

UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

Miljan M. Vučetić

**PRIMENA FAZI LOGIKE U ANALIZI
FUNKCIONALNIH ZAVISNOSTI U
RELACIONIM BAZAMA PODATAKA**

- Doktorska disertacija -

Beograd, 2013

UNIVERSITY OF BELGRADE
FACULTY OF ORGANIZATIONAL SCIENCES

Miljan M. Vučetić

**ANALYSIS OF FUNCTIONAL
DEPENDENCIES IN RELATIONAL
DATABASES USING FUZZY LOGIC**

- Doctoral Dissertation -

Belgrade, 2013

Ova doktorska disertacija rađena je na Fakultetu organizacionih nauka, smer elektronsko poslovanje. Prilikom definsanja teme, sadržaja, naslova i rezultata istraživanja doktorske disertacije konsultovan je **prof. dr Mirko Vujošević**.

Komisija za pregled, ocenu i odbranu doktorske disertacije:

Mentor:

1. **dr Mirko Vujošević**, red. prof.,
Fakultet organizacionih nauka, Beograd

Članovi komisije:

2. **dr Nenad Aničić**, vanr.prof.,
Fakultet organizacionih nauka, Beograd
3. **dr Zoran Avramović**, vanr. prof.
Saobraćajni fakultet, Beograd
4. **dr Slađan Babarogić**, docent,
Fakultet organizacionih nauka, Beograd
5. **dr Dragan Radojević**,
Naučni savetnik Institut Mihajlo Pupin

Datum odbrane: _____

Datum promocije: _____

Primena fazi logike u analizi funkcionalnih zavisnosti u relacionim bazama podataka

Rezime:

Predmet istraživanja u disertaciji su fazi funkcionalne zavisnosti u relacionim bazama podataka, a cilj istraživanja je proučavanja dosadašnjih rezultata u ovom polju, te razvoj sopstvenog algoritma za identifikovanje zavisnosti između atributa relacije koji je zasnovan na primeni fazi implikacija iz fazi logike. Problem analize fazi funkcionalnih zavisnosti nije nov. U ovoj oblasti postoji značajan broj radova u poslednje dve decenije i veliki broj autora se bavio ovom problematikom. Međutim, analizom dostupne literature utvrđeno je da ne postoji algoritam koji bi omogućio identifikovanje zavisnosti između atributa relacione šeme. Ovo ograničenje bilo je motiv za razvoj sopstvenog pristupa u analizu funkcionalnih zavisnosti nad datim skupom atributa. Rešavanje ovog, ne tako novog problema, ne predstavlja samo teorijski značaj i istraživački izazov, već ima i praktični značaj. U tom smislu, cilj istraživanja je implementacija logičkog modela baze podataka koju daje prethodna teorija i koja podržava očekivane rezultate istraživanja, ali i veza sa drugim oblastima i poljima (geografski informacioni sistemi, rudarenje podataka, ekstrakcija informacija, smanjenje redundancije, estimacija nepoznatih vrednosti atributa...).

Polazna pretpostavka je bila da teorija fazi skupova i određeni fragment fazi logike predstavljaju pogodno rešenje kada projektanti informacionih sistema ne mogu jednoznačno da odrede zavisnost između podataka i atributa relacije. U oblasti veštačke inteligencije, fazi skupovi i fazi logika izučavaju se sa aspekta dizajniranja fazi relacionih baza podataka, u okviru inteligentnih sistema, jer teorija fazi skupova i fazi logika predstavljaju moćan alat za manipulisanje nepreciznim i neodređenim informacijama.

Otkrivanje potencijalnih zavisnosti između atributa posmatrane fazi relacije ustvari predstavlja pronalaženje skrivenih i korisnih znanja koja možemo da koristimo za donošenje poslovnih i upravljačkih odluka. U radu je predstavljena nova tehnika koja identifikuje postojanje zavisnosti između atributa na nivou podrelacije (parova n -torki) primenom različitih fazi implikacija, čija karakterizacija je opisana u ovoj disertaciji u kontekstu primene u ovoj problematici. Fazi implikacije koje zadovoljavaju ustanovljene kriterijume prihvatanja koristimo za procenu jačine zavisnosti θ u slučaju postojanja fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$.

Za potrebe demonstracije predložene metodologije i verifikaciju rezultata istraživanja izvršen je eksperiment nad realnim podacima u fazi modelu baze podataka na bazi relacije bliskosti i fazi modelu na bazi mogućnosti distribucije.

Da bismo identifikovali fazi funkcionalne zavisnosti u odgovarajućem fazi modelu potrebno je:

- 1. Pripremiti podatke za analizu,*
- 2. Pronaći podrelacije (parove n -torki) na posmatranim atributima u kojima se potencijalno kriju fazi funkcionalne zavisnosti,*
- 3. Pridružiti fazi funkcionalnim zavisnostima odgovarajuće fazi implikacije,*
- 4. Interpretirati rezultate istraživanja da bismo dobili konačan skup fazi zavisnosti.*

Tragajući za fazi funkcionalnim zavisnostima u fazi relacionim bazama podataka primenom predložene metodologije, neophodno je sa praktične strane gledišta uraditi i analizu efikasnosti. Osnovni nedostatak koji se pojavljuje je ukupan broj transakcija (mogućih parova n -torki sa povezanim vrednostima) koje je neophodno izračunati, a što može biti jako skupo za velike baze podataka. Povećanjem broja n -torki, kao i broja atributa relacione šeme, broj transakcija raste eksponencijalno. U radu predlažemo i pokazujemo da je moguće povećati efikasnost algoritma i smanjiti broj prolaza kroz petlju. Takođe, želimo naglasiti da priroda posmatranog problema određuje efikasnost algoritma. Postoje baze podataka u kojima kompresija nije potrebna. Sa druge strane, kada govorimo o bazama sa velikom količinom podataka i zapisa koristimo različite tehnike kompresije i redukcije zapisa u bazi kao što su fazifikacija, klasterovanje, granulacija, predstavljanje većeg broja sličnih n -torki reprezentom itd.

Generalno, naše istraživanje pokazuje kako da dobijemo inteligentne informacije iz fazi relacionih baza podataka. Da bismo došli do mogućih relacija između atributa fazi relacije najpre je neophodno pripremiti podatke za analizu odnosno krierati tabelu sa parovima n -torki i vrednostima usaglašenosti na svim atributima originalne relacije. U drugom koraku pronalazimo podrelacije u kojima su zadovoljeni kriterijumi za postojanje fazi funkcionalnih zavisnosti. U trećem koraku fazi funkcionalnim zavisnostima pridružujemo odgovarajuće fazi implikacije. Konačno, mi interpetiramo rezultate iz prethodnih koraka kako bismo dobili finalni skup fazi funkcionalnih zavisnosti.

Najvažniji doprinos ove doktorske disertacije predstavlja karakterizacija teorije fazi skupova i fazi implikacija u kontekstu izgradnje alata i algoritma koji omogućava identifikaciju odnosa između atributa u posmatranim relacionim šemama. Treba napomenuti da ovaj rad predstavlja empirijsku studiju i "prof-of-concept" i da nije razvijen komercijalni softver. Nadalje, mi smo diskutovali neke praktične slučajeve u kojima ova metodologija može biti veoma korisna. Eksperimentalni rezultati verifikuju rezultate istraživanja.

Buduće istraživanje podrazumeva primenu predstavljene metodologije u različitim scenarijima, kao što su: procena nedostajućih vrednosti, text mining, ekstrakcija zavisnosti iz praktičnih baza znanja itd.

Ključne reči: *fazi relacija, fazi funkcionalna zavisnost, fazi implikacija, rudarenje podataka.*

Naučna oblast: tehničke nauke

Uža naučna oblast: informacione tehnologije i operaciona istraživanja

UDK: 519.8:004.652.4.

Analysis of Functional Dependencies in Relational Databases Using Fuzzy Logic

Abstract:

The research presented in this dissertation is finding of fuzzy functional dependencies in the fuzzy relational database models. The goal is reviewing and analyzing of recent results in this field and developing our own algorithm for identification of dependencies between attributes in the fuzzy relations based on fuzzy implication in fuzzy logic. The problem of fuzzy functional dependency analyzing in fuzzy relational database models is not recent. In this area there is a significant number of papers and authors dealt with mentioned issue in the last two decades. However, literature analysis showed that there is no algorithm that would enable the identification of attribute relationships in relational schemas. This restriction was the motive for the development of our methodology in the analysis of fuzzy functional dependencies over a given set of attributes. Solving this, not so new problem, is not only research challenge having theoretical importance, but it also has practical significance. Implementation of logical database models provided by the previous theory and integration to other areas and fields (GIS, Rudarenje podataka, Information Retrieval, reducing data redundansy, estimation of NULL values...) are crucial aims in this paper.

The initial assumption was that the theory of fuzzy sets and particular fragment of fuzzy logic are the perfect solution when database managers can't unambiguously determine the relationships between data and attributes within the relation. In the field of artificial intelligence, fuzzy sets and fuzzy logic are studied in terms of fuzzy relational database design within the intelligent systems because fuzzy set theory and fuzzy logic are powerful tools for manipulating and representing imprecise and uncertain information.

Finding potential dependencies between attributes in fuzzy relations is actually rudarenje podataka technique related to hidden and useful knowledge used for decision making. This dissertation presents a new technique for identification of existing dependencies between pairs of tuples using different fuzzy implications which characterization is described by author. Fuzzy implications meeting established acceptance criteria are used for estimation of linguistic strength θ when fuzzy functional dependency $X \xrightarrow{\theta} Y$ is satisfied.

For demonstration of proposed methodology and research verification we performed experiment on real data under two different models: fuzzy database model based on proximity relation and fuzzy database model with possibility distribution.

Discovering potential dependencies between attributes of the observed fuzzy relation actually represents a discovery of hidden and useful knowledge, i.e. a rudarenje podataka technique. In order to identify fuzzy functional dependency within the corresponding fuzzy model, following is necessary:

- 1. Preparing the data for the analysis (preprocessing, data summarization, data warehousing etc.).*
- 2. Finding subrelations (n-tuples) on the observed attributes with potentially hidden fuzzy functional dependencies.*
- 3. Associating fuzzy functional dependencies with corresponding fuzzy implications.*

4. *Interpreting search results for the purpose of getting the final set of fuzzy dependencies.*

From a practical point of view, in search of fuzzy functional dependencies within fuzzy database models through application of the proposed methodology, it is necessary to conduct efficiency analysis as well. Main shortcoming that may appear is the total number of transactions (possible pairs of tuples with associated values) that needs to be computed, which can be very expensive for large databases. Increase of the number of tuples, as well as the number of attributes of relational scheme, exponentially increases the number of tuples. However, in this PhD thesis we propose and show that increase of algorithm's efficiency is possible, as well as decrease of the number of computed transactions. However, we would like to note that the nature of the observed problem determines efficiency of the algorithm. There are databases where compression is not necessary, while in other cases, when we are talking about huge databases, a variety of techniques are proposed for the relation reduction such as data summarization based on fuzzification [], cluster formation, granulation, presenting a number of similar tuples with one representing tuple etc.

Generally, our research shows us how to extract intelligent information from fuzzy relational databases. In order to identify possible relations between attributes of fuzzy relation, we must firstly prepare the data for the analysis i.e. create a table with pairs of tuples and conformance values on all attributes of original relation. Next step is to find sub-relations where criteria for the existence of fuzzy dependencies were met. Third step is to associate fuzzy functional dependencies with corresponding fuzzy implications. Finally, we interpret the results of previous steps in order to get the final set of fuzzy functional dependencies.

The most important contribution of this dissertation is characterization of the theory of fuzzy sets and fuzzy implications in the context of developing tools and algorithms that facilitate identification of relations between attributes in the observed relational schemes. It should be pointed out that this paper represents an empirical study and a “proof-of-concept”, without developed commercial software. Moreover, we discussed certain practical cases where this methodology can be suitable. Our experiments suggest that results provided by algorithm are acceptable.

As a future work we plan to employ this research in different scenarios and applications involving this methodology such as: estimating null values in relational databases, text mining, extraction dependencies from practical knowledge databases etc.

Keywords: *fuzzy relation, fuzzy functional dependency, fuzzy implication, rudarenje podataka.*

Scientific field: technical science

Specialized scientific field: information technology and operational research

UDC: 519.8:004.652.4.

Sadržaj

1. Uvod	9
2. Fazi skupovi i fazi logika	12
2.1. Uvod u fazi skupove	12
2.2. Fundamentalni koncept fazi skupova.....	18
2.3. Princip rezolucije i reprezentacije.....	19
2.4. Operacije nad fazi skupovima.....	21
2.5. t-norma i t-konorma	23
2.6. Fazi relacije.....	25
2.7. Teorija aproksimativnog rezonovanja.....	28
2.8. Semantika sistema aproksimativnog rezonovanja	30
2.9. Dedukcija u sistemu aproksimativnog rezonovanja.....	30
2.10. Kompoziciono pravilo.....	31
2.11. Fazi implikacije.....	32
2.12. Fazi metod zaključivanja.....	34
2.13. Zadovoljivost u fazi logici	35
2.14. Rezolventa u fazi logici.....	35
2.15. Zaključak.....	38
3. Fazi relacione baze podataka.....	39
3.1. Relacioni model baza podataka.....	39
3.1.1. Modeli, atributi, entiteti	39
3.1.2. Predstavljanje skupa entiteta skupom n-torki	39
3.1.3. Relaciona algebra	40
3.1.4. Funkcionalne zavisnosti	43
3.1.5. Višeznačne zavisnosti.....	44
3.1.6. Osobine višeznačnih zavisnosti	46
3.2. Fazi relacioni model baza podataka	46
3.2.1. Nesavršene informacije i teorija fazi skupova.....	48
3.2.2. Predstavljanje podataka u fazi modelu	49
3.2.3. Relacija bliskosti	51
3.2.4. Fazi upiti	53
3.2.5. Redundancija podataka.....	60
3.2.6. Redundacija u fazi relacionim bazama podataka	61
3.3. Fazi objektno-orijentisane baze podataka.....	64
3.4. Fazi sistemi za upravljanje relacionim bazama podataka	65
3.5. Fazi funkcionalne zavisnosti.....	65
3.5.1. Pravila izvođenja za fazi funkcionalne zavisnosti	67
3.5.2. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Raju i Majumdar-a	69
3.5.3. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Shenoj-Melton	71
3.5.4. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Saxena i Tyagi-a	71
3.5.5. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Dubois-Prade	72
3.5.6. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Wei-Yi Liu.....	73
3.5.7. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Sozati i Yazici	74
3.5.8. Primena saglasnosti (conformance) u analizi fazi funkcionalnih zavisnosti.....	77
3.5.9. Analiza podataka i primena funkcionalnih zavisnosti u modelu Cubero -Medina	79
3.5.10. Fazi aproksimativne zavisnosti	81
3.5.11. Fazi višeznačne zavisnosti	84
3.6. Zaključak	85

4. Projektovanje relacionih baza podataka	86
4.1. Logički model baza podataka.....	86
4.2. Logičko projektovanje postupkom normalizacije tabela.....	89
4.2.1. Prva, druga i treća normalna forma	90
4.2.2. Ostale normalne forme	93
4.3. Normalizacija u fazi relacionim modelima baza podataka	93
4.4. Fizička organizacija podataka	97
4.4.1. Fizička reprezentacija relacije	100
4.5. Zaključak.....	102
5. Analiza fazi funkcionalnih zavisnosti u relacionim bazama podataka.....	103
5.1. Uvodna razmatranja.....	103
5.2. Model fazi relacione baze podataka.....	104
5.3. Pretraga fazi funkcionalnih zavisnosti u FRDBMs.....	108
5.4. Uloga fazi logike u ispitivanju fazi funkcionalnih zavisnosti.....	110
5.5. Karakterizacija fazi implikacija u fazi modelima baza podataka.....	111
5.6. Izbor fazi implikacije u algoritmu za identifikovanje fazi funkcionalnih zavisnosti	117
5.7. Model algoritma za pronalaženje fazi funkcionalnih zavisnosti.....	118
5.8. Zaključak	120
6. Algoritam za identifikovanje fazi funkcionalnih zavisnosti	121
6.1. Eksperiment nad realnim fazi relacijama.....	121
6.2. Algoritam za pretragu fazi funkcionalnih zavisnosti u fazi modelu na bazi relacije bliskosti.....	121
6.3. Algoritam za pretragu fazi funkcionalnih zavisnosti u fazi modelu na bazi mogućnosti distribucije	145
6.4. Upotreba fazi priloga	158
6.5. Poređenje primene različitih fazi implikacija	159
6.6. Praktična primena metodologije	160
6.7. Projektovanje fazi relacionih baza podataka.....	162
6.7.1. Prva fazi normalna forma (F1NF)	162
6.7.2. q-Druga fazi normalna forma (q-F2NF)	163
6.7.3. q-Treća fazi normalna forma (q-F3NF).....	164
6.8. Efikasnost algoritma	166
6.9. Zaključak	169
7. Zaključak.....	170
8. Prilog - Programski kod za Algoritam napisan u C#	173
9. Literatura	180
10. Biografija	186

1. Uvod

U ovoj doktorskoj disertaciji predstavljene su rezultati istraživanja u identifikovanju fazi funkcionalnih zavisnosti primenom metoda i alata iz fazi logike. Predmet istraživanja bila je fazi relaciona baza podataka zbog svoje fleksibilnosti i efikasnosti u pogledu pronalaženja zavisnosti između skupa atributa relacione šeme, te mogućnosti primene dobijenih rezultata u logičkom dizajnu i modelovanju iste, definisanju novih tehnika rudarenja podataka i drugih oblasti i polja gde predstavljena metodologija može da nađe svoje mesto i značajnu ulogu. Sama ideja nastala je proučavanjem klasičnih relacionih baza podataka i njihovih ograničenja kao što je nemogućnost interpretacije različitih tipova neodređenosti i nepreciznosti koje se pojavljuju u realnim informacionim sistemima. Integracija ovog ograničenja u klasične relacije dovela nas je do sistema baza podataka koji leže u okviru koncepta veštačke inteligencije. Sinteza dostupnih znanja iz Zadeove teorije fazi skupova i fazi logike, a zatim i razvoj, implementacija i primena fazi relacionih baza podataka bili su osnovni motivi za istraživanje i konkretizaciju rezultata rada u doktorskoj disertaciji.

Savremene korporacije su sve više zavisne od informacija koje se generišu iz baza u kojima se skladište ogromne količine podataka. Upravljanje i manipulacija ovim podacima zahtevaju mehanizme koji omogućavaju pristup informacijama od značaja koje su u većini slučajeva neohodne za donošenje poslovnih odluka. Zbog toga je u disertaciji predstavljen napredni algoritam za pronalaženje fazi funkcionalnih zavisnosti i primena dobijenog skupa zavisnosti u konkretnim situacijama i primerima. Fazi pristup se kontinualno proučava i neprekidno se javljaju nove mogućnosti primene ovog koncepta u analizi problema različite prirode. Problem analize funkcionalnih zavisnosti u fazi relacionim bazama podataka nije nov. U ovoj oblasti postoji značajan broj radova i u poslednje dve decenije veliki broj autora se bavio problemom analize fazi funkcionalnih zavisnosti. Problem kod fazi relacionih baza znanja jeste činjenica da procedura ispitivanja zavisnosti i izvođenja logičkih posledica iz datog skupa atributa u okviru date aksiomatizacije jeste dosta složena. Ne postoji algoritam koji bi odabirao odgovarajuće aksiome i dostupna znanja u datom koraku da bi se došlo do rešenja i odgovarajućih fazi funkcionalnih zavisnosti u bazi podataka, a takođe ne postoji globalna strategija koja bi davala valjane rezultate. Naša pretpostavka je bila da fazi logika predstavlja pogodno rešenje kada projektanti informacionih sistema ne može jednoznačno da odredi zavisnost između podataka i atributa relacije.

Dakle, predmet istraživanja u ovoj doktorskoj disertaciji je fazi relaciona baza podataka. Fazi relacioni model baza podataka je skup relacija uređenih n -torki. Ove n -torke su uzete u formi (d_1, d_2, \dots, d_n) , gde je d_i domen vrednosti datog skupa D_i , d_i je ili neprezan podskup od D_i ili je element takav da pripada D_i . Fazi relacione baze podataka na prirodan način proširuju funkciju klasičnih baza znanja, što obezbeđuje viši nivo adaptacije fazi sistema, kao jedne od osnovnih osobina inteligentnih sistema (pored sistema planiranja, učenja, predikcije, , robota, sistema za pretragu znanja). U samoj biti ovih modela podataka je činjenica da postoji veliki broj aktivnih istraživačkih oblasti koje direktno uključuju ili koriste ove baze znanja. Pitanje nepreciznosti podataka i njihove reprezentacije ja jako zastupljena tema i vrlo važna u raznim oblastima. Navešćemo samo neke od njih: geografski informacioni sistemi (GIS) i reprezentacija prostornih podataka, sistemi za traganje kroz podatke, pretraživanje fazi informacija...Razmatranje sa lingvističke tačke gledišta je veoma važan aspekt koji se koristi u ovim sistemima. Zbog toga, po logici stvari, fazi logika pruža mogućnost karakterizacije, generalizacije i analize podataka u

bazama podataka. Vođeni tom činjenicom, dolazimo do polazne hipoteze da fazi logika može da ima važnu ulogu i u ispitivanju fazi funkcionalnih zavisnosti u bazama podataka.

Navedni predmet istraživanja implicitno ukazuje i na ciljeve istraživanja doktorske disertacije:

- Sistematska studija znanja iz oblasti fazi logike (teorija fazi skupova, fazi implikacije, fazi metod zaključivanja, princip rezolucije, generalizovani modus ponens i modus tolens...)
- Proučavanje postojećih metoda koje se koriste u analizi funkcionalnih zavisnosti u fazi relacionim bazama podataka.
- Istraživanje i razvoj nove metodologije za pronalaženje fazi funkcionalnih zavisnosti koja je zasnovana na određenom fragmentu fazi logike.
- Praktična primena algoritma za pronalaženje fazi funkcionalnih zavisnosti. Rešavanje ovog, ne tako novog problema, ne predstavlja samo teorijski značaj i istraživački izazov, već ima i praktični značaj. Pa je stoga cilj istraživanja i implementacija logičkog modela fazi relacione baze podataka koju daje prethodna teorija i koja podržava očekivane rezultate istraživanja, ali i veza sa drugim oblastima i poljima (GIS, rudarenje podataka, ekstrakcija pravila, statističke baze podataka, redundancija, biznis registri...).
- Analiza efikasnosti i nedostataka prezentovanog rešenja.
- Identifikovanje otvorenih problema i budućih smernica istraživanja u ovoj oblasti.

Celukopna sistematizacija znanja je omogućila generisanje ideje i hipoteze doktorske disertacije. U skladu sa definisanim predmetom i ciljom istraživanja definišemo i hipoteze doktorske disertacije. Kao što smo napomenuli, opšta hipoteza koju smo razmatrali u radu je:

- Teorija fazi skupova i određeni fragment fazi logike mogu da se primenjuju u analizi funkcionalnih zavisnosti u relacionim bazama podataka.

Pored opšte i polazne hipoteze doktorske disertacije, testirali smo i sledeće posebne hipoteze, kroz koje smo nastojali da potvrdimo ili opovrgnemo i opštu hipotezu:

- Razvijena metodologija analize fazi funkcionalnih zavisnosti može da se koristi za logičko dizajniranje fazi relacionih baza podataka.
- Fazi implikacije su adekvatne za procenu jačine zavisnosti između atributa fazi relacione baze podataka.
- Teorija fazi skupova i određeni fragment fazi logike pružaju poboljšane mogućnosti u analizi funkcionalnih zavisnosti.
- Za otkrivanje fazi funkcionalnih zavisnosti mogu da se koriste različite fazi implikacije.
- Fazi funkcionalne zavisnosti povećavaju efikasnost statističkih baza podataka.

Od metoda istraživanja u doktorskoj disertaciji koristili smo teoriju fazi skupova i određene fazi implikacije. Da bi uspešno realizovali istraživanje, zamisao je bila da se koristi i fazi metod zaključivanja, metoda logičkog objašnjenja i naučnog saznanja, metode generalizacije i konkretizacije, kao i eksperiment nad realnim podacima. Naravno, klasične metode prikupljanja i analize podataka su metode koje su korišćene u izradi pristupog rada, te smo i u toku izrade doktorske disertacije pribegavali ovim stručnim izvorima znanja.

Sadržaj doktorske disertacije prikazan je kroz sledeća poglavlja:

- U drugom poglavlju prikazani su osnovni pojmovi i rezultati iz teorije fazi skupova i fazi logike i naveden je značaj i mogućnosti primene ovih rezultata u relacionim bazama podataka. U ovom poglavlju objašnjeni su pojmovi fazi skupa, fazi implikacija, principa rezolucije, fazi rezonovanja, fazi relacije, t-norm i t-konorm.
- U trećem poglavlju definisan je osnovni koncept fazi relacionih baza podataka, funkcionalnih zavisnosti, višeznačnih zavisnosti kao i pregled ostvarenih i najznačajnijih rezultata na polju analize funkcionalnih zavisnosti u fazi relacionim bazama podataka.
- Četrto poglavlje se bavi ulogom i značajom funkcionalnih zavisnosti u projektovanju baza podataka kao osnovnog poligona za primenu dobijenih rezultata istraživanja,
- Peto poglavlje predstavlja rezultat samostalnog istraživanja i razvoj sopstvenih metoda u analizi fazi funkcionalnih zavisnosti. U ovom poglavlju analizirali smo karakterizaciju različitih fazi implikacija u kontekstu njihove primene u otkrivanju fazi funkcionalnih zavisnosti i predstavili model algoritma.
- U šestom poglavlju posmatrali smo rezultate primene algoritma nad realnim podacima, prikazali rezultate eksperimenta i ukazali na praktičnu primenu predložene metodologije.
- U sedmom poglavlju smo se osvrnuli na zaključna razmatranja, predstavili naučni doprinos doktorske disertacije i identifikovali moguće oblasti primene koje mogu da proizađu iz istraživanja, a koje smo predvideli za buduća istraživanja.
- U prilogu je prikazan kod napisan u programskoj jeziku C# koji smo koristili za verifikaciju rezultata istraživanja.

2. Fazi skupovi i fazi logika

U ovom poglavlju uvedene su osnove teorije fazi skupova [3,4,19-22] i dat je prikaz osnovnih koncepata kao i interpretacija fazi skupova, a naročito mogućnost dokazivanja semantičkih operacija na fazi skupovima.

60-tih godina prošlog veka Lotfi Zade uvodi novi koncept kombinujući *crisp* logiku i Lukasiewicz-eve skupove definisane pomoću stepena pripadnosti. Jedna od glavnih Zadeovih ideja kojom se vodio u svojim analizama jeste činjenica da matematika može da se koristi za povezivanje jezika i ljudske inteligencije. Mnoge realne situacije se bolje prikazuju rečima nego matematičkim formulacijama, a upravo fazi logika kroz svoje izraze (fazi skupove) obezbeđuje jednostavnije modeliranje stvarnosti. Stvarni svet se stalno menja i prepun je iznenadnih i nepredvidivih situacija. Drugim rečima, stvarni svet je fazi. Upravo, fazi tehnika nam pruža mogućnost da uspešno upravljamo realnim situacijama.

2.1. Uvod u fazi skupove

U matematičkoj teoriji osobine i koncepti mogu se opisati pomoću specifičnog podskupa datog univerzalnog skupa. Na primer, ako uzmemo da je N skup prirodnih brojeva, univerzalni skup, tada osobina da je broj jednocifren može se izraziti koristeći skup:

$$X = \{1, 2, 3, \dots, 9\} \subseteq N$$

Ova reprezentacija koja je jednostavna u strukturi, ali nije često primenljiva, može se takođe prikazati pomoću predikata tj. jednocifreni broj (n) je istinit za prirodan broj $n \subseteq N$ ako i samo ako decimalna reprezentacija od n sadrži tačno jednu cifru ili

$$X = \{n \in N : \text{jednocifren broj } (n) \}$$

Treći način predstavljanja jednocifrenog broja je pomoću karakteristične funkcije, koja ja za navedeni primer definisana kao:

$$\lambda_X : N \rightarrow \{0,1\}$$

$$\lambda_X(n) = \begin{cases} 1, & \text{ako je jednocifren broj } (n) \\ 0, & \text{u suprotnom} \end{cases}$$

Vrednosti 0 i 1 karakteristične funkcije zovu se i stepenom pripadnosti.

U teoriji fazi skupova, karakteristična funkcija generalizovana je funkcijom pripadnosti tako da svakom elementu iz univerzalnog skupa, umesto dvoelementnog skupa $\{0,1\}$, pridružuje se vrednost iz jedničnog intervala $[0,1]$. Skup koji je definisan na osnovu takve funkcije pripadnosti naziva se fazi skup.

Sledeća definicija formalno definiše koncept fazi skupa.

Definicija 2.1. Neka je X domen. Fazi vrednosti na domenu X okarakterisane sa fazi skupom A na X i funkcijom pripadnosti:

$$\mu_A : X \rightarrow [0,1]$$

Na ovaj način svaki element $x \in X$ ima stepen pripadnosti fazi skupu A , $\mu_A(x) \in [0,1]$. Dakle, fazi skup A kompletno je uređen sa skupom uređenih parova:

$$A = \{(x, \mu_A(x)) \mid x \in X, \mu_A(x) \in [0,1]\}$$

pri čemu je $\mu_A(x)$ stepen pripadnosti elementa x skupu A . Ukoliko je $\mu_A(x)$ veće utoliko ima više istine u tvrdnji da element x pripada skupu A .

U klasičnoj teoriji skupova postoje veoma precizne granice koje razdvajaju elemente koji pripadaju određenom skupu od elemenata van posmatranog skupa. Svaki obični skup (u kontekstu fazi sistema nazvan crisp skup) može se interpretirati preko date karakteristične funkcije, kao specijalan slučaj fazi skupa. Veliki broj skupova sa kojima se susrećemo u realnosti nema precizno određene granice koje razdvajaju elemente u skupu od elemenata izvan skupa. (npr. "kratko" vreme čekanja na semaforu).

Označimo sa $X = \{x_1, x_2, \dots, x_n\}$ konačan skup elemenata x_i , $i = 1, 2, \dots, n$. Skup X može takodje da bude prikazan u obliku:

$$X = x_1 + x_2 + \dots + x_n = \sum_{i=1}^n x_i$$

pri čemu znak '+' označava uniju elemenata.

Fazi skup A definisan na skupu X često se prikazuje u obliku:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i}$$

U slučaju kada X nije konačan skup, kontinulani fazi skup A definisan na domenu X izražava se kao:

$$A = \int_X \frac{\mu_A(x)}{x}$$

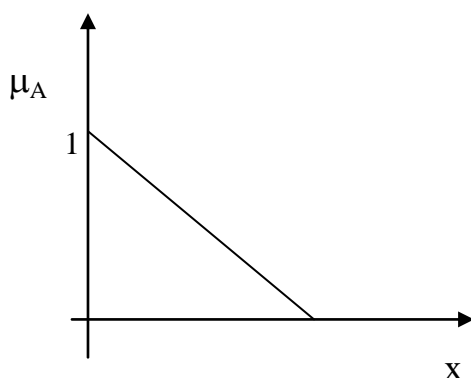
Primer 2.1. $X = \{2, 5, 9, 18, 21, 25\}$. Elementi skupa označavaju broj vozila koja čekaju u redu ispred semafora.

B – "mali broj vozila u redu"

Fazi skup B može, na primer, da bude prikazan kao:

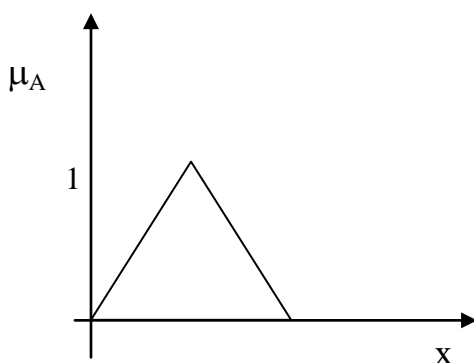
$$B = \frac{0.95}{2} + \frac{0.55}{5} + \frac{0.20}{9} + \frac{0.10}{18} + \frac{0.05}{21} + \frac{0.01}{25}$$

Na sledećoj slici dat je tipičan fazi skup sa funkcijom pripadnosti na jedničnom intervalu:

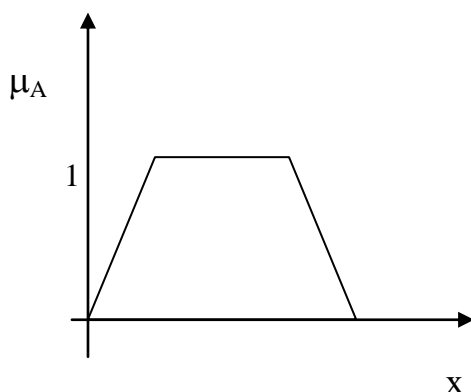


Slika 1. Opadajuća funkcija pripadnosti

Na sledećim slikama ilustrovane su dve važne klase funkcija pripadnosti: trougaona i trapezoidna.

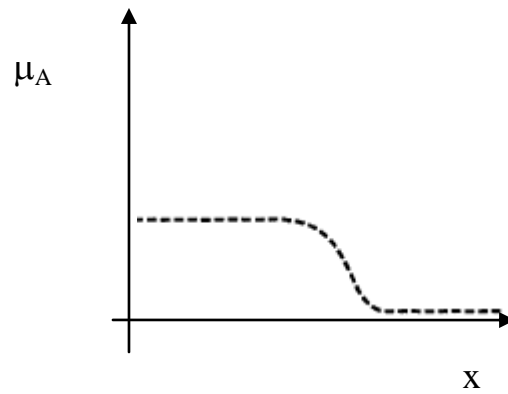


Slika 2. Trougaona forma funkcije pripadnosti

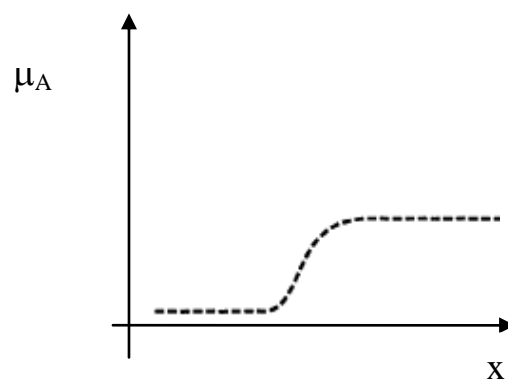


Slika 3. Trapezoidna forma funkcije pripadnosti

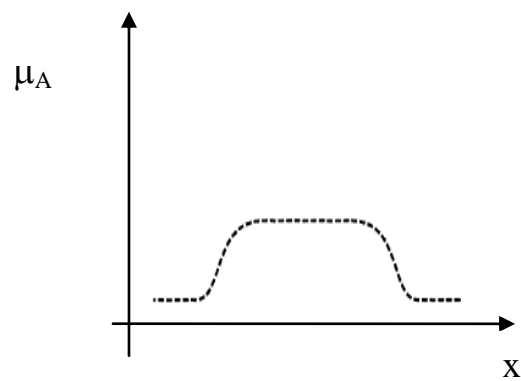
Trouglasta funkcija pripadnosti je najčešće korišćena i najpraktičnija funkcija, ali i drugi oblici koji su prikazani na slikama se koriste. Fazi skup možemo da prikažemo i u vidu kvadratne jednačine, u tom slučaju funkcija pripadnosti je neprekidna kriva, kao što je prikazano na sledećim slikama. Moguća su tri oblika: S-funkcija, Z-funkcija i π -funkcija [69].



Slika 4. Z – funkcija pripadnosti



Slika 5. S – funkcija pripadnosti



Slika 6. π - funkcija pripadnosti

Bilo koji fazi skup A može se opisati pomoću funkcije pripadnosti $\mu_A(x)$, $x \in X$. Ali praktično, ovo je moguće samo ako univerzalni skup (domen) X sadrži konačan broj elemenata. U suprotnom, ako je broj elemenata velik ili beskonačan, tada se stepen pripadnosti $\mu_A(x)$ najbolje određuje uz pomoć odgovarajućih parametara, gde ovi parametri moraju biti usklađeni datom problemu.

Ako je $X=\mathbb{R}$, tada u mnogim slučajevima lingvistički izrazi kao “Visok“, “Oko m “ i “približno između a i d “, mogu biti tretirani kao elementi koji odgovaraju klasi fazi skupova, tako da zadovoljavaju uslov $\exists x \in X : \mu(x)=1$. U ovom slučaju za fazi skup kažemo da je normalan.

U aplikacijama, intepretacija “Visok“ je monotona neopadajuća funkcija sa dva parametra, definisana kao:

$$\mu_A(x) = \begin{cases} 0, & \text{ako je } x \leq a, \\ \frac{x-a}{x-b}, & \text{ako je } a \leq x \leq b, \\ 1, & \text{ako je } x \geq b \end{cases}$$

Za interpretaciju lingvističkog izraza “Oko m “ može se definisati trougaona funkcija kao:

$$\mu_A(x) = \begin{cases} 1 - \left| \frac{m-x}{d} \right|, & \text{ako je } m-d \leq x \leq m+d, \\ 0, & x < m-d \text{ ili } x > m+d \end{cases}$$

Analogno izraz “približno između a i d “ može se okarakterisati pomoću trapezoidne funkcije sa četiri parametra, definisana kao:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{ako je } a \leq x \leq b, \\ 1, & \text{ako je } b \leq x \leq c, \\ \frac{x-d}{c-d}, & \text{ako je } c \leq x \leq d, \\ 0, & \text{ako je } x < a \text{ ili } x > d \end{cases}$$

U prethodim primerima fazi skupovi (u oznaci A) su dati pomoću funkcije pripadnosti $\mu : X \rightarrow [0,1]$. Ova reprezentacija odgovara razmatranju funkcije, čija slika elemenata x , uzeta iz univerzalnog skupa X , uzima vrednost $\mu(x)$. S obzirom na karakterističnu reprezentaciju funkcije pripadnosti μ , ovo nazivamo vertikalnom reprezentacijom odgovarajućeg fazi skupa. Horizontalna reprezentacija fazi skupa definisana je pomoću α -preseka [26].

Definicija 2.2. Neka je A fazi podskup od X i neka je $\alpha \in [0,1]$. Tada se skup

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}$$

naziva α - presek skupa A .

Fazi skupovi su relevantni u mnogim problemima sa kojima se susrećemo: klasifikacija i analiza podataka u savremenim bazama, problemi odlučivanja i aproksimativno rezonovanje. Naravno, ove tipologije problema nisu jedine gde fazi skupovi i funkcija pripadnosti imaju eksplicitnu primenu. Uopšteno posmatrano, većina drugih poslova adresirana na fazi skupove (na primer upiti u relacionim bazama podataka) uključuje ovakvu vrstu istraživanja, pa se u praksi sve češće koristi termin fazi informacioni inženjering.

Tri glavna značenja za funkciju pripadnosti, koja se susreću u literaturi su: sličnost, preferencija i neizvesnost. [75] Svaka od navedenih semantika se nalazi u osnovi posebnih klasa aplikacija. Funkcija pripadnosti fazi skupova ponekad predstavlja i funkciju korisnosti koja predstavlja fleksibilno ograničenje u problemima donošenja odluka. U narednim odeljcima definisana su referentna tumačenja primene funkcije pripadnosti:

Stepen sličnosti: Pomoću funkcije pripadnosti može da se definiše stepen bliskosti i sličnosti posmatranih elemenata. Ovo je, ujedno i najstarija semantika koju uvode Belman i ostali. Ovaj prikaz je naročito značajan u klaster analizi i regresionoj analizi, gde se susrećemo sa problemom predstavljanja podataka i određivanja bliskosti između informacija.

Stepen preferencije: Fazi skup A predstavlja skup manje ili više željenih objekata ili vrednosti varijable na osnovu koje se donose odluke. U tom slučaju, $A(x)$ predstavlja intenzitet prednosti u korist objekta x , odnosno preferenciju izbora objekta (vrednosti) x . Fazi skup je tada kriterijum ili fleksibilno ograničenje. Ovaj pristup koji su izneli Belman i Zade, predstavlja danas osnovu problema fazi optimizacije, fazi linearnog programiranja i analize odluka. Aproksimativno rezonovanje na bazi varijabli i ograničenja koja mogu biti fazi naročito je pogodno za primenu ovakvog pristupa.

Stepen neizvesnosti: Ovo tumačenje je predložio Zade u svojoj teoriji fazi skupova. $\mu_A(x)$ predstavlja stepen pripadnosti elementa x fazi skupu A . Funkcija pripadnosti rangira vrednosti u smislu njihove verodostojnosti. Ovakav pristup se koristi u ekspertnim sistemima.

Jedna od najvažnijih osobina fazi logike jeste njena mogućnost da izrazi stepen neodređenosti u čovekovom razmišljanju i njegovu subjektivnost. Fazi logika se često pogrešno poistovećuje sa verovatnoćom. Osnovna razlika između fazi logike i teorije verovatnoće sastoji se u tome da fazi logika operiše sa determinističkim nedorečenostima i neodređenostima, dok se verovatnoća bavi verodostojnošću stohastičkih događaja i iza nje suštinski stoji eksperiment. Fazi logika prosto ima za cilj prevazilaženje problema u komunikaciji vezanim za razlike između pravila koja nameću formalne teorije i načina razmišljanja koji opisuju ponašanje ljudskog uma, dok se verovatnoća generalno bavi fenomenom ponavljanja koji se simbolizuje slučajnošću (slučajnim promenljivim i slučajnim procesima). Drugim rečima, fazi i slučajni su dva atributa koji se razlikuju u svojoj prirodi, odnosno, oni opisuju drugačiji aspekt neodređenosti. Dakle, fazi logika pokriva subjektivnost ljudskog mišljenja, osećanja, jezika, dok verovatnoća pokriva objektivnu statistiku u prirodnim naukama. Analogno, fazi modeli i modeli formirani na bazi verovatnoće nose drugačiji vid informacija: fazi funkcija pripadnosti predstavlja

sličnost objekata u kontekstu neprecizne definicije osobina, dok verovatnoća daje informaciju o frekvenciji ponavljanja.

Sa matematičke tačke gledišta, fazi skup je preslikavanje skupa elemenata u opseg koji ima strukturu rešetke i obično linearni poredak. Takva osnovna ideja skupa sa funkcijom pripadnosti ili ponderisanim elementima pokazala se veoma korisnom u analizi znanja i predstavljanju informacija. Zaista, stepen pripadnosti, kao što smo i naveli može da se koristi u različitim kontekstima: sličnost, preferencija, neodređenost. U bazama podataka, nepoznata vrednost atributa može da se predstavi fazi skupom (manje ili više verodostojnim vrednostima). Kao generalizacija teorije skupova, fazi skupovi mogu da se kombinuju sa operacijama i relacijama definisanim u teoriji skupova na osnovu principa proširenja, a o tome će biti reči u narednim poglavljima.

2.2. Fundamentalni koncept fazi skupova

Ovde, kao prvo, uvodimo koncepte u teoriji fazi skupova koji ne postoje u klasičnoj teoriji skupova. Fazi skupovi su okarakterisani sa visinom i nosačem [4,19], pa ćemo najpre dati sledeće definicije.

Definicija 2.3. Visina fazi skupa A je najveći stepen pripadnosti elementa u A:

$$\text{Visina}(A) = \sup_{x \in X} (\mu_A(x))$$

Definicija 2.4. Nosač fazi skupa je klasičan skup elemenata $x \in X$ takvih da je $\mu_A(x) > 0$:

$$\text{Supp}(A) = \{x \mid \mu_A(x) > 0, x \in X\}$$

Definicija 2.5. Fazi skup A univerzalnog skupa X zove se normalnim, ako postoji bar jedan element $x \in X$ takav da je $\mu_A(x) = 1$. Fazi skup koji nije normalan zove se subnormalnim. Svi klasični skupovi izuzev praznog skupa su normalni.

Definicija 2.6. Neka je A fazi skup od X. Jezgro od A, je podskup od X, koji sadrži sve elemente sa stepenom pripadnosti 1:

$$\text{Jezgro}(A) = \{x \mid \mu_A(x) = 1, x \in X\}$$

Normalni fazi skup ima nenulto jezgro, dok subnormalni fazi skup ima nulto jezgro. Sada ćemo dati proširene definicije na fazi skupovima :

Definicija 2.7. Neka su A i B dva fazi skupa definisana na univerzalnom skupu X. Kažemo da je A podskup fazi skupa B, $A \subseteq B$, ako je $\mu_B(x) \geq \mu_A(x)$, za svako $x \in X$.

Primer 2.2. Neka je $A = \{0.1/a, 0.6/b, 0.1/c\}$ i $B = \{0.4/a, 0.8/b, 0.1/c\}$. Na osnovu prethodne definicije sledi da je $A \subset B$. Odavde vidimo da ova definicija proširuje ideju klasičnih skupova.

Neka je univerzalni skup $X = \{a, b, c, d, e\}$ i neka su data dva skupa $A = \{a, b\}$ i $B = \{a, b, c\}$ na X. Očigledno je da je $A \subset B$. U fazi skupovima imamo $A = \{1/a, 1/b, 0/c, 0/d, 0/e\}$, $B = \{1/a, 1/b, 1/c, 0/d, 0/e\}$, a s obzirom da je $\mu_B(x) \geq \mu_A(x)$, za svako $x \in X$ sledi da je $A \subset B$.

Definicija 2.8. Neka su A i B dva fazi skupa definisana na skupu X. Kažemo da je fazi skup A jednak fazi skupu B, $A = B$, ako je $A \subset B$ i $B \subset A$, odnosno $A = B$ ako i samo ako je $\mu_A(x) = \mu_B(x)$, za svako $x \in X$.

Definicija 2.9. Nulti fazi podskup od X je definisan kao fazi podskup θ od X takav da je $\theta(x) = 0$ za $x \in X$. Primetimo da za bilo koji podskup A od X je $\theta \subset A$. Dalje, pošto X

može da bude fazi skup za koje je $\mu_X(x) = 1$ za svako $x \in X$, tada je za bilo koje A , $A \subset X$.

Definicija 2.10. Neka je A fazi skup od X gde je $X = \{x_1, \dots, x_n\}$. Tada se u fazi skupu A , definiše diskretna notacija kao $\sum_{i=1}^n \mu_A(x_i)$. Primetimo da je za klasične skupove *diskretna notacija* ekvivalentna kardinalitetu skupa.

Definicija 2.11. Neka je A fazi skup na univerzalnom skupu X . Radni domen od skupa A , označen sa $R_{\text{dom}}(A)$ je isto što i nosač skupa $\text{Supp}(A)$.

Primer 2.3. Pretpostavimo da je mogući broj osvojenih bodova na nekom ispitu jedan od elemenata skupa $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. Ovo je, u svetlu prethodne priče, univerzalni skup X . Posmatrajmo potom tri fazi skupa: $A = \text{“Odličan uspeh“}$, $B = \text{“Osrednji uspeh“}$ i $C = \text{“Slab uspeh“}$, čije su funkcije pripadnosti prikazane sledećom tabelom:

Broj poena	Odličan uspeh (A)	Osrednji uspeh (B)	Slab uspeh (C)
10	0	0	1
20	0	0	1
30	0	0.1	0.9
40	0	0.5	0.7
50	0.1	0.8	0.5
60	0.3	1	0.3
70	0.5	0.8	0.1
80	0.8	0.5	0
90	1	0	0
100	1	0	0

Tabela 1. Funkcije pripadnosti fazi skupova A, B, C

Tada je:

$$\text{Supp}(A) = \{50, 60, 70, 80, 90, 100\}$$

$$\text{Supp}(B) = \{30, 40, 50, 60, 70, 80\}$$

$$\text{Supp}(C) = \{10, 20, 30, 40, 50, 60, 70\}$$

Definicija 2.12. Fazi skup A čija se podrška sastoji od samo jednog elementa univerzalnog skupa X i pri tome je $\mu_A(x)=1$, naziva se **fazi singleton (fazi singleton)**.

Definicija 2.13. Element $x \in X$ za koji važi $\mu_A(x)=0.5$ naziva se **tačka prolaska (crossover point)**.

2.3. Princip rezolucije i reprezentacije

Na osnovu definisanih α -preseka, moguće je uvesti jednu veoma važnu osobinu teorije fazi skupova koja se naziva **princip rezolucije (resolution principle)** [57]. Ova osobina otvara mogućnost ekspanzije fazi skupa u niz njegovih α -preseka.

Teorema 1. Neka je A fazi skup definisan na nekom univerzalnom skupu X . Tada se funkcija pripadnosti ovog fazi skupa može izraziti na osnovu α -preseka u sledećem obliku:

$$\mu_A(x) = \sup [\alpha \wedge \mu_{A_\alpha}(x)] \quad \forall x \in X, \alpha \in [0, 1]$$

gde operator „ \wedge “ označava operator minimuma i pri tome $\mu_{A_\alpha}(x)$ označava funkciju pripadnosti konvencionalnog skupa A_α

$$A_\alpha = \left\{ \begin{array}{l} 1, x \in A_\alpha \\ 0, x \notin A_\alpha \end{array} \right\}$$

Dokaz: Neka „ \vee “ označava operator maksimuma. Kako je $\mu_{A_\alpha}(x)=1$ kada $x \in A_\alpha$ i $\mu_{A_\alpha}(x)=0$ kada $x \notin A_\alpha$, imamo:

$$\begin{aligned} \sup_{\alpha \in [0,1]} [\alpha \wedge \mu_{A_\alpha}(x)] &= \sup_{\alpha \in [0, \mu_A(x)]} [\alpha \wedge \mu_{A_\alpha}(x)] \vee \sup_{\alpha \in [\mu_A(x), 1]} [\alpha \wedge \mu_{A_\alpha}(x)] \\ &= \sup_{\alpha \in [0, \mu_A(x)]} [\alpha \wedge 1] \vee \sup_{\alpha \in [\mu_A(x), 1]} [\alpha \wedge 0] = \sup_{\alpha \in [0, \mu_A(x)]} \alpha = \mu_A(x) \end{aligned}$$

Ovim je dokaz teoreme završen.

Ova teorema direktno vodi ka reprezentaciji fazi skupova na osnovu principa rezolucije. Neka je A fazi skup definisan na univerzalnom skupu X . Neka je, dalje, αA_α fazi skup čija je funkcija pripadnosti:

$$\mu_{\alpha A_\alpha}(x) = [\alpha \wedge \mu_{A_\alpha}(x)], \forall x \in X$$

To znači da je αA_α takav fazi skup (vrlo sličan konvencionalnom) koji svakom elementu čija je funkcija pripadnosti skupu A veća ili jednaka α pridružuje funkciju pripadnosti jednaku α , dok svim ostalim elementima pridružuje funkciju pripadnosti jednaku 0. Tada se, korišćenjem principa rezolucije, skup A može predstaviti na sledeći način:

$$A = \bigcup_{\alpha \in A} \alpha A_\alpha \quad \text{ili} \quad A = \int_0^1 \alpha A_\alpha$$

Teorema o dekompoziciji (A. Kofman): Svaki fazi skup A može se razložiti na proizvod običnih podskupova sa datim koeficijentima α_i .

Princip rezolucije označava da se fazi skup A može dekomponovati na skup fazi skupova αA_α , $\alpha \in [0,1]$. Sa druge strane, fazi skup A može biti predstavljen unijom odgovarajućih αA_α skupova, što se naziva **teoremom reprezentacije**. Drugim rečima, fazi skup se može predstaviti u obliku odgovarajućih α -preseka bez gubitka informacije o funkciji pripadnosti.

Primer 2.4. Ako posmatramo fazi skup A definisan u prethodnom primeru, možemo pisati:

$$A = \frac{0.1}{50} + \frac{0.3}{60} + \frac{0.5}{70} + \frac{0.8}{80} + \frac{1}{90} + \frac{1}{100}$$

Koristeći se principom rezolucije, skup A se može napisati na sledeći način:

$$\begin{aligned} A &= \frac{0.1}{50} + \frac{0.3}{60} + \frac{0.5}{70} + \frac{0.8}{80} + \frac{1}{90} + \frac{1}{100} = \frac{0.1}{50} + \frac{0.1}{60} + \frac{0.1}{70} + \frac{0.1}{80} + \frac{0.1}{90} + \frac{0.1}{100} + \frac{0.3}{60} + \\ &+ \frac{0.3}{70} + \frac{0.3}{80} + \frac{0.3}{90} + \frac{0.3}{100} + \frac{0.5}{70} + \frac{0.5}{80} + \frac{0.5}{90} + \frac{0.5}{100} + \frac{0.8}{80} + \frac{0.8}{90} + \frac{0.8}{100} + \frac{1}{90} + \frac{1}{100} \end{aligned}$$

ili u domenu αA_α skupova:

$$\begin{aligned}
A &= 0.1\left(\frac{1}{50} + \frac{1}{60} + \frac{1}{70} + \frac{1}{80} + \frac{1}{90} + \frac{1}{100}\right) + 0.3\left(\frac{1}{60} + \frac{1}{70} + \frac{1}{80} + \frac{1}{90} + \frac{1}{100}\right) + \\
&0.5\left(\frac{1}{70} + \frac{1}{80} + \frac{1}{90} + \frac{1}{100}\right) + 0.8\left(\frac{1}{80} + \frac{1}{90} + \frac{1}{100}\right) + 1\left(\frac{1}{90} + \frac{1}{100}\right) = \\
&= 0.1A_{0.1} + 0.3A_{0.3} + 0.5A_{0.5} + 0.8A_{0.8} + 1A_1
\end{aligned}$$

Dakle, ovim primerom ilustrujemo princip rezolucije. Sledećim primerom će biti ilustrovan **princip reprezentacije**. Ako su dati sledeći α -preseki: $A_{0.1}=\{1,2,3,4,5\}$, $A_{0.4}=\{2,3,5\}$, $A_{0.8}=\{2,3\}$, $A_1=\{3\}$, tada na osnovu teoreme o reprezentaciji možemo napisati:

$$\begin{aligned}
A &= 0.1A_{0.1} + 0.4A_{0.4} + 0.8A_{0.8} + 1A_1 = 0.1\left(\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5}\right) + 0.4\left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5}\right) + \\
&+ 0.8\left(\frac{1}{2} + \frac{1}{3}\right) + 1\left(\frac{1}{3}\right) = \frac{0.1}{1} + \frac{0.8}{2} + \frac{1}{3} + \frac{0.1}{4} + \frac{0.4}{5}
\end{aligned}$$

2.4. Operacije nad fazi skupovima

U ovoj sekciji ćemo proširiti operacije iz teorije klasičnih skupova na fazi skupove. Pomoću prethodnih notacija i definicija fazi skupova, u mogućnosti smo da predstavimo neke osnovne operacije koje se nad fazi skupovima (i između njih) mogu izvoditi. Neka su A i B fazi skupovi na univerzalnom skupu X .

Definicija 2.14. Za date fazi skupove A i B čije funkcije pripadnosti su $\mu_A(x)$ i $\mu_B(x)$, funkcija pripadnosti unije fazi skupova $A \cup B$, u oznaci $\mu_{A \cup B}(x)$, je definisana kao :

$$\mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)], \forall x \in X$$

Definicija 2.15. Za date fazi skupove A i B čije funkcije pripadnosti su $\mu_A(x)$ i $\mu_B(x)$, funkcija pripadnosti preseka fazi skupova $A \cap B$, u oznaci $\mu_{A \cap B}(x)$, je definisana kao :

$$\mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)], \forall x \in X$$

Za bilo koje $a \in A$ i $b \in B$, operacije \max i \min mogu biti određene i algebarskim izrazom:

$$\begin{aligned}
\max(a, b) &= \frac{a + b + |a - b|}{2} \\
\min(a, b) &= \frac{a + b - |a - b|}{2}
\end{aligned}$$

Primer 2.5. Za fazi skupove definisane u primeru 2.3. imamo:

$$\begin{aligned}
A &= \frac{1}{10} + \frac{1}{20} + \frac{1}{30} + \frac{1}{40} + \frac{0.9}{50} + \frac{0.7}{60} + \frac{0.5}{70} + \frac{0.2}{80} \\
A \cap B &= \frac{0.1}{50} + \frac{0.3}{60} + \frac{0.5}{70} + \frac{0.5}{80} \\
A \cup B &= \frac{0.1}{30} + \frac{0.5}{40} + \frac{0.8}{50} + \frac{1}{60} + \frac{0.8}{70} + \frac{0.8}{80} + \frac{1}{90} + \frac{1}{100}
\end{aligned}$$

Neka su A , B i C fazi skupovi i neka je $x \in A$, $y \in B$ i $z \in C$. Tada iz definicija unije i preseka sledi da vrede sledeće osobine:

1. Komutativnost

$$\mu_A(x) \cup \mu_B(y) = \mu_B(y) \cup \mu_A(x)$$

$$\mu_A(x) \cap \mu_B(y) = \mu_B(y) \cap \mu_A(x)$$

2. Idempotenstnost

$$\mu_A(x) \cup \mu_A(x) = \mu_A(x)$$

$$\mu_B(y) \cap \mu_B(y) = \mu_B(y)$$

3. Asocijativnost

$$\mu_A(x) \cup [\mu_B(y) \cup \mu_C(z)] = [\mu_A(x) \cup \mu_B(y)] \cup \mu_C(z)$$

$$\mu_A(x) \cap [\mu_B(y) \cap \mu_C(z)] = [\mu_A(x) \cap \mu_B(y)] \cap \mu_C(z)$$

4. Distributivnost

$$\mu_A(x) \cup [\mu_B(y) \cap \mu_C(z)] = [\mu_A(x) \cup \mu_B(y)] \cap [\mu_A(x) \cup \mu_C(z)]$$

$$\mu_A(x) \cap [\mu_B(y) \cup \mu_C(z)] = [\mu_A(x) \cap \mu_B(y)] \cup [\mu_A(x) \cap \mu_C(z)]$$

5. Identitet

$$\mu_A(x) \cup \mu_\emptyset = \mu_A(x)$$

$$\mu_A(x) \cap \mu_\emptyset = \mu_A(x)$$

gde μ_\emptyset označava stepen pripadnosti praznom (nultom) skupu.

6. Apsorpcija

$$\mu_A(x) \cup [\mu_A(x) \cap \mu_B(y)] = \mu_A(x)$$

$$\mu_A(x) \cap [\mu_A(x) \cup \mu_B(y)] = \mu_A(x)$$

Sada proširimo definiciju operacije relativnoga komplementa i komplementa (ili negacije) u fazi podskupovima.

Definicija 2.16. Neka su A i B dva fazi skupa na univerzalnom skupu X . Relativni komplement od B sa respektom na A , $E = A - B$ je definisan kao fazi skup E od X gde za svako $x \in X$ je:

$$\mu_E(x) = \max [0, \mu_A(x) - \mu_B(x)]$$

Definicija 2.17. Neka su A i B dva fazi skupa na univerzalnom skupu X . Komplement ili negacija od A , označena sa $\neg A$, je definisana kao fazi skup $\neg A$ gde za svako $x \in X$ funkcija pripadnosti:

$$\neg \mu_A(x) = 1 - \mu_A(x)$$

Sada navodimo listu osobina koje uključuju komplement i relativni komplement fazi skupova A i B od X :

1. Dupla negacija

$$\neg [\neg \mu_A(x)] = \mu_A(x), x \in X$$

2. De Morganovi zakoni

$$\neg [\mu_A(x) \cup \mu_B(y)] = \neg\mu_A(x) \cup \neg\mu_B(y)$$

$$\neg [\mu_A(x) \cap \mu_B(y)] = \neg\mu_A(x) \cap \neg\mu_B(y)$$

$$3. \neg \mu_\theta = X, \neg X = \theta$$

$$4. \text{ Ako je } \mu_A(x) \leq \mu_B(y) \text{ tada je } \neg\mu_A(x) > \neg\mu_B(y)$$

$$5. \mu_A(x) - \mu_A(x) = \theta$$

$$6. \mu_A(x) - \theta = \mu_A(x), \theta - \mu_A(x) = \theta$$

Ovde primećujemo da definicije za relativni komplement i komplement su ekvivalentne definicijama za klasične skupove.

Druga značajna osobina jeste da generalno ne vredi, odnosno da je $\neg A \cup A \neq X$ i $\neg A \cap A \neq \theta$. Ovi zakoni koji važe sa svaki konvencionalni skup više nisu istiniti i validni u teoriji fazi skupova.

Sada uvedimo operacije nad fazi skupovima koje nisu korištene u klasičnoj teoriji skupova:

Definicija 2.18. Neka je A fazi skup od X i neka je α proizvoljan nenegativan broj. Tada se sa A^α označava fazi skup B od X takav da je:

$$\mu_B(x) = (\mu_A(x))^\alpha$$

Lako je pokazati da ako je $\alpha > 1$ tada je A^α fazi podskup od A . Isto tako ako je $\alpha < 1$ tada je $A \subset A^\alpha$. Ako je A klasičan skup i $\alpha > 0$ tada je $A^\alpha = A$. U slučaju da je $\alpha > 1$ gornju operaciju nazivamo **koncentracijom**, a u slučaju da je $\alpha < 1$ tada je zovemo **dilatacijom**.

2.5. t-norma i t-konorma

Razmotrimo sada operacije preseka i unije nad fazi skupovima, koje se ubrajaju u **trougaone norme (t-norms)** i **trougaone konorme (t-conorms)**, respektivno. Počevši sa operacijom preseka, mi pretpostavljamo da presek $\mu \cap \mu'$ dva fazi skupa, koji želimo definisati, može biti izračunat tačku po tačku. [26,84]

Definicija 2.19. **t-norma** je dvoparameterska funkcije i predstavlja se u formi:

$$t: [0,1] \times [0,1] \rightarrow [0,1]$$

tako da važi:

$$\mu_A \cap_B = t[\mu_A(x), \mu_B(y)]$$

pri čemu funkcija t zadovoljava sledeće uslove:

Ograničenost: $t(\mu_A(x), 1) = \mu_A(x)$,

Monotonost: $\mu_A(x) < \mu_B(x) \Rightarrow t(\mu_A(x), \mu_C(x)) < t(\mu_B(x), \mu_C(x))$

Komutativnost: $t(\mu_A(x), \mu_B(x)) = t(\mu_B(x), \mu_A(x))$

Asocijativnost: $t(\mu_A(x), t(\mu_B(x), \mu_C(x))) = t(t(\mu_A(x), \mu_B(x)), \mu_C(x))$

t-norme i t-konorme se često koriste za definisanje drugih operacija. Tipične **neparametarske t-norme** (jednostavnosti radi, koristićemo sledeće notacije: $\mu_A(x)=a$ i $\mu_B(x)=b$) koje su često korištene u aplikacijama su:

$$t_{\min}(a,b) = \min\{a,b\},$$

$$t_{\text{Lukasiewicz}}(a,b) = \max\{0, a + b - 1\},$$

$$t_{\text{prod}}(a,b) = ab.$$

Dualni zapis t - norme jeste t - konorm, koja definiše operaciju unije.

Definicija 2.20. t-konorma (takođe se zove i s-norma) je dvoparametarska funkcija i predstavlja se u formi:

$$s: [0,1] \times [0,1] \rightarrow [0,1]$$

tako da važi:

$$\mu_A \cup_B = s[\mu_A(x), \mu_B(y)]$$

uz ispunjenost sledećih uslova:

Ograničenost: $s(\mu_A(x), 1) = 1,$

Monotonost: $\mu_A(x) < \mu_B(x) \Rightarrow s(\mu_A(x), \mu_C(x)) < s(\mu_B(x), \mu_C(x)),$

Komutativnost: $s(\mu_A(x), \mu_B(x)) = s(\mu_B(x), \mu_A(x)),$

Asocijativnost: $s(\mu_A(x), s(\mu_B(x), \mu_C(x))) = s(s(\mu_A(x), \mu_B(x)), \mu_C(x)),$

Iz gornjih uslova, lako je videti da vredi $s(1,1) = s(1,0) = s(0,1) = 1, s(0,0) = 0$. Poznati primeri t-konorma su sledeći:

$$s_{\max}(a,b) = \max\{a,b\},$$

$$s_{\text{Lukasiewicz}}(a,b) = \min\{1, a + b\},$$

$$s_{\text{prod}}(a,b) = a + b - ab.$$

t-norme i t-konorme predstavljaju vrstu operacija agregacije nad fazi skupovima, tj. ovo su operacije pomoću kojih se kombinuje nekoliko fazi skupova da proizvedu jedan fazi skup. Stoga ove operacije igraju važnu ulogu u smislu odluka koje se donose u fazi okruženju.

Z a d e L. [20-22] uvodi sledeće operatore:

$$(\mu \cup_T \mu')(x) = \max(\mu(x), \mu'(x))$$

$$(\mu \cap_T \mu')(x) = \min(\mu(x), \mu'(x))$$

$$\neg \mu(x) = 1 - \mu(x)$$

gde je operacija negacija definisana na sledeći način.

Definicija 2.21. Preslikavanje $N: [0,1] \rightarrow [0,1]$ zovemo operatorom negacije ako je:

1. $N(0) = 1, N(1) = 0$
2. $a < b$ implicira $N(a) > N(b)$
3. $N(N(a)) = a$

Korištenjem ove definicije, operator negacije za fazi skupove definiše se kao:

$$\neg \mu(x) = N(\mu(x))$$

Definicija negacije, $\neg \mu(x) = 1 - \mu(x)$, za fazi skupove je zasnovana na upotrebi operatora negacije $N(a) = 1 - a$, koji očigledno zadovoljava gornje uslove.

Uvođenjem operatora negacije, u mogućnosti smo da odredimo uopštene De - Morganove zakone, koji dovode u vezu t - normu i t - konormu. Pretpostavimo da je t bilo koja proizvoljna t - norma. Tada se za operator s definisan pomoću:

$$s(a,b) = 1 - t(\neg a, \neg b)$$

može pokazati da je t - konorma. Za dati t operator, odgovarajući s operator definisan prethodno, naziva se njegov **dual**.

Takođe, može se dokazati da ako je s t - konorma, tada $t(a,b) = 1 - s(\neg a, \neg b)$ je t - norma. Važna implikacija ovoga jeste da, ako želimo definisati presek i uniju u fazi skupovima, tako da De Morganovi zakoni budu zadovoljeni:

$$\begin{aligned} \neg(\mu \cup \mu') &= (\neg\mu \cap \neg\mu') \text{ i} \\ \neg(\mu \cap \mu') &= (\neg\mu \cup \neg\mu') \end{aligned}$$

tada t i s operatori moraju biti dualni jedan drugome.

Veoma važne, sa matematičkog stanovišta, su Arhimedova t - norma i t - konorma koje su definisane na sledeći način:

Definicija 2.22. Neka su $t: [0,1]^2 \rightarrow [0,1]$ i $s: [0,1]^2 \rightarrow [0,1]$ dve funkcije.

a) t se naziva **Arhimedovom t - normom** ako i samo ako je t *neprekidna* t-norma i vredi $t(a,a) < a$, za svako $a \in (0,1)$.

b) s se naziva **Arhimedovom t -konormom** ako i samo ako je s *neprekidna* t-konorma i vredi $t(a,a) > a$, za svako $a \in (0,1)$.

Arhimedova t - norma može biti reprezentovana na sledeći način :

Teorema: Funkcija $t: [0,1]^2 \rightarrow [0,1]$ je Arhimedova t - norma ako i samo ako postoji striktno opadajuća neprekidna funkcija $f: [0,1] \rightarrow [0,\infty]$ sa $f(1) = 0$ i $t(a,b) = f^{(-1)}(f(a) + f(b))$, gde je $f^{(-1)}$ pseudo-inverzna funkcija od f, tj.

$$f^{(-1)}(y) = \begin{cases} x \in [0,1]: f(x) = y, y \in [0, f(0)] \\ 0, y \in [f(0), \infty] \end{cases}$$

Ako je $f(0) = \infty$, tada je t striktno rastuća.

2.6. Fazi relacije

U ovom poglavlju proširujemo koncept relacije na fazi skup, uvođenjem koncepta fazi relacije [19,55,84]. Pojam fazi relacije moguće je objasniti na sličan način kao što je objašnjen pojam fazi skupa. Između dva klasična skupa postoji određena relacija ukoliko su njihovi elementi međuzavisni, povezani, odnosno uslovljeni jedan drugim. Prelaz od pojma klasične relacije ka pojmu tzv. fazi relacije moguće je obezbediti uvođenjem koncepta “*jačine*” relacije između pojedinih elemenata.

Kažemo, da ako su X i Y dva obična skupa, tada je njihov Dekartov (Cartezian) proizvod, skup $X \times Y$, koji sadrži sve parove (x,y), gde je $x \in X, y \in Y$.

Definicija 2.23. Neka su A i B dva fazi podskupa od X i Y. Njihov Dekartov proizvod $A \times B$ je fazi relacija R na skupu $X \times Y$, zapisana kao $R = A \times B$, gde je:

$$\mu_R(x,y) = \min[\mu_A(x), \mu_B(x)]$$

ili uopšteno, ako su A_1, A_2, \dots, A_n , fazi podskupovi od X_1, X_2, \dots, X_n , respektivno, tada njihov Dekartov proizvod $A_1 \times A_2 \times \dots \times A_n$, je fazi podskup od $X_1 \times X_2 \times \dots \times X_n$ zapisan kao $R = A_1 \times A_2 \times \dots \times A_n$, gde je:

$$\mu_R(x_1, \dots, x_n) = \min \{ \mu_{A_i}(x_i) \mid x_i \in X_i, i = 1, 2, 3, \dots, n \}.$$

Primer 2.6. Neka su $A = \{1/a, 0.6/b, 0.3/c\}$ i $B = \{1/1, 0.5/2, 0/3\}$ dva fazi podskupa na univerzalnim skupovima $X = \{a, b, c\}$ i $Y = \{1, 2, 3\}$. Tada je njihov Dekartov proizvod:

$$A \times B = \{ 1/(a,1), 0.5/(a,2), 0/(a,3), 0.6/(b,1), 0.5/(b,2), 0/(b,3), 0.3/(c,1), 0.3/(c,2), 0/(c,3) \}$$

Nakon što smo definisali osnovnu fazi relaciju nad fazi skupovima, možemo da prikazemo operacije nad fazi relacijama. Neka su R_1 i R_2 dve fazi relacije na X i Y sa funkcijama pripadnosti μ_{R_1} i μ_{R_2} respektivno. Tada je:

1. Unija: $R_1 \cup R_2$ je fazi relacija od X na Y sa funkcijom pripadnosti:

$$\mu_{R_1 \cup R_2}: X \times Y \rightarrow [0,1],$$

tako da za $\forall(x,y) \in X \times Y$ je:

$$\mu_{R_1 \cup R_2}(x,y) = \max(\mu_{R_1}(x,y), \mu_{R_2}(x,y))$$

2. Presek: $R_1 \cap R_2$ je fazi relacija od X u Y sa funkcijom pripadnosti:

$$\mu_{R_1 \cap R_2}: X \times Y \rightarrow [0,1],$$

tako da za $\forall(x,y) \in X \times Y$ je:

$$\mu_{R_1 \cap R_2}(x,y) = \min(\mu_{R_1}(x,y), \mu_{R_2}(x,y))$$

3. Komplement: $C \circ R$ je fazi relacija od X u Y sa funkcijom pripadnosti:

$$\mu_{C \circ R}: X \times Y \rightarrow [0,1]$$

tako da za $\forall(x,y) \in X \times Y$ je:

$$\mu_{C \circ R}(x,y) = 1 - \mu_{R_1}(x,y).$$

4. Razlika: $R_1 - R_2$ je fazi relacija od X u Y sa funkcijom pripadnosti:

$$\mu_{R_1 - R_2}: X \times Y \rightarrow [0,1],$$

tako da za $\forall(x,y) \in X \times Y$ je:

$$\mu_{R_1 - R_2}(x,y) = \min(\mu_{R_1}(x,y), 1 - \mu_{R_2}(x,y))$$

5. Prirodno spajanje: Neka je R_1 fazi relacija od X u Y sa $\mu_{R_1}(x,y): X \times Y \rightarrow [0,1]$ i neka je R_2 fazi relacija od Y u Z sa $\mu_{R_2}(y,z): Y \times Z \rightarrow [0,1]$. Tada je $R_1 * R_2$ fazi relacija između X, Y, Z sa $\mu_{R_1 * R_2}(x,y,z): X \times Y \times Z \rightarrow [0,1]$, tako da za $\forall(x,y) \in X \times Y \times Z$ je:

$$\mu_{R_1 * R_2}(x,y,z) = \min(\mu_{R_1}(x,y), \mu_{R_2}(y,z))$$

6. Projekcija: Neka je R fazi relacija između X_1, \dots, X_n sa $\mu_R: X_1 \times \dots \times X_n \rightarrow [0,1]$. Projekcija (ili marginalna fazi restrikcija) od R , zapisana kao $\Pi_{X_{i_1} \dots X_{i_k}}(R)$, gde je $i_1, \dots, i_k \in (1, \dots, n)$, definisana je kao fazi relacija između X_{i_1}, \dots, X_{i_k} sa funkcijom pripadnosti $\mu_{\Pi_{X_{i_1} \dots X_{i_k}}}(R)$ tako da za $\forall(x_{i_1}, \dots, x_{i_k}) \in X_{i_1} \times \dots \times X_{i_k}$:

$$\mu_{\Pi_{X_{i_1} \dots X_{i_k}}}(R)(x_{i_1}, \dots, x_{i_k}) = \sup \mu_R(u_1, \dots, u_n), u_i = x_{i_j}, i=1, \dots, k.$$

7. Cilindrično proširenje: Specijalan slučaj Dekartovog proizvoda zovemo cilindrično proširenje. Neka je R fazi relacija između X_1, X_2, \dots, X_k sa $\mu_R: X_1 \times \dots \times X_k \rightarrow [0,1]$. Cilindrično proširenje od R u $X_1 \times X_2 \dots \times X_k \times X_{k+1} \times X_{k+2} \dots \times X_n$ označeno sa R' je fazi relacija između $X_1 \times \dots \times X_k \times X_{k+1} \times X_{k+2} \dots \times X_n$ sa $\mu_{R'}: X_1 \times \dots \times X_k \times X_{k+1} \times X_{k+2} \dots \times X_n \rightarrow [0,1]$ tako da za $\forall (x_1, x_2, \dots, x_k, x_{k+1}, x_{k+2}, \dots, x_n) \in X_1 \times \dots \times X_k \times X_{k+1} \times X_{k+2} \dots \times X_n$:

$$\mu_{R'}(x_1, x_2, \dots, x_k, x_{k+1}, x_{k+2}, \dots, x_n) = \mu_R(x_1, x_2, \dots, x_k).$$

Još jedna važna operacija u fazi relacijama jeste kompozicija koja se definiše na sledeći način. Neka su X, Y i Z neprazni skupovi i neka je R_1 fazi relacija iz X u Y , i neka je R_2 fazi relaciji iz Y u Z . Max-min kompozicija fazi relacija R_1 i R_2 , označena kao $R_1 \circ R_2$, definiše fazi relaciju iz X u Z tako da za $\forall (x, z) \in X \times Z$ vredi:

$$\mu_{R_1 \circ R_2}(x, z) = \sup \min(\mu_{R_1}(x, y), \mu_{R_2}(y, z)), y \in Y$$

U fazi relacijama vrede sledeće osobine:

- a) Asocijativnost: $(R_1 \circ R_2) \circ R_3 = R_1 \circ (R_2 \circ R_3)$,
- b) Monotonost: $R_1 \subseteq R_2 \Rightarrow R_1 \circ R_3 \subseteq R_2 \circ R_3$
- c) Distributivnost: $(R_1 \cup R_2) \circ R_3 = (R_1 \circ R_3) \cup (R_2 \circ R_3)$

$$(R_1 \cap R_2) \circ R_3 = (R_1 \circ R_3) \cap (R_2 \circ R_3)$$

Definicija 2.24. Neka je R binarna fazi relacija na X (tj. iz X u X) sa $\mu_R: X \times X \rightarrow [0,1]$. R je relacija sličnosti ako i samo ako je:

- refleksivna: $\forall x \in X, \mu_R(x, x) = 1$,
- simetrična: $\forall (x_1, x_2) \in X \times X, \mu_R(x_1, x_2) = \mu_R(x_2, x_1)$,
- max-min tranzitivna $\forall (x_1, x_3) \in X \times X, \mu_R(x_1, x_3) \geq \sup \min(\mu_R(x_1, x_2), \mu_R(x_2, x_3)), x_2 \in X$.

Fazi logika [56-59] je zasnovana na konceptu fazi skupova i simboličke logike. Logički operatori konjunkcije, disjunkcije i negacije definišu se na sledeći način:

- a) $x_1 \wedge x_2 = \min(x_1, x_2)$
- b) $x_1 \vee x_2 = \max(x_1, x_2)$
- c) $\neg x = 1 - x$

gde je x_i ($i=1, 2, \dots, n$) promenljiva sa značenjem iz intervala $[0,1]$.

Fazi formula definiše funkcije iz $[0,1]^n$ u $[0,1]$. Definišimo sada fazi formulu pomoću sledećih pravila:

- a) 0 i 1 su fazi formule,
- b) Fazi promenljiva x_i je fazi formula,
- c) Ako je f fazi formula, tada je $\neg f$ fazi formula,
- d) Ako su f i g fazi formule tada su to $f \vee g$,
- e) Drugih formula nema.

U fazi logici istinita vrednost formule može biti bilo koja vrednost iz intervala $[0,1]$. Neka je $T(S)$ zapis za istinitu vrednost formule S . Tada je na snazi: [58].

- a) $T(S) = T(A)$ ako je $S=A$ i A je atomik formula,
- b) $T(S)=1-T(R)$ ako je $S= \neg R$,
- c) $T(S) = \min [T(S_1), T(S_2)]$ ako je $S = S_1 \wedge S_2$,
- d) $T(S) = \max [T(S_1), T(S_2)]$ ako je $S = S_1 \vee S_2$,

- e) $T(S) = \inf [T(B(x)), x \in D]$ ako je $S = \oplus B$, D je domen od x ,
 f) $T(S) = \sup [T(B(x)), x \in D]$ ako je $S = \leq B$, D je domen od x .

2.7. Teorija aproksimativnog rezonovanja

Ova teorija, zasnovana na fazi skupovima, omogućava rezonovanje sa neodređenim informacijama. Originalna ideja donošenja odluka na osnovu *nepreciznih, kvalitativnih* podataka kombinovanjem deskriptivnih lingvističkih pravila potiče od Zadea. Proceduru kombinovanja nepreciznih logičkih pravila u jedinstvenu upravljačku strategiju Zade (1973) naziva **aproksimativnim** ili **fazi rezonovanjem**. Fazi rezonovanje (aproksimativno rezonovanje) predstavlja proceduru zaključivanja, odnosno generisanje zaključka na osnovu premisa koje sadrže lingvističke izraze.

Centralna osobina ove teorije je reprezentovanje propozicija kao naredbi koje su pridružene fazi skupovima. Ulazne i izlazne promenljive u fazi sistemima su lingvističke promenljive ("Visoka zarada", "Kratko vreme čekanja",...). Izlazni rezultat je dat u kontinualnoj formi (svim mogućim vrednostima izlazne promenljive pridružen je odgovarajući stepen pripadnosti). **Defazifikacija** predstavlja izbor jedne vrednosti izlazne promenljive.

U ovoj sekciji uvešćemo sintaksu i operacije u sistemu aproksimativnog rezonovanja [86,87].

Primarni element u sistemu aproksimativnog rezonovanja je skup varijabli zvanih atomskim (baznim) varijablama i pridruženi skupovi X_1, \dots, X_n , gde je X_k osnovni (univerzalni) skup od V_k . Zajednička varijabla je bilo koji skup jedne ili više različitih atomskih varijabli. Propozicija u ovome sistemu je naredba u formi:

$$V \text{ je } M$$

gde je V zajednička varijabla i M je fazi podskup Dekartovog proizvoda osnovnog skupa atomskih varijabli.

Naredba ili propozicija koja uključuje samo jednu varijablu zove se kanonična naredba. Sledi definicija koncepta mogućnosti propozicije:

Definicija 2.25. Za datu propoziciju $V \text{ je } A$ možemo definisati **Mogućnost**[$V \text{ je } A$] = $\text{Max}_x [\mu_A(x)]$.

Propoziciju $V \text{ je } A$ zovemo normalnom ako je $\text{Mogućnost}[V \text{ je } A] = 1$ tj. postoji bar jedan element x u osnovnom skupu A takav da je $\mu_A(x)=1$. Propozicija se zove kontradikcija ako je $\text{Mogućnost}[V \text{ je } A] = 0$.

Definicija 2.26. Propozicija $V \text{ je } A$ naziva se tautologijom ako je $\mu_A(x)=1$ za svako x iz domena V .

Rad sa propozicijama u sistemu aproksimativnog rezonovanja je zasnovano na dve fundamentalne operacije. Prva operacija je konjunkcija, a druga operacija je biti sadržan.

Definicija 2.27. Neka su V_A i V_B dve zajedničke varijable na osnovnim skupovima X i Y respektivno i neka su $V_A \text{ je } M$ i $V_B \text{ je } N$ dve propozicije. Tada je njihova konjunkcija, $V_A \text{ je } M$ i $V_B \text{ je } N$ zapisana kao $V_A \text{ je } M \times V_B \text{ je } N$ je propozicija $V \text{ je } P$ gde je V zajednička varijabla koja sadrži uniju atomskih varijabli koje čine V_A i V_B , i P je fazi podskup domena V takav da za svako z iz domena V je $\mu_P(z) = \mu_M(x) \times \mu_N(y)$

U slučaju da je $V_A = V_B$ tada je ova operacija redukovana u operaciju preseka tj.

$$V_A \text{ je } M \text{ i } V_A \text{ je } N = V_A \text{ je } M \cap N.$$

U slučaju kada V_A i V_B nemaju zajedničkih varijabli ova operacija daje klasični Dekartov proizvod:

$$V_A \text{ je } M \cap V_B \text{ je } N = (V_A, V_B) \text{ je } M \times N.$$

Primer za operaciju konjunkcije. Vredi sledeće: Konjunkcija tautologije sa normalnom propozicijom je normalna. Konjunkcija dve tautologije je tautologija. Konjunkcija kontradikcije sa bilo kojom propozicijom je kontradikcija.

Sada ćemo definisati specijalni operator konjunkcije.

Definicija 2.28. Neka su V_A i V_B dve zajedničke varijable takve da V_B sadrži sve varijable iz V_A . Cilindrično proširenje propozicije $V_A \text{ je } M$ u V_B je propozicija $V_B \text{ je } M^\circ$ koja je definisana sa $V_B \text{ je } M^\circ = V_A \text{ je } M \times V_1 \text{ je } X_1 \times \dots \times V_q \text{ je } X_q$ gde su V_1, \dots, V_q atomske varijable iz V_B , a koje nisu V_A , a X_i su osnovni skupovi ovih datih varijabli.

Cilindrično proširenje od $V_A \text{ je } M$ u $V_B \text{ je } M^\circ$ može se izraziti pomoću funkcija pripadnost od M i M° kao $\mu_{M^\circ}(y) = \mu_M(x)$, gde je x element u V_A koji se slaže sa y za delove od V_B koji su u V_A .

Sada uvodimo drugu osnovnu operaciju u sistemu aproksimativnog rezonovanja.

Definicija 2.29. Neka su $V_A \text{ je } M$ i $V_B \text{ je } N$ dve propozicije. Kažemo da $V_A \text{ je } M$ je sadržano u $V_B \text{ je } N$, tj. $(V_A \text{ je } M) \subset (V_B \text{ je } N)$ ako je $\mu_{M^\circ}(z) \geq \mu_{N^\circ}(z)$, gde su M° i N° cilindrična proširenja od M i N na osnovnom skupu varijabli V , koja je unija atomskih varijabli u V_A i V_B .

Primer 2.7. Neka je $X_1 = \{x_1, x_2, x_3\}$ i $X_2 = \{y_1, y_2\}$. Razmotrimo $V_1 \text{ je } M$, gde je $M = \{0.7/x_1, 0.9/x_2, 1.0/x_3\}$, i $V_2 \text{ je } N$ gde je $N = \{0.3/y_1, 0.5/y_2\}$. Tada je:

$$M^\circ = \{0.7/(x_1, y_1), 0.7/(x_1, y_2), 0.9/(x_2, y_1), 0.9/(x_2, y_2), 1/(x_3, y_1), 1/(x_3, y_2)\}$$

$$N^\circ = \{0.3/(x_1, y_1), 0.5/(x_1, y_2), 0.3/(x_2, y_1), 0.5/(x_2, y_2), 0.3/(x_3, y_1), 0.5/(x_3, y_2)\}$$

Kako je za svaki i, j :

$$\mu_{M^\circ}(x_i, y_i) \geq \mu_{N^\circ}(x_i, y_i)$$

imamo da je $V_2 \text{ je } N \subset V_1 \text{ je } M$.

Sledeća teorema uspostavlja relaciju između prethodnih operacija:

Teorema 2. Pretpostavimo da su P_1, P_2 i P_3 tri propozicije. Ako vredi:

$$P_3 = P_1 \times P_2$$

tada je $P_3 \subset P_1$ i $P_3 \subset P_2$. Dokaz sledi neposredno iz definicije konjunkcije. Sada uvedimo koncept ekvivalencije dve propozicije.

Definicija 2.30. Dve propozicije P_1 i P_2 su ekvivalentne ako je $P_1 \subset P_2$ i $P_2 \subset P_1$.

Teorema 3. Ako su V_A i V_B razdvojene varijable tada propozicije:

$$V_A \text{ je } M \text{ i } V_B \text{ je } N$$

su ekvivalentne ako i samo ako postoji neko α iz $[0,1]$ takvo da je $\mu_M(x) = \alpha$ za svako x iz V_A i $\mu_N(y) = \alpha$, za svaki y iz V_B .

2.8. Semantika sistema aproksimativnog rezonovanja

U ovom poglavlju uvedena je semantika u sistemu aproksimativnog rezonovanja. Kada je V atomska varijabla data na univerzalnom skupu X , tada sledi da X je skup svih mogućih vrednosti koju varijabla V može uzeti. Propozicija V_i je A_j označava da vrednost varijable V_i leži u skupu A_j .

Primer 2.8. Neka su date dve varijable V_1 (Grad osobe) i V_2 (Godine osobe). Ako imamo propoziciju:

$$(V_1, V_2) = (\text{Grad}, \text{Godine}) = \{ \mu_i(v_1, v_2) / (v_1, v_2) \}$$

$$(V_1, V_2) = \{ 1/(\text{Bg}, 32), 1/(\text{NS}, 29) \}$$

Ova informacija pokazuje da osoba ili živi u Beogradu i ima 32 godine ili živi u Novom Sadu i ima 29 godina. Preko ove semantike vidimo da operacija konjunkcije označava skup mogućih vrednosti na kojoj se obadve propozicije slažu.

Za de L. uvodi posebna pravila koja nam dozvoljavaju reprezentovanje zajedničkih lingvističkih naredbi u izrazima propozicija. Sada dajemo neka od ovih pravila.

Negacija-propozicije. Naredba: ne V je A prevodi u V je B gde je B negacija od A odnosno:

$$\mu_B(x) = 1 - \mu_A(x).$$

I-propozicija. Naredba: V_A je A i V_B je B definisana je preko operatora konjunkcije kao:

$$V \text{ je } A \wedge B$$

gde je V unija atomskih varijabli u V_A i V_B .

II-propozicija. Naredba: V_A je A ili V_B je B definisana je ne (V_A je ne A i V_B je ne B). Može se pokazati da

$$V_A \text{ je } A \text{ ili } V_B \text{ je } B = V \text{ je } A^\circ \cup B^\circ$$

gde je V unija od V_A i V_B , A° i B° su cilindrična proširenja od A i B i \cup je unija u značenju da:

$$F = A^\circ \cup B^\circ$$

je definisano kao:

$$\mu_F(z) = \max(\mu_{A^\circ}(z), \mu_{B^\circ}(z)) = \mu_{A^\circ}(z) \vee \mu_{B^\circ}(z)$$

Operator implikacije. Ako V_A je A tada V_B je B prevodi se u propoziciju V je E gde je V unija od V_A i V_B , i E je podskup osnovnog skupa V . Skup E se može definisati na više različitih načina.

2.9. Dedukcija u sistemu aproksimativnog rezonovanja

Dedukcija predstavlja fundamentalni mehanizam zaključivanja koji se koristi u teoriji aproksimativnog rezonovanja. Ovaj mehanizam zaključivanja biće zasnovan na upotrebi prethodno uvedenih operatora. Sada dajemo osnovna pravila zaključivanja.

Pravilo 1. (P1) Od propozicije V je A možemo izvoditi propoziciju V_B je B ako je V_A je $A \subset V_B$ je B . Ovo pravilo reflektuje očiglednu činjenicu: Ako vrednost date varijable leži u nekom skupu tada ta vrednost mora ležati i u većem skupu.

Pravilo 2 (P2). Od propozicija V_A je A i V_B je B možemo izvesti propoziciju:

$$(V_A \text{ je } A) \wedge (V_B \text{ je } B)$$

Proces dedukcije formira osnovni mehanizam u kome počevši sa skupom premisa i izvođenja dobijamo nove rezultate.

Definicija 2.31. Dedukcija iz skupa premisa (P_1, \dots, P_q) u sistemu aproksimativnog rezonovanja je niz dobro definisanih formula B_1, \dots, B_n , gde je svaka B_k jedna od sledećeg:

1. Premisa (P_1, \dots, P_q) ,
2. Tautologija,
3. Za neko $i < k$, B_k je posledica od B_i , kao rezultat pravila 1 (P1),
4. Za neko $i, j < k$, B_k je posledica od B_i i B_j , kao rezultat pravila 2 (PI2).

Teorema 3. (monotonost) . Ako je $(P_1, \dots, P_n) \vdash B$ tada je i $(P_1, \dots, P_{n+1}) \vdash B$. Dokaz sledi neposredno iz definicije dedukcije.

Teorema 4. Jedine teoreme u sistemu aproksimativnog rezonovanja su tautologije.

Sledeći rezultat prikazuje sve propozicije izvedive iz skupa propozicija.

Teorema 5. Propozicija B može da se izvede iz skupa premisa $(P_1, \dots, P_n) \vdash B$ ako i samo ako B sadrži konjunkciju svih premisa odnosno:

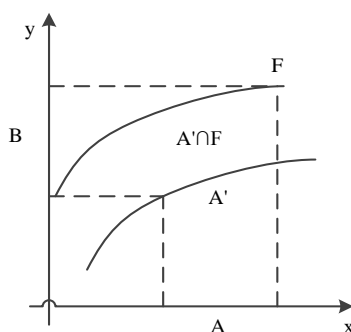
$$(P_1, \dots, P_n) \subset B.$$

Primedba. Implikacija ove teoreme može biti kao osnova za mehanizam rezonovanja u teoriji aproksimativnog rezonovanja.

2.10. Kompoziciono pravilo

Osnovno pravilo izvođenja u klasičnoj logici je *modus ponens* u kome ako je dati iskaz A istinit i ako je istinito da $A \Rightarrow B$, tada je i iskaz B istinit. U mnogim običnim rasuđivanjima, pravilo modus ponensa, izražava se ne u tačnoj već u približnoj formi. Tako npr. ako znamo da je A istinito i da $A^* \Rightarrow B$, gde je A^* u nekom smislu približno A , tada iz $A^* \Rightarrow B$ možemo zaključiti da je B približno istinito. Sada dajemo formalizaciju približnih rasuđivanja, na osnovu pojmova koji su uvedeni u prethodnom poglavlju. Za razliku od klasične logike, ovde predstavljamo *kompoziciono pravilo zaključivanja*, gde je modus ponens samo specijalan slučaj ovoga izvođenja [88,89].

Pretpostavimo da je A fazi skup definisa na osi x , a F fazi relacija na Dekartovom domenu $X \times Y$. Nađimo cilindrično proširenje A' sa osnovom A , a zatim njen presek sa fazi relacijom F kao na donjoj slici. Tako dobivamo $A \cap F$. Projektujmo zatim $A' \cap F$ na osu y . Na taj način dobivamo fazi podskupa B , na y osi.



Slika 7. Ilustracija kompozicionog pravila

Neka $\mu_A, \mu_{A'}, \mu_F$ i μ_B označavaju funkcije pripadnosti skupova A, A', F i B respektivno. Tada po definiciji cilindrično proširenje od A je:

$$\mu_{A'}(x,y) = \mu_A(x)$$

Oдавde je i :

$$\mu_{A \cap F} = \mu_{A'}(x,y) \wedge \mu_F(x,y)$$

Projektujmo skup $A' \cap F$ na y osu i tada imamo (na osnovu definicije projekcije):

$$\mu_B(y) = \mu_A(x) \wedge \mu_F(x,y).$$

Iz ovoga prozilazi da se skup B može definisati kao:

$$B = A \circ F$$

gde “ \circ ” definiše max-min operaciju kompozicije, tj:

$$\mu_B(y) = \max_x \min(\mu_A(x), \mu_F(x,y))$$

Kompoziciono pravilo. Neka su U i V dva univerzalna skupa sa datim varijablama u i v respektivno. Neka $R(u)$, $R(u,v)$ i $R(v)$ označavaju ograničenja na u , (u,v) i v i neka predstavljaju fazi relacije na $U, U \times V$ i V respektivno. Neka su A i F fazi podskupovi skupova U i $U \times V$. Tada kompoziciono pravilo tvrdi, da rešenje jednačine:

$$R(u) = A$$

$$R(u,v) = F$$

ima oblik:

$$R(v) = A \circ F$$

gde je $A \circ F$ kompozicija od A i F .

Primer 2.9. Neka je $U = V = \{1,2,3,4\}$ univerzalni skup. Neka je dat fazi skup $A = \text{malen} = \{1/1, 0.6/2, 0.2/3\}$ i neka je fazi relacija zadata sa:

$$F = \text{približno jednaki} = \{1/(1,1), 1/(2,2), 1/(3,3), 1/(4,4), 0.5/(1,2), 0.5/(2,1), 0.5/(2,3), 0.5/(3,2), 0.5/(3,4), 0.5/(4,3)\}$$

drugim rečima, neka je A unarna fazi relacija na U , nazvana malen, F binarna fazi relacija na $U \times V$, nazvana približno jednaki. Gornja jednačina u tom slučaju ima oblik:

$$R(u) = \text{malen}$$

$$R(u,v) = \text{približno jednaki}$$

i otuda:

$$R(v) = \text{malen} \circ \text{približno jednaki} = R(u) \circ R(u,v) = [1 \ 0.6 \ 0.5 \ 0.2]$$

2.11. Fazi implikacije

Neka su U i V univerzalni skupovi i neka su A, B i C fazi podskupovi od U, V i V . Kao prvo definišimo smisao iskaza *Ako x je A tada y je B inače y je C* , a zatim definišimo *Ako x je A tada y je B* kao specijalan slučaj iskaza *Ako x je A tada y je B inače y je C* .

Definicija 2.32. Iskaz *Ako x je A tada y je B inače y je C* je binarna fazi relacija na $U \times V$ definisana sa izrazom:

$$\text{Ako } x \text{ je } A \text{ tada } y \text{ je } B \text{ inače } y \text{ je } C = A \times B \cup \neg A \times C$$

Odnosno, ako su A , B i C unarne fazi relacije na U , V i V tada **Ako x je A tada y je B inače C** je binarna fazi relacija na $U \times V$, koja je unija Dekartovog proizvoda A i B i Dekartovog proizvoda negacije A i C .

Pošto iskaz **Ako x je A tada y je B inače y je C** možemo posmatrati kao specijalan slučaj iskaza **Ako x je A tada y je B inače y je C** , pri $C = V$ tada taj iskaz možemo definisati kao:

$$\text{Ako } x \text{ je } A \text{ tada } y \text{ je } B = A \times B \cup A \times V$$

Z a d e L. uvodi pravila (max-min pravilo i aritmetičko pravilo) za prevođenje fazi uslovne propozicije "Ako x je A tada y je B inače y je C " u fazi relaciju na $U \times V$. Označimo fazi skupove A , B i C :

$$A = \int_U \mu_A(u)/u \quad B = \int_V \mu_B(v)/v \quad C = \int_V \mu_C(v)/v$$

tada imamo:

1. Max- Min pravilo

$$R_m = (A \times B) \cup (\neg A \times C)$$

$$R_m = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v)) \vee (1 - \mu_A(u) \wedge \mu_C(v)) / (u, v)$$

2. Aritmetičko pravilo

$$R_a = (\neg A \times V \oplus U \times B) \cap (A \times V \oplus U \times C)$$

$$R_a = \int_{U \times V} 1 \wedge (1 - \mu_A(u) + \mu_B(v)) \wedge (\mu_A(u) + \mu_C(v)) / (u, v)$$

gde \times , \cup , \cap , \neg i \oplus označavaju Dekartov proizvod, uniju, presek, komplement i graničnu sumu za fazi skupove.

Primedba. Ako se C zameni sa V (univerzalnim skupom od C) koji je interpretiran kao 'nepoznat' tada fazi uslovna propozicija "Ako x je A tada y je B inače y je C " je redukovana u propoziciju "Ako x je A tada y je B inače y je nepoznato" odnosno u "Ako x je A tada y je B ". Tada max - min pravilo R_m i aritmetičko pravilo R_a postaje, za $C = V$:

$$R_m = (A \times B) \cup (\neg A \times V)$$

$$R_m = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v)) \vee (1 - \mu_A(u) \wedge \mu_C(v)) / (u, v)$$

odnosno:

$$R_a = (\neg A \times V \oplus U \times B)$$

$$R_a = \int_{U \times V} 1 \wedge (1 - \mu_A(u) + \mu_B(v)) / (u, v)$$

U slučaju kada je $C =$ prazan skup max - min pravilo je redukovano sa min pravilom R_c koje je uveo Mamdani [86].

$$R_c = A \times B$$

$$R_c = \int_{U \times V} (\mu_A(u) \wedge \mu_B(v)) / (u, v)$$

Za propoziciju “Ako x je A tada y je B inače y je C “, takođe, se definiše translatorno pravilo R_b koje je zasnovano na implikaciji u binarnoj logici:

3. Fazi binarno pravilo R_b

$$R_b = (\neg A \times V \cup U \times B) \cap (A \times V \cup U \times C)$$

$$R_b = \int_{U \times V} (1 - \mu_A(u) \vee \mu_B(v)) \wedge (\mu_A(u) \wedge \mu_C(v)) / (u, v)$$

U slučaju da je C =prazan skup imamo:

$$R_b = (\neg A \times V) \cup (U \times B)$$

$$R_b = \int_{U \times V} (1 - \mu_A(u) \vee \mu_B(v)) / (u, v)$$

Mizumoto [88] uvodi sledeće fazi implikacije:

1. Pravilo R_s :

$$R_s = A \times V \Rightarrow_s U \times B$$

$$R_s = \int_{U \times V} (\mu_A(u) \rightarrow_s \mu_B(v)) / (u, v)$$

gde je:

$$\mu_A(u) \rightarrow_s \mu_B(v) = \begin{cases} 1, & \mu_A(u) \leq \mu_B(v) \\ 0, & \mu_A(u) > \mu_B(v) \end{cases}$$

2. Pravilo R_g :

$$R_g = A \times V \Rightarrow_g U \times B$$

$$R_g = \int_{U \times V} (\mu_A(u) \rightarrow_g \mu_B(v)) / (u, v)$$

gde je:

$$\mu_A(u) \rightarrow_g \mu_B(v) = \begin{cases} 1, & \mu_A(u) \leq \mu_B(v) \\ \mu_B(v), & \mu_A(u) > \mu_B(v) \end{cases}$$

2.12. Fazi metod zaključivanja

Razmatrimo sledeću formu zaključivanja:

Premisa 1: Ako x je A tada y je B

Premisa 2: x je A'

Zaključak: y je B'

gde su x i y imena objekata, a A , A' , B i B' su fazi koncepti određeni sa fazi skupovima na univerzalnim skupovima U , U , V i V respektivno. Ova forma zaključivanja može se posmatrati kao **generalizovani (fazi) modus ponens** u odnosu na **klasični modus ponens** kada je $A'=A$ i $B'=B$. Validnost modus ponensa nam govori da zaključak mora biti istinit ukoliko su sve premise istinite.

Sada posmatrajmo sledeću formu zaključivanja:

Premisa 1: Ako x je A tada y je B

Premisa 2: y je B'

Zaključak: x je A'

Ovaj zaključak može biti uveden kao **generalizovan (fazi) modus tollens**, u odnosu na klasični modus tollens kada je $B' = \text{ne } B$ i $A' = \text{ne } A$. U logici, **modus tollens** je formalni naziv za validan **indirektan dokaz** ili **dokaz kontrapozicijom**.

Generalizovani modus ponens i modus tollens se lako primenjuju i u fazi situacijama. Uočimo to na sledećem primeru:

Ako je ova jabuka vrlo crvena

I crvena jabuka je zrela jabuka

Onda je ova jabuka vrlo zrela.

U generalizovanom modus ponensu, konsekvenca B' u zaključku može biti dobivena iz Premise 1 i Premise 2 koristeći max-min (\circ) kompoziciju između fazi skupa A' i fazi relacija datih prethodno. Tako za metod R_m imamo:

$$B_m = A' \circ R_m = A' \circ [(A \times B) \cup (\neg A \times V)]$$

Funkcija pripadnosti fazi skupa B_m u univerzalnom skupu V je:

$$\mu_{B_m} = \vee_u \{ \mu_{A'}(u) \wedge \mu_{R_m}(u, v) \}$$

ili:

$$\mu_{B_m} = \vee_u \{ \mu_{A'}(u) \wedge [(\mu_A(u) \wedge \mu_B(v)) \vee (1 - \mu_A(u))] \}$$

2.13. Zadovoljivost u fazi logici

Definicija 2.32. Za fazi formulu $f \in S$, gde je S skup fazi formula, kažemo da je zadovoljiva ako postoji interpretacija I u kojoj je $T(f) \geq 0.5$. U interpretaciji I kažemo da je f protivrečna ako je $T(f) < 0.5$. Za fazi formulu f kažemo da je nezadovoljiva ako je protivrečna u svim njenim interpretacijama.

Definicija 2.33. Fazi formulu f zovemo konjunktivnom normalnom formom ako je $f = C_1 \wedge \dots \wedge C_p$, $p \geq 1$, gde je C_j disjunkt. Fazi formulu f zovemo disjunktivnom normalnom formom ako je $f = C_1 \vee \dots \vee C_p$, $p \geq 1$, gde je C_j konjunkt

Definicija 2.34. Za formulu g kažemo da je logička konsekvenca date formule f ako i samo ako je $f \wedge \neg g$ nezadovoljivo tj. $T(f \wedge \neg g) < 0.5$.

2.14. Rezolventa u fazi logici

Definicija 2.35. Neka su $D_1: L_1 \vee D_1'$ i $D_2: L_2 \vee D_2'$ dva disjunkta, a L_1 i L_2 , kontrarni (suprotni) par literala tj. $L_2: \neg L_1$: neka D_1' i D_2' ne sadrže ni jedan takav par. Tada disjunkt $D_1' \vee D_2'$ zovemo rezolventom (Res) disjunkta D_1 i D_2 sa ključnom reči L_1 .

Primer 2.9. Neka su $D_1: \neg P \vee Q$ i $D_2: P$ dva disjunkta, i neka je 'i' njihova interpretacija. Neka je ispunjeno $i(P) = 0.3$, $i(Q) = 0.2$. Tada je:

$$i(D_1 \wedge D_2) = \min[\max[i(\neg P), i(Q)], i(P)] = \min[\max[0.7, 0.2], 0.3] = \min [0.7, 0.3] = 0.3$$

Q je rezolventa od D_1 i D_2 , pa imamo da je tada $i(\text{Res}(Q)) = 0.2$. Vidimo da je u datom primeru da je $i(Q) < i(D_1 \wedge D_2)$.

Kriterijum koji osigurava da primena pravila rezolucije, u kontekstu fazi logike, kao rezultat daje valjane logičke posledice dat je sledećom teoremom.

Teorema 6. Neka su D_1 i D_2 dva disjunktka i neka je $\text{Res}(D, D')$ njihova rezolventa s literarom L kao ključnom reči. Tada vredi:

a) Ako je $i(L \wedge \neg L) \leq i(\text{Res}(D_1, D_2))$ tada je $i(D_1 \wedge D_2) \leq i(\text{Res}(D_1, D_2))$

b) Ako je $i(L \wedge \neg L) \geq i(\text{Res}(D_1, D_2))$ tada je $i(D_1 \wedge D_2) \geq i(\text{Res}(D_1, D_2))$

Dokaz : Neka su $D_1: L \vee D_1'$ i $D_2: \neg L \vee D_2'$ dva disjunktka i pretpostavimo da ni u D_1' niti u D_2' ne nastupa ni jedan kontrarni par literala. Sud $(D_1 \wedge D_2)$ i rezolventa $\text{Res}(D_1, D_2)$ tada imaju prikaz:

$$D_1 \wedge D_2: (L_1 \vee D_1') \wedge (\neg L \vee D_2') : (L \wedge \neg L) \vee (L \wedge D_2') \vee (\neg L \wedge D_1') \vee (D_1' \wedge D_2')$$

Odnosno $\text{Res}(D_1, D_2) = D_1' \vee D_2'$. Pretpostavimo da je i interpretacija za koju vredi 1), to jest:

$$i(L \wedge \neg L) \leq i(D_1' \vee D_2') \quad (1)$$

Naredne tri nejednakosti lako je proveriti koristeći definiciju interpretacije fazi skupova i svojstva operacija min i max.

$$i(L \wedge D_2') \leq i(D_1' \vee D_2') \quad (2)$$

$$i(L \wedge D_1') \leq i(D_1' \vee D_2') \quad (3)$$

$$i(D_1' \wedge D_2') \leq i(D_1' \vee D_2') \quad (4)$$

Ilustracije radi proverimo samo prvu. Tada je:

$$i(L \wedge D_2') = \min \{i(L), i(D_2')\} \quad (*)$$

$$i(D_1' \vee D_2') = \max \{i(D_1'), i(D_2')\}$$

Ako je minimum desne strane u (*) jednak $i(L)$, onda je $i(L \wedge D_2') = i(L) \leq i(D_2') \leq \max\{i(D_1'), i(D_2')\}$. Time je tvrdnja 2) ispunjena. Ako je taj minimum jednak $i(D_2')$, nejednakost 2) vredi tim prije.

Uzimanjem maksimuma levih i desnih strana nejednakosti (1) do (4) dobivamo sledeću nejednakost:

$$\max\{i(L \wedge \neg L), i(L \wedge D_2'), i(\neg L \wedge D_1')\} \leq i(D_1' \vee D_2') (**)$$

Činjenica da je leva strana u (**) upravo $i(D_1 \wedge D_2)$, a desna rezolventa $\text{Res}(D_1, D_2)$ potvrđuje valjanost tvrdnje 1) datog teorema.

Ako je sada ispunjena pretpostavka iz tačke 2) teorema, tj.

$i(\text{Res}(D_1, D_2)) \leq i(L \wedge \neg L)$, tada je tim prije na snazi nejednakost:

$$i(\text{Res}(D_1, D_2)) \leq \max\{i(L \wedge \neg L), i(L \wedge D_2'), i(\neg L \wedge D_1'), i(D_1' \vee D_2')\}$$

odnosno:

$$i(\text{Res}(D_1, D_2)) \leq i(D_1 \wedge D_2), \text{ a što je i trebalo dokazati.}$$

Naredne dve teoreme su neposredna posledice date teoreme.

Teorema 7. Neka su D_1 i D_2 dva disjunkta i neka je $\text{Res}(D_1, D_2)$ njihova rezolventa s literarom L kao ključnom reči. Ako vredi $i(\text{Res}(D_1, D_2)) \geq 0.5$ tada vredi $i(D_1 \wedge D_2) \leq i(\text{Res}(D_1, D_2))$.

Dokaz : Očigledno je da za svaki literal L vredi $i(L \wedge \neg L) \leq 0.5$. Kako je iz uslova teoreme je $i(\text{Res}(D_1, D_2)) \geq 0.5$, tada i vredi da je $i(L \wedge \neg L) \leq i(\text{Res}(D_1, D_2))$. Prema tački 1) prethodne teoreme tada je $i(D_1 \wedge D_2) \leq i(\text{Res}(D_1, D_2))$.

Iz ove teoreme zaključujemo da je svaka rezolventa dvaju disjunkta sa vrednošću interpretacije ne manjom od 0.5 smisljena je logička posledica tih disjunkta.

Teorema 8. Neka su D_1 i D_2 dva disjunkta i neka je $i(D_1 \wedge D_2) > 0.5$. Tada je $i(D_1 \wedge D_2) < i(\text{Res}(D_1, D_2))$.

Dokaz. Zbog

$$i(D_1 \wedge D_2) = \max\{i(L \wedge \neg L), i(L \wedge D_2'), i(\neg L \wedge D_1'), i(D_1' \wedge D_2')\}$$

i činjenice da je $i(L \wedge \neg L) \leq 0.5$ i pretpostavke teoreme, tada vredi bar jedna od narednih nejednakosti:

- a) $i(L \wedge D_2') > 0.5$
- b) $i(\neg L \wedge D_1') > 0.5$
- c) $i(D_1' \wedge D_2') > 0.5$

Otuda je $i(D_1') > 0.5$ ili je $i(D_2') > 0.5$. Samim tim je i

$$i(D_1' \vee D_2') = i(\text{Res}(D_1, D_2)) > 0.5.$$

U suprotnom da je $i(\text{Res}(D_1, D_2)) \leq i(D_1 \vee D_2)$ tada bi bila ispunjena bar jedna od tri naredne nejednakosti:

- a) $\max\{i(D_1'), i(D_2')\} < \min\{i(L), i(D_2')\}$
- b) $\max\{i(D_1'), i(D_2')\} < \min\{i(\neg L), i(D_1')\}$
- c) $\max\{i(D_1'), i(D_2')\} < \min\{i(D_1'), i(D_2')\}$

što je očigledno nemoguće. Dakle vredi tvrdnja teoreme.

Gornja teorem pokazuje da je za smislenost rezolvente kao logičke posledice dvaju disjunkta dovoljno da istinita vrednost svake interpretacije njihove konjunkcije bude veća od 0.5.

Definicija 2.36. Neka je D skup disjunkta. Indukcijom po $n \in \mathbb{N}$ definišemo n -ti rezolventni skup $\text{Res}^n D$ na sledeći način:

$$\text{Res}^n D = \begin{cases} D & n = 0 \\ \text{Res}(\text{Res}^{n-1}(D)), & \text{u suprotnom.} \end{cases}$$

Teorema 9. Neka je $D = \{D_1, \dots, D_n\}$ skup disjunkta i neka je $D^* \in \text{Res}^n(D)$. Ako je $i(D) \geq 0.5$ za svaki disjunkt D iz D tada vredi:

$$i(D_1 \wedge \dots \wedge D_n) \leq i(D^*)$$

Dokaz. Ovo ćemo dokazati indukcijom po n . Neka je $n = 1$, tada $D^* \in \text{Res}^1(D) = \text{Res}(D)$. Bez gubitka uopštenosti, pretpostavimo da je $D^* \in \text{Res}(D) - D$. U tom slučaju postoje disjunkt D_1 i D_2 takvi da je $D : \text{Res}(D_1, D_2)$.

Po pretpostavci teoreme vredi $i(D_1 \wedge D_2) > 0.5$. Sada na osnovu prethodne teoreme vredi sledeći niz nejednakosti:

$$0.5 < i(D_1 \wedge \dots \wedge D_n) < i(D_1 \wedge D_2) < i(\text{Res}(D_1, D_2)) = i(D^*).$$

Sada pretpostavimo da je D^* element skupa $\text{Res}^i(D) - \text{Res}^{i-1}(D)$. Tada postoje disjunkti D_i' i D_{i-1}'' iz skupa $\text{Res}^{i-1}(D)$, takvi da vredi:

$$D^*: \text{Res}(D_{i-1}' \wedge D_{i-1}'').$$

Po pretpostavci indukcije vredi:

$$0.5 < i(D_1 \wedge \dots \wedge D_n) \leq i(D_{i-1}') \text{ i } 0.5 < i(D_1 \wedge \dots \wedge D_n) \leq i(D_{i-1}'')$$

Sada je:

$$0.5 < i(D_1 \wedge \dots \wedge D_n) \leq \min\{i(D_{i-1}'), i(D_{i-1}'')\} = \{i(D_{i-1}') \wedge i(D_{i-1}'')\} \leq i(\text{Res}(D_{i-1}', D_{i-1}'')) = i(D^*).$$

Ova poslednja nejednakost je posledica prethodne teoreme.

Ova teorema tvrdi da je dovoljno da minimalna vrednost istinitosti disjunkta iz datog skupa bude veća od 0.5 pa da svi rezolventni skupovi tog skupa sadrže samo valjane logičke posledice.

2.15. Zaključak

U ovom poglavlju navedeni su osnovni pojmovi kao i već poznati rezultati iz teorije fazi skupova. Kao prvo, uvedena je definicija fazi skupa, a zatim su date operacije s kojima se operiše nad fazi skupovima.

Zatim su uvedeni i razmatrani rezultati iz teorije fazi logike. Najpre, uvedena je teorija aproksimativnog rezonovanja, zatim su definisani primarni elementi u sistemu aproksimativnog rezonovanja i data su kompoziciona pravila zaključivanja. Na kraju je definisana rezolventa i dato je pravilo rezolucije u fazi logici. Sve navedene osobine su analizirane radi razmatranja i analize fazi relacionih modela baza podataka.

Fazi logika i fazi sistemi se koriste za procenu i donošenje odluka u velikom broju upravljačkih sistema. U ovom radu rezultati fazi logike primenjeni su na konkretnom primeru relacionih baza podataka. Primena teorije fazi skupova u bazama podataka pruža mogućnost upravljanja i manipulisanja nepreciznim i nepotpunim informacijama. Praktično, danas se fazi logika koristi u bezbroj aplikacija, a fazi relacioni modeli baza podataka se primenjuju i čine osnovu sistema zasnovanih na znanju, sistema za pretragu znanja itd. Takođe, fazi modeli baza podataka se koriste na višem, praktičnom nivou, kao što je proces donošenja odluka, čime se doprinosi optimizaciji poslovanja i pruža značajna podrška poslovnim aktivnostima.

U protekloj deceniji, fazi sistemi su u velikoj meri postali zamena konvencionalnim tehnologijama u velikom broju naučnih aplikacija i informacionih sistema, naročito u oblasti upravljanja sistemima i prepoznavanju oblika. Teorija fazi skupova se široko primenjuje za proširenje različitih modela baza podataka i primena ove teorije rezultirala je brojnim doprinosima, uglavnom u relacionim modelima baza podataka. Da bi se zadovoljile potrebe za modelovanjem nepreciznih i neodređenih informacija, u poslednjoj deceniji mnoga istraživanja su koncentrisana na fazi relacione i fazi objektno-orjentisane modele baza podataka.

3. Fazi relacione baze podataka

3.1. Relacioni model baza podataka

Sistem baza podataka [90-92] može se posmatrati kao hijerarhijska apstrakcija. Svaki od nivoa u datoj hijerarhiji predstavlja jednu vrstu modela, što znači da predstavlja skup *objekata (entiteta)* i operacija nad tim objektima. Na nivou neposredno ispod modela nalazi se reprezentacija modela predstavljena pomoću *strukture podataka*, a operacija modela predstavlja algoritme koji operišu nad tim strukturama.

3.1.1. Modeli, atributi, entiteti

Pod modelom se podrazumeva reprezentacija nekih objekata i odnosa među njima. Objekte modela i odnose između njih zovemo *entitetima*.

Skup E koji se sastoji od entiteta istoga tipa okarakterisan je skupom atributa A_1, \dots, A_n ili kraće $E(A_1, \dots, A_n)$ gde je $A_i: E \rightarrow D_i$ funkcija čiji se kodomen naziva domenom atributa A_i . Za e iz E , $A_i(e)$ iz D_i se zove vrednost atributa A_i za entitet e .

3.1.2. Predstavljanje skupa entiteta skupom n-torki

Za dati tip entiteta $E(A_1, \dots, A_n)$ skup funkcija $A_1: E \rightarrow D_1, \dots, A_n: E \rightarrow D_n$ koje predstavljaju attribute tog tipa entiteta, određuje jednoznačno funkciju:

$$(A_1, \dots, A_n): E \rightarrow D_1 \times \dots \times D_n$$

gde je $D_1 \times \dots \times D_n$ skup svih n-torki oblika (d_1, \dots, d_n) , $d_i \in D_i$ za $i = 1, 2, \dots, n$, a $D_1 \times \dots \times D_n$ je Dekartov proizvod skupova D_1, \dots, D_n .

Funkcija (A_1, \dots, A_n) definisana je kao:

$$(A_1, \dots, A_n)(e) = (A_1(e), \dots, A_n(e))$$

pri čemu zahtevamo da za dva entiteta e_1 i e_2 istoga tipa E važi:

$$(A_1, \dots, A_n)(e_1) \neq (A_1, \dots, A_n)(e_2)$$

odnosno da postoji neko $j = 1, \dots, n$ takvo da je $A_j(e_1) \neq A_j(e_2)$ ili drugim rečima, entitete istoga tipa razlikujemo po tome što imaju različite vrednosti skupa atributa.

Skup entiteta tipa $E(A_1, \dots, A_n)$ predstavlja se tabelom čije kolone odgovaraju atributima, a redovi pojedinim n - torkama. Ovakva tabela predstavlja relaciju koja se definiše na sledeći način:

Definicija 3.1. R je relacija na skupovima D_1, \dots, D_n , $R \subseteq D_1 \times \dots \times D_n$. *Relaciona baza podataka* je skup relacija koje se mogu menjati tokom vremena. Opis strukture relacije relacione baze se zadaje *relacionom šemom*.

Primer relacione šeme:

Fakultet(Sifra_Fakulteta, Naziv, Grad)

Student(Broj_Indexa, Ime_Prezime, Godina, Sifra_Fakulteta)

Projekat(Projekat, Naziv, Sredstva)

Ucesce(Broj_Indexa, Projekat, Funkcija)

Ova relaciona šema opisuje 4 tipa entiteta: Fakultet, Student, Projekat i Ucesce studenta u projektima. Primer relacione baze za datu relacionu šemu i za mogući vremenski trenutak.

Sifra_Fakulteta	Naziv	Grad
F1	Mašinski	Beograd
F2	Elektrotehnički	Novi Sad
F3	Građevinski	Beograd

Tabela 2. Fakultet

Broj_Indexa	Ime_Prezime	Godina	Sifra_Fakulteta
S1	Mihajlo Ivanović	1.	F1
S2	Miloš Milutinović	2.	F2
S3	Petar Petrović	3.	F1
S4	Marko Marković	1.	F2
S5	Jelene Petrović	4.	F3
S6	Nikola Milošević	4.	F3
S7	Maja Martinović	1.	F1

Tabela 3. Student

Projekat	Naziv	Sredstva
P1	N1	1000
P2	N2	2000
P3	N3	3000

Tabela 4. Projekat

Broj_Indexa	Projekat	Funkcija
S1	P1	Izvršilac
S2	P3	Izvršilac
S2	P2	Rukovodilac
S3	P2	Izvršilac
S3	P3	Izvršilac
S4	P1	Izvršilac
S5	P3	Rukovodilac
S6	P1	Izvršilac
S6	P2	Izvršilac
S6	P3	Izvršilac
S7	P1	Rukovodilac

Tabela 5. Učešće

3.1.3. Relaciona algebra

Sastavni dio modela čini i jezik sa kojim se izražavaju upiti i zadaje ažuriranja baze podataka koja odgovara tom modelu. Osnovu za relacione jezike predstavlja jedan jedinstven formalni jezik koji se naziva relaciona algebra. Ona sadrži niz operacija nad relacijama. Osim poznatih operacija unije, preseka, razlike i Dekartovog proizvoda

relaciona algebra uključuje i neke specifične operacije koje se nazivaju projekcija, restrikcija, spajanje i deljenje.

Projekcija. Neka je $R(A_1, \dots, A_n)$ relacija, a X podskup skupa atributa $\{A_1, \dots, A_n\}$. Neka je Y komplement od X na skupu atributa $\{A_1, \dots, A_n\}$, označan sa $\{A_1, \dots, A_n\} - X$. Tada relaciju R možemo permutovanjem njenih atributa predstaviti kao $R(X, Y)$. Rezultat operacije projekcije relacije R po atributima X je:

$$R[X] = \{x \mid \text{postoji } y \text{ takvo da je } (x, y) \in R(X, Y)\}.$$

Primer projekcije. Naći projekciju za **Student**(Godina, Sifra_Fakulteta) iz prethodne relacione šeme **Student**(Broj_Indexa, Ime_Prezime, Godina, Sifra_Fakulteta).

Godina	Sifra_Fakulteta
1.	F1
2.	F2
3.	F1
1.	F2
4.	F3

Tabela 6. Projekcija

Restrikcija. Neka je $R(A_1, \dots, A_n)$ relacija na skupu atributa $\{A_1, \dots, A_n\}$, a P relacija na skupu $D_1 \times \dots \times D_n$, gde je D_i domen atributa A_i ($i = 1, \dots, n$). Rezultat restrikcije relacije R po uslovu P je:

$$R[P] = \{x \mid x \in R \text{ i } P(x)\}.$$

Primer restrikcije. Izdvojiti sve studente 1. godine fakulteta iz relacione šeme **Student**(Broj_Indexa, Ime_Prezime, Godina, Sifra_Fakulteta).

Broj indexa	Ime_Prezime	Godina	Sifra_Fakulteta
S1	Mihajlo Ivanović	1.	F1
S4	Marko Marković	1.	F2
S7	Maja Martinović	1.	F1

Tabela 7. Restrikcija

Spajanje. Neka su $A(A_1, \dots, A_n)$ i $B(B_1, \dots, B_m)$ relacije, a $X \subseteq (A_1, \dots, A_n)$ i $Y \subseteq (B_1, \dots, B_m)$ skupovi jednakog broja atributa i sa jednakim korespondentnim domenima. Neka je Z komplement od X na skupu atributa $\{A_1, \dots, A_n\}$ u oznaci $\{A_1, \dots, A_n\} - X$ i W komplement od Y na skupu atributa $\{B_1, \dots, B_m\}$, označano sa $\{B_1, \dots, B_m\} - Y$. Tada pogodnim permutovanjem atributa relacije A i B možemo napisati kao $A(Z, X)$ i $B(Y, W)$. Rezultat prirodnog spajanja relacije A i B po atributima X i Y definisan je kao:

$$A[X=Y]B = \{(z, x, w) \mid (z, x) \in A, (y, w) \in B \text{ i } x = y\}$$

Tabelarna interpretacija operacije prirodnog spajanja je formiranje tabele čiji se redovi dobivaju dopisivanjem m -torke iz relacije B , n - torke iz relacije A kada god su vrednosti atributa Y iz m -torke jednake vrednostima X atributa iz n -torke. Kod toga se još projekcijom po atributima iz X i Y eliminišu nepotrebna ponavljanja jednakih vrednosti atributa spajanja.

Primer spajanja. Izvršiti operaciju spajanja tabela **Fakultet**(Sifra_Fakulteta, Naziv, Grad) i **Student**(Broj_Indexa, Ime_Prezime, Godina, Sifra_Fakultet), čiji rezultat daje tabelu koja daje pregled imena studenata i gradova u kojima se nalaze ti fakulteti.

Ime_Prezime	Grad
Mihajlo Ivanović	Beograd
Miloš Milutinović	Novi Sad
Petar Petrović	Beograd
Marko Marković	Novi Sad
Jelena Petrović	Beograd
Nikola Milošević	Beograd
Maja Martinović	Beograd

Tabela 8. Spajanje

Deljenje. Neka su $A(X,Y)$ i $B(Z)$ relacije gde su X , Y i Z skupovi atributa i gde Y i Z predstavljaju isti broj atributa i neka su im domeni korespondentnih atributa jednaki. Rezultat deljenja relacije $A(X,Y)$ sa $B(Z)$ u oznaci $A[Y\#Z]B$ definiše se kao maksimalni podskup projekcije $A[X]$ takav da je njegov proizvod sa $B(Z)$ sadržan u $A(X,Y)$.

Drugim rečima važi:

$$A(X, Y) = A[Y\#Z]B \times B(Z) \cup \Theta(X, Y)$$

gde je $A[Y\#Z]B$ količnik deljenja, a $\Theta(X, Y)$ ostatak deljenja.

Primer deljenja: **Projekat**(Projekat, Naziv, Sredstva)

Učešće(Broj_indexa, Projekat, Funkcija)

Učešće[Projekat#Projekat]**Projekat**[Projekat] predstavlja rezultat deljenja relacije **Učešće** sa projekcijom relacije **Projekat** po atributu Projekat. Količnik deljenja je tabela studenata koji učestvuju na svim projektima u istoj funkciji i njihovih funkcija u okviru tih projekata.

Broj_Indexa	Funkcija
S6	Izvršilac

Tabela 9. Deljenje

Ostatak deljenja je tabela:

Broj_Indexa	Projekat	Funkcija
S1	P1	Izvršilac
S2	P3	Izvršilac
S2	P2	Rukovodilac
S3	P2	Izvršilac
S3	P3	Izvršilac
S4	P1	Izvršilac
S5	P3	Rukovodilac
S7	P1	Rukovodilac

Tabela 10. Ostatak deljenja

3.1.4. Funkcionalne zavisnosti

Posmatrajmo relaciju Studenti(Ime, Adresa, Grad, Poštanski_Broj, Telefon)

Pretpostavimo da su imena svih studenata različita i da svaki od njih može imati samo jednu adresu i telefon.

Logičke veze između atributa relacije Studenti može se opisati:

- Svako ime određuje jednoznačno Adresu, Grad, Poštanski_Broj i Telefon.
- Adresa i Grad određuju jedinstven Poštanski_Broj
- Svakom Poštanskom_Broju pridružen je jednoznačno Grad kojem taj broj pripada.

Ovako opisane logičke veze između atributa u prethodnoj relaciji nazivaju se funkcionalnim zavisnostima. Ovaj se pojam definiše na sledeći način:

Definicija 3.2. Neka je $R(A_1, \dots, A_n)$ relacija, a X i Y podskupovi skupa svih atributa relacije R tj. $X \subseteq (A_1, \dots, A_n)$ i $Y \subseteq (A_1, \dots, A_n)$. Sa $R[X, Y]$ označimo projekciju relacije R po atributima iz X i Y . Kažemo da postoji funkcionalna zavisnost $X \rightarrow Y$ ako i samo ako je za svaki trenutak vremena $R[X, Y]$ funkcija $R[X] \rightarrow R[Y]$.

Navedena definicija označava da kad god su parovi (x, y) i (x, y') elementi relacije $R[XY]$ tada je $y = y'$. Ovo je upravo uslov koji razlikuje funkciju od relacije. Ako funkcionalna zavisnost $X \rightarrow Y$ ne postoji onda relacija $R[X, Y]$ može sadržavati više elemenata koji imaju istu vrednost atributa iz X , a različitu vrednost atributa iz Y . Drugo, $X \rightarrow Y$ je vremenski nepromenljivo svojstvo. Skup n - torki koje opisuju $R(A_1, \dots, A_n)$ se menja u vremenu, a to onda važi i za $R[X, Y]$. Definicija funkcionalne zavisnosti zahteva da su te promene takve da u svakom trenutku vremena $R[X, Y]$ predstavlja ne samo relaciju nego funkciju $R[X] \rightarrow R[Y]$,

Primer funkcionalnih zavisnosti. Za datu relaciju Studenti uz uvedene pretpostavke slede funkcionalne zavisnosti: $\text{Ime} \rightarrow \text{Adresa}$, $\text{Ime} \rightarrow \text{Grad}$, $\text{Ime} \rightarrow \text{Poštanski_Broj}$, $\text{Ime} \rightarrow \text{Telefon}$, $\text{Adresa, Grad} \rightarrow \text{Poštanski_Broj}$, $\text{Poštanski_Broj} \rightarrow \text{Grad}$. Definicija funkcionalne zavisnosti omogućava preciznu definiciju pojma ključa.

Neka je $R(A_1, \dots, A_n)$ relacija, a X podskup skupa atributa relacije R . X je ključ relacije R ako i samo ako važi:

- X funkcionalno određuje sve attribute relacije R tj. $X \rightarrow A_i$, za $i = 1, \dots, n$.
- Nijedan pravi podskup od X nema tu osobinu tj. za $X' \subset X$ ne važi $X' \rightarrow A_j$, za neki $j = 1, \dots, n$.

Na primer, atribut Ime je ključ relacije Studenti, jer određuje sve attribute te relacije.

Ako za X važi samo osobina a) on se naziva *nadključem*, što znači da sadrži ključ. Iz skupa ključeva svake relacije bira se jedan koji preuzima ulogu jedinstvenog identifikatora entiteta predstavljenih relacijom. Taj ključ se zove *primarnim ključem*. On ima ulogu da zahteva da nijedan od njegovih atributa ne sme nikada poprimiti nedefinisano (nepoznatu) vrednost. [60]

Ponašanje funkcionalnih zavisnosti određeno je skupom osnovnih pravila iz kojih se izvode sve druge osobine ove vrste logičkih veza.

- a) Jedinstvenost. Ako su $f: X \rightarrow Y$ i $g: X \rightarrow Y$ funkcionalne zavisnosti tada je $f \equiv g$ tj. postoji samo jedna funkcionalna zavisnost sa datim domenom i kodomenom.
- b) Projektivnost. Ako je $X \subseteq Y$ tada postoji funkcionalna zavisnost $Y \rightarrow X$ tj. svaki skup atributa funkcionalno određuje svaki svoj podskup. (Trivijalna zavisnost)
- c) Tranzitivnost. Ako su $f: X \rightarrow Y$ i $g: Y \rightarrow Z$ funkcionalne zavisnosti tada postoji funkcionalna zavisnost $X \rightarrow Z$.
- d) Aditivnost. Ako su $X \rightarrow Y$ i $X \rightarrow Z$ funkcionalne zavisnosti tada postoji i funkcionalna zavisnost $X \rightarrow YZ$.

Na primer, za relacije Studenti važe na osnovu prethodnih osobina, sledeće funkcionalne zavisnosti:

Adresa, Grad \rightarrow Grad, Adresa, Grad \rightarrow Adresa (Projektivnost)

Iz funkcionalnih zavisnosti na osnovu osobine tranzitivnosti imamo:

Adresa, Grad \rightarrow Poštanski_Broj i Poštanski_Broj \rightarrow Grad dobivamo Adresa, Grad \rightarrow Grad.

Iz funkcionalnih zavisnosti na osnovu osobine aditivnosti imamo:

Ime \rightarrow Adresa, Ime \rightarrow Telefon izvodimo Ime \rightarrow Adresa, Telefon

Iz osnovnih osobina funkcionalnih zavisnosti mogu se izvesti sledeće osobine:

- a) Distributivnost. Ako važi funkcionalna zavisnost $X \rightarrow YZ$ tada važe i funkcionalne zavisnosti $X \rightarrow Y$ i $X \rightarrow Z$.
- b) Pseudotranzitivnost. Ako važi funkcionalne zavisnosti $X \rightarrow Y$, $YW \rightarrow Z$ tada važi i funkcionalna zavisnost $XW \rightarrow Z$.
- c) Proširenje. Ako važi funkcionalna zavisnost $X \rightarrow Y$, $W \rightarrow Z$ tada važi i funkcionalna zavisnost $XW \rightarrow YZ$.

3.1.5. Višeznačne zavisnosti

Funkcionalne zavisnosti predstavljaju specijalan slučaj opštije logičke veze između atributa relacije, koja se naziva višeznačna zavisnost. Višeznačna zavisnost je potpuna veza dva seta atributa u relaciji. Višeznačna zavisnost među atributima neke relacije \mathbf{R} (X, Y, Z) postoji onda kada skup vrednosti atributa Y zavisi od X, a ne zavisi od Z. Rečima iskazano: X višeznačno određuje Y, odnosno Y višeznačno zavisi od X. Ovu vrstu zavisnosti razmotrimo prvo na jednom primeru. Neka je data relacija **Profesor** (Ispit, Ime, Sat, Učionica, Student, Ocena). U ovome primeru postoji samo jedan ispit sa dva studenta. Ispit se može održavati u nekoliko različitih soba i u različito vreme.

Ispit	Ime	Sat	Učionica	Student	Ocena
Baze podataka	M.V.	9:00	1	Marko M.	8
Baze podataka	M.V.	10:00	2	Marko M.	8
Baze podataka	M.V.	11:00	1	Marko M.	8
Baze podataka	M.V.	9:00	1	Jelena J.	9
Baze podataka	M.V.	10:00	2	Jelena J.	9
Baze podataka	M.V.	11:00	1	Jelena J.	9

Tabela 11. Višeznačne zavisnosti

Neka je $R(X,Y,Z)$ relacija sa $m + n + r$ atributa, gde su $X = \{X_1, \dots, X_m\}$, $Y = \{Y_1, \dots, Y_n\}$, $Z = \{Z_1, \dots, Z_r\}$ međusobno disjunktni skupovi atributa, a $m -$ torke (x_1, \dots, x_m) konkretnih vrednosti atributa X_1, \dots, X_m , neke $m + n + r -$ torke relacije R označimo sa x . Analogno važi za y i z . Definišimo skup Y_{xz} na sledeći način:

$$Y_{xz} = \{y \mid (x, y, z) \in R\}$$

Dakle, skup Y_{xz} dobije se tako što izvršimo operaciju restrikcije relacije R birajući samo one $m + n + r -$ torke čije su X komponente jednake x , a Z komponente jednake z , a zatim izvršimo projekciju tako dobivene relacije na attribute Y .

Definicija 3.5. Kažemo da za relaciju $R(X,Y,Z)$, gde su X, Y, Z međusobno disjunktni skupovi atributa, važi višeznačna zavisnost $X \rightarrow\rightarrow Y$ ako i samo ako je $Y_{xz} = Y_{xz'}$ za svako x, z i z' za koje su Y_{xz} i $Y_{xz'}$ neprazni skupovi.

Oдавде vidimo da $Y_{xz} = Y_{xz'}$ važi ako i samo ako važi da kad god su (x, y, z) i (x, y, z') elementi relacije R , tada su to i (x, y', z) i (x, y, z') . Otuda imamo sledeću definiciju potrebnog i dovoljnog uslova za postojanje višeznačne zavisnosti koja je ekvivalentna prethodnoj definiciji.

Definicija 3.4. Za relaciju $R(X,Y,Z)$ važi višeznačna zavisnost $X \rightarrow\rightarrow Y$ ako i samo ako je zadovoljen uslov: kad god su (x, y, z) i (x, y', z') elementi relacije R , tada su to i (x, y', z) i (x, y, z') .

Ove definicije višeznačne zavisnosti mogu se modifikovati tako da se dobiju odgovarajuće definicije funkcionalne zavisnosti. Ako u definiciji 3.3 zahtevamo da Y_{xz} , osim toga što zavisi samo od x , sadrži najviše jedan element, dobijamo uslov za postojanje funkcionalne zavisnosti $X \rightarrow Y$. Analogno, ako u definiciji 3.4. zahtevamo da kad god su (x, y, z) i (x, y', z') elementi relacije R , tada je $y = y'$. Dakle funkcionalne zavisnosti su samo jedan specijalan slučaj višeznačnih zavisnosti.

U gornjem primeru, na osnovu definicije, vredi višeznačna zavisnost:

Ispit $\rightarrow\rightarrow$ Sat i Učionica.

Pored toga vrede i sledeće višeznačne zavisnosti:

Ispit $\rightarrow\rightarrow$ Sat i Ocena i Sat i Učionica $\rightarrow\rightarrow$ Student i Ocena.

Takođe, postoji i trivijalna višeznačne zavisnost:

Sat i Učionica $\rightarrow\rightarrow$ Učionica.

Primetimo da ne vrede sledeće višeznačne zavisnosti:

Ispit $\rightarrow\rightarrow$ Sat,

Ispit $\rightarrow\rightarrow$ Učionica.

Značaj višeznačnih zavisnosti je u tome što one daju potreban i dovoljan uslov da je relacija rezultat prirodnog spajanja svojih projekcija.

3.1.6. Osobine višeznačnih zavisnosti

Višeznačne zavisnosti dosta su složenije od funkcionalnih zavisnosti, pa je i skup osnovnih i izvedenih pravila koji se odnosi na njih dosta složeniji. Na primer, iz $X \rightarrow\rightarrow Y$ i $Y \rightarrow\rightarrow Z$ ne vredi obavezno i $X \rightarrow\rightarrow Z$. Imamo sledeće osobine za višeznačne zavisnosti:

- Komplementno pravilo. Ako vredi $X \rightarrow\rightarrow Y$, tada vredi i $X \rightarrow\rightarrow U - X - Y$
- Tranzitivnost. Ako važi $X \rightarrow\rightarrow Y$ i $Y \rightarrow\rightarrow Z$ tada važi i $X \rightarrow\rightarrow (Z - Y)$.
- Aditivnost. Ako vredi $X \rightarrow\rightarrow Y$ i $V \subseteq W$ tada vredi i $WX \rightarrow\rightarrow VY$.

3.2. Fazi relacioni model baza podataka

U prethodnoj deceniji veliki broj autora [1,2,17,18,39,42] postavio je pitanje uvođenja nepreciznosti i neodređenosti u relacionim bazama podataka, a to nas dovodi do sistema baze podataka koji leže u okviru veštačke inteligencije, jer nam se omogućava da upravljamo informacijama koje su veoma slične prirodnom jeziku. Codd uvodi organizaciju relacione baze podataka koja je zasnovana na teoriji skupova i relacionoj teoriji. Zadeova teorija fazi skupova predstavlja generalizaciju opšte teorije skupova, dok je koncept fazi relacije generalizacija relacione teorije. Zbog toga Buckles i Petry predlažu fazi relacioni model baza podataka koji nudi mogućnost skladištenja i upravljanja nepreciznim i neodređenim informacijama.

Rasplinute, neprecizne informacije (fazi informacije) mogu biti uključene u sistem na dva različita nivoa: prvi nivo razmatra mogućnost definisanja nepreciznih upita u klasičnim relacionim bazama podataka, a drugi se odnosi na problem dodavanja nepreciznih informacija u bazu. U oba slučaja, teorija fazi skupova predstavlja moćan alat za upravljanje neodređenim i nepreciznim podacima. Razmatranje drugog pristupa nas dovodi do *Fazi Relacionih Modela Baza Podataka*.

Grupa autora [5,29] uvodi model nazvan GEFRED (General Fuzzy Relational Database). U ovom delu uvodimo osnovne elemente fazi proširenja relacionog modela baza podataka.

Struktura GEFRED modela može da se prikaže na sledeći način:

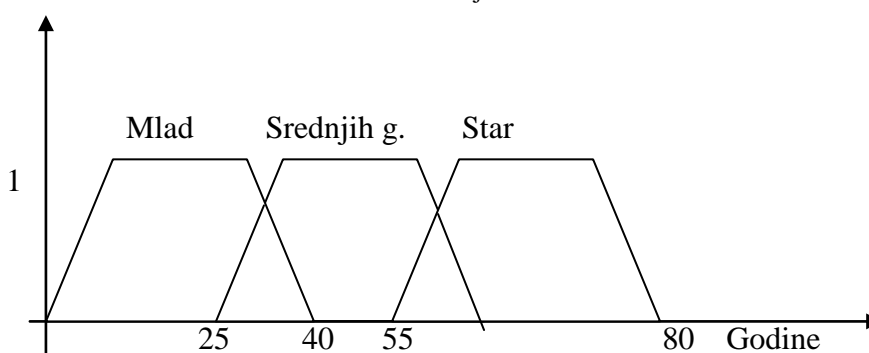
$$R_{FG} \in (D_{G1}, C_1) \times \dots \times (D_{Gn}, C_n),$$

gde je D_{Gi} domen atributa, a C_i "kompatibilnost atributa" koja uzima vrednost iz intervala $[0,1]$. U Fazi Relacionom Modelu kompatibilnost atributa se ne prikazuje, ali u svakoj n-torci vrednost atributa povezana je sa odgovarajućom vrednošću C_i .

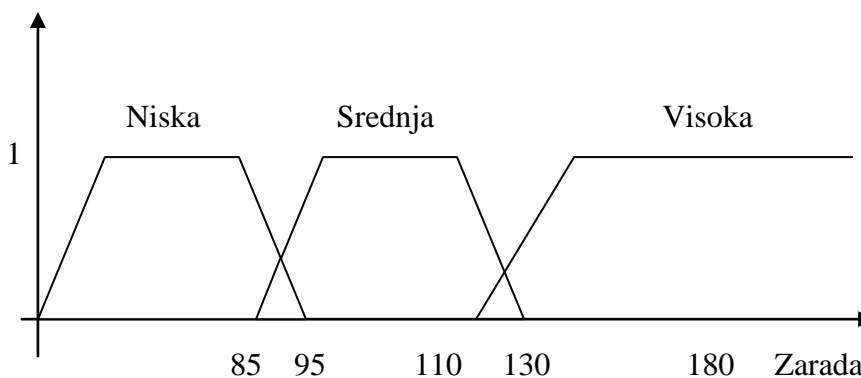
Posmatrajmo sledeći primer koji opisuje proširenje klasične relacione baze podataka u GEFRED model:

Ime	Adresa	Godine	Produktivnost	Zarada
Marko	Bulevar revolucije	31	Dobra	Visoka
Aleksandar	Medaković	Srednjih g.	Zadovoljavajuća	100 000
Nenad	Karaburma	Mlad	Loša	90 .000
Saša	Novi Beograd	Star	Odlična	Niska
Vlada	Cerak	Mlad	Dobra	Srednja
Goran	Rakovica	Oko 28	Dobra	125 000
Milan	Žarkovo	Između 30 i 35	Zadovoljavajuća	105 000

Tabela 11. Fazi relacija GEFRED



Slika 8. Funkcija pripadnosti atributa Godine [5]



Slika 9. Funkcija pripadnosti atributa Zarada [5]

Atributi Ime i Adresa sadrže crisp informacije, pri čemu je primarni ključ atribut Ime. Sa druge strane, atributi Godine i Zarada sadrže fazi informacije, a odgovarajuće funkcije pripadnosti za lingvističke promenljive iz tabele data su na slici 8 i 9. Atribut Produktivnost sadrži informacije iz diskretnog domena (4 tipa vrednosti), pa u tom slučaju moramo da definišemo relaciju sličnosti među elementima ovog domena, a to je prikazano u sledećoj tabeli:

S (d_i, d_j)	Loša	Zadovoljavajuća	Dobra	Odlična
Loša	1	0.8	0.5	0.1
Zadovoljavajuća	0.8	1	0.7	0.5
Dobra	0.5	0.7	1	0.8
Odlična	0.1	0.5	0.8	1

Tabela 12. Relacija sličnosti za atribut Produktivnost [5]

Ovaj teorijski model uključuje neophodne elemente za definisanje Fazi Relacionih Modela Baza Podataka (FRDBM). U okviru ove baze, možemo da ispitamo zavisnost između pojedinih atributa (npr. Godine, Zarada i Produktivnost) i da primenjujući rezultate fazi logike odredimo nepoznate vrednosti atributa, tj. da ih preciziramo. Ovaj model se može posmatrati kao proširenje klasične relacione baze podataka koji je sposoban za predstavljanje i upravljanje nepreciznim i neodređenim informacijama. U ovom primeru, možemo da sagledamo i usvojim prvobitne pristupe u analizi FRDBM:

- Predstavljanje fazi informacija u okviru baza podataka i
- Implementacija FRDBM korišćenjem mehanizama iz klasičnih baza podataka (SQL upiti).

Naredni korak u razvoju ovakvih modela baza podataka bio je fazi proširenje i formulacija SQL jezika. Prvobitne ideje kretale su se u smeru realizacije softvera za prevođenje fazi u klasične SQL upite.

3.2.1. Nesavršene informacije i teorija fazi skupova

Klasični modeli baza podataka često "pate" od svoje nesposobnosti predstavljanja i manipulisanja nepreciznim i neodređenim informacijama koje se javljaju u stvarnom svetu brojnih aplikacija. Od ranih 80-tih godina, Zadeova fazi logika se koristi za proširenje različitih modela podataka. Svrha uvođenja fazi logike u baze podataka jeste mogućnost predstavljanja i upravljanja neodređenim i nepreciznim informacijama.

Nedoslednost, nepreciznost, neodređenost, neizvesnost i nejasnoća su osnovne vrste nesavršenih informacija u sistemima baza podataka [75]:

1. Nedoslednost je vrsta semantičkog konflikta, što znači isti aspekt stvarnog sveta je predstavljen različito u jednoj ili različitim bazama podataka. Npr. Starost jedne osobe je skladištena kao 34 i 37 godina istovremeno.
2. Intuitivno, nepreciznost i neodređenost su relevantni za sadržaj vrednosti atributa, a to znači da vrednost atributa mora da leži u datom opsegu (intervalu ili skupu) vrednosti, ali mi ne znamo koja vrednost će se tačno birati u datom trenutku. U principu, nejasne informacije se predstavljaju lingvističkim varijablama. Na primer, mlad čovek je skup vrednosti {18, 19, 20, 21, 22, 23} što znači da mlad čovek može imati 18, ali i 23 godine.
3. Neizvesnost se odnosi na stepen istine vrednosti atributa, a to znači da možemo da procenimo verodostojnost pojedinih vrednosti atributa. Na primer, mogućnost da je Marko star 35 godina je 97%. Slučajne neizvesnosti koje se opisuju teorijom verovatnoće se ne uzimaju u obzir ovde.
4. Nejasnoća – dvosmislenost znači da nekim elementima modela podataka nedostaje potpuna semantika što dovodi do nekoliko mogućih tumačenja.

Generalno, nekoliko različitih nesavršenosti može da koegzistira u odnosu na isti podatak u bazi. Na primer, starost osobe je podatak iz skupa vrednosti {18,19,20,21} i njihove mogućnosti (stepen pripadnosti) je 70%, 95%, 98% i 85% respektivno. Nepreciznost, neizvesnost i nejasnoća su tri glavne vrste nesavršenih informacija sa kojima se susrećemo u tradicionalnim bazama podataka. Mnogi od postojećih pristupa koji se bave nepreciznošću i neodređenošću informacija se baziraju na teoriji fazi skupova i mogućnosti distribucije. Fazi skup {0.7/18, 0.85/19, 0.95/20, 0.98/21} za starost osobe sadrži

neprecizne informacije (starost osobe može biti 18, 19, 20 ili 21 godinu) i stepen pripadnosti (0.7, 0.85, 0.95 i 0.98) istovremeno.

3.2.2. Predstavljanje podataka u fazi modelu

U klasičnim relacionim bazama podataka [91] sve informacije moraju biti precizno definisane, a u slučaju da osnovne vrednosti nisu precizno definisane, tada one uključuju takozvanu NULL vrednost. Č o d d razmatra NULL vrednost u značenju *kompletno nepoznat* tj. neke od vrednosti u domenu atributa mogu da poprime ovo značenje.

U relacionom modelu baza podataka imamo sledeća tri osnovna uslova:

- a) Za elemente x, y iz domena atributa A , ako x i y nisu identični tada su oni u potpunosti različiti.
- b) Za bilo koju n - torku t u tabeli, $t=(t(A_1), \dots, t(A_n))$, $t(A_i)$ može predstavljati samo jedan element na datom domenu.
- c) Bilo koja n - toraka predstavljena u tabeli u potpunosti pripada toj tabeli i to sa stepenom pripadnosti 1.

Kao drugi način razmatranja ovih nepreciznih informacija je uključivanje fazi vrednosti na domenu atributa. Ove neprecizne informacije su određene Zadeovom teorijom fazi skupova i fazi logikom i one omogućavaju matematički okvir za rad sa nepreciznim informacijama u fazi relacionim bazama podataka.

Neka je data relaciji $R(\text{Ime, Pol, Godine, Visina, Zdravlje, Boja_Kose})$ i neka u toj relaciji u nekom trenutku vremena imamo sledeće činjenice:

T_1 : (Marko, M, {21,22,23}, 176, osrednje, crvena ili smeđa)

T_2 : (Milan, M, srednjih-godina, {0.7/180, 1/183, 0.8/186}, vrlo-dobro, {bela, siva})

T_3 : (Saša, M, 27, 170, dobro, crna): μ , gde je μ pripadnost n -torke relaciji i $\mu \in [0, 1]$.

U datom modelu uzmimo na domenu atributa lingvističke izraze (vrlo, srednjih), fazi skupove (0.7/180, 1/183, 0.8/186), podskupove datoga domena (crvena, smeđa). Zatim, primetimo da n - torke mogu pripadati datoj relaciji sa nekim parcijalnim stepenom pripadnosti.

Za proširenje klasičnog modela podataka u fazi model najpre definišimo modele za predstavljanje podataka koji generiše gore spomenute pretpostavke. U ovom poglavlju daćemo četiri okvira za predstavljanja podataka u fazi modelu [23,28,55,56], i to *osnovni fazi model na bazi sličnosti, fazi model na bazi mogućnosti, fazi model na bazi fazi relacije i fazi model sa proširenom mogućnosti*.

U *fazi modelu na bazi sličnosti (Buckles i Petry)* svaki domen skupa atributa u fazi relacionim bazama podataka, kao prvo asociran je sa relacijom sličnosti, umesto identičnom relacijom, i drugo domen vrednosti je definisan kao podskup osnovnog baznoga skupa, umesto jednoga elementa domena kao u klasičnim relacionim bazama podataka. Tako imamo sledeću definiciju:

Definicija 3.5. Fazi relacija R je podskup skupa Dekartovog proizvoda $2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_n}$ gde je D_i konačan domen i 2^{D_i} partitivni skup skupa D_i . Bilo koji član relacije naziva se jednostavno n - torkom.

Definicija 3.6. Fazi n – torka t_i je bilo koji član od R i $2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_n}$ koja je data u formi $t_i = (d_{i1}, d_{i2}, \dots, d_{ij}, \dots, d_{in})$, gde je $d_{ij} \subset D_j$, $j=1..n$, $d_{ij} \neq \emptyset$.

Relacija sličnosti s_j na D_j je preslikavanje $D_j \times D_j \rightarrow [0,1]$, tako da za $x, y, z \in D_j$ važi:

- a) $s_j(x,x) = 1$ (refleksivna)
- b) $s_j(x,y) = s_j(y,x)$ (simetrična)
- c) $s_j(x,z) > \max(\min(s_j(x,y), s_j(y,z)))$ (max-min tranzitivna).

Za dati domen D_j , *prag sličnosti* definisan je kao:

$$\text{Prag}(D_j) = \min_{\forall i} \left\{ \min_{x,y \in d_{ij}} [s(x,y)] \right\}$$

Primer ovoga modela moguće je predstaviti n-torkom T_1 :

T_1 : (Marko, M, {21,22,23}, 176, osrednje, crvena ili smeđa)

Fazi model na bazi mogućnosti distribucije (Prade i Testemale) proširuje teoriju klasičnih relacionih baza podataka dozvoljavajući upotrebu fazi vrednosti za attribute. Fundamentalni koncept fazi informacija jeste da promenljiva (atribut) nije vezana za konkretnu vrednost. U tom kontekstu se koristi termin *mogućnost distribucije*, koja svaku moguću vrednost promenljive povezuje sa vrednostima iz intervala $[0,1]$. Uopšteno, mogućnost distribucije se identifikuje sa funkcijom stepena pripadnosti. . Zato kažemo da element x pripada fazi skupu A (‘Visina osobe’) sa stepenom pripadnosti 0.9. Onda je mogućnost distribucije elementa $x \in X$ na fazi skupa A , koji je definisan na domenu X , $\pi_A(x)=0.9$.

Definicija 3.7. Fazi relacija R je podskup domena $\Pi(D_1) \times \dots \times \Pi(D_n)$, gde je:

$$\Pi(D_i) = \{ \pi_{A_i} \mid \pi_{A_i} \text{ je moguća distribucija od } A_i \text{ na } D_i \}$$

Odgovarajuća n - torka je data u formi $t_i = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_n})$.

Nadalje, u ovom modelu uveden je jedan poseban element e za takozvane *neprimenljive* situacije koje mogu poprimiti odgovarajuće vrednosti atributa. U tu svrhu definišemo sledeću funkciju:

$$\pi_{A_i}: D \cup e \rightarrow [0,1]$$

Primer ovoga modela moguće je predstaviti n-torkom T_2 :

T_2 : (Milan, M, srednjih-godina, {0.7/180,1/183,0.8/186}, vrlo-dobro, {bela, siva}).

U fazi modelu na bazi fazi relacije (Baldwin), fazi relacija definisana je na sledeći način:

Definicija 3.8. Fazi relacija R na $D_1 \times \dots \times D_n$ određena je sa funkcijom pripadnosti

$$\mu_R : D_1 \times \dots \times D_n \rightarrow [0,1], \text{ gde je } D_i, \text{ domen atributa } A_i$$

Generalna forma binarne relacije R na $D_1 \times D_2$ predstavljena je kao:

$$R = \{ \mu_R(u_1, v_1) / (u_1, v_1), \dots, \mu_R(u_m, v_n) / (u_m, v_n) \}$$
 u n - torci date sa

$$R = \{ u_1, v_1, \mu_R(u_1, v_1), \dots, u_m, v_n, \mu_R(u_m, v_n) \}$$

gde je $u_j \in D_1$, $j = 1,2,\dots,m$ i $v_k \in D_2$, $k = 1,2,\dots,n$.

Ovaj model određuje da je pripadnost neke n - torke datoj relaciji jeste fazi, dok individualne vrednosti atributa ne moraju biti fazi, ili mogu biti lingvističke varijable, ali tada se tretiraju kao atomske ili kao jedno - vrednosne varijable.

Primer ovoga modela moguće je predstaviti n-torkom T_3 :

T_3 : (Sasa, M, 27, 170, dobro, crna): μ

Fazi model sa proširenom mogućnošću proširuje fazi model na bazi mogućnosti distribucije, ne samo dozvoljavajući moguću distribuciju vrednostima atributa, već dozvoljavajući i relaciju bliskosti koja je u vezi sa datim domenom. Ovo proširenje generalizuje klasični relacioni model baza podataka. Naznačimo da je relacija sličnosti samo specijalni slučaj relacije bliskosti kod koje se zahteva da bude refleksivna i simetrična.

Definicija 3.9. Relacija bliskosti c_i na domenu D_i je preslikavanje $D_i \times D_i \rightarrow [0,1]$ tako da za svako $x,y \in D_i$ je:

- a) $c_i(x,x) = 1$ (refleksivna)
- b) $c_i(x,y) = c_i(y,x)$ (simetrična)

Na taj način gore spomenuti modeli postaju specijalni slučajevi *fazi modela sa proširenom mogućnošću*.

Definicija 3.10. Fazi relacija R je podskup domena $\Pi(D_1) \times \dots \times \Pi(D_n)$, gde je:

$$\Pi(D_i) = \{\pi_{A_i} \mid \pi_{A_i} \text{ je moguća distribucija od } A_i \text{ na } D_i\}$$

Odgovarajuća n - torka je data u formi $t_i = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_n})$.

Na taj način pomoću relacije bliskosti određujemo bliskost elemenata odgovarajućih atributa na datom domenu. Primer ovoga modela moguće je predstaviti n-torkom T_2 :

T_2 : (Milan, M, srednjih-godina, {0.7/180, 1/183, 0.8/186}, vrlo-dobro, {bela, siva}).

3.2.3. Relacija bliskosti

Relacija sličnosti je jedna od najvažnijih relacija koja se primenjuje u analizi domena skalarnih fazi atributa. Fazi relacioni model baza podataka zasniva se, između ostalog, na relaciji sličnosti skalarnih domena. Ove relacije su, kao što smo napomenuli refleksivne, simetrične i max-min tranzitivne. Buckles i Petry u svojim radovima predlažu uvođenje fazi informacija u relacione strukture baza podataka. U ovim strukturama, kao i kod klasične teorije relacionih baza, podaci se definišu kao uređene n-torke.

Međutim, fazi relaciona baza podataka R je podskup Dekartovog proizvoda:

$$R \subset 2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_n},$$

gde je D_i domen atributa A_i , a t_i n-torka koja pripada relaciji R .

Ovakva struktura nam daje mogućnost da koristimo podskup vrednosti domena D_i kao komponente uređene n-torke da bismo na efikasniji način mogli da upravljamo sa rasplnutim, neodređenim i nepreciznim informacijama u bazi podataka. Što je veća rasplnutost informacija veći je podskup iz kojih može da se uzme odgovarajuća vrednost.

Fazi relacioni model baza podataka Buckles-a i Petry-ija oslanja se na specifikaciju sličnosti za svaki poseban skalarni domen u bazi. Relacija sličnosti mapira sličnost u zatvoreni interval [0,1] za svaki par elementa posmatranog domena.

Međutim, postoje kritički osvrti na upotrebu relacije sličnosti u fazi relacionim bazama podataka. Razloge ovih kritika možemo lako da ilustrujemo i adekvatnim primerom. Posmatrajmo sličnost između brojeva 1,2,3 koji mogu da budu elementi nekog skalarnog domena. Pretpostavimo da je sličnost između 1 i 2 izražena sa 0.8, a između 2 i 3 takođe sa 0.8. Prema osobini max-min tranzitivnosti koja važi za relacije sličnosti, zaključujemo da sličnost između 1 i 3 ne može biti manja od 0.8. Odavde je uočljivo da se 1, 2 i 3 nalaze u istoj klasi ekvivalencije. Osobina tranzitivnosti uvodi vrlo restriktivno ograničenje, što može biti kontraintuitivno za određene domene. U tom smislu, fazi relacioni model baza podataka Buckles-a i Petry-ija se proširuje uvođenjem relacije bliskosti na sklaranom domenu. Svojstva refleksivnosti i simetrije su veoma pogodna za izražavanje stepena bliskosti ili blizine između elemenata skalarnog domena.[62] Sada je jasno da je relacija sličnosti specijalni slučaj relacije bliskosti. Važna činjenica koju nam obezbeđuje relacija bliskosti je particionisanje i klastorevanje skalarnog domena. Ako se vratimo na prethodni primer, uočićemo da sada možemo da bliskost između brojeva 1 i 3 izrazimo sa vrednošću 0.6. Ovo nam daje značajnu mogućnost kada je reč o klasterovanju domena u fazi relacionim bazama podataka i efikasnije kreiranje SQL upita nad podacima u bazi.

Primer: Posmatrajmo FRDBM u kojoj je prisutan fazi atribut BOJA KOSE kao fizička karakteristika neke individue. Relacija bliskosti između elementa domena BOJA KOSE prikazana je u sledećoj tabeli:

Boja Kose	C	TB	K	C	S	SB	P
Crna	1.0	0.8	0.6	0.5	0.4	0.3	0.1
Tamno braon	0.8	1	0.7	0.6	0.6	0.5	0.2
Kestenjasta	0.6	0.7	1	0.8	0.7	0.4	0.3
Crvenkasta	0.5	0.6	0.8	1	0.7	0.5	0.4
Svetla	0.4	0.6	0.7	0.7	1	0.7	0.5
Svetlo braon	0.3	0.5	0.4	0.5	0.7	1	0.8
Plava	0.1	0.2	0.3	0.4	0.5	0.8	1

Tabela 13. Relacija bliskosti za Atribut BOJA KOSE [62]

Pretpostavimo sada situaciju da su pojedinci iz baze osumnjičeni za podmetanje požara. Jedini dokaz koji je dostupan u istrazi je boja kosa počinioca (svetlo braon), koju je uspeo da zapazi jedan od čuvara. Sada smo u mogućnosti da formulišemo fazi upit:

```
SELECT *
FROM Fizicke karaktersitike
WHERE Boja_Kose = 'SB'
WITH LEVEL (Boja_Kose) = 0.8)
```

Pogledajmo sada bazu podataka:

Ime	Boja kose	Građa
Marko	Crna	Veliki
Miloš	Kestenjasta	Srednje razvijen
Petar	Plava	Veoma mali
Milorad	Svetlo braon	Mali
Rade	Tamno braon	Veliki
Momčilo	Crna	Veoma veliki

Tabela 14. Baza podataka

Na osnovu formulisanog upita, dobijamo iz baze podataka da su osumnjičene dve osobe:

Petar	Plava	Veoma mali
Milorad	Svetlo braon	Mali

Tabela 15. Rezultat upita

Dakle, prema dokazima sa kojim se raspolaže preporuka je da se u dalja ispitivanja uključe Petar i Milorad. Sada možemo da diskutujemo i o efektu uvođenja relacije bliskosti. Jedna od najzanimljivijih stvari jeste klasterizacija domena. U našem primeru, uočavamo sledeće klustere za različite vrednosti α :

$\alpha \in (0.8, 1]$ {Crna}, {Tamno braon}, {Kestenjasta}, {Crvenkasta}, {Svetla}, {Plava}.

$\alpha \in (0.7, 0.8]$ {Crna, Tamno braon}, {Kestenjasta, Crvenkasta}, {Plava, Svetlo braon}

$\alpha \in (0.5, 0.7]$ {Crna, Tamno Braon, Crvena, Crvenkasta}, {Svetla, Svetlo braon}

$\alpha \in (0, 0.5]$ {Crna, Tamno braon, Crvenkasta, Kestenjasa, Plava, Svetla, Svetlo braon}.

3.2.4. Fazi upiti

Relacija bliskosti i particija domena [49,64] su veoma važni alati koje se koriste u kreiranju fazi upita u fazi relacionim modelima baza podataka. Neka su dati domen D i odgovarajuća relacija bliskosti p definisana na domenu D. Vrednosti p(x,y) je potrebno rasporediti u rastućem redosledu:

$$m_1 < m_2 < \dots < m_k=1$$

Podintervali $(m_r, m_{r+1}]$, uključujući i $[0, m_1]$ nazvani p-particije se označavaju na sledeći način:

$$I_1 = \begin{cases} [0, m_1], & \text{ako je } m_1 \neq 0 \\ 0 & \text{ako je } m_1 = 0 \end{cases}$$

$$I_2 = (m_1, m_2],$$

$$I_3 = (m_2, m_3],$$

...

$$I_k = (m_{k-1}, m_k].$$

pri čemu je k broj particija domena.

Za $\alpha \in I_1$ particija (klaster) domena D se označava sa P_1 ,

$\alpha \in I_2$ particija (klaster) domena D se označava sa P_2 ,

....

$\alpha \in I_k$ particija (klaster) domena D se označava sa P_k .

Sledeća nejednakost služi za definisanje početnog broja particija (klastera) domena D:

$$k \leq \frac{c^2 - c}{2} + 1$$

pri čemu je c kardinalnost domena D.

Ako je $P_i = P_{i+1} \neq P_{i+2}$ ($P_{i-1} \neq P_i$) tada intervale I_i i I_{i+1} spajamo i dobijamo prošireni interval. Na ovaj način dolazimo do optimalnog broja klastera nekog domena D, pa imamo sledeću notaciju:

Za $\forall \alpha \in J_1$ particija (klaster) domena D se označava sa Q_1 ,

$\forall \alpha \in J_2$ particija (klaster) domena D se označava sa Q_2 ,

....

$\forall \alpha \in J_m$ particija (klaster) domena D se označava sa Q_m .

J_1, J_2, \dots, J_m zovemo optimalne p-particije ili jednostavnije rečeno optimalne particije domena D, pri čemu je jasno da je $m \leq k$. Krajnje tačke intervala J_1, J_2, \dots, J_m zovu se čvorovi optimalne podele. Ove čvorne tačke su jako važne prilikom donošenja odluka o izboru nivoa vrednosti prilikom kreiranja fazi upita.

Na sledećem primeru prikazana je optimalna klasterizacija domena. Posmatrajmo relaciju **Fizičke karakteristike** (Ime, Boja_Kose, Građa).

Ime	Boja Kose	Građa
Marko	Crna	Veliki
Miloš	Tamno braon	Srednje razvijen
Petar	Kestenjasta	Srednje razvijen
Ivan	Svetla braon	Mali
Vlada	Tamno braon	Srednje razvijen
Goran	Crvenkasta	Veoma mali
Saša	Plava	Veoma veliki
Ognjen	Svetla braon	Srednje razvijen
Tomo	Kestenjasta	Veliki
Boško	Svetla braon	Veliki

Tabela 16. Tabela fizičke karakteristike

Građa	Veoma veliki (VV)	Veliki (V)	Srednje r. (S)	Mali (M)	Veoma mali (VM)
VV	1.0	0.8	0.5	0.3	0.1
V	0.8	1.0	0.6	0.4	0.2
S	0.5	0.6	1.0	0.6	0.4
M	0.3	0.4	0.6	1.0	0.7
VM	0.1	0.2	0.4	0.7	1.0

Tabela 17. Relacija bliskosti na domenu atributa Građa

Boja_Kose={Crna, Tamno braon, Kestenjasta, Crvenkasta, Svetla, Svetlo braon, Plava}

Građa={Veoma veliki, Veliki, Srednje razvijen, Mali, Veoma mali}.

Tabela 16. reprezentuje fizičke karakteristike osoba, pri čemu svaka n-torka sadrži sledeće informacije: ime osobe, njegova boja kose i građa. Vrednosti domena i nad njima generisane relacije bliskosti su date u tabelama 13. i 17. Usmerimo našu pažnju na atribut Boja_Kose.

Vrednosti iz relacije bliskosti date u rastućem redosledu:

$$0.1 < 0.2 < 0.3 < 0.4 < 0.5 < 0.6 < 0.7 < 0.8 < 0.9 < 1.0.$$

Vidimo da ja $9 < \frac{c^2 - c}{2} + 1$, pa su podintervali:

$$I_1 = [0, 0.1],$$

$$I_2 = (0.1, 0.2],$$

$$I_3 = (0.2, 0.3],$$

$$I_4 = (0.3, 0.4],$$

$$I_5 = (0.4, 0.5],$$

$$I_6 = (0.5, 0.6],$$

$$I_7 = (0.6, 0.7],$$

$$I_8 = (0.7, 0.8],$$

$$I_9 = (0.8, 1.0],$$

Odgovarajuće particije domena Boje_Kose uzimajući u obzir relaciju bliskosti na ovom domenu:

$$P_1 = \{C, TB, K, C, SB, P\},$$

$$P_2 = \{C, TB, K, C, SB, P\},$$

$$P_3 = \{C, TB, K, C, SB, P\},$$

$$P_4 = \{C, TB, K, C, SB, P\},$$

$$P_5 = \{C, TB, K, C, SB, P\},$$

$$P_6 = \{C, TB, K, C\} \{SB, P\},$$

$$P_7 = \{C, TB, K, C\} \{SB, P\},$$

$$P_8 = \{C, TB\} \{K, C\} \{SB, P\} \text{ i}$$

$$P_9 = \{C\} \{TB\} \{K\} \{C\} \{SB\} \{P\}.$$

Primećujemo da nismo navodili vrednost Svetla (S) za atribut Boja_Kose jer se ne pojavljuje u relaciji **Fizičke karakteristike**. Iz datih klastera vidimo da je $P_1 = P_2 = P_3 = P_4 = P_5$ i $P_6 = P_7$. Prema tome, optimalna podela domena atributa Boje_Kose na intervalu $[0, 1]$ je:

$$J_1 = I_1 \cup I_2 \cup I_3 \cup I_4 \cup I_5 = [0, 0.5],$$

$$J_2 = I_6 \cup I_7 = (0.5, 0.7],$$

$$J_3 = I_8 = (0.7, 0.8]$$

$$J_4 = I_9 = (0.8, 1.0].$$

a odgovarajuće optimalne particije (klasteri) domena vrednosti su:

$$Q_1 = \{C, TB, K, C, SB, P\},$$

$$Q_2 = \{C, TB, K, C\} \{SB, P\},$$

$$Q_3 = \{C, TB\} \{K, C\} \{SB, P\},$$

$$Q_4 = \{C\} \{TB\} \{K\} \{C\} \{SB\} \{P\}.$$

Ako je $\alpha (\neq 0)$ čvor definišemo α^- i α^+ kao granične tačke klasterizovanog intervala:

$$\alpha^- = \begin{cases} [\alpha_p, \alpha], & \text{ako je } \alpha_p \neq 0 \\ [0, \alpha] & \text{ako je } \alpha_p = 0 \end{cases}$$

gde je α_p prethodni čvor od α . Ako je $\alpha (\neq 1)$ tada je:

$$\alpha^+ = \begin{cases} [\alpha, \alpha_n], & \text{ako je } \alpha \neq 0 \\ [0, \alpha_n] & \text{ako je } \alpha = 0 \end{cases}$$

gde je α_n sledeći čvor od α .

Za čvor α definišemo gornju i donju toleranciju:

$$lt(\alpha) = \frac{\text{lenght}(\alpha^-)}{\text{lenght of } [0,1]}$$

$$up(\alpha) = \frac{\text{lenght}(\alpha^+)}{\text{lenght of } [0,1]}$$

Čvor igra veoma važnu ulogu u donošenju odluka prilikom izbora nivoa vrednosti za fazi upite iz dva razloga:

1. Promenljive vrednosti tolerancije $lt(\alpha)$ i $up(\alpha)$ za različite vrednosti α . U prethodnom primeru $lt(0.5)=50\%$ i $up(0.5)=20\%$.

2. Za bilo koje vrednosti $\alpha_1 \in \alpha^-$ i $\alpha_2 \in \alpha^+$, odgovarajuće particije domena D značajno su različite bez obzira na vrednost razlike $\alpha_2 - \alpha_1$.

Ako α nije čvorna tačka onda se donja tolerancija računa po formuli:

$$lt(\alpha) = (\alpha - \alpha_p) = [100 \times (\alpha - \alpha_p)] \%$$

i gornja tolerancija:

$$ut(\alpha) = (\alpha_n - \alpha) = [100 \times (\alpha_n - \alpha)] \%$$

Ako izabrani nivo vrednosti nije čvor, može se videti da izborom odgovarajuće vrednosti α povećavamo ili smanjujemo vrednosti gornje i donje tolerancije, ali se ne menja odgovor na fazi upite. Međutim, ako je izabrani nivo vrednosti α čvorna tačka, a pri tome α povećavamo za bilo koji iznos, odgovori na fazi upite se drastično menjaju, dok smanjenjem vrednosti α unutar opsega ne menja rezultat fazi upita.

U prethodnom primeru možemo da formulišemo sledeći upit:

Pronaći osobe koje imaju manje ili više svetlo braon boju kose i koje su manje ili više velike?

Ovaj upit može da se prevede u sledeću formu u fazi relacionoj algebri:

```
SELECT *
FROM Fizčke karakteristike
WHERE Boja_Kose='Svetlo braon'
AND Građa='Veliki'
WITH LEVEL (Boja_Kose)=0.8
LEVEL (Građa)=0.8.
```

Rezultat ovog fazi upita prikazan je u sledećoj tabeli:

Ime	Boja_Kose	Građa
Saša	Plava	Veoma veliki
Boško	Svetlo braon	Veliki

Tabela 18. Rezultat fazi upita

Ako umesto $LEVEL(Boja_Kose)=0.8 \in 0.8^-$ uzmemo $0.801 \in 0.8^+$ rezultat fazi upita će izgledati:

Ime	Boja_Kose	Građa
Saša	Plava	Veoma veliki

Tabela 19. Rezultat fazi upita posle promene vrednosti α

što znači da smo promenom nivoa α za 0.001 dobili različite rezultate fazi upita. To je zbog činjenica da je $\alpha=0.8$ čvorna tačka. Međutim, ako za α uzmemo vrednost 0.9 koja nije čvorna tačka, imaćemo rezultat koji se ne razlikuje pri gornjoj i donjoj toleranciji od 10%.

Fazi relacioni model baza podataka predložen od strane Buckles-a i Petry-a, zasnovan na teoriji sličnosti, predstavlja generalizaciju klasične relacione baze podataka. Sheno i Melton [25,27,62] zamenjuju relaciju sličnosti sa relacijom bliskosti i na taj način proširuju fazi relacioni model baza podataka. Uvođenjem relacije bliskosti značajno se pojednostavljuje generisanje fazi upita nad fazi relacionim bazama podataka, što je i prikazano u ovom poglavlju.

Što se tiče kreiranja fazi upita nad relacionim bazama podataka, pored predložene strategije koja podrazumeva da je prag (α) povezan sa fazi upitom, druga strategija se oslanja na razlaganje fazi upita u osnovne operacije i procenu ovih operacija putem efikasnih algoritama. U poređenju sa drugom strategijom koja podrazumeva izgradnju potpuno novog sistema za implementaciju fazi upita, prva strategija može da se implementira pomoću regularnog DBMS-a. Z.M. Ma i Li Yan predlažu metod prevođenja fazi upita tako da se prag prevodi u fazi brojeve [52]. Treba napomenuti da fazi uslovi u fazi upitima mogu biti opisani fazi relacijama i fazi uslovima. Kombinujući fazi logiku i osnovne operacije kao što su $>$, $<$, $=$, \leq , \geq , \neq , te lingvističke varijable blizu, oko, najmanje, najviše sa poznatim vrednostima formiraju se fazi uslovi. Prevođenje fazi upita je korisno u sledećim situacijama:

- Fazi upit se sastoji od regularnih relacija ‘=’ i fazi uslova,
- Fazi upit se sastoji od fazi relacija (‘oko’, ‘malo više’) i poznatih vrednosti.

Za ilustraciju prethodne diskusije, posmatrajmo osnovni fazi upit sa fazi operandima:

```
SELECT XX
FROM RR
WHERE AθY
WITH [threshold]
```

gde je XX atribut(i), RR je relacija i AθY je fazi uslov sa fazi operandima. θ može biti $>$, $<$, $=$, \leq , \geq , \neq , between ili not between. Ako je θ between ili not between, AθY ima sledeću formu A between (not between) Y_1 and Y_2 .

U zavisnosti od Y, sledeći slučajevi se mogu identifikovati prilikom translacije fazi upita:

1. Y je jednostavan fazi uslov (fazi broj)

Neka je poznata vrednost praga α , a to je vrednost iz intervala [0,1]. Manja vrednost α podrazumeva da je veći broj n-torki koji zadovoljavaju dati uslov. Korisnik može da izabere vrednost α u skladu sa svojim zahtevima i svojim znanjem. Ako je $Y = [a,b]$, onda se fazi upiti mogu prevesti u klasične upite na sledeći način:

Fazi upit	Klasični upit – nakon translacije
SELECT XX FROM RR WHERE A=Y WITH α	SELECT XX FROM RR WHERE $A \geq a$ AND $A \leq b$
SELECT XX FROM RR WHERE $A > Y$ WITH α	SELECT XX FROM RR WHERE $A > b$
SELECT XX FROM RR WHERE $A < Y$ WITH α	SELECT XX FROM RR WHERE $A < a$
SELECT XX FROM RR WHERE $A \neq Y$ WITH α	SELECT XX FROM RR WHERE $A < a$ OR $A > b$
SELECT XX FROM RR WHERE $A \geq Y$ WITH α	SELECT XX FROM RR WHERE $A \geq a$
SELECT XX FROM RR WHERE $A \leq Y$ WITH α	SELECT XX FROM RR WHERE $A \leq b$

Tabela 20. Translacija fazi upita 1 [52]

Fazi uslov "A between Y_1 AND Y_2 " može biti posmatran kao $A \geq Y_1$ AND $A \leq Y_2$.

```
SELECT XX
FROM RR
WHERE A between  $Y_1$  AND  $Y_2$  WITH  $\alpha$ 
```

Ovaj fazi upit se prvo prevodi u:

```
SELECT XX
FROM RR
WHERE  $A \geq Y_1$  AND  $A \leq Y_2$  WITH  $\alpha$ 
```

I onda se dobija:

```
SELECT XX
FROM RR
WHERE  $A \geq b_1$  AND  $A \leq a_2$ .
```

Lako je videti da uslov $A \geq b_1$ AND $A \leq a_2$ ima smisla ako i samo ako je $b_1 \leq a_2$.

2. Y je složeni fazi uslov (fazi broj) uz upotrebu operatora \cup :

Neka je data vrednost praga α . Ako Y ima formu $[a,b]$ or $[c,d]$, onda se fazi upiti u ovakvoj formi mogu translirati na način prikazan u tabeli 21. Ovo je unija dva fazi broja koja daje fazi broj koji dolazi kao rezultat operacije unije.

```
SELECT XX
FROM RR
WHERE A > Mlad  $\cup$  Veoma_Star WITH  $\alpha$ 
```

Ovaj fazi upit je ekvivalentan sa:

```
SELECT XX
FROM RR
WHERE A between Y1 AND Y2 WITH  $\alpha$ 
```

Odnosno važi, kao u prethodnom slučaju:

```
SELECT XX
FROM RR
WHERE A  $\geq$  Y1 AND A  $\leq$  Y2 WITH  $\alpha$ 
```

Fazi upit	Klasični upit – nakon translacije
SELECT XX FROM RR WHERE A=Y WITH α	SELECT XX FROM RR WHERE A \geq a AND A \leq b OR A \geq c AND A \leq d
SELECT XX FROM RR WHERE A>Y WITH α	SELECT XX FROM RR WHERE A>max(b,d)
SELECT XX FROM RR WHERE A<Y WITH α	SELECT XX FROM RR WHERE A<min(a,c)
SELECT XX FROM RR WHERE A \neq Y WITH α	SELECT XX FROM RR WHERE A<a OR A>b OR A<c OR A>d
SELECT XX FROM RR WHERE A \geq Y WITH α	SELECT XX FROM RR WHERE A \geq min(a,c)
SELECT XX FROM RR WHERE A \leq Y WITH α	SELECT XX FROM RR WHERE A \leq max(b,d)

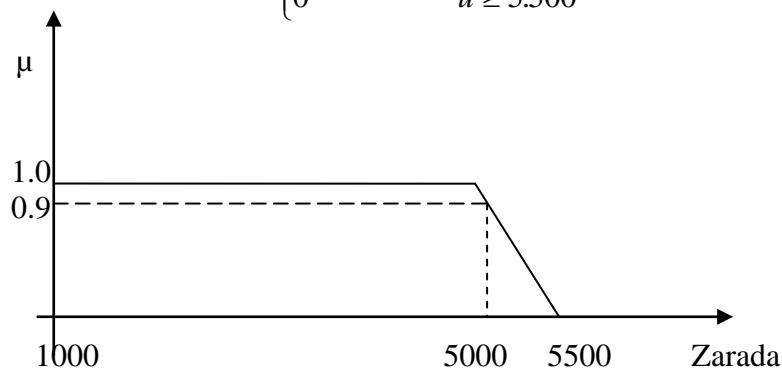
Tabela 21. Translacija fazi upita 2, [52]

Posmatrajmo relaciju **Radnik** (Ime, Godine, Zarada) i fazi upit:

```
SELECT Ime
FROM Radnik
WHERE Zarada najviše 5000 WITH 0.9
```

Pretpostavili smo da je vrednost praga $\alpha=0.9$. Osim toga, pretpostavka je da je domen atributa Zarada [1.000,7.000] i fazi uslov najviše 5.000 je predstavljen funkcijom pripadnosti:

$$\mu_{\text{najviše } 5000}(u) = \begin{cases} 1 & u \leq 5.000 \\ \frac{5.500 - u}{500} & 5.000 < u < 5.500 \\ 0 & u \geq 5.500 \end{cases}$$



Slika 10. Funkcija pripadnosti atributa Zarada

Fazi uslov ‘najviše 5.000’ za vrednost $\alpha=0.9$ prikazan je na slici 10. i rezultat je [1000,5050]. Prema tome fazi upit prevodimo:

```
SELECT Ime
FROM Radnik
WHERE Zarada ≥ 1000 AND Zarada ≤ 5000
```

3.2.5. Redundancija podataka

Redundancija podataka je jedan od primarnih problema u klasičnim bazama podataka. Kao prvo, redundancija označava postojanje duplih n - torke, kojima nije dozvoljeno da se pojavljuju u pojedinačnim tabelama. To znači da neće postojati n- torke t i t' u R, takve da su t i t' identične u odnosu na odgovarajuće vrednosti atributa.

Definicija 3.11. Za n-torku $t = (t(A_1), t(A_2), \dots, t(A_n))$ i n-torku $t' = (t'(A_1), t'(A_2), \dots, t'(A_n))$ u relaciji R, kažemo da su jednake ako i samo ako je $t(A_i) = t'(A_i)$, $i = 1, 2, \dots, n$.

Alternativno, redundancija n - torke može se takođe definisati na osnovu primarnog ključa K, na osnovu koncepta zavisnosti podataka. Primarni ključ K relacije R na relacionoj šemi R(U) je podskup od U takav da za jednake vrednosti n - torke na K impliciraju jednake vrednosti n - torke za svaki atribut $A_i \in U$, pa su tada dve n - torke t i t' redundantne ako i samo ako su njihove odgovarajuće vrednosti n - torke na K jednake. Drugim rečima, budući da iz $t(K) = t'(K)$ implicira $t(A_i) = t'(A_i)$, n - torke t i t' su redundantne ako i samo ako je $t(K) = t'(K)$.

Osim toga, redundancija takođe znači dupliranje vrednosti u odnosu na neke atribute u skupu. U fazi relacionim bazama podataka ne postoji jasan način potvrđivanja da su dve vrednosti atributa jednake ili različite, već se posmatra kada su one približno jednake. Zbog toga klasična razmatranja ne važe i problem redundancije nije više trivijalan i zahteva dodatna razmatranja.

Prvi problem nastaje zbog toga što vrednost nekog atributa n - torke može biti podskup domena ili može biti i moguća distribucija. Na primer, $Visina = \{170, 180, 190\}$ ili $Visina = \{0.3/150, 0.9/175, 0.1/200\}$.

Drugi problem nastaje kada se neke fazi relacije definišu na datim domenima, na primer relacija sličnosti ili relacija bliskosti. Postoje situacije kada je potrebno uvesti takve relacije nad domenima. U klasičnom slučaju na domenu $D = \{crn, plav, braon, siv, zelen, crven, ljubičast, žut, bel\}$ atribut *Boja* na D može uzeti bilo koju vrednost iz domena D . Ali, kada kažemo da je nešto tamno? Da li je jasno tada određena granica? Dalje, ako je dozvoljeno da vrednosti atributa budu podskupovi domena D , sa ciljem da predstavimo neodređenosti kao $\{crn, zelen\}$, $\{braon, crven\}$, $\{crn, siv, plav\}$ itd., kako onda odrediti jednakost ili sličnost za neki par vrednosti.

Na kraju veći problemi nastaju kada su vrednosti atributa distribucije mogućnosti uzete zajedno sa nekom fazi relacijom definisanom na domenu atributa.

3.2.6. Redundancija u fazi relacionim bazama podataka

U ovom odeljku daćemo neke od postojećih definicija redundancije u fazi relacionim bazama podataka, koje su rezultat radova [23,28,55]. Ove definicije baziraju se uglavnom na pojmu relacije sličnosti, relacije bliskosti, te nekim varijantama Zadeove relacije jednakosti.

a) U radu *Buckles i Petry* redundancija n - torke definiše se na osnovu relacije sličnosti među vrednostima n - torke.

Definicija 3.12. Za date domene D_1, D_2, \dots, D_n i relaciju $R \subseteq 2^{D_1} \times 2^{D_2} \times \dots \times 2^{D_n}$, dve n - torke t i t' su α - redundantne, gde je $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in [0,1]^n$, ako i samo ako je za $i = 1, 2, \dots, n$:

$$\min_{x, y \in d_i \cup d_i'} (s_i(x, y)) \geq \alpha_i$$

gde je $t = (d_1, \dots, d_i, \dots, d_n)$, $t' = (d_1, \dots, d_i', \dots, d_n)$, $d_i, d_i' \subseteq D_i$ i $d_i, d_i' \neq \emptyset$.

b) U radu *Shenoi i Melton* [8,27] dato je proširenje *Buckles i Petry* modela tako što relaciju sličnost s_i , zamenjuju sa relacijom bliskosti p_i , na datom domenu. Autori žele kontrolisati neku osobinu u odnosu na particiju domena, pa zato uvode i relaciju α - bliskost.

Definicija 3.13. Ako je p_i , relacija bliskosti na D_i tada za dati α_i , $0 < \alpha_i < 1$, dva elementa iz D_i , x i z , su α_i - bliski (u oznaci $xP_{\alpha_i}z$) ako i samo ako je $p_i(x, z) > \alpha_i$ ili ako postoji konačan niz $y_1, y_2, \dots, y_r \in D_i$ takav da je:

$$p_i(x, y_1) > \alpha_i, p_i(y_1, y_2) > \alpha_i, \dots, p_i(y_{r-1}, y_r) > \alpha_i, p_i(y_r, z) > \alpha_i,$$

Definicija 3.14. Neka su dati $p_i, \alpha_i = (1, \dots, n)$ i P_{α_i} . Dve n - torke t i t' iz R su α - redundantne, jedna u odnosu na drugu, ako i samo ako $xP_{\alpha_i}y$ važi za svako $x, y \in d_i \cup d'_i$, $i = 1, \dots, n$, gde je $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)$ i $t = (d_1, \dots, d_i, \dots, d_n)$, $t' = (d_1, \dots, d'_i, \dots, d_n)$.

Dakle dve n - torke su redundantne ako i samo ako za sve njihove vrednosti atributa respektivno, elementi iz $d_i \cup d'_i$ pripadaju istoj klasi ekvivalencije.

c) Rudenstajner i ostali autori baziraju njihova razmatranja na pojmu relacije (Res_i) na domenu D_i koja je u suštini relacija bliskosti. U njihovom radu redundancija n - torke je definisana na osnovu α - preklasa.

Definicija 3.15. Dve n - torke t i t' su α - β - redundantne jedna u odnosu na drugu, gde je $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in [0,1]^n$, $\beta \in [0,1]$, ako i samo ako za $i = 1, \dots, n$ važi:

$$a) \min_{\substack{x, y \in A_i \cup tA_i \\ \pi_{A_i}(x) > 0 \\ \pi_{A_i}(y) > 0}} (Res_i(x, y)) \geq \alpha_i$$

$$b) \min_{z \in D_i} (1 - |\pi_{A_i}(z) - \pi'_{A_i}(z)|) \geq \beta$$

gde je $t = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_n})$, $t' = (\pi'_{A_1}, \pi'_{A_2}, \dots, \pi'_{A_n})$, $t_{A_i} = \{w : \pi_{A_i}(w) > 0, w \in D_i\}$ i $t'_{A_i} = \{w : \pi'_{A_i}(w) > 0, w \in D_i\}$

d) Sada dajemo definiciju jednakosti n - torke u *fazi modelu na bazi mogućnosti distribucije*.

Za date dve n - torke $t_i = (\pi_{A_1}, \pi_{A_2}, \dots, \pi_{A_n})$, $t'_i = (\pi'_{A_1}, \pi'_{A_2}, \dots, \pi'_{A_n})$, propozicija da t je jednako sa t' definisana je preko mogućnosti distribucije G na $\{T, F\}$, gde T označava Tačno i F označava Lažno. Simbolički pišemo:

$$G(t, t') = Poss((\pi_{A_1} = \pi'_{A_1}) \text{ and } \dots \text{ and } (\pi_{A_n} = \pi'_{A_n})) = E(\pi_{A_1} = \pi'_{A_1}) \wedge \dots \wedge E(\pi_{A_n} = \pi'_{A_n}) = \{g^1/T, g^2/F\}$$

gde je $0 < g^1 < 1$, $0 < g^2 < 1$, E je preslikavanje iz $\Pi(D_i) \times \Pi(D_i)$ u $\Pi(\{T, F\})$ i \wedge je preslikavanje iz $\Pi(\{T, F\}) \times \Pi(\{T, F\})$ u $\Pi(\{T, F\})$. Drugim rečima, E i \wedge su dve mogućnosti distribucije na $\{T, F\}$. Vrednost $E(\pi_{A_1}, \pi'_{A_1})(T)$ i $E(\pi_{A_1}, \pi'_{A_1})(F)$ računa se na sledeći način:

$$E(\pi_{A_i}, \pi'_{A_i})(T) = \sup \min(\pi_{A_i}(x), \pi'_{A_i}(x)), x \in D_i, i = 1, 2, \dots, n$$

$$E(\pi_{A_i}, \pi'_{A_i})(F) = \sup \min(\pi_{A_i}(x), \pi'_{A_i}(y)), x, y \in D_i, x \neq y, i = 1, 2, \dots, n$$

Primenjujući Zadeov princip proširenja na operaciju \wedge imamo:

$$g^1 = [E(\pi_{A_1}, \pi'_{A_1}) \wedge \dots \wedge E(\pi_{A_n}, \pi'_{A_n})](T) = \sup_{e_1 \wedge e_2 \wedge \dots \wedge e_n = T} \min(E(\pi_{A_1}, \pi'_{A_1})(e_1) \wedge \dots \wedge E(\pi_{A_n}, \pi'_{A_n})(e_n)),$$

$$g^1 = \min(E(\pi_{A_1}, \pi'_{A_1})(T) \wedge \dots \wedge E(\pi_{A_n}, \pi'_{A_n})(T)),$$

$$g^2 = [E(\pi_{A_1}, \pi'_{A_1}) \wedge \dots \wedge E(\pi_{A_n}, \pi'_{A_n})](F) = \sup_{e_1 \wedge e_2 \wedge \dots \wedge e_n = F} \min(E(\pi_{A_1}, \pi'_{A_1})(e_1) \wedge \dots \wedge E(\pi_{A_n}, \pi'_{A_n})(e_n)),$$

Istinita vrednost g da je $t = t'$ može se dobiti na osnovu g^1 i g^2 na sledeći način:

$$g = (G(T) + 1 - G(F)) / 2 = (g^1 + 1 - g^2) / 2$$

Primer 3.1. Neka je domen $D = \{a,b,c,d\}$ i neka su date dve mogućnosti distribucije $\pi_1 = \{1/a, 0.2/b\}$ i $\pi_2 = \{1/c, 0.3/d\}$.

Prema prethodnoj definiciji mogućnost da $\pi_1 = \pi_2$ je T jednako je :

$$E(\pi_1, \pi_2) (T) = \sup_{x \in D_1} \min(\pi_1(x), \pi_2(x)) = \max(\min(1,0), \min(0.2,0), \min(0,1), \min(0,0.3)) = 0$$

i mogućnost da $\pi_1 = \pi_2$ je lazno (F) jednako je :

$$E(\pi_1, \pi_2)(F) = \sup_{x \neq y} \min(\pi_1(x), \pi_2(y)) = \max(\min(1,1), \min(1,0.3), \min(0.2,1), \min(0.2,0.3)) = 1$$

Sledi da je nemoguće da π_1 i π_2 budu jednaki.

Sada ćemo predstaviti podatke u fazi modelu sa proširenom mogućnošću.

Definicija 3.16. Neka je c_i relacija bliskosti na domenu D_i i neka je α_i odgovarajući prag za c_i $0 < \alpha_i < 1$. Neka je E preslikavanje iz $\Pi(D_i) \times \Pi(D_i)$ u $\Pi(\{T,F\})$ i neka su t i t' dve proizvoljne n - torke relacije $R \subset \Pi(D_1) \times \dots \times \Pi(D_n)$ sa $t = (\pi_{A1}, \pi_{A2}, \dots, \pi_{An})$ i $t' = (\pi'_{A1}, \pi'_{A2}, \dots, \pi'_{An})$. Mogućnost distribucije za $t = t'$ označena sa $G_c(t, t')$ je preslikavanje iz $R \times R$ u $\Pi(\{T,F\})$:

$$G_c(t, t') = \{ g^1 \text{ } / T, g^2 \text{ } / F \}$$

gde je:

$$g^1_c = [E_c(\pi_{A1}, \pi'_{A1}) \wedge \dots \wedge E_c(\pi_{An}, \pi'_{An})] (T) = \sup_{e1 \wedge e2 \wedge \dots \wedge en = T} \min(E_c(\pi_{A1}, \pi'_{A1})(e_1) \wedge \dots \wedge E_c(\pi_{An}, \pi'_{An})(e_n))$$

$$(e_n) = \min(E_c(\pi_{A1}, \pi'_{A1})(T), \dots, E_c(\pi_{Ai}, \pi'_{Ai})(T))$$

i

$$g^2_c = [E_c(\pi_{A1}, \pi'_{A1}) \wedge \dots \wedge E_c(\pi_{An}, \pi'_{An})] (F) = \sup_{e1 \wedge e2 \wedge \dots \wedge en = F} \min(E_c(\pi_{A1}, \pi'_{A1})(e_1) \wedge \dots \wedge E_c(\pi_{An}, \pi'_{An})(e_n))$$

$$(e_n) = \min(E_c(\pi_{A1}, \pi'_{A1})(F), \dots, E_c(\pi_{An}, \pi'_{An})(F))$$

i gde je: $E_c(\pi_{A1}, \pi'_{A1}) (T) = \sup \min(\pi_{Ai}(x), \pi'_{Ai}(y)), x, y \in D_i, c_i(x, y) \geq \alpha_i, i = 1, 2, \dots, n$

$$E_c(\pi_{A1}, \pi'_{A1}) (F) = \sup \min(\pi_{Ai}(x), \pi'_{Ai}(y)), x, y \in D_i, c_i(x, y) < \alpha_i, i = 1, 2, \dots, n$$

Primer 3.2. Neka su D , π_1 i π_2 dati kao u prethodnom primeru i neka je c_i na D dato u sledećoj tabeli:

c_i	a	b	c	d
a	1.0	0.2	0.9	0.1
b		1.0	0.2	0.95
c			1.0	0.1
d				1.0

Tabela 12. Relacija bliskosti za domen D

Mogućnost distribucije da je $\pi_1 = \pi_2$ sa E_c je za $\alpha_i = 0.9$.

$$E(\pi_1, \pi_2) (T) = \sup_{c_i(x, y) \geq \alpha_i} \min(\pi_1(x), \pi_2(y)) = \max(\min(1,1), \min(0.2,0.3)) = 1$$

$$E(\pi_1, \pi_2) (F) = \sup_{c_i(x, y) < \alpha_i} \min(\pi_1(x), \pi_2(y)) = \max(\min(1,0.3), \min(0.2,1)) = 0.3$$

3.3. Fazi objektno-orijentisane baze podataka

Klasične relacione baze podataka i njihovo proširenje u vidu fazi modela ne zadovoljavaju u potpunosti potrebu za modelovanjem kompleksnih objekata sa nepreciznim i neodređenim informacijama. Objektno-orijentisani modeli baza podataka mogu da predstavljaju složene strukture objekata bez fragmentacije agragiranih podataka i da se pri tome modeluju odnosi između atributa. Trenutni naponi u ovoj oblasti istraživanja su koncentrisani na nadogradnju postojećih modela kako bi se došlo do mogućnosti da se upravlja kompleksnim objektima i fazi informacijama istovremeno. Zbog toga u nastavku dajemo pregled fazi objektno-orijentisanih modela baza podataka.

U slučaju fazi objektno-orijentisanih baza podataka (FOODBs), fazi koncept može da se posmatra na nivou objekta ili na nivou klase i njihovih veza. Postoji veći broj radova i modela u kojima je analizirana upotreba "fuziness-a" u objektno-orijentisanim bazama podataka. U ovom radu dajemo pregled najvažnijih.

Umano, Imada i ostali [92] predstavljaju fazi objektno-orijentisani model u kojem atributi uzimaju fazi vrednosti sa određenim faktorom odnosno stepenom pripadnosti, dok se za manipulaciju podacima koristi SQL jezik. Sa druge strane, Gysegham i Caluwe [93] uvode neodređenost i rasplinutost u objektno-orijentisane baze podataka preko fazi skupova. Ponašanje i struktura objekata su nepotpuno definisani rezultati sa stanovišta njihovog instanciranja. Delimično nasleđivanje, uslovno nasleđivanje, kao i višestruko nasleđivanje su dozvoljeni u fazi hijerarhiji.

Grupa autora Bordogna, Pasi i Lucarella [94,95] predlaže fazi objektno-orijentisani model zasnovan na grafovima. Pojam snage izražen pomoću jezičkih kvalifikatora se koristi za iskazivanje odnosa između instanci, kao i odnosa između objekata i klasa. Fazi klase i fazi hijerarhija su takođe modelovani u ovom radu. Pored toga, ovi autori analiziraju i pretragu i selekciju podataka u predloženom modelu definišući operacije nad podacima u fazi objektno-orijentisanim modelima, a koji mogu biti i crisp i fazi.

U modelu koji predlažu Dubois, Prade i Rossazza [97] koristi se relacije sličnosti. Opseg vrednosti za svaki atribut posmatrane klase reprezentuje skup dozvoljenih vrednosti, a između posmatranih vrednosti definiše se stepen sličnosti. U zavisnosti od uključivanja stvarnih vrednosti atributa datog objekta iz raspona vrednosti za datu klasu može da se izračuna i stepen pripadnosti objekta klasi. Shodno tome, definišu se jaka i slaba hijerarhija klasa na osnovu monotono opadajuće ili rastuće funkcije pripadnosti podklasa superklasi.

George, Srikanth, Petry i Buckles [98] proširuju prethodni model uvodeći teoriju verovatnoće. Neodređenost i neizvesnost su zastupljene u klasnoj hijerarhiji. Stepem uključivanja objekata u nadtip ili superklasu zavisi od fazi vrednosti atributa i stepena verovatnoće odnosno pripadnosti generičkom tipu. Na taj način se vrši apstrakcija u kojoj se skup sličnih tipova objekata tretira kao generički tip objekta (nadtip, superklasa). "Slični" tipovi objekata su oni tipovi koji imaju neke zajedničke attribute, veze i/ili operacije. Specijalizacija je inverzni postupak u kome se za neki tip objekta, definišu njegovi podtipovi, koji imaju neke njima specifične attribute, veze i/ili operacije. Na primer, skup tipova Radnik, Student, Penzioner, Dete može se predstaviti generičkim tipom Gradjanin, a za svaku od klasa se definiše stepen pripadnosti višoj klasi. Isto tako se tip Radnik, može specijalizovati u podtipove Vozač, Inženjer, Lekar i druge sa određenim stepenom verovatnoće. Ova ideja prisutna je i u naučnim radovima Zong M. Ma, čiji se doprinos u ovoj oblasti smatra jako značajnim. Na taj način su sve velike ideje objektno-orijentisanih modela kao što su objekti, klase, veze objekti-klase, podklase, superklase,

nasleđivanje itd. inkorporirane u fazi informaciono okruženje. Pravila i operacije koje važe u objektno-orijentisanim bazama podataka prisutna su i u generičkim modelima FOODB. U fazi generalizacionoj hijerarhiji objekata, takođe, važi pravilo nasleđivanja osobina i pravilo nasleđivanja kao i operacija, kao što su:

- Podtipovi nasledjuju sve attribute i veze svoga nadtipa.
- Podtipovi nasledjuju sve operacije svoga nadtipa.

Poslednjih godina učinjeni su neki napori na uspostavljanju jedinstvenog okvira za fazi objektno-orijentisane baze podataka koji su zasnovani na standardu koji predlaže posebno formirana grupa za upravljanje objektnim podacima (Object Data Management Group – ODMG). U tom smislu predstavljen je koncept ‘Nivo – 2 fazi skupova’ koji se bavi jedinstvenom i najpovoljnijom zastupljenošću crisp i fazi informacija prilikom predstavljanja realnog sveta i podataka. Cilj ove grupacije je da ilustruje i diskutuje kako podaci iz realnog sveta mogu biti generalizovani kako bi se sa njima upravljalo na najpovoljniji način.

3.4. Fazi sistemi za upravljanje relacionim bazama podataka

Sistemi za upravljanje bazama podataka (DBMS) kao što su SQL Server i Oracle koriste se za implementiranje klasičnih relacionih baza podataka. Međutim, ovi sistemi trenutno ne podržavaju i mehanizme za rad sa fazi relacionim bazama podataka. Naravno, to ne znači da ne postoje različiti teorijski modeli i koncepti koji analiziraju proširenje postojećih sistema za upravljanje bazama podataka, a koje bi trebalo da omoguće implementaciju fazi relacionih baza podataka. U literaturi se mogu naći različiti teorijski pristupi. Nažalost, posledice ovih radova na praktičnom planu su zanemarljive. Fazi sistemi za upravljanje relacionim bazama podataka (FRDBMS) proširuju mogućnosti klasičnih RDBMS kako bi se omogućilo upravljanje fazi informacijama. Suštinska ideja u ovim radovima je proširenje SQL upitnog jezika i dodavanje dodatnog sloja/layer-a na postojeći RDBMS. Medina, Pons i ostali [99] predlažu teorijski model i predstavljaju ideje koje za cilj imaju izgradnju sistema i operativnu implementaciju GEFRED modela. Tačnije, oni predlažu FRDBMS arhitekturu zasnovanu na klijent/server pristupu i modulima koji omogućavaju manipulaciju podacima (fazi i crisp): pristup podacima, generisanje fleskibilnih upita, razvoj različitih aplikacija i podrška odlučivanju, replikaciju podataka iz drugih baza. Ovaj model nazivaju Fazi SQL Server. Fazi SQL server inkorporira elemente i algoritme koji omogućavaju parsiranje relacione šeme i konverziju fazi SQL upita u standardnu SQL sintaksu. Touzi i Hassine [100] proširuju prethodni model, oslanjajući se više na Oracle-ov sistem za upravljanje bazama podataka. Ovaj FRDBMS nudi sve funkcionalnosti klasičnog sistema za upravljanje podacima (RDBMS). Cilj ovih radova je da se omogući korisniku jednostavno upravljanje i manipulacija sa fazi informacijama.

3.5. Fazi funkcionalne zavisnosti

Važnu ulogu u logičkom dizajnu baza podataka predstavlja integritet podataka. Među ovim, najinteresantnije su zavisnosti podataka. Jedna od najvažnijih zavisnosti podataka jeste funkcionalna zavisnost. One dovode u relaciju vrednosti jednoga skupa atributa sa vrednostima drugoga skupa atributa. Ukoliko funkcionalne zavisnosti uključimo u fazi model baze podataka [68,73], tada nju nije moguće direktno primeniti u tom model jer se ona temelji na konceptu jednakosti. Pošto ne postoji jasan način potvrđivanja kada su dve neprecizne vrednosti jednake, tada definicija funkcionalnih zavisnosti mora da se proširi

odnosno generalizuje. Ta proširena/generalizovana verzija funkcionalnih zavisnosti naziva se fazi funkcionalna zavisnost [2,6,14,47,50,51,68,74,80]. Postoji više različitih definicija fazi funkcionalnih zavisnosti, koje su dobivene kao rezultat upotrebe fazi logike u klasičnoj funkcionalnoj zavisnosti i sve takve definicije fazi funkcionalnih zavisnosti asocirane su sa datim okvirom u fazi bazama podataka. Zbog toga su jedino primenljive unutar datoga okvira, iako postoje osnovne odnosno generalne osobine i karakteristike koje su potrebne da fazi funkcionalne zavisnosti poseduju.

U klasičnim relacionim bazama podataka, funkcionalne zavisnosti određuju kada vrednost neke n-torke jednoga skupa atributa X, jedinstveno određuje njegove vrednosti u drugom skupu atributa Y ili strožije rečeno:

Definicija 3.17. Neka je $R(A_1, A_2, \dots, A_n)$ relaciona šema na domenima D_1, D_2, \dots, D_n sa $\text{Dom}(A_i) = D_i$ i neka su X i Y podskupovi skupa atributa $U = \{A_1, A_2, \dots, A_n\}$ tj. $X, Y \subseteq U$ i neka je r relacija od R, $r \subseteq D_1 \times \dots \times D_n$. Kažemo da relacija r zadovoljava funkcionalnu zavisnost $X \rightarrow Y$, ako za svake dve n - torke t i t' $\subseteq r$, za koje vredi $t(X) = t'(X)$ implicira da vredi i $t(Y) = t'(Y)$.

U opštem slučaju kada vrednosti atributa ne poprimaju samo atomske elemente domena, već i mogućnosti distribucije, tada stepen $X \rightarrow Y$ nije nužno potrebno da bude 1, već može da bude u jediničnom intervalu [0,1]. Tako da se prirodno nameću sledeća pitanja. Kako odrediti $t(X) = t'(X)$ i $t(Y) = t'(Y)$ kada su t(X), t'(X), t(Y) i t'(Y) sve neprecizne vrednosti predstavljene kao mogućnosti distribucije. Kao drugo, kako odrediti stepen propozicije Ako je t(X) = t'(X) tada je t(Y) = t'(Y) kada su t(X) = t'(X) i t(Y) = t'(Y) delimično istiniti sa stepenom pripadnosti iz intervala [0,1]. I na kraju, kako proceniti stepen od $X \rightarrow Y$ ako različiti parovi n - torki daju različite istinitne vrednosti za propoziciju Ako ... tada.

Otuda proizilazi da su ova pitanja povezana sa problemima fazi bliskosti podataka, fazi logičke implikacije i fazi i (and) operatora. Sada definišimo fazi funkcionalne zavisnosti na sledeći način.

Definicija 3.18. Neka je $R(A_1, A_2, \dots, A_n)$ relaciona šema na domenima D_1, D_2, \dots, D_n sa $\text{Dom}(A_i) = D_i$ i neka su X i Y podskupovi skupa atributa $U = \{A_1, A_2, \dots, A_n\}$ tj. $X, Y \subseteq U$ i neka je r relacija od R, $r \subseteq \Pi(D_1) \times \dots \times \Pi(D_n)$, gde je $\Pi(D_i) = \{\pi \mid \pi \text{ je moguća distribucija od } A_i \text{ na } D_i\}$, $i=1,2,\dots,n$. Kažemo da X fazi funkcionalno određuje Y sa stepenom θ , označeno kao $X \xrightarrow{\theta} Y$ ako i samo ako za $\forall r \in R$:

$$\min I(t(X)=_c t'(X), t(Y)=_c t'(Y)) \geq \theta, t, t' \in r.$$

gde je $\theta \in [0,1] =_c [0,1] \times [0,1] \rightarrow [0,1]$ mera bliskosti i $I: [0,1] \times [0,1] \rightarrow [0,1]$ je fazi operator implikacije.

Ova definicija daje generalnu formu fazi funkcionalnih zavisnosti, koja izražava semantiku da bliske Y vrednosti odgovaraju bliskim X vrednostima (sa stepenom θ). Ako se uzme odgovarajuća mera bliskosti za $=_c$ i odgovarajuća fazi implikaciju za I, tada se može i odrediti odgovarajuća forma za fazi funkcionalne zavisnosti. Na primer:

$$\text{Truth}(t(X)=_c t'(X)) = G_c(t(X)=_c t'(X)) (T),$$

$$\text{Truth}(t(Y)=_c t'(Y)) = G_c(t(Y)=_c t'(Y)) (T)$$

gde je:

$$\approx (t(A), t'(A)) = E_c(t(A), t'(A)) (T) = \sup \min (\pi_A(u), \pi'_A(v)), \quad u, v \in D_A, c_A \geq \alpha_A, \forall A \in X.$$

$$\approx (t(B), t'(B)) = E_c(t(B), t'(B)) (T) = \sup \min (\pi_B(w), \pi'_B(z)), \quad w, z \in D_B, c_B \geq \alpha_B, \forall B \in Y.$$

Ovde je $t(X) =_c t'(X)$ uzeto kao minimum svih bliskih vrednosti za sve atribute A u X, dok je $t(Y) =_c t'(Y)$ uzeto kao minimum svih bliskih vrednosti za sve atribute B u Y. Ako mogućnost distribucije na odgovarajućim vrednostima atributa nije normalizovana, tada za $\approx (t(A), t'(A))$ i $\approx (t(B), t'(B))$:

$$\approx (t(A), t'(A)) = \begin{cases} 1 & t(A) \text{ i } t'(A) \text{ su identicni} \\ \sup_{\substack{u, v \in D_A \\ C_A(u, v) \geq \alpha_A}} \min(\pi_A(u), \pi'_A(v)), & \text{inace} \end{cases}$$

$$\approx (t(B), t'(B)) = \begin{cases} 1 & t(B) \text{ i } t'(B) \text{ su identicni} \\ \sup_{\substack{w, z \in D_B \\ C_B(w, z) \geq \alpha_B}} \min(\pi_B(w), \pi'_B(z)), & \text{inace} \end{cases}$$

3.5.1. Pravila izvođenja za fazi funkcionalne zavisnosti

U klasičnom relacionom modelu, često je potrebno, da na osnovu datog skupa zavisnih podataka, odrediti i neke druge zavisnosti nad istom bazom podataka, a koje su posledica već datog skupa zavisnosti. Tako u klasičnim relacionim bazama podataka, postoje tri pravila izvođenja poznata kao *Armstrongove aksiome*, a koje se koriste da iz datih funkcionalnih zavisnosti izvodimo nove funkcionalne zavisnosti. Sada dajemo ova pravila izvođenja [56]. Neka su X, Y, Z i W proizvoljan skup atributa.

A₁: Ako je $Y \subseteq X$ tada postoji funkcionalna zavisnost $X \rightarrow Y$.

A₂: Ako važi funkcionalna zavisnost $X \rightarrow Y$ tada važi i funkcionalna zavisnost $XZ \rightarrow YZ$.

A₃: Ako važe funkcionalne zavisnosti $X \rightarrow Y$ i $Y \rightarrow Z$ tada važi i funkcionalna zavisnost $X \rightarrow Z$.

Iz osnovnih osobina funkcionalnih zavisnosti mogu se izvesti i sledeće osobine:

A₄: Ako važe funkcionalne zavisnosti $X \rightarrow Z$ i $X \rightarrow Y$ tada važi i funkcionalna zavisnost $X \rightarrow YZ$.

A₅: Ako važe funkcionalne zavisnosti $X \rightarrow Y$ i $WY \rightarrow Z$ tada važi i funkcionalna zavisnost $XW \rightarrow Z$.

A₆: Ako važi funkcionalna zavisnost $X \rightarrow Y$ i ako je $Z \subseteq Y$ tada važi funkcionalna zavisnost $X \rightarrow Z$.

Fazi funkcionalne zavisnosti izražavaju relacije između skupa atributa. Ako je data relaciona šema na skupu atributa U i fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$ koja je zadovoljena u R, tada $X \xrightarrow{\theta} Y$ označava da je zadovoljena u svim relacijama od R. Čak šta više, za dati skup F fazi funkcionalnih zavisnosti, koji je zadovoljen u F, ne garantuje nam samo da su ove fazi funkcionalne zavisnosti u F zadovoljene u svim relacijama od R, već nam i garantuje da je i svaka fazi funkcionalna zavisnost logički implicirana sa F

zadovoljena u svim relacijama od R. Tako, na primer, ako znamo da vredi $X \xrightarrow{\theta} Y$ u R, tada možemo očekivati da u R vredi i $X \xrightarrow{\alpha} Y$ za $\alpha \leq \theta$ iz $[0,1]$. Takođe, ako vredi $X \xrightarrow{\theta} Y$ tada možemo očekivati da XZ funkcionalno određuje YZ do stepena koji je najmanje jednak θ , za sve relacije od R. Dalje, ako vredi $X \xrightarrow{\alpha} Y$ i $Y \xrightarrow{\beta} Z$ u R tada možemo očekivati da X funkcionalno određuje Z sa određenim stepenom λ iz $[0,1]$. Intuitivno možemo pretpostaviti da je $\lambda = \min(\alpha, \beta)$ iz sledećih razloga:

- a) Ako X funkcionalno određuje Y sa stepenom θ i Y funkcionalno određuje Z sa stepenom θ , tada možemo očekivati da i X funkcionalno određuje Z sa istim stepenom.
- b) Ako X funkcionalno određuje Y sa stepenom $\alpha \geq \theta$ i Y funkcionalno određuje Z sa stepenom $\beta \geq \theta$, tada očekujemo da X funkcionalno određuje Z sa stepenom θ . Na taj način, prethodno date tri Armstrongove aksiome u klasičnoj relacionoj teoriji, proširene su sa tri pravila izvođenja za fazi funkcionalne zavisnosti:

A₁': Ako je $Y \subseteq X$ tada postoji fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$ za $\forall \theta$.

A₂': Ako važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$ tada važi i fazi funkcionalna zavisnost $XZ \xrightarrow{\theta} YZ$.

A₃': Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y$ i $Y \xrightarrow{\beta} Z$ tada važi i fazi funkcionalna zavisnost $X \xrightarrow{\lambda} Z$ sa $\lambda = \min(\alpha, \beta)$, gde je $\lambda, \alpha, \beta, \theta \in [0,1]$.

Iz osnovnih osobina fazi funkcionalnih zavisnosti mogu se izvesti i sledeće osobine:

A₄': Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y$ i $X \xrightarrow{\beta} Z$ tada važi i fazi funkcionalna zavisnost $X \xrightarrow{\lambda} YZ$ sa $\lambda = \min(\alpha, \beta)$.

A₅': Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y$ i $WY \xrightarrow{\beta} Z$ tada važi i fazi funkcionalna zavisnost $XW \xrightarrow{\lambda} Z$ sa $\lambda = \min(\alpha, \beta)$.

A₆': Ako važi fazi funkcionalna zavisnost $X \xrightarrow{\alpha} Y$ i ako je $Z \subseteq Y$ tada važi fazi funkcionalna zavisnost $X \xrightarrow{\alpha} Z$.

A₇': Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y$ tada važi $X \xrightarrow{\beta} Y$ za $\forall \beta \leq \alpha$.

Osnovni zahtev koji moraju ispunjavati pravila izvođenja jeste da su ona korektna. To je osigurano sledećom teoremom:

Teorema 1. Neka je F skup fazi funkcionalnih zavisnosti na relacionoj šemi R(U) i neka je P skup fazi funkcionalnih zavisnosti koji su logički implicirani iz F. Tada je aksiomatski sistem koga čine aksiome A₁', A₂', A₃' korekatan.

A₁': Ako je $Y \subseteq X$ tada postoji fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y \in F^+$ za $\forall \theta$ iz $[0,1]$.

A'₂: Ako važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y \in F^+$ tada važi i fazi funkcionalna zavisnost $XZ \xrightarrow{\theta} YZ \in F^+$.

A'₃: Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y \in F^+$ i $Y \xrightarrow{\beta} Z \in F^+$ tada važi fazi funkcionalna zavisnost $X \xrightarrow{\lambda} Z \in F^+$ sa $\lambda = \min(\alpha, \beta)$, gde je $\lambda, \alpha, \beta, \theta \in [0, 1]$.

Kao posledica ovih pravila izvođenja, vrede sledeći rezultati (zapisani kao A'₄, A'₅, A'₆, A'₇, A'₈):

A'₄: Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y \in F^+$ i $X \xrightarrow{\beta} Z \in F^+$ tada važi i fazi funkcionalna zavisnost $X \xrightarrow{\lambda} YZ \in F^+$ sa $\lambda = \min(\alpha, \beta)$.

A'₅: Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y \in F^+$ i $WY \xrightarrow{\beta} Z \in F^+$ tada važi i fazi funkcionalna zavisnost $XW \xrightarrow{\lambda} Z \in F^+$ sa $\lambda = \min(\alpha, \beta)$.

A'₆: Ako važi fazi funkcionalna zavisnost $X \xrightarrow{\alpha} Y \in F^+$ i ako je $Z \subseteq Y$ tada važi fazi funkcionalna zavisnost $X \xrightarrow{\alpha} Z \in F^+$.

A'₇: Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{\alpha} Y \in F^+$ tada važi $X \xrightarrow{\beta} Y \in F^+$ za $\forall \beta \leq \alpha$.

A'₈: $X \xrightarrow{\theta} Y_1, Y_2, \dots, Y_k \in F^+$ ako i samo ako $X \xrightarrow{\theta} Y_i \in F^+$ za $i=1, 2, \dots, k$.

Aksiomatski sistem je skup pravila zaključivanja $\{A'_1, A'_2 \text{ i } A'_3\}$ koji se koristi za izvođenje novih fazi funkcionalnih zavisnosti koje su logička posledica skupa fazi funkcionalnih zavisnosti F , za neku relaciju šemu R . Neka je F^+ skup fazi funkcionalnih zavisnosti, koji je logički impliciran sa skupom F i neka je F^A skup fazi funkcionalnih zavisnosti izveden pomoću aksiomatskog sistema iz skupa F . Tada postoje dve važne činjenice. Prva se odnosi kao soundness (adekvatnost) $F^A \subseteq F^+$ aksiomatskoga sistema, a druga se odnosi na completeness (potpunost) $F^+ \subseteq F^A$ aksiomatskog sistema. Sledeća teorema pokazuje da je dati aksiomatski sistem sound i complete [11]*.

Teorema 2. Aksiomatski sistem koga čine A'_1, A'_2 i A'_3 je sound i complete. Tako je $F^+ = F^A$.

3.5.2. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Raju i Majumdar-a

Fazi relacioni model Raju i Majumdara [53] dopušta da komponente n -torki uzimaju kako atomik tako i neatomik vrednosti. U zavisnosti od kompleksnosti domena, oni fazi relaciju dele na dve kategorije. U prvom tipu fazi relacije, domen može biti samo fazi skup ili klasični skup. U drugom tipu fazi relacije, omogućava se da svaki domen atributa može biti klasičan skup, fazi skup ili skup fazi podskupova (ili mogućnost distribucije). Svaka fazi relacija u ovom modelu reprezentovana je tabelom koja ima dodatnu kolonu, koja određuje vrednost pripadnosti date n -torke odgovarajućoj relaciji. Tako oni definišu fazi relacioni model na sledeći način:

Definicija 3.19. Neka je $R(A_1, \dots, A_n, \mu_r)$ fazi relaciona šema. Fazi relacija r na R je fazi skup ili skup fazi podskupova na domenu $\text{dom}(A_1) \times \dots \times \text{dom}(A_n)$ koji je određen sa funkcijom pripadnosti: $\mu_r(t_i): \text{dom}(A_1) \times \dots \times \text{dom}(A_n) \rightarrow [0, 1]$.

Prvi tip fazi relacije izražava se kao:

$$r = \sum_{\forall t_i \in r} \mu(t_i) / t_i$$

$$r = \sum_{\forall t_i \in r} \mu(d_{i1}, \dots, d_{in}) / (d_{i1}, \dots, d_{in})$$

gde je: $d_{ij} = \mu(a_{ij}) / a_{ij}$ i $\mu(d_{i1}, \dots, d_{in}) = \min(\mu(a_{i1}), \dots, \mu(a_{in}))$, $a_{ij} \in \text{dom}(A_j)$ i $j=1,2,\dots,n$.

U drugom tipu fazi relacije svaka komponenta d_{ij} od t_i data je kao:

$$d_{ij} = \{ \mu(a_{ij1}) / a_{ij1}, \dots, \mu(a_{ijk}) / a_{ijk} \}$$

gde je $a_{ijl} \in \text{dom}(A_j)$, $l=1,2,\dots,k$ i $\mu(a_{ijl})$ stepen pripadnosti od a_{ijl} .

Tako pripadna vrednost $\mu_r(t_i)$ mora zadovoljavati sledeću jednakost:

$$\mu_r(t_i) = \min(\mu(d_{i1}), \dots, \mu(d_{in}))$$

gde je $\mu(d_{ij}) = \max(\mu(a_{ij1}), \dots, \mu(a_{ijk}))$ za svako $a_{ijl} \in d_{ij}$ i $l=1,2,\dots,k$

R a j u i M a j u m d a r definišu fazi funkcionalne zavisnosti na sledeći način:

Definicija 3.20. Fazi funkcionalna zavisnost $X \rightarrow Y$, $X, Y \subseteq R$, vredni u fazi relaciji r na R , ako za sve n -torke t_1 i t_2 iz r ($\mu_r(t_i) > 0$, $i=1,2$), vredni:

$$\mu_{EQ}(t_1[X], t_2[X]) \leq \mu_{EQ}(t_1[Y], t_2[Y])$$

gde je Equal(EQ) fazi relacija bliskosti na univerzalnom skupu U i definisana je kao fazi podskup na $U \times U$ i gde je μ_{EQ} funkcija pripadnosti koja zadovoljava sledeće uslove: Za svako $a, b \in U$, $\mu_{EQ}(a, a) = 1$ (refleksivna) i $\mu_{EQ}(a, b) = \mu_{EQ}(b, a)$ (simetrična).

Pravila zaključivanja u modelu Raju i Majumdar.

FF1: Refleksivnost: Ako je $Y \subseteq X$ tada postoji fazi funkcionalna zavisnost $X \xrightarrow{RM} Y$

FF2: Aditivnost: Ako važi fazi funkcionalna zavisnost $X \xrightarrow{RM} Y$ tada važi i fazi funkcionalna zavisnost $XZ \xrightarrow{RM} YZ$.

FF3: Tranzitivnost. Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{RM} Y$ i $Y \xrightarrow{RM} Z$ tada važi i fazi funkcionalna zavisnost $X \xrightarrow{RM} Z$.

Na osnovu ovih aksioma slede dodatna pravila:

FF4: Unija: Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{RM} Y$ i $X \xrightarrow{RM} Z$ tada važi i fazi funkcionalna zavisnost $X \xrightarrow{RM} YZ$.

FF5: Dekompozicija. Ako važi fazi funkcionalna zavisnost $X \xrightarrow{RM} YZ$ tada važe i fazi funkcionalne zavisnosti $X \xrightarrow{RM} Y$ i $X \xrightarrow{RM} Z$.

FF6: Pseudotranzitivnost. Ako važe fazi funkcionalne zavisnosti $X \xrightarrow{RM} Y$ i $YW \xrightarrow{RM} Z$ tada važi i fazi funkcionalna zavisnost $XW \xrightarrow{RM} Z$.

3.5.3. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Shenoi-Melton

Njihova strategija za izražavanje nepreciznih informacija temelji se na ideji sakupljanja istovrsnih elemenata konačnog domena u blokove elemenata koji se ne razlikuju sa određenim nivoom preciznosti na tom domenu. Ova ideja je izražena u odnosu na klase ekvivalencije na domenu particije. Particija domena D_k je skup nepraznih disjunktih podskupova ili klasa ekvivalencije od D_k sa osobinom da svaki element iz D_k je u tačno jednoj klasi ekvivalencije. [8,65] Particionisanje domena na klase ekvivalencije je ključ za očuvanje nekoliko važnih osobina u klasičnom modelu. Na osnovu toga oni definišu redundansu na sledeći način:

Definicija 3.21. Neka su t i t' dve fazi n - torke. Komponente t_k i t'_k su α_k - redundantne označeno sa $t_k \approx_{\alpha_k} t'_k$, kada su t_k i t'_k podskupovi iste klase ekvivalencije za α_k - particije temporalnog domena D_k .

Fazi relaciju definišu na sledeći način:

Definicija 3.22. Neka je R relaciona šema sa atributima (A_1, \dots, A_n) i pridruženim particijama sa nivoom $(\alpha_1, \dots, \alpha_n)$. Neka je $r(R) \subseteq 2^{D^1} \times 2^{D^2} \times \dots \times 2^{D^n}$. Tada je r fazi relacija u odnosu na fazi relacionu šemu R i temporalni domen D'_1, \dots, D'_m , ako je r skup neredundantnih fazi n - torki sa respektom $\alpha_1, \dots, \alpha_n$ - nivoom particija na D'_1, \dots, D'_m , respektivno.

Izražavanje redundancije u klasičnim funkcionalnim zavisnostima omogućilo je i definisanje fazi funkcionalne zavisnosti u fazi relacionim bazama podataka. Pošto je fazi redundancija ograničena sa nivoom preciznosti, tada ovaj nivo preciznosti mora biti takođe deo fazi funkcionalne zavisnosti. Tako, fazi funkcionalna zavisnost vredi kada god postoji redundancija sa određenim nivoom preciznosti na levoj strani atributa, tada postoji redundancija na desnoj strani odgovarajućih atributa, sa određenim nivoom preciznosti. Tako imamo:

Definicija 3.23. Neka je R fazi relaciona šema sa atributima U i nivoom particije α_U . Neka su X i Y skupovi atributa u U sa asociiranim nivoom $\alpha_X = (\alpha_p, \dots, \alpha_q)$ i $\alpha_Y = (\alpha_r, \dots, \alpha_s)$ u α_U . Neka je r fazi relacija sa temporalnim domenima D'_p, \dots, D'_q , i D'_r, \dots, D'_s , za skupove atributa X i Y respektivno. Relacija r zadovoljava fazi funkcionalnu zavisnost iz X u Y sa nivoom (α_X, α_Y) na datim particijama, kada za bilo koje n - torke $t, t' \in r(R)$ vredi:

$$t \approx_{\alpha_X} t' \text{ implicira } t \approx_{\alpha_Y} t'$$

Na kraju oni dokazuju da ovako definisane fazi funkcionalne zavisnosti zadovoljavaju Armstrong-ove aksiome.

3.5.4. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Saxena i Tyagi-a

U ovom fazi relacionom modelu fazi vrednostima atributa dopušta se mogućnost da atribut nije primenljiv za dati objekat. Da bi bili u mogućnosti da rade sa neodređenim vrednostima podataka, oni definišu fazi relaciju na sledeći način.

Neka je 2^{A_i} , $i = 1, 2, \dots, n$ skup fazi podskupova na domenu $\text{dom } A_i \cup \{e\}$, gde je e ekstra element koji dopušta mogućnost da fazi vrednost atributa nije primenjena za dati objekat, takav da za svaki skup $a \in 2^{A_i}$, funkcija pripadnosti od a zadovoljava uslov $\mu_a(e) = 0$ ili $\mu_a(e) = 1$ skupa sa $\mu_a(u) = 0$ za svako u . Tada je fazi relacija r na $R = (A_1, \dots, A_n)$ definisana kao fazi podskup od Dekartovog proizvoda $2^{A_1} \times 2^{A_2} \times \dots \times 2^{A_n} = 2^R$, okarakterisana sa funkcijom pripadnosti:

$$\mu_r: 2^R \rightarrow [0,1].$$

Svaka n - toraka $t = (a_1, \dots, a_n)$ gde je $a_i \in 2^{A_i}$, $i = 1, 2, \dots, n$ u r može se posmatrati kao mogućnost distribucije za $D = (\text{dom } A_1 \cup \{e\}) \times \dots \times (\text{dom } A_n \cup \{e\})$ kao:

$$\text{Poss}(t[A_1] = u_1, \dots, t[A_n] = u_n) = \min \{ \mu_r(t), \mu_{a_1}(u_1), \dots, \mu_{a_n}(u_n) \},$$

gde je $u_i \in A_i, \cup \{e\}, i = 1 \dots n$. [61]

U ovome fazi relacionom modelu sličnost između elemenata datoga domena definiše se na sledeći način.

$$\mu_{EQ}(a_1, a_2) = \min_{u \in \text{dom} A} \psi(\mu_{a_1}(u), \mu_{a_2}(u))$$

Ovde je fazi relacija EQ na $2^A - \{\emptyset\}$, gde je sa \emptyset označeno da za atribut A, fazi skup $a \in 2^A$ je takav da je $\mu_a(e) = 1$ i $\mu_a(u) = 0$ za $u \neq e$, je definisana kao fazi podskup $(2^A - \{\emptyset\}) \times (2^A - \{\emptyset\})$ tako da je njena funkcija pripadnosti definisana kao $\mu_{EQ}: (2^A - \{\emptyset\}) \times (2^A - \{\emptyset\}) \rightarrow [0,1]$ i koja zadovoljava $\mu_{EQ}(a, a) = 1$ (refleksivna), $a \in 2^A - \{\emptyset\}$ i $\mu_{EQ}(a_1, a_2) = \mu_{EQ}(a_2, a_1)$ (simetrična), $a_1, a_2 \in 2^A - \{\emptyset\}$. ψ je fazi relacija sličnosti.

Na osnovu prethodnih razmatranja oni definišu fazi funkcionalne zavisnosti.

Definicija 3.24. Fazi funkcionalna zavisnost $X \rightarrow Y$, $X, Y \in R$, vredi u fazi relaciji r na R , ako za bilo koje n-torke t_1 i t_2 iz r $\mu_r(t_i) > 0, i = 1, 2$ vredi $\mu_{EQ}(t_1[X], t_2[X]) > 0$, $t_1[Y] = t_2[Y] = \emptyset$ ili postoji neprazan skup $Y' \subseteq Y$ takav da je $t_1[A] = \emptyset = t_2[A]$ za svako $A \in Y'$, $t_1[Y - Y'] = t_2[Y - Y'] = \emptyset$ i $\mu_{EQ}(t_1[X], t_2[X]) \leq \mu_{EQ}(t_1[Y'], t_2[Y'])$.

Uvodeći pravila izvođenja, data niže, oni dokazuju da su ta pravila sound i complete.

FF1': Refleksivnost: Ako je $Y \subseteq X$ tada postoji fazi funkcionalna zavisnost $X \rightarrow Y$,

FF2': Aditivnost: Ako važi fazi funkcionalna zavisnost $X \rightarrow Y$ i $V \subseteq W$ tada važi i fazi funkcionalna zavisnost $XW \rightarrow YV$.

FF3': Unija: Ako važe fazi funkcionalne zavisnosti $X \rightarrow Y$ i $X \rightarrow Z$ tada važi i fazi funkcionalna zavisnost $X \rightarrow YZ$.

FF4': Dekompozicija: Ako važi fazi funkcionalna zavisnost $X \rightarrow YZ$ tada važe i fazi funkcionalne zavisnosti $X \rightarrow Y$ i $X \rightarrow Z$.

3.5.5. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Dubois-Prade

Dubois i Prade [82,83] fazi funkcionalne zavisnosti uvode na sledeći način. Ako su vrednosti atributa A za n - torke t i t' jednake tada i vrednosti atributa B za t i t' ne smeju biti daleko jedna od druge. Ovu ideju oni modeliraju sa ispoljavanjem fazi relacije bliskosti P (koja je refleksivna $\forall d, \mu_p(d, d) = 1$ i simetrična $\forall d, d' \mu_p(d, d') = \mu_p(d', d)$), a koja je definisana na oblasti značenja atributa B.

$$\text{Ako je } t(A) = t'(A) \text{ tada je } \mu_B(t(B), t'(B)) > \theta$$

gde θ predstavlja zadati prag.

Smatrajući da je $t(B)$ i $t'(B)$ određeno u terminu funkcije raspodele mogućnosti, imamo:

$$\text{Ako je } t(A) = t'(A) \text{ tada je } \Pi(t(B) \approx_p t'(B)) > \theta$$

Ova mogućnost data je sa:

$$\Pi(t(B) \approx_p t'(B)) = \sup_{v,w \in D_B \times D_B} \min(\mu_P(v,w), \pi_{t(B)}(v), \pi_{t'(B)}(v))$$

gde je π_B funkcija raspodele mogućnosti koja ograničava moguće značenje atributa B za n-torku t, a μ_P funkcija pripadnosti fazi relacije bliskosti P.

3.5.6. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Wei-Yi Liu

U ovome radu najpre se definiše koncept semantičke distance između dve fazi vrednosti atributa, dok su fazi funkcionalne zavisnosti predstavljene sa fazi semantičkom distancom. Na osnovu semantičke blizine daje se definicija fazi funkcionalnih zavisnosti [7,11,47].

Stepen bliskosti između dve fazi vrednosti je opisan pomoću semantičke blizine. Semantička blizina zasnovana je na konceptu intervala i označavamo je sa $SD(f_1, f_2)$, pri čemu je $0 \leq SD(f_1, f_2) \leq 1$. Sledeće karakteristike trebalo bi da budu zadovoljene:

Neka su $f_1=[a_1, b_1]$, $f_2=[a_2, b_2]$, $g_1=[c_1, d_1]$, $g_2=[c_2, d_2]$. Tada važi:

1. $SD(f_1, f_2)=1$ ako i samo ako je $a_1 = a_2 = b_1 = b_2$,
2. $SD(f_1, f_2)=0$ ako i samo ako $f_1 \cap f_2 = \emptyset$,
3. Ako je $a_1 = a_2$, $b_1 = b_2$, $c_1 = c_2$, $d_1 = d_2$ i $|d_1 - c_1| > |b_1 - a_1|$ onda je i $SD(f_1, f_2) \geq SD(g_1, g_2)$,
4. Ako je $|a_2 - b_2| = |a_1 - b_1|$ i $f_1 \cap g_1 \geq f_2 \cap g_1$ onda je i $SD(f_1, g_1) \geq SD(f_2, g_1)$.

Semantičku distancu (blizinu) računamo po sledećoj formuli:

$$SD(f_1, f_2) = \frac{\|f_1 \cap f_2\|}{\|f_1 \cup f_2\| - \|f_1 \cap f_2\|} / \alpha$$

gde je $\|h\|$ modularno sa odgovarajućim intervalom:

$$\|h\| = \begin{cases} 0, & h \neq 0 \\ \delta & h = [a, a] \\ |b - a| & h = [a, b] \\ \alpha & h = \infty \end{cases}$$

α je koeficijent takav da je $\alpha \geq \|f_1 \cup f_2\|$, a δ veoma mali broj. Npr. $\delta = \alpha / 10.000$ se obično uzima u konkretnim primerima.

Ako su $t_1 = (x_{11}, x_{12}, \dots, x_{1n})$ i $t_2 = (x_{21}, x_{22}, \dots, x_{2n})$ dve n-torke u relaciji, onda se semantička blizina između njih označava kao $SD(t_1, t_2)$ i računa kao $SD(t_1, t_2) = \min\{SD(x_{1i}, x_{2i})\}$.

Semantička distanca $SD(f_1, f_2)$ od dve fazi vrednosti $f_1(X)$ i $f_2(X)$ definisana je pomoću neke norme $|f_1(X) - f_2(X)|$. Npr. $SD(f_1, f_2) = \max |f_1(X) - f_2(X)|$, $x \in \text{dom}(A_i)$. Komplement od $SD(f_1, f_2)$ u oznaci $SS(f_1, f_2)$ definisan je kao $SS(f_1, f_2) = 1 - SD(f_1, f_2)$. Semantička distanca fazi vrednosti opisanih na drugi način se može definisati na sličan način.

Prethodno navedene karakteristike prikazane su i na sledećim primerima:

1. Pretpostavimo da je $f_1=[10,10]$ i $f_2=[10,10]$, $\alpha=300$ onda je $SD(f_1, f_2) = \delta / \delta - \delta / \alpha = 1$.

2. Ako je $f_1=[2,5]$ i $f_2=[6,7]$ onda je semantička distanca $SD(f_1,f_2)= 0/5 - 0/\alpha = 0$.
3. Za $f_1=[2,5]$, $f_2=[2,5]$, $g_1=[2,8]$ i $g_2=[2,8]$ važi $SD(f_1,f_2)= 3/3 - 3/300 = 0.99$ i $SD(g_1,g_2)= 6/6 - 6/300 = 0.98$. Vidimo da je $SD(f_1,f_2) \geq SD(g_1,g_2)$.
4. Za $f_1=[3,5]$, $f_2=[2,4]$ i $g_1=[3,6]$ je $SD(f_1,g_1)= 2/3 - 2/300 = 0.66$ i $SD(f_2,g_1)= 1/4 - 1/300 = 0.25$. Imamo da je $SD(f_1,g_1) \geq SD(f_2,g_1)$.

Definicija fazi funkcionalne zavisnosti:

Definicija 3.25. Neka je r fazi relacija na relacionoj šemi $R(A_1, \dots, A_n)$ i neka je U univerzalan skup atributa A_1, \dots, A_n i neka su X i Y podskupovi od U . Kažemo da fazi relacija r zadovoljava fazi funkcionalnu zavisnost $X \rightarrow Y$, $X, Y \subseteq R$ ako za svaki par n - torke t_1 i t_2 u r vredi:

$$SS(t_1[X], t_2[X]) \leq SS(t_1[Y], t_2[Y])$$

Pravila izvođenja za fazi funkcionalne zavisnosti u modelu W e i - Y i L i u :

L_1 : Ako je $Y \subseteq X$ tada postoji fazi funkcionalna zavisnost $X \rightarrow Y$.

L_2 : Ako važi fazi funkcionalna zavisnost $X \rightarrow Y$ tada važi i fazi funkcionalna zavisnost $XZ \rightarrow YZ$.

L_3 : Ako važe fazi funkcionalne zavisnosti $X \rightarrow Y$ i $Y \rightarrow Z$ tada važi i fazi funkcionalna zavisnost $X \rightarrow Z$

Teorema 3. Pravila izvođenja $\{L_1, L_2, L_3\}$ su sound i complete.

3.5.7. Fazi funkcionalne zavisnosti i pravila izvođenja u modelu Sozati i Yazici

Jedna od najtemeljnijih definicija fazi funkcionalnih zavisnosti data je u radu [44]. U analizi fazi funkcionalnih zavisnosti se polazi od sledećeg razmatranja.

Ako je $t[X]$ sličan sa $t'[X]$, tada je i $t[Y]$ slično sa $t'[Y]$, Pri tome sličnost između Y vrednosti je veća ili jednaka sličnosti između X vrednosti. Takva zavisnost označava sa

$X \xrightarrow{F} Y$. Kao primer takve zavisnosti navodimo: "Službenici sa sličnim iskustvima moraju

imati slične plate". U ovom slučaju vrednosti atributa Iskustvo i Plata mogu biti neprecizne, dok je sama definicija zavisnosti precizno definisana. Međutim, ova definicija fazi funkcionalne zavisnosti, nije u potpunosti određena, u smislu da i zavisnost sama po sebi može biti neprecizna. Kao primer takve fazi funkcionalne zavisnosti je "Nivo inteligencije osobe *više ili manje* određuje njen uspeh" gde *više ili manje* u rečenici određuje nepreciznu zavisnost. Ako se zna da je osoba inteligentna, tada se može zaključiti da će ona biti uspešna. Međutim, ovaj nivo uspeha nije sasvim jasno ni precizno određen sa inteligencijom. Osoba može biti vrlo uspešna, manje uspešna itd. Tako ova zavisnost ne određuje precizan nivo uspeha, ali ipak osigurava minimalan nivo uspeha. Pretpostavimo da postoje dve osobe sa identičnim inteligencijama i pretpostavimo da je prva osoba vrlo uspešna. Odavde se ne može zaključiti da je i druga osoba vrlo uspešna, ali se može reći da nivo uspeha druge osobe će biti više ili manje sličan nivou uspeha prve osobe. Jedan način da se definiše ovakva zavisnost jeste da se prihvati lingvistička jačina zavisnosti kao prag. Na primer zavisnost "Službenici sa sličnim iskustvima moraju imati slične plate" ima lingvističku jačinu 1, dok zavisnost "Nivo inteligencije više ili manje određuje uspeh" ima lingvističku jačinu 0.7. Tako vrednost praga određuje jačinu zavisnosti i ona je zapisana u

formi $X \xrightarrow[\theta]{F} Y$, gde je θ jačina zavisnosti.

Koncept sličnosti je veoma važan u fazi relacionim bazama podataka jer nam omogućava da se proširi koncept identiteta u jasan model za rukovanje nepreciznim i neodređenim informacijama. U ‘‘Crisp’’ modelima baza podataka, dve n-torke su identične na posmatranom atributu ako i samo ako su im vrednosti tog atributa identične. U Fazi modelima baza podataka, sličnost vrednosti atributa se posmatra u odnosu na to koliko su te vrednosti usaglašene na posmatranom atributu. U ovom fazi modelu, sličnost između vrednosti atributa, definiše se kao saglasnost (conformant) dveju n-torki na atributu. Ovo je vrlo važan aspekt predložen od strane Bosc-a i ostalih koji služi za poređenje vrednosti nepreciznih, fazi atributa korištenjem koncepta usaglašenosti. Relacija usaglašenosti je simetrična, refleksivna i tranzitivna.

Definicija 3.26. Saglasnost atributa A_k definisana na domenu D_k za bilo koje n-torke t_i i t_j prezentovane u relaciji r i označene kao $C(A_k[t_i, t_j])$ data je sa:

$$C(A_k[t_i, t_j]) = \min \left\{ \min_{x \in d_i} \left\{ \max_{y \in d_j} \{s(x, y)\} \right\}, \min_{x \in d_j} \left\{ \max_{y \in d_i} \{s(x, y)\} \right\} \right\},$$

gde je d_i vrednost atributa A_k za n - torku t_i , d_j je vrednost atributa A_k za n - torku t_j , $s(x, y)$ je relacija sličnosti za vrednosti x i y , i s je preslikavanje svakog para elemenata iz domena D_k u interval $[0,1]$,

Ako je $C(A_k[t_i, t_j]) > \theta$, za relaciju r , za n - torke t_i i t_j kažemo da su saglasne na atributu A sa jačinom zavisnosti θ . Ova definicija proširena je na opisivanje bliskosti za dve n - torke na skupu atributa. Posmatrajmo sledeći primer:

Usaglašenost na atributu CENA za n-torke t_2 i t_5 može da se prikaže kao:

$$\begin{aligned} C(\text{CENA}[t_2, t_5]) &= \min \{ \min \{ \max \{s(S, S)\}, \max \{s(P, S)\} \}, \\ &\quad \min \{ \max \{s(S, S), s(S, P)\} \} \} \\ &= \min \{ \min \{ \max \{1\}, \max \{0.8\} \}, \\ &\quad \min \{ \max \{1, 0.8\} \} \} \\ &= \min \{ \min \{1, 0.8\}, \min \{1\} \} \\ &= \min \{0.8, 1\} = 0.8 \end{aligned}$$

Treba napomenuti da je moguća vrednost atributa CENA skromna (S) i prihvatljiva (P) i da je sličnost između njih 0.8. Primetimo da je usaglašenost između $\{\text{Skromna}, \text{Prihvatljiva}\}$ i $\{\text{Skromna}\}$ 0.8, dok je u slučaju $\{\text{Skromna}, \text{Prihvatljiva}\}$ i $\{\text{Skromna}, \text{Prihvatljiva}\}$ 1.

Relacioni model upoređuje dve n-torke na skupu atributa upoređujući vrednosti atributa. Definicija usaglašenosti se proširuje kako bi se opisala bliskost između dve n-torke na skupu atributa.

Definicija 3.27. Saglasnost na skupu atributa X za bilo koje dve n - torke t_i i t_j date u relaciji r i označena sa $C(X[t_i, t_j])$ data je je kao:

$$C(X[t_i, t_j]) = \min_{A_k \in X} \{C(A_k[t_i, t_j])\}$$

Saglasnost atributa je veoma važna osobina koja se koristi u dokazivanju i izvođenju pravila. Sada ćemo navesti osobine saglasnosti u vidu propozicija, a one će se koristiti u dokazivanju teorema.

Propozicija 1. Ako je $X \supseteq Y$, tada je $C(Y[t_i, t_j]) \geq C(X[t_i, t_j])$.

Dokaz. Pretpostavimo da je $Y = \{A_1, \dots, A_n\}$, $n \geq 0$ i $X = \{A_1, \dots, A_n, A_{n+1}, \dots, A_m\}$, $m \geq n$.

Tada je:

$$C(Y[t_i, t_j]) = \min \{C(A_1[t_i, t_j]), \dots, C(A_n[t_i, t_j])\}$$

$$C(X-Y[t_i, t_j]) = \min \{C(A_{n+1}[t_i, t_j]), \dots, C(A_m[t_i, t_j])\} \text{ i}$$

$$C(X[t_i, t_j]) = \min \{C(A_1[t_i, t_j]), \dots, C(A_m[t_i, t_j])\}$$

U tom slučaju možemo da pišemo da je:

$$C(X[t_i, t_j]) = \min \{C(Y[t_i, t_j]), C(X-Y[t_i, t_j])\}$$

Odakle proizilazi da je:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j]).$$

Analogno prethodnom, lako se dokazuju i sledeće propozicije:

Propozicija 2. Ako je $X(A_1, \dots, A_n)$ i $C(A_k[t_i, t_j]) \geq \theta$ za svako $k, k=1, 2, \dots, n$ tada je $C(X[t_i, t_j]) \geq \theta$ za bilo koje dve n-torke t_i i t_j u r .

Propozicija 3. Ako je $C(X[t_i, t_j]) \geq \theta$ i $C(X[t_j, t_k]) \geq \theta$ tada je i $C(X[t_i, t_k]) \geq \theta$ za svako t_i, t_j, t_k u r .

Definicija fazi funkcionalnih zavisnosti:

Definicija 3.28. Neka je r fazi relacija na relacionoj šemi $R(A_1, \dots, A_n)$ i neka je U univerzalan skup atributa A_1, \dots, A_n i neka su X i Y podskupovi od U . Kažemo da fazi relacija r zadovoljava fazi funkcionalnu zavisnost $X \xrightarrow[F]{\theta} Y$, ako za svaki par n-torki t_i i t_j u r vredi:

$$C(Y[t_i, t_j]) \geq \min(\theta, C(X[t_i, t_j]))$$

gde je θ je realan broj iz $[0, 1]$ i opisuje lingvističku jačinu zavisnosti.

Fazi funkcionalne zavisnosti su generalizacija funkcionalnih zavisnosti. Tu činjenicu bi trebalo i matematički dokazati. Da bismo dokazali konzistentnost, u definiciji FFD primenićemo originalne uslove relacione teorije funkcionalnih zavisnosti.

Teorama 4. Fazi funkcionalne zavisnosti su konzistentne.

Dokaz. Relaciona teorija definiše da u slučaju postojanja funkcionalnih zavisnosti $\theta=1$. Slučaj $\theta=0$ je takođe validan, ali u tom slučaju funkcionalna zavisnost ne postoji. Ako posmatramo slučaj $\theta=1$ i primenimo to na definiciju fazi funkcionalnih zavisnosti, u tom slučaju možemo da pišemo da je:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j])$$

Ako koncept saglasnosti zamenimo sa konceptom identiteta (relaciju sličnosti zamenimimo sa relacijom identiteta), onda $C(X[t_i, t_j])$ može imati vrednost 0 ili 1. Kada je $C(X[t_i, t_j]) = 1$, tada je $X[t_1] = X[t_2]$ jer su vrednosti atributa iste. Prema definiciji $C(Y[t_i, t_j]) \geq C(X[t_i, t_j])$, ako i samo ako je $C(Y[t_i, t_j]) \geq 1$, to jeste $Y[t_1] = Y[t_2]$.

Pravila zaključivanja za fazi funkcionalne zavisnosti u modelu Sozati-a i Yazici-ja:

P1. Ako je $X \xrightarrow[F]{\theta_1} Y$ i $\theta_1 \geq \theta_2$ tada je i $X \xrightarrow[F]{\theta_2} Y$ – Pravilo inkluzivnosti.

P2. Ako je $X \supseteq Y$, onda je $X \xrightarrow[F]{\theta} Y$ zadržana. – Pravilo refleksivnosti.

P3. $\{ X \xrightarrow[F]{\theta} Y \} \mid \Rightarrow XZ \xrightarrow[F]{\theta} YZ$ – Pravilo proširenja.

$$P4. \{X \xrightarrow[F]{\theta_1} Y, Y \xrightarrow[F]{\theta_2} Z\} \mid \Rightarrow X \xrightarrow[F]{\min(\theta_1, \theta_2)} Z - \text{Pravilo tranzitivnosti.}$$

Dodatna pravila izvođenja za fazi funkcionalne zavisnosti:

$$P5. \{X \xrightarrow[F]{\theta_1} Y, X \xrightarrow[F]{\theta_2} Z\} \mid \Rightarrow X \xrightarrow[F]{\min(\theta_1, \theta_2)} YZ - \text{Pravilo unije}$$

$$P6. \{X \xrightarrow[F]{\theta_1} Y, WY \xrightarrow[F]{\theta_2} Z\} \mid \Rightarrow WX \xrightarrow[F]{\min(\theta_1, \theta_2)} Z - \text{Pravilo pseudotranzitivnosti}$$

$$P7. \text{Ako vredi } X \xrightarrow[F]{\theta} Y \text{ i } Z \subseteq Y \text{ tada vredi } X \xrightarrow[F]{\theta} Z. - \text{Pravilo dekompozicije.}$$

Važno svojstvo fazi funkcionalnih zavisnosti jeste da ova pravila izvođenja su sound. To garantuje sledeća teorema.

Teorema 5. Pravila izvođenja P1, P2, P3, P4 su sound.

Da bi se mogla formulisati i dokazati teorema potpunosti uvodi se pojam zatvarača skupa fazi funkcionalnih zavisnosti.

Definicija 3.29. Neka je F skup fazi funkcionalnih zavisnosti na skupu atributa U . Zatvarač od F , zapisano kao F_+ , je najmanji skup svih fazi funkcionalnih zavisnosti je koji je zatvoren na primenu pravila u skupu $\{P1, P2, P3, P4\}$.

Definicija 3.30. Neka je F skup fazi funkcionalnih zavisnosti na skupu atributa U . Limit jačine zavisnosti od $X \xrightarrow[F]{\theta} Y$ je definisan kao realan broj θ_1 u $[0,1]$ takav da je:

$$X \xrightarrow[F]{\theta_1} Y \text{ u } F_+ \text{ i za svaki } \theta_1 > \theta' \text{ je } X \xrightarrow[F]{\theta'} Y \text{ je u } F_+.$$

Zatvarač skupa atributa

Definicija 3.31. Neka je F skup fazi funkcionalnih zavisnosti na skupu atributa U i neka je X podskup od U . Tada $X_+(\theta)$, zatvarač od X , je skup atributa A tako da je:

$$X \xrightarrow[F]{\theta} Y \text{ je iz } F_+.$$

Sada na osnovu ovih razmatranja sledi teorema [44].

Teorema 6. Pravila izvođenja P1, P2, P3, P4 su sound i complete.

3.5.8. Primena saglasnosti (conformance) u analizi fazi funkcionalnih zavisnosti

Za određivanje istinite vrednosti atributa koristi se definicija saglasnosti (conformance) [44], koja će poslužiti da se odredi kada je istinita vrednost atributa zadovoljiva.

Neka je r podrelacija na relacionoj šemi R , sa tačno dve n -torke t_1 i t_2 . Podrelacija r se može upotrebiti za definisanje istinitih vrednosti dodeljenih atributima u relacionoj šemi R , kada se oni uzimaju kao varijable. Sada dajemo definiciju za istinitu vrednost atributa.

Definicija 3.32. Neka je $R(A_1, \dots, A_n)$ relaciona šema i neka je $r(R) = \{t_1, t_2\}$ dvoelementna poderlacija. Relaciji r pridružena valuacija je preslikavanje $i_r: R \rightarrow [0,1]$ za koje vredi:

$$i_r(A_k) = \begin{cases} > 0.5 & \text{ako je } C(A_k[t_i, t_j]) \geq \theta \\ \leq 0.5 & \text{ako je } C(A_k[t_i, t_j]) < \theta \end{cases}$$

gde je $\theta \in [0,1]$ lingvistička jačina zavisnosti.

Sledeći primer pokazuje istinitu vrednost atributa nad datom relacijom, gde se ova istinita vrednost određuje na osnovu bliskosti za dve n-torke nad odgovarajućim atributom.

Neka je **Student**(Ime_Prezime, Inteligencija, Predmet, Godine, Uspeh) relaciona šema i neka je dvoelementna podrelacija r data tabelom 22. Ovde su domeni atributa Ime_Prezime i Predmet klasični skupovi, dok su domeni atributa Inteligencija, Godine i Uspeh fazi podskupovi.

Milan	{Nadaren, Bistar}	Matematika	{Vrlo mlad}	{veoma}
Marko	{Nadaren}	Informatika	{Star}	{Više manje uspešan}

Tabela 22. Podrelacija r

gde domen atributa Inteligencija može biti dat sa:

$$\text{Dom(Inteligencija)} = \{\text{Nadaren, Bistar, Prosječna, Iznad proseka}\}$$

Tabela 23. ilustruje relaciju sličnosti na domenu Inteligencija:

Sličnost	Nadaren	Bistar	Prosečan	Iznad proseka
Nadaren	1	0.7	0.7	0.3
Bistar	0.7	1	0.8	0.3
Prosečan	0.7	0.8	1	0.3
Iznad proseka	0.3	0.3	0.3	1

Tabela 23. Relacija sličnosti na dom(Inteligencija)

Na sličan način može se dati i relacija sličnosti za ostale attribute. Sada izračunajmo stepen saglasnosti na atributu inteligencija:

$$\begin{aligned} C(\text{Inteligencija}[t_1, t_2]) &= \min\{\min\{\max\{s(\text{Nadaren}, \text{Nadaren})\}, \max\{s(\text{Bistar}, \text{Nadaren})\}\}, \\ &\quad \min\{\max\{s(\text{Nadaren}, \text{Nadaren}), s(\text{Nadaren}, \text{Bistar})\}\}\} \\ &= \min\{\min\{\max\{1\}, \max\{0.7\}\}, \\ &\quad \min\{\max\{1, 0.7\}\}\} \\ &= \min\{\min\{1, 0.7\}, \min\{1\}\} \\ &= \min\{0.7, 1\} = 0.7 \end{aligned}$$

Oдавde sledi da je stepen saglasnosti na atributu Inteligencija jednaka 0.7. Neka je jačina zavisnosti $\theta = 0.6$. Otuda je i $C(\text{Inteligencija}[t_1, t_2]) > \theta$. Ili ako je Inteligencija prve osobe data sa {Nadaren, Bistar}, a druge {Nadaren} tada je stepen bliskosti između njih jednak 0.7, odnosno istinita vrednost atributa Inteligencija nad datom relacijom veća je od 0.5.

Na sličan način neka je stepen saglasnosti na atributu Uspeh za dve n-torke t_1 i t_2 , na istoj relaciji veća ili jednaka od θ , gde je $\theta \in [0,1]$, tj. neka je $C(\text{Uspeh}[t_1, t_2]) \geq \theta$. Uzmimo dalje da je stepen saglasnosti na atributu Godine za dve n-torke t_1 i t_2 , na istoj relaciji, manja od θ , tj. $C(\text{Godine}[t_1, t_2]) < \theta$.

Pošto su vrednosti atributa Ime_Prezime i Predmet atomik, tada $C(\text{Ime_Prezime}[t_1, t_2])$ i $C(\text{Predmet}[t_1, t_2])$ mogu biti samo 0 ili 1, pa je i tada u datoj relaciji, $C(\text{Ime_Prezime}[t_1, t_2]) = 0$ i $C(\text{Predmet}[t_1, t_2]) = 0$. Dakle, u ovom primeru imamo da je:

1. $C(\text{Inteligencija}[t_1, t_2]) \geq \theta$ i $C(\text{Uspeh}[t_1, t_2]) \geq \theta$
2. $C(\text{Godine}[t_1, t_2]) < \theta$, $C(\text{Ime_Prezime}[t_1, t_2]) < \theta$ i $C(\text{Predmet}[t_1, t_2]) < \theta$.

Sada je očigledno na osnovu definicije 3.1 istinita vrednost atributa Inteligencija i Uspeh nad datom relacijom r:

$$i_r(\text{Inteligencija}) > 0.5 \text{ i } i_r(\text{Uspeh}) > 0.5,$$

dok je istinita vrednost atributa Ime_Prezime, Predmet i Godine:

$$i_r(\text{Ime_Prezime}) \leq 0.5, \quad i_r(\text{Predmet}) \leq 0.5 \text{ i } i_r(\text{Godine}) \leq 0.5.$$

Dakle, na ovom primeru su date istinite vrednosti pojedinih atributa u datoj relaciji r.

3.5.9. Analiza podataka i primena funkcionalnih zavisnosti u modelu Cubero - Medina

Teorija normalizacije koju je uveo Codd je sistematski pristup pravilnom projektovanju baza podataka. Glavna ideja je da ako smo suočeni sa relacijama u bazi podataka koje zadovoljavaju funkcionalnu zavisnost (ne uključujući primarni ključ), onda može da postoji redundantnost i ažuriranje postojeće baze. Kako bismo izbegli redundantnost, mi možemo originalnu relaciju razložiti, tj. napraviti dekompoziciju, a da pri tome ne dođe do gubitka informacija. Inače, u realnim bazama podataka, nije uobičajeno da su date striktno stroge zavisnosti u odnosima. Ipak, mi možemo da pronađemo funkcionalne i fazi funkcionalne zavisnosti, kao što je "Težina osobe zavisi više ili manje od njegove visine i uzrasta". U ovakvim situacijama predlaže se proces dekompozicije [37] i izdvajanja informacija, poštujući datu zavisnost, i kompresiju originalnih podataka u relacionoj bazi podataka. Ideja je primena teorije fazi skupova i tolerisanje nekih nepreciznosti u bazi, koje nam omogućava da objedinimo više n-torki u jednu. Razmotrimo, na primer, odnos koji se pojavljuje u sledećoj relaciji. Poseban operator pridruživanja se koristi za ponovno dobijanje originalnih podataka iz relacije R, tj. recovering.

X	Visina	Težina
X1	180	86
X2	170	74
X3	170	73

Tabela 24. Originalna relacija R

X	Visina	Težina
X1	Visok	oko 85 kg
X2	170	74 kg
X3	170	oko 73 kg

Tabela 25. Uvođenje fazi vrednostu u relacionu strukturu

X	Visina
X1	180
X2	170
X3	170

Tabela 26. Relacija r_1

Visina	Težina
Visok	Oko 85 kg
170	74 ili oko 73 kg

Tabela 27. Dekompozicija R - relacija r_2

Dekompozicija relacije R je data sa relacijama (projekcijama) r_1 i r_2 , kao što je prikazano u prethodnim tabelama. Kao što se može videti u ovom primeru, imamo smanjenu redundansu jer su druga i treća n-torka spojeni u jednu u relaciji r_2 . Od primarnog značaja u ovakvim situacijama je da kvantifikujemo koliku nepreciznost možemo tolerisati, da bismo garantovali da su fazi vrednosti kao što su "Oko 85 kg" dovoljno blizu originalnom podatku 86 kg. Da bismo to uradili, mi moramo koristiti meru sličnosti između elemenata podataka. Za nove podatke u bazi, mi možemo da testiramo fazi zavisnost posmatranjem n-torki u relaciji r_2 . Na ovaj način količina podataka smeštena u baze r_1 i r_2 je manja nego u relaciji R. Originalni podaci koji se pojavljuju u R mogu se dobiti spajanjem relacija r_1 i r_2 .

Važno je naglasiti da jezičke varijable, kao što je "Oko 85 kg", su definisane i date od strane stručnjaka – iskusnih projektanata informacionih sistema. Dakle, trebalo bi da budemo u stanju da otkrijemo neka znanja u vidu fazi pravila koja će nam omogućiti da posle dekompozicije originalne relacije, smanjimo i uklonimo redundantnost. To nam, svakako omogućava da dobijemo bolje razumevanje realnog sveta, jer su fazi zavisnosti izolovane u poseban odnos.

Definicija fazi funkcionalni zavisnosti u modelu Cubero-Medina:

Definicija 3.33. Ako je $R_i(X_i(t_1), X_i(t_2)) \geq \alpha_i \forall i$ onda mora da važi $R_j(Y_j(t_1), Y_j(t_2)) \geq \beta_j \forall j$. Kao specijalan slučaj imamo $\forall t_1, t_2 \in r$ važi da je $X_i(t_1)=X_i(t_2) \forall i$, tada je $R_j(Y_j(t_1), Y_j(t_2)) \geq \beta_j \forall j$, za slučaj postojanja fazi funkcionalne zavisnosti.

U slučaju klasičnih relacionih baza podataka, kada u relaciji R postoji zavisnost između X i Y, projekcija Π_{XY} bira vrednost tako da su XY n-torke sa istim vrednostima X (a samimi tim imaju iste i Y vrednosti). Cilj je da u novodobijenoj relaciji smanjimo količinu informacija. Sledeći korak je da se spoje n-torke sa istim X vrednostima. Sada za dve n-torke t i t' sa $X(t) = X(t')$ može biti $Y(t) \neq Y(t')$, ali da bismo zadovoljili postojeću zavisnost moramo osigurati $Y(t) \approx_{\beta} Y(t')$. To nam zapravo garantuje da nećemo dobiti previše rasplnute informacije (na primer "između 130 i 210"), čime ćemo očuvati postojeću fazi funkcionalnu zavisnost i u projekciji Π_{XY} . Osim toga, zavisnost može da se testira i za nove n-torke koje se dodaju u novu relaciju, preko već projektovanih odnosa koje smo koristili i za originalnu relaciju. Posmatrajmo sledeći primer:

Z	Visina	Težina
Z1	185	86
Z2	185	90
Z3	192	99
Z4	194	95
Z5	160	65
Z6	176	79
Z7	160	65
Z8	160	66

Tabela 28. Relacija R

Ova relacija ne zadovoljava klasičnu funkcionalnu zavisnost $Visina \rightarrow Težina$. Ako je $R_T(86,90) \geq \beta$ i $R_T(65,66) \geq \beta$, kažemo da postoji fazi funkcionalna zavisnost $Visina \xrightarrow{(1,\beta)} Težina$.

Visina	Težina
185	oko 86
192	99
194	95
160	65
176	79
160	oko 65

Tabela 29. Projekcija Π_{VT}

U prethodnom primeru, primetimo da je originalna relacija R klasična relaciona baza podataka, dok je odgovarajuća projekcija Π_{VT} fazi relaciona baza podataka. Na osnovu prethodno rečenog, možemo da konstatujemo da je zadovoljena fazi funkcionalna zavisnost $Visina \xrightarrow{(1,\beta)} Težina$ ako i samo ako je ista zadovoljena i u projekciji Π_{VT} .

Dakle, Cubero, Medina i ostali uvode fazi funkcionalne zavisnosti u relacioni model baza podataka. Takav pristup nam omogućava da otkrijemo veze između atributa koje nisu otkrivene klasičnom aproksimacijom i izvršimo dekompoziciju poštujući uspostavljene fazi zavisnosti. Na taj način, smanjujemo redundansu u bazama podataka, štedimo računarske resurse, a da pri tome nema gubitka informacija. Dokaz za to je mogućnost dobijanja originalnih podataka koje smo imali u početnoj bazi, a koje pri tome nisu fazi. Nedostatak ovog pristupa je činjenica da je za efikasan dizajn i uspostavljanje fazi funkcionalnih zavisnosti neophodna pomoć stručnjaka i iskusnih projektanata. Rešenje za ovakvu vrstu problema mogli bi da budu sistemi za rudarenje podataka odnosno dizajniranje efikasnih algoritama za otkrivanje fazi funkcionalnih zavisnosti bez uključivanja subjektivnog ljudskog faktora.

3.5.10. Fazi aproksimativne zavisnosti

U analizi podataka u relacionim bazama, vrlo zanimljivo pitanje jeste otkrivanje mogućih odnosa između vrednosti atributa, a na višem nivou i odnosa između samih atributa, tj. analiza funkcionalnih i višeznačnih zavisnosti. U slučaju prisustva neizvesnosti i neodređenosti podataka, specifične metode rudarenje podataka se koriste u otkrivanju znanja. Kao što smo to već napomenuli u ovoj disertaciji, teorija fazi skupova je korisna alatka za postizanje tog cilja. Berzal, Blanco i ostali [10] predlažu algoritam za računanje približne fazi zavisnosti i različitih tipova odnosa između atributa u fazi relacionim modelima baza podataka.

Metode otkrivanja znanja u bazama podataka se bave pronalaženjem nepoznatih i potencijalno korisnih znanja iz baze. Taj proces se sastoji od tri koraka:

- a) Priprema podataka,
- b) Rudarenje podataka (pronalaženje skrivenih znanja) i
- c) Interpretacija rezultata rudarenje podataka čime se obezbeđuju nova znanja.

Teorija fazi skupova, i u ovom slučaju se pokazuje kao pogodan alat, kojim dolazimo do novih saznanja i izražavamo odnose između podataka. Najvažnija činjenica u celoj ovoj priči jeste veza između fazi skupova i sistema za otkrivanje znanja: u mnogim slučajevima podaci su neprecizni i neodređeni, ali fazi relacioni modeli i fazi metod

zaključivanja su razvijeni kako bi se nosili sa ovim potencijalnim problemima. Analiza informacija zahteva specifične alate i fazi proširenja postojećih zavisnosti između podataka, te u tom kontekstu se govori o fazi približnim zavisnostima, koje navode u svom radu Berzal i ostali [15]. Grubo govoreći, približne zavisnosti su funkcionalne zavisnosti sa pojedinim izuzecima. Ove zavisnosti se takođe koriste za predstavljanje odnosa između atributa.

Da bismo ilustrovali ove tvrdnje, poslužićemo se jednostavnim primerom. Neka Kategorija i Zarada predstavljaju dva atributa koji preciziraju platu i poziciju zaposlenog. Pretpostavimo da su vrednosti atributa Kategorija, koji ukazuju na vrstu posla, unutar skupa (Menadžer, Komercijalista, Podrška...), a plata je numerički podatak. Uobičajeno je da postoji odnos između kategorije i plate (viša kategorija, veća zarada). Međutim, ako želimo opisati ovaku vrstu odnosa pomoću pravila kao što su: ‘If Kategorija=Menadžer then Zarada=100.000’ nisu najbolje rešenje jer postoje mnoge različite numeričke vrednosti za plate menadžera. Sa druge strane, semantički sadržaj ovakvih pravila je veoma siromašan. U takvim okolnostima, mi preferiramo pravilima kao što su ‘If Kategorija=Menadžer then Zarada=Visoka’. Da bi se otkrile i procenile ovakve vrste pravila i odnosa između podataka neophodni su specifični alati, kao što je algoritam za otkrivanje fazi približnih zavisnosti.

U realnim bazama podataka, susrećemo se sa dve različite vrste odnosa. Sa jedne strane, postoje odnosi koji nisu implicitni, kod kojih su odnosi između vrednosti atributa skriveni i koji nisu dovoljno jasni u prvom trenutku. Ovaj tip zavisnosti se dobiva kroz samu analizu baze podataka. Sa druge strane, često se susrećemo sa eksplicitnim odnosima između atributa koji se lako detektuju (Npr. Grad i Poštanski broj). Ove dve vrste odnosa između atributa u relacionim strukturama baza podataka predstavljaju ograničenja integriteta koja se nameću u toku dizajniranja baza podataka. U ovim slučajevima govorimo da postoji funkcionalna zavisnost ili približna zavisnost između atributa.

Pretraga funkcionalnih zavisnosti u relacionim bazama podataka je predmet interesovanja u oblasti rudarenje podataka, jer se ovaj vid poslovne inteligencije vrlo strogo bavi strukturom podataka. Međutim, jako teško je savršeno otkriti funkcionalne zavisnosti u bazama podataka, jer jedan jedini izuzetak u pravilima utiče na gubljenje zavisnosti. Ako taj broj izuzetaka nije veliki, ‘fazi funkcionalne zavisnosti sa izuzecima’ mogu nam prikazati interesantne pravilnosti sadržane u podacima. Šta više, određuje se i prikazuje stepen zavisnosti koji postoji između podataka. Ideja je da se meri ne samo tačnost zavisnosti, već i podrška (proporcija n-torki u kojima se posmatrana zavisnost pojavljuje). Dakle, za procenu zavisnosti koriste se poverenje (Confidence) – uslovna verovatnoća $p(Y/X)$, u oznaci $Conf(X \rightarrow Y)$ i podrška (Support) – verovatnoća $p(X \cup Y)$, u oznaci $S(X \rightarrow Y)$.

Problem kod poverenja predstavlja činjenica da se ne uzimaju u obzir negativne zavisnosti, pa mogu da se dobiju visoki procenti poverenja u tim slučajevima koji mogu da dovedu u zabludu. Zbog toga se u radovima predlaže korišćenje faktora sigurnosti CF:

$$CF(X \rightarrow Y) = \begin{cases} \frac{Conf(X \rightarrow Y) - S(Y)}{1 - S(Y)}, & Conf(X \rightarrow Y) > S(Y) \\ \frac{Conf(X \rightarrow Y) - S(Y)}{S(Y)}, & Conf(X \rightarrow Y) < S(Y) \\ 0 & Inace \end{cases}$$

Faktor sigurnosti uzima vrednosti iz intervala $[-1,1]$ i ukazuje nam u kojoj meri je naše uverenje o postojanju zavisnosti tačno. $CF = 1$ u situacijama kada je $X=True$ tada $Y=True$, a $CF=-1$ u suprotnom. Dva ekstremna slučaja su kada je $S(Y)=0$ i $S(Y)=1$. U oba slučaja rezultat je trivijalan, pa je logično da tada uzimamo vrednost $CF=0$.

Vraćajući se na ideju funkcionalnih zavisnosti, približne zavisnosti se definišu kao pravila koja se odnose na prisustvo zavisnosti u n -torkama posmatrane relacije. Sledeće interpretacije i koncepti se koriste u analizi aproksimativnih zavisnosti:

- Stavka je objekat povezan sa posmatranim atributom u relaciji R . Za svaki atribut $A_{tk} \in R$ sa $it_{A_{tk}}$ označavamo povezane stavke.

- Uvodimo skup I_X kao:

$$I_X = \{ it_{A_{tk}} \mid A_{tk} \in X \}$$

- T_r je T-skup koji za svaki par n -torki $(t,s) \in r \times r$ sadrži transakciju $ts \in T_r$ tako da vredi:

$$it_{A_{tk}} \in ts \Leftrightarrow t[A_{tk}] = s[A_{tk}]$$

Jasno je da je $|T_r| = |r \times r| = n^2$.

Na primer, razmotrimo relaciju r prikazanu u tabeli 30. Prema prethodnim interpretacijama T-skup T_r je prikazan u tabeli 31.

Sada približna zavisnost $X \rightarrow Y$ iz relacije r prelazi u pridruženo pravilo $I_X \Rightarrow I_Y$ u T_r . Podrška i faktor sigurnosti mere interes i tačnost zavisnosti $I_X \Rightarrow I_Y$.

n-torke	A	B	C
t_1	a_1	b_1	c_1
t_2	a_2	b_1	c_2
t_3	a_1	b_1	c_3
t_4	a_3	b_2	c_3

Tabela 30. Relacija r

T_r	it_A	it_B	it_C
t_1t_1	1	1	1
t_1t_2	0	1	0
t_1t_3	1	1	0
t_1t_4	0	0	0
t_2t_1	0	1	0
t_2t_2	1	1	1
t_2t_3	0	1	0
t_2t_4	0	0	0
t_3t_1	1	1	0
t_3t_2	0	1	0
t_3t_3	1	1	1
t_3t_4	0	0	1
t_4t_1	0	0	0
t_4t_2	0	0	0
t_4t_3	0	0	1
t_4t_4	1	1	1

Tabela 31. Relacija T_r

Definicija funkcionalne zavisnosti:

Definicija 3.33. Ako je $CF(X \rightarrow Y) = 1$ (a to implicira i da je $Conf(X \rightarrow Y) = 1$) onda je $X \rightarrow Y$ funkcionalna zavisnost.

Podrška i tačnost približne zavisnosti $X \rightarrow Y$ može da se tumačiti kao agregacija podrške i tačnosti pridruženih pravila koja se odnose na vrednosti atributa X i Y . Zbog toga, aproksimativne zavisnosti se mogu posmatrati kao rezime informacija sadržanih u ovim pravilima. Konkretno, primenom ove metodologije u relacionim bazama podataka moguće je izdvojiti sve postojeće funkcionalne i fazi funkcionalne zavisnosti. Dobijeni skup zavisnosti može da se smanji i redukuje u zavisnosti od potreba korisnika i faktora sigurnosti CF . Tehnike rudarenja podataka mogu imati velike koristi od teorije fazi skupova jer nam ovakav način posmatranja problema omogućava da dobijemo više razumljive odnose između podataka.

3.5.11. Fazi višeznačne zavisnosti

Višeznačne zavisnosti, koje predstavljaju drugu važnu formu zavisnost, pomažu u otklanjanju redundancije podataka u kreiranju baza podataka. Definicija višeznačne zavisnosti kaže da ako su dve n -torke jednake na skupu atributa X tada postoji n -torak t_3 u datoj relaciji r takva da je $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$ i $t_3[Z] = t_2[Z]$ odnosno, imamo sledeću definiciju:

Definicija 3.34. Neka je R relaciona šema, $X, Y \subseteq R$, $X \cap Y = \emptyset$ i $Z = R - [XY]$. Kažemo da relacija r zadovoljava višeznačnu zavisnost $X \twoheadrightarrow Y$ ako vredi da za svake dve n -torke t_1 i t_2 iz r za koje je ispunjeno $t_1[X] = t_2[X]$, postoji n -torak t_3 u r , takva da je $t_3[X] = t_1[X]$, $t_3[Y] = t_1[Y]$ i $t_3[Z] = t_2[Z]$.

Pošto ova definicija višeznačne zavisnosti nije primenljiva u fazi modelu, jer ne postoji jasan način određivanja kada su dve neprecizne informacije jednake, nju je potrebno proširiti sa nepreciznim informacijama. Te proširene verzija višeznačnih zavisnosti nazivaju se fazi višeznačne zavisnosti [44,45,50,54].

Definicija 3.35. Neka je r fazi relacija na relacionoj šemi $R(A_1, \dots, A_n)$ i neka je U univerzalni skup atributa A_1, \dots, A_n i neka su X i Y podskupovi od U . Kažemo da fazi relacija r zadovoljava fazi višeznačnu zavisnost $X \twoheadrightarrow Y$ ako za svaki par n -torke t_1 i t_2 iz r postoji n -torak t_3 u r , takva da je:

$$C(X[t_1, t_3]) \geq \min(\theta, C(X[t_1, t_2])),$$

$$C(Y[t_1, t_3]) \geq \min(\theta, C(X[t_1, t_2])),$$

$$C(Z[t_2, t_3]) \geq \min(\theta, C(X[t_1, t_2])).$$

gde je θ , realan broj iz $[0,1]$ i koji označava lingvističku jačinu zavisnosti.

Da ovako definisane fazi višeznačne zavisnosti osiguravaju konzistentnost govori naredna teorema.

Teorema 6. Fazi višeznačne zavisnosti su konzistentne.

Sada dajemo šemu aksioma za fazi višeznačne zavisnosti.

A1: Ako vredi $X \xrightarrow[F]{\theta_1} Y$ i $\theta_1 \geq \theta_2$ tada je i $X \xrightarrow[F]{\theta_2} Y$ (Inkluzivno pravilo).

A2: $\{X \xrightarrow[F]{\theta} Y\} | \Rightarrow X \xrightarrow[F]{\theta} U\text{-}XY$ (Komplementarno pravilo).

A3 : Ako vredi $\{X \xrightarrow[F]{\theta} Y\}$ i $Z \subseteq W$ tada vredi i $W X \xrightarrow[F]{\theta} YZ$ (Šema aksioma pribrajanja).

A4: $\{X \xrightarrow[F]{\theta_1} Y, Y \xrightarrow[F]{\theta_2} Z\} | \Rightarrow X \xrightarrow[F]{\min(\theta_1, \theta_2)} Z\text{-}U$ (Šema aksioma tranzitivnosti).

A5 : $\{X \xrightarrow[F]{\theta} Y\} | \Rightarrow X \xrightarrow[F]{\theta} Y$ (Replicitno pravilo).

A6: Ako vredi $X \xrightarrow[F]{\theta_1} Y$ i $Z \subseteq Y$ i za neki W iz Y vredi $W \xrightarrow[F]{\theta_2} Z$ tada vredi $X \xrightarrow[F]{\theta_2} Z$.

Dodatna pravila izvođenja za fazi višeznačne zavisnosti:

A7: $\{X \xrightarrow[F]{\theta_1} Y, X \xrightarrow[F]{\theta_2} Z\} | \Rightarrow X \xrightarrow[F]{\min(\theta_1, \theta_2)} YZ$.

A8: $\{X \xrightarrow[F]{\theta_1} Y, WY \xrightarrow[F]{\theta_2} Z\} | \Rightarrow WX \xrightarrow[F]{\min(\theta_1, \theta_2)} Z$.

A9: $\{X \xrightarrow[F]{\theta_1} Y, WY \xrightarrow[F]{\theta_2} Z\} | \Rightarrow WX \xrightarrow[F]{\min(\theta_1, \theta_2)} Z\text{-}WY$.

A10: Ako vredi $X \xrightarrow[F]{\theta_1} Y, X \xrightarrow[F]{\theta_2} Z$ tada vredi $X \xrightarrow[F]{\min(\theta_1, \theta_2)} Y \cap Z, X \xrightarrow[F]{\min(\theta_1, \theta_2)} Y\text{-}Z,$

$X \xrightarrow[F]{\min(\theta_1, \theta_2)} Z\text{-}Y$.

A11: $\{X \xrightarrow[F]{\theta_1} Y, XY \xrightarrow[F]{\theta_2} Z\} | \Rightarrow WX \xrightarrow[F]{\min(\theta_1, \theta_2)} Z\text{-}Y$.

Da su ova pravila sound i complete tvrdi naredna teorema.

Teorema 7. Praviia izvođenja $\{A1, A2, A3, A4, A5, A6\}$ su sound i complete.

Dokaz kompletnosti ovih aksimoma dat je u radu [44].

3.6. Zaključak

U ovome poglavlju posmatrana je uopšteno kako klasična relaciona baza podataka tako i fazi relaciona baza podataka. Najpre je uveden relacioni model baza podataka i u njemu su definisani pojmovi iz relacione algebre, a zatim su date A r m s t r o n g o v e aksiome i odgovarajuća pravila izvođenja. Pored fazi relacione baze podataka, predstavljena su dostignuća u oblasti fazi objektno-orijentisanih baza podataka.

Zatim je uvedena definicija fazi funkcionalnih i višeznačnih zavisnosti i data su pravila izvođenja tih zavisnosti u već poznatim modelima.

Sve navedene osobine i pojmovi u ovome poglavlju, kao i već poznati rezultati iz teorije fazi relacionih baza podataka, iskorišćeni su u svrhu naprednih istraživanja u ovoj doktorskoj disertaciji.

4. Projektovanje relacionih baza podataka

Poseban značaj u radu sa relacionim bazama podataka ima struktuiranje i organizacija samih podataka. Podatke je na nivou korisnika potrebno struktuirati, a na fizičkom nivou organizovati tako da njihovo održavanje bude najlakše, a manipulisanje njima najefikasnije. Skup postupaka kojima se dolazi do dobro struktuiranih podataka u bazi podataka naziva se metodama logičkog projektovanja baze podataka. Skup postupaka kojima se podaci fizički organizuju u bazi, tako da im je pristup i održavanje najefikasnije, naziva se metodama fizičkog projektovanja, tj. metodama fizičke organizacije podataka. Ogromnu ulogu u logičkom dizajnu baza podataka imaju funkcionalne zavisnosti, višeznačne zavisnosti, zavisnosti pridruživanja o kojima je bilo reči u prethodnim poglavljima. U ovom radu su prikazani različiti pristupi u analizi funkcionalnih zavisnosti koje koriste projektantni informacionih sistema prilikom dizajniranja baza podataka. Zato ćemo u uvom poglavlju pokušati da damo kraću analizu logičkog modeliranja i fizičke organizacije podataka u relacionim bazama, kako bi prethodna teorijska razmatranja mogla da se implementiraju u potpunosti, a pod tim podrazumevamo realizaciju fazi relacionih baza podataka koju daje prethodna analiza i teorija.

4.1. Logički model baza podataka

Relacioni model baze podataka sastoji se od dva dela i to:

- logičkog i
- fizičkog modela.

Pod logičkim projektovanjem relacione baze podataka podrazumeva se projektovanje šeme relacione baze podataka, tj. grupisanje pojedinih atributa u posebne relacije, ili izdvajanje uočenih entiteta i odnosa među entitetima u posebne relacije, a sve to na osnovu analize informacionih zahteva i informacionog okruženja. Značaj dobrog logičkog projektovanja relacione baze podataka leži u činjenici da je takvu bazu moguće efikasnije održavati, i da je manipulisanje podacima u njoj jednostavnije i pouzdanije [71].

Logički model baze podataka nastaje kao rezultat:

- analize postojećih i / ili dobijenih novih podataka,
- definisanja tabela (relacija) i veza među njima, te
- dovođenja modela na relacioni oblik

uz prethodno definisanu strukturu i oblik podataka koji će činiti sadržaj baze.

Kvalitet logičkog dizajna baza podataka uvek zavisi od toga koliko uspešno dizajn modela odgovara realnom svetu. Klasične relacione baze podataka nam omogućavaju da dizajniramo bazu sa informacijama iz realnog sveta koje su sve precizne i tačne. Zapravo, ova pretpostavka predstavlja restrikciju, jer u bazu mogu da se skladište samo precizne informacije. Ovo ima za krajnju posledicu gubitak informacija. Neprecizne informacije su posebno važne u sistemima i aplikacijama koji su uključeni u proces donošenja odluka. Najpogodniji način da se neprecizne informacije sačuvaju u bazi jeste da se povežu sa konceptom fazi logike. Tako podaci u bazi dobijuju prefiks fazi. U radu su navedeni načini i modeli predstavljanja fazi informacija u Fazi Relacionim Bazama podataka.

U logičkom dizajnu baza podataka, ograničenja integriteta imaju ključnu ulogu. Među tim ograničenjima, zavisnost između podataka predstavlja važnu i možda najšire ispitivanu klasu. Razne vrste zavisnosti, kao što su funkcionalne zavisnosti, višeznačne zavisnosti, zavisnosti pridruživanja itd. se koriste prilikom projektovanja baza podataka. Druga, vrlo važna pretpostavka u relacionim bazama podataka jeste činjenica da neki podaci u potpunosti određuju druge podatke ili su podaci potpuno nezavisni. Međutim, u mnogim realnim situacijama, postoje podaci koji nisu 100% zavisni, ali ipak zavisnost postoji (npr. "Nivo inteligencije osobe manje ili više određuje uspeh u poslu.") U ovom primeru i stepen uspeha i nivo inteligencije su fazi atributi, pa govorimo o fazi zavisnostima. Ako znamo da je čovek inteligentan, mi znamo da će biti uspešan, ali njegov uspeh nije precizno određen i definisan.

Vrlo važna stavka u analizi zavisnosti podataka je postojanje pravila zaključivanja. To je skup pravila koji projektantima informacionih sistema ostavlja mogućnost ispravnog logičkog dizajniranja i zaključivanja. Da bi se takva pravila i primenjevala, neophodno je da budu *sound i complete*.

Značaj funkcionalnih zavisnosti se ogleda pre svega u mogućnosti da prilikom dizajniranja relacione šeme baze, eliminišemo redundansu koja je svakako prisutna. Za par atributa X i Y kažemo da postoji funkcionalna zavisnost $X \rightarrow Y$, ako je $t[X]=t'[X]$ onda važi i $t[Y]=t'[Y]$. tj:

$$t[X]=t'[X] \Rightarrow t[Y]=t'[Y]$$

pri čemu je $t[X]$ projekcija n-torke (vrste) t na attribute X.

Kod fazi funkcionalnih zavisnosti se ne možemo osloniti na koncept jednakosti. Sozat i Yazici, Bosc i ostali [38,44] u analizi fazi funkcionalnih zavisnosti koriste relaciju sličnosti. U tom slučaju kažemo: Ako je $t[X]$ slično sa $t'[X]$, $t[Y]$ je takođe slično sa $t'[Y]$, i pišemo:

$$X \xrightarrow[F]{\theta} Y.$$

Tipičan primer FFD: "Zaposleni sa sličnim iskustvom imaju slične plate". Iskustvo i plata su atributi koji mogu da imaju vrednosti u vidu lingvističkih izraza. Jedan od načina da izrazimo stepen zavisnosti je koristeći prag vrednosti. Za prethodna dva primera imamo: "Zaposleni sa sličnim iskustvom imaju slične plate" – stepen zavisnosti možemo da iskažemo sa vrednošću 0.9 ili "Nivo inteligencije osobe manje ili više određuje uspeh u poslu" – lingvistička snaga 0.6. Ova vrednost se zove snaga, odnosno stepen fazi funkcionalne zavisnosti, i označavamo je sa θ

Primer 4.1 Posmatrajmo bazu podataka koja je predstavljena sledećom relacionom šemom:

P (P_SIF, IME, BR_NASLOVA, DRZAVA)

I (I_SIF, NAZIV, STATUS, DRZAVA)

K (K_SIF, NASLOV, OBLAST)

KP (K_SIF, P_SIF, R_BROJ)

KI (K_SIF, I_SIF, IZDANJE, GODINA, TIRAZ)

Prve tri relacije predstavljaju tipove entiteta PISAC, IZDAVAČ i KNJIGA redom, i imaju sličnu strukturu kao i odgovarajući tipovi entiteta. Nov je prvi atribut u svakoj od relacija, koji predstavlja šifru, tj. jedinstveni identifikator pisca, izdavača i knjige. Na

primer, šifra knjige, K_SIF, može biti ISBN (International Standard Book Number), P_SIF i I_SIF mogu biti matični broj pisca i izdavača, redom. Relacija KP predstavlja apstraktni tip entiteta AUTOR, tj. odnos između tipova entiteta KNJIGA i PISAC, dok relacija KI predstavlja apstraktni tip entiteta IZDAVAŠTVO, tj. odnos između tipova entiteta KNJIGA i IZDAVAČ.

Ova relacionalna baza podataka je dobro logički projektovana jer omogućuje jednostavno unošenje, brisanje i izmenu objekata (ili njihovih odnosa) predstavljenih svakom pojedinačnom tabelom. Nešto formalnije, “dobro” logičko projektovanje relacione baze podataka zasnovano je na karakterističnim zavisnostima koje postoje među atributima svake pojedinačne relacije. Na primer, u relaciji I, jedinstveni identifikator vrste jeste šifra izdavača (atribut I_SIF), tj. I_SIF jednoznačno određuje vrednost svakog od atributa NAZIV, STATUS, DRZAVA (svaki od atributa NAZIV, STATUS, DRZAVA funkcionalno zavisi od atributa I_SIF). Ovo jednoznačno određenje, tj. funkcionalna zavisnost, označava se na sledeći način:

$I_SIF \rightarrow NAZIV$

$I_SIF \rightarrow STATUS$

$I_SIF \rightarrow DRZAVA$

U relaciji I, atribut I_SIF je *ključ* jer svi ostali atributi od njega funkcionalno zavise. Razne n-torke relacije I moraju imati različite vrednosti atributa I_SIF. Relacije opisane izdavačke baze podataka su dobro projektovane jer su sve zavisnosti posledice ključa (na levoj strani imaju ceo ključ).

Primer 4.2. Razmotrimo primer loše isprojektovane relacije R:

I_SIF	K_SIF	NAZIV
I1	K1	Prosveta
I3	K2	Dečje novine
I3	K3	Dečje novine
I4	K1	Matica srpska
I5	K1	Prosveta
I6	K2	Prosveta

Tabela 32. Primer projektovanja baze podataka

U ovoj relaciji postoje problemi sa unošenjem, brisanjem i izmenom podataka, tzv. “anomalije”, jer nisu sve zavisnosti posledica ključa.

Različite metodologije logičkog projektovanja dolaze do šeme relacione baze različitim postupcima, pri čemu se već u opisu značenja “logičkog projektovanja” uočavaju dva globalno različita postupka: jedan se zasniva na grupisanju atributa – svojstava informacionog okruženja – u posebne relacije, a drugi na izdvajanju entiteta i odnosa u posebne relacije. Prvi je “sintaksno” orijentisan i vođen vezama među atributima izraženim kroz koncepte zavisnosti (funkcionalne, višeznačne, zavisnosti spajanja), a drugi je “semantički” orijentisan i vođen je vezama među entitetima informacionog okruženja. Metoda *normalnih formi* dosledno i kompletno realizuje prvi postupak logičkog projektovanja, dok razne metode *semantičkog modeliranja* realizuju drugi postupak. [71,73].

Dakle, problemu sinteze logičkog modela pristupa se na dva načina i to:

- postupkom normalizacije tabela ili
- preko modela objekat-veze (MOV, Entity-Relationship Model, skraćeno E-R model), pri čemu treba naglasiti da jedan pristup ne isključuje drugi, nego se naprotiv, često, međusobno i dopunjuju.

4.2. Logičko projektovanje postupkom normalizacije tabela

Logičko projektovanje može, po jednom kriterijumu, da bude vođeno pravilima o grupisanju atributa u relacije na osnovu logičkih veza među atributima. Za relacije koje se dobiju primenom takvih pravila kaže se da se nalaze u odgovarajućim normalnim formama [71]. Zbog toga su normalne forme čisto sintaksna svojstva zasnovana na algebri zavisnosti (funkcionalnih, višeznačnih i zavisnosti spajanja). Dakle, projektovanju informacionog sistema može se prići prostim sakupljanjem i registrovanjem svih raspoloživih atributa. U tom slučaju mora se posebno obratiti pažnja na to da logički model ne sadrži redundansu podataka. Tabele sa sirovo "nabacanim" atributima retko kada zadovoljavaju ovaj uslov, jer se pri njihovom modelovanju nije vodilo računa o izboru tabela i njihovih atributa.

Optimalan izbor relacija i njihovih atributa naziva se *normalizacija sistema*. Normalizacija je u stvari postupak izmene (najčešće dekompozicije, rastavljanja na više relacija) prvokoncipiranih tabela (relacija).

Dekompozicijom relacije dobija se uvek dve ili više novih, međusobno povezanih relacija, koje se nalaze i u nekoj od normalnih formi. U slučaju kada se projektovanju sistema pristupa preko E-R modela, i ako se isti korektno izvede, onda će i relacije koje iz njega slede već biti u nekoj višoj normalnoj formi pa mnogi tako i proveravaju kvalitet dobijenog E-R modela.

Tehnički postupak izvođenja normalizacije svodi se na operacije relacione algebre - projekciju i spajanje, i to:

- na dekompoziciju tabela generisanjem vertikalnih podskupova (operacija projekcije), i
- na generisanje novih relacija iz dvaju ili više dobijenih vertikalnih podskupova (operacija spajanja).

Ovakav pristup sintezi informacionog sistema naziva se vertikalna normalizacija u kojoj postoji pet normalnih formi. Za praksu su najvažnije druga i treća i na njima se često proces normalizacije i zaustavlja. Pomenimo da pored vertikalne postoji i horizontalna normalizacija.

Šta je to normalizovana, a šta nenormalizovana forma neke relacije (tabele), najbolje se može videti na sledećem primeru:

Primer 4.2. Pretpostavimo da neka trgovačka firma želi da vodi evidenciju o svome poslovanju. U tu svrhu skuplja i obrađuje podatke koji se nalaze u tabeli NARUDŽBA. To su: šifra kupca, ime i prezime kupca, adresa kupca, količina kupljene robe, šifra robe (artikla), naziv robe, kvalitet i cena robe. Bez dublje analize problem poslovanja se može predstaviti u vidu jedne tabele **NARUDZBA** (Sifra_Kup#, Ime, Adresa, Kolicina, Sifra_Art#, Naziv, Kvalitet, Cena).

Sifra_Kup#	Ime	Adresa	Kolicina	Sifra_Art#	Naziv	Kvalitet	Cena
K1	Nikola	Beograd	100	A1	Lak	2	220
K1	Nikola	Beograd	200	A2	Boja	1	130
K1	Nikola	Beograd	50	A3	Gips	3	20
K1	Nikola	Beograd	300	A4	Četke	2	70
...
K17	Miloš	Beograd	200	A2	Boja	1	130

Tabela 33. Table Nardudžba

Ovako koncipiran model je loš iz sledećih razloga:

1. Šifra kupca (Sifra_Kup#) i šifra robe (Sifra_Art#) su atributi tabele NARUDŽBA, iako su u realnom svetu identifikatori dva različitih objekta (KUPAC, ROBA). Tako, na primer, kvalitet nije atribut kupca nego nekog drugog objekta koji bi mogao da se zove ROBA.
2. Model sadrži redundansu jer se adresa i ime kupca javljaju onoliko puta koliko puta je neki kupac kupovao robu u toj trgovini. Isti zaključak vredi i za artikle. Na primer: opis artikla sa šifrom A2 (kvalitet, naziv i cena) javlja se dva puta, jer su dva različita kupca nabavljala taj isti artikl.

Prisustvo redundanse u bazi podataka dovodi do niza problema prvenstveno kod izmene (ažuriranja), ali i kod korišćenja podataka. Ti problemi se nazivaju anomalije. To su pre svega:

a. Anomalije pri upisu podataka

Ova anomalija sastoji se u tome da upis podataka o kupcu nije moguć sve dok neki kupac nešto ne kupi, iako on kao potencijalni kupac postoji i marketinški je interesantan. Analogno vredi i za artikle. Na primer: kupac sa šifrom K2 nema još ništa naručeno, i njegovi podaci nisu uneseni u tabelu NARUDŽBA (iako je možda upravo taj kupac veoma interesantan kao potencijalni kupac), ili podatke o artiklu sa šifrom artikla A5 ne možemo imati u bazi (iako taj artikl postoji) sve dok neko ne izvrši porudžbinu.

b. Anomalije pri brisanju podataka

Brisanjem jedne narudžbe iz tabele može se desiti da se izgube svi podaci o kupcu ili artiklu - robi. U konkretnom primeru: brisanjem kupca sa šifrom K17 biće izgubljeni svi podaci o njemu s obzirom da je on izvršio narudžbu samo jedanput. Na sreću, slučajno neće biti izgubljeni i podaci o artiklu koji je on kupio, jer postoji još jedan kupac (sa šifrom K2) koji je kupio isti artikl. Ali zato, brisanjem nekoliko narudžbi i takva greška može da se desi.

c. Anomalije pri izmeni podataka

Promenom imena ili adrese kupca nastaje problem izmene podataka na onoliko mesta na koliko je kupac bio upisan. Isti zaključak odnosi se i na artikle prilikom promene imena ili svojstva (na primer cene) nekog artikla.

4.2.1. Prva, druga i treća normalna forma

Svi pobrojani nedostaci mogu biti eliminisani ako se prvobitno koncipirane relacije - tabele normalizuju i dovedu u potrebnu normalnu formu. Pri tome treba samo obratiti pažnju da tokom postupka normalizacije (koji se, kako rekosmo, svodi na dekompoziciju i ponovo spajanje tabela) ne dođe i do gubitka informacija.

Prva normalna forma. Relacija se nalazi u prvoj normalnoj formi (1NF) ako i samo ako je njen domen (podaci u tabeli) skup atomskih vrednosti. Budući da je ovo i uslov da bi neka tabela uopšte bila relacija, sledi da se svaka relacija nalazi u prvoj normalnoj formi.

Pojava redundancije koja je u prvoj normalnoj formi praktično skoro uvek prisutna (kao što smo to videli u prethodnom primeru), te mogućnost svih pratećih grešaka, ukazuju na to da prva normalna forma nije ni izdaleka dovoljna za sintezu upotrebljive baze podataka. Neki autori stoga ovu formu nazivaju nenormalizovanom formom ili nultom normalnom formom, tako da se kod raznih autora može naći i različit ukupan broj normalnih formi.

Druga normalna forma. Druga normalna forma nema nekog većeg praktičnog značaja, jer je većina pomenutih anomalija i dalje prisutna kod relacija koje su dovedene u drugu normalnu formu. Pomenimo je stoga samo kao prethodnika treće normalne forme koja za praksu ima najveći značaj.

Po definiciji, relacija se nalazi u drugoj normalnoj formi ako svaki atribut, koji nije ključni, zavisi potpuno (ne delimično) od ključnog atributa.

Posmatrajmo opet prethodni primer. Relacija NARUDŽBA nije u drugoj normalnoj formi (2NF) jer u toj relaciji je jedini mogući kandidat za ključ složeni ključ (Sifra_Kup#, Sifra_Art#), a atribut kvalitet očigledno zavisi samo od dela (Sifra_Art#), a ne i celoga ključa. Prevedimo stoga relaciju NARUDŽBA u dve relacije, od kojih prva R1 sadrži podatke o kupcu, a druga R2, o artiklu koji je taj kupac kupio, i to na sledeći način:

R1 (Sifra_Kup#, Ime, Adresa) i

R2 (Sifra_Kup#, Sifra_Art#, Naziv, Kvalitet, Cena, Količina).

Relacija R1 je u 2NF, ali R2 još uvek nije jer ponovo atribut kvalitet zavisi samo od dela ključa (Sifra_Art#), ali ne od celoga ključa (Sifra_Kup#, Sifra_Art#). Relaciju R2 moramo stoga opet razbiti na dve od kojih, prva R21 sadrži samo podatke o robi, a druga R22 o kupcu i robi koji je on kupio:

R21 (Sifra_Art#, Kvalitet, Cena)

R22 (Sifra_Kup#, Sifra_Art#, Količina).

Sada su sve tri relacije R1, R21 i R22 u 2NF, jer R1 i R21 nemaju složeni ključ, a atribut količina u R22 zavisi potpuno, a ne delimično, od složenog ključa te relacije.

Treća normalna forma. Relacije R1, R21 i R22, u ovom primeru, zadovoljavaju i uslov treće normalne forme (3NF), što nikako ne znači da je to i u svakom drugom slučaju tako, jer se može desiti da dekompozicijom 1NF postignemo samo 2NF, a ne i više od toga. Pokažimo to na drugom primeru.

Pretpostavimo da unutar informacionog sistema nekog instituta imamo objekat - entitet LABORANT sa atributima:

LABORANT (Sif_Lab#, Ime, Prezime, Broj_Sobe, Telefon)

Relacija LABORANT je već u 2NF jer ima samo jedan ključni atribut (Sif_Lab# - šifra laboranta), i niz atributa. Međutim, između atributa Broj_Sobe i Telefon može da postoji međusobna zavisnost (telefon se nalazi u sobi) pa će se stoga neki broj telefona u laboratoriji pojaviti u relaciji onoliko puta koliko laboranata radi u njoj, pod uslovom da koriste isti telefon.

Prema tome, iako je relacija LABORANT u 2NF, pojava redundancije podataka u toj formi nije otklonjena, a time i sve anomalije pri održavanju kao na primer:

- Broj jednog od telefona i broj sobe neke laboratorije nije moguće uneti u informacijski sistem sve dok se ne unesu podaci o bar jednom laborantu koji radi u njoj,
- Brisanjem podatka o poslednjem laborantu neke laboratorije, gubi se podatak i o broju telefona u njoj (iako laboratorija, prostorija i dalje postoji), te konačno,
- ako laboratorija promeni broj telefona, izmenu treba uneti onoliko puta koliko laboranata radi u njoj.

Navedene slabosti ukazuju da relacija LABORANT mora biti dalje dekomponovana i na taj način dovedena do 3NF. Dekompozicija se izvodi tako da se neključni atributi, koji su u međusobnoj funkcionalnoj vezi, izdvoje u zasebnu relaciju, te da se tako nastala relacija poveže sa ostatkom prvobitne preko jednog atributa. Konkretno, u poslednjem primeru od relacije LABORANT treba napraviti dve:

LABORATORIJA (Broj_Sobe, Telefon)

LABORANT (Sif_Lab#, Ime, Prezime, Broj_Sobe)

U ovom primeru funkcionalna zavisnost između atributa Sif_Lab#, Broj_Sobe i Telefon postoji i dalje. Broj telefona, naime zavisi od broja sobe u kojoj se telefon nalazi, a broj sobe je u vezi sa laborantom koji sedi u njoj. Ne bi bilo dobro da smo dekompozicijom tabele izgubili tu zavisnost, jer model sistema ne bi više bio realna slika stvarnosti.

Relacija LABORANT nije prema tome pre dekompozicije bila u trećoj normalnoj formi, jer je postojala veza među atributima.

Relacija se nalazi u trećoj normalnoj formi ako, i samo ako, neključni atributi nisu ni funkcionalno ni tranzitivno zavisni od ključnog atributa, što znači da se svaka relacija koja se nalazi u trećoj normalnoj formi nalazi i u drugoj, dok obrnuto ne važi.

Definicija 4.1. Neka su X, Y, Z skupovi atributa relacije $R(A_1, \dots, A_n)$. Kaže se da skup atributa Z tranzitivno zavisi od X ako važi $X \rightarrow Y$ i $Y \rightarrow Z$. Relacija R je u trećoj normalnoj formi (u oznaci 3NF) ako je u 1NF i nijedan njen neključni atribut ne zavisi tranzitivno ni od jednog njenog ključa.

Ako između atributa unutar jedne n -torke postoje sledeće funkcionalne zavisnosti: $X \rightarrow Y$, $X \rightarrow Z$ i $Y \rightarrow Z$ onda se kaže da je atribut Z tranzitivno i funkcionalno zavistan od atributa X .

Pomenute međusobne zavisnosti atributa (funkcionalna i tranzitivna) nisu veštačke veze među atributima, nego veze koje postoje i u stvarnosti i zato se, preslikavaju iz realnog sveta u informacijski sistem. Dekompozicijom relacije ne smemo takve veze potpuno raskinuti, jer će model u tom slučaju izgubiti svoju verodostojnost. To praktično znači da svaka zavisnost unutar neke relacije mora logički slediti i u relacijama koje su generisane dekompozicijom, odnosno iz njih se mora dobiti operacijom prirodnog spajanja. U protivnom, može doći do gubitka informacija. Pravilo prilikom dekompozicije kojega se treba držati glasi:

Relacija se dekomponuje bez opasnosti gubitaka funkcionalnih zavisnosti samo ako se dekompozicija vrši prema funkcionalnoj zavisnosti koja ne ide od kandidata ključa (kao što je urađeno sa relacijama koje povezuju laboratorije i laborante).

4.2.2. Ostale normalne forme

Pored pomenutih normalnih formi postoje još i:

- **Boyce - Codd-ova normalna forma (BCNF)** koja je u stvari stroži oblik treće normalne forme i relevantna je za one šeme relacija koje imaju više kandidata za ključ, koji se pri tome još i međusobno prekrivaju. U praksi su ovakvi slučajevi retki.
- **Četvrta normalna forma** eliminiše redundansu koja je u trećoj normalnoj formi moguća ako među atributima postoji višeznačna zavisnost, **čime se eliminišu i anomalije pri održavanju.**

Posmatrajmo, na kraju, još jedan primer kako bi i ova definicija postala razumljiva.

Pretpostavimo da relacija NASTAVA ima tri atributa i to: Sifra_Predmeta#, Sifra_Nastavnika#, Sifra_Udzbenika#:

NASTAVA (Sifra_Predmeta#, Sifra_Nastavnika#, Sifra_Udzbenika#)

te da jedan predmet može da predaje više nastavnika, i da se za jedan predmet može koristiti više udžbenika. Tako u relaciji NASTAVA imamo dve višeznačne zavisnosti jer jednom predmetu (na primer fizici) pripada više nastavnika (na primer Marković, Pavlović, Marić) i više udžbenika (na primer "Fizika materijala" i "Atomska i molekularna fizika).

Ovako koncipirana relacija NASTAVA sadrži prema tome redundansu, jer se zapis o tome da pojedini nastavnik predaje neki predmet javlja onoliko puta, koliko različitih udžbenika on koristi za taj predmet.

I ova redundansa se može dekompozicijom izbeći, i tada se kaže da je sistem u četvrtoj normalnoj formi. Konkretno, u ovom slučaju relaciju NASTAVA treba predstaviti dvema relacijama:

NASTAVA1 (Sifra_Predmeta#, Sifra_nastavnika#)

NASTAVA2 (Sifra_Predmeta#, Sifra_Udzbenika#)

Peta normalna forma je poslednji stupanj dekompozicije, koja rečima iskazana glasi: relacija se nalazi u petoj normalnoj formi ako se ne može dalje, sa nekim smislom, dekomponovati, odnosno, peta normalna forma se odnosi na relacije koje predstavljaju veći broj entiteta ili njihovih odnosa (više od 2). U takvim slučajevima, moguća je dekompozicija relacije u više projekcija, bez gubljenja informacija. Ovo svojstvo označava prisustvo zavisnosti spajanja koje se zamenu relacije njenim projekcijama eliminiše.

4.3. Normalizacija u fazi relacionim modelima baza podataka

U fazi relacionim modelima baza podataka, gde su vrednosti atributa predstavljeni pomoću mogućnosti distribucije, a domeni definisani pomoću relacije bliskosti, problem anomalija i redundanse podataka i dalje postoji. Chen, Kerre i Vandenbulcke [12] proširuju teoriju normalizacije relacionog modela baza podataka kako bi obezbedili smernice za projektovanje fazi relacionih baza podataka. Normalizacija fazi baza podataka zasnovana je na pojmu fazi funkcionalnih zavisnosti, a jedan broj pojmova kao što su ključevi i normalne forme su generalizovani. Kao rezultat toga, formulišu se q-ključevi, prva fazi normalna forma (F1NF), q - druga fazi normalna forma (q-F2NF), q - treća fazi normalna forma (q-F3NF) i q- Boyce-Codd fazi normalna forma.

U fazi relacionim bazama podataka, u kojima egzistiraju fazi funkcionalne zavisnosti, ključ K relacije R(U) se može definisati na način da K funkcionalno određuje U sa određenim stepenom.

Teorijske smernice fazi normalnih formi se definišu kako bi se objasnilo grupisanje atributa u relacionim šemama. Jedan od osnovnih problema, koji egzistira u ovakvim modelima baza podataka, je postojanje delimičnih, nepotpunih i/ili tranzitivnih fazi funkcionalnih zavisnosti koji, kao i prethodnom slučaju nameću, postajanje redundantnosti i anomalija. Broj normalnih formi se definiše tako da se adekvatno definišu odnosi između atributa.

Generalna forma fazi funkcionalnih zavisnosti koja se koristi prilikom definisanja fazi normalnih formi koristi relaciju bliskosti \approx za mogućnost distribucije na odgovarajućem domenu, fazi operator implikacije I za pravila zaključivanja IF-THEN i min za operator AND. Konkretno X funkcionalno određuje Y sa stepenom q ako i samo ako je:

$$\min I(\approx(t(X), t'(X)), \approx(t(Y), t'(Y))) \geq q \quad t, t' \in R.$$

Na primer, Ako je $\approx(t(X), t'(X)) = 0.95$ i $\approx(t(Y), t'(Y)) = 0.90$ onda fazi funkcionalna zavisnost zadovoljava $X \xrightarrow{0.9} Y$.

Sada definišemo fazi normalne forme:

Definicija 4.2. Neka je D_i domen atributa A_i . Relacija se nalazi u prvoj fazi normalnoj formi (F1NF) ako i samo ako u relaciji R postoji vrednost atributa koja predstavlja isključujuću mogućnost distribucije.

Posmatrajmo sledeću relaciju koja zadovoljava uslove prve fazi normalne forme:

R (Ime, Godine, Jezik)

Ime	Godine	Jezik
Marko	21	E
Dušan	{0.7/16, 1/17, 0.8/18}	{0.3/F, 0.6/N, 1/E}
Petar	{25, 26}	F

Tabela 34. Primer F1NF

Napomenimo, da relacija nije u prvoj fazi normalnoj formi ako postoje višeznačni atributi. Na primer, ako i Dušan govori francuski sa stepenom pripadnosti 0.3 i Nemacki sa stepenom 0.6 i Engleski sa 1. Dakle, samo jednoznačni atributi, koji su po prirodi isključivi, zadovoljavaju uslove prve fazi normalne forme.

Definicija 4.3. Neka je F skup fazi funkcionalnih zavisnosti relacije R, a K q-ključ relacije. Relacija je u q-drugoj fazi normalnoj formi (q-F2NF), ako i samo ako je R u F1NF i ako za bilo koji drugi atribut A, koji nije ključni, važi $K \xrightarrow{a} A$ u F^+ (potpuna fazi funkcionalna zavisnost), $0 \leq a \leq q$.

Definicija 4.4. Neka su $X, Y \subseteq U$. Y je potpuno funkcionalno zavisno od X sa stepenom q, u oznaci $X \xrightarrow{q} Y$, ako i samo ako je $X \xrightarrow{q} Y$ i ne postoji $X' \subset X, X' \neq \emptyset$, tako da je $X' \xrightarrow{q} Y$. U suprotnom, kažemo da je Y parcijalno zavisno od X sa stepenom q.

Razmotrimo prethodnu relaciju sa $F = \{ \text{Ime} \xrightarrow{0.9} \text{Godine} \}$. $\{ \text{Ime}, \text{Jezik} \}$ je 0.9-ključ jer je:

$$\text{Ime}, \text{Jezik} \xrightarrow{0.9} \text{Godine},$$

$$\text{Ime}, \text{Jezik} \xrightarrow{1.0} \text{Ime}$$

$$\text{Ime}, \text{Jezik} \xrightarrow{1.0} \text{Jezik}.$$

Međutim, R nije u q-F2NF, jer Godine nisu potpuno zavisne od $\{ \text{Ime}, \text{Jezik} \}$.

Definicija 4.5. Neka je F skup fazi funkcionalnih zavisnosti relacije R, a K q-ključ relacije. Relacija je u q-trećoj fazi normalnoj formi (q-F3NF), ako i samo ako je R u F1NF i za bilo koju zavisnost $X \xrightarrow{f} A$ u F^+ , tako da A nije u X, važi da je ili X q-superključ ili A je q-primarni ključ.

Definicija 4.6. Neka su $K, S \subseteq U$ i F skup fazi funkcionalnih zavisnosti u relaciji R. Onda K zovemo q-ključ (ili q-ključ kandidat) relacije R ako i samo ako je $K \xrightarrow{q} U \in F^+$ i ako je $K \xrightarrow{q} U$ potpuna funkcionalna zavisnost. Ako relacija R ima više od jednog q-ključa, onda možemo da odredimo jedan od ključeva kao q-primarni ključ relacije R. S se zove q-superključ ako i samo S sadrži q-ključ.

Definicija 4.7. Neka su $A \in U, X \subseteq U$ i K q-ključ relacije R. A se zove q-primarni atribut ako i samo ako $A \in K$. Isto tako, X se zove q-primarom ako i samo ako je $X \subseteq K$. Za ostale attribute kaže se da su neključni.

Posmatrajmo relaciju R (Student#, Ime, Adresa, Godine) i skup fazi funkcionalnih zavisnosti $F = \{ \text{Student\#} \xrightarrow{0.9} \text{Godine}, \text{Student\#} \xrightarrow{0.8} \text{Adresa}, \text{Student\#} \xrightarrow{1} \text{Ime}, \text{Ime} \xrightarrow{1} \text{Student\#} \}$. Student# je 0.8-ključ. Takođe, iz zadnje fazi funkcionalne zavisnosti Ime je 0.8-ključ. Bilo koja FFD u F^+ zadovoljava ograničenja 0.8-F3NF. Prema tome, relacija je u 0.8-trećoj fazi normalnoj formi.

Definicija 4.8. Neka je F skup fazi funkcionalnih zavisnosti i K q-ključ u relaciji R. Za R kažemo da zadovoljava uslove q-fazi Boyce-Codd normalne forme, u oznaci q-FBCNF, ako i samo ako je R u F1NF i za bilo koju zavisnost $X \xrightarrow{f} A$ u F^+ , tako da A nije u X, važi da je X q-superključ.

Pozivajući se na teoriju normalizacije za klasične relacione baze podataka, kada dekomponujemo fazi relacionu šemu u veći broj jednostavnijih relacija, da bismo eliminisali neželjene zavisnosti između atributa, moramo da vodimo računa da ne dođe do gubitka informacija, a da se pri tome očuvaju postojeće zavisnosti. Zakonitost mora da bude zadovoljena u svim fazi relacijama koje su u q-trećoj fazi normalnoj formi (q-F3NF). Da bismo ovo postigli koristićemo definiciju *delimično/parcijalno slobodnih fazi funkcionalnih zavisnosti*.

Definicija 4.9. Fazi funkcionalna zavisnost $Y \xrightarrow{\theta} B$ je parcijalno slobodna ako i samo ako ne postoji $X \xrightarrow{\alpha} B$ u F^+ tako da je $X \subset Y$ i $\alpha < \theta$. Ako je svaka fazi funkcionalna zavisnost u posmatranom skupu F parcijalno slobodna, tada za skup F kažemo da je parcijalno slobodan.

Navedimo sada i teoremu na osnovu koje i vršimo dekompoziciju relacione šeme u q-F3NF relacije.

Teorema 1. Neka je R relaciona šema, F minimalni skup fazi funkcionalnih zavisnosti za R i $\rho = \{R_1, R_2, \dots, R_k\}$ dekompozicija relacione šeme R . Ako je skup F parcijalno slobodan onda ρ zadovoljava zahtev za očuvanjem zavisnosti i svaka od R_i , $i=1,2,\dots,k$ je u q-trećoj fazi normalnoj formi q-F3NF tako da je $\Pi_{U_i}(F)$, $YB=U_i$, a $Y \xrightarrow{q} B$ u F .

Dokaz. Kao prvo, ρ zadovoljava zahtev za očuvanjem zavisnosti. Drugo, mi dokazujemo da je R_i u q-F3NF. Pretpostavimo suprotno. Onda $\exists X \xrightarrow{\alpha} A$ u $\Pi_{U_i}(F)$ tako da A ne pripada X , X nije q-superključ i A nije primarni ključ.

Za slučaj $A=B$ imamo da je $XB \subseteq YB$ i B ne pripada X što implicira $X \in Y$. Nadalje, iz činjenice da X nije superključ, mi imamo da je $X \subset Y$. A kako je skup F parcijalno slobodan $Y \xrightarrow{q} B$ je parcijalno slobodna fazi funkcionalna zavisnost. Međutim, za $X \xrightarrow{\alpha} B$ mora da važi $\alpha \geq q$. A to znači da postoji $X \xrightarrow{q} B$ u F^+ koja može da zameni $Y \xrightarrow{q} B$ u F , što je kontradiktorno činjenici da je F skup sa najmanjom pokrivenošću.

Napomenimo da ako skup F nije parcijalno slobodan onda se $R_i(U_i)$ u ρ odnosi na skup fazi funkcionalnih zavisnosti $\Pi_{U_i}(F)|_{\theta} = \{V \xrightarrow{\alpha} W \mid V \xrightarrow{\alpha} W \text{ u } F^+, VW \subseteq U_i\}$.

Na primer, neka je $R = \{A, B, C, D, E\}$ i skup fazi funkcionalnih zavisnosti $F = \{CD \xrightarrow{\theta} A, CD \xrightarrow{\theta} B, AD \xrightarrow{\rho} E, CD \xrightarrow{\theta} E, A \xrightarrow{\alpha} B, B \xrightarrow{\beta} E\}$ i $\varphi > \min(\alpha, \beta)$. Prema teoremi 1. $\rho = \{CDA, CDB, ADE, CDE, AB, BE\}$. A kako F nije parcijalno slobodan zbog postojanja $A \xrightarrow{\min(\alpha, \beta)} E$ u F^+ . ADE je u φ -F3NF u odnosu na $\Pi_{ADE}(F)|_{\varphi}$, a ne u odnosu na $\Pi_{ADE}(F)$.

Teorema 2. Neka je R relaciona šema, F minimalni skup fazi funkcionalnih zavisnosti za R i K θ -ključ za R . Onda je relaciona šema R_K sa $U'=K$ u 1-F3NF u odnosu na $\Pi_K(F)$.

Dokaz. Pretpostavimo da R_K nije u 1-F3NF. Onda $\exists V \xrightarrow{\beta} B \in \Pi_K(F)$ tako da B nije u V , V nije 1-superključ za R_K i B nije 1-primarni ključ. Mi isto tako znamo da je $VB \subseteq K$ i $K \xrightarrow{1} K$.

Međutim, važi da je K 1-ključ od K (ako nije, prema definiciji za 1-ključ, mora da postoji $Z \subset K$ tako da važi $Z \xrightarrow{1} K$). Kako je $K \xrightarrow{\theta} U$, a imamo i da je $Z \xrightarrow{\theta} U$, što je kontradiktorno

činjenici da je K θ -ključ relacije šeme R). Sa druge strane, iz $VB \subseteq K$ imamo da je B 1-primarni ključ u R_K što je suprotno polaznoj pretpostavci jer važi da je R_K u 1-F3NF.

Sada datu relacionu šemu R i K kao θ -ključ od R možemo da dekomponujemo kao: $\sigma = \rho \cup R_K$, gde je ρ dekompozicija koju smo dobili algoritmom kreiranom iz teoreme 1. Napomenimo da i σ omogućava spajanje bez gubitka informacija, analogno prethodnom.

Definišimo sada i pravila koja ćemo koristiti za projektovanje fazi relacije R na osnovu poznatog skupa fazi funkcionalnih zavisnosti, a koji će da bude u q -trećoj fazi normalnoj formi:

1. Input: Fazi relacija R i skup fazi funkcionalnih zavisnosti F ,
2. Output: dekompozicija ρ fazi relacije R ,
- P1. Ako postoje neki atributi fazi relacije R koji nisu sadržani ni u jednoj fazi funkcionalnoj zavisnosti u skupu F (bilo na levoj ili desnoj strani zavisnosti), onda ovi atributi mogu formirati relaciju $\Pi_{A_i}(F)$ i njih eliminišemo iz fazi relacije R .
- P2. Ako neka od fazi funkcionalnih zavisnosti uključuje sve attribute fazi relacije R onda je $\rho = \{R\}$.
- P3. Za svaku fazi funkcionalnu zavisnosti $X \xrightarrow{\theta} Y$ u skupu F , kreiraćemo relaciju u ρ tako da je $R_i \in \rho$, $\Pi_{U_i}(F)$, $U_i = XY$.

4.4. Fizička organizacija podataka

Fizički model baze podataka određuje kako su podaci memorisani na memoriji računara i kako se njima manipuliše - danas najčešće na disku. Fizičko projektovanje je prevođenje logičkog projekta na konkretnu opremu, konkretne softvere koji uključuju konkretne baze podataka. [71] Strukture podataka (različiti tipovi datoteka), zajedno sa metodama pristupa i obrade podataka, ustvari predstavljaju fizičku organizaciju podataka.

Osnovna motivacija za izučavanje različitih fizičkih organizacija je činjenica da je vreme pristupa disku mnogo veće (za četiri do pet redova veličine) nego vreme pristupa unutrašnjoj memoriji. Zato izbor fizičke organizacije ima za cilj minimizaciju broja pristupa disku pri radu sa podacima.

Ne postoji optimalna fizička organizacija, jer se različite fizičke organizacije pokazuju manje ili više efikasnim u različitim aplikacijama. Zato sistemi za upravljanje bazama podataka obično podržavaju veći broj fizičkih organizacija, tako da se za razne delove baza podataka mogu koristiti najpogodnije strukture i algoritmi.

Najniži nivo arhitekture sistema baza podataka je unutrašnji nivo. To je fizička reprezentacija (obično na disku) cele baze podataka, niskog nivoa, koja se sastoji od velikog broja fizičkih slogova (ili, kraće, slogova¹) različitih tipova. Fizički slogovi su prema svom tipu, s tačke gledišta DBMS-a (SUBP), grupisani u datoteke koje imaju jedinstvene identifikatore.

Fizički slog odgovara logičkom slogu koji predstavlja (npr. n -torci specifične relacije), ima jedinstveni identifikator i sastoji se od niza bajtova. Prvih nekoliko bajtova sadrži informaciju o tipu fizičkog sloga, sledećih nekoliko bajtova je rezervisano za dužinu

¹ Terminologija, kao i rešenja pojedinih problema u ovoj oblasti bitno se razlikuju od sistema do sistema, ali principi fizičke reprezentacije i pristupa podacima uglavnom su standardni.

sloga u bajtovima ako je tip sloga promenljive dužine, a zatim slede oznake i vrednosti fizičkih polja.

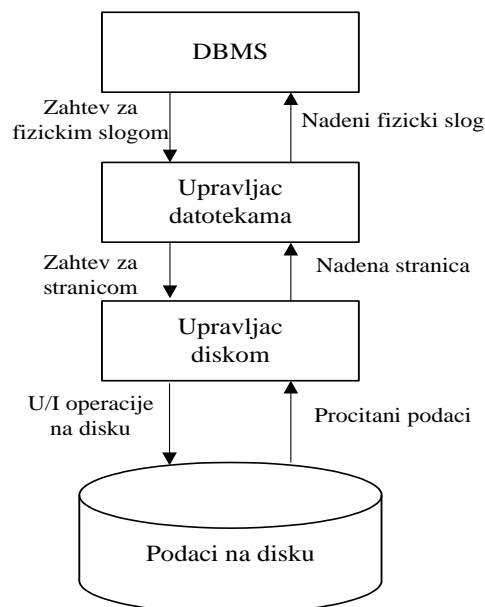
Zadatak DBMS-a je da zahteve koje korisnik postavlja na konceptualnom nivou nad logičkim tipovima podataka (npr. relacijama i n-torkama) preslika, na unutrašnjem nivou, u zahteve nad datotekama i fizičkim slogovima (preslikavanje skupa relacija u skup datoteka ne mora biti 1–1. Zahtevi koje DBMS može da proizvede na unutrašnjem nivou su sledeći:

1. Pronaći fizički slog s iz datoteke d,
2. Izmeniti fizički slog s u datoteci d,
3. Dodati novi fizički slog datoteci d i dobiti njegov identifikator,
4. Izbrisati fizički slog s iz datoteke d,
5. Kreirati novu datoteku d,
6. Ukloniti datoteku d.

Da bi se ovi zahtevi izvršili, DBMS u svojstvu korisnika operativnog sistema koristi funkcionalnost komponente operativnog sistema – upravljača datotekama (engl. file manager), a upravljač datotekama svoju funkcionalnost izgrađuje koristeći funkcije komponente operativnog sistema nižeg nivoa – upravljača diskom.

Upravljač diskom ima informacije o svim fizičkim adresama na disku, i odgovoran je za sve fizičke ulazno/izlazne operacije na disku (slika 12.). Za upravljač diskom (kao i za upravljač datotekama), struktura fizičkog sloga nije od interesa i on predstavlja samo niz bajtova na specifičnim adresama.

Upravljač datotekama “vidi” disk kao skup blokova fiksne veličine koji se zovu stranice. Stranice su numerisane tako da linearno uređenje skupa stranica odgovara njihovom fizičkom redosledu. Veličina stranice je obično izražena brojem bajtova, tj. brojem znakova. Stranice su logički grupisane u grupe stranica od kojih svaka ima svoj redni broj (tj. identifikator). Stranice (odnosno grupe stranica) su, sa tačke gledišta upravljača datotekama, dalje grupisane u disjunktne stranične skupove (engl. page sets). Svaki stranični skup ima jedinstveni identifikator. Jedan stranični skup može da sadrži fizičke slogove jednog ili raznih tipova, odnosno može da uključi i fizičke slogove koji, sa tačke gledišta DBMS-a, pripadaju različitim datotekama.



Slika 11. Slika DBMS, upravljač datotekama i upravljač diskom [71]

Preslikavanje stranice u fizičke adrese na disku koje odgovaraju toj stranici, ostvaruje upravljač diskom (slika 11.). Zato je upravljač datotekama u mogućnosti da postavi (a upravljač diskom da izvrši) sledeće vrste zahteva:

1. Pretražiti stranicu p iz straničnog skupa st ,
2. Izmeniti stranicu p u straničnom skupu st ,
3. Dodati novu stranicu straničnom skupu st i dobiti broj te stranice, p ,
4. Ukloniti stranicu p iz straničnog skupa st .

Preslikavanje datoteke u stranične skupove ostvaruje upravljač datotekama. Tako se zahtevi koji se odnose na fizički slog, a koje DBMS postavlja na unutrašnjem nivou ANSI arhitekture, realizuju pozivom upravljača tekama. On određuje broj stranice koja sadrži traženi fizički slog i poziva upravljač diskom koji realizuje ulazno/izlazne operacije nad podacima koji odgovaraju traženoj stranici (slika 11.).

Mada ne mora da učestvuje u upravljanju stranicama, DBMS mora da “zna” za njihovo postojanje, da bi izborom odgovarajuće fizičke reprezentacije povećao efikasnost pristupa podacima.

Odnos između načina na koje upravljač datotekama i DBMS “vide” podatke na disku ilustrovan je slikom 12. Stranice sa brojem 1 i 3 (sa ove slike) pripadaju jednom straničnom skupu, a stranice sa brojem 2, 4 i sve ostale stranice – drugom straničnom skupu, npr. skupu slobodnih stranica. Oznake “f.k1” – “f.k5” odnosno “f.i1” – “f.i10” označavaju fizičke slogove kojima su predstavljene na disku n -torke relacije K sa vrednostima $k1$ – $k5$ atributa K_SIF , odnosno n -torke relacije I sa vrednostima $i1$ – $i10$ atributa I_SIF (Primer 4.1). Stranice diska (kako ih vidi upravljač datotekama) obeležene su brojevima 1, 2, 3, 4, itd. Isti stranični skup na unutrašnjem nivou DBMS-a vidi se kao dve datoteke, a na konceptualnom nivou – kao dve relacije.

Mada upravljač datotekama postoji kao komponenta operativnog sistema, veći broj DBMS-a izgrađuje svoj sopstveni upravljač datotekama da bi povećao efikasnost pristupa podacima.

DBMS (Konceptualni nivo):

k1
k2
k3
k4
k5

Relacija – 1 (K)

i1
i2
i3
i4
i5
i6
i7
i8
i9
i10

Relacija – 2 (I)

DBMS (Unutrašnji nivo):

1	2	3	4
f.k1		f.k4	
f.k2		f.k5	
f.k3			

Datoteka – 1

1	2	3	4
f.i1	f.i2	f.i5	f.i6
f.i3	f.i4		
		f.i7	f.i8
		f.i9	f.i10

Datoteka – 2

Upravljač datotekama:

stranični skup 1

1	2	3	4
f.k1		f.k4	
f.i1	f.i2	f.i5	f.i6
f.i3	f.i4	f.k5	
f.k2		f.i7	f.i8
f.k3		f.i9	f.i10

Slika 12. Fizička organizacija podataka

4.4.1. Fizička reprezentacija relacije

Jedna relacija može imati fizičku reprezentaciju koja se sastoji od većeg broja datoteka (sa tačke gledišta DBMS-a), tj. od većeg broja straničnih skupova (sa tačke gledišta operativnog sistema). U tekstu koji sledi biće razmatrana samo fizička reprezentacija sa tačke gledišta DBMS-a koji uključuje funkcije upravljača datotekama.

Fizička reprezentacija jedne relacije uključuje, obavezno, datoteku podataka, u oznaci DP, koja sadrži fizičke slogove koji odgovaraju podacima koji se “vide” na logičkom nivou (npr. n-torkama relacije). Međutim, da bi se postigla što veća efikasnost obrade podataka u datoteci podataka, u zavisnosti od prirode podataka i vrste obrade, fizička reprezentacija može da uključi i druge datoteke – datoteku promena, datoteku prekoračenja, u koje može da se “širi” datoteka podataka, i datoteke koje ubrzavaju pristup podacima u datoteci podataka i koje se zovu indeksne datoteke. Tako se, uz odgovarajuće algoritme obrade podataka, dolazi do različitih fizičkih organizacija podataka – sekvencijalne, indeks-sekvencijalne, indeksne, direktne.

Efikasnost pojedine fizičke organizacije podataka izražava se vremenom obrade traženog fizičkog sloga, a to vreme uključuje vreme pristupa grupi stranica, vreme pristupa stranici unutar grupe (koja sadrži traženi slog), vreme donošenja odabrane stranice sa diska u unutrašnju memoriju odnosno vreme upisa stranice iz unutrašnje memorije na odabranu poziciju, kao i vreme obrade sloga u unutrašnjoj memoriji. Vreme pristupa grupi stranica i vreme pristupa stranici unutar grupe zavisi od karakteristika uređaja (brzina pomeranja glave diska), i neće se uzimati u obzir pri daljem razmatranju. Zato će se vreme čitanja, odnosno upisa stranice smatrati jednakim za sve stranice na disku. Takođe, vreme obrade stranice u unutrašnjoj memoriji je za nekoliko redova veličine manje od vremena čitanja, odnosno upisa stranice na disk, pa se ni ono neće uzimati u obzir. Tako će jedina mera efikasnosti pojedine fizičke organizacije biti broj stranica koje je potrebno upisati, odnosno pročitati sa diska, u realizaciji upita pri toj fizičkoj organizaciji.

Razmotrimo sledeći primer. Neka datoteka podataka sadrži n slogova grupisanih u stranice, od kojih svaka može da primi po m slogova. Dakle, datoteka podataka sadrži (logički, jer stranični skup koji joj odgovara može da sadrži i slogove drugog tipa) n/m stranica. Neka je vreme čitanja, odnosno upisa pojedine stranice jednako t . Pretpostavimo da je potrebno obraditi sve slogove iz te datoteke podataka. Ako se slogovima pristupa serijski (redom, kako su upisani), za obradu svih slogova potrebno je vreme:

$$(n/m) \times t,$$

tj. vreme obrade pojedinačnog sloga pri ovakvom pristupu je:

$$((n/m) \times t) = t/m.$$

Ako postoji mogućnost direktnog pristupa slogu (po nekom svojstvu tog sloga), i ako slogovi nisu grupisani po tom svojstvu, realna je pretpostavka da se dva sloga koja se obrađuju jedan za drugim nalaze na različitim stranicama. Tada je za obradu cele datoteke podataka potrebno vreme:

$$n \times t,$$

tj. vreme obrade pojedinačnog sloga pri ovakvom pristupu je:

$$n \times t / n = t.$$

Ovo razmatranje dovodi do zaključka da, kada je potrebno obraditi veliki broj slogova iz datoteke podataka, vreme obrade po slogu, pa i celokupno vreme obrade svih slogova daleko je manje u slučaju da se slogovima pristupa sekvencijalno – u fizičkom redosledu stranica, nego u slučaju da je pristup stranicama slobodan (“preko reda”).

4.5. Zaključak

U ovom poglavlju opisan je način i pristup projektovanju relacionih i fazi relacionih baza podataka jer projektovanje predstavlja osnovni poligon za primenu rezultata i analiza iz oblasti funkcionalnih zavisnosti, tj. funkcionalne i fazi funkcionalne zavisnosti igraju veoma važnu ulogu u dizajniranju baza podataka. Problemi redundancije i pojave anomalija u bazama podataka navode nas na zaključak da baza nije dobro projektovana. Upravo iz tog razloga, data je kraća analiza i smernice koje obezbeđuju teorijsku osnovu u dizajniranju istih.

Pored toga, prikazana je i fizička organizacija podataka u bazi, kako bi se stvorila jasna slika celokupnog koncepta i pristupa projektovanju.

Naš cilj je da rezultate razmatranja fazi funkcionalnih zavisnosti u ovoj doktorskoj disertaciji primenimo i implementiramo u dizajniranju baza podataka, što je i urađeno u šestom poglavlju, jer oblast veštačke inteligencije, teorija fazi skupova i drugi principi prikazani u ovom radu izučavaju se sa aspekta dizajniranja FRDBM.

5. Analiza fazi funkcionalnih zavisnosti u relacionim bazama podataka

5.1. Uvodna razmatranja

U ovom poglavlju predstavimo vezu između teorije fazi funkcionalnih zavisnosti u fazi relacionim bazama podataka i koncepta fazi logike, tj. fazi implikacija. Hipoteza od koje smo krenuli u dalja istraživanja jeste da prilikom ispitivanja fazi funkcionalnih zavisnosti svoje mesto imaju i metode zasnovane na fragmentu fazi logike. S obzirom da fazi logika opisuje nepreciznost ljudske percepcije odgovarajućim matematičkim alatom, ovu metodologiju možemo da iskoristimo u identifikaciji relacija između podataka skladištenih u fazi relacijama i na taj način utvrdimo skrivene zavisnosti između atributa relacije koje nisu uočljive na prvi momenat. Skup fazi funkcionalnih zavisnosti možemo dobiti analizom fazi baze podataka i primenom fazi implikacija što je i potvrđeno analizom u ovoj doktorskoj disertaciji.

Otkrivanje potencijalnih zavisnosti između atributa posmatrane fazi relacije ustvari predstavlja pronalaženje skrivenih i korisnih znanja odnosno rudarenje podataka, koje možemo da koristimo za donošenje upravljačkih odluka. Da bismo identifikovali fazi funkcionalne zavisnosti u odgovarajućoj fazi relaciji potrebno je:

1. Pripremiti podatke za analizu,
2. Pronaći podrelacije (parove n-torki) na posmatranim atributima u kojima se potencijalno kriju fazi funkcionalne zavisnosti,
3. Pridružiti fazi funkcionalnim zavisnostima odgovarajuće fazi implikacije,
4. Interpretirati rezultate istraživanja da bismo dobili konačan skup fazi funkcionalnih zavisnosti.

Dakle, u ovoj disertaciji uvodimo relativno novu tehniku i algoritam za računanje fazi funkcionalnih zavisnosti. Koliko je poznato, predložena metodologija nije pronađena u postojećoj bibliografiji. U svakom slučaju, dostupni pristupi i koncepti su navedeni u prethodnim poglavljima.

Pronalaženje funkcionalnih zavisnosti u relacionim bazama podataka je predmet interesovanja rudarenje podataka, jer su one veoma informativne za donošenje upravljačkih odluka, a naročito u situacijama kada ne raspolažemo sa preciznim i egzaktinim informacijama. U tom slučaju menadžeri mogu da donose odluke i procene u realnom vremenu na bazi lingvističkih varijabli. Da bismo ovo ilustrovali, posmatrajmo sledeći jednostavan primer: ako želimo da opišemo odnos između atributa fazi relacije *Dužina puteva* i *Finansijska sredstva* fazi funkcionalnom zavisnošću

$Dužina\ puteva \xrightarrow{0.8} Finansijska\ sredstva$ (Dužina puteva prilično određuje potrebna finansijska sredstva projekta sa stepenom jačine 0.8), onda možemo da preferiramo pravilima kao što su "IF *Dužina puteva* velika THEN *Finansijska sredstva* su veoma visoka". Poznavajući ovakav skup pravila fazi rezonovanjem (aproksimativnim zaključivanjem) možemo da kreiramo jedinstvenu upravljačku strategiju i generišemo adekvatne zaključke, a defazifikacijom dođemo i do konkretnih vrednosti i procena neophodnih za donošenje odluka. Odavde zaključujemo da fazi funkcionalne zavisnosti možemo da koristimo i prilikom estimacije nepoznatih vrednosti atributa. Za procenu i detektovanje ovih pravila odnosno fazi funkcionalnih zavisnosti neophodno je razviti poseban metod koji može da se upotrebi kao specifičan alat, naročito interesantan u analizi

statističkih modela baza podataka, geografskih informacionih sistema, sistema za pronalaženje informacija...

5.2. Model fazi relacije baze podataka

U ovom radu analiziraćemo postojanje fazi funkcionalnih zavisnosti u fazi relaciji na bazi relacije bliskosti i fazi relaciji na bazi mogućnosti distribucije (fazi model zasnovan na fazi skupovima). Prvi model je pogodan za opisivanje i upravljanje atributima koji vrednost uzimaju iz diskretnih domena, dok drugi predloženi model više odgovara situaciji kada su vrednosti atributa iz numeričkog domena. Mi želimo da razmotrimo postojanje fazi funkcionalnih zavisnosti u oba modela kako bismo dokazali predloženi metod i verifikovali rezultate samostalnog istraživanja.

Neka je **Radnik** (Ime_Prezime, Godine, Iskustvo, Produktivnost, Jezik, Pozicija, Zarada) fazi baza podataka zasnovana na relaciji bliskosti. Domeni atributa *Ime_Prezime* i *Pozicija* su klasični skupovi, dok su domeni atributa *Godine*, *Iskustvo*, *Produktivnost*, *Jezik* i *Zarada* fazi podskupovi. Predstavljajući podatke na ovaj način mi smo uveli određene nepreciznosti i neodređenosti u relacionu bazu podataka i dobili fazi relaciju. Dakle, u prvoj predloženoj relaciji kao vrednosti atributa mogu da se pojave lingvističke varijable ili podskupovi diskretnog domena.

Ime_Prezime	Godine	Iskustvo	Produktivnost	Jezik	Pozicija	Zarada
IP1	{veoma mlad}	{početnik}	{zadovoljavajuća}	{Engleski, Ruski}	Sistem inženjer	{prosečna}
IP2	{star}	{specijalista, ekspert}	{dobra, odlična}	{Engleski}	Menadžer	{prosečna, visoka}
IP3	{mlad, srednjih godina}	{senior, specijalista}	{dobra}	{Francuski}	Direktor	{veoma visoka}
IP4	{veoma mlad, mlad}	{početnik, senior}	{loša}	{Ruski}	Prodaja	{mala, prosečna}
IP5	{mlad}	{senior}	{zadovoljavajuća, dobra}	{Engleski, Francuski}	Nabavka	{visoka}

Tabela 35. Fazi model na bazi relacije bliskosti

Domeni atributa posmatrane fazi relacije su:

Dom (Godine) = {veoma mlad, mlad, srednjih godina, star}

Dom (Iskustvo) = {početnik, senior, specijalista, ekspert}

Dom (Produktivnost) = {loša, zadovoljavajuća, dobra, odlična}

Dom (Jezik) = {Engleski, Francuski, Ruski}

Dom (Pozicija) = {nabavka, prodaja, sistem inženjer, menadžer, direktor}

Dom (Zarada) = {mala, prosečna, visoka, veoma visoka}

Tabela 36. prikazuje relaciju bliskosti na domenu atributa Godine.

S_{Godine}	veoma mlad	mlad	srednjih godina	star
veoma mlad	1	0.95	0.70	0.30
mlad		1	0.90	0.40
srednjih godina			1	0.50
star				1

Tabela 36. Relacija bliskosti na domenu Godine

Tabela 37. ilustruje relaciju bliskosti na domenu Iskustvo.

S _{Iskustvo}	početnik	senior	Specijalista	ekspert
početnik	1	0.70	0.70	0.30
senior		1	0.85	0.30
specijalista			1	0.50
ekspert				1

Tabela 37. Relacija bliskosti na domenu Iskustvo

Tabela 38. ilustruje relaciju bliskosti na domenu Produktivnost.

S _{Produktivnost}	loša	zadovoljavajuća	dobra	odlična
loša	1	0.80	0.50	0.10
zadovoljavajuća		1	0.70	0.50
dobra			1	0.80
odlična				1

Tabela 38. Relacija bliskosti na domenu Produktivnost

Tabela 39. ilustruje relaciju bliskosti na domenu Jezik.

S _{Jezik}	Engleski	Francuski	Ruski
Engleski	1	0.60	0.20
Francuski		1	0.30
Ruski			1

Tabela 39. Relacija bliskosti na domenu Jezik

Tabela 40. ilustruje relaciju bliskosti na domenu Pozicija.

S _{Pozicija}	sistem inženjer	menadžer	direktor	prodaja	nabavka
sistem inženjer	1	0.70	0.60	0.50	0.50
menadžer		1	0.90	0.60	0.60
direktor			1	0.70	0.70
prodaja				1	0.80
nabavka					1.00

Tabela 40. Relacija bliskosti na domenu Pozicija

Tabela 41. ilustruje relaciju bliskosti na domenu Zarada.

S _{Zarada}	mala	prosečna	visoka	veoma visoka
mala	1	0.85	0.55	0.25
prosečna		1	0.75	0.35
visoka			1	0.80
veoma visoka				1

Tabela 41. Relacija bliskosti na domenu Zarada

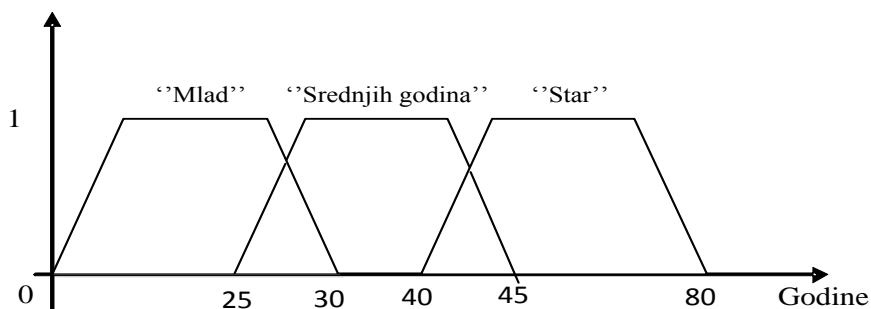
Prilikom određivanja relacije bliskosti nad elementima posmatranih domena koriste se ekspertska znanja, tj. možemo da ih dobijemo intervjuisanjem eksperata. Napomenimo i to da ćemo zbog prirode problema i potreba daljih istraživanja usvojiti da je sličnost između vrednosti atributa Ime_Prezime 0.7, tj. sličnost između imena i prezimena osoba u fazi relaciji jednaka je minimalnoj jačini zavisnosti između atributa relacije koju ćemo koristiti prilikom ispitivanja postojanja fazi funkcionalnih zavisnosti.

Druga relacija u kojoj ćemo tragati za fazi funkcionalnim zavisnostima jeste fazi model na bazi mogućnosti distribucije. Formalno, neka je **Radnik** (Ime_Prezime, Godine, Iskustvo, Produktivnost, Jezik, Pozicija, Zarada) relaciona šema i r fazi relacija nad relacijskom šemom Radnik. Vrednost atributa A u n-torci t_i označićemo sa a_i , a pridruženi stepen pripadnosti $\mu(A_{t_i})$, te dobijamo uređeni par $(a_i, \mu(A_{t_i}))$. U ovom slučaju nešto što je lingvistički izraz možemo da zemenimo sa fazi vrednošću (fazi brojem). Na ovaj način možemo da fazifikujemo relacionu bazu podataka i dobijemo GEFRED model za potrebe istraživanja i ispitivanja različitih tema u ovoj oblasti ili da generalizujemo postojeću bazu, kako bismo pretraživali skrivena znanja. U ovakvim situacijama, prilikom definisanja funkcija pripadnosti fazi atributa relacije r, uvek je neophodno konstatovati se sa ekspertima – projektantima IS i koristiti softvere za fazifikaciju i simulaciju vrednosti posmatranog atributa. Tabela 42. prikazuje fazi relaciju r na bazi mogućnosti distribucije koja će nam poslužiti za dalja istraživanja i dokazivanje ekvivalentnosti između fazi implikacija i fazi funkcionalnih zavisnosti.

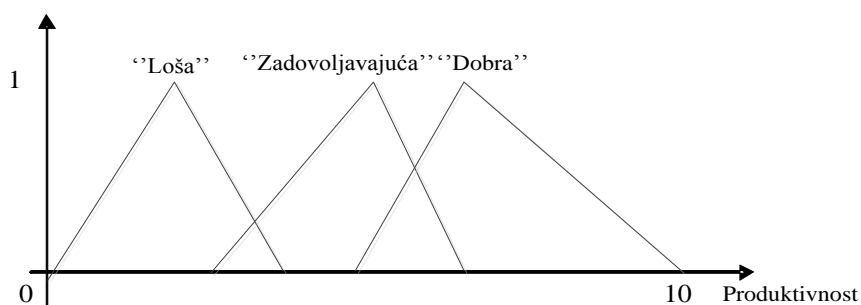
Ime_Prezime	Godine	Iskustvo	Produktivnost	Jezik	Pozicija	Zarada
IP1, 1	$g_1, 0.61$	$i_1, 0.76$	$p_2, 0.65$	Engleski, 1	Sistem inženjer, 1	$z_2, 0.81$
IP2, 1	$g_3, 0.21$	$i_3, 0.46$	$p_3, 0.79$	Engleski, 1	Menadžer, 1	$z_3, 0.92$
IP3, 1	$g_2, 0.82$	$i_2, 0.71$	$p_2, 0.82$	Francuski, 1	Direktor, 1	$z_4, 0.89$
IP4, 1	$g_2, 0.78$	$i_1, 0.89$	$p_1, 0.90$	Ruski, 1	Prodaja, 1	$z_1, 0.67$
IP5, 1	$g_1, 0.85$	$i_2, 0.73$	$p_3, 0.60$	Francuski, 1	Nabavka, 1	$z_2, 0.32$

Tabela 42. Fazi relacija r

U ovoj relaciji pretpostavili smo da atributi *Godine*, *Iskustvo*, *Produktivnost*, *Jezik* i *Zarada* mogu da uzme neku od predefinisanih vrednosti. Na primer, vrednost atributa *Godine* može da bude g_1 , g_2 , g_3 što može da odgovara lingvističkim izrazima mlad, srednjih godina i star, respektivno. Dakle, vrednost g_2 u posmatranoj n-torci pripada fazi skupu "Srednjih godina" sa stepenom pripadnosti $\mu_i(g_2)$, gde je i oznaka i-te n-torke. Analogno tome, vrednosti z_1 , z_2 , z_3 , z_4 mogu da pripadaju fazi skupovima "Mala zarada", "Prosečna zarada", "Visoka zarada", "Veoma visoka zarada" sa odgovarajućim stepenima pripadnosti. Attribute koji ne mogu da uzmu numeričku vrednost iz posmatranog fazi skupa, kao što su Iskustvo i Produktivnost, možemo da skaliramo. Tako fazi skupove "Loša produktivnost", "Zadovoljavajuća produktivnost" i "Dobra produktivnost" možemo da predstavimo trougaonim fazi brojevima (a_i, b_i, c_i) , gde Produktivnost predstavljamo i ocenjujemo na skali od 1 do 10. Isto važi i za atribut Iskustvo, a to znači da vrednosti atributa pripadaju nekom od fazi skupova "Nedovoljno iskusan", "Srednje iskusan" i "Veoma iskusan". Atributi Ime_Prezime, Jezik i Pozicija sadrže crisp informacije, a da bismo napravili ekvivalenciju sa fazi skupovima pisaćemo da je stepen pripadnosti kojom odgovarajuća vrednost pripada nekom od ovih atributa jednaka 1. Predstavljajući podatke na ovaj način možemo da uspostavimo vezu između dva predložena modela, a koji model ćemo da istražujemo i ispitujemo zavisi od naših potreba i prirode problema, te dostupnosti podataka koje prikupljamo i ekspertskih znanja za procenu lingvističkih i neodređenih izraza. Ta znanja ustvari nam omogućavaju definisanje odgovarajućih funkcija pripadnosti. Na sledećim slikama prikazani su primeri funkcija pripadnosti za attribute *Godine* i *Produktivnost*, koje možemo da definišemo i koristimo za potrebe određivanja stepena pripadnosti pojedinih vrednosti atributa. Za ovakvu predstavu atributa fazi relacije neophodna su i odgovarajuća ekspertiska znanja, koja možemo da dobijemo od stručnjaka iz ove oblasti.



Slika 13. Primer funkcije pripadnosti za atribut Godine



Slika 14. Primer funkcije pripadnosti za atribut Produktivnost

Međutim, da bismo pristupili problemu utvrđivanja fazi funkcionalnih zavisnosti, nama je neophodno da poznajemo stepen sličnosti između odgovarajućih fazi vrednosti. Iz tog razloga uvodimo fazi relaciju bliskosti na posmatranim atributima, u oznaci s_{A_i} . Za ovu relaciju pretpostavljamo da ima osobine simetričnosti i refleksivnosti i kao takvu usvajamo je za vrednost koja predstavlja stepen sa kojom se n -torke t_1 i t_2 slažu na atributu A_i . Po ovom osnovu mi definišemo fazi funkcionalne zavisnosti. U sledećim tabelama prikazane su fazi relacije bliskosti za attribute *Godine*, *Iskustvo*, *Produktivnost* i *Zarada*.

S_{Godine}	g_1	g_2	g_3
g_1	1	0.70	0.30
g_2		1	0.50
g_3			1

Tabela 43. Fazi relacija bliskosti za domen Godine

S_{Iskustvo}	i_1	i_2	i_3
i_1	1	0.80	0.40
i_2		1	0.60
i_3			1

Tabela 44. Fazi relacija bliskosti za domen Iskustvo

$S_{\text{Produktivnost}}$	p_1	p_2	p_3
p_1	1	0.75	0.45
p_2		1	0.65
p_3			1

Tabela 45. Fazi relacija bliskosti za domen Produktivnost

SZarada	Z ₁	Z ₂	Z ₃	Z ₄
Z ₁	1	0.95	0.55	0.25
Z ₂		1	0.90	0.45
Z ₃			1	0.90
Z ₄				1

Tabela 46. Fazi relacija bliskosti za domen Zarada

Ova dva pristupa predstavljaju različite okvire za predstavljanje fazi podataka, o kojima je bilo reči u prethodnim poglavljima. Fazi relacije (Tabela 35. i Tabela 42.) poslužiće nam za dokazivanje polazne hipoteze doktorske disertacije. Međutim, na predloženim relacijama možemo da istražujemo i druge teme iz oblasti teorije fazi relacionih baza podataka, kao što su projektovanje i dizajn baze, postojanje redundantnih podataka, dekompozicija bez gubitka zavisnosti, primena Armstrongovih aksioma, procena NULL vrednosti, a o čemu smo takođe pisali u prethodnim poglavljima ovog rada. Fokus istraživanja stavili smo na utvrđivanje postojanja zavisnosti između atributa datih relacija, te utvrđivanje jačine (stepana) zavisnosti primenom fazi implikacija, čime smo predstavili novi pristup u izvođenju fazi funkcionalnih zavisnosti i obezbedili teorijsku osnovu za primenu teorije normalizacije u fazi relacionim bazama podataka koja je opisana u četvrtom poglavlju ovog rada.

5.3. Pretraga fazi funkcionalnih zavisnosti u FRDBMs

Polazeći od definicije fazi funkcionalnih zavisnosti, ideja je da, pošto $X \xrightarrow{\theta} Y$ možemo da posmatramo kao pravilo koje se odnosi na bliskost vrednosti atributa na odgovarajućem paru n-torki $t_i t_j$, interpretiramo koncept u kojem možemo da otkrivamo moguće odnose između atributa datih relacija. Dakle, zamisao je da identifikujemo podrelacije (parove n-torki) na posmatranim atributima u kojima se potencijalno kriju fazi funkcionalne zavisnosti jer na taj način dolazimo do prethodno nepoznatih i potencijalno korisnih saznanja iz same baze. Podrelacija $t_i t_j$ se može upotrebiti za analizu podataka u datim fazi relacijama koristeći tumačenje Berzala i ostalih. Shodno tome, moguće relacije između vrednosti atributa iskoristićemo za detektovanje fazi funkcionalnih zavisnosti nad posmatranim skupom atributa, a za postizanje tog cilja iskoristićemo mogućnosti fazi logike.

Za potrebe istraživanja u ovoj doktorskoj disertaciji modifikovali smo relaciju T_r kao T-skup transakcija, i to na sledeći način:

- $|T_r| = |r \times r|$, gde je r posmatrana fazi relacija.
- Za svaki atribut $A_i \in r$, računamo stepen saglasnosti između svih parova n-torki $C(A_i[t_i, t_j])$.
- T_r predstavlja relaciju čije su kolone $C(A_i[t_i, t_j])$, a redovi n-torke $t_i t_j$ tako da vredi:

$$\forall t_i, t_j \in r, i \neq j \text{ par n-torki } (t_i t_j) \in r \times r, \text{ tj. } t_i t_j \in T_r,$$

$$C(A_i[t_i, t_j]) \in [0, 1].$$

- θ_k predstavlja kritičnu jačinu zavisnosti koja predstavlja parametar za detektovanje zavisnosti između n-torki posmatrane relacije. θ_k je realan broj iz $[0, 1]$, tj. $\theta_k \in [0, 1]$.

U tabeli 47. prikazana je relacija T_r prema prethodnim interpretacijama.

$\theta = \theta_k$	$C(A_i[t_i, t_j])$	$C(A_j[t_i, t_j])$	$C(A_k[t_i, t_j])$
$t_1 t_2$	$C(A_i[t_1 t_2])$	$C(A_j[t_1 t_2])$	$C(A_k[t_1 t_2])$
$t_1 t_3$	$C(A_i[t_1 t_3])$	$C(A_j[t_1 t_3])$	$C(A_k[t_1 t_3])$
$t_1 t_4$	$C(A_i[t_1 t_4])$	$C(A_j[t_1 t_4])$	$C(A_k[t_1 t_4])$
$t_1 t_5$	$C(A_i[t_1 t_5])$	$C(A_j[t_1 t_5])$	$C(A_k[t_1 t_5])$
$t_2 t_3$	$C(A_i[t_2 t_3])$	$C(A_j[t_2 t_3])$	$C(A_k[t_2 t_3])$
$t_2 t_4$	$C(A_i[t_2 t_4])$	$C(A_j[t_2 t_4])$	$C(A_k[t_2 t_4])$
$t_2 t_5$	$C(A_i[t_2 t_5])$	$C(A_j[t_2 t_5])$	$C(A_k[t_2 t_5])$
$t_3 t_4$	$C(A_i[t_3 t_4])$	$C(A_j[t_3 t_4])$	$C(A_k[t_3 t_4])$
$t_3 t_5$	$C(A_i[t_3 t_5])$	$C(A_j[t_3 t_5])$	$C(A_k[t_3 t_5])$
$t_4 t_5$	$C(A_i[t_4 t_5])$	$C(A_j[t_4 t_5])$	$C(A_k[t_4 t_5])$

Tabela 47. Relacija T_r dobijena iz fazi relacije r

Za procenu zavisnosti primenjujemo definiciju saglasnosti (conformance) $C(A_i[t_i, t_j])$ na paru n -torke (podrelaciji $r(R)$ date relacijske šeme). U fazi relacionom modelu na bazi relacije bliskosti $C(A_i[t_i, t_j])$ računamo po sledećoj formuli:

$$C(A_i[t_i, t_j]) = \min \left\{ \min_{x \in d_i} \left\{ \max_{y \in d_j} \{s(x, y)\} \right\}, \min_{x \in d_j} \left\{ \max_{y \in d_i} \{s(x, y)\} \right\} \right\},$$

gde je d_i vrednost atributa A_i za n - torke t_i , d_j je vrednost atributa A_i za n - torke t_j , $s(x, y)$ je relacija bliskosti za vrednosti x i y i s je preslikavanje svakog para elemenata iz domena D_i u interval $[0, 1]$, tj. relacija bliskosti na domenu D_i .

U fazi relacionom modelu na bazi mogućnosti distribucije $C(A_i[t_i, t_j])$ računamo prema sledećem:

$$C(A_i[t_i, t_j]) = \min (\mu_{t_i}(A_i), \mu_{t_j}(A_i), s(t_i(A_i), t_j(A_i))),$$

gde je $\mu_{t_i}(A_i)$ stepen pripadnosti vrednosti atributa A_i n -torci t_i , $\mu_{t_j}(A_i)$ stepen pripadnosti vrednosti atributa A_i n -torci t_j , a $s(t_i(A_i), t_j(A_i))$ fazi relacija bliskosti na domenu D_i .

Ako je $\theta = \theta_k$, tada možemo da formiramo tabelu (tabela 48.) u kojoj sa \top označavamo da je $C(A_i[t_i, t_j]) \geq \theta_k$, tj. stepen bliskosti posmatranih n -torke na atributu A_i je veći od kritične jačine zavisnosti. Označimo dalje sa \perp situaciju kada je stepen saglasnosti za dve n -torke na posmatranom atributu A_i manji od θ_k , $C(A_i[t_i, t_j]) < \theta_k$. Na ovaj način možemo preciznije da sagledamo i uočimo potencijalne zavisnosti između podataka u fazi relaciji, ali za određivanje i računanje paramtera θ mi moramo da koristimo predloženi algoritam baziran na fazi operatorima implikacije. Za povezivanje tj. agregaciju parova n -torke u kojima postoje uočene zavisnosti koristi se minimum t -norma.

Interesantna stvar je i ideja da merimo procenat n -torke u kojima se fazi funkcionalna zavisnost pojavljuje. Sledeći pristup u ovom radu, mi otkrivamo fazi funkcionalne zavisnosti, ali i uvodimo metriku pomoću koje određujemo proporciju n -torke P u fazi relaciji gde su fazi funkcionalne zavisnosti sadržane.

Formalno, mi dobijamo parametar P na sledeći način:

$$P = \frac{C}{C_{uk}}$$

gde je C broj podrelacija u kojima postoji posmatrana fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, a C_{uk} ukupan broj parova n-torki relacije T_r . C_{uk} računamo po sledećoj formuli:

$$C_{uk} = \frac{n!}{k!(n-k)!}$$

gde je n broj redova (n-torki) fazi relacije, a k=2 s obzirom da se radi o parovima n-točlanog skupa.

$\theta = \theta_k$	$C(A_i[t_i, t_j])$	$C(A_j[t_i, t_j])$	$C(A_k[t_i, t_j])$
$t_1 t_2$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_1 t_3$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_1 t_4$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_1 t_5$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_2 t_3$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_2 t_4$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_2 t_5$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_3 t_4$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_3 t_5$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$
$t_4 t_5$	$\top \setminus \perp$	$\top \setminus \perp$	$\top \setminus \perp$

Tabela 48. True i False vrednosti za $C(A_i[t_i, t_j]) > \theta_k$

5.4. Uloga fazi logike u ispitivanju fazi funkcionalnih zavisnosti

U klasičnim relacionim bazama podataka funkcionalna zavisnost $X \rightarrow Y$ se predstavlja klasičnom implikacijom I_c . Dakle, u klasičnoj logici pravilo IF X THEN Y pišemo u obliku $X \rightarrow Y$. "X povlači Y", u našem slučaju, znači da jednakim X-vrednostima odgovaraju jednake Y-vrednosti. Potpuna interpretacija ove operacije ukazuje na to da je Y bar toliko tačno koliko je tačno X odnosno:

$$X \rightarrow Y \text{ je tačno} \Leftrightarrow \tau(X) \leq \tau(Y)$$

$$X \rightarrow Y = \begin{cases} 1, & \text{if } t(X) = t'(X) \text{ then } t(Y) = t'(Y) \\ 0, & \text{ostalo} \end{cases}$$

Međutim, u fazi relacionim bazama podataka gde u suštini vrednosti atributa ne moraju da budu pojedinačni elementi domena, već i podskupovi osnovnog skupa ili mogućnost distribucije, stepen funkcionalne zavisnosti ne mora implicitno da bude jednak 1 kao u crisp slučajevima. Prirodno, može da se očekuje da to bude vrednost iz intervala [0,1]. Zbog toga definiciju funkcionalnih zavisnosti generalizujemo i dobijamo fazi funkcionalne zavisnosti FFD. U tom slučaju umesto klasične implikacije, FFD predstavljamo fazi implikacijom, a relaciju jednakosti zamenjujemo relacijom bliskosti.

Fazi implikacija je tada proširenje klasične implikacije $X \rightarrow Y$ na domenu $\{0,1\}$. Dakle, "IF-THEN" pravilo u slučaju fazi funkcionalnih zavisnosti predstavljamo fazi implikacijom I_f i pišemo $X \xrightarrow{\theta} Y$, tj. $\min I_f(t(X)=_c t'(X), t(Y)=_c t'(Y)) \geq \theta$, $t, t' \in r$, gde je $\theta \in [0,1] =_c [0,1] \times [0,1] \rightarrow [0,1]$ mera bliskosti i $I: [0,1] \times [0,1] \rightarrow [0,1]$ je fazi operator implikacije. Prema tome, funkcionalnim zavisnostima pridružujemo odgovarajuću implikaciju I , kao što je navedeno:

$$I = \begin{cases} I_c, & \text{if } t(X) = t'(X) \text{ then } t(Y) = t'(Y) \\ I_f, & \text{if } t(X) \approx t'(X) \text{ then } t(Y) \approx t'(Y) \end{cases}$$

U ovoj doktorskoj disertaciji naš cilj je da iskoristimo fazi implikacije i postavimo teorijske osnove algoritma za pronalaženje fazi funkcionalnih zavisnosti. Ovde ćemo odgovarajućim FFD pridružiti fazi implikacije i ispitivati postojanje zavisnosti, tj. umesto fazi zavisnosti posmatraćemo fazi implikacije. Dalji naš cilj jeste da ova teorijska razmatranja implementiramo na predloženim modelima fazi relacija, a nadalje da iskoristimo dobijene rezultate za potrebe projektovanja i dizajniranja fazi baze podataka.

5.5. Karakterizacija fazi implikacija u fazi modelima baza podataka

Osnovni doprinos dokorskog rada je analiza fazi implikacija u smislu njihovog ponašanja u bilo kom modelu fazi relacionih baza podataka. Mi razmatramo pod kojim uslovima ovi fazi operatori imaju smisao iz inženjerske perspektive da se primenjuju u analizi fazi funkcionalnih zavisnosti. Sa naše tačke gledišta postoje faze implikacije koje zadovoljavaju kriterijume prihvatanja u problemima ispitivanja i identifikovanja zavisnosti između podataka. Sa druge strane, fazi operatori kao što su Lukasiewicz-eva i Gödel-ova implikacija, ne mogu da se koriste u ovim konotacijama. Mi želimo da okarakterišemo pravilno korišćenje implikacija u analizi fazi funkcionalnih zavisnosti u datim fazi relacijama. Tragajući za osnovnim svojstvima fazi implikacija u kontekstu njihove primene u raznim aplikacijama, za prihvatljive se smatraju one koje verifikuju osobine interpretirane u ovom radu.

Mi možemo da razmatramo ponašanje različitih fazi implikacija u analizi fazi funkcionalnih zavisnosti. Fazi implikacije koje su odabrane za analizu i upotrebu u ovoj doktorskoj disertaciji prikazane su u tabeli 49.

Fazi implikacija	Korespodentna formula za $X \xrightarrow{\theta} Y$
Mamdani	$I_{MM} = \min(x, y)$
Lukasiewicz	$I_L = \min(1, 1 - x + y)$
Kleene-Diense	$I_{KL} = \max(1 - x, y)$
Early-Zade	$I_{EA} = \max(1 - x, \min(x, y))$
Willmot	$I_w = \min[\max(1 - x, y), \max(x, 1 - x), \max(y, 1 - y)]$

Tabela 49. Fazi implikacije

Cilj doktorskog rada i sprovedenog istraživanja je da pokažemo da ovaj pristup, za odgovarajuće fazi implikacije, daje željene rezultate kada je u pitanju procena jačine zavisnosti između podataka, tj. određivanje jačine fazi funkcionalnih zavisnosti. Ova studija daje nam odgovore na neka kritična pitanja kada koristimo koncept fazi logike u razvoju sistema zasnovanih na znanju. U ovom poglavlju uspostavili smo vezu između IF-THEN operatora, tj. fazi implikacija i posmatranih relacija u kontekstu ispitivanja zavisnosti između podataka relacije. Efikasnost predloženog pristupa analiziraćemo kroz izlazni skup fazi funkcionalnih zavisnosti. Istu metodologiju primenićemo i na problem dizajniranja fazi relacije.

Fazi implikacije imaju niz atraktivnih karakteristika koje nam omogućavaju da dokažemo početnu hipotezu istraživanja. Prvo, pogodne su za primenu u fazi relacijama jer je moguće izraziti jačinu zavisnosti između posmatranih atributa stvarnom realnom vrednošću. Drugo, fazi implikacije obezbeđuju prirodni okvir koji inkorporira IF-THEN pravila neophodna za razvoj novih tehnika pretraživanja znanja.

Prvi korak koji je neophodno sprovesti da bi se uspešno realizovalo istraživanje jeste preciziranje metoda logičkog objašnjenja. Dakle, neophodno je definisati kriterijume prema kojima će fazi implikacije biti zadovoljive u kontekstu ispitivanja fazi funkcionalnih zavisnosti u fazi bazama podataka. U bogatoj literaturi iz fazi logike, predložen je veliki broj fazi implikacija. Naš doprinos u ovom radu ogleda se u interpretaciji operatora implikacije pomoću kojih možemo da razvijemo efikasan algoritam za identifikovanje zavisnosti između podataka i otkrivanje novih znanja. Na osnovu predložene analize projektanti mogu direktno da izaberu implikacije koje će koristiti u svom radu.

Neka je $X \xrightarrow{\theta} Y$ fazi funkcionalna zavisnost. Pridruzićemo ovoj zavisnosti Mamdanijevu implikaciju $I_{MM} = \min(x, y)$. Da bismo proverili da li Mamdanijevu implikaciju možemo da primenimo u validaciji ove zavisnosti iskoristićemo naučna saznanja i definicije prezentovane u prethodnim poglavljima ovog rada, a to su:

- Saglasnost atributa X za n-torke t_i i t_j , $C(X[t_i, t_j])$:

Fazi model na bazi relacije bliskosti:

$$C(X[t_i, t_j]) = \min \left\{ \min_{x \in d_i} \left\{ \max_{y \in d_j} \{s(x, y)\} \right\}, \min_{x \in d_j} \left\{ \max_{y \in d_i} \{s(x, y)\} \right\} \right\},$$

Fazi model na bazi mogućnosti distribucije:

$$C(X[t_i, t_j]) = \min (\mu_{t_i}(A_i), \mu_{t_j}(A_i), s(t_i(A_i), t_j(A_i))),$$

- Saglasnost atributa Y za n-torke t_i i t_j , $C(Y[t_i, t_j])$:

Fazi model na bazi relacije bliskosti:

$$C(Y[t_i, t_j]) = \min \left\{ \min_{x \in d_i} \left\{ \max_{y \in d_j} \{s(x, y)\} \right\}, \min_{x \in d_j} \left\{ \max_{y \in d_i} \{s(x, y)\} \right\} \right\},$$

Fazi model na bazi mogućnosti distribucije:

$$C(Y[t_i, t_j]) = \min (\mu_{t_i}(A_i), \mu_{t_j}(A_i), s(t_i(A_i), t_j(A_i))),$$

- Definicija za fazi funkcionalnu zavisnosti $X \xrightarrow{\theta} Y$ (Sözat i Yazici):

$$C(Y[t_i, t_j]) \geq \min (\theta, C(X[t_i, t_j]))$$

a) Određivanje jačine (θ) fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ primenom Mamdani-ijeve implikacije I_{MM} iz fazi logike:

Pretpostavimo da važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, gde su X i Y atributi fazi relacije $r(R)$, a θ jačina zavisnosti, $\theta \in [0,1]$.

Zbog pretpostavke da vredi fazi funkcionalna zavisnost imamo:

$$C(Y[t_i, t_j]) \geq \min(\theta, C(X[t_i, t_j]))$$

U ovom slučaju parametar θ odredićemo primenom Mamdani-ijeve implikacije:

$$I_{MM} = \min(x, y), \quad x, y \in [0,1].$$

Da bi pridružena fazi implikacija bila zadovoljiva mora da važi:

$$\min(x, y) \geq \theta_k$$

gde je θ_k kritična jačina zavisnosti, $\theta_k \in [0,1]$, $\theta \geq \theta_k$ i $\theta_k \geq 0.5$.

Kako smo ovde konstatovali da ćemo koristiti saglasnost n-torki nad posmatatranim atributima, na osnovu prethodnog sledi:

$$x = C(X[t_i, t_j]) \geq \theta_k,$$

$$y = C(Y[t_i, t_j]) \geq \theta_k,$$

A zbog pretpostavke da vredi fazi funkcionalna zavisnost, imamo:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j]) \geq \theta_k$$

Na ovaj način smo verifikovali ekvivalentnost između Mamdani-ijeve implikacije i fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$. Prema tome, jačina fazi funkcionalne zavisnosti θ jednaka je:

$$\theta = \begin{cases} p \vee q, & \text{ako je } C(X[t_i, t_j]) = C(Y[t_i, t_j]) \geq 0.5 \\ p, & \text{ako je } C(Y[t_i, t_j]) > C(X[t_i, t_j]) \geq 0.5 \end{cases}$$

gde je sa p i q označeno: $p = x = C(X[t_i, t_j])$ i $q = y = C(Y[t_i, t_j])$.

b) Određivanje jačine (θ) fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ primenom Lukasiewicz-eve implikacije I_L iz fazi logike:

Pretpostavimo, kao i u prethodnom slučaju, da važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, gde su X i Y atributi fazi relacije $r(R)$, a θ jačina zavisnosti, $\theta \in [0,1]$.

Zbog pretpostavke da vredi fazi funkcionalna zavisnost imamo:

$$C(Y[t_i, t_j]) \geq \min(\theta, C(X[t_i, t_j]))$$

U ovom slučaju parametar θ pokušaćemo da odredimo primenom Lukasiewicz-eve implikacije:

$$I_L = \min(1, 1 - x + y), \quad x, y \in [0,1]$$

Kako vrediti fazi funkcionalna zavisnost, mora da bude zadovoljeno:

$$\min(1, 1-x+y) \geq \theta_k$$

θ_k je kritična jačina zavisnosti. Da bismo između dva atributa konstatovali postojanje zavisnosti, ova vrednost mora biti veća od 0.5. Kako vrediti da je:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j]) \geq 0.5$$

iz ovoga sledi da je:

$$1-x+y \geq 1,$$

tada mora biti:

$$\min(1, 1-x+y) \geq 1.$$

Prema tome, jačina fazi funkcionalne zavisnosti θ , u najboljem slučaju, iznosi 1. S obzirom da je za vrednost $\theta=1$ reč o klasičnoj funkcionalnoj zavisnosti, konstatujemo da je Lukasiewicz-evu fazi implikaciju nije moguće primeniti u analizi fazi funkcionalnih zavisnosti u fazi relacionim bazama podataka.

c) Određivanje jačine (θ) fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ primenom Kleene-Diense implikacije I_{KL} iz fazi logike:

Dokažimo sada ekvivalentnost fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ sa Kleene-Diense-ovom implikacijom u fazi logici. U ovom slučaju koristimo isti metod kao u analizi prethodnih tvrdnji.

Pretpostavimo da važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, gde su X i Y atributi fazi relacije $r(R)$, a θ jačina zavisnosti, $\theta \in [0,1]$.

Zbog ove pretpostavke važi da je:

$$C(Y[t_i, t_j]) \geq \min(\theta, C(X[t_i, t_j]))$$

Vrednost parametra θ u ovom primeru izračunaćemo koristeći Kleene-Diense implikaciju:

$$I_{KL} = \max(1-x, y), \quad x, y \in [0,1]$$

U klasičnoj Bulovoj logici implikacija $X \rightarrow Y$ piše se u obliku:

$$X \rightarrow Y = \neg X \vee Y$$

Kleene-Diense implikacija proširuje klasičnu implikaciju i izvodi se iz prethodne relacije. Kada se zameni operator \vee sa \max operatorom, a za negaciju se primeni definicija negacije u fazi logici, dobije se ova poznata fazi implikacija. Već ova činjenica nas navodi na ekvivalentnost fazi funkcionalnih zavisnosti i Kleene-Diense fazi implikacije, a što dokazujemo sledećom interpretacijom.

Kako vrediti fazi funkcionalna zavisnost, mora da bude zadovoljeno:

$$\max(1-x, y) \geq \theta_k$$

gde je θ_k kritična jačina zavisnosti, $\theta_k \in [0,1]$, $\theta \geq \theta_k$ i $\theta_k \geq 0.5$.

Da bi prethodna nejednakost važila, mora da bude zadovoljeno:

$$1-x \geq \theta_k, \text{ tj. } 1-C(X[t_i, t_j]) \geq \theta_k \text{ i}$$

$$y \geq \theta_k, \text{ tj. } C(Y[t_i, t_j]) \geq \theta_k.$$

Kako zbog naše pretpostavke da vredi fazi funkcionalna zavisnost imamo da je:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j]) \geq \theta_k$$

iz prethodnog zaključujemo da važi:

$$1 - C(X[t_i, t_j]) \leq C(Y[t_i, t_j]).$$

Označimo sa p i q: $p=x=C(X[t_i, t_j])$ i $q=y=C(Y[t_i, t_j])$.

Tada važi da je θ :

$$\theta = \begin{cases} p \vee q, & \text{ako je } C(X[t_i, t_j]) = C(Y[t_i, t_j]) \geq 0.5 \\ q, & \text{ako je } C(Y[t_i, t_j]) > C(X[t_i, t_j]) \geq 0.5 \end{cases}$$

Zaključujemo, dakle, da i Kleene-Diense implikacija može da se koristi u analizi fazi funkcionalnih zavisnosti, što ćemo i koristiti u fazi relacijama u ovom radu.

d) Određivanje jačine (θ) fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ primenom Early-Zade implikacije I_{EA} iz fazi logike:

U daljim istraživanjima proširićemo analizu i na Early-Zade fazi implikaciju, tj. ispitaćemo ponašanje ovog operatora u situacijama ispitivanja zavisnosti između podataka u fazi bazama podataka.

Kao i u prethodnim slučajevima, pretpostavimo da važi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, gde su X i Y atributi fazi relacije $r(R)$, a θ jačina zavisnosti, $\theta \in [0,1]$.

Otuda proizilazi da je:

$$C(Y[t_i, t_j]) \geq \min(\theta, C(X[t_i, t_j]))$$

U ovom slučaju iskoristićemo Early-Zade fazi implikaciju za izračunavanje jačine zavisnosti θ između atributa X i Y:

$$I_{EA} = \max(1-x, \min(x, y)), \quad x, y \in [0, 1]$$

Kako je u klasičnoj logici implikacija $X \rightarrow Y$ ekvivalentna i sa:

$$X \rightarrow Y = (X \wedge Y) \vee \neg X$$

primenjujući definicije iz fazi logike dobija se Early-Zade implikacija.

Dokažimo sada primenljivost I_{EA} u interpretaciji jačine zavisnosti θ . Zbog pretpostavke o postojanju fazi funkcionalne zavisnosti, mora da važi:

$$\max(1-x, \min(x, y)) \geq \theta_k$$

gde je θ_k kritična jačina zavisnosti, $\theta_k \in [0,1]$, $\theta \geq \theta_k$ i $\theta_k \geq 0.5$.

Da bi prethodna nejednakost važila, mora da bude zadovoljeno:

$$1-x \geq \theta_k, \text{ tj. } 1 - C(X[t_i, t_j]) \geq \theta_k \text{ i}$$

$$\min(x, y) \geq \theta_k, \text{ tj. } \min(C(X[t_i, t_j]), C(Y[t_i, t_j])) \geq \theta_k.$$

Kako zbog naše pretpostavke da vredi fazi funkcionalna zavisnost imamo da je:

$$C(Y[t_i, t_j]) \geq C(X[t_i, t_j]) \geq \theta_k$$

iz prethodnog zaključujemo da važi:

$$1 - C(X[t_i, t_j]) \leq C(Y[t_i, t_j])$$

Označimo sa p i q : $p=x=C(X[t_i,t_j])$ i $q=y=C(Y[t_i,t_j])$.

Tada važi da je θ :

$$\theta = \begin{cases} p \vee q, \text{ ako je } C(X[t_i, t_j]) = C(Y[t_i, t_j]) \geq 0.5 \\ p, \text{ ako je } C(Y[t_i, t_j]) > X(Y[t_i, t_j]) \geq 0.5 \end{cases}$$

Da ovaj odnos između fazi implikacija i fazi funkcionalnih zavisnosti nije slučajan, dokazuje i Early-Zade fazi implikacija. Na osnovu parametra θ zaključujemo da je kalkulacija jačine zavisnosti između posmatranih atributa u slučaju primene Kleene-Diense i Early-Zade fazi implikacije nije identična.

e) Određivanje jačine (θ) fazi funkcionalne zavisnosti $X \xrightarrow{\theta} Y$ primenom Willmot-ove implikacije I_w iz fazi logike:

Kao i u prethodnim slučajevima pokazaćemo ekvivalentnost između fazi funkcionalnih zavisnosti i fazi implikacije po Willmotu koristeći isti princip i metod zaključivanja.

Krenimo od pretpostavke da vredi fazi funkcionalna zavisnost $X \xrightarrow{\theta} Y$, gde su X i Y atributi fazi relacije $r(R)$, a θ jačina zavisnosti, $\theta \in [0,1]$.

Zbog toga važi da je:

$$C(Y[t_i,t_j]) \geq \min(\theta, C(X[t_i,t_j]))$$

Jačinu zavisnosti θ između atributa X i Y odredićemo koristeći fazi implikaciju po Willmotu:

$$I_w = \min[\max(1-x,y), \max(x,1-x), \max(y,1-y)], x,y \in [0,1]$$

Kako smo na početku pretpostavili da između atributa X i Y postoji fazi funkcionalna zavisnost, zaključujemo da za pridruženu fazi implikaciju I_w važi $I_w \geq \theta_k$, tj.:

$$\min[\max(1-x,y), \max(x,1-x), \max(y,1-y)] \geq \theta_k$$

gde je θ_k kritična jačina zavisnosti, $\theta_k \in [0,1]$, $\theta \geq \theta_k$ i $\theta_k \geq 0.5$.

Da bi ova nejednakost bila zadovoljena, mora biti:

$$\max(1-x,y) \geq \theta_k \wedge \max(x,1-x) \geq \theta_k \wedge \max(y,1-y) \geq \theta_k \text{ tj.}$$

$$\max(1-C(X[t_i,t_j]), C(Y[t_i,t_j])) \geq \theta_k \text{ i}$$

$$\max(C(X[t_i,t_j]), 1-C(X[t_i,t_j])) \geq \theta_k \text{ i}$$

$$\max(C(Y[t_i,t_j]), 1-C(Y[t_i,t_j])) \geq \theta_k$$

Kako zbog naše pretpostavke da vredi fazi funkcionalna zavisnost imamo da je:

$$C(Y[t_i,t_j]) \geq C(X[t_i,t_j]) \geq \theta_k$$

iz prethodnog zaključujemo da važi:

$$1-C(X[t_i,t_j]) \leq C(Y[t_i,t_j]) \text{ i}$$

$$1-C(X[t_i,t_j]) \leq C(X[t_i,t_j]) \text{ i}$$

$$1-C(Y[t_i,t_j]) \leq C(Y[t_i,t_j])$$

Označimo sa p i q : $p=x=C(X[t_i,t_j])$ i $q=y=C(Y[t_i,t_j])$

Tada važi da je θ :

$$\theta = \begin{cases} p \vee q, \text{ ako je } C(X[t_i, t_j]) = C(Y[t_i, t_j]) \geq 0.5 \\ p, \text{ ako je } C(Y[t_i, t_j]) > X(Y[t_i, t_j]) \geq 0.5 \end{cases}$$

Karakterizacija operatora implikacije na prezentovani način pruža nam mogućnost njihove verifikacije u kontekstu analize fazi funkcionalnih zavisnosti. Ovaj metod logičkog objašnjenja determiniše kriterijume koje moramo da uzmemo u obzir kada primenjujemo ovaj fragment fazi logike u fazi relacionim bazama podataka. Neke tipove fazi implikacija primenićemo u predloženom algoritmu da bismo došli do novih znanja i procene jačine zavisnosti θ . Implementacijom prethodnih saznanja dokazaćemo postojanje fazi funkcionalnih zavisnosti u fazi relacijama i na taj način automatizovati izvođenje skrivenih zavisnosti.

5.6. Izbor fazi implikacije u algoritmu za identifikovanje fazi funkcionalnih zavisnosti

Fazi logika se široko primenjuju u realnim aplikacijama, o čemu može da se nađe veliki broj radova u bogatoj literaturi iz ove oblasti. Najveća praktična primena ostvarena je u industrijskim rešenjima i fazi sistemima za upravljanje i kontrolu. U ovoj doktorskoj disertaciji predstavljena je jedna nova perspektiva u smislu primene koncepta fazi implikacija u relacionim bazama podataka. Ovaj novi pristup pojašnjava osnov algoritma koji koristimo u identifikovanju fazi funkcionalnih zavisnosti i otvara nove ideje u danas veoma aktuelnoj i izazovnoj oblasti istraživanja – istraživanju podataka i sistemima zasnovanim na znanju. Veliki broj informacionih sistema koristi relacione baze podataka. Ova činjenica navodi nas na zaključak da ovaj koncept ima perspektivu, a fazi logika se još jednom pokazala kao moćan alat u analizi podataka.

U prethodnim razmatranjima prikazali smo na koji način možemo da okarakterišemo pravilnu upotrebu fazi implikacija u fazi relacijama. Stoga se postavlja logično pitanje: Koju od ovih fazi implikacija ćemo koristiti u konkretnoj aplikaciji ili bazi podataka? Različiti projektanti informacionih sistema različito interpretiraju znanja i zavisnosti koji se identifikuju u posmatranom modelu. U analizi literature ustanovilo smo da je selekcija i izbor optimalnog operatora fazi implikacije jedan od fundamentalnih problema u dizajnu fazi sistema. Ova tematika je diskutovana od strane velikog broja autora (Bendler, Kendel, Mizimuto, Fukami i dr.).

U istraživanju sprovedenom u ovom doktorskom radu dokazano je da samo neke fazi implikacije mogu da se primenjuju, dok druge nisu pogodne u analizi fazi funkcionalnih zavisnosti pod određenim kriterijumima. Kod primene fazi implikacija u fazi modelima baza podataka ključna uloga je na projektantima informacionih sistema. Proces identifikovanja zavisnosti između podataka blizak je aproksimativnom razmišljanju. Dizajner baze podataka svoje preferencije najbolje opisuje izborom odgovarajuće fazi implikacije. Primena konkretne fazi implikacije u prvom redu zavisi od prirode problema koji želimo da prikazemo u formi baze podataka, a u drugom redu od procene, želje i zahteva projektanta baze podataka. Izlazni skup fazi funkcionalnih zavisnosti se razlikuje po tome koji fazi operatori se primenjuju u algoritmu. Prema tome, model fazi baze podataka nije isti u slučaju primene Mamdani-ijeve i Kleene-Diense fazi implikacije. Dakle, u slučaju primene fazi implikacija u različitim bazama podataka, od izbora i procena projektanta zavisi koji ćemo operator koristiti. Tu pre svega mislimo na ekspertska znanja koja oni poseduju, jer oni jasno mogu da preciziraju kakvo rešenje i u kakvom obliku žele da imaju. Time projektanti dobijaju slobodu u modeliranju relacija jer imaju pregled celokupnog procesa izvođenja fazi funkcionalnih zavisnosti i ukoliko im neki

rezultati izvođenja ne odgovaraju, mogu da ih modifikuju. Zbog toga je donosiocima odluka i analitičarima potrebna asistencija projektanata, pre svega radi pripreme ulaznih podataka u format pogodan za zaključivanje. Svaka otkrivena i identifikovana zavisnost odgovara IF-THEN pravilima. Na taj način rešenja dobijena primenom predloženog algoritma prikazujemo u upotrebljivoj formi za krajnjeg korisnika. Naravno, projektanti informacionih sistema mogu da koriste i veći broj fazi implikacija koje zadovoljavaju kriterijume analize fazi funkcionalnih zavisnosti. U tom slučaju oni će izabrati onaj izlazni skup zavisnosti i onaj model koji odgovara početnim ciljevima i zahtevima projektovanja baze podataka, jer fazi implikacije određuju stepen zadovoljenja zaključka.

Mamdanijeva aplikacija je najčešće korišćena implikacija u fazi sistemima. Tu činjenicu ćemo zadržati i kada je reč o primeni ovog koncepta fazi logike i u različitim modelima baza podataka. Razlog tome je što su fazi funkcionalne zavisnosti predstavljene IF-THEN pravilima za koje je poznato da su lokalnog tipa. Ako kažemo da "Visoka inteligencija značajno određuje uspeh u studiranju" sa jačinom zavisnosti 0.9, to znači da smo analizom baze podataka došli do takvog zaključka. Ali to ne podrazumeva i druge interpretacije skrivenih znanja koje postoje između atributa relacije Inteligencija i Uspeh.

Druge prednosti korišćenja Mamdanijeve fazi implikacije u analizi fazi funkcionalnih zavisnosti u relacionim bazama podataka:

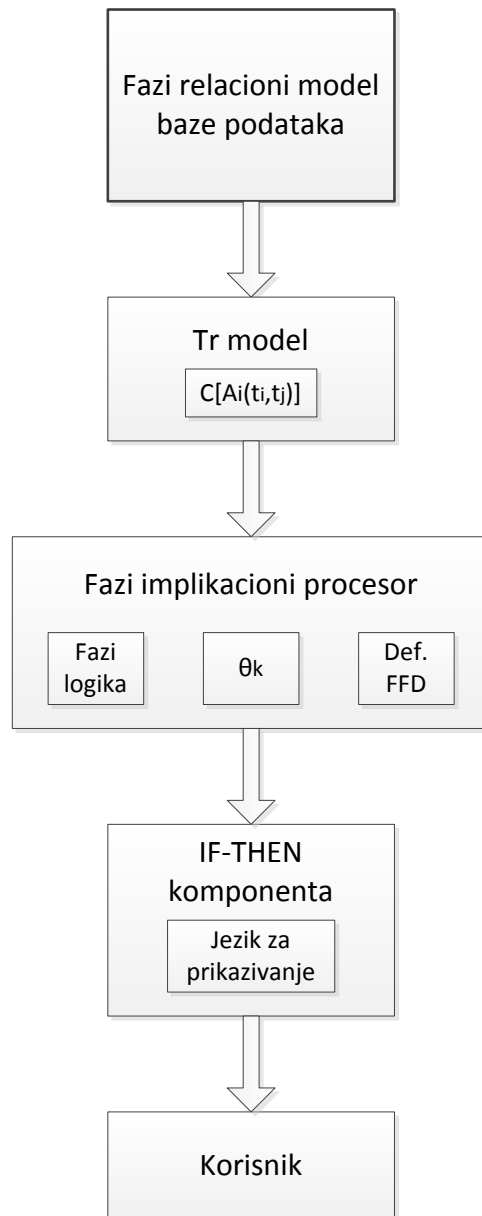
- intuitivan operator,
- ima široku primenu u aplikativnim rešenjima gde se izlaz predstavlja fazi pravilima,
- algoritamska efikasnost,
- računarska efikasnost,
- jednostavnija za korišćenje,
- izlazni rezultati su prikazani u upotrebljivom obliku.

Da bismo uporedili ponašanje različitih fazi implikacija, mi ćemo uraditi komparativnu analizu na izabranom fazi modelu baze podataka, a o čemu će biti reči u narednim poglavljima doktorske disertacije.

5.7. Model algoritma za pronalaženje fazi funkcionalnih zavisnosti

Model algoritma za pronalaženje fazi funkcionalnih zavisnosti predstavljen u ovoj doktorskoj disertaciji sadrži 5 modula/komponenti:

1. Fazi relaciju,
2. T_r relaciju,
3. Fazi implikacioni procesor,
4. IF-THEN dijalog komponentu i
5. Korisnika/Menadžera.



Slika 15. Model za pronalaženje fazi funkcionalnih zavisnosti u fazi relacijama

Na slici 15. prikazana je arhitektura modela i koraci u procesu pronalaženja fazi funkcionalnih zavisnosti. Fazi relacija sadrži podatke o zaposlenim radnicima u jednoj kompaniji. Na osnovu fazi baze podataka mi smo definisali T_r relaciju u kojoj smo izračunali usaglašenost n -torki na svim atributima prvobitne baze. Ovako strukturirani podaci se koriste u fazi implikacionom procesoru za utvrđivanje potencijalnih zavisnosti unutar skupa atributa. To je ujedno i bazična komponenta modela. IF-THEN dijalog komponenta je predviđena za translaciju informacija dobijenih u trećem koraku u lingvističke izraze pogodne za krajnje korisnike odnosno menadžere koji direktno ili indirektno učestvuju u donošenju odluka. Ovu komponentu možemo da smatramo semantičkim nivoom jer interpretira rezultate iz prethodnog koraka i predstavlja interfejs prema krajnjoj instanci. Poslednja komponenta je, dakle, korisnik koji želi da na jednostavan, relativno lagan i razumljiv način dođe do željenih informacija.

5.8. Zaključak

U ovom poglavlju predstavili smo rezultate samostalnog istraživanja u pronalaženju fazi funkcionalnih zavisnosti. Predstavljen je model i algoritam koji ćemo koristiti u sledećem poglavlju za eksperiment nad realnim podacima. Iako postoje različiti pristupi u posmatranju fazi funkcionalnih zavisnosti, oni nisu zadovoljavali suštinske potrebe u analizi podataka. Zbog toga je razvijena metodologija koja korisnicima omogućava dobijanje novih i ineresatnih saznanja iz baza podataka, a koji su iskoristivi u različitim poljima. Kao prvo, opisana je uloga fazi logike u identifikovanju fazi funkcionalnih zavisnosti, zatim karakterizacija različitih fazi implikacija, izbor određenih fazi implikacija i predstavljene su fazi relacije koje su korišćene za verifikaciju rezultata istraživanja.

6. Algoritam za identifikovanje fazi funkcionalnih zavisnosti

6.1. Eksperiment nad realnim fazi relacijama

Prethodna razmatranja interpretiraćemo na fazi relacijama prikazanim u Tabelama 35. i 42., uzimajući u obzir postojeće relacije bliskosti između vrednosti atributa (Tabela 36-41. i Tabela 43.-46.). Iskoristićemo ovaj primer da bismo dokazali polaznu (početnu) hipotezu istraživanja u ovoj doktorskoj disertaciji. U tom smislu, mi verujemo da ova metodologija donosi nove mogućnosti kada je u pitanju izražavanje relacija između podataka i atributa posmatrane baze. Dodatno, obezbedili smo metodologiju kojom smo definisali algoritam za otkrivanje fazi funkcionalnih zavisnosti u fazi bazama podataka.

U sledećim primerima prikazaćemo ekstrakciju fazi funkcionalnih zavisnosti iz relanih podataka. Kao što je diskutovano, određeni fragment fazi logike može da se primeni u otkrivanju novih znanja i interpretaciji istraživanja podataka.

Fazi relacije u Tabelama 35. i 42. sadrže podatke o radnicima zaposlenim u nekom preduzeću. Prikupljeni su podaci u formi fazi relacije za pet radnika (Ime, Godine, Iskustvo, Produktivnost, Jezik, Pozicija, Zarada). Vrednosti atributa sadrže crisp i fazi informacije. Navođenje kompletne fazi relacije nema preveliki značaj za predstavljanje metodologije istraživanja u ovom radu. Razmatraćemo dve situacije, dva aspekta iste relacione šeme: fazi model baze podataka na bazi relacije bliskosti i fazi model baze podataka na bazi mogućnosti distribucije.

Informacioni sistemi sadrže veliki broj podataka koji su najčešće skladišteni u formi relacione baze podataka. U takvoj situaciji, istakli smo značaj utvrđivanja zavisnosti između atributa relacije i otkrivanje novih znanja naročito iz pozicije menadžera i donosioca odluka. Podaci, naročito u fazi relacijama, sami po sebi ne predstavljaju značajan izvor upravljačkih informacija. Zbog toga je važna analiza relacija između podataka i atributa i pretraga korisnih informacija i novih znanja sadržanih u njima. Zbog toga u ovom poglavlju analiziramo ovu problematiku.

6.2. Algoritam za pretragu fazi funkcionalnih zavisnosti u fazi modelu na bazi relacije bliskosti

U ovom poglavlju pokazaćemo kako naš algoritam radi na praktičnom primeru. Tabela 35. prikazuje fazi model na bazi relacije bliskosti. Za svaki atribut koji je definisana je relacija bliskosti. Pretraga fazi funkcionalnih zavisnosti u fazi modelu na bazi relacije bliskosti vrši se primenom definicije saglasnosti na paru n-torki. Na relaciji T_r je moguće primenjivati fazi implikacije i vršiti ekstrakciju fazi funkcionalnih zavisnosti. Prema našoj definiciji, vezivni operator koji povezuje uočene zavisnosti na parovima n-torki je uvek minimum t-norma. Skup fazi funkcionalnih zavisnosti može da se proširi primenom pravila izvođenja za fazi funkcionalne zavisnosti.

Stepen saglasnosti n-torki na atributu **Ime_Prezime**:

$$\begin{aligned} C(\text{Ime_Prezime}[t_1, t_2]) &= \min\{\min\{\max\{s(\text{IP1}, \text{IP2})\}\}, \min\{\max\{s(\text{IP2}, \text{IP1})\}\}\}, \\ &= \min\{\min\{\max\{0.7\}\}, \min\{\max\{0.7\}\}\}, \\ &= \min\{\min\{0.7\}, \min\{0.7\}\} \\ &= \min\{0.7, 0.7\} = 0.7 \end{aligned}$$

Stepen saglasnosti n-torki na atributu **Godine**:

$$\begin{aligned}
 C(\text{Godine}[t_1, t_2]) &= \min\{\min\{\max\{s(\text{veoma mlad, star})\}, \\
 &\quad \min\{\max\{s(\text{star, veoma mlad})\}\}\}, \\
 &= \min\{\min\{\max\{0.30\}\}, \\
 &\quad \min\{\max\{0.30\}\}\} \\
 &= \min\{\min\{0.30\}, \min\{0.30\}\} \\
 &= \min\{0.30, 0.30\} = 0.30
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_1, t_3]) &= \min\{\min\{\max\{s(\text{veoma mlad, mlad}), s(\text{veoma mlad, srednjih godina})\}\}, \\
 &\quad \min\{\max\{s(\text{mlad, veoma mlad}), \max\{s(\text{srednjih g., veoma mlad})\}\}\}\}, \\
 &= \min\{\min\{\max\{0.95, 0.70\}\}, \\
 &\quad \min\{\max\{0.95\}, \max\{0.70\}\}\} \\
 &= \min\{\min\{0.95\}, \min\{0.95, 0.70\}\} \\
 &= \min\{0.95, 0.70\} = 0.70
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_1, t_4]) &= \min\{\min\{\max\{s(\text{veoma mlad, veoma mlad}), s(\text{veoma mlad, mlad})\}\}, \\
 &\quad \min\{\max\{s(\text{veoma mlad, veoma mlad}), \max\{s(\text{mlad, veoma mlad})\}\}\}\}, \\
 &= \min\{\min\{\max\{1, 0.95\}\}, \\
 &\quad \min\{\max\{1\}, \max\{0.95\}\}\} \\
 &= \min\{\min\{1\}, \min\{1, 0.95\}\} \\
 &= \min\{1, 0.95\} = 0.95
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_1, t_5]) &= \min\{\min\{\max\{s(\text{veoma mlad, mlad})\}\}, \\
 &\quad \min\{\max\{s(\text{mlad, veoma mlad})\}\}\}, \\
 &= \min\{\min\{\max\{0.95\}\}, \\
 &\quad \min\{\max\{0.95\}\}\} \\
 &= \min\{\min\{0.95\}, \min\{0.95\}\} \\
 &= \min\{0.95, 0.95\} = 0.95
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_2, t_3]) &= \min\{\min\{\max\{s(\text{star, mlad}), s(\text{star, srednjih godina})\}\}, \\
 &\quad \min\{\max\{s(\text{mlad, star}), \max\{s(\text{srednjih godina, star})\}\}\}\}, \\
 &= \min\{\min\{\max\{0.40, 0.50\}\}, \\
 &\quad \min\{\max\{0.40\}, \max\{0.50\}\}\} \\
 &= \min\{\min\{0.50\}, \min\{0.40, 0.50\}\} \\
 &= \min\{0.50, 0.40\} = 0.40
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_2, t_4]) &= \min\{\min\{\max\{s(\text{star, veoma mlad}), s(\text{star, mlad})\}\}, \\
 &\quad \min\{\max\{s(\text{veoma mlad, star}), \max\{s(\text{mlad, star})\}\}\}\}, \\
 &= \min\{\min\{\max\{0.30, 0.40\}\}, \\
 &\quad \min\{\max\{0.30\}, \max\{0.40\}\}\} \\
 &= \min\{\min\{0.40\}, \min\{0.30, 0.40\}\} \\
 &= \min\{0.40, 0.30\} = 0.30
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Godine}[t_2, t_5]) &= \min\{\min\{\max\{s(\text{star, mlad})\}\}, \\
 &\quad \min\{\max\{s(\text{mlad, star})\}\}\}, \\
 &= \min\{\min\{\max\{0.40\}\}, \\
 &\quad \min\{\max\{0.40\}\}\} \\
 &= \min\{\min\{0.40\}, \min\{0.40\}\} \\
 &= \min\{0.40, 0.40\} = 0.40
 \end{aligned}$$

$$\begin{aligned}
C(\text{Godine}[t_3,t_4]) &= \min\{\min\{\max\{s(\text{mlad,veoma mlad}), s(\text{mlad,mlad})\}, \max\{s(\text{srednjih godina,veoma mlad}), s(\text{srednjih godina,mlad})\}\}, \min\{\max\{s(\text{veoma mlad,mlad}), s(\text{veoma mlad,srednjih godina})\}, \max\{s(\text{mlad,mlad}), s(\text{mlad,srednjih godina})\}\}\}, \\
&= \min\{\min\{\max\{0.95,1\}, \max\{0.70,0.90\}\}, \\
&\quad \min\{\max\{0.95,0.70\}, \max\{1,0.90\}\}\}, \\
&= \min\{\min\{1,0.90\}, \min\{0.95,1\}\} \\
&= \min\{0.90, 0.95\} = 0.90
\end{aligned}$$

$$\begin{aligned}
C(\text{Godine}[t_3,t_5]) &= \min\{\min\{\max\{s(\text{mlad,mlad})\}, \max\{s(\text{srednjih godina,mlad})\}\}, \\
&\quad \min\{\max\{s(\text{mlad,mlad}),s(\text{mlad,srednjih godina})\}\}\}, \\
&= \min\{\min\{\max\{1\}, \max\{0.90\}\}, \\
&\quad \min\{\max\{1,0.90\}\}\}, \\
&= \min\{\min\{1,0.90\}, \min\{1\}\} \\
&= \min\{0.90, 1\} = 0.90
\end{aligned}$$

$$\begin{aligned}
C(\text{Godine}[t_4,t_5]) &= \min\{\min\{\max\{s(\text{veoma mlad,mlad})\}, \max\{s(\text{mlad,mlad})\}\}, \\
&\quad \min\{\max\{s(\text{mlad,veoma mlad}),s(\text{mlad,mlad})\}\}\}, \\
&= \min\{\min\{\max\{0.95\}, \max\{1\}\}, \\
&\quad \min\{\max\{0.95,1\}\}\}, \\
&= \min\{\min\{0.95,1\}, \min\{1\}\} \\
&= \min\{0.95, 1\} = 0.95
\end{aligned}$$

Stepen saglasnosti n-torki na atributu **Iskustvo**:

$$\begin{aligned}
C(\text{Iskustvo}[t_1,t_2]) &= \min\{\min\{\max\{s(\text{početnik,specijalista}),s(\text{početnik,ekspert})\}\}, \\
&\quad \min\{\max\{s(\text{specijalista,početnik})\}, \max\{s(\text{ekspert,početnik})\}\}\}, \\
&= \min\{\min\{\max\{0.70,0.30\}\}, \\
&\quad \min\{\max\{0.70\}, \max\{0.30\}\}\}, \\
&= \min\{\min\{0.70\}, \min\{0.70,0.30\}\} \\
&= \min\{0.70,0.30\} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_1,t_3]) &= \min\{\min\{\max\{s(\text{početnik,senior}),s(\text{početnik,specijalista})\}\}, \\
&\quad \min\{\max\{s(\text{senior,početnik})\}, \max\{s(\text{specijalista,početnik})\}\}\}, \\
&= \min\{\min\{\max\{0.70,0.70\}\}, \\
&\quad \min\{\max\{0.70\}, \max\{0.70\}\}\}, \\
&= \min\{\min\{0.70\}, \min\{0.70,0.70\}\} \\
&= \min\{0.70,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_1,t_4]) &= \min\{\min\{\max\{s(\text{početnik,početnik}),s(\text{početnik,senior})\}\}, \\
&\quad \min\{\max\{s(\text{početnik,početnik})\}, \max\{s(\text{senior,početnik})\}\}\}, \\
&= \min\{\min\{\max\{1,0.70\}\}, \\
&\quad \min\{\max\{1\}, \max\{0.70\}\}\}, \\
&= \min\{\min\{1\}, \min\{1,0.70\}\} \\
&= \min\{1,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_1,t_5]) &= \min\{\min\{\max\{s(\text{početnik,senior})\}\}, \\
&\quad \min\{\max\{s(\text{senior,početnik})\}\}\}, \\
&= \min\{\min\{\max\{0.70\}\}, \\
&\quad \min\{\max\{0.70\}\}\}, \\
&= \min\{\min\{0.70\}, \min\{0.70\}\} \\
&= \min\{0.70,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_2, t_3]) &= \min\{\min\{\max\{s(\text{specijalista, senior}), s(\text{specijalista, specijalista})\}, \\
&\max\{s(\text{ekspert, senior}), s(\text{ekspert, specijalista})\}\}, \min\{\max\{s(\text{senior, specijalista}), \\
&s(\text{senior, ekspert})\}, \max\{s(\text{specijalista, specijalista}), s(\text{specijalista, ekspert})\}\}, \\
&= \min\{\min\{\max\{0.85, 1\}, \max\{0.30, 0.50\}\}, \\
&\quad \min\{\max\{0.85, 0.30\}, \max\{1, 0.50\}\}\}, \\
&= \min\{\min\{1, 0.50\}, \min\{0.85, 1\}\} \\
&= \min\{0.50, 0.85\} = 0.50
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_2, t_4]) &= \min\{\min\{\max\{s(\text{specijalista, početnik}), s(\text{specijalista, senior})\}, \\
&\max\{s(\text{ekspert, početnik}), s(\text{ekspert, senior})\}\}, \min\{\max\{s(\text{početnik, specijalista}), \\
&s(\text{početnik, ekspert})\}, \max\{s(\text{senior, specijalista}), s(\text{senior, ekspert})\}\}, \\
&= \min\{\min\{\max\{0.70, 0.85\}, \max\{0.30, 0.30\}\}, \\
&\quad \min\{\max\{0.70, 0.30\}, \max\{0.85, 0.30\}\}\}, \\
&= \min\{\min\{0.85, 0.30\}, \min\{0.70, 0.85\}\} \\
&= \min\{0.30, 0.70\} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_2, t_5]) &= \min\{\min\{\max\{s(\text{specijalista, senior})\}, \max\{s(\text{ekspert, senior})\}\}, \\
&\quad \min\{\max\{s(\text{senior, specijalista}), s(\text{senior, ekspert})\}\}, \\
&= \min\{\min\{\max\{0.85\}, \max\{0.30\}\}, \\
&\quad \min\{\max\{0.85, 0.30\}\}\}, \\
&= \min\{\min\{0.85, 0.30\}, \min\{0.85\}\} \\
&= \min\{0.30, 0.85\} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_3, t_4]) &= \min\{\min\{\max\{s(\text{senior, početnik}), s(\text{senior, senior})\}, \\
&\max\{s(\text{specijalista, početnik}), s(\text{specijalista, senior})\}\}, \min\{\max\{s(\text{početnik, senior}), \\
&s(\text{početnik, specijalista})\}, \max\{s(\text{senior, senior}), s(\text{senior, specijalista})\}\}, \\
&= \min\{\min\{\max\{0.70, 1\}, \max\{0.70, 0.85\}\}, \\
&\quad \min\{\max\{0.70, 0.70\}, \max\{1, 0.85\}\}\}, \\
&= \min\{\min\{1, 0.85\}, \min\{0.70, 1\}\} \\
&= \min\{0.85, 0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_3, t_5]) &= \min\{\min\{\max\{s(\text{senior, senior})\}, \max\{s(\text{specijalista, senior})\}\}, \\
&\quad \min\{\max\{s(\text{senior, senior}), s(\text{senior, specijalista})\}\}, \\
&= \min\{\min\{\max\{1\}, \max\{0.85\}\}, \\
&\quad \min\{\max\{1, 0.85\}\}\}, \\
&= \min\{\min\{1, 0.85\}, \min\{1\}\} \\
&= \min\{0.85, 1\} = 0.85
\end{aligned}$$

$$\begin{aligned}
C(\text{Iskustvo}[t_4, t_5]) &= \min\{\min\{\max\{s(\text{početnik, senior})\}, \max\{s(\text{senior, senior})\}\}, \\
&\quad \min\{\max\{s(\text{senior, početnik}), s(\text{senior, senior})\}\}, \\
&= \min\{\min\{\max\{0.70\}, \max\{1\}\}, \\
&\quad \min\{\max\{0.70, 1\}\}\}, \\
&= \min\{\min\{0.70, 1\}, \min\{1\}\} \\
&= \min\{0.70, 1\} = 0.70
\end{aligned}$$

Stepen saglasnosti n-torki na atributu **Produktivnost**:

$$\begin{aligned}
 C(\text{Produktivnost}[t_1, t_2]) &= \min \{ \min \{ \max \{ s(\text{zadovolj.}, \text{dobra}), s(\text{zadovolj.}, \text{odlična}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{dobra}, \text{zadovolj.}), \max \{ s(\text{odlična}, \text{zadovolj.}) \} \} \} \\
 &= \min \{ \min \{ \max \{ 0.70, 0.50 \} \}, \\
 &\quad \min \{ \max \{ 0.70 \}, \max \{ 0.50 \} \} \} \\
 &= \min \{ \min \{ 0.50 \}, \min \{ 0.70, 0.50 \} \} \\
 &= \min \{ 0.50, 0.50 \} = 0.50
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_1, t_3]) &= \min \{ \min \{ \max \{ s(\text{zadovoljavajuća}, \text{dobra}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{dobra}, \text{zadovoljavajuća}) \} \} \} \\
 &= \min \{ \min \{ \max \{ 0.70 \} \}, \\
 &\quad \min \{ \max \{ 0.70 \} \} \} \\
 &= \min \{ \min \{ 0.70 \}, \min \{ 0.70 \} \} \\
 &= \min \{ 0.70, 0.70 \} = 0.70
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_1, t_4]) &= \min \{ \min \{ \max \{ s(\text{zadovoljavajuća}, \text{loša}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{loša}, \text{zadovoljavajuća}) \} \} \} \\
 &= \min \{ \min \{ \max \{ 0.80 \} \}, \\
 &\quad \min \{ \max \{ 0.80 \} \} \} \\
 &= \min \{ \min \{ 0.80 \}, \min \{ 0.80 \} \} \\
 &= \min \{ 0.80, 0.80 \} = 0.80
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_1, t_5]) &= \min \{ \min \{ \max \{ s(\text{zadovolj.}, \text{zadovolj.}), s(\text{zadovolj.}, \text{dobra}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{zadovolj.}, \text{zadovolj.}), \max \{ s(\text{dobra}, \text{zadovolj.}) \} \} \} \\
 &= \min \{ \min \{ \max \{ 1, 0.70 \} \}, \\
 &\quad \min \{ \max \{ 1 \}, \max \{ 0.70 \} \} \} \\
 &= \min \{ \min \{ 1 \}, \min \{ 1, 0.70 \} \} \\
 &= \min \{ 1, 0.70 \} = 0.70
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_2, t_3]) &= \min \{ \min \{ \max \{ s(\text{dobra}, \text{dobra}) \}, \max \{ s(\text{odlična}, \text{dobra}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{dobra}, \text{dobra}), s(\text{dobra}, \text{odlična}) \} \} \}, \\
 &= \min \{ \min \{ \max \{ 1 \}, \max \{ 0.80 \} \}, \\
 &\quad \min \{ \max \{ 1, 0.80 \} \} \}, \\
 &= \min \{ \min \{ 1, 0.80 \}, \min \{ 1 \} \} \\
 &= \min \{ 0.80, 1 \} = 0.80
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_2, t_4]) &= \min \{ \min \{ \max \{ s(\text{dobra}, \text{loša}) \}, \max \{ s(\text{odlična}, \text{loša}) \} \}, \\
 &\quad \min \{ \max \{ s(\text{loša}, \text{dobra}), s(\text{loša}, \text{odlična}) \} \} \}, \\
 &= \min \{ \min \{ \max \{ 0.50 \}, \max \{ 0.10 \} \}, \\
 &\quad \min \{ \max \{ 0.50, 0.10 \} \} \}, \\
 &= \min \{ \min \{ 0.50, 0.10 \}, \min \{ 0.50 \} \} \\
 &= \min \{ 0.10, 0.50 \} = 0.10
 \end{aligned}$$

$$\begin{aligned}
 C(\text{Produktivnost}[t_2, t_5]) &= \min \{ \min \{ \max \{ s(\text{dobra}, \text{zadovoljavajuća}), s(\text{dobra}, \text{dobra}) \}, \\
 &\quad \max \{ s(\text{odlična}, \text{zadovoljavajuća}), s(\text{odlična}, \text{dobra}) \} \}, \min \{ \max \{ s(\text{zadovoljavajuća}, \text{dobra}), \\
 &\quad s(\text{zadovoljavajuća}, \text{odlična}) \}, \max \{ s(\text{dobra}, \text{dobra}), s(\text{dobra}, \text{odlična}) \} \} \}, \\
 &= \min \{ \min \{ \max \{ 0.70, 1 \}, \max \{ 0.50, 0.80 \} \}, \\
 &\quad \min \{ \max \{ 0.70, 0.50 \}, \max \{ 1, 0.80 \} \} \}, \\
 &= \min \{ \min \{ 1, 0.80 \}, \min \{ 0.70, 1 \} \} \\
 &= \min \{ 0.80, 0.70 \} = 0.70
 \end{aligned}$$

$$\begin{aligned}
C(\text{Produktivnost}[t_3,t_4]) &= \min \{ \min \{ \max \{ s(\text{dobra, loša}) \}, \\
&\quad \min \{ \max \{ s(\text{loša, dobra}) \} \} \}, \\
&= \min \{ \min \{ \max \{ 0.50 \} \}, \\
&\quad \min \{ \max \{ 0.50 \} \} \}, \\
&= \min \{ \min \{ 0.50 \}, \min \{ 0.50 \} \} \\
&= \min \{ 0.50, 0.50 \} = 0.50
\end{aligned}$$

$$\begin{aligned}
C(\text{Produktivnost}[t_3,t_5]) &= \min \{ \min \{ \max \{ s(\text{dobra, zadovoljavajuća}), s(\text{dobra, dobra}) \}, \\
&\quad \min \{ \max \{ s(\text{zadovoljavajuća, dobra}) \}, \max \{ s(\text{dobra, dobra}) \} \} \}, \\
&= \min \{ \min \{ \max \{ 0.70, 1 \} \}, \\
&\quad \min \{ \max \{ 0.70 \}, \max \{ 1 \} \} \}, \\
&= \min \{ \min \{ 1 \}, \min \{ 0.70, 1 \} \} \\
&= \min \{ 1, 0.70 \} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Produktivnost}[t_4,t_5]) &= \min \{ \min \{ \max \{ s(\text{loša, zadovoljavajuća}), s(\text{loša, dobra}) \}, \\
&\quad \min \{ \max \{ s(\text{zadovoljavajuća, loša}) \}, \max \{ s(\text{dobra, loša}) \} \} \}, \\
&= \min \{ \min \{ \max \{ 0.80, 0.50 \} \}, \\
&\quad \min \{ \max \{ 0.80 \}, \max \{ 0.50 \} \} \}, \\
&= \min \{ \min \{ 0.80 \}, \min \{ 0.80, 0.50 \} \} \\
&= \min \{ 0.80, 0.50 \} = 0.50
\end{aligned}$$

Stepen saglasnosti n-torki na atributu **Jezik**:

$$\begin{aligned}
C(\text{Jezik}[t_1,t_2]) &= \min \{ \min \{ \max \{ s(\text{Engleski, Engleski}) \}, \max \{ s(\text{Ruski, Engleski}) \} \}, \\
&\quad \min \{ \max \{ s(\text{Engleski, Engleski}), s(\text{Engleski, Ruski}) \} \}, \\
&= \min \{ \min \{ \max \{ 1 \}, \max \{ 0.20 \} \}, \\
&\quad \min \{ \max \{ 1, 0.20 \} \} \}, \\
&= \min \{ \min \{ 1, 0.20 \}, \min \{ 1 \} \} \\
&= \min \{ 0.20, 1 \} = 0.20
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_1,t_3]) &= \min \{ \min \{ \max \{ s(\text{Engleski, Francuski}) \}, \max \{ s(\text{Ruski, Francuski}) \} \}, \\
&\quad \min \{ \max \{ s(\text{Francuski, Engleski}), s(\text{Francuski, Ruski}) \} \}, \\
&= \min \{ \min \{ \max \{ 0.60 \}, \max \{ 0.30 \} \}, \\
&\quad \min \{ \max \{ 0.60, 0.30 \} \} \}, \\
&= \min \{ \min \{ 0.60, 0.30 \}, \min \{ 0.60 \} \} \\
&= \min \{ 0.30, 0.60 \} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_1,t_4]) &= \min \{ \min \{ \max \{ s(\text{Engleski, Ruski}) \}, \max \{ s(\text{Ruski, Ruski}) \} \}, \\
&\quad \min \{ \max \{ s(\text{Ruski, Engleski}), s(\text{Ruski, Ruski}) \} \}, \\
&= \min \{ \min \{ \max \{ 0.20 \}, \max \{ 1 \} \}, \\
&\quad \min \{ \max \{ 0.20, 1 \} \} \}, \\
&= \min \{ \min \{ 0.20, 1 \}, \min \{ 1 \} \} \\
&= \min \{ 0.20, 1 \} = 0.20
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_1,t_5]) &= \min \{ \min \{ \max \{ s(\text{Engleski, Engleski}), s(\text{Engleski, Francuski}) \}, \\
&\quad \max \{ s(\text{Ruski, Engleski}), s(\text{Ruski, Francuski}) \} \}, \min \{ \max \{ s(\text{Engleski, Engleski}), \\
&\quad s(\text{Engleski, Ruski}) \}, \max \{ s(\text{Francuski, Engleski}), s(\text{Francuski, Ruski}) \} \}, \\
&= \min \{ \min \{ \max \{ 1, 0.60 \}, \max \{ 0.20, 0.30 \} \}, \\
&\quad \min \{ \max \{ 1, 0.20 \}, \max \{ 0.60, 0.30 \} \} \}, \\
&= \min \{ \min \{ 1, 0.30 \}, \min \{ 1, 0.60 \} \} \\
&= \min \{ 0.30, 0.60 \} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_2,t_3]) &= \min\{\min\{\max\{s(\text{Engleski,Francuski})\}, \\
&\quad \min\{\max\{s(\text{Francuski,Engleski})\}\}\} \\
&= \min\{\min\{\max\{0.60\}\}, \\
&\quad \min\{\max\{0.60\}\}\} \\
&= \min\{\min\{0.60\}, \min\{0.60\}\} \\
&= \min\{0.60,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_2,t_4]) &= \min\{\min\{\max\{s(\text{Engleski,Ruski})\}, \\
&\quad \min\{\max\{s(\text{Ruski,Engleski})\}\}\} \\
&= \min\{\min\{\max\{0.20\}\}, \\
&\quad \min\{\max\{0.20\}\}\} \\
&= \min\{\min\{0.20\}, \min\{0.20\}\} \\
&= \min\{0.20,0.20\} = 0.20
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_2,t_5]) &= \min\{\min\{\max\{s(\text{Engleski,Engleski}),s(\text{Engleski,Francuski})\}, \\
&\quad \min\{\max\{s(\text{Engleski,Engleski})\},\max\{s(\text{Francuski,Engleski})\}\}\} \\
&= \min\{\min\{\max\{1,0.60\}\}, \\
&\quad \min\{\max\{1\},\max\{0.60\}\}\} \\
&= \min\{\min\{1\}, \min\{1,0.60\}\} \\
&= \min\{1,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_3,t_4]) &= \min\{\min\{\max\{s(\text{Francuski,Ruski})\}, \\
&\quad \min\{\max\{s(\text{Ruski,Francuski})\}\}\} \\
&= \min\{\min\{\max\{0.30\}\}, \\
&\quad \min\{\max\{0.30\}\}\} \\
&= \min\{\min\{0.30\}, \min\{0.30\}\} \\
&= \min\{0.30,0.30\} = 0.30
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_3,t_5]) &= \min\{\min\{\max\{s(\text{Francuski,Engleski}),s(\text{Francuski,Francuski})\}, \\
&\quad \min\{\max\{s(\text{Engleski,Francuski})\},\max\{s(\text{Francuski,Francuski})\}\}\} \\
&= \min\{\min\{\max\{0.60,1\}\}, \\
&\quad \min\{\max\{0.60\},\max\{1\}\}\} \\
&= \min\{\min\{1\}, \min\{0.60,1\}\} \\
&= \min\{1,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Jezik}[t_3,t_5]) &= \min\{\min\{\max\{s(\text{Ruski,Engleski}),s(\text{Ruski,Francuski})\}, \\
&\quad \min\{\max\{s(\text{Engleski,Ruski})\},\max\{s(\text{Francuski,Ruski})\}\}\} \\
&= \min\{\min\{\max\{0.20,0.30\}\}, \\
&\quad \min\{\max\{0.20\},\max\{0.30\}\}\} \\
&= \min\{\min\{0.30\}, \min\{0.20,0.30\}\} \\
&= \min\{0.30,0.20\} = 0.20
\end{aligned}$$

Stepen saglasnosti n-torki na atributu **Pozicija**:

$$\begin{aligned}
C(\text{Pozicija}[t_1,t_2]) &= \min\{\min\{\max\{s(\text{sistem inženjer,menadžer})\}, \\
&\quad \min\{\max\{s(\text{menadžer,sistem inženjer})\}\}\} \\
&= \min\{\min\{\max\{0.70\}\}, \\
&\quad \min\{\max\{0.70\}\}\} \\
&= \min\{\min\{0.70\}, \min\{0.70\}\} \\
&= \min\{0.70,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_1,t_3]) &= \min\{\min\{\max\{s(\text{sistem inženjer,direktor})\}, \\
&\quad \min\{\max\{s(\text{direktor,sistem inženjer})\}\}\}, \\
&= \min\{\min\{\max\{0.60\}\}, \\
&\quad \min\{\max\{0.60\}\}\}, \\
&= \min\{\min\{0.60\}, \min\{0.60\}\} \\
&= \min\{0.60,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_1,t_4]) &= \min\{\min\{\max\{s(\text{sistem inženjer,prodaja})\}, \\
&\quad \min\{\max\{s(\text{prodaja,sistem inženjer})\}\}\}, \\
&= \min\{\min\{\max\{0.50\}\}, \\
&\quad \min\{\max\{0.50\}\}\}, \\
&= \min\{\min\{0.50\}, \min\{0.50\}\} \\
&= \min\{0.50,0.50\} = 0.50
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_1,t_5]) &= \min\{\min\{\max\{s(\text{sistem inženjer,nabavka})\}, \\
&\quad \min\{\max\{s(\text{nabavka,sistem inženjer})\}\}\}, \\
&= \min\{\min\{\max\{0.50\}\}, \\
&\quad \min\{\max\{0.50\}\}\}, \\
&= \min\{\min\{0.50\}, \min\{0.50\}\} \\
&= \min\{0.50,0.50\} = 0.50
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_2,t_3]) &= \min\{\min\{\max\{s(\text{menadžer,direktor})\}, \\
&\quad \min\{\max\{s(\text{direktor,menadžer})\}\}\}, \\
&= \min\{\min\{\max\{0.90\}\}, \\
&\quad \min\{\max\{0.90\}\}\}, \\
&= \min\{\min\{0.90\}, \min\{0.90\}\} \\
&= \min\{0.90,0.90\} = 0.90
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_2,t_4]) &= \min\{\min\{\max\{s(\text{menadžer,prodaja})\}, \\
&\quad \min\{\max\{s(\text{prodaja,menadžer})\}\}\}, \\
&= \min\{\min\{\max\{0.60\}\}, \\
&\quad \min\{\max\{0.60\}\}\}, \\
&= \min\{\min\{0.60\}, \min\{0.60\}\} \\
&= \min\{0.60,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_2,t_5]) &= \min\{\min\{\max\{s(\text{menadžer,nabavka})\}, \\
&\quad \min\{\max\{s(\text{nabavka,menadžer})\}\}\}, \\
&= \min\{\min\{\max\{0.60\}\}, \\
&\quad \min\{\max\{0.60\}\}\}, \\
&= \min\{\min\{0.60\}, \min\{0.60\}\} \\
&= \min\{0.60,0.60\} = 0.60
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_3,t_4]) &= \min\{\min\{\max\{s(\text{direktor,prodaja})\}, \\
&\quad \min\{\max\{s(\text{prodaja,direktor})\}\}\}, \\
&= \min\{\min\{\max\{0.70\}\}, \\
&\quad \min\{\max\{0.70\}\}\}, \\
&= \min\{\min\{0.70\}, \min\{0.70\}\} \\
&= \min\{0.70,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_3,t_5]) &= \min\{\min\{\max\{s(\text{direktor,nabavka})\}, \\
&\quad \min\{\max\{s(\text{nabavka,direktor})\}\}\}, \\
&= \min\{\min\{\max\{0.70\}\}, \\
&\quad \min\{\max\{0.70\}\}\}, \\
&= \min\{\min\{0.70\}, \min\{0.70\}\} \\
&= \min\{0.70,0.70\} = 0.70
\end{aligned}$$

$$\begin{aligned}
C(\text{Pozicija}[t_4,t_5]) &= \min\{\min\{\max\{s(\text{prodaja,nabavka})\}, \\
&\quad \min\{\max\{s(\text{nabavka,prodaja})\}\}\}, \\
&= \min\{\min\{\max\{0.80\}\}, \\
&\quad \min\{\max\{0.80\}\}\}, \\
&= \min\{\min\{0.80\}, \min\{0.80\}\} \\
&= \min\{0.80,0.80\} = 0.80
\end{aligned}$$

Stepen saglasnosti n-torki na atributu **Zarada**:

$$\begin{aligned}
C(\text{Zarada}[t_1,t_2]) &= \min\{\min\{\max\{s(\text{prosečna,prosečna}),s(\text{prosečna,visoka})\}\}, \\
&\quad \min\{\max\{s(\text{prosečna,prosečna}),\max\{s(\text{visoka,prosečna})\}\}\}, \\
&= \min\{\min\{\max\{1,0.75\}\}, \\
&\quad \min\{\max\{1\},\max\{0.75\}\}\}, \\
&= \min\{\min\{1\}, \min\{1,0.75\}\} \\
&= \min\{1,0.75\} = 0.75
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_1,t_3]) &= \min\{\min\{\max\{s(\text{prosečna,veoma visoka})\}\}, \\
&\quad \min\{\max\{s(\text{veoma visoka,prosečna})\}\}\}, \\
&= \min\{\min\{\max\{0.35\}\}, \\
&\quad \min\{\max\{0.35\}\}\}, \\
&= \min\{\min\{0.35\}, \min\{0.35\}\} \\
&= \min\{0.35,0.35\} = 0.35
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_1,t_4]) &= \min\{\min\{\max\{s(\text{prosečna,mala}),s(\text{prosečna,prosečna})\}\}, \\
&\quad \min\{\max\{s(\text{mala,prosečna}),\max\{s(\text{prosečna,prosečna})\}\}\}, \\
&= \min\{\min\{\max\{0.85,1\}\}, \\
&\quad \min\{\max\{0.85\},\max\{1\}\}\}, \\
&= \min\{\min\{1\}, \min\{0.85,1\}\} \\
&= \min\{1,0.85\} = 0.85
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_1,t_5]) &= \min\{\min\{\max\{s(\text{prosečna,visoka})\}\}, \\
&\quad \min\{\max\{s(\text{visoka,prosečna})\}\}\}, \\
&= \min\{\min\{\max\{0.75\}\}, \\
&\quad \min\{\max\{0.75\}\}\}, \\
&= \min\{\min\{0.75\}, \min\{0.75\}\} \\
&= \min\{0.75,0.75\} = 0.75
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_2,t_3]) &= \min\{\min\{\max\{s(\text{prosečna,veom.visoka}),\max\{s(\text{visoka,veom.visoka})\}\}, \\
&\quad \min\{\max\{s(\text{veoma visoka,prosečna}),s(\text{veoma visoka,visoka})\}\}\}, \\
&= \min\{\min\{\max\{0.35\},\max\{0.80\}\}, \\
&\quad \min\{\max\{0.35,0.80\}\}\}, \\
&= \min\{\min\{0.35,0.80\}, \min\{0.80\}\} \\
&= \min\{0.35,0.80\} = 0.35
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_2, t_4]) &= \min \{ \min \{ \max \{ s(\text{prosečna, mala}), s(\text{prosečna, prosečna}) \}, \\
&\max \{ s(\text{visoka, mala}), s(\text{visoka, prosečna}) \} \}, \min \{ \max \{ s(\text{mala, prosečna}), s(\text{mala, visoka}) \}, \\
&\max \{ s(\text{prosečna, prosečna}), s(\text{prosečna, visoka}) \} \}, \\
&= \min \{ \min \{ \max \{ 0.85, 1 \}, \max \{ 0.55, 0.75 \} \}, \\
&\quad \min \{ \max \{ 0.85, 0.55 \}, \max \{ 1, 0.75 \} \} \}, \\
&= \min \{ \min \{ 1, 0.75 \}, \min \{ 0.85, 1 \} \} \\
&= \min \{ 0.75, 0.85 \} = 0.75
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_2, t_5]) &= \min \{ \min \{ \max \{ s(\text{prosečna, visoka}) \}, \max \{ s(\text{visoka, visoka}) \} \}, \\
&\quad \min \{ \max \{ s(\text{visoka, prosečna}), s(\text{visoka, visoka}) \} \}, \\
&= \min \{ \min \{ \max \{ 0.75 \}, \max \{ 1 \} \}, \\
&\quad \min \{ \max \{ 0.75, 1 \} \} \}, \\
&= \min \{ \min \{ 0.75, 1 \}, \min \{ 1 \} \} \\
&= \min \{ 0.75, 1 \} = 0.75
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_3, t_4]) &= \min \{ \min \{ \max \{ s(\text{veoma visoka, mala}), s(\text{veoma visoka, prosečna}) \} \}, \\
&\quad \min \{ \max \{ s(\text{mala, veoma visoka}), \max \{ s(\text{prosečna, veoma visoka}) \} \} \} \\
&= \min \{ \min \{ \max \{ 0.25, 0.35 \} \}, \\
&\quad \min \{ \max \{ 0.25 \}, \max \{ 0.35 \} \} \} \\
&= \min \{ \min \{ 0.35 \}, \min \{ 0.25, 1 \} \} \\
&= \min \{ 0.35, 0.25 \} = 0.25
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_3, t_5]) &= \min \{ \min \{ \max \{ s(\text{veoma visoka, visoka}) \} \}, \\
&\quad \min \{ \max \{ s(\text{visoka, veoma visoka}) \} \} \} \\
&= \min \{ \min \{ \max \{ 0.80 \} \}, \\
&\quad \min \{ \max \{ 0.80 \} \} \} \\
&= \min \{ \min \{ 0.80 \}, \min \{ 0.80 \} \} \\
&= \min \{ 0.80, 0.80 \} = 0.80
\end{aligned}$$

$$\begin{aligned}
C(\text{Zarada}[t_4, t_5]) &= \min \{ \min \{ \max \{ s(\text{mala, visoka}) \}, \max \{ s(\text{prosečna, visoka}) \} \}, \\
&\quad \min \{ \max \{ s(\text{visoka, mala}), s(\text{visoka, prosečna}) \} \} \}, \\
&= \min \{ \min \{ \max \{ 0.55 \}, \max \{ 0.75 \} \}, \\
&\quad \min \{ \max \{ 0.55, 0.75 \} \} \}, \\
&= \min \{ \min \{ 0.55, 0.75 \}, \min \{ 0.75 \} \} \\
&= \min \{ 0.55, 0.75 \} = 0.55
\end{aligned}$$

Na ovaj način izračunali smo saglasnost n-torki na atributima fazi relacije koristeći relaciju bliskosti između elemenata datih domena. Poznavajući ove vrednosti, mi nadalje možemo da pristupimo otkrivanju fazi funkcionalnih zavisnosti koristeći metodologiju zasnovanu na fazi implikacijama koju smo predstavili u prethodnom poglavlju, a koja nam omogućava da odredimo jačinu zavisnosti između atributa fazi relacije. Vrednosti $C(A_i[t_i, t_j])$ prikazane su u tabeli 50. Dobijene zavisnosti predstavljaju IF – THEN pravila u kojima koristimo lingvističke kvantifikatore kao što su prilično, značajno, itd...U slučajevima kada između atributa fazi relacije postoji fazi funkcionalna zavisnost u više n-torki koriste se t-norma funkcije, a zbog efikasnosti algoritma odabrali smo min t-normu T_{\min} .

θ_k	C(IP[t _i ,t _j])	C(God[t _i ,t _j])	C(Isk[t _i ,t _j])	C(Pro[t _i ,t _j])	C(Jez[t _i ,t _j])	C(Poz[t _i ,t _j])	C(Zar[t _i ,t _j])
t ₁ t ₂	0.70	0.30	0.30	0.50	0.20	0.70	0.75
t ₁ t ₃	0.70	0.70	0.70	0.70	0.30	0.60	0.35
t ₁ t ₄	0.70	0.95	0.70	0.80	0.20	0.50	0.85
t ₁ t ₅	0.70	0.95	0.70	0.70	0.30	0.50	0.75
t ₂ t ₃	0.70	0.40	0.50	0.80	0.60	0.90	0.35
t ₂ t ₄	0.70	0.30	0.30	0.10	0.20	0.60	0.75
t ₂ t ₅	0.70	0.40	0.30	0.70	0.60	0.60	0.75
t ₃ t ₄	0.70	0.90	0.70	0.50	0.30	0.70	0.25
t ₃ t ₅	0.70	0.90	0.85	0.70	0.60	0.70	0.80
t ₄ t ₅	0.70	0.95	0.70	0.50	0.20	0.80	0.55

Tabela 50. C(A_i[t_i,t_j]) na atributima fazi relacije na bazi relacije bliskosti

U interpretaciju rezultata istraživanja krenućemo od tabele 48. koju ćemo primeniti na datom fazi modelu baze podataka. Kao što je objašnjeno u prethodnom poglavlju, iskoristićemo vrednosti C(A_i[t_i,t_j]) da bismo utvrdili zadovoljivost pojedinih atributa za primenu u pridruženom algoritmu. Minimalnu vrednost kritične jačine zavisnosti θ_k fiksiraćemo na prag 0.7 za identifikovanje fazi funkcionalnih zavisnosti.

0.7	C(IP[t _i ,t _j])	C(God[t _i ,t _j])	C(Isk[t _i ,t _j])	C(Pro[t _i ,t _j])	C(Jez[t _i ,t _j])	C(Poz[t _i ,t _j])	C(Zar[t _i ,t _j])
t ₁ t ₂	T	×	×	×	×	T	T
t ₁ t ₃	T	T	T	T	×	×	×
t ₁ t ₄	T	T	T	T	×	×	T
t ₁ t ₅	T	T	T	T	×	×	T
t ₂ t ₃	T	×	×	T	×	T	×
t ₂ t ₄	T	×	×	×	×	×	T
t ₂ t ₅	T	×	×	T	×	×	T
t ₃ t ₄	T	T	T	×	×	T	×
t ₃ t ₅	T	T	T	T	×	T	T
t ₄ t ₅	T	T	T	×	×	T	×

Tabela 51. C(A_i[t_i,t_j]) na atributima fazi relacije na bazi relacije bliskosti sa C(A_i[t_i,t_j]) ≥ 0.70

U skladu sa ovim vrednostima, definicijom fazi funkcionalnih zavisnosti $C(Y[t_i,t_j]) \geq \min(\theta_k, C(X[t_i,t_j]))$ i metodologijom predloženom u ovoj doktorskoj disertaciji identifikujemo sledeće fazi funkcionalne zavisnosti u fazi relaciji prikazanoj u Tabeli 35.:

a) Otkrivanje fazi funkcionalnih zavisnosti primenom Mamdani-ijeve implikacije I_{MM} za slučaj $\theta_k=0.70$:

Ime_Prezime $\xrightarrow{0.70}$ **Godine, P=60%**

$$T_{\min}(\text{Ime_Prezime,Godine}) = \min(\min(0.70,0.70), \min(0.70,0.95), \min(0.70,0.95), \min(0.70,0.90), \min(0.70,0.90), \min(0.70,0.95)) = \min(0.70,0.70,0.70,0.70,0.70,0.70)=0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Iskustvo, P=60%**

$$T_{\min}(\text{Ime_Prezime,Iskustvo}) = \min(\min(0.70,0.70), \min(0.70,0.70), \min(0.70,0.70), \min(0.70,0.70), \min(0.70,0.85), \min(0.70,0.70)) = \min(0.70,0.70,0.70,0.70,0.70,0.70)=0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime ^{0.70} → **Produktivnost, P=60%**

$$T_{\min}(\text{Ime_Prezime, Produktivnost}) = \min(\min(0.70, 0.70), \min(0.70, 0.80), \min(0.70, 0.70), \min(0.70, 0.80), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime ^{0.70} → **Pozicija, P=50%**

$$T_{\min}(\text{Ime_Prezime, Pozicija}) = \min(\min(0.70, 0.70), \min(0.70, 0.90), \min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.80)) = \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Ime_Prezime ^{0.70} → **Zarada, P=60%**

$$T_{\min}(\text{Ime_Prezime, Zarada}) = \min(\min(0.70, 0.75), \min(0.70, 0.85), \min(0.70, 0.75), \min(0.70, 0.75), \min(0.70, 0.80)) = \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Godine ^{0.70} → **Iskustvo, P=10%**

$$T_{\min}(\text{Godine, Iskustvo}) = \min(\min(0.70, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Godine ^{0.70} → **Produktivnost, P=10%**

$$T_{\min}(\text{Godine, Produktivnost}) = \min(\min(0.70, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo ^{0.70} → **Produktivnost, P=30%**

$$T_{\min}(\text{Iskustvo, Produktivnost}) = \min(\min(0.70, 0.70), \min(0.70, 0.80), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Iskustvo ^{0.70} → **Pozicija, P=20%**

$$T_{\min}(\text{Iskustvo, Pozicija}) = \min(\min(0.70, 0.70), \min(0.70, 0.80)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo ^{0.70} → **Zarada, P=20%**

$$T_{\min}(\text{Iskustvo}, \text{Zarada}) = \min(\min(0.70, 0.85), \min(0.70, 0.75)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Produktivnost ^{0.70} → **Pozicija, P=20%**

$$T_{\min}(\text{Produktivnost}, \text{Pozicija}) = \min(\min(0.80, 0.90), \min(0.70, 0.70)) = \min(0.80, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Produktivnost ^{0.70} → **Zarada, P=40%**

$$T_{\min}(\text{Produktivnost}, \text{Zarada}) = \min(\min(0.80, 0.95), \min(0.70, 0.75), \min(0.70, 0.75), \min(0.70, 0.80)) = \min(0.80, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Pozicija ^{0.70} → **Zarada, P=20%**

$$T_{\min}(\text{Pozicija}, \text{Zarada}) = \min(\min(0.70, 0.75), \min(0.70, 0.80)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine ^{0.70} → **Ime_Prezime, P=10%**

$$T_{\min}(\text{Godine}, \text{Ime_Prezime}) = \min(\min(0.70, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo ^{0.70} → **Ime_Prezime, P=50%**

$$T_{\min}(\text{Iskustvo}, \text{Ime_Prezime}) = \min(\min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Iskustvo ^{0.70} → **Godine, P=60%**

$$T_{\min}(\text{Iskustvo}, \text{Godine}) = \min(\min(0.70, 0.70), \min(0.70, 0.95), \min(0.70, 0.95), \min(0.70, 0.90), \min(0.85, 0.90), \min(0.70, 0.95)) = \min(0.70, 0.70, 0.70, 0.70, 0.85, 0.70) = 0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Produktivnost ^{0.70} → **Ime_Prezime, P=40%**

$$T_{\min}(\text{Produktivnost, Ime_Prezime}) = \min(\min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Produktivnost ^{0.70} → **Godine, P=40%**

$$T_{\min}(\text{Produktivnost, Godine}) = \min(\min(0.70, 0.70), \min(0.80, 0.95), \min(0.70, 0.95), \min(0.70, 0.90)) = \min(0.70, 0.80, 0.70, 0.70) = 0.70$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Produktivnost ^{0.70} → **Iskustvo, P=30%**

$$T_{\min}(\text{Produktivnost, Iskustvo}) = \min(\min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.85)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Ime_Prezime, P=30%**

$$T_{\min}(\text{Pozicija, Ime_Prezime}) = \min(\min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Godine, P=30%**

$$T_{\min}(\text{Pozicija, Godine}) = \min(\min(0.70, 0.90), \min(0.70, 0.90), \min(0.80, 0.95)) = \min(0.70, 0.70, 0.80) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Iskustvo, P=20%**

$$T_{\min}(\text{Pozicija, Iskustvo}) = \min(\min(0.70, 0.70), \min(0.70, 0.85)) = \min(0.70, 0.85) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija ^{0.70} → **Produktivnost, P=10%**

$$T_{\min}(\text{Pozicija, Produktivnost}) = \min(\min(0.70, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.75}$ **Godine, P=30%**

$$T_{\min}(\text{Zarada, Godine}) = \min(\min(0.85, 0.95), \min(0.75, 0.95), \min(0.80, 0.90)) = \min(0.85, 0.75, 0.80) = 0.75$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Zarada $\xrightarrow{0.80}$ **Iskustvo, P=10%**

$$T_{\min}(\text{Zarada, Iskustvo}) = \min(\min(0.80, 0.85)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pridružujući predloženu metodologiju na skup atributa prikazanim u fazi relaciji (tabela 35.) i uzimajući u obzir relacije bliskosti nad vrednostima posmatranih atributa dobili smo sledeći skup fazi funkcionalnih zavisnosti:

$\text{Ime_Prezime} \xrightarrow{0.70} \text{Godine, P=60\%}$	$\text{Produktivnost} \xrightarrow{0.70} \text{Iskustvo, P=30\%}$
$\text{Godine} \xrightarrow{0.70} \text{Ime_Prezime, P=10\%}$	$\text{Iskustvo} \xrightarrow{0.70} \text{Pozicija, P=20\%}$
$\text{Ime_Prezime} \xrightarrow{0.70} \text{Iskustvo, P=60\%}$	$\text{Pozicija} \xrightarrow{0.70} \text{Iskustvo, P=20\%}$
$\text{Iskustvo} \xrightarrow{0.70} \text{Ime_Prezime, P=50\%}$	$\text{Iskustvo} \xrightarrow{0.70} \text{Zarada, P=20\%}$
$\text{Ime_Prezime} \xrightarrow{0.70} \text{Produktivnost, P=60\%}$	$\text{Zarada} \xrightarrow{0.80} \text{Iskustvo, P=10\%}$
$\text{Produktivnost} \xrightarrow{0.70} \text{Ime_Prezime, P=40\%}$	$\text{Produktivnost} \xrightarrow{0.70} \text{Pozicija, P=20\%}$
$\text{Ime_Prezime} \xrightarrow{0.70} \text{Pozicija, P=50\%}$	$\text{Pozicija} \xrightarrow{0.70} \text{Produktivnost, P=10\%}$
$\text{Pozicija} \xrightarrow{0.70} \text{Ime_Prezime, P=30\%}$	$\text{Produktivnost} \xrightarrow{0.70} \text{Zarada, P=40\%}$
$\text{Ime_Prezime} \xrightarrow{0.70} \text{Zarada, P=60\%}$	$\text{Pozicija} \xrightarrow{0.70} \text{Zarada, P=20\%}$
$\text{Godine} \xrightarrow{0.70} \text{Iskustvo, P=10\%}$	$\text{Pozicija} \xrightarrow{0.70} \text{Godine, P=30\%}$
$\text{Iskustvo} \xrightarrow{0.70} \text{Godine, P=60\%}$	$\text{Zarada} \xrightarrow{0.75} \text{Godine, P=30\%}$
$\text{Godine} \xrightarrow{0.70} \text{Produktivnost, P=10\%}$	
$\text{Produktivnost} \xrightarrow{0.70} \text{Godine, P=40\%}$	
$\text{Iskustvo} \xrightarrow{0.70} \text{Produktivnost, P=30\%}$	

Tabela 52. Fazi funkcionalne zavisnosti dobijene primenom I_{MM} . za $\theta_k=0.7$

b) Otkrivanje fazi funkcionalnih zavisnosti primenom Kleene-Diense implikacije I_{KL} za slučaj $\theta_k=0.70$:

Ime_Prezime $\xrightarrow{0.70}$ **Godine, P=60%**

$$\begin{aligned} T_{\min}(\text{Ime_Prezime}, \text{Godine}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.95), \max(1-0.70, 0.95), \\ &\max(1-0.70, 0.90), \max(1-0.70, 0.90), \max(1-0.70, 0.95)) = \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.95), \max(0.30, 0.95), \max(0.30, 0.90), \max(0.30, 0.90), \\ &\max(0.30, 0.95)) = \min(0.70, 0.95, 0.95, 0.90, 0.90, 0.95) = 0.70 \end{aligned}$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Iskustvo, P=60%**

$$\begin{aligned} T_{\min}(\text{Ime_Prezime}, \text{Iskustvo}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70), \\ &\max(1-0.70, 0.70), \max(1-0.70, 0.85), \max(1-0.70, 0.70)) = \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.85), \\ &\max(0.30, 0.70)) = \min(0.70, 0.70, 0.70, 0.70, 0.85, 0.70) = 0.70 \end{aligned}$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Produktivnost, P=60%**

$$\begin{aligned} T_{\min}(\text{Ime_Prezime}, \text{Iskustvo}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.80), \max(1-0.70, 0.70), \\ &\max(1-0.70, 0.80), \max(1-0.70, 0.70), \max(1-0.70, 0.70)) = \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.80), \max(0.30, 0.70), \max(0.30, 0.80), \max(0.30, 0.70), \\ &\max(0.30, 0.70)) = \min(0.70, 0.80, 0.70, 0.80, 0.70, 0.70) = 0.70 \end{aligned}$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Pozicija, P=50%**

$$\begin{aligned} T_{\min}(\text{Ime_Prezime}, \text{Pozicija}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.90), \max(1-0.70, 0.70), \\ &\max(1-0.70, 0.70), \max(1-0.70, 0.80)) = \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.90), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.80)) = \\ &= \min(0.70, 0.90, 0.70, 0.70, 0.80) = 0.70 \end{aligned}$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Zarada, P=60%**

$$\begin{aligned} T_{\min}(\text{Ime_Prezime}, \text{Iskustvo}) &= \min(\max(1-0.70, 0.75), \max(1-0.70, 0.85), \max(1-0.70, 0.75), \\ &\max(1-0.70, 0.75), \max(1-0.70, 0.75), \max(1-0.70, 0.80)) = \\ &= \min(\max(0.30, 0.75), \max(0.30, 0.85), \max(0.30, 0.75), \max(0.30, 0.75), \max(0.30, 0.75), \\ &\max(0.30, 0.80)) = \min(0.75, 0.85, 0.75, 0.75, 0.75, 0.80) = 0.70 \end{aligned}$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Godine $\xrightarrow{0.70}$ **Iskustvo, P=10%**

$$T_{\min}(\text{Godine}, \text{Iskustvo}) = \min(\max(1-0.70, 0.70)) = \min(\max(0.30, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Godine $\xrightarrow{0.70}$ **Produktivnost, P=10%**

$$T_{\min}(\text{Godine}, \text{Produktivnost}) = \min(\max(1-0.70, 0.70)) = \min(\max(0.30, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo $\xrightarrow{0.70}$ **Produktivnost, P=30%**

$$\begin{aligned} T_{\min}(\text{Iskustvo}, \text{Produktivnost}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.80), \max(1-0.70, 0.70)) \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.80), \max(0.30, 0.70)) = \min(0.70, 0.80, 0.70) = 0.70 \end{aligned}$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Iskustvo $\xrightarrow{0.70}$ **Pozicija, P=20%**

$$\begin{aligned} T_{\min}(\text{Iskustvo}, \text{Pozicija}) &= \min(\max(1-0.70, 0.70), \max(1-0.70, 0.80)) = \\ &= \min(\max(0.30, 0.70), \max(0.30, 0.80)) = \min(0.70, 0.80) = 0.70 \end{aligned}$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo $\xrightarrow{0.75}$ **Zarada, P=20%**

$$\begin{aligned} T_{\min}(\text{Iskustvo}, \text{Zarada}) &= \min(\max(1-0.70, 0.85), \max(1-0.70, 0.75)) = \\ &= \min(\max(0.30, 0.85), \max(0.30, 0.75)) = \min(0.85, 0.75) = 0.75 \end{aligned}$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Produktivnost $\xrightarrow{0.70}$ **Pozicija, P=20%**

$$\begin{aligned} T_{\min}(\text{Produktivnost}, \text{Pozicija}) &= \min(\max(1-0.80, 0.90), \max(1-0.70, 0.70)) = \\ &= \min(\max(0.20, 0.90), \max(0.30, 0.70)) = \min(0.90, 0.70) = 0.70 \end{aligned}$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Produktivnost ^{0.75} → **Zarada, P=40%**

$$T_{\min}(\text{Produktivnost}, \text{Zarada}) = \min(\max(1-0.80, 0.85), \max(1-0.70, 0.75), \max(1-0.70, 0.75), \max(1-0.70, 0.80)) = \min(\max(0.20, 0.85), \max(0.30, 0.75), \max(0.30, 0.75), \max(0.30, 0.80)) = \min(0.85, 0.75, 0.75, 0.80) = 0.75$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Pozicija ^{0.75} → **Zarada, P=20%**

$$T_{\min}(\text{Pozicija}, \text{Zarada}) = \min(\max(1-0.70, 0.75), \max(1-0.70, 0.80)) = \min(\max(0.30, 0.75), \max(0.30, 0.80)) = \min(0.75, 0.80) = 0.75$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Zarada ^{0.85} → **Iskustvo, P=10%**

$$T_{\min}(\text{Zarada}, \text{Iskustvo}) = \min(\max(1-0.80, 0.85)) = \min(\max(0.20, 0.85)) = 0.85$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada ^{0.90} → **Godine, P=30%**

$$T_{\min}(\text{Zarada}, \text{Godine}) = \min(\max(1-0.85, 0.95), \max(1-0.75, 0.95), \max(1-0.80, 0.90)) = \min(\max(0.15, 0.95), \max(0.25, 0.95), \max(0.20, 0.90)) = \min(0.95, 0.95, 0.90) = 0.90$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Produktivnost, P=10%**

$$T_{\min}(\text{Pozicija}, \text{Produktivnost}) = \min(\max(1-0.70, 0.70)) = \min(\max(0.30, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija ^{0.70} → **Iskustvo, P=20%**

$$T_{\min}(\text{Pozicija}, \text{Iskustvo}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.85)) = \min(\max(0.30, 0.70), \max(0.30, 0.85)) = \min(0.70, 0.85) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija ^{0.90} → **Godine, P=30%**

$$T_{\min}(\text{Pozicija}, \text{Godine}) = \min(\max(1-0.70, 0.90), \max(1-0.70, 0.90), \max(1-0.80, 0.95)) = \min(\max(0.30, 0.90), \max(0.30, 0.90), \max(0.20, 0.95)) = \min(0.90, 0.90, 0.95) = 0.90$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Ime_Prezime, P=30%**

$$T_{\min}(\text{Pozicija}, \text{Ime_Prezime}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70)) \\ = \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Produktivnost ^{0.70} → **Iskustvo, P=30%**

$$T_{\min}(\text{Produktivnost}, \text{Iskustvo}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.85)) \\ = \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.85)) = \min(0.70, 0.70, 0.85) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Produktivnost ^{0.70} → **Godine, P=40%**

$$T_{\min}(\text{Produktivnost}, \text{Godine}) = \min(\max(1-0.70, 0.70), \max(1-0.80, 0.95), \max(1-0.70, 0.95), \\ \max(1-0.70, 0.90)) = \min(\max(0.30, 0.70), \max(0.20, 0.95), \max(0.30, 0.95), \max(0.30, 0.90)) = \\ = \min(0.70, 0.95, 0.95, 0.90) = 0.70$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Produktivnost ^{0.70} → **Ime_Prezime, P=40%**

$$T_{\min}(\text{Produktivnost}, \text{Ime_Pr}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70), \\ \max(1-0.70, 0.70)) = \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70)) = \\ = \min(0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{4}{10} = 0.40, \text{ tj. } 40\%$$

Iskustvo ^{0.70} → **Godine, P=60%**

$$T_{\min}(\text{Iskustvo}, \text{Godine}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.95), \max(1-0.70, 0.95), \\ \max(1-0.70, 0.90), \max(1-0.85, 0.90), \max(1-0.70, 0.95)) = \\ = \min(\max(0.30, 0.70), \max(0.30, 0.95), \max(0.30, 0.95), \max(0.30, 0.90), \max(0.15, 0.90), \\ \max(0.30, 0.95)) = \min(0.70, 0.95, 0.95, 0.90, 0.90, 0.95) = 0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Iskustvo ^{0.70} → **Ime_Prezime, P=50%**

$$T_{\min}(\text{Iskustvo}, \text{Ime_Prezime}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70)) =$$

$$= \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70)) =$$

$$= \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Godine ^{0.70} → **Ime_Prezime, P=10%**

$$T_{\min}(\text{Godine}, \text{Ime_Prezime}) = \min(\max(1-0.70, 0.70)) = \min(\max(0.30, 0.70)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

<p>^{0.70} Ime_Prezime → Godine, P=60%</p> <p>^{0.70} Godine → Ime_Prezime, P=10%</p> <p>^{0.70} Ime_Prezime → Iskustvo, P=60%</p> <p>^{0.70} Iskustvo → Ime_Prezime, P=50%</p> <p>^{0.70} Ime_Prezime → Produktivnost, P=60%</p> <p>^{0.70} Produktivnost → Ime_Prezime, P=40%</p> <p>^{0.70} Ime_Prezime → Pozicija, P=50%</p> <p>^{0.70} Pozicija → Ime_Prezime, P=30%</p> <p>^{0.70} Ime_Prezime → Zarada, P=60%</p> <p>^{0.70} Godine → Iskustvo, P=10%</p> <p>^{0.70} Iskustvo → Godine, P=60%</p> <p>^{0.70} Godine → Produktivnost, P=10%</p> <p>^{0.70} Produktivnost → Godine, P=40%</p> <p>^{0.70} Iskustvo → Produktivnost, P=30%</p> <p>^{0.70} Produktivnost → Iskustvo, P=30%</p> <p>^{0.70} Iskustvo → Pozicija, P=20%</p>	<p>^{0.70} Pozicija → Iskustvo, P=20%</p> <p>^{0.75} Iskustvo → Zarada, P=20%</p> <p>^{0.85} Zarada → Iskustvo, P=10%</p> <p>^{0.70} Produktivnost → Pozicija, P=20%</p> <p>^{0.70} Pozicija → Produktivnost, P=10%</p> <p>^{0.75} Produktivnost → Zarada, P=40%</p> <p>^{0.75} Pozicija → Zarada, P=20%</p> <p>^{0.90} Pozicija → Godine, P=30%</p> <p>^{0.90} Zarada → Godine, P=30%</p>
--	--

Tabela 53. Fazi funkcionalne zavisnosti dobijene primenom I_{KL} za $\theta_k=0.7$

Ukoliko bismo minimalni prag jačine zavisnosti θ_k povećali na vrednost 0.8, finalni skup fazi funkcionalnih zavisnosti izgledao bi značajno drugačije. Iz tabele 54. vidimo da je broj parova n-torki u kojima postoji saglasnost na posmatranim atributima mnogo manji u poređenju sa tabelom 51. Stoga, logično je da će skup fazi funkcionalnih zavisnosti u ovom slučaju biti drastično promenjen.

0.8	C(IP[t _i ,t _j])	C(God[t _i ,t _j])	C(Isk[t _i ,t _j])	C(Pro[t _i ,t _j])	C(Jez[t _i ,t _j])	C(Poz[t _i ,t _j])	C(Zar[t _i ,t _j])
t ₁ t ₂	×	×	×	×	×	×	×
t ₁ t ₃	×	×	×	×	×	×	×
t ₁ t ₄	×	T	×	T	×	×	T
t ₁ t ₅	×	T	×	×	×	×	×
t ₂ t ₃	×	×	×	T	×	T	×
t ₂ t ₄	×	×	×	×	×	×	×
t ₂ t ₅	×	×	×	×	×	×	×
t ₃ t ₄	×	T	×	×	×	×	×
t ₃ t ₅	×	T	T	×	×	×	T
t ₄ t ₅	×	T	×	×	×	T	×

Tabela 54. C(A_i[t_i,t_j]) na atributima fazi relacije na bazi relacije bliskosti sa C(A_i[t_i,t_j]) ≥ 0.80

c) Otkrivanje fazi funkcionalnih zavisnosti primenom Kleene-Diense implikacije I_{KL} za slučaj $\theta_k=0.80$:

Produktivnost $\xrightarrow{0.90}$ **Pozicija, P=10%**

$$T_{\min}(\text{Produktivnost}, \text{Pozicija}) = \min(\max(1-0.80, 0.90)) = \min(\max(0.20, 0.90)) = 0.90$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost $\xrightarrow{0.85}$ **Zarada, P=10%**

$$T_{\min}(\text{Produktivnost}, \text{Zarada}) = \min(\max(1-0.80, 0.85)) = \min(\max(0.20, 0.85)) = 0.85$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.85}$ **Iskustvo, P=10%**

$$T_{\min}(\text{Zarada}, \text{Iskustvo}) = \min(\max(1-0.80, 0.85)) = \min(\max(0.20, 0.85)) = 0.85$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.90}$ **Godine, P=20%**

$$T_{\min}(\text{Zarada}, \text{Godine}) = \min(\max(1-0.85, 0.95), \max(1-0.80, 0.90)) = \\ = \min(\max(0.15, 0.95), \max(0.20, 0.90)) = \min(0.95, 0.90) = 0.90$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija $\xrightarrow{0.95}$ **Godine, P=10%**

$$T_{\min}(\text{Pozicija}, \text{Godine}) = \min(\max(1-0.80, 0.95)) = \min(\max(0.20, 0.95)) = 0.95$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost $\xrightarrow{0.95}$ **Godine, P=10%**

$$T_{\min}(\text{Produktivnost, Godine}) = \min(\max(1-0.80, 0.95)) = \min(\max(0.20, 0.95)) = 0.95$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo $\xrightarrow{0.90}$ **Godine, P=10%**

$$T_{\min}(\text{Iskustvo, Godine}) = \min(\max(1-0.85, 0.90)) = \min(\max(0.15, 0.90)) = 0.90$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

d) Otkrivanje fazi funkcionalnih zavisnosti primenom Mamdani-ijeve implikacije I_{MM} za slučaj $\theta_k = 0.80$:

Produktivnost $\xrightarrow{0.80}$ **Pozicija, P=10%**

$$T_{\min}(\text{Produktivnost, Pozicija}) = \min(\min(0.80, 0.90)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost $\xrightarrow{0.80}$ **Zarada, P=10%**

$$T_{\min}(\text{Produktivnost, Zarada}) = \min(\min(0.80, 0.85)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.80}$ **Iskustvo, P=10%**

$$T_{\min}(\text{Zarada, Iskustvo}) = \min(\min(0.80, 0.85)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.80}$ **Godine, P=20%**

$$T_{\min}(\text{Zarada, Godine}) = \min(\min(0.85, 0.95), \min(0.80, 0.90)) = \min(0.85, 0.80) = 0.80$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija $\xrightarrow{0.80}$ **Godine, P=10%**

$$T_{\min}(\text{Pozicija, Godine}) = \min(\min(0.80, 0.95)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost $\xrightarrow{0.80}$ **Godine, P=10%**

$$T_{\min}(\text{Produktivnost, Godine}) = \min(\min(0.80, 0.95)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo $\xrightarrow{0.85}$ **Godine, P=10%**

$$T_{\min}(\text{Iskustvo, Godine}) = \min(\min(0.85, 0.90)) = 0.85$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Može se videti da se izborom odgovarajuće vrednosti za θ_k finalni skup fazi funkcionalnih zavisnosti menja u značajnoj meri. Projektant informacionog sistema, na bazi svoje procene i interesa i krajnjeg cilja istraživanja, bira vrednost. Ovde ističemo da na izbor vrednosti θ_k utiče iskustvo i ekspertsko znanje samog dizajnera.

U tabeli 55. su prikazane fazi funkcionalne zavisnosti dobijene primenom Mamdani-ijeve I_{MM} i Kleene-Diense I_{KL} fazi implikacije kada smo vrednost kritične jačine zavisnosti θ_k promenili sa 0.70 na 0.80.

$\theta_k=0.80$	Kleene-Diense	Mamdani	P
1.	Produktivnost $\xrightarrow{0.90}$ Pozicija,	Produktivnost $\xrightarrow{0.80}$ Pozicija	10%
2.	Produktivnost $\xrightarrow{0.85}$ Zarada,	Produktivnost $\xrightarrow{0.80}$ Zarada	10%
3.	Zarada $\xrightarrow{0.85}$ Iskustvo	Zarada $\xrightarrow{0.80}$ Iskustvo	10%
4.	Zarada $\xrightarrow{0.90}$ Godine	Zarada $\xrightarrow{0.80}$ Godine	20%
5.	Pozicija $\xrightarrow{0.95}$ Godine	Pozicija $\xrightarrow{0.80}$ Godine	10%
6.	Produktivnost $\xrightarrow{0.95}$ Godine	Produktivnost $\xrightarrow{0.80}$ Godine	10%
7.	Iskustvo $\xrightarrow{0.90}$ Godine	Iskustvo $\xrightarrow{0.85}$ Godine	10%

Tabela 55. Poređenje FFDs dobijenih primenom I_{MM} i I_{KL} za $\theta_k=0.8$

Na datim skupovima zavisnih podataka možemo da koristimo pravila izvođenja za fazi funkcionalne zavisnosti, a koja nam omogućavaju da dođemo do novih zavisnosti. Ove zavisnosti predstavljaju logičku posledicu otkrivenih funkcionalnih zavisnosti. Ova pravila prikazana su u poglavlju 3. (tačka 3.5.1). Za slučaj primene I_{KL} imamo:

Produktivnost $\xrightarrow{0.90}$ Godine

Produktivnost $\xrightarrow{0.85}$ Iskustvo, Godine

Zarada $\xrightarrow{0.85}$ Godine

Produktivnost $\xrightarrow{0.85}$ Pozicija, Zarada, Godine

Zarada $\xrightarrow{0.85}$ Iskustvo, Godine

Na ovaj način obogatili smo skup fazi funkcionalnih zavisnosti F sa novim zavisnostima. Prošireni skup označavamo sa F^+ .

6.3. Algoritam za pretragu fazi funkcionalnih zavisnosti u fazi modelu na bazi mogućnosti distribucije

Nakon što smo utvrdili postojanje fazi funkcionalnih zavisnosti u fazi modelu na bazi relacije bliskosti, predloženi algoritam ćemo koristiti i u fazi modelu na bazi mogućnosti distribucije. Osnovna razlika između ova dva fazi modela baza podataka leži u načinu utvrđivanja saglasnosti n -torki na posmatranim atributima. Naime, u ovom slučaju saglasnost n -torki $t = (\pi_{A1}, \pi_{A2}, \dots, \pi_{An})$ i $t' = (\pi'_{A1}, \pi'_{A2}, \dots, \pi'_{An})$ na atributu A_i zavise od mogućnosti distribucija π_{A_i} i fazi relacije bliskosti definisane na mogućim vrednostima datog atributa. Na primeru iz tabele 42., saglasnost n -torki na datim atributima su:

Saglasnost n -torki na atributu **Ime_Prezime:**

$$C(\text{Ime_Prezime}[t_1, t_2]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_1, t_3]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_1, t_4]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_1, t_5]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_2, t_3]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_2, t_4]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_2, t_5]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_3, t_4]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_3, t_5]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Ime_Prezime}[t_4, t_5]) = \min(1, 1, 0.70) = 0.70$$

Saglasnost n -torki na atributu **Godine:**

$$C(\text{Godine}[t_1, t_2]) = \min(0.61, 0.21, 0.30) = 0.21$$

$$C(\text{Godine}[t_1, t_3]) = \min(0.61, 0.82, 0.70) = 0.61$$

$$C(\text{Godine}[t_1, t_4]) = \min(0.61, 0.78, 0.70) = 0.61$$

$$C(\text{Godine}[t_1, t_5]) = \min(0.61, 0.85, 1) = 0.61$$

$$C(\text{Godine}[t_2, t_3]) = \min(0.21, 0.82, 0.50) = 0.21$$

$$C(\text{Godine}[t_2, t_4]) = \min(0.21, 0.58, 0.50) = 0.21$$

$$C(\text{Godine}[t_2, t_5]) = \min(0.21, 0.85, 0.30) = 0.21$$

$$C(\text{Godine}[t_3, t_4]) = \min(0.82, 0.78, 1) = 0.78$$

$$C(\text{Godine}[t_3, t_5]) = \min(0.82, 0.85, 0.70) = 0.70$$

$$C(\text{Godine}[t_4, t_5]) = \min(0.78, 0.85, 0.70) = 0.70$$

Saglasnost n-torki na atributu **Iskustvo:**

$$C(\text{Iskustvo}[t_1, t_2]) = \min(0.76, 0.46, 0.40) = 0.40$$

$$C(\text{Iskustvo}[t_1, t_3]) = \min(0.76, 0.71, 0.80) = 0.71$$

$$C(\text{Iskustvo}[t_1, t_4]) = \min(0.76, 0.89, 1) = 0.76$$

$$C(\text{Iskustvo}[t_1, t_5]) = \min(0.76, 0.73, 0.80) = 0.73$$

$$C(\text{Iskustvo}[t_2, t_3]) = \min(0.46, 0.71, 0.60) = 0.46$$

$$C(\text{Iskustvo}[t_2, t_4]) = \min(0.46, 0.89, 0.40) = 0.40$$

$$C(\text{Iskustvo}[t_2, t_5]) = \min(0.46, 0.73, 0.60) = 0.46$$

$$C(\text{Iskustvo}[t_3, t_4]) = \min(0.71, 0.89, 0.80) = 0.71$$

$$C(\text{Iskustvo}[t_3, t_5]) = \min(0.71, 0.73, 1) = 0.71$$

$$C(\text{Iskustvo}[t_4, t_5]) = \min(0.89, 0.73, 0.80) = 0.73$$

Saglasnost n-torki na atributu **Produktivnost:**

$$C(\text{Produktivnost}[t_1, t_2]) = \min(0.65, 0.79, 0.65) = 0.65$$

$$C(\text{Produktivnost}[t_1, t_3]) = \min(0.65, 0.82, 1) = 0.65$$

$$C(\text{Produktivnost}[t_1, t_4]) = \min(0.65, 0.90, 0.75) = 0.65$$

$$C(\text{Produktivnost}[t_1, t_5]) = \min(0.65, 0.60, 0.65) = 0.60$$

$$C(\text{Produktivnost}[t_2, t_3]) = \min(0.79, 0.82, 0.65) = 0.65$$

$$C(\text{Produktivnost}[t_2, t_4]) = \min(0.79, 0.90, 0.45) = 0.45$$

$$C(\text{Produktivnost}[t_2, t_5]) = \min(0.79, 0.60, 1) = 0.60$$

$$C(\text{Produktivnost}[t_3, t_4]) = \min(0.82, 0.90, 0.75) = 0.75$$

$$C(\text{Produktivnost}[t_3, t_5]) = \min(0.82, 0.60, 0.65) = 0.60$$

$$C(\text{Produktivnost}[t_4, t_5]) = \min(0.90, 0.60, 0.45) = 0.45$$

Saglasnost n-torki na atributu **Jezik:**

$$C(\text{Jezik}[t_1, t_2]) = \min(1, 1, 1) = 1$$

$$C(\text{Jezik}[t_1, t_3]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Jezik}[t_1, t_4]) = \min(1, 1, 0.20) = 0.20$$

$$C(\text{Jezik}[t_1, t_5]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Jezik}[t_2, t_3]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Jezik}[t_2, t_4]) = \min(1, 1, 0.20) = 0.20$$

$$C(\text{Jezik}[t_2, t_5]) = \min(1, 1, 0.30) = 0.30$$

$$C(\text{Jezik}[t_3, t_4]) = \min(1, 1, 1) = 1$$

$$C(\text{Jezik}[t_3, t_5]) = \min(1, 1, 1) = 1$$

$$C(\text{Jezik}[t_4, t_5]) = \min(1, 1, 0.30) = 0.30$$

Saglasnost n-torki na atributu **Pozicija**:

$$C(\text{Pozicija}[t_1, t_2]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Pozicija}[t_1, t_3]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Pozicija}[t_1, t_4]) = \min(1, 1, 0.50) = 0.50$$

$$C(\text{Pozicija}[t_1, t_5]) = \min(1, 1, 0.50) = 0.50$$

$$C(\text{Pozicija}[t_2, t_3]) = \min(1, 1, 0.90) = 0.90$$

$$C(\text{Pozicija}[t_2, t_4]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Pozicija}[t_2, t_5]) = \min(1, 1, 0.60) = 0.60$$

$$C(\text{Pozicija}[t_3, t_4]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Pozicija}[t_3, t_5]) = \min(1, 1, 0.70) = 0.70$$

$$C(\text{Pozicija}[t_4, t_5]) = \min(1, 1, 0.80) = 0.80$$

Saglasnost n-torki na atributu **Zarada**:

$$C(\text{Zarada}[t_1, t_2]) = \min(0.81, 0.92, 0.90) = 0.81$$

$$C(\text{Zarada}[t_1, t_3]) = \min(0.81, 0.89, 0.45) = 0.45$$

$$C(\text{Zarada}[t_1, t_4]) = \min(0.81, 0.67, 0.95) = 0.67$$

$$C(\text{Zarada}[t_1, t_5]) = \min(0.81, 0.32, 1) = 0.32$$

$$C(\text{Zarada}[t_2, t_3]) = \min(0.92, 0.89, 0.90) = 0.89$$

$$C(\text{Zarada}[t_2, t_4]) = \min(0.92, 0.67, 0.55) = 0.55$$

$$C(\text{Zarada}[t_2, t_5]) = \min(0.92, 0.32, 0.90) = 0.32$$

$$C(\text{Zarada}[t_3, t_4]) = \min(0.89, 0.67, 0.25) = 0.25$$

$$C(\text{Zarada}[t_3, t_5]) = \min(0.89, 0.32, 0.45) = 0.32$$

$$C(\text{Zarada}[t_4, t_5]) = \min(0.67, 0.32, 0.95) = 0.32$$

Vrednosti usaglašenosti n-torki na atributima fazi relacije na bazi mogućnosti distribucije prikazane u tabeli 56.

θ_k	$C(\text{IP}[t_i, t_j])$	$C(\text{God}[t_i, t_j])$	$C(\text{Isk}[t_i, t_j])$	$C(\text{Pro}[t_i, t_j])$	$C(\text{Jez}[t_i, t_j])$	$C(\text{Poz}[t_i, t_j])$	$C(\text{Zar}[t_i, t_j])$
$t_1 t_2$	0.70	0.21	0.40	0.65	1	0.70	0.81
$t_1 t_3$	0.70	0.61	0.71	0.65	0.60	0.60	0.45
$t_1 t_4$	0.70	0.61	0.76	0.65	0.20	0.50	0.67
$t_1 t_5$	0.70	0.61	0.73	0.60	0.60	0.50	0.32
$t_2 t_3$	0.70	0.21	0.46	0.65	0.60	0.90	0.89
$t_2 t_4$	0.70	0.21	0.40	0.45	0.20	0.60	0.55
$t_2 t_5$	0.70	0.21	0.46	0.60	0.30	0.60	0.32
$t_3 t_4$	0.70	0.78	0.71	0.75	1	0.70	0.25
$t_3 t_5$	0.70	0.70	0.71	0.60	1	0.70	0.32
$t_4 t_5$	0.70	0.70	0.73	0.45	0.30	0.80	0.32

Tabela 56.. $C(A_i[t_i, t_j])$ na atributima fazi relacije na bazi mogućnosti distribucije

Kao i u prethodnom slučaju, u zavisnosti od kritične jačine zavisnosti θ_k pristupićeemo analizi fazi funkcionalnih zavisnosti u fazi relaciji na bazi mogućnosti distribucije. Za slučaj $\theta_k=0.70$. u sledećoj tabeli prikazani su parovi n-torki u kojima se potencijalno kriju fazi funkcionalne zavisnosti.

0.7	C(IP[t _i ,t _j])	C(God[t _i ,t _j])	C(Isk[t _i ,t _j])	C(Pro[t _i ,t _j])	C(Jez[t _i ,t _j])	C(Poz[t _i ,t _j])	C(Zar[t _i ,t _j])
t ₁ t ₂	T	×	×	×	T	T	T
t ₁ t ₃	T	×	T	×	×	×	×
t ₁ t ₄	T	×	T	×	×	×	×
t ₁ t ₅	T	×	T	×	×	×	×
t ₂ t ₃	T	×	×	T	×	T	T
t ₂ t ₄	T	×	×	×	×	×	×
t ₂ t ₅	T	×	×	×	×	×	×
t ₃ t ₄	T	T	T	T	T	T	×
t ₃ t ₅	T	T	T	×	T	T	×
t ₄ t ₅	T	T	T	×	×	T	×

Tabela 57. C(A_i[t_i,t_j]) na atributima fazi relacije na bazi mogućnosti distribucije za slučaj C(A_i[t_i,t_j]) \geq 0.70

a) Otkrivanje fazi funkcionalnih zavisnosti u fazi modelu na bazi mogućnosti distribucije primenom Mamdani-ijeve implikacije I_{MM} za slučaj $\theta_k=0.70$:

Ime_Prezime $\xrightarrow{0.70}$ **Godine, P=30%**

$$T_{\min}(\text{Ime_Prezime,Godine}) = \min(\min(0.70,0.78), \min(0.70,0.70), \min(0.70,0.70)) = \min(0.70,0.70,0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Iskustvo, P=60%**

$$T_{\min}(\text{Ime_Prezime,Iskustvo}) = \min(\min(0.70,0.71), \min(0.70,0.76), \min(0.70,0.73), \min(0.70,0.71), \min(0.70,0.71), \min(0.70,0.73)) = \min(0.70,0.70,0.70,0.70,0.70,0.70) = 0.70$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Produktivnost, P=10%**

$$T_{\min}(\text{Ime_Prezime,Produktivnost}) = \min(\min(0.70,0.75)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Ime_Prezime $\xrightarrow{0.70}$ **Jezik, P=30%**

$$T_{\min}(\text{Ime_Prezime,Jezik}) = \min(\min(0.70,1), \min(0.70,1), \min(0.70,1)) = \min(0.70,0.70,0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Ime_Prezime^{0.70} → **Pozicija, P=50%**

$$T_{\min}(\text{Ime_Prezime}, \text{Pozicija}) = \min(\min(0.70, 0.70), \min(0.70, 0.90), \min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.80)) = \min(0.70, 0.70, 0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Ime_Prezime^{0.70} → **Zarada, P=20%**

$$T_{\min}(\text{Ime_Prezime}, \text{Zarada}) = \min(\min(0.70, 0.81), \min(0.70, 0.89)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine^{0.70} → **Iskustvo, P=20%**

$$T_{\min}(\text{Godine}, \text{Iskustvo}) = \min(\min(0.70, 0.71), \min(0.70, 0.73)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine^{0.70} → **Jezik, P=20%**

$$T_{\min}(\text{Godine}, \text{Jezik}) = \min(\min(0.78, 1), \min(0.70, 1)) = \min(0.78, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine^{0.70} → **Pozicija, P=20%**

$$T_{\min}(\text{Godine}, \text{Pozicija}) = \min(\min(0.70, 0.70), \min(0.70, 0.80)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo^{0.71} → **Produktivnost, P=10%**

$$T_{\min}(\text{Iskustvo}, \text{Produktivnost}) = \min(\min(0.71, 0.75)) = 0.71$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo^{0.71} → **Jezik, P=20%**

$$T_{\min}(\text{Iskustvo}, \text{Jezik}) = \min(\min(0.71, 1), \min(0.71, 1)) = \min(0.71, 0.71) = 0.71$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo^{0.73} → **Pozicija, P=10%**

$$T_{\min}(\text{Iskustvo}, \text{Pozicija}) = \min(\min(0.73, 0.80)) = 0.73$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost ^{0.75} → **Jezik, P=10%**

$$T_{\min}(\text{Produktivnost, Jezik}) = \min(\min(0.75, 1)) = 0.75$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija ^{0.70} → **Zarada, P=10%**

$$T_{\min}(\text{Pozicija, Zarada}) = \min(\min(0.70, 0.81)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada ^{0.89} → **Pozicija, P=10%**

$$T_{\min}(\text{Zarada, Pozicija}) = \min(\min(0.89, 0.90)) = 0.89$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada ^{0.81} → **Jezik, P=10%**

$$T_{\min}(\text{Zarada, Jezik}) = \min(\min(0.81, 1)) = 0.81$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija ^{0.70} → **Jezik, P=30%**

$$T_{\min}(\text{Pozicija, Jezik}) = \min(\min(0.70, 1), \min(0.70, 1), \min(0.70, 1)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija ^{0.70} → **Produktivnost, P=10%**

$$T_{\min}(\text{Pozicija, Produktivnost}) = \min(\min(0.70, 0.75)) = 0.70$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija ^{0.70} → **Godine, P=20%**

$$T_{\min}(\text{Pozicija, Godine}) = \min(\min(0.70, 0.78), \min(0.70, 0.70)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija ^{0.70} → **Ime_Prezime, P=30%**

$$T_{\min}(\text{Pozicija, Ime_Prezime}) = \min(\min(0.70, 0.70), \min(0.70, 0.70), \min(0.70, 0.70)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Produktivnost $\xrightarrow{0.75}$ **Godine, P=10%**

$$T_{\min}(\text{Produktivnost, Godine}) = \min(\min(0.75, 0.78)) = 0.75$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo $\xrightarrow{0.71}$ **Godine, P=10%**

$$T_{\min}(\text{Iskustvo, Godine}) = \min(\min(0.71, 0.78)) = 0.71$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Godine $\xrightarrow{0.70}$ **Ime_Prezime, P=20%**

$$T_{\min}(\text{Godine, Ime_Prezime}) = \min(\min(0.70, 0.70), \min(0.70, 0.70)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Godine, P=30%</p>	<p>$\xrightarrow{0.71}$ Iskustvo \rightarrow Jezik, P=20%</p>
<p>$\xrightarrow{0.70}$ Godine \rightarrow Ime_Prezime, P=20%</p>	<p>$\xrightarrow{0.73}$ Iskustvo \rightarrow Pozicija, P=10%</p>
<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Iskustvo, P=60%</p>	<p>$\xrightarrow{0.75}$ Produktivnost \rightarrow Jezik, P=10%</p>
<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Produktivnost, P=10%</p>	<p>$\xrightarrow{0.70}$ Pozicija \rightarrow Zarada, P=10%</p>
<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Jezik, P=30%</p>	<p>$\xrightarrow{0.89}$ Zarada \rightarrow Pozicija, P=10%</p>
<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Pozicija, P=50%</p>	<p>$\xrightarrow{0.81}$ Zarada \rightarrow Jezik, P=10%</p>
<p>$\xrightarrow{0.70}$ Pozicija \rightarrow Ime_Prezime, P=30%</p>	<p>$\xrightarrow{0.70}$ Pozicija \rightarrow Jezik, P=30%</p>
<p>$\xrightarrow{0.70}$ Ime_Prezime \rightarrow Zarada, P=20%</p>	<p>$\xrightarrow{0.70}$ Pozicija \rightarrow Produktivnost, P=10%</p>
<p>$\xrightarrow{0.70}$ Godine \rightarrow Iskustvo, P=20%</p>	<p>$\xrightarrow{0.75}$ Produktivnost \rightarrow Godine, P=10%</p>
<p>$\xrightarrow{0.71}$ Iskustvo \rightarrow Godine, P=10%</p>	
<p>$\xrightarrow{0.70}$ Godine \rightarrow Jezik, P=20%</p>	
<p>$\xrightarrow{0.70}$ Godine \rightarrow Pozicija, P=20%</p>	
<p>$\xrightarrow{0.70}$ Pozicija \rightarrow Godine, P=20%</p>	
<p>$\xrightarrow{0.71}$ Iskustvo \rightarrow Produktivnost, P=10%</p>	

Tabela 58. Fazi funkcionalne zavisnosti u fazi modelu na bazi moguć.distr.dobijene primenom I_{MM} . za $\theta_k=0.7$

U tabeli 58. prikazane su identifikovane fazi funkcionalne zavisnosti između atributa koji definišu fazi model na bazi mogućnosti distribucije. Ove zavisnosti možemo da prikažemo u vidu IF-THEN pravila ili lingvističkim izrazima kao što su:

"Ime i prezime službenika u većoj meri određuje njegovo iskustvo" je zavisnost sa lingvističkom jačinom 0.70 ili

"Službenici približnih godina otprilike imaju i slično iskustvo" ima lingvističku jačinu 0.70 ili

"Zarada službenika u manjoj meri određuje i jezik koji govori" ima lingvističku jačinu 0.81.

Ove zavisnosti ne određuju precizan nivo skrivenih znanja, ali ipak osiguravaju potreban nivo neophodan za donošenje upravljačkih odluka. O praktičnim primenama fazi funkcionalnih zavisnosti biće reči u ovom poglavlju doktorske disertacije.

b) Otkrivanje fazi funkcionalnih zavisnosti u fazi modelu na bazi mogućnosti distribucije primenom Kleene-Diense implikacije I_{KL} za slučaj $\theta_k=0.70$:

Ime_Prezime ^{0.70} → Godine, P=30%

$$T_{\min}(\text{Ime_Prezime}, \text{Godine}) = \min(\max(1-0.70, 0.78), \max(1-0.70, 0.70), \max(1-0.70, 0.70)) \\ = \min(\max(0.30, 0.78), \max(0.30, 0.70), \max(0.30, 0.70)) = \min(0.78, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Ime_Prezime ^{0.71} → Iskustvo, P=60%

$$T_{\min}(\text{Ime_Prezime}, \text{Iskustvo}) = \min(\max(1-0.70, 0.71), \max(1-0.70, 0.76), \max(1-0.70, 0.73), \\ \max(1-0.70, 0.71), \max(1-0.70, 0.71), \max(1-0.70, 0.73)) = \min(\max(0.30, 0.71), \\ \max(0.30, 0.76), \max(0.30, 0.73), \max(0.30, 0.71), \max(0.30, 0.71), \max(0.30, 0.73)) = \\ = \min(0.71, 0.76, 0.73, 0.71, 0.71, 0.73) = 0.71$$

$$P = \frac{6}{10} = 0.60, \text{ tj. } 60\%$$

Ime_Prezime ^{0.75} → Produktivnost, P=10%

$$T_{\min}(\text{Ime_Prezime}, \text{Produktivnost}) = \min(\max(1-0.70, 0.75)) = \min(\max(0.30, 0.75)) = 0.75$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Ime_Prezime ¹ → Jezik, P=30%

$$T_{\min}(\text{Ime_Prezime}, \text{Jezik}) = \min(\max(1-0.70, 1), \max(1-0.70, 1), \max(1-0.70, 1)) = \\ = \min(\max(0.30, 1), \max(0.30, 1), \max(0.30, 1)) = \min(1, 1, 1) = 1$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Ime_Prezime^{0.70} → Pozicija, P=50%

$$T_{\min}(\text{Ime_Prezime}, \text{Pozicija}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.90), \max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.80)) = \min(\max(0.30, 0.70), \max(0.30, 0.90), \max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.80)) = \min(0.70, 0.90, 0.70, 0.70, 0.80) = 0.70$$

$$P = \frac{5}{10} = 0.50, \text{ tj. } 50\%$$

Ime_Prezime^{0.81} → Zarada, P=20%

$$T_{\min}(\text{Ime_Prezime}, \text{Zarada}) = \min(\max(1-0.70, 0.81), \max(1-0.70, 0.89)) = \min(\max(0.30, 0.81), \max(0.30, 0.89)) = \min(0.81, 0.89) = 0.81$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine^{0.71} → Iskustvo, P=20%

$$T_{\min}(\text{Godine}, \text{Iskustvo}) = \min(\max(1-0.70, 0.71), \max(1-0.70, 0.73)) = \min(\max(0.30, 0.71), \max(0.30, 0.73)) = \min(0.71, 0.73) = 0.71$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine¹ → Jezik, P=20%

$$T_{\min}(\text{Godine}, \text{Jezik}) = \min(\max(1-0.78, 1), \max(1-0.70, 1)) = \min(\max(0.22, 1), \max(0.30, 1)) = \min(1, 1) = 1$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Godine^{0.70} → Pozicija, P=20%

$$T_{\min}(\text{Godine}, \text{Pozicija}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.80)) = \min(\max(0.30, 0.70), \max(0.30, 0.80)) = \min(0.70, 0.80) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo^{0.75} → Produktivnost, P=10%

$$T_{\min}(\text{Iskustvo}, \text{Produktivnost}) = \min(\max(1-0.71, 0.75)) = \min(\max(0.29, 0.75)) = 0.75$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo¹ → Jezik, P=20%

$$T_{\min}(\text{Iskustvo}, \text{Jezik}) = \min(\max(1-0.71, 1), \max(1-0.71, 1)) = \min(\max(0.29, 1), \max(0.21, 1)) = \min(1, 1) = 1$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Iskustvo $\xrightarrow{0.80}$ **Pozicija, P=10%**

$$T_{\min}(\text{Iskustvo}, \text{Pozicija}) = \min(\max(1-0.73, 0.80)) = \min(\max(0.27, 0.80)) = 0.80$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Produktivnost $\xrightarrow{1}$ **Jezik, P=10%**

$$T_{\min}(\text{Produktivnost}, \text{Jezik}) = \min(\max(1-0.75, 1)) = \min(\max(0.25, 1)) = 1$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija $\xrightarrow{0.81}$ **Zarada, P=10%**

$$T_{\min}(\text{Pozicija}, \text{Zarada}) = \min(\max(1-0.70, 0.81)) = \min(\max(0.30, 0.81)) = 0.81$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{0.90}$ **Pozicija, P=10%**

$$T_{\min}(\text{Zarada}, \text{Pozicija}) = \min(\max(1-0.89, 0.90)) = \min(\max(0.11, 0.90)) = 0.90$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Zarada $\xrightarrow{1}$ **Jezik, P=10%**

$$T_{\min}(\text{Zarada}, \text{Jezik}) = \min(\max(1-0.81, 1)) = \min(\max(0.19, 1)) = 1$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija $\xrightarrow{1}$ **Jezik, P=30%**

$$T_{\min}(\text{Pozicija}, \text{Jezik}) = \min(\max(1-0.70, 1), \max(1-0.70, 1), \max(1-0.70, 1)) = \\ = \min(\max(0.30, 1), \max(0.30, 1), \max(0.30, 1)) = \min(1, 1, 1) = 1$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Pozicija $\xrightarrow{0.75}$ **Produktivnost, P=10%**

$$T_{\min}(\text{Pozicija}, \text{Produktivnost}) = \min(\max(1-0.70, 0.75)) = \min(\max(0.30, 0.75)) = 0.75$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Pozicija $\xrightarrow{0.70}$ **Godine, P=20%**

$$T_{\min}(\text{Pozicija}, \text{Godine}) = \min(\max(1-0.70, 0.78), \max(1-0.70, 0.70)) = \min(\max(0.30, 0.78), \\ \max(0.30, 0.70)) = \min(0.78, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Pozicija ^{0.70} → **Ime_Prezime, P=30%**

$$T_{\min}(\text{Pozicija}, \text{Ime_Prezime}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70), \max(1-0.70, 0.70)) \\ = \min(\max(0.30, 0.70), \max(0.30, 0.70), \max(0.30, 0.70)) = \min(0.70, 0.70, 0.70) = 0.70$$

$$P = \frac{3}{10} = 0.30, \text{ tj. } 30\%$$

Produktivnost ^{0.78} → **Godine, P=10%**

$$T_{\min}(\text{Produktivnost}, \text{Godine}) = \min(\max(1-0.75, 0.78)) = \min(\max(0.25, 0.78)) = 0.78$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Iskustvo ^{0.78} → **Godine, P=10%**

$$T_{\min}(\text{Iskustvo}, \text{Godine}) = \min(\max(1-0.71, 0.78)) = \min(\max(0.29, 0.78)) = 0.78$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Godine ^{0.70} → **Ime_Prezime, P=20%**

$$T_{\min}(\text{Godine}, \text{Ime_Prezime}) = \min(\max(1-0.70, 0.70), \max(1-0.70, 0.70)) = \\ = \min(\max(0.30, 0.70), \max(0.30, 0.70)) = \min(0.70, 0.70) = 0.70$$

$$P = \frac{2}{10} = 0.20, \text{ tj. } 20\%$$

Tabela broj 59. prikazuje otkrivene fazi funkcionalne zavisnosti koje smo dobili primenom različitih fazi implikacija, tj. Mamdanijeve I_{MM} i Kleene – Diense I_{KD} . Iz tabele zaključujemo da je lingvistička jačina zavisnosti veća ili jednaka kada je računamo pomoću Kleene – Diense fazi implikacije. O ovome će biti reči u narednim odeljcima ove doktorske disertacije.

0.70 Ime_Prezime → Godine, P=30%	0.70 Ime_Prezime → Godine, P=30%
0.70 Godine → Ime_Prezime, P=20%	0.70 Godine → Ime_Prezime, P=20%
0.70 Ime_Prezime → Iskustvo, P=60%	0.71 Ime_Prezime → Iskustvo, P=60%
0.70 Ime_Prezime → Produktivnost, P=10%	0.75 Ime_Prezime → Produktivnost, P=10%
0.70 Ime_Prezime → Jezik, P=30%	1 Ime_Prezime → Jezik, P=30%
0.70 Ime_Prezime → Pozicija, P=50%	0.70 Ime_Prezime → Pozicija, P=50%
0.70 Pozicija → Ime_Prezime, P=30%	0.70 Pozicija → Ime_Prezime, P=30%
0.70 Ime_Prezime → Zarada, P=20%	0.81 Ime_Prezime → Zarada, P=20%
0.70 Godine → Iskustvo, P=20%	0.71 Godine → Iskustvo, P=20%
0.71 Iskustvo → Godine, P=10%	0.78 Iskustvo → Godine, P=10%
0.70 Godine → Jezik, P=20%	1 Godine → Jezik, P=20%
0.70 Godine → Pozicija, P=20%	0.70 Godine → Pozicija, P=20%
0.70 Pozicija → Godine, P=20%	0.70 Pozicija → Godine, P=20%
0.71 Iskustvo → Produktivnost, P=10%	0.75 Iskustvo → Produktivnost, P=10%
0.71 Iskustvo → Jezik, P=20%	1 Iskustvo → Jezik, P=20%
0.73 Iskustvo → Pozicija, P=10%	0.80 Iskustvo → Pozicija, P=10%
0.75 Produktivnost → Jezik, P=10%	1 Produktivnost → Jezik, P=10%
0.70 Pozicija → Zarada, P=10%	0.81 Pozicija → Zarada, P=10%
0.89 Zarada → Pozicija, P=10%	0.90 Zarada → Pozicija, P=10%
0.81 Zarada → Jezik, P=10%	1 Zarada → Jezik, P=10%
0.70 Pozicija → Jezik, P=30%	1 Pozicija → Jezik, P=30%
0.70 Pozicija → Produktivnost, P=10%	0.75 Pozicija → Produktivnost, P=10%
0.75 Produktivnost → Godine, P=10%	0.78 Produktivnost → Godine, P=10%

Tabela 59. FFDs dobijene primenom Mamdani-ijeve i Kleene-Diense fazi implikacije

I u ovom primeru ćemo da izračunamo i odredimo fazi funkcionalne zavisnosti kada prag kritične jačine zavisnosti θ_k povećamo na vrednost 0.80. U sledećoj tabeli su prikazani parovi n-torki u kojima je vrednost $C(A_i[t_i, t_j]) \geq 0.80$.

0.8	C(IP[t _i ,t _j])	C(God[t _i ,t _j])	C(Isk[t _i ,t _j])	C(Pro[t _i ,t _j])	C(Jez[t _i ,t _j])	C(Poz[t _i ,t _j])	C(Zar[t _i ,t _j])
t ₁ t ₂	×	×	×	×	T	×	T
t ₁ t ₃	×	×	×	×	×	×	×
t ₁ t ₄	×	×	×	×	×	×	×
t ₁ t ₅	×	×	×	×	×	×	×
t ₂ t ₃	×	×	×	×	×	T	T
t ₂ t ₄	×	×	×	×	×	×	×
t ₂ t ₅	×	×	×	×	×	×	×
t ₃ t ₄	×	×	×	×	T	×	×
t ₃ t ₅	×	×	×	×	T	×	×
t ₄ t ₅	×	×	×	×	×	T	×

Tabela 60. $C(A_i[t_i, t_j])$ na atributima fazi relacije na bazi mogućnosti distribucije za slučaj $C(A_i[t_i, t_j]) \geq 0.80$

Iz ove tabele vidimo da je moguće utvrditi postojanje dve fazi funkcionalne zavisnosti, a to su:

c) Otkrivanje fazi funkcionalnih zavisnosti u fazi modelu na bazi mogućnosti distribucije primenom Mamdani-ijeve implikacije I_{MM} za slučaj $\theta_k=0.80$:

$$\text{Zarada} \xrightarrow{0.89} \text{Pozicija, P=10\%}$$

$$T_{\min}(\text{Zarada, Pozicija}) = \min(\min(0.89, 0.90)) = 0.89$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

$$\text{Zarada} \xrightarrow{0.81} \text{Jezik, P=10\%}$$

$$T_{\min}(\text{Zarada, Jezik}) = \min(\min(0.81, 1)) = 0.81$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

d) Otkrivanje fazi funkcionalnih zavisnosti primenom Kleene-Diense implikacije I_{KL} za slučaj $\theta_k=0.80$:

$$\text{Zarada} \xrightarrow{0.90} \text{Pozicija, P=10\%}$$

$$T_{\min}(\text{Zarada, Pozicija}) = \min(\max(1-0.89, 0.90)) = \min(\max(0.11, 0.90)) = 0.90$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

$$\text{Zarada} \xrightarrow{1} \text{Jezik, P=10\%}$$

$$T_{\min}(\text{Zarada, Jezik}) = \min(\max(1-0.81, 1)) = \min(\max(0.19, 1)) = 1$$

$$P = \frac{1}{10} = 0.10, \text{ tj. } 10\%$$

Može se videti da smo u slučaju primene Kleene-Diense fazi implikacije dobili čistu funkcionalnu zavisnost $Zarada \xrightarrow{1} Jezik$. S obzirom da se u našoj fazi relaciji vrednost za zaradu z_2 pojavljuje u dve n-torke, t_1 i t_5 , vrednosti atributa Jezik u ovom slučaju nisu identične jer z_2 reprezentuje jako udaljene vrednosti.

Primenom pravila izvođenja za fazi funkcionalne zavisnosti možemo dobiti skup F proširiti u F^+ dodajući novu zavisnost $Zarada \xrightarrow{\ominus} Pozicija, Jezik$. Za slučaj primene I_{MM} $\theta=0.81$, a kada koristimo I_{KL} vrednost jačine zavisnosti $\theta=0.90$.

$\theta_k=0.80$	Kleene-Diense	Mamdani	P
1.	$Zarada \xrightarrow{0.90} Pozicija,$	$Zarada \xrightarrow{0.89} Pozicija,$	10%
2.	$Zarada \xrightarrow{1} Jezik,$	$Zarada \xrightarrow{0.81} Jezik,$	10%
3.	$Zarada \xrightarrow{0.90} Pozicija, Jezik$	$Zarada \xrightarrow{0.81} Pozicija, Jezik$	10%

Tabela 61. Skup F^+ za $\theta_k=0.8$

6.4. Upotreba fazi priloga

Korišćenjem fazi priloga fazi funkcionalne zavisnosti možemo da transliramo u lingvistička pravila i da na taj način omogućimo korisnicima informacija i saznanja iz baza podataka da ih koriste na prirodan način, u tzv. ljudski-orijentisanom modu (human-oriented mode). Ova logika dobija potvrdu u činjenici da i prirodni jezik obuhvata mnoge nepreciznosti i neodređenosti koje su kompatibilne sa prirodom fazi funkcionalnih zavisnosti. U ovoj doktorskoj disertaciji navodili smo primere fazi funkcionalnih zavisnosti, kao što su:

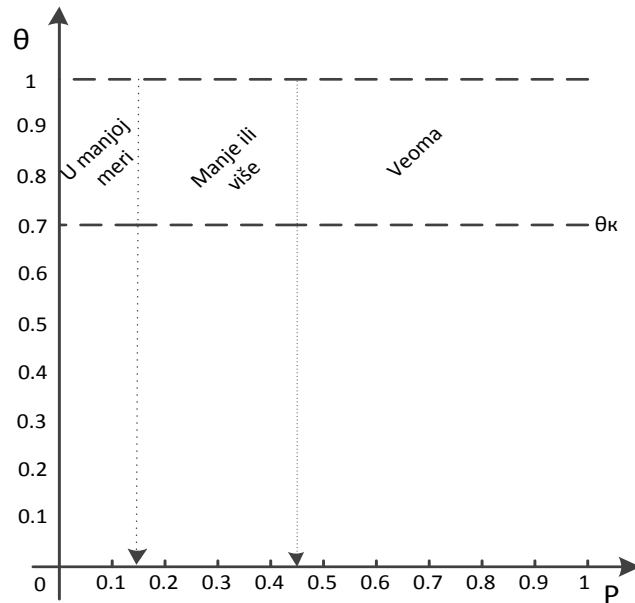
"Inteligencija manje ili više određuje uspeh studenta"

"Iskustvo veoma utiče na zaradu zaposlenog"

"Zarada službenika u manjoj meri određuje i jezik koji govori"

Upotreba fazi priloga kao što su manje ili više, otprilike, veoma, prosečno, u manjoj meri, većinom, približno itd. dozvoljava nam fleksibilnost u interpretaciji algoritma za identifikovanje fazi funkcionalnih zavisnosti. Dakle, zavisnost tipa $Pozicija \xrightarrow{\ominus} Zarada$ može da se iskaže i u lingvističkoj formi koja je prihvatljivija krajnjim korisnicima saznanja.

Da bismo reprezentovali upotrebu fazi priloga u kontekstu i interpretaciji fazi funkcionalnih zavisnosti predlažemo korišćenje jačine zavisnosti θ i parametra P za opis primene istih. Na sledećoj slici prikazan je primer kako možemo da upotrebimo pojedine lingvističke izraze u interpretaciji zavisnosti u formi prirodnog jezika. Eksperti i projektanti informacionih sistema predlažu i definišu način i modele, kao što je ovaj primer, korišćenja fazi priloga. Razmatrajući da neki od priloga mogu da imaju isto ili slično značenje, mi koristimo izraze iz definisisanog modela.



Slika 16. Model upotrebe fazi priloga

Šta donosi metrika? Procenat n-torki u kojima postoji posmatrana zavisnost ukazuje nam na intenzitet i bogatstvo zavisnosti u datoj fazi relaciji. Što je vrednost parametra P veća, fazi funkcionalna zavisnost je sadržana u većem broju n-torki. U tom kontekstu, predložena metodologija donosi nove aspekte analize fazi funkcionalnih zavisnosti. U sistemima za podršku odlučivanju to znači da veći prioritet u donošenju odluka imaju one zavisnosti, tj. pravila IF-THEN tipa, koja imaju veću vrednost parametra P. To podrazumeva veću proporciju parova n-torki koji zadovoljavaju opisane uslove. A to nam dalje ukazuje na činjenicu da takve zavisnosti možemo da primenjujemo u proceni nepoznatih (NULL) vrednosti atributa i da sa određenom verovatnoćom/stepenom istine pretpostavljamo da je nepoznata vrednost atributa upravo ona koju smo dobili primenom ove metodologije. Što je vrednost parametra P manja, tada zaključujemo da je fazi funkcionalna zavisnost sadržana samo u pojedinačnom ili manjem broju parova n-torki, pa je faktor sigurnosti procenjene nepoznate vrednosti znatno niži. Za donosioca odluke to je znak za dalju analizu i procenu prilikom donošenja odluka.

Naročito ističemo, u kontekstu predložene metodologije, da ona omogućava ekstrakciju svih fazi funkcionalnih zavisnosti. Dobijeni skup zavisnosti može kasnije da se redukuje ili proširi prema korisnikovim potrebama i zamislama, ali i da se koristi za estimaciju NULL vrednosti sa određenim faktorom sigurnosti i verovatnoće.

6.5. Poređenje primene različitih fazi implikacija

Sa praktične strane gledišta, mi primećujemo da u identifikovanim fazi funkcionalnim zavisnostima postoji dvoznačnost kada primenjujemo različite fazi implikacije. U ovom poglavlju analiziramo upotrebu različitih operatora fazi implikacije kako bi se u konkretnom primeru mogao izabrati najpogodniji. U prethodnim poglavljima prikazane su tabele sa fazi funkcionalnim zavisnostima koje su dobijene primenom Mamdani-ijeve i Kleene-Diense fazi implikacije. Ako uporedimo rezultate dobijene primenom fazi implikacija koji zadovoljavaju kriterijume prihvatanja u analizi opisane u petom poglavlju, možemo da zaključimo sledeće:

- Kleene-Diense fazi implikacija daje nam veće vrednosti jačine zavisnosti θ u poređenju sa Mamdani-ijevom, Wilmot-ovom i Early-Zade fazi implikacijom (Tabela 62.).
- Mamdani, Willmot i Early-Zade fazi operatori daju nam iste rezultate i vrednosti jačine zavisnosti θ , s tim da ta vrednost nije uvek manja vrednosti koja je dobijena primenom Kleene-Diense fazi implikacije.
- Proporcija/procentat n-torki u kojima postoje fazi funkcionalne zavisnosti je nezavisna od izbora odgovarajućeg operatora fazi implikacije.
- Dobijene vrednosti jačine zavisnosti θ su zavisne od usaglašenosti n-torki na posmatranim atributima. U ovoj doktorskoj disertaciji prikazali smo dve različite fazi relacije i dve formule za računanje saglasnosti n-torki na atributima datih fazi modela.

U sledećoj tabeli prikazane su vrednosti jačine zavisnosti θ za različite fazi implikacije koje smo koristili u identifikovanju fazi funkcionalnih zavisnosti u ovoj doktorskoj disertaciji.

θ	I_{MM}	I_{KL}	I_{EA}	I_W
$\theta \geq \theta_k$	θ_1	θ_2	θ_1	θ_1

Tabela 62. Vrednosti jačine zavisnosti θ dobijene primenom različitih fazi implikacija

Napomenimo da je vrednost $\theta_2 \geq \theta_1$, a što smo uočili i konstatovali na primeru fazi funkcionalnih zavisnosti koje smo identifikovali primenom predložene metodologije na fazi relacijama u ovom radu.

6.6. Praktična primena metodologije

Postoji veliki broj situacija i scenarija u kome ovaj koncept može biti jako koristan. Neki primeri su:

- Identifikovane fazi funkcionalne zavisnosti reprezentovane su sa fazi implikacijama koje nam omogućavaju ekstrakciju relevantnih i prethodno nepoznatih informacija neophodnih za donošenje upravljačkih i poslovnih odluka. Menadžeri u savremenim korporacijama sve više se oslanjaju na podatke iz baza podataka kako bi došli do novih saznanja i ažuriranih informacija. Na taj način određeni fragment fazi logike omogućava nam definisanje novih tehnika rudarenja podataka odnosno istraživanje i analizu velikih količina podataka s ciljem otkrivanja značajnih paterna i pravila. To je praktično samo deo pronalaženja znanja u bazama podataka, ali sa pravom možemo reći i ključni deo.
- Istraživanje podataka ima sve veći značaj u procesu upravljanja zbog neograničenih mogućnosti primene, uključujući prodaju i marketing, službu podrške kupcima, razvoj baze znanja itd. Izgradnja prediktivnih modela ponašanja potrošača danas je jako zastupljena tema u velikom broju kompanija koje su tržišno orjentisane. Prikupljeni podaci se modeliraju, vrši se analiza, targetiranje, izveštavanje i identifikacija potencijalno novih područija za povećanje profita. U takvim okolnostima, ova metodologija ima veliki značaj. No, pretraga znanja je zastupljena i u drugim modelima baza podataka.

- Predložena metodologija može da se koristi i u pronalaženju funkcionalnih zavisnosti u crisp relacijama, tj. klasičnim relacionim bazama podataka. Prilikom kreiranja logičkog modela nove baze podataka, pronalaženje crisp funkcionalnih zavisnosti vodi direktno u redukciju broja atributa za koje treba prikupljati podatke i zapamtiti u bazi podataka. Na primer, ako je jedan atribut u potpunosti zavistan od nekoliko atributa, onda se on može izračunati. Podskup atributa X funkcionalno određuje atribut Y ($X \rightarrow Y$) ako je svaki X asociran sa tačno jednim Y . Na primer atributi teritorija i gustina stanovnika. Kod fazi funkcionalnih zavisnosti između atributa, pitanje je koliko jako treba da su atributi zavisni da se za njih ne prikupljali podaci. Odnosno, kada prikupljanje podataka za zavisne attribute ima, a kada nema smisla. Fazi funkcionalne zavisnosti treba pažljivije koristiti kod redukcije atributa u tabelama. Kod postojećih baza podataka cilj je redukcija broja entitetovih atributa za buduća ažuriranja manjeg broja atributa. Ovo je kombinacija modifikacije logičkog modela i ekstrakcije informacija iz baza podataka.
- Fazi funkcionalne zavisnosti imaju široku primenu kod ekstrakcije informacija i pravila iz podataka. Na primer da li većina entiteta iz baze podataka sa osobinom A ima osobinu B ? Gde su A i B izrazi prirodnog jezika tipa “visoka zarada”, “malo preduzeće” itd. Tema je interesantna s pogleda mogućih primena i u oficijalnoj statistici: intrastat, popisi stanovništva, registri, generisanju fazi upita...
- Navise, pronalaženjem fazi funkcionalnih zavisnosti se može redukovati ne samo broj atributa, već i broj entiteta se može redukovati ili se mogu samo raspoznati veoma slični entiteti i na taj način se smanjuje redundantnost.
- U ovoj doktorskoj disertaciji mi smo se fokusirali na fazi relacione baze podataka, jer su praktičnije za analizu. Razlog je jednostavan. Pretpostavimo da je kardinalnost domena atributa Zarada velika u poređenju sa brojem entiteta, tj. $\text{dom}(\text{Zarada}) \subseteq \mathbb{R}$. Da bismo utvrdili postojanje fazi funkcionalnih zavisnosti koristimo relaciju bliskosti između vrednosti atributa. Mnogo jednostavniji i efikasniji način za analizu relacija između atributa posmatrane baze je da umesto crisp podataka poredimo međusobno lingvističke izraze kao što su *mala, srednja, visoka, veoma visoka* zarada. U tom slučaju domen atributa Zarada je skup četiri vrednosti $\text{dom}(\text{Zarada}) = \{\text{mala, srednja, visoka, veoma visoka}\}$. Poređenjem lingvističkih izraza ili računanjem semantičke distance između dva fazi broja značajno pojednostavljujemo proceduru predstavljenu u ovom radu. Dakle, daje se prednost i ukazuje na važnost fazifikacije podataka. U literaturi postoji različiti načini i modeli fazifikacije klasičnih baza podataka. Ove činjenice nas navode na zaključak da su fazi relacione baze podataka efikasnije za dobijanje informacija neophodnih za donošenje upravljačkih odluka. Cilj nije dobiti što više informacija iz ogromne količine podataka skladištnih u relacionim bazama, već bolje i kvalitetnije informacije.
- U pogledu primene fazi funkcionalnih zavisnosti značajno mesto zauzima i analiza i procena nepoznatih odnosno NULL vrednosti. To su one vrednosti koje postoje, a koje ne znamo. Fazi funkcionalne zavisnosti posmatramo kao rezultat istraživanja, a koji može da ima praktičnu primenu. Algoritam predstavljen u ovoj doktorskoj disertaciji korišćemo u budućim istraživanjima za procenu nepoznatih vrednosti atributa koji se odnose opštine u jednom regionu u Republici Slovačkoj (npr. broj snežnih dana u godini za pojedine opštine). Na osnovu poznatog skupa fazi funkcionalnih zavisnosti možemo da generišemo sistem fazi pravila koja sa određenom verovatnoćom manifestuju nepoznatu vrednost atributa. Dakle, fazi

pravila generišemo direktno iz relacione baze podataka koristeći otkrivene fazi funkcionalne zavisnosti.

- Poseban značaj metodologije predstavljene u ovom radu ogleda se u mogućnosti logičkog projektovanja fazi relacionih baza podataka. Kao i u slučaju klasičnih relacionih baza podataka problem redundancije i anomalija je prisutan i u fazi modelima ukoliko nisu adekvatno dizajnirani. Chen, Kerre i Vandenbulke su u svojim radovima predstavili teorijske smernice za logičko projektovanje fazi relacija na osnovu poznatog skupa fazi funkcionalnih zavisnosti. Zbog toga algoritam pomoću kojeg dolazimo do datog skupa je veoma važan korak u pravilnom dizajniranju. U radovima pomenutih autora nije prikazano kako možemo doći do skupa fazi funkcionalnih zavisnosti koji u stvari predstavlja input za radni okvir predstavljen u njihovom radu. Zbog toga ćemo na sledećem primeru prikazati kako u našem slučaju fazi relacije mogu da se dekomponuju pomoću skupa zavisnosti do kojih smo došli primenom algoritma predstavljenog u ovoj doktorskoj disertaciji.

6.7. Projektovanje fazi relacionih baza podataka

U četvrtom poglavlju predstavili smo teorijske smernice za normalizaciju fazi relacionih baza podataka. Istakli smo da fazi funkcionalne zavisnosti imaju veoma važnu ulogu u projektovanju istih. Slično kao i kod klasičnih relacija, problem anomalija i redundancije prisutan je i kod fazi relacija kada nisu pravilno dizajnirane. Chen, Kerre i Vandenbulcke [12] proširuju teoriju normalizacije prilagođavajući je fazi modelima. Zapravo, oni formulišu q-fazi normalne forme koje omogućavaju dekompoziciju prvobitnih relacija/tabela uz očuvanje postojanih fazi funkcionalnih zavisnosti. Dekomponovane tabele moguće je ponovo spojati, a da pri tome ne dođe do gubitka informacija. Napomenimo i da se problem anomalija i redundantnosti podataka smatra manje ozbiljnim u fazi relacijama pre svega zbog raspoloživosti i raznovrsnosti vrednosti atributa.

Ovaj radni okvir je razvijen za fazi modele na bazi mogućnosti distribucije i fazi modele na bazi relacije bliskosti, koji su i analizirani u ovoj doktorskoj disertaciji. Stoga ćemo da analiziramo projektovanje fazi relacija prikazanih u tabelama 35. i 42. U analizi ćemo koristiti skupove fazi funkcionalnih zavisnosti koji su dobijeni primenom Mamdani-ijeve aplikacije. Primenjujući pravila izvođenja i Armstrongove aksiome zavisnosti, vršićemo komprimovanje početnog skupa kako bismo dobili minimalne skupove odnosno skupove fazi funkcionalnih zavisnosti sa najmanjom pokrivenošću.

6.7.1. Prva fazi normalna forma (F1NF)

S obzirom da su vrednosti atributa u fazi relacijama prikazanim u tabelama 35. i 42. jednoznačne odnosno u slučaju fazi modela na bazi mogućnosti distribucije vrednosti atributa predstavljaju isključujuću mogućnost distribucije, konstatujemo da se obe fazi relacije nalaze u prvoj fazi normalnoj formi jer zadovoljavaju kriterijume za F1NF. Kao vrednosti atributa u datim fazi relacijama pojavljuju se podskupovi osnovnog skupa – domena, lingvističke varijable i mogućnost distribucije i sve one zadovoljavaju uslove prve fazi normalne forme. Sa druge strane, u slučaju postojanja višeznačnih vrednosti, mogućnost distribucije nije isključujuća jer su vrednosti/elementi fazi skupa povezani operatorom AND. U tom slučaju, da bismo dobili fazi model koji zadovoljava kriterijume prve fazi normalne forme, mogućnost distribucije $\pi_A = \{a_1/x_1, a_2/x_2, \dots, a_n/x_n\}$ mora da bude transformisana u veći broj isključujućih, i to: $\pi_{A_i} = \{a_i/x_i\}$, $i=1,2,\dots,n$.

6.7.2. q-Druge fazi normalna forma (q-F2NF)

Da bi fazi relacija bila u q-drugoj fazi normalnoj formi, mora da bude u F1NF i da svi atributi neključni potpuno zavise od q-ključa fazi relacije. U našem primeru, prvi preduslov je ispunjen, tj. obe fazi relacije su u F1NF. Ispitaćemo sada da li neključni atributi potpuno ili parcijalno zavise od q-ključa relacije.

Posmatrajmo prvo fazi model na bazi relacije bliskosti i skup fazi funkcionalnih zavisnosti F koji smo dobili primenom algoritma osmišljenog u ovoj doktorskoj disertaciji (Tabela 52.). {Ime_Prezime, Jezik} je 0.7-ključ ove fazi relacije jer važi:

$$\text{Ime_Prezime, Jezik} \xrightarrow{1} \text{Ime_Prezime,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{0.7} \text{Godine,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{0.7} \text{Iskustvo,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{0.7} \text{Produktivnost,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{1} \text{Jezik,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{0.7} \text{Pozicija,}$$

$$\text{Ime_Prezime, Jezik} \xrightarrow{0.7} \text{Zarada.}$$

Međutim, ova fazi relacija nije u 0.7-F2NF jer atributi Godine, Iskustvo, Produktivnost, Pozicija i Zarada ne zavise potpuno od složenog ključa relacije {Ime_Prezime, Jezik} već parcijalno odnosno delimično. Naime, zbog postojanja fazi funkcionalnih zavisnosti {Ime_Prezime^{0.7} → Godine, Ime_Prezime^{0.7} → Iskustvo,

Ime_Prezime^{0.7} → Produktivnost, Ime_Prezime^{0.7} → Pozicija, Ime_Prezime^{0.7} → Zarada} u skupu

F, zaključujemo da dati atributi potpuno zavise samo od dela ključa, tj. od primarnog ključa Ime_Prezime, ali ne i od 0.7-ključa {Ime_Prezime, Jezik}. Prema tome, nisu ispunjeni uslovi da konstatujemo da je data fazi relacija u 0.7-drugoj fazi normalnoj formi.

Analizirajmo sada fazi model na bazi mogućnosti distribucije i identifikovani skup fazi funkcionalnih zavisnosti koji je prikazan u tabeli 58. U ovom primeru Ime_Prezime je 0.7-ključ relacije. Kako svi ostali atributi, tj. atributi koji nisu ključni, potpuno zavise od 0.7-ključa Ime_Prezime, $q = \min(0.7, 0.7, 0.7, 0.7, 0.7, 0.7) = 0.70$:

$$\text{Ime_Prezime} \xrightarrow{0.7} \text{Godine,}$$

$$\text{Ime_Prezime} \xrightarrow{0.7} \text{Iskustvo,}$$

$\text{Ime_Prezime} \xrightarrow{0.7} \text{Produktivnost},$

$\text{Ime_Prezime} \xrightarrow{0.7} \text{Jezik},$

$\text{Ime_Prezime} \xrightarrow{0.7} \text{Pozicija},$

$\text{Ime_Prezime} \xrightarrow{0.7} \text{Zarada},$

zaključujemo da je ova fazi relacija R u 0.7-F2NF.

Dakle, fazi model na bazi relacije bliskosti nije u 0.7-drugoj fazi normalnoj formi, dok to ne važi za fazi model na bazi mogućnosti distribucije. Napomenimo ovde da q-druga fazi normalna forma q-F2NF nema veći praktični značaj za projektovanje fazi relacionih baza podataka, za razliku od q-treće fazi normalne forme q-F3NF. Zbog toga nije od presudne važnosti da li neka fazi relacija zadovoljava kriterijume q-F2NF. U definiciji za q-F3NF vidimo da je dovoljan i potreban uslov da je fazi relacija u q-F3NF da je u F1NF i da ispunjava kriterijume treće fazi normalne forme. Dakle, q-druga fazi normalna forma ne rešava problem anomalija i redundancije u fazi relacijama. Kao i kod klasičnih relacionih baza podataka i u slučaju fazi relacionih modela važi:

$$q\text{-F3NF} \subset q\text{-F2NF} \subset \text{F1NF}.$$

6.7.3. q-Treća fazi normalna forma (q-F3NF)

Za dovođenje fazi relacija u q-treću fazi normalnu formu koristićemo teoremu, pravila i definicije koje su prikazane u četvrtom poglavlju ove doktorske disertacije. Ova fazi normalna forma ima najveći praktični značaj kada je reč o projektovanju fazi relacionih baza podataka. Stoga ćemo u analizi koristiti skupove fazi funkcionalnih zavisnosti koje smo dobili analizom fazi modela na bazi mogućnosti distribucije i fazi modela na bazi relacije bliskosti za slučajeve $\theta_k=0.70$ i $\theta_k=0.80$. Napomenimo da algoritam prikazan u 4.3. poglavlju omogućava dekompoziciju fazi relacija u q-F3NF uz očuvanje postojećih fazi funkcionalnih zavisnosti i spajanje bez gubitka informacija. Da bismo analizu učinili jednostavnijom i preglednijom atribut fazi relacija (Ime_Prezime, Godine, Iskustvo, Produktivnost, Jezik, Pozicija, Zarada) označićemo slovnim oznakama A, B, C, D, E, F,G respektivno.

Posmatrajmo prvo fazi model na bazi relacije bliskosti i skup fazi funkcionalnih zavisnosti F koje smo dobili za slučaj $\theta_k=0.70$. {Ime_Prezime, Jezik} je 0.7-ključ. Preduslov da je relacija u F1NF je ispunjen. Međutim, fazi model na bazi relacije bliskosti nije u q-F3NF. Kako {Ime_Prezime, Jezik} parcijalno određuje ostale attribute fazi relacije i atribut Jezik se ne nalazi ni u jednoj od fazi funkcionalnih zavisnosti u identifikovanom skupu, izvršićemo dekompoziciju na dve relacije R_1 i R_2 :

$$\rho=\{R_1,R_2\}$$

pri čemu je $R_1=\Pi_{ABCDEFG}(F)$, a $R_2=\Pi_{AE}(F)$ koje su u 0.7-F3NF i 1-F3NF.

U drugom primeru, fazi modelu na bazi mogućnosti distribucije koji je u F1NF, 0.7-ključ Ime_Prezime potpuno određuje sve ostale attribute fazi relacije, pa kako fazi

funkcionalna zavisnost $A \xrightarrow{0.7} B, C, D, E, F, G$ uključuje sve attribute fazi relacije, onda prema pravilu za dekompoziciju relacije u q-F3NF, zaključujemo da je:

$$\rho = \{R\},$$

tj. fazi relacija na bazi mogućnosti distribucije je u 0.7-F3NF. Ovo je scenario koji važi za $\theta_k = 0.70$

Pređimo sada na slučaj relacione šeme sa skupom fazi funkcionalnih zavisnosti dobijenih za $\theta_k = 0.80$. U prvom primeru, u tabeli 55., prikazan je skup fazi funkcionalnih zavisnosti F za fazi model na bazi relacije bliskosti. Kao što smo naveli, posmatraćemo skup dobijen primenom Mamdani-ijeve aplikacije. Da bismo mogli da izvršimo pravilnu dekompoziciju na osnovu poznatih fazi funkcionalnih zavisnosti neophodan nam je minimalni skup F tj. skup F sa najmanjom pokrivenošću fazi funkcionalnih zavisnosti. U konkretnom primeru F možemo da predstavimo kao:

$$F = \{C \xrightarrow{0.80} D, D \xrightarrow{0.80} F, G \xrightarrow{0.80} C\}.$$

odnosno:

$$\text{Iskustvo, Produktivnost, Pozicija, Zarada} \xrightarrow{0.80} \text{Godine},$$

$$\text{Produktivnost} \xrightarrow{0.80} \text{Pozicija, Zarada},$$

$$\text{Zarada} \xrightarrow{0.80} \text{Iskustvo}.$$

Dati skup F je parcijalno slobodan. Atributi Ime_Prezime i Jezik se ne pojavljuju ni u jednoj fazi funkcionalnoj zavisnosti, pa ih prema utvrđenim pravilima eliminišemo iz relacije R i kreiramo relacionu šemu R'. {Produktivnost, Zarada} je 0.8-ključ relacione šeme R. Iz svega navedenog zaključujemo da relacionu šemu R možemo da dekomponujemo kao:

$$\rho = R' \cup \{R_1, R_2, R_3\},$$

gde je:

$$R'(A, E) = \Pi_{AE},$$

$$R_1 = \Pi_{CDEFG}(F),$$

$$R_2 = \Pi_{BFG}(F),$$

$$R_3 = \Pi_{GC}(F).$$

Relacije R_1, R_2, R_3 su u trećoj fazi normalnoj formi sa stepenima 0.8, 0.8 i 0.8 respektivno. Dakle, problem anomalija i redundanse u relacionoj šemi R zasnovanoj na relaciji bliskosti možemo rešiti tako što izvršimo dekompoziciju u odnosu na minimalni skup fazi funkcionalnih zavisnosti: $\Pi_{AE}, \Pi_{CDEFG}(F), \Pi_{BFG}(F), \Pi_{GC}(F)$.

Kako je {Produktivnost, Zarada} 0.8-ključ. Prema teoremi 2. iz poglavlja 4.3. imamo da je:

$$\sigma = \rho \cup \{D, G\} = \rho \cup \{\text{Produktivnost, Zarada}\}.$$

a da je pri tome {Produktivnost,Zarada} u 1-F3NF. Pored toga, i σ omogućava spajanje bez gubitka informacija.

Osvrnimo se sada na fazi model na bazi mogućnosti distribucije i skup fazi funkcionalnih zavisnosti F koji smo dobili za slučaj $\theta_k=0.80$. U tabeli 61. prikazan je skup F, gde vidimo da je u slučaju primene Mamdan-ijeve aplikacije:

$$F=\{G \xrightarrow{0.81} EF\},$$

odnosno:

$$F=\{Zarada \xrightarrow{0.81} Jezik,Pozicija\}.$$

Kako u navedenom primeru atributi Ime_Prezime, Godine, Iskustvo i Produktivnost nisu obuhvaćeni fazi funkcionalnim zavisnostima u skupu F, kreiraćemo relaciju šemu R' (Ime_Prezime,Godine, Iskustvo, Produktivnost). S obzirom da je skup F parcijalno slobodan, prema algoritmu za dekompoziciju imamo:

$$\rho = R' \cup \{R_1\},$$

gde važi da je:

$$R'(A,B,C,D) = \Pi_{ABCD} i$$

$$R_1 = \Pi_{GEF}(F).$$

Zarada je 0.81-ključ fazi relacije R, koju smo dobili nakon eliminisanja atributa koji se ne pojavljuju u skupu fazi funkcionalnih zavisnosti F. Dodajmo ovde i to da se za proveru da li je neki skup fazi funkcionalnih zavisnosti parcijalno slobodan ili ne koristi algoritam koji proizilazi iz same definicije.

6.8. Efikasnost algoritma

Tragajući za fazi funkcionalnim zavisnostima u fazi relacijama primenom predložene metodologije, neophodno je sa praktične strane gledišta uraditi i analizu efikasnosti. Mi smo predložili ovaj algoritam zbog njegove fleksibilnosti i intuitivnosti, a u ovom poglavlju razmatraćemo i njegovu efikasnost.

Naša definicija kreće od principa usaglašenosti na paru n-torki. Da bismo došli do skupa fazi funkcionalnih zavisnosti u fazi relaciji neophodno je primeniti odgovarajuća pravila i transformisati originalnu relaciju u oblik pogodan za ekstrakciju odgovarajućih zavisnosti.

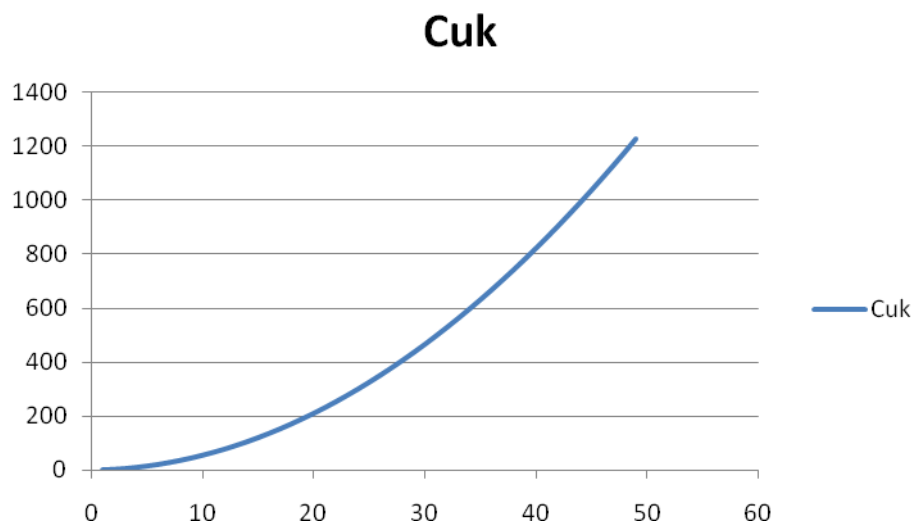
Pretpostavimo da je n broj n-torki u originalnoj fazi relaciji, a m ukupan broj atributa. U našem slučaju, prema modelu koji smo razvili, kreiramo parove n-torki da bismo dobili relaciju T_r i za svaki par n-torki računamo usaglašenost $C(A_k[t_i,t_j])$, $k=1,\dots,m$, $i,j=1,\dots,n$ i $i \neq j$ na posmatranom atributu. Dakle, mi razmatramo $\binom{n}{2}$ parova n-torki na svakom od atributa. U najgorem slučaju, ukupan broj parova S_{uk} sa odgovarajućim vrednostima je $m^* \binom{n}{2}$, tj.:

$$S_{uk} = \frac{m * n!}{2!(n - 2)!}$$

Osnovni nedostatak koji se pojavljuje je ukupan broj transakcija (mogućih parova n-torki sa povezanim vrednostima) koje je neophodno izračunati, a što može biti jako skupo za velike baze podataka. Povećanjem broja n-torki, kao i broja atributa relacione šeme, broj transakcija raste eksponencijalno, stim da S_{uk} ima brži rast. Na grafiku 1. prikazan je broj transakcija u zavisnosti od broja entiteta relacione šeme, a na grafiku 2. ukupan broj transakcija u zavisnosti od broja atributa m iste relacije.

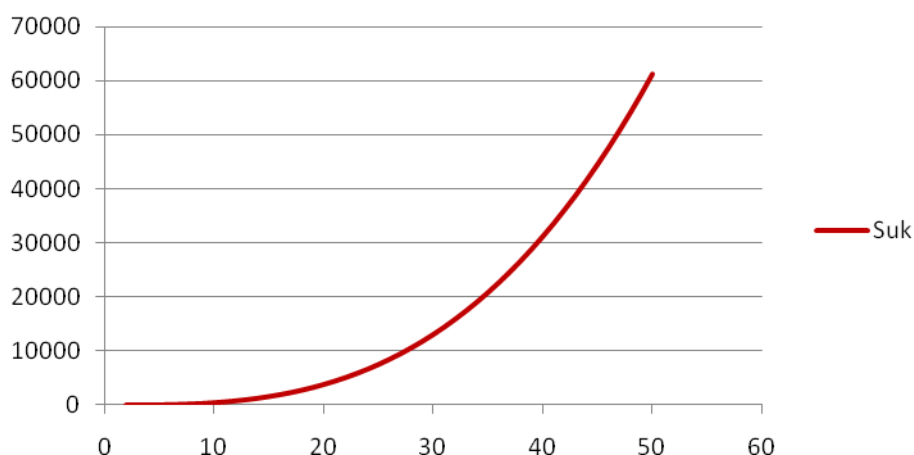
n	C_{uk}	m	S_{uk}
2	1	2	2
5	10	5	50
10	45	10	450
15	105	15	1575
20	190	20	3800
25	300	25	7500
30	435	30	13050
35	595	35	20825
40	780	40	31200
45	990	45	44550
50	1225	50	61250

Tabela 63. Broj transakcija algoritma



Grafik 1. Ukupan broj parova n-torki za koje računamo usaglašenost na posmatranom atributu

Suk



Grafik 2. Ukupan broj transakcija u relaciji T_r

Međutim, u radu predlažemo i pokazujemo da je moguće povećati efikasnost algoritma i smanjiti broj transakcija za koje računamo $C(A_k[t_i, t_j])$:

- Primenom projekcije i restrikcije nad originalnom bazom smanjujemo broj atributa odnosno n-torki ($p \leq n$, $r \leq m$), iz čega sledi da je:

$$E(p,r) \leq E(n,m).$$

Da bismo mogli da redukujemo ukupan broj transakcija neophodna su nam ekspertska znanja i iskustvo. Na taj način moguće je predvideti najznačajnije delove relacije čijom analizom možemo doći do skupa fazi funkcionalnih zavisnosti. To nam omogućava značajno unapređenje efikasnosti algoritma.

- Sa druge strane, savremeni računarski sistemi, sa moćnim procesorima i memorijskim čipovima novije generacije, pružaju nam mogućnosti obrade velike količine podataka, što može da bude dodatna pogodnost kada je reč o relacijama sa velikim brojem zapisa. Virtuelizacija, novi hard diskovi sa SSD tehnologijom čitanja i upisa podataka i sve veći broj jezgara u procesorima značajno smanjuje vreme izvršenja predloženog algoritma. Ovo predstavlja praktično posmatranje. Majkrosoft SQL Server 2008 pruža mogućnost generisanja različitih statističkih izveštaja. Na primer, SELECT upit nad jedim atribtom u relaciji sa 50.000 zapisa se izvrši za 30ms. Veličina baze koju smo koristili je 300 MB i relaciona šema ima ukupno 19 atributa. SQL server je instaliran na virtuelnoj mašini sa 4vCPUs i 4.84 GB RAM memorije. Baza podataka koju smo koristili za aproksimaciju se skladišti na 4x600GB hard diskovima u RAID5 strukturi. Koristeći ovaj podatak, mi možemo aproksimativno izračunati vreme izvršenja našeg algoritma. U relaciji sa 5 atributa i 50.000 rekorda aproksimativno vreme generisanja funkcionalnih zavisnosti je 625s. Sa teorijske tačke gledišta, napredne računarske tehnologije ne mogu da rešavaju praktične probleme eksponencijelne prirode u prihvatljivom vremenskom intervalu. Međutim, želimo naglasiti da priroda posmatranog problema određuje efikasnost algoritma. Postoje baze podataka u kojima kompresija nije neophodna, dok u drugim slučajevima, kada govorimo o ogromnim

bazama podataka, različite tehnike za kompresiju i redukciju količine zapisa se koriste, kao što su fazifikacija, klasterizacija, granulacija, predstavljanje sličnih rekorda sa jednim reprezentom itd.

Međutim, da bismo identifikovali fazi funkcionalne zavisnosti, mi moramo da primenimo fazi implikacije na način opisan u ovoj doktorskoj disertaciji. Prema utvrđenoj metodologiji, relacije između atributa se utvrđuju primenom fazi logike. Ako je m broj atributa posmatrane fazi baze podataka, onda je ukupan broj mogućih fazi funkcionalnih zavisnosti $2^* \binom{m}{2}$. Teoretski, ovo je najveći mogući broj u skupu fazi funkcionalnih zavisnosti. Ako primenjujemo različite fazi implikacije, onda imamo više iteracija i u svakoj od iteracija $2^* \binom{m}{2}$ mogućih fazi funkcionalnih zavisnosti. Ako tome dodamo da je

po definiciji fazi funkcionalna zavisnost $A_1 \xrightarrow{\theta} A_2$ prisutna ako i samo ako je $C(A_2[t_i, t_j]) \geq \min(\theta, C(A_1[t_i, t_j]))$, onda dolazimo do činjenice da upoređujemo međusobne vrednosti usaglašenosti n -torki na svakom paru atributa, pa je ukupan broj kalkulacija $C_{uk}^* \binom{m}{2}$, gde je C_{uk} broj parova n -torki fazi relacije. Međutim, ovaj broj je u praksi značajno manji zbog postojanja logičkih uslova. Naime sve vrednosti $C(A_k[t_i, t_j]) < \theta_k$, gde je θ_k kritična jačina zavisnosti, se ne uzimaju u razmatranje, pa je broj kalkulacija mnogo manji. U našem primeru, za $\theta_k = 0.80$ u fazi modelu na bazi mogućnosti distribucije, od mogućih $10^* \binom{7}{2} = 210$ ispitivanja ispunjenosti uslova postojanja fazi funkcionalnih zavisnosti, razmatrali smo svega dve situacije i utvrdili postojanje dve fazi funkcionalne zavisnosti. Napomenimo i to da je primer i eksperiment vršen nad realnim podacima.

6.9. Zaključak

U ovom poglavlju dokazana je metodologija razvijena za potrebe pronalaženja fazi funkcionalnih zavisnosti u fazi relacijama. U tom cilju, najpre smo uradili analizu u fazi modelu na bazi relacije bliskosti i korišćenjem Mamdani-ijeve i Kleene-Diense fazi implikacije ustanovili skup fazi funkcionalnih zavisnosti. Isti metod rada primenili smo i u slučaju fazi relacije na bazi mogućnosti distribucije. Zatim smo ukazali na mogućnosti primene rezultata rada i dokazali njihovo korišćenje u slučaju logičkog dizajniranja fazi modela baza podataka i konverziju zavisnosti u lingvističke forme koje su intuitivnije sa stanovišta krajnjeg korisnika informacija. Na kraju uradili smo i analizu efikasnosti kako bismo uključili i ispitivanje nedostatka u predloženom algoritmu.

7. Zaključak

U oblasti fazi logike postoje mehanizmi koji nalaze primenu u definisanju i pronalaženju fazi funkcionalnih zavisnosti u bazama podataka. Fazi relacione baze podataka na prirodan način proširuju funkciju klasičnih relacija integrišući nepreciznosti, neodređenosti i nejednoznačnosti koje se ne mogu zanemariti u realnim aplikacijama i sistemima. Fazi funkcionalne zavisnosti predstavljaju veći nivo adaptacije fazi bazi znanja u njihovoj svakodnevnoj upotrebi. U savremenim korporacijama sve veći broj ljudi, od običnih inženjera i menadžera, pa sve do eksperata su zavisni od informacija iz baza podataka, koje su im neophodne za obavljanje svakodnevnih zadataka. Ove činjenice dovode nas do shvatanja značaja i uloge fazi funkcionalnih zavisnosti, koje nam omogućavaju da pristupimo informacijama iz baze na jedan fleksibilniji i pristupačniji način. Cilj nije da dobijemo što više informacija, već korisnija i efikasnija znanja, u smislu posjedovanja filtriranih informacija umesto ogromne količine podataka koje su skladištene u posmatranim relacijama. Zbog toga se menadžeri prilikom donošenja upravljačkih odluka oslanjaju na korišćenje saznanja dobijenih analizom fazi funkcionalnih zavisnosti.

U disertaciji je dat pregled više okvira za definisanje fazi funkcionalnih zavisnosti kao i odgovarajuća pravila izvođenja, zasnovana na Armstrongovim aksiomama, a koja nam omogućavaju da iz datog skupa zavisnosti logički dobijemo i neke druge fazi funkcionalne zavisnosti. Međutim, problem kod fazi relacionih baza podataka jeste činjenica da nije postojao algoritam ili procedura pronalaženja fazi funkcionalnih zavisnosti. U ovom doktorskom radu razvijena je originalna metodologija koja omogućava da analizom zapisa dođemo do zavisnosti između atributa u fazi relacionim bazama podataka. Problem pronalaženja fazi funkcionalnih zavisnosti rešen je na jedan potpuno nov način, primenom relacije bliskosti i odgovarajućeg fragmenta fazi logike. Važna tema u ovom istraživanju bila je karakterizacija fazi implikacija (Mamdani, Lukasiewicz, Early-Zade itd.) i definisanje pravila za predstavljanje zavisnosti u formi pogodnoj za ljudsku percepciju. Dakle, u ovoj doktorskoj disertaciji adresiran je problem i predložen je algoritam za njegovo rešavanje, a čiji fokus je reprezentovanje odnosa između atributa.

Teorijsko razmatranje i eksperiment nad realnim podacima pokazali su da važi polazna hipoteza doktorske disertacije:

- Teorija fazi skupova i određeni fragment fazi logike mogu da se primenjuju u analizi fazi funkcionalnih zavisnosti u fazi relacionim bazama podataka.

Sprovedenim istraživanjem pokazali smo da fazi funkcionalnim zavisnostima možemo da pridružimo odgovarajuće fazi implikacije kako bismo došli do relacija između atributa u posmatranoj bazi podataka.

Pored opšte i polazne hipoteze istraživanja u doktorskoj disertaciji dokazali smo da vrede i sledeće posebne hipoteze:

- Razvijena metodologija analize fazi funkcionalnih zavisnosti može da se koristi za logičko dizajniranje fazi relacionih baza podataka.
- Fazi implikacije su adekvatne za procenu jačine zavisnosti između atributa fazi relacione baze podataka.
- Teorija fazi skupova i određeni fragment fazi logike pružaju poboljšane mogućnosti u analizi funkcionalnih zavisnosti.

- Za otkrivanje fazi funkcionalnih zavisnosti mogu da se koriste različite fazi implikacije.
- Fazi funkcionalne zavisnosti povećavaju efikasnost statističkih baza podataka.

Dokazivanjem navedenih hipoteza ostvarili smo temeljni cilj istraživanja u ovoj doktorskoj disertaciji koji se odnosi na ispitivanje zavisnosti između atributa fazi relacionih baza podataka.

Generalno, naše istraživanje pokazuje kako da dobijemo inteligentne informacije iz fazi relacija. Da bismo došli do mogućih relacija između atributa fazi baze najpre je neohodno pripremiti podatke za analizu odnosno kreirati tabelu sa parovima n-torki i vrednostima usaglašenosti na svim atributima originalne relacije. U drugom koraku pronalazimo podrelacije u kojima su zadovoljeni kriterijumi za postojanje fazi funkcionalnih zavisnosti. U trećem koraku fazi funkcionalnim zavisnostima pridružujemo odgovarajuće fazi implikacije. Konačno, mi interpretiramo rezultate iz prethodnih koraka kako bismo dobili finalni skup fazi funkcionalnih zavisnosti.

Najvažniji doprinos ove doktorske disertacije predstavlja karakterizacija teorije fazi skupova i fazi implikacija u kontekstu izgradnje alata i algoritma koji omogućava identifikaciju odnosa između atributa u posmatranim relacionim šemama. Treba napomenuti da ovaj rad predstavlja empirijsku studiju i "prof-of-concept" i da nije razvijen komercijalni softver. No, krajnji rezultat istraživanja je prototip koji definiše sve faze i korake u projektovanju i programiranju softvera koji bi rešavao adresirani problem. Sa druge strane, mi verujemo da ideja prezentovana u ovom istraživanju obezbeđuje metodologiju koja može biti u značajnoj meri eksploatisana u budućim konkretnim istraživanjima.

U sažetku, doprinos i značaj ove doktorske disertacije ogleda se u praktičnim primenama predložene metodologije kao što su:

- Modelovanje i projektovanje fazi relacionih baza podataka,
- Podrška sistemima za podršku odlučivanju,
- Ekvivalencija između fazi implikacija i fazi funkcionalnih zavisnosti,
- Predstavljanje fazi funkcionalnih zavisnosti u lingvističkoj formi koja omogućava bolje razumevanje dobijenih rezultata,
- Izvođenje fazi funkcionalnih zavisnosti na nivou podrelacija,
- Povećanje efikasnosti i redukovanje opterećenja respondenata prilikom prikupljanja i slanja podataka u statističkim modelima baza podataka,
- Smanjenje redundantnosti otkrivanjem sličnih entiteta,
- Integracija različitih pristupa u definisanju fazi funkcionalnih zavisnosti,
- Izvođenje finalnog skupa fazi funkcionalnih zavisnosti,
- Metodologija može da se primenjuje u svim fazi modelima gde je moguće primeniti relaciju bliskosti ili mogućnost distribucije.

U toku rada identifikovali smo i pravce budućih istraživanja i neke otvorene probleme koji proizilaze iz ove teme:

- Primena opisane metodologije u proceni NULL vrednosti u relacionim modelima baza podataka. Prvi koraci u istraživanju ove tematike su napravljeni na primeru statističkih modela baza podataka. Ovo može da bude jako interesantna oblast za istraživanje zbog prednosti koja se može iskoristiti prilikom prikupljanja podataka. To je naročito važno za organizacije koje manipulišu sa velikim količinom podataka. Na taj način postupak procene i određivanja nepoznatih vrednosti atributa približujemo aproksimativnom rezonovanju, a to znači da estimacija IF-THEN pravilima leži unutar fazi logike.
- Proširenje i nadgradnja sistema za upravljanje bazama podataka. Postojeći sistemi za upravljanje bazama podataka nemaju implementirane mehanizme za integraciju i upravljanje fazi informacija. Model opisan u ovoj doktorskoj disertaciji može da bude prototip za eliminisanje ograničenja u sistemima za upravljanje bazama podataka poput SQL i Oracla. S obzirom na širinu sagledavanja problema prilagođavanje ne bi trebalo da ima dodatne poteškoće.
- Mogućnost definisanja i implementacije novih implikacija u fazi logici i njihova integracija u predloženu metodologiju.
- Predstavljanje novih tehnika i metoda u okviru rudarenje podataka-a. Rudarenje podataka koristi različite vrste alata za analizu podataka i otkrivanje zakonitosti i veza između istih koje se mogu dalje iskoristiti za izradu validnih projekcija. Rudarenje podataka nije pojedinačna tehnika ili tehnologija već je to skup srodnih metoda i metodologija koje su usmerene ka pronalaženju i automatskom izvlačenju pravila, šablona, paterna i drugih karakterističnih struktura iz podataka. Koncept zasnovan na fazi logici i teoriji fazi skupova svakako predstavlja novu metodologiju koja može aktivno da se koristi u različitim pretragama, a izbor alata zavisi od prirode zadatka. Jedna od tehnika rudarenje podataka-a, pored web mining-a i rudarenje podataka multimedije, u kojoj istraživanje predstavljeno u ovoj doktorskoj disertaciji može da bude interesantno jeste i text mining. Tekst mining predstavlja analizu teksta u cilju izvlačenja termina, entiteta i odnosa između tih termina i entiteta. To je deo šireg polja otkrivanja znanja koje, počinjući od grupe podataka relevantnih za rešenje specifičnog problema, traži zanimljive i prethodno nepoznate šablone. U okviru text mininga koriste se različite tehnike, kao što su izvlačenje termina, izvlačenje informacija i analiza veza. Pretpostavka je da analiza zavisnosti zasnovana na fazi implikacijama i relaciji bliskosti može da dovede do novih interpretacija i pronalaženja zanimljivih informacija u velikim skupovima tekstova, gde se pod zanimljivim informacijama podrazumevaju netrivialne, skrivene, prethodno nepoznate i potencijalno korisne.

8. Prilog - Programski kod za Algoritam napisan u C#

```
namespace Fuzzy
{
    public class FuzzyProcessor
    {
        public FuzzyProcessor()
        { }

        public static string Calculate(int exampleId, float undefinedCloseness, float
criticalDependence)
        {
            List<int> tupleIds = GetTupleIds(exampleId);
            List<int> attributeIds = GetAttributeIds(exampleId);
            var dt = GetAttributeValues(exampleId);
            return xxx(exampleId, tupleIds, attributeIds, dt, undefinedCloseness,
criticalDependence);
        }

        public static string Calculation1(int exampleId, List<int> tupleIds, List<int>
attributeIds, dsFuzzy.AttributeValuesDataTable dt, float undefinedCloseness, float
criticalDependence)
        {
            dsFuzzy.AttributeValueClosenessDataTable dtClosenes =
GetAttributeValueCloseness(exampleId);
            int d = tupleIds.Count;// *(tupleIds.Count - 1) / 2;
            float[, ] array = new float[d, d, attributeIds.Count];
            for (int i = 0; i < tupleIds.Count; i++)
            {
                for (int j = i+1; j < tupleIds.Count; j++)
                {
                    for (int a = 0; a < attributeIds.Count; a++)
                    {
                        float adI = dt.Where(x => x.TupleId == tupleIds[i] && x.AttributeId ==
attributeIds[a]).First().AffiliationDegree;
                        int adomainI = dt.Where(x => x.TupleId == tupleIds[i] && x.AttributeId
== attributeIds[a]).First().AttributeDomainId;
                        float adJ = dt.Where(x => x.TupleId == tupleIds[j] && x.AttributeId ==
attributeIds[a]).First().AffiliationDegree;
                        int adomainJ = dt.Where(x => x.TupleId == tupleIds[j] && x.AttributeId
== attributeIds[a]).First().AttributeDomainId;
```

```

        var closenessItem = dtClosenes.Where(x => x.AttributeId == attributeIds[a]
&& (x.AttributeValueCandidateA==adomainI &&
x.AttributeValueCandidateB==adomainJ
        || x.AttributeValueCandidateA == adomainJ &&
x.AttributeValueCandidateB == adomainI)).FirstOrDefault();

        float closeness = closenessItem != null ? (float)closenessItem.Closeness :
undefinedCloseness;

        array[i, j, a] = Math.Min(adI, adJ);
        array[i, j, a] = Math.Min(array[i, j, a], closeness);
        array[j, i, a] = array[i, j, a];
        //if (a==4)
        System.Diagnostics.Debug.WriteLine("i:" + i + " j:" + j + " a:" + a + " =>
" + array[i, j, a]);
    }
    // int x = Math.Min( tupleIds[i], attributeIds[j]);
    // System.Diagnostics.Debug.WriteLine(x);
}
}
return Calculation2(exampleId, tupleIds, attributeIds, array, criticalDependence);
}

private static string Calculation2(int exampleId, List<int> tupleIds, List<int>
attributeIds, float[, ,] array, float criticalDependence)
{
    StringBuilder sb = new StringBuilder();
    int dt = tupleIds.Count * (tupleIds.Count - 1) / 2;
    int da = attributeIds.Count * (attributeIds.Count - 1) / 2;
    float[, , ,] final = new float[dt, dt, da, da];
    for (int a = 0; a < attributeIds.Count; a++)
    {
        for (int b = a+1; b < attributeIds.Count; b++)
        {
            //if (a != b)
            {
                int percent = 0;
                float teta = 1;

```

```

    for (int i = 0; i < tupleIds.Count; i++)
    {
        for (int j = i + 1; j < tupleIds.Count; j++)
        {
            //System.Diagnostics.Debug.WriteLine(" a:" + a + " b:" + b + " i:" + i
+ " j:" + j);
            //System.Diagnostics.Debug.WriteLine(" aValue:" + array[i, j, a] + "
bValue:" + array[i, j, b]);
            // final[i, j, a, b] = 0;
            if (array[i, j, a] <= array[i, j, b])
            {
                if (array[i, j, a] >= criticalDependence)
                {
                    System.Diagnostics.Debug.WriteLine("i:" + i + " j:" + j + " *** "
+ GetAttributeName(attributeIds[a]) + " & " + GetAttributeName(attributeIds[b]) + "
=> a: " + array[i, j, a] + "   b: " + array[i, j, a]);
                    percent++;
                    if (teta > array[i, j, a])
                        teta = array[i, j, a];
                }
            }
        }
        //System.Diagnostics.Debug.WriteLine(" a:" +
GetAttributeName(attributeIds[a]) + " b:" + GetAttributeName(attributeIds[b]) + "
percent:" + percent/dt + " teta:" + teta);
        if (((float)percent / dt > 0) && teta >= 0.7f)
            sb.AppendLine(GetAttributeName(attributeIds[a]) + " =>" +
GetAttributeName(attributeIds[b]) + "   Percent:" + (float)percent / dt + "   Teta:" + teta);
    }
}
for (int b = 0; b < attributeIds.Count; b++)
{
    for (int a = b + 1; a < attributeIds.Count; a++)
    {
        //if (a != b)

```



```

    {
        int percent = 0;
        float teta = 1;
        for (int i = 0; i < tupleIds.Count; i++)
        {
            for (int j = i + 1; j < tupleIds.Count; j++)
            {
                if (array[i, j, a] <= array[i, j, b])
                {
                    if (array[i, j, a] >= criticalDependence)
                    {
                        percent++;
                        if (teta > array[i, j, a])
                            teta = array[i, j, a];
                    }
                }
            }
        }
        if (((float)percent / dt > 0) && teta >= 0.7f)
            sb.AppendLine(GetAttributeName(attributeIds[a]) + " =>" +
                GetAttributeName(attributeIds[b]) + "    Percent:" + (float)percent / dt + "    Teta:" + teta);
    }
}
return sb.ToString();
}
private static List<int> GetTupleIds(int exampleId)
{
    List<int> tupleIds = new List<int>();
    SqlConnection dataConnection = new SqlConnection();
    dataConnection.ConnectionString =
        ConfigurationManager.ConnectionStrings["Fuzzy.Properties.Settings.FuzzyConnectionString"].ConnectionString;
    SqlCommand dataCommand = new SqlCommand();
    dataCommand.Connection = dataConnection;
    dataCommand.CommandType = CommandType.Text;
}

```

```

        dataCommand.CommandText = ("select id from dbo.Tuple where exampleid = " +
exampleId);
    try
    {
        dataConnection.Open();
        SqlDataReader reader = dataCommand.ExecuteReader();

        while (reader.Read())
        {
            tupleIds.Add(reader.GetInt32(0));
        }
        reader.Close();
    }
    catch
    {
    }
    finally
    {
        dataConnection.Close();
    }
    return tupleIds;
}
private static List<int> GetAttributeIds(int exampleId)
{
    List<int> attributeIds = new List<int>();
    SqlConnection dataConnection = new SqlConnection();
    dataConnection.ConnectionString =
ConfigurationManager.ConnectionStrings["Fuzzy.Properties.Settings.FuzzyConnectionStri
ng"].ConnectionString;
    SqlCommand dataCommand = new SqlCommand();
    dataCommand.Connection = dataConnection;
    dataCommand.CommandType = CommandType.Text;
    dataCommand.CommandText = ("select id from dbo.Attribute where exampleid = "
+ exampleId);
    try
    {

```

```

        dataConnection.Open();
        SqlDataReader reader = dataCommand.ExecuteReader();
        while (reader.Read())
        {
            attributeIds.Add(reader.GetInt32(0));
        }
        reader.Close();
    }
    catch
    {
    }
    finally
    {
        dataConnection.Close();
    }
    return attributeIds;
}

private static string GetAttributeName(int attributeId)
{
    SqlConnection dataConnection = new SqlConnection();
    dataConnection.ConnectionString =
ConfigurationManager.ConnectionStrings["Fuzzy.Properties.Settings.FuzzyConnectionStri
ng"].ConnectionString;
    SqlCommand dataCommand = new SqlCommand();
    dataCommand.Connection = dataConnection;
    dataCommand.CommandType = CommandType.Text;
    dataCommand.CommandText = ("select AttributeName from dbo.Attribute where
id = " + attributeId);
    try
    {
        dataConnection.Open();
        object result = dataCommand.ExecuteScalar();
        //while (reader.Read())
        {
            return result.ToString();
        }
    }
}

```

```

    }
    catch
    {
    }
    finally
    {
        dataConnection.Close();
    }
    return "unknown!";
}
private static dsFuzzy.AttributeValuesDataTable GetAttributeValues(int exampleId)
{
    dsFuzzy.AttributeValuesDataTable dtAttributeValues = new
dsFuzzy.AttributeValuesDataTable();
    dsFuzzyTableAdapters.AttributeValuesTableAdapter adapter = new
dsFuzzyTableAdapters.AttributeValuesTableAdapter();
    dtAttributeValues = adapter.GetDataBy(exampleId);
    return dtAttributeValues;
}
private static dsFuzzy.AttributeValueClosenessDataTable
GetAttributeValueCloseness(int exampleId)
{
    dsFuzzy.AttributeValueClosenessDataTable dtAttributeValues = new
dsFuzzy.AttributeValueClosenessDataTable();
    dsFuzzyTableAdapters.AttributeValueClosenessTableAdapter adapter = new
dsFuzzyTableAdapters.AttributeValueClosenessTableAdapter();
    dtAttributeValues = adapter.GetDataBy(exampleId);

    return dtAttributeValues;
}
}
}
}

```

9. Literatura

1. P. Bosc, D. H. Kraft, F. Petry, Fuzzy sets in database and information systems, status and opportunities, *Fuzzy Sets and Systems*, 156 (3) (2005) pp. 418-426.
2. D. H. Kraft, F. Petry, Fuzzy information systems, managing uncertainty in databases and information retrieval systems, *Fuzzy Sets and Systems*, 90 (2) (1997) pp. 183-191.
3. D. Dubois, H. Prade, *Fuzzy sets and systems, theory and application*, Academic Press New York, 1980.
4. A. Kaufman, *An introduction to the theory of fuzzy subsets*, Academic Press New York, 1985.
5. J. M. Medina, M. A. Vila, J. C. Cubero, O. Pons, Towards the implementation of a generalized fuzzy relational database model, *Fuzzy Sets and Systems*, 75 (3) (1995) pp. 273-289.
6. P. Bosc, O. Pivert, On the impact of regular functional dependencies when moving to a possibilistic database framework, *Fuzzy Sets and Systems*, 140 (1) (2003) pp. 207-227.
7. W.Y. Liu, Constraints on fuzzy values and fuzzy functional dependencies, *Information Sciences*, 78 (1994) pp. 303-309.
8. S. Sheno, A. Melton, L.T. Fan, Functional dependencies and normal forms in the fuzzy relational database model, *Information Sciences*, 60 (1992) pp. 1-28.
9. B.K. Tyagi, A. Sharfuddin, R.N. Dutta, D. K. Tayal, A complete axiomatization of fuzzy functional dependencies using fuzzy function, *Fuzzy Sets and Systems*, 151 (2) (2005) pp. 363-379.
10. F. Berzal, I. Blanco, D. Sánchez, J.M. Serrano, M.A. Vila, A definition for fuzzy approximate dependencies, *Fuzzy Sets and Systems*, 149 (1) (2005) pp. 105-12.
11. W.Y. Liu, A relational data model with fuzzy inheritance dependencies *Fuzzy Sets and Systems*, 89 (2) (1997) pp. 205-213.
12. G. Chen, E. E. Kerre, J. Vandenbulcke, Normalization based on fuzzy functional dependency in a fuzzy relational data model, *Information Systems*, 21 (3) (1996) pp. 299-310.
13. J. C. Cubero, J. M. Medina, O. Pons, M. A. Vila, Transitive fuzzy dependencies (II), *Fuzzy Sets and Systems*, 106 (3) (1999) pp. 433-448.
14. W.Y. Liu, Fuzzy data dependencies and implication of fuzzy data dependencies, *Fuzzy Sets and Systems*, 92 (3) (1997) pp. 341-348.
15. F. Berzal, I. Blanco, D. Sánchez, J.M. Serrano, M.A. Vila, A definition for fuzzy approximate dependencies, *Fuzzy Sets and Systems*, 149 (1) (2005) pp. 105-129.
16. T. Radecki, A theoretical background for applying fuzzy set theory in information retrieval, *Fuzzy Sets and Systems*, 10 (1983) pp. 169-183.
17. E. A. Rundensteiner, L. W. Hawkes, W. Bandler, On nearness measures in fuzzy relational data models, *International Journal of Approximate Reasoning*, 3 (3) (1989) pp. 267-298.
18. L.A. Zadeh, Fuzzy sets, *Information and control*, 8 (1965) pp.338-353.

19. L.A. Zadeh, Fuzzy sets as a basis of theory of possibility, *Fuzzy Sets and Systems*, (1978) pp. 3-28.
20. L.A. Zadeh, *The role of fuzzy logic in the management of uncertainty in expert systems and approximate reasoning*, Elsevier Science Publishers, (1985) pp. 3-27.
21. L.A. Zadeh, *The concept of a linguistic variable and its application to approximate reasoning*, American Elsevier Publishing Company, 1973.
22. B.P. Buckles, F. E. Petry, Extending the fuzzy database with fuzzy numbers, *Information Sciences*, 2 (1984) pp.145-159.
24. A. Melton, S. Sheno, Fuzzy relations and fuzzy relational databases, *Computers & Mathematics with Applications*, 21 (11-12) (1991) pp.129-138.
25. R. Kruse, J. Gebhard, F.Klawonn, *Foundation of fuzzy sets*, John Wiley and Sons, England 1994.
26. S. Sheno, A. Melton, An extended version of the fuzzy relational database model, *Information Sciences*, 52 (1) (1990) pp. 35-52.
27. J. Kacprzyk, B.P. Buckles, F.E. Petry, Fuzzy information and database systems, *Fuzzy Sets and Systems*, 38 (2) (1990) pp. 133-135.
28. J.M. Medina, O. Pons, M. Amparo Vila, GEFRED: A generalized model of fuzzy relational databases, *Information Sciences*, 76 (1-2) (1994) pp. 87-109.
29. A. Gonzalez, R. Perez, Completeness and consistency conditions for learning fuzzy rules, *Fuzzy Sets and Systems*, 96 (1) (1998) pp. 37-51.
30. D.H. Hong, S.Y. Hwang, A note on the value similarity of fuzzy systems variables, *Fuzzy Sets and Systems*, 66 (3) (1994) pp. 383-386.
31. L. Li, J. Zhang, Attribute reduction in fuzzy concept lattices based on the T- implication, *Knowledge-Based Systems*, (2010).
32. G. Moreno, V. Pascual, A hybrid programming scheme combining fuzzy-logic and functional-logic resources, *Fuzzy Sets and Systems*, 160 (10) (2009) pp. 1402-1419.
33. T. Sudkamp, Similarity, interpolation and fuzzy rule construction, *Fuzzy Sets and Systems*, 58 (1) (1993) pp. 73-86.
34. M. Schneider, Design and implementation of finite resolution crisp and fuzzy spatial objects, *Data & Knowledge Engineering*, 44 (1) (2003) pp. 81-108.
35. V. B. Robinson, Some implications of fuzzy set theory applied to geographic databases, *Computers, Environment and Urban Systems*, 12 (2) (1988).
36. J. C. Cubero, J. M. Medina, O. Pons, M. A. Vila, Data summarization in relational databases through fuzzy dependencies, *Information Sciences*, 121 (3-4) (1999) pp. 233-270.
37. S.M. Chen, C.M. Huang, A new approach to generate weighted fuzzy rules using genetic algorithms for estimating null values, *Expert Systems with Applications*, 35 (3) (2008) pp. 905-917.
38. P. Bosc, M. Galibourg, Indexing principles for a fuzzy database, *Information Systems*, 14 (6) (1989) pp. 493-499.

39. G. Chen, E.E. Kerre, J. Vandenbulcke, The dependency-preserving decomposition and a testing algorithm in a fuzzy relational data model, *Fuzzy Sets and Systems*, 72 (1) (1995) pp. 27-37.
40. J. C. Cubero, J. M. Medina, O. Pons, M. A. Vila, Non-transitive fuzzy dependencies (I), *Fuzzy Sets and Systems*, 106 (3) (1999) pp. 401-431.
41. D.A. Chiang, L.R. Chow, N.C. Hsien, Fuzzy information in extended fuzzy relational databases, *Fuzzy Sets and Systems*, 92 (1) (1997) pp. 1-20.
42. J. Galindo, J.M. Medina, M.C. Aranda-Garrido, Fuzzy division in fuzzy relational databases - an approach, *Fuzzy Sets and Systems*, 121 (3) (1998).
43. M.I. Sözat, A. Yazici, A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations, *Fuzzy Sets and Systems*, 117 (2) (2001) pp. 161-181.
44. T.K. Bhattacharjee, A.K. Mazumdar, Axiomatisation of fuzzy multivalued dependencies in a fuzzy relational data model, *Fuzzy Sets and Systems*, 96 (3) (1998) pp. 343-352.
45. L. Weiyi, The reduction of the fuzzy data domain and fuzzy consistent join, *Fuzzy Sets and Systems*, 50 (1) (1992) pp. 89-96.
46. W.Y. Liu, Extending the relational model to deal with fuzzy values, *Fuzzy Sets and Systems*, 60 (2) (1993) pp. 207-218.
47. B. Bhuniya, P. Niyogi, Lossless join property in fuzzy relational databases, *Data & Knowledge Engineering*, 11 (2) (1993) pp. 109-124.
48. S. Kumar De, R. Biswas, A.R.Roy, On extended fuzzy relational database model with proximity relations, *Fuzzy Sets and Systems*, 117 (2) (2001) pp. 195-201.
49. S. Jyothi, M. S. Babu, Multivalued dependencies in fuzzy relational databases and lossless join decomposition, *Fuzzy Sets and Systems*, 88 (3) (1997) pp. 315-332.
50. R. Vandenbergh, A. V. Schooten, R. De Caluwe, E. E. Kerre, Some practical aspects of fuzzy database techniques - an example, *Information Systems*, 14 (6) (1989) pp. 465-472.
51. Z.M. Ma, L. Yan, Generalization of strategies for fuzzy query translation in classical relational databases, *Information and Software Technology*, 49 (2) (2007) pp. 172-180.
52. K. V. S. V. N. Raju, A. K. Majumdar, The study of joins in fuzzy relational databases, *Fuzzy Sets and Systems*, 21 (1) (1987) pp. 19-34.
53. R. C. Tripathy, P. C. Saxena, Multivalued dependencies in fuzzy relational databases, *Fuzzy Sets and Systems*, 38 (3) (1990) pp. 267-279.
54. B.P. Buckles, F. E. Petry, A fuzzy representation of data for relational databases, *Fuzzy Sets and Systems*, 7 (3) (1982) pp. 213-226.
55. G.Chen, *Fuzzy logic in data modeling, semantics, constraints and database design*, Kluwer Academic Publisher, 1998.
56. C.W.Enteman, A Fuzzy logic with interval truth value, *Fuzzy Sets and Systems*, (2000) pp. 161-189.
57. H.Habiballa, *Fuzzy general resolution*, AFM, 2000.

58. R.C.T.Lee, Fuzzy logic and resolution principle, *Journal of the Association for Computing*, (1982) pp. 109-119.
59. H.B. Potoczny, On similarity relations in fuzzy relational databases, *Fuzzy Sets and Systems*, 12 (3) (1984) pp. 231-235.
60. P. C. Saxena, B. K. Tyagi, Fuzzy functional dependencies and independencies in extended fuzzy relational database models, *Fuzzy Sets and Systems*, 69 (1) (1995) pp. 65-89.
61. S. Sheno, A. Melton, Proximity relations in the fuzzy relational database model, *Fuzzy Sets and Systems*, 100 (1999) pp. 51-62.
62. X. Tang, G. Chen, Equivalence and transformation of extended algebraic operators in fuzzy relational databases, *Fuzzy Sets and Systems*, 157 (12) (2006) pp. 1581-1596.
63. B. P. Buckles, F. E. Petry, H. S. Sachar, A domain calculus for fuzzy relational databases, *Fuzzy Sets and Systems*, 29 (3) (1989) pp. 327-340.
64. S. Sheno, A. Melton, L. T. Fan, An equivalence classes model of fuzzy relational databases, *Fuzzy Sets and Systems*, 38 (2) (1990) pp. 153-170.
65. W. Siler, J. Buckley, *Fuzzy expert systems and Fuzzy reasoning*, Wiley&Sons 2005.
66. Fuzzy logic toolbox for the use with Matlab, MathWorks, 2001
67. R. Belohlavek, Codd's Relational Model from the Point of View of Fuzzy Logic, *Journal of Logic and Computation Advance Access*, 21 (2009) pp.851-862.
68. P. Bosc, P. Pivert, O. Ughetto, Database mining for the discovery of extended functional dependencies, *Fuzzy Information Processing Society, 18th International Conference of the North American*, 2002.
69. T. Hussai, M. M. Awais, S. Shamail, *Applying fuzzy logic to measure completeness of a conceptual model*, 2006.
70. G.P. Lazetić, *Uvod u relacione baze podataka*, Matematički fakultet Beograd, 2008.
71. M. McNeill, E. Thro, *Fuzzy logic - A practical approach*, Academic Press, 1994.
72. Z. Ma, *Fuzzy database modeling of imprecise and uncertain engineering information*, Springer, 2006.
73. J. Haris, *Fuzzy logic application in engineering science*, Springer, 2006.
74. Z.M. Ma, Li Yan, A literature overview of Fuzzy database models, *Information Science and Engineering*, (2008) pp. 172-180.
75. S.B. Yahia, H. Ounalli, A. Jaoua, An extension of classical functional dependency, dynamic fuzzy functional dependency, *Information Science*, 119 (1999) pp. 219-234.
76. P.Bosc, O. Pivert, D. Rocacher, Characterizing the result of the division of fuzzy relations, *International Journal of Approximate Reasoning*, 45 (2007) pp. 511-530.
77. R. Bouaziz, S. Chakhar, V. Mousseau, S. Ram, A. Telmoudi, Database design and querying within the fuzzy semantic model, *Information Sciences*, 177 (2007) pp. 4598–4620.

78. R. A. Ribeiro, A. M. Moreira, Fuzzy query interface for a business database, *Int. J. Human-Computer Studies*, 58 (2003) pp. 63–391.
79. R. B. Bhatt, M. Gopal, On the extension of functional dependency degree from crisp to fuzzy partitions, *Pattern Recognition Letters*, 27 (2006) pp. 487–491.
80. F. E. Petry, L. Zhao, Rudarenje podataka by attribute generalization with fuzzy hierarchies in fuzzy databases, *Fuzzy Sets and Systems*, 160 (2009) pp. 2206 – 2223.
81. D. Dubois, H. Prade, The three semantics of fuzzy sets, *Fuzzy Sets and Systems*, 90 (1997) pp. 141-150.
82. D. Dubois, H. Prade, Fuzzy set and possibility theory-based methods in artificial intelligence, *Artificial Intelligence*, 148 (2003) pp. 1–9.
83. D. Driankov, H. Hellendorn, M. Reinfrank, *An introduction to fuzzy control*, Springer, 1996.
84. D. Dubois, H. Prade, Operations in a fuzzy valued logic, *Information and Control*, (1989) pp. 224-240.
85. J. F. Baldwin, A new approach to approximate reasoning using fuzzy logic, *Fuzzy Sets and Systems*, (1989) pp. 308-325.
86. E. H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic systems, *IEEE Transc Computer*, 26 (1987) pp. 1189-1191.
87. P. Magrez, P. Smets, Fuzzy modus ponens, a new model suitable for application in knowledge – based systems, *Intelligent System*, (1989) pp. 181-200.
88. M. Mizumoto, Fuzzy reasoning under new new competition rules of inference, *Kybernetika*, (1985) pp. 107-117.
89. E. F. Codd, *A relational model of data for large shared databases*, ACM, 1970.
90. D. Maier, *The theory of relational databases*, Computer Science Press, 1985.
91. J. D. Ullman, *Principles of database and knowledge-based systems*, Computer Science Press, 1989.
92. M. Umam, T. Imada, I. Hatono and H. Tamura, Fuzzy object-oriented databases and implementation of its SQL-type data manipulation language, in *Proceedings of the 7th IEEE International Conference on Fuzzy Systems*, 2 (1998) pp. 1344- 1349.
93. V. Cross, R. Caluwe and N. van Gyseghem, A perspective from the fuzzy object data management group (FODMG), in *Proceedings of the 6th IEEE International Conference on Fuzzy Systems*, 2 (1997) pp. 721-728.
94. G. Bordogna and G. Pasi, Graph-based interaction in a fuzzy object oriented database, *International Journal of Intelligent Systems*, 16 (2001) pp. 821-841.
95. G. Bordogna, G. Pasi, and D. Lucarella, A fuzzy object-oriented data model for managing vague and uncertain information, *International Journal of Intelligent Systems*, 14 (1999) pp. 623-651.

96. G. de Tré and R. de Caluwe, Level-2 fuzzy sets and their usefulness in object-oriented database modeling, *Fuzzy Sets and Systems*, 140 (2003) pp. 29-49.
97. D. Dubois, H. Prade and J. P. Rossazza, Vagueness, typicality, and uncertainty in class hierarchies, *International Journal of Intelligent Systems*, 6 (1991) pp. 167-183.
98. R. George, R. Srikanth, F. E. Petry and B. P. Buckles, Uncertainty management issues in the object-oriented data model, *IEEE Transactions on Fuzzy Systems*, 4 (1996) pp. 179-192.
99. J.M.Medina, O.Pons, M.A.Vila and J. C. Cubero, Client/Server architecture for Fuzzy Relational Databases, *Mathware&Soft Computing* 3, 3 (1996) pp.415-424.
100. A.G. Touzi and M. A.B. Hassine, New Architecture of Fuzzy Database Management Systems, *The International Arab Journal of Information Technology*, 6 (3) (2009) pp.213-222.

10. Biografija

Miljan Vučetić rođen je 28.03.1984.godine u Sarajevu. Osnovnu školu i gimnaziju završio je u Sokocu. Za uspeh u osnovnoj i srednjoj školi nagrađen je "Vukovom diplomom". Petogodišnje akademske studije završio je na Saobraćajnom fakultetu u Beogradu, smer telekomunikacije, sa prosečnom ocenom 9.93. U dosadašnjem školovanju dobio je niz pohvala i priznanja za ostvarene rezultate, a kruna dosadašnjeg rada i najznačajnija priznanja su: student generacije, najbolji student od osnivanja Saobraćajnog fakultetu, povelja Univerziteta u Beogradu za najbolje studente generacije 2008/2009. dodeljena na svečanosti organizovanoj povodom dana Univerziteta u Beogradu. Doktorske studije upisao je 2009. godine na Fakultetu organizacionih nauka, smer elektronsko poslovanje.

Naučno-istraživačkim radom bavi se od 2007. godine. Naime, na 10-toj Internacionalnoj konferenciji "Upravljanje kvalitetom i pouzdanošću"-ICDQM 2007, održanoj 13. i 14. juna 2007. u Beogradu, zajedno sa prof.dr Zoranom Avramovićem objavio je rad pod nazivom "O prirodi sebi-sličnosti u eternet saobraćaju". U sklopu nastave na predmetima Elektronika i Statistička teorija telekomunikacija saradivao je i bio angažovan kao saradnik u nastavi na Saobraćajnom fakultetu u Beogradu. U dosadašnjem naučno-istraživačkom radu objavio je veći broj radova na domaćim i međunarodnim konferencijama, časopisima InfoM, JITA, Arhiv za tehničke nauke, te časopisima sa SCI liste Expert System with Applications i TTEM. U oktobru 2011. učestvovao je na CyberC 2011 međunarodnoj konferenciji u Pekingu, Kina, sa radom "E-health transformation model in Serbia: design, architecture and developing" indeksiranom u IEEE Xplore bazi.

Zbog želje za neprekidnim stručnim usavršavanjem i ovladavanjem novih znanja i vještina, uspešno je pohađao kurseve Cisco CCNA, Gustav Kaiser i Microsoft Training Program čime je stekao status sertifikovanog inženjera za različite oblasti. Pored toga poseduje sledeće Microsoft sertifikate Windows Server 2008 Server Administrator, Windows Server 2008 Network Infrastructure, Windows Server 2008 AD Configuring.

Praktična iskustva sticao je na Saobraćajnom fakultetu u Beogradu, različitim projektima na kojima je bio angažovan, te kompanijama Telekom Srpske, VeratNet i E-Smart Systems u kojoj je trenutno i zaposlen.

Spisak objavljenih radova:

1. Miljan Vučetić, Zoran Avramović: "*O prirodi sebi-sličnosti u eternet saobraćaju*" - XIV Internacionalna kofrenecija ICDQM, Beograd 2007,
2. Miljan Vučetić, Slaviša Nogoštić: "*Simulacija rada rutera u telekomunikacionim mrežama primenom algoritama inspirisanih ponašanjem mrava*", SYMOPIS, Ivanjica 2009.
3. Slaviša Nogoštić, Miljan Vučetić: "*Bajes-Nešov ekvilibrijum u zapečaćenim aukcijama*", SYMPOIS, Ivanjica 2009,
4. Miljan Vučetić: "*Strategija razvoja e-zdravstva u Srbiji: primena smart card tehnologije u realizaciji zdravstvenog informacionog sistema*", SYMOPIS, Tara 2010.
5. Nenad Gligorić, Miljan Vučetić, Ana Uzelac, Ljiljan Milosavljević: "*Napredni sistemi za pomoć pri vožnji bazirani na interakciji sa drugim vozilima i infrastrukturnoj komunikaciji*", InfoM, Vol. 34 2010.
6. Miljan Vučetić, Ana Uzelac, Nenad Gligorić: "*Transformacioni model e-zdravstva u Srbiji*", InfoM, Vol. 36 2010.
7. Učešće u izradi *Zbirke zadataka iz Elektronike*, štampanoj na Saobraćajnom fakultetu u Beogradu 2008.
8. Ana Uzelac, Dragan Zoranović, Miljan Vučetić, Nenad Gligorić, Ljiljana Milosavljević: "*Promene u zdravstvenim sistemima zemalja u razvoju omogućene razvojem mobilnih tehnologija sa primerima*", Arhiv za tehničke nauke Bijaljna, No.5 2011.,
9. Miljan Vučetić: "*Analiza podataka u fazi relacionim bazama podataka*", SYMOPIS konferencija, septembar 2011, Zlatibor.
10. Miljan Vučetić: "*Fazi relacioni modeli baza podataka*", SYMOPIS konferencija, septembar 2011, Zlatibor.
11. Miljan Vučetić: "*Functional dependencies analysis in fuzzy relational database model*", Journal of Information Technology and Applications, vol. 1, No2., Banjaluka 2011.
12. Miljan Vučetić, Ana Uzelac, Nenad Gligorić: "*E-health transformation model in Serbia: design, architecture and developing*", CyberC 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, October 2011.
13. Miljan Vučetić: "*Primena fazi logike u relacionim bazama podataka*", InfoM, vol.40 2011.
14. Miljan Vučetić, Mirko Vujošević: "A literature overview of functional dependencies in fuzzy relational database models", *Techniques Technologies Education Managment*, vol. 7, No.4., 2012 IF=0.351.
15. Miljan Vučetić, Miroslav Hudec, Mirko Vujošević: "A new method for computing fuzzy functional dependencies in relational database systems", *Expert Systems with Applications*, Vol. 40 (7) Jun 2013 IF=2.455.

16. M. Stefanović, V. Pejović, M. Vučetić: "*Komunikacioni protokol između AMM centra i koncentratora podataka kao podrška interoperabilnosti celokupnog sistema*", CIRED 2012 konferencija, Vrnjačka banja.
17. Miljan Vučetić: "*Teorijski model za generisanje upita u fazi relacijama*", SYMOPIS 2012 konferencija, septembar 2012, Tara.
18. Miljan Vučetić: "*Projektovanje fazi relacionih baza podataka*", InfoM, vol.44 2012.

Прилог 1.

Изјава о ауторству

Потписани-а **Вучетић Миљан**

број индекса **40/2008**

Изјављујем

да је докторска дисертација под насловом

"Примена фази логике у анализи функционалних зависности у релационим базама података"

- резултат сопственог истраживачког рада,
- да предложена дисертација у целини ни у деловима није била предложена за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

Потпис докторанда

У Београду, 01.02.2013.



Прилог 2.

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора: **Миљан Вучетић**

Број индекса : **40/2008**

Студијски програм: **електронско пословање**

Наслов рада: **"Примена фази логике у анализи функционалних зависности у релационим базама података"**

Ментор: **Мирко Вујошевић**

Потписани/а: **Миљан Вучетић**

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла за објављивање на порталу **Дигиталног репозиторијума Универзитета у Београду**.

Дозвољавам да се објаве моји лични подаци везани за добијање академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис докторанда

У Београду, 01.02.2013.



Прилог 3.

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

"Примена фази логике у анализи функционалних зависности у релационим базама података"

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигитални репозиторијум Универзитета у Београду могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство

2. Ауторство - некомерцијално

3. Ауторство – некомерцијално – без прераде

4. Ауторство – некомерцијално – делити под истим условима

5. Ауторство – без прераде

6. Ауторство – делити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

Потпис докторанда

У Београду, 01.02.2013.



1. Ауторство - Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. Ауторство – некомерцијално. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. Ауторство - некомерцијално – без прераде. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. Ауторство - некомерцијално – делити под истим условима. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. Ауторство – без прераде. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. Ауторство - делити под истим условима. Дозвољаваате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.