the number of its top-level processes change dynamically. Whenever we need to extract processes located at $p_{t,\rho}$, we first substitute $Y$ in process $\mathtt{out}^{\mathtt{s}}$ (cf. (3.2)) and in functions $\mathtt{ch}(t,\cdot)$ and $\mathtt{nl}(l_1,\cdot)$ (cf. Definition 3.2.1), by the current content of the location $t$.
  - We use the reserved name $h_t$ (introduced by $\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle$) to control the execution of failure signals; it is particularly useful for error notifications that occur sequentially (one after another in the form of a prefix, e.g. $\bar{t}.\overline{t_1}.\dots.\overline{t_n}$).
  - Once the translation of protected blocks has been moved out of $t$, the location only contains "garbage": we can then erase the location $t$ and its contents. To this end, we use the prefix $t\langle\!\langle\dagger\rangle\!\rangle$ (cf. Convention 3.1.1), which is also introduced by $\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle$.

- In case of an internal error notification $\bar{t}$, function $\mathtt{ch}(t,\cdot)$ does the following: it searches for processes of the form $h_t.P$ within the current content at $t$ and replaces them with $h_t.\mathbf{0}$. This is done before the update prefix $t\langle\langle\dagger\rangle\rangle$ deletes both location $t$ and processes located at $t$, as described above. Notice that we would need to preserve synchronizations between input $h_t$ and its corresponding output $\overline{h_t}$.

With the above intuitions, translations for the remaining constructs should be self-explanatory.

### 3.2.3 Translation Correctness

We now establish that the translation $[\![\cdot]\!]_\rho$ is a valid encoding. To this end, we address the five criteria in Definition 2.3.5: compositionality, name invariance, operational correspondence, divergence refection, and success sensitiveness.

Our results apply for well-formed processes as in Definition 2.2.4. Consider $P = t_1\big[a \mid t_2[b,\bar{b}],\bar{a}\big] \mid \overline{t_1} \mid \overline{t_2}$, the ill-formed process presented in (2.2). Intuitively, $P$ is not well-formed because it can either compensate $t_1$ or $t_2$ in a non-deterministic fashion:

- if $t_1$ is compensated then the failure signal on $t_2$ will not be able to synchronize;

- if $t_2$ is compensated then $t_1$ can still be compensated.

That is, $P \longrightarrow^* \langle\bar{b}\rangle \mid \langle\bar{a}\rangle$. Consider how this possibility would be mimicked by $[\![P]\!]_\epsilon$, the encoding of $P$:

$$
\begin{aligned}
[\![P]\!]_\epsilon = {} & t_1\big[a \mid t_2\big[b\big] \mid t_2.\big(t_2\langle\langle(Y).t_2[Y] \mid \mathtt{ch}(t_2,Y) \mid t_2\langle\langle\dagger\rangle\rangle.\overline{h_{t_2}}\rangle\rangle \mid p_{t_1}[\bar{b}]\big)\,\big] \\
& \mid t_1.\big(t_1\langle\langle(Y).t_1[Y] \mid \mathtt{ch}(t_1,Y) \mid \mathtt{outd^s}(p_{t_1},p_\epsilon,\mathtt{nl}(p_{t_1},Y),t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}})\rangle\rangle \mid p_\epsilon[\bar{a}]\big) \\
& \mid \overline{t_1}.h_{t_1} \mid \overline{t_2}.h_{t_2} \\
\longrightarrow {} & t_1\big[a \mid t_2\big[b\big] \mid t_2.\big(t_2\langle\langle(Y).t_2[Y] \mid \mathtt{ch}(t_2,Y) \mid \overline{h_{t_2}}\rangle\rangle \mid p_{t_1}[\bar{b}]\big)\,\big] \\
& \mid t_1.\big(t_1\langle\langle(Y).t_1[Y] \mid \mathtt{ch}(t_1,Y) \mid \mathtt{outd^s}(p_{t_1},p_\epsilon,\mathtt{nl}(p_{t_1},Y),t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}})\rangle\rangle \mid p_\epsilon[\bar{a}]\big) \\
& \mid \overline{t_1}.h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & t_1\big[a \mid t_2\big[b\big] \mid t_2\langle\langle(Y).t_2[Y] \mid \mathtt{ch}(t_2,Y) \mid t_2\langle\langle\dagger\rangle\rangle.\overline{h_{t_2}}\rangle\rangle \mid p_{t_1,\epsilon}[\bar{b}]\big] \\
& \mid t_1\langle\langle(Y).t_1[Y] \mid \mathtt{ch}(t_1,Y) \mid \mathtt{outd^s}(p_{t_1},p_\epsilon,\mathtt{nl}(p_{t_1},Y),t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}})\rangle\rangle \mid p_\epsilon[\bar{a}] \mid h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & t_1\big[a \mid t_2[b] \mid \mathtt{ch}(t_2,b) \mid t_2\langle\langle\dagger\rangle\rangle.\overline{h_{t_2}} \mid p_{t_1,\epsilon}[\bar{b}]\big] \\
& \mid t_1\langle\langle(Y).t_1[Y] \mid \mathtt{ch}(t_1,Y) \mid \mathtt{outd^s}(p_{t_1},p_\epsilon,\mathtt{nl}(p_{t_1},Y),t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}})\rangle\rangle \mid p_\epsilon[\bar{a}] \mid h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & t_1\big[a \mid t_2[b] \mid t_2\langle\langle\dagger\rangle\rangle.\overline{h_{t_2}} \mid p_{t_1,\epsilon}[\bar{b}]\big] \mid p_{t_1,\varepsilon}\langle\langle(X_1).\big(p_\varepsilon[X_1] \mid t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}}\big)\rangle\rangle \mid p_\epsilon[\bar{a}] \mid h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & t_1\big[a \mid t_2[b] \mid t_2\langle\langle\dagger\rangle\rangle.\overline{h_{t_2}}\big] \mid p_\varepsilon[\bar{b}] \mid t_1\langle\langle\dagger\rangle\rangle.\overline{h_{t_1}} \mid p_\epsilon[\bar{a}] \mid h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & p_\varepsilon[\bar{b}] \mid \overline{h_{t_1}} \mid p_\epsilon[\bar{a}] \mid h_{t_1} \mid h_{t_2} \\
\longrightarrow {} & p_\epsilon[\bar{b}] \mid p_\epsilon[\bar{a}] \mid h_{t_2}
\end{aligned}
$$

Hence, when applied into ill-formed processes, encoding induces target processes with "garbage processes" (such as $h_{t_2}$ above), which do not satisfy operational correspondence as defined in Definition 2.3.5. Specifically, the soundness property would not hold, because $[\![P]\!]_\epsilon$ would have behaviors not present in $P$. A similar conclusion can be drawn for the other two ill-formed processes presented in (2.2).

#### 3.2.3.1 Structural Criteria

We prove the two criteria, following the order in which they were introduced in Definition 2.3.5, i.e., compositionality and name invariance.

#### 3.2.3.1.1 Compositionality

Compositionality is not a trivial criterion at all. It should be borne in mind that well-known encoding from the asynchronous $\pi$-calculus into the join-calculus in [19] is not compositional. Also, in the recent work [49] is presented that the encoding from $\pi$-calculus into $C_\pi$ calculus is not compositional.

The compositionality criterion says that the translation of a composite term must be defined in terms of the translations of its subterms. The translation is initially parametrized with $\varepsilon$ (i.e., without external names); afterwards, when applied to nested subterms, the list of parameters is extended with transaction names $\rho$, as specified in Definition 2.3.5. Accordingly, we consider compositional contexts that depend on an arbitrary list $\rho$ of external transaction names. Nevertheless, encoding still preserves the main principles of the notion of compositionality. We can translate compensable terms by translating their operator without need to analyze the structure of the subterms. Another peculiarity appears in the process $\mathtt{extrd}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle$, which is defined in Definition 3.2.2. It depends on the function $\mathtt{nl}(l_1, Y)$ that dynamically counts the current number of locations $l_1$ in the content of $t$. To mediate between these translations of subterms, we define a *context* for each process operator, which depends on free names of the subterms:

**Definition 3.2.4** (Compositional context for $\mathcal{C}_{\mathtt{D}}$). For every process operator from $\mathcal{C}_{\mathtt{D}}$, we define a compositional context in $\mathcal{S}$ as follows:

$$
\begin{array}{ll}
C_{\langle\rangle,\rho}[\bullet] = p_\rho\big[[\bullet]\big] & C_{t[,],\rho}[\bullet_1,\bullet_2] = t\big[[\bullet_1]\big] \mid t.\big(\mathtt{extrd}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle \mid p_\rho\big[[\bullet_2]\big]\big) \\
C_{\mid}[\bullet_1,\bullet_2] = [\bullet_1] \mid [\bullet_2] & C_{\overline{t}.}[\bullet] = \overline{t}.h_t.[\bullet] \\
C_{\overline{a}.}[\bullet] = \overline{a}.[\bullet] & C_{a.}[\bullet] = a.[\bullet] \\
C_{(\nu x)}[\bullet] = (\nu x)[\bullet] & C_{!\pi.}[\bullet] = !\pi.[\bullet]
\end{array}
$$

Using this definition, we may now state the following result:

**Theorem 3.2.2** (Compositionality for $[\![\cdot]\!]_\rho$). Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}_{\mathtt{D}}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$
\begin{array}{lll}
[\![\langle P\rangle]\!]_\rho = C_{\langle\rangle,\rho}\,[[\![P]\!]_\varepsilon] & [\![t[P,Q]]\!]_\rho = C_{t[,],\rho}\,[[\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon] & [\![P \mid Q]\!]_\rho = C_{\mid}\,[[\![P]\!]_\rho, [\![Q]\!]_\rho] \\
[\![a.P]\!]_\rho = C_{a.}\,[[\![P]\!]_\rho] & [\![\overline{t}.P]\!]_\rho = C_{\overline{t}.}\,[[\![P]\!]_\rho] & [\![(\nu x)P)]\!]_\rho = C_{(\nu x)}\,[[\![P]\!]_\rho] \\
[\![\overline{a}.P]\!]_\rho = C_{\overline{a}.}\,[[\![P]\!]_\rho] & [\![!\pi.P]\!]_\rho = C_{!\pi.}\,[[\![P]\!]_\rho] &
\end{array}
$$

*Proof.* Follows directly from the definition of contexts (Definition 3.2.4) and from the definition of $[\![\cdot]\!]_\rho : \mathcal{C}_{\mathtt{D}} \to \mathcal{S}$ (Figure 3.3). Indeed, for all operators and all well-formed compensable processes $P$ and $Q$ we have:

$$
\begin{aligned}
[\![P \mid Q]\!]_\rho &= C_{\mid}\,[[\![P]\!]_\rho, [\![Q]\!]_\rho] = [\![P]\!]_\rho \mid [\![Q]\!]_\rho \\
[\![t[P,Q]]\!]_\rho &= C_{t[,],\rho}\,[[\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon] = t\big[[\![P]\!]_{t,\rho}\big] \mid t.\,(\mathtt{extrd}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle \mid p_\rho[[\![Q]\!]_\varepsilon])\,. \\
[\![\langle P\rangle]\!]_\rho &= C_{\langle\rangle,\rho}\,[[\![P]\!]_\varepsilon] = p_\rho\big[[\![P]\!]_\varepsilon\big] \\
[\![a.P]\!]_\rho &= C_{a.}\,[[\![P]\!]_\rho] = a.[\![P]\!]_\rho \\
[\![\overline{a}.P]\!]_\rho &= C_{\overline{a}.}\,[[\![P]\!]_\rho] = \overline{a}.[\![P]\!]_\rho \\
[\![\overline{t}.P]\!]_\rho &= C_{\overline{t}.}\,[[\![P]\!]_\rho] = \overline{t}.h_t.[\![P]\!]_\rho \\
[\![(\nu x)P)]\!]_\rho &= C_{(\nu x)}\,[[\![P]\!]_\rho] = (\nu x)[\![P]\!]_\rho \\
[\![!\pi.P]\!]_\rho &= C_{!\pi.}\,[[\![\pi.P]\!]_\rho] = ![\![\pi.P]\!]_\rho.
\end{aligned}
$$

■

#### 3.2.3.1.2 Name invariance

Below we consider *name invariance* but first, we have the following remark:

**Remark 3.2.3.** We will say that a function $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ is a *valid substitution* if it is the identity except on a finite set and it respects syntactically the partition of $\mathcal{N}_c$ into subsets $\mathcal{N}_s$ and $\mathcal{N}_t$, i.e., $\sigma(\mathcal{N}_s) \subseteq \mathcal{N}_s$ and $\sigma(\mathcal{N}_t) \subseteq \mathcal{N}_t$ where $\sigma$ is injective (due to the condition that we observe only well-formed compensable processes for the translation). If $\rho = t_1, \dots, t_n, \varepsilon$, we write $\sigma(\rho)$ to denote the sequence $\sigma(t_1), \dots, \sigma(t_n), \varepsilon$.

We now state name invariance, by relying on the renaming policy in Definition 3.2.3 (a).

**Theorem 3.2.4** (Name invariance for $[\![\cdot]\!]_\rho$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{[\![\cdot]\!]_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]_\rho}(x)\}$, and

(ii) $[\![\sigma(P)]\!]_{\sigma(\rho)} = \sigma'([\![P]\!]_\rho)$.

*Proof.* We define the substitution $\sigma'$ as follows:

$$\sigma'(x) = \begin{cases} \sigma(x) & \text{if } x = a \text{ or } x = t \\ h_{\sigma(t)} & \text{if } x = h_t \\ p_{\sigma(\rho)} & \text{if } x = p_\rho. \end{cases} \tag{3.5}$$

Now we provide proofs for (i) and (ii):

(i) Since $\mathcal{N}_c = \mathcal{N}_t \cup \mathcal{N}_s$, we consider two sub-cases for $x$:

- if $x \in \mathcal{N}_s$ then it follows that:

$$\{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]_\rho}(x)\} = \{\sigma'(y) : y \in \{x\}\} = \{\sigma'(x)\} = \{\sigma(x)\} = \varphi_{[\![\cdot]\!]_\rho}(\sigma(x)).$$

- if $x \in \mathcal{N}_t$ then:
  - by Definition 3.2.3:

$$\varphi_{[\![\cdot]\!]_{\sigma(\rho)}}(\sigma(x)) = \{\sigma(x), h_{\sigma(x)}\} \cup \{p_{\sigma(\rho)} : \sigma(x) \in \sigma(\rho)\}$$

  - by definition of $\sigma'$:

$$\begin{aligned} & \{\sigma(x), h_{\sigma(x)}\} \cup \{p_{\sigma(\rho)} : \sigma(x) \in \sigma(\rho)\} \\ = & \{\sigma'(x), \sigma'(h_x)\} \cup \{\sigma'(p_\rho) : \sigma'(x) \in \sigma'(\rho)\} \\ = & \{\sigma'(y) : y \in \{x, h_x\}\} \cup \{\sigma'(y) : y \in \{p_\rho : \sigma(x) \in \sigma(\rho)\}\} \\ = & \{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]_\rho}(x)\} \end{aligned}$$

(ii) The proof proceeds by structural induction on $P$. In the following, given a name $x$, a path $\rho$, and process $P$, we write $\sigma x$, $\sigma\rho$, and $\sigma P$ to stand for $\sigma(x)$, $\sigma(\rho)$, and $\sigma(P)$, respectively.

***Base case:*** The statement holds for $P = \mathbf{0}$: $[\![\sigma(\mathbf{0})]\!]_{\sigma\rho} = \sigma'([\![\mathbf{0}]\!]_\rho) \Leftrightarrow \mathbf{0} = \mathbf{0}$.

***Inductive step:*** There are six cases, but we show only the following three cases: transaction scope, protected block, and input/output prefix. The proof for all the other cases proceeds similarly.

- *Case* $P = t[P_1,Q_1]$: We first apply the substitution $\sigma$ on process $P$:

$$[\![\sigma(t[P_1,Q_1])]\!]_{\sigma\rho} = [\![\sigma t[\sigma(P_1),\sigma(Q_1)]]\!]_{\sigma\rho}.$$

By expanding the definition of the translation in Definition 3.2.3, we have:

$$[\![\sigma(t[P_1,Q_1])]\!]_{\sigma\rho} = \sigma t\big[[\![\sigma(P_1)]\!]_{\sigma t,\sigma\rho}\big] \mid \sigma t.\,(\texttt{extrd}\langle\!\langle \sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\!\rangle \mid p_{\sigma\rho}[[\![\sigma(Q_1)]\!]_{\varepsilon}])$$

By induction hypothesis it follows:

$$[\![\sigma(t[P_1,Q_1])]\!]_{\sigma\rho} = \sigma t\big[\sigma'\left([\![P_1]\!]_{t,\rho}\right)\big] \mid \sigma t.\,\big(\texttt{extrd}\langle\!\langle \sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\!\rangle \mid p_{\sigma\rho}[\sigma'\left([\![Q_1]\!]_{\varepsilon}\right)]\big) \quad (3.6)$$

On the other side, when we apply definition of substitution $\sigma'$ on $[\![P]\!]_{\rho}$ the following holds:

$$\begin{aligned}\sigma'\left([\![t[P_1,Q_1]]\!]_{\rho}\right) &= \sigma'\left(t\big[[\![P_1]\!]_{t,\rho}\big] \mid t.\,(\texttt{extrd}\langle\!\langle t, p_{t,\rho}, p_{\rho}\rangle\!\rangle \mid p_{\rho}[[\![Q_1]\!]_{\varepsilon}])\right)\\ &= \sigma' t\big[\sigma'\left([\![P_1]\!]_{t,\rho}\right)\big] \mid \sigma' t.(\texttt{extrd}\langle\!\langle \sigma' t, p_{\sigma' t,\sigma'\rho}, p_{\sigma'\rho}\rangle\!\rangle \mid p_{\sigma'\rho}[\sigma'\left([\![Q_1]\!]_{\varepsilon}\right)]).\end{aligned} \quad (3.7)$$

Given that it is valid $\sigma'(t) = \sigma(t)$ (cf. (3.5)), it is easy to conclude that (3.6) is equal to (3.7).

- *Case* $P = \langle P_1 \rangle$*:* We apply substitution $\sigma$ on process $P$:

$$[\![\sigma(\langle P_1\rangle)]\!]_{\sigma\rho} = [\![\langle\sigma(P_1)\rangle]\!]_{\sigma\rho}$$

By Definition 3.2.3, $[\![\sigma(\langle P_1\rangle)]\!]_{\sigma\rho} = p_{\sigma\rho}\big[[\![\sigma(P_1)]\!]_{\varepsilon}\big]$, and by induction hypothesis:

$$[\![\sigma(\langle P_1\rangle)]\!]_{\sigma\rho} = p_{\sigma\rho}\big[\sigma'([\![P_1]\!]_{\varepsilon})\big]. \quad (3.8)$$

On the other side, when we apply substitution $\sigma'$ on $[\![P]\!]_{\rho}$ the following holds:

$$\sigma'([\![\langle P_1\rangle]\!]_{\rho}) = \sigma'(p_{\rho}\big[[\![P_1]\!]_{\varepsilon}\big]) = p_{\sigma'\rho}\big[\sigma'([\![P_1]\!]_{\varepsilon})\big]. \quad (3.9)$$

Based on definition of the function $\sigma'$, i.e. $\sigma'(p_{\rho}) = p_{\sigma(\rho)}$ and $\sigma'(t) = \sigma(t)$ (cf. (3.5)), it is easy to conclude that (3.8) is equal to (3.9).

- *Case* $P = \pi.P_1$*:* Here we distinguish two sub-cases. In the first sub-case we consider input on name $a \in \mathcal{N}_s$ (proof follows similarly for output). In the second sub-case we consider that the output message is an error notification on name $t \in \mathcal{N}_t$.

  - *Case* $P = a.P_1$*:* We apply substitution $\sigma$ on process $P$:

  $$[\![\sigma(a.P_1)]\!]_{\sigma\rho} = [\![\sigma a.\sigma(P_1)]\!]_{\sigma\rho}.$$

  Next, we apply Definition 3.2.3: $[\![\sigma(a.P_1)]\!]_{\sigma\rho} = \sigma a.[\![\sigma(P_1)]\!]_{\sigma\rho}$. By induction hypothesis it follows:

  $$[\![\sigma(a.P_1)]\!]_{\sigma\rho} = \sigma a.\sigma'([\![P_1]\!]_{\rho}). \quad (3.10)$$

  We now apply substitution $\sigma'$ on $[\![P]\!]_{\rho}$:

  $$\sigma'([\![(a.P_1)]\!]_{\rho}) = \sigma'(a.[\![P_1]\!]_{\rho}) = \sigma' a.\sigma'([\![P_1]\!]_{\rho}). \quad (3.11)$$

  By definition of $\sigma'$ (cf. (3.5)), $\sigma'(a) = \sigma(a)$ and so we conclude that (3.10) is equal to (3.11).

- *Case* $P = \overline{t}.P_1$: We apply substitution $\sigma$ on process $P$:

$$[\![\sigma(\overline{t}.P_1)]\!]_{\sigma\rho} = [\![\sigma\overline{t}.\sigma(P_1)]\!]_{\sigma\rho}.$$

Next, we apply Definition 3.2.3: $[\![\sigma(\overline{t}.P_1)]\!]_{\sigma\rho} = \sigma\overline{t}.h_{\sigma t}.[\![\sigma(P_1)]\!]_{\sigma\rho}$. By induction hypothesis:

$$[\![\sigma(\overline{t}.P_1)]\!]_{\sigma\rho} = \sigma\overline{t}.h_{\sigma t}.\sigma'([\![P_1]\!]_{\rho}). \tag{3.12}$$

We apply substitution $\sigma'$ on $[\![P]\!]_{\rho}$:

$$\sigma'([\![(\overline{t}.P_1)]\!]_{\rho}) = \sigma'(\overline{t}.h_t.[\![P_1]\!]_{\rho}) = \sigma'\overline{t}.h_{\sigma' t}.\sigma'([\![P_1]\!]_{\rho}). \tag{3.13}$$

By definition of $\sigma'$ (cf. (3.5)), $\sigma'(t) = \sigma(t)$ and so we conclude that (3.12) is equal to (3.13).

■

#### 3.2.3.2 Semantic Criteria

We prove the three semantic criteria, following the order in which they were introduced in Definition 2.3.5: operational correspondence, divergence reflection, and success sensitiveness.

##### 3.2.3.2.1 Operational Correspondence

Among the semantic criteria, operational correspondence is usually the most interesting one, but also the most delicate to prove. We are interested in giving a statement of operational correspondence that includes the number of reductions required in $\mathcal{S}$ to correctly mimic a reduction in $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$. Indeed, this will allow us to support our claim that subjective updates are more efficient than objective updates (cf. Definition 4.1.7). To precisely state completeness results we introduce some auxiliary notions: $\mathtt{pb}_{\mathtt{D}}(P)$, $\mathtt{pb}_{\mathtt{P}}(P)$ and $\mathtt{pb}_{\mathtt{A}}(P)$. Whenever a notion coincides for the all semantics, we shall avoid decorations $\mathtt{D}, \mathtt{P}$ and $\mathtt{A}$.

**Definition 3.2.5.** Given a compensable process $P$, we will write $\mathtt{pb}(P)$ to denote the number of protected blocks in $P$ — see Figure 3.4 for a definition.

Given a transaction $t[P,Q]$, the following lemma ensures that the number of protected blocks in the default activity $P$ is equal to the number of locations $p_{t,\rho}$ in $[\![P]\!]_{t,\rho}$ (Definition 3.2.1).

**Lemma 3.2.5.** Let $t[P,Q]$ and $\rho$ be a well-formed compensable process and an arbitrary path, respectively. Then it holds that $\mathtt{pb}_{\mathtt{D}}(P) = \mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho})$.

*Proof.* The proof is by induction on the structure of $P$.

- *Case* $P = \mathbf{0}$ or $P = \pi.P_1$ or $P = !\pi.P_1$: By Definition 3.2.1, Definition 3.2.5 and Definition 3.2.3, we can derive $\mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho}) = 0 = \mathtt{pb}_{\mathtt{D}}(P)$.
- *Case* $P = \langle P_1 \rangle$: By Definition 3.2.1, Definition 3.2.5 and Definition 3.2.3,

$$\mathtt{nl}(p_{t,\rho}, [\![\langle P_1\rangle]\!]_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, p_{t,\rho}\big[[\![P_1]\!]_{\varepsilon}\big]) = 1 = \mathtt{pb}_{\mathtt{D}}(\langle P_1\rangle).$$

- $P = s[P_1,Q_1]$: By Definition 3.2.3,

$$[\![s[P_1,Q_1]]\!]_{t,\rho} = s\big[[\![P_1]\!]_{s,t,\rho}\big] \mid s.\left(\mathtt{extrd}\langle\!\langle s, p_{s,t,\rho}\,,\, p_{t,\rho}\rangle\!\rangle \mid p_{t,\rho}[[\![Q_1]\!]_{\varepsilon}]\right).$$

Noticing that $\mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{s,t,\rho}) = 0$, by application of Definition 3.2.1 and Definition 3.2.5, we get $\mathtt{nl}(p_{t,\rho}, [\![s[P_1,Q_1]]\!]_{t,\rho}) = 0 = \mathtt{pb}_{\mathtt{D}}(s[P_1,Q_1])$.

| | |
|---|---|
| $\mathtt{pb}(\langle P \rangle) = 1$ | $\mathtt{pb}_{\mathtt{A}}(t[P,Q]) = 1 + \mathtt{pb}_{\mathtt{A}}(P)$ |
| $\mathtt{pb}_{\mathtt{D}}(t[P,Q]) = \mathtt{pb}_{\mathtt{P}}(t[P,Q]) = 0$ | $\mathtt{pb}(!\pi.P) = \mathtt{pb}(\pi.P) = \mathtt{pb}(\mathbf{0}) = 0$ |
| $\mathtt{pb}(P \mid Q) = \mathtt{pb}(P) + \mathtt{pb}(Q)$ | $\mathtt{pb}((\nu x)P) = \mathtt{pb}(P)$ |

Figure 3.4: Number of protected blocks for $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$

- *Case* $P = P_1 \mid Q_1$: By Definition 3.2.1 and Definition 3.2.3,

$$\mathtt{nl}(p_{t,\rho}, [\![P_1|Q_1]\!]_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}|[\![Q_1]\!]_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}) + \mathtt{nl}(p_{t,\rho}, [\![Q_1]\!]_{t,\rho}).$$

By induction hypothesis, we conclude $\mathtt{nl}(p_{t,\rho}, [\![P_1|Q_1]\!]_{t,\rho}) = \mathtt{pb}_{\mathtt{D}}(P_1) + \mathtt{pb}_{\mathtt{D}}(Q_1)$.

- *Case* $P = (\nu x)P_1$: By Definition 3.2.1 and Definition 3.2.3,

$$\mathtt{nl}(p_{t,\rho}, [\![(\nu x)P_1]\!]_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, (\nu x)[\![P_1]\!]_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}).$$

By induction hypothesis and Definition 3.2.5,

$$\mathtt{nl}(p_{t,\rho}, [\![(\nu x)P_1]\!]_{t,\rho}) = \mathtt{pb}_{\mathtt{D}}(P_1) = \mathtt{pb}_{\mathtt{D}}((\nu x)P_1).$$

■

The following example illustrates this claim.

**Example 3.2.6.** Notably, $P = t[P_1,d]$ is a well-formed compensable process, with default activity $P_1 = \langle a \rangle \mid \langle b \rangle \mid c$. By Figure 3.4, we have $\mathtt{pb}_{\mathtt{D}}(P_1) = 2$. Also, by Definition 3.2.3, we have:

$$[\![P]\!]_\rho = t\Big[[\![P_1]\!]_{t,\rho}\Big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho\big[d\big]\big),$$

such that $[\![P_1]\!]_{t,\rho} = p_{t,\rho}\big[a\big] \mid p_{t,\rho}\big[b\big] \mid c$. Now, by Definition 3.2.1 it is clear that $\mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}) = 2$.

We shall now prove that the translation $[\![\cdot]\!]_\rho$ satisfies operational correspondence (completeness and soundness).

**Theorem 3.2.7** (Operational Correspondence for $[\![\cdot]\!]_\rho$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{D}}$.

(1) If $P \longrightarrow P'$ then $[\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,
b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ it follows $k = 4 + \mathtt{pb}_{\mathtt{D}}(P_1)$,
c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q \rangle]$ it follows $k = 4 + \mathtt{pb}_{\mathtt{D}}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

(2) If $[\![P]\!]_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* [\![P']\!]_\varepsilon$.

In the Theorem 3.2.7, Case (1) concerns completeness while Case (2) describes soundness. Case (1)(a) concerns usual synchronizations, which are translated by $[\![\cdot]\!]_\rho$. Cases (1)-(b) and (c) concern synchronizations due to compensation signals; here the analysis distinguishes four cases, as the failure signal can be external or internal (cf. (2.1), Page 14) and the transaction can be replicated or not. In all cases, the number of reduction steps required to mimic the source transition depends on the number of protected blocks of the transaction being canceled. Therefore, in the proof we will consider completeness and soundness (Parts (1) and (2)) separately.

Due to the complexity of the proof for Theorem 3.2.7 we first present some auxiliary results for completeness and soundness, respectively.

#### 3.2.3.2.2 Auxiliary Results for Completeness

For the proof of operational correspondence, we introduce a mapping of evaluation contexts for compensable (cf. Definition 2.2.3) into evaluation contexts for adaptable processes (cf. Definition 2.2.6). For this mapping we rely directly on translation $\mathcal{C}_{\text{D}}$ into $\mathcal{S}$ (cf. Definition 3.2.3).

**Definition 3.2.6.** Let $\rho$ be a path. We define mapping $[\![\cdot]\!]_\rho$ from evaluation contexts of compensable processes into evaluation contexts of adaptable processes as follows:

$$\begin{aligned}
[\![[\bullet]]\!]_\rho &= [\bullet] \\
[\![\langle C[\bullet]\rangle]\!]_\rho &= p_\rho[[\![C[\bullet]]\!]_\varepsilon] \\
[\![t[C[\bullet],Q]]\!]_\rho &= t\big[[\![C[\bullet]]\!]_{t,\rho}\big] \mid t.\,(\texttt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]) \\
[\![C[\bullet] \mid P]\!]_\rho &= [\![C[\bullet]]\!]_\rho \mid [\![P]\!]_\rho \\
[\![(\nu x)C[\bullet]]\!]_\rho &= (\nu x)[\![C[\bullet]]\!]_\rho
\end{aligned}$$

**Convention 3.2.8.** We will use $[\![C]\!]_\rho[P]$ to denote the process that is obtained when the only hole of context $[\![C[\bullet]]\!]_\rho$ is replaced with process $P$.

**Lemma 3.2.9.** Let $P$ be a well-formed compensable process, $C[\bullet]$ an evaluation context, $\rho$ an arbitrary path, and $\rho'$ the path to the hole in $C[\bullet]$. Then, $[\![C[P]]\!]_\rho = [\![C]\!]_\rho[[\![P]\!]_{\rho'}]$.

*Proof.* The proof proceeds by induction on the structure of $C[\bullet]$.

***Base cases:*** Assume that $C[\bullet] = [\bullet]$ then $C[P] = P$, and $[\![C[P]]\!]_\rho = [\![P]\!]_\rho$.

***Inductive step:*** There are four cases to consider. They all proceed by Definition 3.2.3, Definition 3.2.6, and the inductive hypothesis:

- *Case* $C[\bullet] = \langle C_1[\bullet]\rangle$: Then $C[P] = \langle C_1[P]\rangle$ and

$$\begin{aligned}
[\![C[P]]\!]_\rho = [\![\langle C_1[P]\rangle]\!]_\rho &\stackrel{\text{Def.}}{=} p_\rho[[\![C_1[P]]\!]_\varepsilon] \\
&\stackrel{\text{I.H.}}{=} p_\rho[[\![C_1]\!]_\varepsilon[[\![P]\!]_{\rho'}]] = p_\rho[[\![C_1]\!]_\varepsilon]\,[[\![P]\!]_{\rho'}] \\
&\stackrel{\text{Def.}}{=} [\![C]\!]_\rho\,[[\![P]\!]_{\rho'}]
\end{aligned}$$

- *Case* $C[\bullet] = C_1[\bullet] \mid Q$: Then $C[P] = C_1[P] \mid Q$ and

$$\begin{aligned}
[\![C[P]]\!]_\rho = [\![C_1[P] \mid Q]\!]_\rho &\stackrel{\text{Def.}}{=} [\![C_1[P]]\!]_\rho \mid [\![Q]\!]_\rho \\
&\stackrel{\text{I.H.}}{=} [\![C_1]\!]_\rho[[\![P]\!]_{\rho'}] \mid [\![Q]\!]_\rho \\
&= [\![C_1 \mid Q]\!]_\rho[[\![P]\!]_{\rho'} = [\![C]\!]_\rho\,[[\![P]\!]_{\rho'}]
\end{aligned}$$

- *Case* $C[\bullet] = t[C_1[\bullet],Q]$: Then $C[P] = t[C_1[P],Q]$ and

$$\begin{aligned}
[\![C[P]]\!]_\rho = [\![t[C_1[P],Q]]\!]_\rho &\stackrel{\text{Def.}}{=} t\big[[\![C_1[P]]\!]_{t,\rho}\big] \mid t.\,(\texttt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]) \\
&\stackrel{\text{I.H.}}{=} t\big[[\![C_1]\!]_{t,\rho}[[\![P]\!]_{\rho'}]\big] \mid t.\,(\texttt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]) \\
&= \big(t\big[[\![C_1]\!]_{t,\rho}\big] \mid t.\,(\texttt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon])\big)\,[[\![P]\!]_{\rho'}] \\
&= [\![C]\!]_\rho\,[[\![P]\!]_{\rho'}]
\end{aligned}$$

- *Case* $C[\bullet] = (\nu x)C_1[\bullet]$: Then $C[P] = (\nu x)C_1[P]$ and

$$\begin{aligned}
[\![C[P]]\!]_\rho = [\![(\nu x)C_1[P]]\!]_\rho &\stackrel{\text{Def.}}{=} (\nu x)[\![C_1[P]]\!]_\rho \\
&\stackrel{\text{I.H.}}{=} (\nu x)[\![C_1]\!]_\rho[[\![P]\!]_{\rho'}] = [\![C]\!]_\rho\,[[\![P]\!]_{\rho'}].
\end{aligned}$$

∎

#### 3.2.3.2.3 Auxiliary Results for Soundness

For the proof of soundness, we will need the converse of Lemma 3.2.9, which is stated by the following two results.

**Lemma 3.2.10.** Let $P$ be a well-formed compensable process and $\rho$ a path. If $[\![P]\!]_\rho \equiv C[P']$ then there are $C_1[\bullet]$ and $P_1$ such that $C[\bullet] = [\![C_1[\bullet]]\!]_\rho$ and $P' = [\![P_1]\!]_{\rho'}$, where $\rho'$ is the path to the hole in $C_1[\bullet]$.

*Proof.* The proof is by induction on structure of context $C[\bullet]$.

***Base case:*** If $C[\bullet] = [\bullet]$ and $[\![P]\!]_\rho = P'$ then it follows directly that $C_1[\bullet] = [\bullet]$ and $P_1 = P$.

***Inductive step:*** We consider the following three cases:

- *Case* $C[\bullet] = l[C'[\bullet]]$: Let $[\![P]\!]_\rho \equiv l[C'[P']]$. By Definition 3.2.3, we have that $l = p_\rho$ and there is $P_1'$ such that $[\![P_1']\!]_\rho = C'[P']$. By the induction hypothesis, there are $C_1'[\bullet]$ and $P_1$ such that $C'[\bullet] = [\![C_1'[\bullet]]\!]_\rho$ and $P' = [\![P_1]\!]_{\rho'}$, where $\rho'$ is the path to the hole in $C_1'[\bullet]$. By Definition 3.2.6, $C[\bullet] = p_\rho[[\![C_1'[\bullet]]\!]_\rho] = [\![\langle C_1'[\bullet]\rangle]\!]_\rho$, and hence $C_1[\bullet] = \langle C_1'[\bullet]\rangle$.

- *Case* $C[\bullet] = (\nu x)C'[\bullet]$: Let $[\![P]\!]_\rho \equiv (\nu x)C'[P']$. By Definition 3.2.3, there is $P_1'$ such that $[\![P_1']\!]_\rho = C'[P']$. By induction hypothesis, there are $C_1'[\bullet]$ and $P_1$ such that $C'[\bullet] = [\![C_1'[\bullet]]\!]_\rho$ and $P' = [\![P_1]\!]_{\rho'}$, where $\rho'$ is the path to the hole in $C_1'[\bullet]$. Now, we have that $C[\bullet] = (\nu x)[\![C_1'[\bullet]]\!]_\rho = [\![(\nu x)C_1'[\bullet]]\!]_\rho$ and hence $C_1[\bullet] = (\nu x)C_1'[\bullet]$.

- *Case* $C[\bullet] = C'[\bullet] \mid Q$: Let $[\![P]\!]_\rho \equiv C'[P'] \mid Q$. By Definition 3.2.3, we have two possibilities:

  (i) If $Q = t.(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q']\!]_\varepsilon])$ and $C'[P'] \equiv t[[\![P_1']\!]_{t,\rho}]$ for some $t, P_1', Q'$, then $[\![P_1']\!]_{t,\rho} = C_1'[P']$ for some $C_1'$. By induction hypothesis, there are $C_1''[\bullet]$ and $P_1$ such that $C_1'[\bullet] = [\![C_1''[\bullet]]\!]_\rho$ and $P' = [\![P_1]\!]_{\rho'}$, where $\rho'$ is the path to the hole in $C_1''[\bullet]$. We complete the proof by choosing $C[\bullet] = t[C_1''[\bullet],Q']$ and $P' = [\![P_1]\!]_{t,\rho'}$.

  (ii) If $C'[P'] = [\![Q_1]\!]_\rho$ and $Q = [\![Q_2]\!]_\rho$ for some $Q_1, Q_2$, then, by induction hypothesis, there are $C_1'[\bullet]$ and $P_1$ such that $C'[\bullet] = [\![C_1'[\bullet]]\!]_\rho$ and $P' = [\![P_1]\!]_{\rho'}$, where $\rho'$ is the path to the hole in $C_1'[\bullet]$. In this case, $C[\bullet] = C_1'[\bullet] \mid Q_2$.

■

As a direct consequence of Case 3 in the previous proof, we can identify two possibilities for a process that is obtained via our translation and equals to a parallel composition of processes.

**Corollary 3.2.11.** Let $P$ be a well-formed compensable process and $\rho$ a path.
If $[\![P]\!]_\rho \equiv C[P'] \mid D[Q']$ then either:

(i) There are $C_1[\bullet]$, $D_1[\bullet]$, $P_1$, and $Q_1$ such that:

- $C[\bullet] = [\![C_1[\bullet]]\!]_\rho$,
- $D[\bullet] = [\![D_1[\bullet]]\!]_\rho$,
- $P' = [\![P_1]\!]_{\rho'}$ and $Q' = [\![Q_1]\!]_{\rho''}$, where $\rho'$ and $\rho''$ are paths to holes in $C[\bullet]$ and $D[\bullet]$, respectively.

(ii) There are $C_1[\bullet]$, $P_1$, $Q, t$ such that $Q' \equiv t.(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon])$, $D[\bullet] = [\bullet]$, and $C[\bullet] = t[C_1[\bullet]]$.

The proof of soundness proceeds by induction on $n$. The base case uses the following lemma. In cases (b) and (c), we use a process of the form $I_t^{(1)}([\![P]\!]_{t,\rho''}, [\![Q]\!]_\varepsilon)$ and $O_u^{(1)}([\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}], [\![Q]\!]_\varepsilon)$ where $t, u$ are names and "1" intuitively denotes the first intermediate process in the translation. In fact, processes of the form $I_t^{(p)}([\![P]\!]_{t,\rho''}, [\![Q]\!]_\varepsilon)$ and $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}], [\![Q]\!]_\varepsilon)$ with $p, q \geq 1$, that are introduced in Figure 3.6 and Figure 3.7, will be important in the proof of soundness.

**Lemma 3.2.12.** Suppose $[\![P]\!]_\rho \longrightarrow R$. Then one of the following holds for $P$ and $R$:

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $R \equiv [\![E]\!]_\rho\Big[[\![C]\!]_{\rho_1}[[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![P_2]\!]_{\rho''}]\Big]$, or

b) $P \equiv E\Big[C[\overline{t}.P_1] \mid D[t[P_2,Q]]\Big]$ and
$R \equiv [\![E]\!]_\rho\Big[[\![C]\!]_{\rho_1}[h_t.[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[I_t^{(1)}([\![P_2]\!]_{t,\rho''}, [\![Q]\!]_\varepsilon)]\Big]$ where
$I_t^{(1)}([\![P_2]\!]_{t,\rho''}, [\![Q]\!]_\varepsilon) = t\big[[\![P_2]\!]_{t,\rho''}\big] \mid \texttt{extrd}\langle\!\langle t, p_{t,\rho''}, p_{\rho''}\rangle\!\rangle \mid p_{\rho''}[[\![Q]\!]_\varepsilon]$, or

c) $P \equiv E[u[C[\overline{u}.P_1],Q]]$ and $R \equiv [\![E]\!]_\rho\Big[O_u^{(1)}([\![C]\!]_{u,\rho_1}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon)\Big]$ where

$$O_u^{(1)}([\![C]\!]_{u,\rho_1}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon) = u\big[[\![C]\!]_{u,\rho_1}[h_u.[\![P_1]\!]_{\rho'}]\big] \mid \texttt{extrd}\langle\!\langle t, p_{u,\rho_1}, p_{\rho_1}\rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon],$$

for some contexts $C[\bullet], D[\bullet], E[\bullet]$ and processes $P_1, P_2, Q$. Also, paths $\rho$, $\rho'$, and $\rho''$ are paths to holes in contexts $E[\bullet]$, $C[\bullet]$, and $D[\bullet]$, respectively.

*Proof.* The proof is by induction on the reduction $[\![P]\!]_\rho \longrightarrow R$. There are three base cases, which can be obtained by applying Rule (R-In-Out)) with $x = a$ or $x = t$.

a) $[\![P]\!]_\rho = E'[C'[\overline{a}.P_1'] \mid D'[a.P_2']] \longrightarrow E'[C'[P_1'] \mid D'[P_2']] = R$.
By Lemma 3.2.10, Corollary 3.2.11, Definition 3.2.3 and Lemma 3.2.9, we get the following derivation:

$$\begin{aligned}
[\![P]\!]_\rho &= [\![E]\!]_\rho[[\![S]\!]_{\rho_1}], && (1)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[[\![S_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![S_2]\!]_{\rho''}]], && (2)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[\overline{a}.[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[a.[\![P_2]\!]_{\rho''}]], && (3)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[[\![\overline{a}.P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![a.P_2]\!]_{\rho''}]]\\
&= [\![E[C[\overline{a}.P_1] \mid D[a.P_2]]]\!]_\rho
\end{aligned}$$

where

$$\begin{aligned}
&(1)\quad [\![S]\!]_{\rho_1} = C'[\overline{a}.P_1'] \mid D'[a.P_2'],\\
&(2)\quad [\![S_1]\!]_{\rho'} = \overline{a}.P_1', [\![S_2]\!]_{\rho''} = a.P_2',\\
&(3)\quad [\![P_1]\!]_{\rho'} = P_1', [\![P_2]\!]_{\rho''} = P_2',
\end{aligned}$$

and $[\![D]\!]_{\rho_1}[\bullet] = D'[\bullet]$, $[\![C[\bullet]]\!]_{\rho_1} = C'[\bullet]$, and $\rho_1$, $\rho'$ and $\rho''$ are paths to holes in $E[\bullet]$, $C[\bullet]$ and $D[\bullet]$, respectively.

b) $[\![P]\!]_\rho = E'[C'[\overline{t}.P_1'] \mid D'[t.P_2']] \longrightarrow E'[C'[P_1'] \mid D'[P_2']] = R$ and $D'[\bullet] \neq [\bullet]$.
By Lemma 3.2.10, Corollary 3.2.11 and Definition 3.2.3, we get the following derivation:

$$
\begin{aligned}
[\![P]\!]_\rho &= [\![E]\!]_\rho[[\![S]\!]_{\rho_1}], && (1)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[[\![S_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![S_2]\!]_{\rho''}]], && (2)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[\overline{t}.h_t.[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[t[[\![P_2]\!]_{\rho''}] && (3)\\
&\quad \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho''}, p_{\rho''}\rangle\!\rangle \mid p_{\rho''}[[\![Q]\!]_\varepsilon]\big)], && (4)\\
&= [\![E]\!]_\rho[[\![C]\!]_{\rho_1}[[\![\overline{t}.P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![t[P_2,Q]]\!]_{\rho''}]]\\
&= [\![E[C[\overline{t}.P_1] \mid D[t[P_2,Q]]]]\!]_\rho
\end{aligned}
$$

where

$$
\begin{aligned}
&(1)\quad [\![S]\!]_{\rho_1} = C'[\overline{t}.P_1'] \mid D''[t[P_2''] \mid t.P_2']\\
&(2)\quad [\![S_1]\!]_{\rho'} = \overline{t}.P_1', [\![S_2]\!]_{\rho''} = t[P_2''] \mid t.P_2'\\
&(3)\quad P_1' = h_t.[\![P_1]\!]_{\rho'}, P_2'' = [\![P_2]\!]_{\rho''},\\
&(4)\quad P_2' = \mathtt{extrd}\langle\!\langle t, p_{t,\rho''}, p_{\rho''}\rangle\!\rangle \mid p_{\rho''}[[\![Q]\!]_\varepsilon]
\end{aligned}
$$

and $[\![D]\!]_{\rho_1}[\bullet] = D''[\bullet]$, $[\![C[\bullet]]\!]_{\rho_1} = C'[\bullet]$, and $\rho_1$, $\rho'$, $\rho''$ are paths to holes in $E[\bullet]$, $C[\bullet]$, $D[\bullet]$, respectively.

c) $[\![P]\!]_\rho = E'[C'[\overline{u}.P_1'] \mid D'[u.P_2']] \longrightarrow E'[C'[P_1'] \mid D'[P_2']] = R$ and $D'[\bullet] = [\bullet]$ and $C'[\bullet] = u[C''[\bullet]]$.
By Lemma 3.2.10, Corollary 3.2.11 and Definition 3.2.3, we get the following derivation:

$$
\begin{aligned}
[\![P]\!]_\rho &= E'[u[C''[\overline{u}.P_1']] \mid u.P_2']\\
&= [\![E]\!]_\rho[[\![S]\!]_{\rho_1}] && (1)\\
&= [\![E]\!]_\rho[u[[\![P_1'']\!]_{u,\rho_1}] && (2)\\
&\quad \mid\ u.\,(\mathtt{extrd}\langle\!\langle u, p_{u,\rho_1}, p_{\rho_1}\rangle\!\rangle \mid p_{\rho_1}[[\![Q]\!]_\varepsilon])] && (3)\\
&= [\![E]\!]_\rho[u[[\![C]\!]_{u,\rho_1}[[\![\overline{u}.P_1]\!]_{\rho'}]] \mid u.\,(\mathtt{extrd}\langle\!\langle u, p_{u,\rho_1}, p_{\rho_1}\rangle\!\rangle \mid p_{\rho_1}[[\![Q]\!]_\varepsilon])]\\
&= [\![E[u[C[\overline{u}.P_1],Q]]]\!]_\rho.
\end{aligned}
$$

where

$$
\begin{aligned}
&(1)\quad [\![S]\!]_{\rho_1} = u[C''[\overline{u}.P_1']] \mid u.P_2'\\
&(2)\quad [\![P_1'']\!]_{u,\rho_1} = C''[\overline{u}.P_1'] = t[P_2''] \mid t.P_2'\\
&(3)\quad P_2' = \mathtt{extrd}\langle\!\langle u, p_{u,\rho_1}, p_{\rho_1}\rangle\!\rangle \mid p_{\rho_1}[[\![Q]\!]_\varepsilon]
\end{aligned}
$$

and $[\![C[\bullet]]\!]_{t,\rho_1} = C''[\bullet]$ and $\rho_1$ and $\rho'$ are paths to holes in $E[\bullet]$ and $C[\bullet]$, respectively.
Note that since we analyze only one (first) reduction step, i.e. $[\![P]\!]_\rho \longrightarrow R$, the case of a reduction derived by Rule (R-SUB-UPD) is excluded by definition of translation.

Finally, the inductive step considers cases when the last step was derived by Rule (R-STR). In that way, we get case with "≡" instead of "=" in the three base cases. ■

Starting from an adaptable process $P$ that results from our translation, we single out those processes that $P$ reduces to but that do not correspond to the translation of any compensable process. Such processes always appear after a synchronization on some name $t$ and before synchronization on the reserved name $h_t$. We will first consider computations of a process that results from translating the parallel composition of a transaction and its failure signal (possibly with some continuation).

Recall that function $\mathsf{ch}(t, R)$ (cf. Definition 3.2.1) checks whether $R$ is structurally equivalent to a process of the form $C[h_t.S]$, for some context $C[\bullet]$ and process $S$: if this is not the case, then $\mathsf{ch}(t, R) = \mathbf{0}$. In a process obtained from our translation, process $h_t.S$ always occurs within a process of the form $\bar{t}.h_t.S$ (cf. Figure 3.3), directly implying that any process $[\![P]\!]_\rho$ cannot be congruent with $C[h_t.S]$. This is stated by the following lemma.

**Lemma 3.2.13.** Let $P$ be a well-formed compensable process. If $[\![P]\!]_\rho{=}\pi.Q$ then $\pi = a$ or $\pi = \bar{a}$ or $\pi = \bar{t}$, for some $a \in \mathcal{N}_s$ and $t \in \mathcal{N}_t$.

*Proof.* Follows directly from definition of the translation (cf. Definition 3.2.3). ■

The following lemma holds also for all translations that will be considered in the thesis, i.e., for translation $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$.

**Lemma 3.2.14.** Let $P$ be a well-formed compensable process, $t$ a transaction name, and $\rho$ a path. Then, it holds that $\mathsf{ch}(t, [\![P]\!]_\rho) = \mathbf{0}$.

*Proof.* By contradiction. Suppose, for the sake of contradiction, that $\mathsf{ch}(t, [\![P]\!]_\rho) = h_t.\mathbf{0}$. Then, $[\![P]\!]_\rho \equiv C[h_t.S]$. By Lemma 3.2.10, there are $C_1[\bullet]$ and $Q$ such that $[\![C_1[\bullet]]\!]_\rho = C[\bullet]$ and $[\![Q]\!]_{\rho'} = h_t.S$, where $\rho'$ is the path to $[\bullet]$ in $C_1[\bullet]$. But this contradicts Lemma 3.2.13: it is not possible that $[\![Q]\!]_{\rho'} = h_t.S$ since, necessarily, $h_t$ is a reserved name in $\mathcal{N}_s^r$; by Definition 3.1.2, $\mathcal{N}_s^r \cap \mathcal{N}_t = \emptyset$ and $\mathcal{N}_s^r \cap \mathcal{N}_s = \emptyset$. ■

In studying the processes that are obtained by translating the parallel composition of a transaction and its (externally triggered) failure signal (and its computation), we come to the lemmas that identify processes that are created before a synchronization on $h_t$.

**Lemma 3.2.15.** If $[\![E]\!]_\rho[P] \mid Q \equiv C[S]$ where $S = \pi.R$ or $S = \checkmark$ then there exist contexts $E'[\bullet]$, $E''[\bullet]$, and $E'''[\bullet]$ such that:

1. $[\![E[\bullet]]\!]_\rho = [\![E'[\bullet]]\!]_\rho \mid [\![E'']\!]_\rho[S]$ and for $S = \pi.R$ it holds $\pi \in \{x, \overline{x}\}$, or
2. $P \equiv E'''[S]$, or
3. $Q \equiv E'''[S]$.

*Proof.* The proof proceeds by induction on the structure of context $C[\bullet]$. ■

The following definition formalizes all possible forms for the process $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$. Recall that function $\mathtt{nl}(l, P)$, defined in Definition 3.2.1 (1), returns the number of locations $l$ in process $P$.

**Definition 3.2.7.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$ (Figure 3.6) depending on $m = \mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho})$:

1. if $m = 0$ then $p \in \{1, 2, 3\}$;
2. otherwise, if $m > 0$ then $[\![P]\!]_{t,\rho} = \prod_{k=1}^{m} p_{t,\rho}[[\![P'_k]\!]_\varepsilon] \mid S$ and $p \in \{1, \ldots, m+3\}$.

Figure 3.5 illustrates how intermediate processes relate to the encoding of well-formed compensable processes. The main role of these processes is to extract all processes $p_{t,\rho}[\cdot]$ from $[\![P]\!]_{t,\rho}$ using the process $\mathtt{outd}$.

**Lemma 3.2.16.** Let $P_1$ be a well-formed compensable process such that

- $[\![P_1]\!]_\varepsilon \equiv [\![E]\!]_\varepsilon \left[[\![G]\!]_\rho \left[[\![C]\!]_{\rho'} \left[[\![t[P_t, Q_t]]\!]_{\rho''}\right] \mid [\![D]\!]_{\rho'} \left[[\![\bar{t}.S_t]\!]_{\rho'''}\right] \mid M_1\right] \mid M_2\right] \mid M_3$ and

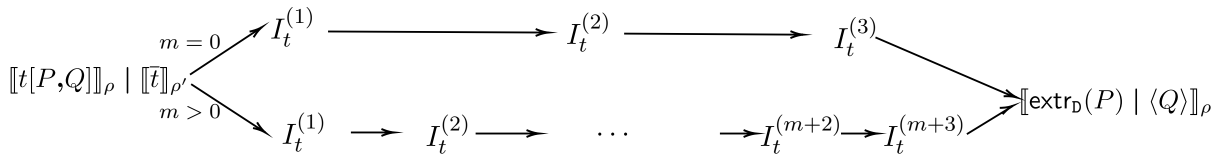



Figure 3.5: Process $I_t^{(p)}$.

| $(p)$ | $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$ for $\mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho}) = 0$ |
|---|---|
| (1) | $t\big[[\![P]\!]_{t,\rho}\big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]$ <br> $\equiv t\big[[\![P]\!]_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t,Y) \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t} \rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]$ |
| (2) | $t\big[[\![P]\!]_{t,\rho}\big] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t} \mid p_\rho[[\![Q]\!]_\varepsilon]$ |
| (3) | $\overline{h_t} \mid p_\rho[[\![Q]\!]_\varepsilon]$ |

| $(p)$ | $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$ for $\mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho}) > 0$ |
|---|---|
| (1) | $t\big[[\![P]\!]_{t,\rho}\big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]$ <br> $\equiv t\big[[\![P]\!]_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t,Y)$ <br> $\mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}\,,\, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t}) \rangle\!\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]$ |
| $(j+2), 0 \leq j \leq m-1$ | $t\big[[\![P]\!]_{t,\rho}\big] \mid p_{t,\rho}\langle\!\langle (X_1, \ldots, X_{m-j}).\big(\prod_{k=1}^{m-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[[\![P'_k]\!]_\varepsilon]$ <br> $\mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t}\big) \rangle\!\rangle \mid p_\rho[[\![Q_t]\!]_\varepsilon]$ |
| $(m+2)$ | $t\big[[\![P']\!]_{t,\rho}\big] \mid \prod_{k=1}^{m} p_\rho[[\![P'_k]\!]_\varepsilon] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t} \mid p_\rho[[\![Q]\!]_\varepsilon]$ |
| $(m+3)$ | $\prod_{k=1}^{m} p_\rho[[\![P'_k]\!]_\varepsilon] \mid \overline{h_t} \mid p_\rho[[\![Q]\!]_\varepsilon]$ |

Figure 3.6: Process $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$ with $p \geq 1$.

- $[\![P_1]\!]_\varepsilon \longrightarrow^{n-1} R$,

$$R \equiv [\![E_1]\!]_\varepsilon \Big[[\![G_1]\!]_\rho \Big[[\![C_1]\!]_{\rho'} \Big[I_t^{(p)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)\Big] \mid [\![D_1]\!]_{\rho'} \big[h_t.[\![S_t]\!]_{\rho'''}\big] \mid M'_1\Big] \mid M'_2\Big] \mid M'_3,$$

where $I_t^{(p)}([\![P_t]\!]_{t,\rho''}, [\![Q_t]\!]_\varepsilon)$ in $R$ is as in Definition 3.2.7. If $R \longrightarrow R'$ then either

I) $R' \equiv [\![E_1]\!]_\varepsilon \Big[[\![G_1]\!]_\rho \Big[[\![C_1]\!]_{\rho'} \Big[I_t^{(p+1)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)\Big] \mid [\![D_1]\!]_{\rho'} \big[h_t.[\![S_t]\!]_{\rho'''}\big] \mid M'_1\Big] \mid M'_2\Big] \mid M'_3$
or

II) $R' \equiv [\![E_2]\!]_\varepsilon \Big[[\![G_2]\!]_\rho \Big[[\![C_2]\!]_{\rho'} \Big[I_t^{(p)}([\![P''_t]\!]_{t,\rho''}, [\![Q''_t]\!]_\varepsilon)\Big] \mid [\![D_2]\!]_{\rho'} \big[h_t.[\![S_t]\!]_{\rho'''}\big] \mid M''_1\Big] \mid M''_2\Big] \mid M''_3.$

where:

- $n > 1$;
- $\rho$ is the path to holes in $[\![E[\bullet]]\!]_\varepsilon$ and $[\![E_k[\bullet]]\!]_\varepsilon$ and $k \in \{1,2\}$;
- $\rho'$ is the path to holes in $[\![G[\bullet]]\!]_\rho$, and $[\![G_k[\bullet]]\!]_\rho$ and $k \in \{1,2\}$;

- $\rho''$ is the path to the hole in $[\![C[\bullet]]\!]_{\rho'}$ and $[\![C_k[\bullet]]\!]_{\rho'}$ and $k \in \{1,2\}$;
- $\rho'''$ is the path to hole in $[\![D[\bullet]]\!]_{\rho'}$ and $[\![D_k[\bullet]]\!]_{\rho'}$ and $k \in \{1,2\}$.

*Proof.* By Definition 3.2.3, we get:

$$\begin{aligned}[\![P_1]\!]_\varepsilon \equiv & [\![E]\!]_\varepsilon \big[[\![G]\!]_\rho \big[[\![C]\!]_{\rho'} \big[t\big[[\![P_t]\!]_{t,\rho''}\big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho''}, p_{\rho''}\rangle\!\rangle \mid p_{\rho''}[[\![Q_t]\!]_\varepsilon]\big)\big] \\ & \mid [\![D]\!]_{\rho'}\big[\overline{t}.h_t.[\![S_t]\!]_{\rho'''}\big] \mid M_1\big] \mid M_2\big] \mid M_3\end{aligned} \tag{3.14}$$

We continue with the proof by case analysis for the step, $R \longrightarrow R'$, that can be realized. The analysis depends on the shape $I_t^{(p)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)$. Hence, there are multiple cases, for $p \in \{1,\dots,m+3\}$ and $m \geq 0$. We detail only one case, namely $p = 1$; all other cases proceed similarly.

If $p = 1$ then

$$R \equiv [\![E_1]\!]_\varepsilon \Big[[\![G_1]\!]_\rho \Big[[\![C_1]\!]_{\rho'} \Big[I_t^{(1)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)\Big] \mid [\![D_1]\!]_{\rho'} \big[h_t.[\![S_t]\!]_{\rho'''}\big] \mid M'_1\Big] \mid M'_2\Big] \mid M'_3. \tag{3.15}$$

In the analysis, we will use the following representation of process $R$:

$$\begin{aligned} & R \equiv [\![E_1]\!]_\rho[P'] \mid M'_3 \text{ where} \\ & P' \equiv [\![G_1]\!]_\rho \Big[[\![C_1]\!]_{\rho'} \Big[I_t^{(1)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)\Big] \mid [\![D_1]\!]_{\rho'} \big[h_t.[\![S_t]\!]_{\rho'''}\big] \mid M'_1\Big] \mid M'_2\end{aligned} \tag{3.16}$$

For $R \longrightarrow R'$ we analyze the following two sub-cases, based on the rules from reduction semantics for adaptable processes: Rule (R-In-Out) and Rule (R-Sub-Upd) (cf. Figure 2.5).

A) By using Rule (R-In-Out):

$$R \equiv E\Big[C[\overline{x}.P] \mid D[x.Q]\Big] \longrightarrow E\Big[C[P] \mid D[Q]\Big] \equiv R'.$$

Therefore, we have:

$$\begin{aligned} & R \equiv E'[x.Q] \text{ where } E'[\bullet] = E[C[\overline{x}.P] \mid D[\bullet]] \text{ and} \\ & R \equiv E''[\overline{x}.P] \text{ where } E''[\bullet] = E[C[\bullet] \mid D[x.Q]].\end{aligned}$$

In (3.16), based on Lemma 3.2.15 for $R \equiv E'[x.Q]$, the following holds:

(i) $[\![E_1[\bullet]]\!]_\rho = [\![E'_1[\bullet]]\!]_\rho \mid [\![E''_1]\!]_\rho[x.Q]$, or

(ii) $P' = E'''[x.Q]$,or

(iii) $M'_3 = E'''[x.Q]$.

In the following, we present detailed analysis only for case (i). Proofs of cases (ii) and (iii) follow in a similar way, i.e., with the case analysis that is result of applying Lemma 3.2.15.

Therefore, if (i) holds then $R \equiv [\![E_1]\!]_\rho[P'] \mid M'_3 = [\![E'_1]\!]_\rho[P'] \mid [\![E''_1]\!]_\rho[x.Q] \mid M'_3$. For $R \equiv E''[\overline{x}.P]$ the following holds based on Lemma 3.2.15:

(a) $[\![E'_1[\bullet]]\!]_\rho = [\![E'_2[\bullet]]\!]_\rho \mid [\![E''_2]\!]_\rho[\overline{x}.P]$, or

(b) $P' = E'''_2[\overline{x}.P]$, or

(c) $M'_3 \mid [\![E''_1]\!]_\rho[x.P] = E'''_2[\overline{x}.P]$.

In the following we analyze the sub-cases. It should be noted that in all sub-cases, obtained process $R'$ corresponds to the case II) from the statement:

(a)

$$R \equiv [\![E_2'[P']]\!]_\rho \mid [\![E_2'']\!]_\rho[\overline{x}.P] \mid [\![E_1'']\!]_\rho[x.Q] \mid M_3' \\ \longrightarrow [\![E_2'[P']]\!]_\rho \mid [\![E_2'']\!]_\rho[P] \mid [\![E_1'']\!]_\rho[Q] \mid M_3' \equiv R', \text{ or}$$

(b) $R \equiv [\![E_1]\!]_\rho[E_2''[\overline{x}.P]] \mid E_1''[x.Q] \mid M_3' \longrightarrow [\![E_1]\!]_\rho[E_2''[P]] \mid E_1''[Q] \mid M_3' \equiv R'$,or

(c) we distinguish two cases based on Lemma 3.2.15:

- $M_3' \equiv E_1^{iv}[\overline{x}.P]$ and it follows

$$R \equiv [\![E_1']\!]_\rho[P'] \mid [\![E_1'']\!]_\rho[x.Q] \mid E_1^{iv}[\overline{x}.P] \longrightarrow [\![E_1']\!]_\rho[P'] \mid [\![E_1'']\!]_\rho[Q] \mid E_1^{iv}[P] \equiv R', \text{ or}$$

- $[\![E_1'']\!]_\rho[\bullet] = [\![E_1^{iv}[\bullet]]\!]_\rho \mid [\![E_1^v]\!]_\rho[\overline{x}.P]$ and it follows:

$$R \equiv [\![E_1^{iv}]\!]_\rho[P'] \mid E_1^v[\overline{x}.P] \mid \;\mid E_1^{''}[x.Q] \mid M_3' \longrightarrow [\![E_1^{iv}]\!]_\rho[P'] \mid E_1^v[P] \mid \;\mid E_1^{''}[Q] \mid M_3' \equiv R'.$$

B) By using Rule (R-Sub-Upd):

$$R \equiv E\Big[C\big[l[P]\big] \mid D\big[l\langle\!\langle (X).Q\rangle\!\rangle.S\big]\Big] \longrightarrow E\Big[C\big[\mathbf{0}\big] \mid D\big[Q\{^P\!/X\} \mid S\big]\Big] \equiv R'.$$

Therefore, we have that $R \equiv E'[l\langle\!\langle (X).Q\rangle\!\rangle.S]$ such that $E'[\bullet] = E[C[l[P]] \mid D[\bullet]]$. In (3.16), based on Lemma 3.2.15, the following holds:

(i) $M_3' \equiv E'''[l\langle\!\langle (X).Q\rangle\!\rangle.S]$, or

(ii) $P' \equiv E'''[l\langle\!\langle (X).Q\rangle\!\rangle.S]$.

By Definition 3.2.3, for every process $P_1$ a location name in $[\![P_1]\!]_\varepsilon$ is either a transaction name or a reserved name $p_{s,\rho}$ for some $s, \rho$. Therefore, if interaction on them exists then they should be part of some process $I_s^{(p)}([\![P_s']\!]_{s,\rho}, [\![Q_s']\!]_\varepsilon)$, i.e.,

$$I_s^{(1)}([\![P_s']\!]_{s,\rho}, [\![Q_s']\!]_\varepsilon) \;\equiv s\big[[\![P_s']\!]_{s,\rho''}\big] \mid s\langle\!\langle (Y).s[Y] \mid \mathtt{ch}(s,Y) \\ \mid \mathtt{outd}^{\mathtt{s}}(p_{s,\rho''}\,,\, p_{\rho''}, \mathtt{nl}(p_{s,\rho''}, Y), s\langle\!\langle \dagger\rangle\!\rangle.\overline{h_s})\rangle\!\rangle \mid p_{\rho'}[[\![Q_s']\!]_\varepsilon]$$

(cf. Figure 3.6 for the other forms). This directly provides that process in the form $s[P]$ (i.e., $p_{s,\rho}[P]$) and update $s\langle\!\langle (X).Q\rangle\!\rangle.S$ (i.e., $p_{s,\rho}\langle\!\langle (X).Q\rangle\!\rangle.S$) have to be in parallel composition. In the following, we analyze cases (i) and (ii). It should be noted that in all obtained cases and sub-cases, except sub-case (2.2.2) below, we have that process $R'$ corresponds to the case I) from the statement. Process $R'$ obtained in (2.2.2) corresponds to the case II).

(1) If (i) holds, then $R \equiv [\![E_1]\!]_\rho[P'] \mid E'''[l\langle\!\langle (X).Q\rangle\!\rangle.S]$, cf. (3.16). In the following we analyze where location $l[P]$ can occur. We have the following cases:

- $E'''[\bullet] = E_1'''[\bullet] \mid l[P]$ and for this case it follows that:

$$R' \equiv [\![E_1]\!]_\varepsilon\left[P'\right] \mid M_3'' \text{ where } M_3'' = E_1'''[Q\{^P\!/X\} \mid S] \mid \mathbf{0}, \text{ or}$$

- $[\![E_1]\!]_\rho[\bullet] \equiv E_2[\bullet] \mid l[P] \mid M_3'$

$$R' \equiv [\![E_2]\!]_\varepsilon\left[P'\right] \mid M_3'' \text{ where } M_3'' = \mathbf{0} \mid E'''[Q\{^P\!/X\} \mid S].$$

(2) If (ii) holds then

$$E'''[l\langle\!\langle (X).Q\rangle\!\rangle.R] \;\equiv [\![G_1]\!]_\rho[[\![C_1]\!]_{\rho'}[I_t^{(1)}([\![P_t']\!]_{t,\rho''}, [\![Q_t']\!]_\varepsilon)] \mid [\![D_1]\!]_{\rho'}[h_t.[\![S_t]\!]_{\rho'''}] \mid M_1'] \mid M_2'.$$

In the following analysis we consider two sub-cases:

(2.1) By exploiting (i) it holds that $[\![G_1]\!]_\rho[P''] \mid M_2'$ where

$$\begin{aligned} P'' &\equiv [\![C_1]\!]_{\rho'}\left[I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon)\right] \mid [\![D_1]\!]_{\rho'}\left[h_t.[\![S_t]\!]_{\rho'''}\right] \mid M_1' \text{ and} \\ M_2' &\equiv E_1'''[l\langle\!\langle(X).Q'\rangle\!\rangle.R']. \end{aligned}$$

We have that $E'''[\bullet] = E_1'''[\bullet] \mid l[P]$ and for this case the following holds:

$$R' \equiv [\![E_1]\!]_\varepsilon\left[[\![G_1]\!]_\rho\left[P''\right] \mid M_2''\right] \mid M_3'.$$

(2.2) By exploiting case (ii) it holds that $[\![G_1]\!]_\rho[P''] \mid M_2'$

$$\begin{aligned} P'' &\equiv E'''[l\langle\!\langle(X).Q\rangle\!\rangle.R] \\ &\equiv [\![C_1]\!]_{\rho'}\left[I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon)\right] \mid [\![D_1]\!]_{\rho'}\left[h_t.[\![S_t]\!]_{\rho'''}\right] \mid M_1'. \end{aligned}$$

We consider the following two sub-cases:

(2.2.1) By exploiting case (i) it holds $[\![C_1]\!]_\rho[P'''] \mid M_1'$ for

$$\begin{aligned} P''' &\equiv [\![C_1]\!]_{\rho'}\left[I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon)\right] \mid [\![D_1]\!]_{\rho'}\left[h_t.[\![S_t]\!]_{\rho'''}\right] \\ M_1' &\equiv E_1'''[l\langle\!\langle(X).Q'\rangle\!\rangle.S]. \end{aligned}$$

We have that $E'''[\bullet] = E_1'''[\bullet] \mid l[P]$ then the following holds:

$$R' \equiv [\![E_1]\!]_\varepsilon\left[[\![G_1]\!]_\rho\left[P''' \mid M_1''\right] \mid M_2'\right] \mid M_3'.$$

(2.2.2) By exploiting case (ii) it holds $[\![C_1]\!]_\rho[P'''] \mid Q'''$ for

$$\begin{aligned} P''' &\equiv E'''[l\langle\!\langle(X).Q\rangle\!\rangle.R] \equiv I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon) \text{ and} \\ Q''' &\equiv [\![D_1]\!]_{\rho'}\left[h_t.[\![S_t]\!]_{\rho'''}\right] \mid M_1', \end{aligned}$$

and follows directly that

$$\begin{aligned} I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon) &\longrightarrow I_t^{(2)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon) \text{ for } \mathtt{nl}(p_{t,\rho''},[\![P_t]\!]_{t,\rho''}) > 0, \text{ or} \\ I_t^{(1)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon) &\longrightarrow I_t^{(2)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon) \text{ for } \mathtt{nl}(p_{t,\rho''},[\![P_t]\!]_{t,\rho''}) = 0. \end{aligned}$$

Therefore, process $R'$ is as presented in the following, where $I_t^{(2)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon$ has an appropriate form, that is described above:

$$\begin{aligned} R' \equiv [\![E_1]\!]_\varepsilon[[\![G_1]\!]_\rho[[\![C_1]\!]_{\rho'}[I_t^{(2)}([\![P_t']\!]_{t,\rho''},[\![Q_t']\!]_\varepsilon] \mid [\![D_1]\!]_{\rho'}[h_t.[\![S_t]\!]_{\rho'''}] \\ \mid M_1'] \mid M_2'] \mid M_3 \end{aligned} \tag{3.17}$$

■

The following lemma formalizes all possible forms for the process $O_u^{(q)}([\![F]\!]_{\rho''}[h_u.[\![P_u]\!]_{\rho'''}],[\![Q_u']\!]_\varepsilon)$ for $m \geq 0$ and $q \in \{1,\ldots,m+4\}$ .

**Definition 3.2.8.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}],[\![Q]\!]_\varepsilon)$ (Figure 3.7) depending on $m = \mathtt{nl}(p_{u,\rho},[\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}])$:

1. for $m = 0$ we have $q \in \{1,2,3,4\}$, and
2. for $m > 0$ and $[\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}] = \prod_{k=1}^{m} p_{u,\rho}[[\![P_k']\!]_\varepsilon] \mid S$ we have $q \in \{1,\ldots,m+4\}$.

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}], \llbracket Q\rrbracket_\varepsilon)$, $\mathtt{nl}(p_{u,\rho}, \llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]) = 0$ |
| $(1)$ | $u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid \mathtt{extrd}\langle\langle u, p_{u,\rho}, p_\rho\rangle\rangle \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ <br> $\equiv u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid u\langle\langle(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid u\langle\langle\dagger\rangle\rangle.\overline{h_u}\rangle\rangle \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |
| $(2)$ | $u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid h_u \mid u\langle\langle\dagger\rangle\rangle.\overline{h_u} \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |
| $(3)$ | $h_u \mid \overline{h_u} \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |
| $(4)$ | $p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}], \llbracket Q\rrbracket_\varepsilon)$, $\mathtt{nl}(p_{u,\rho}, \llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]) > 0$ |
| $(1)$ | $u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid \mathtt{extrd}\langle\langle u, p_{u,\rho}, p_\rho\rangle\rangle \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ <br> $\equiv u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid u\langle\langle(Y).u[Y] \mid \mathsf{ch}(u,Y)$ <br> $\mid \mathtt{outd}^{\mathsf{s}}(p_{u,\rho}\,,\, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), u\langle\langle\dagger\rangle\rangle.\overline{h_u})\rangle\rangle \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |
| $(j+2)$ <br> $0 \le j \le m-1$ | $u\big[\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\langle\langle(X_1,\dots,X_{m-j}).$ <br> $(\prod_{k=1}^{m-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\llbracket P'_k\rrbracket_\varepsilon] \mid u\langle\langle\dagger\rangle\rangle.\overline{h_u})\rangle\rangle \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |
| $(m+2)$ | $u\big[\llbracket F'\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}]\big] \mid \prod_{k=1}^{m} p_\rho[\llbracket P'_k\rrbracket_\varepsilon] \mid h_u \mid p_\rho[\llbracket Q\rrbracket_\varepsilon] \mid u\langle\langle\dagger\rangle\rangle.\overline{h_u}$ |
| $(m+3)$ | $\prod_{k=1}^{m} p_\rho[\llbracket P'_k\rrbracket_\varepsilon] \mid h_u \mid p_\rho[\llbracket Q\rrbracket_\varepsilon] \mid \overline{h_u}$ |
| $(m+4)$ | $\prod_{k=1}^{m} p_\rho[\llbracket P'_k\rrbracket_\varepsilon] \mid p_\rho[\llbracket Q\rrbracket_\varepsilon]$ |

Figure 3.7: Process $O_u^{(q)}(\llbracket F\rrbracket_\rho[h_u.\llbracket P\rrbracket_{\rho'}], \llbracket Q\rrbracket_\varepsilon)$ with $q \ge 1$.

We now continue with the analysis of adaptable processes that can be obtained starting from the translation of a transaction that contains failure signal in its body, which is triggered internally.

**Lemma 3.2.17.** Let $P_1$ be a well-formed compensable process such that

- $\llbracket P_1\rrbracket_\varepsilon \equiv \llbracket E\rrbracket_\varepsilon\,[\llbracket G\rrbracket_\rho\,[\llbracket L\rrbracket_{\rho'}\,[\llbracket u[F[\overline{u}.P_u], Q_u]\rrbracket_{\rho''}] \mid M_1] \mid M_2] \mid M_3$, and
- $\llbracket P_1\rrbracket_\varepsilon \longrightarrow^{n-1} R$,

$$R \equiv \llbracket E_1\rrbracket_\varepsilon\Big[\llbracket G_1\rrbracket_\rho\big[\llbracket L_1\rrbracket_{\rho'}\,[O_u^{(q)}(\llbracket F_1\rrbracket_{\rho''}[h_u.\llbracket P_u\rrbracket_{t,\rho'''}], \llbracket Q'_u\rrbracket_\varepsilon)] \mid M'_1\big] \mid M'_1\Big] \mid M'_3,$$

where $O_u^{(q)}(\llbracket F\rrbracket_{\rho''}[h_u.\llbracket P_u\rrbracket_{t,\rho'''}], \llbracket Q_u\rrbracket_\varepsilon)$ in $R$ is a process from Definition 3.2.8.
If $R \longrightarrow R'$ then either

I) $R' \equiv \llbracket E_1\rrbracket_\varepsilon\,\Big[\llbracket G_1\rrbracket_\rho\,\Big[\llbracket L_1\rrbracket_{\rho'}\,\Big[O_u^{(q+1)}(\llbracket F_1\rrbracket_{\rho''}[h_u.\llbracket P_u\rrbracket_{t,\rho'''}], \llbracket Q'_u\rrbracket_\varepsilon)\Big] \mid M'_1\Big] \mid M'_2\Big] \mid M'_3$, or

II) $R' \equiv \llbracket E_2\rrbracket_\varepsilon\,\Big[\llbracket G_2\rrbracket_\rho\,\Big[\llbracket L_2\rrbracket_{\rho'}\,\Big[O_u^{(q)}(\llbracket F_2\rrbracket_{\rho''}[h_u.\llbracket P_u\rrbracket_{t,\rho'''}], \llbracket Q''_u\rrbracket_\varepsilon)\Big] \mid M''_1\Big] \mid M''_2\Big] \mid M''_3$,

where:

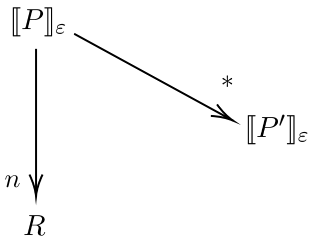



Figure 3.8: Diagram of Lemma 3.2.19.

- $n > 1$;
- $\rho$ is the path to hole in $\llbracket E \rrbracket_\varepsilon [\bullet]$;
- $\rho'$ is the path to hole in $\llbracket G \rrbracket_\rho [\bullet]$ and $\llbracket G_k \rrbracket_\rho [\bullet]$ and $k \in \{1,2\}$;
- $\rho''$ is the path to the hole in $\llbracket L \rrbracket_{\rho'} [\bullet]$; $\llbracket L_k \rrbracket_{\rho'} [\bullet]$ and $k \in \{1,2\}$;
- $\rho'''$ is the path to the hole in $\llbracket F \rrbracket_{\rho''} [\bullet]$ and $\llbracket F_k \rrbracket_{\rho'} [\bullet]$ and $k \in \{1,2\}$.

*Proof.* By Definition 3.2.3, we get

$$\begin{aligned}\llbracket P_1 \rrbracket_\varepsilon \equiv & \llbracket E \rrbracket_\varepsilon \Big[ \llbracket G \rrbracket_\rho \big[ \llbracket L \rrbracket_{\rho'} \big[ u \big[ \llbracket F \rrbracket_{\rho''} [\overline{u}.h_u.\llbracket P_u \rrbracket_{\rho'''}] \big] \\ & \mid u.(\texttt{extrd}\langle\langle u, p_{u,\rho''}, p_{\rho''} \rangle\rangle \mid p_{\rho''}[\llbracket Q_u \rrbracket_\varepsilon]) \big] \mid M_1 \big] \mid M_2 \Big] \mid M_3,\end{aligned}$$

and the proof continues by case analysis for the step, $R \longrightarrow R'$, that can be realized. The proof follows the same idea that is presented for the proof of Lemma 3.2.16. ∎

**Remark 3.2.18.** We will use the following abbreviations, where we use $i, c, k, w$ as indexes of $t, u$ and $F$:

$$\begin{aligned} I^{(p)}_{t_{i,k,w}} &= I^{(p)}_{t_{i,k,w}}(\llbracket P'_{t_{i,k,w}} \rrbracket_{t,\rho''}, \llbracket Q'_{t_{i,k,w}} \rrbracket_\varepsilon), \\ O^{(q)}_{u_{c,k,w}} &= O^{(q)}_{u_{c,k,w}}(\llbracket F_{c,k,w} \rrbracket_{\rho''}[h_{u_{c,k,w}}.\llbracket P_{u_{c,k,w}} \rrbracket_{\rho'''}], \llbracket Q'_{u_{c,k,w}} \rrbracket_\varepsilon).\end{aligned}$$

The following lemma is crucial for the proof of soundness, and it is illustrated in Figure 3.8. In Figure 3.8 we have $\llbracket P \rrbracket_\varepsilon \longrightarrow^n R$ and lemma will provide the shape of process $R$. On the other side, lemma will provide the shape of the process $P'$. The proof of soundness will provide that by successive application of completeness on the derivation $P \longrightarrow^* P'$ it holds that $\llbracket P \rrbracket_\varepsilon \longrightarrow^* \llbracket P' \rrbracket_\varepsilon$.

**Lemma 3.2.19.** Let $I^{(p)}_{t_{i,k,w}}$ and $O^{(q)}_{u_{c,k,w}}$ be processes from Definition 3.2.7 and Definition 3.2.8, respectively. If $\llbracket P \rrbracket_\varepsilon \longrightarrow^n R$, with $n \geq 1$, then

1)

$$\begin{aligned} R \equiv \prod_{w=1}^{z} \llbracket E_w \rrbracket_\varepsilon \Big[ \prod_{k=1}^{s_w} \llbracket G_{k,w} \rrbracket_{\rho_w} \big[ \prod_{i=1}^{l_k} \llbracket C_{i,k,w} \rrbracket_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k} \llbracket D_{j,k,w} \rrbracket_{\rho'_{k,w}} [h_{t_{j,k,w}}.\llbracket S_{t_{j,k,w}} \rrbracket_{\rho''_{k,w}}] \\ \mid \prod_{c=1}^{m_k} \llbracket L_{c,k,w} \rrbracket_{\rho'_{k,w}} [O^{(q)}_{u_{c,k,w}}] \big] \Big] \end{aligned} \tag{3.18}$$

and $P \longrightarrow^* P'$, where $P'$ is of the following form:

2)

$$P' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}},Q_{t_{i,k,w}}]\big]\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big] \\ \mid \prod_{c=1}^{m_k} L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}],Q_{u_{c,k,w}}]\big]\Big], \tag{3.19}$$

for some $E_w[\bullet]$, $G_{k,w}[\bullet]$, $C_{i,k,w}[\bullet]$, $D_{j,k,w}[\bullet]$, and $L_{c,k,w}[\bullet]$ where $w \in \{1,\dots,z\}$, $k \in \{1,\dots,s_w\}$, $i \in \{1,\dots,l_k\}$, $j \in \{1,\dots,r_k\}$, and $c \in \{j,\dots,m_k\}$.

*Proof.* The proof proceeds by induction on $n$.

***Base case:*** Assume that $n = 1$, i.e. $[\![P]\!]_\varepsilon \longrightarrow R$. By application of Lemma 3.2.12 there are three possible cases:

- *Case* $P \equiv E'[C'[a.P_2] \mid D'[\overline{a}.P_1]]$ and $R \equiv [\![E']\!]_\varepsilon\Big[[\![C']\!]_\rho[[\![P_1]\!]_{\rho'}] \mid [\![D']\!]_\rho[[\![P_2]\!]_{\rho''}]\Big]$:
  In this case we have: $z = 1$ and $s_1 = 0$ and it holds $E_1[\bullet] = [\bullet] \mid E'\big[C'[a.P_2] \mid D'[\overline{a}.P_1]\big]$ and $P = P'$.
- *Case* $P \equiv E'\Big[C'[t[P_2,Q]] \mid D'[\overline{t}.P_1]\Big]$ and
  $R \equiv [\![E']\!]_\varepsilon\Big[[\![C']\!]_\rho[t\big[[\![P_2']\!]_{t,\rho'}\big] \mid \texttt{extrd}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]] \mid [\![D']\!]_\rho[h_t.[\![P_1]\!]_{\rho''}]\Big]$:
  In this case we have: $z = 1, s_1 = 1, l_1 = 1, r_1 = 1$ and $m_1 = 0$. Therefore, the following holds: $E_1[\bullet] = [\bullet], G_{1,1}[\bullet] = E'[\bullet]\ C_{1,1,1}[\bullet] = C'[\bullet], D_{1,1,1}[\bullet] = D'[\bullet], P_{t_{1,1,1}} = P_2, Q_{t_{1,1,1}} = Q, S_{t_{1,1,1}} = P_1$ and $I^{(1)}_{t_{1,1,1}} \equiv t\big[[\![P_2']\!]_{t,\rho'}\big] \mid \texttt{extrd}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]$ and $P = P'$.
- *Case* $P \equiv C'[u[D'[\overline{u}.P_1],Q]]$ and
  $R \equiv [\![C']\!]_\varepsilon\Big[u\big[[\![D']\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}]\big] \mid \texttt{extrd}\langle\!\langle u, p_{u,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]\Big]$:
  In this case we have: $z = 1, s_1 = 1, l_1 = 0, r_1 = 0$ and $m_1 = 1$. Therefore, the following holds: $E_1[\bullet] = [\bullet], G_{1,1}[\bullet] = C'[\bullet], L_{1,1,1}[\bullet] = D'[\bullet], P_{u_{1,1,1}} = P_1, Q_{u_{1,1,1}} = Q$ and
  $I^{(1)}_{u_{1,1,1}} \equiv u\big[[\![D']\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}]\big] \mid \texttt{extrd}\langle\!\langle u, p_{u,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]$ and $P = P'$.

***Inductive hypothesis:*** Assume that the statement holds for $n-1$ reduction steps, i.e. if $[\![P]\!]_\varepsilon \longrightarrow^{n-1} R_1$ then the statement holds.

***Inductive step:*** We consider that $[\![P]\!]_\varepsilon \longrightarrow^{n-1} R_1 \longrightarrow R$. We know, by inductive hypothesis:

1) $R_1$ has the following form:

$$R_1 \equiv \prod_{w=1}^{z} [\![E_w]\!]_\varepsilon\Big[\prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w}\big[\prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}}\big[I^{(p)}_{t_{i,k,w}}\big] \\ \mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}}\big[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\big] \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}}\big[O^{(q)}_{u_{c,k,w}}\big]\big]\Big],$$

2) $P \longrightarrow^* P''$ such that $P''$ has the following form:

$$P'' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}},Q_{t_{i,k,w}}]\big]\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big] \\ \mid \prod_{c=1}^{m_k} L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}],Q_{u_{c,k,w}}]\big]\Big].$$

We continue with the proof by case analysis for the last step, $R_1 \longrightarrow R$, that can be realized. In the following we consider six interesting cases.

(1) Let $I_t^{(1)}$ is a process that has the form as presented in Definition 3.2.7 where $t = t_{1,1,1}, G_1[\bullet] = G_{1,1}[\bullet], C_1[\bullet] = C_{1,1,1}[\bullet], D_1[\bullet] = D_{1,1,1}[\bullet]$ and

$$R_1 \equiv [\![E_1]\!]_\varepsilon [[\![G_{1,1}]\!]_{\rho_1} [[\![C_{1,1,1}]\!]_{\rho_{1,1}} [I^{(1)}_{t_{1,1,1}}([\![P'_{t_{1,1,1}}]\!]_{t,\rho''}, [\![Q'_{t_{1,1,1}}]\!]_\varepsilon)] \\ \mid [\![D_{1,1,1}]\!]_{\rho_{1,1}} [h_{t_{1,1,1}}.[\![S_{t_{1,1,1}}]\!]_{\rho''_{1,1}}] \mid M'_1] \mid M'_2] \mid M'_3.$$

According to the Lemma 3.2.16, it follows that $R_1 \longrightarrow R$ such that $R$ has the form

I)

$$R \equiv [\![E_1]\!]_\varepsilon [[\![G_{1,1}]\!]_\rho [[\![C_{1,1,1}]\!]_{\rho'} [I^{(2)}_{t_{1,1,1}}([\![P'_{t_{1,1,1}}]\!]_{t_{1,1,1},\rho''}, [\![Q'_{t_{1,1,1}}]\!]_\varepsilon)] \\ \mid [\![D_1]\!]_{\rho'} [h_{t_{1,1,1}}.[\![S_{t_{1,1,1}}]\!]_{\rho'''}] \mid M'_1] \mid M'_2] \mid M'_3, \text{or}$$

II)

$$R \equiv [\![E'_1]\!]_\varepsilon [[\![G'_{1,1}]\!]_\rho [[\![C'_{1,1,1}]\!]_{\rho'} [I^{(1)}_{t_{1,1,1}}([\![P''_{t_{1,1,1}}]\!]_{t_{1,1,1},\rho''}, [\![Q''_{t_{1,1,1}}]\!]_\varepsilon)] \\ \mid [\![D'_{1,1,1}]\!]_{\rho'} [h_{t_{1,1,1}}.[\![S_{t_{1,1,1}}]\!]_{\rho'''}] \mid M''_1] \mid M''_2] \mid M''_3. \tag{3.20}$$

Here we comment case II), while case I) follows the idea that is given in case a) from Base case.
In case when we get the form II) it directly follows that $P'' = P'$.
Similarly, for all $I_t^{(1)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)$ with $t \in \{t_{2,1,1}, \dots, t_{l_{s_z},s_z,z}\}$.
Similarly, for cases $I^{(p)}_{t_{1,1,1}}([\![P'_{t_{1,1,1}}]\!]_{t_{1,1,1},\rho''}, [\![Q'_{t_{1,1,1}}]\!]_\varepsilon)$ where $p \in \{2, 4, \dots, m+3\}$.

(2) Let $O_u^{(1)}$ is a process that has a form as presented in Definition 3.2.8, where $u = u_{1,1,1}, G_1[\bullet] = G_{1,1}[\bullet], \mathcal{L}_1[\bullet] = L_{1,1,1}[\bullet], F_1[\bullet] = F_{1,1,1}[\bullet]$ and

$$R_1 \equiv [\![E_1]\!]_\varepsilon [[\![G_{1,1}]\!]_{\rho_1} [[\![L_{1,1,1}]\!]_{\rho'_{1,1}} [O^{(1)}_{u_{1,1,1}}([\![F_{1,1,1}]\!]_{\rho''}[h_{u_{1,1,1}}.[\![P_{u_{1,1,1}}]\!]_{u_{1,1,1},\rho'''}], [\![Q'_{u_{1,1,1}}]\!]_\varepsilon)] \\ \mid M'_1] \mid M'_2\Big] \mid M'_3.$$

According to the Lemma 3.2.17, it follows that $R_1 \longrightarrow R$ such that $R$ has the form

I)

$$R \equiv [\![E_1]\!]_\varepsilon \Big[[\![G_{1,1}]\!]_\rho [[\![L_{1,1,1}]\!]_{\rho'} [O^{(2)}_u([\![F_{1,1,1}]\!]_{\rho''}[h_{u_{1,1,1}}.[\![P_{u_{1,1,1}}]\!]_{t,\rho'''}], [\![Q'_{u_{1,1,1}}]\!]_\varepsilon)] \\ \mid M'_1] \mid M'_2\Big] \mid M'_3 \ \text{ or}$$

II)

$$R \equiv [\![E'_1]\!]_\varepsilon \Big[[\![G'_{1,1}]\!]_\rho [[\![L'_{1,1,1}]\!]_{\rho'} [O^{(1)}_{u_{1,1,1}}([\![F'_{1,1,1}]\!]_{\rho''}[h_u.[\![P_{u_{1,1,1}}]\!]_{u_{1,1,1},\rho'''}], [\![Q''_{u_{1,1,1}}]\!]_\varepsilon)] \\ \mid M''_1] \mid M''_2\Big] \mid M''_3. \tag{3.21}$$

Here we comment on the case II), while case I) follows the idea that is given in case a) from the base case.
In case when we get the form II) it directly follows that $P'' = P'$.
Similarly, for all $O_u^{(1)}([\![F]\!]_{\rho''}[h_u.[\![P_u]\!]_{t,\rho'''}], [\![Q'_u]\!]_\varepsilon)$ with $u \in \{u_{2,1,1}, \dots, u_{m_{s_z},s_z,z}\}$.
Similarly, for cases $O_u^{(q)}([\![F]\!]_{\rho''}[h_u.[\![P_u]\!]_{t,\rho'''}], [\![Q'_u]\!]_\varepsilon)$ where $q \in \{2, 4, \dots, m+4\}$.

(3) Let $I_t^{(3)}$ is a process that has the form as presented in Definition 3.2.7, where where $t = t_{1,1,1}, G_1[\bullet] = G_{1,1}[\bullet], C_1[\bullet] = C_{1,1,1}[\bullet], D_1[\bullet] = D_{1,1,1}[\bullet]$ and

$$R_1 \equiv [\![E_1]\!]_\varepsilon \Big[ [\![G_{1,1}]\!]_{\rho_1} \big[ [\![C_{1,1,1}]\!]_{\rho_{1,1}} \big[ \overline{h_{t_{1,1,1}}} \mid p_{\rho'_{1,1}}[ [\![Q'_{t_{1,1,1}}]\!]_\epsilon ] \big] \mid [\![D_{1,1,1}]\!]_{\rho_{1,1}} \big[ h_{t_{1,1,1}} . [\![S_{t_{1,1,1}}]\!]_{\rho''_{1,1}} \big] \mid M'_1 \big] \mid M'_2 \Big] \mid M'_3.$$

According to the Lemma 3.2.16, it follows that we can derive $R_1 \longrightarrow R$ such that $R$ has the form of (3.20) or

$$R \equiv [\![E_1]\!]_\varepsilon \Big[ [\![G_{1,1}]\!]_{\rho_1} \big[ [\![C_{1,1,1}]\!]_{\rho_{1,1}} \big[ p_{\rho'_{1,1}}[ [\![Q'_{t_{1,1,1}}]\!]_\epsilon ] \big] \mid [\![D_{1,1,1}]\!]_{\rho_{1,1}} \big[ [\![S_{t_{1,1,1}}]\!]_{\rho''_{1,1}} \big] \mid M'_1 \big] \mid M'_2 \Big] \mid M'_3. \tag{3.22}$$

In case when we get the form (3.22) it holds that $P'' \longrightarrow P'$ where:

$$\begin{aligned}
P' &\equiv E_1 \Big[ G_{1,1} \big[ C_{1,1,1}[\langle Q'_{t_{1,1,1}} \rangle] \mid D_{1,1,1}[S_{t_{1,1,1}}] \mid M'_1 \big] \mid M'_2 \Big] \mid M'_3 \\
&\equiv E_1 \Big[ G_{1,1} \big[ \prod_{i=2}^{l_1} C'_{i,1,1} \big[ t_{i,1,1}[P_{t_{i,1,1}}, Q_{t_{i,1,1}}] \big] \mid \prod_{j=2}^{r_1} D'_{j,1,1} \big[ \overline{t_{j,1,1}} . S_{t_{j,1,1}} \big] \\
&\quad \mid \prod_{c=1}^{m_1} L_{c,1,1} \big[ u_{c,1,1}[F_{c,1,1}[\overline{u_{c,1,1}} . P_{u_{c,1,1}}], Q_{u_{c,1,1}}] \big] \Big] \\
&\quad \mid \prod_{w=2}^{z} E_w \Big[ \prod_{k=1}^{s_w} G_{k,w} \big[ \prod_{i=1}^{l_k} C_{i,k,w} \big[ t_{i,k,w}[P_{t_{i,k,w}}, Q_{t_{i,k,w}}] \big] \mid \prod_{j=1}^{r_k} D_{j,k,w} \big[ \overline{t_{j,k,w}} . S_{t_{j,k,w}} \big] \\
&\quad \mid \prod_{c=1}^{m_k} L_{c,k,w} \big[ u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}} . P_{u_{c,k,w}}], Q_{u_{c,k,w}}] \big] \Big],
\end{aligned}$$

such that:

- $C'_{2,1,1}[\bullet] = C_{2,1,1}[\bullet] \mid N$, where $N = C_{1,1,1}[\langle Q'_{t_{1,1,1}} \rangle]$ and $C'_{i,1,1}[\bullet] = C_{i,1,1}[\bullet]$ for $i \in \{3, \ldots, l_1\}$, and
- $D'_{2,1,1}[\bullet] = D_{2,1,1}[\bullet] \mid N_1$, where $N_1 = D_{1,1,1}[S_{t_{1,1,1}}]$ and $D'_{i,1,1}[\bullet] = D_{i,1,1}[\bullet]$ for $i \in \{3, \ldots, l_1\}$.

Similarly, for all $I_t^{(3)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)$with $t \in \{t_{2,1,1}, \ldots, t_{l_{s_z}, s_z, z}\}$.
Similarly, for $I_t^{(m+3)}([\![P'_t]\!]_{t,\rho''}, [\![Q'_t]\!]_\varepsilon)$ in Definition 3.2.7.

(4) $O_u^{(3)}$ is a process that has a form as presented in Definition 3.2.8, where, where $u = u_{1,1,1}, G_1[\bullet] = G_{1,1}[\bullet], \mathcal{L}_1[\bullet] = L_{1,1,1}[\bullet], F_1[\bullet] = F_{1,1,1}[\bullet]$ and

$$R_1 \equiv [\![E_1]\!]_\varepsilon \Big[ [\![G_{1,1}]\!]_{\rho_1} \Big[ [\![L_{1,1,1}]\!]_{\rho'_{1,1}} \Big[ h_{u_{1,1,1}} \mid \overline{h_{u_{1,1,1}}} \mid p_{\rho''}[ [\![Q'_{u_{1,1,1}}]\!]_\varepsilon ] \Big] \mid M'_1 \Big] \mid M'_2 \Big] \mid M'_3.$$

According to the proof of Lemma 3.2.17, it follows that and $R_1 \longrightarrow R$ such that $R$ has the form (3.21) or

$$R_1 \equiv [\![E_1]\!]_\varepsilon \Big[ [\![G_{1,1}]\!]_{\rho_1} \Big[ [\![L_{1,1,1}]\!]_{\rho'_{1,1}} \Big[ p_{\rho''}[ [\![Q'_{u_{1,1,1}}]\!]_\varepsilon ] \Big] \mid M'_1 \Big] \mid M'_2 \Big] \mid M'_3. \tag{3.23}$$

In case when we get the form (3.23) it holds $P'' \longrightarrow P'$ where

$$P' \equiv E_1 \Big[ G_{1,1} \big[ L_{1,1,1}[\langle Q'_{u_{1,1,1}} \rangle] \mid M'_1 \big] \mid M'_2 \Big] \mid M'_3$$

$$\equiv E_1\Big[G_{1,1}[\prod_{i=1}^{l_1} C_{i,1,1}[t_{i,1,1}[P_{t_{i,1,1}},Q_{t_{i,1,1}}]] \mid \prod_{j=1}^{r_1} D_{j,1,1}[\overline{t_{j,1,1}}.S_{t_{j,1,1}}]$$
$$\mid \prod_{c=2}^{m_1} L'_{c,1,1}[u_{c,1,1}[F_{c,1,1}[\overline{u_{c,1,1}}.P_{u_{c,1,1}}],Q_{u_{c,1,1}}]]\Big]$$
$$\mid \prod_{w=2}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}[\prod_{i=1}^{l_k} C_{i,k,w}[t_{i,k,w}[P_{t_{i,k,w}},Q_{t_{i,k,w}}]] \mid \prod_{j=1}^{r_k} D_{j,k,w}[\overline{t_{j,k,w}}.S_{t_{j,k,w}}]$$
$$\mid \prod_{c=1}^{m_k} L_{c,k,w}[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}],Q_{u_{c,k,w}}]]\Big],$$

such that $L'_{2,1,1}[\bullet] = L_{2,1,1}[\bullet] \mid N$, where $N = L_{1,1,1}[\langle Q'_{u_{1,1,1}}\rangle]$ and $L'_{i,1,1}[\bullet] = L_{i,1,1}[\bullet]$ for $i \in \{3,\ldots,m_1\}$,

Similarly, for all $O_u^{(3)}([\![F]\!]_{\rho''}[h_u.[\![P_u]\!]_{t,\rho'''}],[\![Q'_u]\!]_\varepsilon)$ with $u \in \{u_{2,1,1},\ldots,u_{m_{s_z}s_z z}\}$.
Similarly, for case of $O_u^{(m+3)}([\![F]\!]_{\rho''}[h_u.[\![P_u]\!]_{t,\rho'''}],[\![Q'_u]\!]_\varepsilon)$ in Definition 3.2.8.

(5) In this case let us consider the following context:

$$G_{1,1}[\bullet] = G'_{1,1}[\bullet] \mid C_{(l_1+1)1,1}[t_{(l_1+1)1,1}[P_{t_{(l_1+1)1,1}},Q_{t_{(l_k+1)1,1}}]] \mid D_{(r_1+1)1,1}[\overline{t_{(r_1+1)1,1}}.S_{t_{(r_1+1)1,1}}]. \tag{3.24}$$

Therefore, the following holds:

$$R_1 \equiv [\![E_1]\!]_\varepsilon\Big[[\![G_{1,1}]\!]_{\rho_1}[\prod_{i=1}^{l_k}[\![C_{i,1,1}]\!]_{\rho'_{1,1}}[I^{(p)}_{t_{i,1,1}}] \mid \prod_{j=1}^{r_k}[\![D_{j,1,1}]\!]_{\rho'_{1,1}}[h_{t_{j,1,1}}.[\![S_{t_{j,1,1}}]\!]_{\rho''_{1,1}}] \mid M'_1] \mid M'_2\Big] \mid M'_3$$
$$\equiv [\![E_1]\!]_\varepsilon\Big[[\![G'_{1,1}]\!]_{\rho_1}[\prod_{i=1}^{l_k}[\![C_{i,1,1}]\!]_{\rho'_{1,1}}[I^{(p)}_{t_{i,1,1}}] \mid \prod_{j=1}^{r_k}[\![D_{j,1,1}]\!]_{\rho'_{1,1}}[h_{t_{j,1,1}}.[\![S_{t_{j,1,1}}]\!]_{\rho''_{1,1}}]$$
$$\mid [\![C_{(l_1+1)1,1}]\!]_{\rho'_{1,1}}[t_{(l_1+1)1,1}[[\![P]\!]_{t_{(l_1+1)1,1},\rho''_{1,1}}]$$
$$\mid t_{(l_1+1)1,1}.(\texttt{extrd}\langle\!\langle t_{(l_1+1)1,1},p_{t_{(l_1+1)1,1},\rho''_{1,1}},p_{\rho''_{1,1}}\rangle\!\rangle \mid p_{\rho''_{1,1}}[[\![Q]\!]_\varepsilon])]$$
$$\mid [\![D_{(r_1+1)1,1}]\!]_{\rho'_{1,1}}[\overline{t_{(r_1+1)1,1}}.h_{t_{(r_1+1)1,1}}.[\![S_{t_{(r_1+1)1,1}}]\!]_{\rho''_{1,1}}] \mid M'_1] \mid M'_2\Big] \mid M'_3.$$

For process $R$, which is obtained from $R_1 \longrightarrow R$, one possible reduction is caused by synchronization on name input $t_{(r_1+1)1,1}$, as presented in the following:

$$\begin{aligned}R \equiv [\![E_1]\!]_\varepsilon\Big[&[\![G'_{1,1}]\!]_{\rho_1}[\prod_{i=1}^{l_k}[\![C_{i,1,1}]\!]_{\rho'_{1,1}}[I^{(p)}_{t_{i,1,1}}] \mid \prod_{j=1}^{r_k}[\![D_{j,1,1}]\!]_{\rho'_{1,1}}[h_{t_{j,1,1}}.[\![S_{t_{j,1,1}}]\!]_{\rho''_{1,1}}]\\ &\mid [\![C_{(l_1+1)1,1}]\!]_{\rho'_{1,1}}[I^{(1)}_{t_{(l_1+1)1,1}}] \mid [\![D_{(r_1+1)1,1}]\!]_{\rho'_{1,1}}[h_{t_{(r_1+1)1,1}}.[\![S_{t_{(r_1+1)1,1}}]\!]_{\rho''_{1,1}}]\\ &\mid M'_1] \mid M'_2\Big] \mid M'_3,\end{aligned} \tag{3.25}$$

where

$$\begin{aligned}I^{(1)}_{t_{(l_1+1)1,1}} = t_{(l_1+1)1,1}&[[\![P_{t_{(l_1+1)1,1}}]\!]_{t_{(l_1+1)1,1},\rho''_{1,1}}] \mid \texttt{extrd}\langle\!\langle t_{(l_1+1)1,1},p_{t_{(l_1+1)1,1},\rho''_{1,1}},p_{\rho''_{1,1}}\rangle\!\rangle\\ &\mid p_{\rho''_{1,1}}[[\![Q_{t_{(l_1+1)1,1}}]\!]_\varepsilon]\end{aligned}$$

In case when we get (3.25) it follows that $P'' = P'$.
It should be noted that here we considered one particular case. Precisely, we consider

scenario where for transaction $t_{(l_1+1)1,1}[P_{t_{(l_1+1)1,1}}, Q_{t_{(l_1+1)1,1}}]$ error notification comes from context $D_{(r_1+1)1,1}[\bullet]$, but that is not the only possible case. For the other cases, when the error notification $\overline{t_{(l_1+1)1,1}}$ comes from some other context, i.e. from $D_{j,1,1}[\bullet], j \in \{1,\ldots,r_1\}$ or $C_{i,1,1}[\bullet], i \in \{1,\ldots,l_1\}$ or $G'_{1,1}[\bullet]$ or $E_1[\bullet]$, discussion follows similarly.

(6) In this case let us consider that:

$$G_{1,1}[\bullet] = G'_{1,1}[\bullet] \mid L_{(m_1+1),1,1}[u_{(m_1+1),1,1}[F_{(m_1+1)1,1}[\overline{u_{(m_1+1),1,1}}.P_{u_{(m_1+1),1,1}}], Q_{u_{(m_1+1),1,1}}]] \tag{3.26}$$

Therefore, the following holds:

$$\begin{aligned} R_1 \equiv \llbracket E_1 \rrbracket_\varepsilon \Big[ & \llbracket G'_{1,1} \rrbracket_{\rho_1} [ \prod_{c=1}^{m_k} \llbracket L_{c,1,1} \rrbracket_{\rho'_{1,1}} [I^{(q)}_{u_{c,1,1}}] \\ & \mid \llbracket L_{(m_1+1),1,1} \rrbracket_{\rho'_{1,1}} [u_{(m_1+1),1,1}[\llbracket F_{(m_k+1),1,1} \rrbracket_{\rho''_{1,1}} [\overline{u_{(m_k+1),1,1}}.h_{u_{(m_1+1),1,1}}.\llbracket P_{u_{(m_1+1),1,1}} \rrbracket_{\rho''_{1,1}}]] \\ & \mid u_{(m_1+1),1,1}.(\mathtt{extrd}\langle\langle u_{(m_1+1),1,1}, p_{u_{(m_1+1),1,1},\rho''_{1,1}}, p_{\rho''_{1,1}} \rangle\rangle \mid p_{\rho''_{1,1}}[\llbracket Q_{u_{(m_1+1),1,1}} \rrbracket_\varepsilon]) \\ & \mid M'_1] \mid M'_2 \Big] \mid M'_3 \end{aligned}$$

For process $R$, which is obtained form $R_1 \longrightarrow R$, one possible reduction is caused by a synchronization on name $u_{(m_1+1),1,1}$, as presented in the following:

$$R \equiv \llbracket E_1 \rrbracket_\varepsilon \Big[ \llbracket G'_{1,1} \rrbracket_{\rho_1} [ \prod_{c=1}^{m_k} \llbracket L_{c,1,1} \rrbracket_{\rho'_{1,1}} [O^{(q)}_{u_{c,1,1}}] \mid \llbracket L_{(m_1+1),1,1} \rrbracket_{\rho'_{1,1}} [O^{(1)}_{u_{(m_1+1),1,1}} \mid M'_1] \mid M'_2 \Big] \mid M'_3, \tag{3.27}$$

where

$$\begin{aligned} O^{(1)}_{u_{(m_1+1),1,1}} = \; & u_{(m_1+1),1,1}[\llbracket F_{(m_1+1),1,1} \rrbracket_{\rho''_{1,1}} [h_{u_{(m_1+1),1,1}}.\llbracket P_{u_{(m_1+1),1,1}} \rrbracket_{\rho''_{1,1}}]] \\ & \mid \mathtt{extrd}\langle\langle u_{(m_1+1),1,1}, p_{u_{(m_1+1),1,1},\rho''_{1,1}}, p_{\rho''_{1,1}} \rangle\rangle \mid p_{\rho''_{1,1}}[\llbracket Q_{u_{(m_1+1),1,1}} \rrbracket_\varepsilon]. \end{aligned}$$

In case when we get (3.27) it follows that $P'' = P'$. ■

**Lemma 3.2.20.** Let processes $I^{(p)}_t(\llbracket P'_t \rrbracket_{t,\rho''}, \llbracket Q'_t \rrbracket_\varepsilon)$ and $O^{(q)}_u(\llbracket F \rrbracket_{\rho''}[h_u.\llbracket P_u \rrbracket_{\rho'''}], \llbracket Q'_u \rrbracket_\varepsilon)$ be defined as in Definition 3.2.7 and Definition 3.2.8. For any contexts $C[\bullet]$, $D[\bullet]$, and $L[\bullet]$ the following holds:

$$C\big[I^{(p)}_t(\llbracket P'_t \rrbracket_{t,\rho''}, \llbracket Q'_t \rrbracket_\varepsilon)\big] \mid D\big[h_t.\llbracket S_t \rrbracket_\rho\big] \longrightarrow^* C\big[\llbracket \mathsf{extr}(P'_t) \rrbracket_{\rho'} \mid \llbracket \langle Q'_t \rangle \rrbracket_{\rho'}\big] \mid D\big[\llbracket S_t \rrbracket_\rho\big] \quad \text{and} \tag{3.28}$$

$$L\big[O^{(q)}_u(\llbracket F \rrbracket_{\rho''}[h_u.\llbracket P_u \rrbracket_{\rho'''}], \llbracket Q'_u \rrbracket_\varepsilon)\big] \longrightarrow^* L\big[\llbracket \mathsf{extr}(F_1[P_u]) \rrbracket_{\rho'} \mid \llbracket \langle Q'_u \rangle \rrbracket_{\rho'}\big] \tag{3.29}$$

*Proof.* The proof proceeds directly by application of the reduction rules from Figure 2.5. ■

**Lemma 3.2.21.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\llbracket P \rrbracket_\varepsilon \equiv \llbracket Q \rrbracket_\varepsilon$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

#### 3.2.3.2.4 Proof of Operational Correspondence — Theorem 3.2.7

Due to the complexity of the proof for Theorem 3.2.7 we first present an overview of the proof.

#### 3.2.3.2.5 A Roadmap for the Proofs

Part (1) of Theorem 3.2.7 is ***completeness***, i.e.,

$$\text{If } P \longrightarrow P' \text{ then } [\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$$

where $k \geq 1$ is given precisely by our statement. This property ensures that our translation faithfully simulates the behavior of compensable processes. The proof is by induction on the derivation of $P \longrightarrow P'$ and uses:

- Proposition 2.2.3 (Page 18) for determining three base cases.
- Definition 3.2.3 (Page 39), i.e., the definition of translation.
- Lemma 3.2.9 (Page 47), which maps evaluation contexts in $\mathcal{C}_{\mathsf{D}}$ into evaluation contexts of $\mathcal{S}$.
- Lemma 3.2.14 (Page 51), which concerns function $\mathsf{ch}(\cdot,\cdot)$.

Part (2) of Theorem 3.2.7 is ***soundness***, i.e.,

$$\text{If } [\![P]\!]_\varepsilon \longrightarrow^n R \text{ then there is } P' \text{ such that } P \longrightarrow^* P' \text{ and } R \longrightarrow^* [\![P']\!]_\varepsilon$$

This property ensures that target terms never exhibit behavior that can not be attributed to some compensable process. As usual, proving soundness is more challenging than proving completeness.

The proof of soundness is by induction on $n$, i.e., the length of the reduction $[\![P]\!]_\rho \longrightarrow^n R$. We rely crucially on two lemmas (Lemma 3.2.19 and Lemma 3.2.20). Lemma 3.2.19 concerns the shape of processes $R$ and $P'$, whereas Lemma 3.2.20 ensures that the obtained adaptable process $R$ can evolve until reaching a process that corresponds to the translation of a compensable process. More in details:

- By analyzing the processes obtained by translating the composition of a transaction and its externally triggered failure signal (and its computation), we come to Lemma 3.2.16 (Page 51), which identifies processes that are created before a synchronization on $h_t$.
- Similarly, the analysis of the processes obtained by translating a transaction and its internally triggered failure signal (and its computation) leads us to Lemma 3.2.17 (Page 56), which identifies processes that are created before a synchronization on $h_u$.
- In the statement of Lemma 3.2.16 and Lemma 3.2.17 we use the definition of intermediate processes given by Definition 3.2.7 and Definition 3.2.8, respectively. The proofs proceed by case analysis for the step $R \longrightarrow R'$.
- Lemma 3.2.19 (Page 57) is about the shape of process $R$, and also ensures that there is a process $P'$ with an appropriate shape. The proof proceeds by induction on $n$. The base case uses Lemma 3.2.12 (Page 49); in the inductive step, we exploit the fact that the target term $R_1$ has a specific shape, which is in turn ensured by Lemma 3.2.16 and Lemma 3.2.17.
- Lemma 3.2.20 (Page 62) ensures that the adaptable process obtained thanks to Lemma 3.2.16 and Lemma 3.2.17 can evolve until reaching a process that corresponds to the translation of a compensable process.

Using these guidelines as a proof sketch, we now repeat Theorem 3.2.7 (Page 46) and present its proof in full detail:

**Theorem 3.2.7** (Operational Correspondence for $[\![\cdot]\!]_\rho$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathsf{D}}$.

(1) If $P \longrightarrow P'$ then $[\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 4 + \mathtt{pb_D}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_D}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 4 + \mathtt{pb_D}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

(2) If $[\![P]\!]_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* [\![P']\!]_\varepsilon$.

*Proof.* (1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases *a*), *b*) and *c*) of Proposition 2.2.3 (Page 18). In all cases, we use Definition 3.2.3, Lemma 3.2.9 (Page 47), and Lemma 3.2.21.

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have the following derivation:

$$\begin{aligned}
[\![P]\!]_\varepsilon \equiv\ & [\![E\big[C[\overline{a}.P_1] \mid D[a.P_2]\big]]\!]_\varepsilon \\
=\ & [\![E]\!]_\varepsilon\big[[\![C[\overline{a}.P_1] \mid D[a.P_2]]\!]_\rho\big] \\
=\ & [\![E]\!]_\varepsilon[[\![C]\!]_\rho[[\![\overline{a}.P_1]\!]_{\rho'}] \mid [\![D]\!]_\rho[[\![a.P_2]\!]_{\rho''}]] \\
=\ & [\![E]\!]_\varepsilon\big[[\![C]\!]_\rho[\overline{a}.[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_\rho[a.[\![P_2]\!]_{\rho''}]\big] \\
\longrightarrow\ & [\![E]\!]_\varepsilon\big[[\![C]\!]_\rho[[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_\rho[[\![P_2]\!]_{\rho''}]\big] \\
=\ & [\![E]\!]_\varepsilon\big[[\![C[P_1] \mid D[P_2]]\!]_\rho\big] \\
=\ & [\![E\big[C[P_1] \mid D[P_2]\big]]\!]_\varepsilon \\
\equiv\ & [\![P']\!]_\varepsilon
\end{aligned} \tag{3.30}$$

Therefore, the thesis holds with $k = 1$.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $m = \mathtt{pb_D}(P_1)$, and $P' \equiv E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation:

$$\begin{aligned}
[\![P]\!]_\varepsilon \equiv\ & [\![E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]]\!]_\varepsilon \\
=\ & [\![E]\!]_\varepsilon\Big[[\![C[t[P_1,Q]]]\!]_\rho \mid [\![D[\overline{t}.P_2]]\!]_\rho\Big] \\
=\ & [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho[[\![t[P_1,Q]]\!]_{\rho'}] \mid [\![D]\!]_\rho[[\![\overline{t}.P_2]\!]_{\rho''}]\Big] \\
=\ & [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho\Big[t\big[[\![P_1]\!]_{t,\rho'}\big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q]\!]_\varepsilon]\big)\Big] \\
& \mid [\![D]\!]_\rho[\overline{t}.h_t.[\![P_2]\!]_{\rho''}]\Big] \\
\longrightarrow\ & [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho\big[I_t^{(1)}([\![P_1]\!]_{t,\rho'}, [\![Q]\!]_\varepsilon)\big] \mid [\![D]\!]_\rho[h_t.[\![P_2]\!]_{\rho''}]\Big] \\
\longrightarrow^{m+2}\ & [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho\big[[I_t^{(m+3)}([\![P_1]\!]_{t,\rho'}, [\![Q]\!]_\varepsilon)\big] \mid [\![D]\!]_\rho\big[h_t.[\![P_2]\!]_{\rho''}\big]\Big] \\
\longrightarrow\ & [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho\big[[\![\mathsf{extr_D}(P_1) \mid \langle Q\rangle]\!]_{\rho'}\big] \mid [\![D]\!]_\rho\big[[\![P_2]\!]_{\rho''}\big]\Big] \\
=\ & [\![E]\!]_\varepsilon\Big[[\![C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle]]\!]_\rho \mid [\![D[P_2]]\!]_\rho\Big] \\
=\ & [\![E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]]\!]_\varepsilon \\
\equiv\ & [\![P']\!]_\varepsilon
\end{aligned}$$

Since we have that the error notification is external, in $\texttt{extrd}\langle\langle t, p_{t,\rho'}, p_{\rho'}\rangle\rangle$ (cf. eq. (3.3)) we get that $\mathsf{ch}(t, [\![P_1]\!]_\rho) = \mathbf{0}$. (cf. Lemma 3.2.14 for more details). The order/nature of these reduction steps is as follows:

i) The first synchronization concerns $t$ and $\bar{t}$.

ii) The following $m+2$ synchronizations can be explained as follows:

- First, we have a process relocation through the update of location $t$, as enforced by the definition of process $\texttt{extr}$. Process $I_t^{(1)}([\![P_1]\!]_{t,\rho'}, [\![Q]\!]_\varepsilon)$ is as in Definition 3.2.7 (Figure 3.6); as shown in Figure 3.5, there are two possibilities for reduction, depending on $m$.
- Subsequently, due to process

$$\texttt{outd}^{\mathsf{s}}(p_{t,\rho'}, p_{\rho'}, \texttt{nl}(p_{t,\rho'}, [\![P_1]\!]_{t,\rho'}), t\langle\langle\dagger\rangle\rangle.\overline{h_t})$$

we have $m$ reduction steps that relocate processes on location $p_{t,\rho'}$ to location $p_{\rho'}$, as also shown in Figure 3.5.
- The final reduction corresponds to the erasure of the location $t$ with all its contents, obtained by updating prefix $t\langle\langle\dagger\rangle\rangle$.

iii) Finally, we have a synchronization between $h_t$ and $\overline{h_t}$, which serves to signal that all synchronizations related to location $t$ have been completed.

Therefore, we can conclude that for $[\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$ where $k = 4+m$.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[u[F[\overline{u}.P_1], Q]]$, with $m = \mathsf{pb}_{\mathsf{D}}(F[P_1])$, and $P' \equiv C[\mathsf{extr}_{\mathsf{D}}(F[P_1]) \mid \langle Q\rangle]$. Then we have the following derivation:

$$\begin{aligned}
[\![P]\!]_\varepsilon &\equiv [\![C\big[u[F[\overline{u}.P_1], Q]\big]]\!]_\varepsilon \\
&= [\![C]\!]_\varepsilon\big[[\![u[F[\overline{u}.P_1], Q]]\!]_\rho\big] \\
&= [\![C]\!]_\varepsilon\big[u\big[[\![F[\overline{u}.P_1]]\!]_{u,\rho}\big] \mid u.\,(\texttt{extrd}\langle\langle u, p_{u,\rho}, p_\rho\rangle\rangle \mid p_\rho[[\![Q]\!]_\varepsilon]))]\big] \\
&= [\![C]\!]_\varepsilon\big[u\big[[\![F]\!]_{u,\rho}[\overline{u}.h_u.[\![P_1]\!]_{\rho'}]\big] \mid u.(\texttt{extrd}\langle\langle u, p_{u,\rho}, p_\rho\rangle\rangle \mid p_\rho[[\![Q]\!]_\varepsilon])\big] \\
&\longrightarrow [\![C]\!]_\varepsilon\big[O_u^{(1)}([\![F]\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon)\big] \\
\longrightarrow^{m+2}\ & [\![C]\!]_\varepsilon\big[O_u^{(m+3)}([\![F]\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon)\big] \\
&\longrightarrow [\![C]\!]_\varepsilon\big[O_u^{(m+4)}([\![F]\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon)\big] \\
&\equiv [\![C]\!]_\varepsilon\big[[\![\mathsf{extr}_{\mathsf{D}}(F[P_1])]\!]_\rho \mid p_\rho[[\![Q]\!]_\varepsilon]\big] \\
&= [\![C\big[\mathsf{extr}_{\mathsf{D}}(F[P_1]) \mid \langle Q\rangle\big]]\!]_\varepsilon \\
&\equiv [\![P']\!]_\varepsilon
\end{aligned}$$

Process $O_u^{(q)}([\![F]\!]_{u,\rho}[h_u.[\![P_1]\!]_{\rho'}], [\![Q]\!]_\varepsilon)$, where $q \in \{1, \ldots, m+4\}$, is as in Definition 3.2.8 (Figure 3.7). It should be noted that the location on name $u$ and its content will be erased before interaction on name $h_u$ and $\overline{h_u}$ (cf. Figure 3.7 for $q = (m+2)$ and $q = (m+3)$). Therefore, in this case, the role of function $\mathsf{ch}(u, \cdot)$ is central: indeed, $\mathsf{ch}(u, [\![F]\!]_{u,\rho}\big[h_u.[\![P_1]\!]_{\rho'}\big])$ provides the input $h_u$ which is necessary to achieve operational correspondence.

The order/nature/number of reduction steps can be explained as in Case b) above. We can then conclude that $[\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$ for $k = 4+m$.

(2) **Part (2) – Soundness:** Given $[\![P]\!]_\varepsilon \longrightarrow^n R$, by Lemma 3.2.19, process $R$ has the following form:

$$R \equiv \prod_{w=1}^{z} [\![E_w]\!]_\varepsilon\Big[\prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w}\big[\prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}}[I^{(p)}_{t_{i,k,w}}]$$

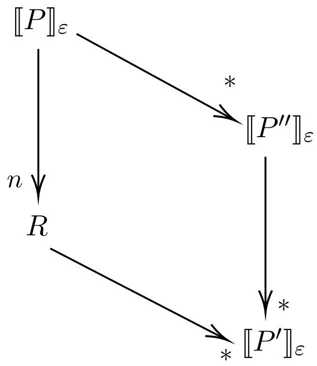


Figure 3.9: Diagram of the statement of soundness for $[\![\cdot]\!]_\varepsilon$

$$\mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}} \big[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\big] \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}} \big[O^{(q)}_{u_{c,k,w}}]]\big].$$

Also by Lemma 3.2.19, we have $P \longrightarrow^* P''$ where

$$P'' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}},Q_{t_{i,k,w}}]\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big]$$
$$\mid \prod_{c=1}^{m_k} L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}],Q_{u_{c,k,w}}]\big]\Big],$$

where by successive application of completeness it follows that $[\![P]\!]_\varepsilon \longrightarrow^* [\![P'']\!]_\varepsilon$.

By Lemma 3.2.20, i.e., by $l_k$ successive applications of (3.28) and $m_k$ successive applications of (3.29) on process $R$, it follows that:

$$R \longrightarrow^* \prod_{w=1}^{z} [\![E_w]\!]_\varepsilon\Big[\prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w}\big[\prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}}\big[[\![\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}})]\!]_{\rho''_{k,w}} \mid [\![\langle Q'_{t_{i,k,w}}\rangle]\!]_{\rho''_{k,w}}\big]$$
$$\mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}}\big[[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\big] \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}}\big[[\![\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}])]\!]_{\rho''_{k,w}}$$
$$\mid [\![\langle Q'_{u_{c,k,w}}\rangle]\!]_{\rho''_{k,w}}]]\Big]$$
$$\equiv [\![\prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[S_{t_{j,k,w}}\big]$$
$$\mid \prod_{c=1}^{m_k} L_{c,k,w}\big[\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle\big]\big]\Big]]\!]_\varepsilon \equiv [\![P']\!]_\varepsilon.$$

Therefore, it follows that

$$P' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[S_{t_{j,k,w}}\big]$$
$$\mid \prod_{c=1}^{m_k} L_{c,k,w}\big[\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle\big]\big]\Big].$$

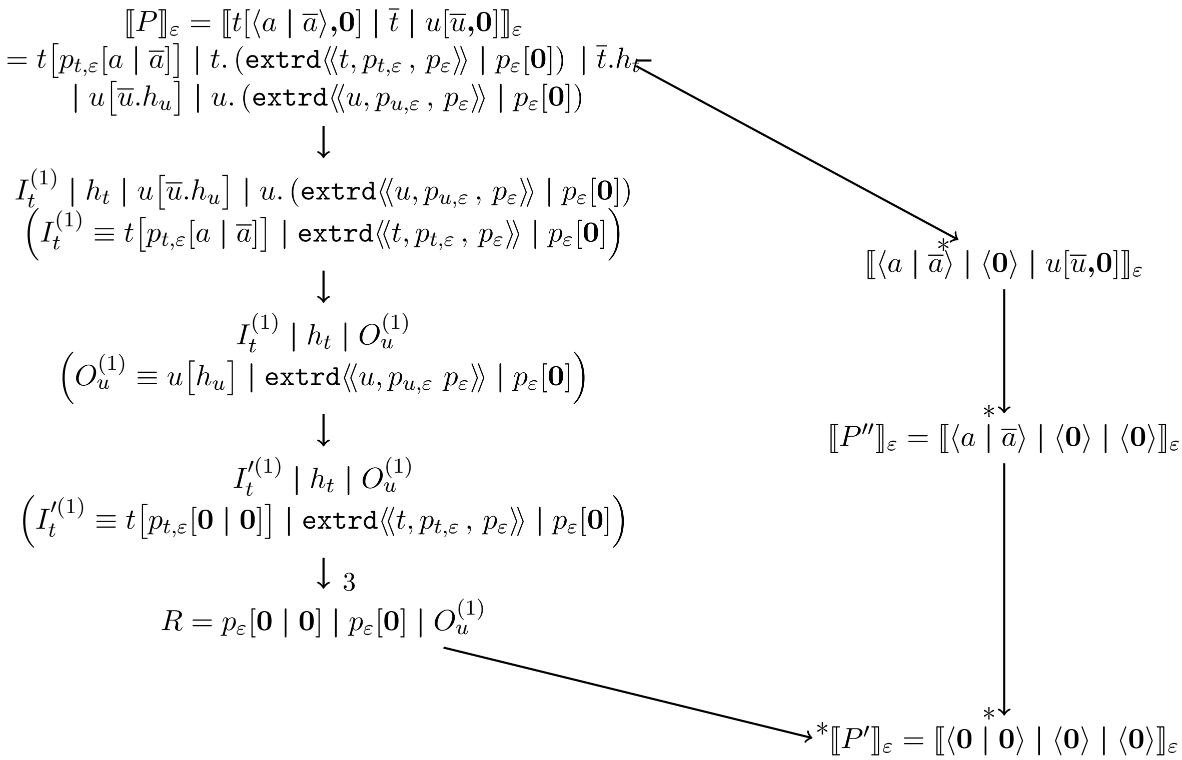



Figure 3.10: Example for operational soundness.

Also, by Proposition 2.2.3, i.e., by $l_k$ successive applications of case b) and and $m_k$ successive applications of case c) on process $P''$, it follows that: $P'' \longrightarrow^* P'$.
By successive application of **(1) – Completeness** on the derivation $P'' \longrightarrow^* P'$ it follows that $[\![P'']\!]_\varepsilon \longrightarrow^* [\![P']\!]_\varepsilon$. The proof scheme is shown in Figure 3.9.

■

We illustrate the encoding with the following examples.

**Example 3.2.22.** The example presented in Figure 3.10 illustrates the statement of soundness (Figure 3.9).

**Example 3.2.23.** Notably, $P = s\big[t[\langle a\rangle \mid \langle b\rangle \mid c,d],\mathbf{0}\big] \mid \overline{t}.\overline{s}$ is a well-formed compensable process. By the LTS of $\mathcal{C}_D$ (cf. Figure 2.3), we have:

$$P \xrightarrow{\tau}_D s[\langle a\rangle \mid \langle b\rangle \mid \langle d\rangle,\mathbf{0}] \mid \overline{s} \xrightarrow{\tau}_D \langle a\rangle \mid \langle b\rangle \mid \langle d\rangle.$$

The encoding of $P$ is obtained by expanding Definition 3.2.3:

$$\begin{aligned}
[\![P]\!]_\varepsilon &= s\Big[t\big[p_{t,s}[a] \mid p_{t,s}[b] \mid c\big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,s}, p_s\rangle\!\rangle \mid p_s[d]\big)\Big] \mid s.\mathtt{extrd}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle \mid \overline{t}.h_t.\overline{s}.h_s \\
&\longrightarrow^6 s\Big[p_s[a] \mid p_s[b] \mid p_s[d]\Big] \mid s.\mathtt{extrd}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle \mid \overline{s}.h_s \\
&\longrightarrow^7 p_\varepsilon[a] \mid p_\varepsilon[b] \mid p_\varepsilon[d] \\
&= [\![\langle a\rangle \mid \langle b\rangle \mid \langle d\rangle]\!]_\varepsilon.
\end{aligned}$$

Let us write $P_1$ to denote the process $\langle a\rangle \mid \langle b\rangle \mid c$ (the default activity of transaction $t$) and $P_2$ to denote the process $\langle a\rangle \mid \langle b\rangle \mid \langle d\rangle$ (the process obtained above). Our operational correspondence result ensures that $k$ in $[\![P]\!]_\varepsilon \longrightarrow^k [\![P_2]\!]_\varepsilon$ is equal to

$$k = \underbrace{4 + \mathtt{pb}_D(P_1)}_{\text{for transaction } t} + \underbrace{4 + \mathtt{pb}_D(P_2)}_{\text{for transaction } s}$$

$$= 6 + 7 = 13$$

Let us analyze in detail these reduction steps:

i) The first step corresponds to the synchronization between $t$ and $\overline{t}$.

ii) Once process $\mathtt{extrd}\langle\langle t, p_{t,s}, p_s\rangle\rangle$ is released, the second step is a synchronization on update prefix $t\langle\langle(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outd}^{\mathtt{s}}(p_{t,s}\,,\, p_s, \mathtt{nl}(p_{t,s}, Y), t\langle\langle\dagger\rangle\rangle.\overline{h_t})\rangle\rangle$ and location $t\big[p_{t,s}[a] \mid p_{t,s}[b] \mid c\big]$.

iii) Since we get process $\mathtt{outd}^{\mathtt{s}}(p_{t,s}\,,\, p_s, 2, t\langle\langle\dagger\rangle\rangle.\overline{h_t})$, the third and fourth steps correspond to synchronizations between locations $p_{t,s}[a]$ and $p_{t,s}[b]$ with the (nested) update prefix $p_{t,s}\langle\langle(X_1, X_2).p_s[X_1] \mid p_s[X_2] \mid t\langle\langle\dagger\rangle\rangle.\overline{h_t}\rangle\rangle$, which relocates the encoding of protected blocks.

iv) The fifth step is the synchronization between update prefix $t\langle\langle\dagger\rangle\rangle$ and location $t[\ldots]$, whereby the location is deleted together with its content (cf. equation (3.1));

v) The sixth reduction step is a synchronization on name $h_t$, which enables behavior corresponding to the encoding of transaction $s$.

To encode the failure of transaction $s$, we repeat the exact same steps as before. For location $s$ we have one more reduction step, because in process $\mathtt{outd}^{\mathtt{s}}(p_s\,,\, p_\varepsilon, 3, s\langle\langle\dagger\rangle\rangle.\overline{h_s})$ we have three locations $p_s[\ldots]$ that have to be relocated on location $p_\varepsilon[\ldots]$.

We illustrate the encoding also on the *Hotel booking scenario* discussed earlier (Example 2.2.1, Page 15).

**Example 3.2.24.** Recall the hotel booking scenario (§ 2.2, Example 2.2.1) where the client wants to cancel a reservation after booking and paying.

$$
\begin{aligned}
[\![Reservation]\!]_\varepsilon = \;& t\big[book.pay.invoice\big] \mid t.(\mathtt{extrd}\langle\langle t, p_t, p_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]) \mid \overline{book}.\overline{pay}.\overline{t}.h_t.refund \\
\longrightarrow \;& t\big[pay.invoice\big] \mid t.(\mathtt{extrd}\langle\langle t, p_t, p_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]) \mid \overline{pay}.\overline{t}.h_t.refund \\
\longrightarrow \;& t\big[invoice\big] \mid t.(\mathtt{extrd}\langle\langle t, p_t, p_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]) \mid \overline{t}.h_t.refund \\
\longrightarrow \;& t\big[invoice\big] \mid t\langle\langle(Y).t[Y] \mid \mathsf{ch}(t,Y) \\
& \mid \mathtt{outd}^{\mathtt{s}}(p_t\,,\, p_\varepsilon, \mathtt{nl}(p_t, Y), t\langle\langle\dagger\rangle\rangle.\overline{h_t})\rangle\rangle \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\longrightarrow \;& t\big[invoice\big] \mid \mathsf{ch}(t, invoice) \mid \mathtt{outd}^{\mathtt{s}}(p_t\,,\, p_\varepsilon, \mathtt{nl}(p_t, invoice), t\langle\langle\dagger\rangle\rangle.\overline{h_t}) \\
& \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\equiv \;& t\big[invoice\big] \mid \mathtt{outd}^{\mathtt{s}}(p_t\,,\, p_\varepsilon, 0, t\langle\langle\dagger\rangle\rangle.\overline{h_t}) \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
= \;& t\big[invoice\big] \mid t\langle\langle\dagger\rangle\rangle.\overline{h_t} \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\longrightarrow \;& \overline{h_t} \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\longrightarrow \;& p_\varepsilon[\overline{refund}] \mid refund \\
\longrightarrow \;& p_\varepsilon[\mathbf{0}]
\end{aligned}
$$

Therefore, we get $[\![Reservation]\!]_\varepsilon \longrightarrow^7 p_\varepsilon[\mathbf{0}]$. There are three reduction steps as a result of synchronizations on input prefixes: $book, pay$ and $t$ with corresponding outputs. Now, the structure of the default activity of transaction is changed and we have one reduction step for updating its current content. After that, there are three more reduction steps: one for erasing the location $t$ and its content, and two reduction steps as result of synchronizations on input names $h_t$ and $refund$ with corresponding outputs.

| | | |
|---|---|---|
| $\mathtt{npb}(\langle P\rangle) = 1 + \mathtt{npb}(P)$ | $\mathtt{npb}(P \mid Q) = \mathtt{npb}(P) + \mathtt{npb}(Q)$ | $\mathtt{npb}((\nu x)P) = \mathtt{npb}(P)$ |
| $\mathtt{npb}(!\pi.P) = \mathtt{npb}(\pi.P) = 0$ | $\mathtt{npb}(t[P,Q]) = 1 + \mathtt{npb}(P) + \mathtt{npb}(Q)$ | $\mathtt{npb}(\mathbf{0}) = 0$ |

Figure 3.11: Number of protected blocks.

#### 3.2.3.2.6 Divergence Reflection

In the following, we are going to prove that the encoding does not introduce divergent computations. We need the following definition, which counts all protected blocks in process $P$.

**Definition 3.2.9.** Given a well-formed compensable process $P$, we will write $\mathtt{npb}(P)$ to denote the number of protected blocks in $P$ — see Figure 3.11 for a definition.

Notice that $\mathtt{npb}(P)$ is different from $\mathtt{pb}(P)$ in Definition 3.2.5. The difference is in the definition for processes $\langle P\rangle$ and $t[P,Q]$. In Definition 3.2.5 we count all processes that may become protected, e.g., after a reduction of the considered compensable process. Intuitively, with $\mathtt{npb}(P)$ we are looking for protected blocks at all levels of the observed compensable process.

To establish divergence reflection, we relate a sequence of adaptable processes and a sequence of compensable processes. One reduction of an adaptable process from the sequence corresponds either to one reduction of the corresponding compensable process or to equal subsequent compensable processes. This reflects that a single reduction of a compensable processes is mimicked by several reductions of a corresponding adaptable process. The following lemma proves that such a relation does exist, providing also the upper bound for the number of successive, non-equivalent, adaptable processes with the property that their corresponding adaptable processes are equal. This last property directly induces that the set of compensable processes is infinite, too.

**Lemma 3.2.25.** Let $\{R_i\}_{i\geq 0}$ be a sequence of adaptable processes such that $R_i \longrightarrow R_{i+1}$, with $R_0 = [\![P_0]\!]_\rho$, for some compensable process $P_0$ and path $\rho$. Then for every $i \geq 1$ there is $P_i$ such that:

(i) $R_i \longrightarrow^* [\![P_i]\!]_\rho$,

(ii) $P_{i-1} = P_i$ or $P_{i-1} \longrightarrow P_i$, and

(iii) $R_i \not\equiv R_{i+1} \not\equiv \ldots \not\equiv R_{i+m}$ and $P_i = P_{i+1} = \ldots = P_{i+m}$ imply $m \leq 4 + \mathtt{npb}(P_0)$.

*Proof.* The proof for (i) and (ii) proceeds by induction on $i$.
***Base case:*** Assume that $i = 1$. By the proof of Lemma 3.2.19, i.e. its ***Base case***, we have three cases:

a) $P_0 \equiv E[C[a.P_1'] \mid D[\overline{a}.P_2]]$ and $R_1 \equiv [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho[[\![P_1']\!]_{\rho'}] \mid [\![D]\!]_\rho[[\![P_2]\!]_{\rho''}]\Big] = [\![P_1]\!]_\rho$, it follows $P_0 \longrightarrow P_1$ (cf. Proposition 2.2.3 (a)).

b) $P_0 \equiv E\Big[C[t[P_2,Q]] \mid D[\overline{t}.P_1']\Big]$ and
$R_1 \equiv [\![E]\!]_\varepsilon\Big[[\![C]\!]_\rho[t\big[[\![P_2']\!]_{t,\rho'}\big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]] \mid [\![D]\!]_\rho[h_t.[\![P_1']\!]_{\rho''}]\Big]$. There is $P_1$ such that by Lemma 3.2.20 (3.28) it follows $R_1 \longrightarrow^* [\![P_1]\!]_\rho$. Also, it follows $P_0 \longrightarrow P_1$ (cf. Proposition 2.2.3 (b)).

c) $P_0 \equiv C[u[D[\overline{u}.P_1'],Q]]$ and $R_1 \equiv [\![C]\!]_\varepsilon\Big[u\big[[\![D]\!]_{u,\rho}[h_u.[\![P_1']\!]_{\rho'}]\big] \mid \mathtt{extrd}\langle\!\langle u, p_{u,\rho'}, p_{\rho'}\rangle\!\rangle \mid p_{\rho'}[[\![Q']\!]_\varepsilon]\Big]$. There is $P_1$ such that by Lemma 3.2.20 (3.29) it follows $R_1 \longrightarrow^* [\![P_1]\!]_\rho$. Also, it follows $P_0 \longrightarrow P_1$ (cf. Proposition 2.2.3 (c)).

***Inductive step:*** By inductive hypothesis, there are processes $P_1, \ldots, P_{i-1}, P_i$ such that $R_{i-1} \longrightarrow^* [\![P_{i-1}]\!]_\rho$ and either $P_{i-1} = P_i$ or $P_{i-1} \longrightarrow P_i$. Let us now consider $R_i \longrightarrow R_{i+1}$ (i.e., $[\![P_0]\!]_\rho \longrightarrow^i R_i \longrightarrow R_{i+1}$). By the proof of Lemma 3.2.19, i.e., its ***Inductive step***, we get that there is $P_{i+1}$ such that either $P_i = P_{i+1}$ or $P_i \longrightarrow P_{i+1}$ (cf. for example (3.20) and (3.22)). By Lemma 3.2.20 it follows $R_{i+1} \longrightarrow^* [\![P_{i+1}]\!]_\rho$.
Now, we are going to prove the last assertion in the statement, (iii). In the following, we give guidelines on how to obtain the proof since it follows from (the proof of) Lemma 3.2.19:

(1) The form of process $R_i$ is given with (3.18), and process $P_i$ has a form given with (3.19).

(2) In the proof, its ***Inductive step***, we consider only cases such that $R_i \not\equiv R_{i+1}$ and $P_i = P_{i+1}$. Therefore, we consider the cases in which intermediate processes $I^{(p)}_{t_{i,k,w}}$ and $O^{(q)}_{u_{c,k,w}}$ inside process $R_i$ (cf. Definition 3.2.7 and Definition 3.2.8, respectively) have been changed.

(3) From Figure 3.6 and Figure 3.7 we obtain the form and the number of all intermediate processes. We remind the reader that the number of intermediate processes directly depends on the number of protected blocks in the observed transaction, more precisely in its compensation activity (cf. for example Figure 3.5).

(4) We conclude, for each $l \in \{1, \ldots, m\}$ it follows that $m$ is at most $4 + \mathtt{npb}(P_0)$, i.e., $m = 4 + \mathtt{pb}(Q') \leq 4 + \mathtt{npb}(P_0)$, for some $Q'$ that appears in $P_0$.

■

The following theorem concerns *infinite* reduction sequences: it says that an infinite reduction sequence originating from a target term can only arise from an infinite reduction sequence of a corresponding source term. Hence, it suffices to establish divergence reflection, as in Definition 2.3.5:

**Theorem 3.2.26** (Divergence Reflection for $[\![\cdot]\!]_\rho$)**.** Let $\{R_i\}_{i\geq 0}$ be an infinite sequence of adaptable processes such that

$$(1)\ R_0 = [\![P_0]\!]_\rho \text{ for some } P_0 \text{ and } \rho, \qquad \text{and} \qquad (2)\ \ R_i \longrightarrow R_{i+1} \text{ for any } i \geq 0.$$

Then there is an infinite sequence of adaptable processes $\{P'_j\}_{j\geq 0}$ such that

$$(3)\ P'_0 = P_0, \qquad \text{and} \qquad (4)\ \ P'_j \longrightarrow P'_{j+1} \text{ for any } j \geq 0.$$

*Proof.* By Lemma 3.2.25, there is a sequence $\{P_i\}_{i\geq 0}$ such that

$$\text{(i) } R_i \longrightarrow^* [\![P_i]\!]_\rho \qquad \text{and} \qquad \text{(ii) } \ P_{i-1} = P_i \text{ or } P_{i-1} \longrightarrow P_i.$$

Consider now a sequence of compensable processes $P'_0, P'_1, P'_2, \ldots$ such that

(1) $P'_{j-1} \longrightarrow P'_j$, for any $j \geq 1$, and

(2) for every $i$ there is $j$ such that $P_i = P'_j$.

By Lemma 3.2.25, at most $4 + \mathtt{npb}(P_0)$ reduction steps from the sequence $\{R_i\}_{i\geq 0}$ correspond to one reduction step of $\{P'_j\}_{j\geq 0}$. Hence, the number of processes in $\{P'_j\}_{j\geq 0}$ is not less then the number of processes $\{R_i\}_{i\geq 0}$ divided by $4 + \mathtt{npb}(P_0)$. Since the sequence $\{R_i\}_{i\geq 0}$ is infinite, the same holds for $\{P'_j\}_{j\geq 0}$. ■

#### 3.2.3.2.7 Success Sensitiveness

To prove that the translation satisfies success sensitiveness we need first to extend Definition 3.2.3 with $[\![\checkmark]\!]_\rho = \checkmark$.

Further, we adapt the definition of may-succeed (Definition 2.3.4) to adaptable and compensable processes. It is defined in exactly the same way for the two calculi, but it relies on different definitions of operational semantics and evaluation contexts.

**Definition 3.2.10.** Let $P$ be an adaptable/compensable process. We say that $P$ may-succeed, denoted $P\Downarrow$, if $P \longrightarrow^* P'$ and $P' = C[\checkmark]$ for some process $P'$ and evaluation context $C[\bullet]$.

**Theorem 3.2.27** (Success Sensitiveness for $[\![\cdot]\!]_\rho$)**.** Let $P$ be a well-formed compensable process and $\rho$ an arbitrary path. Then $P\Downarrow$ if and only if $[\![P]\!]_\rho \Downarrow$.

We shall prove that success sensitiveness holds for the translation $[\![\cdot]\!]_\rho$. The first part of the statement

$$P \Downarrow \text{ implies } [\![P]\!]_\rho \Downarrow$$

follows directly from operational completeness (Theorem 3.2.7 (1)) and Lemma 3.2.9. The proof for the opposite direction

$$[\![P]\!]_\rho \Downarrow \text{ implies } P \Downarrow$$

is derived through the following steps:

- By Definition 3.2.10, if $[\![P]\!]_\rho \Downarrow$ then $[\![P]\!]_\rho \longrightarrow^k R$ and $R = C[\checkmark]$ for some context $C[\bullet]$.
- By Lemma 3.2.19, we conclude that process $R$ has the form given in (3.18).
- Assuming that $R = C[\checkmark]$, we identify all possible positions of $\checkmark$ in the form (3.18). For that purpose, we introduce some auxiliary lemmas:
  - By Lemma 3.2.15, either $\checkmark$ appears at top level of some context (in parallel), in a form $[\![C'[\checkmark]]\!]_\rho$, or it is nested inside some locations. There are four additional nested places that we consider separately and list in the following items.
  - Lemma 3.2.28 considers the case with $I_t^{(p)}([\![P_1]\!]_{t,\rho}, [\![Q_1]\!]_\varepsilon) = C''[\checkmark]$ and $\mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}) = 0$ and $p \in \{1,2,3\}$, where $I_t^{(p)}([\![P_1]\!]_{t,\rho}, [\![Q_1]\!]_\varepsilon)$ is given in Figure 3.6.
  - Lemma 3.2.29 considers the case with $I_t^{(p)}([\![P_1]\!]_{t,\rho}, [\![Q_1]\!]_\varepsilon) = C''[\checkmark]$ and $\mathtt{nl}(p_{t,\rho}, [\![P_1]\!]_{t,\rho}) = m > 0$ and $p \in \{1,2,\ldots,m+3\}$, where $I_t^{(p)}([\![P_1]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$ is given in Figure 3.6.
  - Lemma 3.2.30 considers the case with $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P_1]\!]_{\rho'}], [\![Q_1]\!]_\varepsilon) = C''[\checkmark]$ and $\mathtt{nl}(p_{u,\rho}, [\![F]\!]_\rho[h_u.[\![P_1]\!]_{\rho'}]) = 0$ and $p \in \{1,2,3,4\}$, where $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P_2]\!]_{\rho'}], [\![Q_1]\!]_\varepsilon)$ is given in Figure 3.7.
  - Lemma 3.2.31 considers the case with $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P_1]\!]_{\rho'}], [\![Q_1]\!]_\varepsilon) = C''[\checkmark]$ and $\mathtt{nl}(p_{u,\rho}, [\![F]\!]_\rho[h_u.[\![P_1]\!]_{\rho'}]) = m > 0$ and $p \in \{1,\ldots,m+4\}$, where $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P_2]\!]_{\rho'}], [\![Q_1]\!]_\varepsilon)$ is given in Figure 3.7.
- Finally, after identifiying the place of $\checkmark$, using (3.19) of Lemma 3.2.19, we get the proof.

We proceed to introduce the auxiliary lemmas that consider nested appearances of $\checkmark$.

**Lemma 3.2.28.** Let $t$ be a name, $\rho$ a path, and $P, Q$ well-formed compensable processes such that $\mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho}) = 0$. If $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon) = C[\checkmark]$, for $p \in \{1,2,3\}$ and some context $C[\bullet]$, then

(i) either $[\![P]\!]_{t,\rho} = C_1[\checkmark]$, (ii) or $[\![Q]\!]_\varepsilon = C_1[\checkmark]$

for some context $C_1[\bullet]$.

*Proof.* There are three possible forms of $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon)$, given in the first three rows of Figure 3.6.

- $p \in \{1,2\}$: If $t\big[[\![P]\!]_{t,\rho}\big] \mid R \mid p_\rho[[\![Q]\!]_\varepsilon] = C[\checkmark]$ and $(R \equiv t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid t\{\dagger\}.\overline{h_t}\}$ or $R = t\{\dagger\}.\overline{h_t})$, by Definition 2.2.6, we have the following two possibilities:
  - (i) $C[\bullet] = t\big[C_1[\bullet]\big] \mid R \mid p_\rho[[\![Q]\!]_\varepsilon]$ and $C_1[\checkmark] = [\![P]\!]_{t,\rho}$, or
  - (ii) $C[\bullet] = t\big[[\![P]\!]_{t,\rho}\big] \mid R \mid p_\rho[C_1[\bullet]]$ and $C_1[\checkmark] = [\![Q]\!]_\varepsilon$.
- $p = 3$: If $\overline{h_t} \mid p_\rho[[\![Q]\!]_\varepsilon] = C[\checkmark]$, by Definition 2.2.6, $C[\bullet] = \overline{h_t} \mid p_\rho[C_1[\bullet]]$ and therefore $C_1[\checkmark] = [\![Q]\!]_\varepsilon$.

■

**Lemma 3.2.29.** Let $t$ be a name, $\rho$ a path, and $P, Q$ well-formed compensable processes such that $[\![P]\!]_{t,\rho} = \prod_{k=1}^{m} p_{t,\rho}[[\![P'_k]\!]_\varepsilon] \mid S$ with $\mathtt{nl}(p_{t,\rho}, [\![P]\!]_{t,\rho}) = m$. If $I_t^{(p)}([\![P]\!]_{t,\rho}, [\![Q]\!]_\varepsilon) = C[\checkmark]$, for $p \in \{1, \dots, m+3\}$ and some context $C[\bullet]$, then

(i) $[\![P'_k]\!]_\varepsilon = C_1[\checkmark]$, or (ii) $[\![Q]\!]_\varepsilon = C_1[\checkmark]$, or (iii) $S = C_1[\checkmark]$

for some context $C_1[\bullet]$ and $k \in \{1, \dots, m\}$.

*Proof.* The proof is similar to the proof of Lemma 3.2.28 and follows directly from Definition 2.2.6 and the process defined in Figure 3.6. ■

**Lemma 3.2.30.** Let $u$ be a name, $\rho$ a path, and $P, Q$ well-formed compensable processes such that $\mathtt{nl}(p_{u,\rho}, [\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}]) = 0$. If $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}], [\![Q]\!]_\varepsilon) = C[\checkmark]$, for $p \in \{1,2,3,4\}$ and some context $C[\bullet]$, then

(i) either $[\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}] = C_1[\checkmark]$, (ii) or $[\![Q]\!]_\varepsilon = C_1[\checkmark]$

for some context $C_1[\bullet]$.

*Proof.* The proof is similar to the proof of Lemma 3.2.28 and follows directly from Definition 2.2.6, and the process defined in Figure 3.7. ■

**Lemma 3.2.31.** Let $u$ be a name, $\rho$ a path, and $P, Q$ well-formed compensable processes such that $[\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}] = \prod_{k=1}^{m} p_{u,\rho}[[\![P'_k]\!]_\varepsilon] \mid S$ with $\mathtt{nl}(p_{u,\rho}, [\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}]) = m$. If $O_u^{(q)}([\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}], [\![Q]\!]_\varepsilon) = C[\checkmark]$, for $p \in \{1, \dots, m+4\}$ and some context $C[\bullet]$, then

(i) $[\![F]\!]_\rho[h_u.[\![P]\!]_{\rho'}] = C_1[\checkmark]$, or (ii) $[\![Q]\!]_\varepsilon = C_1[\checkmark]$, or (iii) $[\![P'_k]\!]_\varepsilon = C_1[\checkmark]$

for some context $C_1[\bullet]$ and $k \in \{1, \dots, m+4\}$.

*Proof.* Similar to the proof of Lemma 3.2.28 and follows directly from Definition 2.2.6 and the process defined in Figure 3.7. ■

Now we will give the proof of the success sensitiveness (cf. Theorem 3.2.27):

*Proof of Success Sensitiveness – Theorem 3.2.27.* (⇒) Let $P \Downarrow$, i.e., $P \longrightarrow^* P'$ and $P' = C[\checkmark]$. By Theorem 3.2.7 (1) – Completeness we have that $[\![P]\!]_\rho \longrightarrow^k [\![P']\!]_\rho = [\![C[\checkmark]]\!]_\rho$. By Convention 3.2.8 and Lemma 3.2.9 it follows:

$$[\![C]\!]_\rho[\checkmark] = [\![C[\bullet]]\!]_\rho[[\![\checkmark]\!]_{\rho'}],$$

where $\rho'$ is a path to hole in context $C[\bullet]$. By $[\![\checkmark]\!]_\rho = \checkmark$ we have that $[\![P]\!]_\rho \longrightarrow^k [\![C[\bullet]]\!]_\rho[\checkmark]$. This implies that $[\![P]\!]_\rho \Downarrow$.
($\Leftarrow$) Conversely, Let $[\![P]\!]_\rho \Downarrow$, i.e., $[\![P]\!]_\rho \longrightarrow^k R$ and $R \equiv C[\checkmark]$. By Lemma 3.2.19 it follows:

$$C[\checkmark] \equiv \prod_{w=1}^{z} [\![E_w]\!]_\varepsilon \Big[ \prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w} \big[ \prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}} [h_{t_{j,k,w}} . [\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}] \\ \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}} [O^{(q)}_{u_{c,k,w}}] \big] \Big] \tag{3.31}$$

By Lemma 3.2.15, Lemma 3.2.28, Lemma 3.2.29, Lemma 3.2.30, and Lemma 3.2.31 we analyze all possible places where $\checkmark$ occurs in (3.31). By Lemma 3.2.15,

(1) either

$$\begin{aligned} C[\checkmark] \equiv\ & [\![E'']\!]_\varepsilon[\checkmark] \mid \prod_{w=1}^{z} [\![E'_w]\!]_\varepsilon \Big[ \prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w} \big[ \prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}] \\ & \mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}} [h_{t_{j,k,w}} . [\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}] \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}} [O^{(q)}_{u_{c,k,w}}] \big] \end{aligned} \tag{3.32}$$

(2) or, there are $\omega \in \{1, \ldots, z\}$ and $C_1[\bullet]$ such that

$$\begin{aligned} C_1[\checkmark] \equiv \prod_{k=1}^{s_w} [\![G_{k,w}]\!]_{\rho_w} \big[ \prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}} [h_{t_{j,k,w}} . [\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}] \\ \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}} [O^{(q)}_{u_{c,k,w}}] \big] \Big] \end{aligned} \tag{3.33}$$

By Lemma 3.2.15,

(2.1) either

$$\begin{aligned} C_1[\checkmark] \equiv\ & [\![G''_w]\!]_{\rho_w}[\checkmark] \mid \prod_{k=1}^{s_w} [\![G'_{k,w}]\!]_{\rho_w} \big[ \prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}] \\ & \mid \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]_{\rho'_{k,w}} [h_{t_{j,k,w}} . [\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}] \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]_{\rho'_{k,w}} [O^{(q)}_{u_{c,k,w}}] \big] \Big] \end{aligned} \tag{3.34}$$

(2.2) or, there are $C_2[\bullet]$ and $k \in \{1, \ldots, s_\omega\}$ such that one of the following three cases holds:

(2.2.1) $C_2[\checkmark] \equiv [\![C_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}]$ :

(2.2.1.1) either $C_2[\checkmark] \equiv [\![C''_{i,k,w}]\!]_{\rho'_{k,w}}[\checkmark] \mid \prod_{i=1}^{l_k} [\![C'_{i,k,w}]\!]_{\rho'_{k,w}} [I^{(p)}_{t_{i,k,w}}]$

(2.2.1.2) or, there are $C_3[\bullet]$ and $i \in \{1, \ldots, l_k\}$ such that
$C_3[\checkmark] = I^{(p)}_{t_{i,k,w}}([\![P'_{t_{i,k,w}}]\!]_{t,\rho''}, [\![Q'_{t_{i,k,w}}]\!]_\varepsilon)$.
Assume that $\mathtt{nl}(p_{t,\rho''}, [\![P'_{t_{i,k,w}}]\!]_{t,\rho''}) = 0$ and $p \in \{1, 2, 3\}$ (other cases are similar). By Lemma 3.2.29,
(2.2.1.2.1) $[\![P'_{t_{i,k,w}}]\!]_{t,\rho''} = C_4[\checkmark]$, or
(2.2.1.2.2) $[\![Q]\!]_\varepsilon = C_4[\checkmark]$
for some $C_4[\bullet]$.

(2.2.2) $C_2[\checkmark] \equiv [\![D_{j,k,w}]\!]_{\rho'_{k,w}} \left[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\right]$: By Lemma 3.2.15,
$[\![D_{j,k,w}]\!]_{\rho'_{k,w}} \left[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\right] = [\![D'_{j,k,w}]\!]_{\rho'_{k,w}} [\checkmark] \mid [\![D_{j,k,w}]\!]_{\rho'_{k,w}} \left[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}\right]$.

(2.2.3) $C_2[\checkmark] \equiv [\![L_{c,k,w}]\!]_{\rho'_{k,w}} \left[O^{(q)}_{u_{c,k,w}}\right]$:

(2.2.3.1) either $C_2[\checkmark] \equiv [\![L'_{c,k,w}]\!]_{\rho'_{k,w}} [\checkmark] \mid [\![L''_{c,k,w}]\!]_{\rho'_{k,w}} \left[O^{(q)}_{u_{c,k,w}}\right]$

(2.2.3.2) or, there are $C_4[\bullet]$ and $c \in \{1, \dots, m_k\}$ such that
$C_4[\checkmark] = O^{(q)}_{u_{c,k,w}}([\![F_{c,k,w}]\!]_{\rho''}[h_{u_{c,k,w}}.[\![P_{u_{c,k,w}}]\!]_{\rho'''}], [\![Q'_{u_{c,k,w}}]\!]_{\varepsilon})$.
Assume that $\mathtt{nl}(u_{c,k,w}, \rho''', [\![F_{c,k,w}]\!]_{\rho''}[h_{u_{c,k,w}}.[\![P_{u_{c,k,w}}]\!]_{\rho'''}])) = 0$ and $q \in \{1,2,3,4\}$ (other cases are similar). By Lemma 3.2.30,
(2.2.3.2.1) either $[\![F_{c,k,w}]\!]_{\rho''}[h_{u_{c,k,w}}.[\![P_{u_{c,k,w}}]\!]_{\rho'''}] = C_5[\checkmark]$
(2.2.3.2.2) or, $[\![Q'_{u_{c,k,w}}]\!]_{\varepsilon}) = C_5[\checkmark]$
for some $C_5[\bullet]$.

In all the cases listed above, it follows directly by Lemma 3.2.19 that $P \Downarrow$ since

$$P \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}}, Q_{t_{i,k,w}}]\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big] \mid \prod_{c=1}^{m_k} L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}], Q_{u_{c,k,w}}]\big]\Big], \tag{3.35}$$

Other cases are similar. ■

## 3.3 Translating $\mathcal{C}_\mathtt{P}$ into $\mathcal{S}$

In this section we concentrate on a specific source calculus, namely the calculus in [29] with *static recovery* and *preserving semantics*. Before giving a formal presentation of the encoding $\mathcal{C}_\mathtt{P}$ into $\mathcal{S}$ we introduce some useful conventions and intuitions.

### 3.3.1 The Translation, Informally

The translation $\mathcal{C}_\mathtt{P}$ into $\mathcal{S}$, denoted $(\!|\cdot|\!)_\rho$, uses very similar ideas as the encoding $[\![\cdot]\!]_\rho$. This way, the translation of a protected block found at path $\rho$, is defined as:

$$(\!|\langle P\rangle|\!)_\rho = p_\rho\big[(\!|P|\!)_\varepsilon\big].$$

To encode a preserving semantics we use the base sets given in Definition 3.1.1. We use the set of *reserved location names* with name $\beta_\rho$, because besides protected blocks we have to keep transactions that are in default activity $P$ (cf. Figure 2.2) in the case that a failure signal is fired. We use a revised auxiliary process, denoted $\mathtt{outp}^\mathtt{s}(t, P, l_1, l'_1, l_2, l'_2, n, m)$, which:

(i) moves $n$ processes from location $l_1$ to location $l'_1$;

(ii) moves $m$ processes from location $l_2$ to location $l'_2$.

To define process $\mathtt{outp}^\mathtt{s}$, we need some auxiliary notions. In the case when we move $m$ processes from location $l_2$ to location $l'_2$ it will be necessary to remove some names from the path in processes that are enclosed in $l'_2$. The following function removes a name from a path:

**Definition 3.3.1.** Let $\rho = t_1, \dots, t_n$ be a path and $r$ be a name in $\mathcal{N}_t$. We define the function $\rho/r$ as follows:

$$\rho/r = \begin{cases} t_1, t_2, \dots, t_{i-1}, t_{i+1}, \dots, t_n & \text{if } t_i = r \\ \rho & \text{if } t_i \neq r \text{ for } 1 \leq i \leq n. \end{cases}$$

It should be noted that name $r$ can occur only one time in $\rho$ (cf. Definition 2.2.4 (i)).

The following definition serves to remove names mentioned in an adaptable process. This is important: if a transaction $t$ had nested transactions and location name $t$ is lost, then we have to remove $t$ from all the paths that contained it.

**Definition 3.3.2.** Let $P$ be a closed adaptable process, and let $\rho$ be a path that contains name $s$. The function $\mathcal{E}(P, s)$ is defined as follows:

$$
\begin{aligned}
&\mathcal{E}(l[P], s) = l[\mathcal{E}(P, s)] \text{ if } l \notin \mathcal{N}_l^r && \mathcal{E}(p_\rho[P], s) = p_{\rho/s}[\mathcal{E}(P, s)] \\
&\mathcal{E}(\beta_\rho[P], s) = \beta_{\rho/s}[\mathcal{E}(P, s)] && \mathcal{E}(P \mid Q, s) = \mathcal{E}(P, s) \mid \mathcal{E}(Q, s) \\
&\mathcal{E}(\pi.P, s) = \pi.\mathcal{E}(P, s) && \mathcal{E}(!\pi.P, s) = !\pi.\mathcal{E}(P, s) \\
&\mathcal{E}((\nu x)P, s) = (\nu x)\mathcal{E}(P, s) && \mathcal{E}(\mathbf{0}, s) = \mathbf{0} \\
&\mathcal{E}(X, s) = X
\end{aligned}
$$

The auxiliary processes $\mathtt{outp}^{\mathsf{s}}$ should depend on all location names which derive from the names of nested transactions (e.g., $t_1, \ldots, t_m$). The following definition has just that role, to denote the list of location names from adaptable process $P$ that are nested (at top level) in $l$.

**Definition 3.3.3.** Let $l$ be a name and $P$ an adaptable process. Function $\mathsf{top}(l, P)$ denotes the list of location names from $P$ that are nested (at top level) in $l$. It is defined as follows:

$$
\mathsf{top}(l, l'[P]) = \begin{cases} \{l''\} & \text{if } l = l' \text{ and } P = l''[Q] \mid R \text{ for some } Q \text{ and } R \text{ and } l'' \in \mathcal{N}_t \\ \emptyset & \text{otherwise} \end{cases}
$$

$$
\begin{aligned}
&\mathsf{top}(l, P \mid Q) = \mathsf{top}(l, P) \cup \mathsf{top}(l, Q) && \mathsf{top}(l, \mathbf{0}) = \mathsf{top}(l, X) = \emptyset \\
&\mathsf{top}(l, (\nu x)P) = \mathsf{top}(l, P) && \mathsf{top}(l, \pi.P) = \mathsf{top}(l, !\pi.P) = \emptyset
\end{aligned}
$$

The following example illustrates this definition.

**Example 3.3.1.** Given $P = \beta_{t,\rho}[t_1[m_1] \mid p_{t,\rho}[m_2]] \mid \beta_{t,\rho}[m_3 \mid t_2[m_4] \mid t_3[m_5]]$, by Definition 3.3.3 we have the following list of location names from $P$:

$$
\mathsf{top}(\beta_{t,\rho}, P) = \{t_1, t_2, t_3\}.
$$

We assume that function $\mathcal{E}(\cdot, \cdot)$ operates only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that it is applied once it is provided with an argument.

For the definition of $\mathtt{outp}^{\mathsf{s}}(t, P, l_1, l_1', l_2, l_2', n, m)$ we introduce the following auxiliary processes:

$$
\begin{aligned}
&\mathtt{outp}^{\mathsf{s}}_1(t, l_1, l_1', n) = l_1\langle\!\langle (X_1, \ldots, X_n). \left( \prod_{i=1}^{n} l_1'[X_i] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t \right) \rangle\!\rangle; \\
&\mathtt{outp}^{\mathsf{s}}_2(t, t_1, \ldots, t_m, l_2, l_2', m) = l_2\langle\!\langle (Y_1, \ldots, Y_m). \\
&\qquad \left( r_t. \left( \prod_{k=1}^{m} (l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}) \right) \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t} \right) \rangle\!\rangle; \\
&\mathtt{outp}^{\mathsf{s}}_3(t, t_1, \ldots, t_m, l_1, l_1', l_2, l_2', n, m) = l_1\langle\!\langle (X_1, \ldots, X_n).l_2\langle\!\langle (Y_1, \ldots, Y_m). \\
&\qquad \left( \prod_{i=1}^{n} l_1'[X_i] \mid r_t. \left( \prod_{k=1}^{m} (l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}) \right) \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t} \right) \rangle\!\rangle \rangle\!\rangle.
\end{aligned}
$$

We are now ready to define process $\mathtt{outp}^{\mathsf{s}}(t, P, l_1, l_1', l_2, l_2', n, m)$.

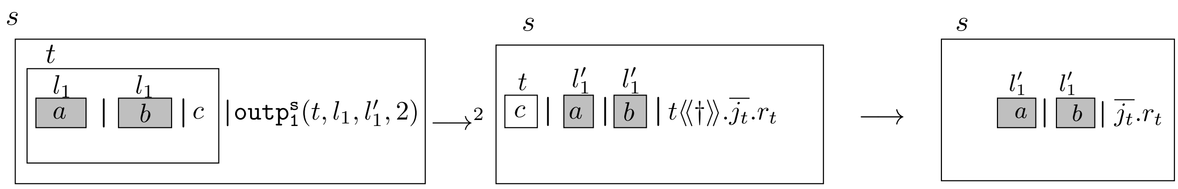



Figure 3.12: Illustrating $\mathtt{outp}^{\mathtt{s}}_{1}(t, l_1, l'_1, 2)$ from Example 3.3.2 .

The auxiliary process $\mathtt{outp}^{\mathtt{s}}(t, P, l_1, l'_1, l_2, l'_2, n, m)$, where $\mathtt{top}(l_2, P) = \{t_1, \ldots, t_m\}$ for $m > 0$, is defined as follows:

$$\mathtt{outp}^{\mathtt{s}}(t, P, l_1, l'_1, l_2, l'_2, n, m) = \begin{cases} t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t & \text{if } n, m = 0 \\ \mathtt{outp}^{\mathtt{s}}_{1}(t, l_1, l'_1, n) & \text{if } n > 0, m = 0 \\ \mathtt{outp}^{\mathtt{s}}_{2}(t, t_1, \ldots, t_m, l_2, l'_2, m) & \text{if } n = 0, m > 0 \\ \mathtt{outp}^{\mathtt{s}}_{3}(t, t_1, \ldots, t_m, l_1, l'_1, l_2, l'_2, n, m) & \text{if } n, m > 0 \end{cases} \tag{3.36}$$

The following example illustrates process $\mathtt{outp}^{\mathtt{s}}$ (cf. (3.36)). A more detailed explanation is given later on.

**Example 3.3.2.** We illustrate the revised auxiliary process:

$$\begin{aligned} & s\Big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid \mathtt{outp}^{\mathtt{s}}_{1}(t, l_1, l'_1, 2)\Big] \\ = \; & s\Big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\langle\!\langle(X_1, X_2).\big(l'_1[X_1] \mid l'_1[X_2] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t\big)\rangle\!\rangle\Big] \\ \longrightarrow \; & s\Big[t\big[l_1[b] \mid c\big] \mid l_1\langle\!\langle(X_2).\big(l'_1[a] \mid l'_1[X_2] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t\big)\rangle\!\rangle\Big] \\ \longrightarrow \; & s\Big[t\big[c\big] \mid l'_1[a] \mid l'_1[b] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t\Big] \\ \longrightarrow \; & s\Big[l'_1[a] \mid l'_1[b] \mid \overline{j_t}.r_t\Big] \end{aligned}$$

Above, the two reduction steps are used for relocation of $l_1[a]$ and $l_1[b]$ that are nested in location $t$ (with omitted trailing occurrences of $\mathbf{0}$). The third step is the synchronization between update prefix $t\langle\!\langle\dagger\rangle\!\rangle$ and location $t[c]$, where the update deletes the location and its content. This is illustrated in Figure 3.12.

### 3.3.2 The Translation, Formally

In order to give a precise account of the number of computation steps, i.e., reductions required in $\mathcal{S}$ to correctly mimic a reductions in $\mathcal{C}_{\mathtt{P}}$ we use $\mathtt{pb}(P)$ as presented in the Definition 3.2.5 (cf. Figure 3.4). We need the following definition for $\mathtt{ts}_{\mathtt{P}}(P)$, which counts all transactions in process $P$. Since translations $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$ will also need $\mathtt{ts}_{\mathtt{A}}(P)$, in the following definition we will include it.

**Definition 3.3.4.** Given a compensable process $P$, we will write $\mathtt{ts}_{\mathtt{P}}(P)$ and $\mathtt{ts}_{\mathtt{A}}(P)$ to denote the number of transactions in $P$ for $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$, respectively — see Figure 3.13 for a definition.

Whenever a notion coincides for the both semantics, we shall avoid decorations $\mathtt{P}$ and $\mathtt{A}$. It should be noted that the number of protected blocks and transactions in the default activity of the transaction corresponds to the number of locations $p_{t,\rho}[\cdot]$ and $\beta_{t,\rho}[\cdot]$ after encoding of protected blocks and transactions in this transaction. The last ingredient we need to translate

$$\mathtt{ts}_{\mathtt{P}}(t[P,Q]) = 1 \qquad \mathtt{ts}_{\mathtt{A}}(t[P,Q]) = 1 + \mathtt{ts}_{\mathtt{A}}(P) \qquad \mathtt{ts}(\pi.P) = \mathtt{ts}(\langle P \rangle) = 0$$
$$\mathtt{ts}(P \mid Q) = \mathtt{ts}(P) + \mathtt{ts}(Q) \qquad \mathtt{ts}((\nu x)P) = \mathtt{ts}(P) \qquad \mathtt{ts}(!\pi.P) = \mathtt{ts}(\mathbf{0}) = 0$$

Figure 3.13: Number of transactions.

$$\begin{aligned}
(\!|\langle P \rangle|\!)_\rho &= p_\rho\big[(\!|P|\!)_\varepsilon\big] \\
(\!|t[P,Q]|\!)_\rho &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}\big] \mid t.\,(\mathtt{extrp}\langle\!\langle t, (\!|P|\!)_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho \rangle\!\rangle \mid p_\rho[(\!|Q|\!)_\varepsilon])\Big] \\
&\qquad \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
(\!|\bar{t}.P|\!)_\rho &= \bar{t}.h_t.(\!|P|\!)_\rho
\end{aligned}$$

Figure 3.14: Translating $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$.

$\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$ is the following auxiliary process, where we use functions $\mathtt{nl}(l,\cdot)$ and $\mathtt{ch}(t,\cdot)$ defined in Definition 3.2.1. Note that, we assume that functions $\mathtt{nl}(\cdot,\cdot)$ and $\mathtt{ch}(\cdot,\cdot)$ operate only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that they are applied once they are provided with an argument.

**Definition 3.3.5** (Update Prefix for Extraction)**.** Let $t, l_1, l_1', l_2, l_2'$ be names and $P$ is an adaptable process. We write $\mathtt{extrp}\langle\!\langle t, P, l_1, l_1', l_2, l_2' \rangle\!\rangle$ to stand for the following (subjective) update prefix:

$$\mathtt{extrp}\langle\!\langle t, P, l_1, l_1', l_2, l_2' \rangle\!\rangle = t\langle\!\langle (Y).t[Y] \mid \mathtt{ch}(t,Y) \mid \mathtt{outp}^{\mathtt{s}}(t, P, l_1, l_1', l_2, l_2', \mathtt{nl}(l_1,Y), \mathtt{nl}(l_2,Y)) \rangle\!\rangle \tag{3.37}$$

Now, we may formally define translation $(\!|\cdot|\!)_\rho$:

**Definition 3.3.6** (Translating $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with preserving semantics into (subjective) adaptable processes as a tuple $((\!|\cdot|\!)_\rho, \varphi_{(\!|\cdot|\!)_\rho})$ where:

(a) The renaming policy $\varphi_{(\!|\cdot|\!)_\rho} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$\varphi_{(\!|\cdot|\!)_\rho}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, j_x, r_x\} \cup \{p_\rho, \beta_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{3.38}$$

(b) The translation $(\!|\cdot|\!)_\rho : \mathcal{C}_{\mathtt{P}} \to \mathcal{S}$ is as in Figure 3.14 and as a homomorphism for other operators.

As in previous presented encodings, here again the main challenge in the translation is in representing transactions and protected blocks as adaptable processes. More in details:

- The translation of a protected block found at path $\rho$ will be enclosed in the location $p_\rho$.
- In the translation of $t[P,Q]$ we represent processes $P$ and $Q$ independently, using processes in separate locations. More in details:
  - As in the encoding $[\![\cdot]\!]_\rho$ (cf. Figure 3.3), the structure of a transaction and the number of its top-level processes dynamically changes if there is a failure signal.

- Whenever we need to extract processes located at $p_{t,\rho}$ and $\beta_{t,\rho}$ we will first substitute $Y$ in process $\mathtt{outp^s}$ (cf. (3.36)) by the content of the location $t$ and count the current number of locations $p_{t,\rho}$ and $\beta_{t,\rho}$.
- The translation of the transaction body $P$ with location $t$ is nested in location $\beta_\rho$, and the compensation activity $Q$ is encoded as a protected block and nested in location $p_\rho$.
- If $P$ contains $n$ top-level protected blocks and $m$ top-level transaction scopes (with $n, m > 0$) when the failure signal $\bar{t}$ is activated, after synchronizations on $t$ and updates, the translation will release $n + m$ successive update prefixes by using auxiliary processes $\mathtt{outp^s}$.
- Indeed, thanks to processes $\mathtt{outp^s}$, $n$ protected blocks at location $p_{t,\rho}$ and $m$ transaction scopes at location $\beta_{t,\rho}$ will be moved to their parent locations ($p_\rho$ and $\beta_\rho$, respectively). Subsequently, there is a synchronization on location $t$ that discards it with its content.
- After that, there are synchronizations on names $j_t, \beta_\rho, r_t$, and $h_t$. If transaction $t$ had nested transactions and location name $t$ is lost, we have to take the name $t$ out also from all the paths that contained it. To this end we use function $\mathcal{E}(P, s)$ (cf. Definition 3.3.2).

- With the above intuitions, translations for the remaining constructs should be self-explanatory.

The next subsection provides correctness of the translation presented in Definition 3.3.6 and includes proofs of *structural criteria* and *operational correspondence*.

### 3.3.3 Translation Correctness

In this section, we address the two criteria in Definition 2.3.5: *name invariance* and *operational correspondence*. We will investigate the other criteria as a part of future work. Our results apply for well-formed processes as in Definition 2.2.4.

#### 3.3.3.1 Semantic Criteria - Name invariance

We now consider *name invariance*. For name invariance we use Remark 3.2.3. In the following theorem we state name invariance, by relying on the renaming policy in Definition 3.3.6 (a).

**Theorem 3.3.3** (Name invariance for $(\!|\cdot|\!)_\rho$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c :\ \varphi_{(\!|\cdot|\!)_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)_\rho}(x)\}$, and

(ii) $(\!|\sigma(P)|\!)_{\sigma(\rho)} = \sigma'((\!|P|\!)_\rho)$.

*Proof.* We define the substitution $\sigma'$ as follows:

$$\sigma'(x) = \begin{cases} \sigma(x) & \text{if } x = a \text{ or } x = t \\ h_{\sigma(t)} & \text{if } x = h_t \\ j_{\sigma(t)} & \text{if } x = j_t \\ r_{\sigma(t)} & \text{if } x = r_t \\ p_{\sigma(\rho)} & \text{if } x = p_\rho \\ \beta_{\sigma(\rho)} & \text{if } x = \beta_\rho \end{cases} \tag{3.39}$$

Now we provide proofs for (i) and (ii):

(i) Since $\mathcal{N}_c = \mathcal{N}_t \cup \mathcal{N}_s$, we consider two sub-cases for $x$:

- if $x \in \mathcal{N}_s$ then it follows that:

$$\{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)_\rho}(x)\} = \{\sigma'(y) : y \in \{x\}\} = \{\sigma'(x)\} = \{\sigma(x)\} = \varphi_{(\!|\cdot|\!)_\rho}(\sigma(x)).$$

- if $x \in \mathcal{N}_t$ then:
  - by Definition 3.3.6:

$$\varphi_{(\!|\cdot|\!)_{\sigma(\rho)}}(\sigma(x)) = \{\sigma(x), h_{\sigma(x)}, j_{\sigma(t)}, r_{\sigma(t)}\} \cup \{p_{\sigma(\rho)}, \beta_{\sigma(\rho)} : \sigma(x) \in \sigma(\rho)\}$$

  - by definition of $\sigma'$:

$$\begin{aligned}
&\{\sigma(x), h_{\sigma(x)}, j_{\sigma(t)}, r_{\sigma(t)}\} \cup \{p_{\sigma(\rho)}, \beta_{\sigma(\rho)} : \sigma(x) \in \sigma(\rho)\}\\
&= \{\sigma'(x), \sigma'(h_x), \sigma'(j_x), \sigma'(r_x)\} \cup \{\sigma'(p_\rho), \sigma'(\beta_\rho) : \sigma'(x) \in \sigma'(\rho)\}\\
&= \{\sigma'(y) : y \in \{x, h_x, j_x, r_x\}\} \cup \{\sigma'(y) : y \in \{p_\rho, \beta_\rho : x \in \rho\}\}\\
&= \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)_\rho}(x)\}
\end{aligned}$$

(ii) The proof proceeds by structural induction on $P$. In the following, given a name $x$, a path $\rho$, and process $P$, we write $\sigma x$, $\sigma\rho$, and $\sigma P$ to stand for $\sigma(x)$, $\sigma(\rho)$, and $\sigma(P)$, respectively.

***Base case:*** The statement holds for $P = \mathbf{0}$: $(\!|\sigma(\mathbf{0})|\!)_{\sigma\rho} = \sigma'((\!|\mathbf{0}|\!)_\rho) \Leftrightarrow \mathbf{0} = \mathbf{0}$.

***Inductive step:*** There are six cases, but we content ourselves by showing the case for transaction scope. Proof for all other cases are similar as in the proof of Theorem 3.2.4.

- Assume that $P = t[P_1, Q_1]$. We first apply the substitution $\sigma$ on process $P$:

$$(\!|\sigma(t[P_1, Q_1])|\!)_{\sigma\rho} = (\!|\sigma t[\sigma(P_1), \sigma(Q_1)]|\!)_{\sigma\rho}.$$

By expanding the definition of the translation in Definition 3.3.6, we have:

$$\begin{aligned}
(\!|\sigma(t[P_1, Q_1])|\!)_{\sigma\rho} = \beta_{\sigma\rho}\Big[&\sigma t\big[(\!|\sigma(P_1)|\!)_{\sigma t, \sigma\rho}\big]\\
&\mid \sigma t.\big(\mathtt{extrp}\langle\!\langle \sigma t, (\!|\sigma(P_1)|\!)_{\sigma t,\sigma\rho}, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}, \beta_{\sigma t,\sigma\rho}, \beta_{\sigma\rho}\rangle\!\rangle\\
&\mid p_{\sigma\rho}[(\!|\sigma(Q_1)|\!)_\varepsilon]\big)\Big] \mid j_{\sigma t}.\beta_{\sigma\rho}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{\sigma t}}.\overline{h_{\sigma t}}
\end{aligned}$$

By induction hypothesis it follows:

$$\begin{aligned}
(\!|\sigma(t[P_1, Q_1])|\!)_{\sigma\rho} = \beta_{\sigma\rho}\Big[&\sigma t\big[\sigma'\left((\!|P_1|\!)_{t,\rho}\right)\big]\\
&\mid \sigma t.\big(\mathtt{extrp}\langle\!\langle \sigma t, \sigma'\left((\!|P_1|\!)_{t,\rho}\right), p_{\sigma t,\sigma\rho}, p_{\sigma\rho}, \beta_{\sigma t,\sigma\rho}, \beta_{\sigma\rho}\rangle\!\rangle\\
&\mid p_{\sigma\rho}[\sigma'\left((\!|Q_1|\!)_\varepsilon\right)]\big)\Big] \mid j_{\sigma t}.\beta_{\sigma\rho}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{\sigma t}}.\overline{h_{\sigma t}}
\end{aligned} \tag{3.40}$$

On the other side, when we apply definition of substitution $\sigma'$ on $(\!|P|\!)_\rho$ the following holds:

$$\begin{aligned}
\sigma'\left((\!|t[P_1, Q_1]|\!)_\rho\right) &= \sigma'\Big(\beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}\big] \mid t.\left(\mathtt{extrp}\langle\!\langle t, (\!|P_1|\!)_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho\rangle\!\rangle \mid p_\rho[(\!|Q|\!)_\varepsilon]\right)\Big]\\
&\qquad \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\Big)\\
&= \beta_{\sigma'\rho}\Big[\sigma' t\big[\sigma'\left((\!|P|\!)_{t,\rho}\right)\big]\\
&\qquad \mid \sigma' t.\big(\mathtt{extrp}\langle\!\langle \sigma' t, \sigma'\left((\!|P|\!)_{t,\rho}\right), p_{\sigma' t,\sigma'\rho}, p_{\sigma'\rho}, \beta_{\sigma' t,\sigma'\rho}, \beta_{\sigma'\rho}\rangle\!\rangle\\
&\qquad \mid p_{\sigma'\rho}[(\!|Q|\!)_\varepsilon]\big)\Big] \mid j_{\sigma' t}.\beta_{\sigma'\rho}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{\sigma' t}}.\overline{h_{\sigma' t}}
\end{aligned} \tag{3.41}$$

Given that it is valid $\sigma'(t) = \sigma(t)$ (cf. (3.39)), it is easy to conclude that (3.40) is equal to (3.41).

■

#### 3.3.3.2 Semantic Criteria - Operational Correspondence

We now shall prove that the translation $(\!|\cdot|\!)_\rho$ satisfies operational correspondence (completeness and soundness). We now state our operational correspondence result:

**Theorem 3.3.4** (Operational Correspondence for $(\!|\cdot|\!)_\varepsilon$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{P}}$.

(1) If $P \longrightarrow P'$ then $(\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$ where for

- a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,
- b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 7 + \mathtt{pb}_{\mathtt{P}}(P_1) + \mathtt{ts}_{\mathtt{P}}(P_1)$,
- c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{P}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 7 + \mathtt{pb}_{\mathtt{P}}(F[P_1]) + \mathtt{ts}_{\mathtt{P}}(F[P_1])$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$ processes $P_1, Q, P_2$ and names $t, u$.

(2) If $(\!|P|\!)_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* (\!|P'|\!)_\varepsilon$.

In Theorem 3.3.4, Case (1) concerns completeness while Case (2) describes soundness. Case (1)(a) concerns usual synchronizations, which are translated by $(\!|\cdot|\!)_\rho$. Cases (1)-(b) and (c) concern synchronizations due to compensation signals; here the analysis distinguishes four cases, as the failure signal can be external or internal (cf. (2.1), Page 14) and the transaction can be replicated or not. In all cases, the number of reduction steps required to mimic the source transition depends on the number of protected blocks and nested transactions of the transaction being canceled.

Before we present the proof with all details, we first present all ingredients of the proof for Theorem 3.3.4.

##### 3.3.3.2.1 Auxiliary Results for Completeness and Soundness.

Given a transaction $t[P,Q]$, the following lemma ensures that the number of protected blocks and transactions in the default activity $P$ is equal to the number of locations $p_{t,\rho}$ and $\beta_{t,\rho}$ in $(\!|P|\!)_{t,\rho}$, respectively.

**Lemma 3.3.5.** Let $t[P,Q]$ and $\rho$ be a well-formed compensable process and an arbitrary path, respectively. Then it holds that $\mathtt{pb}_{\mathtt{P}}(P) = \mathtt{nl}(p_{t,\rho}, (\!|P|\!)_{t,\rho})$ and $\mathtt{ts}_{\mathtt{P}}(P) = \mathtt{nl}(\beta_{t,\rho}, (\!|P|\!)_{t,\rho})$.

*Proof.* The proof is by induction on the structure of $P$. In what follows, we illustrate two cases, $P = \langle P_1\rangle$ and $P = s[P_1,Q_1]$. The proofs for the other cases proceed similarly as in the proof of Lemma 3.2.5.

- *Case* $P = \langle P_1\rangle$: By Definition 3.2.1, Definition 3.2.5, Definition 3.4.5 and Definition 3.3.6 we have:

$$\mathtt{nl}(p_{t,\rho}, (\!|\langle P_1\rangle|\!)_{t,\rho}) = \mathtt{nl}(p_{t,\rho}, p_{t,\rho}\Big[(\!|P_1|\!)_\varepsilon\Big]) = 1 = \mathtt{pb}_{\mathtt{P}}(\langle P_1\rangle), \text{ and}$$

$$\mathtt{nl}(\beta_{t,\rho}, (\!|\langle P_1\rangle|\!)_{t,\rho}) = \mathtt{nl}(\beta_{t,\rho}, p_{t,\rho}\Big[(\!|P_1|\!)_\varepsilon\Big]) = 0 = \mathtt{ts}_{\mathtt{P}}(\langle P_1\rangle).$$

- $P = s[P_1,Q_1]$: By Definition 3.3.6,

$$[\![s[P_1,Q_1]]\!]_{t,\rho} = \beta_{t,\rho}\Big[s\big[(\!|P_1|\!)_{s,t,\rho}\big] \mid s.\,(\mathtt{extrp}\langle\!\langle s, (\!|P_1|\!)_{s,t,\rho}, p_{s,t,\rho}, p_{t,\rho}, \beta_{s,t,\rho}, \beta_{t,\rho}\rangle\!\rangle \mid p_{t,\rho}[(\!|Q_1|\!)_\varepsilon])\Big]$$
$$\mid j_s.\beta_{t,\rho}\langle\!\langle (X).X\rangle\!\rangle.\overline{r}_s.\overline{h_s}.$$

Noticing that $\mathtt{nl}(p_{t,\rho}, (\!|P_1|\!)_{s,t,\rho}) = 0$, by application of Definition 3.2.1 and Definition 3.2.5, we get $\mathtt{nl}(p_{t,\rho}, (\!|s[P_1,Q_1]|\!)_{t,\rho}) = 0 = \mathtt{pb}_{\mathtt{P}}(s[P_1,Q_1])$.
Also, $\mathtt{nl}(\beta_{t,\rho}, (\!|P_1|\!)_{s,t,\rho}) = 0$, by application of Definition 3.2.1 and Definition 3.4.5, we get $\mathtt{nl}(\beta_{t,\rho}, (\!|s[P_1,Q_1]|\!)_{t,\rho}) = 1 = \mathtt{ts}_{\mathtt{P}}(s[P_1,Q_1])$.

■

To simplify proofs of correctness, we start by defining a mapping of evaluation contexts for compensable (cf. Definition 2.2.3) into evaluation contexts for adaptable processes (cf. Definition 2.2.6). For this mapping we rely directly on translation $\mathcal{C}_\mathtt{P}$ into $\mathcal{S}$ (cf. Definition 3.3.6).

**Definition 3.3.7.** Let $\rho$ be a path. We define the following mapping $\llparenthesis \cdot \rrparenthesis_\rho$ from evaluation contexts of compensable processes into evaluation contexts of adaptable processes:

$$
\begin{aligned}
&\llparenthesis [\bullet] \rrparenthesis_\rho = [\bullet] \\
&\llparenthesis \langle C[\bullet] \rangle \rrparenthesis_\rho = p_\rho[\llparenthesis C[\bullet] \rrparenthesis_\varepsilon] \\
&\llparenthesis C[\bullet] \mid P \rrparenthesis_\rho = \llparenthesis C[\bullet] \rrparenthesis_\rho \mid \llparenthesis P \rrparenthesis_\rho \\
&\llparenthesis (\nu x) C[\bullet] \rrparenthesis_\rho = (\nu x) \llparenthesis C[\bullet] \rrparenthesis_\rho \\
&\llparenthesis t[C[\bullet], Q] \rrparenthesis_\rho = \beta_\rho\Big[ t\big[\llparenthesis C[\bullet] \rrparenthesis_{t,\rho}\big] \mid t.\,(\mathtt{extrp}\langle\!\langle t, \llparenthesis C[\bullet] \rrparenthesis_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]) \Big] \\
&\qquad\qquad\qquad \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}
$$

The following definition formalizes all possible forms for the process $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$. Recall that function $\mathtt{nl}(l, P)$, defined in Definition 3.2.1 (1), returns the number of locations $l$ in process $P$.

**Definition 3.3.8.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ (cf. Table 3.1) depending on $n = \mathtt{nl}(p_{t,\rho}, \llparenthesis P \rrparenthesis_{t,\rho})$ and $m = \mathtt{nl}(\beta_{t,\rho}, \llparenthesis P \rrparenthesis_{t,\rho})$:

1. if $n = 0$ and $m = 0$ then $p \in \{1, \ldots, 6\}$;
2. if $n > 0$ and $m = 0$ then then $\llparenthesis P \rrparenthesis_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid S$ and $p \in \{1, \ldots, n+6\}$;
3. if $n = 0$ and $m > 0$ then $\llparenthesis P \rrparenthesis_{t,\rho} = \prod_{i=1}^{m} \beta_{t,\rho}[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid S$ and $p \in \{1, \ldots, m+6\}$;
4. otherwise, if $n > 0$ and $m > 0$ then $\llparenthesis P \rrparenthesis_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid \prod_{i=1}^{m} \beta_{t,\rho}[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid S$ and $p \in \{1, \ldots, n+m+6\}$.

Table 3.1: Process $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ with $p \geq 1$. We use abbreviation $\mathtt{outp^s}$ for process $\mathtt{outp^s}(t, \llparenthesis P \rrparenthesis_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho, \mathtt{nl}(p_{t,\rho}, Y), \mathtt{nl}(\beta_{t,\rho}, Y))$.

| $(p)$ | $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ for $n, m = 0$ |
|---|---|
| (1) | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t, Y) \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big]$ $\mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| (2) | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| (3) | $\beta_\rho\Big[\overline{j_t}.r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| (4) | $\beta_\rho\Big[r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |

| | |
|---|---|
| (5) | $r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |
| (6) | $p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{h_t}$ |
| $(p)$ | $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ for $n > 0, m = 0$ |
| (1) | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outp}^{\mathsf{s}} \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+j)$ <br> $0 \leq j \leq n-1$ | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid p_{t,\rho}\langle\!\langle (X_1, \ldots, X_{n-j}).\big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t\big)$ <br> $\mid \prod_{i=1}^{j} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+n)$ | $\beta_\rho\Big[t\big[\llparenthesis P' \rrparenthesis_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(3+n)$ | $\beta_\rho\Big[\prod_{i=1}^{n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid \overline{j_t}.r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(4+n)$ | $\beta_\rho\Big[\prod_{i=1}^{n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid r_t \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(5+n)$ | $r_t \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |
| $(6+n)$ | $\prod_{i=1}^{n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{h_t}$ |
| $(p)$ | $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ for $n = 0, m > 0$ and <br> $\mathtt{Nested_t} = \prod_{k=1}^{m} \beta_\rho[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}$ |
| (1) | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outp}^{\mathsf{s}} \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+s)$ <br> $0 \leq s \leq m-1$ | $\beta_\rho\Big[t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid \beta_{t,\rho}\langle\!\langle (Y_1, \ldots, Y_{m-s}).\Big(r_t.\big(\prod_{k=1}^{m-s} \big(\beta_\rho[\mathcal{E}(Y_k, t)]$ <br> $\mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid \prod_{k=1}^{s} \big(\beta_\rho[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)$ <br> $\mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}\Big) \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+m)$ | $\beta_\rho\Big[t\big[\llparenthesis P' \rrparenthesis_{t,\rho}\big] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t} \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(3+m)$ | $\beta_\rho\Big[\overline{j_t} \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(4+m)$ | $\beta_\rho\Big[\ r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(5+m)$ | $r_t.\big(\mathtt{Netsed_t}\big) \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |

| | |
|---|---|
| $(6+m)$ | $\mathtt{Netsed_t} \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon] \mid \overline{h_t}$ |

| | |
|---|---|
| $(p)$ | $I_t^{(p)}((\!\lvert P \rvert\!)_{t,\rho}, (\!\lvert Q \rvert\!)_\varepsilon)$ for $n, m > 0$ |
| $(1)$ | $\beta_\rho\Big[t\big[(\!\lvert P \rvert\!)_{t,\rho}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathtt{ch}(t,Y) \mid \mathtt{outp^s} \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon]\rangle\!\rangle\Big]$ $\mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+j+s)$ $0 \le j \le n-1$ $0 \le s \le m-1$ | $\beta_\rho\Big[t\big[(\!\lvert P \rvert\!)_{t,\rho}\big] \mid p_{t,\rho}\langle\!\langle (X_1,\dots,X_{n-j}).\beta_{t,\rho}\langle\!\langle (Y_1,\dots,Y_{m-s}).\Big(\prod_{i=1}^{n-j} p_\rho[X_i]$ $\mid \prod_{i=1}^{j} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{m-s} (\beta_\rho[\mathcal{E}(Y_k,t)] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}})$ $\mid \prod_{k=1}^{s} (\beta_\rho[(\!\lvert P'_k \rvert\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}})\big) \mid t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}\Big)\rangle\!\rangle\rangle\!\rangle$ $\mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(2+n+m)$ | $\beta_\rho\Big[t\big[(\!\lvert P' \rvert\!)_{t,\rho}\big] \mid t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t} \mid \prod_{i=1}^{n} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon]\Big]$ $\mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(3+n+m)$ | $\beta_\rho\Big[\overline{j_t} \mid \prod_{i=1}^{n} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(4+n+m)$ | $\beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$ |
| $(5+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid r_t.\big(\mathtt{Nested_t}\big) \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |
| $(6+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!\lvert P'_i \rvert\!)_\varepsilon] \mid \mathtt{Nested_t} \mid p_\rho[(\!\lvert Q \rvert\!)_\varepsilon] \mid \overline{h_t}$ |

**Definition 3.3.9.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \ge 1$, we define the intermediate processes $O_u^{(q)}((\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}], (\!\lvert Q \rvert\!)_\varepsilon)$ (cf. Table 3.2 ) depending on $m = \mathtt{nl}(p_{u,\rho}, (\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}])$ and $n = \mathtt{nl}(\beta_{u,\rho}, (\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}])$:

1. if $n = 0$ and $m = 0$ then $q \in \{1, \dots, 7\}$;
2. if $n > 0, m = 0$ then $(\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[(\!\lvert P'_k \rvert\!)_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+n\}$;
3. if $n = 0, m > 0$ then $(\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}] = \prod_{i=1}^{m} \beta_{u,\rho}[(\!\lvert P'_k \rvert\!)_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+m\}$;
4. otherwise, if $n > 0$ and $m > 0$ then $(\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[(\!\lvert P'_k \rvert\!)_\varepsilon] \mid \prod_{i=1}^{m} \beta_{u,\rho}[(\!\lvert P'_k \rvert\!)_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+n+m\}$.

Table 3.2: Process $O_u^{(q)}((\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}, (\!\lvert Q \rvert\!)_\varepsilon])$ with $q \ge 1$. We use abbreviation $\mathtt{outp^s}$ for process $\mathtt{outp^s}(u, (\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}], p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho, \mathtt{nl}(p_{u,\rho}, Y), \mathtt{nl}(\beta_{u,\rho}, Y))$.

| | |
|---|---|
| $(q)$ | $O_u^{(q)}((\!\lvert F \rvert\!)_\rho[h_u.(\!\lvert P \rvert\!)_{\rho'}, (\!\lvert Q \rvert\!)_\varepsilon])$ for $n = m = 0$ |

| | |
|---|---|
| (1) | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y) \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u\rangle\!\rangle$ <br> $\mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| (2) | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}\big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| (3) | $\beta_\rho\Big[h_u \mid \overline{j_u}.r_u \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| (4) | $\beta_\rho\Big[h_u \mid r_u \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| (5) | $h_u \mid r_u \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| (6) | $h_u \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon] \mid \overline{h_u}$ |
| (7) | $p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}, \llparenthesis Q\rrparenthesis_\varepsilon])$ for $n > 0, m = 0$ |

| | |
|---|---|
| (1) | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y) \mid \mathtt{outp^s}\rangle\!\rangle$ <br> $\mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(2+j)$ <br> $0 \le j \le n-1$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\langle\!\langle (X_1,\ldots,X_{n-j}).$ <br> $\Big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u\Big)\rangle\!\rangle \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(2+n)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P'\rrparenthesis_{\rho'}]\big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big]$ <br> $\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(3+n)$ | $\beta_\rho\Big[h_u \mid \overline{j_u}.r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(4+n)$ | $\beta_\rho\Big[h_u \mid r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(5+n)$ | $h_u \mid r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| $(6+n)$ | $h_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon] \mid \overline{h_u}$ |
| $(7+n)$ | $\prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}, \llparenthesis Q\rrparenthesis_\varepsilon])$ for $n = 0, m > 0$ and <br> $\mathtt{Nested_u} = \prod_{k=1}^{m} \beta_\rho[\llparenthesis P'_k\rrparenthesis_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}$ |

| | |
|---|---|
| (1) | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis_\rho[h_u.\llparenthesis P\rrparenthesis_{\rho'}]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y) \mid \mathtt{outp^s}\rangle\!\rangle$ |

| | |
|---|---|
| | $\mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(2+s)$<br><br>$0 \leq s \leq m-1$ | $\beta_\rho\Big[u\big[(\!\vert F\vert\!)_\rho[h_u.(\!\vert P\vert\!)_{\rho'}]\big] \mid h_u \mid \beta_{u,\rho}\langle\!\langle (Y_1,\ldots,Y_{m-s}).$<br>$\Big(r_u.\big(\prod_{k=1}^{m-s}\big(\beta_\rho[\mathcal{E}(Y_k,u)] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big) \mid \prod_{k=1}^{s}\big(\beta_\rho[(\!\vert P'_k\vert\!)_\varepsilon]$<br>$\mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\Big)\rangle\!\rangle$<br>$\mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(2+m)$ | $\beta_\rho\Big[u\big[(\!\vert F\vert\!)_\rho[h_u.(\!\vert P'\vert\!)_{\rho'}]\big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u} \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big]$<br>$\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(3+m)$ | $\beta_\rho\Big[h_u \mid \overline{j_u} \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(4+m)$ | $\beta_\rho\Big[h_u \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(5+m)$ | $h_u \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| $(6+m)$ | $h_u \mid \mathtt{Nested_u} \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid \overline{h_u}$ |
| $(7+m)$ | $\mathtt{Nested_u} \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}((\!\vert F\vert\!)_\rho[h_u.(\!\vert P\vert\!)_{\rho'}, (\!\vert Q\vert\!)_\varepsilon])$ for $n, m > 0$ |

| | |
|---|---|
| $(1)$ | $\beta_\rho\Big[u\big[(\!\vert F\vert\!)_\rho[h_u.(\!\vert P\vert\!)_{\rho'}]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathtt{ch}(u,Y) \mid \mathtt{outp^s}\rangle\!\rangle$<br>$\mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(j+s+2)$<br><br>$0 \leq j \leq m-1$<br>$0 \leq s \leq n-1$ | $\beta_\rho\Big[u\big[(\!\vert F\vert\!)_\rho[h_u.(\!\vert P\vert\!)_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\langle\!\langle (X_1,\ldots,X_{n-j}).\beta_{u,\rho}\langle\!\langle (Y_1,\ldots,Y_{m-s}).$<br>$\Big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!\vert P'_i\vert\!)_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m-s}\big(\beta_\rho[\mathcal{E}(Y_k,u)]$<br>$\mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big) \mid \prod_{k=1}^{s}\big(\beta_\rho[(\!\vert P'_k\vert\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big)$<br>$\mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\Big)\rangle\!\rangle\rangle\!\rangle \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(2+n+m)$ | $\beta_\rho\Big[u\big[(\!\vert F\vert\!)_\rho[h_u.(\!\vert P'\vert\!)_{\rho'}]\big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u} \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)_\varepsilon] \mid r_u.\big(\mathtt{Nested_u}\big)$<br>$\mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(3+n+m)$ | $\beta_\rho\Big[h_u \mid \overline{j_u} \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)_\varepsilon] \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big]$<br>$\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(4+n+m)$ | $\beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)_\varepsilon] \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$ |
| $(5+n+m)$ | $h_u \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)_\varepsilon] \mid r_u.\big(\mathtt{Nested_u}\big) \mid p_\rho[(\!\vert Q\vert\!)_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |

| | |
|---|---|
| $(6+n+m)$ | $h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \mathtt{Nested_u} \mid p_\rho[(\!|Q|\!)_\varepsilon] \mid \overline{h_u}$ |
| $(7+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \mathtt{Nested_u} \mid p_\rho[(\!|Q|\!)_\varepsilon]$ |

For the proof of operational correspondence we also need the following statement:

**Lemma 3.3.6.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $(\!|P|\!)_\rho \equiv (\!|Q|\!)_\rho$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

#### 3.3.3.2.2 Proof of Operational Correspondence — Theorem 3.3.4

Before we provide the proof of operational correspondence (cf. Theorem 3.3.4) in detail in the following we present the roadmap of the proof. The proof follows the idea that is presented in Paragraph 3.2.3.2.5 with modified: definitions, lemmas, and theorems for translation $(\!|\cdot|\!)_\rho$.

#### 3.3.3.2.3 A Roadmap for the Proofs

Part (1) of Theorem 3.3.4 is ***completeness***, i.e.,

$$\text{If } P \longrightarrow P' \text{ then } (\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$$

where $k \geq 1$ is given precisely by our statement. The proof is by induction on the derivation of $P \longrightarrow P'$ and uses:

- Proposition 2.2.3 (Page 18) for determining three base cases.
- Definition 3.2.3 (Page 77), i.e., the definition of translation.
- Lemma 3.2.9 (Page 47), which maps evaluation contexts in $\mathcal{C}_\mathtt{D}$ into evaluation contexts of $\mathcal{S}$. This lemma completely applies for mapping evaluation contexts in $\mathcal{C}_\mathtt{P}$ into evaluation contexts of $\mathcal{S}$ (cf. Definition 3.3.7).
- Lemma 3.2.14 (Page 51), which concerns function $\mathsf{ch}(\cdot,\cdot)$. It holds also for the $(\!|\cdot|\!)_\rho$, and the proof proceeds in the same direction.
- Definition 3.3.8 and Definition 3.3.9 (Page 81 and Page 83, respectively). These definitions are significant because they formalize the intermediate processes which appear during derivation.

Part (2) of Theorem 3.3.4 is ***soundness***, i.e.,

$$\text{If } (\!|P|\!)_\varepsilon \longrightarrow^n R \text{ then there is } P' \text{ such that } P \longrightarrow^* P' \text{ and } R \longrightarrow^* (\!|P'|\!)_\varepsilon$$

Proof is by induction on $n$, i.e., the length of the reduction $(\!|P|\!)_\rho \longrightarrow^n R$. We rely on several auxiliary results:

- Lemma 3.2.19 (Page 57) holds also for translation $(\!|\cdot|\!)_\rho$. We emphasize that for translation $\mathcal{C}_\mathtt{P}$ into $\mathcal{S}$ this lemma uses Definition 3.3.8 and Definition 3.3.9. To remind the reader: Lemma 3.2.19 is about the shape of process $R$, and also ensures that there is a process $P'$ with an appropriate shape. The proof proceeds by induction on $n$. We will omit the proof since it can be derived in the same way for translation $(\!|\cdot|\!)_\rho$.

- Lemma 3.2.12 (Page 49) holds also for $(\!|\cdot|\!)_\rho$. For the proof we use Lemma 3.2.10 and Corollary 3.2.11 (Page 48) that are adapted to $(\!|P|\!)_\rho$. Also, the proof uses Definition 3.3.6.

- The statement of Lemma 3.2.16 (Page 51) and Lemma 3.2.17 (Page 56) hold also for $(\!|\cdot|\!)_\rho$. These lemmas use the definition of intermediate processes given by Definition 3.3.8 and Definition 3.3.9, respectively. The proofs proceed by case analysis for the step $R \longrightarrow R'$ and uses Lemma 3.2.15.

- Lemma 3.2.20 (Page 62) also holds also for $(\!|\cdot|\!)_\rho$, and use Definition 3.3.8 and Definition 3.3.9.

Using these guidelines as a proof sketch, we now repeat Theorem 3.3.4 (Page 80) and present its proof in full detail:

**Theorem 3.3.4** (Operational Correspondence for $(\!|\cdot|\!)_\varepsilon$). Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{P}}$.

(1) If $P \longrightarrow P'$ then $(\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 7 + \mathtt{pb}_{\mathtt{P}}(P_1) + \mathtt{ts}_{\mathtt{P}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{P}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 7 + \mathtt{pb}_{\mathtt{P}}(F[P_1]) + \mathtt{ts}_{\mathtt{P}}(F[P_1])$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$ processes $P_1, Q, P_2$ and names $t, u$.

(2) If $(\!|P|\!)_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* (\!|P'|\!)_\varepsilon$.

*Proof.* We consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases $a$), $b$) and $c$) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 3.3.6, Definition 3.3.6, and Lemma 3.2.9 (Page 47) which holds for $(\!|\cdot|\!)_\rho$ .

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have the following derivation:

$$
\begin{aligned}
(\!|P|\!)_\varepsilon &\equiv (\!|E\big[C[\overline{a}.P_1] \mid D[a.P_2]\big]|\!)_\varepsilon \\
&= (\!|E|\!)_\varepsilon\big[(\!|C[\overline{a}.P_1] \mid D[a.P_2]|\!)_\rho\big] \\
&= (\!|E|\!)_\varepsilon[(\!|C|\!)_\rho[(\!|\overline{a}.P_1|\!)_{\rho'}] \mid (\!|D|\!)_\rho[(\!|a.P_2|\!)_{\rho''}]] \\
&= (\!|E|\!)_\varepsilon\big[(\!|C|\!)_\rho[\overline{a}.(\!|P_1|\!)_{\rho'}] \mid (\!|D|\!)_\rho[a.(\!|P_2|\!)_{\rho''}]\big] \\
&\longrightarrow (\!|E|\!)_\varepsilon\big[(\!|C|\!)_\rho[(\!|P_1|\!)_{\rho'}] \mid (\!|D|\!)_\rho[(\!|P_2|\!)_{\rho''}]\big] \\
&= (\!|E|\!)_\varepsilon\big[(\!|C[P_1] \mid D[P_2]|\!)_\rho\big] \\
&= (\!|E\big[C[P_1] \mid D[P_2]\big]|\!)_\varepsilon \\
&\equiv (\!|P'|\!)_\varepsilon
\end{aligned}
\tag{3.42}
$$

Therefore, the thesis holds with $k = 1$.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb}_{\mathtt{P}}(P_1)$ and $m = \mathtt{ts}_{\mathtt{P}}(P_1)$, and $P' \equiv E[C[\mathsf{extr}_{\mathtt{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation:

$$(\!|P|\!)_\varepsilon \equiv (\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!)_\varepsilon$$

$$
\begin{aligned}
&= (\!|E|\!)_\varepsilon\Big[(\!|C[t[P_1,Q]]|\!)_\rho \mid (\!|D[\overline{t}.P_2]|\!)_\rho\Big]\\
&= (\!|E|\!)_\varepsilon\Big[(\!|C|\!)_\rho[(\!|t[P_1,Q]|\!)_{\rho'}] \mid (\!|D|\!)_\rho[(\!|\overline{t}.P_2|\!)_{\rho''}]\Big]\\
&= (\!|E|\!)_\varepsilon\Big[(\!|C|\!)_\rho[\beta_{\rho'}\Big[t\big[(\!|P_1|\!)_{t,\rho'}\big] \mid t.\big(\mathtt{extrp}\langle\!\langle t, (\!|P_1|\!)_{t,\rho'}, p_{t,\rho'}, p_{\rho'}, \beta_{t,\rho'}, \beta_{\rho'}\rangle\!\rangle \mid p_{\rho'}[(\!|Q|\!)_\varepsilon]\big)\Big]\\
&\quad \mid j_t.\beta_{\rho'}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\big] \mid (\!|D|\!)_\rho[\overline{t}.h_t.(\!|P_2|\!)_{\rho''}]\Big]\\
&\longrightarrow (\!|E|\!)_\varepsilon\Big[(\!|C|\!)_\rho\big[I_t^{(1)}((\!|P_1|\!)_{t,\rho'}, (\!|Q|\!)_\varepsilon)\big] \mid (\!|D|\!)_\rho[h_t.(\!|P_2|\!)_{\rho''}]\Big]\\
&\longrightarrow^{n+m+5} (\!|E|\!)_\varepsilon\Big[(\!|C|\!)_\rho\big[\big[I_t^{(n+m+6)}((\!|P_1|\!)_{t,\rho'}, (\!|Q|\!)_\varepsilon)\big] \mid (\!|D|\!)_\rho\big[h_t.(\!|P_2|\!)_{\rho''}\big]\Big]\\
&\longrightarrow (\!|E|\!)_\varepsilon\Big[(\!|C|\!)_\rho\big[(\!|\mathsf{extr_P}(P_1) \mid \langle Q\rangle|\!)_{\rho'}\big] \mid (\!|D|\!)_\rho\big[(\!|P_2|\!)_{\rho''}\big]\Big]\\
&= (\!|E|\!)_\varepsilon\Big[(\!|C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle]|\!)_\rho \mid (\!|D[P_2]|\!)_\rho\Big]\\
&= (\!|E[C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle] \mid D[P_2]]|\!)_\varepsilon\\
&\equiv (\!|P'|\!)_\varepsilon
\end{aligned}
$$

The order/nature of these reduction steps is as follows:

i) The first synchronization concerns $t$ and $\overline{t}$.

ii) The following $n+m+5$ synchronizations can be explained as follows:

- First, we have a process relocation through the update of location $t$, as enforced by the definition of process $\mathtt{extrp}$. Process $I_t^{(1)}((\!|P_1|\!)_{t,\rho'}, (\!|Q|\!)_\varepsilon)$ is as in Definition 3.3.8 (cf. Table 3.1 ); there are four possibilities for reduction, depending on $n$ and $m$.
- Subsequently, thanks to process

$$\mathtt{outp}^{\mathsf{s}}(t, (\!|P_1|\!)_{t,\rho'}, p_{t,\rho'}, p_{\rho'}, \beta_{t,\rho'}, \beta_{\rho'}, \mathtt{nl}(p_{t,\rho'}, (\!|P_1|\!)_{t,\rho'}), \mathtt{nl}(\beta_{t,\rho'}, (\!|P_1|\!)_{t,\rho'}))$$

  we have $n+m$ reduction steps that relocate processes on location $p_{t,\rho'}$ and $\beta_{t,\rho'}$ to locations $p_{\rho'}$ and $\beta_{t,\rho'}$, respectively.
- Next reduction corresponds to the erasure of the location $t$ with all its contents, obtained by updating prefix $t\langle\!\langle \dagger\rangle\!\rangle$.
- The following reduction step is synchronization on $j_t$ and $\overline{j_t}$.
- Next, it follows to delete the location name $\beta_{\rho'}\Big[\ldots\Big]$ with $\beta_{\rho'}\langle\!\langle (X).X\rangle\!\rangle$
- The final reduction corresponds to the synchronization on names $r_t$ and $\overline{r_t}$.

iii) Finally, we have a synchronization between $h_t$ and $\overline{h_t}$, which serves to signal that all synchronizations related to location $t$ have been completed.

Therefore, we can conclude that $(\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$ for $k = 7+n+m$.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[u[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb}_{\mathsf{P}}(F[P_1])$ and $m = \mathtt{ts}_{\mathsf{P}}(F[P_1])$, and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle]$. Then we have the following derivation:

$$
\begin{aligned}
(\!|P|\!)_\varepsilon &\equiv (\!|C\big[u[F[\overline{u}.P_1],Q]\big]|\!)_\varepsilon\\
&= (\!|C|\!)_\varepsilon\big[(\!|u[F[\overline{u}.P_1],Q]|\!)_\rho\big]\\
&= (\!|C|\!)_\varepsilon\big[\beta_\rho\Big[u\big[(\!|F[\overline{u}.P_1]|\!)_{u,\rho}\big] \mid u.\big(\mathtt{extrp}\langle\!\langle u, p_{u,\rho'}, p_{\rho'}, \beta_{u,\rho'}, \beta_{\rho'}\rangle\!\rangle \mid p_\rho[(\!|Q|\!)_\varepsilon]\big)\big]\Big]\\
&\quad \mid j_u.\beta_{\rho'}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}\\
&\longrightarrow (\!|C|\!)_\varepsilon\Big[O_u^{(1)}((\!|F|\!)_{u,\rho}[h_u.(\!|P_1|\!)_{\rho'}], (\!|Q|\!)_\varepsilon)\Big]
\end{aligned}
$$

$$
\begin{aligned}
&\longrightarrow^{n+m+5} (\!|C|\!)_\varepsilon\big[O_u^{(n+m+6)}((\!|F|\!)_{u,\rho}[h_u.(\!|P_1|\!)_{\rho'}], (\!|Q|\!)_\varepsilon)\big] \\
&\longrightarrow (\!|C|\!)_\varepsilon\big[O_u^{(n+m+7)}((\!|F|\!)_{u,\rho}[h_u.(\!|P_1|\!)_{\rho'}], (\!|Q|\!)_\varepsilon)\big] \\
&\equiv (\!|C|\!)_\varepsilon\big[(\!|\mathsf{extr_P}(F[P_1])|\!)_\rho \mid p_\rho[(\!|Q|\!)_\varepsilon]\big] \\
&= (\!|C\big[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle\big]|\!)_\varepsilon \\
&\equiv (\!|P'|\!)_\varepsilon
\end{aligned}
$$

Process $O_u^{(q)}((\!|F|\!)_{u,\rho}[h_u.(\!|P_1|\!)_{\rho'}], (\!|Q|\!)_\varepsilon)$, where $q \in \{1, \dots, 7+n+m\}$, is as in Definition 3.3.9 ( cf. Table 3.2 ). In this case, the role of function $\mathsf{ch}(u,\cdot)$ is central: indeed, $\mathsf{ch}(u, (\!|F|\!)_{u,\rho}[h_u.(\!|P_1|\!)_{\rho'}])$ provides the input $h_u$ which is necessary to achieve operational correspondence.
The order/nature/number of reduction steps can be explained as in Case b) above. We can then conclude that $(\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$ for $k = 7+n+m$.

(2) **Part (2) – Soundness:** The proof of soundness follows the explanation presented in Roadmap 3.3.3.2.3 and the similar derivation that that we present in the proof of soundness for translation $\mathcal{C}_{\mathsf{D}}$ into $\mathcal{S}$ (cf. Theorem 3.2.7).

Given $(\!|P|\!)_\varepsilon \longrightarrow^n R$, by Lemma 3.2.19 which holds also for $(\!|\cdot|\!)_\varepsilon$, process $R$ has the following form:

$$
\begin{aligned}
R \equiv \prod_{w=1}^{z}(\!|E_w|\!)_\varepsilon\Big[\prod_{k=1}^{s_w}(\!|G_{k,w}|\!)_{\rho_w}\big[\prod_{i=1}^{l_k}(\!|C_{i,k,w}|\!)_{\rho'_{k,w}}\big[I^{(p)}_{t_{i,k,w}}\big] \\
\mid \prod_{j=1}^{r_k}(\!|D_{j,k,w}|\!)_{\rho'_{k,w}}\big[h_{t_{j,k,w}}.(\!|S_{t_{j,k,w}}|\!)_{\rho''_{k,w}}\big] \mid \prod_{c=1}^{m_k}(\!|L_{c,k,w}|\!)_{\rho'_{k,w}}\big[O^{(q)}_{u_{c,k,w}}\big]\big]\Big].
\end{aligned}
$$

Also by Lemma 3.2.19, we have $P \longrightarrow^* P''$ where

$$
\begin{aligned}
P'' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}}, Q_{t_{i,k,w}}]\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big] \\
\mid \prod_{c=1}^{m_k} L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}], Q_{u_{c,k,w}}]\big]\big]\Big],
\end{aligned}
$$

where by successive application of completeness it follows that $(\!|P|\!)_\varepsilon \longrightarrow^* (\!|P''|\!)_\varepsilon$.

Lemma 3.2.20 also hols for $(\!|\cdot|\!)_\rho$. Therefore, by $l_k$ successive applications of (3.28) and $m_k$ successive applications of (3.29) on process $R$, it follows that:

$$
\begin{aligned}
R \longrightarrow^* &\prod_{w=1}^{z}(\!|E_w|\!)_\varepsilon\Big[\prod_{k=1}^{s_w}(\!|G_{k,w}|\!)_{\rho_w}\big[\prod_{i=1}^{l_k}(\!|C_{i,k,w}|\!)_{\rho'_{k,w}}\big[(\!|\mathsf{extr_P}(P'_{t_{i,k,w}})|\!)_{\rho''_{k,w}} \mid (\!|\langle Q'_{t_{i,k,w}}\rangle|\!)_{\rho''_{k,w}}\big] \\
&\mid \prod_{j=1}^{r_k}(\!|D_{j,k,w}|\!)_{\rho'_{k,w}}\big[(\!|S_{t_{j,k,w}}|\!)_{\rho''_{k,w}}\big] \mid \prod_{c=1}^{m_k}[\![L_{c,k,w}]\!]_{\rho'_{k,w}}\big[(\!|\mathsf{extr_P}(F_{c,k,w}[P_{u_{c,k,w}}])|\!)_{\rho''_{k,w}} \\
&\mid (\!|\langle Q'_{u_{c,k,w}}\rangle|\!)_{\rho''_{k,w}}\big]\big]\Big] \\
= &(\!|\prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[\mathsf{extr_P}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[S_{t_{j,k,w}}\big] \\
&\mid \prod_{c=1}^{m_k} L_{c,k,w}\big[\mathsf{extr_P}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle\big]\big]\Big]|\!)_\varepsilon \equiv (\!|P'|\!)_\varepsilon.
\end{aligned}
$$

Therefore, it follows that:

$$P' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}\big[\mathsf{extr}_{\mathrm{P}}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w}\big[S_{t_{j,k,w}}\big] \mid \prod_{c=1}^{m_k} L_{c,k,w}\big[\mathsf{extr}_{\mathrm{P}}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle\big]\big]\Big].$$

Also, by Proposition 2.2.3, i.e., by $l_k$ successive applications of case b) and and $m_k$ successive applications of case c) on process $P''$, it follows that: $P'' \longrightarrow^* P'$.
By successive application of **(1) – Completeness** on the derivation $P'' \longrightarrow^* P'$ it follows that $(\!|P''|\!)_\varepsilon \longrightarrow^* (\!|P'|\!)_\varepsilon$.

■

The following example illustrates the translation.

**Example 3.3.7.** Let $P$ be a well-formed compensable process such that $P = s\big[t[v[b,\mathbf{0}],c] \mid \langle d\rangle,\mathbf{0}\big] \mid \bar{t}.\bar{s}$. By the LTS (cf. Figure 2.3), we have

$$P \xrightarrow{\tau}_{\mathrm{P}} s[v[b,\mathbf{0}] \mid \langle c\rangle \mid \langle d\rangle,\mathbf{0}] \mid \bar{s} \xrightarrow{\tau}_{\mathrm{P}} v[b,\mathbf{0}] \mid \langle c\rangle \mid \langle d\rangle.$$

We have that $\mathtt{pb}_{\mathrm{P}}(P) = 0$ and $\mathtt{ts}_{\mathrm{P}}(P) = 1$. Let $P_1 = t[v[b,\mathbf{0}],c]$: by Figure 3.4 it follows that $\mathtt{pb}_{\mathrm{P}}(P_1) = 0$; by Figure 3.13 we have $\mathtt{ts}_{\mathrm{P}}(P_1) = 1$. We have a sequential error notification such that activation starts from nested transactions on name $t$. By expanding Definition 3.3.6, we have the following translation and derivation:

$$\begin{aligned}
(\!|P|\!)_\varepsilon = {} & \beta_\varepsilon\Big[s\Big[\beta_s\Big[t\Big[\beta_{t,s}\Big[v\Big[b\Big] \mid v.\,(\mathtt{extrp}\langle\!\langle v, (\!|b|\!)_{v,t,s}, p_{v,t,s}, p_{t,s}, \beta_{v,t,s}, \beta_{t,s}\rangle\!\rangle)\Big]\\
& \mid j_v.\beta_{t,s}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_v}.\overline{h_v}\Big]\\
& \mid t.\,\big(\mathtt{extrp}\langle\!\langle t, (\!|v[b,\mathbf{0}]|\!)_{t,s}, p_{t,s}, p_s, \beta_{t,s}, \beta_s\rangle\!\rangle \mid p_s\big[c\big]\big)\Big] \mid j_t.\beta_s\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid p_s\big[d\big]\Big]\\
& \mid s.\,(\mathtt{extrp}\langle\!\langle s, (\!|P_1 \mid \langle d\rangle|\!)_s, p_s, p_\varepsilon, \beta_s, \beta_\varepsilon\rangle\!\rangle)\Big] \mid j_s.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \mid \bar{t}.h_t.\bar{s}.h_s\\
= {} & \beta_\varepsilon\Big[s\Big[\beta_s\Big[t\Big[\beta_{t,s}\Big[v\Big[b\Big] \mid v.\,(\mathtt{extrp}\langle\!\langle v, (\!|b|\!)_{v,t,s}, p_{v,t,s}, p_{t,s}, \beta_{v,t,s}, \beta_{t,s}\rangle\!\rangle)\Big]\\
& \mid j_v.\beta_{t,s}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_v}.\overline{h_v}\Big] \mid t.(t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t,Y)\\
& \mid \mathtt{outp}^{\mathsf{s}}(t, (\!|v[b,\mathbf{0}]|\!)_{t,s}, p_{t,s}, p_s, \beta_{t,s}, \beta_s, \mathtt{nl}(p_{t,s},Y), \mathtt{nl}(\beta_{t,s},Y))\rangle\!\rangle \mid p_s\big[c\big])\Big]\\
& \mid j_t.\beta_s\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid p_s\big[d\big]\Big]\\
& \mid s.\,(\mathtt{extrp}\langle\!\langle s, (\!|P_1 \mid \langle d\rangle|\!)_s, p_s, p_\varepsilon, \beta_s, \beta_\varepsilon\rangle\!\rangle)\Big] \mid j_s.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \mid \bar{t}.h_t.\bar{s}.h_s\\
\longrightarrow^8 {} & \beta_\varepsilon\Big[s\Big[\beta_s\Big[v\Big[b\Big] \mid v.\,(\mathtt{extrp}\langle\!\langle v, (\!|b|\!)_{v,s}, p_{v,s}, p_s, \beta_{v,s}, \beta_s\rangle\!\rangle)\Big]\\
& \mid j_v.\beta_s\langle\!\langle (X).X\rangle\!\rangle.\overline{r_v}.\overline{h_v} \mid p_s\big[c\big] \mid p_s\big[d\big]\Big]\\
& \mid s.\,(s\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(s,Y) \mid \mathtt{outp}^{\mathsf{s}}(s, (\!|P_1 \mid \langle d\rangle|\!)_s, p_s, p_\varepsilon, \beta_s, \beta_\varepsilon, \mathtt{nl}(p_s,Y), \mathtt{nl}(\beta_s,Y))\rangle\!\rangle)\Big]\\
& \mid j_s.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \mid \bar{s}.h_s\\
\longrightarrow^9 {} & \beta_\varepsilon\Big[v\Big[b\Big] \mid v.\,(\mathtt{extrp}\langle\!\langle v, (\!|b|\!)_v, p_v, p_\varepsilon, \beta_v, \beta_\varepsilon\rangle\!\rangle)\Big] \mid j_v.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_v}.\overline{h_v} \mid p_\varepsilon\big[c\big] \mid p_\varepsilon\big[d\big]\\
= {} & (\!|v[b,\mathbf{0}] \mid \langle c\rangle \mid \langle d\rangle|\!)_\varepsilon
\end{aligned}$$

Therefore, the number of reduction steps is $k = 17$. Indeed, we have 8 reduction steps for location $t$ and 9 reduction steps and for location $s$:

i) the first step is a synchronization on name $t$;

ii) now the process $\mathtt{extrp}\langle\langle t, (\!|b|\!)_{t,s}, p_{t,s}, p_s, \beta_{t,s}, \beta_s\rangle\rangle$ is released and the second step is synchronization on update prefix and location, respectively:

$$t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outp}^{\mathsf{s}}(t, (\!|v[b,\mathbf{0}]|\!)_{t,s}, p_{t,s}, p_s, \beta_{t,s}, \beta_s, \mathtt{nl}(p_{t,s}, Y), \mathtt{nl}(\beta_{t,s}, Y))\rangle\rangle,$$
$$t\Big[\beta_{t,s}\Big[v\Big[b\Big] \mid v.\,(\mathtt{extrp}\langle\langle v, (\!|b|\!)_{v,t,s}, p_{v,t,s}, p_{t,s}, \beta_{v,t,s}, \beta_{t,s}\rangle\rangle)\Big] \mid j_v.\beta_{t,s}\langle\langle (X).X\rangle\rangle.\overline{r_v}.\overline{h_v}\Big];$$

iii) as a result, process $\mathtt{outp}^{\mathsf{s}}(t, v, p_{t,s}, p_s, \beta_{t,s}, \beta_s, 0, 1)$ triggers the third step: the synchronization of location $\beta_{t,s}[\ldots]$ with update prefixe $\beta_{t,s}\langle\langle (X_1).\beta_s[X_1] \mid t\langle\langle\dagger\rangle\rangle.\overline{j_t}\rangle\rangle$;

iv) the fourth step is the synchronization between update prefix $t\langle\langle\dagger\rangle\rangle$ and location $t[\ldots]$, where the update deletes the location and its content (cf. (3.1));

v) the fifth step is a synchronization on name $j_t$;

vi) the sixth step is a synchronization between $\beta_s\langle\langle (X).X\rangle\rangle$ and $\beta_s\Big[\ldots\Big]$, which deletes the location $\beta_s$;

vii) the seventh step is a synchronization on name $r_t$;

viii) the eighth step is a synchronization on $h_t$, which activates visit to location on name $s$.

At this point, we have the same reduction steps but for location $s$. We have one more reduction step, though, since in process $\mathtt{outp}^{\mathsf{s}}(s, t, p_s, p_\varepsilon, \beta_s, \beta_\varepsilon, 1, 1)$ we have a location $p_s[\ldots]$ that has to be relocated to $p_\varepsilon[\ldots]$. Consequently, we have 9 reduction steps for handling the location on name $s$.

We illustrate the encoding also on the *Hotel booking scenario* discussed earlier (cf. Example 2.2.1, Page 15).

**Example 3.3.8.** Recall the hotel booking scenario where the client wants to cancel a reservation after booking and paying. In compensable processes for preserving semantics we have that:

$$\begin{aligned} Reservation \;&\xrightarrow{\tau}_{\mathsf{P}}\; t[pay.\overline{invoice},\overline{refund}] \mid \overline{pay}.(invoice + \overline{t}.refund) \\ &\xrightarrow{\tau}_{\mathsf{P}}\; t[\overline{invoice},\overline{refund}] \mid invoice + \overline{t}.refund \\ &\xrightarrow{\tau}_{\mathsf{P}}\; \langle\overline{refund}\rangle \mid refund \\ &\xrightarrow{\tau}_{\mathsf{P}}\; \langle\mathbf{0}\rangle. \end{aligned}$$

We apply the translation (cf. Definition 3.3.6) on process *Reservation*: *Reservation*:

$$\begin{aligned} (\!|Reservation|\!)_\varepsilon = \beta_\varepsilon\Big[&t\Big[book.pay.invoice\Big] \\ &\mid t.\,\big(\mathtt{extrp}\langle\langle t, book.pay.invoice, p_t, p_\varepsilon, \beta_t, \beta_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]\big)\Big] \\ &\mid j_t.\beta_\varepsilon\langle\langle (X).X\rangle\rangle.\overline{r_t}.\overline{h_t} \mid \overline{book}.\overline{pay}.\overline{t}.h_t.refund \\ \longrightarrow \beta_\varepsilon\Big[&t\Big[pay.invoice\Big] \mid t.\,\big(\mathtt{extrp}\langle\langle t, pay.invoice, p_t, p_\varepsilon, \beta_t, \beta_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]\big)\Big] \\ &\mid j_t.\beta_\varepsilon\langle\langle (X).X\rangle\rangle.\overline{r_t}.\overline{h_t} \mid \overline{pay}.\overline{t}.h_t.refund \\ \longrightarrow \beta_\varepsilon\Big[&t\Big[invoice\Big] \mid t.\,\big(\mathtt{extrp}\langle\langle t, invoice, p_t, p_\varepsilon, \beta_t, \beta_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]\big)\Big] \\ &\mid j_t.\beta_\varepsilon\langle\langle (X).X\rangle\rangle.\overline{r_t}.\overline{h_t} \mid \overline{t}.h_t.refund \\ \longrightarrow \beta_\varepsilon\Big[&t\Big[invoice\Big] \mid t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t,Y) \end{aligned}$$

$$\mid \mathtt{outp^s}(t, book.pay.invoice, p_t, p_\varepsilon, \beta_t, \beta_\varepsilon, \mathtt{nl}(p_t, Y), \mathtt{nl}(\beta_t, Y))\rangle\!\rangle \mid p_\varepsilon[\overline{refund}]\Big]$$
$$\mid j_t.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid h_t.refund$$
$$\longrightarrow \beta_\varepsilon\Big[t[invoice] \mid t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\varepsilon[\overline{refund}]\Big] \mid j_t.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid h_t.refund$$
$$\longrightarrow \beta_\varepsilon\Big[\overline{j_t}.r_t \mid p_\varepsilon[\overline{refund}]\Big] \mid j_t.\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid h_t.refund$$
$$\longrightarrow \beta_\varepsilon\Big[r_t \mid p_\varepsilon[\overline{refund}]\Big] \mid \beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \mid h_t.refund$$
$$\longrightarrow r_t \mid p_\varepsilon[\overline{refund}] \mid \overline{r_t}.\overline{h_t} \mid h_t.refund$$
$$\longrightarrow p_\varepsilon[\overline{refund}] \mid \overline{h_t} \mid h_t.refund$$
$$\longrightarrow p_\varepsilon[\overline{refund}] \mid refund$$
$$\longrightarrow p_\varepsilon[\mathbf{0}]$$

Therefore, $(\!|Reservation|\!)_\varepsilon \longrightarrow^{10} p_\varepsilon[\mathbf{0}]$. There are three reduction steps as a result of synchronizations on names *book*, *pay*, and $t$. Now, the structure of the default activity of transaction is changed and we have one reduction step for updating its current content. After that, there are six additional reduction steps: one for erasing location $t$ and its content, a synchronization between $j_t$ with $\overline{j_t}$, an update on location $\beta_\varepsilon\Big[\overline{j_t}.r_t \mid p_\varepsilon[\overline{refund}]\Big]$ with $\beta_\varepsilon\langle\!\langle (X).X\rangle\!\rangle$, and three reduction steps that result from synchronizations on names $r_t$, $h_t$, and $refund$.

## 3.4 Translating $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$

In this section we concentrate on a specific source calculus, namely the calculus in [29] with *static recovery* and *aborting semantics*. Before giving a formal presentation of the encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ we introduce some useful conventions and intuitions.

#### 3.4.0.1 The Translation, Informally

The translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$, denoted $\langle\!|\cdot|\!\rangle_\rho$, relies on the key ideas of encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$, $[\![\cdot]\!]_\rho$. The most interesting processes for translation are protected block and transaction.

The translation of a protected block $\langle P\rangle$ found at path $\rho$ is defined as before:

$$\langle\!|\langle P\rangle|\!\rangle_\rho = p_\rho\big[\langle\!|P|\!\rangle_\varepsilon\big].$$

To translate a transaction $t[P,Q]$ we use the base sets in Definition 3.1.1. Also, we use auxiliary process $\mathtt{outd^s}$ (cf. (3.2)) that is defined for encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$. As expected, this translation also requires the introduction of some additional auxiliary processes.

The aborting semantics keeps not only top-level protected blocks of a transaction, but also protected blocks from nested transactions (cf. Figure 2.2). To handle this, we define the *activation prefixes* of a process, which captures the hierarchical structure of its nested locations. Nested locations arise as a result of a translated transaction with its nested transactions. The activation prefixes contain the names of the nested locations. These names originate exclusively from its corresponding transaction name and the names of its nested transactions (i.e., locations on names $p_\rho$ are not included in the activation prefixes).

**Definition 3.4.1** (Activation Prefixes)**.** Given a located process $l[P]$, we denote by $St(l[P])$ the *containment structure* of process $l[P]$: the labeled tree (with root $l$) in which nodes are labeled with names from $\mathcal{N}_t$ such that sub-trees capture nested locations. The *activation prefixes* for $l[P]$, denoted $\mathcal{T}_l(P)$, are obtained by a post-order search in $St(l[P])$ in which the visit to a node labeled $l_i$ adds prefixes $\overline{r_{l_i}}.k_{l_i}$.

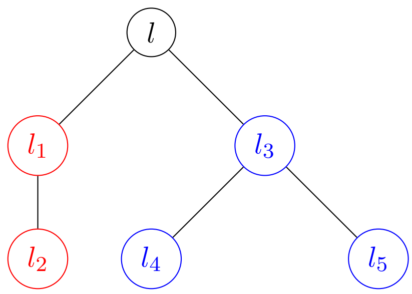



Figure 3.15: Tree for process $P$ in Example 3.4.1

**Example 3.4.1.** Given $l[P]$ with $P = l_1[l_2[p_\rho[m_1]] \mid m_2] \mid l_3[m_3 \mid l_4[m_4] \mid l_5[m_5]]$, by Definition 3.4.1 we use post-order search, therefore the root node $l$ is visited last. First, we traverse the left subtree, then the right subtree, and finally the root node (cf. Figure 3.15). In that manner we have the following activation prefixes:

$$\mathcal{T}_l(P) = \overline{r_{l_2}}.k_{l_2}.\overline{r_{l_1}}.k_{l_1}.\overline{r_{l_4}}.k_{l_4}.\overline{r_{l_5}}.k_{l_5}.\overline{r_{l_3}}.k_{l_3}.\overline{r_l}.k_l.$$

Once again, it should note that locations on names $p_\rho$ are not included in the activation prefixes.

We assume that $\mathcal{T}_l(\cdot)$ operates only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that they are applied once they are provided with an argument.

### 3.4.1 The Translation, Formally

As we discussed before, a failure signal extracts all nested protected blocks and erases nested locations; our translation does the same with the corresponding located processes and nested locations. We define the following auxiliary process, where we use functions $\mathtt{nl}(l,\cdot)$ and $\mathtt{ch}(t,\cdot)$ defined in Definition 3.2.1. Note that, we assume that functions $\mathtt{nl}(\cdot,\cdot)$ and $\mathtt{ch}(\cdot,\cdot)$ operate only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that they are applied once they are provided with an argument.

**Definition 3.4.2** (Update Prefix for Extraction)**.** Let $t, l_1$, and $l_2$ be names and $\mathtt{outd}^{\mathsf{s}}(\cdot)$ is defined with (3.2). We write $\mathtt{extra}\langle\langle t, l_1, l_2\rangle\rangle$ to stand for the following update prefix:

$$\mathtt{extra}\langle\langle t, l_1, l_2\rangle\rangle = t\langle\langle (Y).t[Y] \mid \mathtt{ch}(t, Y) \mid \mathtt{outd}^{\mathsf{s}}(l_1, l_2, \mathtt{nl}(l, Y), t\langle\langle \dagger\rangle\rangle.\overline{k_t})\rangle\rangle. \tag{3.43}$$

Now we can present the translation $\langle\!\langle\cdot\rangle\!\rangle_\rho$ formally. It is defined as follows:

**Definition 3.4.3** (Translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with aborting semantics into adaptable processes as a tuple $(\langle\!\langle\cdot\rangle\!\rangle_\rho, \varphi_{\langle\!\langle\cdot\rangle\!\rangle_\rho})$ where:

(a) The renaming policy $\varphi_{\langle\!\langle\cdot\rangle\!\rangle_\rho} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$\varphi_{\langle\!\langle\cdot\rangle\!\rangle_\rho}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, k_x, r_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases}$$

(b) The translation $\langle\!\langle\cdot\rangle\!\rangle_\rho : \mathcal{C}_{\mathtt{A}} \to \mathcal{S}$ is as in Figure 3.16 and as a homomorphism for other operators.

In the following we comment encoding more in details:

- The translation of a protected block found at path $\rho$ will be enclosed in the location $p_\rho$.

$$\langle\!\langle \langle P \rangle \rangle\!\rangle_\rho = p_\rho[\langle\!| P |\!\rangle_\varepsilon]$$
$$\langle\!| t[P,Q] |\!\rangle_\rho = t\big[\langle\!| P |\!\rangle_{t,\rho}\big] \mid r_t.\,(\texttt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]) \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t} \rangle\!\rangle$$
$$\langle\!| \,\bar{t}.P\, |\!\rangle_\rho = \bar{t}.h_t.\langle\!| P |\!\rangle_\rho$$

Figure 3.16: Translating $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$.

- As in the previously presented encodings, the translation of $t[P,Q]$ represent processes $P$ and $Q$ independently by using processes in separate locations:
  - The presence of a failure signal dynamically changes the structure of a located process on transaction name (e.g. $t$) and the number of its nested processes. Therefore, we need first to substitute $Y$ in activation prefixes $\mathcal{T}_t(Y)$ by the content of location $t$.
  - For the same reason, whenever we need to extract processes located at $p_{t,\rho}$ we will substitute $Y$ in process $\texttt{outd}^{\texttt{s}}$ by the content of the location $t$.
  - We count the current number of locations $p_{t,\rho}$ using function $\texttt{nl}(\cdot,\cdot)$ (cf. Definition 3.2.1).
  - We use the reserved name $h_t$ to control the execution of failure signals.
- Translations for the remaining constructs should be self-explanatory.

### 3.4.2 Translation Correctness

In this subsection we give proof of correctness of the translation presented in Definition 3.4.3 which includes proofs of *structural* and *semantic criteria*.

#### 3.4.2.1 Structural Criteria

In this subsection we prove the two criteria *compositionality* and *name invariance*.

##### 3.4.2.1.1 Compositionality

For the proof of compositionality criterion, we need to define a context for each process operator, which depends on free names of the subterms. This definition relies entirely on the definition of compositional context for $\mathcal{C}_{\mathtt{D}}$ (cf. Definition 3.2.4, Paragraph 3.2.3.1.1) by using $\langle\!| \cdot |\!\rangle_\rho$ instead of $[\![\cdot]\!]_\rho$. The process $\texttt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle$, is defined in Definition 3.3.5 and it depends on the function $\texttt{nl}(l_1, Y)$ that dynamically counts the current number of locations $l_1$ in the content of $t$. Another main point is $\mathcal{T}_l(Y)$ that dynamically generates activation prefixes. To mediate between translations of subterms, we define a compositional context for each process operator, which depends on free names of the subterms:

**Definition 3.4.4** (Compositional context for $\mathcal{C}_{\mathtt{A}}$)**.** For all process operator from $\mathcal{C}_{\mathtt{A}}$, instead transaction, we define a compositional context in $\mathcal{S}$ as in Definition 3.2.4. For transaction compositional context is:

$$C_{t[,],\rho}[\bullet_1, \bullet_2] = t\big[[\bullet_1]\big] \mid r_t.\,(\texttt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho[[\bullet_2]]) \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t} \rangle\!\rangle$$

Using this definition, we may now state the following result:

**Theorem 3.4.2** (Compositionality for $\langle\!\langle\cdot\rangle\!\rangle_\rho$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}_\mathtt{A}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$\begin{array}{lll}
\langle\!\langle\langle P\rangle\rangle\!\rangle_\rho = C_{\langle\rangle,\rho}[\langle\!\langle P\rangle\!\rangle_\varepsilon] & \langle\!\langle t[P,Q]\rangle\!\rangle_\rho = C_{t[,],\rho}[\langle\!\langle P\rangle\!\rangle_{t,\rho},\langle\!\langle Q\rangle\!\rangle_\varepsilon] & \langle\!\langle P\mid Q\rangle\!\rangle_\rho = C_{\mid}[\langle\!\langle P\rangle\!\rangle_\rho,\langle\!\langle Q\rangle\!\rangle_\rho] \\
\langle\!\langle a.P\rangle\!\rangle_\rho = C_{a.}[\langle\!\langle P\rangle\!\rangle_\rho] & \langle\!\langle \overline{t}.P\rangle\!\rangle_\rho = C_{\overline{t}.}[\langle\!\langle P\rangle\!\rangle_\rho] & \langle\!\langle(\nu x)P)\rangle\!\rangle_\rho = C_{(\nu x)}[\langle\!\langle P\rangle\!\rangle_\rho] \\
\langle\!\langle \overline{a}.P\rangle\!\rangle_\rho = C_{\overline{a}.}[\langle\!\langle P\rangle\!\rangle_\rho] & \langle\!\langle !\pi.P\rangle\!\rangle_\rho = C_{!\pi.}[\langle\!\langle P\rangle\!\rangle_\rho] &
\end{array}$$

*Proof.* Follows directly from the definition of contexts Definition 3.4.4 and from the definition of translation $\langle\!\langle\cdot\rangle\!\rangle_\rho : \mathcal{C}_\mathtt{A} \to \mathcal{S}$ (cf. Figure 3.16). Therefore, considering these definitions we have similar derivation as the proof of Theorem 3.2.2. ∎

### 3.4.2.1.2 Name invariance

We now state *name invariance*, by relying on the renaming policy that is presented in Definition 3.4.3 (a) and using Remark 3.2.3.

**Theorem 3.4.3** (Name invariance for $\langle\!\langle\cdot\rangle\!\rangle_\rho$ )**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c :\ \varphi_{\langle\!\langle\cdot\rangle\!\rangle_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{\langle\!\langle\cdot\rangle\!\rangle_\rho}(x)\}$, and

(ii) $\langle\!\langle\sigma(P)\rangle\!\rangle_{\sigma(\rho)} = \sigma'(\langle\!\langle P\rangle\!\rangle_\rho)$.

*Proof.* We define the substitution $\sigma'$ as follows:

$$\sigma'(x) = \begin{cases} \sigma(x) & \text{if } x = a \text{ or } x = t \\ h_{\sigma(t)} & \text{if } x = h_t \\ k_{\sigma(t)} & \text{if } x = k_t \\ r_{\sigma(t)} & \text{if } x = r_t \\ p_{\sigma(\rho)} & \text{if } x = p_\rho \end{cases} \tag{3.44}$$

Now we provide proofs for (i) and (ii):

(i) The poof uses substitution $\sigma'$ (cf. (3.44)) and has the same derivation as the proof of Theorem 3.2.4 (i).

(ii) The proof proceeds by structural induction on $P$. In the following, given a name $x$, a path $\rho$, and process $P$, we write $\sigma x$, $\sigma\rho$, and $\sigma P$ to stand for $\sigma(x)$, $\sigma(\rho)$, and $\sigma(P)$, respectively.

***Base case:*** The statement holds for $P = \mathbf{0}$: $\langle\!\langle\sigma(\mathbf{0})\rangle\!\rangle_{\sigma\rho} = \sigma'\Big(\langle\!\langle\mathbf{0}\rangle\!\rangle_\rho\Big) \Leftrightarrow \mathbf{0} = \mathbf{0}$.

***Inductive step:*** There are six cases, but we content ourselves by showing the case for transaction scope. Proof for all other cases are similar as in the proof of Theorem 3.2.4.

- *Case* $P = t[P_1,Q_1]$: We first apply the substitution $\sigma$ on process $P$:

$$\langle\!\langle\sigma(t[P_1,Q_1])\rangle\!\rangle_{\sigma\rho} = \langle\!\langle\sigma t[\sigma(P_1),\sigma(Q_1)]\rangle\!\rangle_{\sigma\rho}.$$

By expanding the definition of the translation in Definition 3.4.3, we have:

$$\langle\!\langle\sigma(t[P_1,Q_1])\rangle\!\rangle_{\sigma\rho} = \sigma t\big[\langle\!\langle\sigma P_1\rangle\!\rangle_{\sigma t,\sigma\rho}\big] \mid r_{\sigma t}.(\mathtt{extra}\langle\!\langle\sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\!\rangle \mid p_{\sigma\rho}[\langle\!\langle\sigma(Q_1)\rangle\!\rangle_\varepsilon])$$

$$| \sigma t.\sigma t\langle\langle (Y).\sigma t[Y] \mid \mathcal{T}_{\sigma t}(Y).\overline{h_{\sigma t}}\rangle\rangle$$

By induction hypothesis it follows:

$$\langle\!| \sigma(t[P_1,Q_1]) |\!\rangle_{\sigma\rho} = \sigma t\big[\langle\!| \sigma' P_1 |\!\rangle_{\sigma t,\sigma\rho}\big] \mid r_{\sigma t}.\ \big(\texttt{extra}\langle\langle \sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\rangle \mid p_{\sigma\rho}[\langle\!| \sigma'(Q_1) |\!\rangle_{\varepsilon}]\big) \quad (3.45)$$

On the other side, when we apply definition of substitution $\sigma'$ on $\langle\!| P |\!\rangle_\rho$ the following holds:

$$\begin{aligned} \sigma'\,(\langle\!| t[P_1,Q_1] |\!\rangle_\rho) &= \sigma'\big(t\big[\langle\!| P_1 |\!\rangle_{t,\rho}\big] \mid r_t.\,(\texttt{extra}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle \mid p_\rho[\langle\!| Q_1 |\!\rangle_\varepsilon]) \\ &\qquad \mid t.t\langle\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\rangle\big) \\ &= \sigma' t\big[\sigma'(\langle\!| P |\!\rangle_{t,\rho})\big] \mid r_{\sigma' t}.\ \big(\texttt{extra}\langle\langle \sigma' t, p_{\sigma' t,\sigma'\rho}, p_{\sigma'\rho}\rangle\rangle \mid p_{\sigma'\rho}[\langle\!| Q |\!\rangle_\varepsilon]\big) \\ &\qquad \mid \sigma' t.\sigma' t\langle\langle (Y).\sigma' t[Y] \mid \mathcal{T}_{\sigma' t}(Y).\overline{h_{\sigma' t}}\rangle\rangle \end{aligned} \quad (3.46)$$

Given that it is valid $\sigma'(t) = \sigma(t)$ (cf. (3.44)), it is easy to conclude that (3.45) is equal to (3.46).

■

#### 3.4.2.1.3 Semantic Criteria - Operational Correspondence

The analysis of operational correspondence follows the same ideas as in the translations $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$.

Precisely, for the proof of operational correspondence, we fully rely on Roadmap 3.2.3.2.5, Paragraph 3.2.3.2.2 and Paragraph 3.2.3.2.3. We use $\langle\!| \cdot |\!\rangle_\rho$ instead $[\![\cdot]\!]_\rho$.

In the following we present definitions that differs from those represented in the Roadmap 3.2.3.2.5 and some additional auxiliary definitions. Also, we introduce some auxiliary notions to precisely describe the number of required reduction steps.

**Definition 3.4.5.** Given a compensable process $P$, we will write $\texttt{ts}_{\mathtt{A}}(P)$ to denote the number of transaction scopes in $P$ — see Figure 3.13 for a definition.

**Remark 3.4.4.** The number of protected blocks $\texttt{pb}_{\mathtt{A}}(P)$ is as in Figure 3.4. Also, it should be noted that directly from Figure 3.4 and Figure 3.13 the following holds:

- If $\texttt{pb}_{\mathtt{A}}(P) = 0$ then $\texttt{ts}_{\mathtt{A}}(P) = 0$;
- If $\texttt{pb}_{\mathtt{A}}(P) = m$ and $m > 0$ then $\texttt{ts}_{\mathtt{A}}(P) = n$ and $0 \leq n \leq m$.

In addition to the listed functions, we need two more functions, denoted with $\texttt{d}(P)$ and $\texttt{S}(P)$, that are given in the following definition:

**Definition 3.4.6.** Let $P$ be a well-formed compensable process.

1. Function $\texttt{d}(P)$ denotes the set of default activities of transactions in $P$. It is defined as follows:

$$\texttt{d}(t[P,Q]) = \{P\} \cup \texttt{d}(P) \qquad \texttt{d}(P \mid Q) = \texttt{d}(P) \cup \texttt{d}(Q)$$
$$\texttt{d}((\nu x)P) = \texttt{d}(P) \qquad \texttt{d}(!\pi.P) = \texttt{d}(\pi.P) = \texttt{d}(\mathbf{0}) = \texttt{d}(\langle P\rangle) = \emptyset$$

2. Function $\texttt{S}(P)$ is defined as follows:

$$\texttt{S}(P) = \begin{cases} \texttt{pb}_{\mathtt{A}}(P) & \text{if } |\texttt{d}(P)| = 0 \\ \texttt{pb}_{\mathtt{A}}(P) + \sum\limits_{i=1}^{n} \texttt{pb}_{\mathtt{A}}(P_i) & \text{if } \texttt{d}(P) = \{P_1, \ldots, P_n\} \end{cases}$$

The following definition formalizes the intermediate processes that appear during derivation, denoted with $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$. As in previously presented encodings, it plays a significant role in proving completeness and soundness.

**Definition 3.4.7.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ (cf. Table 3.3) depending on $n = \mathtt{nl}(p_{t,\rho}, \llparenthesis P \rrparenthesis_{t,\rho})$, $m = \mathtt{ts_A}(P)$ and $s = \mathtt{S}(P)$:

1. if $n = 0$ then $p \in \{1, \ldots, 6\}$;
2. otherwise, if $n > 0$ and $m \geq 0$ then $\llparenthesis P \rrparenthesis_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid S$ and $p \in \{1, \ldots, 6 + n + 4m\}$.

Table 3.3: Process $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ with $p \geq 1$.

| $(p)$ | $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ for $n = 0$ |
|---|---|
| (1) | $t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]) \mid t\langle\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\rangle$ |
| (2) | $t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid r_t.\big(t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\langle\langle\dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\big) \mid \overline{r_t}.k_t.\overline{h_t}$ |
| (3) | $t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\langle\langle\dagger\rangle\rangle.\overline{k_t})\rangle\rangle) \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid k_t.\overline{h_t}$ |
| (4) | $t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid t\langle\langle\dagger\rangle\rangle.\overline{k_t} \mid k_t.\overline{h_t}$ |
| (5) | $p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{k_t} \mid k_t.\overline{h_t}$ |
| (6) | $p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid \overline{h_t}$ |
| $(p)$ | $I_t^{(p)}(\llparenthesis P \rrparenthesis_{t,\rho}, \llparenthesis Q \rrparenthesis_\varepsilon)$ for $n > 0$ |
| (1) | $t\big[\llparenthesis P \rrparenthesis_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\langle\langle t, p_{t,\rho}, p_\rho\rangle\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]) \mid t\langle\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\rangle$ |
| $(2 + 4m + s - n)$ | $t\big[\llparenthesis P' \rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_{t,\rho}[\llparenthesis P'_i \rrparenthesis_\varepsilon]\big] \mid r_t.\big(t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\langle\langle\dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon]\big) \mid \overline{r_t}.k_t.\overline{h_t}$ |
| $(3 + 4m + s - n)$ | $t\big[\llparenthesis P' \rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon]\big] \mid t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\langle\langle\dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid k_t.\overline{h_t}$ |
| $(4 + 4m + s - n + j)$ $0 \leq j \leq n - 1$ | $t\big[\llparenthesis P' \rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_\rho[\llparenthesis P'_i \rrparenthesis_\varepsilon]\big] \mid p_{t,\rho}\langle\langle (X_1, \ldots, X_{n-j}).$ $\big(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid t\langle\langle\dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\rho[\llparenthesis Q_t \rrparenthesis_\varepsilon] \mid k_t.\overline{h_t}$ |
| $(4 + 4m + s)$ | $t\big[\llparenthesis P' \rrparenthesis_{t,\rho}\big] \mid \prod_{k=1}^{n} p_\rho[\llparenthesis P'_k \rrparenthesis_\varepsilon] \mid t\langle\langle\dagger\rangle\rangle.\overline{k_t} \mid p_\rho[\llparenthesis Q \rrparenthesis_\varepsilon] \mid k_t.\overline{h_t}$ |

| | |
|---|---|
| $(5+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!| P'_k |\!\rangle_\varepsilon]\overline{k_t} \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid k_t.\overline{h_t}$ |
| $(6+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!| P'_k |\!\rangle_\varepsilon] \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid \overline{h_t}$ |

The following definition formalizes all possible forms for the process $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$.

**Definition 3.4.8.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$ (Table 3.4) depending on $n = \mathtt{nl}(p_{u,\rho}, \langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}])$, $m = \mathtt{ts_A}(F[P])$ and $s = \mathtt{S}(F[P])$:

1. for $n = 0$ we have $q \in \{1, \ldots, 7\}$, and
2. for $n > 0$ and $\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[\langle\!| P'_k |\!\rangle_\varepsilon] \mid S$ we have $q \in \{1, \ldots, 7+4m+s\}$.

Table 3.4: Process $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$ with $q \geq 1$.

| $(q)$ | $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$, $n = 0$ |
|---|---|
| (1) | $u\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon])$ $\mid u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle$ |
| (2) | $u\big[\mathbf{end}\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\big] \mid r_u.\big(u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y)$ $\mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u})\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]\big) \mid \overline{r_u}.k_u.\overline{h_u}$ |
| (3) | $u\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y)$ $\mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u})\rangle\!\rangle\big) \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid k_u.\overline{h_u}$ |
| (4) | $u\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\big] \mid h_u \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u} \mid k_u.\overline{h_u}$ |
| (5) | $p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid h_u \mid \overline{k_u} \mid k_u.\overline{h_u}$ |
| (6) | $p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid h_u \mid \overline{h_u}$ |
| (7) | $p_\rho[\langle\!| Q |\!\rangle_\varepsilon]$ |
| $(q)$ | $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$ for $n > 0$ |
| (1) | $t\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon])$ $\mid u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle$ |
| $(2+4m+s-n)$ | $u\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_{u,\rho}[\langle\!| P'_i |\!\rangle_\varepsilon]\big] \mid r_u.\big(u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y)$ $\mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u})\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]\big) \mid \overline{r_u}.k_u.\overline{h_u}$ |
| $(3+4m+s-n)$ | $u\big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P' |\!\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_\rho[\langle\!| P'_i |\!\rangle_\varepsilon]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathsf{ch}(u,Y)$ $\mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u})\rangle\!\rangle \mid h_u \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon] \mid k_u.\overline{h_u}$ |

| | |
|---|---|
| $(4+4m+s-n+j)$<br><br>$0 \le j \le n-1$ | $u\big[\langle\!\langle F\rangle\!\rangle_\rho[h_u.\langle\!\langle P'\rangle\!\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_\rho[\langle\!\langle P'_i\rangle\!\rangle_\varepsilon]\big] \mid p_{u,\rho}\langle\!\langle(X_1,\dots,X_{n-j}).$<br>$(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\langle\!\langle P'_k\rangle\!\rangle_\varepsilon] \mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u})\rangle\!\rangle \mid h_u \mid p_\rho[\langle\!\langle Q_u\rangle\!\rangle_\varepsilon] \mid k_u.\overline{h_u}$ |
| $(4+4n+s)$ | $u\big[\langle\!\langle F\rangle\!\rangle_\rho[h_u.\langle\!\langle P'\rangle\!\rangle_{\rho'}]\big] \mid \prod_{k=1}^{n} p_\rho[\langle\!\langle P'_k\rangle\!\rangle_\varepsilon] \mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{k_u} \mid h_u \mid p_\rho[\langle\!\langle Q\rangle\!\rangle_\varepsilon] \mid k_u.\overline{h_u}$ |
| $(5+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\langle P'_k\rangle\!\rangle_\varepsilon] \mid h_u \mid \overline{k_u} \mid p_\rho[\langle\!\langle Q\rangle\!\rangle_\varepsilon] \mid k_u.\overline{h_u}$ |
| $(6+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\langle P'_k\rangle\!\rangle_\varepsilon] \mid h_u \mid p_\rho[\langle\!\langle Q\rangle\!\rangle_\varepsilon] \mid \overline{h_u}$ |
| $(7+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\langle P'_k\rangle\!\rangle_\varepsilon] \mid p_\rho[\langle\!\langle Q\rangle\!\rangle_\varepsilon]$ |

For the proof of operational correspondence we need the following statement:

**Lemma 3.4.5.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\langle\!\langle P\rangle\!\rangle_\rho \equiv \langle\!\langle Q\rangle\!\rangle_\rho$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

We now state our operational correspondence result:

**Theorem 3.4.6** (Operational Correspondence for $\langle\!\langle\cdot\rangle\!\rangle_\rho$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{A}}$.

(1) If $P \longrightarrow P'$ then $\langle\!\langle P\rangle\!\rangle_\varepsilon \longrightarrow^k \langle\!\langle P'\rangle\!\rangle_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{P}}(F[P_1])$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$ processes $P_1, Q, P_2$ and names $t, u$.

(2) If $\langle\!\langle P\rangle\!\rangle_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* \langle\!\langle P'\rangle\!\rangle_\varepsilon$.

*Proof.* As in all previously presented encodings, in the following we consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases *a*), *b*) and *c*) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 3.4.5, Definition 3.4.3, and Lemma 3.2.9 (Page 47) that applies also for $\langle\!\langle\cdot\rangle\!\rangle_\rho$.

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have that derivation corresponds to the derivation presented in (3.30). Therefore, the thesis holds with $k = 1$.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb_A}(P_1)$, $m = \mathtt{ts_A}(P_1)$ and $s = \mathtt{S}(P_1)$, and $P' \equiv E[C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation:

$$
\begin{aligned}
\langle\!|P|\!\rangle_\varepsilon &\equiv \langle\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!\rangle_\varepsilon \\
&= \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C[t[P_1,Q]]|\!\rangle_\rho \mid \langle\!|D[\overline{t}.P_2]|\!\rangle_\rho\Big] \\
&= \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho[\langle\!|t[P_1,Q]|\!\rangle_{\rho'}] \mid \langle\!|D|\!\rangle_\rho[\langle\!|\overline{t}.P_2|\!\rangle_{\rho''}]\Big] \\
&= \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho\Big[t\big[\langle\!|P|\!\rangle_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!|Q|\!\rangle_\varepsilon]) \\
&\qquad \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle\Big] \mid \langle\!|D|\!\rangle_\rho[\overline{t}.h_t.\langle\!|P_2|\!\rangle_{\rho''}]\Big] \\
&\longrightarrow \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho\big[I_t^{(1)}(\langle\!|P_1|\!\rangle_{t,\rho'}, \langle\!|Q|\!\rangle_\varepsilon)\big] \mid \langle\!|D|\!\rangle_\rho\big[h_t.\langle\!|P_2|\!\rangle_{\rho''}\big]\Big] \\
\longrightarrow^{2+4m+s-n}\ &\langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho\big[\big[I_t^{(3+4m+s-n)}(\langle\!|P_1|\!\rangle_{t,\rho'}, \langle\!|Q|\!\rangle_\varepsilon) \mid \langle\!|D|\!\rangle_\rho\big[h_t.\langle\!|P_2|\!\rangle_{\rho''}\big]\Big] \\
\longrightarrow^{n+3}\ &\langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho\big[\big[I_t^{(6+4m+s)}(\langle\!|P_1|\!\rangle_{t,\rho'}, \langle\!|Q|\!\rangle_\varepsilon) \mid \langle\!|D|\!\rangle_\rho\big[h_t.\langle\!|P_2|\!\rangle_{\rho''}\big]\Big] \\
&\longrightarrow \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C|\!\rangle_\rho\big[\langle\!|\mathsf{extr_A}(P_1) \mid \langle Q\rangle|\!\rangle_{\rho'}\big] \mid \langle\!|D|\!\rangle_\rho\big[\langle\!|P_2|\!\rangle_{\rho''}\big]\Big] \\
&= \langle\!|E|\!\rangle_\varepsilon\Big[\langle\!|C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle]|\!\rangle_\rho \mid \langle\!|D[P_2]|\!\rangle_\rho\Big] \\
&= \langle\!|E[C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle] \mid D[P_2]]|\!\rangle_\varepsilon \\
&\equiv \langle\!|P'|\!\rangle_\varepsilon
\end{aligned}
$$

The order/nature of these reduction steps is as follows:

i) The first synchronization concerns $t$ and $\overline{t}$.

ii) In the next step we have a process relocation through the update of location $t$.

iii) There are $4m + s - n$ reduction steps and their description is given in the following:
- $4m$ reduction steps correspond to:
  - relocation through the update of (nested) location $t_i$, where $i \in \{1, \ldots, m\}$,
  - synchronization on names $\overline{r_{t_i}}, k_{t_i}$ with corresponding $r_{t_i}$ and $\overline{k_{t_i}}$, respectively. These steps come from activation prefix $\mathcal{T}_t(\langle\!|P|\!\rangle_{t,\rho})$ and,
  - erasure of the location $t_i$ with all its contents, obtained by updating prefix $t_i\langle\!\langle \dagger\rangle\!\rangle$.
- $s - n$ reduction sreps occur as a consequence of relocation all processes that are on location $p_{t_i,t,\rho'}$ to location $p_{t,\rho'}$ (i.e., in the calculus of compensable processes this means that we have to extract all protected blocks from transaction that are nested in $t[P,Q]$).

iv) We have reduction step as a synchronization on $r_t$ and $\overline{r_t}$.

v) We have again a process relocation through the update of location $t$, as enforced by the definition of process `extra`.

vi) Subsequently, thanks to process

$$\mathtt{outd}^{\mathsf{s}}(p_{t,\rho'}, p_{\rho'}, \mathtt{nl}(p_{t,\rho'}, [\![P_1]\!]_{t,\rho'}), t\langle\!\langle\dagger\rangle\!\rangle.\overline{k_t})$$

we have $n$ reduction steps that relocate processes on location $p_{t,\rho'}$ to location $p_{\rho'}$.

vii) Next reduction corresponds to the erasure of the location $t$ with all its contents, obtained by updating prefix $t\langle\!\langle\dagger\rangle\!\rangle$.

viii) Next reduction corresponds to the synchronization on names $k_t$ and $\overline{k_t}$.

ix) Finally, we have a synchronization between $h_t$ and $\overline{h_t}$, which serves to signal that all synchronizations related to location $t$ have been completed.

Therefore, we can conclude that for $\langle\!| P |\!\rangle_\varepsilon \longrightarrow^k \langle\!| P' |\!\rangle_\varepsilon$ such that $k = 7 + 4m + s$.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[t[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb}_{\mathtt{A}}(F[P_1])$, $m = \mathtt{ts}_{\mathtt{A}}(P_1)$, $s = \mathtt{S}(P_1)$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q \rangle]$. Then we have the following derivation:

$$
\begin{aligned}
\langle\!| P |\!\rangle_\varepsilon &\equiv \langle\!| C[u[F[\overline{u}.P_1],Q]] |\!\rangle_\varepsilon \\
&= \langle\!| C |\!\rangle_\varepsilon \big[\langle\!| u[F[\overline{u}.P_1],Q] |\!\rangle_\rho\big] \\
&= \langle\!| C |\!\rangle_\varepsilon \Big[u\big[\langle\!| F[\overline{u}.P_1] |\!\rangle_{u,\rho}\big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]\big) \\
&\qquad \mid u.u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle]\Big]\Big] \\
&= \langle\!| C |\!\rangle_\varepsilon \Big[u\big[\langle\!| F |\!\rangle_{u,\rho}[\overline{u}.h_u.\langle\!| P_1 |\!\rangle_{\rho'}]\big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]\big) \\
&\qquad \mid u.u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle\Big] \\
&\longrightarrow \langle\!| C |\!\rangle_\varepsilon \Big[O_u^{(1)}(\langle\!| F |\!\rangle_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon) \\
&\longrightarrow^{2+4m+s-n} \langle\!| C |\!\rangle_\varepsilon \big[O_u^{(3+4m+s-n)}(\langle\!| F |\!\rangle_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)\big] \\
&\longrightarrow^{n+4} \langle\!| C |\!\rangle_\varepsilon \big[O_u^{(7+4m+s)}(\langle\!| F |\!\rangle_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)\big] \\
&\equiv \langle\!| C |\!\rangle_\varepsilon \big[\langle\!| \mathsf{extr}_{\mathtt{A}}(F[P_1]) |\!\rangle_\rho \mid p_\rho[\langle\!| Q |\!\rangle_\varepsilon]\big] \\
&= \langle\!| C\big[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q \rangle\big] |\!\rangle_\varepsilon \\
&\equiv \langle\!| P' |\!\rangle_\varepsilon
\end{aligned}
$$

Process $O_u^{(q)}(\langle\!| F |\!\rangle_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle_{\rho'}], [\![Q]\!]_\varepsilon)$, where $q \in \{1, \ldots, 7 + 4m + n\}$, is as in Definition 3.4.8 (Table 3.4). In this case, the role of function $\mathsf{ch}(u, \cdot)$ is central: indeed, $\mathsf{ch}(u, \langle\!| F |\!\rangle_{u,\rho}\big[h_u.\langle\!| P_1 |\!\rangle_{\rho'}\big])$ provides the input $h_u$ which is necessary to achieve operational correspondence.

The order and number of reduction steps can be explained as in Case b) above. We can then conclude that $\langle\!| P |\!\rangle_\varepsilon \longrightarrow^k \langle\!| P' |\!\rangle_\varepsilon$ such that $k = 7 + 4m + s$.

(2) **Part (2) – Soundness:** For the proof of soundness we use auxiliary results presented in Paragraph 3.2.3.2.3 applied to encoding of aborting semantics instead encoding of discarding semantics. Also the proof use Definition 3.4.7 and Definition 3.4.8. Therefore, the proof of soundness follows the explanation presented in Roadmap 3.2.3.2.5 and the same derivation that is presented in the proof of soundness for translation $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ (cf. Theorem 3.2.7 – Soundness).

■

The following example illustrates the translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$.

**Example 3.4.7.** Notably, $P$ is well-formed compensable process from Example 3.2.23. By the Figure 2.3 we have:

$$P \xrightarrow{\tau}_{\mathtt{A}} s[\langle a \rangle \mid \langle b \rangle \mid \langle d \rangle, \mathbf{0}] \mid \overline{s} \xrightarrow{\tau}_{\mathtt{A}} \langle a \rangle \mid \langle b \rangle \mid \langle d \rangle.$$

We apply the translation $\langle\!| \cdot |\!\rangle_\rho$ on $P$ and illustrate its behaviors:

$$
\begin{aligned}
\langle\!| P |\!\rangle_\varepsilon = s\Big[&t\big[p_{t,s}\big[a\big] \mid p_{t,s}\big[b\big] \mid c\big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,s}, p_s\rangle\!\rangle \mid p_s\big[d\big] \\
&\mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle\big)\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle\big) \\
&\mid s.s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \mid \overline{t}.h_t.\overline{s}.h_s \\
\longrightarrow^2 \; s\Big[&t\big[p_{t,s}\big[a\big] \mid p_{t,s}\big[b\big] \mid c\big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,s}, p_s\rangle\!\rangle \mid p_s\big[d\big]\big)\Big] \mid \overline{r_t}.k_t.\overline{h_t}
\end{aligned}
$$

$$
\begin{aligned}
&\mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle\big) \mid s.s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \mid h_t.\overline{s}.h_s \\
\longrightarrow^7\ & s\Big[p_s\big[a\big] \mid p_s\big[b\big] \mid p_s\big[d\big]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle\big) \mid s.s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \mid \overline{s}.h_s \\
\longrightarrow^2\ & s\Big[p_s\big[a\big] \mid p_s\big[b\big] \mid p_s\big[d\big]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_s, p_\varepsilon\rangle\!\rangle\big) \mid \overline{r_s}.k_s.\overline{h_s} \mid h_s \\
\longrightarrow^8\ & p_\varepsilon\big[a\big] \mid p_\varepsilon\big[b\big] \mid p_\varepsilon\big[d\big].
\end{aligned}
$$

The total number of reduction steps is $k = 19$. There are 9 steps for location $t$ and 10 steps for location $s$. In the following we provide explanation for these steps:

i) the first step is a synchronization on name $t$;

ii) the second step is the synchronization between $t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle$ and $t\big[p_{t,s}\big[a\big] \mid p_{t,s}\big[b\big] \mid c\big]$;

iii) the third step is synchronization on name $r_t$, where $\overline{r_t}$ comes from $\mathcal{T}_t(p_{t,s}\big[a\big] \mid p_{t,s}\big[b\big] \mid c) = \overline{r_t}.k_t$;

iv) now the process $\mathtt{extrp}\langle\!\langle t, p_{t,s}, p_s\rangle\!\rangle$ is released and the fourth step is the synchronization between update prefix and location, respectively:

$$t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t, Y) \mid \mathtt{outd}^{\mathsf{s}}(p_{t,s}, p_s, \mathtt{nl}(p_{t,s}, Y), t\langle\!\langle\dagger\rangle\!\rangle.\overline{k_t})\rangle\!\rangle \mid t\big[p_{t,s}\big[a\big] \mid p_{t,s}\big[b\big] \mid c\big];$$

v) we get process $\mathtt{outd}^{\mathsf{s}}(p_{t,s}, p_s, 2, t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t})$, which triggers the fifth and sixth reductions: the synchronizations between $p_{t,s}[a]$ and $p_{t,s}[b]$ and update prefixes $p_{t,s}\langle\!\langle (X_1, X_2).p_s[X_1] \mid p_s[X_2] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}\rangle\!\rangle$;

vi) the seventh step is the synchronization between update prefix $t\langle\!\langle\dagger\rangle\!\rangle$ and location $t[c]$, where the update prefix deletes the location together with its content (cf. (3.1));

vii) the eighth and ninth reduction steps are synchronizations on names $k_t$ and $h_t$.

At this point, we have the same reduction steps for location $s$. We have one more reduction step, though, since in process $\mathtt{outd}^{\mathsf{s}}(p_s, p_\varepsilon, 3, s\langle\!\langle\dagger\rangle\!\rangle.\overline{k_s})$ we have 3 locations $p_s[\ldots]$ that have to be relocated to $p_\varepsilon[\ldots]$. Consequently, we have 10 reduction steps for handling location $s$.

We illustrate the encoding also on the *Hotel booking scenario* discussed earlier (§ 2, Example 2.2.1, Page 15).

**Example 3.4.8.** Recall the hotel booking scenario where the client wants to cancel a reservation after booking and paying. Using compensable processes with aborting semantics we have the following derivation:

$$
\begin{aligned}
\mathit{Reservation}\ &\xrightarrow{\tau}_{\mathtt{A}}\ t[pay.\overline{invoice}, \overline{refund}] \mid \overline{pay}.(invoice + \overline{t}.refund) \\
&\xrightarrow{\tau}_{\mathtt{A}}\ t[\overline{invoice}, \overline{refund}] \mid invoice + \overline{t}.refund \\
&\xrightarrow{\tau}_{\mathtt{A}}\ \langle \overline{refund}\rangle \mid refund \\
&\xrightarrow{\tau}_{\mathtt{A}}\ \langle \mathbf{0}\rangle.
\end{aligned}
$$

We apply the translation in Definition 3.4.3 to process *Reservation*:

$$
\begin{aligned}
\llparenthesis \mathit{Reservation}\rrparenthesis_\varepsilon &= t\big[book.pay.invoice\big] \mid r_t.\,\big(\mathtt{extra}\langle\!\langle t, p_t, p_\varepsilon\rangle\!\rangle \mid p_\varepsilon[\overline{refund}]\big) \\
&\quad \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle \mid \overline{book}.\overline{pay}.\overline{t}.h_t.refund \\
&\longrightarrow t\big[pay.invoice\big] \mid r_t.\,\big(\mathtt{extra}\langle\!\langle t, p_t, p_\varepsilon\rangle\!\rangle \mid p_\varepsilon[\overline{refund}]\big) \\
&\quad \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle \mid \overline{pay}.\overline{t}.h_t.refund
\end{aligned}
$$

$$\begin{aligned}
&\longrightarrow t\big[invoice\big] \mid r_t.\,\big(\mathtt{extra}\langle\langle t, p_t, p_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]\big)\\
&\qquad \mid t.t\langle\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\rangle \mid \overline{t}.h_t.refund\\
&\longrightarrow t\big[invoice\big] \mid r_t.\,\big(\mathtt{extra}\langle\langle t, p_t, p_\varepsilon\rangle\rangle \mid p_\varepsilon[\overline{refund}]\big)\\
&\qquad \mid t\langle\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\rangle \mid h_t.refund\\
&\longrightarrow t\big[invoice\big] \mid r_t.(t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t,Y)\\
&\qquad \mid \mathtt{outd}^{\mathtt{s}}(p_t, \mathtt{nl}(p_t,Y), t\langle\langle \dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\varepsilon[\overline{refund}]) \mid \overline{r_t}.k_t.\overline{h_t} \mid h_t.refund\\
&\longrightarrow t\big[invoice\big] \mid t\langle\langle (Y).t[Y] \mid \mathsf{ch}(t,Y)\\
&\qquad \mid \mathtt{outd}^{\mathtt{s}}(p_t, \mathtt{nl}(p_t,Y), t\langle\langle \dagger\rangle\rangle.\overline{k_t})\rangle\rangle \mid p_\varepsilon[\overline{refund}] \mid k_t.\overline{h_t} \mid h_t.refund\\
&\longrightarrow t\big[invoice\big] \mid \mathtt{outd}^{\mathtt{s}}(p_t, p_\varepsilon, 0, t\langle\langle \dagger\rangle\rangle.\overline{k_t}) \mid p_\varepsilon[\overline{refund}] \mid k_t.\overline{h_t} \mid h_t.refund\\
&\;\equiv t\big[invoice\big] \mid t\langle\langle \dagger\rangle\rangle.\overline{k_t} \mid p_\varepsilon[\overline{refund}] \mid k_t.\overline{h_t} \mid h_t.refund\\
&\longrightarrow \overline{k_t} \mid p_\varepsilon[\overline{refund}] \mid k_t.\overline{h_t} \mid h_t.refund\\
&\longrightarrow p_\varepsilon[\overline{refund}] \mid \overline{h_t} \mid h_t.refund\\
&\longrightarrow p_\varepsilon[\overline{refund}] \mid refund\\
&\longrightarrow p_\varepsilon[\mathbf{0}]
\end{aligned}$$

The three steps taken at the beginning are explained in all variants of this example (cf. Example 3.3.8). The fourth step is the update on location $t$; the fifth step is a synchronization on name $r_t$; the sixth step is again an update on the location $t$; the seventh step is a synchronization on name $k_t$; the eighth step deletes location $t$ with its content; the last two steps are synchronizations on names $h_t$ and $refund$. Therefore, we get $\langle\!\langle Reservation\rangle\!\rangle_\varepsilon \longrightarrow^{10} p_t[\mathbf{0}]$.

**Brief summary of the chapter**:
In this chapter, we introduced all preliminaries for *encodings* $\mathcal{C}$ *into* $\mathcal{A}$ and informally acquainted the reader with the basic intuition of encoding. Also, the main result is the *valid encodings* of calculus for compensable processes into the calculus of adaptable processes with the subjective update (encodings $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, and $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$).

Encodings $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, and $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$, which is analyzed in the next chapter, follow and mimic the basic intuition of the encoding presented in this chapter. Therefore, we believe that it will be easier for the reader to follow and adopt the results from the upcoming chapter.

# CHAPTER 4

# Encoding Compensable into Adaptable Processes with Objective Update

In the previous chapter, we provided a detailed explanation of the *translation* of the calculus of compensable processes with static recovery ($\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}_{\mathtt{A}}$) into the calculus of adaptable processes with subjective update ($\mathcal{S}$). This chapter turns our attention to the translation of $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, and $\mathcal{C}_{\mathtt{A}}$ into adaptable processes with *objective update* ($\mathcal{O}$), where a located process is reconfigured in own its context by an update prefix residing in a different context. Also, it introduces an encodability criterion called *efficiency*. In the following we present a brief structure of the chapter:

**Section 4.1** introduces the translation of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$ in an informal way to acquaint the reader with the basic intuition and the difference concerning the translation of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$. Then the formal definition of encoding follows. The main result is *valid encoding* $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$. Also, in this section we introduce a criterion called *efficiency*. We prove that encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$ because they induce tighter operational correspondences.

**Section 4.2** presents the translation of $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$ informally and formally. Then the formal definition of encoding follows. Also, we prove that encodings satisfy *name invariance* and *operational correspondence (completeness and soundness)*. We compare encodings $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$ with encodings $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$. Based on the *efficiency* criterion, we prove that encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$ because they induce tighter operational correspondences.

**Section 4.3** presents the translation of $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$ informally, then goes on to give a formal definition of translation. Then the formal definition of encoding follows. We prove that the encoding satisfies *compositionality*, *name invariance*, and *operational correspondence (completeness and soundness)*. By using the *efficiency* criterion, we prove that encoding $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$ because they induce tighter operational correspondences.

## 4.1 Translating $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$

In this section we concentrate on a source calculus with *static recovery* and *discarding semantics.* Before a formal presentation of the translation we introduce some useful conventions and intuitions. It is important to emphasize that the encoding of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$ relies on the basic idea/intuition that we presented in the encoding of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ (cf. Section 3.2, § 2.2.3).

### 4.1.1 The Translation, Informally

To encode transactions and their extraction function we use the auxiliary process $\mathtt{outd}^{\mathtt{o}}(l_1, l_2, n, Q)$, which is similar to the process $\mathtt{outd}^{\mathtt{s}}(l_1, l_2, n, Q)$ (cf. (3.2)) that we used in the encoding $[\![\cdot]\!]_\rho$. Using objective update prefixes, we define this auxiliary process as follows:

$$\mathtt{outd}^{\mathtt{o}}(l_1, l_2, n, Q) = \begin{cases} Q & \text{if } n = 0 \\ l_1\{(X_1, \ldots, X_n).z_t\{(Z).\prod_{i=1}^{n} l_2[X_i] \mid Q\}\}.z_t[\mathbf{0}] & \text{if } n > 0. \end{cases} \tag{4.1}$$

It is instructive to compare processes $\mathtt{outd}^{\mathtt{s}}$ (3.2) and $\mathtt{outd}^{\mathtt{o}}$ (4.1), because differences between them will reflect directly on the efficiency of encodings.

**Remark 4.1.1** (Comparing $\mathtt{outd}^{\mathtt{s}}$ and $\mathtt{outd}^{\mathtt{o}}$)**.** We consider two cases:

- If $n = 0$ then we have that both processes $\mathtt{out}^{\mathtt{s}}$ and $\mathtt{out}^{\mathtt{o}}$ are equal to some process $Q$. Therefore, these processes are identical, and the difference is reflected in the use of an appropriate update.

- Consider the case $n > 0$. In $\mathtt{outd}^{\mathtt{o}}$ the process $\prod_{i=1}^{n} l_2[X_i] \mid Q$ appears enclosed inside an update prefix on name $z_t$, while in $\mathtt{outd}^{\mathtt{s}}$ this is not the case. In $\mathtt{outd}^{\mathtt{o}}$, once $n$ updates on name $l_1$ have been executed, the resulting process $\prod_{i=1}^{n} l_2[P_i] \mid Q$ will be enclosed in a location (say, $t$). Process $\prod_{i=1}^{n} l_2[P_i] \mid Q$ must be relocated and $t$ must be deleted. In (4.1), this relocation is achieved via a synchronization on name $z_t$. In contrast, because $\mathtt{outd}^{\mathtt{s}}$ uses subjective updates, process reconfiguration follows in the opposite direction. This ensures that, after $n$ updates, process $\prod_{i=1}^{n} l_2[P_i] \mid Q$ will remain in its original location, and so no relocation using $z_t$ is needed — see Example 3.2.1 and Figure 3.2.

**Example 4.1.2** (Example 3.2.1, revisited)**.** Consider process

$$P' = s\Big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid \mathtt{outd}^{\mathtt{o}}(l_1,\, l_2, 2, Q)\Big],$$

similar to process $P$ in Example 3.2.1. $P'$ has the following reductions, which are illustrated in Figure 4.1:

$$\begin{aligned}
P' &= s\Big[t\big[l_1[a] \mid l_1[b] \mid c\big] \mid l_1\{(X_1, X_2).z_t\{(Z).l_2[X_1] \mid l_2[X_2] \mid Q\}\}.z_t[\mathbf{0}]\Big] \\
&\longrightarrow s\Big[t\big[l_1\{(X_2).z_t\{(Z).l_2[a] \mid l_2[X_2] \mid Q\}\} \mid l_1[b] \mid c\big] \mid z_t[\mathbf{0}]\Big] \\
&\longrightarrow s\Big[t\big[z_t\{(Z).l_2[a] \mid l_2[b] \mid Q\} \mid c\big] \mid z_t[\mathbf{0}]\Big] \\
&\longrightarrow s\Big[t\big[c \mid z_t\{(Z).l_2[a] \mid l_2[b] \mid Q\}\big] \mid z_t[\mathbf{0}]\Big] \\
&\longrightarrow s\Big[t\big[c\big] \mid l_2[a] \mid l_2[b] \mid Q\Big]
\end{aligned}$$

In this case, the wrong location is $t$: the last reduction is needed to move process $l_2[a] \mid l_2[b] \mid Q$ out of $z_t$.

Notice that the number $n$ of protected blocks in the default activity of the transaction scope is directly related to the number of reduction steps induced by our translations. If $n = 0$ then the number of reduction steps will be the same for subjective and objective updates; otherwise, if $n > 0$, the translation with subjective update will exhibit less reduction steps than the translation with objective update.

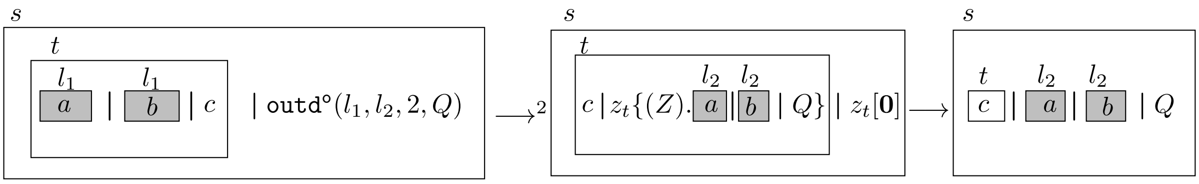



Figure 4.1: Illustrating $\mathtt{outd}^{\circ}(l_1,\, l_2, 2, Q)$.

$$
\begin{aligned}
[\![\langle P\rangle]\!]^{\circ}_{\rho} &= p_{\rho}[[\![P]\!]^{\circ}_{\varepsilon}] \\
[\![t[P,Q]]\!]^{\circ}_{\rho} &= t\Big[[\![P]\!]^{\circ}_{t,\rho}\Big] \mid t.\,(\mathtt{extrd}\{t, p_{t,\rho}, p_{\rho}\} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]) \\
[\![\,\overline{t}.P\,]\!]^{\circ}_{\rho} &= \overline{t}.h_t.[\![P]\!]^{\circ}_{\rho}
\end{aligned}
$$

Figure 4.2: Translating $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$.

### 4.1.2 The Translation, Formally

The function for determining the number of locations $\mathtt{nl}(\cdot,\cdot)$ in an adaptable process and the function $\mathtt{ch}(t,\cdot)$ are as introduced in Definition 3.2.1. We assume that function $\mathtt{nl}(\cdot,\cdot)$ and $\mathtt{ch}(t,\cdot)$ operate only over *closed processes* and, in the style of a *call-by-need evaluation strategy*, we assume that they are applied once they are provided with an argument. We now define process $\mathtt{extrd}\{t, l_1, l_2\}$:

**Definition 4.1.1** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extrd}\{t, l_1, l_2\}$ to stand for the following (objective) update prefix:

$$
\mathtt{extrd}\{t, l_1, l_2\} \;= t\{(Y).t[Y] \mid \mathtt{ch}(t, Y) \mid \mathtt{outd}^{\circ}(l_1\,,\, l_2, \mathtt{nl}(l, Y), t\{\dagger\}.\overline{h_t})\}. \tag{4.2}
$$

The intuitions for process $\mathtt{extrd}\{t, l_1, l_2\}$ are just as for process $\mathtt{extrd}\langle\!\langle t, l_1, l_2\rangle\!\rangle$ given in Definition 3.2.2. We can now formally define the translation of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$:

**Definition 4.1.2** (Translation $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$)**.** Let $\rho$ be a path. We define the translation of compensable processes into objective adaptable processes as a tuple $([\![\cdot]\!]^{\circ}_{\rho}, \varphi_{[\![\cdot]\!]^{\circ}_{\rho}})$ where:

(a)

$$
\varphi_{[\![\cdot]\!]_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, z_x\} \cup \{p_{\rho} : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{4.3}
$$

(b) $[\![\cdot]\!]^{\circ}_{\rho} : \mathcal{C}_{\mathtt{D}} \longrightarrow \mathcal{O}$ is as defined in Figure 4.2 and as a homomorphism for other operators.

The intuition for the translation of $t[P,Q]$ is as in the case of subjective update. For erasing the location and all unnecessary processes in it, we need an update prefix denoted with $t\{\dagger\}$ (cf. Convention 3.1.1).

### 4.1.3 Translation Correctness

We prove that the translation $[\![\cdot]\!]^{\circ}_{\rho}$ is a *valid encoding* (cf. Definition 2.3.5). We thus consider five criteria: compositionality, name invariance, and operational correspondence, divergence reflection, and success sensitiveness.

#### 4.1.3.1 Structural Criteria

We consider two criteria *compositionality* and *name invariace*.

##### 4.1.3.1.1 Compositionality

The first property is compositionality. Compositionality for $[\![\cdot]\!]^{\circ}_{\rho}$ as well as compositionality for $[\![\cdot]\!]_{\rho}$ (cf. Theorem 3.2.2) includes a path $\rho$ in its formulation. Initially, we need to define a *compositional context.*

**Definition 4.1.3** (Compositional context for $\mathcal{C}_{\mathtt{D}}$)**.** For every process operator from $\mathcal{C}_{\mathtt{D}}$, instead transaction scope, we define a compositional context in $\mathcal{O}$ as in Definition 3.2.4. For transaction scope a compositional context is defined with:

$$C_{t[,],\rho}[\bullet_1, \bullet_2] = t\big[[\bullet_1]\big] \mid t.\,\big(\mathtt{extrd}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho\big[[\bullet_2]\big]\big)$$

Using this definition, we may now state the following result:

**Theorem 4.1.3** (Compositionality for $[\![\cdot]\!]^{\circ}_{\rho}$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}_{\mathtt{D}}$ and for all compensable processes $P$ and $Q$ it holds that:

$$[\![\langle P\rangle]\!]^{\circ}_{\rho} = C_{\langle\rangle,\rho}\,[[\![P]\!]^{\circ}_{\varepsilon}] \qquad [\![t[P,Q]]\!]^{\circ}_{\rho} = C_{t[,],\rho}\,[[\![P]\!]^{\circ}_{t,\rho}, [\![Q]\!]^{\circ}_{\varepsilon}] \qquad [\![P \mid Q]\!]^{\circ}_{\rho} = C_{\mid}\,[[\![P]\!]^{\circ}_{\rho}, [\![Q]\!]^{\circ}_{\rho}]$$

$$[\![a.P]\!]^{\circ}_{\rho} = C_{a.}\,[[\![P]\!]^{\circ}_{\rho}] \qquad [\![\overline{t}.P]\!]^{\circ}_{\rho} = C_{\overline{t}.}\,[[\![P]\!]^{\circ}_{\rho}] \qquad [\![(\nu x)P)]\!]^{\circ}_{\rho} = C_{(\nu x)}\,[[\![P]\!]^{\circ}_{\rho}]$$

$$[\![\overline{a}.P]\!]^{\circ}_{\rho} = C_{\overline{a}.}\,[[\![P]\!]^{\circ}_{\rho}] \qquad [\![!\pi.P]\!]^{\circ}_{\rho} = C_{!\pi.}\,[[\![P]\!]^{\circ}_{\rho}]$$

*Proof.* Follows directly from the definition of contexts, Definition 3.2.4, and from the definition of $[\![\cdot]\!]^{\circ}_{\rho} : \mathcal{C}_{\mathtt{D}} \to \mathcal{O}$ (Figure 4.2) and has the same derivation as the proof of Theorem 3.2.2. ■

##### 4.1.3.1.2 Name invariance

The second property is name invariance with respect to the renaming policy in Definition 3.2.3 Case (b).

**Theorem 4.1.4** (Name invariance for $[\![\cdot]\!]^{\circ}_{\rho}$)**.** For every *well-formed* compensable processes $P$ and substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c : \varphi_{[\![\cdot]\!]^{\circ}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]^{\circ}_{\rho}}(x)\}$, and

(ii) $[\![\sigma(P)]\!]^{\circ}_{\sigma(\rho)} = \sigma'([\![P]\!]^{\circ}_{\rho})$

*Proof.* The proof proceeds in the same way as the proof of Theorem 3.2.4 by using $[\![\cdot]\!]^{\circ}_{\rho}$ instead of $[\![\cdot]\!]_{\rho}$. ■

#### 4.1.3.2 Semantic Criteria

We consider three criteria *operational correspondence* and *divergence reflection* and *success sensitiveness.*

##### 4.1.3.2.1 Operational Correspondence

In this section we shall prove that operational correspondence (completeness and soundness) holds for the translation $[\![\cdot]\!]^{\circ}_{\rho}$.

As before, we are interested in precisely accounting for the number of computation steps induced by our translation. We need the following definition:

**Definition 4.1.4.** Let $P$ be a well-formed compensable process, then function $\mathtt{Z_d}(P)$ is defined as follows:

$$\mathtt{Z_d}(P) = \begin{cases} 0 & \text{if } \mathtt{pb}(P) = 0, \\ 1 & \text{if } \mathtt{pb}(P) > 0. \end{cases}$$

The number of reduction steps required for translating transaction scopes depends on the number of protected blocks in the default activity of that transaction. As already mentioned, if the transaction scope in the default activity contains at least one protected block then in the translation of such transaction there is an update location on name $z_t$ (it occurs in process $\mathtt{outd}^{\circ}$, see equation (4.1)); otherwise (if the number of protected blocks is zero) the number of reduction steps is the same as in the subjective case. This fact is presented by using the function $\mathtt{Z_d}(P)$ in the following theorem for operational correspondence.

Most of the lemmas, definitions, and theorems we have introduced to prove the operational correspondence for the translation with subjective update (cf. Theorem 3.2.7), can be adapted for the translation with objective update. Therefore, we will reuse the following statements for $[\![\cdot]\!]^{\circ}_{\rho}$, assuming the expected modifications:

- Definition 3.2.6 (Page 47) and Lemma 3.2.9, (Page 47), that are related with a mapping of evaluation contexts for $\mathcal{C}_{\mathtt{D}}$ into evaluation contexts of $\mathcal{S}$.
- Lemma 3.2.10, (Page 48) and Corollary 3.2.11, (Page 48), are the converse of Lemma 3.2.9.
- Lemma 3.2.14, (Page 51), shows that $\mathtt{ch}(t, [\![P]\!]^{\circ}_{\rho}) = \mathbf{0}$ for all $[\![P]\!]^{\circ}_{\rho}$ and use Lemma 3.2.13, (Page 51), for the proof.
- Lemma 3.2.15, (Page 51), identify processes that are created before a synchronization on $h_t$.

We first present an overview of the auxiliary results (and proofs) that are different from those presented in Paragraph 3.2.3.2.2 and Paragraph 3.2.3.2.3. The following definition formalizes all possible forms for the process $I^{(p)}_t([\![P]\!]^{\circ}_{t,\rho}, [\![Q]\!]^{\circ}_{\varepsilon})$.

**Definition 4.1.5.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I^{(p)}_t([\![P]\!]^{\circ}_{t,\rho}, [\![Q]\!]^{\circ}_{\varepsilon})$ (Figure 4.3) depending on $n = \mathtt{nl}(p_{t,\rho}, [\![P]\!]^{\circ}_{t,\rho})$:

1. if $n = 0$ then $p \in \{1, 2, 3\}$;
2. otherwise, if $n > 0$ then $[\![P]\!]^{\circ}_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid S$ and $p \in \{1, \ldots, 4+n\}$.

The following lemma formalizes all possible forms for the process $O^{(q)}_u([\![F]\!]^{\circ}_{\rho''}[h_u.[\![P_u]\!]^{\circ}_{\rho'''}], [\![Q'_u]\!]_{\varepsilon})$.

**Definition 4.1.6.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O^{(q)}_u([\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}], [\![Q]\!]^{\circ}_{\varepsilon})$ (Figure 4.4) depending on $m = \mathtt{nl}(p_{u,\rho}, [\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}])$:

1. for $n = 0$ we have $q \in \{1, 2, 3, 4\}$, and
2. for $n > 0$ and $[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid S$ we have $q \in \{1, \ldots, n+5\}$.

The following lemmas, which we established for the translation with subjective update $[\![\cdot]\!]_{\rho}$, hold also for translation with objective update $[\![\cdot]\!]^{\circ}_{\rho}$; the difference is that they use Definition 4.1.5 and Definition 4.1.6 instead of Definition 3.2.7 and Definition 3.2.8, respectively:

| $(p)$ | $I_t^{(p)}([\![P]\!]^\circ_{t,\rho}, [\![Q]\!]^\circ_\varepsilon)$ for $\mathtt{nl}(p_{t,\rho}, [\![P]\!]^\circ_{t,\rho}) = 0$ |
|---|---|
| (1) | $t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid \mathtt{extrd}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ <br> $\equiv t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid t\{\dagger\}.\overline{h_t}\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |
| (2) | $t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid t\{\dagger\}.\overline{h_t} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |
| (3) | $\overline{h_t} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |

| $(p)$ | $I_t^{(p)}([\![P]\!]^\circ_{t,\rho}, [\![Q]\!]^\circ_\varepsilon)$ for $\mathtt{nl}(p_{t,\rho}, [\![P]\!]^\circ_{t,\rho}) > 0$ |
|---|---|
| (1) | $t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid \mathtt{extrd}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ <br> $\equiv t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y)$ <br> $\mid \mathtt{outd}^\circ(p_{t,\rho}\,, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\{\dagger\}.\overline{h_t})\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |
| (2) | $t\big[[\![P]\!]^\circ_{t,\rho}\big] \mid p_{t,\rho}\{(X_1,\ldots,X_m).z_t\{(Z).(\prod_{k=1}^{m} p_\rho[X_k]$ <br> $\mid t\{\dagger\}.\overline{h_t})\}\}.z_t[\mathbf{0}] \mid p_\rho[[\![Q_t]\!]^\circ_\varepsilon]$ |
| $(2+j)$ <br> $1 \leq j \leq m-1$ | $t\big[[\![P]\!]^\circ_{t,\rho} \mid p_{t,\rho}\{(X_1,\ldots,X_{m-j}).z_t\{(Z).(\prod_{k=1}^{m-j} p_\rho[X_k] \mid t\{\dagger\}.\overline{h_t})$ <br> $\mid \prod_{k=1}^{j} p_\rho[[\![P'_k]\!]^\circ_\varepsilon]\}\}\big] \mid z_t[\mathbf{0}] \mid p_\rho[[\![Q_t]\!]^\circ_\varepsilon]$ |
| $(2+m)$ | $t\big[[\![P']\!]^\circ_{t,\rho} \mid z_t\{(Z).\prod_{k=1}^{m} p_\rho[[\![P'_k]\!]^\circ_\varepsilon] \mid t\{\dagger\}.\overline{h_t}\}\big] \mid z_t[\mathbf{0}] \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |
| $(3+m)$ | $t\big[[\![P']\!]^\circ_{t,\rho}\big] \mid \prod_{k=1}^{m} p_\rho[[\![P'_k]\!]^\circ_\varepsilon] \mid t\{\dagger\}.\overline{h_t} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |
| $(4+m)$ | $\prod_{k=1}^{m} p_\rho[[\![P'_k]\!]^\circ_\varepsilon] \mid \overline{h_t} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]$ |

Figure 4.3: Process $I_t^{(p)}([\![P]\!]^\circ_{t,\rho}, [\![Q]\!]^\circ_\varepsilon)$ with $p \geq 1$.

- Lemma 3.2.19,(Page 57)), is about the shape of process $R$ in $[\![P]\!]_\varepsilon \longrightarrow^n R$, and also ensures that there is a process $P'$ with an appropriate shape. The proof proceeds by induction on $n$.
- Lemma 3.2.12,(Page 49), is used as the base case in the proof of Lemma 3.2.19.
- Lemma 3.2.16 (Page 51) and Lemma 3.2.17 (Page 56) are used in the inductive step of the proof of Lemma 3.2.19.
- Lemma 3.2.20,(Page 62), ensures that the adaptable process obtained thanks to Lemma 3.2.16 and Lemma 3.2.17 can evolve until reaching a process that corresponds to the translation of a compensable process.

For the proof of operational correspondence we need the following statement:

**Lemma 4.1.5.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $[\![P]\!]^\circ_\rho \equiv [\![Q]\!]^\circ_\rho$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ∎

| $(q)$ | $O_u^{(q)}([\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}], [\![Q]\!]^{\circ}_{\varepsilon})$, $\mathtt{nl}(p_{u,\rho}, [\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]) = 0$ |
|---|---|
| $(1)$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid \mathtt{extrd}\{u, p_{u,\rho}, p_{\rho}\} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$<br>$\equiv u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid u\{\dagger\}.\overline{h_u}\} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(2)$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid h_u \mid u\{\dagger\}.\overline{h_u} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(3)$ | $h_u \mid \overline{h_u} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(4)$ | $p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |

| $(q)$ | $O_u^{(q)}([\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}], [\![Q]\!]^{\circ}_{\varepsilon})$, $\mathtt{nl}(p_{u,\rho}, [\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]) > 0$ |
|---|---|
| $(1)$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid \mathtt{extrd}\{u, p_{u,\rho}, p_{\rho}\} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$<br>$\equiv u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y)$<br>$\mid \mathtt{outd}^{\circ}(p_{u,\rho}\,, p_{\rho}, \mathtt{nl}(p_{u,\rho}, Y), u\{\dagger\}.\overline{h_u})\} \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(2)$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\{(X_1, \ldots, X_n).z_t\{(Z).(\prod_{k=1}^{n} p_{\rho}[X_k]$<br>$\mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u})\}\}.z_t[\mathbf{0}] \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(2+j)$<br>$1 \le j \le n-1$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}] \mid p_{u,\rho}\{(X_1, \ldots, X_{n-j}).z_t\{(Z).(\prod_{k=1}^{m-j} p_{\rho}[X_k]$<br>$\mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u}) \mid \prod_{k=1}^{j} p_{\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}]\}\}\big] \mid h_u \mid z_t[\mathbf{0}] \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(2+n)$ | $u\big[[\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'} \mid z_t\{(Z).(\prod_{k=1}^{n-j} p_{\rho}[X_k] \mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u})\}\big] \mid h_u \mid z_t[\mathbf{0}]$<br>$\mid \prod_{k=1}^{j} p_{\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |
| $(3+n)$ | $u\big[[\![F']\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}]\big] \mid \prod_{k=1}^{n} p_{\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid h_u \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}] \mid u\{\dagger\}.\overline{h_u}$ |
| $(4+n)$ | $\prod_{k=1}^{n} p_{\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid h_u \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}] \mid \overline{h_u}$ |
| $(5+n)$ | $\prod_{k=1}^{n} p_{\rho}[[\![P'_k]\!]^{\circ}_{\varepsilon}] \mid p_{\rho}[[\![Q]\!]^{\circ}_{\varepsilon}]$ |

Figure 4.4: Process $O_u^{(q)}([\![F]\!]^{\circ}_{\rho}[h_u.[\![P]\!]^{\circ}_{\rho'}], [\![Q]\!]^{\circ}_{\varepsilon})$ with $q \ge 1$.

In the following we prove that operational correspondence hols for translation $[\![\cdot]\!]^{\circ}_{\rho}$.

**Theorem 4.1.6** (Operational Correspondence for $[\![\cdot]\!]^{\circ}_{\varepsilon}$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{D}}$.

(1) If $P \longrightarrow P'$ then $[\![P]\!]^{\circ}_{\varepsilon} \longrightarrow^{k} [\![P']\!]^{\circ}_{\varepsilon}$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 4 + \mathtt{pb}_{\mathtt{D}}(P_1) + \mathtt{Z}_{\mathtt{d}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q\rangle]$, it follows $k = 4 + \mathtt{pb}_{\mathtt{D}}(F[P_1]) + \mathtt{Z}_{\mathtt{d}}(F[P_1])$.

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

(2) If $[\![P]\!]^{\circ}_{\varepsilon} \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow P'$ and $R \longrightarrow [\![P']\!]^{\circ}_{\varepsilon}$.

*Proof.* We consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We have three base cases, corresponding to cases $a), b)$ and $c)$ of Proposition 2.2.3 (Page 18). Also, we prove all cases by using Lemma 4.1.5, Definition 4.1.2, and Lemma 3.2.9.

a) This case corresponds to an input-output synchronization, such that $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$. The derivation that corresponds to this case is as the derivation presented in **Part (1) – Completeness** case ($a$) for translation with subjective update (cf. Derivation 3.30). Therefore, the thesis holds with $k = 1$.

b) This case corresponds to a synchronization due to an external error notification for a transaction scope. Therefore, for this case we consider that $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We will consider two sub-cases depending on whether process $P_1$ contains or not protected blocks. Below, we will use that $n = \mathtt{pb_D}(P_1)$.

(i) In this sub-case $n = 0$. Therefore, we have the following derivation:

$$
\begin{aligned}
[\![P]\!]^{\circ}_{\varepsilon} &\equiv [\![E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]]\!]^{\circ}_{\varepsilon}\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C[t[P_1,Q]]]\!]^{\circ}_{\rho} \mid [\![D[\overline{t}.P_2]]\!]^{\circ}_{\rho}\Big]\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[[\![t[P_1,Q]]\!]^{\circ}_{\rho'}] \mid [\![D]\!]^{\circ}_{\rho}[[\![\overline{t}.P_2]\!]^{\circ}_{\rho''}]\Big]\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[t\big[[\![P_1]\!]^{\circ}_{t,\rho'}\big] \mid t.\,\big(\mathtt{extrd}\{t, p_{t,\rho'}, p_{\rho'}\} \mid p_{\rho'}[[\![Q]\!]^{\circ}_{\varepsilon}]\big)] \mid [\![D]\!]^{\circ}_{\rho}[\overline{t}.h_t.[\![P_2]\!]^{\circ}_{\rho''}]\Big]\\
&\longrightarrow [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}\Big[I^{(1)}_t([\![P_1]\!]^{\circ}_{t,\rho'}, [\![Q]\!]^{\circ}_{\varepsilon})\Big] \mid [\![D]\!]^{\circ}_{\rho}[h_t.[\![P_2]\!]^{\circ}_{\rho''}]\Big]\\
&\longrightarrow^2 [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}\Big[I^{(3)}_t([\![P_1]\!]^{\circ}_{t,\rho'}, [\![Q]\!]^{\circ}_{\varepsilon})\Big] \mid [\![D]\!]^{\circ}_{\rho}\big[h_t.[\![P_2]\!]^{\circ}_{\rho''}\big]\Big]\\
&\longrightarrow [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[[\![\langle Q\rangle]\!]^{\circ}_{\rho'}] \mid [\![D]\!]^{\circ}_{\rho}[[\![P_2]\!]^{\circ}_{\rho''}]\Big]\\
&= [\![E]\!]^{\circ}_{\varepsilon}[[\![C[\langle Q\rangle]]\!]^{\circ}_{\rho} \mid [\![D[P_2]]\!]^{\circ}_{\rho}]\\
&= [\![E[C[\langle Q\rangle] \mid D[P_2]]]\!]^{\circ}_{\varepsilon}\\
&\equiv [\![P']\!]^{\circ}_{\varepsilon}
\end{aligned}
$$

Thus, the number of reduction steps is $k = 4$. Notice that here $\longrightarrow^2$ tells us that there have been two reduction steps: the first one is an update on location name $t$; the second reduction step "kills" with $t\{\dagger\}$ both the location $t$ and the process it hosts.

(ii) In this sub-case we consider $n > 0$, i.e., this is when there is at least one protected block in the default activity $P_1$. We have the following derivation:

$$
\begin{aligned}
[\![P]\!]^{\circ}_{\varepsilon} &\equiv [\![E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]]\!]^{\circ}_{\varepsilon}\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C[t[P_1,Q]]]\!]^{\circ}_{\rho} \mid [\![D[\overline{t}.P_2]]\!]^{\circ}_{\rho}\Big]\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[[\![t[P_1,Q]]\!]^{\circ}_{\rho'}] \mid [\![D]\!]^{\circ}_{\rho}[[\![\overline{t}.P_2]\!]^{\circ}_{\rho''}]\Big]\\
&= [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[t\big[[\![P_1]\!]^{\circ}_{t,\rho'}\big] \mid t.\,\big(\mathtt{extrd}\{t, p_{t,\rho'}, p_{\rho'}\} \mid p_{\rho'}[[\![Q]\!]^{\circ}_{\varepsilon}]\big)] \mid [\![D]\!]^{\circ}_{\rho}[\overline{t}.h_t.[\![P_2]\!]^{\circ}_{\rho''}]\Big]\\
&\longrightarrow [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}[I^{(1)}_t([\![P_1]\!]^{\circ}_{t,\rho'}, [\![Q]\!]^{\circ}_{\varepsilon})] \mid [\![D]\!]^{\circ}_{\rho}[h_t.[\![P_2]\!]^{\circ}_{\rho''}]\Big]\\
&\longrightarrow^{n+3} [\![E]\!]^{\circ}_{\varepsilon}\Big[[\![C]\!]^{\circ}_{\rho}\Big[I^{(n+4)}_t([\![P_1]\!]^{\circ}_{t,\rho'}, [\![Q]\!]^{\circ}_{\varepsilon})\Big] \mid [\![D]\!]^{\circ}_{\rho}\big[h_t.[\![P_2]\!]^{\circ}_{\rho''}\big]\Big]
\end{aligned}
$$

$$\begin{aligned}
&\longrightarrow [\![E]\!]^\circ_\varepsilon\Big[[\![C]\!]^\circ_\rho\Big[I_t^{(n+5)}([\![P_1]\!]^\circ_{t,\rho'}, [\![Q]\!]^\circ_\varepsilon)\Big] \mid [\![D]\!]^\circ_\rho[h_t.[\![P_2]\!]^\circ_{\rho''}]\Big]\\
&\equiv [\![E]\!]^\circ_\varepsilon\Big[[\![C]\!]^\circ_\rho\Big[[\![\mathsf{extr_D}(P_1)]\!]^\circ_{\rho'} \mid p_{\rho'}[[\![Q]\!]^\circ_\varepsilon]\Big] \mid [\![D]\!]^\circ_\rho[[\![P_2]\!]^\circ_{\rho''}]\Big]\\
&= [\![E]\!]^\circ_\varepsilon[[\![C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle]]\!]^\circ_\rho \mid [\![D[P_2]]\!]^\circ_\rho]\\
&= [\![E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]]\!]^\circ_\varepsilon\\
&\equiv [\![P']\!]^\circ_\varepsilon
\end{aligned}$$

Therefore, $k = 4 + n + \mathsf{Z_d}(P_1) = 5 + n$, where:

- 4 steps are as described in Theorem 3.2.7 and under a semantics with objective update, after $n$ updates, processes located at $p_{t,\rho'}$ will stay at location $t$, and
- $\mathsf{Z_d}(P_1)$ gives 1 more step; to avoid leaving such processes in the wrong location, the translation in [16] use an (objective) update on auxiliary location $z_t$, so to take them out of $t$ once $m$ updates on $p_{t,\rho'}$ have been executed. This additional synchronization step on name $z_t$ is the key to the efficiency gains when moving from objective to subjective updates (cf. Definition 4.1.4, Page 109).

c) In this case we consider that error notification arrives from the default activity of transaction; the error notification is internal. Again, according to Proposition 2.2.3 we consider the following case. Let $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_D}(F[P_1]) \mid \langle Q\rangle]$. Letting $n = \mathtt{pb}(F[P_1])$, we consider two cases $n = 0$ and the second is when $n > 0$:

(i) If $n = 0$ then there is the following derivation:

$$\begin{aligned}
[\![P]\!]^\circ_\varepsilon &\equiv [\![C[u[F[\overline{u}.P_1],Q]]]\!]^\circ_\varepsilon\\
&= [\![C]\!]^\circ_\varepsilon[[\![u[F[\overline{u}.P_1],Q]]\!]^\circ_\rho]\\
&= [\![C]\!]^\circ_\varepsilon[u\big[[\![F[\overline{u}.P_1]]\!]^\circ_{u,\rho}\big] \mid u.\,(\mathtt{extrd}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon])]\\
&= [\![C]\!]^\circ_\varepsilon[u\big[[\![F]\!]^\circ_{u,\rho}[\overline{u}.h_u.[\![P_1]\!]^\circ_{\rho'}]\big] \mid u.(\mathtt{extrd}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon])]\\
&\longrightarrow [\![C]\!]^\circ_\varepsilon\Big[O_u^{(1)}([\![F]\!]^\circ_\rho[h_u.[\![P_1]\!]^\circ_{\rho'}], [\![Q]\!]^\circ_\varepsilon)\Big]\\
&\longrightarrow^3 [\![C]\!]^\circ_\varepsilon\big[O_u^{(4)}([\![F]\!]^\circ_\rho[h_u.[\![P_1]\!]^\circ_{\rho'}], [\![Q]\!]^\circ_\varepsilon)\big]\\
&\equiv [\![P']\!]^\circ_\varepsilon
\end{aligned}$$

Therefore, the number of reduction steps is $k = 4$.

(ii) If $n > 0$ then there is the following derivation:

$$\begin{aligned}
[\![P]\!]^\circ_\varepsilon &\equiv [\![C[u[F[\overline{u}.P_1],Q]]]\!]^\circ_\varepsilon\\
&= [\![C]\!]^\circ_\varepsilon[[\![u[F[\overline{u}.P_1],Q]]\!]^\circ_\rho]\\
&= [\![C]\!]^\circ_\varepsilon[u\big[[\![F[\overline{u}.P_1]]\!]^\circ_{u,\rho}\big] \mid u.\,(\mathtt{extrd}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon])]\\
&= [\![C]\!]^\circ_\varepsilon[u\big[[\![F]\!]^\circ_{u,\rho}[\overline{u}.h_u.[\![P_1]\!]^\circ_{\rho'}]\big] \mid u.(\mathtt{extrd}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[[\![Q]\!]^\circ_\varepsilon])]\\
&\longrightarrow [\![C]\!]^\circ_\varepsilon\Big[O_u^{(1)}([\![F]\!]^\circ_\rho[h_u.[\![P_1]\!]^\circ_{\rho'}], [\![Q]\!]^\circ_\varepsilon)\Big]\\
&\longrightarrow^{n+3} [\![C]\!]^\circ_\varepsilon\Big[O_u^{(n+4)}([\![F]\!]^\circ_\rho[h_u.[\![P_1]\!]^\circ_{\rho'}], [\![Q]\!]^\circ_\varepsilon)\\
&\longrightarrow [\![C]\!]^\circ_\varepsilon\Big[O_u^{(n+5)}([\![F]\!]^\circ_\rho[h_u.[\![P_1]\!]^\circ_{\rho'}], [\![Q]\!]^\circ_\varepsilon)\Big]\\
&= [\![C]\!]^\circ_\varepsilon\Big[[\![\mathsf{extr_D}(F[P_1])]\!]^\circ_\rho \mid p_\rho[[\![Q]\!]^\circ_\varepsilon]\Big]\\
&= [\![C[\mathsf{extr_D}(F[P_1]) \mid \langle Q\rangle]\!]^\circ_\varepsilon\\
&\equiv [\![P']\!]^\circ_\varepsilon
\end{aligned}$$

Therefore, the number of reduction steps is $k = 4 + n + \mathsf{Z_d}(F[P_1]) = 5 + n$.

(2) **Part (2) – Soundness:** The proof for soundness follows the approach described in detail for encoding with subjective update (cf. page 65). Therefore, Given $[\![P]\!]^\circ_\varepsilon \longrightarrow^n R$, by Lemma 3.2.19 (which also applies to $[\![\cdot]\!]^\circ_\rho$), process $R$ has the following form:

$$R \equiv \prod_{w=1}^{z} [\![E_w]\!]^\circ_\varepsilon\Big[\prod_{k=1}^{s_w}[\![G_{k,w}]\!]^\circ_{\rho_w}\big[\prod_{i=1}^{l_k}[\![C_{i,k,w}]\!]^\circ_{\rho'_{k,w}}[I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k}[\![D_{j,k,w}]\!]^\circ_{\rho'_{k,w}}[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]^\circ_{\rho''_{k,w}}] \mid \prod_{c=1}^{m_k}[\![L_{c,k,w}]\!]^\circ_{\rho'_{k,w}}[O^{(q)}_{u_{c,k,w}}]\big]\Big],$$

where $I^{(p)}_{t_{i,k,w}}$ and $O^{(q)}_{u_{c,k,w}}$ are processes from Figure 4.3 and Figure 4.4, respectively.

Also by Lemma 3.2.19, we have $P \longrightarrow^* P''$ where

$$P'' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}[t_{i,k,w}[P_{t_{i,k,w}}, Q_{t_{i,k,w}}]] \mid \prod_{j=1}^{r_k} D_{j,k,w}[\overline{t_{j,k,w}}.S_{t_{j,k,w}}] \mid \prod_{c=1}^{m_k} L_{c,k,w}[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}], Q_{u_{c,k,w}}]]\big]\Big],$$

where by successive application of completeness it follows that $[\![P]\!]^\circ_\varepsilon \longrightarrow^* [\![P'']\!]^\circ_\varepsilon$.

By Lemma 3.2.20 (which also applies to $[\![\cdot]\!]^\circ_\rho$), i.e., by $l_k$ successive applications of (3.28) and $m_k$ successive applications of (3.29) on process $R$, it follows that:

$$\begin{aligned}
R \longrightarrow^* & \prod_{w=1}^{z} [\![E_w]\!]^\circ_\varepsilon\Big[\prod_{k=1}^{s_w}[\![G_{k,w}]\!]^\circ_{\rho_w}\big[\prod_{i=1}^{l_k}[\![C_{i,k,w}]\!]^\circ_{\rho'_{k,w}}[[\![\mathsf{extr}_\mathsf{D}(P'_{t_{i,k,w}})]\!]^\circ_{\rho''_{k,w}} \mid [\![\langle Q'_{t_{i,k,w}}\rangle]\!]^\circ_{\rho''_{k,w}}] \\
& \mid \prod_{j=1}^{r_k}[\![D_{j,k,w}]\!]^\circ_{\rho'_{k,w}}[[\![S_{t_{j,k,w}}]\!]^\circ_{\rho''_{k,w}}] \mid \prod_{c=1}^{m_k}[\![L_{c,k,w}]\!]^\circ_{\rho'_{k,w}}[[\![\mathsf{extr}_\mathsf{D}(F_{c,k,w}[P_{u_{c,k,w}}])]\!]^\circ_{\rho''_{k,w}} \\
& \mid [\![\langle Q'_{u_{c,k,w}}\rangle]\!]^\circ_{\rho''_{k,w}}]\big]\Big] \\
\equiv & [\![\prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}[\mathsf{extr}_\mathsf{D}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle] \mid \prod_{j=1}^{r_k} D_{j,k,w}[S_{t_{j,k,w}}] \\
& \mid \prod_{c=1}^{m_k} L_{c,k,w}[\mathsf{extr}_\mathsf{D}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle]\big]\Big]]\!]^\circ_\varepsilon \\
\equiv & [\![P']\!]^\circ_\varepsilon.
\end{aligned}$$

Therefore, it follows that

$$P' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}[\mathsf{extr}_\mathsf{D}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}}\rangle] \mid \prod_{j=1}^{r_k} D_{j,k,w}[S_{t_{j,k,w}}] \mid \prod_{c=1}^{m_k} L_{c,k,w}[\mathsf{extr}_\mathsf{D}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}}\rangle]\big]\Big].$$

Also, by Proposition 2.2.3, i.e., by $l_k$ successive applications of case b) and and $m_k$ successive applications of case c) on process $P''$, it follows that: $P'' \longrightarrow^* P'$.

By successive application of (1) — **Completeness** on the derivation $P'' \longrightarrow^* P'$ it follows that $[\![P'']\!]^\circ_\varepsilon \longrightarrow^* [\![P']\!]^\circ_\varepsilon$.

■

The following example illustrates the operational correspondence property:

**Example 4.1.7.** Let $P$ be a process as in Example 3.2.23 (Page 67). Expanding Definition 4.1.2 the following holds:

$$
\begin{aligned}
[\![P]\!]^{\circ}_{\varepsilon} &= s\Big[t\big[p_{t,s}[a] \mid p_{t,s}[b \mid c] \mid t.\big(\mathtt{extrd}\{t, p_{t,s}, p_s\} \mid p_s[d]\big)\big]\Big] \\
&\quad \mid s.\mathtt{extrd}\{s, p_s, p_\varepsilon\} \mid \overline{t}.h_t.\overline{s}.h_s \\
&= s\Big[t\big[p_{t,s}[a] \mid p_{t,s}[b] \mid c\big] \mid t.\big(t\{(Y).t[Y] \mid \mathsf{ch}(t, Y) \\
&\quad \mid \mathtt{outd}^{\circ}(p_{t,s}\,,\, p_s, \mathtt{nl}(p_{t,s}, Y), t\{\dagger\}.\overline{h_t})\} \mid p_s[d]\big)\Big] \\
&\quad \mid s.\mathtt{extrd}\{s, p_s, p_\varepsilon\} \mid \overline{t}.h_t.\overline{s}.h_s \\
&\longrightarrow^2 s\Big[t\big[p_{t,s}[a] \mid p_{t,s}[b] \mid c\big] \mid \mathtt{outd}^{\circ}(p_{t,s}\,,\, p_s, 2, t\{\dagger\}.\overline{h_t}) \mid p_s[d]\Big] \\
&\quad \mid s.\mathtt{extrd}\{s, p_s, p_\varepsilon\} \mid h_t.\overline{s}.h_s \\
&\longrightarrow^2 s\Big[t\big[z_t\{(Z).p_s[a] \mid p_s[b] \mid t\{\dagger\}.h_t\}\big] \mid z_t[\mathbf{0}] \mid p_s[d]\Big] \\
&\quad \mid s.\mathtt{extrd}\{s, p_s, p_\varepsilon\} \mid \overline{h_t}.\overline{s}.h_s \\
&\longrightarrow^3 s\Big[p_s[a] \mid p_s[b] \mid p_s[d]\Big] \mid s.\mathtt{extrd}\{s, p_s, p_\varepsilon\} \mid \overline{s}.h_s \\
&\longrightarrow^8 p_\varepsilon[a] \mid p_\varepsilon[b] \mid p_\varepsilon[d] \\
&= [\![\langle a\rangle \mid \langle b\rangle \mid \langle d\rangle]\!]^{\circ}_{\varepsilon}.
\end{aligned}
$$

We have $[\![P]\!]^{\circ}_{\varepsilon} \longrightarrow^k [\![P_2]\!]^{\circ}_{\varepsilon}$ with $k = 15$:

$$
\begin{aligned}
k &= \underbrace{4 + \mathtt{pb}_{\mathtt{D}}(P_1) + \mathtt{Z}_{\mathtt{d}}(t[\langle a\rangle \mid \langle b\rangle \mid c, d])}_{\text{for transaction } t} + \underbrace{4 + \mathtt{pb}_{\mathtt{D}}(P_2) + \mathtt{Z}_{\mathtt{d}}(s[\langle a\rangle \mid \langle b\rangle \mid \langle d\rangle, \mathbf{0}])}_{\text{for transaction } s} \\
&= 4 + 2 + 1 + 4 + 3 + 1 = 15.
\end{aligned}
$$

We briefly analyze the reduction steps that are related to the translation of transactions on name $t$ and $s$:

- For location $t$ there are 7 reduction steps: synchronization on $t$ and $\overline{t}$, updating location $t$, two steps as relocation of process on location $p_{t,s}$ by process $\mathtt{outd}^{\circ}$ to location $p_s$, update on location $z_t$, erasing location $t$ using $t\{\dagger\}$ and synchronization $h_t$ with corresponding output $\overline{h_t}$.
- For location $s$ there are 8 reduction steps, one more step than for location $t$; because now location $s$ contains three processes on location $p_s$ that has to be relocated on location $p_\varepsilon$ by process $\mathtt{outd}^{\circ}$.

**Example 4.1.8.** Recall process *Reservation* from the hotel booking scenario (cf. Example 2.2.1, Page 15). We use encoding with objective update:

$$
\begin{aligned}
[\![Reservation]\!]^{\circ}_{\varepsilon} = \;& t[book.pay.invoice] \mid t.(\mathtt{extrd}\{t, p_t, p_\varepsilon\} \mid p_\varepsilon[\overline{refund}]) \mid \overline{book}.\overline{pay}.\overline{t}.h_t.refund \\
\longrightarrow^3 \;& t[invoice] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t, Y) \\
& \mid \mathtt{outd}^{\circ}(p_t\,,\, p_\varepsilon, \mathtt{nl}(p_t, Y), t\{\dagger\}.\overline{h_t})\} \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\longrightarrow \;& t[invoice] \mid \mathsf{ch}(t, invoice) \mid \mathtt{outd}^{\circ}(p_t\,,\, p_\varepsilon, \mathtt{nl}(p_t, invoice), t\{\dagger\}.\overline{h_t}) \\
& \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
\equiv \;& t[invoice] \mid \mathtt{outd}^{\circ}(p_t\,,\, p_\varepsilon, 0, t\{\dagger\}.\overline{h_t}) \mid p_\varepsilon[\overline{refund}] \mid h_t.refund \\
= \;& t[invoice] \mid t\{\dagger\}.\overline{h_t} \mid p_\varepsilon[\overline{refund}] \mid h_t.refund
\end{aligned}
$$

$$\longrightarrow^3 \quad p_\varepsilon[\mathbf{0}]$$

Therefore, we get $[\![Reservation]\!]^\circ_\varepsilon \longrightarrow^7 p_\varepsilon[\mathbf{0}]$ and so the number and justification for the obtained reduction steps is the same as in Example 3.2.24. This is because the transaction $t$ does not contain protected blocks. In turn, in encodings, this means that there are no differences between processes $\mathtt{outd}^{\mathsf{s}}$ and $\mathtt{outd}^{\mathsf{o}}$, i.e., they are equal to some process $Q$. In this example, $Q = t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t}$.

#### 4.1.3.2.2 Divergence Reflection

We are going to prove that the encoding does not introduce divergent computations. We need Definition 3.2.9, also we need to revise Lemma 3.2.25 as in the following.

**Lemma 4.1.9.** Let $\{R_i\}_{i\geq 0}$ be a sequence of adaptable processes such that $R_i \longrightarrow R_{i+1}$, with $R_0 = [\![P_0]\!]^\circ_\rho$, for some compensable process $P_0$ and path $\rho$. Then for every $i \geq 1$ there is $P_i$ such that:

(i) $R_i \longrightarrow^* [\![P_i]\!]^\circ_\rho$,

(ii) $P_{i-1} = P_i$ or $P_{i-1} \longrightarrow P_i$, and

(iii) $R_i \not\equiv R_{i+1} \not\equiv \ldots \not\equiv R_{i+m}$ and $P_i = P_{i+1} = \ldots = P_{i+m}$ imply $m \leq 5 + \mathtt{npb}(P_0)$.

*Proof.* The proof follows the same idea as a proof of Lemma 3.2.25. Therefore, the proof for (i) and (ii) proceeds by induction on $i$. The proof for (iii) is obtain from (the proof of) Lemma 3.2.19 which holds also for $[\![\cdot]\!]^\circ_\rho$. ■

**Theorem 4.1.10** (Divergence Reflection for $[\![\cdot]\!]^\circ_\rho$)**.** Let $\{R_i\}_{i\geq 0}$ be an infinite sequence of adaptable processes such that

$$(1)\ R_0 = [\![P_0]\!]^\circ_\rho \text{ for some } P_0 \text{ and } \rho, \qquad \text{and} \qquad (2)\ \ R_i \longrightarrow R_{i+1} \text{ for any } i \geq 0.$$

Then there is an infinite sequence of adaptable processes $\{P'_j\}_{j\geq 0}$ such that

$$(3)\ P'_0 = P_0, \qquad \text{and} \qquad (4)\ \ P'_j \longrightarrow P'_{j+1} \text{ for any } j \geq 0.$$

*Proof.* By Lemma 4.1.9, there is a sequence $\{P_i\}_{i\geq 0}$ such that

$$(\mathrm{i})\ R_i \longrightarrow^* [\![P_i]\!]^\circ_\rho \qquad \text{and} \qquad (\mathrm{ii})\ \ P_{i-1} = P_i \text{ or } P_{i-1} \longrightarrow P_i.$$

Consider now a sequence of compensable processes $P'_0, P'_1, P'_2, \ldots$ such that

(1) $P'_{j-1} \longrightarrow P'_j$, for any $j \geq 1$, and

(2) for every $i$ there is $j$ such that $P_i = P'_j$.

By Lemma 4.1.9, at most $5 + \mathtt{npb}(P_0)$ reduction steps from the sequence $\{R_i\}_{i\geq 0}$ correspond to one reduction step of $\{P'_j\}_{j\geq 0}$. Hence, the number of processes in $\{P'_j\}_{j\geq 0}$ is not less then the number of processes $\{R_i\}_{i\geq 0}$ divided by $5 + \mathtt{npb}(P_0)$. Since the sequence $\{R_i\}_{i\geq 0}$ is infinite, the same holds for $\{P'_j\}_{j\geq 0}$. ■

#### 4.1.3.2.3 Success Sensitiveness

**Theorem 4.1.11** (Success Sensitiveness for $[\![\cdot]\!]^\circ_\rho$)**.** Let $P$ be a well-formed compensable process and $\rho$ an arbitrary path. Then $P \Downarrow$ if and only if $[\![P]\!]^\circ_\rho \Downarrow$.

*Proof.* The proof proceeds similarly as the proof for Theorem 3.2.27. ■

### 4.1.4 Comparing Subjective vs Objective update

In this section we exploit the correctness properties of encodings to distinguish between subjective and objective updates in calculi for concurrency. We introduce an encodability criterion called *efficiency*. Since subjective updates induce tighter operational correspondences, we can formally declare that subjective updates are more suited to encode compensation handling than objective updates.

Having introduced encodings of compensable processes with discarding semantics into adaptable processes, here we compare their *efficiency*. We define efficiency in abstract terms, considering the number of reduction steps that a target language requires to mimic the behavior of a source language:

**Definition 4.1.7** (Efficient Encoding). Let $\mathcal{L}_i = (\mathcal{P}_i, \longrightarrow_i, \approx_i)$ (with $i \in \{1,2,3\}$) be calculi as in Definition 2.3.1. Suppose $[\![\cdot]\!]_1 : \mathcal{P}_1 \longrightarrow \mathcal{P}_2$ and $[\![\cdot]\!]_2 : \mathcal{P}_1 \longrightarrow \mathcal{P}_3$ are encodings as in Definition 2.3.5. We say that $[\![\cdot]\!]_1$ is *as or more efficient* than $[\![\cdot]\!]_2$ if for every process $P$ from $\mathcal{P}_1$ the following implication holds (with $k_1, k_2 > 0$):

$$\text{If } P \longrightarrow P' \text{ and } [\![P]\!]_2 \longrightarrow^{k_2} [\![P']\!]_2 \text{ then there is } k_1 \text{ such that } [\![P]\!]_1 \longrightarrow^{k_1} [\![P']\!]_1 \text{ and } k_1 \leq k_2.$$

The following statement is a corollary of Theorem 4.1.6, where $\mathtt{Z_d}(\cdot)$ is defined in Definition 4.1.4.

**Corollary 4.1.12.** Let $P$ be a well-formed process in $\mathcal{C}$. If $P \longrightarrow P'$ and $[\![P]\!]^{\circ}_{\varepsilon} \longrightarrow^{k} [\![P']\!]^{\circ}_{\varepsilon}$ then:

b) If $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ then $k \geq 4 + \mathtt{pb_D}(P_1) + \mathtt{Z_d}(P_1)$,

c) If $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q\rangle]$ then $k \geq 4 + \mathtt{pb_D}(F[P_1]) + \mathtt{Z_d}(F[P_1])$.

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

We then have the following theorem:

**Theorem 4.1.13.** The encoding $[\![\cdot]\!]_{\rho} : \mathcal{C}_{\mathtt{D}} \longrightarrow \mathcal{S}$ is as or more efficient than $[\![\cdot]\!]^{\circ}_{\rho} : \mathcal{C}_{\mathtt{D}} \longrightarrow \mathcal{O}$.

*Proof.* Let $P \longrightarrow P'$ and $[\![P]\!]^{\circ}_{\varepsilon} \longrightarrow^{k_2} [\![P']\!]^{\circ}_{\varepsilon}$. By Theorem 3.2.7, there is $k_1$ such that $[\![P]\!]_{\varepsilon} \longrightarrow^{k_1} [\![P']\!]_{\varepsilon}$ and

a) $k_1 = 1$ if $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$,

b) $k_1 = 4 + \mathtt{pb_D}(P_1)$ if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$,

c) $k_1 = 4 + \mathtt{pb_D}(F[P_1])$ if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q\rangle]$.

By Theorem 4.1.6, Proposition 2.2.3 and Corollary 4.1.12 we have the following three cases, for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$, processes $P_1, Q, P_2$ and names $t, u$:

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ and $k_2 \geq 1 = k_1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ and $k_2 \geq 4 + \mathtt{pb_D}(P_1) + \mathtt{Z_d}(P_1) \geq 4 + \mathtt{pb_D}(P_1) = k_1$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}(F[P_1]) \mid \langle Q\rangle]$ and $k_2 \geq 4 + \mathtt{pb_D}(F[P_1]) + \mathtt{Z_d}(F[P_1]) \geq 4 + \mathtt{pb_D}(F[P_1]) = k_1$.

Thus, in all three cases $k_1 \leq k_2$; by Definition 4.1.7 we conclude that $[\![\cdot]\!]_{\varepsilon}$ is as or more efficient than $[\![\cdot]\!]^{\circ}_{\varepsilon}$. ■

Let us dwell a bit on the content of the previous theorem, to understand better the differences between the two encodings (and between objective and subjective update). Recall that the main difference between encodings is in the auxiliary processes

$$\texttt{extrd}\langle\!\langle t, l_1, l_2\rangle\!\rangle \quad \text{and} \quad \texttt{extrd}\{t, l_1, l_2\}$$

which are used in the encoding of transaction scopes in the subjective and objective case, respectively. In turn, those auxiliary processes rely on processes $\texttt{outd}^{\mathsf{s}}(l_1, l_2, n, Q)$ and $\texttt{outd}^{\mathsf{o}}(t, l_1, l_2, n, Q)$ (cf. (3.2) and (4.1), respectively), which extract $n$ processes located at $l_1$ in $Q$ and relocate them to $l_2$.

A closer look at $\texttt{outd}^{\mathsf{s}}(l_1, l_2, n, Q)$ and $\texttt{outd}^{\mathsf{o}}(t, l_1, l_2, n, Q)$ reveals that they differ in the use of name $z_t$, which is used in the objective case (when $n > 0$) but not in the subjective case. The use of $z_t$ appears indispensable: under a semantics with objective update, after $n$ updates, the located processes will stay at the wrong location (i.e. $t$). To avoid this, we use $z_t$ as an auxiliary location. This auxiliary location enables us to move processes out of $t$ and to relocate them to their parent location.

This synchronization step on name $z_t$ is the key to the efficiency gains obtained when moving from objective to subjective updates—clearly, the improvement will be proportional to the number of compensation operations in the source process. Consider again Example 3.2.23 and Example 4.1.7, which translate process $P = s\big[t[\langle a\rangle \mid \langle b\rangle \mid c, d], \mathbf{0}\big] \mid \bar{t}.\bar{s}$ using subjective and objective updates, respectively. Here the subjective encoding outperforms the objective encoding by two reduction steps. In Example 4.1.7, these two steps correspond to two synchronisations on name $z_t$, which are not needed in Example 3.2.23.

Finally, as the proof of Theorem 4.1.13 makes explicit, the two encodings are equivalent, in terms of efficiency, when $n = 0$ in $\texttt{outd}^{\mathsf{s}}(l_1, l_2, n, Q)$ and $\texttt{outd}^{\mathsf{o}}(t, l_1, l_2, n, Q)$. This is because in this case we do not need to save any process from the default activity of the transaction scope—we need an equal number of reduction steps for achieving operational correspondence.

## 4.2 Translating $\mathcal{C}_{\mathrm{P}}$ into $\mathcal{O}$

In this subsection we concentrate on a source calculus with *static recovery* and *preserving semantics*. Initially, we introduce some useful conventions and intuitions for the encoding and then we give the formal presentations of the encoding.

### 4.2.1 The Translation, Informally

To encode transactions and their extraction function we use Definition 3.3.1 and Definition 3.3.2. Also, we need auxiliary process $\texttt{outp}^{\mathsf{o}}(t, P, l_1, l_1', l_2, l_2', n, m)$, which is similar to the process $\texttt{outp}^{\mathsf{s}}(t, P, l_1, l_1', l_2, l_2', n, m)$ (cf. 3.36) that we used in the encoding $(\!|\cdot|\!)_\rho$. Therefore, auxiliary process $\texttt{outp}^{\mathsf{o}}(t, P, l_1, l_1', l_2, l_2', n, m)$:

(i) moves $n$ processes from location $l_1$ to location $l_1'$;

(ii) moves $m$ processes from location $l_2$ to location $l_2'$.

Initially, for the definition of $\mathtt{outp}^{\mathsf{o}}(t, P, l_1, l'_1, l_2, l'_2, n, m)$ we introduce the following auxiliary processes:

$$\mathtt{outp}^{\mathsf{o}}_1(t, l_1, l'_1, n) = l_1\{(X_1, \ldots, X_n).z_t\{(Z).\left(\prod_{i=1}^{n} l'_1[X_i] \mid t\{\dagger\}.\overline{j_t}.r_t\right)\}\}.z_t[\mathbf{0}];$$

$$\mathtt{outp}^{\mathsf{o}}_2(t, t_1, \ldots, t_m, l_2, l'_2, m) = l_2\{(Y_1, \ldots, Y_m).$$
$$z_t\{(Z).\left(r_t.\left(\prod_{k=1}^{m} (l'_2[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l'_2\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}})\right) \mid t\{\dagger\}.\overline{j_t}\right)\}\}.z_t[\mathbf{0}];$$

$$\mathtt{outp}^{\mathsf{o}}_3(t, t_1, \ldots, t_m, l_1, l'_1, l_2, l'_2, n, m) = l_1\{(X_1, \ldots, X_n).l_2\{(Y_1, \ldots, Y_m).$$
$$z_t\{(Z).\left(\prod_{i=1}^{n} l'_1[X_i] \mid r_t.\left(\prod_{k=1}^{m} (l'_2[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l'_2\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}})\right) \mid t\{\dagger\}.\overline{j_t}\right)\}\}\}.z_t[\mathbf{0}]$$

The auxiliary process $\mathtt{outp}^{\mathsf{o}}(t, P, l_1, l'_1, l_2, l'_2, n, m)$ where $\mathtt{top}(l_2, P) = \{t_1, \ldots, t_m\}$ for $m > 0$ is now defined as follows:

$$\mathtt{outp}^{\mathsf{s}}(t, P, l_1, l'_1, l_2, l'_2, n, m) = \begin{cases} t\langle\langle\dagger\rangle\rangle.\overline{j_t}.r_t & \text{if } n, m = 0 \\ \mathtt{outp}^{\mathsf{o}}_1(t, l_1, l'_1, n) & \text{if } n > 0, m = 0 \\ \mathtt{outp}^{\mathsf{o}}_2(t, t_1, \ldots, t_m, l_2, l'_2, m) & \text{if } n = 0, m > 0 \\ \mathtt{outp}^{\mathsf{o}}_3(t, t_1, \ldots, t_m, l_1, l'_1, l_2, l'_2, n, m) & \text{if } n, m > 0 \end{cases} \tag{4.4}$$

We compare processes $\mathtt{outp}^{\mathsf{s}}$ (3.36) and $\mathtt{outp}^{\mathsf{o}}$ (4.4), because differences between them will reflect directly on the efficiency of encodings.

**Remark 4.2.1** (Comparing $\mathtt{outp}^{\mathsf{s}}$ and $\mathtt{outp}^{\mathsf{o}}$)**.** Differences appear when parameter $n > 0$ and/or $m > 0$. In these cases, it should be noted that in $\mathtt{outp}^{\mathsf{o}}$ (cf. (4.4), i.e., $\mathtt{outp}^{\mathsf{o}}_1$, $\mathtt{outp}^{\mathsf{o}}_2$, $\mathtt{outp}^{\mathsf{o}}_3$) appears processes that are enclosed inside an update prefix on name $z_t$, while in process $\mathtt{out}^{\mathsf{s}}$ (cf. (3.36)) this is not the case. In $\mathtt{outp}^{\mathsf{o}}$, once:

- $n$ updates on name $l_1$, in the case $n > 0, m = 0$, or
- $m$ updates on name $l_2$, in the case $n = 0, m > 0$, or
- $n + m$ updates on names $l_1, l_2$ in the case $n > 0, m > 0$,

have been executed, the resulting processes:

$$\prod_{i=1}^{n} l'_1[X_i] \mid t\{\dagger\}.\overline{j_t}.r_t \tag{4.5}$$

$$r_t.(\prod_{k=1}^{m} l'_2[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l'_2\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}) \mid t\{\dagger\}.\overline{j_t} \tag{4.6}$$

$$\prod_{i=1}^{n} l'_1[X_i] \mid r_t.(\prod_{k=1}^{m} l'_2[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l'_2\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}) \mid t\{\dagger\}.\overline{j_t} \tag{4.7}$$

will be enclosed in a (wrong) location (say, $t$). Such a location should be deleted, but before doing so process enclosed in update prefix on $z_t$, must be relocated. In eq. (4.4), this relocation is achieved via a synchronization on name $z_t$. In contrast, because process $\mathtt{outp}^{\mathsf{s}}$ uses subjective updates, process reconfiguration follows in the opposite direction. This ensures that, after $n$, or $m$, or $n + m$ updates, the processes given with (4.5, 4.6, 4.7) will remain in their original location, and so no relocation using $z_t$ is needed.

$$
\begin{aligned}
(\!|\langle P\rangle|\!)^{\circ}_{\rho} &= p_\rho\big[(\!|P|\!)^{\circ}_{\varepsilon}\big] \\
(\!|t[P,Q]|\!)^{\circ}_{\rho} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid t.\big(\mathtt{extrp}\{t,(\!|P|\!)^{\circ}_{t,\rho},p_{t,\rho},p_\rho,\beta_{t,\rho},\beta_\rho\} \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\big)\Big] \\
&\qquad \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t} \\
(\!|\overline{t}.P|\!)^{\circ}_{\rho} &= \overline{t}.h_t.(\!|P|\!)^{\circ}_{\rho}
\end{aligned}
$$

Figure 4.5: Translating $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$.

At this point, we are going to specify an important fact: the number $n$ of protected blocks and the number $m$ of nested transactions in the default activity of the transaction is directly related to the number of reduction steps induced by our translation. If $n = 0$ and $m = 0$ then the number of reduction steps will be the same for subjective and objective updates; otherwise, if $n > 0, m = 0$, or $n = 0, m > 0$, or $n, m > 0$ the translation with subjective update will exhibit less reduction steps than the translation with objective update. Therefore, translation with the subjective update will be more efficient than the translation with the objective update (cf. Definition 4.1.7). We will prove this statement in Section 4.2.4.

### 4.2.2 The Translation, Formally

The function for determining the number of locations $\mathtt{nl}(\cdot,\cdot)$ in an adaptable process and the function $\mathsf{ch}(t,\cdot)$ are as introduced in Definition 3.2.1. Also we use Definition 3.3.4 to count the number of nested transactions.

We now define process $\mathtt{extrp}\{t, P, l_1, l'_1, l_2, l'_2\}$:

**Definition 4.2.1** (Update Prefix for Extraction)**.** Let $t$, $l_1, l'_1, l_2, l'_2$ be names and $P$ is an adaptable process. We write $\mathtt{extrp}\{t, P, l_1, l'_1, l_2, l'_2\}$ to stand for the following (objective) update prefix:

$$
\mathtt{extrp}\{t,P,l_1,l'_1,l_2,l'_2\} \ = t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outp}^{\circ}(t,P,l_1,l'_1,l_2,l'_2,\mathtt{nl}(l_1,Y),\mathtt{nl}(l_2,Y))\} \tag{4.8}
$$

Now, we may formally define $(\!|\cdot|\!)^{\circ}_{\rho}$, and the intuition for the translation of $t[P,Q]$ is as in the case of subjective update.

**Definition 4.2.2** (Translating $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with preserving semantics into (objective) adaptable processes as a tuple $((\!|\cdot|\!)^{\circ}_{\rho}, \varphi_{(\!|\cdot|\!)^{\circ}_{\rho}})$ where:

(a) The renaming policy $\varphi_{(\!|\cdot|\!)^{\circ}_{\rho}} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$
\varphi_{(\!|\cdot|\!)^{\circ}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, j_x, r_x\} \cup \{p_\rho, \beta_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{4.9}
$$

(b) The translation $(\!|\cdot|\!)^{\circ}_{\rho} : \mathcal{C}_{\mathtt{P}} \to \mathcal{O}$ is as in Figure 4.5 and as a homomorphism for other operators.

In the upcoming section, we prove the correctness of the translation for $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$.

### 4.2.3 Translation Correctness

To this end, we address the two criteria in Definition 2.3.5: *name invariance* and *operational correspondence*. We will investigate the other criteria as a part of future work. Our results apply for well-formed processes as in Definition 2.2.4.

#### 4.2.3.1 Semantic Criteria - Name invariance

We prove name invariance with respect to the renaming policy in Definition 3.2.3 Case (b).

**Theorem 4.2.2** (Name invariance for $(\!|\cdot|\!)^{\circ}_{\rho}$)**.** For every *well-formed* compensable processes $P$ and substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c : \varphi_{(\!|\cdot|\!)^{\circ}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)^{\circ}_{\rho}}(x)\}$, and

(ii) $(\!|\sigma(P)|\!)^{\circ}_{\sigma(\rho)} = \sigma'((\!|P|\!)^{\circ}_{\rho})$

*Proof.* The proof proceeds in the same way as the proof of Theorem 3.2.4 by using $(\!|\cdot|\!)^{\circ}_{\rho}$ instead of $[\![\cdot]\!]_{\rho}$. ■

#### 4.2.3.2 Semantic Criteria - Operational Correspondence

In this section we shall prove that *operational correspondence* (completeness and soundness) holds for the translation $(\!|\cdot|\!)^{\circ}_{\rho}$. We need the following auxiliary definition.

**Definition 4.2.3.** Let $P$ be a well-formed compensable process, then function $\mathtt{Z_p}(P)$ is defined as follows:

$$\mathtt{Z_p}(P) = \begin{cases} 0 & \text{if } \mathtt{pb_P}(P) = 0 \text{ and } \mathtt{ts_P}(P) = 0, \\ 1 & \text{if } \mathtt{pb_P}(P) > 0 \text{ or } \mathtt{ts_P}(P) > 0. \end{cases}$$

The number of reduction steps for the translation of transaction scope depends on the number of protected blocks and transactions in the default activity of that transaction. This dependence occurs if the transaction scope in the default activity contains at least one protected block or at least one transaction. In the translation of such transaction, there is an update location on name $z_t$ (it occurs in process $\mathtt{outp}^{\circ}$, see equation (4.4)). Otherwise, if the number of protected blocks and transaction is equal to zero in a default activity of transaction, which is attacked by an abortion signal, then the number of reduction steps is equal in both considered updates. Therefore, update on name $z_t$ make a significant difference, in terms of efficiency to achieve operational correspondence, between translation with the subjective and objective update. In the following, we prove that translation with the objective update also satisfies operational correspondence.

Most of the lemmas, definitions, and theorems we have introduced to prove the operational correspondence for the translation with subjective update (Section 3.3.3, Theorem 3.3.4), can be adapted for the translation with objective update. We first present an overview of the auxiliary results (and proofs) that are different from those presented in Paragraph 3.3.3.2.1. We start with the following definition which formalizes all possible forms for the process $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$.

**Definition 4.2.4.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$ (cf. Table 4.1 ) depending on $n = \mathtt{nl}(p_{t,\rho}, (\!|P|\!)^{\circ}_{t,\rho})$ and $m = \mathtt{nl}(\beta_{t,\rho}, (\!|P|\!)^{\circ}_{t,\rho})$:

1. if $n = 0$ and $m = 0$ then $p \in \{1, \dots, 6\}$;
2. if $n > 0$ and $m = 0$ then then $(\!|P|\!)^{\circ}_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[(\!|P'_k|\!)^{\circ}_{\varepsilon}] \mid S$ and $p \in \{1, \dots, 7+n\}$;
3. if $n = 0$ and $m > 0$ then $(\!|P|\!)^{\circ}_{t,\rho} = \prod_{i=1}^{m} \beta_{t,\rho}[(\!|P'_k|\!)^{\circ}_{\varepsilon}] \mid S$ and $p \in \{1, \dots, 7+m\}$;
4. otherwise, if $n > 0$ and $m > 0$ then $(\!|P|\!)^{\circ}_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[(\!|P'_k|\!)^{\circ}_{\varepsilon}] \mid \prod_{i=1}^{m} \beta_{t,\rho}[(\!|P'_k|\!)^{\circ}_{\varepsilon}] \mid S$ and $p \in \{1, \dots, 7+n+m\}$.

Table 4.1: Process $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$ with $p \geq 1$. We use abbreviation $\texttt{outp}^{\circ}$ for process $\texttt{outp}^{\circ}(t, (\!|P|\!)^{\circ}_{t,\rho}, p_{t,\rho}, p_{\rho}, \beta_{t,\rho}, \beta_{\rho}, \texttt{nl}(p_{t,\rho}, Y), \texttt{nl}(\beta_{t,\rho}, Y))$.

| $(p)$ | $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$ for $n, m = 0$ |
|---|---|
| (1) | $\beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid t\{\dagger\}.\overline{j_t}.r_t\} \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big]$ $\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| (2) | $\beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid t\{\dagger\}.\overline{j_t}.r_t\Big] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| (3) | $\beta_\rho\Big[\overline{j_t}.r_t \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| (4) | $\beta_\rho\Big[r_t \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big] \mid \beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| (5) | $r_t \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}] \mid \overline{r}_t.\overline{h_t}$ |
| (6) | $p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}] \mid \overline{h_t}$ |
| $(p)$ | $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$ for $n > 0, m = 0$ |
| (1) | $\beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \texttt{outp}^{\circ}\} \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big]$ $\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(2+j)$ <br> $0 \leq j \leq n-1$ | $\beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid p_{t,\rho}\{(X_1,\ldots,X_{n-j}).z_t\{(Z).\prod_{i=1}^{n-j} p_\rho[X_i]$ $\mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid t\{\dagger\}.\overline{j_t}.r_t\}\Big)\}.\overline{z_t}[\mathbf{0}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(2+m)$ | $\beta_\rho\Big[t\big[(\!|P'|\!)^{\circ}_{t,\rho} \mid z_t\{(Z).\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid t\{\dagger\}.\overline{j_t}.r_t\}\big] \mid \overline{z_t}[\mathbf{0}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big]$ $\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(3+n)$ | $\beta_\rho\Big[t\big[(\!|P'|\!)^{\circ}_{t,\rho}\big] \mid t\{\dagger\}.\overline{j_t}.r_t \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big]$ $\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(4+n)$ | $\beta_\rho\Big[\overline{j_t}.r_t \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(5+n)$ | $\beta_\rho\Big[r_t \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big] \mid \beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(6+n)$ | $r_t \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}] \mid \overline{r_t}.\overline{h_t}$ |
| $(7+n)$ | $\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^{\circ}_{\varepsilon}] \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}] \mid \overline{h_t}$ |
| $(p)$ | $I_t^{(p)}((\!|P|\!)^{\circ}_{t,\rho}, (\!|Q|\!)^{\circ}_{\varepsilon})$ for $n = 0, m > 0$ and $\texttt{Nested}_\texttt{t} = \prod_{k=1}^{m} \beta_\rho[(\!|P'_k|\!)^{\circ}_{\varepsilon}] \mid j_{t_k}.\beta_\rho\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}$ |
| (1) | $\beta_\rho\Big[t\big[(\!|P|\!)^{\circ}_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \texttt{outp}^{\circ}\} \mid p_\rho[(\!|Q|\!)^{\circ}_{\varepsilon}]\Big]$ $\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |

| | |
|---|---|
| $(2+s)$<br><br>$0 \le s \le m-1$ | $\beta_\rho\Big[t\big[(\!\vert P\vert\!)^\circ_{t,\rho}\big] \mid \beta_{t,\rho}\{(Y_1,\ldots,Y_{m-s}).z_t\{(Z).\Big(r_t.\big(\prod_{k=1}^{m-s}(\beta_\rho[\mathcal{E}(Y_k,t)]$<br>$\mid j_{t_k}.\beta_\rho\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}) \mid \prod_{k=1}^{s}(\beta_\rho[(\!\vert P'_k\vert\!)^\circ_\varepsilon] \mid j_{t_k}.\beta_\rho\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}})\big)$<br>$\mid t\{\dagger\}.\overline{j_t}\Big)\}\}.z_t[\mathbf{0}] \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(2+m)$ | $\beta_\rho\Big[t\big[(\!\vert P'\vert\!)^\circ_{t,\rho} \mid z_t\{(Z).t\{\dagger\}.\overline{j_t} \mid r_t.(\mathtt{Nested_t})\}\big] \mid z_t[\mathbf{0}] \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big]$<br>$\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(3+m)$ | $\beta_\rho\Big[t\big[(\!\vert P'\vert\!)^\circ_{t,\rho}\big] \mid t\{\dagger\}.\overline{j_t} \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(4+m)$ | $\beta_\rho\Big[\overline{j_t} \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(5+m)$ | $\beta_\rho\Big[\ r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(6+m)$ | $r_t.(\mathtt{Netsed_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |
| $(7+m)$ | $\mathtt{Netsed_t} \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon] \mid \overline{h_t}$ |

| | |
|---|---|
| $(p)$ | $I_t^{(p)}((\!\vert P\vert\!)^\circ_{t,\rho}, (\!\vert Q\vert\!)^\circ_\varepsilon)$ for $n, m > 0$ |
| $(1)$ | $\beta_\rho\Big[t\big[(\!\vert P\vert\!)^\circ_{t,\rho}\big] \mid t\{(Y).t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outp}^\circ\}$<br>$\mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(2+j+s)$<br><br>$0 \le j \le n-1$<br>$0 \le s \le m-1$ | $\beta_\rho\Big[t\big[(\!\vert P\vert\!)^\circ_{t,\rho}\big] \mid p_{t,\rho}\{(X_1,\ldots,X_{n-j}).\beta_{t,\rho}\{(Y_1,\ldots,Y_{m-s}).$<br>$z_t\{(Z).\Big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.\big(\prod_{k=1}^{m-s}(\beta_\rho[\mathcal{E}(Y_k,t)]$<br>$\mid j_{t_k}.\beta_\rho\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}})$<br>$\mid \prod_{k=1}^{s}(\beta_\rho[(\!\vert P'_k\vert\!)^\circ_\varepsilon] \mid j_{t_k}.\beta_\rho\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}})\big) \mid t\{\dagger\}.\overline{j_t}\Big)\}\}\}.z_t[\mathbf{0}]$<br>$\mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(2+n+m)$ | $\beta_\rho\Big[t\big[(\!\vert P'\vert\!)^\circ_{t,\rho} \mid t\{\dagger\}.\overline{j_t} \mid z_t\{(Z).\prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.(\mathtt{Nested_t})\}\big] \mid z_t[\mathbf{0}]$<br>$\mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(3+n+m)$ | $\beta_\rho\Big[t\big[(\!\vert P'\vert\!)^\circ_{t,\rho}\big] \mid t\{\dagger\}.\overline{j_t} \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big]$<br>$\mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(4+n+m)$ | $\beta_\rho\Big[\overline{j_t} \mid \prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid j_t.\beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(5+n+m)$ | $\beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r}_t.\overline{h_t}$ |
| $(6+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!\vert P'_i\vert\!)^\circ_\varepsilon] \mid r_t.(\mathtt{Nested_t}) \mid p_\rho[(\!\vert Q\vert\!)^\circ_\varepsilon] \mid \overline{r_t}.\overline{h_t}$ |

| | |
|---|---|
| $(7+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)^\circ_\varepsilon] \mid \mathtt{Nested_t} \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon] \ \mid \overline{h_t}$ |

**Definition 4.2.5.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}((\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}], (\!|Q|\!)^\circ_\varepsilon)$ (cf. Table 4.2) depending on $n = \mathtt{nl}(p_{u,\rho}, (\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}])$ and $m = \mathtt{nl}(\beta_{u,\rho}, (\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}])$:

1. for $n = 0$ and $m = 0$ we have $q \in \{1, \dots, 6\}$;
2. for $n > 0, m = 0$ and $(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[(\!|P_k'|\!)^\circ_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+n\}$;
3. for $n = 0, m > 0$ and $(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}] = \prod_{i=1}^{m} \beta_{u,\rho}[(\!|P_k'|\!)^\circ_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+m\}$;
4. otherwise, for $n, m > 0$ and $(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[(\!|P_k'|\!)^\circ_\varepsilon] \mid \prod_{i=1}^{m} \beta_{u,\rho}[(\!|P_k'|\!)^\circ_\varepsilon] \mid S$ we have $q \in \{1, \dots, 7+n+m\}$.

Table 4.2: Process $O_u^{(q)}((\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}, (\!|Q|\!)^\circ_\varepsilon])$ with $q \geq 1$. We use abbreviation $\mathtt{outp}^\circ$ for process $\mathtt{outp}^\circ(t, (\!|P|\!)^\circ_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho, \mathtt{nl}(p_{t,\rho}, Y), \mathtt{nl}(\beta_{t,\rho}, Y))$.

| $(q)$ | $O_u^{(q)}((\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}, (\!|Q|\!)^\circ_\varepsilon])$ for $n, m = 0$ |
|---|---|
| (1) | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}]\mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid u\{\dagger\}.\overline{j_u}.r_u\}$ $\mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| (2) | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}]\mid h_u \mid u\{\dagger\}.\overline{j_u}.r_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big]$ $\mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| (3) | $\beta_\rho\Big[h_u \mid \overline{j_u}.r_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| (4) | $\beta_\rho\Big[h_u \mid r_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| (5) | $h_u \mid r_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon] \mid \overline{r}_u.\overline{h_u}$ |
| (6) | $h_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon] \mid \overline{h_u}$ |
| (7) | $p_\rho[(\!|Q|\!)^\circ_\varepsilon]$ |

| $(q)$ | $O_u^{(q)}((\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}, (\!|Q|\!)^\circ_\varepsilon])$ for $n > 0, m = 0$ |
|---|---|
| (1) | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid \mathtt{outp}^\circ\}$ $\mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| $(2+j)$ <br> $0 \leq j \leq n-1$ | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\{(X_1, \dots, X_{n-j}).z_t\{(Z).\big(\prod_{i=1}^{n-j} p_\rho[X_i]$ $\mid \prod_{i=1}^{j} p_\rho[(\!|P_i'|\!)^\circ_\varepsilon] \mid u\{\dagger\}.\overline{j_u}.r_u\big)\}\}.z_t[\mathbf{0}] \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(2+n)$ | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P'|\!)^\circ_{\rho'}] \mid z_t\{(Z).u\{\dagger\}.\overline{j_u}.r_u \mid \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)^\circ_\varepsilon]\}\big] \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]$ |

| | |
|---|---|
| | $\mid h_u \mid z_t[\mathbf{0}]\big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(3+n)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P'\rrparenthesis^\circ_{\rho'}]\big] \mid u\{\dagger\}.\overline{j_u}.r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]$<br>$\mid h_u\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(4+n)$ | $\beta_\rho\Big[h_u \mid \overline{j_u}.r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| $(5+n)$ | $\beta_\rho\Big[h_u \mid r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(6+n)$ | $h_u \mid r_u \mid \prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| $(7+n)$ | $h_u \mid \prod_{i=1}^{m} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon] \mid \overline{h_u}$ |
| $(8+n)$ | $\prod_{i=1}^{n} p_\rho[\llparenthesis P'_i\rrparenthesis^\circ_\varepsilon] \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P\rrparenthesis^\circ_{\rho'}, \llparenthesis Q\rrparenthesis^\circ_\varepsilon])$ for $n=0, m>0$ and<br>$\mathtt{Nested_u} = \prod_{k=1}^{m} \beta_\rho[\llparenthesis P'_k\rrparenthesis^\circ_\varepsilon] \mid j_{u_k}.\beta_\rho\{(X).X\}.\overline{r_{u_k}}.\overline{h_{u_k}}$ |

| | |
|---|---|
| $(1)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P\rrparenthesis^\circ_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid \mathtt{outp}^\circ\}$<br>$\mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| $(2+s)$<br>$0 \leq s \leq m-1$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P\rrparenthesis^\circ_{\rho'}]\big] \mid h_u \mid \beta_{u,\rho}\{(Y_1,\ldots,Y_{m-s}).z_t\{(Z).$<br>$\Big(r_u.\big(\prod_{k=1}^{m-s}(\beta_\rho[\mathcal{E}(Y_k,u)] \mid j_{u_k}.\beta_\rho\{(X).X\}.\overline{r_{u_k}}.\overline{h_{u_k}}) \mid \prod_{k=1}^{s}(\beta_\rho[\llparenthesis P'_k\rrparenthesis^\circ_\varepsilon]$<br>$\mid j_{u_k}.\beta_\rho\{(X).X\}.\overline{r_{u_k}}.\overline{h_{u_k}})\big) \mid u\{\dagger\}.\overline{j_u}\Big)\}\}.z_t[\mathbf{0}]$<br>$\mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(2+m)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P'\rrparenthesis^\circ_{\rho'}]\big] \mid z_t\{(Z).u\{\dagger\}.\overline{j_u} \mid r_u.(\mathtt{Nested_u})\}\Big] \mid z_t[\mathbf{0}] \mid h_u$<br>$\mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(3+m)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P'\rrparenthesis^\circ_{\rho'}]\big] \mid u\{\dagger\}.\overline{j_u} \mid r_u.(\mathtt{Nested_u}) \mid h_u$<br>$\mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(4+m)$ | $\beta_\rho\Big[h_u \mid \overline{j_u} \mid r_u.(\mathtt{Nested_u}) \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r}_u.\overline{h_u}$ |
| $(5+m)$ | $\beta_\rho\Big[h_u \mid r_u.(\mathtt{Nested_u}) \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(6+m)$ | $h_u \mid r_u.(\mathtt{Nested_u}) \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| $(7+m)$ | $h_u \mid \mathtt{Nested_u} \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]\Big] \mid \overline{h_u}$ |
| $(8+m)$ | $\mathtt{Nested_u} \mid p_\rho[\llparenthesis Q\rrparenthesis^\circ_\varepsilon]$ |

| | |
|---|---|
| $(q)$ | $O_u^{(q)}(\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P\rrparenthesis^\circ_{\rho'}, \llparenthesis Q\rrparenthesis^\circ_\varepsilon])$ for $n,m>0$ |

| | |
|---|---|
| $(1)$ | $\beta_\rho\Big[u\big[\llparenthesis F\rrparenthesis^\circ_\rho[h_u.\llparenthesis P\rrparenthesis^\circ_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y) \mid \mathtt{outp}^\circ\}$ |

| | |
|---|---|
| | $\mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(j+s+2)$ <br> $0 \le j \le n-1$ <br> $0 \le s \le m-1$ | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P|\!)^\circ_{\rho'}]\big] \mid h_u \mid p_{u,\rho}\{(X_1,\ldots,X_{n-j}).\beta_{u,\rho}\{(Y_1,\ldots,Y_{m-s}).$ <br> $z_t\{(Z).\big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m-s} (\beta_\rho[\mathcal{E}(Y_k,u)]$ <br> $\mid j_{u_k}.\beta_\rho\{(X).X\}.\overline{r_{u_k}}.\overline{h_{u_k}}) \mid \prod_{k=1}^{s} (\beta_\rho[(\!|P'_k|\!)^\circ_\varepsilon] \mid j_{u_k}.\beta_\rho\{(X).X\}.\overline{r_{u_k}}.\overline{h_{u_k}}))$ <br> $\mid u\{\dagger\}.\overline{j_u}\big)\}\}\}.z_t[\mathbf{0}] \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(2+n+m)$ | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P'|\!)^\circ_{\rho'}] \mid z_t\{(Z).\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.(\texttt{Nested}_\texttt{u} \mid u\{\dagger\}.\overline{j_u})\}\big]$ <br> $\mid z_t[\mathbf{0}] \mid h_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(3+n+m)$ | $\beta_\rho\Big[u\big[(\!|F|\!)^\circ_\rho[h_u.(\!|P'|\!)^\circ_{\rho'}]\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.(\texttt{Nested}_\texttt{u} \mid u\{\dagger\}.\overline{j_u})$ <br> $\mid h_u \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(4+n+m)$ | $\beta_\rho\Big[h_u \mid \overline{j_u} \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.(\texttt{Nested}_\texttt{u}) \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big]$ <br> $\mid j_u.\beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(5+n+m)$ | $\beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.(\texttt{Nested}_\texttt{u}) \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\Big] \mid \beta_\rho\{(X).X\}.\overline{r_u}.\overline{h_u}$ |
| $(6+n+m)$ | $h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid r_u.(\texttt{Nested}_\texttt{u}) \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon] \mid \overline{r_u}.\overline{h_u}$ |
| $(7+n+m)$ | $h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid \texttt{Nested}_\texttt{u} \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon] \mid \overline{h_u}$ |
| $(8+n+m)$ | $\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)^\circ_\varepsilon] \mid \texttt{Nested}_\texttt{u} \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]$ |

For the proof of operational correspondence we need the following statement:

**Lemma 4.2.3.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $(\!|P|\!)^\circ_\rho \equiv (\!|Q|\!)^\circ_\rho$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

In the following we prove that operational correspondence hols for translation $(\!|\cdot|\!)^\circ_\rho$.

**Theorem 4.2.4** (Operational Correspondence for $(\!|\cdot|\!)^\circ_\rho$)**.** Let $P$ be a well-formed process in $\mathcal{C}_\texttt{P}$.

(1) If $P \longrightarrow P'$ then $(\!|P|\!)^\circ_\varepsilon \longrightarrow^k (\!|P'|\!)^\circ_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_\mathtt{P}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ it follows $k = 7 + \mathtt{pb}_\mathtt{P}(P_1) + \mathtt{ts}_\mathtt{P}(P_1) + \mathtt{Z}_\mathtt{P}(P)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ it follows $k = 7 + \mathtt{pb}_\mathtt{P}(F[P_1]) + \mathtt{ts}_\mathtt{P}(F[P_1]) + \mathtt{Z}_\mathtt{P}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$ processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

(2) If $(\!|P|\!)^\circ_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* (\!|P'|\!)^\circ_\varepsilon$.

*Proof.* We consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases *a*), *b*) and *c*) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 4.2.3, Definition 4.2.2, and Lemma 3.2.9 (Page 47) which holds for $(\!|\cdot|\!)^{\circ}_{\rho}$.

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have that derivation corresponds to eq. (3.42) (Page 87). Therefore, the thesis holds with $k = 1$.

(i) In this sub-case $n, m = 0$. Therefore, we have the following derivation:

$$
\begin{aligned}
(\!|P|\!)^{\circ}_{\varepsilon} \equiv\ & (\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!)^{\circ}_{\varepsilon}\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C[t[P_1,Q]]|\!)^{\circ}_{\rho} \mid (\!|D[\overline{t}.P_2]|\!)^{\circ}_{\rho}\Big]\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}[(\!|t[P_1,Q]|\!)^{\circ}_{\rho'}] \mid (\!|D|\!)^{\circ}_{\rho}[(\!|\overline{t}.P_2|\!)^{\circ}_{\rho''}]\Big]\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}[t\big[(\!|P_1|\!)^{\circ}_{t,\rho'}\big] \mid t.\big(\mathtt{extrp}\{t, (\!|P_1|\!)^{\circ}_{t,\rho'}, p_{t,\rho'}, p_{\rho'}, \beta_{t,\rho'}, \beta_{\rho'}\} \mid p_{\rho'}[(\!|Q|\!)^{\circ}_{\varepsilon}]\big)]\\
\mid\ & (\!|D|\!)^{\circ}_{\rho}[\overline{t}.h_t.(\!|P_2|\!)^{\circ}_{\rho''}]\Big]\\
\longrightarrow\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}\Big[I^{(1)}_t((\!|P_1|\!)^{\circ}_{t,\rho'}, (\!|Q|\!)^{\circ}_{\varepsilon})\Big] \mid (\!|D|\!)^{\circ}_{\rho}[h_t.(\!|P_2|\!)^{\circ}_{\rho''}]\Big]\\
\longrightarrow^5\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}\Big[I^{(6)}_t((\!|P_1|\!)^{\circ}_{t,\rho'}, (\!|Q|\!)^{\circ}_{\varepsilon})\Big] \mid (\!|D|\!)^{\circ}_{\rho}\big[h_t.(\!|P_2|\!)^{\circ}_{\rho''}\big]\Big]\\
\longrightarrow\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}[(\!|\langle Q\rangle|\!)^{\circ}_{\rho'}] \mid (\!|D|\!)^{\circ}_{\rho}[(\!|P_2|\!)^{\circ}_{\rho''}]\Big]\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}[(\!|C[\langle Q\rangle]|\!)^{\circ}_{\rho} \mid (\!|D[P_2]|\!)^{\circ}_{\rho}]\\
=\ & (\!|E[C[\langle Q\rangle] \mid D[P_2]]|\!)^{\circ}_{\varepsilon}\\
\equiv\ & (\!|P'|\!)^{\circ}_{\varepsilon}
\end{aligned}
\tag{4.10}
$$

Thus, the number of reduction steps is $k = 7$, and it has the following description:

- The first synchronization concerns $t$ and $\overline{t}$.
- Notice that here $\longrightarrow^5$ tells us that there have been five reduction steps:
  - the first one is an update on location name $t$;
  - the second reduction step "kills" with $t\{\dagger\}$ both the location $t$ and the process it hosts;
  - the third reduction step is synchronization on name $j_t$ and $\overline{j_t}$.
  - the fourth reduction step is an update on location name $\beta_{\rho}$;
  - the fifth reduction step is synchronization on name $r_t$ and $\overline{r_t}$.
- The last synchronization concerns $h_t$ and $\overline{h_t}$.

(ii) In this sub-case we consider the following: $n > 0, m = 0$ or $n = 0, m > 0$ or $n, m > 0$ , i.e., this is when there is at least one protected block and/or transaction scope in the default activity $P_1$. We have the following derivation:

$$
\begin{aligned}
(\!|P|\!)^{\circ}_{\varepsilon} \equiv\ & (\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!)^{\circ}_{\varepsilon}\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C[t[P_1,Q]]|\!)^{\circ}_{\rho} \mid (\!|D[\overline{t}.P_2]|\!)^{\circ}_{\rho}\Big]\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\Big[(\!|C|\!)^{\circ}_{\rho}[(\!|t[P_1,Q]|\!)^{\circ}_{\rho'}] \mid (\!|D|\!)^{\circ}_{\rho}[(\!|\overline{t}.P_2|\!)^{\circ}_{\rho''}]\Big]\\
=\ & (\!|E|\!)^{\circ}_{\varepsilon}\big[(\!|C|\!)^{\circ}_{\rho}\Big[t\big[(\!|P_1|\!)^{\circ}_{t,\rho'}\big] \mid t.\big(\mathtt{extrp}\{t, (\!|P_1|\!)^{\circ}_{t,\rho'}, p_{t,\rho'}, p_{\rho'}, \beta_{t,\rho'}, \beta_{\rho'}\}
\end{aligned}
$$

$$
\begin{aligned}
& \mid p_{\rho'}[(\!|Q|\!)^\circ_\varepsilon])\Big] (\!|D|\!)^\circ_\rho[\overline{t}.h_t.(\!|P_2|\!)^\circ_{\rho''}]] \\
\longrightarrow\ & (\!|E|\!)^\circ_\varepsilon\Big[(\!|C|\!)^\circ_\rho\big[I_t^{(1)}((\!|P_1|\!)^\circ_{t,\rho'}, (\!|Q|\!)^\circ_\varepsilon)\big] \mid (\!|D|\!)^\circ_\rho[\overline{t}.h_t.(\!|P_2|\!)^\circ_{\rho''}]\Big] \\
\longrightarrow^{6+n+m}\ & (\!|E|\!)^\circ_\varepsilon\Big[(\!|C|\!)^\circ_\rho\big[\big[I_t^{(n+m+7)}((\!|P_1|\!)^\circ_{t,\rho'}, (\!|Q|\!)^\circ_\varepsilon)\big] \mid (\!|D|\!)^\circ_\rho\big[h_t.(\!|P_2|\!)^\circ_{\rho''}\big]\Big] \\
\longrightarrow\ & (\!|E|\!)^\circ_\varepsilon\Big[(\!|C|\!)^\circ_\rho\big[(\!|\mathsf{extr_P}(P_1) \mid \langle Q\rangle|\!)^\circ_{\rho'}\big] \mid (\!|D|\!)^\circ_\rho\big[(\!|P_2|\!)^\circ_{\rho''}\big]\Big] \\
=\ & (\!|E|\!)^\circ_\varepsilon\Big[(\!|C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle]|\!)^\circ_\rho \mid (\!|D[P_2]|\!)^\circ_\rho\Big] \\
=\ & (\!|E[C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle] \mid D[P_2]]|\!)^\circ_\varepsilon \\
\equiv\ & (\!|P'|\!)^\circ_\varepsilon
\end{aligned}
$$

Notice that by Lemma 3.2.14 we know that $\mathsf{ch}(t, (\!|P|\!)^\circ_\rho) = \mathbf{0}$. The order/nature of these reduction steps is similar as in the proof of **Part (1) – Completeness** 1 for translation $\mathcal{C}_\mathsf{P}$ into $\mathcal{S}$ (cf. explanation 1b). The difference is in the synchronization on update prefix and location with name $z_t$. This extra step we need for the relocation of processes $p_{\rho'}[\cdot]$, which are enclosed on location $t$. Therefore, we can conclude that $(\!|P|\!)_\varepsilon \longrightarrow^k (\!|P'|\!)_\varepsilon$ where $k = 8 + n + m$.

Process $I_t^{(p)}((\!|P_1|\!)^\circ_{t,\rho'}, (\!|Q|\!)^\circ_\varepsilon)$ is as in Definition 4.2.4 (cf. Table 4.1).

b) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[u[F[\overline{u}.P_1],Q]]$, with $m = \mathtt{pb_P}(F[P_1])$ and $n = \mathtt{ts_P}(F[P_1])$, and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle]$. Then we analyze the following two cases:

(i) In this sub-case we consider $n, m = 0$, and we have the derivation that is like derivation presented with eq. (4.10). The difference is that in this case we use $O_u^{(q)}((\!|F|\!)^\circ_{u,\rho}[h_u.(\!|P_1|\!)^\circ_{\rho'}], (\!|Q|\!)^\circ_\varepsilon)$ with $q \in \{1, \dots, 6\}$ (cf. Table 4.2) instead $I_t^{(p)}((\!|P_1|\!)^\circ_{t,\rho'}, (\!|Q|\!)^\circ_\varepsilon)$ with $p \in \{1, \dots, 6\}$ (cf. Table 4.1). Thus, the number of reduction steps is $k = 7$.

(ii) This sub-case analyses scenarios in which: $m > 0, n = 0$ or $m = 0, n > 0$ or $m > 0, n > 0$. We have the following derivation:

$$
\begin{aligned}
(\!|P|\!)^\circ_\varepsilon \equiv\ & (\!|C\big[u[F[\overline{u}.P_1],Q]\big]|\!)^\circ_\varepsilon \\
=\ & (\!|C|\!)^\circ_\varepsilon\big[(\!|u[F[\overline{u}.P_1],Q]|\!)^\circ_\rho\big] \\
=\ & (\!|C|\!)^\circ_\varepsilon\big[u\big[(\!|F[\overline{u}.P_1]|\!)^\circ_{u,\rho}\big] \\
=\ & (\!|C|\!)^\circ_\varepsilon\big[u\big[(\!|F|\!)^\circ_{u,\rho}[\overline{u}.h_u.(\!|P_1|\!)^\circ_{\rho'}]\big] \\
& \mid u.(\mathtt{extrp}\{u, (\!|F[\overline{u}.P_1]|\!)^\circ_{u,\rho}, p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho\} \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon])\big] \\
\longrightarrow\ & (\!|C|\!)^\circ_\varepsilon\Big[O_u^{(1)}((\!|F|\!)^\circ_{u,\rho}[h_u.(\!|P_1|\!)^\circ_{\rho'}], (\!|Q|\!)^\circ_\varepsilon)\Big] \\
\longrightarrow^{7+n+m}\ & (\!|C|\!)^\circ_\varepsilon\big[O_u^{(8+n+m)}((\!|F|\!)^\circ_{u,\rho}[h_u.(\!|P_1|\!)^\circ_{\rho'}], (\!|Q|\!)^\circ_\varepsilon)\big] \\
\equiv\ & (\!|C|\!)^\circ_\varepsilon\big[(\!|\mathsf{extr_P}(F[P_1])|\!)^\circ_\rho \mid p_\rho[(\!|Q|\!)^\circ_\varepsilon]\big] \\
=\ & (\!|C\big[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle\big]|\!)^\circ_\varepsilon \\
\equiv\ & (\!|P'|\!)^\circ_\varepsilon
\end{aligned}
$$

Process $O_u^{(q)}((\!|F|\!)^\circ_{u,\rho}[h_u.(\!|P_1|\!)^\circ_{\rho'}], (\!|Q|\!)^\circ_\varepsilon)$, where $q \in \{1, \dots, 7+n+m\}$, is as in Definition 4.2.5 ( cf. Table 4.2 ). In this case, the role of function $\mathsf{ch}(u, \cdot)$ is central: indeed, $\mathsf{ch}(u, (\!|F|\!)^\circ_{u,\rho}\big[h_u.(\!|P_1|\!)^\circ_{\rho'}\big])$ provides the input $h_u$ which is necessary to achieve operational correspondence.
The order/nature/number of reduction steps can be explained as in Case b) above. We can then conclude that $(\!|P|\!)^\circ_\varepsilon \longrightarrow^k (\!|P'|\!)^\circ_\varepsilon$ where $k = 8 + n + m$.

(2) **Part (2) – Soundness:** The proof of soundness follows the explanation presented in Roadmap 3.3.3.2.3 and the same derivation that is presented in the proof of soundness for translation $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{O}$ (Theorem 4.1.6).

■

### 4.2.4 Comparing Subjective vs Objective update

In this subsection, we provide that subjective updates are better suited to encode compensation handling with preserving semantics than objective updates, again because they induce tighter operational correspondences.

The following statement is a corollary of Theorem 4.2.4.

**Corollary 4.2.5.** Let $P$ be a well-formed process in $\mathcal{C}$. If $P \longrightarrow P'$ and $(\!|P|\!)^{\circ}_{\varepsilon} \longrightarrow^{k} (\!|P'|\!)^{\circ}_{\varepsilon}$ then:

b) if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_P}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ then $k \geq 7 + \mathsf{pb_P}(P_1) + \mathsf{Z_P}(P_1)$,

c) if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q \rangle]$ then $k \geq 7 + \mathsf{pb_P}(F[P_1]) + \mathsf{Z_P}(F[P_1])$.

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

We then have the following theorem:

**Theorem 4.2.6.** The encoding $(\!|\cdot|\!)_{\rho} : \mathcal{C}_{\mathtt{P}} \longrightarrow \mathcal{S}$ is as or more efficient than $(\!|\cdot|\!)^{\circ}_{\rho} : \mathcal{C}_{\mathtt{P}} \longrightarrow \mathcal{O}$.

*Proof.* Let $P \longrightarrow P'$ and $(\!|P|\!)^{\circ}_{\varepsilon} \longrightarrow^{k_2} (\!|P'|\!)^{\circ}_{\varepsilon}$. By Theorem 3.3.4, there is $k_1$ such that $(\!|P|\!)_{\varepsilon} \longrightarrow^{k_1} (\!|P'|\!)_{\varepsilon}$ and

a) $k_1 = 1$ if $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$,

b) $k_1 = 7 + \mathsf{pb_P}(P_1)$ if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_P}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$,

c) $k_1 = 7 + \mathsf{pb_P}(F[P_1])$ if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q \rangle]$.

By Theorem 4.2.4, Proposition 2.2.3 and Corollary 4.2.5 we have the following tree cases, for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$, processes $P_1, Q, P_2$ and names $t, u$:

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ and $k_2 \geq 1 = k_1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_P}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ and $k_2 \geq 7 + \mathsf{pb_P}(P_1) + \mathsf{Z_P}(P_1) \geq 7 + \mathsf{pb_P}(P_1) = k_1$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q \rangle]$ and $k_2 \geq 7 + \mathsf{pb_P}(F[P_1]) + \mathsf{Z_P}(F[P_1]) \geq 7 + \mathsf{pb_P}(F[P_1]) = k_1$.

Thus, in all three cases $k_1 \leq k_2$; by Definition 4.1.7 we conclude that $(\!|\cdot|\!)_{\varepsilon}$ is as or more efficient than $(\!|\cdot|\!)^{\circ}_{\varepsilon}$. ■

## 4.3 Translating $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$

In the this subsection we concentrate on a source calculus with *static recovery* and *aborting semantics*. We introduce useful conventions and intuitions for the translation and then we give the formal presentations of it.

$$
\begin{aligned}
\llparenthesis \langle P\rangle \rrparenthesis^{\circ}_{\rho} &= p_\rho\big[\llparenthesis P\rrparenthesis^{\circ}_{\varepsilon}\big]\\
\llparenthesis t[P,Q]\rrparenthesis^{\circ}_{\rho} &= t\big[\llparenthesis P\rrparenthesis^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\llparenthesis Q\rrparenthesis^{\circ}_{\varepsilon}]) \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}\\
\llparenthesis \bar{t}.P\rrparenthesis^{\circ}_{\rho} &= \bar{t}.h_t.\llparenthesis P\rrparenthesis^{\circ}_{\rho}
\end{aligned}
$$

Figure 4.6: Translating $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$.

### 4.3.1 The Translation, Formally

The translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$, denoted $\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}$, also relies on the key ideas of encoding $\llparenthesis\cdot\rrparenthesis_{\rho}$, Section 3.4. We use paths and reserved names given in Definition 3.1.1 and Definition 3.1.2, respectively. For encoding protected block we use the same intuition as in all previous presented encodings. To encode transactions and their extraction function we use the auxiliary process $\mathtt{outd}^{\circ}(l_1, l_2, n, Q)$, given with eq. (4.1).

A failure signal extracts all nested protected blocks and erases nested locations; our translation does the same with the corresponding located processes and nested locations. We define the following auxiliary process:

**Definition 4.3.1** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names and $\mathtt{out}^{\circ}(\cdot)$ is defined with 4.1. We write $\mathtt{extra}\{t, l_1, l_2\}$ to stand for the following update prefix:

$$\mathtt{extra}\{t, l_1, l_2\} = t\{(Y).t[Y] \mid \mathsf{ch}(t, Y) \mid \mathtt{outd}^{\circ}(l_1, l_2, \mathtt{nl}(l, Y), t\{\dagger\}.\overline{k_t})\}. \tag{4.11}$$

Now we can present the translation $\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}$ formally. It is defined as follows:

**Definition 4.3.2** (Translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with aborting semantics into adaptable processes as a tuple $(\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}, \varphi_{\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}})$ where:

(a) The renaming policy $\varphi_{\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$\varphi_{\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, k_x, r_x, z_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases}$$

(b) The translation $\llparenthesis\cdot\rrparenthesis^{\circ}_{\rho} : \mathcal{C}_{\mathtt{A}} \to \mathcal{S}$ is as in Figure 4.6 and as a homomorphism for other operators.

The basic intuition for the translation of $\langle Q\rangle$ and $t[P,Q]$ is as in the case of the encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$. The difference is in the processes $\mathtt{out}^{\mathtt{s}}$ and $\mathtt{out}^{\circ}$, and it has been explained in Remark 4.1.1.

### 4.3.2 Translation Correctness

In this subsection we give proof of correctness of the translation presented in Definition 4.3.2 which includes proofs of *structural criteria semantic criteria.*

#### 4.3.2.1 Structural Criteria

In this subsection, we prove the compositionality and name invariance.

#### 4.3.2.1.1 Compositionality

The first property is compositionality. For the proof of compositionality criterion, we need to define a *context* for each process operator, which depends on free names of the subterms. This definition relies entirely on Definition 3.2.4 by using $\langle\!|\cdot|\!\rangle_\rho^\circ$ instead of $\langle\!|\cdot|\!\rangle_\rho$.

**Definition 4.3.3** (Compositional context for $\mathcal{C}_{\mathtt{A}}$)**.** For all process operator from $\mathcal{C}_{\mathtt{A}}$, instead transaction we define a compositional context in $\mathcal{O}$ as in Definition 3.4.4. For transaction compositional context is:

$$C_{t[,],\rho}[\bullet_1,\bullet_2] = t\big[[\bullet_1]\big] \mid r_t.\,(\mathtt{extra}\{t,p_{t,\rho},p_\rho\} \mid p_\rho[[\bullet_2]]) \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}$$

Using this definition, we may now state the following result:

**Theorem 4.3.1** (Compositionality for $\langle\!|\cdot|\!\rangle_\rho^\circ$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}_{\mathtt{A}}$ and for all compensable processes $P$ and $Q$ it holds that:

$$\langle\!|\langle P\rangle|\!\rangle_\rho^\circ = C_{\langle\rangle,\rho}[\langle\!|P|\!\rangle_\varepsilon^\circ] \qquad \langle\!|t[P,Q]|\!\rangle_\rho^\circ = C_{t[,],\rho}[\langle\!|P|\!\rangle_{t,\rho}^\circ,\langle\!|Q|\!\rangle_\varepsilon^\circ] \qquad \langle\!|P \mid Q|\!\rangle_\rho^\circ = C_{\mid}\,[\langle\!|P|\!\rangle_\rho^\circ,\langle\!|Q|\!\rangle_\rho^\circ]$$
$$\langle\!|a.P|\!\rangle_\rho^\circ = C_{a.}\left[\langle\!|P|\!\rangle_\rho^\circ\right] \qquad \langle\!|\overline{t}.P|\!\rangle_\rho^\circ = C_{\overline{t}.}[\langle\!|P|\!\rangle_\rho^\circ] \qquad \langle\!|(\nu x)P)|\!\rangle_\rho^\circ = C_{(\nu x)}[\langle\!|P|\!\rangle_\rho^\circ]$$
$$\langle\!|\overline{a}.P|\!\rangle_\rho^\circ = C_{\overline{a}.}[\langle\!|P|\!\rangle_\rho^\circ] \qquad \langle\!|!\pi.P|\!\rangle_\rho^\circ = C_{!\pi.}[\langle\!|P|\!\rangle_\rho^\circ]$$

*Proof.* Follows directly from the definition of contexts (Definition 3.2.4) and from the definition of $\langle\!|\cdot|\!\rangle_\rho^\circ : \mathcal{C}_{\mathtt{A}} \to \mathcal{O}$ (Figure 4.6) and has the same derivation as the proof of Theorem 3.2.2. ■

#### 4.3.2.1.2 Name invariance

The second property is name invariance. We analyze this property with respect to the renaming policy presented in Definition 3.2.3 (b).

**Theorem 4.3.2** (Name invariance for $\langle\!|\cdot|\!\rangle_\rho^\circ$)**.** For every *well-formed* compensable processes $P$ and substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c : \varphi_{\langle\!|\cdot|\!\rangle_{\sigma(\rho)}^\circ}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{\langle\!|\cdot|\!\rangle_\rho^\circ}(x)\}$, and

(ii) $\langle\!|\sigma(P)|\!\rangle_{\sigma(\rho)}^\circ = \sigma'(\langle\!|P|\!\rangle_\rho^\circ)$

*Proof.* The proof proceeds in the same way as the proof of Theorem 3.2.4 by using $\langle\!|\cdot|\!\rangle_\rho^\circ$ instead of $[\![\cdot]\!]_\rho$. ■

### 4.3.2.2 Semantic Criteria

In the following we prove that translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$ satisfied operational correspondence. We leave the analysis of divergence reflection and success sensitiveness for future research.

#### 4.3.2.2.1 Operational Correspondence

In this section we shall prove that operational correspondence (completeness and soundness) holds for the translation $\langle\!|\cdot|\!\rangle_\rho^\circ$.

We are interested in precisely accounting for the number of computation steps induced by our translation. We need the following definition.

**Definition 4.3.4.** Let $P$ be a well-formed compensable process, then function $\mathtt{Z_a}(P)$ is defined as follows:

$$\mathtt{Z_a}(P) = \begin{cases} 1 & \text{if } P = \langle P_1 \rangle, \\ 1 + \mathtt{Z_a}(P_1) & \text{if } P = t[P_1, Q_1], \\ \mathtt{Z_a}(P_1) + \mathtt{Z_a}(P_2) & \text{if } P = P_1 \mid P_2, \\ \mathtt{Z_a}(P_1) & \text{if } P = (\nu a)P_1, \\ 0 & \text{otherwise.} \end{cases}$$

Motivation for this definition is similar with the motivation in the discarding and preserving semantics. When the number of protected blocks is at least one we have that there exist update location on name $z_t$. The level of protection that has abortion semantics orders that when the number of nested transactions is at least one (there exists at least one nested transaction), the structure of these transactions must be researched. When in nested transaction, the number of protected blocks in default activity is at least one, the number of update location on name $z_t$ will be increased for one.

Most of the lemmas, definitions, and theorems we have introduced to prove the operational correspondence for the translation with subjective update (cf. Theorem 3.4.6), can be easily adapted for the translation with objective update.

The following definition formalizes the intermediate processes that appear during derivation, denoted with $I_t^{(p)}(\llbracket P \rrbracket^{\circ}_{t,\rho}, \llbracket Q \rrbracket^{\circ}_{\varepsilon})$. As in previously presented encodings, it plays a significant role in proving completeness and soundness.

**Definition 4.3.5.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}(\llbracket P \rrbracket^{\circ}_{t,\rho}, \llbracket Q \rrbracket^{\circ}_{\varepsilon})$ (Table 4.3) depending on $n = \mathtt{nl}(p_{t,\rho}, \llbracket P \rrbracket^{\circ}_{t,\rho})$, $m = \mathtt{ts_A}(P)$ and $s = \mathtt{S}(P)$:

1. if $n = 0$ then $p \in \{1, \ldots, 6\}$;
2. otherwise, if $n, m > 0$ then $\llbracket P \rrbracket^{\circ}_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\llbracket P'_k \rrbracket^{\circ}_{\varepsilon}] \mid S$ and $p \in \{1, \ldots, 6 + 4m + s\}$.

Table 4.3: Process $I_t^{(p)}(\llbracket P \rrbracket^{\circ}_{t,\rho}, \llbracket Q \rrbracket^{\circ}_{\varepsilon})$ with $p \geq 1$.

| $(p)$ | $I_t^{(p)}(\llbracket P \rrbracket^{\circ}_{t,\rho}, \llbracket Q \rrbracket^{\circ}_{\varepsilon})$ for $n = 0$ |
|---|---|
| (1) | $t\big[\llbracket P \rrbracket^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}])$ $\mid t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}$ |
| (2) | $t\big[\llbracket P \rrbracket^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}]) \mid \mathcal{T}_t(\llbracket P \rrbracket^{\circ}_{t,\rho}).\overline{h_t}$ $\equiv t\big[\llbracket P \rrbracket^{\circ}_{t,\rho}\big] \mid r_t.\big(t\{(Y).(t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\circ}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\{\dagger\}.\overline{k_t}))\} \mid p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}]\big) \mid \overline{r_t}.k_t.\overline{h_t}$ |
| (3) | $t\big[\llbracket P \rrbracket^{\circ}_{t,\rho}\big] \mid t\{(Y).(t[Y] \mid \mathsf{ch}(t, Y)$ $\mid \mathtt{outd}^{\circ}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\{\dagger\}.\overline{k_t}))\} \mid p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| (4) | $t\big[\llbracket P \rrbracket^{\circ}_{t,\rho}\big] \mid p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}] \mid t\{\dagger\}.\overline{k_t} \mid k_t.\overline{h_t}$ |
| (5) | $p_\rho[\llbracket Q \rrbracket^{\circ}_{\varepsilon}] \mid \overline{k_t} \mid k_t.\overline{h_t}$ |

| | |
|---|---|
| (6) | $p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid \overline{h_t}$ |

| $(p)$ | $I^{(p)}_t(\llparenthesis P \rrparenthesis^{\circ}_{t,\rho}, \llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon})$ for and $n, m > 0$ |
|---|---|
| (1) | $t\big[\llparenthesis P \rrparenthesis^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}]) \mid t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}$ |
| $(2+4m+s-n)$ | $t\big[\llparenthesis P' \rrparenthesis^{\circ}_{t,\rho} \mid \prod_{i=1}^{s-n} p_{t,\rho}[\llparenthesis P'_i \rrparenthesis^{\circ}_{\varepsilon}]\big] \mid r_t.\big(t\{(Y).(t[Y] \mid \mathsf{ch}(t,Y)$<br>$\mid \mathtt{outd}^{\circ}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\{\dagger\}.\overline{k_t}))\} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}]\big) \mid \overline{r_t}.k_t.\overline{h_t}$ |
| $(3+4m+s-n)$ | $t\big[\llparenthesis P' \rrparenthesis^{\circ}_{t,\rho} \mid \prod_{i=1}^{s-n} p_\rho[\llparenthesis P'_i \rrparenthesis^{\circ}_{\varepsilon}]\big] \mid t\{(Y).(t[Y] \mid \mathsf{ch}(t,Y)$<br>$\mid \mathtt{outd}^{\circ}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), t\{\dagger\}.\overline{k_t}))\} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| $(4+4m+s-n+j)$<br><br>$0 \leq j \leq n-1$ | $t\big[\llparenthesis P' \rrparenthesis^{\circ}_{t,\rho} \mid \prod_{i=1}^{s-n} p_\rho[\llparenthesis P'_i \rrparenthesis^{\circ}_{\varepsilon}] \mid p_{t,\rho}\{(X_1, \ldots, X_{n-j}).$<br>$z_t\{(Z).\big(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}] \mid t\{\dagger\}.\overline{k_t}\big)\}\}\big] \mid z_t[\mathbf{0}]$<br>$\mid p_\rho[\llparenthesis Q_t \rrparenthesis^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| $(4+4m+s)$ | $t\big[\llparenthesis P' \rrparenthesis^{\circ}_{t,\rho} \mid z_t\{(Z).\prod_{k=1}^{n} p_\rho[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{k_t}\}\big] \mid z_t[\mathbf{0}] \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| $(5+4m+s)$ | $t\big[\llparenthesis P' \rrparenthesis^{\circ}_{t,\rho}\big] \mid \prod_{k=1}^{n} p_\rho[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{k_t} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| $(6+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}]\overline{k_t} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid k_t.\overline{h_t}$ |
| $(7+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}] \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}] \mid \overline{h_t}$ |

The following definition formalizes all possible forms for the process $O^{(q)}_u(\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}], \llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon})$.

**Definition 4.3.6.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O^{(q)}_u(\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}], \llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon})$ (Table 4.4) depending on $n = \mathtt{nl}(p_{u,\rho}, \llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}])$, $m = \mathtt{ts}_{\mathtt{A}}(F[P])$ and $s = \mathtt{S}(F[P])$:

1. for $n = 0$ we have $q \in \{1, \ldots, 7\}$, and
2. for $n, m > 0$ and $\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[\llparenthesis P'_k \rrparenthesis^{\circ}_{\varepsilon}] \mid S$ we have $q \in \{1, \ldots, 7+4m+s\}$.

Table 4.4: Process $O^{(q)}_u(\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}], \llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon})$ with $q \geq 1$.

| $(q)$ | $O^{(q)}_u(\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}], \llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon})$, $n = 0$ |
|---|---|
| (1) | $u\big[\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}])$<br>$\mid u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}$ |
| (2) | $u\big[\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\circ}_{\varepsilon}])$<br>$\mid \mathcal{T}_u(\llparenthesis F \rrparenthesis^{\circ}_{\rho}[h_u.\llparenthesis P \rrparenthesis^{\circ}_{\rho'}]).\overline{h_u}$ |

| | |
|---|---|
| | $\equiv u\big[\langle\!\|P\|\!\rangle^{\circ}_{u,\rho}\big] \mid r_u.\big(u\{(Y).u[Y] \mid \mathsf{ch}(u,Y)$<br>$\mid \mathtt{outd}^{\mathtt{u}}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho},Y), u\{\dagger\}.\overline{k_u})\} \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}]\big) \mid \overline{r_u}.k_u.\overline{h_u}$ |
| (3) | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P\|\!\rangle^{\circ}_{\rho'}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y)$<br>$\mid \mathtt{outd}^{\circ}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho},Y), u\{\dagger\}.\overline{k_u})\}\big) \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| (4) | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P\|\!\rangle^{\circ}_{\rho'}]\big] \mid h_u \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid u\{\dagger\}.\overline{k_u} \mid k_u.\overline{h_u}$ |
| (5) | $p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid h_u \mid \overline{k_u} \mid k_u.\overline{h_u}$ |
| (6) | $p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid h_u \mid \overline{h_u}$ |
| (7) | $p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}]$ |
| $(q)$ | $O^{(q)}_u(\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P\|\!\rangle^{\circ}_{\rho'}], \langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon})$ for $n, m > 0$ |
| (1) | $t\big[\langle\!\|F\|\!\rangle_{\rho}[h_u.\langle\!\|P\|\!\rangle_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[\langle\!\|Q\|\!\rangle_{\varepsilon}])$<br>$\mid u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}$ |
| $(2+4m+s-n)$ | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P\|\!\rangle^{\circ}_{\rho'} \mid \prod_{i=1}^{s-n} p_{u,\rho}[\langle\!\|P'_i\|\!\rangle^{\circ}_{\varepsilon}]\big] \mid r_u.\big(u\{(Y).u[Y] \mid \mathsf{ch}(u,Y)$<br>$\mid \mathtt{outd}^{\mathtt{u}}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho},Y), u\{\dagger\}.\overline{k_u})\} \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}]\big) \mid \overline{r_u}.k_u.\overline{h_u}$ |
| $(3+4m+s-n)$ | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P'\|\!\rangle^{\circ}_{\rho'} \mid \prod_{i=1}^{s-n} p_\rho[\langle\!\|P'_i\|\!\rangle^{\circ}_{\varepsilon}]\big] \mid u\{(Y).u[Y] \mid \mathsf{ch}(u,Y)$<br>$\mid \mathtt{outd}^{\mathtt{s}}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho},Y), u\{\dagger\}.\overline{k_u})\} \mid h_u \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| $(4+4m+s-n+j)$<br><br>$0 \leq j \leq n-1$ | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P'\|\!\rangle^{\circ}_{\rho'} \mid \prod_{i=1}^{s-n} p_\rho[\langle\!\|P'_i\|\!\rangle^{\circ}_{\varepsilon}] \mid p_{u,\rho}\{(X_1,\ldots,X_{n-j}).$<br>$z_t\{(Z).(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid u\{\dagger\}.\overline{k_u})\}\}\big] \mid z_t[\mathbf{0}]$<br>$\mid h_u \mid p_\rho[\langle\!\|Q_u\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| $(4+4m+s)$ | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P'\|\!\rangle^{\circ}_{\rho'}] \mid z_t\{(Z).\prod_{k=1}^{n} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid u\{\dagger\}.\overline{k_u}\}\big] \mid z_t[\mathbf{0}]$<br>$\mid h_u \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| $(5+4m+s)$ | $u\big[\langle\!\|F\|\!\rangle^{\circ}_{\rho}[h_u.\langle\!\|P'\|\!\rangle^{\circ}_{\rho'}]\big] \mid \prod_{k=1}^{n} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid u\{\dagger\}.\overline{k_u} \mid h_u \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| $(6+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid h_u \mid \overline{k_u} \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid k_u.\overline{h_u}$ |
| $(7+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid h_u \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}] \mid \overline{h_u}$ |
| $(8+4m+s)$ | $\prod_{k=1}^{n} p_\rho[\langle\!\|P'_k\|\!\rangle^{\circ}_{\varepsilon}] \mid p_\rho[\langle\!\|Q\|\!\rangle^{\circ}_{\varepsilon}]$ |

For the proof of operational correspondence we need the following statement:

**Lemma 4.3.3.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\langle\!|P|\!\rangle^{\circ}_{\rho} \equiv \langle\!|Q|\!\rangle^{\circ}_{\rho}$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

We now state our operational correspondence result:

**Theorem 4.3.4** (Operational Correspondence for $\langle\!|\cdot|\!\rangle^{\circ}_{\rho}$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{A}}$.

(1) If $P \longrightarrow P'$ then $\langle\!|P|\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{k} \langle\!|P'|\!\rangle^{\circ}_{\varepsilon}$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1) + \mathtt{Z}_{\mathtt{a}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$, it follows $k = 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{P}}(F[P_1]) + \mathtt{Z}_{\mathtt{a}}(F[P_1])$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$ processes $P_1, Q, P_2$ and names $t, u$.

(2) If $\langle\!|P|\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{n} R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^{*} P'$ and $R \longrightarrow^{*} \langle\!|P'|\!\rangle^{\circ}_{\varepsilon}$.

*Proof.* We consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases *a*), *b*) and *c*) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 4.3.3, Definition 4.3.2, and Lemma 3.2.9 (Page 47) that applies also for $\langle\!|\cdot|\!\rangle^{\circ}_{\rho}$.

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have that derivation corresponds to the derivation presented in (3.30) but here we use Definition 4.3.2 instead Definition 3.2.3. Therefore, the thesis holds with $k = 1$.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb}_{\mathtt{A}}(P_1)$, $m = \mathtt{ts}_{\mathtt{A}}(P_1)$ and $s = \mathtt{S}(P_1)$, and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We consider the following two cases and their derivation. Process $I^{(p)}_{t}(\langle\!|P_1|\!\rangle^{\circ}_{t,\rho'}, \langle\!|Q|\!\rangle^{\circ}_{\varepsilon})$ is presented in the Table 4.3.

(i) In this sub-case $n = 0$. Therefore, we have the following derivation:

$$
\begin{aligned}
\langle\!|P|\!\rangle^{\circ}_{\varepsilon} &\equiv \langle\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!\rangle^{\circ}_{\varepsilon} \\
&= \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C[t[P_1,Q]]|\!\rangle^{\circ}_{\rho} \mid \langle\!|D[\overline{t}.P_2]|\!\rangle^{\circ}_{\rho}\Big] \\
&= \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C|\!\rangle_{\rho}[\langle\!|t[P_1,Q]|\!\rangle^{\circ}_{\rho'}] \mid \langle\!|D|\!\rangle^{\circ}_{\rho}[\langle\!|\overline{t}.P_2|\!\rangle^{\circ}_{\rho''}]\Big] \\
&= \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C|\!\rangle^{\circ}_{\rho}\big[t\big[\langle\!|P|\!\rangle^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\langle\!|Q|\!\rangle^{\circ}_{\varepsilon}]) \\
&\qquad \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}\big] \mid \langle\!|D|\!\rangle^{\circ}_{\rho}[\overline{t}.h_t.\langle\!|P_2|\!\rangle^{\circ}_{\rho''}]\Big] \\
&\longrightarrow \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C|\!\rangle^{\circ}_{\rho}\big[I^{(1)}_{t}(\langle\!|P_1|\!\rangle^{\circ}_{t,\rho'}, \langle\!|Q|\!\rangle^{\circ}_{\varepsilon}) \mid \langle\!|D|\!\rangle^{\circ}_{\rho}\big[h_t.\langle\!|P_2|\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&\longrightarrow^{5} \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C|\!\rangle^{\circ}_{\rho}\big[\big[I^{(6)}_{t}(\langle\!|P_1|\!\rangle^{\circ}_{t,\rho'}, \langle\!|Q|\!\rangle^{\circ}_{\varepsilon}) \mid \langle\!|D|\!\rangle^{\circ}_{\rho}\big[h_t.\langle\!|P_2|\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&\longrightarrow \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C|\!\rangle^{\circ}_{\rho}\big[\langle\!|\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle|\!\rangle^{\circ}_{\rho'}\big] \mid \langle\!|D|\!\rangle^{\circ}_{\rho}\big[\langle\!|P_2|\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&= \langle\!|E|\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!|C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle]|\!\rangle^{\circ}_{\rho} \mid \langle\!|D[P_2]|\!\rangle^{\circ}_{\rho}\Big] \\
&= \langle\!|E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]|\!\rangle^{\circ}_{\varepsilon} \\
&\equiv \langle\!|P'|\!\rangle^{\circ}_{\varepsilon}
\end{aligned}
$$

Therefore, we can conclude that $\langle\!|P|\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{k} \langle\!|P'|\!\rangle^{\circ}_{\varepsilon}$ where $k = 7$.

(ii) In this sub-case $n > 0$ and $m > 0$. Therefore, we have the following derivation:

$$\begin{aligned}
\langle\!| P |\!\rangle^{\circ}_{\varepsilon} &\equiv \langle\!| E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]] |\!\rangle^{\circ}_{\varepsilon} \\
&= \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C[t[P_1,Q]] |\!\rangle^{\circ}_{\rho} \mid \langle\!| D[\overline{t}.P_2] |\!\rangle^{\circ}_{\rho}\Big] \\
&= \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C |\!\rangle_{\rho}[\langle\!| t[P_1,Q] |\!\rangle^{\circ}_{\rho'}] \mid \langle\!| D |\!\rangle^{\circ}_{\rho}[\langle\!| \overline{t}.P_2 |\!\rangle^{\circ}_{\rho''}]\Big] \\
&= \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\big[\langle\!| C |\!\rangle^{\circ}_{\rho}\Big[t\big[\langle\!| P |\!\rangle^{\circ}_{t,\rho}\big] \mid r_t.\,(\mathtt{extra}\{t, p_{t,\rho}, p_{\rho}\} \mid p_{\rho}[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}]) \\
&\qquad \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\}\Big] \mid \langle\!| D |\!\rangle^{\circ}_{\rho}[\overline{t}.h_t.\langle\!| P_2 |\!\rangle^{\circ}_{\rho''}]\big] \\
&\longrightarrow \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\circ}_{\rho}\big[I^{(1)}_{t}(\langle\!| P_1 |\!\rangle^{\circ}_{t,\rho'}, \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\big] \mid \langle\!| D |\!\rangle^{\circ}_{\rho}[\overline{t}.h_t.\langle\!| P_2 |\!\rangle^{\circ}_{\rho''}]\Big] \\
&\longrightarrow^{2+4m+s-n} \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\circ}_{\rho}\big[\big[I^{(3+4m+s-n)}_{t}(\langle\!| P_1 |\!\rangle^{\circ}_{t,\rho'}, \langle\!| Q |\!\rangle^{\circ}_{\varepsilon}) \mid \langle\!| D |\!\rangle^{\circ}_{\rho}\big[h_t.\langle\!| P_2 |\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&\longrightarrow^{4+n} \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\circ}_{\rho}\big[\big[I^{(7+4m+s)}_{t}(\langle\!| P_1 |\!\rangle^{\circ}_{t,\rho'}, \langle\!| Q |\!\rangle^{\circ}_{\varepsilon}) \mid \langle\!| D |\!\rangle^{\circ}_{\rho}\big[h_t.\langle\!| P_2 |\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&\longrightarrow \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\circ}_{\rho}\big[\langle\!| \mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle |\!\rangle^{\circ}_{\rho'}\big] \mid \langle\!| D |\!\rangle^{\circ}_{\rho}\big[\langle\!| P_2 |\!\rangle^{\circ}_{\rho''}\big]\Big] \\
&= \langle\!| E |\!\rangle^{\circ}_{\varepsilon}\Big[\langle\!| C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] |\!\rangle^{\circ}_{\rho} \mid \langle\!| D[P_2] |\!\rangle^{\circ}_{\rho}\Big] \\
&= \langle\!| E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]] |\!\rangle^{\circ}_{\varepsilon} \\
&\equiv \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}
\end{aligned}$$

The order/nature of these reduction steps is similar as in the proof of **Part (1) – Completeness** 1 for translation $\mathcal{C}_{\mathtt{A}}$ nto $\mathcal{S}$ (cf. explanation 1b). The difference exists in the synchronization on update prefix and location with name $z_t$. This extra step we need for the relocation of processes $p_{\rho'}[\cdot]$, which are enclosed on location $t$ out of it.

Therefore, we can conclude that $\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{k} \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}$ for $k = 8 + 4m + s$.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[t[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb}_{\mathtt{A}}(F[P_1])$, $m = \mathtt{ts}_{\mathtt{A}}(P_1)$, $s = \mathtt{S}(P_1)$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$. Then we consider two cases and have the following derivations.

(i) In this sub-case we consider that $n = 0$.

$$\begin{aligned}
\langle\!| P |\!\rangle^{\circ}_{\varepsilon} &\equiv \langle\!| C\big[u[F[\overline{u}.P_1],Q]\big] |\!\rangle^{\circ}_{\varepsilon} \\
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[\langle\!| u[F[\overline{u}.P_1],Q] |\!\rangle^{\circ}_{\rho}\big] \\
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[u\big[\langle\!| F[\overline{u}.P_1] |\!\rangle^{\circ}_{u,\rho}\big] \mid r_u.\,(\mathtt{extra}\{u, p_{u,\rho}, p_{\rho}\} \mid p_{\rho}[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}]) \\
&\qquad \mid u.u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}]\big]\big] \\
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[u\big[\langle\!| F |\!\rangle^{\circ}_{u,\rho}[\overline{u}.h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\{u, p_{u,\rho}, p_{\rho}\}) \mid p_{\rho}[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}] \\
&\qquad \mid u.u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}\big] \\
&\longrightarrow \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\Big[O^{(1)}_{u}(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\Big] \\
&\longrightarrow^{6} \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[O^{(7)}_{u}(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\big] \\
&\equiv \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[\langle\!| \mathsf{extr}_{\mathtt{A}}(F[P_1]) |\!\rangle^{\circ}_{\rho} \mid p_{\rho}[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}]\big] \\
&= \langle\!| C\big[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle\big] |\!\rangle^{\circ}_{\varepsilon} \\
&\equiv \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}
\end{aligned}$$

Therefore, we can conclude that $\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{k} \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}$ for $k = 8$.

(ii) In this sub-case we consider that $n > 0$ and $m \geq 0$.

$$\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \equiv \langle\!| C\big[u[F[\overline{u}.P_1],Q]\big] |\!\rangle^{\circ}_{\varepsilon}$$

$$
\begin{aligned}
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[\langle\!| u[F[\overline{u}.P_1],Q] |\!\rangle^{\circ}_{\rho}\big]\\
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[u\big[\langle\!| F[\overline{u}.P_1] |\!\rangle^{\circ}_{u,\rho}\big] \mid r_u.\,(\mathtt{extra}\{u, p_{u,\rho}, p_\rho\} \mid p_\rho[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}])\\
&\quad \mid u.u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}\big]\big]\\
&= \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[u\big[\langle\!| F |\!\rangle^{\circ}_{u,\rho}[\overline{u}.h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}]\big] \mid r_u.\,(\mathtt{extra}\{u, p_{u,\rho}, p_\rho\})\\
&\quad \mid p_\rho[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}] \mid u.u\{(Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\}\big]\\
\longrightarrow\ & \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\Big[O^{(1)}_u(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\Big]\\
\longrightarrow^{2+4m+s-n}\ & \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[O^{(3+4m+s-n)}_u(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\big]\\
\longrightarrow^{n+5}\ & \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[O^{(8+4m+s)}_u(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})\big]\\
&\equiv \langle\!| C |\!\rangle^{\circ}_{\varepsilon}\big[\langle\!| \mathsf{extr}_{\mathtt{A}}(F[P_1]) |\!\rangle^{\circ}_{\rho} \mid p_\rho[\langle\!| Q |\!\rangle^{\circ}_{\varepsilon}]\big]\\
&= \langle\!| C\big[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q \rangle\big] |\!\rangle^{\circ}_{\varepsilon}\\
&\equiv \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}
\end{aligned}
$$

Process $O^{(q)}_u(\langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}], \langle\!| Q |\!\rangle^{\circ}_{\varepsilon})$, where $q \in \{1, \ldots, 8+4m+s\}$, is as in Definition 4.3.6 (cf. Table 4.4). In this case, the role of function $\mathsf{ch}(u,\cdot)$ is central: indeed, $\mathsf{ch}(u, \langle\!| F |\!\rangle^{\circ}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\circ}_{\rho'}])$ provides the input $h_u$ which is necessary to achieve operational correspondence.
The order/nature/number of reduction steps can be explained as in Case b) above. We can then conclude that $\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \longrightarrow^k \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}$ for $k = 8+4m+s$.

(2) **Part (2) – Soundness:** The proof of soundness follows the explanation presented in Roadmap 3.2.3.2.5 and the same derivation that is presented in the proof of soundness for translation $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ (Item 2).

■

### 4.3.3 Comparing Subjective vs Objective update

In this subsection, we provide that subjective updates are better suited to encode compensation handling with aborting semantics than objective updates.

The following statement is a corollary of Theorem 4.3.4.

**Corollary 4.3.5.** Let $P$ be a well-formed process in $\mathcal{C}$. If $P \longrightarrow P'$ and $\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \longrightarrow^k \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}$ then:

b) if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ then $k \geq 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1) + \mathtt{Z}_{\mathtt{a}}(P_1)$,

c) if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q \rangle]$ then $k \geq 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{A}}(F[P_1]) + \mathtt{Z}_{\mathtt{a}}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

We then have the following theorem:

**Theorem 4.3.6.** The encoding $\langle\!| \cdot |\!\rangle_{\rho} : \mathcal{C}_{\mathtt{A}} \longrightarrow \mathcal{S}$ is as or more efficient than $\langle\!| \cdot |\!\rangle^{\circ}_{\rho} : \mathcal{C}_{\mathtt{A}} \longrightarrow \mathcal{O}$.

*Proof.* Let $P \longrightarrow P'$ and $\langle\!| P |\!\rangle^{\circ}_{\varepsilon} \longrightarrow^{k_2} \langle\!| P' |\!\rangle^{\circ}_{\varepsilon}$. By Theorem 3.4.6, there is $k_1$ such that $\langle\!| P |\!\rangle_{\varepsilon} \longrightarrow^{k_1} \langle\!| P' |\!\rangle_{\varepsilon}$ and

a) $k_1 = 1$ if $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$,

b) $k_1 = 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1)$ if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$,

c) $k_1 = 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts_A}(F[P_1])$ if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_A}(F[P_1]) \mid \langle Q \rangle]$.

By Theorem 4.3.4, Proposition 2.2.3 and Corollary 4.3.5 we have the following tree cases, for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet]$, processes $P_1, Q, P_2$ and names $t, u$:

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ and $k_2 \geq 1 = k_1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr_A}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ and $k_2 \geq 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts_A}(P_1) + \mathtt{Z_a}(P_1) \geq 7 + \mathtt{S}(P_1) + 4\,\mathtt{ts_A}(P_1) = k_1$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr_A}(F[P_1]) \mid \langle Q \rangle]$ and $k_2 \geq 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts_A}(F[P_1]) + \mathtt{Z_a}(F[P_1]) \geq 7 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts_A}(F[P_1]) = k_1$.

Thus, in all three cases $k_1 \leq k_2$; by Definition 4.1.7 we conclude that $\langle\!|\cdot|\!\rangle_{\varepsilon}$ is as or more efficient than $\langle\!|\cdot|\!\rangle^{\mathsf{o}}_{\varepsilon}$. ■

**Brief summary of the chapter**:
In this chapter, we have two main results: (i) we presented the *encodings* of calculi for compensable processes into the calculi of adaptable processes with the objective update (encodings $\mathcal{C}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{P}}$, $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{O}$); (ii) we exploit the correctness properties of encodings to distinguish between subjective and objective updates in calculi for concurrency. We introduce an encodability criterion called *efficiency*. Since subjective updates induce tighter operational correspondences, we can formally declare that subjective updates are more suited to encode compensation handling than objective updates.

In the next chapter, we introduce dynamic compensation processes. We proved encodings of dynamic compensation processes into adaptable processes with subjective updates (i.e., encodings $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}^{\lambda}_{\mathtt{P}}$, and $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{S}$).

# CHAPTER 5

# Encoding Dynamic Compensation Processes into Adaptable Processes with Subjective Update

Previous chapters of the thesis have dealt with encodings of the calculus of compensable processes with static compensations under discarding, preserving, and aborting semantics into the calculus of adaptable processes with *subjective* mobility. In this chapter, we discuss how to extend the previous encodings to account for compensation updates $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P$. Therefore, we developed encodings of calculus of compensable processes with dynamic compensations, denoted $\mathcal{C}^{\lambda}$, under discarding, preserving, and aborting semantics into calculus of adaptable processes with *subjective* mobility. In the following is given a brief structure of the chapter:

**Section 5.1** introduces preliminaries for *encodings of* $\mathcal{C}^{\lambda}$ *into* $\mathcal{A}$. More specifically, the section contains an extension of syntax for compensable processes presented (cf. Section 2.2.1) for a dynamic update. We also discuss the appropriate extension of well-formed compensable processes.

**Section 3.2** presents the translation of $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{S}$. Then the formal definition of the encoding follows. We prove that the encoding satisfies *compositionality*, *name invariance* and *operational correspondence (completeness and soundness)*.

**Section 5.3** presents the translation of $\mathcal{C}^{\lambda}_{\mathtt{P}}$ into $\mathcal{S}$. We introduce the formal definition of encoding. We prove that the encoding satisfies *name invariance* and *operational correspondence (completeness and soundness)*.

**Section 5.4** presents the translation of $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{S}$. Then the formal definition of the encoding follows. We prove that translation satisfies *compositionality*, *name invariance* and *operational correspondence (completeness and soundness)*.

## 5.1 Compensable Processes with Compensation Update

Compensable processes, which realize *general dynamic recovery*, extend static recovery processes presented in Paragraph 2.2.1.2.1. In the following, we present compensable processes with compensation update.

#### 5.1.0.0.1 Syntax.

The calculus of *compensable processes* considers prefixes $\pi$ and processes $P, Q, \ldots$ defined as:

$$\pi \quad ::= \quad a \;\big|\; \overline{x}$$

$$P, Q \quad ::= \quad \textit{Static recovery processes} \;\Big|\; Y \;\Big|\; \texttt{inst}\lfloor \lambda Y.R \rfloor.P$$

The main difference comparing with static recovery is that in compensable processes, the body $P$ of transaction $t[P,Q]$ can update the compensation $Q$. Compensation update is performed by a new operator $\texttt{inst}\lfloor \lambda Y.R \rfloor.P'$, where function $\lambda Y.R$ is the compensation update ($Y$ can occur inside $R$). Applying such a compensation update to compensation $Q$ produces a new compensation $R\{^Q/_Y\}$ after the internal transition. Note that $R$ may not occur at all in the resulting compensation, and it may also occur more than once. For instance, $\lambda Y.\mathbf{0}$ deletes the current compensation. A compensation update has priority over other transitions; that is, if process $P$ in transaction $t[P,Q]$ has a compensation update at top-level then it will be performed before any change of the current state.

#### 5.1.0.0.2 Operational Semantics

Following [29, 30], the semantics of compensable processes with compensation update is given in terms of an LTS. The LTS is parametric in an extraction function, which is defined as in Figure 2.2 and additionally extended with the following rule:

$$\mathsf{extr}(\texttt{inst}\lfloor \lambda Y.Q \rfloor.P) = \mathbf{0}. \tag{5.1}$$

As before, error notifications can be *internal* or *external* to the transaction. The rules (L-Out), (L-In), (L-Rep), (L-Par1), (L-Res) and (L-Block) are as in Figure 2.3. The other LTS rules (L-Scope-Out), (L-Recover-Out), (L-Recover-In), (L-Inst) and (L-Scope-Close) are presented in Figure 5.1. The final two rules are peculiar of processes with dynamic compensations. We comment briefly on each of them:

- Rules (L-Rec-Out) and (L-Rec-In) have the explanation as in Figure 2.3, with the addition of a premise noComp(P);

- Rule (L-Scope-Out) allows the default activity $P$ of a transaction to progress, provided that the performed action is not a compensation update and that there is no pending compensation update to be executed. The latter is ensured by condition noComp(P), this condition guarantees that there is no pending compensation update in $P$ and is defined in Definition 5.1.1). The condition is true if and only if process $P$ does not have compensation update which waits for execution. This means that a compensation update has priority over other transitions;

- Rule (L-Inst) performs a compensation update;

- Rule (L-Scope-Close) updates the compensation of a transaction.

**Definition 5.1.1** (noComp(•) predicate)**.** The predicate noComp(P) that verifies that there are no pending compensation updates inside $P$ is true in the cases specified in Figure 5.2 and false otherwise.

The following proposition and Proposition 2.2.3 are key to operational correspondence statements. First, we present one auxiliary result that is needed for the proof of the Proposition.

**Lemma 5.1.1.** Let $P$ be a compensable process. If $P \xrightarrow{\lambda Y.R} P'$ then $P \equiv H[\texttt{inst}\lfloor \lambda Y.R \rfloor.P_1]$ and $P' = H[P_1]$ for some evaluation context $H[\bullet]$ and processes $P_1$ and $R$.

*Proof.* The proof is by induction on the derivation of $P \xrightarrow{\alpha} P'$. ■

(L-SCOPE-OUT)
$$\frac{P \xrightarrow{\alpha} P' \quad \alpha \neq \lambda Y.Q \quad \mathsf{noComp}(\mathsf{P})}{t[P,Q] \xrightarrow{\alpha} t[P',Q]}$$

(L-RECOVER-OUT)
$$\frac{\mathsf{noComp}(\mathsf{P})}{t[P,Q] \xrightarrow{t} \mathsf{extr}(P) \mid \langle Q \rangle}$$

(L-RECOVER-IN)
$$\frac{P \xrightarrow{\overline{t}} P' \quad \mathsf{noComp}(\mathsf{P})}{t[P,Q] \xrightarrow{\tau} \mathsf{extr}(P') \mid \langle Q \rangle}$$

(L-INST)
$$\mathtt{inst}\lfloor \lambda Y.Q \rfloor.P \xrightarrow{\lambda Y.Q} P$$

(L-SCOPE-CLOSE)
$$\frac{P \xrightarrow{\lambda Y.R} P'}{t[P,Q] \xrightarrow{\tau} t[P', R\{Q/Y\}]}$$

Figure 5.1: LTS for compensable processes with compensation update.

| | |
|---|---|
| $\mathsf{noComp}(\mathbf{0})$ | $\mathsf{noComp}(\langle \mathsf{P} \rangle)$ if $\mathsf{noComp}(\mathsf{P})$ |
| $\mathsf{noComp}(\pi.\mathsf{P})$ | $\mathsf{noComp}(\mathsf{t}[\mathsf{P};\mathsf{Q}])$ if $\mathsf{noComp}(\mathsf{P})$ |
| $\mathsf{noComp}(!\mathsf{P})$ | $\mathsf{noComp}(\mathsf{P}\vert\mathsf{Q})$ if $\mathsf{noComp}(\mathsf{P})$ and $\mathsf{noComp}(\mathsf{Q})$ |
| | $\mathsf{noComp}((\nu \mathsf{a})\mathsf{P})$ if $\mathsf{noComp}(\mathsf{P})$ |

Figure 5.2: No pending compensation update predicate.

**Proposition 5.1.2.** Let $P$ be a compensable process. If $P \xrightarrow{\tau} P'$ then $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1],Q]$ and $P' \equiv C[s[H[P_1], R\{Q/Y\}]]$, for some contexts $C, H$ processes $R, P_1, Q$ and name $s$.

*Proof.* The proof proceeds by induction on the inference of $P \xrightarrow{\tau} P'$. We will show that the proposition is true for the base cases, whereas the inductive step follows directly. By LTS (cf. Figure 5.1), in accordance with the Rule (L-SCOPE-CLOSE) we have the following: if $P \equiv s[P_1',Q]$ and $P_1' \xrightarrow{\lambda Y.R} P_1$ then $P' \equiv s[P_1,R\{Q/Y\}]$ and by Lemma 5.1.1, we conclude that $P_1' \equiv H[\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1]$ and $P_1 \equiv H[P_1]$ for some process $P_1$. ■

### 5.1.1 Well-formed Compensable Processes

For compensable processes with compensation updates, the definition of well-formed processes must account for compensation updates. Therefore, in the following, we present revised well-formed compensable processes presented in Section 2.2.2.

**Remark 5.1.3** (Well-Formed Processes with Dynamic Recovery)**.** We revisit the notion of well-formed compensable processes, now with compensation updates. An example of a process that is not well-formed is the following:

$$\times \qquad P_1 = t_1[\mathtt{inst}\lfloor \lambda Y.t_2[X,a] \rfloor.b,c] \mid \overline{t_1} \mid \overline{t_2} \xrightarrow{\tau} t_1[b,t_2[c,a]] \mid \overline{t_1} \mid \overline{t_2}. \tag{5.2}$$

Process $P_1$ has concurrent error notifications (on $t_1$ and $t_2$), and a pair of nested transactions (i.e., $(t_1, t_2)$) that is hard to capture properly in the representation that we shall give in terms of adaptable processes. In contrast, we would like to consider as well-formed the following processes (where $t_1 \neq t_2$):

$$\checkmark \qquad P = t_1[\mathtt{inst}\lfloor \lambda Y.t_2[X,a] \rfloor.b,c] \mid \overline{t_1}.\overline{t_2} \xrightarrow{\tau} t_1[b,t_2[c,a]] \mid \overline{t_1}.\overline{t_2}. \tag{5.3}$$

For $\mathcal{C}^\lambda$ processes, the relation for well-formed compensable processes (cf. Figure 2.4) should be extended with the following rule:

$$\text{(W-Inst)}\quad \frac{\Gamma_1;\Delta_1 \vdash_{\gamma_1;\delta_1;p_1} P \quad \Gamma_2;\Delta_2 \vdash_{\gamma_2;\delta_2;p_2} Q \quad \Gamma_3;\Delta_3 \vdash_{\gamma_3;\delta_3;p_3} R \qquad \Gamma^s \cap \Delta^t = \emptyset}{\Gamma;\Delta \vdash_{\bigcup_{i=1}^{3}\gamma_i;\bigcup_{i=1}^{3}\delta_i;\bigvee_{i=1}^{3}p_i} t[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P, Q]}$$

where

$$\Gamma(\gamma_1,\gamma_2,\gamma_3) = \{(t',t'') : t' \in \gamma_1 \wedge t'' \in \gamma_2 \cup \gamma_3\}\ ,\ \ \Delta(t,\delta) = \{(t,t') : t' \in \delta\}, \tag{5.4}$$

$\mathcal{P}(P) = (\Gamma_1,\Delta_1,\gamma_1,\delta_1)$, $\mathcal{P}(Q) = (\Gamma_2,\Delta_2,\gamma_2,\delta_2)$, $\mathcal{P}(R) = (\Gamma_3,\Delta_3,\gamma_3,\delta_3)$ and

$$\begin{aligned} f_\lambda(\mathcal{P}(P),\mathcal{P}(Q),\mathcal{P}(R)) &= \Big(\bigcup_{i=1}^{3}\Gamma_i \cup \Gamma(\gamma_1,\gamma_2,\gamma_3), \bigcup_{i=1}^{3}\Delta_i \cup \Delta(t,\delta_1\cup\delta_2\cup\delta_3\cup\gamma_1\cup\gamma_2\cup\gamma_3)\Big) \\ &= (\Gamma,\Delta) \end{aligned} \tag{5.5}$$

Rule (W-Inst) specifies the conditions for $t[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P, Q]$ to be well-formed; it relies on the key ideas of the Rule (W-Trans). Therefore, $\delta = \{t\}$. The set of pairs of parallel failure signals is the union of the respective sets for $P$, $Q$ and $R$ and the set whose elements are pairs of failure signals; in the pair, one element belongs to the set of failure signals of $P$, the second element is from the union of sets of failure signals of $Q$ and $R$. This extension with $\Gamma(\gamma_1,\gamma_2,\gamma_3)$ is necessary for $t[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P, Q]$, because $P$ may contain protected blocks which will be composed in parallel with $R\{^Q/_Y\}$ in case of a failure signal. The set of pairs of nested transactions is obtained from those for $P$, $Q$, and $R$ also considering further pairs as specified by $\Delta(t,\delta_1\cup\delta_2\cup\delta_3\cup\gamma_1\cup\gamma_2\cup\gamma_3)$ (cf. (5.4)). The rule additionally enforces that the sets of parallel failure signals and nested transaction names in the parallel composition are disjoint. For example, for process (5.3) above we can derive:

$$\emptyset;\{(t_1,t_2)\} \vdash_{\{t_1,t_2\};\{t_1\};\bot} t_1[\mathtt{inst}\lfloor \lambda Y.t_2[X,a]\rfloor.b,c] \mid \overline{t_1}.\overline{t_2}.$$

In contrast, process (5.2) does not satisfy the predicate, since its sets of pairs of parallel failure signals and nested transaction names are not disjoint: they are both equal to $\{(t_1,t_2)\}$.

Building upon the syntax defined in Section 2.2.1, we shall write $\mathcal{C}^\lambda_{\mathtt{D}}$, $\mathcal{C}^\lambda_{\mathtt{P}}$, $\mathcal{C}^\lambda_{\mathtt{A}}$ to denote compensable processes with compensation updates. Also, translations of $\mathcal{C}^\lambda_{\mathtt{D}}$, $\mathcal{C}^\lambda_{\mathtt{P}}$, $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$ will be defined for well-formed compensable processes.

## 5.2 Translating $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$

The translation $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$, denoted $[\![\cdot]\!]^\lambda_\rho$, extends the key ideas of the encoding $[\![\cdot]\!]_\rho$ (cf. Section 3.2).

**Remark 5.2.1** (Reserved names)**.** The translation requires sets of *reserved names* and therefore we need to revised Definition 3.1.2 as in the following:

(i) the set of *reserved location names* $\mathcal{N}^r_l$ is unchanged and,

(ii) the set of *reserved synchronization names* is extended such that

$$\mathcal{N}^r_s = \{h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x \mid x \in \mathcal{N}_t\}.$$

**Remark 5.2.2.** The function for determining the number of locations (cf. Definition 3.2.1) should be extended with the following:

$$\mathtt{nl}(l, \mathtt{inst}\lfloor \lambda Y.R \rfloor.P) = \mathtt{nl}(l, P), \text{ and} \tag{5.6}$$

$$\mathsf{ch}(t, \mathtt{inst}\lfloor \lambda Y.R \rfloor.P) = \mathsf{ch}(t, P) \tag{5.7}$$

We will use process $\mathtt{outd^s}$ as defined for $[\![\cdot]\!]_\rho$ (cf. (3.2)). We need some additional auxiliary process.

**Definition 5.2.1** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extrd}\langle\!\langle t, l_1, l_2 \rangle\!\rangle$ to stand for the following (subjective) update prefix:

$$\mathtt{extrd}\langle\!\langle t, l_1, l_2 \rangle\!\rangle = t\langle\!\langle (Y).\, \big(t[Y] \mid \mathsf{ch}(t, Y) \mid \mathtt{outd^s}(l_1, l_2, \mathtt{nl}(l, Y), \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{h_t})\big) \rangle\!\rangle \tag{5.8}$$

The intuition for the process $\mathtt{extrd}\langle\!\langle t, l_1, l_2 \rangle\!\rangle$ is the same as in the translation of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ with static recovery (cf. Definition 3.2.2). The only difference is in the third parameter for process $\mathtt{outd^s}$, which enables us to have a controlled execution of adaptable processes, which is important to establish operational correspondence. The prefix $t\langle\!\langle \dagger \rangle\!\rangle$ and name $h_t$ have the same roles as in $[\![\cdot]\!]_\rho$. The differences concern names $m_t$ and $k_t$: while name $m_t$ ensures that every translation of compensation $Q$ is updated if the translation of compensation update exists, name $k_t$ controls the execution of failure signals.

Using well-formed compesable processes (cf. Section 5.1.1), the translation of $\mathcal{C}_{\mathtt{D}}^{\lambda}$ into $\mathcal{S}$ extends Definition 3.2.3 (see Page 39) as follows:

**Definition 5.2.2** (Translating $\mathcal{C}_{\mathtt{D}}^{\lambda}$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with dynamic recovery into (subjective) adaptable processes as a tuple $([\![\cdot]\!]^{\lambda}_\rho, \varphi_{[\![\cdot]\!]^{\lambda}_\rho})$ where:

(a) The renaming policy

$$\varphi_{[\![\cdot]\!]^{\lambda}_\rho}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t. \end{cases}$$

(b) The translation $[\![\cdot]\!]^{\lambda}_\rho : \mathcal{C}_{\mathtt{D}}^{\lambda} \to \mathcal{S}$ is as in Figure 5.3.

Key elements in Figure 5.3 are the translations of $t[P,Q]$ and $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1$, which are closely related to each other. Indeed, these translations share location names $u_t$, $v_t$, and $e_t$ (as well as names $f_t$ and $g_t$) in order to account for the possible replacement of $Q$ in $t[P,Q]$ with $R$ in $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1$, using updates. As stated earlier, $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P$ produces a new compensation behavior $R\{Q/Y\}$ after an internal transition.

### 5.2.1 Translation Correctness

We now establish that the translation $[\![\cdot]\!]^{\lambda}_\rho$ is a valid encoding. To this end, we address the three criteria in Definition 2.3.5: compositionality, name invariance, and operational correspondence. Other criteria have been left as a research topic for future work.

#### 5.2.1.1 Structural Criteria

We prove the two criteria, following the order in which they were introduced in Definition 2.3.5, i.e., compositionality and name invariance.

$$
\begin{aligned}
[\![\langle P\rangle]\!]^\lambda_\rho &= p_\rho\big[[\![P]\!]^\lambda_\varepsilon\big]\\
[\![t[P,Q]]\!]^\lambda_\rho &= t\Big[[\![P]\!]^\lambda_{t,\rho}\Big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle\\
&\qquad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big)\\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^\lambda_\varepsilon] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
[\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P]\!]^\lambda_{t,\rho} &= u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[[\![R]\!]^\lambda_\varepsilon]\\
&\qquad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\![P]\!]^\lambda_{t,\rho}\\
[\![\bar{t}.P]\!]^\lambda_\rho &= \bar{t}.h_t.[\![P]\!]^\lambda_\rho\\
[\![Y]\!]^\lambda_\rho &= Y\\
[\![a.P]\!]^\lambda_\rho &= a.[\![P]\!]^\lambda_\rho\\
[\![\overline{a}.P]\!]^\lambda_\rho &= \overline{a}.[\![P]\!]^\lambda_\rho\\
[\![\mathbf{0}]\!]^\lambda_\rho &= \mathbf{0}\\
[\![(\nu x)P]\!]^\lambda_\rho &= (\nu x)[\![P]\!]^\lambda_\rho\\
[\![P_1 \mid P_2]\!]^\lambda_\rho &= [\![P_1]\!]^\lambda_\rho \mid [\![P_2]\!]^\lambda_\rho\\
[\![!\pi.P]\!]^\lambda_\rho &= ![\![\pi.P]\!]^\lambda_\rho
\end{aligned}
$$

Figure 5.3: Translating $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$.

#### 5.2.1.1.1 Compositionality

As previously stated, the compositionality criterion states that a composite term's translation must be defined in terms of its subterms' translations. To mediate between these translations of subterms, we define a *context* for each process operator, which depends on free names of the subterms:

**Definition 5.2.3** (Compositional context for $\mathcal{C}^\lambda_{\mathtt{D}}$)**.** For all process operator from $\mathcal{C}^\lambda_{\mathtt{D}}$, instead transaction we define a compositional context in $\mathcal{S}$ as in Definition 3.2.4. For transaction and compensation update compositional contexts are as follows:

$$
\begin{aligned}
C_{t[,],\rho}[\bullet_1,\bullet_2] &= t\Big[[\bullet_1]\Big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big)\\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\bullet_2]] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
C_{\mathtt{inst},\rho}[\bullet_1,\bullet_2] &= u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[[\bullet_1]] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\bullet_2]\\
C_Y[\bullet_1] &= [\bullet_1]
\end{aligned}
$$

Using this definition, we may now state the following result:

**Theorem 5.2.3** (Compositionality for $[\![\cdot]\!]^\lambda_\rho$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}^\lambda_{\mathtt{D}}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$
\begin{array}{lll}
[\![\langle P\rangle]\!]^\lambda_\rho = C_{\langle\rangle,\rho}\left[[\![P]\!]^\lambda_\varepsilon\right] & [\![t[P,Q]]\!]^\lambda_\rho = C_{t[,],\rho}\left[[\![P]\!]^\lambda_{t,\rho}, [\![Q]\!]^\lambda_\varepsilon\right] & [\![P \mid Q]\!]^\lambda_\rho = C_{\mid}\left[[\![P]\!]^\lambda_\rho, [\![Q]\!]^\lambda_\rho\right]\\
[\![a.P]\!]^\lambda_\rho = C_{a.}\left[[\![P]\!]^\lambda_\rho\right] & [\![\bar{t}.P]\!]^\lambda_\rho = C_{\bar{t}.}\left[[\![P]\!]^\lambda_\rho\right] & [\![(\nu x)P)]\!]^\lambda_\rho = C_{(\nu x)}\left[[\![P]\!]^\lambda_\rho\right]\\
[\![\overline{a}.P]\!]^\lambda_\rho = C_{\overline{a}.}\left[[\![P]\!]^\lambda_\rho\right] & [\![!\pi.P]\!]^\lambda_\rho = C_{!\pi.}\left[[\![P]\!]^\lambda_\rho\right] & \\
[\![Y]\!]^\lambda_\rho = C_Y\left[[\![Y]\!]^\lambda_\rho\right] & [\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P]\!]^\lambda_{t,\rho} = C_{\mathtt{inst},\rho}[[\![R]\!]^\lambda_\varepsilon, [\![P]\!]^\lambda_{t,\rho}] &
\end{array}
$$

*Proof.* Follows directly from the definition of contexts (cf. Definition 6.1.3) and from the definition of $[\![\cdot]\!]^{\lambda}_{\rho} : \mathcal{C}^{\lambda}_{\mathrm{D}} \to \mathcal{S}$ (cf. Figure 5.3). Therefore, considering Definition 6.1.3 and Figure 5.3 we present derivation for: transaction, compensation update, process variable and all well-formed compensable processes $P, Q$ and $R$. The other operators have the same derivation as in the proof of Theorem 3.2.2. Therefore, the following holds:

$$\begin{aligned}
[\![t[P,Q]]\!]^{\lambda}_{\rho} &= C_{t[,],\rho}\left[[\![P]\!]^{\lambda}_{t,\rho}, [\![Q]\!]^{\lambda}_{\varepsilon}\right] \\
&= t\left[[\![P]\!]^{\lambda}_{t,\rho}\right] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_{\rho}\rangle\!\rangle \mid m_t.p_{\rho}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big) \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\
[\![\mathtt{inst}\lfloor \lambda Y.R\rfloor.P]\!]^{\lambda}_{t,\rho} &= C_{\mathtt{inst},\rho}[[\![R]\!]^{\lambda}_{\varepsilon}, [\![P]\!]^{\lambda}_{t,\rho}] \\
&= u_t\left[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[[\![R]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\right] \mid [\![P]\!]^{\lambda}_{t,\rho} \\
[\![Y]\!]^{\lambda}_{\rho} &= C_Y\left[[\![Y]\!]^{\lambda}_{\rho}\right] = [\![Y]\!]^{\lambda}_{\rho}
\end{aligned}$$

■

#### 5.2.1.1.2 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 3.2.3 (a).

**Theorem 5.2.4** (Name invariance for $[\![\cdot]\!]^{\lambda}_{\rho}$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{[\![\cdot]\!]^{\lambda}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]^{\lambda}_{\rho}}(x)\}$, and

(ii) $[\![\sigma(P)]\!]^{\lambda}_{\sigma(\rho)} = \sigma'([\![P]\!]^{\lambda}_{\rho})$.

*Proof.* We define the substitution $\sigma'$ as follows:

$$\sigma'(x) = \begin{cases}
\sigma(x) & \text{if } x = a \text{ or } x = t \\
h_{\sigma(t)} & \text{if } x = h_t \\
m_{\sigma(t)} & \text{if } x = m_t \\
k_{\sigma(t)} & \text{if } x = k_t \\
u_{\sigma(t)} & \text{if } x = u_t \\
v_{\sigma(t)} & \text{if } x = v_t \\
e_{\sigma(t)} & \text{if } x = e_t \\
g_{\sigma(t)} & \text{if } x = g_t \\
f_{\sigma(t)} & \text{if } x = f_t \\
p_{\sigma(\rho)} & \text{if } x = p_{\rho}.
\end{cases} \tag{5.9}$$

Now we provide proofs for (i) and (ii):

(i) The poof uses 5.9 and has the same derivation as the proof of Theorem 3.2.4 (i).

(ii) The proof proceeds by structural induction on $P$. In the following, given a name $x$, a path $\rho$, and process $P$, we write $\sigma x$, $\sigma\rho$, and $\sigma P$ to stand for $\sigma(x)$, $\sigma(\rho)$, and $\sigma(P)$, respectively.

***Base case:*** The statement holds for $P = \mathbf{0}$: $[\![\sigma(\mathbf{0})]\!]^{\lambda}_{\sigma\rho} = \sigma'([\![\mathbf{0}]\!]^{\lambda}_{\rho}) \Leftrightarrow \mathbf{0} = \mathbf{0}$.

***Inductive step:*** There are seven cases, but we show only the case for *transaction scope* and *compensation update*. Proof for all other cases are similar as in the proof of Theorem 3.2.4.

- ***Case*** $P = t[P_1,Q_1]$: We first apply the substitution $\sigma$ on process $P$:

$$[\![\sigma(t[P_1,Q_1])]\!]^{\lambda}_{\sigma\rho} = [\![\sigma t[\sigma(P_1),\sigma(Q_1)]]\!]^{\lambda}_{\sigma\rho}.$$

By expanding the definition of the translation in Definition 5.2.2, we have:

$$\begin{aligned}[\![\sigma(t[P_1,Q_1])]\!]^{\lambda}_{\sigma\rho} &= \sigma t\big[[\![\sigma(P_1)]\!]^{\lambda}_{\sigma t,\sigma\rho}\big] \mid \sigma t.\big(\mathtt{extrd}\langle\!\langle \sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\!\rangle \\ &\mid m_{\sigma t}.p_{\sigma\rho}\big[v_{\sigma t}\langle\!\langle (X).(X \mid u_{\sigma t}[\overline{f_{\sigma t}}.\overline{g_{\sigma t}}.k_{\sigma t}])\rangle\!\rangle\big]\big) \\ &\mid v_{\sigma t}\big[u_{\sigma t}\langle\!\langle (Z).(Z \mid e_{\sigma t}[[\![Q_1]\!]^{\lambda}_{\varepsilon}] \mid f_{\sigma t}.e_{\sigma t}\langle\!\langle (X).X\rangle\!\rangle.g_{\sigma t})\rangle\!\rangle\big]\end{aligned}$$

By induction hypothesis it follows:

$$\begin{aligned}[\![\sigma(t[P_1,Q_1])]\!]^{\lambda}_{\sigma\rho} &= \sigma t\Big[\sigma'([\![P]\!]^{\lambda}_{t,\rho})\Big] \mid \sigma t.\big(\mathtt{extrd}\langle\!\langle \sigma t, p_{\sigma t,\sigma\rho}, p_{\sigma\rho}\rangle\!\rangle \\ &\mid m_{\sigma t}.p_{\sigma\rho}\big[v_{\sigma t}\langle\!\langle (X).(X \mid u_{\sigma t}[\overline{f_{\sigma t}}.\overline{g_{\sigma t}}.k_{\sigma t}])\rangle\!\rangle\big]\big) \\ &\mid v_t\big[u_{\sigma t}\langle\!\langle (Z).(Z \mid e_{\sigma t}[\sigma'([\![Q_1]\!]^{\lambda}_{\varepsilon})] \mid f_{\sigma t}.e_{\sigma t}\langle\!\langle (X).X\rangle\!\rangle.g_{\sigma t})\rangle\!\rangle\big]\end{aligned} \tag{5.10}$$

On the other side, when we apply definition of substitution $\sigma'$ on $[\![P]\!]^{\lambda}_{\rho}$ the following holds:

$$\begin{aligned}\sigma'\Big([\![t[P_1,Q_1]]\!]^{\lambda}_{\rho}\Big) &= \sigma'\Big(t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_{\rho}\rangle\!\rangle \\ &\mid m_t.p_{\rho}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big) \\ &\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big) \\ &= \sigma' t\big[\sigma'\Big([\![P_1]\!]^{\lambda}_{t,\rho}\Big)\big] \mid \sigma' t.\big(\mathtt{extrd}\langle\!\langle \sigma' t, p_{\sigma' t,\sigma'\rho}, p_{\sigma'\rho}\rangle\!\rangle \\ &\mid m_{\sigma' t}.p_{\sigma'\rho}\big[v_{\sigma' t}\langle\!\langle (X).(X \mid u_{\sigma' t}[\overline{f_{\sigma' t}}.\overline{g_{\sigma' t}}.k_{\sigma' t}])\rangle\!\rangle\big]\big) \\ &\mid v_{\sigma' t}\big[u_{\sigma' t}\langle\!\langle (Z).(Z \mid e_{\sigma' t}[\sigma'([\![Q]\!]^{\lambda}_{\varepsilon})] \mid f_{\sigma' t}.e_{\sigma' t}\langle\!\langle (X).X\rangle\!\rangle.g_{\sigma' t})\rangle\!\rangle\big]\big).\end{aligned} \tag{5.11}$$

Given that it is valid $\sigma'(t) = \sigma(t)$ (cf. (5.9)), it is easy to conclude that (5.10) is equal to (5.11).

- ***Case*** $P = \mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1$: We apply substitution $\sigma$ on process $P$:

$$[\![\sigma(\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1)]\!]^{\lambda}_{\sigma t,\sigma\rho} = [\![\mathtt{inst}\lfloor \lambda Y.\sigma R\rfloor.\sigma P_1]\!]^{\lambda}_{\sigma t,\sigma\rho}$$

By Definition 5.2.2

$$\begin{aligned}[\![\sigma(\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1)]\!]^{\lambda}_{\sigma t,\sigma\rho} = u_{\sigma t}\Big[&e_{\sigma t}\langle\!\langle (Y).(\overline{g_{\sigma t}}.u_{\sigma t}\langle\!\langle (Z).(Z \mid e_{\sigma t}[\sigma([\![R]\!]^{\lambda}_{\varepsilon})] \\ &\mid f_{\sigma t}.e_{\sigma t}\langle\!\langle (X).X\rangle\!\rangle.g_{\sigma t})\rangle\!\rangle)\rangle\!\rangle.(\overline{f_{\sigma t}}.e_{\sigma t}[\mathbf{0}])\Big] \mid [\![\sigma(P_1)]\!]^{\lambda}_{\sigma t,\sigma\rho}\end{aligned}$$

and by induction hypothesis:

$$\begin{aligned}[\![\sigma(\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1)]\!]^{\lambda}_{\sigma\rho} = u_{\sigma t}\Big[&e_{\sigma t}\langle\!\langle (Y).(\overline{g_{\sigma t}}.u_{\sigma t}\langle\!\langle (Z).(Z \mid e_{\sigma t}[\sigma'([\![R]\!]^{\lambda}_{\varepsilon})] \\ &\mid f_{\sigma t}.e_{\sigma t}\langle\!\langle (X).X\rangle\!\rangle.g_{\sigma t})\rangle\!\rangle)\rangle\!\rangle.(\overline{f_{\sigma t}}.e_{\sigma t}[\mathbf{0}])\Big] \mid \sigma'([\![P_1]\!]^{\lambda}_{\sigma t,\sigma\rho})\end{aligned} \tag{5.12}$$

On the other side, when we apply substitution $\sigma'$ on $[\![P]\!]^{\lambda}_{\rho}$ the following holds:

$$\begin{aligned}\sigma'([\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1]\!]^{\lambda}_{t,\rho}) &= \sigma'(u_t\Big[e_t\langle\!\langle(Y).(\overline{g_t}.u_t\langle\!\langle(Z).(Z \mid e_t[[\![R]\!]^{\lambda}_{\varepsilon}] \\ &\quad \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\![P]\!]^{\lambda}_{t,\rho}) \\ &= u_{\sigma' t}\Big[e_{\sigma' t}\langle\!\langle(Y).(\overline{g_{\sigma' t}}.u_{\sigma' t}\langle\!\langle(Z).(Z \mid e_{\sigma' t}[\sigma'([\![R]\!]^{\lambda}_{\varepsilon})] \\ &\quad \mid f_{\sigma' t}.e_{\sigma' t}\langle\!\langle(X).X\rangle\!\rangle.g_{\sigma' t})\rangle\!\rangle)\rangle\!\rangle.(\overline{f}_{\sigma' t}.e_{\sigma' t}[\mathbf{0}])\Big] \mid \sigma'([\![P]\!]^{\lambda}_{t,\rho})\end{aligned} \tag{5.13}$$

Based on definition of the function $\sigma'$, i.e. $\sigma'(p_\rho) = p_{\sigma(\rho)}$ and $\sigma'(t) = \sigma(t)$ (cf. (5.9)), it is easy to conclude that (5.12) is equal to (5.13).

■

#### 5.2.1.2 Semantic Criteria

In this subsection we prove that translation $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{S}$ satisfies operational correspondence (completeness and soundness). The other two criteria, divergence reflection and success sensitiveness, are left for future work.

##### 5.2.1.2.1 Operational Correspondence

The following statement formalizes the encoding of process $R\{{}^{Q}\!/_{Y}\}$, and holds also for $(\!|\cdot|\!)^{\lambda}_{\rho}$ and $\langle\!|\cdot|\!\rangle^{\lambda}_{\rho}$:

**Lemma 5.2.5.** Suppose $R$ is a well-formed compensable process. Then $[\![R\{{}^{Q}\!/_{Y}\}]\!]^{\lambda}_{\rho} = [\![R]\!]^{\lambda}_{\rho}\{{}^{[\![Q]\!]^{\lambda}_{\rho}}\!/_{X}\}$.

We are likewise interested in giving a precise account of the number of calculation steps for establishing operational correspondence in encoding dynamic compensable processes into adaptable processes. We claim that subjective updates are more efficient than objective updates, and this is to support it.

The following definition formalizes all possible forms of the process $I^{(p)}_t([\![P]\!]^{\lambda}_{t,\rho}, [\![Q]\!]^{\lambda}_{\varepsilon})$. Also, due to the simplicity of writing for the process $I^{(p)}_t([\![P]\!]^{\lambda}_{t,\rho}, [\![Q]\!]^{\lambda}_{\varepsilon})$, we will use the abbreviation $I^{(p)}_t$ in all places where we do not violate the rationing of the content.

**Definition 5.2.4.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I^{(p)}_t([\![P]\!]^{\lambda}_{t,\rho}, [\![Q]\!]^{\lambda}_{\varepsilon})$ depending on $n = \mathtt{nl}(p_{t,\rho}, [\![P]\!]^{\lambda}_{t,\rho})$:

1. if $n = 0$ then $p \in \{1, \dots, 10\}$ and

$$\begin{aligned}I^{(1)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\ &\quad \mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\ &\equiv t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid t\langle\!\langle(Y).\big(t[Y] \mid \mathtt{ch}(t,Y) \mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t})\big)\rangle\!\rangle \\ &\quad \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\ I^{(2)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \\ &\quad \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\ I^{(3)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\end{aligned}$$

$$
\begin{aligned}
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
I^{(4)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\\
I^{(5)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\\
I^{(6)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{g_t}.k_t \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\\
I^{(7)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{g_t}.k_t \mid [\![Q]\!]^{\lambda}_{\varepsilon} \mid g_t\big]\\
I^{(8)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[k_t \mid [\![Q]\!]^{\lambda}_{\varepsilon}\big]\\
I^{(9)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[[\![Q]\!]^{\lambda}_{\varepsilon}\big]\\
I^{(10)}_t &= \overline{h_t} \mid p_\rho\big[[\![Q]\!]^{\lambda}_{\varepsilon}\big]
\end{aligned}
$$

2. otherwise, if $n > 0$ then $[\![P]\!]^{\lambda}_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid S$ and $p \in \{1,\dots,n+10\}$, $0 \le j \le n-1$ and

$$
\begin{aligned}
I^{(1)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
I^{(j+2)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid p_{t,\rho}\langle\!\langle (X_1,\dots,X_{n-j}).(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t})\rangle\!\rangle\\
&\mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
I^{(n+2)}_t &= t\Big[[\![P']\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
I^{(n+3)}_t &= t\Big[[\![P']\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\\
I^{(n+4)}_t &= t\Big[[\![P']\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t}\\
&\mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\\
I^{(n+5)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\\
I^{(n+6)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{g_t}.k_t \mid e_t[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\\
I^{(n+7)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[\overline{g_t}.k_t \mid [\![Q]\!]^{\lambda}_{\varepsilon} \mid g_t\big]\\
I^{(n+8)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[k_t \mid [\![Q]\!]^{\lambda}_{\varepsilon}\big]\\
I^{(n+9)}_t &= t\Big[[\![P]\!]^{\lambda}_{t,\rho}\Big] \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^{\lambda}_{\varepsilon}] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{h_t} \mid p_\rho\big[[\![Q]\!]^{\lambda}_{\varepsilon}\big]
\end{aligned}
$$

$$I_t^{(n+10)} = \prod_{k=1}^{n} p_\rho[\llbracket P'_k \rrbracket^\lambda_\varepsilon] \mid \overline{h_t} \mid p_\rho[\llbracket Q \rrbracket^\lambda_\varepsilon].$$

The following lemma formalizes all possible forms for the process $O_u^{(q)}(\llbracket F \rrbracket^\lambda_{\rho''}[h_u.\llbracket P_u \rrbracket^\lambda_{\rho'''}], \llbracket Q'_u \rrbracket^\lambda_\varepsilon)$ for $n \geq 0$ and $q \in \{1, \ldots, n+11\}$. Due to the simplicity of writing for the process $O_u^{(q)}(\llbracket F \rrbracket^\lambda_{\rho''}[h_u.\llbracket P_u \rrbracket^\lambda_{\rho'''}], \llbracket Q'_u \rrbracket^\lambda_\varepsilon)$, we will use the abbreviation $Q_u^{(q)}$ in all places where we do not violate the rationing of the content.

**Definition 5.2.5.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}(\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}], \llbracket Q \rrbracket^\lambda_\varepsilon)$ depending on $n = \mathtt{nl}(p_{u,\rho}, \llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}])$:

1. for $n = 0$ we have $q \in \{1, \ldots, 11\}$, and

$$\begin{aligned}
O_u^{(1)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid u\langle\!\langle (Y).\big(u[Y] \mid \mathsf{ch}(u, Y) \\
&\quad \mid \mathtt{outd}^{\mathsf{s}}(p_{u,\rho'}, p_{\rho'}, \mathtt{nl}(p_{u,\rho'}, Y), \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u})\big)\rangle\!\rangle \\
&\quad \mid m_u.p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(2)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid m_u.p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(3)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(4)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \\
&\quad \mid p_{\rho'}\big[u_u\langle\!\langle (Z).(Z \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_u}.k_u]\big] \\
O_u^{(5)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{f_u}.\overline{g_u}.k_u \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u\big] \\
O_u^{(6)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{g_u}.k_u \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid e_u\langle\!\langle (X).X \rangle\!\rangle.g_u\big] \\
O_u^{(7)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{g_u}.k_u \mid \llbracket Q \rrbracket^\lambda_\varepsilon \mid g_u\big] \\
O_u^{(8)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[k_u \mid \llbracket Q \rrbracket^\lambda_\varepsilon\big] \\
O_u^{(9)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u \mid u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\llbracket Q \rrbracket^\lambda_\varepsilon\big] \\
O_u^{(10)} &= \ h_u \mid \overline{h_u} \mid p_{\rho'}\big[\llbracket Q \rrbracket^\lambda_\varepsilon\big] \\
O_u^{(11)} &= \ p_{\rho'}\big[\llbracket Q \rrbracket^\lambda_\varepsilon\big]
\end{aligned}$$

2. for $n > 0$ and $\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[\llbracket P'_k \rrbracket^\lambda_\varepsilon] \mid S$ we have $q \in \{1, \ldots, n+11\}$ and $0 \leq j \leq n-1$.

$$\begin{aligned}
O_u^{(1)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid u\langle\!\langle (Y).\big(u[Y] \mid \mathsf{ch}(u, Y) \\
&\quad \mid \mathtt{outd}^{\mathsf{s}}(p_{u,\rho'}, p_{\rho'}, \mathtt{nl}(p_{u,\rho'}, Y), \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{h_u})\big)\rangle\!\rangle \\
&\quad \mid m_u.p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\llbracket Q \rrbracket^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(j+2)} &= u\Big[\llbracket F \rrbracket^\lambda_\rho[h_u.\llbracket P \rrbracket^\lambda_{\rho'}]\Big] \mid h_u
\end{aligned}$$

$$
\begin{aligned}
&\mid p_{u,\rho}\langle\!\langle (X_1,\ldots,X_{n-j}).\big(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u}\big)\rangle\!\rangle \\
&\mid m_u.p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(n+2)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \\
&\mid m_u.\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid \overline{m_u}.p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(n+3)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \\
&\mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big] \\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(n+4)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \\
&\mid p_{\rho'}\big[u_u\langle\!\langle (Z).(Z \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_u}.k_u]\big] \\
O_u^{(n+5)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \\
&\mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{f_u}.\overline{g_u}.k_u \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big] \\
O_u^{(n+6)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \\
&\mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{g_u}.k_u \mid e_u[[\![Q]\!]^\lambda_\varepsilon] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big] \\
O_u^{(n+7)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[\overline{g_u}.k_u \mid [\![Q]\!]^\lambda_\varepsilon \mid g_u\big] \\
O_u^{(n+8)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[k_u \mid [\![Q]\!]^\lambda_\varepsilon\big] \\
O_u^{(n+9)} &= u\Big[[\![F]\!]^\lambda_\rho[h_u.[\![P]\!]^\lambda_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid u\langle\!\langle\dagger\rangle\!\rangle.\overline{h_u} \mid p_{\rho'}\big[[\![Q]\!]^\lambda_\varepsilon\big] \\
O_u^{(n+10)} &= \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid h_u \mid \overline{h_u} \mid p_{\rho'}\big[[\![Q]\!]^\lambda_\varepsilon\big] \\
O_u^{(n+11)} &= \prod_{k=1}^{n} p_\rho[[\![P'_k]\!]^\lambda_\varepsilon] \mid p_{\rho'}\big[[\![Q]\!]^\lambda_\varepsilon\big].
\end{aligned}
$$

The following definition formalizes all possible forms for the process $U_s^{(r)}([\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}], [\![R]\!]^\lambda_\varepsilon, [\![Q]\!]^\lambda_\varepsilon)$. Due to the simplicity of writing for the process $U_s^{(r)}([\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}], [\![R]\!]^\lambda_\varepsilon, [\![Q]\!]^\lambda_\varepsilon)$, we will use the abbreviation $U_s^{(r)}$ in all places where we do not violate the rationing of the content.

**Definition 5.2.6.** Let $P, Q, R$ be well-formed compensable processes. Given a name $s$, a path $\rho$, and $r \geq 1$, we define the intermediate processes $U_s^{(r)}$ as in the following:

$$
U_s^{(1)} = s\Big[[\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}]\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle
$$

$$
\begin{aligned}
& \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon] \\
& \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}]) \mid e_s[[\![Q]\!]^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]
\end{aligned}
$$

$$
\begin{aligned}
U_s^{(2)} = {} & s\Big[[\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}]\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
& \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon\{^{[\![Q]\!]^\lambda_\varepsilon}/_Y\}] \\
& \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid \overline{f_s}.e_s[\mathbf{0}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]
\end{aligned}
$$

$$
\begin{aligned}
U_s^{(3)} = {} & s\Big[[\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}]\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
& \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon\{^{[\![Q]\!]^\lambda_\varepsilon}/_Y\}] \\
& \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid e_s[\mathbf{0}] \mid e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]
\end{aligned}
$$

$$
\begin{aligned}
U_s^{(4)} = {} & s\Big[[\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}]\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
& \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon\{^{[\![Q]\!]^\lambda_\varepsilon}/_Y\}] \\
& \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid g_s\big]
\end{aligned}
$$

$$
\begin{aligned}
U_s^{(5)} = {} & s\Big[[\![H]\!]^\lambda_s[[\![P]\!]^\lambda_{s,\rho}]\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
& \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[u\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon\{^{[\![Q]\!]^\lambda_\varepsilon}/_Y\}] \\
& \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big].
\end{aligned}
$$

**Corollary 5.2.6.** Let $P$ be a well-formed compensable process and $\rho$ a path. If $[\![P]\!]^\lambda_\rho \equiv C[P'] \mid D[Q']$ then either:

(i) there are $C_1[\bullet]$, $D_1[\bullet]$, $P_1$, and $Q_1$ such that

- $C[\bullet] = [\![C_1[\bullet]]\!]^\lambda_\rho$
- $D[\bullet] = [\![D_1[\bullet]]\!]^\lambda_\rho$
- $P' = [\![P_1]\!]^\lambda_{\rho'}$ and $Q' = [\![Q_1]\!]^\lambda_{\rho''}$, where $\rho'$ and $\rho''$ are paths to holes in $C[\bullet]$ and $D[\bullet]$, respectively.

(ii) there are $C_1[\bullet]$, $P_1$, $Q, t$ such that

$$
\begin{aligned}
Q' \equiv {} & t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big) \\
& \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^\lambda_\varepsilon] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\
P' \equiv {} & u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[[\![R]\!]^\lambda_\varepsilon] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\![P_1]\!]^\lambda_{t,\rho},
\end{aligned}
$$

and $D[\bullet] = [\bullet]$, and $C[\bullet] = t[C_1[\bullet]]$.

**Lemma 5.2.7.** Suppose $[\![P]\!]^\lambda_\rho \longrightarrow R$. Then one of the following holds for $P$ and $R$:

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $R \equiv [\![E]\!]^\lambda_\rho\Big[[\![C]\!]^\lambda_{\rho_1}[[\![P_1]\!]_{\rho'}] \mid [\![D]\!]_{\rho_1}[[\![P_2]\!]_{\rho''}]\Big]$, or

b) $P \equiv E\big[C[\overline{t}.P_1] \mid D[t[P_2,Q]]\big]$ and $R \equiv [\![E]\!]^\lambda_\rho\Big[[\![C]\!]^\lambda_{\rho_1}[h_t.[\![P_1]\!]^\lambda_{\rho'}] \mid [\![D]\!]^\lambda_{\rho_1}[I_t^{(1)}([\![P_2]\!]^\lambda_{t,\rho''}, [\![Q]\!]^\lambda_\varepsilon)]\Big]$ where $I_t^{(1)}([\![P_2]\!]^\lambda_{t,\rho''}, [\![Q]\!]^\lambda_\varepsilon)$ is given in Definition 5.2.4. or

c) $P \equiv E[u[C[\overline{u}.P_1],Q]]$ and $R \equiv [\![E]\!]^\lambda_\rho\Big[O_u^{(1)}([\![C]\!]^\lambda_{u,\rho_1}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]_\varepsilon)\Big]$ where $O_u^{(1)}([\![C]\!]^\lambda_{u,\rho_1}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]^\lambda_\varepsilon)$ is given in Definition 5.2.5.

d) $P \equiv E[s[C[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]$ and $R \equiv [\![E]\!]^\lambda_\rho\Big[U_s^{(1)}([\![C]\!]^\lambda_{\rho_1}[[\![P_1]\!]^\lambda_{s,\rho'}], [\![R]\!]^\lambda_\varepsilon, [\![Q]\!]^\lambda_\varepsilon)\Big]$ where $U_s^{(1)}([\![C]\!]^\lambda_{\rho_1}[[\![P_1]\!]^\lambda_{s,\rho'}], [\![R]\!]^\lambda_\varepsilon, [\![Q]\!]^\lambda_\varepsilon)$ is given in Definition 5.2.6.

for some contexts $C[\bullet], D[\bullet], E[\bullet]$ and processes $P_1, P_2, Q, R$. Also, paths $\rho_1$ is path to holes in contexts $E[\bullet]$ and $N[\bullet]$, $\rho'$ is path to holes in contexts $C[\bullet]$ and $H[\bullet]$ and $\rho''$ is path for $D[\bullet]$.

*Proof.* The proof is by induction on the reduction $[\![P]\!]_\rho \longrightarrow R$. There are three base cases, which can be obtained by applying Rule (R-In-Out)) with $x = a$ or $x = t$, and this follows the same idea presented in the proof of Lemma 3.2.12. There are one base cases, which can be obtained by Rule (R-Sub-Upd) and the proof is in the following. We consider

$$[\![P]\!]_\rho = E'[C'[P_1'] \mid D'[s.P_2' \mid P_3']] \longrightarrow E'[C'[P_1''] \mid D'[s.P_2' \mid P_3'']] = R$$

and $D'[\bullet] = [\bullet]$ and $C'[\bullet] = s[C''[\bullet]]$

The proof use Lemma 3.2.10, Corollary 5.2.6 and Definition 5.2.2.

$$\begin{aligned}
[\![P]\!]_\rho &= [\![E]\!]^\lambda_\rho[[\![S]\!]^\lambda_{\rho_1}] && (1)\\
&= [\![E]\!]^\lambda_\rho[s[[\![P_1'']\!]^\lambda_{s,\rho_1}] && (2)\\
&\quad \mid s.\big(\texttt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) && (3)\\
&\quad \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[[\![Q]\!]^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big] && (4)\\
&= [\![E]\!]^\lambda_\rho[s[[\![C]\!]_{s,\rho_1}[[\![P_1]\!]^\lambda_{\rho'}]]\\
&\quad \mid s.\big(\texttt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big)\\
&\quad \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[[\![Q]\!]^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\\
&= [\![E[s[C[\texttt{inst}\lfloor \lambda Y.R\rfloor.P'], Q]]]\!]_\rho.
\end{aligned}$$

where

(1) $[\![S]\!]_{\rho_1} = s[C''[P_1']] \mid s.P_2' \mid P_3'$

(2) $[\![P_1'']\!]^\lambda_{s,\rho_1} = C''[P_1']$, and

$P_1' = u_s\Big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}])\Big] \mid [\![P']\!]^\lambda_{s,\rho}$

(3) $P_2' = \texttt{extrd}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]$

(4) $P_3' = v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[[\![Q]\!]^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]$

where $[\![C[\bullet]]\!]^\lambda_{t,\rho_1} = C''[\bullet]$ and $\rho_1$ and $\rho'$ are paths to holes in $E[\bullet]$ and $C[\bullet]$, respectively. ■

In the following, we analyze adaptable processes obtained from the translation of a transactions that contains a failure signal in its body, which can be internal or external. For this we use the Lemma 3.2.17 and Lemma 3.2.16, where we use $[\![\cdot]\!]^\lambda_\rho$ instead $[\![\cdot]\!]_\rho$. In the following lemma we will consider analysis of adaptable processes that can be obtained starting from the translation of a transaction that contains compensation update.

**Lemma 5.2.8.** Let $P_1$ be a well-formed compensable process such that

- $$[\![P_1]\!]^\lambda_\varepsilon \equiv [\![E]\!]^\lambda_\varepsilon\Big[[\![G]\!]^\lambda_\rho\big[[\![N]\!]^\lambda_{\rho'}\big[[\![s[H[\texttt{inst}\lfloor \lambda Y.R_s\rfloor.P_s], Q_s]]\!]^\lambda_{\rho''}\big] \mid M_1\big] \mid M_2 \mid M_3\Big] \mid M_4 \text{ and}$$
- $[\![P_1]\!]^\lambda_\varepsilon \longrightarrow^{n-1} R$,

$$R \equiv [\![E_1]\!]^\lambda_\varepsilon\Big[[\![G_1]\!]^\lambda_\rho\Big[[\![N_1]\!]^\lambda_{\rho'}\Big[I^{(p)}_t([\![P_t']\!]_{t,\rho''}, [\![Q_t']\!]_\varepsilon)\Big] \mid M_1'\Big] \mid M_2' \mid M_3'\Big] \mid M_4',$$

where $I^{(p)}_t([\![P_t]\!]_{t,\rho''}, [\![Q_t]\!]_\varepsilon)$ in $R$ is as in Definition 5.2.4. If $R \longrightarrow R'$ then either

I)

$$R' \equiv [\![E_1]\!]^{\lambda}_{\varepsilon}\Big[[\![G_1]\!]^{\lambda}_{\rho}[[\![N_1]\!]^{\lambda}_{\rho'}[U^{(r+1)}_s([\![H_1]\!]^{\lambda}_{\rho''}[[\![P_s]\!]^{\lambda}_{t,\rho'''}], [\![R'_s]\!]^{\lambda}_{\varepsilon}, [\![Q'_s]\!]^{\lambda}_{\varepsilon})] \mid M'_1] \mid M'_2 \mid M'_3] \mid M'_4,$$

or

II)

$$R' \equiv [\![E_2]\!]^{\lambda}_{\varepsilon}\Big[[\![G_2]\!]^{\lambda}_{\rho}[[\![N_2]\!]^{\lambda}_{\rho'}[U^{(r)}_s([\![H_2]\!]^{\lambda}_{\rho''}[[\![P_s]\!]^{\lambda}_{t,\rho'''}], [\![R''_s]\!]^{\lambda}_{\varepsilon}, [\![Q''_s]\!]^{\lambda}_{\varepsilon})] \mid M''_1] \mid M''_2 \mid M''_3\Big] \mid M''_4.$$

where

- $n > 1$;
- $\rho$ is the path to holes in $[\![E[\bullet]]\!]^{\lambda}_{\varepsilon}$ and $[\![E_k[\bullet]]\!]^{\lambda}_{\varepsilon}$ and $k \in \{1, 2\}$;
- $\rho'$ is the path to holes in $[\![G[\bullet]]\!]^{\lambda}_{\rho}$, and $[\![G_k[\bullet]]\!]^{\lambda}_{\rho}$ and $k \in \{1, 2\}$;
- $\rho''$ is the path to the hole in $[\![N[\bullet]]\!]^{\lambda}_{\rho'}$ and $[\![N_k[\bullet]]\!]^{\lambda}_{\rho'}$ and $k \in \{1, 2\}$;
- $\rho'''$ is the path to hole in $[\![H[\bullet]]\!]^{\lambda}_{\rho'}$ and $[\![H_k[\bullet]]\!]^{\lambda}_{\rho'}$ and $k \in \{1, 2\}$.

*Proof.* The proof follows the same idea that is presented for the proof of Lemma 3.2.16. ■

The following lemma is crucial for the proof of soundness. We use abbreviation as presented in Remark 3.2.18 and additionally we use

$$U^{(r)}_{s_{\phi,k,w}} = U^{(r)}_{s_{\phi,k,w}}([\![H_{\phi,k,w}]\!]^{\lambda}_{\rho''}[[\![P_{s_{\phi,k,w}}]\!]^{\lambda}_{\rho'''}], [\![R'_{s_{\phi,k,w}}]\!]^{\lambda}_{\varepsilon}, [\![Q'_{s_{\phi,k,w}}]\!]^{\lambda}_{\varepsilon})$$

**Lemma 5.2.9.** Let $I^{(p)}_{t_{i,k,w}}$, $O^{(q)}_{u_{c,k,w}}$ and $U^{(r)}_{s_{\phi,k,w}}$ be processes from Definition 5.2.4, Definition 5.2.5 and Definition 5.2.6 respectively. If $[\![P]\!]^{\lambda}_{\varepsilon} \longrightarrow^n R$, with $n \geq 1$, then

1)

$$\begin{aligned} R \equiv \prod_{w=1}^{z}[\![E_w]\!]_{\varepsilon}\Big[\prod_{k=1}^{s_w}[\![G_{k,w}]\!]_{\rho_w}\big[\prod_{i=1}^{l_k}[\![C_{i,k,w}]\!]_{\rho'_{k,w}}[I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k}[\![D_{j,k,w}]\!]_{\rho'_{k,w}}[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]_{\rho''_{k,w}}] \\ \mid \prod_{\phi=1}^{o_k}[\![N_{\phi,k,w}]\!]_{\rho'_{k,w}}[U^{(r)}_{s_{\phi,k,w}}] \mid \prod_{c=1}^{m_k}[\![L_{c,k,w}]\!]_{\rho'_{k,w}}[O^{(q)}_{u_{c,k,w}}]\big]\Big] \end{aligned} \tag{5.14}$$

and $P \longrightarrow^* P'$, where $P'$ is of the following form:

2)

$$\begin{aligned} P' \equiv \prod_{w=1}^{z} E_w\Big[\prod_{k=1}^{s_w} G_{k,w}\big[\prod_{i=1}^{l_k} C_{i,k,w}[t_{i,k,w}[P_{t_{i,k,w}}, Q_{t_{i,k,w}}]] \mid \prod_{j=1}^{r_k} D_{j,k,w}[\overline{t_{j,k,w}}.S_{t_{j,k,w}}] \\ \mid \prod_{\phi=1}^{o_k} N_{\phi,k,w}[s_{\phi,k,w}[H_{\phi,k,w}[\mathtt{inst}\lfloor \lambda Y_{\phi,k,w}.R_{\phi,k,w}\rfloor.P_{\phi,k,w}], Q_{\phi,k,w}]] \\ \mid \prod_{c=1}^{m_k} L_{c,k,w}[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}], Q_{u_{c,k,w}}]]\big]\Big], \end{aligned} \tag{5.15}$$

for some $E_w[\bullet]$, $G_{k,w}[\bullet]$, $C_{i,k,w}[\bullet]$, $D_{j,k,w}[\bullet]$, $N_{\phi,k,w}[\bullet]$, $H_{\phi,k,w}[\bullet]$ and $L_{c,k,w}[\bullet]$ where $w \in \{1, \dots, z\}$, $k \in \{1, \dots, s_w\}$, $i \in \{1, \dots, l_k\}$, $j \in \{1, \dots, r_k\}$, $\phi \in \{1, \dots o_k\}$ and $c \in \{1, \dots, m_k\}$.

*Proof.* The proof proceeds by induction on $n$ and follows the same idea as presented in the proof of Lemma 3.2.19. ∎

**Lemma 5.2.10.** Let processes $I_t^{(p)}(\llbracket P_t' \rrbracket_{t,\rho''}^{\lambda}, \llbracket Q_t' \rrbracket_{\varepsilon}^{\lambda})$, $O_u^{(q)}(\llbracket F \rrbracket_{\rho''}^{\lambda}[h_u.\llbracket P_u \rrbracket_{\rho'''}^{\lambda}], \llbracket Q_u' \rrbracket_{\varepsilon}^{\lambda})$ and $U_s^{(r)}(\llbracket H \rrbracket_{\rho'}^{\lambda}[\llbracket P_s \rrbracket_{s,\rho''}^{\lambda}], \llbracket R_s \rrbracket_{\varepsilon}^{\lambda}, \llbracket Q_s \rrbracket_{\varepsilon}^{\lambda})$ be defined as in Definition 5.2.4, Definition 5.2.5 and Definition 5.2.6, respectively. For any contexts $C[\bullet]$, $D[\bullet]$, $H[\bullet]$, $N[\bullet]$ and $L[\bullet]$ the following holds:

$$C\big[I_t^{(p)}(\llbracket P_t' \rrbracket_{t,\rho''}^{\lambda}, \llbracket Q_t' \rrbracket_{\varepsilon}^{\lambda})\big] \mid D\big[h_t.\llbracket S_t \rrbracket_{\rho}^{\lambda}\big] \longrightarrow^* C\big[\llbracket \mathsf{extr}_{\mathtt{D}}(P_t') \rrbracket_{\rho'}^{\lambda} \mid \llbracket \langle Q_t' \rangle \rrbracket_{\rho'}^{\lambda}\big] \mid D\big[\llbracket S_t \rrbracket_{\rho}^{\lambda}\big], \tag{5.16}$$

$$N\big[U_s^{(r)}(\llbracket H \rrbracket_{\rho'}^{\lambda}[\llbracket P_s \rrbracket_{s,\rho''}^{\lambda}], \llbracket R_s \rrbracket_{\varepsilon}^{\lambda}, \llbracket Q_s \rrbracket_{\varepsilon}^{\lambda})\big] \longrightarrow^* N[\llbracket s[H[P_s], R_s\{^{Q_s}/_{Y_s}\}] \rrbracket_{\rho'}^{\lambda}] \tag{5.17}$$

$$L\big[O_u^{(q)}(\llbracket F \rrbracket_{\rho''}[h_u.\llbracket P_u \rrbracket_{\rho'''}], \llbracket Q_u' \rrbracket_{\varepsilon})\big] \longrightarrow^* L\big[\llbracket \mathsf{extr}_{\mathtt{D}}(F_1[P_u]) \rrbracket_{\rho'} \mid \llbracket \langle Q_u' \rangle \rrbracket_{\rho'}\big] \tag{5.18}$$

*Proof.* The proof proceeds directly by application of the reduction rules from Figure 2.5. ∎

For the proof of operational correspondence we need the following statement:

**Lemma 5.2.11.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\llbracket P \rrbracket_{\rho}^{\lambda} \equiv \llbracket Q \rrbracket_{\rho}^{\lambda}$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ∎

Operational correspondence for the translation of dynamic compensable processes with discarding semantics into adaptable processes with subjective update is given in the following theorem:

**Theorem 5.2.12** (Operational Correspondence for $\llbracket\cdot\rrbracket_{\rho}^{\lambda}$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{D}}^{\lambda}$. We have:

1. If $P \xrightarrow{\tau} P'$ then $\llbracket P \rrbracket_{\varepsilon}^{\lambda} \longrightarrow^k \llbracket P' \rrbracket_{\varepsilon}^{\lambda}$ where either
   a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,
   b) $P \equiv E[C[t[P_1, Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q \rangle] \mid D[P_2]]$ it follows $k = 11 + \mathtt{pb}_{\mathtt{D}}(P_1)$ or
   c) $P \equiv C[u[F[\overline{u}.P_1], Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q \rangle]$ it follows $k = 11 + \mathtt{pb}_{\mathtt{D}}(F[P_1])$,
   d) $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1], Q]]$ and $P' \equiv C[s[H[P_1], R\{^{Q}/_{Y}\}]]$ it follows $k = 5$,

   for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$, processes $P_1, Q, P_2, R$, and name $t, u, s$.
2. If $\llbracket P \rrbracket_{\varepsilon}^{\lambda} \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* \llbracket P' \rrbracket_{\varepsilon}^{\lambda}$.

*Proof.* Case (1) concerns completeness and Case (2) describes soundness. Case (1)(a) concerns usual synchronizations, which are translated by $\llbracket\cdot\rrbracket_{\rho}^{\lambda}$. Cases (1)-(b) and (c) concern synchronizations due to compensation signals. The fault signal can be external or internal to the transaction, which implies that the analysis has two cases. Case (1)-(d) concerns synchronization due to dynamic update of compensation in transaction. In all cases, the number of reduction steps required to mimic the source transition depends on the number of protected blocks of the transaction being canceled. In the following, we consider the proof of completeness and soundness separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases $a$), $b$) and $c$) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 5.2.11, Definition 5.2.2 and Lemma 3.2.9 (Page 47).

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have a derivation that is as *Part* (1) – *Completeness* for discarding semantics and static recovery processes, in case $a$) (cf. page 64, (3.30)). Here we use Definition 5.2.2 instead Definition 3.2.3.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb_D}(P_1)$, and $P' \equiv E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation where we use Definition 5.2.4 for process $I_t^{(p)}([\![P]\!]^\lambda_{t,\rho}, [\![Q]\!]^\lambda_\varepsilon)$, $p \in \{1,\dots,n+10\}$:

$$
\begin{aligned}
[\![P]\!]^\lambda_\varepsilon &\equiv [\![E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]]\!]^\lambda_\varepsilon \\
&= [\![E]\!]^\lambda_\varepsilon\Big[[\![C[t[P_1,Q]]]\!]^\lambda_\rho \mid [\![D[\overline{t}.P_2]]\!]^\lambda_\rho\Big] \\
&= [\![E]\!]^\lambda_\varepsilon\Big[[\![C]\!]^\lambda_\rho[[\![t[P_1,Q]]\!]^\lambda_{\rho'}] \mid [\![D]\!]^\lambda_\rho[[\![\overline{t}.P_2]\!]^\lambda_{\rho''}]\Big] \\
&= [\![E]\!]^\lambda_\varepsilon\Big[[\![C]\!]^\lambda_\rho\big[t\big[[\![P_1]\!]^\lambda_{t,\rho'}\big] \mid t.\big(\mathtt{extrd}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}\rangle\!\rangle \\
&\qquad \mid m_t.p_{\rho'}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big) \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\![Q]\!]^\lambda_\varepsilon] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \mid [\![D]\!]^\lambda_\rho[\overline{t}.h_t.[\![P_2]\!]^\lambda_{\rho''}]\Big] \\
&\longrightarrow [\![E]\!]^\lambda_\varepsilon\Big[[\![C]\!]^\lambda_\rho[I_t^{(1)}([\![P_1]\!]^\lambda_{t,\rho'}, [\![Q]\!]^\lambda_\varepsilon)] \mid [\![D]\!]^\lambda_\rho[h_t.[\![P_2]\!]^\lambda_{\rho''}]\Big] \\
&\longrightarrow^{n+9} [\![E]\!]^\lambda_\varepsilon\Big[[\![C]\!]^\lambda_\rho[I_t^{(n+10)}([\![P_1]\!]^\lambda_{t,\rho'}, [\![Q]\!]^\lambda_\varepsilon)] \mid [\![D]\!]^\lambda_\rho[h_t.[\![P_2]\!]^\lambda_{\rho''}]\Big] \\
&\longrightarrow [\![E]\!]^\lambda_\varepsilon\Big[[\![C]\!]^\lambda_\rho[[\![\mathsf{extr_D}(P_1) \mid \langle Q\rangle]\!]^\lambda_{\rho'}] \mid [\![D]\!]^\lambda_\rho[[\![P_2]\!]^\lambda_{\rho''}]\Big] \\
&= [\![E]\!]^\lambda_\varepsilon\Big[[\![C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle]]\!]^\lambda_\rho \mid [\![D[P_2]]\!]^\lambda_\rho\Big] \\
&= [\![E[C[\mathsf{extr_D}(P_1) \mid \langle Q\rangle] \mid D[P_2]]]\!]^\lambda_\varepsilon \\
&\equiv [\![P']\!]^\lambda_\varepsilon
\end{aligned}
$$

Therefore, $k = 11 + n$.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[u[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb_D}(F[P_1])$, and $P' \equiv C[\mathsf{extr_D}(F[P_1]) \mid \langle Q\rangle]$. Then we have the following derivation:

$$
\begin{aligned}
[\![P]\!]^\lambda_\varepsilon &\equiv [\![C\big[u[D[\overline{u}.P_1],Q]\big]]\!]^\lambda_\varepsilon \\
&= [\![C]\!]^\lambda_\varepsilon\big[[\![u[F[\overline{u}.P_1],Q]]\!]^\lambda_\rho\big] \\
&= [\![C]\!]^\lambda_\varepsilon\big[u\big[[\![F[\overline{u}.P_1]]\!]^\lambda_{u,\rho}\big] \mid u.\big(\mathtt{extrd}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]^\lambda_\varepsilon]\big)]\big] \\
&= [\![C]\!]^\lambda_\varepsilon\big[u\big[[\![F]\!]^\lambda_{u,\rho}[\overline{u}.h_u.[\![P_1]\!]^\lambda_{\rho'}]\big] \mid u.(\mathtt{extrd}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid p_\rho[[\![Q]\!]^\lambda_\varepsilon])\big] \\
&\longrightarrow [\![C]\!]^\lambda_\varepsilon\Big[O_u^{(1)}([\![F]\!]^\lambda_{u,\rho}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]^\lambda_\varepsilon)\Big] \\
&\longrightarrow^{n+9} [\![C]\!]^\lambda_\varepsilon\big[O_u^{(n+10)}([\![F]\!]^\lambda_{u,\rho}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]^\lambda_\varepsilon)\big] \\
&\longrightarrow [\![C]\!]^\lambda_\varepsilon\big[O_u^{(n+11)}([\![F]\!]^\lambda_{u,\rho}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]^\lambda_\varepsilon)\big] \\
&\equiv [\![C]\!]^\lambda_\varepsilon\big[[\![\mathsf{extr_D}(F[P_1])]\!]^\lambda_\rho \mid p_\rho[[\![Q]\!]^\lambda_\varepsilon]\big] \\
&= [\![C\big[\mathsf{extr_D}(F[P_1]) \mid \langle Q\rangle\big]]\!]^\lambda_\varepsilon \\
&\equiv [\![P']\!]^\lambda_\varepsilon
\end{aligned}
$$

Process $O_u^{(q)}([\![F]\!]^\lambda_{u,\rho}[h_u.[\![P_1]\!]^\lambda_{\rho'}], [\![Q]\!]^\lambda_\varepsilon)$, where $q \in \{1,\dots,n+11\}$, is as in Definition 5.2.5. In this case, the role of function $\mathsf{ch}(u,\cdot)$ is central. Therefore, $k = 11 + n$.

d) We have that $P \equiv C[s[H[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^{Q}\!/\!_{Y}\}]]$. We will use Definition 5.2.6 for process $U^{(r)}_s([\![H]\!]^{\lambda}_{s,\rho}[[\![P_1]\!]^{\lambda}_{s,\rho'}], (\!|Q|\!)^{\lambda}_{\varepsilon})$ where $r \in \{1,\dots,5\}$ and have the following:

$$\begin{aligned}
[\![P]\!]^{\lambda}_{\varepsilon} &= [\![C[s[H[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]]\!]^{\lambda}_{\varepsilon}\\
&= [\![C]\!]^{\lambda}_{\varepsilon}\big[s\Big[[\![H]\!]^{\lambda}_{s,\rho}[[\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1]\!]^{\lambda}_{s,\rho'}]\Big]\\
&\quad \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho'}, p_{\rho'}\rangle\!\rangle \mid m_s.p_{\rho'}\,[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle]\big)\big]\\
&\quad \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\\
&= [\![C]\!]^{\lambda}_{\varepsilon}\big[s\Big[[\![H]\!]^{\lambda}_{s,\rho}[u_s\Big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^{\lambda}_{\varepsilon}]\\
&\quad \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}])\Big] \mid [\![P_1]\!]^{\lambda}_{s,\rho}\Big] \mid s.\big(\mathtt{extrd}\langle\!\langle s, p_{s,\rho'}, p_{\rho'}\rangle\!\rangle\\
&\quad \mid m_s.p_{\rho}\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big)\\
&\quad \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[[\![Q]\!]^{\lambda}_{\varepsilon}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\big]\\
&\longrightarrow [\![C]\!]^{\lambda}_{\varepsilon}\big[U^{(1)}_s([\![H]\!]^{\lambda}_{s,\rho}[[\![P_1]\!]^{\lambda}_{s,\rho'}], [\![Q]\!]^{\lambda}_{\varepsilon})\big]\\
&\longrightarrow^4 [\![C]\!]^{\lambda}_{\varepsilon}\big[U^{(5)}_s([\![H]\!]^{\lambda}_{s,\rho}[[\![P_1]\!]^{\lambda}_{s,\rho'}], [\![Q]\!]^{\lambda}_{\varepsilon})\\
&\equiv [\![C[s[H[P_1],R\{^{Q}\!/\!_{Y}\}]]]\!]^{\lambda}_{\varepsilon}
\end{aligned}$$

Therefore, $k = 5$. Let us analyze these reduction steps:

i) In $[\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1]\!]^{\lambda}_{s,\rho}$ we find process $[\![R]\!]^{\lambda}_{\varepsilon}$ on location $u_s$, which is composed in parallel with process $[\![P]\!]^{\lambda}_{s,\rho}$. This location may synchronize with the update prefix on name $u_s$ that is implemented in $[\![s[P,Q]]\!]^{\lambda}_{\rho}$: such a step would move $[\![R]\!]^{\lambda}_{\varepsilon}$ from location $s$ to location $v_s$, leaving $[\![P]\!]^{\lambda}_{s,\rho}$ in $s$.

ii) In the translation of $s[P,Q]$, process $[\![Q]\!]^{\lambda}_{\varepsilon}$ resides in location $e_s$. This location may synchronize with the update prefix implemented in $[\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1]\!]^{\lambda}_{s,\rho}$, which contains $[\![R]\!]^{\lambda}_{\varepsilon}$: such a step allows us to obtain $[\![R]\!]^{\lambda}_{\rho}\{[\![Q]\!]^{\lambda}_{\rho}/Y\}$ (cf. Lemma 5.2.5).

iii) The translations use synchronizations on $f_s$, $e_s$, and $g_s$ to preserve operational correspondence.

(2) **Part (2) – Soundness:** Given $[\![P]\!]^{\lambda}_{\varepsilon} \longrightarrow^n R$, by Lemma 5.2.9, process $R$ has the following form:

$$\begin{aligned}
R \equiv \prod_{w=1}^{z}[\![E_w]\!]^{\lambda}_{\varepsilon}\Big[\prod_{k=1}^{s_w}[\![G_{k,w}]\!]^{\lambda}_{\rho_w}\big[\prod_{i=1}^{l_k}[\![C_{i,k,w}]\!]^{\lambda}_{\rho'_{k,w}}\,[I^{(p)}_{t_{i,k,w}}] \mid \prod_{j=1}^{r_k}[\![D_{j,k,w}]\!]^{\lambda}_{\rho'_{k,w}}\,[h_{t_{j,k,w}}.[\![S_{t_{j,k,w}}]\!]^{\lambda}_{\rho''_{k,w}}]\\
\mid \prod_{\phi=1}^{o_k}[\![N_{\phi,k,w}]\!]^{\lambda}_{\rho'_{k,w}}\,[U^{(r)}_{s_{\phi,k,w}}] \mid \prod_{c=1}^{m_k}[\![L_{c,k,w}]\!]^{\lambda}_{\rho'_{k,w}}\,[O^{(q)}_{u_{c,k,w}}]\big]\Big]
\end{aligned}$$

Also by Lemma 5.2.9, we have $P \longrightarrow^* P''$ where

$$\begin{aligned}
P'' \equiv \prod_{w=1}^{z}E_w\Big[\prod_{k=1}^{s_w}G_{k,w}\big[\prod_{i=1}^{l_k}C_{i,k,w}\big[t_{i,k,w}[P_{t_{i,k,w}},Q_{t_{i,k,w}}]\big] \mid \prod_{j=1}^{r_k}D_{j,k,w}\big[\overline{t_{j,k,w}}.S_{t_{j,k,w}}\big]\\
\mid \prod_{\phi=1}^{o_k}N_{\phi,k,w}\big[s_{\phi,k,w}[H_{\phi,k,w}[\mathtt{inst}\lfloor\lambda Y_{\phi,k,w}.R_{\phi,k,w}\rfloor.P_{\phi,k,w}],Q_{\phi,k,w}]\big]\\
\mid \prod_{c=1}^{m_k}L_{c,k,w}\big[u_{c,k,w}[F_{c,k,w}[\overline{u_{c,k,w}}.P_{u_{c,k,w}}],Q_{u_{c,k,w}}]\big]\Big],
\end{aligned}$$

where by successive application of completeness it follows that $[\![P]\!]^\lambda_\varepsilon \longrightarrow^* [\![P'']\!]^\lambda_\varepsilon$.
By Lemma 5.2.10, i.e., by $l_k$ successive applications of (5.16), $o_k$ successive applications of (5.17) and $m_k$ successive applications of (5.18) on process $R$, it follows that:

$$\begin{aligned}
R \longrightarrow^* & \prod_{w=1}^{z} [\![E_w]\!]^\lambda_\varepsilon \Big[ \prod_{k=1}^{s_w} [\![G_{k,w}]\!]^\lambda_{\rho_w} \big[ \prod_{i=1}^{l_k} [\![C_{i,k,w}]\!]^\lambda_{\rho'_{k,w}} [ [\![\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}})]\!]^\lambda_{\rho''_{k,w}} \mid [\![\langle Q'_{t_{i,k,w}} \rangle]\!]_{\rho''_{k,w}} ] \mid \\
& \prod_{j=1}^{r_k} [\![D_{j,k,w}]\!]^\lambda_{\rho'_{k,w}} [ [\![S_{t_{j,k,w}}]\!]^\lambda_{\rho''_{k,w}} ] \\
& \mid \prod_{\phi=1}^{o_k} [\![N_{\phi,k,w}]\!]^\lambda_{\rho'_{k,w}} [ [\![s_{\phi,k,w}[H_{\phi,k,w}[P_{s_{\phi,k,w}}], R_{s_{\phi,k,w}}\{^{Q'_{s_{\phi,k,w}}}/_{Y}\}]]\!]^\lambda_{\rho''_{k,w}} ] \\
& \mid \prod_{c=1}^{m_k} [\![L_{c,k,w}]\!]^\lambda_{\rho'_{k,w}} \big[ [\![\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}])]\!]^\lambda_{\rho''_{k,w}} \mid [\![\langle Q'_{u_{c,k,w}} \rangle]\!]^\lambda_{\rho''_{k,w}} ]\big] \Big] \\
\equiv & [\![ \prod_{w=1}^{z} E_w \Big[ \prod_{k=1}^{s_w} G_{k,w} \big[ \prod_{i=1}^{l_k} C_{i,k,w} \big[\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}} \rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w} \big[S_{t_{j,k,w}}\big] \\
& \qquad \mid \prod_{\phi=1}^{o_k} N_{\phi,k,w} \big[s_{\phi,k,w}[H_{\phi,k,w}[P_{s_{\phi,k,w}}], R_{s_{\phi,k,w}}\{^{Q'_{s_{\phi,k,w}}}/_{Y}\}]\big] \\
& \qquad \mid \prod_{c=1}^{m_k} L_{c,k,w} \big[\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}} \rangle\big]\big] \Big] ]\!]^\lambda_\varepsilon \\
\equiv & [\![P']\!]^\lambda_\varepsilon
\end{aligned}$$

Therefore, it follows that

$$\begin{aligned}
P' \equiv \prod_{w=1}^{z} E_w \Big[ & \prod_{k=1}^{s_w} G_{k,w} \big[ \prod_{i=1}^{l_k} C_{i,k,w} \big[\mathsf{extr}_{\mathsf{D}}(P'_{t_{i,k,w}}) \mid \langle Q'_{t_{i,k,w}} \rangle\big] \mid \prod_{j=1}^{r_k} D_{j,k,w} \big[S_{t_{j,k,w}}\big] \\
& \mid \prod_{\phi=1}^{o_k} N_{\phi,k,w} \big[s_{\phi,k,w}[H_{\phi,k,w}[P_{s_{\phi,k,w}}], R_{s_{\phi,k,w}}\{^{Q'_{s_{\phi,k,w}}}/_{Y}\}]\big] \\
& \mid \prod_{c=1}^{m_k} L_{c,k,w} \big[\mathsf{extr}_{\mathsf{D}}(F_{c,k,w}[P_{u_{c,k,w}}]) \mid \langle Q'_{u_{c,k,w}} \rangle\big]\big] \Big].
\end{aligned}$$

Also, by Proposition 2.2.3 (i.e., by $l_k$ successive applications of case b) and and $m_k$ successive applications of case c)) and by Proposition 5.1.2 (i.e., by $o_k$ applications of it) on process $P''$, it follows that: $P'' \longrightarrow^* P'$.
By successive application of **(1)** – **Completeness** on the derivation $P'' \longrightarrow^* P'$ it follows that $[\![P'']\!]^\lambda_\varepsilon \longrightarrow^* [\![P']\!]^\lambda_\varepsilon$.

■

## 5.3 Translating $\mathcal{C}^\lambda_{\mathsf{P}}$ into $\mathcal{S}$

The translation $\mathcal{C}^\lambda_{\mathsf{P}}$ into $\mathcal{S}$, denoted $(\!|\cdot|\!)^\lambda_\rho$. This translation relies on the main idea and principles of encoding $\mathcal{C}_{\mathsf{P}}$ into $\mathcal{S}$ (cf. Section 3.3).

**Remark 5.3.1** (Reserved names)**.** We require sets of *reserved names* and need to revised Definition 3.1.2 as in the following:

(i) the set of *reserved location names* $\mathcal{N}_l^r$ is unchanged and,

(ii) the set of *reserved synchronization names* is extended such that

$$\mathcal{N}_s^r = \{h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, j_x \mid x \in \mathcal{N}_t\}.$$

Accordingly, the function that counts the number of protected blocks is as Figure 3.4, while the function that counts the number of transactions represents the extension of the function in Figure 3.13, as in the following:

**Definition 5.3.1** (Number of transactions). Let $P = \mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1$ be a well-formed compensable process. The number of transactions which occur in $P$, denoted $\mathtt{ts_P}(P)$, is equal to $\mathtt{ts_P}(P_1)$.

Below we give a formal definition of the translation $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{S}$. We instruct the reader that this translation relies directly on the ideas that are presented in Section 3.3.1 and Section 3.3.2.

### 5.3.1 Translation Correctness

We need auxiliary processes $\mathtt{extr_P}$, that represent appropriate extensions of processes (3.37). For $\mathtt{extr_P}$ we will use process $\mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', n, m)$ which is defined similarly as $(\!|\cdot|\!)_\rho$ (cf. (3.36)). Therefore, the auxiliary process $\mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', n, m)$: (i) moves $n$ processes from location $l_1$ to location $l_1'$; (ii) moves $m$ processes from location $l_2$ to location $l_2'$.

For the definition of $\mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', n, m)$ we introduce the following auxiliary processes for $n, m > 0$:

$$\begin{aligned}
&\mathtt{outp}_1^{\mathtt{s}}(t, l_1, l_1', n) = l_1\langle\!\langle (X_1, \ldots, X_n).\left(\prod_{i=1}^{n} l_1'[X_i] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t\right)\rangle\!\rangle;\\
&\mathtt{outp}_2^{\mathtt{s}}(t, t_1, \ldots, t_m, l_2, l_2', m) = l_2\langle\!\langle (Y_1, \ldots, Y_m).\\
&\qquad \left(r_t.\left(\prod_{k=1}^{m} \left(l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\right)\right) \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}\right)\rangle\!\rangle;\\
&\mathtt{outp}_3^{\mathtt{s}}(t, t_1, \ldots, t_m, l_1, l_1', l_2, l_2', n, m) = l_1\langle\!\langle (X_1, \ldots, X_n).l_2\langle\!\langle (Y_1, \ldots, Y_m).\\
&\qquad \left(\prod_{i=1}^{n} l_1'[X_i] \mid r_t.\left(\prod_{k=1}^{m} \left(l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\right)\right) \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}\right)\rangle\!\rangle\rangle\!\rangle.
\end{aligned}$$

The auxiliary process $\mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', n, m)$, where $\mathtt{top}(l_2, P) = \{t_1, \ldots, t_m\}$ (cf. Definition 3.3.3) for $m > 0$, is now defined as follows:

$$\mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', n, m) = \begin{cases} \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t}.r_t & \text{if } n, m = 0 \\ \mathtt{outp}_1^{\mathtt{s}}(t, l_1, l_1', n) & \text{if } n > 0, m = 0 \\ \mathtt{outp}_2^{\mathtt{s}}(t, t_1, \ldots, t_m, l_2, l_2', m) & \text{if } n = 0, m > 0 \\ \mathtt{outp}_3^{\mathtt{s}}(t, t_1, \ldots, t_m, l_1, l_1', l_2, l_2', n, m) & \text{if } n, m > 0 \end{cases} \tag{5.19}$$

**Definition 5.3.2** (Update Prefix for Extraction). Let $t$, $l_1, l_1', l_2$, and $l_2'$ be names and $P$ is an adaptable process. We write $\mathtt{extrp}\langle\!\langle t, P, l_1, l_1', l_2, l_2' \rangle\!\rangle$ to stand for the following (subjective) update prefix:

$$\begin{aligned} \mathtt{extrp}\langle\!\langle t, P, l_1, l_1', l_2, l_2' \rangle\!\rangle = \; & t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t, Y) \\ & \mid \mathtt{outp^s}(t, P, l_1, l_1', l_2, l_2', \mathtt{nl}(l_1, Y), \mathtt{nl}(l_2, Y))\rangle\!\rangle \end{aligned} \tag{5.20}$$

$$
\begin{aligned}
(\!|\langle P\rangle|\!)^{\lambda}_{\rho} &= p_{\rho}\big[(\!|P|\!)^{\lambda}_{\epsilon}\big]\\
(\!|t[P,Q]|\!)^{\lambda}_{\rho} &= \beta_{\rho}\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid t.\big(\texttt{extrp}\langle\!\langle t, (\!|P|\!)^{\lambda}_{t,\rho}, p_{t,\rho}, p_{\rho}, \beta_{t,\rho}, \beta_{\rho}\rangle\!\rangle\\
&\qquad \mid m_t.p_{\rho}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big)\\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big]\\
&\qquad \mid j_t.\beta_{\rho}\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
(\!|\texttt{inst}\lfloor\lambda Y.R\rfloor.P|\!)^{\lambda}_{t,\rho} &= u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[(\!|R|\!)^{\lambda}_{\epsilon}]\\
&\qquad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid (\!|P|\!)^{\lambda}_{t,\rho}\\
(\!|\overline{t}.P|\!)^{\lambda}_{\rho} &= \overline{t}.h_t.(\!|P|\!)^{\lambda}_{\rho}
\end{aligned}
$$

Figure 5.4: Translating $\mathcal{C}^{\lambda}_{\mathrm{P}}$ into $\mathcal{S}$.

The intuition for the process $\texttt{extrp}\langle\!\langle t, P, l_1, l_1', l_2, l_2'\rangle\!\rangle$ in preserving semantics with dynamic recovery is the same as in static recovery (cf. Definition 3.3.5). The only difference comparing with static compensation is the third parameter. This parameter enables us to have a controlled execution of adaptable processes due to achieve the operational correspondence. The explanation of names $m_t$ and $k_t$ is the same as presented in the encoding discarding semantics with dynamic update.

Based on the above modifications, the encoding of processes with dynamic compensations is given with the following definition:

**Definition 5.3.3** (Translating $\mathcal{C}^{\lambda}_{\mathrm{P}}$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with preserving semantics into (subjective) adaptable processes as a tuple $((\!|\cdot|\!)^{\lambda}_{\rho}, \varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}})$ where:

(a) The renaming policy $\varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$
\varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, j_x, r_x\} \cup \{p_{\rho}, \beta_{\rho} : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{5.21}
$$

(b) The translation $(\!|\cdot|\!)^{\lambda}_{\rho} : \mathcal{C}_{\mathrm{P}} \to \mathcal{S}$ is as in Figure 5.4 and as a homomorphism for other operators.

#### 5.3.1.1 Structural Criteria

In the following, we prove that translation $\mathcal{C}^{\lambda}_{\mathrm{P}}$ into $\mathcal{S}$ satisfies name invariance (cf. Definition 2.3.5). Analysis of compositionality is left for future research work.

##### 5.3.1.1.1 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 5.3.3 (a).

**Theorem 5.3.2** (Name invariance for $(\!|\cdot|\!)^{\lambda}_{\rho}$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{(\!|\cdot|\!)^{\lambda}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}}(x)\}$, and

(ii) $(\!|\sigma(P)|\!)^{\lambda}_{\sigma(\rho)} = \sigma'((\!|P|\!)^{\lambda}_{\rho})$.

*Proof.* The proof follows the idea presented in the proof of Theorem 5.2.4. ∎

#### 5.3.1.2 Semantic Criteria

In this subsection we prove that translation $\mathcal{C}_{\mathrm{P}}^{\lambda}$ into $\mathcal{S}$ satisfies operational correspondence. Analysis of divergence reflection and success sensitiveness are left for future research work.

##### 5.3.1.2.1 Operational Correspondence

We use Lemma 5.2.5 for the proof of operational correspondence. Also, as in previously presented encodings, we are interested in giving a precise account of the number of computation steps for achieving operational correspondence. We claim again that subjective updates are more efficient than objective updates. We will use Definition 3.2.1 (1) and Remark 5.2.2 (eq. (5.6)), since we need function $\mathtt{nl}(l, P)$ to give us the number of locations $l$ in process $P$.
The following definition formalizes all possible forms for the process $I_t^{(p)}((\!|P|\!)_{t,\rho}^{\lambda}, (\!|Q|\!)_{\varepsilon}^{\lambda})$. Due to the simplicity of writing for the process $I_t^{(p)}((\!|P|\!)_{t,\rho}^{\lambda}, (\!|Q|\!)_{\varepsilon}^{\lambda})$, we will use the abbreviation $I_t^{(p)}$ in all places where we do not violate the rationing of the content.

**Definition 5.3.4.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}((\!|P|\!)_{t,\rho}^{\lambda}, (\!|Q|\!)_{\varepsilon}^{\lambda})$ depending on $n = \mathtt{nl}(p_{t,\rho}, (\!|P|\!)_{t,\rho}^{\lambda})$ and $m = \mathtt{nl}(\beta_{t,\rho}, (\!|P|\!)_{t,\rho}^{\lambda})$:

1. if $n = 0$ and $m = 0$ then $p \in \{1, \ldots, 13\}$;

$$
\begin{aligned}
I_t^{(1)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \mathtt{extrp}\langle\!\langle t, (\!|P|\!)_{t,\rho}^{\lambda}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho\rangle\!\rangle \\
&\qquad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
&\equiv \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid t\langle\!\langle (Y).t[Y] \mid \mathsf{ch}(t, Y) \\
&\qquad \mid \mathtt{outp}^{\mathtt{s}}(t, (\!|P|\!)_{t,\rho}^{\lambda}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho, \mathtt{nl}(p_{t,\rho}, Y), \mathtt{nl}(\beta_{t,\rho}, Y))\rangle\!\rangle \\
&\qquad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(2)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(3)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\qquad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(4)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle \\
&\qquad \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(5)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \\
&\qquad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(6)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{g_t}.k_t \mid e_t[(\!|Q|\!)_{\epsilon}^{\lambda}] \mid e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\Big] \\
&\qquad \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(7)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{g_t}.k_t \mid (\!|Q|\!)_{\epsilon}^{\lambda} \mid g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(8)} &= \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^{\lambda}\big] \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[k_t \mid (\!|Q|\!)_{\epsilon}^{\lambda}\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}
$$

$$I_t^{(9)} = \beta_\rho\Big[t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$
$$I_t^{(10)} = \beta_\rho\Big[\overline{j_t}.r_t \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$
$$I_t^{(11)} = \beta_\rho\Big[r_t \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid \beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$
$$I_t^{(12)} = r_t \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big] \mid \overline{r_t}.\overline{h_t}$$
$$I_t^{(13)} = p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big] \mid \overline{h_t}$$

2. if $n > 0$ and $m = 0$ then then $(\!|P|\!)_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ and $p \in \{1,\ldots,n+13\}$, $0 \le j \le n-1$ and:

$$\begin{aligned}
I_t^{(1)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \texttt{extrp}\langle\!\langle t, (\!|P|\!)_{t,\rho}^\lambda, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho\rangle\!\rangle \\
&\mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(2+j)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid p_{t,\rho}\langle\!\langle(X_1,\ldots,X_{n-j}).\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)_\varepsilon] \\
&\mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t\big)\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(2+n)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \\
&\mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(3+n)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(n+4)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \\
&\mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(n+5)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid \\
&e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(n+6)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{g_t}.k_t \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \\
&\mid e_t\langle\!\langle(X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(n+7)} = \beta_\rho\Big[&t\big[(\!|P|\!)_{t,\rho}^\lambda\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[\overline{g_t}.k_t \mid (\!|Q|\!)_\epsilon^\lambda \mid g_t\big]\Big] \\
&\mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$I_t^{(n+8)} = \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[k_t \mid (\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$

$$I_t^{(n+9)} = \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{m} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}.r_t \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$

$$I_t^{(n+10)} = \beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{j_t}.r_t \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$

$$I_t^{(n+11)} = \beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid \beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}$$

$$I_t^{(n+12)} = \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{r_t}.\overline{h_t}$$

$$I_t^{(n+13)} = \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{h_t}$$

3. if $n = 0$ and $m > 0$ then $(\!|P|\!)_{t,\rho} = \prod_{k=1}^{m} \beta_{t,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ and, $p \in \{1, \ldots, m+13\}$, $0 \le j \le m-1$ and:

$$\begin{aligned}
I_t^{(1)} = \beta_\rho\Big[& t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \mathtt{extrp}\langle\!\langle t, (\!|P|\!)^{\lambda}_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho\rangle\!\rangle \\
&\mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(2+j)} = \beta_\rho\Big[& t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \beta_{t,\rho}\langle\!\langle(Y_1, \ldots, Y_{m-s}).\Big(r_t.\big(\prod_{k=1}^{m-s}\big(\beta_\rho[\mathcal{E}(Y_k, t)] \\
&\mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid \prod_{k=1}^{s}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \\
&\mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}\Big)\rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(2+m)} = \beta_\rho\Big[& t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \\
&\mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(3+m)} = \beta_\rho\Big[& t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \\
&\mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big] \\
&\mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$\begin{aligned}
I_t^{(4+m)} = \beta_\rho\Big[& t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \\
&\mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \\
&\mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}
\end{aligned}$$

$$
\begin{aligned}
I_t^{(5+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid e_t[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(6+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[\overline{g_t}.k_t \mid e_t[(\!|Q|\!)^\lambda_\epsilon] \mid e_t\langle\!\langle(X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(7+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[\overline{g_t}.k_t \mid (\!|Q|\!)^\lambda_\epsilon \mid g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(8+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid \overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[k_t \mid (\!|Q|\!)^\lambda_\epsilon \mid \big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(9+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(10+m)} &= \beta_\rho\Big[r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\quad \mid \overline{j_t} \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(11+m)} &= \beta_\rho\Big[r_t.\big(\prod_{k=1}^{m}\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid \beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(12+m)} &= r_t.\big(\prod_{k=1}^{m}\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{r_t}.\overline{h_t}\\
I_t^{(13+m)} &= \prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{h_t};
\end{aligned}
$$

4. otherwise, if $n > 0$ and $m > 0$ then $(\!|P|\!)_{t,\rho} = \prod_{i=1}^{n} p_{t,\rho}[(\!|P'_i|\!)_\varepsilon] \mid \prod_{k=1}^{m} \beta_{t,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ and $p \in \{1,\ldots,n+m+13\}$, $0 \le j \le n-1$, $0 \le s \le m-1$ and:

$$
\begin{aligned}
I_t^{(1)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid \texttt{extrp}\langle\!\langle t, (\!|P|\!)^\lambda_{t,\rho}, p_{t,\rho}, p_\rho, \beta_{t,\rho}, \beta_\rho\rangle\!\rangle\\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\quad \mid v_t\big[u_t\langle\!\langle(Z).(Z \mid e_t[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_t\langle\!\langle(X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(2+j+s)} &= \beta_\rho\Big[t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid p_{t,\rho}\langle\!\langle(X_1,\ldots,X_{n-j}).\beta_{t,\rho}\langle\!\langle(Y_1,\ldots,Y_{m-s}).\\
&\quad \Big(\prod_{i=1}^{m-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{n-s}\big(\beta_\rho[\mathcal{E}(Y_k,t)] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\\
&\quad \mid \prod_{k=1}^{s}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \mid \overline{m_t}.\overline{k_t}.t\langle\!\langle\dagger\rangle\!\rangle.\overline{j_t}\Big)\rangle\!\rangle\rangle\!\rangle\\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(2+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\mid \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t} \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(3+m+n)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{m} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_t.\big(\prod_{k=1}^{n}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\\
&\mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(4+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid\\
&f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(5+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}\\
&\mid p_\rho\big[\overline{f_t}.\overline{g_t}.k_t \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(6+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{m} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_t.\big(\prod_{k=1}^{n}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t}\\
&\mid p_\rho\big[\overline{g_t}.k_t \mid e_t[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(7+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big]\\
&\mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)\\
&\mid \overline{k_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{j_t} \mid p_\rho\big[\overline{g_t}.k_t \mid (\!|Q|\!)^{\lambda}_{\epsilon} \mid g_t\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\\
I_t^{(8+n+m)} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda}_{t,\rho}\big] \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big)
\end{aligned}
$$

$$
\begin{aligned}
& \mid \overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t} \mid p_\rho\big[k_t \mid (\!|Q|\!)^\lambda_\epsilon \mid \big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(9+n+m)} &= \beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid t\big[(\!|P|\!)^\lambda_{t,\rho}\big] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \\
& \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{j_t} \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(10+n+m)} &= \beta_\rho\Big[\prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \\
& \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big)\big) \mid \overline{j_t} \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_t.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(11+n+m)} &= \beta_\rho\Big[\prod_{i=1}^{m} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{n}\beta_\rho[(\!|P'_k|\!)_\varepsilon] \\
& \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid \beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_t}.\overline{h_t} \\
I_t^{(12+n+m)} &= \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_t.\big(\prod_{k=1}^{m}\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{r_t}.\overline{h_t} \\
I_t^{(13+n+m)} &= \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{t_k}.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_{t_k}}.\overline{h_{t_k}}\big) \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{h_t}
\end{aligned}
$$

The following lemma formalizes all possible forms for the process $O_u^{(q)}((\!|F|\!)^\lambda_{\rho''}[h_u.(\!|P_u|\!)^\lambda_{\rho'''}], (\!|Q'_u|\!)^\lambda_\varepsilon)$ for $n, m \geq 0$ and $q \in \{1, \dots, n+m+14\}$. Due to the simplicity of writing for the process $O_u^{(q)}((\!|F|\!)^\lambda_{\rho''}[h_u.(\!|P_u|\!)^\lambda_{\rho'''}], (\!|Q'_u|\!)^\lambda_\varepsilon)$, we will use the abbreviation $O_u^{(q)}$ in all places where we do not violate the rationing of the content.

**Definition 5.3.5.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}((\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}], (\!|Q|\!)_\varepsilon)$ depending on $n = \mathtt{nl}(p_{u,\rho}, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}])$ and $m = \mathtt{nl}(\beta_{u,\rho}, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}])$:

1. for $n = 0$ and $m = 0$ we have $q \in \{1, \dots, 14\}$ and

$$
\begin{aligned}
O_u^{(1)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid \mathtt{extrp}\langle\!\langle u, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}], p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho \rangle\!\rangle \\
& \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u]) \rangle\!\rangle\big] \\
& \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u) \rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
&\equiv \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid u\langle\!\langle (Y).u[Y] \mid \mathtt{ch}(u, Y) \\
& \mid \mathtt{outp}^{\mathtt{s}}(u, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}], p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho, \mathtt{nl}(p_{u,\rho}, Y), \mathtt{nl}(\beta_{u,\rho}, Y)) \rangle\!\rangle \\
& \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u]) \rangle\!\rangle\big] \\
& \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u) \rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(2)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{j_u}.r_u \\
& \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u]) \rangle\!\rangle\big] \\
& \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u) \rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(3)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger \rangle\!\rangle.\overline{j_u}.r_u \\
& \mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u]) \rangle\!\rangle\big] \\
& \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X \rangle\!\rangle.g_u) \rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X \rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$\begin{aligned}
O_u^{(4)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \\
&\quad \mid p_\rho\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_t}.k_u]\big]\Big] \\
&\quad \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(5)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \\
&\quad \mid p_\rho\big[\overline{f_u}.\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(6)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \\
&\quad \mid p_\rho\big[\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(7)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \\
&\quad \mid p_\rho\big[\overline{g_t}.k_u \mid (\!|Q|\!)^\lambda_\epsilon \mid g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(8)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[k_u \mid (\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(9)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(10)} &= \beta_\rho\Big[h_u \mid \overline{j_u}.r_u \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(11)} &= \beta_\rho\Big[h_u \mid r_u \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(12)} &= h_u \mid r_u \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{r_u}.\overline{h_u} \\
O_u^{(13)} &= h_u \mid p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big] \mid \overline{h_u} \\
O_u^{(14)} &= p_\rho\big[(\!|Q|\!)^\lambda_\epsilon\big].
\end{aligned}$$

2. for $n > 0, m = 0$ and $(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}] = \prod_{k=1}^{n} p_{u,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ we have $q \in \{1, \ldots, 14+n\}$ and $0 \leq j \leq n-1$ and

$$\begin{aligned}
O_u^{(1)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid \mathtt{extrp}\langle\!\langle u, p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho\rangle\!\rangle \\
&\quad \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(2+j)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \\
&\quad \mid p_{u,\rho}\langle\!\langle (X_1, \ldots, X_{n-j}).\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u\Big)\rangle\!\rangle \\
&\quad \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(2+n)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \\
&\quad \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big] \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(3+n)} &= \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \\
&\quad \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]
\end{aligned}$$

$$\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle]\big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(4+n)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_t}.k_u]\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(5+n)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[\overline{f_u}.\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(6+m)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{m} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(7+n)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[\overline{g_t}.k_u \mid (\!|Q|\!)^{\lambda}_{\epsilon} \mid g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(8+n)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[k_u \mid (\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(9+n)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(10+n)} = \beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid \overline{j_u}.r_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(11+n)} = \beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(12+n)} = \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid h_u \mid r_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{r_u}.\overline{h_u}$$

$$O_u^{(13+n)} = \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid h_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{h_u}$$

$$O_u^{(14+n)} = \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big].$$

3. for $n = 0, m > 0$ and $(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}] = \prod_{k=1}^{m} \beta_{u,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ we have $q \in \{1, \ldots, 14+m\}$ and $0 \leq j \leq m-1$ and:

$$O_u^{(1)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid \mathtt{extrp}\langle\!\langle u, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}], p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho\rangle\!\rangle$$

$$
\begin{aligned}
&\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(2+j)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u\\
&\mid \beta_{u,\rho}\langle\!\langle (Y_1,\ldots,Y_{m-s}).\Big(r_u.\big(\prod_{k=1}^{m-s}\big(\beta_\rho[\mathcal{E}(Y_k,u)] \mid j_{t_u}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{t_u}}.\overline{h_{t_u}}\big)\\
&\mid \prod_{k=1}^{s}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\Big)\rangle\!\rangle\\
&\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(2+m)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\\
&\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(3+m)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(4+m)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid\\
& r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_t}.k_u]\big]\Big]\\
&\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(5+m)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[\overline{f_u}.\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(6+m)} = {}& \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid\\
& r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}.r_u\\
&\mid p_\rho\big[\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
O_u^{(7+m)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid
$$

$$\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u} \mid p_\rho\big[\overline{g_t}.k_u \mid (\!|Q|\!)^{\lambda}_{\epsilon} \mid g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(8+m)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big)$$
$$\mid \overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u} \mid p_\rho\big[k_u \mid (\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(9+m)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid$$
$$u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u} \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(10+m)} = \beta_\rho\Big[h_u \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big)$$
$$\mid \overline{j_u} \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(11+m)} = \beta_\rho\Big[h_u \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big]\Big]$$
$$\mid \beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(12+m)} = r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid h_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{r_u}.\overline{h_u}$$

$$O_u^{(13+m)} = \big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid h_u \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big] \mid \overline{h_u}$$

$$O_u^{(14+m)} = \prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big) \mid p_\rho\big[(\!|Q|\!)^{\lambda}_{\epsilon}\big].$$

4. otherwise, for $n, m > 0$ and $(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}] = \prod_{i=1}^{n} p_{u,\rho}[(\!|P'_k|\!)_\varepsilon] \mid \prod_{k=1}^{m} \beta_{u,\rho}[(\!|P'_k|\!)_\varepsilon] \mid S$ we have $q \in \{1, \ldots, 14+n+m\}$ and $0 \le j \le n-1$ and $0 \le k \le m-1$ and

$$O_u^{(1)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid \texttt{extrp}\langle\!\langle u, (\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}], p_{u,\rho}, p_\rho, \beta_{u,\rho}, \beta_\rho\rangle\!\rangle$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle(X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]$$
$$\mid v_u\big[u_u\langle\!\langle(Z).(Z \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_u\langle\!\langle(X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(2+j)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u$$
$$\mid p_{u,\rho}\langle\!\langle(X_1, \ldots, X_{n-j}).\beta_{u,\rho}\langle\!\langle(Y_1, \ldots, Y_{m-s}).$$
$$\Big(\prod_{i=1}^{n-j} p_\rho[X_i] \mid \prod_{i=1}^{j} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m-s}\big(\beta_\rho[\mathcal{E}(Y_k, u)] \mid j_{u_u}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_u}}.\overline{h_{u_u}}\big)$$
$$\mid \prod_{k=1}^{s}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_u}.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle\dagger\rangle\!\rangle.\overline{j_u}\Big)\rangle\!\rangle\rangle\!\rangle$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle(X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]$$
$$\mid v_u\big[u_u\langle\!\langle(Z).(Z \mid e_u[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_t.e_u\langle\!\langle(X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle(X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}$$

$$O_u^{(2+n+m)} = \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]$$

$$
\begin{aligned}
&\mid r_u.\big(\prod_{k=1}^{n}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(3+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_t}.k_u])\rangle\!\rangle\big]\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(4+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid\\
&r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \mid u_u[\overline{f_u}.\overline{g_t}.k_u]\big]\Big]\\
&\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(5+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[\overline{f_u}.\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(6+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid\\
&r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}.r_u\\
&\mid p_\rho\big[\overline{g_t}.k_u \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(7+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon] \mid\\
&r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[\overline{g_t}.k_u \mid (\!|Q|\!)^\lambda_\epsilon \mid g_u\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(8+n+m)} = \beta_\rho\Big[&u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P'_i|\!)_\varepsilon]\\
&\mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P'_k|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u}\\
&\mid p_\rho\big[k_u \mid (\!|Q|\!)^\lambda_\epsilon\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}
\end{aligned}
$$

$$
\begin{aligned}
O_u^{(9+n+m)} = &\ \beta_\rho\Big[u\big[(\!|F|\!)_\rho[h_u.(\!|P|\!)_{\rho'}]\big] \mid h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid \\
&\ r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{j_u} \\
&\ \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(10+n+m)} = &\ \beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \mid \\
&\ \overline{j_u} \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(11+n+m)} = &\ \beta_\rho\Big[h_u \mid \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \\
&\ \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]\Big] \mid \beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u} \\
O_u^{(12+n+m)} = &\ \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid r_u.\big(\prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big)\big) \\
&\ \mid h_u \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big] \mid \overline{r_u}.\overline{h_u} \\
O_u^{(13+n+m)} = &\ \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid \prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big) \mid h_u \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big] \mid \overline{h_u} \\
O_u^{(14+n+m)} = &\ \prod_{i=1}^{n} p_\rho[(\!|P_i'|\!)_\varepsilon] \mid \prod_{k=1}^{m}\big(\beta_\rho[(\!|P_k'|\!)_\varepsilon] \mid j_{u_k}.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_{u_k}}.\overline{h_{u_k}}\big) \mid p_\rho\big[(\!|Q|\!)_\epsilon^\lambda\big]
\end{aligned}
$$

The following lemma formalizes all possible forms for the process $U_s^{(r)}((\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda], (\!|R|\!)_\varepsilon^\lambda, (\!|Q|\!)_\varepsilon^\lambda,)$ for $r \in \{1,\ldots,5\}$. Due to the simplicity of writing for the process $U_s^{(r)}((\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda], (\!|R|\!)_\varepsilon^\lambda, (\!|Q|\!)_\varepsilon^\lambda,)$, we will use the abbreviation $U_s^{(r)}$ in all places where we do not violate the rationing of the content.

**Definition 5.3.6.** Let $P, Q, R$ be well-formed compensable processes. Given a name $s$, a path $\rho$, and $r \geq 1$, we define the intermediate processes $U_s^{(r)}((\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda], (\!|R|\!)_\varepsilon^\lambda, (\!|Q|\!)_\varepsilon^\lambda,)$

$$
\begin{aligned}
U_s^{(1)} = &\beta_\rho\Big[s\Big[(\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda]\Big] \mid s.\big(\texttt{extrp}\langle\!\langle s, p_{s,\rho}, p_\rho, \beta_{s,\rho}, \beta_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[(\!|R|\!)_\varepsilon^\lambda] \\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}]) \mid e_s[(\!|Q|\!)_\varepsilon^\lambda] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]\Big] \\
&\mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \\
U_s^{(2)} = &\ \beta_\rho\Big[s\Big[(\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda]\Big] \mid s.\big(\texttt{extrp}\langle\!\langle s, p_{s,\rho}, p_\rho, \beta_{s,\rho}, \beta_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[(\!|R|\!)_\varepsilon^\lambda\{^{(\!|Q|\!)_\varepsilon^\lambda}/_Y\}] \\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid \overline{f_s}.e_s[\mathbf{0}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]\Big] \\
&\mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \\
U_s^{(3)} = &\ \beta_\rho\Big[s\Big[(\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda]\Big] \mid t.\big(\texttt{extrp}\langle\!\langle s, p_{s,\rho}, p_\rho, \beta_{s,\rho}, \beta_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[(\!|R|\!)_\varepsilon^\lambda\{^{(\!|Q|\!)_\varepsilon^\lambda}/_Y\}] \\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid e_s[\mathbf{0}] \mid e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big]\Big] \\
&\mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s}
\end{aligned}
$$

$$
\begin{aligned}
U_s^{(4)} = \beta_\rho\Big[t\Big[(\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda]\Big] \mid s.\big(\mathtt{extrp}\langle\!\langle s, p_{s,\rho}, p_\rho, \beta_{s,\rho}, \beta_\rho\rangle\!\rangle \\
\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[(\!|R|\!)_\varepsilon^\lambda\{^{(\!|Q|\!)_\varepsilon^\lambda}/_Y\}] \\
\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid g_s\big]\Big] \mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s} \\
U_s^{(5)} = \beta_\rho\Big[s\Big[(\!|H|\!)_s^\lambda[(\!|P|\!)_{s,\rho}^\lambda]\Big] \mid s.\big(\mathtt{extrp}\langle\!\langle s, p_{s,\rho}, p_\rho, \beta_{s,\rho}, \beta_\rho\rangle\!\rangle \\
\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid v_s\big[u\langle\!\langle (Z).(Z \mid e_s[(\!|R|\!)_\varepsilon^\lambda\{^{(\!|Q|\!)_\varepsilon^\lambda}/_Y\}] \\
\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\Big] \mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s}.
\end{aligned}
$$

Operational correspondence for the translation of dynamic compensable processes into adaptable processes with subjective update, is given in the following theorem:

**Theorem 5.3.3** (Operational Correspondence for $(\!|\cdot|\!)_\rho^\lambda$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathrm{P}}^{\lambda}$.

(1) If $P \longrightarrow P'$ then $(\!|P|\!)_\varepsilon^\lambda \longrightarrow^k (\!|P'|\!)_\varepsilon^\lambda$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and and $P' \equiv E[C[\mathsf{extr}_{\mathrm{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 14 + \mathtt{pb}_{\mathrm{P}}(P_1) + \mathtt{ts}_{\mathrm{P}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathrm{P}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 14 + \mathtt{pb}_{\mathrm{P}}(F[P_1]) + \mathtt{ts}_{\mathrm{P}}(F[P_1])$,

d) $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$ it follows $k = 5$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$, processes $P_1, Q, P_2, R$, and name $t, u, s$.

(2) If $(\!|P|\!)_\varepsilon^\lambda \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* (\!|P'|\!)_\varepsilon^\lambda$.

*Proof.* Case (1) concerns completeness and Case (2) describes soundness. Therefore, in the following we consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases $a$), $b$) and $c$) of Proposition 2.2.3 (Page 18). In all cases, we use Definition 5.2.2 and Lemma 3.2.9 (Page 47).

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have a derivation that is as *Part* (1) – *Completeness* for preserving semantics and static recovery processes, in case $a$). Here we use Definition 5.3.3 instead Definition 3.3.6.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb}_{\mathrm{P}}(P_1)$ and $m = \mathtt{ts}_{\mathrm{P}}(P_1)$, and $P' \equiv E[C[\mathsf{extr}_{\mathrm{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation where we use Definition 5.3.4 for process $I_t^{(p)}([\![P]\!]_{t,\rho}^\lambda, [\![Q]\!]_\varepsilon^\lambda)$, $p \in \{1, \ldots, 14 + n + m\}$:

$$
\begin{aligned}
(\!|P|\!)_\varepsilon^\lambda \equiv & (\!|E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]|\!)_\varepsilon^\lambda \\
= & (\!|E|\!)_\varepsilon^\lambda\Big[(\!|C[t[P_1,Q]]|\!)_\rho^\lambda \mid (\!|D[\overline{t}.P_2]|\!)_\rho^\lambda\Big] \\
= & (\!|E|\!)_\varepsilon^\lambda\Big[(\!|C|\!)_\rho^\lambda[(\!|t[P_1,Q]|\!)_{\rho'}^\lambda] \mid (\!|D|\!)_\rho^\lambda[(\!|\overline{t}.P_2|\!)_{\rho''}^\lambda]\Big] \\
= & (\!|E|\!)_\varepsilon^\lambda\Big[(\!|C|\!)_\rho^\lambda\Big[\beta_{\rho'}\Big[t\big[(\!|P_1|\!)_{t,\rho'}^\lambda\big] \mid t.\big(\mathtt{extrp}\langle\!\langle t, p_{t,\rho'}, p_{\rho'}, \beta_{t,\rho'}, \beta_{\rho'}\rangle\!\rangle \\
& \mid m_t.p_{\rho'}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\rangle\!\rangle\big]\big) \\
& \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)_\epsilon^\lambda] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big]
\end{aligned}
$$

$$
\begin{aligned}
&\mid j_t.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_t}.\overline{h_t}\Big] \mid (\!|D|\!)^\lambda_\rho[\overline{t}.h_t.(\!|P_2|\!)^\lambda_{\rho''}]\big]\\
\longrightarrow\ & (\!|E|\!)^\lambda_\varepsilon\Big[(\!|C|\!)^\lambda_\rho\big[I^{(1)}_t((\!|P_1|\!)^\lambda_{t,\rho'},(\!|Q|\!)^\lambda_\varepsilon)\big] \mid (\!|D|\!)^\lambda_\rho[h_t.(\!|P_2|\!)^\lambda_{\rho''}]\Big]\\
\longrightarrow^{m+n+12}\ & (\!|E|\!)^\lambda_\varepsilon\Big[(\!|C|\!)^\lambda_\rho\big[\big[I^{(m+n+13)}_t((\!|P_1|\!)^\lambda_{t,\rho'},(\!|Q|\!)^\lambda_\varepsilon)\big] \mid (\!|D|\!)^\lambda_\rho\big[h_t.(\!|P_2|\!)^\lambda_{\rho''}\big]\Big]\\
\longrightarrow\ & (\!|E|\!)^\lambda_\varepsilon\Big[(\!|C|\!)^\lambda_\rho\big[(\!|\mathsf{extr_P}(P_1) \mid \langle Q\rangle|\!)^\lambda_{\rho'}\big] \mid (\!|D|\!)^\lambda_\rho\big[(\!|P_2|\!)^\lambda_{\rho''}\big]\Big]\\
=\ & (\!|E|\!)^\lambda_\varepsilon\Big[(\!|C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle]|\!)^\lambda_\rho \mid (\!|D[P_2]|\!)^\lambda_\rho\Big]\\
=\ & (\!|E[C[\mathsf{extr_P}(P_1) \mid \langle Q\rangle] \mid D[P_2]]|\!)^\lambda_\varepsilon\\
\equiv\ & (\!|P'|\!)^\lambda_\varepsilon
\end{aligned}
$$

Therefore, $k = 14 + n + m$.

c) This case concerns a synchronization due to an internal error notification. Here we have $P \equiv C[u[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb_P}(F[P_1])$, $m = \mathtt{ts_P}(F[P_1])$ and and $P' \equiv C[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle]$. We have the following derivation where we use Definition 5.3.5 for process $O^{(q)}_u((\!|F|\!)^\lambda_{\rho''}[h_u.(\!|P_u|\!)^\lambda_{\rho'''}],(\!|Q'_u|\!)^\lambda_\varepsilon)$, and $q \in \{1,\ldots,m+n+14\}$

$$
\begin{aligned}
(\!|P|\!)^\lambda_\varepsilon \equiv\ & (\!|C\big[u[F[\overline{u}.P_1],Q]\big]|\!)^\lambda_\varepsilon\\
=\ & (\!|C|\!)^\lambda_\varepsilon\big[(\!|u[F[\overline{u}.P_1],Q]|\!)^\lambda_\rho\big]\\
=\ & (\!|C|\!)^\lambda_\varepsilon\Big[\beta_\rho\Big[u\big[(\!|F[\overline{u}.P_1]|\!)^\lambda_{u,\rho}\big] \mid u.\big(\mathtt{extrp}\langle\!\langle u,p_{u,\rho},p_\rho,\beta_{u,\rho},\beta_\rho\rangle\!\rangle\\
&\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.k_u])\rangle\!\rangle\big]\big)\\
&\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[(\!|Q|\!)^\lambda_\epsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\Big]\Big]\\
&\mid j_u.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_u}.\overline{h_u}\\
\longrightarrow\ & (\!|C|\!)^\lambda_\varepsilon\Big[O^{(1)}_u((\!|F|\!)^\lambda_{u,\rho}[h_u.(\!|P_1|\!)^\lambda_{\rho'}],(\!|Q|\!)^\lambda_\varepsilon)\Big]\\
\longrightarrow^{13+n+m}\ & (\!|C|\!)_\varepsilon\big[O^{(14+n+m)}_u((\!|F|\!)^\lambda_{u,\rho}[h_u.(\!|P_1|\!)^\lambda_{\rho'}],(\!|Q|\!)^\lambda_\varepsilon)\big]\\
\equiv\ & (\!|C|\!)_\varepsilon\big[(\!|\mathsf{extr_P}(F[P_1])|\!)_\rho \mid p_\rho[(\!|Q|\!)_\varepsilon]\big]\\
=\ & (\!|C\big[\mathsf{extr_P}(F[P_1]) \mid \langle Q\rangle\big]|\!)_\varepsilon\\
\equiv\ & (\!|P'|\!)_\varepsilon
\end{aligned}
$$

We can then conclude that $(\!|P|\!)^\lambda_\varepsilon \longrightarrow^k (\!|P'|\!)^\lambda_\varepsilon$ where $k = 14 + n + m$.

d) We have that $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv s[C[P_1],R\{^Q/_Y\}],$. We will use Definition 5.3.6 for process $U^{(r)}_s((\!|H|\!)^\lambda_{s,\rho}[[\![P_1]\!]^\lambda_{s,\rho'}],(\!|Q|\!)^\lambda_\varepsilon)$ where $r \in \{1,\ldots,5\}$ and have the following:

$$
\begin{aligned}
(\!|P|\!)^\lambda_\varepsilon =\ & (\!|C[s[H[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]|\!)^\lambda_\varepsilon\\
=\ & (\!|C|\!)^\lambda_\varepsilon\Big[\beta_\rho\Big[s\big[(\!|H|\!)^\lambda_{s,\rho}[(\!|\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1|\!)^\lambda_{s,\rho'}]\big]\\
&\mid s.\big(\mathtt{extrp}\langle\!\langle s,p_{s,\rho'},p_{\rho'},\beta_{s,\rho'},\beta_{\rho'}\rangle\!\rangle\\
&\mid m_s.p_{\rho'}\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big)\\
&\mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[(\!|Q|\!)^\lambda_\epsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\Big]\\
&\mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s}\big]\\
=\ & (\!|C|\!)^\lambda_\varepsilon\Big[\beta_\rho\Big[s\big[(\!|H|\!)^\lambda_{s,\rho}[u_s\Big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[[\![R]\!]^\lambda_\varepsilon]\\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}])\Big] \mid [\![P]\!]^\lambda_{s,\rho'}]\big]
\end{aligned}
$$

$$
\begin{aligned}
&\mid s.\big(\mathtt{extrp}\langle\!\langle s, p_{s,\rho'}, p_{\rho'}, \beta_{s,\rho'}, \beta_{\rho'}\rangle\!\rangle \\
&\mid m_s.p_{\rho'}\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \\
&\mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[(\!|Q|\!)^{\lambda}_{\epsilon}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\big] \\
&\mid j_s.\beta_\rho\langle\!\langle (X).X\rangle\!\rangle.\overline{r_s}.\overline{h_s}] \\
\longrightarrow\ & (\!|C|\!)^{\lambda}_{\varepsilon}[U_s^{(1)}((\!|H|\!)^{\lambda}_{s,\rho}[(\!|P_1|\!)^{\lambda}_{s,\rho'}], (\!|Q|\!)^{\lambda}_{\varepsilon})] \\
\longrightarrow^4\ & (\!|C|\!)^{\lambda}_{\varepsilon}[U_s^{(5)}((\!|H|\!)^{\lambda}_{s,\rho}[(\!|P_1|\!)^{\lambda}_{s,\rho'}], (\!|Q|\!)^{\lambda}_{\varepsilon})] \\
\equiv\ & (\!|C[s[H[P_1],R\{^{Q}/_{Y}\}]]|\!)^{\lambda}_{\varepsilon}
\end{aligned}
$$

Therefore, $k = 5$. The explanation for the reduction steps is as in the proof of Theorem 5.2.12 (Part (1) – Completeness 1–(d)).

(2) **Part (2) – Soundness:** The proof of soundness follows the explanation presented in the proof of *Part (2) – Soundness* for Theorem 5.2.12 (cf. 2).

■

## 5.4 Translating $\mathcal{C}_{\mathtt{A}}^{\lambda}$ into $\mathcal{S}$

The translation $\mathcal{C}_{\mathtt{A}}^{\lambda}$ into $\mathcal{S}$, denoted $\langle\!|\cdot|\!\rangle^{\lambda}_{\rho}$, relies on the idea and principles of encoding $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ (cf. Section 3.4). We also require sets of *reserved names* as in Remark 5.2.1 and we use function for determining the number of locations as in Remark 5.2.2. We will use process $\mathtt{outd}^{\mathtt{s}}$ as defined for $[\![\cdot]\!]_\rho$ (cf. (3.2) and Example 3.2.1). We need some additional auxiliary processes.

**Definition 5.4.1** (Update Prefix for Extraction). Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extra}\langle\!\langle t, l_1, l_2\rangle\!\rangle$ to stand for the following (subjective) update prefix:

$$
\mathtt{extra}\langle\!\langle t, l_1, l_2\rangle\!\rangle = t\langle\!\langle (Y).\,\big(t[Y] \mid \mathtt{ch}(t, Y) \mid \mathtt{outd}^{\mathtt{s}}(l_1, l_2, \mathtt{nl}(l, Y), \overline{m_t}.\overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t})\big)\rangle\!\rangle \tag{5.22}
$$

The intuition for the process $\mathtt{extra}\langle\!\langle t, l_1, l_2\rangle\!\rangle$ is the same as in the translation of $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ with static recovery (cf. Section 3.4.0.1 and Section 3.4.1). The difference is in the third parameter for process $\mathtt{outd}^{\mathtt{s}}$, which enables us to have a controlled execution of adaptable processes, which is important to establish operational correspondence. The prefix $t\langle\!\langle \dagger\rangle\!\rangle$ and name $h_t$ have the same roles as in $\langle\!|\cdot|\!\rangle_\rho$. The differences concern names $m_t$ and $j_t$: name $m_t$ ensures that every translation of compensation $Q$ is updated if the translation of compensation update exists, name $j_t$ controls the erasing of location on name $t$ with its contents. Using well-formed compesable processes, the translation of $\mathcal{C}_{\mathtt{A}}^{\lambda}$ into $\mathcal{S}$ is as follows:

**Definition 5.4.2** (Translating $\mathcal{C}_{\mathtt{A}}^{\lambda}$ into $\mathcal{S}$). Let $\rho$ be a path. We define the translation of compensable processes with dynamic recovery into (subjective) adaptable processes as a tuple $(\langle\!|\cdot|\!\rangle^{\lambda}_{\rho}, \varphi_{\langle\!|\cdot|\!\rangle^{\lambda}_{\rho}})$ where:

(a) The renaming policy

$$
\varphi_{\langle\!|\cdot|\!\rangle^{\lambda}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, r_x, u_x, v_x, e_x, g_x, f_x, j_x, n_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t. \end{cases}
$$

(b) The translation $\langle\!|\cdot|\!\rangle^{\lambda}_{\rho} : \mathcal{C}_{\mathtt{A}}^{\lambda} \to \mathcal{S}$ is as in Figure 5.5 and as a homomorphism for other operators.

$$
\begin{aligned}
\langle\!\langle P \rangle\!\rangle^{\lambda}_{\rho} &= p_\rho[\llparenthesis P \rrparenthesis^{\lambda}_{\varepsilon}] \\
\llparenthesis t[P,Q] \rrparenthesis^{\lambda}_{\rho} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid p_\rho[\llparenthesis Q \rrparenthesis_{\varepsilon}] \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big]\big) \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\
\llparenthesis \mathtt{inst}\lfloor \lambda Y.R \rfloor.P \rrparenthesis^{\lambda}_{t,\rho} &= u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis R \rrparenthesis^{\lambda}_{\varepsilon}] \\
&\quad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid \llparenthesis P \rrparenthesis^{\lambda}_{t,\rho} \\
\llparenthesis \overline{t}.P \rrparenthesis^{\lambda}_{\rho} &= \overline{t}.h_t.\llparenthesis P \rrparenthesis^{\lambda}_{\rho}
\end{aligned}
$$

Figure 5.5: Translating $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{S}$.

As in previous introduced encodings, key elements in Figure 5.5 are the translations of $t[P,Q]$ and $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1$. Indeed, these translations share location names $u_t$, $v_t$, and $e_t$ (as well as names $f_t$ and $g_t$) in order to account for the possible replacement of $Q$ in $t[P,Q]$ with $R$ in $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P_1$, using updates. As stated earlier, $\mathtt{inst}\lfloor \lambda Y.R \rfloor.P$ produces a new compensation behavior $R\{^{Q}/_{Y}\}$ after an internal transition.

### 5.4.1 Translation Correctness

We now prove that $\llparenthesis \cdot \rrparenthesis^{\lambda}_{\rho}$ satisfies the three criteria in Definition 2.3.5: compositionality, name invariance, and operational correspondence. The other criteria are left as a research topic for future work.

#### 5.4.1.1 Structural Criteria

In the following, we prove the two criteria, compositionality and name invariance which are introduced in Definition 2.3.5.

##### 5.4.1.1.1 Compositionality

As we described in the previous encodings, to mediate between translations of subterms, we define a *context* for each process operator, which again depends on free names of the subterms:

**Definition 5.4.3** (Compositional context for $\mathcal{C}^{\lambda}_{\mathtt{A}}$)**.** For all process operator from $\mathcal{C}^{\lambda}_{\mathtt{D}}$, instead transaction we define a compositional context in $\mathcal{S}$ as in Definition 3.2.4. For transaction and compensation update compositional contexts are as follows:

$$
\begin{aligned}
C_{t[,],\rho}[\bullet_1, \bullet_2] &= t\Big[[\bullet_1]\Big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big]\big) \\
&\quad \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[[\bullet_2]] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\
C_{\mathtt{inst},\rho}[\bullet_1, \bullet_2] &= u_t\Big[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[[\bullet_1]] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\bullet_2] \\
C_Y[\bullet_1] &= [\bullet_1]
\end{aligned}
$$

Using this definition, we may now state the following result:

**Theorem 5.4.1** (Compositionality for $(\!|\cdot|\!)^{\lambda}_{\rho}$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}^{\lambda}_{\mathtt{A}}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$
\begin{array}{lll}
(\!|\langle P\rangle|\!)^{\lambda}_{\rho} = C_{\langle\rangle,\rho}\left[(\!|P|\!)^{\lambda}_{\varepsilon}\right] & (\!|t[P,Q]|\!)^{\lambda}_{\rho} = C_{t[,],\rho}\left[(\!|P|\!)^{\lambda}_{t,\rho},(\!|Q|\!)^{\lambda}_{\varepsilon}\right] & (\!|P \mid Q|\!)^{\lambda}_{\rho} = C_{\mid}\left[(\!|P|\!)^{\lambda}_{\rho},(\!|Q|\!)^{\lambda}_{\rho}\right] \\
(\!|a.P|\!)^{\lambda}_{\rho} = C_{a.}\left[(\!|P|\!)^{\lambda}_{\rho}\right] & (\!|\bar{t}.P|\!)^{\lambda}_{\rho} = C_{\bar{t}.}\left[(\!|P|\!)^{\lambda}_{\rho}\right] & (\!|(\nu x)P|\!)^{\lambda}_{\rho} = C_{(\nu x)}\left[(\!|P|\!)^{\lambda}_{\rho}\right] \\
(\!|\overline{a}.P|\!)^{\lambda}_{\rho} = C_{\overline{a}.}\left[(\!|P|\!)^{\lambda}_{\rho}\right] & (\!|!\pi.P|\!)^{\lambda}_{\rho} = C_{!\pi.}\left[(\!|P|\!)^{\lambda}_{\rho}\right] & \\
(\!|Y|\!)^{\lambda}_{\rho} = C_{Y}\left[(\!|Y|\!)^{\lambda}_{\rho}\right] & (\!|\mathtt{inst}\lfloor\lambda Y.R\rfloor.P|\!)^{\lambda}_{t,\rho} = C_{\mathtt{inst},\rho}[(\!|R|\!)^{\lambda}_{\varepsilon},(\!|P|\!)^{\lambda}_{t,\rho}] &
\end{array}
$$

*Proof.* Follows directly from the definition of contexts (cf. Definition 5.4.3) and from the definition of $(\!|\cdot|\!)^{\lambda}_{\rho} : \mathcal{C}^{\lambda}_{\mathtt{A}} \to \mathcal{S}$ (cf. Figure 5.5). Therefore, considering Definition 5.4.3 and Figure 5.5 we present derivation for: transaction, compensation update, process variable and all well-formed compensable processes $P, Q$ and $R$. The other operators have the same derivation as in the proof of Theorem 3.2.2. Hence, the following holds:

$$
\begin{aligned}
(\!|t[P,Q]|\!)^{\lambda}_{\rho} &= C_{t[,],\rho}\left[(\!|P|\!)^{\lambda}_{t,\rho},(\!|Q|\!)^{\lambda}_{\varepsilon}\right] \\
&= t\left[(\!|P|\!)^{\lambda}_{t,\rho}\right] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_{\rho}\rangle\!\rangle \mid m_t.p_{\rho}\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big]\big) \\
&\quad \mid t.t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[(\!|Q|\!)^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big] \\
(\!|\mathtt{inst}\lfloor\lambda Y.R\rfloor.P|\!)^{\lambda}_{t,\rho} &= C_{\mathtt{inst},\rho}[(\!|R|\!)^{\lambda}_{\varepsilon},(\!|P|\!)^{\lambda}_{t,\rho}] \\
&= u_t\left[e_t\langle\!\langle (Y).(\overline{g_t}.u_t\langle\!\langle (Z).(Z \mid e_t[(\!|R|\!)^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_t}.e_t[\mathbf{0}])\right] \mid (\!|P|\!)^{\lambda}_{t,\rho} \\
(\!|Y|\!)^{\lambda}_{\rho} &= C_{Y}\left[(\!|Y|\!)^{\lambda}_{\rho}\right] = (\!|Y|\!)^{\lambda}_{\rho}
\end{aligned}
$$

■

#### 5.4.1.1.2 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 5.4.2 (a).

**Theorem 5.4.2** (Name invariance for $(\!|\cdot|\!)^{\lambda}_{\rho}$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{(\!|\cdot|\!)^{\lambda}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}}(x)\}$, and

(ii) $(\!|\sigma(P)|\!)^{\lambda}_{\sigma(\rho)} = \sigma'((\!|P|\!)^{\lambda}_{\rho})$.

*Proof.* The proof proceeds in the same direction as the proof of Theorem 5.2.4. ■

### 5.4.1.2 Semantic Criteria - Operational Correspondence

The analysis of operational correspondence follows the same ideas as in the translations $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ (cf. Section 3.4). Therefore, we use Definition 3.4.5, Remark 3.4.4 and Definition 3.4.6. Below we state the premise (theorems and lemmas) that are necessary for the proof of operation correspondence.

The following definition formalizes the intermediate processes that appear during derivation, denoted with $I^{(p)}_t((\!|P|\!)^{\lambda}_{t,\rho},(\!|Q|\!)^{\lambda}_{\varepsilon})$. Also, due to the simplicity of writing for the process $I^{(p)}_t((\!|P|\!)^{\lambda}_{t,\rho},(\!|Q|\!)^{\lambda}_{\varepsilon})$, we will use the abbreviation $I^{(p)}_t$ in all places where we do not violate the rationing of the content. As in previously presented encodings, it plays a significant role in proving completeness and soundness.

**Definition 5.4.4.** Let $P, Q$ be well-formed compensable processes. Given a name $t$, a path $\rho$, and $p \geq 1$, we define the intermediate processes $I_t^{(p)}(\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}, \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon})$ depending on $n = \mathtt{nl}(p_{t,\rho}, \llparenthesis P \rrparenthesis_{t,\rho})$, $m = \mathtt{ts_A}(P)$ and $s = \mathtt{S}(P)$:

1. if $n = 0$ then $p \in \{1, \ldots, 14\}$;

$$
\begin{aligned}
I_t^{(1)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.n_t.k_t]) \rangle\!\rangle\big]\big) \\
&\quad \mid t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t} \rangle\!\rangle \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
&\equiv t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid r_t.\big(t\langle\!\langle (Y).\big(t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), \overline{m_t}.\overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t})\big) \rangle\!\rangle \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.n_t.k_t]) \rangle\!\rangle\big]\big) \mid t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t} \rangle\!\rangle \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
I_t^{(2)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid r_t.\big(t\langle\!\langle (Y).\big(t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), \overline{m_t}.\overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t})\big) \rangle\!\rangle \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]) \rangle\!\rangle\big]\big) \mid \overline{r_t}.k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
I_t^{(3)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid t\langle\!\langle (Y).\big(t[Y] \mid \mathsf{ch}(t,Y) \mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho}, Y), \overline{m_t}.\overline{k_t}.t\langle\!\langle \dagger \rangle\!\rangle)\big) \rangle\!\rangle \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]) \rangle\!\rangle\big] \mid k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
I_t^{(4)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid \overline{m_t}.\overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]) \rangle\!\rangle\big] \mid k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
I_t^{(5)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]) \rangle\!\rangle\big] \mid k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big] \\
I_t^{(6)} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle \\
&\quad \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]\big] \mid k_t.\overline{h_t} \\
I_t^{(7)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t} \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t\big] \mid k_t.\overline{h_t} \\
I_t^{(8)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{g_t}.j_t.n_t.\overline{k_t} \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid e_t\langle\!\langle (X).X \rangle\!\rangle.g_t\big] \mid k_t.\overline{h_t} \\
I_t^{(9)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{g_t}.j_t.n_t.\overline{k_t} \mid \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon} \mid g_t\big] \mid k_t.\overline{h_t} \\
I_t^{(10)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid \overline{j_t}.t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[j_t.n_t.\overline{k_t} \mid \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t} \\
I_t^{(11)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid t\langle\!\langle \dagger \rangle\!\rangle.\overline{n_t} \mid p_\rho\big[n_t.\overline{k_t} \mid \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t} \\
I_t^{(12)} &= \overline{n_t} \mid p_\rho\big[n_t.\overline{k_t} \mid \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t} \\
I_t^{(13)} &= p_\rho\big[\overline{k_t} \mid \llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t} \\
I_t^{(14)} &= p_\rho\big[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}\big] \mid \overline{h_t}
\end{aligned}
$$

2. otherwise, if $n > 0$ then $\llparenthesis P \rrparenthesis_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\llparenthesis P'_k \rrparenthesis_{\varepsilon}] \mid S$ and $p \in \{1, \ldots, 14 + n + 4m\}$.

$$
\begin{aligned}
I_t^{(1)} &= t\Big[\llbracket P \rrbracket^{\lambda}_{t,\rho}\Big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho \rangle\!\rangle \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]) \rangle\!\rangle\big]\big) \\
&\quad \mid t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t} \rangle\!\rangle \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X \rangle\!\rangle.g_t) \rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(2+4m+s-n)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_{t,\rho}[\llparenthesis P'_i\rrparenthesis_{\varepsilon}]\Big] \mid r_t.\big(\mathtt{extra}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big]\big) \mid \overline{r_t}.k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(3+4m+s-n)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_{t,\rho}[\llparenthesis P'_i\rrparenthesis_{\varepsilon}]\Big] \mid \mathtt{extrd}\langle\!\langle t, p_{t,\rho}, p_\rho\rangle\!\rangle \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big] \mid k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(4+4m+s-n+j)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho} \mid \prod_{i=1}^{s-n} p_{t,\rho}[\llparenthesis P'_i\rrparenthesis_{\varepsilon}]\Big] \mid p_{t,\rho}\langle\!\langle (X_1, \ldots, X_{n-j}). \\
&\quad \big(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid m_t.\overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle\rangle\!\rangle.\overline{n_t}\big) \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big] \mid k_t.\overline{h_t} \\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(4+4m+s)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid m_t.\overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big] \\
&\quad \mid k_t.\overline{h_t} \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(5+4m+s)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\rangle\!\rangle\big] \\
&\quad \mid k_t.\overline{h_t} \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(6+4m+s)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[u_t\langle\!\langle (Z).(Z \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \\
&\quad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]\big] \mid k_t.\overline{h_t}
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(7+4m+s)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t} \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \\
&\quad \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big] \mid k_t.\overline{h_t}
\end{aligned}
$$

$$
\begin{aligned}
I_t^{(8+4m+s)} &= t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{g_t}.j_t.n_t.\overline{k_t} \mid e_t[\llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}] \\
&\quad \mid e_t\langle\!\langle (X).X\rangle\!\rangle.g_t\big] \mid k_t.\overline{h_t}
\end{aligned}
$$

$$
I_t^{(9+4m+s)} = t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[\overline{g_t}.j_t.n_t.\overline{k_t} \mid \llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon} \mid g_t\big] \mid k_t.\overline{h_t}
$$

$$
I_t^{(10+4m+s)} = t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid \overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[j_t.n_t.\overline{k_t} \mid \llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t}
$$

$$
I_t^{(11+4m+s)} = t\Big[\llparenthesis P'\rrparenthesis_{t,\rho}\Big] \mid \prod_{k=1}^{j} p_\rho[\llparenthesis P'_k\rrparenthesis_{\varepsilon}] \mid t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t} \mid p_\rho\big[n_t.\overline{k_t} \mid \llparenthesis Q\rrparenthesis^{\lambda}_{\varepsilon}\big] \mid k_t.\overline{h_t}
$$

$$I_t^{(12+4m+s)} = \prod_{k=1}^{j} p_\rho[\langle\!| P'_k |\!\rangle_\varepsilon] \mid \overline{n_t} \mid p_\rho\big[n_t.\overline{k_t} \mid \langle\!| Q |\!\rangle_\varepsilon^\lambda\big] \mid k_t.\overline{h_t}$$

$$I_t^{(13+4m+s)} = \prod_{k=1}^{j} p_\rho[\langle\!| P'_k |\!\rangle_\varepsilon] \mid p_\rho\big[\overline{k_t} \mid \langle\!| Q |\!\rangle_\varepsilon^\lambda\big] \mid k_t.\overline{h_t}$$

$$I_t^{(14+4m+s)} = \prod_{k=1}^{j} p_\rho[\langle\!| P'_k |\!\rangle_\varepsilon] \mid p_\rho\big[\langle\!| Q |\!\rangle_\varepsilon^\lambda\big] \mid \overline{h_t}$$

The following definition formalizes all possible forms for the process $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$. Also, due to the simplicity of writing for the process $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$, we will use the abbreviation $O_u^{(q)}$ in all places where we do not violate the rationing of the content.

**Definition 5.4.5.** Let $P, Q$ be well-formed compensable processes. Given a name $u$, paths $\rho, \rho'$, and $q \geq 1$, we define the intermediate processes $O_u^{(q)}(\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}], \langle\!| Q |\!\rangle_\varepsilon)$ depending on $n = \mathtt{nl}(p_{u,\rho}, \langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}])$, $m = \mathtt{ts_A}(F[P])$ and $s = \mathtt{S}(F[P])$:

1. if $n = 0$ then $p \in \{1, \dots, 15\}$;

$$\begin{aligned}
O_u^{(1)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.n_u.k_u])\rangle\!\rangle\big]\big) \\
&\quad \mid u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
&\equiv u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid r_u.\big(u\langle\!\langle (Y).\big(u[Y] \mid \mathsf{ch}(u, Y) \\
&\quad \mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), \overline{m_u}.\overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u})\big)\rangle\!\rangle \\
&\quad \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.n_u.k_u])\rangle\!\rangle\big]\big) \mid u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(2)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid r_u.\big(u\langle\!\langle (Y).\big(u[Y] \mid \mathsf{ch}(u, Y) \\
&\quad \mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), \overline{m_u}.\overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u})\big)\rangle\!\rangle \\
&\quad \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big]\big) \mid \overline{r_u}.k_u.\overline{h_u} \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(3)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid u\langle\!\langle (Y).\big(u[Y] \mid \mathsf{ch}(u, Y) \\
&\quad \mid \mathtt{outd^s}(p_{u,\rho}, p_\rho, \mathtt{nl}(p_{u,\rho}, Y), \overline{m_u}.\overline{k_u}.u\langle\!\langle \dagger\rangle\!\rangle)\big)\rangle\!\rangle \\
&\quad \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big] \mid k_u.\overline{h_u} \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(4)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid h_u \mid \overline{m_u}.\overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big] \\
&\quad \mid k_u.\overline{h_u} \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(5)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big] \mid k_u.\overline{h_u} \\
&\quad \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big] \\
O_u^{(6)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle \\
&\quad \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}]\big] \mid k_u.\overline{h_u} \\
O_u^{(7)} &= u\Big[\langle\!| F |\!\rangle_\rho[h_u.\langle\!| P |\!\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u} \mid e_u[\langle\!| Q |\!\rangle_\varepsilon^\lambda] \\
&\quad \mid f_t.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big] \mid k_u.\overline{h_u}
\end{aligned}$$

$$O_u^{(8)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[\overline{g_u}.j_u.n_u.\overline{k_u}$$
$$\mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon] \mid e_u\langle\!\langle (X).X\rangle\!\rangle.g_u\big] \mid k_u.\overline{h_u}$$
$$O_u^{(9)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[\overline{g_u}.j_u.n_u.\overline{k_u} \mid \langle\!| Q|\!\rangle^\lambda_\varepsilon \mid g_u\big] \mid k_u.\overline{h_u}$$
$$O_u^{(10)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[j_u.n_u.\overline{k_u} \mid \langle\!| Q|\!\rangle^\lambda_\varepsilon\big] \mid k_u.\overline{h_u}$$
$$O_u^{(11)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u} \mid p_\rho\big[n_u.\overline{k_u} \mid \langle\!| Q|\!\rangle^\lambda_\varepsilon\big] \mid k_u.\overline{h_u}$$
$$O_u^{(12)} = h_u \mid \overline{n_u} \mid p_\rho\big[n_u.\overline{k_u} \mid \langle\!| Q|\!\rangle^\lambda_\varepsilon\big] \mid k_u.\overline{h_u}$$
$$O_u^{(13)} = h_u \mid p_\rho\big[\overline{k_u} \mid \langle\!| Q|\!\rangle^\lambda_\varepsilon\big] \mid k_u.\overline{h_u}$$
$$O_u^{(14)} = h_u \mid p_\rho\big[\langle\!| Q|\!\rangle^\lambda_\varepsilon\big] \mid \overline{h_u}$$
$$O_u^{(15)} = p_\rho\big[\langle\!| Q|\!\rangle^\lambda_\varepsilon\big]$$

2. otherwise, if $n > 0$ then $\langle P\rangle_{t,\rho} = \prod_{k=1}^{n} p_{t,\rho}[\langle\!| P'_k|\!\rangle_\varepsilon] \mid S$ and $p \in \{1, \ldots, 15+n+4m\}$.

$$O_u^{(1)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big]\big)$$
$$\mid u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon]$$
$$\mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]$$
$$O_u^{(2+4m+s-n)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_{u,\rho}[\langle\!| P'_i|\!\rangle_\varepsilon]\Big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big]\big) \mid \overline{r_u}.k_u.\overline{h_u}$$
$$\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]$$
$$O_u^{(3+4m+s-n)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_{u,\rho}[\langle\!| P'_i|\!\rangle_\varepsilon]\Big] \mid \mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big] \mid k_u.\overline{h_u}$$
$$\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]$$
$$O_u^{(4+4m+s-n+j)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}] \mid \prod_{i=1}^{s-n} p_{u,\rho}[\langle\!| P'_i|\!\rangle_\varepsilon]\Big] \mid h_u \mid p_{u,\rho}\langle\!\langle (X_1, \ldots, X_{n-j}).$$
$$(\prod_{k=1}^{n-j} p_\rho[X_k] \mid \prod_{k=1}^{j} p_\rho[\langle\!| P'_k|\!\rangle_\varepsilon] \mid m_u.\overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle\rangle\!\rangle.\overline{n_u})$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big] \mid k_u.\overline{h_u}$$
$$\mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]$$
$$O_u^{(4+4m+s)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle\!| P'_k|\!\rangle_\varepsilon] \mid m_u.\overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u}$$
$$\mid m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big]$$
$$\mid k_u.\overline{h_u} \mid v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q|\!\rangle^\lambda_\varepsilon] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]$$
$$O_u^{(5+4m+s)} = u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle\!| P'_k|\!\rangle_\varepsilon] \mid \overline{j_u}.u\langle\!\langle \dagger\rangle\!\rangle.\overline{n_u}$$
$$\mid p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}])\rangle\!\rangle\big]$$

$$
\begin{aligned}
& \mid k_u.\overline{h_u} \mid v_u\big[u_u\langle\langle(Z).(Z \mid e_u[\langle Q\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\langle(X).X\rangle\rangle.g_u)\rangle\rangle\big] \\
O_u^{(6+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{j_u}.u\langle\langle\dagger\rangle\rangle.\overline{n_u} \\
& \mid p_\rho\big[u_u\langle\langle(Z).(Z \mid e_u[\langle Q\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\langle(X).X\rangle\rangle.g_u)\rangle\rangle \\
& \mid \ \mid u_u[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u}]\big] \mid k_u.\overline{h_u} \\
O_u^{(7+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{j_u}.u\langle\langle\dagger\rangle\rangle.\overline{n_u} \\
& \mid p_\rho\big[\overline{f_u}.\overline{g_u}.j_u.n_u.\overline{k_u} \mid e_u[\langle Q\rangle_\varepsilon^\lambda] \mid f_u.e_u\langle\langle(X).X\rangle\rangle.g_u\big] \mid k_u.\overline{h_u} \\
O_u^{(8+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{j_u}.u\langle\langle\dagger\rangle\rangle.\overline{n_u} \\
& \mid p_\rho\big[\overline{g_u}.j_u.n_u.\overline{k_u} \mid e_u[\langle Q\rangle_\varepsilon^\lambda] \mid e_u\langle\langle(X).X\rangle\rangle.g_u\big] \mid k_u.\overline{h_u} \\
O_u^{(9+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{j_u}.u\langle\langle\dagger\rangle\rangle.\overline{n_u} \\
& \mid p_\rho\big[\overline{g_u}.j_u.n_u.\overline{k_u} \mid \langle Q\rangle_\varepsilon^\lambda \mid g_u\big] \mid k_u.\overline{h_u} \\
O_u^{(10+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{j_u}.u\langle\langle\dagger\rangle\rangle.\overline{n_u} \\
& \mid p_\rho\big[j_t.n_t.\overline{k_t} \mid \langle Q\rangle_\varepsilon^\lambda\big] \mid k_u.\overline{h_u} \\
O_u^{(11+4m+s)} &= u\Big[\langle F\rangle_\rho[h_u.\langle P\rangle_{\rho'}]\Big] \mid h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid u\langle\langle\dagger\rangle\rangle.\overline{n_u} \mid p_\rho\big[n_t.\overline{k_t} \mid \langle Q\rangle_\varepsilon^\lambda\big] \mid k_u.\overline{h_u} \\
O_u^{(12+4m+s)} &= h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \overline{n_t} \mid p_\rho\big[n_t.\overline{k_t} \mid \langle Q\rangle_\varepsilon^\lambda\big] \mid k_u.\overline{h_u} \\
O_u^{(13+4m+s)} &= h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \quad \mid p_\rho\big[\overline{k_u} \mid \langle Q\rangle_\varepsilon^\lambda\big] \mid k_u.\overline{h_u} \\
O_u^{(14+4m+s)} &= h_u \mid \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \quad \mid p_\rho\big[\langle Q\rangle_\varepsilon^\lambda\big] \mid \overline{h_u} \\
O_u^{(15+4m+s)} &= \prod_{k=1}^{j} p_\rho[\langle P'_k\rangle_\varepsilon] \mid \quad \mid p_\rho\big[\langle Q\rangle_\varepsilon^\lambda\big]
\end{aligned}
$$

The following definition formalizes all possible forms for the process $U_s^{(r)}(\langle H\rangle_s^\lambda[\langle P\rangle_{s,\rho}^\lambda], \langle R\rangle_\varepsilon^\lambda, \langle Q\rangle_\varepsilon^\lambda)$. Due to the simplicity of writing for the process $U_s^{(r)}(\langle H\rangle_s^\lambda[\langle P\rangle_{s,\rho}^\lambda], \langle R\rangle_\varepsilon^\lambda, \langle Q\rangle_\varepsilon^\lambda)$, we will use the abbreviation $U_s^{(r)}$ in all places where we do not violate the rationing of the content.

**Definition 5.4.6.** Let $P, Q, R$ be well-formed compensable processes. Given a name $s$, a path $\rho$, and $r \geq 1$, we define the intermediate processes $U_s^{(r)}$ as in the following:

$$
\begin{aligned}
U_s^{(1)} = s\Big[[\![H]\!]_s^\lambda[[\![P]\!]_{s,\rho}^\lambda]\Big] & \mid r_s.\big(\mathtt{extra}\langle\langle s, p_{s,\rho}, p_\rho\rangle\rangle \\
& \mid m_s.p_\rho\big[v_s\langle\langle(X).(X \mid u_s[\overline{f_s}.\overline{g_s}.n_s.k_s])\rangle\rangle\big]\big) \mid s\langle\langle(Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\rangle \\
& \mid v_s\big[e_s\langle\langle(Y).(\overline{g_s}.u_s\langle\langle(Z).(Z \mid e_s[[\![R]\!]_\varepsilon^\lambda] \\
& \mid f_s.e_s\langle\langle(X).X\rangle\rangle.g_s)\rangle\rangle)\rangle\rangle.(\overline{f_s}.e_s[\mathbf{0}]) \mid e_s[[\![Q]\!]_\varepsilon^\lambda] \mid f_s.e_s\langle\langle(X).X\rangle\rangle.g_s\big]
\end{aligned}
$$

$$\begin{aligned}
U_s^{(2)} = s\Big[&[\![H]\!]_s^{\lambda}[[\![P]\!]_{s,\rho}^{\lambda}]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \\
&\mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[\langle\!| R |\!\rangle_\varepsilon^\lambda\{^{\langle\!| Q |\!\rangle_\varepsilon^\lambda}/_Y\}] \\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid \overline{f_s}.e_s[\mathbf{0}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big] \\
U_s^{(3)} = s\Big[&[\![H]\!]_s^{\lambda}[[\![P]\!]_{s,\rho}^{\lambda}]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \\
&\mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[\langle\!| R |\!\rangle_\varepsilon^\lambda\{^{\langle\!| Q |\!\rangle_\varepsilon^\lambda}/_Y\}] \\
&\mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle \mid e_s[\mathbf{0}] \mid e_s\langle\!\langle (X).X\rangle\!\rangle.g_s\big] \\
U_s^{(4)} = s\Big[&[\![H]\!]_s^{\lambda}[[\![P]\!]_{s,\rho}^{\lambda}]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \\
&\mid v_s\big[\overline{g_s}.u\langle\!\langle (Z).(Z \mid e_s[\langle\!| R |\!\rangle_\varepsilon^\lambda\{^{\langle\!| Q |\!\rangle_\varepsilon^\lambda}/_Y\}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}]) \mid g_s\big] \\
U_s^{(5)} = s\Big[&[\![H]\!]_s^{\lambda}[[\![P]\!]_{s,\rho}^{\lambda}]\Big] \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho}, p_\rho\rangle\!\rangle \\
&\mid m_s.p_\rho\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.k_s])\rangle\!\rangle\big]\big) \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \\
&\mid v_s\big[u\langle\!\langle (Z).(Z \mid e_s[\langle\!| R |\!\rangle_\varepsilon^\lambda\{^{\langle\!| Q |\!\rangle_\varepsilon^\lambda}/_Y\}] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]
\end{aligned}$$

For the proof of operational correspondence we need the following statement:

**Lemma 5.4.3.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\langle\!| P |\!\rangle_\rho^\lambda \equiv \langle\!| Q |\!\rangle_\rho^\lambda$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ■

We now state our operational correspondence result:

**Theorem 5.4.4** (Operational Correspondence for $\langle\!| \cdot |\!\rangle_\rho^\lambda$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{A}}$.

(1) If $P \longrightarrow P'$ then $\langle\!| P |\!\rangle_\varepsilon^\lambda \longrightarrow^k \langle\!| P' |\!\rangle_\varepsilon^\lambda$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 15 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 15 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{P}}(F[P_1])$,

d) $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$ it follows $k = 5$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$ processes $P_1, Q, P_2, R$ and names $t, u, s$.

(2) If $\langle\!| P |\!\rangle_\varepsilon^\lambda \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* \langle\!| P' |\!\rangle_\varepsilon^\lambda$.

*Proof.* As in all previously presented encodings, in the following we consider completeness and soundness (Parts (1) and (2)) separately.

(1) **Part (1) – Completeness:** The proof proceeds by induction on the derivation of $P \longrightarrow P'$. We consider three base cases, corresponding to cases *a*), *b*) and *c*) of Proposition 2.2.3 (Page 18). In all cases, we use Lemma 5.4.3, Definition 5.4.2 and Lemma 3.2.9 (Page 47) that applies also for $\langle\!| \cdot |\!\rangle_\rho^\lambda$.

a) This case concerns an input-output synchronization on a name $a \in \mathcal{N}_s$. Therefore, we observe that $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$, and we have that derivation corresponds to the derivation presented in (3.30). Therefore, the thesis holds with $k = 1$.

b) This case concerns a synchronization due to an external error notification for a transaction scope. We consider $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$, with $n = \mathtt{pb_A}(P_1)$, $m = \mathtt{ts_A}(P_1)$ and $s = \mathtt{S}(P_1)$, and $P' \equiv E[C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$. We have the following derivation:

$$
\begin{aligned}
\langle\!| P |\!\rangle^{\lambda}_{\varepsilon} &\equiv \langle\!| E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]] |\!\rangle^{\lambda}_{\varepsilon}\\
&= \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C[t[P_1,Q]] |\!\rangle^{\lambda}_{\rho} \mid \langle\!| D[\overline{t}.P_2] |\!\rangle^{\lambda}_{\rho}\Big]\\
&= \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}[\langle\!| t[P_1,Q] |\!\rangle^{\lambda}_{\rho'}] \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[\langle\!| \overline{t}.P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
&= \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}\Big[t\Big[\langle\!| P |\!\rangle^{\lambda}_{t,\rho}\Big] \mid r_t.\big(t\langle\!\langle (Y).\big(t[Y] \mid \mathsf{ch}(t,Y)\\
&\quad \mid \mathtt{outd}^{\mathtt{s}}(p_{t,\rho}, p_\rho, \mathtt{nl}(p_{t,\rho},Y), \overline{m_t}.\overline{j_t}.t\langle\!\langle \dagger\rangle\!\rangle.\overline{n_t}))\rangle\!\rangle\\
&\quad \mid m_t.p_\rho\big[v_t\langle\!\langle (X).(X \mid u_t[\overline{f_t}.\overline{g_t}.n_t.k_t])\rangle\!\rangle\big]\big) \mid t\langle\!\langle (Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\rangle\!\rangle\\
&\quad \mid v_t\big[u_t\langle\!\langle (Z).(Z \mid e_t[\langle\!| Q |\!\rangle^{\lambda}_{\varepsilon}] \mid f_t.e_t\langle\!\langle (X).X\rangle\!\rangle.g_t)\rangle\!\rangle\big]\Big] \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[\overline{t}.h_t.\langle\!| P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
&\longrightarrow \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}[I^{(1)}_t(\langle\!| P_1 |\!\rangle^{\lambda}_{t,\rho'}, \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon})] \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[\overline{t}.h_t.\langle\!| P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
\longrightarrow^{2+4m+s-n}\ &\langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}[[I^{(3+4m+s-n)}_t(\langle\!| P_1 |\!\rangle^{\lambda}_{t,\rho'}, \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon}) \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[h_t.\langle\!| P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
\longrightarrow^{n+11}\ &\langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}[[I^{(14+4m+s)}_t(\langle\!| P_1 |\!\rangle^{\lambda}_{t,\rho'}, \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon}) \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[h_t.\langle\!| P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
&\longrightarrow \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C |\!\rangle^{\lambda}_{\rho}[\langle\!| \mathsf{extr_A}(P_1) \mid \langle Q\rangle |\!\rangle^{\lambda}_{\rho'}] \mid \langle\!| D |\!\rangle^{\lambda}_{\rho}[\langle\!| P_2 |\!\rangle^{\lambda}_{\rho''}]\Big]\\
&= \langle\!| E |\!\rangle^{\lambda}_{\varepsilon}\Big[\langle\!| C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle] |\!\rangle^{\lambda}_{\rho} \mid \langle\!| D[P_2] |\!\rangle^{\lambda}_{\rho}\Big]\\
&= \langle\!| E[C[\mathsf{extr_A}(P_1) \mid \langle Q\rangle] \mid D[P_2]] |\!\rangle^{\lambda}_{\varepsilon}\\
&\equiv \langle\!| P' |\!\rangle^{\lambda}_{\varepsilon}
\end{aligned}
$$

Therefore, we can conclude that $\langle\!| P |\!\rangle_{\varepsilon} \longrightarrow^{k} \langle\!| P' |\!\rangle^{\lambda}_{\varepsilon}$ for $k = 15 + 4m + s$. Process $I^{(p)}_t(\langle\!| P_1 |\!\rangle^{\lambda}_{t,\rho'}, \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon})$, where $p \in \{1, \ldots, 15 + 4m + n\}$, is as in Definition 5.4.4.

c) This case concerns a synchronization due to an internal error notification (i.e., the error comes from the default activity of transaction). Here we have $P \equiv C[t[F[\overline{u}.P_1],Q]]$, with $n = \mathtt{pb_A}(F[P_1])$, $m = \mathtt{ts_A}(P_1)$, $s = \mathtt{S}(P_1)$ and $P' \equiv C[\mathsf{extr_A}(F[P_1]) \mid \langle Q\rangle]$. Then we have the following derivation:

$$
\begin{aligned}
\langle\!| P |\!\rangle^{\lambda}_{\varepsilon} &\equiv \langle\!| C\big[u[F[\overline{u}.P_1],Q]\big] |\!\rangle^{\lambda}_{\varepsilon}\\
&= \langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[\langle\!| u[F[\overline{u}.P_1],Q] |\!\rangle^{\lambda}_{\rho}\big]\\
&= \langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[u\big[\langle\!| F[\overline{u}.P_1] |\!\rangle^{\lambda}_{u,\rho}\big] \mid r_u.\big(\mathtt{extra}\langle\!\langle u, p_{u,\rho}, p_\rho\rangle\!\rangle\\
&\quad \mid\ m_u.p_\rho\big[v_u\langle\!\langle (X).(X \mid u_u[\overline{f_u}.\overline{g_u}.n_u.\overline{k_u}])\rangle\!\rangle\big]\big) \mid u.u\langle\!\langle (Y).u[Y] \mid \mathcal{T}_u(Y).\overline{h_u}\rangle\!\rangle\\
&\quad \mid\ v_u\big[u_u\langle\!\langle (Z).(Z \mid e_u[\langle\!| Q |\!\rangle^{\lambda}_{\varepsilon}] \mid f_u.e_u\langle\!\langle (X).X\rangle\!\rangle.g_u)\rangle\!\rangle\big]\big]\\
&\longrightarrow \langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[O^{(1)}_u(\langle\!| F |\!\rangle^{\lambda}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\lambda}_{\rho'}], \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon})\big]\\
\longrightarrow^{2+4m+s-n}\ &\langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[O^{(3+4m+s-n)}_u(\langle\!| F |\!\rangle^{\lambda}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\lambda}_{\rho'}], \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon})\big]\\
\longrightarrow^{n+12}\ &\langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[O^{(15+4m+s)}_u(\langle\!| F |\!\rangle^{\lambda}_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^{\lambda}_{\rho'}], \langle\!| Q |\!\rangle^{\lambda}_{\varepsilon})\big]\\
&\equiv \langle\!| C |\!\rangle^{\lambda}_{\varepsilon}\big[\langle\!| \mathsf{extr_A}(F[P_1]) |\!\rangle^{\lambda}_{\rho} \mid p_\rho[\langle\!| Q |\!\rangle^{\lambda}_{\varepsilon}]\big]\\
&= \langle\!| C\big[\mathsf{extr_A}(F[P_1]) \mid \langle Q\rangle\big] |\!\rangle^{\lambda}_{\varepsilon}
\end{aligned}
$$

$$\equiv \langle\!| P' |\!\rangle^\lambda_\varepsilon$$

Process $O^{(q)}_u(\langle\!| F |\!\rangle^\lambda_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^\lambda_{\rho'}], [\![Q]\!]_\varepsilon)$, where $q \in \{1,\ldots,15+4m+n\}$, is as in Definition 5.4.5. The role of function $\mathsf{ch}(u,\cdot)$ is central. Indeed, $\mathsf{ch}(u, \langle\!| F |\!\rangle^\lambda_{u,\rho}[h_u.\langle\!| P_1 |\!\rangle^\lambda_{\rho'}])$ provides the input $h_u$ which is necessary to achieve operational correspondence.
The order and number of reduction steps can be explained as in Case b) above. We can then conclude that $[\![P]\!]_\varepsilon \longrightarrow^k [\![P']\!]_\varepsilon$ where $k = 15+4m+s$.

d) We have that $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1,Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$. We will use Definition 5.2.6 for process $U^{(r)}_s(\langle\!| H |\!\rangle^\lambda_{s,\rho}[\langle\!| P_1 |\!\rangle^\lambda_{s,\rho'}], (\!|Q|\!)^\lambda_\varepsilon)$ where $r \in \{1,\ldots,5\}$ and have the following:

$$\begin{aligned}
\langle\!| P |\!\rangle^\lambda_\varepsilon &\equiv \langle\!| C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1,Q]]] |\!\rangle^\lambda_\varepsilon\\
&= \langle\!| C |\!\rangle^\lambda_\varepsilon\big[s\Big[\langle\!| H |\!\rangle^\lambda_{s,\rho}[\langle\!| \mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1 |\!\rangle^\lambda_{s,\rho'}]\Big]\\
&\quad \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho'}, p_{\rho'}\rangle\!\rangle \mid m_s.p_{\rho'}\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.n_s.\overline{k_s}])\rangle\!\rangle\big]\big)\big]\\
&\quad \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[\langle\!| Q |\!\rangle^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\\
&= \langle\!| C |\!\rangle^\lambda_\varepsilon\big[s\Big[\langle\!| H |\!\rangle^\lambda_{s,\rho}[u_s\Big[e_s\langle\!\langle (Y).(\overline{g_s}.u_s\langle\!\langle (Z).(Z \mid e_s[\langle\!| R |\!\rangle^\lambda_\varepsilon]\\
&\quad \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle)\rangle\!\rangle.(\overline{f_s}.e_s[\mathbf{0}])\Big] \mid \langle\!| P_1 |\!\rangle^\lambda_{s,\rho}]\Big]\\
&\quad \mid r_s.\big(\mathtt{extra}\langle\!\langle s, p_{s,\rho'}, p_{\rho'}\rangle\!\rangle \mid m_s.p_{\rho'}\big[v_s\langle\!\langle (X).(X \mid u_s[\overline{f_s}.\overline{g_s}.n_s.\overline{k_s}])\rangle\!\rangle\big]\big)\big]\\
&\quad \mid s\langle\!\langle (Y).s[Y] \mid \mathcal{T}_s(Y).\overline{h_s}\rangle\!\rangle \mid v_s\big[u_s\langle\!\langle (Z).(Z \mid e_s[\langle\!| Q |\!\rangle^\lambda_\varepsilon] \mid f_s.e_s\langle\!\langle (X).X\rangle\!\rangle.g_s)\rangle\!\rangle\big]\big]\\
&\longrightarrow \langle\!| C |\!\rangle^\lambda_\varepsilon\big[U^{(1)}_s(\langle\!| H |\!\rangle^\lambda_{s,\rho}[\langle\!| P_1 |\!\rangle^\lambda_{s,\rho'}], \langle\!| Q |\!\rangle^\lambda_\varepsilon)\big]\\
&\longrightarrow^4 \langle\!| C |\!\rangle^\lambda_\varepsilon\big[U^{(5)}_s(\langle\!| H |\!\rangle^\lambda_{s,\rho}[\langle\!| P_1 |\!\rangle^\lambda_{s,\rho'}], \langle\!| Q |\!\rangle^\lambda_\varepsilon)\\
&\equiv \langle\!| C[s[H[P_1],R\{^Q/_Y\}]] |\!\rangle^\lambda_\varepsilon
\end{aligned}$$

Therefore, $k = 5$.

(2) **Part (2) – Soundness:** For the proof of soundness we use auxiliary results presented in Paragraph 3.2.3.2.3 by use of encoding of aborting semantics instead of encoding of discarding semantics. Also the proof use Definition 5.4.4, Definition 5.4.5, and Definition 5.4.6. Therefore, the proof of soundness follows the explanation presented in Roadmap 3.2.3.2.5. Also, the proof uses the same derivation that is presented in the proof of soundness for translation $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$ (cf. Item 2 – Soundness).

■

**Brief summary of the chapter**:
In this chapter, we introduced all preliminaries for encodings $\mathcal{C}^\lambda$ into $\mathcal{A}$ and informally acquaint the reader with the basic intuition of the encodings. Also, the main result is the *valid encodings* of calculus for compensable processes with dynamic update into the calculus of adaptable processes with the subjective update (encodings $\mathcal{C}^\lambda_{\mathtt{D}}$, $\mathcal{C}^\lambda_{\mathtt{P}}$, and $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{S}$).

Encodings $\mathcal{C}^\lambda_{\mathtt{D}}$, $\mathcal{C}^\lambda_{\mathtt{P}}$, and $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$, which is analyzed in the next chapter, follow and mimic the basic intuition of the encoding presented in this chapter. Therefore, we believe that it will be easier for the reader to follow and understand the results from the upcoming chapter.

# CHAPTER 6

## Encoding Dynamic Compensation Processes into Adaptable Processes with Objective Update

In this chapter, we developed translations of compensable processes with dynamic compensations, denoted $\mathcal{C}^\lambda$, under discarding, preserving, and aborting semantics into adaptable processes with *objective* mobility. In the following is given a brief structure of the chapter:

**Section 6.1** presents the translation of $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{O}$. Then the formal definition of the encoding follows. We prove that the encoding satisfies *compositionality*, *name invariance* and *operational correspondence (completeness and soundness)*. Also, in this section we discuss a *efficiency* criterion. The encoding $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$ provides tighter operational correspondences result. Therefore, we prove that the encoding $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{O}$.

**Section 6.2** presents the translation of $\mathcal{C}^\lambda_{\mathtt{P}}$ into $\mathcal{O}$. Also, we present the formal definition of the encoding. We prove that the encoding satisfies *name invariance* and *operational correspondence (completeness and soundness)*. The encoding $\mathcal{C}^\lambda_{\mathtt{P}}$ into $\mathcal{S}$ provides tighter operational correspondences result. Therefore, we prove that the encoding $\mathcal{C}^\lambda_{\mathtt{P}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}^\lambda_{\mathtt{P}}$ into $\mathcal{O}$.

**Section 6.3** presents the translation of $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$. Also, we present the formal definition of the encoding and prove that it satisfies *compositionality*, *name invariance* and *operational correspondence (completeness and soundness)*. The encoding $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{S}$ provides tighter operational correspondences result. Therefore, we prove that the encoding $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{S}$ is better suited than the encoding $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$.

### 6.1 Translating $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{O}$

The translation $\mathcal{C}^\lambda_{\mathtt{D}}$ into $\mathcal{O}$, denoted $[\![\cdot]\!]^{\lambda_o}_\rho$, extends the key ideas of the encoding $[\![\cdot]\!]^{o}_\rho$ (cf. Section 4.1).

**Remark 6.1.1.** The translation requires sets of *reserved names* and therefore we need to revised Definition 3.1.2 as in the following:

(i) the set of *reserved location names* $\mathcal{N}^r_l$ is unchanged,

(ii) the set of *reserved synchronization names* is extended with name $z_x$ such that:

$$\mathcal{N}^r_s = \{h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, z_x \mid x \in \mathcal{N}_t\}.$$

$$
\begin{aligned}
[\![\langle P\rangle]\!]^{\lambda_o}_{\rho} &= p_\rho\big[[\![P]\!]^{\lambda_o}_{\varepsilon}\big] \\
[\![t[P,Q]]\!]^{\lambda_o}_{\rho} &= t\Big[[\![P]\!]^{\lambda_o}_{t,\rho}\Big] \mid t.\big(\mathtt{extrd}\{t, p_{t,\rho}, p_\rho\} \mid m_t.p_\rho\big[u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\big) \\
&\qquad \mid v_t\big[u_t\{(Z).(Z \mid e_t[[\![Q]\!]^{\lambda_o}_{\varepsilon}] \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\}\big] \\
[\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P]\!]^{\lambda_o}_{t,\rho} &= u_t\big[e_t\{(Y).(\overline{g_t}.v_t\{(X).(v_t\big[u_t\{(Z).(Z \mid e_t[[\![R]\!]^{\lambda_o}_{\varepsilon}] \\
&\qquad \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\})\big]\}\}.(\overline{f_t}.e_t[\mathbf{0}])\big] \mid [\![P]\!]^{\lambda_o}_{t,\rho} \\
[\![\bar{t}.P]\!]^{\lambda_o}_{\rho} &= \bar{t}.h_t.[\![P]\!]^{\lambda_o}_{\rho}
\end{aligned}
$$

Figure 6.1: Translating $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{O}$.

We use the function for determining the number of locations as in Remark 5.2.2. Also, we use process $\mathtt{outd}^{\circ}$ as defined for $[\![\cdot]\!]^{\circ}_{\rho}$ (cf. (4.1)).

We need the following additional auxiliary process:

**Definition 6.1.1** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extrd}\{t, l_1, l_2\}$ to stand for the following (subjective) update prefix:

$$
\mathtt{extrd}\{t, l_1, l_2\} = t\{(Y).\,\big(t[Y] \mid \mathtt{ch}(t, Y) \mid \mathtt{outd}^{\circ}(l_1, l_2, \mathtt{nl}(l, Y), \overline{m_t}.\overline{k_t}.t\{\dagger\}.\overline{h_t})\big)\} \tag{6.1}
$$

The intuition for the process $\mathtt{extrd}\{t, l_1, l_2\}$ is the same as in the translation of $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{S}$ (cf. Definition 5.2.1).

Using well-formed compesable processes (cf. Section 5.1.1), the translation of $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{S}$ extends Definition 3.2.3 (see Page 39) as follows:

**Definition 6.1.2** (Translating $\mathcal{C}^{\lambda}_{\mathtt{D}}$ into $\mathcal{S}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with dynamic recovery into (subjective) adaptable processes as a tuple $([\![\cdot]\!]^{\lambda}_{\rho}, \varphi_{[\![\cdot]\!]^{\lambda}_{\rho}})$ where:

(a) The renaming policy

$$
\varphi_{[\![\cdot]\!]^{\lambda}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, z_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t. \end{cases}
$$

(b) The translation $[\![\cdot]\!]^{\lambda_o}_{\rho} : \mathcal{C}^{\lambda}_{\mathtt{D}} \to \mathcal{O}$ is as in Figure 6.1 and as a homomorphism for other operators.

### 6.1.1 Translation Correctness

We now establish that the translation $[\![\cdot]\!]^{\lambda_o}_{\rho}$ is a valid encoding. To this end, we address the three criteria in Definition 2.3.5: compositionality, name invariance, and operational correspondence. Other criteria are left as a research topic for future work.

#### 6.1.1.1 Structural Criteria

We prove the two criteria, compositionality and name invariance.

##### 6.1.1.1.1 Compositionality

As previously stated, the compositionality criterion states that a composite term's translation must be defined in terms of its subterms' translations. To mediate between these translations of subterms, we define a *context* for each process operator, which depends on free names of the subterms:

**Definition 6.1.3** (Compositional context for $\mathcal{C}_{\mathtt{D}}^{\lambda}$)**.** For all process operator from $\mathcal{C}_{\mathtt{D}}^{\lambda}$, instead transaction we define a compositional context in $\mathcal{O}$ as in Definition 5.2.3. For transaction and compensation update compositional contexts are as follows:

$$\begin{aligned} C_{t[,],\rho}[\bullet_1, \bullet_2] = {} & t\Big[[\bullet_1]\Big] \mid t.\big(\mathtt{extrd}\{t, p_{t,\rho}, p_\rho\} \mid m_t.p_\rho\big[v_t\{(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.k_t])\}\big]\big) \\ & \mid v_t\big[u_t\{(Z).(Z \mid e_t[[\bullet_2]] \mid f_t.e_t\{(X).X\}.g_t)\}\big] \\ C_{\mathtt{inst},\rho}[\bullet_1, \bullet_2] = {} & u_t\Big[e_t\{(Y).(\overline{g_t}.u_t\{(Z).(Z \mid e_t[[\bullet_1]] \mid f_t.e_t\{(X).X\}.g_t)\})\}.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\bullet_2] \\ C_Y[\bullet_1] = {} & [\bullet_1] \end{aligned}$$

Using this definition, we may now state the following result.

**Theorem 6.1.2** (Compositionality for $[\![\cdot]\!]_\rho^\lambda$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}_{\mathtt{D}}^{\lambda}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$\begin{array}{lll} [\![\langle P\rangle]\!]_\rho^{\lambda_o} = C_{\langle\rangle,\rho}\,[[\![P]\!]_\varepsilon^{\lambda_o}] & [\![t[P,Q]]\!]_\rho^{\lambda_o} = C_{t[,],\rho}\,\Big[[\![P]\!]_{t,\rho}^{\lambda_o}, [\![Q]\!]_\varepsilon^{\lambda_o}\Big] & [\![P \mid Q]\!]_\rho^{\lambda_o} = C_{\mid}\,[[\![P]\!]_\rho^{\lambda_o}, [\![Q]\!]_\rho^{\lambda_o}] \\ [\![a.P]\!]_\rho^{\lambda_o} = C_{a.}\,[[\![P]\!]_\rho^{\lambda_o}] & [\![\overline{t}.P]\!]_\rho^{\lambda_o} = C_{\overline{t}.}\,[[\![P]\!]_\rho^{\lambda_o}] & [\![(\nu x)P)]\!]_\rho^{\lambda_o} = C_{(\nu x)}\,[[\![P]\!]_\rho^{\lambda_o}] \\ [\![\overline{a}.P]\!]_\rho^{\lambda_o} = C_{\overline{a}.}\,[[\![P]\!]_\rho^{\lambda_o}] & [\![!\pi.P]\!]_\rho^{\lambda_o} = C_{!\pi.}\,[[\![P]\!]_\rho^{\lambda_o}] & \\ [\![Y]\!]_\rho^{\lambda_o} = C_Y\,[[\![Y]\!]_\rho^{\lambda_o}] & [\![\mathtt{inst}\lfloor\lambda Y.R\rfloor.P]\!]_{t,\rho}^{\lambda_o} = C_{\mathtt{inst},\rho}[[\![R]\!]_\varepsilon^{\lambda_o}, [\![P]\!]_{t,\rho}^{\lambda_o}] & \end{array}$$

*Proof.* The proof proceeds in the same direction as the proof of Theorem 5.2.3. ■

##### 6.1.1.1.2 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 6.1.2 (a).

**Theorem 6.1.3** (Name invariance for $[\![\cdot]\!]_\rho^{\lambda_o}$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{[\![\cdot]\!]_{\sigma(\rho)}^{\lambda_o}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{[\![\cdot]\!]_\rho^{\lambda_o}}(x)\}$, and

(ii) $[\![\sigma(P)]\!]_{\sigma(\rho)}^{\lambda_o} = \sigma'([\![P]\!]_\rho^{\lambda_o})$.

*Proof.* The proof proceeds in the same direction as the proof of Theorem 5.2.4. ■

#### 6.1.1.2 Semantic Criteria

In this subsection we prove that translation $\mathcal{C}_{\mathtt{D}}^{\lambda}$ into $\mathcal{O}$ satisfied operational correspondence (completeness and soundness).

##### 6.1.1.2.1 Operational Correspondence

For the proof of operational correspondence we need the following statement:

**Lemma 6.1.4.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $[\![P]\!]_\rho^{\lambda_o} \equiv [\![Q]\!]_\rho^{\lambda_o}$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ∎

Operational correspondence for translation of dynamic compensable processes with discarding semantics into adaptable processes with objective update is given in the following theorem:

**Theorem 6.1.5** (Operational Correspondence for $[\![\cdot]\!]_\rho^{\lambda_o}$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{D}}^{\lambda}$. We have:

1. If $P \xrightarrow{\tau} P'$ then $[\![P]\!]_\varepsilon^{\lambda_o} \longrightarrow^k [\![P']\!]_\varepsilon^{\lambda_o}$ where either

   a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,
   b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 11 + \mathtt{pb}_{\mathtt{D}}(P_1) + \mathtt{Z}_{\mathtt{d}}(P_1)$ or
   c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 11 + \mathtt{pb}_{\mathtt{D}}(F[P_1]) + \mathtt{Z}_{\mathtt{d}}(F[P_1])$,
   d) $P \equiv C[s[H[\mathtt{inst}\lfloor \lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$ it follows $k = 6$,

   for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$, processes $P_1, Q, P_2, R$, and name $t, u, s$.

2. If $[\![P]\!]_\varepsilon^{\lambda_o} \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* [\![P']\!]_\varepsilon^{\lambda_o}$.

*Proof.* The proof proceeds in the same direction as the proof of Theorem 5.2.12. ∎

### 6.1.2 Comparing Subjective vs Objective update

In this subsection, we provide a theorem which states that subjective updates are better suited to encode compensation handling with dynamic compensation and discarding semantics than objective updates.

The following statement is a corollary of Theorem 6.1.5.

**Corollary 6.1.6.** Let $P$ be a well-formed process in $\mathcal{C}^{\lambda}$. If $P \longrightarrow P'$ and $[\![P]\!]_\varepsilon^{\lambda_o} \longrightarrow^k [\![P']\!]_\varepsilon^{\lambda_o}$ then:

b) if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{D}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ then $k \geq 11 + \mathtt{pb}_{\mathtt{D}}(P_1) + \mathtt{Z}_{\mathtt{d}}(P_1)$,

c) if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{D}}(F[P_1]) \mid \langle Q\rangle]$ then $k \geq 11 + \mathtt{pb}_{\mathtt{D}}(F[P_1]) + \mathtt{Z}_{\mathtt{d}}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

**Theorem 6.1.7.** The encoding $[\![\cdot]\!]_\rho^{\lambda} : \mathcal{C}_{\mathtt{D}}^{\lambda} \longrightarrow \mathcal{S}$ is as or more efficient than $[\![\cdot]\!]_\rho^{\lambda_o} : \mathcal{C}_{\mathtt{D}}^{\lambda} \longrightarrow \mathcal{O}$.

*Proof.* The proof proceeds similarly to the proof for Theorem 4.1.13 by using: Theorem 5.2.12, theorem 6.1.5, Proposition 2.2.3, and Corollary 6.1.6. ∎

## 6.2 Translating $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{O}$

In this section the translation $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{O}$, denoted $(\!|\cdot|\!)_\rho^{\lambda_o}$ is presented. This translation relies on the idea and principles of encoding of $\mathcal{C}_{\mathtt{P}}$ into $\mathcal{O}$ (cf. Section 4.2) and encoding of $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{S}$ (cf. Section 5.3).

**Remark 6.2.1.** We require sets of *reserved names* and need to revise Definition 3.1.2 as in the following:

(i) the set of *reserved location names* $\mathcal{N}_l^r$ is unchanged and,

(ii) the set of *reserved synchronization names* is extended such that

$$\mathcal{N}_s^r = \{h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, j_x, r_x, z_x \mid x \in \mathcal{N}_t\}.$$

Accordingly, the function that counts the number of protected blocks is as in Figure 3.4, while the function that counts the number of transactions is as in Definition 5.3.1.

Below we give a formal definition of the translation $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{O}$. We instruct the reader that this translation relies directly on the ideas that are presented in Section 5.3.

### 6.2.1 Translation Correctness

For process extrp we will use process $\mathtt{outp}^{\circ}(t, P, l_1, l_1', l_2, l_2', n, m)$. Initially, for the definition of $\mathtt{outp}^{\circ}(t, P, l_1, l_1', l_2, l_2', n, m)$ we introduce the following auxiliary processes:

$$\mathtt{outp}_1^{\circ}(t, l_1, l_1', n) = l_1\{(X_1, \ldots, X_n).z_t\{(Z).\left(\prod_{i=1}^{n} l_1'[X_i] \mid \overline{m_t}.\overline{k_t}.t\{\dagger\}.\overline{j_t}.r_t\right)\}\}.z_t[\mathbf{0}];$$

$$\mathtt{outp}_2^{\circ}(t, t_1, \ldots, t_m, l_2, l_2', m) = l_2\{(Y_1, \ldots, Y_m).$$
$$z_t\{(Z).\left(r_t.\left(\prod_{k=1}^{m}\left(l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}\right)\right) \mid \overline{m_t}.\overline{k_t}.t\{\dagger\}.\overline{j_t}\right)\}\}.z_t[\mathbf{0}];$$

$$\mathtt{outp}_3^{\circ}(t, t_1, \ldots, t_m, l_1, l_1', l_2, l_2', n, m) = l_1\{(X_1, \ldots, X_n).l_2\{(Y_1, \ldots, Y_m).$$
$$z_t\{(Z).\left(\prod_{i=1}^{n} l_1'[X_i] \mid r_t.\left(\prod_{k=1}^{m}\left(l_2'[\mathcal{E}(Y_k, t)] \mid j_{t_k}.l_2'\{(X).X\}.\overline{r_{t_k}}.\overline{h_{t_k}}\right)\right) \mid \overline{m_t}.\overline{k_t}.t\{\dagger\}.\overline{j_t}\right)\}\}\}.z_t[\mathbf{0}]$$

The auxiliary process $\mathtt{outp}^{\circ}(t, P, l_1, l_1', l_2, l_2', n, m)$ where $\mathtt{top}(l_2, P) = \{t_1, \ldots, t_m\}$ (cf. Definition 3.3.3) for $m > 0$ is now defined as follows:

$$\mathtt{outp}^{\mathtt{s}}(t, P, l_1, l_1', l_2, l_2', n, m) = \begin{cases} \overline{m_t}.\overline{k_t}.t\{\dagger\}.\overline{j_t}.r_t & \text{if } n, m = 0 \\ \mathtt{outp}_1^{\circ}(t, l_1, l_1', n) & \text{if } n > 0, m = 0 \\ \mathtt{outp}_2^{\circ}(t, t_1, \ldots, t_m, l_2, l_2', m) & \text{if } n = 0, m > 0 \\ \mathtt{outp}_3^{\circ}(t, t_1, \ldots, t_m, l_1, l_1', l_2, l_2', n, m) & \text{if } n, m > 0 \end{cases} \tag{6.2}$$

This process is similar to the process (4.4). The difference is in the process that is placed in the objective update on the name $z_t$. In (6.2) there is additional name $\overline{m_t}$.

**Definition 6.2.1** (Update Prefix for Extraction)**.** Let $t$, $l_1, l_1', l_2$, and $l_2'$ be names nd $P$ is an adaptable process. We write $\mathtt{extrp}\{t, P, l_1, l_1', l_2, l_2'\}$ to stand for the following (subjective) update prefix:

$$\begin{aligned}\mathtt{extrp}\{t, P, l_1, l_1', l_2, l_2'\} = t\{(Y).t[Y] \mid \mathsf{ch}(t, Y) \\ \mid \mathtt{outp}^{\circ}(t, P, l_1, l_1', l_2, l_2', \mathtt{nl}(l_1, Y), \mathtt{nl}(l_2, Y))\}\end{aligned} \tag{6.3}$$

The intuition for the process $\mathtt{extrp}\{t, P, l_1, l_1', l_2, l_2'\}$ in preserving semantics with dynamic recovery is the same as in static recovery (cf. Definition 4.2.1).
Based on the above modifications, the encoding of processes with dynamic compensations is given with the following definition:

**Definition 6.2.2** (Translating $\mathcal{C}_{\mathtt{P}}^{\lambda}$ into $\mathcal{O}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with preserving semantics into (subjective) adaptable processes as a tuple $(\llparenthesis\cdot\rrparenthesis_{\rho}^{\lambda_o}, \varphi_{\llparenthesis\cdot\rrparenthesis_{\rho}^{\lambda_o}})$ where:

$$
\begin{aligned}
(\!|\langle P\rangle|\!)^{\lambda_o}_{\rho} &= p_\rho\big[(\!|P|\!)^{\lambda_o}_{\epsilon}\big]\\
(\!|t[P,Q]|\!)^{\lambda_o}_{\rho} &= \beta_\rho\Big[t\big[(\!|P|\!)^{\lambda_o}_{t,\rho}\big] \mid t.\big(\mathtt{extrp}\{t,(\!|P|\!)^{\lambda_o}_{t,\rho},p_{t,\rho},p_\rho,\beta_{t,\rho},\beta_\rho\} \mid m_t.p_\rho\big[u_t[\overline{f_t}.\overline{g_t}.k_t]\big]\big)\\
&\qquad \mid v_t\big[u_t\{(Z).(Z \mid e_t[(\!|Q|\!)^{\lambda_o}_{\epsilon}] \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\}\big]\Big]\\
&\qquad \mid j_t.\beta_\rho\{(X).X\}.\overline{r_t}.\overline{h_t}\\
(\!|\mathtt{inst}\lfloor\lambda Y.R\rfloor.P|\!)^{\lambda_o}_{t,\rho} &= u_t\Big[e_t\{(Y).(\overline{g_t}.v_t\{(X).v_t[u_t\{(Z).(Z \mid e_t[(\!|R|\!)^{\lambda_o}_{\epsilon}]\\
&\qquad \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\})]\}\}.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid (\!|P|\!)^{\lambda_o}_{t,\rho}\\
(\!|\overline{t}.P|\!)^{\lambda_o}_{\rho} &= \overline{t}.h_t.(\!|P|\!)^{\lambda_o}_{\rho}
\end{aligned}
$$

Figure 6.2: Translating $\mathcal{C}^{\lambda}_{\mathtt{P}}$ into $\mathcal{O}$.

(a) The renaming policy $\varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}} : \mathcal{N}_c \to \mathcal{P}(\mathcal{N}_a)$ is defined with

$$
\varphi_{(\!|\cdot|\!)^{\lambda}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, u_x, v_x, e_x, g_x, f_x, j_x, r_x, z_x\} \cup \{p_\rho, \beta_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t \end{cases} \tag{6.4}
$$

(b) The translation $(\!|\cdot|\!)^{\lambda_o}_{\rho} : \mathcal{C}^{\lambda}_{\mathtt{P}} \to \mathcal{O}$ is as in Figure 6.2 and as a homomorphism for other operators.

#### 6.2.1.1 Structural Criteria

We prove that translation $\mathcal{C}^{\lambda}_{\mathtt{P}}$ ito $\mathcal{O}$ satisfies name invariance (cf. Definition 2.3.5). Analysis of compositionality is left for future research work.

##### 6.2.1.1.1 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 5.3.3 (a).

**Theorem 6.2.2** (Name invariance for $(\!|\cdot|\!)^{\lambda_o}_{\rho}$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{(\!|\cdot|\!)^{\lambda_o}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{(\!|\cdot|\!)^{\lambda_o}_{\rho}}(x)\}$, and

(ii) $(\!|\sigma(P)|\!)^{\lambda_o}_{\sigma(\rho)} = \sigma'((\!|P|\!)^{\lambda_o}_{\rho})$.

*Proof.* The proof follows the idea presented in the proof of Theorem 5.2.4. ∎

#### 6.2.1.2 Semantic Criteria

In this subsection we prove that translation $\mathcal{C}^{\lambda}_{\mathtt{P}}$ into $\mathcal{O}$ satisfies operational correspondence.

##### 6.2.1.2.1 Operational Correspondence

For the proof of operational correspondence we need the following statement:

**Lemma 6.2.3.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $(\!|P|\!)^{\lambda_o}_{\rho} \equiv (\!|Q|\!)^{\lambda_o}_{\rho}$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ∎

Operational correspondence for the translation of dynamic compensable processes into adaptable processes with objective update, is given in the following theorems:

**Theorem 6.2.4** (Operational Correspondence for $(\!|\cdot|\!)^{\lambda_o}_\rho$)**.** Let $P$ be a well-formed process in $\mathcal{C}^\lambda_{\mathtt{P}}$.

(1) If $P \longrightarrow P'$ then $(\!|P|\!)^{\lambda_o}_\varepsilon \longrightarrow^k (\!|P'|\!)^{\lambda_o}_\varepsilon$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 14 + \mathtt{pb}_{\mathtt{P}}(P_1) + \mathtt{ts}_{\mathtt{P}}(P_1) + \mathtt{Z}_{\mathtt{p}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{P}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 14 + \mathtt{pb}_{\mathtt{P}}(F[P_1]) + \mathtt{ts}_{\mathtt{P}}(F[P_1]) + \mathtt{Z}_{\mathtt{p}}(F[P_1])$,

d) $P \equiv C[s[H[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$ it follows $k = 6$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$, processes $P_1, Q, P_2, R$, and name $t, u, s$.

(2) If $(\!|P|\!)^{\lambda_o}_\varepsilon \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* (\!|P'|\!)^{\lambda_o}_\varepsilon$.

*Proof.* The proof uses the same ideas as Theorem 5.3.3. ∎

### 6.2.2 Comparing Subjective vs Objective update

In this subsection, we provide a theorem which states that subjective updates are better suited to encode compensation handling with dynamic compensation and preserving semantics than objective updates.

The following statement is a corollary of Theorem 6.2.4.

**Corollary 6.2.5.** Let $P$ be a well-formed process in $\mathcal{C}^\lambda$. If $P \longrightarrow P'$ and $(\!|P|\!)^{\lambda_o}_\varepsilon \longrightarrow^k (\!|P'|\!)^{\lambda_o}_\varepsilon$ then:

b) if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{P}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ then $k \geq 14 + \mathtt{pb}_{\mathtt{P}}(P_1) + \mathtt{ts}_{\mathtt{P}}(P_1) + \mathtt{Z}_{\mathtt{p}}(P_1)$,

c) if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{P}}(F[P_1]) \mid \langle Q\rangle]$ then $k \geq 14 + \mathtt{pb}_{\mathtt{P}}(F[P_1]) + \mathtt{ts}_{\mathtt{P}}(F[P_1]) + \mathtt{Z}_{\mathtt{p}}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

**Theorem 6.2.6.** The encoding $(\!|\cdot|\!)^{\lambda}_\rho : \mathcal{C}^\lambda_{\mathtt{P}} \longrightarrow \mathcal{S}$ is as or more efficient than $(\!|\cdot|\!)^{\lambda_o}_\rho : \mathcal{C}^\lambda_{\mathtt{P}} \longrightarrow \mathcal{O}$.

*Proof.* The proof proceeds in a similar manner to that employed in Theorem 4.2.6 by using: Theorem 5.3.3, Theorem 6.2.4, Proposition 2.2.3, and Corollary 6.2.5. ∎

## 6.3 Translating $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$

The translation $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$, denoted $\langle\!|\cdot|\!\rangle^\lambda_\rho$. This translation relies on the idea and principles of encoding $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{O}$ (cf. Section 4.3). We also require sets of *reserved names* as in Remark 5.2.1 and we use function for determining the number of locations as in Remark 5.2.2. We will use process $\mathtt{outd}^\circ$ as defined for $[\![\cdot]\!]^\circ_\rho$. We need some additional auxiliary processes.

$$
\begin{aligned}
\llparenthesis \langle P \rangle \rrparenthesis^{\lambda_o}_{\rho} &= p_\rho\big[\llparenthesis P \rrparenthesis^{\lambda_o}_{\varepsilon}\big] \\
\llparenthesis t[P,Q] \rrparenthesis^{\lambda_o}_{\rho} &= t\Big[\llparenthesis P \rrparenthesis^{\lambda_o}_{t,\rho}\Big] \mid r_t.\big(\mathtt{extra}\{t, p_{t,\rho}, p_\rho\} \mid p_\rho[\llparenthesis Q \rrparenthesis^{\lambda_o}_{\varepsilon}] \mid m_t.p_\rho\big[u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}]\big]\big) \\
&\qquad \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\} \\
&\qquad \mid v_t\big[u_t\{(Z).(Z \mid e_t[\llparenthesis Q \rrparenthesis^{\lambda}_{\varepsilon}] \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\}\big] \\
\llparenthesis \mathtt{inst}\lfloor \lambda Y.R \rfloor.P \rrparenthesis^{\lambda_o}_{t,\rho} &= u_t\Big[e_t\{(Y).(\overline{g_t}.v_t\{(X).(v_t[u_t\{(Z).(Z \mid e_t[\llparenthesis R \rrparenthesis^{\lambda_o}_{\varepsilon}] \\
&\qquad \mid f_t.e_t\{(X).X\}.v_t\{(X).X\}.g_t)\}))]\}\}.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid \llparenthesis P \rrparenthesis^{\lambda_o}_{t,\rho} \\
\llparenthesis \overline{t}.P \rrparenthesis^{\lambda_o}_{\rho} &= \overline{t}.h_t.\llparenthesis P \rrparenthesis^{\lambda_o}_{\rho}
\end{aligned}
$$

Figure 6.3: Translating $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{O}$.

**Definition 6.3.1** (Update Prefix for Extraction)**.** Let $t$, $l_1$, and $l_2$ be names. We write $\mathtt{extra}\{t, l_1, l_2\}$ to stand for the following (subjective) update prefix:

$$
\mathtt{extra}\{t, l_1, l_2\} = t\{(Y).\,\big(t[Y] \mid \mathtt{ch}(t, Y) \mid \mathtt{outd}^{\mathrm{o}}(l_1, l_2, \mathtt{nl}(l, Y), \overline{m_t}.\overline{j_t}.t\{\dagger\}.\overline{n_t})\big)\} \qquad (6.5)
$$

The intuition for the process $\mathtt{extra}\{t, l_1, l_2\}$ is the same as in the translation of $\mathcal{C}_{\mathtt{A}}$ into $\mathcal{S}$ with static recovery (cf. Section 3.4.0.1 and Section 3.4.1). Using well-formed compensable processes, the translation of $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{O}$ is as follows:

**Definition 6.3.2** (Translating $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{O}$)**.** Let $\rho$ be a path. We define the translation of compensable processes with dynamic recovery into (subjective) adaptable processes as a tuple $(\llparenthesis \cdot \rrparenthesis^{\lambda_o}_{\rho}, \varphi_{\llparenthesis \cdot \rrparenthesis^{\lambda_o}_{\rho}})$ where:

(a) The renaming policy

$$
\varphi_{\llparenthesis \cdot \rrparenthesis^{\lambda_o}_{\rho}}(x) = \begin{cases} \{x\} & \text{if } x \in \mathcal{N}_s \\ \{x, h_x, m_x, k_x, r_x, u_x, v_x, e_x, g_x, f_x, j_x, n_x, z_x\} \cup \{p_\rho : x \in \rho\} & \text{if } x \in \mathcal{N}_t. \end{cases}
$$

(b) The translation $\llparenthesis \cdot \rrparenthesis^{\lambda_o}_{\rho} : \mathcal{C}^{\lambda}_{\mathtt{A}} \to \mathcal{O}$ is as in Figure 6.3 and as a homomorphism for other operators.

### 6.3.1 Translation Correctness

We now establish that the translation $\llparenthesis \cdot \rrparenthesis^{\lambda_o}_{\rho}$ is a valid encoding. To this end, we address the three criteria in Definition 2.3.5: compositionality, name invariance, and operational correspondence. The other criteria are left as a research topic for future work.

#### 6.3.1.1 Structural Criteria

We prove the two criteria, compositionality and name invariance which are introduced in Definition 2.3.5.

#### 6.3.1.1.1 Compositionality

As we described in the previous encodings, to mediate between translations of subterms, we define a *context* for each process operator, which again depends on free names of the subterms:

**Definition 6.3.3** (Compositional context for $\mathcal{C}^\lambda_{\mathtt{A}}$)**.** For all process operator from $\mathcal{C}^\lambda_{\mathtt{D}}$, instead transaction we define a compositional context in $\mathcal{O}$ as in Definition 3.2.4. For transaction and compensation update compositional contexts are as follows:

$$\begin{aligned} C_{t[,],\rho}[\bullet_1,\bullet_2] &= t\Big[[\bullet_1]\Big] \mid r_t.\big(\mathtt{extra}\{t,p_{t,\rho},p_\rho\} \mid m_t.p_\rho\big[v_t\{(X).(X \mid u_t[\overline{f_t}.\overline{g_t}.j_t.n_t.\overline{k_t}])\}\big]\big) \\ &\qquad \mid t.t\{(Y).t[Y] \mid \mathcal{T}_t(Y).\overline{h_t}\} \mid v_t\big[u_t\{(Z).(Z \mid e_t[[\bullet_2]] \mid f_t.e_t\{(X).X\}.g_t)\}\big] \\ C_{\mathtt{inst},\rho}[\bullet_1,\bullet_2] &= u_t\Big[e_t\{(Y).(\overline{g_t}.u_t\{(Z).(Z \mid e_t[[\bullet_1]] \mid f_t.e_t\{(X).X\}.g_t)\})\}.(\overline{f_t}.e_t[\mathbf{0}])\Big] \mid [\bullet_2] \\ C_Y[\bullet_1] &= [\bullet_1] \end{aligned}$$

Using this definition, we may now state the following result:

**Theorem 6.3.1** (Compositionality for $\langle\!|\cdot|\!\rangle^{\lambda_o}_\rho$)**.** Let $\rho$ be an arbitrary path. For every process operator in $\mathcal{C}^\lambda_{\mathtt{A}}$ and for all well-formed compensable processes $P$ and $Q$ it holds that:

$$\begin{array}{lll} \langle\!|\langle P\rangle|\!\rangle^{\lambda_o}_\rho = C_{\langle\rangle,\rho}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\varepsilon\Big] & \langle\!|t[P,Q]|\!\rangle^{\lambda_o}_\rho = C_{t[,],\rho}\Big[\langle\!|P|\!\rangle^{\lambda_o}_{t,\rho},\langle\!|Q|\!\rangle^{\lambda_o}_\varepsilon\Big] & \langle\!|P \mid Q|\!\rangle^{\lambda_o}_\rho = C_{\mid}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho,\langle\!|Q|\!\rangle^{\lambda_o}_\rho\Big] \\ \langle\!|a.P|\!\rangle^{\lambda_o}_\rho = C_{a.}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho\Big] & \langle\!|\overline{t}.P|\!\rangle^{\lambda_o}_\rho = C_{\overline{t}.}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho\Big] & \langle\!|(\nu x)P|\!\rangle^{\lambda_o}_\rho = C_{(\nu x)}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho\Big] \\ \langle\!|\overline{a}.P|\!\rangle^{\lambda_o}_\rho = C_{\overline{a}.}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho\Big] & \langle\!|!\pi.P|\!\rangle^{\lambda_o}_\rho = C_{!\pi.}\Big[\langle\!|P|\!\rangle^{\lambda_o}_\rho\Big] & \\ \langle\!|Y|\!\rangle^{\lambda_o}_\rho = C_Y\Big[\langle\!|Y|\!\rangle^{\lambda_o}_\rho\Big] & \langle\!|\mathtt{inst}\lfloor\lambda Y.R\rfloor.P|\!\rangle^{\lambda_o}_{t,\rho} = C_{\mathtt{inst},\rho}[\langle\!|R|\!\rangle^{\lambda_o}_\varepsilon,\langle\!|P|\!\rangle^{\lambda_o}_{t,\rho}] & \end{array}$$

*Proof.* Follows directly from the definition of contexts (cf. Definition 5.4.3) and from the definition of $\langle\!|\cdot|\!\rangle^{\lambda_o}_\rho : \mathcal{C}^\lambda_{\mathtt{A}} \to \mathcal{O}$ (cf. Figure 6.3). ∎

#### 6.3.1.1.2 Name invariance

We now state name invariance, by relying on the renaming policy in Definition 6.3.2 (a).

**Theorem 6.3.2** (Name invariance for $\langle\!|\cdot|\!\rangle^{\lambda_o}_\rho$)**.** For every well-formed compensable process $P$ and valid substitution $\sigma : \mathcal{N}_c \to \mathcal{N}_c$ there is a $\sigma' : \mathcal{N}_a \to \mathcal{N}_a$ such that:

(i) for every $x \in \mathcal{N}_c$ : $\varphi_{\langle\!|\cdot|\!\rangle^{\lambda_o}_{\sigma(\rho)}}(\sigma(x)) = \{\sigma'(y) : y \in \varphi_{\langle\!|\cdot|\!\rangle^{\lambda_o}_\rho}(x)\}$, and

(ii) $\langle\!|\sigma(P)|\!\rangle^{\lambda_o}_{\sigma(\rho)} = \sigma'(\langle\!|P|\!\rangle^{\lambda_o}_\rho)$.

*Proof.* The proof proceeds in the same direction as the proof of Theorem 5.2.4. ∎

### 6.3.1.2 Semantic Criteria - Operational Correspondence

The analysis of operational correspondence follows the same ideas as in the translations $\mathcal{C}^\lambda_{\mathtt{A}}$ into $\mathcal{S}$ (cf. Subsection 5.4.1.2). Therefore, we use Definition 3.4.5, Remark 3.4.4 and Definition 3.4.6. Also, we need the following lemma:

**Lemma 6.3.3.** If $P$ and $Q$ are well-formed compensable processes such that $P \equiv Q$ then $\langle\!|P|\!\rangle^{\lambda_o}_\rho \equiv \langle\!|Q|\!\rangle^{\lambda_o}_\rho$.

*Proof.* The proof is by induction on the derivation $P \equiv Q$, and perform the case analysis on the last rule applied. In all cases the proof follows directly. ∎

We now state our operational correspondence result:

**Theorem 6.3.4** (Operational Correspondence for $\langle\!|\cdot|\!\rangle_\rho^{\lambda_o}$)**.** Let $P$ be a well-formed process in $\mathcal{C}_{\mathtt{A}}$.

(1) If $P \longrightarrow P'$ then $\langle\!|P|\!\rangle_\varepsilon^{\lambda_o} \longrightarrow^k \langle\!|P'|\!\rangle_\varepsilon^{\lambda_o}$ where for

a) $P \equiv E[C[\overline{a}.P_1] \mid D[a.P_2]]$ and $P' \equiv E[C[P_1] \mid D[P_2]]$ it follows $k = 1$,

b) $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ it follows $k = 15 + \mathtt{S}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1) + \mathtt{Z}_{\mathtt{a}}(P_1)$,

c) $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$ it follows $k = 15 + \mathtt{S}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{A}}(F[P_1]) + \mathtt{Z}_{\mathtt{a}}(F[P_1])$,

d) $P \equiv C[s[H[\mathtt{inst}\lfloor\lambda Y.R\rfloor.P_1],Q]]$ and $P' \equiv C[s[H[P_1],R\{^Q/_Y\}]]$ it follows $k = 6$,

for some contexts $C[\bullet], D[\bullet], E[\bullet], F[\bullet], H[\bullet]$ processes $P_1, Q, P_2, R$ and names $t, u, s$.

(2) If $\langle\!|P|\!\rangle_\varepsilon^{\lambda_o} \longrightarrow^n R$ with $n > 0$ then there is $P'$ such that $P \longrightarrow^* P'$ and $R \longrightarrow^* \langle\!|P'|\!\rangle_\varepsilon^{\lambda_o}$.

*Proof.* The proof follows idea presented in Theorem 3.4.6 and Theorem 5.4.4. ■

### 6.3.2 Comparing Subjective vs Objective update

In this subsection, we provide a theorem which states that subjective updates are better suited to encode compensation handling with dynamic compensation and aborting semantics than objective updates.
The following statement is a corollary of Theorem 6.2.4.

**Corollary 6.3.5.** Let $P$ be a well-formed process in $\mathcal{C}^\lambda$. If $P \longrightarrow P'$ and $\langle\!|P|\!\rangle_\varepsilon^{\lambda_o} \longrightarrow^k \langle\!|P'|\!\rangle_\varepsilon^{\lambda_o}$ then:

b) if $P \equiv E[C[t[P_1,Q]] \mid D[\overline{t}.P_2]]$ and $P' \equiv E[C[\mathsf{extr}_{\mathtt{A}}(P_1) \mid \langle Q\rangle] \mid D[P_2]]$ then $k \geq 15 + \mathtt{pb}_{\mathtt{A}}(P_1) + 4\,\mathtt{ts}_{\mathtt{A}}(P_1) + \mathtt{Z}_{\mathtt{a}}(P_1)$,

c) if $P \equiv C[u[F[\overline{u}.P_1],Q]]$ and $P' \equiv C[\mathsf{extr}_{\mathtt{A}}(F[P_1]) \mid \langle Q\rangle]$ then $k \geq 15 + \mathtt{pb}_{\mathtt{A}}(F[P_1]) + 4\,\mathtt{ts}_{\mathtt{A}}(F[P_1]) + \mathtt{Z}_{\mathtt{a}}(F[P_1])$,

for some contexts $C[\bullet]$, $D[\bullet]$, $E[\bullet]$, $F[\bullet]$, processes $P_1$, $Q$, $P_2$ and names $t$, $u$.

**Theorem 6.3.6.** The encoding $\langle\!|\cdot|\!\rangle_\rho^{\lambda} : \mathcal{C}_{\mathtt{A}}^\lambda \longrightarrow \mathcal{S}$ is as or more efficient than $\langle\!|\cdot|\!\rangle_\rho^{\lambda_o} : \mathcal{C}_{\mathtt{A}}^\lambda \longrightarrow \mathcal{O}$.

*Proof.* The proof proceeds similarly to the proof for Theorem 4.3.6 by using: Theorem 5.4.4, Theorem 6.3.4, Proposition 2.2.3, and Corollary 6.3.5. ■

**Brief summary of the chapter**:
In this chapter, we have two main results: (i) we presented the *encodings* of calculi for compensable processes into the calculi of adaptable processes with the objective update (encodings $\mathcal{C}_{\mathtt{D}}^\lambda$, $\mathcal{C}_{\mathtt{P}}^\lambda$, $\mathcal{C}_{\mathtt{A}}^\lambda$ into $\mathcal{O}$); (ii) we exploit the correctness properties of encodings to distinguish between subjective and objective updates in calculi for concurrency. We again analyzed the *efficiency* criterion of the encoding. We concluded that subjective updates induce tighter operational correspondences. Therefore, we can formally declare that subjective updates are more suited to encode compensation handling than objective updates.

In the next chapter, we conclude the dissertation by providing a review of the work presented in the thesis and some insight for future work.

# CHAPTER 7

# Conclusions and Perspectives

Finally, we conclude by providing a review of the work presented in the thesis and other research conducted by the candidate during her PhD studies. Also, this chapter suggests future research subjects that may be of interest to the reader.

## 7.1 Concluding Remarks

The quest for programming abstractions that suit emerging computational settings is a multi-faceted issue. Rather than creating new languages from the ground up, one option is to build on existing languages for mobile, autonomic, and service-oriented computing.

In this thesis, we have developed connections between programming abstractions for compensation handling (typical of models for services and long-running transactions) and run-time adaptation. Specifically, we compared from the point of view of relative expressiveness two related and yet fundamentally different process models: the calculus of compensable processes [29] and the calculus of adaptable processes [7].

We provide a unified, comprehensive presentation of twelve *processes translations* between the calculus of compensation handling (with static and dynamic compensations under discarding, preserving, and aborting semantics) into the calculus of adaptable processe swith *subjective* and *objective* mobility.

We have proved that encodings satisfied all or some well-established criteria [22]. Precisely, we prove that translations of $\mathcal{C}_{\mathtt{D}}$ into $\mathcal{S}$ and $\mathcal{O}$ are *valid encodings* — they satisfy *compositionality, name invariance, operational correspondence, divergence reflection* and *success sensitiveness* properties that bear witness to the robustness of translations. For translation of $\mathcal{C}^{\lambda}_{\mathtt{D}}$, $\mathcal{C}_{\mathtt{A}}$, $\mathcal{C}^{\lambda}_{\mathtt{A}}$ into $\mathcal{S}$ and $\mathcal{O}$ we prove that they satisfy: *compositionality, name invariance* and *operational correspondence*. We establish that translations of $\mathcal{C}_{\mathtt{P}}$ and $\mathcal{C}^{\lambda}_{\mathtt{P}}$ into $\mathcal{S}$ and $\mathcal{O}$ satisfy *name invariance* and *operational correspondence*, the analysis of the other criteria are left for future work. The encodings not only constitute a non-trivial application of two sensible forms of mobility for adaptable processes, but they also shed light on the intricate semantics of compensable processes.

We exploit our twelve translations to clearly distinguish between *subjective* and *objective updates* in calculi for concurrency. Therefore, we compared our encodings from the point of view of *efficiency*. Efficiency is a new comparison criterion, defied in abstract terms, considering the number of reduction steps that a target language requires to mimic the behavior of a source language. In this sense, subjective mobility allows us to encode compensable processes more efficiently than objective mobility. The efficiency gains induced by subjective mobility depend on the number of compensation actions in the source process. In the thesis, to formalize encodings, we developed the class of *well-formed* compensable processes, for which error notifications are crucial. Precisely, this class of processes disallows certain non-deterministic interactions that involve nested transactions and error notifications.

Interestingly, the work presented in the thesis uncovers an interesting dichotomy: should one appeal to objective or subjective updates? A subjective update would appear more "autonomous" than an objective update because it is determined by a located process itself, not by its environment. Still, we believe that the choice between objective and subjective updates is largely dependent on the application at hand: it is easy to imagine real-world scenarios of dynamic reconfiguration where each form of the update is more appropriate. Hence, a general specification language should probably include both objective and subjective updates.

During the candidate's PhD studies, in addition to the research work presented in the thesis, she has been also involved in research in the field of distributed computing particularly in edge computing as a service (micro clouds). Briefly, the results in [53] show how geodistributed edge nodes can be dynamically organized into micro data centers to cover any arbitrary area and increase capacity, availability, and reliability.

## 7.2 Future work

We intend further study the relationship between subjective and objective updates in future research work. An initial insight is the following: subjective updates can represent objective updates, at least in an ad-hoc manner. Consider process $S = C_1\big[l[P] \mid R_1\big] \mid C_2\big[l\{(X).Q\}.R_2\big]$, which, as we have seen, reduces to $C_1\big[Q\{^P\!/X\} \mid R_1\big] \mid C_2\big[R_2\big]$. Now consider $S'$, a process similar to $S$ but with subjective update prefixes:

$$S' = C_1\big[l[P] \mid l_1\langle\!\langle (X).X \rangle\!\rangle \mid R_1\big] \mid C_2\big[l\langle\!\langle (X).l_1[Q] \rangle\!\rangle.R_2\big]$$

In $S'$, we assume that name $l_1$ does not occur in $P$, $Q$, $R_1$, and $R_2$. Using two reductions, $S'$ emulates the movement induced by the reduction step originated in $S$:

$$\begin{aligned} S' &\longrightarrow C_1\big[\mathbf{0} \mid l_1\langle\!\langle (X).X \rangle\!\rangle \mid R_1\big] \mid C_2\big[l_1[Q\{^P\!/X\}].R_2\big] \\ &\longrightarrow C_1\big[Q\{^P\!/X\} \mid R_1\big] \mid C_2\big[\mathbf{0} \mid R_2\big] \end{aligned}$$

That is, the update prefix $l_1\langle\!\langle (X).X \rangle\!\rangle$ serves as an "anchor" to bring the reconfigured process $Q\{^P\!/X\}$ back to its original context $C_1[\bullet]$.

Similarly, we can represent subjective updates using objective prefixes. Consider process $L = C_1\big[l[P] \mid R_1\big] \mid C_2\big[l\langle\!\langle (X).Q \rangle\!\rangle.R_2\big]$, which reduces to $C_1\big[\mathbf{0} \mid R_1\big] \mid C_2\big[Q\{^P\!/X\} \mid R_2\big]$. Now consider process $L'$:

$$L' = C_1\big[l[P] \mid R_1\big] \mid C_2\big[l\{(X).l_1\{(Y).Q\}.\mathbf{0}\}.R_2 \mid l_1[\mathbf{0}]\big]$$

As in process $S'$, in $L'$ we assume that name $l_1$ is fresh; also, we assume that $P$ and $Q$ do not contain free occurrences of variable $Y$. Process $L'$ uses two reduction steps to mimic the reduction step originated in $L$:

$$\begin{aligned} L' &\longrightarrow C_1\big[l_1\{(Y).Q\{^P\!/X\}\}.\mathbf{0} \mid R_1\big] \mid C_2\big[R_2 \mid l_1[\mathbf{0}]\big] \\ &\longrightarrow C_1\big[\mathbf{0} \mid R_1\big] \mid C_2\big[R_2 \mid Q\{^P\!/X\}\big] \end{aligned}$$

Here, we use location $l_1[\mathbf{0}]$ to bring the reconfigured process $Q\{^P\!/X\}$ back to its original context $C_2[\bullet]$.

Crucially, these examples show that the ability of emulating a certain style of process mobility (subjective or objective) comes at the price of additional reduction steps, which could entail inefficient encodings. This observation reinforces our claim that a specification language should natively support both forms of update.

We addressed the encodability of compensable processes into adaptable processes. We also plan to consider the *reverse direction*, i.e., encodings of adaptable processes into compensable processes. We conjecture that there is no encoding of adaptable processes into a language with static compensations: compensation updates $\texttt{inst}\lfloor \lambda X.Q \rfloor.P$ seem essential to model an update prefix $l\{(X).Q\}.P$ — the semantics of both constructs induces process substitutions. Even if we consider a language with dynamic compensations, an encoding of adaptable processes is far from obvious. This claim is based on the fact that the semantics of compensation updates dynamically modifies the behavior of the compensation activity, the inactive part of a transaction. As a part of future work, it will be interesting to see how these (non) encodability claims can be formalized.

# Bibliography


[1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1 – 70, 1999.

[2] S. Alves, A. Arkin, S. Askary, C. Barreto, B. Bloch, F. Curbera, M. G. Ford, Y. Goland, A. Guzar, N. Kartha, C. K. Liu, R. Khalaf, D. König, M. Marin, V. Mehta, S. Thatte, D. Rijn, P. Yendluri, and A. Yiu. Web services business process execution language version 2.0 (oasis standard). 2007.

[3] J. Baeten. A brief history of process algebra. *Theoretical Computer Science*, 335(2):131 –146, 2005. Process Algebra.

[4] J. A. Bergstra and J. W. Klop. Fixed point semantics in process algebras. 1982.

[5] L. Bocchi, C. Laneve, and G. Zavattaro. A calculus for long-running transactions. In *Proc. of FMOODS 2003*, volume 2884 of *LNCS*, pages 124–138. Springer, 2003.

[6] D. Brand and P. Zafiropulo. On communicating finite-state machines. *J. ACM*, 30(2):323–342, Apr. 1983.

[7] M. Bravetti, C. D. Giusto, J. A. Pérez, and G. Zavattaro. Adaptable processes. *Logical Methods in Computer Science*, 8(4), 2012.

[8] M. Bravetti and G. Zavattaro. On the expressive power of process interruption and compensation. *Mathematical Structures in Computer Science*, 19(3):565–599, 2009.

[9] R. Bruni, C. Ferreira, and A. Kersten Kauer. First-order dynamic logic for compensable processes. In M. Sirjani, editor, *Coordination Models and Languages*, pages 104–121, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] M. J. Butler, C. Ferreira, and M. Y. Ng. Precise modelling of compensating business transactions and its application to BPEL. *J. Univers. Comput. Sci.*, 11(5):712–743, 2005.

[11] L. Caires, C. Ferreira, and H. T. Vieira. A process calculus analysis of compensations. In *Proc. of TGC 2008*, volume 5474 of *LNCS*, pages 87–103. Springer, 2009.

[12] L. Cardelli and A. D. Gordon. Mobile ambients. *Theor. Comput. Sci.*, 240(1):177–213, 2000.

[13] G. Castagna, J. Vitek, and F. Z. Nardelli. The seal calculus. *Inf. Comput.*, 201(1):1–54, 2005.

[14] J. Dedeić, J. Pantović, and J. A. Pérez. Efficient compensation handling via subjective updates. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 51–58, Marrakesh, Morocco, 2017. ACM.

[15] J. Dedeić, J. Pantović, and J. A. Pérez. On primitives for compensation handling as adaptable processes. *Journal of Logical and Algebraic Methods in Programming*, page 100675, 2021.

[16] J. Dedeić, J. Pantoviić, and J. A. Pérez. On compensation primitives as adaptable processes. In *EXPRESS/SOS 2015*, volume 190 of *EPTCS*, pages 16–30, 2015.

[17] C. Ferreira, I. Lanese, A. Ravara, H. T. Vieira, and G. Zavattaro. Advanced mechanisms for service combination and transactions. In *Results of SENSORIA*, volume 6582 of *LNCS*, pages 302–325. Springer, 2011.

[18] W. Fokkink. *Introduction to Process Algebra.* Springer Publishing Company, Incorporated, 1st edition, 2010.

[19] C. Fournet and G. Gonthier. The reflexive cham and the join-calculus. In *Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '96, pages 372–385, New York, NY, USA, 1996. Association for Computing Machinery.

[20] C. Fournet, C. Laneve, L. Maranget, and D. Rémy. Inheritance in the join calculus. *The Journal of Logic and Algebraic Programming*, 57(1):23 – 69, 2003.

[21] Y. Fu and H. Lu. On the expressiveness of interaction. *Theoretical Computer Science*, 411(11):1387 – 1451, 2010.

[22] D. Gorla. Towards a unified approach to encodability and separation results for process calculi. *Inf. Comput.*, 208(9):1031–1053, 2010.

[23] M. Hennessy. *A Distributed Pi-Calculus.* Cambridge University Press, USA, 2007.

[24] M. Hennessy and J. Riely. Resource access control in systems of mobile agents. *Information and Computation*, 173(1):82 –120, 2002.

[25] T. Hildebrandt, J. C. Godskesen, and M. Bundgaard. Bisimulation congruences for homer–a calculus of higher order mobile embedded resources. *Technical Report TR-2004–52*, 2004.

[26] C. A. R. Hoare. Communicating sequential processes. *Commun. ACM*, 21(8):666–677, Aug. 1978.

[27] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[28] K. Honda, V. T. Vasconcelos, and M. Kubo. Language primitives and type discipline for structured communication-based programming. In C. Hankin, editor, *Programming Languages and Systems*, pages 122–138, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[29] I. Lanese, C. Vaz, and C. Ferreira. On the expressive power of primitives for compensation handling. In *Proc. of ESOP 2010*, volume 6012 of *LNCS*, pages 366–386. Springer, 2010.

[30] I. Lanese and G. Zavattaro. Decidability results for dynamic installation of compensation handlers. In *COORDINATION 2013*, volume 7890 of *LNCS*, pages 136–150. Springer, 2013.

[31] C. Laneve and G. Zavattaro. Foundations of web transactions. In *Proc. of FOSSACS 2005*, volume 3441 of *LNCS*, pages 282–298. Springer, 2005.

[32] R. Milner. *Communication and concurrency.* PHI Series in computer science. Prentice Hall, 1989.

[33] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I. *Inf. Comput.*, 100(1):1–40, 1992.

[34] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, ii. *Information and Computation*, 100(1):41 – 77, 1992.

[35] U. Nestmann. *On Determinacy and Nondeterminacy in Concurrent Programming*. Doctoral thesis, Universität Erlangen-Nürnberg, 1996.

[36] U. Nestmann. What is a "good" encoding of guarded choice? *Information and Computation*, 156(1-2):287–319, 2000.

[37] U. Nestmann. Welcome to the jungle: A subjective guide to mobile process calculi. In *Proceedings of the 17th International Conference on Concurrency Theory*, CONCUR'06, pages 52–63, Berlin, Heidelberg, 2006. Springer-Verlag.

[38] R. D. Nicola. A gentle introduction to process algebras. IMT – Institute for Advanced Studies Lucca, 2013.

[39] C. Palamidessi. Comparing the expressive power of the synchronous and asynchronous *pi*-calculi. *Mathematical Structures in Computer Science*, 13(5):685–719, 2003.

[40] J. Parrow. Expressiveness of process algebras. *Electronic Notes in Theoretical Computer Science*, 209:173 – 186, 2008. Proceedings of the LIX Colloquium on Emerging Trends in Concurrency Theory (LIX 2006).

[41] J. Parrow and B. Victor. The fusion calculus: expressiveness and symmetry in mobile processes. In *Proceedings. Thirteenth Annual IEEE Symposium on Logic in Computer Science (Cat. No.98CB36226)*, pages 176–185, 1998.

[42] J. A. Pérez. *Higher-Order Concurrency: Expressiveness and Decidability Results*. Doctoral thesis, University of Bologna, Department of Computer Science, 2010.

[43] K. Peters. *Translational expressiveness: comparing process calculi using encodings*. Doctoral thesis, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, Berlin, 2012.

[44] K. Peters. Comparing process calculi using encodings. In J. A. Pérez and J. Rot, editors, *Proceedings Combined 26th International Workshop on Expressiveness in Concurrency and 16th Workshop on Structural Operational Semantics, EXPRESS/SOS 2019, Amsterdam, The Netherlands, 26th August 2019*, volume 300 of *EPTCS*, pages 19–38, 2019.

[45] K. Peters and R. J. van Glabbeek. Analysing and comparing encodability criteria. In S. Crafa and D. Gebler, editors, *Proceedings of the Combined 22nd International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, EXPRESS/SOS 2015, Madrid, Spain, 31st August 2015*, volume 190 of *EPTCS*, pages 46–60, 2015.

[46] C. A. Petri. *Kommunikation mit Automaten*. Doctoral thesis, Universität Hamburg, 1962.

[47] C. Priami. Stochastic pi-calculus. *The Computer Journal*, 38(7):578–589, 1995.

[48] I. Prokić. *Formal modeling and analysis of resource usage and sharing in distributed software systems*. Doctoral thesis, Faculty of Technical Sciences, University of Novi Sad, 2019.

[49] I. Prokić and H. T. Vieira. The $c_\pi$-calculus: A model for confidential name passing. *Journal of Logical and Algebraic Methods in Programming*, 119:100622, 2021.

[50] W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies.* Springer Publishing Company, Incorporated, 2013.

[51] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher Order Paradigms.* PhD thesis, University of Edinburgh, 1992.

[52] A. Schmitt and J. Stefani. The kell calculus: A family of higher-order distributed process calculi. In C. Priami and P. Quaglia, editors, *Global Computing, IST/FET International Workshop, GC 2004, Rovereto, Italy, March 9-12, 2004, Revised Selected Papers*, volume 3267 of *Lecture Notes in Computer Science*, pages 146–178. Springer, 2004.

[53] M. Simić, I. Prokić, J. Dedeić, G. Sladić, and B. Milosavljević. Towards edge computing as a service: Dynamic formation of the micro data-centers. *IEEE Access*, 9:114468–114484, 2021.

[54] C. Vaz and C. Ferreira. On the analysis of compensation correctness. *J. Log. Algebr. Program.*, 81(5):585–605, 2012.

[55] M. G. Vigliotti, I. Phillips, and C. Palamidessi. Tutorial on separation results in process calculi via leader election problems. *Theoretical Computer Science*, 388(1):267 – 289, 2007.

[56] N. Yoshida. Minimality and separation results on asynchronous mobile processes - representability theorems by concurrent combinators. *Theoretical Computer Science*, 274(1):231 –276, 2002. 9th International Conference on Concurrency Theory 1998.

*Овај Образац чини саставни део докторске дисертације, односно докторског уметничког пројекта који се брани на Универзитету у Новом Саду. Попуњен Образац укоричити иза текста докторске дисертације, односно докторског уметничког пројекта.*

## План третмана података

**Назив пројекта/истраживања**

Релативна експресивност процесних рачуна који поседују могућност адаптације и динамичког ажурирања током извршавања/ Relative Expressiveness of Process Calculi with Dynamic Update and Runtime Adaptation

**Назив институције/институција у оквиру којих се спроводи истраживање**

а) Факултет техничких наука, Универзитет у Новом Саду

б) Бернули институт за математику, рачунарске науке и вештачку интелигенцију, Универзитет у Гронингену, Холандија

**Назив програма у оквиру ког се реализује истраживање**

Математика у техници – докторска дисертација

**1. Опис података**

*1.1* Врста студије

*Укратко описати тип студије у оквиру које се подаци прикупљају*

**Докторска дисертација**

1.2 Врсте података

а) квантитативни

б) **квалитативни**

1.3. Начин прикупљања података

а) анкете, упитници, тестови

б) клиничке процене, медицински записи, електронски здравствени записи

в) генотипови: навести врсту ___________________________________

г) административни подаци: навести врсту _________________________

д) узорци ткива: навести врсту____________________________________

ђ) снимци, фотографије: навести врсту_______________________________

е) текст, навести врсту _**Актуелна литература у области истраживања**_______

ж) мапа, навести врсту ___________________________________________ з) остало: описати _______________________________________________

1.3 Формат података, употребљене скале, количина података

1.3.1 Употребљени софтвер и формат датотеке:

a) Excel фајл, датотека ____________________

b) SPSS фајл, датотека ____________________

c) PDF фајл, датотека _____________________

d) Текст фајл, датотека ____________________

e) JPG фајл, датотека _____________________

f) Остало, датотека ______________________

1.3.2. Број записа (код квантитативних података)

а) број варијабли ________________________________________________________

б) број мерења (испитаника, процена, снимака и сл.) ________________________

1.3.3. Поновљена мерења

а) да

б) не

Уколико је одговор да, одговорити на следећа питања:

а) временски размак између поновљених мера је _________________________________

б) варијабле које се више пута мере односе се на ___________________________________

в) нове верзије фајлова који садрже поновљена мерења су именоване као _____________

Напомене: ____________________________________________________________________

*Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?*

*а) Да*

*б) Не*

*Ако је одговор не, образложити* ___________________________________________________

________________________________________________________________________________

## 2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

а) експеримент, навести тип ______________________________________________________

б) корелационо истраживање, навести тип ____________________________________________ ц) анализа текста, навести тип ___________________________________________________

д) остало, навести шта ______________________________________________________

*2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).*

__________________________________________________________________________

__________________________________________________________________________

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

а) Да ли матрица садржи недостајуће податке? Да Не

Ако је одговор да, одговорити на следећа питања:

а) Колики је број недостајућих података? ________________________

б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не

в) Ако је одговор да, навести сугестије за третман замене недостајућих података

__________________________________________________________________________

2.2.2. На који начин је контролисан квалитет података? Описати

__________________________________________________________________________

__________________________________________________________________________

2.2.3. На који начин је извршена контрола уноса података у матрицу?

__________________________________________________________________________

__________________________________________________________________________

## 3. Третман података и пратећа документација

3.1. Третман и чување података

*3.1.1. Подаци ће бити депоновани у* ________________________________ *репозиторијум.*

*3.1.2. URL адреса* __________________________________________________________

*3.1.3. DOI* _________________________________________________________________

*3.1.4. Да ли ће подаци бити у отвореном приступу?*

*а) Да*

*б) Да, али после ембарга који ће трајати до* ________________________________

*в) Не*

*Ако је одговор не, навести разлог* ______________________________________

*3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.*

*Образложење*

__________________________________________________________________________

___________________________________________________________________________

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен? _________________________________

3.2.1. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

___________________________________________________________________________

_____________________________________________________

*Ако је потребно, навести методе које се користе за преузимање података, аналитичке и процедуралне информације, њихово кодирање, детаљне описе варијабли, записа итд.*

___________________________________________________________________________

___________________________________________________________________________

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму? ______________________

3.3.2. Да ли ће подаци бити депоновани под шифром? Да Не

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да Не

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена?

Да Не

Образложити

___________________________________________________________________________

___________________________________________________________________________

## 4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размотрити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности (*https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html*) и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да **Не**

Ако је одговор Да, навести датум и назив етичке комисије која је одобрила истраживање

___________________________________________________________________________

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? Да **Не**

Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

а) Подаци нису у отвореном приступу

б) Подаци су анонимизирани

ц) Остало, навести шта

______________________________________________________________________________

______________________________________________________________________________

**5. Доступност података**

*5.1. Подаци ће бити*

*а)* ***јавно доступни***

*б) доступни само уском кругу истраживача у одређеној научној области*

*ц) затворени*

*Ако су подаци доступни само уском кругу истраживача, навести под којим условима могу да их користе:*

_______________________________________________________________________

______________________________________________________________________________

*Ако су подаци доступни само уском кругу истраживача, навести на који начин могу приступити подацима:*

______________________________________________________________________________

______________________________________________________________________________

*5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.*

______________________________________________________________________________

**6. Улоге и одговорност**

*6.1. Навести име и презиме и мејл адресу власника (аутора) података*

__________ Јована Дедеић, radenovicj@uns.ac.rs

*6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима*

______________________________________________________________________________

*6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима*

__________________________________________________________________________

## PERSONAL INFORMATION

# Jovana Dedeić


Miše Dimitrijevića 66A, Novi Sad, 21000, Serbia

radenovicj@uns.ac.rs

http://imft.ftn.uns.ac.rs/math/People/JovanaRadenovi%C4%87

Female | Date, place, and country of birth:: 23/10/1987, Novi Sad, Serbia


## ACADEMIC QUALIFICATION

| | |
|---|---|
| October 2012 - December 2021 | Ph.D. degree in Applied Mathematics from the Faculty of Technical Sciences, University of Novi Sad, Serbia<br>**Advisors:**<br>- Prof. Jovanka Pantović (http://imft.ftn.uns.ac.rs/~vanja)<br>- Prof. Jorge Andres Pérez Parra (https://www.jperez.nl/)<br>**Area of research:** concurrency, semantics of programming languages, process calculi, compensation handling, dynamic update, expressiveness |
| October 2010 - November 2011 | Master in mathematics<br>Applied mathematics study programme<br>Faculty of Sciences, University of Novi Sad, Department of Mathematics and Informatics<br>GPA: 9.63/10 |
| October 2006 - September 2010 | Bachelor with honours in mathematics<br>Mathematics study programme<br>Faculty of Sciences, University of Novi Sad, Department of Mathematics and Informatics<br>GPA: 9.58/10 |

## PROFESIONAL EXPERIENCE

| | |
|---|---|
| Since 2013 | Faculty of Technical Sciences, University of Novi Sad,<br>Teaching assistant within the Department of Fundamentals Sciences |
| 2012 - 2013 | Faculty of Technical Sciences, University of Novi Sad,<br>Teaching Associate within the Department of Fundamentals Sciences |
| 2011 - 2012 | City Administration for Economy, Novi Sad, Finance Department, Graduate trainee |

## PUBLICATIONS

| | |
|---|---|
| | **JOURNAL PAPER** |
| 2021 | J. Dedeić, J. Pantović, J.A. Pérez: *On primitives for compensation handling as adaptable processes.* Journal of Logical and Algebraic Methods in Programming, vol. 121, pp. 100675, 2021, ISSN 2352-2208, doi: https://doi.org/10.1016/j.jlamp.2021.100675. |
| 2021 | M. Simić, I. Prokić, J. Dedeić, G. Sladić and B. Milosavljević: *Towards Edge Computing as a Service: Dynamic Formation of the Micro Data-Centers*, in IEEE Access, vol. 9, pp. 114468-114484, 2021, doi: 10.1109/ACCESS.2021.3104475. |
| | **CONFERENCE PAPER** |
| 2021 | J. Dedeić, J. Pantović, J.A. Pérez: *On primitives for compensation handling as adaptable processes* (Oral Communication), ICE 2021 - 14th Interaction and Concurrency Experience. |
| 2021 | B. Čelić and J. Dedeić: *Synchronous and asynchronous learning in online education*, XXVII Skup Trendovi razvoja: "On-line nastava na univerzitetima", Srbija, 2021 |
| 2020 | R. Božić, J. Dedeić, S. Milićević i I. Kovačević: *Primena Geogebre u nastavi matematike*, XXVI Skup Trendovi razvoja: "Inovacije u modernom obrazovanju" , Kopaonik, Srbija, 2020. |
| 2016 | J. Dedeić, J. Pantović, J.A. Pérez: *Efficient Compensation Handling via Subjective Updates,*<br>The 32nd ACM Symposium on Applied Computing - SAC'17 (CAS track), ACM Press<br>April 3-6, 2017, Marrakesh, Morocco |
| 2015 | J. Dedeić, J. Pantović, J.A. Pérez: *On Compensation Primitives as Adaptable Processes,*<br>EPTCS 190, 2015, pp. 16-30. Combined 22nd International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, and 12th Workshop on Structural Operational Semantics, EXPRESS/SOS 2015, Madrid, Spain, 31st August 2015. (DOI: 10.4204/EPTCS.190.2) |

## ADDITIONAL INFORMATION

| | |
|---|---|
| | **CONFERENCES** |
| June 18, 2021 | ICE 2021 - 14th Interaction and Concurrency Experience, virtual event |
| October 3 - 5, 2019 | Congress of Young Mathematicians, Novi Sad, Serbia |
| September 21 - 25, 2015 | Logic and Applications 2015 (LAP 2015), Dubrovnik, Croatia. |
| September 1 - 4, 2015 | 26th International Conference on Concurrency Theory (CONCUR 2015), Madrid, Spain. |
| August 31, 2015 | Combined 22nd International Workshop on Expressiveness in Concurrency and 12th Workshop on Structural Operational Semantics, and 12th Workshop on Structural Operational Semantics (EXPRESS/SOS 2015), Madrid, Spain |
| February 10 - 14, 2014 | Mathematical Structures of Computation, Lyon, France. |
| September 26 - 27, 2013 | Probabilistic logics and applications, Belgrade, Serbia. |

| | |
|---|---|
| March 18 - 22, 2013 | 84th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM 2013), Novi Sad, Serbia. |
| | **SUMMER SCHOOLS** |
| June 27 - July 1, 2016 | Second International Summer School on Behavioural Types, Limassol, Cyprus |
| July 21 - 25, 2014 | Summer School on the Interactions between Modern Foundations of Mathematics and Contemporary Philosophy, Benedictine nunnery on Fraueninsel (as island in Chiemsee), Germany. |
| June 30 - July 4, 2014 | First International Summer School on Behavioural Types, Lovran, Croatia |
| July 13 - 25, 2013 | European Summer School for Visual Mathematics and Education, Eger, Hungary. |
| | **PROJECTS** |
| 2011 - 2021 | Representation of logical structure and formal language and their application in computing,<br>Ministry of Education and Science, Project IO 174026, Serbian National Project. |
| 2012 - 2016 | EU Cost Action IC1201: Behavioural Types for Reliable Large-Scale Software Systems |
| 2012 - 2014 | Tempus IV Project: Visuality & Mathematics: Experiential Education of Mathematics through Visual Arts,<br>Sciences and Playful Activities (http://vismath.ektf.hu/). |
| | **STAYS ABROAD** |
| April 2018 | Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen<br>Prof. Jorge A. Pérez; 1 week (Erasmus+ International Credit Mobility (ICM)) |
| January - February 2016<br>September 2017 | Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen,<br>The Netherlands, 2016 - 4 weeks, and 2017 - 1 week<br>Research topic: Typed approaches to compensation handling via sessions and adaptable processes<br>Supervisor: Prof. Jorge A. Pérez. |
| January 2015 | Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen<br>The Netherlands, 3 weeks<br>Research topic: Expressive power of sessions and adaptable processes<br>Supervisor: Prof. Jorge A. Pérez |
| April - May 2014 | Institute of Art, Science and Education; University of applied arts Vienna, Austria, 4 weeks<br>Research topic: Applied Design Thinking: Visualizing Mathematical topics through Art and Design<br>Supervisor: Dr. Ruth Mateus-Berr |
| | **SCHOLARSHIPS** |
| April 2018 | Erasmus+ International Credit Mobility (ICM) |
| Jun 2016 | Travel/accommodation grant for participation in Second International Summer School on Behavioural Types, Limassol, Cyprus. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |
| March 2016 | Travel/accommodation grant for attending BETTY Working Group meeting, Valletta, Malta. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |
| January 2016 | Grant for Short Term Scientific Mission (STSM), Groningen, The Netherlands. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |
| September 2015 | Travel/accommodation grant for attending BETTY Working Group meeting, Madrid, Spain. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |
| July 2015 | Travel/accommodation grant for participation in First International Summer School on Behavioural Types, Lovran, Croatia. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |
| January 2015 | Grant for Short Term Scientific Mission (STSM), Groningen, The Netherlands. COST Action IC1201 - Behavioural Types for Reliable Large-Scale Software Systems (BETTY). |