University of Niš
Faculty of Sciences and Mathematics
PhD School of Mathematics

Marko S. Stanković

BISIMULATIONS FOR KRIPKE MODELS OF FUZZY MULTIMODAL LOGICS

DOCTORAL DISSERTATION

Niš, 2022.

University of Niš
Faculty of Sciences and Mathematics
PhD School of Mathematics

Marko S. Stanković

BISIMULATIONS FOR KRIPKE MODELS OF FUZZY MULTIMODAL LOGICS

DOCTORAL DISSERTATION

Niš, 2022.

Универзитет у Нишу
Природно-математички факултет
Докторска школа математике

Марко С. Станковић

БИСИМУЛАЦИЈЕ ЗА КРИПКЕОВЕ МОДЕЛЕ ФАЗИ МУЛТИМОДАЛНИХ ЛОГИКА

ДОКТОРСКА ДИСЕРТАЦИЈА

Ниш, 2022.

Supervisor:

Miroslav Ćirić
Full Professor
Faculty of Sciences and Mathematics
University of Niš, Serbia

Members of the Commission:

Jelena Ignjatović
Full Professor
Faculty of Sciences and Mathematics
University of Niš, Serbia

Manfred Droste
Full Professor
Institute of Computer Science
University of Leipzig, Germany

Andreja Tepavčević
Full Professor
Faculty of Sciences
University of Novi Sad, Serbia

Zoran Ognjanović
Research Professor
Mathematical Institute
Serbian Academy of Sciences and Arts, Serbia

Nada Damljanović
Full Professor
Faculty of Technical Sciences in Čačak
University of Kragujevac, Serbia

Date of Defense:_____________

Data on Doctoral Dissertation

Doctoral Supervisor:	Miroslav D. Ćirić, Full Professor at Faculty of Sciences and Mathematics, University of Niš.
Title:	Bisimulations for Kripke models of Fuzzy Multimodal Logics
Abstract:	The main objective of the dissertation is to provide a detailed study of several different types of simulations and bisimulations for Kripke models of fuzzy multimodal logics. Two types of simulations (forward and backward) and five types of bisimulations (forward, backward, forward-backward, backward-forward and regular) are presented hereby. For each type of simulation and bisimulation, an algorithm is created to test the existence of the simulation or bisimulation and, if it exists, the algorithm computes the greatest one. The dissertation presents the application of bisimulations in the state reduction of fuzzy Kripke models, while preserving their semantic properties. Next, weak simulations and bisimulations were considered and the Hennessy-Milner property was examined. Finally, an algorithm was created to compute weak simulations and bisimulations for fuzzy Kripke models over locally finite algebras.
Scientific Field:	Mathematics
Scientific Discipline:	Fuzzy logic and fuzzy sets
Key Words:	Simulations, Bisimulations, Kripke models, Fuzzy logic, Modal logic
UDC:	510.643/.644(043.3) 510.22(043.3)
CERIF Classification:	**03B45** Modal logic, **03B50** Many-valued logic, **03B52** Fuzzy logic; logic of vagueness, **03E72** Theory of fuzzy sets, etc.
Creative Commons License Type:	

Подаци о докторској дисертацији

Ментор:	Мирослав Д. Ћирић, редовни професор, Универзитет у Нишу, Природно-математички факултет.
Наслов:	Бисимулације за Крипкеове моделе фази мултимодалних логика
Резиме:	Главни задатак дисертације јесте да пружи детаљну студију више различитих типова симулација и бисимулација за Крипкеове моделе фази мултимодалних логика. Представљена су два типа симулација (директне и повратне) и пет типова бисимулација (директне, повратне, директно-повратне, повратно-директне и регуларне). За сваки тип симулација и бисимулација креиран је алгоритам који тестира постојање симулације или бисимулације и, уколико иста постоји, алгоритам израчунава највећу. У дисертацији је приказана примена бисимулација у редуковању броја светова фази Крипкеових модела уз очување њихових семантичких својстава. Даље, разматране су слабе симулације и бисимулације и испитано је Хенеси-Милнерово (Hennessy-Milner) својство. На крају, креиран је алгоритам за израчунавање слабих симулација и бисимулација за фази Крипкеове моделе над локално коначним алгебрама.
Научна област:	Математика
Научна дисциплина:	Фази логика и фази скупови
Кључне речи:	симулације, бисимулације, Крипкеови модели, фази логика, модална логика
УДК:	510.643/.644(043.3) 510.22(043.3)
CERIF класификација:	**03B45** Модална логика, **03B50** Вишевредносне логике, **03B52** Фази логика; логика непрецизности, **03E72** Теорија фази скупова, и сл.
Тип лиценце Креативне заједнице:	**CC BY-NC-ND**

Preface

My doctoral dissertation entitled “Bisimulations for Kripke models of Fuzzy Multimodal Logics” includes the results of my previous research during my doctoral studies at University of Niš, Faculty of Sciences and Mathematics, PhD School of Mathematics, module: Algebra and mathematical logic. I sincerely hope that the results presented in this dissertation will be useful to mathematicians, logicians, philosophers, computer scientists, programmers and other people whose focus of interest is fuzzy modal logic or some related scientific areas.

This doctoral dissertation is the crown of my professional and personal development. I am taking the opportunity to sincerely thank the exceptional people who had been with me on this journey.

I herewith thank all my dear friends and colleagues for all the things, both significant and tiny, they have done for me. First of all, I thank them for their understanding during the writing of the dissertation and for every interesting and inspiring moment, we spend together.

I thank my family, above all my dear parents Jelica and Stanko for their love, unconditional support and being there when I needed them most. Special thanks to my twin brother Miloš who is my immeasurable support in life and who provided me with tremendous help while writing my dissertation, especially with the writing program which implements the results of this dissertation. I would like to thank him for the hard work and time he devoted.

I also take this opportunity to thank all the members of the commission for their effort and thorough reading of this dissertation and for their useful suggestions that have greatly improved the quality of the dissertation.

Finally, I owe gratitude to my mentor, Professor Miroslav Ćirić, for the trust and enormous assistance he provided me during the preparation of this dissertation. His advice during learning, conceiving and writing the thesis was of immense value to me. I am glad that I had the opportunity to learn from a leading professor and scientist with vast experience.

Once again, thanks to everyone who encouraged me on this journey.

Niš, 2022. Marko Stanković

Contents

Introduction

> "Mathematics is the most beautiful and most powerful creation of the human spirit."
>
> *Stefan Banach*

The research in this dissertation combines parts of very important mathematical theories, such as Modal logic, Fuzzy logic, Coinduction, etc. Compared to algebra and geometry, Fuzzy logic and Coinduction are very young mathematical disciplines. On the other hand, Modal logic is as old as logic itself. However, the development of modern modal logic began in the twentieth century, so this scientific discipline can also be considered young. So first, we will say a few words about these disciplines, their importance, and their interconnectedness.

* * *

The beginnings of modal logic can be found in antiquity in the ancient Greek philosopher Aristotle (384–322 BC), who was the first to study logic and logical systems systematically. He is the creator of syllogisms, deductive schemes of logical terms and operators and the structures that make it possible to infer true conclusions from given premises. Aristotle also developed modal syllogisms by adding the qualifications "necessarily" and "possibly" to their premises in various ways. It turned out that modal syllogisms are very difficult for satisfiability and interpretation. Therefore, categorical syllogisms have become an important part of classical education, while modal logic is rejected as a failure. Modal logic and semantics were also discussed in the Middle Ages. After the Middle Ages, we will only mention Gottfried Wilhelm Leibniz (1646–1716), who dealt with possible worlds and had the thesis that the existing world is the best world (*best of all possible worlds*). However, these beginnings are negligible compared to the expansion in the 20th century.

The beginnings of modern modal logic appear in the series of articles by philosopher C. I. Lewis at the beginning of 1912, and the first modal systems were introduced in 1918 (for all details, see [80]). Later, modal logic developed primarily through the works of A. N. Prior, J. Hintikka, G. H. von Wright, S. Kanger and others. The real expansion of modal logic came in the 1960s when the semantics was proposed independently by J. Hintikka, S. Kanger and S. A. Kripke (for some notable Kripke works, we refer to [78, 79]). The introduced relational semantics we call today *Kripke semantics* has influenced the continuing development of the field. Kripke semantics show that modal logics are, in fact, logics for reasoning about relational structures. After formalization, modal logics reach the focus of interest of mathematicians, philosophers and computer scientists.

Modal logic is a part of mathematical logic that deals with the *qualification* of sentences, i.e., sentences that are necessarily true, possibly true, provable, obligatory, etc. The initial formalization of modal logic is *alethic modal logic* (from the Greek word *aletheia* which means truth), because modalities of necessity ($\Box$) and possibility ($\Diamond$) are called alethic modalities. The modalities $\Box$ and $\Diamond$ have now become standard notations in modal logic literature. In addition to these standard interpretations and notations, there are several other variants of modal logics with different classes of modalities. For example, *temporal logic* was invented in 1953 by A. N. Prior to dealing with tense operators and now has important application in computer science and formal verification. *Deontic logic* (from the Greek word *déon* meaning "of that which is binding") was founded in 1951 by G. H. von Wright to deal with obligation, permission, prohibition, and related concepts which are characterized as deontic modalities, also with important application in computer science. *Epistemic logic* (from the Greek *episteme* which means knowledge) was also founded by G. H. von Wright to deal with beliefs and knowledge. *Dynamic logic* is modal logic for representing the states and the events of dynamic systems like computer programs, linguistics, artificial intelligence, etc.

The languages of propositional modal logic are the languages of propositional logic enriched by the so-called modal operators or modalities. Modal operators are characterized by great expressive power so that even in the basic modal language, it is possible to express the essential properties of sentences. Let us also mention that *multimodal logic* is a modal logic with more than one modal operator.

* * *

The world's perception is interwoven with concepts that are not completely clear and do not have clearly defined boundaries. Therefore, classical logic, which deals with bivalent propositions (propositions that are either true or false), is not suitable for describing various phenomena in the physical world and deals with vagueness in human thinking and reasoning.

In the history of logic, several times philosophers have considerations on the vagueness of human concepts. Significant progress and ideas can be noticed at the end of the 19th century, when the outlines of multivalued logic can be discerned (a lot of details from this period can be seen in the book [6]).

At the beginning of the 20th century, the work of Jan Łukasiewicz stands out as probably the most influential on the development of multivalued logic. He developed a three-valued logic which he later generalized to n-valued logic where truth values are equidistant rational numbers in $[0, 1]$. In his work with A. Tarski [84], formulae that describe every n-valued generalization of Łukasiewicz's three-valued logic are presented. Also, Łukasiewicz considered in his works the case when the set of truth values are all real numbers in the unit interval $[0, 1]$ (first individually in [83] and later with Tarski in the aforementioned paper [84]). Therefore, the number of works and new ideas in this field has been growing rapidly since the 1920s, preparing the ground for what would follow.

The real flourish in this field follows from the paper "Fuzzy sets" by L. A. Zadeh (see [147]). Because of this work, Zadeh is considered to be the founder of fuzzy logic. Here is Zadeh's original definition of the fuzzy set:

"A *fuzzy set (class)* A in X is characterized by a *membership (characteristic) function* $f_A(A)$ which associates with each point in X a real number in the interval

$[0,1]$, with the value of $f_A(x)$ at x representing the "grade of membership" of x in A."

The original notation was later changed. According to the Zadeh definition, fuzzy sets are sets whose elements have degrees of membership in the real unit interval $[0,1]$ they are generalizations of classical sets. When membership functions of fuzzy sets only take values 0 and 1, we obtain classical bivalent sets, usually called *crisp sets*. Based on the fuzzy sets, the fuzzy logic developed, which is a generalization of classical logic. Somewhat later, after Zadeh's work, J. A. Goguen in [58] proposed the study of the fuzzy sets that take truth values in an arbitrary lattice.

The impact of Zadeh's work [147] became incredibly significant in a short time. According to a study in the journal Nature from 2014, the paper entered the top 100 most cited articles in science (cf. [104]). The number of citations and the influence of fuzzy logic on other scientific fields have been analyzed in detail in the book [6]. Fascinating is the number of citations in other sciences such as Computer Science, Engineering, Decision Sciences, Environmental Science, etc., even in Biochemistry, Genetics, Molecular Biology, Medicine, Psychology, etc.

Fuzzy sets have proven to be an excellent tool for modeling uncertainties, vagueness, ambiguities, linguistic uncertainties, etc. That is why the fuzzy approach has quickly been applied in many areas of mathematics. We are interested in the fuzzy approach that gave fuzzy automata, fuzzy labelled transition systems, and especially the fuzzy modal logic. For an early approach where the fuzzy sets were applied to modal operators, we refer to [131]. Later, the development of fuzzy modal logic progressed with great prosperity (for example, see [16, 50, 51, 106, 115], etc.).

* * *

Mathematical induction is a well-known proof methodology in Mathematics and Computer Science for defining objects and reasoning on their properties. In addition to the standard inductive technique on the domain of positive integers, there are several more techniques such as structural induction, induction on derivation proofs, transition induction, well-founded induction, etc. Coinduction is a dual concept of induction and a powerful technique for reasoning about the behavioral properties of objects in Concurrency Theory. This concept has been discovered and studied in recent years with growing interest and increasing application possibilities in computer science, mathematics, philosophy and physics (cf. [128]).

The most famous instance of coinduction is the concept of a *bisimulation*. Origins of bisimulations can be found in the work of R. Milner [92] and D. Park [108] with the original purpose of modeling behavioral equivalence among processes and reducing the state space of processes. Approximately at the same time, but independently, bisimulations were discovered in modal logic by J. van Benthem [8] as an equivalence principle between Kripke structures. M. Forti and F. Honsell [54] introduced bisimulation in set theory as a natural principle replacing extensionality in the context of non-well-founded sets.

Today, bisimulations are being studied in many areas of computer science. They are employed in functional languages, object-oriented languages, types, data types, domains, databases, compiler optimizations, program analysis, verification tools, etc. More detailed information on the origins of bisimulations and their applications can be found in [127, 128].

Bisimulation is a binary relation between two models (in modal logic, Labelled Transition Systems (LTSs), automaton, etc.) and bisimilar states have the same local properties and match each other's moves (transition possibilities). With such correspondence and the notion of bisimulation, all models can be viewed in the same way and evaluation of formula in modal logic can be viewed as computing LTS or automata computation and vice versa.

Also, since bisimulations are used to reduce the number of states of automata, bisimulation can be used to reduce the number of worlds in modal logic, so it is of great importance to have an algorithm for the computation of bisimulations.

Simulations and bisimulations have so far been most often studied on LTS and in the Automata theory.

* * *

So, the fuzzy modal logic, as well as the mentioned mathematical disciplines, are very young, and they are full of open problems. Some prominent problems concern the axiomatization of the fuzzy modal structures, completeness results, model theory, computational complexity, etc. However, in this dissertation we will deal with problems related to simulations, bisimulations as well as their computations.

Observing structural differences, we can distinguish two different types of simulations/bisimulations. The first ones are known as strong simulations and strong bisimulations, or just simulations and bisimulations which (bi)simulate local properties of worlds and their transition patterns. Bisimulations can be used to reduce the number of worlds in modal logic, so efficient algorithms for the computation of bisimulations are of great importance. The potential for the possible application of these algorithms would be significant, considering the expressive power of Kripke's syntax.

In Kripke models, bisimulations preserve the truth values of formulae, which means that bisimilar worlds are equivalent in the sense that they satisfy the same set of formulae. However, the converse of this assertion is generally not valid because of the finitary characteristic of modal formulae, i.e., equivalent worlds that satisfy the same set of formulae do not have to be bisimilar. The special class of models to which this applies is said to have the Hennessy-Milner property, and the Hennessy-Milner theorem more closely determines such a class. In the fuzzy modal logics, Hennessy-Milner's property is not sufficiently examined.

The second type of simulations and bisimulations are known as weak simulations and weak bisimulations, and they are used for (bi)simulating internal systems' actions (such as automata languages, transitions in labelled transition systems, formulae in Kripke models, etc.). It is generally known that a weak bisimulation on some structures is a fuzzy equivalence called *weak bisimulation equivalence* and this concept is widely used in formal verification and model checking. Weak bisimulation equivalences provide better state reductions of the model than the ordinary strong bisimulations while at the same time they preserve the semantic properties of the model. However, the computation of weak simulations and bisimulations is a computationally hard problem in the general case.

The main aim of the dissertation is to provide a comprehensive study of simulations and bisimulations for Kripke models of fuzzy multimodal logics. Two types of simulations and five types of bisimulations are presented and an algorithm is created to test the existence of the simulation or bisimulation. Then, the dissertation provides the application of bisimulations in the state reduction of the fuzzy Kripke models, while preserving their semantic properties. Next, weak simulations and bisimulations were considered and the Hennessy-Milner property was examined. Finally, an algorithm was created to compute weak simulations and bisimulations for fuzzy Kripke models over locally finite algebras.

The results presented in the dissertation may have various potential applications. We list some of the possible applications. The defined modal language syntax is inter-translatable with the syntax of the fuzzy description logics (cf. [13, 14, 61]), fuzzy temporal logic [29] and social network analysis [48, 49, 71]. In fact, a weighted social network can easily be transformed into a Kripke model (see section 3 from [49]).

The doctoral dissertation consists of six chapters and one appendix.

Chapter 1 serves to provide the basic concepts that will be used throughout the dissertation. In the beginning, some important notions from set theory, universal algebras and lattices are given. In the following, we deal with algebraic structures, especially residuated lattices and Heyting algebras over which the vast majority of results are exposed. Afterwards, the notions of fuzzy sets and fuzzy relations will be introduced as well as uniform fuzzy relations.

Chapter 2 defines Kripke semantics for fuzzy multimodal logics over a complete Heyting algebra. Also, the chapter provides some important properties of the fuzzy formulae as well as the examples of fuzzy Kripke models. At the end of the chapter, we provide some notions of afterset Kripke models.

Chapter 3 defines two types of simulations and five types of bisimulations and gives their characteristics. This chapter gives an algorithm for testing the existence and computation of the greatest simulations and bisimulations of each type, which is one of the main results of the dissertation. As these algorithms do not always terminate in a finite number of steps, we also provide their modifications which determine whether there are crisp simulations or bisimulations of a given type and compute the greatest ones when they exist. Such algorithms always terminate in finitely many steps.

Further, we provide an application of bisimulations in reducing the size of fuzzy Kripke models while preserving their semantic properties. Using an arbitrary fuzzy quasi-order on a given fuzzy Kripke model, we construct a new model called the afterset fuzzy Kripke model. When regular, forward, or backward bisimulation is fuzzy quasi-order, we show that the corresponding afterset model is equivalent to the original one with respect to all modal formulae, to all plus formulae, or all minus formulae.

The chapter abounds in computational examples for computing simulations and bisimulations and reducing the states of the Kripke models.

Chapter 4 defines weak simulations and bisimulations for a given non-empty set Ψ of modal formulae. Also, a lot of characterization of weak simulation and bisimulation has been provided. The concept of weak bisimulation can be used to express the degree of modal equivalence between worlds w and w' with respect to formulae from Ψ. The main result of the chapter is several Hennessy-Milner type theorems. The first theorem determines that the greatest weak bisimulation for the set of plus-formulae between image-finite fuzzy Kripke models coincides with the greatest forward bisimulation. The second theorem follows from duality, i.e., the theorem determines that the greatest weak bisimulation for the set of minus-formulae between domain-finite fuzzy Kripke models coincides with the greatest backward bisimulation. Finally, the third theorem is a consequence of the previous two, i.e., the theorem determines that the greatest weak bisimulation for the set of all modal formulae between the degree-finite fuzzy Kripke models coincides with the greatest regular bisimulation.

Hennessy-Milner type theorems are important for the following reason. The modal equivalence for a given set of formulae can be obtained by computing the greatest weak bisimulation for the corresponding set of formulae, which is generally a computationally hard problem. Results presented in this section reduce these problems to the problems of computing the greatest forward, backward and regular bisimulations, for which efficient algorithms have been developed in Chapter 3. And in this chapter, we provide computational examples which demonstrate the application of the Hennessy-Milner type Theorems.

In this chapter, we also use the concept of uniform relations introduced in [24], in order to study weak simulations and bisimulations that are uniform fuzzy relations. Uniform weak bisimulations are a powerful tool for studying when two fuzzy Kripke models are equivalent, similar to the equivalence between fuzzy automata (cf. [30, 73, 91]).

Chapter 5 deals with the computation of weak simulations and bisimulations. The computation of weak simulations and bisimulations inevitably leads to the formulae explosion problem. Nevertheless, we first developed an algorithm for reachable fuzzy sets for the Kripke model, which terminates in a finite number of steps. The algorithm can be applied to all locally finite algebras. Afterwards, we determined the complexity of this algorithm. Next, we provide an algorithm for computing simulations and bisimulations that is based on the algorithm for reachable fuzzy sets. This chapter also abounds with computational examples for both of the algorithms.

Chapter 6 provides some generalized results from the previous chapters and brings forth some interesting computational examples.

Chapter A is an appendix that provides the implementation of the algorithms developed in the previous chapters in the Java programming language and shows the corresponding source codes.

At the end of the dissertation, List of Abbreviations, List of Symbols, Index and Bibliography, are given to make it easier to navigate through the document.

Chapter 1

Fundamental concepts

"Nature is written in mathematical language."

Galileo Galilei

This chapter contains basic concepts and notations as well as some known results which will be used in the thesis. To make the dissertation as self-contained as possible, the chapter consists of eight sections.

First, in Section 1.1, some basic concepts and notations from set theory will be defined, which is necessary for further work. We define binary relations and give special attention to equivalence relations. In the dissertation, we also use terms from universal algebras and we give an overview of basic ones in Section 1.2. The terms are according to the classical textbook of S. Burris and H. P. Sankappanavar [17]. In Section 1.3, the elementary notions of a partially ordered set and a lattice will be introduced, based on influential books of G. Birkhoff [10], T. S. Blyth [12], B. Davey and H. Priestley [31] and S. Roman [122]. At the end of the section, we present the famous Knaster-Tarski Theorem from the fixed point theory, which is essential for obtaining some important results presented in the dissertation.

Since we work with several different algebraic structures, in Section 1.4, we give an overview of those that are most important to us, their properties and mutual relations. The following Section 1.5 gives the basic properties of the most general structure we work with, i.e., residuated lattices. The notations in this section are according to the book of Bělohlávek and Vychodil [7]. Then, Section 1.6 provides some properties of Heyting algebras on which the vast majority of the results are given. Section 1.7 provides basic notions of fuzzy sets and fuzzy relations. At the end of the section some features of fuzzy equivalences and fuzzy quasi-orders are presented. Last Section 1.8, contains results for uniform fuzzy relations from work of M. Ćirić, J. Ignjatović, S. Bogdanović [24]. These results will be especially important to us when dealing with uniform weak simulations and bisimulations.

1.1 Sets and relations

We will use the terms and symbols from Set Theory as is usual in this theory. We denote the cardinal number of the set A by $|A|$. The family of sets indexed by a set I will be denoted by A_i, $i \in I$, or $\{A_i \mid i \in I\}$ or $\{A_i\}_{i\in I}$. If the indexed set is finite and has n elements, then we usually write $I = \{1, 2, \dots, n\}$ and the family indexed by I is denoted by $A_1, A_2, \dots, A_n$ or $\{A_i\}_{i=1}^n$.

Definition 1.1. Let $\{A_i\}_{i\in I}$ be the non-empty family (possibly infinite) of non-empty sets. Then, the *Cartesian product* (or *direct product*) of the sets $\{A_i\}_{i\in I}$ is the following set of functions:

$$\prod_{i\in I} A_i = \left\{ a : I \to \bigcup_{i\in I} A_i \mid (\forall i)(a(i) \in A_i) \right\}. \tag{1.1}$$

Hence, the Cartesian product is the set of all function defined on the index set such that function takes an element $i \in I$ and maps it to an element $a(i) \in A_i$. For the sake of simplicity, the element $a \in A = \prod_{i\in I} A_i$ we write as $(a_i)_{i\in I}$, or just shorter (a_i), where I is given set of indices, and a_i is *ith coordinate* of a, for $i \in I$.

In particular, if the indexed set I is finite and has n elements, then the direct product of the family $\{A_i\}_{i\in I}$ is called *n-ary product* and is defined by the following set of functions:

$$\prod_{i\in I} A_i = \{a : \{1, \dots, n\} \to A_1 \cup \ldots \cup A_n \mid a(i) \in A_i \text{ for every } i \in \{1, \dots, n\}\}.$$

More specific, n-ary Cartesian product is usually interpreted as

$$\prod_{i\in I} A_i = A_1 \times A_2 \times \cdots \times A_n = \{(a_1, \dots, a_n) \mid a_i \in A_i \text{ for every } i \in \{1, \dots, n\}\}.$$

In that case, an arbitrary element $(a_1, \dots, a_n) \in A_1 \times A_2 \times \cdots \times A_n$ is called an *ordered n-tuple*, and if $n = 2$, then we call it an *ordered pair*. Moreover, if $A_i = A$, for every $i \in I$, then the direct product $\prod_{i\in I} A_i$ is denoted by A^I and called *Cartesian power* of the set A, and if I has n elements, then we denote $\prod_{i\in I} A_i$ by A^n. Additionally, we define $A^0 = \{\emptyset\}$.

Definition 1.2. Let A, B be non-empty sets. A *binary relation over sets A and B* is any subset R of the Cartesian product $A \times B$.

Let's emphasize that R can be even an empty subset. When $A = B$, we will say that R is a *binary relation on A*. Since we usually work with binary relations, we simply call them *relations*.

Here are some special relations on a set A that are often used:

- the *empty relation* usually denoted by $\emptyset$;
- the *identity relation* $\Delta_A = \{(a, a) \mid a \in A\}$;
- the *universal relation* $\nabla_A = \{(a, b) \mid a, b \in A\}$.

Let R be a relation on a set A. If elements $a, b \in A$ are in a relation R, it can be written $(a, b) \in R$, or more usual aRb.

Definition 1.3. For non-empty sets A, B and C, and relations $R \subseteq A \times B$ and $S \subseteq B \times C$, the *composition* of relations R and S on the sets A and C is a relation $R \circ S \subseteq A \times C$ defined by:

$$R \circ S = \{(a, c) \in A \times C \mid (\exists b \in B)(a, b) \in R \text{ and } (b, c) \in S\}. \tag{1.2}$$

If R is a relation over sets A and B, then $R^{-1} = \{(a, b) \mid (b, a) \in R\}$ is the *inverse* relation of the given relation R over A and B. Further, the set of all first components of the ordered pairs of R is called the *domain* of R. The set of all second components of the ordered pairs of R is called the *image* of R (or codomain or *range*). Formally, we have

$$\begin{aligned}\mathrm{Dom}(R) &= \{a \in A \mid (\exists b \in B)(a, b) \in R\},\\ \mathrm{Im}(R) &= \{b \in B \mid (\exists a \in A)(a, b) \in R\}.\end{aligned}$$

Clearly, it is valid $\mathrm{Dom}(R^{-1}) = \mathrm{Im}(R)$ and $\mathrm{Im}(R^{-1}) = \mathrm{Dom}(R)$.

Now, we define basic properties of binary relations on a set A. Given relation R on a non-empty set A is called:

- *reflexive* if $(a, a) \in R$ for every $a \in A$, that is, if $\Delta_A \subseteq R$;
- *symmetric* if $(a, b) \in R$ implies $(b, a) \in R$ for all $a, b \in A$, i.e., $R^{-1} \subseteq R$;
- *antisymmetric* if $(a, b) \in R$ and $(b, a) \in R$ implies $a = b$ for all $a, b \in A$, i.e., $R^{-1} \cap R = \Delta_A$.
- *transitive* if $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$ for all $a, b, c \in A$, i.e., $R \circ R \subseteq R$.

A reflexive, antisymmetric and transitive relation on the non-empty set A is called a *partial order* on A, or briefly an *order* on A. A reflexive, symmetric and transitive relation on the non-empty set A is called an *equivalence relation* on A.

We will now give some basic concepts related to equivalence relations. For relation R on A we define the two subsets

$$R_a = \{x \in A \mid (a, x) \in R\}, \qquad R^a = \{x \in A \mid (x, a) \in R\}.$$

Definition 1.4. Let R be an equivalence relation on A and a be an arbitrary element from A. Then, the *equivalence class of* a, denoted R_a (or $[a]_R$), is the set of all elements of A which are equivalent to a, i.e., $R_a = \{b \in A \mid (a, b) \in R\}$.

Usually, when relation R is clear from the context, we omit it from $[a]_R$ and write $[a]$. The set of all equivalence classes of A is denoted by A/R called the *factor set* (or *quotient set*). Notationally, we write:

$$A/R = \{R_a \mid a \in A\}.$$

Further, every element from some class R_a can be chosen to represent the class and such element is called a *representative* of the class. Equivalence classes have a property that in some sense "covers" set A. Set A is partitioned into equivalence classes and so we come to the following definition.

Definition 1.5. Let A be a set and $\{A_i \mid i \in I\}$ be subsets of A. Then $\{A_i \mid i \in I\}$ is a *partition* of A, if and only if the following hold:

(1) $\bigcup_{i \in I} A_i = A$;

(2) if $A_i \neq A_j$, then $A_i \cap A_j = \emptyset$, for every i, j from I.

Hence, every element of the set A belongs to one and only one equivalence class.

Let α be the function from A onto A/R defined by $\alpha(a) = R_a$. The function α is usually called the *natural function*, or *canonical map* (or the *canonical surjection* or the *natural projection*) from A to A/R. Also, let us note that f can be seen as a relation over set A and A/R defined by

$$\alpha(a, R_b) = R(a, b), \qquad \text{for all } a, b \in A.$$

Now, we have the following definition.

Definition 1.6. Let $\alpha : A \to B$ be any function. Then,

$$\ker(\alpha) = \{(a, b) \in A^2 \mid f(a) = f(b)\} \tag{1.3}$$

is called the *kernel* of α.

It is easy to see that $\ker(\alpha)$ is an equivalence relation on A.

The following two theorems state the fundamental properties of the quotient set A/E, where E is an equivalence relation. For more details, see [85].

Theorem 1.1. *If E is an equivalence relation on a set A, the natural projection $\alpha : A \to A/E$ is a surjection and $\ker(\alpha)$ is equivalence relation on A.*

Theorem 1.2. *Let $\alpha : A \to B$ be a surjective function. Then, there is bijection β from $A/\ker(\alpha)$ to B defined by $\alpha = \beta \circ \gamma$, where γ is a natural projection from A to $A/\ker(\alpha)$.*

The bijection β in the previous Theorem is uniquely determined. The theorem may be visualized by the diagram in Figure 1.1. If functions α and γ are given, then there exists a unique bijection β which makes the diagram commute.

$$\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "\gamma"'] & B \\
 & A/\ker(\alpha) \arrow[u, "\beta"']
\end{tikzcd}$$

Figure 1.1: Commutative diagram

1.2 Universal algebras

In this section, we recall some general definitions and theorems from universal algebra. In the following chapters, we will use terms such as *subalgebra*, *isomorphism*, etc., so we will define them here. The terms and notions are taken from a noteworthy book in this field [17], with a slight change of notations due to the uniformity of the dissertation. For the missing terms and notions, we refer to the mentioned book.

Definition 1.7. For a nonnegative integer n, an *n-ary operation* (or *function*) *on* A is any function $f : A^n \to A$. Number n is called the *arity* (or *rank*) of f. A *finitary* operation is an n-ary operation, for some n.

In particular, when arity of the operation f is equal to 0 we call it *nullary* operation or *constant*.

Definition 1.8. A *language* (or *type*) of algebras is a set $\mathcal{F}$ of *function symbols* such that a nonnegative integer n is assigned to each member f of $\mathcal{F}$.

Therefore, a language is a pair $(\mathcal{F}, V)$ where $\mathcal{F}$ is a non-empty set of function symbols and V is a function that assigns an arity $n \in \mathbb{N}_0$ to each symbol in $\mathcal{F}$.

The integer n is called the *arity* of f and f is said to be an *n-ary function symbol.* The set $\mathcal{F}$ can be represented in the form $\bigcup_{n\in\mathbb{N}_0} \mathcal{F}_n$ where $\mathcal{F}_n$ is a subset of n-ary functions symbols in $\mathcal{F}$.

Definition 1.9. If $\mathcal{F}$ is a language of algebras then an *algebra* $\mathbf{A}$ of *type* $\mathcal{F}$ is an ordered pair (A, F) where A is a non-empty set and F is a family of finitary operations on A indexed by the language $\mathcal{F}$ such that corresponding to each n-ary function symbol f in $\mathcal{F}$ there is an n-ary operation $f^{\mathbf{A}}$ on A. The set A is called the *universe* (or *underlying set*) of $\mathbf{A} = (A, F)$, and the $f^{\mathbf{A}}$'s are called the *fundamental operations of* $\mathbf{A}$.

In practice, we write just f instead $f^{\mathbf{A}}$, except when it is necessary to emphasize it.

Informally speaking, a *subalgebra* is a subset of an algebra, closed under all its operations. Formally, we give the following definition.

Definition 1.10. Let $\mathbf{A}$ and $\mathbf{B}$ are two algebras of the same type $\mathcal{F}$. Algebra $\mathbf{B}$ is a subalgebra of $\mathbf{A}$, whenever $B \subseteq A$ and every fundamental operation of $\mathbf{B}$ is the restriction of the corresponding operation of $\mathbf{A}$, i.e., if the following hold:

(i) $f^{\mathbf{B}} = f^{\mathbf{A}}$, for each $f \in \mathcal{F}_0$,
(ii) $f^{\mathbf{B}}(a_1, \dots, a_n) = f^{\mathbf{A}}(a_1, \dots, a_n)$, for each $f \in \mathcal{F}_n$, $n \in \mathbb{N}$ and for all $a_1, \dots, a_n \in B$.

Given an algebra $\mathbf{A}$, for arbitrary $H \subseteq A$, there exists the smallest subalgebra containing H. For this algebra we say that is the *subalgebra of the algebra of* $\mathbf{A}$ *generated by* H, which is denoted $\langle H \rangle$. For $H \subseteq A$ we say H *generates* $\mathbf{A}$ (or $\mathbf{A}$ *is generated by* H, or H is a set of generators of $\mathbf{A}$) and in that case, we denote $\langle H \rangle = \mathbf{A}$. Algebra $\mathbf{A}$ is *finitely generated* if it has a finite set of generators.

Definition 1.11. Let $\mathbf{A}$ and $\mathbf{B}$ are two algebras of the same type $\mathcal{F}$. A mapping $\alpha : A \to B$ is called a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ if the following hold:

(i) $\alpha(f^{\mathbf{A}}) = f^{\mathbf{B}}$, for each $f \in \mathcal{F}_0$,
(ii) $\alpha(f^{\mathbf{A}}(a_1, \dots, a_n)) = f^{\mathbf{B}}(\alpha(a_1), \dots, \alpha(a_n))$, for each $f \in \mathcal{F}_n$, $n \in \mathbb{N}$ and for all $a_1, \dots, a_n \in A$.

If, in addition, α is injective then is called *monomorphism* or *embedding*. If α is surjective then is called *epimorphism* and then $\mathbf{B}$ is said to be a *homomorphic image* of $\mathbf{A}$. If α is bijection, we say $\mathbf{A}$ is *isomorphic* to $\mathbf{B}$, and write $\mathbf{A} \cong \mathbf{B}$. In case $\mathbf{A} = \mathbf{B}$ a homomorphism is also called an *endomorphism* and if it is also an isomorphism, then we say it is an *automorphism* of algebra $\mathbf{A}$.

Equivalence relations must respect algebra operations. So we have the following definition.

Definition 1.12. Let $\mathbf{A}$ be an algebra of type $\mathcal{F}$ and let R be an equivalence relation on A. Then R is a *congruence* on $\mathbf{A}$ if R satisfies *compatibility property*, i.e., for each function symbol $f \in \mathcal{F}_n$, $n \in \mathbb{N}$, and elements $a_i, b_i \in A$, if $(a_i, b_i) \in R$ holds for $i \in \{1, \dots, n\}$ then $(f^{\mathbf{A}}(a_1, \dots, a_n), f^{\mathbf{A}}(b_1, \dots, b_n)) \in R$ holds.

Let R be a congruence on an algebra $\mathbf{A}$. Then the *quotient algebra of* $\mathbf{A}$ by R, denoted by $\mathbf{A}/R$, is the algebra whose universe is A/R and whose fundamental operations satisfy

$$f^{\mathbf{A}/R}(R_{a_1}, \dots, R_{a_n}) = R_{f^{\mathbf{A}}(a_1, \dots, a_n)}$$

where $a_1, \dots, a_n \in A$ and $f \in \mathcal{F}_n$, $n \in \mathbb{N}$.

The algebra $\mathbf{A}$ and corresponding quotient algebra $\mathbf{A}/R$ are of the same type.

Let $\mathbf{A}$ be an algebra and let R be a congruence on A. The *natural function* $\alpha : A \to A/R$ is defined by $\alpha_R(a) = R_a$.

Now, analogously like in Definition 1.6, we have the following.

Definition 1.13. Let $\alpha : \mathbf{A} \to \mathbf{B}$ be a homomorphism. Then,

$$\ker(\alpha) = \{(a, b) \in A^2 \mid \alpha(a) = \alpha(b)\}$$

is called the *kernel* of α.

Analogously like in the previous section, $\ker(\alpha)$ is a congruence on $\mathbf{A}$.

The *natural homomorphism* from an algebra $\mathbf{A}$ to an quotient algebra $\mathbf{A}/R$ is given by the natural function.

Theorem 1.3 (First Isomorphism Theorem)**.** *Let $\alpha : \mathbf{A} \to \mathbf{B}$ is a homomorphism onto B. Then, there is an isomorphism β from $\mathbf{A}/\ker(\alpha)$ to $\mathbf{B}$ defined by $\alpha = \beta \circ \gamma$, where γ is natural homomorphism from $\mathbf{A}$ to $\mathbf{A}/\ker(\alpha)$.* (See Figure 1.2 (a).)

$$\begin{tikzcd}
\mathbf{A} \arrow[r, "\alpha"] \arrow[dr, "\gamma"'] & \mathbf{B} \\
& \mathbf{A}/\ker(\alpha) \arrow[u, "\beta"']
\end{tikzcd}$$

(a) First Isomorphism Theorem

$$\begin{tikzcd}
\mathbf{A} \arrow[r, "\gamma_P"] \arrow[dr, "\gamma_Q"'] & \mathbf{A}/P \arrow[r, "\gamma_{Q/P}"] \arrow[d, "\beta"] & (\mathbf{A}/P)/(Q/P) \arrow[dl, "\alpha"] \\
& \mathbf{A}/Q &
\end{tikzcd}$$

(b) Second Isomorphism Theorem

Figure 1.2: Homomorphism theorems

Theorem 1.4 (Second Isomorphism Theorem)**.** *If P, Q are congruences on $\mathbf{A}$ and $P \leqslant Q$, then the map*

$$\alpha : (\mathbf{A}/P)/(Q/P) \to \mathbf{A}/Q$$

defined by

$$\alpha((Q/P)_{P_a}) = Q_a \tag{1.4}$$

is an isomorphism from $(\mathbf{A}/P)/(Q/P)$ to $\mathbf{A}/Q$.

Figure 1.2 (b) explains Theorem 1.4. The maps $\gamma_P : \mathbf{A} \to \mathbf{A}/P$, $\gamma_Q : \mathbf{A} \to \mathbf{A}/Q$, $\gamma_{Q/P} : \mathbf{A}/P \to (\mathbf{A}/P)/(Q/P)$ are natural homomorphisms. Also, from (1.4) it follows $Q/P(P_a, P_b) = Q(a, b)$, for every $a, b \in A$.

1.3 Ordered sets and lattices

The majority of terms and notions in this section are from books [10, 12, 31, 122] with notation adjustment.

We already defined a (partial) order on a set A. The order is usually denoted by $\leqslant$. Hence, $\leqslant$ is an order on A if and only if:

- $a \leqslant a$ for every $a \in A$;
- If $a \leqslant b$ and $b \leqslant a$ implies $a = b$ for all $a, b \in A$;
- If $a \leqslant b$ and $b \leqslant c$ implies $a \leqslant c$ for all $a, b, c \in A$.

Definition 1.14. A pair $(A, \leqslant)$, where A is a non-empty set and $\leqslant$ is an order on A is called a *partially ordered set* or just an *ordered set* or a *poset*.

Commonly, the dual order $\leqslant$ on A is denoted by the symbol $\geqslant$, which we read as “greater than or equal to”. Then, the ordered set $(A, \geqslant)$ is called the *dual* of $(A, \leqslant)$. Therefore, to each statement that concerns an order on a set A there is a dual statement that concerns the corresponding dual order on A. Formally, we have the following principle.

Principle of Duality. To every theorem that concerns an ordered set A there is a corresponding theorem that concerns the dual ordered set. This is obtained by replacing each statement that involves $\leqslant$, explicitly or implicitly, by its dual.

Definition 1.15. An ordered set P is said to satisfy the *ascending chain condition* (ACC) if every ascending sequence of elements of P eventually terminates, i.e., if for every ascending sequence $\{a_k\}_{k\in\mathbb{N}}$ of elements of P there exists $k \in \mathbb{N}$ such that $a_k = a_{k+l}$, for all $l \in \mathbb{N}$. In other words, P satisfies ACC if there is no infinite ascending chain in P.

Dually, we define *descending chain condition*, (DCC).

If $\leqslant$ is an order on A, then $<$ denote the relation on A given by:

$$a < b \text{ if and only if } a \leqslant b \text{ and } a \neq b,$$

and with $\geqslant$ and $>$ we denote the inverse of relations $\leqslant$ and $<$, respectively. An order $\leqslant$ on A is a *linear order* on A if, for every $a, b \in A$ holds $a \leqslant b$ or $b \leqslant a$. In this case, A is a *linearly ordered set*.

Definition 1.16. A mapping ϕ from the ordered set $(A, \leqslant_1)$ to the ordered set $(B, \leqslant_2)$ is called *isotonic* or *order preserving* if $a \leqslant_1 b$ implies $\phi(a) \leqslant_2 \phi(b)$ for all $a, b \in A$. Similarly, a mapping ϕ from the ordered set $(A, \leqslant_1)$ to the ordered set $(B, \leqslant_2)$ is called *antitonic* if $a \leqslant_1 b$ implies $\phi(a) \geqslant_2 \phi(b)$ for all $a, b \in A$. A mapping ϕ is an *isomorphism* of ordered sets A and B, or *ordered isomorphism* from A to B, if ϕ is a bijection from A to B then ϕ and ϕ^{-1} both are isotonic mappings.

Definition 1.17. Let $(A, \leqslant)$ be an ordered set. An element $a \in A$ is called:

- the *minimal element* of the set A, if $x \leqslant a$ implies $x = a$ for every $x \in A$, that is, if there is no element in the set A which is strictly smaller than a;
- *the least element* of the set A, if for every $x \in A$ holds $a \leqslant x$, i.e., if a is less or equal than any element from A.

By Principle of duality, we define the *maximal element* and *the greatest element* of the set A.

Definition 1.18. Let M be a non-empty subset of ordered set $(A, \leq)$. An element $a \in A$ is called:

- the *lower bound* of the set M, if $a \leqslant x$ for every $x \in M$;
- the *greatest lower bound* or the *infimum* of the set M, if it is the greatest element in the set of all lower bounds of M, in other words, if it is the lower bound of M and for any lower bound b of the set M there holds $b \leqslant a$.

Again by duality, we have a notion of upper bound and least upper bound. Hence, an element $a \in A$ is the *upper bound* of the set M, if $x \leqslant a$ for every $x \in M$. The *least upper bound* or the *supremum* of the set M, if it is the least element in the set of all upper bounds of M, in other words, if it is the upper bound of M and for any upper bound b of the set M there holds $a \leqslant b$.

The supremum of the set M, if it exists, is denoted by $\bigvee M$, whereas the infimum of M, if it exists, is usually denoted by $\bigwedge M$. If $M = \{a_i\}_{i \in I}$, instead of $\bigvee M$ and $\bigwedge M$ we can write, respectively:

$$\bigvee_{i \in I} a_i \qquad \text{and} \qquad \bigwedge_{i \in I} a_i.$$

Definition 1.19. A partially ordered set $(L, \leqslant)$, such that every two-element subset has the infimum is called *meet semilattice*.

Equivalent terminology is $\wedge$-*semilattice*. It can be easily proven, by induction, that every finite subset of a meet semilattice has an infimum. However, for an infinite subset of a meet semilattice, it doesn't have to be the case.

Let L be a meet semilattice and let denote infimum of two-element set $\{a, b\}$ with $a \wedge b$. Then, binary operation $\wedge$ on L is defined in the following way:

$$\wedge : (a, b) \mapsto a \wedge b.$$

Operation $\wedge$ is called *intersection* and therefore $\bigwedge M$ is a *intersection of the set* M and $a \wedge b$ is a *intersection of elements* a *and* b.

In addition to order-theoretic Definition 1.19, meet semilattice can be defined equivalently via purely algebraic definition:

Definition 1.20. A meet semilattice is an algebraic structure $(L, \wedge)$ consisting of a set L, binary operation $\wedge$, such that for all $a, b, c \in L$ the following hold:

(SL1) $a \wedge (b \wedge c) = (a \wedge b) \wedge c$ (associativity);

(SL2) $a \wedge b = b \wedge a$ (commutativity);

(SL3) $a \wedge a = a$ (idempotency).

Hence, a meet semilattice is an idempotent commutative semigroup. Dually, in order-theoretic definition of meet semilattice replacing "infimum" with "supremum" leads to the dual concept of a *join semilattice* ($\vee$-*semilattice*). Then, the operation of the *union* is defined $\vee : (a, b) \mapsto a \vee b$. Also, replacing the symbol $\wedge$ with $\vee$ in the algebraic definition of meet semilattice will give an algebraic definition of join semilattice.

Now, we give order-theoretic and algebraic definitions of the lattice.

Definition 1.21. A *lattice* is an ordered set $(L, \leqslant)$ which, concerning its order, is both a meet semilattice and a join semilattice, i.e., every two-element subset has the supremum and the infimum.

Definition 1.22. A lattice is an algebraic structure $(L, \wedge, \vee)$ consisting of a set L, binary operations $\wedge$ and $\vee$, such that for all $a, b, c \in L$ the following hold:

(L1) $a \wedge (b \wedge c) = (a \wedge b) \wedge c, \qquad a \vee (b \vee c) = (a \vee b) \vee c$ (associativity);

(L2) $a \wedge b = b \wedge a, \qquad a \vee b = b \vee a$ (commutativity);

(L3) $a \wedge a = a, \qquad a \vee a = a$ (idempotency);

(L4) $a \wedge (a \vee b) = a, \qquad a \vee (a \wedge b) = a$ (absorption).

The conditions (L1)-(L4) are called *the lattice axioms*.

It is easy to check that Definitions 1.21 and 1.22 are mutually equivalent. Usually, both are used equally, depending on what is needed at the moment.

Definition 1.23. A non-empty subset I of a lattice L is called an *ideal* (or *down-set*) if:

(1) for all $a, x \in L$, by $x \leqslant a$ and $a \in I$ it follows $x \in I$;

(2) $x \vee y \in I$, for all $x, y \in L$.

The dual notion to the notion of an ideal is the *dual ideal* (or *up-set* or *filter*). Namely, a non-empty subset D of a lattice L is a dual ideal if:

(1) for all $a, x \in L$, by $a \leqslant x$ and $a \in D$ it follows $x \in D$;

(2) $x \wedge y \in D$, for all $x, y \in L$.

By a *principal ideal generated by* x we will mean an ideal of the form $I = \{a \in L \mid a \leqslant x\}$. Analogously, *the principal dual ideal generated by* x is the dual ideal of the form $D = \{a \in L \mid x \leqslant a\}$.

The least element of the lattice L, if it exists, is denoted by 0, and the greatest element, if it exists, is denoted by 1. A *bounded lattice* is a lattice that has the greatest element 1 and the least element 0.

One of the most important varieties of lattices are *distributive lattice*. Those lattices satisfy two equivalent distributive identities:

(L5) $a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$, for all $a, b, c \in L$;

(L5') $a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$, for all $a, b, c \in L$;

Definition 1.24. If L is a bounded lattice, then $b \in L$ is a *complement* of $a \in L$ if $a \wedge b = 0$ and $a \vee b = 1$. In this case we also say that a and b are complementary.

According to definitions of lattice, every finite subset of the lattice has an infimum and supremum. Nevertheless, for an arbitrary subset infimum and supremum do not have to exist. Hence, L is a *complete lattice* if L has infimum and supremum for an arbitrary subset. Every complete lattice is bounded. A subset K of a complete lattice L is a *complete sublattice* of L if the infimum and supremum (in L) of every non-empty subset of K belongs to K.

Now, here are some examples of lattices.

Example 1.1. Every chain is a lattice where $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$.

Example 1.2. Structure $(\mathbb{N}, |)$ is a bounded lattice, i.e., the set of natural numbers $\mathbb{N}$ with the operation of division as a partial order. The bottom element is 1 and the top element is 0. In this case $a \wedge b = \mathrm{hcf}\{a, b\}$ and $a \vee b = \mathrm{lcm}\{a, b\}$.

Example 1.3. For every non-empty set A, the set $\mathscr{E}(A)$ of equivalence relations on A is a complete lattice. If $F = \{R_i \mid i \in I\}$ is a family of equivalence relations on A, then operation of infimum in $\mathscr{E}(A)$ is defined as the operation of the intersection of family F, i.e., $\bigwedge_{i \in I} R_i = \bigcap_{i \in I} R_i$. Therefore,

$$(a, b) \in \bigwedge_{i \in I} R_i \qquad \text{iff} \qquad (\forall i \in I)(a, b) \in R_i.$$

Operation of supremum is not a simple union of relations, since the union of two equivalence relations does not have to be an equivalence (in the general case, transitivity does not have to be valid anymore). Hence, for the supremum of family F, consider the relation θ defined by $(a, b) \in \theta$ if and only if there exists a sequence $c_1, \ldots, c_n$ and sequence of relations $R_{i_1}, \ldots R_{i_{n+1}}$ such that

$$a \overset{R_{i_1}}{\equiv} c_1 \overset{R_{i_2}}{\equiv} c_2 \overset{R_{i_3}}{\equiv} \cdots \overset{R_{i_n}}{\equiv} c_n \overset{R_{i_{n+1}}}{\equiv} b.$$

It is clear that $\theta \in \mathscr{E}(A)$. If $(a, b) \in R_i$, i.e., $a \overset{R_i}{\equiv} b$, for any $R_i \in F$ it is obvious that $R_i \leqslant \theta$. Now, by transitivity of θ, every relation which is upper bound of F is also the upper bound of θ. Therefore, we conclude that with previous defined operation of supremum, $\mathscr{E}(A)$ forms a complete lattice. The described relation θ is called the *transitive closure* or *transitive product* of the family $\{R_i \mid i \in I\}$.

Moreover, lattice top and bottom elements are Δ_A and ∇_A, respectively.

Now, we list a few important terms related to fixed points from [122].

Definition 1.25. Let L be an ordered set and a map $f : L \to L$, an element $x \in L$ is called:

- *fixed point* (or *fixpoint*) of f if $f(x) = x$.
- *pre-fixed point* of f if $f(x) \leqslant x$.
- *post-fixed point* of f if $x \leqslant f(x)$.

The corresponding sets of fixed points are denoted by $\mathrm{Fix}(f)$, $\mathrm{Pre}(f)$ and $\mathrm{Post}(f)$. Further, the smallest elements of these sets, if they exist, are denoted by $\mathrm{MinFix}(f)$, $\mathrm{MinPre}(f)$ and $\mathrm{MinPost}(f)$, respectively. The largest elements are denoted by $\mathrm{MaxFix}(f)$, $\mathrm{MaxPre}(f)$ and $\mathrm{MaxPost}(f)$.

The first theorem is the result of Knaster [77], and that result was improved with the work of Tarski [138] which led to the famous Knaster-Tarski Fixed point Theorem.

Theorem 1.5. *If L is a complete lattice and if $f : L \to L$ is an isotone mapping then f has a fixed point.*

Theorem 1.6 (Knaster-Tarski Fixed point Theorem)**.** *Let L be a complete lattice and $f : L \to L$ an isotone map. Then* $\mathrm{Fix}(L)$ *is a complete lattice, with bounds*

$$\mathrm{MaxFix}(f) = \mathrm{MaxPost}(f) = \bigvee \mathrm{Post}(f) \tag{1.5}$$

and

$$\mathrm{MinFix}(f) = \mathrm{MinPre}(f) = \bigwedge \mathrm{Pre}(f). \tag{1.6}$$

Therefore, according to the theorem, we conclude that each of the sets $\mathrm{Fix}(f)$, $\mathrm{Pre}(f)$ and $\mathrm{Post}(f)$ form a complete lattice. For more details about the theorem and its impact, application possibilities, etc., we refer to [31, 122].

1.4 Algebraic structures

The central algebraic structure over which the main results of this dissertation will be given is Heyting algebra. However, we often give examples on other algebraic structures, and in the last chapter, we present some results on residuated lattices. Therefore, in this section, we provide a brief overview of the most important algebraic structures, the axioms that define them and an overview of some of the most interesting interactions between them. We refer to [39, 45, 55, 105, 129, 141] for a thorough approach.

The most general structure we mention is Monoidal Logic (ML) introduced by Höhle (cf. [66]) and it provides a broad framework for various nonclassical logics. Höhle ML is equivalent to FL_{ew}, i.e., Full Lambek calculus with exchange and weakening (see [55]) and $\mathrm{IPC}{*} \setminus c$ (Intuitionistic Propositional Calculus without contraction) (see [1]).

Some authors use the term *bounded integral commutative lattice*. However, the difference in these structures is insignificant for our study. So we will have the same approach as in [129] where ML algebra is identified with residuated lattices. Therefore, below we define the residuated lattice.

Definition 1.26. A *residuated lattice* is an algebra $\mathscr{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ where

(L1) $(L, \wedge, \vee, 0, 1)$ is a lattice with the least element 0 and the greatest element 1,

(L2) $(L, \otimes, 1)$ is a commutative monoid with the unit 1,

(L3) $\otimes$ and $\to$ form an *adjoint pair*, i.e., adjointness property

$$x \otimes y \leqslant z \quad \text{iff} \quad x \leqslant y \to z \tag{1.7}$$

holds for each $x, y, z \in L$, where $\leqslant$ denotes lattice ordering.

If, in addition, $(L, \wedge, \vee, 0, 1)$ is a complete lattice, then $\mathscr{L}$ is called a *complete residuated lattice*.

The operations $\otimes$ (called *multiplication*) and $\to$ (called *residuum*) are intended for modeling the conjunction and implication of the corresponding logical calculus, and *supremum* ($\bigvee$) and *infimum* ($\bigwedge$) are intended for modeling the existential and general quantifier, respectively.

On the complete residuated lattice the following operations can be defined:

$$\textit{biresiduum} \text{ (or } \textit{bi-implication}\text{)}: \ x \leftrightarrow y = (x \to y) \wedge (y \to x), \tag{1.8}$$

$$\textit{negation}: \ \neg x = x \to 0, \tag{1.9}$$

$$\textit{addition}: \ x \oplus y = \neg(\neg x \otimes \neg y), \tag{1.10}$$

$$n\textit{-fold multiplication}: \ x^n = \underbrace{x \otimes \cdots \otimes x}_{n\text{-times}}, \tag{1.11}$$

$$n\textit{-fold addition}: \ nx = \underbrace{x \oplus \cdots \oplus x}_{n\text{-times}}. \tag{1.12}$$

Bi-implication is the operation used for modeling the equivalence of truth values, whereas the negation is used for modeling the complement of a truth value.

Many varieties of ML-algebras, i.e., classes of algebras that are closed under homomorphisms, subalgebras, and direct products, can be obtained by including some additional axioms. Below, we give the list of the most prominent axioms used for refinements of ML-algebras.

(1) Pre-linearity condition: $(x \to y) \vee (y \to x) = 1$, (*Prl*)
(2) Involution: $x = \neg\neg x$, (*Inv*)
(3) Divisibility: $x \wedge y = x \otimes (x \to y)$, (*Div*)
(4) Law of pseudocomplementation: $x \wedge \neg x = 0$, (Π_1)
(5) Law of cancellativity: $\neg\neg z \leqslant ((x \otimes z) \to (y \otimes z)) \to (x \to y)$, ($\Pi_2$)
(6) Idempotency: $x \otimes x = x$, (*G*)
(7) Weak Nilpotent Minimum: $(\neg(x \otimes y)) \vee (x \wedge y \to x \otimes y) = 1$. (*WNM*)

Many authors use different names for some of these conditions, for example, *Dummett's condition* instead of Pre-linearity, *double negation* instead involution, etc. Now, using listed conditions, we define some notable extensions of a residuated lattice.

Definition 1.27. (1) A residuated lattice $\mathscr{L}$ is called an *MTL-algebra* (short for Monoidal t-norm Logic) if it satisfies the (*Prl*) condition.
(2) A residuated lattice $\mathscr{L}$ is called a *Heyting algebra* if it satisfies the (*Inv*) condition.
(3) An MTL-algebra $\mathscr{L}$ is called an *IMTL-algebra* (short for Involutive Monoidal t-norm Logic) if it satisfies the (*Inv*) condition.
(4) An MTL-algebra $\mathscr{L}$ is called a *BL-algebra* (short for Basic Logic Algebra) if it satisfies the (*Div*) condition.

Also, it is generally known that Heyting algebra can be equivalently defined as a residuated lattice $\mathscr{L}$ which satisfies the condition $x \otimes y = x \wedge y$. Moreover, if $\mathscr{L}$ is a complete lattice, than it is called a *complete Heyting algebra*. If the partial order $\leqslant$ in $\mathscr{L}$ is linear, then $\mathscr{L}$ is a *linearly ordered Heyting algebra*.

Now, we define three notable extensions of BL-algebras.

Definition 1.28. (1) A BL-algebra $\mathscr{L}$ is called an *MV-algebra* (short for Many-Valued Algebra) if it satisfies the (Inv) condition.

(2) A BL-algebra $\mathscr{L}$ is called a *Product algebra* if it satisfies (Π_1) and (Π_2) conditions.

(3) A BL-algebra $\mathscr{L}$ is called a *Gödel algebra* if it satisfies the (G) condition.

The MV-algebra is often called Łukasiewicz algebra or **Ł**-algebra while product algebra and Gödel algebra are abbreviated as **Π**-algebra and **G**-algebra, respectively.

Now, Boolean algebra can be defined in two equivalent ways.

Definition 1.29. (1) A Heyting algebra $\mathscr{L}$ is called *Boolean algebra* if it satisfies the (Inv) condition.

(2) An MV-algebra $\mathscr{L}$ is called *Boolean algebra* if it satisfies the (G) condition.

Hence, a Boolean algebra is a residuated lattice which is both Heyting algebra and an MV-algebra. Let us mention another important truth structure.

Definition 1.30. (1) An MTL-algebra $\mathscr{L}$ is called *WNM-algebra* (short for Weak Nilpotent Minimum algebra) if it satisfies the (WNM) condition.

(2) A WNM-algebra $\mathscr{L}$ is called *NM-algebra* (short for Nilpotent Minimum algebra) if it satisfies the (Inv) condition.

(3) An IMTL-algebra $\mathscr{L}$ is called *NM-algebra* if it satisfies the (WNM) condition.

Figure 1.3 shows all the truth structures we defined above and some other relationships between algebras that we did not mention. The Figure also shows the Affine Multiplicative Additive fragment of (propositional) Intuitionistic Linear logic (AMALL or AMAILL). Note that a special case of Boolean algebra can be obtained from Product algebra. The structures that we will deal with in this dissertation are painted in gray. We note once again that the vast majority of the presented results are given over Heyting algebras.

One of the most studied and applied many-valued systems are those corresponding to logical calculi defined over the real interval $[0, 1]$. These systems are induced by the function so-called *triangular norm*, and so we say they are *triangular norm based fuzzy logics*.

Definition 1.31. A *triangular norm* (abbreviated t-norm) is a binary operation $\otimes : [0, 1] \times [0, 1] \to [0, 1]$ satisfying the following conditions:

(1) $(x \otimes y) \otimes z = x \otimes (y \otimes z)$ (associativity)

(2) $x \otimes y = y \otimes x$ (commutativity)

(3) $y \leqslant z \Rightarrow x \otimes y \leqslant x \otimes z$ (monotonicity)

(4) $x \otimes 1 = x$ (1 is the neutral element)

A t-norm $\otimes$ is *left-continuous* if $\lim_{n\to\infty}(x_n \otimes y) = (\lim_{n\to\infty} x_n) \otimes y$ for any non-decreasing sequence $\{x_n \in [0, 1] \mid n = 1, 2, 3, \ldots\}$. Every left-continuous t-norm has a unique residuum operation $\to$, defined by

$$x \to y = \bigvee\{u \in [0, 1] \mid u \otimes x \leqslant y\}. \tag{1.13}$$

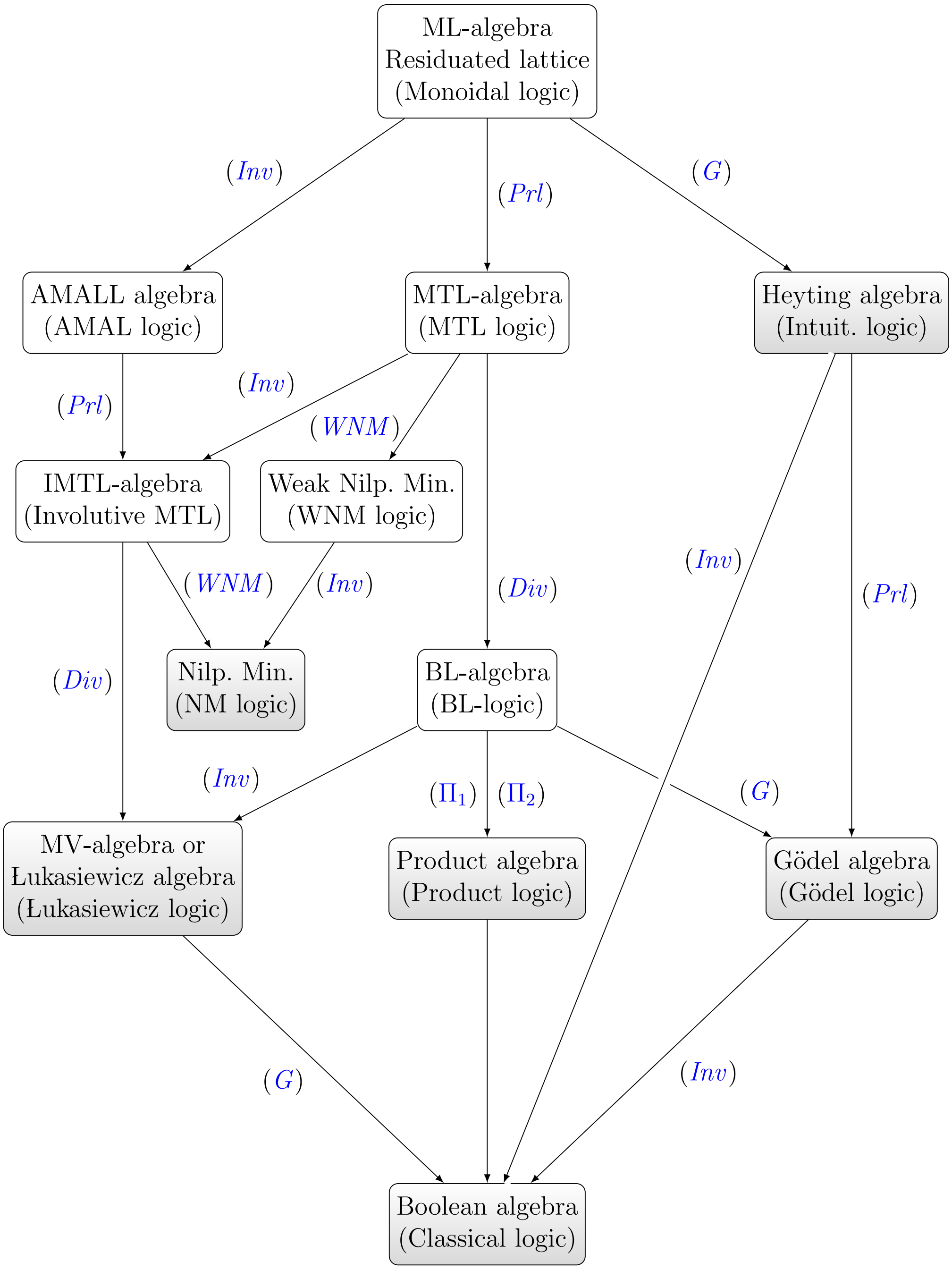

Figure 1.3: The most important truth structures and corresponding logic

The most studied and applied structures of truth values, defined on the real unit interval $[0, 1]$ with:

$$x \wedge y = \min(x, y) \qquad \text{and} \qquad x \vee y = \max(x, y)$$

are: the *Łukasiewicz structure*:

$$x \otimes y = \max(x + y - 1, 0), \quad x \to y = \min(1 - x + y, 1), \tag{1.14}$$

the *Goguen (product) structure*:

$$x \otimes y = x \cdot y, \quad x \to y = \begin{cases} 1, & \text{if } x \leqslant y, \\ x/y, & \text{otherwise,} \end{cases} \tag{1.15}$$

and the *Gödel structure*:

$$x \otimes y = \min(x, y), \quad x \to y = \begin{cases} 1, & \text{if } x \leqslant y, \\ y, & \text{otherwise.} \end{cases} \tag{1.16}$$

Another important set of truth values is the set $\{x_0, x_1, \dots, x_n\}$, $0 = x_0 < \dots < x_n = 1$, with

$$x_k \otimes x_l = x_{\max(k+l-n,0)} \qquad \text{and} \qquad x_k \to x_l = x_{\min(n-k+l,n)}.$$

A special case of the latter algebras is the two-element Boolean algebra of classical logic with the underlying structure $\{0, 1\}$. The only adjoint pair on the two-element Boolean algebra consists of the classical conjunction and implication operations. This structure of truth values we call the *Boolean structure*.

We also mention the *Nilpotent Minimum structure*:

$$x \otimes y = \begin{cases} \min(x, y), & \text{if } x + y > 1, \\ 0, & \text{otherwise,} \end{cases}, \; x \to y = \begin{cases} 1, & \text{if } x \leqslant y, \\ \max(1 - x, y), & \text{otherwise.} \end{cases} \tag{1.17}$$

Nilpotent minimum t-norm was introduced by Fodor in [53] as the first example of an involutive left-continuous but non-continuous t-norm. Later, *nilpotent minimum logic* was formalized by Esteva and Godo in [45]. Figure 1.3 does not show the subvarieties of Nilpotent Minimum algebra (and logics). Hence, for some finitary extensions of the Nilpotent Minimum Logic, we refer to [57].

Figures 1.4 and 1.5 graphically show the difference between the operations of t-norms and fuzzy implications on Gödel and Nilpotent Minimum structures.

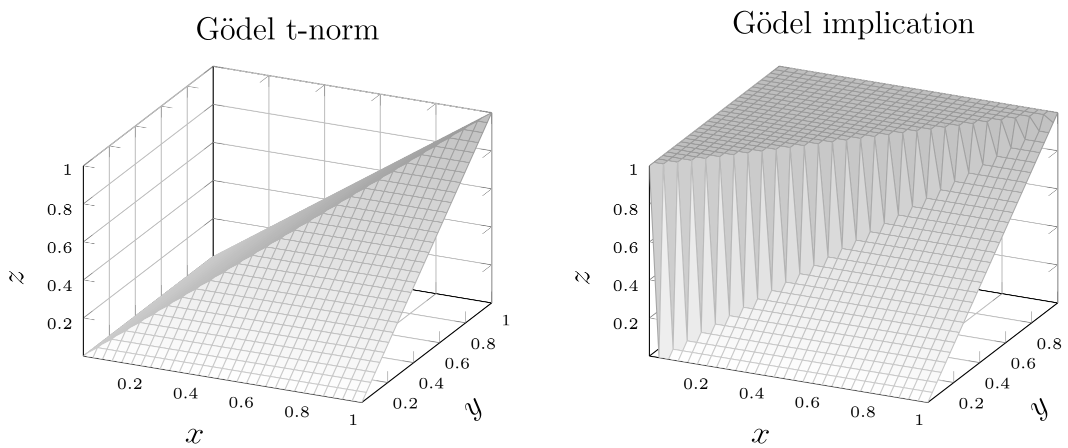

Figure 1.4: Gödel structure

If every finitely generated subalgebra of a residuated lattice $\mathscr{L}$ is finite, then $\mathscr{L}$ is called *locally finite*. For example, Gödel algebra, and hence, the Gödel structure, is locally finite, whereas the product structure is not locally finite.

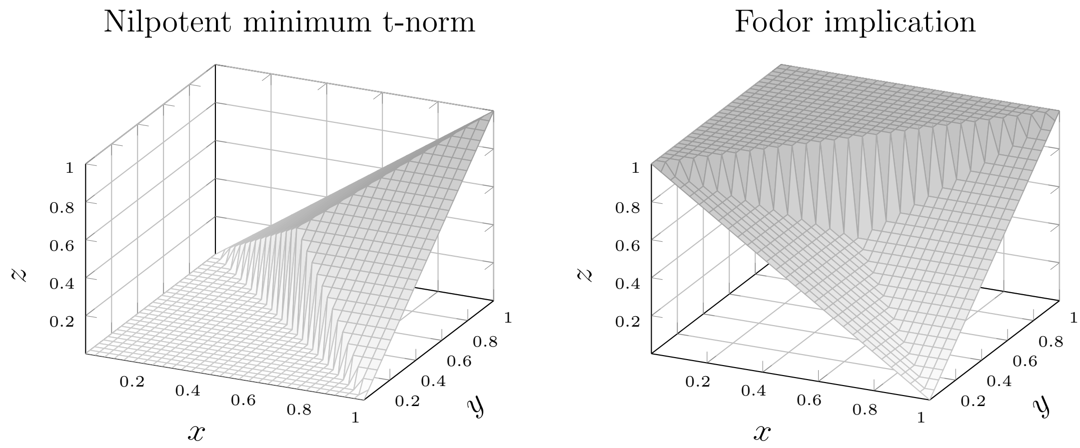

Figure 1.5: Nilpotent Minimum structure

1.5 Properties of complete residuated lattices

In this section, we recall basic properties of complete residuated lattices. For more information, we refer to the book of Bělohlávek and Vychodil [7].

Theorem 1.7. *In every complete residuated lattice, the following assertions hold:*

$$y \leqslant x \to (x \otimes y), \qquad x \leqslant (x \to y) \to y, \tag{1.18}$$

$$x \otimes (x \to y) \leqslant y, \tag{1.19}$$

$$x \leqslant y \quad \Leftrightarrow \quad x \to y = 1, \tag{1.20}$$

$$x \to x = 1, \qquad x \to 1 = 1, \qquad 1 \to x = x, \tag{1.21}$$

$$0 \to x = 1, \tag{1.22}$$

$$x \otimes 0 = 0 \otimes x = 0, \tag{1.23}$$

$$x \otimes y \leqslant x, \qquad x \leqslant y \to x, \tag{1.24}$$

$$x \otimes y \leqslant x \wedge y, \tag{1.25}$$

$$x \otimes y \to z = x \to (y \to z) = y \to (x \to z), \tag{1.26}$$

$$(x \to y) \otimes (y \to z) \leqslant (x \to z), \tag{1.27}$$

$$x \to y \textit{ is the greatest element of } \{z \mid x \otimes z \leqslant y\}, \tag{1.28}$$

$$x \otimes y \textit{ is the least element of } \{z \mid x \leqslant y \to z\}. \tag{1.29}$$

The following theorem indicates that $\otimes$ is an isotone operation in both arguments concerning order $\leqslant$, and the operation $\to$ is isotone in the second and antitone in the first argument.

Theorem 1.8. *In every complete residuated lattice, the following assertions hold:*

$$y_1 \leqslant y_2 \quad \Rightarrow \quad x \otimes y_1 \leqslant x \otimes y_2, \tag{1.30}$$

$$y_1 \leqslant y_2 \quad \Rightarrow \quad x \to y_1 \leqslant x \to y_2, \tag{1.31}$$

$$x_1 \leqslant x_2 \quad \Rightarrow \quad x_2 \to y \leqslant x_1 \to y. \tag{1.32}$$

$$\tag{1.33}$$

Some more properties of residuated lattices will be presented below.

Theorem 1.9. *In every complete residuated lattice, the following inequalities hold:*

$$x \to y \leqslant (x \wedge z) \to (y \wedge z), \tag{1.34}$$
$$x \to y \leqslant (x \vee z) \to (y \vee z), \tag{1.35}$$
$$x \to y \leqslant (x \otimes z) \to (y \otimes z), \tag{1.36}$$
$$x \to y \leqslant (y \to z) \to (x \to z), \tag{1.37}$$
$$x \to y \leqslant (z \to x) \to (z \to y). \tag{1.38}$$

The following theorem gives us a relationship between operations $\otimes$ and $\to$ and operations of supremum (join) and infimum (meet) of any number (possible infinite) of elements from a residuated lattice.

Theorem 1.10. *In every complete residuated lattice the following assertions hold, for every index set I:*

$$x \otimes \bigvee_{i\in I} y_i = \bigvee_{i\in I} (x \otimes y_i), \tag{1.39}$$
$$x \to \bigwedge_{i\in I} y_i = \bigwedge_{i\in I} (x \to y_i), \tag{1.40}$$
$$\left(\bigvee_{i\in I} x_i\right) \to y = \bigwedge_{i\in I} (x_i \to y), \tag{1.41}$$
$$x \otimes \bigwedge_{i\in I} y_i \leqslant \bigwedge_{i\in I} (x \otimes y_i), \tag{1.42}$$
$$\bigvee_{i\in I} (x \to y_i) \leqslant x \to \bigvee_{i\in I} y_i, \tag{1.43}$$
$$\bigvee_{i\in I} (x_i \to y) \leqslant \left(\bigwedge_{i\in I} x_i\right) \to y, \tag{1.44}$$
$$\bigvee_{i\in I} (x_i \to y_i) \leqslant \left(\bigwedge_{i\in I} x_i\right) \to \left(\bigwedge_{i\in I} y_i\right). \tag{1.45}$$

The following theorem gives us some basic properties of negation. Note that many properties of negation are actually special cases of the already mentioned properties.

Theorem 1.11. *In every complete residuated lattice, the following assertions hold:*

$$\neg 0 = 1, \quad \neg 1 = 0, \tag{1.46}$$
$$x \otimes \neg x = 0, \tag{1.47}$$
$$x \leqslant \neg\neg x, \quad \neg x = \neg\neg\neg x, \tag{1.48}$$
$$x \leqslant y \quad \Rightarrow \quad \neg y \leqslant \neg x, \tag{1.49}$$
$$\neg \left(\bigvee_{i\in I} x_i\right) = \bigwedge_{i\in I} \neg x_i, \tag{1.50}$$
$$\neg \left(\bigwedge_{i\in I} x_i\right) \geqslant \bigvee_{i\in I} \neg x_i. \tag{1.51}$$

The following theorem gives us some properties of biresiduum.

Theorem 1.12. *In every complete residuated lattice, the following assertions hold:*

$$0 \leftrightarrow 1 = 1 \leftrightarrow 0 = 0,\ 0 \leftrightarrow 0 = 1 \leftrightarrow 1 = 1, \tag{1.52}$$
$$x \leftrightarrow x = 1, \tag{1.53}$$
$$x \leftrightarrow y = y \leftrightarrow x, \tag{1.54}$$
$$(x \leftrightarrow y) \otimes (y \leftrightarrow z) \leqslant x \leftrightarrow z, \tag{1.55}$$
$$x \leftrightarrow 1 = x, \quad x \leftrightarrow 0 = \neg x, \tag{1.56}$$
$$x \leftrightarrow y = 1 \quad \Leftrightarrow \quad x = y, \tag{1.57}$$
$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \wedge x_2) \leftrightarrow (y_1 \wedge y_2), \tag{1.58}$$
$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \vee x_2) \leftrightarrow (y_1 \vee y_2), \tag{1.59}$$
$$(x_1 \leftrightarrow y_1) \otimes (x_2 \leftrightarrow y_2) \leqslant (x_1 \otimes x_2) \leftrightarrow (y_1 \otimes y_2), \tag{1.60}$$
$$(x_1 \leftrightarrow y_1) \otimes (x_2 \leftrightarrow y_2) \leqslant (x_1 \to x_2) \leftrightarrow (y_1 \to y_2), \tag{1.61}$$
$$\bigwedge_{i\in I}(x_i \leftrightarrow y_i) \leqslant \left(\bigwedge_{i\in I} x_i\right) \leftrightarrow \left(\bigwedge_{i\in I} y_i\right), \tag{1.62}$$
$$\bigwedge_{i\in I}(x_i \leftrightarrow y_i) \leqslant \left(\bigvee_{i\in I} x_i\right) \leftrightarrow \left(\bigvee_{i\in I} y_i\right), \tag{1.63}$$
$$x \leftrightarrow y = (x \vee y) \to (x \wedge y). \tag{1.64}$$

At the end of this chapter we state the important lemma from [28]:

Lemma 1.1. *Let $\mathscr{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ be a complete residuated lattice satisfying condition*

$$\bigwedge_{i\in I}(x \vee y_i) = x \vee \left(\bigwedge_{i\in I} y_i\right), \tag{1.65}$$

for all $x \in L$ and $\{y_i\}_{i\in I} \subseteq L$. Then for all non-increasing sequences $\{x_k\}_{k\in\mathbb{N}}$, $\{y_k\}_{k\in\mathbb{N}} \subseteq L$ we have

$$\bigwedge_{k\in\mathbb{N}}(x_k \vee y_k) = \left(\bigwedge_{k\in\mathbb{N}} x_k\right) \vee \left(\bigwedge_{k\in\mathbb{N}} y_k\right). \tag{1.66}$$

1.6 Heyting algebras

A Dutch mathematician Luitzen Brouwer founded the mathematical philosophy of intuitionism in the early 20th century. His student Arend Heyting developed formal systems to provide a formal basis for Brouwer's programme in 1930 (cf. [65]). The algebras thus obtained are called *Heyting algebras*. Instead of Heyting algebra, certain authors used the term *pseudo-Boolean algebra* or *relatively pseudocomplemented distributive lattice with* 0 (for example, see [120]), and *Brouwerian algebras* for algebraic duals of Heyting algebras (see [88]). For more information about Heyting algebra see [4, 12].

In Section 1.4, we defined Heyting algebra as a residuated lattice that satisfies the condition $x \otimes y = x \wedge y$, but now, we will give another definition of a Heyting algebra.

Definition 1.32. An algebra $\mathscr{H} = (H, \wedge, \vee, \rightarrow, 0, 1)$ with three binary and two nullary operations is a *Heyting algebra* if it satisfies:

(H1) $(H, \wedge, \vee)$ is a distributive lattice;
(H2) $x \wedge 0 = 0, \quad x \vee 1 = 1$;
(H3) $x \rightarrow x = 1$;
(H4) $(x \rightarrow y) \wedge y = y, \quad x \wedge (x \rightarrow y) = x \wedge y$;
(H5) $x \rightarrow (y \wedge z) = (x \rightarrow y) \wedge (x \rightarrow z), \quad (x \vee y) \rightarrow z = (x \rightarrow z) \wedge (y \rightarrow z)$.

By simply checking the conditions, we can see that these two definitions of Heyting algebras are equivalent. Similar to residuated lattices, the binary operation $\rightarrow$ is called *relative pseudocomplementation*, or *residuum*, in many sources. The *relative pseudocomplement* $x \rightarrow z$ *of* x *with respect to* z can be characterized as follows:

$$x \rightarrow z = \bigvee \{y \in H \mid x \wedge y \leqslant z\}. \tag{1.67}$$

Equivalently, we say that operations $\wedge$ and $\rightarrow$ form an *adjoint pair*, i.e., they satisfy the *adjunction property* or *residuation property*: for all $x, y, z \in H$,

$$x \wedge y \leqslant z \qquad \Leftrightarrow \qquad x \leqslant y \rightarrow z. \tag{1.68}$$

If, in addition, $(H, \wedge, \vee, 0, 1)$ is a complete lattice, then $\mathscr{H}$ is called a *complete Heyting algebra*. In the rest of the paper, unless otherwise stated, $\mathscr{H} = (H, \wedge, \vee, \rightarrow, 0, 1)$ stands for a complete Heyting algebra. Operations $\bigvee$, $\bigwedge$ and $\leftrightarrow$ are the same as for residuated lattices.

The following lemma gives some basic properties of Heyting algebras. Note that many properties are special cases of residuated lattices.

Lemma 1.2. *In every complete Heyting algebra* $\mathscr{H} = (H, \wedge, \vee, \rightarrow, 0, 1)$*, the following assertions hold:*

$$x \wedge (y \rightarrow z) = x \wedge (x \wedge y \rightarrow z), \tag{1.69}$$
$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \wedge x_2) \leftrightarrow (y_1 \wedge y_2), \tag{1.70}$$
$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \vee x_2) \leftrightarrow (y_1 \vee y_2), \tag{1.71}$$
$$(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \rightarrow x_2) \leftrightarrow (y_1 \rightarrow y_2), \tag{1.72}$$
$$x \wedge \left(\bigwedge_{i \in I} y_i \right) = \bigwedge_{i \in I} (x \wedge y_i). \tag{1.73}$$

Lemma 1.3. *Let* $\mathscr{L} = (L, \wedge, \vee, \otimes, \rightarrow, 0, 1)$ *be a complete residuated lattice. Then, condition*

$$x \otimes y = x \leftrightarrow y, \text{ for all } x, y \in L \text{ such that } x \neq y, \tag{1.74}$$

is fulfilled if and only if $\mathscr{L}$ *is linearly ordered Heyting algebra.*

Proof. First, let condition (1.74) hold. To prove that $\mathscr{L}$ is linearly ordered Heyting algebra, it is necessary to prove that for all $x, y \in L$ is fulfilled:

$$x \otimes y = x \qquad \text{or} \qquad x \otimes y = y. \tag{1.75}$$

So, let us suppose that $x \otimes y \neq x$. Then, due to (1.25) it follows

$$x \otimes y \leqslant x \wedge y \leqslant x, \, y,$$

and, using (1.20), (1.74) and (1.18) we have

$$\begin{aligned} x \otimes y &= (x \otimes y) \wedge x = (x \otimes y) \leftrightarrow x \\ &= ((x \otimes y) \to x) \wedge (x \to (x \otimes y)) \\ &= x \to (x \otimes y) \geqslant y, \end{aligned}$$

and we can conclude $x \otimes y = y$.

Conversely, let us suppose that $\mathscr{L}$ is linearly ordered Heyting algebra, and let $x, y \in L$ be two different arbitrary elements. If $x < y$, then we have $x \otimes y = x \wedge y = x$ and using (1.20) and (1.18) it follows

$$\begin{aligned} x \leftrightarrow y &= (x \to y) \wedge (y \to x) \\ &= y \to x = y \to (y \otimes x) \\ &\geqslant x = x \otimes y, \end{aligned}$$

and from (1.25) we conclude $x \leftrightarrow y = x \otimes y$. In the same manner, we can prove that from $y < x$ it follows $x \leftrightarrow y = x \otimes y$, which finishes the proof. □

Hence, in a complete, linearly ordered Heyting algebra $\mathscr{H} = (H, \wedge, \vee, \to, 0, 1)$ the following holds:

$$x \wedge y = x \leftrightarrow y, \text{ for all } x, y \in H \text{ such that } x \neq y. \tag{1.76}$$

Lemma 1.1 can be formulated for Heyting algebras.

Lemma 1.4. *Let $\mathscr{H} = (H, \wedge, \vee, \to, 0, 1)$ be a complete Heyting algebra satisfying condition*

$$\bigwedge_{i \in I} (x \vee y_i) = x \vee \left(\bigwedge_{i \in I} y_i \right), \tag{1.77}$$

for all $x \in H$ and $\{y_i\}_{i \in I} \subseteq H$. Then for all non-increasing sequences $\{x_k\}_{k \in \mathbb{N}}$, $\{y_k\}_{k \in \mathbb{N}} \subseteq H$ we have

$$\bigwedge_{k \in \mathbb{N}} (x_k \vee y_k) = \left(\bigwedge_{k \in \mathbb{N}} x_k \right) \vee \left(\bigwedge_{k \in \mathbb{N}} y_k \right), \tag{1.78}$$

and this can be generalized for all non-increasing sequences $\{x_k^j\}_{k \in \mathbb{N}} \subseteq H$, $j \in J$, in the following way:

$$\bigwedge_{k \in \mathbb{N}} \bigvee_{j \in J} x_k^j = \bigvee_{j \in J} \bigwedge_{k \in \mathbb{N}} x_k^j, \tag{1.79}$$

where J is a finite set of indices.

Below are some interesting examples of Heyting algebras.

Example 1.4. Every finite distributive lattice $\mathscr{L} = (L, \wedge, \vee, \to, 0, 1)$ is Heyting algebra where operation of residuum is defined as usual (1.67).

Example 1.5. Every bounded chain $\mathscr{L} = (L, \wedge, \vee, \to, 0, 1)$ is Heyting algebra, with

$$x \to y = \begin{cases} 1 & \text{if } x \leq y, \\ y & \text{if } x > y. \end{cases}$$

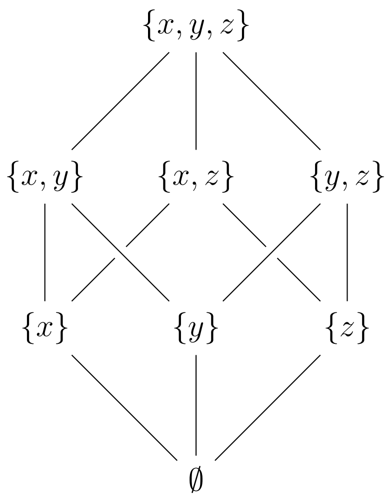

Figure 1.6: The elements of the power set of the set $\{x, y, z\}$ ordered with respect to inclusion

Example 1.6. Every Boolean algebra $\mathscr{B} = (B, \wedge, \vee, \rightarrow, 0, 1)$ is Heyting algebra with $x \rightarrow y = x' \vee y$.

Example 1.7. According to the previous example, the power set algebra of X, i.e. $(\mathcal{P}(X), \cap, \cup, \rightarrow, \emptyset, X)$ is a Boolean algebra, and therefore Heyting algebra. Operation of residuum is defined:

$$A \rightarrow B = A^c \cup B. \tag{1.80}$$

Let consider power set of $X = \{x, y, z\}$. If $A = \{x\}$ and $B = \{y\}$, then $A \rightarrow B = \{y, z\}$.

However, this algebra is not completely linearly ordered with respect to set inclusion because, for example, neither $\{x\} \subseteq \{y\}$ nor $\{y\} \subseteq \{x\}$. The partial order can be seen in Figure 1.6.

Hence, a Heyting algebra is a Boolean algebra if and only if $\neg\neg x = x$ for all x. Also, in Boolean algebra is valid $x \vee \neg x = 1$ while in Heyting algebra such equality does not have to be valid, as the following examples will show.

Example 1.8. The simplest Gödel (and Heyting) algebra which is not Boolean algebra is the structure $\mathscr{G} = (G, \wedge, \vee, \rightarrow, 0, 1)$ with linearly ordered set $G = \{0, \frac{1}{2}, 1\}$ and defined operations:

$\wedge$	0	$\frac{1}{2}$	1
0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
1	0	$\frac{1}{2}$	1

$\vee$	0	$\frac{1}{2}$	1
0	0	$\frac{1}{2}$	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	1
1	1	1	1

$\rightarrow$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	0	1	1
1	0	$\frac{1}{2}$	1

In this algebra law of double negation $\neg\neg x = x$ and law of excluded middle $x \vee \neg x = 1$ do not hold. For example:

$$\neg\neg\frac{1}{2} = \left(\frac{1}{2} \rightarrow 0\right) \rightarrow 0 = 0 \rightarrow 0 = 1$$
$$\frac{1}{2} \vee \neg\frac{1}{2} = \frac{1}{2} \vee 0 = \frac{1}{2}.$$

For the examples below, we need the following definition.

Definition 1.33. Let X be a set. A set τ of subsets of X is called a *topology* if the following properties are satisfied:

(1) $\emptyset, X \in \tau$,
(2) if $\{A_i \mid i \in I\} \subseteq \tau$ then $\bigcup_{i\in I} A_i \in \tau$,
(3) if $A, B \in \tau$ then $A \cap B \in \tau$.

The ordered pair (X, τ) is called a *topological space*.

Hence, topology τ contains an empty set and X and is closed under arbitrary unions and finite intersections.

Example 1.9 (Standard topology of $\mathbb{R}$)**.** Let $\mathbb{R}$ be the set of all real numbers. Let B be the collection of all open intervals

$$(x, y) = \{a \in \mathbb{R} \mid x < a < y\}.$$

Then, $\mathscr{B} = (B, \cap, \cup, \rightarrow, \emptyset, \mathbb{R})$ is a Heyting algebra with the implication

$$A \rightarrow B = int(A^c \cup B).$$

For example, let $A = (0, 1)$. Then, $\neg A = A \rightarrow \emptyset = int(A^c)$, i.e. $\neg A = (-\infty, 0) \cup (1, +\infty)$. Hence, $A \cup \neg A = \mathbb{R} \setminus \{0, 1\} \subseteq \mathbb{R}$.

Definition 1.34. Let (X, τ) be a topological space. If $A \subseteq X$ is such that $A \in \tau$ then A is said to be *Open*. A subset $A \subseteq X$ is said to be *Closed* if $A^c = X \setminus A$ is open. If $A \subseteq X$ are both open and closed, then A is said to be *Clopen*.

Example 1.10. Let $X = \{a, b, c, d\}$ and consider the topology $\tau = \{\emptyset, \{c\}, \{a, b\}, \{c, d\}, \{a, b, c\}, X\}$ (see Figure 1.7). The open sets of X are those sets forming τ, the closed sets of X are the complements of all the open sets and the clopen sets of X are the sets that are both open and closed:

$$\begin{aligned}
&\text{open sets of } X = \{\emptyset, \{c\}, \{a, b\}, \{c, d\}, \{a, b, c\}, X\},\\
&\text{closed sets of } X = \{\emptyset, \{a, b, d\}, \{c, d\}, \{a, b\}, \{d\}, X\},\\
&\text{clopen sets of } X = \{\emptyset, \{a, b\}, \{c, d\}, X\}.
\end{aligned}$$

The operation of residuum is defined in the following way:

$$U \rightarrow V = int(U^c \cup V).$$

For example, let $U = \{a, b, c\}$ and $V = \{a, b\}$, then $U \rightarrow V = int(\{a, b, d\}) = \{a, b\}$.

Hence, the open sets of any topological space X form a Heyting algebra $(\tau, \cap, \cup, \rightarrow, \emptyset, X)$, and obviously, such algebra is not linearly ordered.

Finite sets can have many topologies on them. Consider the set $X = \{a, b, c, d\}$ and the *nested topology* $\tau = \{\emptyset, \{a\}, \{a, b\}, \{a, b, c\}, \{a, b, c, d\}\}$. Such topological space is linearly ordered Heyting algebra.

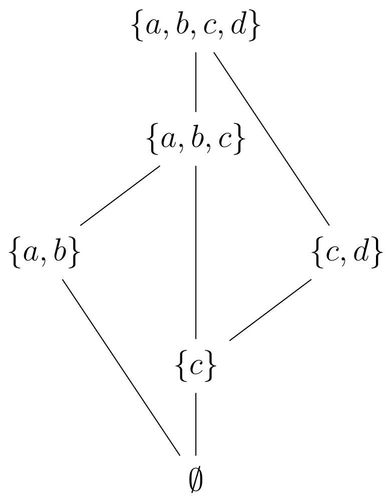

Figure 1.7: Topology $\tau = \{\emptyset, \{c\}, \{a,b\}, \{c,d\}, \{a,b,c\}, X\}$

1.7 Fuzzy sets and fuzzy relations

The terminology and basic notions given in this section are according to [5, 7], but we set them up for a Heyting algebra. For more information about fuzzy logic and its principles, we refer to the book [105].

Definition 1.35. A *fuzzy subset* of a set A *over* $\mathscr{H}$, or simply a *fuzzy subset* of A is any function from A to H. Ordinary crisp subsets of A are considered as fuzzy subsets of A taking membership values in the set $\{0,1\} \subseteq H$.

Let f and g be two fuzzy subsets of A. The *equality* of f and g is defined as the usual equality of functions, i.e., $f = g$ if and only if $f(x) = g(x)$, for every $x \in A$. The *inclusion* $f \leqslant g$ is also defined pointwise: $f \leqslant g$ if and only if $f(x) \leqslant g(x)$, for every $x \in A$. With this partial order, the set $\mathscr{F}(A)$ of all fuzzy subsets of A forms a complete Heyting algebra, in which the meet (intersection) $\bigwedge_{i \in I} f_i$ and the join (union) $\bigvee_{i \in I} f_i$ of an arbitrary family $\{f_i\}_{i \in I}$ of fuzzy subsets of A are functions from A to H defined by

$$\left(\bigwedge_{i \in I} f_i\right)(x) = \bigwedge_{i \in I} f_i(x), \qquad \left(\bigvee_{i \in I} f_i\right)(x) = \bigvee_{i \in I} f_i(x).$$

Note that the equality, inclusion, meet and join of fuzzy sets are all defined pointwise. The *product* $f \wedge g$ is the same as the binary meet: $f \wedge g(x) = f(x) \wedge g(x)$, for every $x \in A$ due to the relationship between Heyting algebra and a residuated lattice.

The *crisp part* of fuzzy subset f of A is a crisp subset $\hat{f} = \{a \in A \mid f(a) = 1\}$ of A. We will also consider $\hat{f}$ as a function $\hat{f} : A \to H$ defined by $\hat{f}(a) = 1$, if $f(a) = 1$, and $\hat{f}(a) = 0$, if $f(a) < 1$.

Definition 1.36. Let A and B be non-empty sets. A *fuzzy relation between sets* A and B (in this order) is any function from $A \times B$ to H, i.e., any fuzzy subset of $A \times B$, and the equality, inclusion (ordering), joins and meets of fuzzy relations are defined as for fuzzy sets.

In particular, a *fuzzy relation on a set* A is any function from $A \times A$ to H, i.e., any fuzzy subset of $A \times A$. The set of all fuzzy relations from A to B will be denoted by $\mathscr{R}(A,B)$, and the set of all fuzzy relations on a set A will be denoted by $\mathscr{R}(A)$. The *inverse* of a fuzzy relation $\varphi \in \mathscr{R}(A,B)$ is a fuzzy relation $\varphi^{-1} \in \mathscr{R}(B,A)$ defined by $\varphi^{-1}(b,a) = \varphi(a,b)$, for all $a \in A$ and $b \in B$. A *crisp relation* is a fuzzy relation which takes values only in the set $\{0,1\}$, and if φ is a crisp relation of A to B, then expressions "$\varphi(a,b) = 1$" and "$(a,b) \in \varphi$" will have the same meaning.

Definition 1.37. For non-empty sets A, B and C, and fuzzy relations $\varphi \in \mathscr{R}(A,B)$ and $\psi \in \mathscr{R}(B,C)$, their *composition* $\varphi \circ \psi$ is a fuzzy relation from $\mathscr{R}(A,C)$ defined by

$$(\varphi \circ \psi)(a,c) = \bigvee_{b\in B} \varphi(a,b) \wedge \psi(b,c), \tag{1.81}$$

for all $a \in A$ and $c \in C$.

If φ and ψ are crisp relations, then $\varphi \circ \psi$ is an ordinary composition of relations in the sense of Definition 1.3. Moreover, if φ and ψ are functions, then $\varphi \circ \psi$ is an ordinary composition of functions, i.e., $(\varphi \circ \psi)(a) = \psi(\varphi(a))$, for every $a \in A$.

Definition 1.38. Let $f \in \mathscr{F}(A)$, $\varphi \in \mathscr{R}(A,B)$ and $g \in \mathscr{F}(B)$, the compositions $f \circ \varphi$ and $\varphi \circ g$ are fuzzy subsets of B and A, respectively, which are defined by

$$(f \circ \varphi)(b) = \bigvee_{a\in A} f(a) \wedge \varphi(a,b), \qquad (\varphi \circ g)(a) = \bigvee_{b\in B} \varphi(a,b) \wedge g(b), \tag{1.82}$$

for every $a \in A$ and $b \in B$.

Let $f, g \in \mathscr{F}(A)$. The composition $f \circ g$ is an element of a fuzzy set A, defined by

$$f \circ g = \bigvee_{a\in A} f(a) \wedge g(a). \tag{1.83}$$

The value $f \circ g$ can be interpreted as the "degree of overlapping" of f and g. In particular, if f and g are crisp sets and φ is a crisp relation, then

$$f \circ \varphi = \{b \in B \mid (\exists a \in f)(a,b) \in \varphi\}, \qquad \varphi \circ g = \{a \in A \mid (\exists b \in g)(a,b) \in \varphi\}.$$

The following lemmas give the basic properties of the composition of fuzzy relations and fuzzy subsets.

Lemma 1.5. *Let A, B, C and D be non-empty sets. Then we have:*

(1) *For any $\varphi_1 \in \mathscr{R}(A,B)$, $\varphi_2 \in \mathscr{R}(B,C)$ and $\varphi_3 \in \mathscr{R}(C,D)$ we have*

$$(\varphi_1 \circ \varphi_2) \circ \varphi_3 = \varphi_1 \circ (\varphi_2 \circ \varphi_3). \tag{1.84}$$

(2) *For $\varphi_0 \in \mathscr{R}(A,B)$, $\varphi_1, \varphi_2 \in \mathscr{R}(B,C)$ and $\varphi_3 \in \mathscr{R}(C,D)$ we have that $\varphi_1 \leqslant \varphi_2$ implies*

$$\varphi_1^{-1} \leqslant \varphi_2^{-1}, \qquad \varphi_0 \circ \varphi_1 \leqslant \varphi_0 \circ \varphi_2 \qquad \textit{and} \qquad \varphi_1 \circ \varphi_3 \leqslant \varphi_2 \circ \varphi_3.$$

(3) *For any $\varphi \in \mathscr{R}(A,B)$, $\psi \in \mathscr{R}(B,C)$, $f \in \mathscr{F}(A)$, $g \in \mathscr{F}(B)$ and $h \in \mathscr{F}(C)$ the following holds:*

$$(f \circ \varphi) \circ \psi = f \circ (\phi \circ \psi), \quad (f \circ \varphi) \circ g = f \circ (\varphi \circ g), \quad (\varphi \circ \psi) \circ h = \varphi \circ (\psi \circ h). \tag{1.85}$$

Consequently, the parentheses in (1.85) can be omitted, as well as the parentheses in (1.84).

Lemma 1.6. *For all $\varphi, \varphi_i \in \mathscr{R}(A,B)(i \in I)$ and $\psi, \psi_i \in \mathscr{R}(B,C)(i \in I)$ we have that*

(1) $(\varphi \circ \psi)^{-1} = \psi^{-1} \circ \varphi^{-1}$;

(2) $\varphi \circ \left(\bigvee_{i\in I} \psi_i\right) = \bigvee_{i\in I}(\varphi \circ \psi_i)$;

(3) $\left(\bigvee_{i\in I} \varphi_i\right) \circ \psi = \bigvee_{i\in I}(\varphi_i \circ \psi)$;

(4) $\left(\bigvee_{i\in I} \varphi_i\right)^{-1} = \bigvee_{i\in I} \varphi_i^{-1}$.

Definition 1.39. Let A and B be fuzzy sets. A fuzzy relation $\varphi \in \mathscr{R}(A,B)$ is called *image-finite* if for every $a \in A$ the set $\{b \in B \mid \varphi(a,b) > 0\}$ is finite, it is called *domain-finite* if for every $b \in B$ the set $\{a \in A \mid \varphi(a,b) > 0\}$ is finite, and it is called *degree-finite* if it is both image-finite and domain finite.

We note that if A, B and C are finite sets of cardinality $|A| = k$, $|B| = m$ and $|C| = n$, then $\varphi \in \mathscr{R}(A,B)$ and $\psi \in \mathscr{R}(B,C)$ can be treated as $k \times m$ and $m \times n$ fuzzy matrices over $\mathscr{H}$, and $\varphi \circ \psi$ is the matrix product. Analogously, for $f \in \mathscr{F}(A)$ and $g \in \mathscr{F}(B)$ we can treat $f \circ \varphi$ as the product of a $1 \times k$ matrix f and a $k \times m$ matrix φ, and $\varphi \circ g$ as the product of a $k \times m$ matrix φ and an $m \times 1$ matrix g^t (the transpose of g).

A fuzzy relation R on A is said to be:

- *reflexive* (or *fuzzy reflexive*) if $R(a,a) = 1$, for every $a \in A$;
- *symmetric* (or *fuzzy symmetric*) if $R(a,b) = R(b,a)$, for all $a, b \in A$;
- *transitive* (or *fuzzy transitive*) if $R(a,b) \wedge R(b,c) \leqslant R(a,c)$, for all $a, b, c \in A$.

It can be easily shown that $R \circ R = R$ holds for any reflexive and transitive relation R on A.

For a fuzzy relation R on the set A, the fuzzy relation R^∞ on A defined by

$$R^\infty = \bigvee_{n\in\mathbb{N}} R^n \tag{1.86}$$

is the least transitive fuzzy relation on A containing R, and it is called *transitive closure* of R.

A reflexive, symmetric and transitive fuzzy relation on A is called a *fuzzy equivalence*. With the respect to the inclusion of fuzzy relations, the set $\mathscr{E}(A)$ of all fuzzy equivalences on A is a complete lattice, in which the infimum coincides with the ordinary intersection of fuzzy relations, but in the general case, the supremum in $\mathscr{E}(A)$ does not coincide with the ordinary union of fuzzy relations (see Example 1.3).

A fuzzy equivalence E on a set A is called *fuzzy equality* if $E(a,b) = 1$ implies $a = b$, for all $a, b \in A$. In other words, E is fuzzy equality if and only if its crisp part $\widehat{E}$ is crisp equality.

Definition 1.40. Let E be a fuzzy equivalence on A and a be an arbitrary element from A. Then, the *equivalence class* of fuzzy relation E on A determined by $a \in A$ is the fuzzy subset denoted E_a (or $[a]_E$) of A defined by

$$E_a(b) = E(a,b), \qquad \text{for every } b \in A.$$

The set of all equivalence classes of A is denoted by $A/E = \{E_a \mid a \in A\}$ called *factor set* (or *quotient set*). The *natural function* from A to A/E is the fuzzy relation $\varphi_E \in \mathscr{R}(A, A/E)$ defined by

$$\varphi_E(a, E_b) = E(a, b), \qquad \text{for all } a, b \in A. \tag{1.87}$$

There are several approaches to how ordinary homomorphism can be generalized to fuzzy homomorphism (for example, see [5] and [67]) and therefore the homomorphism theorems can also be generalized in the fuzzy case according to the approach. However, we will not deal with this, we will only state what is essential for our work.

A reflexive and transitive fuzzy relation on a set A is called a *fuzzy quasi-order*, and a reflexive and transitive crisp relation on A is called a *quasi-order*. Similarly like set $\mathscr{E}(A)$, the set $\mathscr{Q}(A)$ of all fuzzy quasi-orders on A is a complete lattice, in which the infimum coincide with the ordinary intersection of fuzzy relations. However, in the general case, the supremum in $\mathscr{Q}(A)$ does not coincide with the ordinary union of fuzzy relations. Namely, if R is the supremum in $\mathscr{Q}(A)$ of a family $\{R_i \mid i \in I\}$ of fuzzy quasi-orders on A, then using (1.86) R can be presented by:

$$R = \left(\bigvee_{i\in I} R_i\right)^{\infty} = \bigvee_{n\in\mathbb{N}} \left(\bigvee_{i\in I} R_i\right)^{n}.$$

Definition 1.41. The *R-afterset* of a, $a \in A$, is the fuzzy set R_a defined by:

$$R_a(b) = R(a, b), \qquad \text{for every } b \in A, \tag{1.88}$$

while the *R-foreset* of a is the fuzzy set R^a defined by:

$$R^a(b) = R(b, a), \qquad \text{for every } b \in A. \tag{1.89}$$

The set of all R-aftersets will be denoted by A/R, and the set of all R-foresets will be denoted by $A\backslash R$. If R is a fuzzy equivalence, then $A/R = A\backslash R$ is the set of all equivalence classes of R.

For a fuzzy quasi-order R on a set A, a fuzzy relation E_R defined by $E_R = R\wedge R^{-1}$ is a fuzzy equivalence on A, which is called a *natural fuzzy equivalence* of R.

A fuzzy quasi-order R on a set A is a *fuzzy order* if $R(a, b) = R(b, a) = 1$ implies $a = b$, for all $a, b \in A$, i.e., if the natural fuzzy equivalence E_R of R is a fuzzy equality. A fuzzy quasi-order R is a fuzzy order if and only if its crisp part $\widehat{R}$ is a crisp order.

If f is an arbitrary fuzzy subset of A, then fuzzy relations R_f and R^f on A defined by

$$R_f(a, b) = f(a) \to f(b), \qquad R^f(a, b) = f(b) \to f(a), \tag{1.90}$$

for all $a, b \in A$, are fuzzy quasi-orders on A.

Also, for arbitrary fuzzy subset f on A, the fuzzy relation E_f defined by

$$E_f(a, b) = f(a) \leftrightarrow f(b), \qquad \text{for all } a, b \in A, \tag{1.91}$$

is a fuzzy equivalence on A.

The following theorem was proved in [133] (see also [70]). Theorem recalls some important features of quasi orders and natural equivalences.

Theorem 1.13. *Let R be a fuzzy quasi-order on a set A and E the natural fuzzy equivalence of R. Then*

(a) *For arbitrary $a, b \in A$ the following conditions are equivalent:*
  (i) $E(a,b) = 1$;
  (ii) $E_a = E_b$;
  (iii) $R^a = R^b$;
  (iv) $R_a = R_b$.

(b) *Functions $R_a \mapsto E_a$ of A/R to A/E, and $R_a \mapsto R^a$ of A/R to $A\backslash R$ are bijective functions.*

If A is a finite set with cardinality n, then a fuzzy quasi-order R on A is viewed as an $n \times n$ fuzzy matrix with entries in $\mathscr{H}$ (it is usually identified with that matrix, which is called a *fuzzy quasi-order matrix*). In that case R-aftersets are row vectors, whereas R-foresets are column vectors of this matrix. The previous theorem says that the ith and jth row vectors of this matrix are equal if and only if its ith and jth column vectors are equal, and vice versa. Moreover, we have that a fuzzy quasi-order R is a fuzzy order if and only if all its row vectors are different, or equivalently, if and only if all its column vectors are different.

In the continuation of the section, we will deal with the *block representation* of the fuzzy sets and the fuzzy relations. This way of representation can be found in converting two-mode to one-mode fuzzy relational system (for example, see [27]).

If the set A is presented as $A = D \cup E$, where $D \cap E = \emptyset$ then we say that A is a *disjoint union* of sets D and E and denote $A = D \sqcup E$.

For a fuzzy subset $f \in \mathscr{F}(A)$ and $X \subseteq A$, by f_X we denote the restriction of f to X. If the set A is represented as $A = D \sqcup E$, then the expression

$$f = \begin{bmatrix} f_D \\ f_E \end{bmatrix} \tag{1.92}$$

is called the *block representation* of f with blocks f_D and f_E.

For a fuzzy relation $R \in \mathscr{R}(A,B)$ and $X \subseteq A \times B$, by R_X we denote the restriction of R to X. If the sets A and B are represented as $A = D \sqcup E$ and $B = F \sqcup G$, then the expression

$$R = \begin{bmatrix} R_{D\times F} & R_{D\times G} \\ R_{E\times F} & R_{E\times G} \end{bmatrix} \tag{1.93}$$

is called the *block representation* of R with blocks $R_{D\times F}$, $R_{D\times G}$, $R_{E\times F}$ and $R_{E\times G}$. If in addition, $C = I \sqcup J$, and relation $S \in \mathscr{R}(B,C)$, then we have that

$$\begin{aligned} R \circ S &= \begin{bmatrix} R_{D\times F} & R_{D\times G} \\ R_{E\times F} & R_{E\times G} \end{bmatrix} \circ \begin{bmatrix} S_{F\times I} & S_{F\times J} \\ S_{G\times I} & S_{G\times J} \end{bmatrix} \\ &= \begin{bmatrix} R_{D\times F}\circ S_{F\times I} \vee R_{D\times G}\circ S_{G\times I} & R_{D\times F}\circ S_{F\times J} \vee R_{D\times G}\circ S_{G\times J} \\ R_{E\times F}\circ S_{F\times I} \vee R_{E\times G}\circ S_{G\times I} & R_{E\times F}\circ S_{F\times J} \vee R_{E\times G}\circ S_{G\times J} \end{bmatrix} \end{aligned} \tag{1.94}$$

Next, if $f \in \mathscr{F}(A)$, $R \in \mathscr{R}(A,B)$ and $g \in \mathscr{F}(B)$, the composition $f \circ R$ and $R \circ g$ are fuzzy subsets of B and A, respectively, which are defined by

$$f \circ R = \begin{bmatrix} f_D \\ f_E \end{bmatrix} \circ \begin{bmatrix} R_{D\times F} & R_{D\times G} \\ R_{E\times F} & R_{E\times G} \end{bmatrix} = \begin{bmatrix} f_D \circ R_{D\times F} \vee f_D \circ R_{D\times G} \\ f_E \circ R_{E\times F} \vee f_E \circ R_{E\times G} \end{bmatrix} \tag{1.95}$$

$$R \circ g = \begin{bmatrix} R_{D\times F} & R_{D\times G} \\ R_{E\times F} & R_{E\times G} \end{bmatrix} \circ \begin{bmatrix} g_F \\ g_G \end{bmatrix} = \begin{bmatrix} R_{D\times F} \circ g_F \vee R_{D\times G} \circ g_G \\ R_{E\times F} \circ g_F \vee R_{E\times G} \circ g_G \end{bmatrix}. \tag{1.96}$$

Let's note that from (1.93) we have

$$R^{-1} = \begin{bmatrix} R^{-1}_{D\times F} & R^{-1}_{E\times F} \\ R^{-1}_{D\times G} & R^{-1}_{E\times G} \end{bmatrix}. \tag{1.97}$$

A fuzzy relation $\mathbf{0} \in \mathscr{R}(A, B)$ defined by $\mathbf{0}_{A\times B}(a, b) = 0$ for each $(a, b) \in A \times B$, is called the *empty relation* between A and B. For $\mathbf{0}_{A\times A}$ we say that it is the empty relation on A.

1.8 Uniform fuzzy relations

Let A and B be non-empty sets and let E and F be fuzzy equivalences on A and B, respectively. If a fuzzy relation $\varphi \in \mathscr{R}(A, B)$ satisfies

(*EX1*) $\qquad \varphi(a_1, b) \wedge E(a_1, a_2) \leqslant \varphi(a_2, b)$, for all $a_1, a_2 \in A$ and $b \in B$,

then it is called *extensional with respect to* E, and if it satisfies

(*EX2*) $\qquad \varphi(a, b_1) \wedge F(b_1, b_2) \leqslant \varphi(a, b_2)$, for all $a \in A$ and $b_1, b_2 \in B$,

then it is called *extensional with respect to* F. If φ is extensional with respect to E and F, and it satisfies

(*PFF*) $\qquad \varphi(a, b_1) \wedge \varphi(a, b_2) \leqslant F(b_1, b_2)$, for all $a \in A$ and $b_1, b_2 \in B$,

then it is called a *partial fuzzy function* with respect to E and F.

By the adjoint property and the symmetry the conditions (*EX1*) and (*EX2*) are equivalent to:

(*EX1'*) $\qquad E(a_1, a_2) \leqslant \varphi(a_1, b) \leftrightarrow \varphi(a_2, b)$, for all $a_1, a_2 \in A$ and $b \in B$,

(*EX2'*) $\qquad F(b_1, b_2) \leqslant \varphi(a, b_1) \leftrightarrow \varphi(a, b_2)$, for all $a, a \in A$ and $b_1, b_2 \in B$.

For any fuzzy relation $\varphi \in \mathscr{R}(A, B)$ we can define a fuzzy equivalence E^φ_A on A by

$$E^\varphi_A(a_1, a_2) = \bigwedge_{b\in B} \varphi(a_1, b) \leftrightarrow \varphi(a_2, b), \tag{1.98}$$

for all $a_1, a_2 \in A$, and a fuzzy equivalence E^φ_B on B by

$$E^\varphi_B(b_1, b_2) = \bigwedge_{a\in A} \varphi(a, b_1) \leftrightarrow \varphi(a, b_2), \tag{1.99}$$

for all $b_1, b_2 \in B$. They will be called *fuzzy equivalences* on A and B *induced by* φ, and in particular, E^φ_A will be called the *kernel* of φ, and E^φ_B the *cokernel* of φ. According to (*EX1'*) and (*EX2'*), the relations E^φ_A and E^φ_B are the greatest fuzzy equivalences on A and B, respectively, such that φ is extensional with respect to them.

A fuzzy relation $\varphi \in \mathscr{R}(A, B)$ is called just a *partial fuzzy function* if it is a partial fuzzy function with respect to E^φ_A and E^φ_B. Partial fuzzy functions were characterized in [24] as follows:

Theorem 1.14. *Let A and B be non-empty sets and let $\varphi \in \mathscr{R}(A,B)$ be a fuzzy relation. Then the following conditions are equivalent:*

(i) *φ is a partial fuzzy function;*

(ii) *φ^{-1} is a partial fuzzy function;*

(iii) *$\varphi^{-1} \circ \varphi \leqslant E_B^{\varphi}$;*

(iv) *$\varphi \circ \varphi^{-1} \leqslant E_A^{\varphi}$;*

(v) *$\varphi \circ \varphi^{-1} \circ \varphi \leqslant \varphi$.*

A fuzzy relation $\varphi \in \mathscr{R}(A,B)$ is called an *$\mathscr{L}$-function* if for each $a \in A$ there exists $b \in B$ such that $\varphi(a,b) = 1$ [34, 35], and it is called *surjective* if for each $b \in B$ there exists $a \in A$ such that $\varphi(a,b) = 1$, i.e., if φ^{-1} is an $\mathscr{L}$-function, and it is surjective, i.e., if both φ and φ^{-1} are $\mathscr{L}$-functions, then φ is called a *surjective $\mathscr{L}$-function*.

Let us note that a fuzzy relation $\varphi \in \mathscr{R}(A,B)$ is an $\mathscr{L}$-function if and only if there exists a function $\psi : A \to B$ such that $\varphi(a,\psi(a)) = 1$, for all $a \in A$. A function ψ with this property we will call a *crisp description* of φ, and we will denote by $CR(\varphi)$ the set of all such functions.

An $\mathscr{L}$-function which is a partial fuzzy function with respect to E and F is called a *perfect fuzzy function* with respect to E and F. Perfect fuzzy functions were introduced and studied by Demirci [33, 34]. A fuzzy relation $\varphi \in \mathscr{R}(A,B)$ which is a perfect fuzzy function with respect to E_A^{φ} and E_B^{φ} will be called just a *perfect fuzzy function*.

Let A and B be non-empty sets and let $\varphi \in \mathscr{R}(A,B)$ be a partial fuzzy function. If, in addition, φ is a surjective $\mathscr{L}$-function, then it will be called a *uniform fuzzy relation*. In other words, a uniform fuzzy relation is a perfect fuzzy function having the additional property that it is surjective.

Next, we recall the characterizations of uniform fuzzy relation from [24] which will be useful in our further work.

Theorem 1.15. *Let A and B be non-empty sets and let $\varphi \in \mathscr{R}(A,B)$ be a fuzzy relation. Then, the following conditions are equivalent:*

(1) *φ is a uniform fuzzy relation;*

(2) *φ^{-1} is a uniform fuzzy relation;*

(3) *φ is a surjective $\mathscr{L}$-function and $\varphi \circ \varphi^{-1} \circ \varphi = \varphi$;*

(4) *φ is a surjective $\mathscr{L}$-function and $E_A^{\varphi} = \varphi \circ \varphi^{-1}$;*

(5) *φ is a surjective $\mathscr{L}$-function and $E_B^{\varphi} = \varphi^{-1} \circ \varphi$;*

(6) *φ is an $\mathscr{L}$-function, and for all $\psi \in CR(\varphi)$, $a \in A$ and $b \in B$ we have that:*

$$\varphi \text{ is } E_B^{\varphi}\text{-surjective and } \varphi(a,b) = E_B^{\varphi}(\psi(a), b);$$

(7) *φ is an $\mathscr{L}$-function, and for all $\psi \in CR(\varphi)$, $a_1, a_2 \in A$ we have that:*

$$\psi \text{ is } E_B^{\varphi}\text{-surjective and } \varphi(a_1, \psi(a_2)) = E_A^{\varphi}(a_1, a_2).$$

Corollary 1.1. *Let A and B be non-empty sets and let $\varphi \in \mathscr{R}(A,B)$ be a uniform fuzzy relation. Then for all $\psi \in CR(\varphi)$ and $a_1, a_2 \in A$ we have that*

$$E_A^{\varphi}(a_1,a_2) = E_B^{\varphi}(\psi(a_1), \psi(a_2)). \tag{1.100}$$

Let A and B be non-empty sets. According to Theorem 1.15, a fuzzy relation $\varphi \in \mathscr{R}(A,B)$ is a uniform fuzzy relation. Further, from conditions (4) and (5) of the same theorem, we have that the kernel of φ^{-1} is the cokernel of φ and conversely, the cokernel of φ^{-1} is the kernel of φ, that is

$$E_B^{\varphi^{-1}} = E_B^{\varphi} \text{ and } E_A^{\varphi^{-1}} = E_A^{\varphi}.$$

The following theorems will be very useful in our further work.

Theorem 1.16. *Let A and B be non-empty sets, and let $\varphi \in \mathscr{R}(A,B)$ be a uniform fuzzy relation, let $E = E_A^{\varphi}$ and $F = E_B^{\varphi}$, and let $\widetilde{\varphi} : A/E \to B/F$ be the function given by*

$$\widetilde{\varphi}(E_a) = F_{\psi(a)}, \text{ for any } a \in A \text{ and } \psi \in CR(\varphi). \tag{1.101}$$

Then $\widetilde{\varphi}$ is a well-defined function (it does not depend on the choice of $\psi \in CR(\varphi)$ and $a \in A$), it is a bijective function of A/E onto B/F and $(\widetilde{\varphi})^{-1} = \widetilde{\varphi^{-1}}$.

Theorem 1.17. *Let A and B be non-empty sets, and let $\varphi_1, \varphi_2 \in \mathscr{R}(A,B)$ be uniform fuzzy relations. Then the following conditions are equivalent:*

(1) $\varphi_1 \leqslant \varphi_2$;
(2) $\varphi_1^{-1} \leqslant \varphi_2^{-1}$;
(3) $CR(\varphi_1) \subseteq CR(\varphi_2)$ *and* $E_A^{\varphi_1} \leqslant E_A^{\varphi_2}$;
(4) $CR(\varphi_1) \subseteq CR(\varphi_2)$ *and* $E_B^{\varphi_1} \leqslant E_B^{\varphi_2}$.

As a direct consequence of the previous theorem we obtain the following corollary which shows that a uniform fuzzy relation is uniquely determined by its crisp representation and kernel, as well as by its crisp representation and cokernel.

Lemma 1.7. *Let A and B be non-empty sets, and let $\varphi_1, \varphi_2 \in \mathscr{R}(A,B)$ be uniform fuzzy relations. Then the following conditions are equivalent:*

(1) $\varphi_1 = \varphi_2$;
(2) $\varphi_1^{-1} = \varphi_2^{-1}$;
(3) $CR(\varphi_1) = CR(\varphi_2)$ *and* $E_A^{\varphi_1} = E_A^{\varphi_2}$;
(4) $CR(\varphi_1) = CR(\varphi_2)$ *and* $E_B^{\varphi_1} = E_B^{\varphi_2}$.

The composition of two uniform fuzzy relations need not be a uniform fuzzy relation. However, if the cokernel of the first fact of the composition is contained in the kernel of the second factor, then the composition is uniform, as the following theorem shows.

Theorem 1.18. *Let A, B and C be non-empty sets, and let $\varphi_1 \in \mathscr{R}(A,B)$ and $\varphi_2 \in \mathscr{R}(B,C)$.*

(1) *If φ_1 and φ_2 are surjective $\mathscr{L}$-functions, then $\varphi_1 \circ \varphi_2$ is also a surjective $\mathscr{L}$-function.*
(2) *If φ_1 and φ_2 are uniform fuzzy relations such that $E_B^{\varphi_1} \leqslant E_B^{\varphi_2}$, then $\varphi_1 \circ \varphi_2$ is also a uniform fuzzy relation.*

Chapter 2

Fuzzy Multimodal Logics

> "Any necessary truth, whether a priori or a posteriori, could not have turned out otherwise."
>
> *Saul Kripke*

This chapter deals with the fuzzy Kripke models for fuzzy multimodal logics. Kripke models for classical modal logic based on the crisp structure $\{0, 1\}$ can be naturally generalized if we define them over the fuzzy structures. However, such a generalization has been made in several ways, which caused plenty of different fuzzy modal logics that differ in syntax and semantics. So we will now say something more about possible ways to generalize Kripke's models. Also, in fuzzy modal logic, some interesting phenomena can appear that are not common in classical modal logic and should be considered.

First, the fuzzy modal logics can be distinguished by the truth structures on which they are defined (see Section 1.4 and Figure 1.3). For example, the most commonly used truth structures are residuated lattices (cf. [16]), MTL-algebra (cf. [118]), Product algebra (cf. [142, 143]), Gödel algebra (cf. [18, 19, 47]), Łukasiewicz algebra (cf. [62]), Heyting algebra (cf. [42, 50, 51]), etc. Therefore, the properties of underlying algebra are reflected in the properties of defined logics and create interesting differences.

Second, Kripke models' generalization can differ in the values that relational structures can take. The most general approach allows both propositions at possible worlds and accessibility relations can be many-valued (cf. [18, 19, 47, 36]). The second approach allows propositions at the possible worlds can be many-valued while keeping crisp accessibility relations (cf. [142, 143]).

There are various studies of modal expansions of many-valued logics. Fuzzy modal operators are generalizations of operators well-known in modal logics, and they have substantial differences compared to classical modal operators. For a given fuzzy relation R between X and Y, then for every fuzzy subset B of Y and every $x \in X$, we define

$$\Box B = \bigwedge_{y \in Y} (R(x, y) \to B(y)), \tag{2.1}$$

$$\Diamond B = \bigvee_{y \in Y} (R(x, y) \otimes B(y)). \tag{2.2}$$

The above operators are called *fuzzy necessity* and *fuzzy possibility*. In contrast to Propositional Modal Logic (PML), such defined operators are not generally interdefinable in the general cases. This feature allows us to define fuzzy modal logic with only one operator. For example, in [16], Kripke models are defined over bounded commutative residuated lattices with only one modal operator $\Box$. Also, non-interdefinability can cause that $\Box$- and $\Diamond$-fragment to have different characteristics. For example, in [18], it has been shown that in standard Gödel algebra $[0, 1]$, $\Diamond$-fragment has finite model property, while $\Box$-fragment does not.

In addition to fuzzy necessity and fuzzy possibility, Radzikowska in [116] defines *fuzzy sufficiency* and *fuzzy dual sufficiency*. These operators have a natural interpretation in data analysis.

Regardless of modal operators, many authors consider projection operator Δ in their extension of fuzzy multimodal logics. The operator is usually called *Baaz Delta*, or *Monteiro-Baaz Δ operator*, named after its author Baaz (see [2]). The operator Δ is defined on $[0, 1]$ as $\Delta x = 1$ if $x = 1$ and $\Delta x = 0$, otherwise.

Another important thing in the generalization of classical logic is treating degrees of truth. We emphasize the significant work of Pavelka (cf. [109, 110, 111]), who built a propositional many-valued logical system (PL) by introducing truth-constants in the language. He added new constant symbols $\bar{c}$ for appropriate values $c \in [0, 1]$ and determined $V(\bar{c}) = c$ for all truth-evaluations. It turned out that PL is equivalent to Łukasiewicz's logic with a truth-constant $\bar{c}$ where c is a real number from $[0, 1]$ with some additional axioms. This all led to the kind of completeness known as *Pavelka-style completeness*, which differs from strong standard completeness.

Similar rational expansions for a wide class of other t-norm based fuzzy logics have been defined. However, Pavelka-style completeness for these logics could not be obtained except for Łukasiewicz's logic due to continuous truth-functions. For details, see [46].

Further, based on traditional algebraic semantics, expansions with truth-constants, mostly for the Gödel, Product and BL logics have been considered. For example, in [130] the expansion of Product logic with rational constants was studied. Also, for more information, we refer to [44, 46, 141], etc. Furthermore, numerous papers have shown many benefits for expanding t-norm based logics with rational truth-constants and their rational completeness properties.

This chapter defines fuzzy multimodal logics over a complete Heyting algebra where propositions at possible worlds and accessibility relations can be many-valued. Furthermore, we consider multimodal logic with four families of modal operators (fuzzy necessity and fuzzy possibility with their inverse operators) to have the most general syntax due to easier connection with structures (for example, fuzzy automata). Also, we expand logic with *canonical constants*, i.e., constant for each element of the universe. Hence, both the accessibility relations and propositional variables can take the values from Heyting algebras endowed with canonical constants. The logic defined in this way was used in the papers [135, 136].

The chapter consists of five sections. First Section 2.1 contains basic definitions about Kripke semantics in crisp case, i.e., for Propositional Modal Logic. The second Section 2.2 defines Kripke semantics for fuzzy multimodal logics over a complete Heyting algebra. Section 2.3 gives the basic properties of the fuzzy formulae that we will need in the following chapters. Next, Section 2.4 consists of examples of Kripke models and where some interesting properties of the fuzzy modal logics are highlighted. In the last Section 2.5, we deal with afterset Kripke models, which will be especially important in the following chapters.

2.1 Kripke semantics

We assume the reader is familiar with the basic concepts of classical modal logic, Kripke semantics, etc. (cf. [11, 23]). In this section, we deal with Propositional Modal Logic (PML) and expand basic modal language from [11] with inverse modal operators.

The alphabet of PML comprises a set of propositional symbols PV, the logical constant $\overline{0}$, the Boolean connectives $\wedge$ (*conjunction*) and $\rightarrow$ (*implication*) and modal operators $\Diamond$ (*possibility operator*) and $\Diamond^{-}$ (*inverse possibility operator*). Now, we can define the set of well-formed formulae (WFF).

Definition 2.1. Let $\mathscr{B} = (B, \wedge, \vee, \rightarrow, 0, 1)$ be a two-element Boolean algebra and write $\overline{B} = \{\overline{t} \mid t \in B\}$ for the elements of $\mathscr{B}$ viewed as constants. Define the language $\Phi_{\mathscr{B}}$ via the grammar

$$A ::= \overline{0} \mid p \mid A \wedge A \mid A \rightarrow A \mid \Diamond A \mid \Diamond^{-} A \tag{2.3}$$

where p ranges over some set PV of propositional letters.

In addition, we use left and right parentheses, "(" and ")", as auxiliary symbols to avoid ambiguity in WFFs. Let us note that $\overline{1} = \overline{0} \rightarrow \overline{0}$. We also use standard abbreviations:

$\neg A \equiv A \rightarrow \overline{0}$ (*negation*),
$A \leftrightarrow B \equiv (A \rightarrow B) \wedge (B \rightarrow A)$ (*equivalence*),
$A \vee B \equiv \neg(\neg A \wedge \neg B)$ (*disjunction*),
$\Box A \equiv \neg \Diamond \neg A$ (*necessity operator*),
$\Box^{-} A \equiv \neg \Diamond^{-} \neg A$ (*inverse necessity operator*).

A well-formed formula will be simply called *formula*. Let PML^{+} be the set of all formulae with modality $\Diamond$ and its dual operator $\Box$, PML^{-} be the set of all formulae for propositional modal logics with converse modality $\Diamond^{-}$ and its dual operator $\Box^{-}$. Finally, let PML denotes the set of all formulae for propositional modal logic with modalities $\Diamond$ and $\Diamond^{-}$ and with their dual operators $\Box$ and $\Box^{-}$, respectively.

Definition 2.2. A *Kripke frame* is a structure $\mathfrak{F} = (W, R)$ where W is a non-empty set of possible *worlds* (or *states* or *points*) and R is called the *accessibility relation* of the frame.

Definition 2.3. A *Kripke model for PML* is a structure $\mathfrak{M} = (W, R, V)$ such that (W, R) is a Kripke frame and $V : W \times (PV \cup \overline{B}) \rightarrow \{0, 1\}$ is a truth assignment function, called the *evaluation of the model*, which assign B-truth value to propositional variables (and truth constants) in each world, such that $V(w, \overline{t}) = t$, for every $w \in W$ and $t \in B$.

Now, we define the *satisfaction relation*, i.e., the notion when a formula A is *satisfied* or *true* in a world w of model $\mathfrak{M}$. We write $\mathfrak{M}, w \not\models A$ to mean "not $\mathfrak{M}, w \models A$".

Definition 2.4. The *satisfaction of a formula* A in a world w of the model $\mathfrak{M}$, denoted by $\mathfrak{M}, w \models A$, is inductively defined as follows:

(1) $\mathfrak{M}, w \models p$ iff $V(w, p) = 1$ for each $p \in PV$;
(2) $\mathfrak{M}, w \not\models \bot$;
(3) $\mathfrak{M}, w \models A \wedge B$ iff $\mathfrak{M}, w \models A$ and $\mathfrak{M}, w \models B$;
(4) $\mathfrak{M}, w \models A \to B$ iff $\mathfrak{M}, w \not\models A$ or $\mathfrak{M}, w \models B$;
(5) $\mathfrak{M}, w \models \Diamond A$ iff there exists $(w, u) \in R$ such that $\mathfrak{M}, w \models A$;
(6) $\mathfrak{M}, w \models \Diamond^- A$ iff there exists $(u, w) \in R$ such that $\mathfrak{M}, u \models A$.

The class of all Kripke models will be denoted by K.

A formula A is *satisfiable in* K iff there exists a word w of the model $\mathfrak{M} \in \mathsf{K}$ that satisfies A. Further, a set of formulae Ψ is satisfiable in a word w of the model $\mathfrak{M}$, written $\mathfrak{M}, w \models \Psi$, iff $\mathfrak{M}, w \models A$ for every $A \in \Psi$.

A formula A is *valid in the model* $\mathfrak{M}$ if it is satisfied in every word w of the model $\mathfrak{M}$. Further, a formula A is valid in the class K if it is valid in every model $\mathfrak{M}$ of the class K.

2.2 Fuzzy Kripke semantics

Now, we will generalize Kripke semantics from the previous section. Hence, a fuzzy multimodal logic over a Heyting algebra will be defined.

In the sequel, unless otherwise stated, $\mathscr{H} = (H, \wedge, \vee, \to, 0, 1)$ will be a complete Heyting algebra and I will be a non-empty set of indices. An alphabet of a many-valued multimodal logic $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i \in I})$ consists of an enumerable set of *propositional symbols* PV, a set of *truth constants* $\overline{H} = \{\overline{t} \mid t \in H\}$, *logical connectives* $\wedge$ (*conjunction*) and $\to$ (*implication*), and four families of *modal operators*: $\{\Box_i\}_{i \in I}$ and $\{\Box_i^-\}_{i \in I}$ (*necessity operators*) and $\{\Diamond_i\}_{i \in I}$ and $\{\Diamond_i^-\}_{i \in I}$ (*possibility operators*). More formally, we have the following definition.

Definition 2.5. Let $\mathscr{H} = (H, \wedge, \vee, \to, 0, 1)$ be a complete Heyting algebra and write $\overline{H} = \{\overline{t} \mid t \in H\}$ for elements of $\mathscr{H}$ viewed as constants. Let I be some index set. Define the language $\Phi_{I,\mathscr{H}}$ via the grammar

$$A ::= \overline{t} \mid p \mid A \wedge A \mid A \to A \mid \Box_i A \mid \Diamond_i A \mid \Box_i^- A \mid \Diamond_i^- A \tag{2.4}$$

where $\overline{t} \in \overline{H}$, $i \in I$ and p ranges over some set PV of proposition letters.

Hence, the set of formulae $\Phi_{I,\mathscr{H}}$ of a many-valued modal logic is the smallest set containing propositional symbols and truth constants, and is closed under logical connectives and modal operators. The following well-known abbreviations will be used:

$\neg A \equiv A \to \overline{0}$ (*negation*),
$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$ (*equivalence*),
$A \vee B \equiv ((A \to B) \to B) \wedge ((B \to A) \to A)$ (*disjunction*).

Recall that 0 is the least element in $\mathscr{H}$ and $\overline{0}$ is the corresponding truth constant. Also, $\overline{0} \to \overline{0}$ gives $\overline{1}$.

Definition 2.6. A *fuzzy Kripke frame* is a structure $\mathfrak{F} = (W, \{R_i\}_{i\in I})$ where W is a non-empty set of possible *worlds* (or *states* or *points*) and $R_i \in \mathscr{F}(W \times W)$ is a binary fuzzy relation on W, for every i from a finite index set I, called the *accessibility fuzzy relation* of the frame.

Definition 2.7. A *fuzzy Kripke model for* $\Phi_{I,\mathscr{H}}$ is a structure $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ such that $(W, \{R_i\}_{i\in I})$ is a fuzzy Kripke frame and $V : W \times (PV \cup \overline{H}) \to H$ is a truth assignment function, called the *evaluation of the model*, which assigns an H-truth value to propositional variables (and truth constants) in each world, such that $V(w, \overline{t}) = t$, for every $w \in W$ and $t \in H$.

In the case when the finite set I has n elements, then $\mathfrak{F}$ is called a *fuzzy Kripke n-frame* and $\mathfrak{M}$ is called a *fuzzy Kripke n-model.*

Note that the defined notion of a Kripke n-model for $\mathscr{H}$ should not be identified with the notion of an n-model defined in [74], i.e., models with the assignment function V restricted to the propositional variables $p_1, \dots, p_n$ and thereby to n-formulae, formulae formed from $p_1, \dots, p_n$.

The truth assignment function V can be inductively extended to a function $V : W \times \Phi_{I,\mathscr{H}} \to H$ by:

(V1) $V(w, A \wedge B) = V(w, A) \wedge V(w, B)$;

(V2) $V(w, A \to B) = V(w, A) \to V(w, B)$;

(V3) $V(w, \Box_i A) = \bigwedge_{u\in W} R_i(w, u) \to V(u, A)$, for every $i \in I$;

(V4) $V(w, \Diamond_i A) = \bigvee_{u\in W} R_i(w, u) \wedge V(u, A)$, for every $i \in I$;

(V5) $V(w, \Box_i^- A) = \bigwedge_{u\in W} R_i(u, w) \to V(u, A)$, for every $i \in I$;

(V6) $V(w, \Diamond_i^- A) = \bigvee_{u\in W} R_i(u, w) \wedge V(u, A)$, for every $i \in I$.

Note that the same symbols are used for $\wedge$ and $\to$ in both sides of formulae (V1)–(V6). The meaning is clear from the context, so we keep the notation simple. For each world $w \in W$ the truth assignment V determines a function $V_w : \Phi_{I,\mathscr{H}} \to H$ given by $V_w(A) = V(w, A)$, for every $A \in \Phi_{I,\mathscr{H}}$, and vice versa, for each $A \in \Phi_{I,\mathscr{H}}$ the truth assignment V determines a function $V_A : W \to H$ given by $V_A(w) = V(w, A)$, for every $w \in W$.

Usually, we will denote the models with $\mathfrak{M}$, $\mathfrak{M}'$, $\mathfrak{N}$, $\mathfrak{N}'$ etc., not emphasizing specifically the alphabet $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i\in I})$, except when necessary.

For a fuzzy Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$, its *reverse fuzzy Kripke model* is the fuzzy Kripke model $\mathfrak{M}^{-1} = (W, \{R_i^{-1}\}_{i\in I}, V)$.

The following Definition is based on Definition 1.39, where image-finite, domain-finite and degree-finite relation is defined.

Definition 2.8. A fuzzy Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ is called *image-finite* if the relation R_i is image-finite, for every $i \in I$, it is called *domain-finite* if the relation R_i is domain-finite, for every $i \in I$, and it is called *degree-finite* if the relation R_i is degree-finite, for every $i \in I$.

Definition 2.9. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, and let $\Phi \subseteq \Phi_{I,\mathscr{H}}$ be some set of formulae. Worlds $w \in W$ and $w' \in W'$ are said to be Φ-*equivalent* if $V(w, A) = V'(w', A)$, for all $A \in \Phi$. Moreover, $\mathfrak{M}$ and $\mathfrak{M}'$ are said to be Φ-*equivalent fuzzy Kripke models* if each $w \in W$ is Φ-equivalent to some $w' \in W'$, and vice versa, if each $w' \in W'$ is Φ-equivalent to some $w \in W$.

Many authors use the term *modal equivalence* for the relation between two worlds defined as follows: two worlds $w \in W$ and $w' \in W$ are *modally equivalent* if $V(w, A) = V'(w', A)$, where A is from the set of all formulae (cf. [11, 36]). Therefore, Definition 2.9 is more general since the notion of formulae equivalence can be defined for some set of formulae. We also defined formulae equivalence between two Kripke models.

Definition 2.10. Two fuzzy Kripke models $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ are said to be *isomorphic* if there exists a bijective function $\phi : W \to W'$ such that $R_i(u, v) = R'_i(\phi(u), \phi(v))$ and $V(w, p) = V'(\phi(w), p)$, for all $i \in I$, $p \in PV$ and $u, v, w \in W$.

In general, the size of a Kripke model $|\mathfrak{M}|$ is defined to be the number of worlds plus the size of all accessibility relations plus the total number of propositional variables from its worlds. Formally, we have the following definition:

Definition 2.11. The *size of a Kripke model* $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$, denoted by $|\mathfrak{M}|$ is defined as follows:

$$|\mathfrak{M}| = |W| + \sum_{i\in I} |\{R_i(u, v) > 0 \mid u, v \in W\}| + \sum_{w\in W} |\{V(w, p) \mid p \in PV\}|. \qquad (2.5)$$

2.3 Properties of fuzzy formulae

In this section, we will deal with the properties of the fuzzy formula that we will need in the following chapters. First, we will define some subsets of the set of all formulae $\Phi_{I,\mathscr{H}}$.

The set of all formulae over the alphabet $\mathscr{H}(\{\Box_i, \Diamond_i\}_{i\in I})$, i.e., the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i^-$ and $\Diamond_i^-$, $i \in I$, will be denoted by $\Phi^+_{I,\mathscr{H}}$. Similarly, the set of all formulae over the alphabet $\mathscr{H}(\{\Box_i^-, \Diamond_i^-\}_{i\in I})$, i.e., the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i$ and $\Diamond_i$, $i \in I$, will be denoted by $\Phi^-_{I,\mathscr{H}}$. For the sake of simplicity, formulae from $\Phi^+_{I,\mathscr{H}}$ will be called *plus-formulae*, and formulae from $\Phi^-_{I,\mathscr{H}}$ will be called *minus-formulae*.

In the same manner, the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i$, $\Box_i^-$ and $\Diamond_i^-$, $i \in I$, will be denoted by $\Phi^{\Diamond+}_{I,\mathscr{H}}$. Also, the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i$, $\Box_i^-$ and $\Diamond_i$, $i \in I$, will be denoted by $\Phi^{\Diamond-}_{I,\mathscr{H}}$. Finally, with $\Phi^{\Diamond}_{I,\mathscr{H}}$ will be denoted the set of all formulae over the alphabet $\mathscr{H}(\{\Diamond_i, \Diamond_i^-\}_{i\in I})$, i.e., the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i$ and $\Box_i^-$, $i \in I$.

Analogously, the sets of formula $\Phi^{\Box-}_{I,\mathscr{H}}$, $\Phi^{\Box+}_{I,\mathscr{H}}$, $\Phi^{\Box}_{I,\mathscr{H}}$ can be defined with self-explanatory notations. For the sake of simplicity, formulae from $\Phi^{\Diamond}_{I,\mathscr{H}}$ will be called *possibility-fragment*, and formulae from $\Phi^{\Box}_{I,\mathscr{H}}$ will be called *necessity-fragment*. Finally, the set of those formulae from $\Phi_{I,\mathscr{H}}$ that do not contain any of the modal operators $\Box_i$, $\Box_i^-$, $\Diamond_i$ and $\Diamond_i^-$, $i \in I$, will be denoted with $\Phi^{PF}_{I,\mathscr{H}}$, where PF denotes *propositional formulae*.

Therefore, we will consider the following set which consists of previously defined sets of formulae:

$$\{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\} \tag{2.6}$$

It is interesting that this set is a complete lattice, where the order is defined in the same way as for sets (see Figure 2.1).

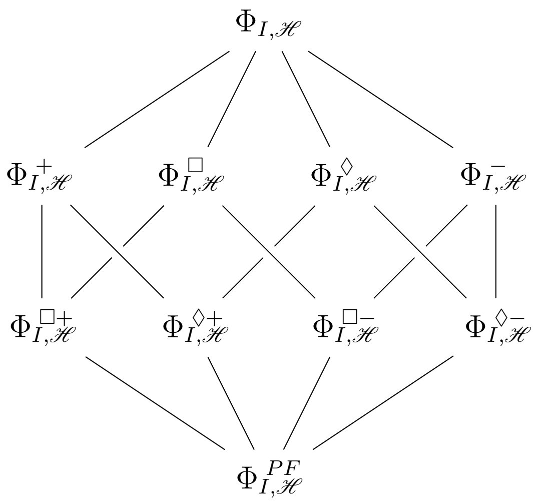

Figure 2.1: Lattice of sets of formulae.

As already mentioned, fuzzy settings have a profound effect on the behaviour of modal formulae. We said that fuzzy modal operators are not interdefinable which is confirmed by the following example.

Example 2.1. Let $\mathfrak{M} = (W, R, V)$ be a fuzzy Kripke model over the Gödel structure from Example 1.8, where $W = \{v, w\}$, and fuzzy relation R and fuzzy set V_p are represented by the following fuzzy matrix and vector:

$$R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}.$$

Now, we have:

$$V_{\Diamond p} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \qquad V_{\neg\Box\neg p} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

However, in [51] was proved that modal operators are not each other's dual unless underlying algebra is a Boolean algebra. Further, modal operators are interdefinable in logics with involutive negation (for example, Łukasiewicz logic). In addition to the impaired interdefinability of modal operators, many other modal axioms are no longer valid in the general case for fuzzy modal logics. Under what conditions some modal axioms remain valid can be seen in [16]. Also, for some modal schemas in MTL algebras we refer to [118].

Let us note that modal operators have a property we will call *reversing duality of modal operators*. Let a mapping $\alpha \mapsto \alpha^d$ from the set of logical operators:

$$\{\wedge, \rightarrow, \Box_i, \Box_i^-, \Diamond_i, \Diamond_i^-\}, \quad \text{for every } i \in I$$

into itself defined as follows:

$$\begin{pmatrix} \wedge & \rightarrow & \Box_i & \Box_i^- & \Diamond_i & \Diamond_i^- \\ \wedge & \rightarrow & \Box_i^- & \Box_i & \Diamond_i^- & \Diamond_i \end{pmatrix}.$$

Hence, non-modal operators are mapped to themselves while modal operators are reversed. For example, a set of formulae Ψ contain formulae $A \rightarrow B$, $\Box_i A$ and $\Diamond_j^- A \wedge \Box_k B$ if and only if Ψ^d contain formulae $A \rightarrow B$, $\Box_i^- A$ and $\Diamond_j A \wedge \Box_k^- B$, for some $i, j, k \in I$.

Therefore, reversing duality of formula can be defined for an arbitrary sets of formulae for Kripke models. We are especially interested in the set (2.6) and let a mapping $\Psi \mapsto \Psi^d$ from the set (2.6) into itself be defined as follows:

$$\begin{pmatrix} \Phi_{I,\mathscr{H}}^{PF} & \Phi_{I,\mathscr{H}}^{\Box+} & \Phi_{I,\mathscr{H}}^{\Diamond+} & \Phi_{I,\mathscr{H}}^{\Box-} & \Phi_{I,\mathscr{H}}^{\Diamond-} & \Phi_{I,\mathscr{H}}^{+} & \Phi_{I,\mathscr{H}}^{-} & \Phi_{I,\mathscr{H}}^{\Box} & \Phi_{I,\mathscr{H}}^{\Diamond} & \Phi_{I,\mathscr{H}} \\ \Phi_{I,\mathscr{H}}^{PF} & \Phi_{I,\mathscr{H}}^{\Box-} & \Phi_{I,\mathscr{H}}^{\Diamond-} & \Phi_{I,\mathscr{H}}^{\Box+} & \Phi_{I,\mathscr{H}}^{\Diamond+} & \Phi_{I,\mathscr{H}}^{-} & \Phi_{I,\mathscr{H}}^{+} & \Phi_{I,\mathscr{H}}^{\Box} & \Phi_{I,\mathscr{H}}^{\Diamond} & \Phi_{I,\mathscr{H}} \end{pmatrix}. \tag{2.7}$$

Now we can state the following corollary.

Proposition 2.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ be fuzzy Kripke model and $\mathfrak{M}^{-1} = (W, \{R_i^{-1}\}_{i \in I}, V)$ corresponding reverse model. An arbitrary set of formulae Ψ defined on $\mathfrak{M}$ is identical to set Ψ^d on $\mathfrak{M}^{-1}$.*

Proof. The proof is a direct consequence of definitions of the reverse Kripke model and reversing duality of formulae. $\square$

In Definition 2.5, we defined a set of all formulae $\Phi_{I,\mathscr{H}}$ of fuzzy multimodal logic in Backus-Naur form (BNF), but now more improvements and precisions will be introduced to pave the way for the following chapters.

Definition 2.12. An alphabet of fuzzy multimodal logic $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i \in I})$ consists of:

(1) enumerable set of propositional symbols PV,
(2) set of truth constants $\overline{H} = \{\overline{t} \mid t \in H\}$,
(3) set of unary logical connectives, $ULC = \{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}$, for every $i \in I$,
(4) set of binary logical connectives, $BLC = \{\wedge, \rightarrow\}$.

Derived operations $\leftrightarrow$ and $\vee$ may also be included in the set BLC. We also exclude operation $\neg$ from ULC, since $\neg A \equiv A \rightarrow 0$.

Definition 2.13. The set $\Phi_{I,\mathscr{H}}$ is the set of *well-formed formulae*, i.e., set formed such that

(1) propositional variables (PV) and set of truth constants $\overline{H}$ are in Ψ,
(2) if $A \in \Psi$, then so are $(*A)$, for every $* \in ULC$,
(3) if $A, B \in \Psi$, then so are $(A \star B)$, for every $\star \in BLC$.

Definition 2.14. The *complexity of a formula* A will be denoted with $c(A)$ and represent the number of occurrences of connectives in it. Then we have:

(1) $c(p) = 0$, for every $p \in PV$,
(2) $c(\bar{t}) = 0$, for every $\bar{t} \in \overline{H}$,
(3) $c(*A) = c(A) + 1$, for every $* \in ULC$,
(4) $c(A \star B) = c(A) + c(B) + 1$, for every $\star \in BLC$.

For a given Kripke model, we can now form sequence $\{F_n\}_{n\in\mathbb{N}_0}$ for every $\Psi \in \{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$ where $F_n = \{A \in \Psi \mid c(A) = n\}$. Actually, it follows:

$$\Psi = \bigcup_{n=0}^{+\infty} F_n. \tag{2.8}$$

The following example shows so-called *state explosion problem* which is one of the biggest obstacles for model checking. In our case, we will call it *formulae explosion problem.*

Example 2.2 (Formulae explosion)**.** Let $\mathfrak{M} = (W, R, V)$ be fuzzy Kripke model $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i\in I})$ model over two-valued Heyting algebra, with one relation R, and p be one propositional letter in the model. Then, for $\Psi = \Phi_{I,\mathscr{H}}$ we have:

$$\begin{aligned}
F_0 =& \{p, 0, 1\}\\
F_1 =& \{\Box p, \Box 0, \Box 1, \Diamond p, \Diamond 0, \Diamond 1, \Box^- p, \Box^- 0, \Box^- 1, \Diamond^- p, \Diamond^- 0, \Diamond^- 1,\\
& p \wedge p, p \wedge 0, p \wedge 1, 0 \wedge p, 0 \wedge 0, 0 \wedge 1, 1 \wedge p, 1 \wedge 0, 1 \wedge 1,\\
& p \to p, p \to 0, p \to 1, 0 \to p, 0 \to 0, 0 \to 1, 1 \to p, 1 \to 0, 1 \to 1\}\\
& \vdots
\end{aligned}$$

As we can see, the number of formulae is increasing very rapidly. In fact, the number of formulae in sets F_n can be calculated for every degree-finite model $\mathfrak{M}$ with finite sets $W, I, PV, \overline{H}$. Now, we have:

$$\begin{aligned}
|F_0| &= |PV| + |\overline{H}|\\
|F_1| &= |F_0| \cdot |ULC| + |F_0| \cdot |BLC| \cdot |F_0|\\
|F_2| &= |F_1| \cdot |ULC| + |F_0| \cdot |BLC| \cdot |F_1| + |F_1| \cdot |BLC| \cdot |F_0|\\
|F_3| &= |F_2| \cdot |ULC| + |F_0| \cdot |BLC| \cdot |F_2| + |F_1| \cdot |BLC| \cdot |F_1| + |F_2| \cdot |BLC| \cdot |F_0|\\
&\vdots\\
|F_n| &= |F_{n-1}| \cdot |ULC| + \sum_{i+j=n-1} |F_i| \cdot |BLC| \cdot |F_j|
\end{aligned}$$

Hence,

$$|F_n| = |F_{n-1}| \cdot |ULC| + |BLC| \sum_{i=0}^{n-1} |F_i| \cdot |F_{n-i-1}|.$$

Now, we can easily see that number of formulae in sets F_n grows exponentially because the growth rate of the sequence $|F_n|$ is bigger than Fibonacci numbers, i.e., $f(n) = f(n-1) + f(n-2)$.

In our example, $|PV| = 1$, $|\overline{H}| = 2$, $ULC = \{\Box, \Diamond, \Box^-, \Diamond^-\}$, so it $|ULC| = 4$, and $|BLC| = 2$. Now, we can compute number of formulae, and we have

$$|F_0| = 3$$

$$
\begin{aligned}
|F_1| &= 30\\
|F_2| &= 480\\
|F_3| &= 9480\\
|F_4| &= 209280\\
|F_5| &= 4946880\\
|F_6| &= 122465280\\
|F_7| &= 3134628480\\
|F_8| &= 82283796480\\
|F_9| &= 2203011425280\\
|F_{10}| &= 59925740666880\\
|F_{11}| &= 1651484601569280\\
|F_{12}| &= 46012170374676480\\
&\vdots
\end{aligned}
$$

Of course, these numbers are much higher when we increase the number of variables, constants, and unary operations.

2.4 Examples of fuzzy Kripke models

This section gives examples of some Kripke models to provide additional clarifications.

Example 2.3. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure, where $W = \{u, v, w\}$, $W' = \{u', v'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R'_1 and fuzzy sets V_p, V_q, V'_p and V'_q are represented by the following fuzzy matrices and column vectors:

$$
R_1 = \begin{bmatrix} 1 & 0 & 0.9 \\ 1 & 0.3 & 0.6 \\ 1 & 0 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}, \tag{2.9}
$$

$$
R'_1 = \begin{bmatrix} 1 & 0.4 \\ 1 & 0.4 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}. \tag{2.10}
$$

Models $\mathfrak{M}$ and $\mathfrak{M}'$ are graphically represented in Figure 2.2. Due to their convenient notation, Kripke models are often used in the representation of various structures. Hence, several interpretations can be made. Usually, an "actual" world is indicated by the double circle (in our case, it is the world w_1). However, the actual world will not be relevant to our consideration. When $R(u, v) = 0$, for some u, v, we do not draw an arrow between worlds u and v.

Example 2.4. Let $\mathfrak{M}^{-1} = (W, \{R_i^{-1}\}_{i\in I}, V)$ and $\mathfrak{M}'^{-1} = (W', \{R_i'^{-1}\}_{i\in I}, V')$ be the reverse Kripke models from Example 2.3. Then, these models are represented by the following fuzzy matrices and vectors:

$$
R_1^{-1} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0.3 & 0 \\ 0.9 & 0.6 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix},
$$

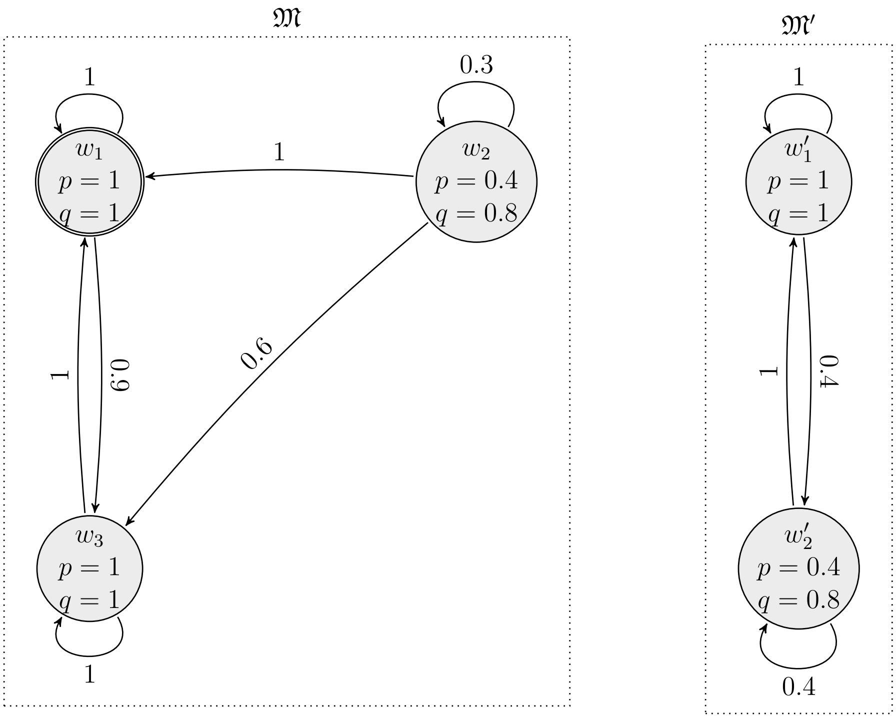

Figure 2.2: Models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 2.3.

$$R_1'^{-1} = \begin{bmatrix} 1 & 1 \\ 0.4 & 0.4 \end{bmatrix}, \qquad V_p' = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}, \qquad V_q' = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}.$$

Models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$ can be graphically presented in the same way as in Figure 2.2, except that the arrows are in the opposite direction.

Example 2.5. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be fuzzy Kripke model over the Gödel structure, where $W = \{w_1, w_2, w_3, w_4\}$, and set $I = \{1, 2\}$. Fuzzy relations R_1, R_2 and fuzzy sets V_p, V_q are represented by the following fuzzy matrices and vectors:

$$R_1 = \begin{bmatrix} 0.7 & 1 & 0.5 & 0.8 \\ 1 & 0.4 & 0.7 & 1 \\ 0.3 & 0.8 & 0.1 & 1 \\ 0.6 & 1 & 0.9 & 0.8 \end{bmatrix}, R_2 = \begin{bmatrix} 1 & 0.1 & 0.2 & 0.6 \\ 0.4 & 0.3 & 0.8 & 1 \\ 0.2 & 0.7 & 0.1 & 1 \\ 0.3 & 0.8 & 0.1 & 0.4 \end{bmatrix}, V_p = \begin{bmatrix} 0.7 \\ 0.8 \\ 1 \\ 1 \end{bmatrix}, V_q = \begin{bmatrix} 0.7 \\ 0.6 \\ 1 \\ 1 \end{bmatrix}.$$

Model $\mathfrak{M}$ is graphically represented in Figure 2.3. The relation R_1 is represented by solid lines, while dotted lines represent the relation R_2.

For the following definition we will use block representation for fuzzy sets (1.92) and fuzzy relations (1.93).

Definition 2.15. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models. The *disjoint union* of $\mathfrak{M}$ and $\mathfrak{M}'$, denoted by $\mathfrak{M} \sqcup \mathfrak{M}'$ is the fuzzy Kripke model $\mathfrak{M}'' = (W'', \{R_i''\}_{i\in I}, V'')$ such that $W'' = W \sqcup W'$,

$$R_i'' = R_i \sqcup R_i' = \begin{bmatrix} R_{i,W\times W} & \mathbf{0}_{W\times W'} \\ \mathbf{0}_{W'\times W} & R_{i,W'\times W'} \end{bmatrix}, \text{ for every } i \in I,$$

and

$$V_p'' = V_p \sqcup V_p' = \begin{bmatrix} V_{p,W} \\ V'_{p,W'} \end{bmatrix}.$$

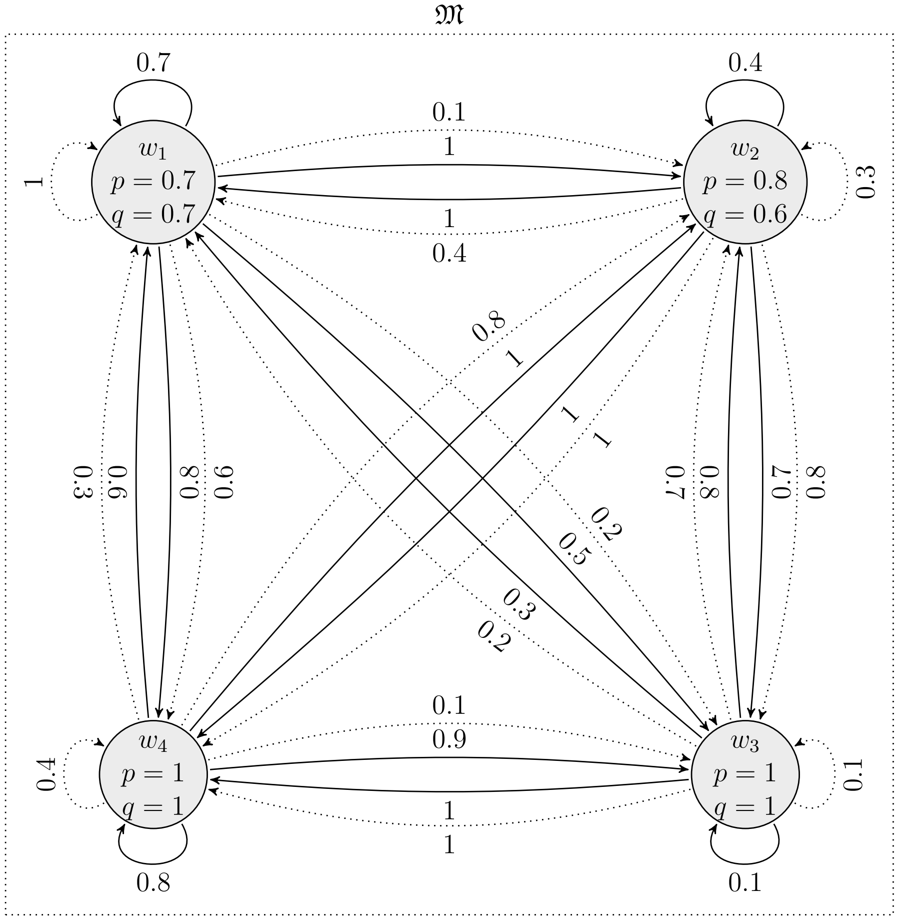

Figure 2.3: Model $\mathfrak{M}$ with two relations from Example 2.5.

Example 2.6. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure $[0,1]$, where $W = \{u, v, w\}$, $W' = \{u', v'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R_1' and fuzzy sets V_p, V_q, V_p' and V_q' are represented by the following fuzzy matrices and vectors:

$$R_1 = \begin{bmatrix} 0.8 & 0.1 & 0.9 \\ 0.2 & 0.8 & 1 \\ 0.6 & 0.7 & 0.9 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \end{bmatrix}, \tag{2.11}$$

$$R_1' = \begin{bmatrix} 0.8 & 0.7 \\ 0.6 & 0.8 \end{bmatrix}, \qquad V_p' = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}, \qquad V_q' = \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix}. \tag{2.12}$$

According to the Definition 2.15, we have $W'' = \{u, v, w, u', v'\}$ since $W = \{u, v, w\}$ and $W' = \{u', v'\}$. Also:

$$R_1'' = R_1 \sqcup R_1' = \begin{bmatrix} 0.8 & 0.1 & 0.9 & 0 & 0 \\ 0.2 & 0.8 & 1 & 0 & 0 \\ 0.6 & 0.7 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0.7 \\ 0 & 0 & 0 & 0.6 & 0.8 \end{bmatrix}, \tag{2.13}$$

$$V_p'' = V_p \sqcup V_p' = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad V_q'' = V_q \sqcup V_q' = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \\ 0.8 \\ 0.7 \end{bmatrix}. \tag{2.14}$$

2.5 Afterset Kripke models

In this section Theorem 1.13 will be applied to define the afterset Kripke model. Therefore, as a direct consequence of the Theorem, we have the following.

Theorem 2.1. *Let Q be a fuzzy quasi-order on a set W and E the natural fuzzy equivalence of Q. Then*

(a) *For arbitrary $w, u \in W$ the following conditions are equivalent:*
  (i) $E(w,u) = 1$;
  (ii) $E_w = E_u$;
  (iii) $Q_w = Q_u$;
  (iv) $Q^w = Q^u$.

(b) *Functions $Q_w \mapsto E_w$ of W/Q to W/E, and $Q_w \mapsto Q^w$ of W/Q to $W\backslash Q$ are bijective functions.*

Let $\mathfrak{F} = (W, \{R_i\}_{i\in I})$ be a fuzzy Kripke frame over $\mathscr{H}$ and let Q be a fuzzy quasi-order on W. For each $i \in I$ we can define a fuzzy relation $R_i^{W/Q} : W/Q \times W/Q \to H$ by

$$R_i^{W/Q}(Q_u, Q_v) = \bigvee_{w,w' \in W} Q(u,w) \wedge R_i(w,w') \wedge Q(w',v), \tag{2.15}$$

or equivalently

$$R_i^{W/Q}(Q_u, Q_v) = (Q \circ R_i \circ Q)(u,v) = Q_u \circ R_i \circ Q^v, \tag{2.16}$$

for all $u, v \in W$. According to the statement (a) of Theorem 2.1, $R_i^{W/Q}$ is well-defined, for each $i \in I$, and we have that $\mathfrak{F}/Q = (W/Q, \{R_i^{W/Q}\}_{i\in I})$ is a fuzzy Kripke frame, called the *afterset fuzzy Kripke frame* of $\mathfrak{F}$ w.r.t. Q.

In addition, if $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ is a fuzzy Kripke model, then we define the fuzzy functions $R_i^{W/Q}$ as in (2.15), for every propositional variable $p \in PV$ we define a fuzzy set $V_p^{W/Q} \in \mathscr{F}(W/Q)$ by

$$V_p^{W/Q}(Q_w) = \bigvee_{u \in W} V_p(u) \wedge Q(u,w) = (V_p \circ Q)(w) = V_p \circ Q^w, \tag{2.17}$$

for any $w \in W$, and we define a function $V^{W/Q} : (W/Q) \times (PV \cup \overline{H}) \to H$ by

$$V^{W/Q}(Q_w, p) = V_p^{W/Q}(Q_w) \quad \text{and} \quad V^{W/Q}(Q_w, \overline{t}) = t,$$

for all $w \in W$, $p \in PV$ and $\overline{t} \in \overline{H}$. We inductively extend $V^{W/Q}$ to a function $V^{W/Q} : (W/Q) \times \Phi_{I,\mathscr{H}} \to H$ as in (V1)-(V6), and for each $A \in \Phi_{I,\mathscr{H}}$ we define a fuzzy set $V_A^{W/Q} \in \mathscr{F}(W/Q)$ by $V_A^{W/Q}(Q_w) = V^{W/Q}(Q_w, A)$, for each $A \in \Phi_{I,\mathscr{H}}$. Then we have that $\mathfrak{M}/Q = (W/Q, \{R_i^{W/Q}\}_{i\in I}, V^{W/Q})$ is a fuzzy Kripke model, which is called the *afterset fuzzy Kripke model* of $\mathfrak{M}$ w.r.t. Q. If E is a fuzzy equivalence, then $\mathfrak{M}/E$ will be called the *factor fuzzy Kripke model* of $\mathfrak{M}$ w.r.t. E.

In the same way, using foresets instead of aftersets, we can define the *foreset fuzzy Kripke model* of $\mathfrak{M}$ w.r.t. Q. However, this does not give anything new because the afterset and the foreset fuzzy Kripke models of $\mathfrak{M}$ w.r.t. Q are isomorphic.

The following theorem can be regarded as a counterpart of the well-known Second Isomorphism Theorem from algebra (cf. [17] §6). The proof of this theorem can be obtained directly from the proof of Theorem 3.3 from [133], so it is omitted.

Theorem 2.2. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model and let P and Q be fuzzy quasi-orders on $\mathfrak{M}$ such that $P \leqslant Q$. Then a fuzzy relation Q/P on W/P defined by*

$$Q/P(P_w, P_u) = Q(w, u), \qquad \text{for all } w, u \in W, \tag{2.18}$$

is a fuzzy quasi-order on W/P and fuzzy Kripke models $\mathfrak{M}/Q$ and $(\mathfrak{M}/P)/(Q/P)$ are isomorphic.

Remark 2.1. For any given fuzzy quasi-order Q on a fuzzy Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$, the rule $w \mapsto Q_w$ defines a surjective function of W onto W/Q. This means that the afterset fuzzy Kripke model $\mathfrak{M}/Q$ has smaller or equal size (cardinality) than the fuzzy Kripke model $\mathfrak{M}$.

The following two theorems are from [133], which will be useful in the further work. In the mentioned paper, the theorems concern the factor fuzzy automata, but we adapt them here for fuzzy Kripke models.

Theorem 2.3. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, and let E and F be fuzzy equivalences on W such that $E \leqslant F$. Then a relation $F/E \in \mathscr{F}(W/E)$ defined by:*

$$F/E(E_{w_1}, E_{w_2}) = F(w_1, w_2), \qquad E_{w_1}, E_{w_2} \in W/E, \tag{2.19}$$

is a fuzzy equivalence on W/E, and the afterset fuzzy Kripke model $(\mathfrak{M}/E)/(F/E)$ and $\mathfrak{M}/F$ are isomorphic.

Theorem 2.4. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model and E a fuzzy equivalence on W.*

The function $\Phi : \mathscr{E}_E(W) \to \mathscr{E}(W/E)$, where $\mathscr{E}_E(W) = \{F \in \mathscr{E}(W) \mid E \leqslant F\}$, defined by

$$\Phi(F) = F/E, \quad \text{for every } F \in \mathscr{E}_E(W), \tag{2.20}$$

is a lattice isomorphism, i.e., it is surjective and

$$F \leqslant G \;\Leftrightarrow\; \Phi(F) \leqslant \Phi(G), \quad \text{for every } F \in \mathscr{E}_E(W). \tag{2.21}$$

Chapter 3

Simulations and bisimulations

> "As complexity rises, precise statements lose meaning and meaningful statements lose precision."
>
> *Lotfi A. Zadeh.*

Simulations and bisimulations play an important role in the Fuzzy Automata Theory, Fuzzy Labelled Transition Systems (FLTS), Fuzzy Modal Logics, and other relational structures. Bisimulations are widely used for modeling equivalence between these systems, as well as in reducing the number of states of the systems. That is why it is especially important to study them as well as to develop algorithms for their computation.

Connections between bisimulations and fuzzy modal logic have been unexplored until recently, but have been intensively studied in recent years (for example see [42, 47, 72, 98, 144]), and also for a special type of fuzzy modal logic - fuzzy description logics (cf. [59, 96, 99, 100, 101, 102]). We also refer to [25, 26, 132, 146] for bisimulations on fuzzy automata and [20, 145] for FLTSs.

The motivation for the results in this chapter came from papers [25, 26] where two types of simulations and four types of bisimulations for fuzzy finite automata were introduced and an efficient algorithms for computation of the greatest simulation/bisimulation between two fuzzy automata on the residuated lattice are given.

In this chapter, we first define forward and backward simulations and two types of corresponding presimulations, which are simulations with relaxed conditions. Consequently, combining notions of forward and backward (pre)simulations we define forward, backward, forward-backward, and backward-forward (pre)bisimulations. We also introduce the fifth type of bisimulation, regular bisimulation, which originates from research on fuzzy social networks.

Then, we provide an efficient algorithms for deciding whether there is a (pre)simulation/(pre)bisimulation of the given type between the given fuzzy Kripke models, and for computing the greatest one, whenever it exists.

The algorithms are of the iterative type and work as follows: First, for each type of (pre)simulations and (pre)bisimulations we determine the corresponding isotone and image-localized function ϕ on the lattice of fuzzy relations. The corresponding initial fuzzy relation π is obtained from propositional variables in the model, i.e., depends only on truth assignment V. Then, the computation of the greatest (pre)simulation/(pre)bisimulation of this type is reduced to the computation of the greatest post-fixed point of ϕ contained in π, by applying Knaster-Tarski Fixed point Theorem. Starting from fuzzy relation π and by iteratively using the function ϕ, a decreasing sequence of fuzzy relations can be built. If this sequence is finite, then it stabilizes and its smallest member is exactly the fuzzy relation which we are searching for, the greatest post-fixed point of ϕ contained in π. In fact, the relation thus obtained is corresponding presimulation/prebisimulation and in order to get corresponding simulation/bisimulation we need to check one more condition from the definition of simulation/bisimulation.

As these algorithms do not always terminate in a finite number of steps, we also provide their modifications which determine whether there are crisp simulations or bisimulations of a given type, and compute the greatest ones when they exist. Such algorithms always terminate in finitely many steps. However, regardless of the existence of simulation/bisimulation of a given type, its corresponding crisp simulation/bisimulation does not have to exist, as will be seen from the examples.

Second, we provide an application of bisimulations in the state reduction of the fuzzy Kripke models, while preserving their semantic properties. In the case when forward, backward or regular bisimulation is fuzzy quasi-order, we create the corresponding afterset model with smaller sets of worlds which is equivalent to the original one with respect to plus-formulae, minus-formulae and all formulae.

The results from this chapter are presented in the paper [135].

The chapter consists of seven sections. In the first Section 3.1, we define simulations and bisimulations and provide propositions that give their basic properties. In Section 3.2 we provide already known definitions and properties of residuals. We also define isotone functions on the lattice of fuzzy relations on which we will later apply the Knaster-Tarski Fixed point Theorem. Section 3.3 provides one of the main results of the dissertation, where an Algorithm for testing the existence and computation of the greatest simulations and bisimulations is given. Then, in Section 3.4, we deal with the computation of crisp simulations and bisimulations. In Section 3.5, we present interesting computational examples which demonstrate applications of the results from the previous two sections. A method for reducing the number of states of fuzzy Kripke models is provided in Section 3.6. The last Section 3.7 provides interesting examples for state reduction of Kripke models.

3.1 Definitions of simulations and bisimulations

In the fuzzy modal logic, fuzzy simulation relates a fuzzy Kripke model to an *abstraction of the model* where the abstraction of the model might have a smaller set of worlds. The simulation guarantees that every local property of the fuzzy Kripke model is also a local property of its abstraction. More precisely, for every world $w \in W$, there is a corresponding world $w' \in W'$ which preserves local properties of w. Also, the simulation has to preserve the steps (represented by the accessibility fuzzy relations $\{R_i\}_{i\in I}$) in the abstraction of the model, but eliminate those steps through the model whose distinction is irrelevant to the simulation requirement. Therefore, fuzzy bisimulations guarantee that two fuzzy Kripke models have the same local properties.

Two types of simulations and four types of bisimulations for fuzzy automata were introduced in [25]. In a similar fashion, we also define two types of simulations and four types of bisimulations between two fuzzy Kripke models. Additionally, we define a fifth type of bisimulation called *regular bisimulation*, as in the case of social networks (cf. [71]). Each of these types of simulations and bisimulations is defined using an appropriate system of fuzzy relation inequations, consisting of three types of inequations.

Definition 3.1. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. If φ satisfies

$$V_p \leqslant V'_p \circ \varphi^{-1}, \qquad \text{for every } p \in PV, \tag{fs-1}$$
$$\varphi^{-1} \circ R_i \leqslant R'_i \circ \varphi^{-1}, \qquad \text{for every } i \in I, \tag{fs-2}$$
$$\varphi^{-1} \circ V_p \leqslant V'_p, \qquad \text{for every } p \in PV, \tag{fs-3}$$

then it is called a *forward simulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if it satisfies only (*fs-2*) and (*fs-3*), then it is called a *forward presimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$. On the other hand, if φ satisfies

$$V_p \leqslant \varphi \circ V'_p, \qquad \text{for every } p \in PV, \tag{bs-1}$$
$$R_i \circ \varphi \leqslant \varphi \circ R'_i, \qquad \text{for every } i \in I, \tag{bs-2}$$
$$V_p \circ \varphi \leqslant V'_p, \qquad \text{for every } p \in PV, \tag{bs-3}$$

then it is called a *backward simulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if it satisfies only (*bs-3*) and (*bs-2*), it is called a *backward presimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

Now, we can define five types of bisimulations by combining notions of forward and backward simulations.

Definition 3.2. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. If both φ and φ^{-1} are forward simulations, i.e., if

$$V_p \leqslant V'_p \circ \varphi^{-1}, \quad V'_p \leqslant V_p \circ \varphi, \qquad \text{for every } p \in PV, \tag{fb-1}$$
$$\varphi^{-1} \circ R_i \leqslant R'_i \circ \varphi^{-1}, \quad \varphi \circ R'_i \leqslant R_i \circ \varphi, \qquad \text{for every } i \in I, \tag{fb-2}$$
$$\varphi^{-1} \circ V_p \leqslant V'_p, \quad \varphi \circ V'_p \leqslant V_p, \qquad \text{for every } p \in PV. \tag{fb-3}$$

then we call φ a *forward bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if φ satisfies only (*fb-2*) and (*fb-3*), then we call it a *forward prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

Similarly, if both φ and φ^{-1} are backward simulation, i.e. if

$$V_p \leqslant \varphi \circ V'_p, \quad V'_p \leqslant \varphi^{-1} \circ V_p, \qquad \text{for every } p \in PV, \tag{bb-1}$$
$$R_i \circ \varphi \leqslant \varphi \circ R'_i, \quad R'_i \circ \varphi^{-1} \leqslant \varphi^{-1} \circ R_i, \qquad \text{for every } i \in I, \tag{bb-2}$$
$$V_p \circ \varphi \leqslant V'_p, \quad V'_p \circ \varphi^{-1} \leqslant V_p, \qquad \text{for every } p \in PV. \tag{bb-3}$$

then we call φ a *backward bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if φ satisfies only (*bb-2*) and (*bb-3*), then we call it a *backward prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

We also define two “mixed” types of bisimulations.

Definition 3.3. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. If φ is a forward simulation and φ^{-1} is a backward simulation, i.e., if

$$V_p \leqslant V_p' \circ \varphi^{-1}, \quad V_p' \leqslant V_p \circ \varphi^{-1}, \qquad \text{for every } p \in PV, \tag{fbb-1}$$
$$\varphi^{-1} \circ R_i = R_i' \circ \varphi^{-1}, \quad \text{for every } i \in I, \tag{fbb-2}$$
$$\varphi^{-1} \circ V_p \leqslant V_p', \quad V_p' \circ \varphi^{-1} \leqslant V_p, \qquad \text{for every } p \in PV, \tag{fbb-3}$$

then we say that φ is a *forward-backward bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if only (*fbb-2*) and (*fbb-3*) hold, we say that φ is a *forward-backward prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

Similarly, if φ is a backward simulation and φ^{-1} is a forward simulation, i.e., if

$$V_p' \leqslant V_p \circ \varphi, \quad V_p \leqslant \varphi \circ V_p', \qquad \text{for every } p \in PV, \tag{bfb-1}$$
$$\varphi \circ R_i' = R_i \circ \varphi, \qquad \text{for every } i \in I, \tag{bfb-2}$$
$$\varphi \circ V_p' \leqslant V_p, \quad V_p \circ \varphi \leqslant V_p', \qquad \text{for every } p \in PV, \tag{bfb-3}$$

then we say that φ is a *backward-forward bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if only (*bfb-2*) and (*bfb-3*) hold, then we say that φ is a *backward-forward prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

Finally, we can define regular bisimulation. Note that the prefix "regular" comes from the social network analysis (cf. [71, 134]).

Definition 3.4. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. If φ is both a forward and backward bisimulation, i.e., if

$$V_p \leqslant V_p' \circ \varphi^{-1}, \quad V_p' \leqslant V_p \circ \varphi, \quad V_p \leqslant \varphi \circ V_p', \quad V_p' \leqslant \varphi^{-1} \circ V_p, \tag{rb-1}$$
for every $p \in PV$,
$$\varphi^{-1} \circ R_i = R_i' \circ \varphi^{-1}, \quad \varphi \circ R_i' = R_i \circ \varphi, \qquad \text{for every } i \in I, \tag{rb-2}$$
$$\varphi^{-1} \circ V_p \leqslant V_p', \quad \varphi \circ V_p' \leqslant V_p, \quad V_p \circ \varphi \leqslant V_p', \quad V_p' \circ \varphi^{-1} \leqslant V_p, \tag{rb-3}$$
for every $p \in PV$,

then we call φ a *regular bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and if φ satisfies only (*rb-2*) and (*rb-3*), then we call it a *regular prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$.

For any $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, a fuzzy relation which satisfies (θ-*1*), (θ-*2*) and (θ-*3*) will be called a *simulation/bisimulation of type* θ or a θ*-simulation/bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$, and a fuzzy relation satisfying (θ-*2*) and (θ-*3*) will be called a *presimulation/prebisimulation of type* θ or a θ*-presimulation/prebisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$. In addition, if $\mathfrak{M}$ and $\mathfrak{M}'$ are the same fuzzy Kripke model, then we use the name *simulation/bisimulation of type* θ or θ*-simulation/bisimulation* on the fuzzy Kripke model $\mathfrak{M}$.

Lemma 3.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\varphi \in \mathscr{R}(W, W')$ be θ-(pre)simulation/(pre)bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. Then, the following hold:*

$$\varphi^{-1} \circ V_p = V_p \circ \varphi, \quad \text{for every } p \in PV, \tag{3.1}$$
$$\varphi \circ V_p' = V_p' \circ \varphi^{-1}, \quad \text{for every } p \in PV. \tag{3.2}$$

Proof. We will prove only the first case. Hence, we have

$$\begin{aligned}\varphi^{-1} \circ V_p(w') &= \bigvee_{w \in W} \varphi^{-1}(w', w) \wedge V_p(w) \\ &= \bigvee_{w \in W} V_p(w) \wedge \varphi(w, w') \\ &= V_p \circ \varphi(w')\end{aligned}$$

for every $w' \in W'$ and consequently, (3.1) holds for any propositional variable $p \in PV$. □

Using the previous Lemma, it follows that the definitions of forward and backward simulations/presimulations differ only in the second conditions (*fs-2*) and (*bs-2*), which are mutually dual. Similarly, the definitions of all five types of bisimulations/prebisimulations differ only in the second conditions (θ-*2*), for $\theta \in \{fb, bb, fbb, bfb, rb\}$, and conjunctions of conditions (θ-*1*) and (θ-*3*) in these definitions can be replaced by

$$V'_p = V_p \circ \varphi, \qquad V_p = \varphi \circ V'_p, \qquad \text{for every } p \in PV. \tag{3.3}$$

However, although the definitions of bisimulations with condition (3.3) seem simpler, in the further text we will see that the original definitions are much more suitable for testing the existence of bisimulations and computing the greatest ones, in cases when they exist.

The meaning of simulations and bisimulations can best be explained in the case when $\mathfrak{M}$ and $\mathfrak{M}'$ are crisp (Boolean-valued) Kripke models and φ is an ordinary crisp (Boolean-valued) binary relation. The condition (*fs-1*) means that if the valuation V assigns the value "true" to the propositional variable p in some world w, then the valuation V' assigns to this variable the value "true" in some world w' which simulates w. On the other hand, the condition (*fs-3*) means that if w' simulates w and the valuation V assigns the value "true" to the propositional variable p in the world w, then the valuation V' also assigns to this variable the value "true" in the world w'. The meaning of the conditions (*fs-2*) and (*bs-2*) can be explained as follows: (*fs-2*) means that if u' simulates u and v is accessible from u, then there is v' accessible from u' which simulates v, and (*bs-2*) means that if u is accessible from v and u' simulates u, then u' is accessible from some v' which simulates v. This is explained in Figure 3.1. In both cases, accessibility is considered with respect to R_i, for each $i \in I$.

Most researchers who have dealt with simulations and bisimulations in different contexts have considered only forward simulations and forward bisimulations, for which they have used the names strong simulations and strong bisimulations, or just simulations and bisimulations (cf., e.g., [47, 93, 94, 121]). The greatest bisimulation equivalence has usually been called a *bisimilarity*. However, our research is motivated by the study of different types of simulations and bisimulations between fuzzy automata, so here we also intend to study different types of simulations and bisimulations between Kripke models of fuzzy multimodal logics.

It has been noted in [25] that every forward simulation between two fuzzy automata is a backward simulation between the reverse fuzzy automata. This means that forward and backward simulations, forward and backward bisimulations, and

$$\begin{tikzcd}
u \arrow[r, "\varphi"] \arrow[d, "R_i"'] & u' \arrow[d, dashed, "R_i'"] \\
v \arrow[r, dashed, "\varphi"'] & v'
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
u \arrow[r, "\varphi"] & u' \\
v \arrow[u, "R_i"] \arrow[r, dashed, "\varphi"'] & v' \arrow[u, dashed, "R_i'"']
\end{tikzcd}$$

Figure 3.1: A forward simulation (the condition (*fs-2*), on the left) and backward simulation (the condition (*bs-2*), on the right)

backward-forward and forward-backward bisimulations, are mutually dual concepts. Here, we consider such duality for fuzzy Kripke models.

Let a mapping $\theta \mapsto \theta^d$ from the set $\{fs, bs, fb, bb, fbb, bfb, rb\}$ into itself be defined as follows:

$$\begin{pmatrix} fs & bs & fb & bb & fbb & bfb & rb \\ bs & fs & bb & fb & bfb & fbb & rb \end{pmatrix}$$

Now we can state and prove the following:

Theorem 3.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models, let $\mathfrak{M}^{-1} = (W, \{R_i^{-1}\}_{i\in I}, V)$ and $\mathfrak{M}'^{-1} = (W', \{R_i'^{-1}\}_{i\in I}, V')$ be the reverse fuzzy Kripke models for $\mathfrak{M}$ and $\mathfrak{M}'$, respectively, let $\varphi \in \mathscr{R}(W, W')$ be a fuzzy relation, and let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$.*

Then the following is true:

(a) *φ is a simulation/bisimulation of type θ between models $\mathfrak{M}$ and $\mathfrak{M}'$ if and only if φ is a simulation/bisimulation of type θ^d between the reverse fuzzy Kripke models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$.*

(b) *The assertion* (a) *remains valid if the terms simulation and bisimulation are replaced with presimulation and prebisimulation, respectively.*

Proof. We will prove only the assertion in (a) concerning the case $\theta = fs$. The others can be proved similarly.

Let φ be forward simulation between $\mathfrak{M}$ and $\mathfrak{M}'$, i.e., let φ satisfy (*fs-1*), (*fs-2*) and (*fs-3*). As we know, conditions (*fs-1*) and (*fs-3*) can be easily transformed into (*bs-1*) and (*bs-3*), respectively, using (3.1) and (3.2).

Also, for each $i \in I$ we have that

$$\varphi^{-1} \circ R_i \leqslant R_i' \circ \varphi^{-1} \quad \Rightarrow \quad \left(\varphi^{-1} \circ R_i\right)^{-1} \leqslant \left(R_i' \circ \varphi^{-1}\right)^{-1} \quad \Rightarrow \quad R_i^{-1} \circ \varphi \leqslant \varphi \circ R_i'^{-1},$$

and it follows that φ satisfies (*bs-2*) for models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$. $\square$

We also state the following lemma that can be easily proved.

Lemma 3.2. *Let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$.*

(a) *If $\{\varphi_\alpha\}_{\alpha\in Y}$ are simulations/bisimulations of type θ between models $\mathfrak{M}$ and $\mathfrak{M}'$, then $\bigvee_{\alpha\in Y} \varphi_\alpha$ is also a simulation/bisimulation of type θ between these models.*

(b) *If φ_1 is a simulation/bisimulation of type θ between models $\mathfrak{M}$ and $\mathfrak{M}'$ and φ_2 is a simulation/bisimulation of type θ between models $\mathfrak{M}'$ and $\mathfrak{M}''$, then $\varphi_1 \circ \varphi_2$ is a simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}''$.*

(c) *The assertions* (a) *and* (b) *remain valid if the terms simulation and bisimulation are replaced with presimulation and prebisimulation, respectively.*

3.2 The residuals

Now, several useful notions and notation will be introduced in the same manner as in [26].

For non-empty sets of worlds W and W' and fuzzy subsets $\eta \in \mathscr{F}(W)$ and $\xi \in \mathscr{F}(W')$, fuzzy relations $\eta\backslash\xi \in \mathscr{R}(W,W')$ and $\eta/\xi \in \mathscr{R}(W,W')$ are defined as follows:

$$(\eta\backslash\xi)(w,w') = \eta(w) \to \xi(w'), \tag{3.4}$$

$$(\eta/\xi)(w,w') = \xi(w') \to \eta(w), \tag{3.5}$$

for arbitrary $w \in W$ and $w' \in W'$. Let us note that $\eta\backslash\xi = (\xi/\eta)^{-1}$ and $\eta/\xi = (\xi\backslash\eta)^{-1}$.

Next we state the well-know results by Sanchez (cf. [124, 125, 126]).

Lemma 3.3. *Let W and W' be non-empty sets of worlds and let $\eta \in \mathscr{F}(W)$ and $\xi \in \mathscr{F}(W')$.*

(a) *The set of all solutions to the inequation $\eta \circ \chi \leqslant \xi$, where χ is an unknown fuzzy relation between W and W', is the principal ideal of $\mathscr{R}(W,W')$ generated by the fuzzy relation $\eta\backslash\xi$.*

(b) *The set of all solutions to the inequation $\chi \circ \xi \leqslant \eta$, where χ is an unknown fuzzy relation between W and W', is the principal ideal of $\mathscr{R}(W,W')$ generated by the fuzzy relation η/ξ.*

In other words, the following residuation properties hold:

$$\eta \circ \chi \leqslant \xi \quad \Leftrightarrow \quad \chi \leqslant \eta\backslash\xi, \qquad \chi \circ \xi \leqslant \eta \quad \Leftrightarrow \quad \chi \leqslant \eta/\xi. \tag{3.6}$$

Note that $(\eta\backslash\xi) \wedge (\eta/\xi) = \eta \leftrightarrow \xi$, where $\eta \leftrightarrow \xi$ is a fuzzy relation between W and W' defined by

$$(\eta \leftrightarrow \xi)(w,w') = \eta(w) \leftrightarrow \xi(w'), \tag{3.7}$$

for arbitrary $w \in W$ and $w' \in W'$.

Next, let W and W' be non-empty sets of worlds and let $\alpha \in \mathscr{R}(W)$, $\beta \in \mathscr{R}(W')$ and $\gamma \in \mathscr{R}(W,W')$. The *right residual* of γ by α is a fuzzy relation $\alpha\backslash\gamma \in \mathscr{R}(W,W')$ defined by

$$(\alpha\backslash\gamma)(w,w') = \bigwedge_{u\in W} \alpha(u,w) \to \gamma(u,w'), \tag{3.8}$$

for all $w \in W$ and $w' \in W'$, and the *left residual* of γ by β is a fuzzy relation $\gamma/\beta \in \mathscr{R}(W,W')$ defined by

$$(\gamma/\beta)(w,w') = \bigwedge_{u'\in W'} \beta(w',u') \to \gamma(w,u'), \tag{3.9}$$

for all $w \in W$ and $w' \in W'$. We think of the right residual $\alpha\backslash\gamma$ as what remains of on the right after "dividing" γ on the left by α, and of the left residual γ/β as what remains of γ on the left after "dividing" γ on the right by β. In other words,

$$\alpha \circ \gamma' \leqslant \gamma \quad \Leftrightarrow \quad \gamma' \leqslant \alpha\backslash\gamma, \qquad \gamma' \circ \beta \leqslant \gamma \quad \Leftrightarrow \quad \gamma' \leqslant \gamma/\beta, \tag{3.10}$$

for all $\alpha \in \mathscr{R}(W)$, $\beta \in \mathscr{R}(W')$ and $\gamma', \gamma \in \mathscr{R}(W, W')$. In the case when $W = W'$, these two concepts become the well-known concepts of right and left residuals of fuzzy relations on a set (cf. [68]).

The following statements in the next lemma are also results by Sanchez (cf. [124, 125, 126]).

Lemma 3.4. *Let W and W' be non-empty sets of worlds and let $\alpha \in \mathscr{R}(W)$, $\beta \in \mathscr{R}(W')$ and $\gamma \in \mathscr{R}(W, W')$.*

(a) *The set of all solutions to the inequation $\alpha \circ \chi \leqslant \gamma$, where χ is an unknown fuzzy relation between W and W', is the principal ideal of $\mathscr{R}(W, W')$ generated by the right residual $\alpha\backslash\gamma$ of γ by α.*
(b) *The set of all solutions to the inequation $\chi \circ \beta \leqslant \gamma$, where χ is an unknown fuzzy relation between W and W', is the principal ideal of $\mathscr{R}(W, W')$ generated by the left residual γ/β of γ by β.*

Now, we will define isotone function ϕ on the lattice of fuzzy relations by which we will reduce problem of computation of the greatest (pre)simulation/(pre)bisimulation to the problem of computing the greatest post-fixed point, contained in a given fuzzy relation. Let's emphasize once again that greatest simulation/bisimulation do not always have to exist and in that case we just have decision-making procedure whether there is a simulation or bisimulation of a given type. First, we define initial fuzzy relations which are obtained from residuals and propositional variables in the model.

Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models. We define fuzzy relations $\pi^\theta \in \mathscr{R}(W, W')$, for $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, in the following way:

$$\pi^{fs} = \pi^{bs} = \bigwedge_{p\in PV} (V_p\backslash V'_p), \tag{3.11}$$

$$\pi^{fb} = \pi^{bb} = \pi^{fbb} = \pi^{bfb} = \pi^{rb} = \bigwedge_{p\in PV} [(V_p\backslash V'_p) \wedge (V_p/V'_p)] = \bigwedge_{p\in PV} (V_p \leftrightarrow V'_p). \tag{3.12}$$

Moreover, we define functions $\phi^\theta : \mathscr{R}(W, W') \to \mathscr{R}(W, W')$, for $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, as follows:

$$\phi^{fs}(\varphi) = \bigwedge_{i\in I} [(R'_i \circ \varphi^{-1})/R_i]^{-1}, \tag{3.13}$$

$$\phi^{bs}(\varphi) = \bigwedge_{i\in I} R_i\backslash(\varphi \circ R'_i), \tag{3.14}$$

$$\phi^{fb}(\varphi) = \bigwedge_{i\in I} [(R'_i \circ \varphi^{-1})/R_i]^{-1} \wedge [(R_i \circ \varphi)/R'_i] = \phi^{fs}(\varphi) \wedge [\phi^{fs}(\varphi^{-1})]^{-1}, \tag{3.15}$$

$$\phi^{bb}(\varphi) = \bigwedge_{i\in I} [R_i\backslash(\varphi \circ R'_i)] \wedge [R'_i\backslash(\varphi^{-1} \circ R_i)]^{-1} = \phi^{bs}(\varphi) \wedge [\phi^{bs}(\varphi^{-1})]^{-1}, \tag{3.16}$$

$$\phi^{fbb}(\varphi) = \bigwedge_{i\in I}[(R'_i \circ \varphi^{-1})/R_i]^{-1} \wedge [R'_i\backslash(\varphi^{-1} \circ R_i)]^{-1} = \phi^{fs}(\varphi) \wedge [\phi^{bs}(\varphi^{-1})]^{-1}, \quad (3.17)$$

$$\phi^{bfb}(\varphi) = \bigwedge_{i\in I}[R_i\backslash(\varphi \circ R'_i)] \wedge [(R_i \circ \varphi)/R'_i] = \phi^{bs}(\varphi) \wedge [\phi^{fs}(\varphi^{-1})]^{-1}, \quad (3.18)$$

$$\begin{aligned}\phi^{rb}(\varphi) &= \bigwedge_{i\in I}[R_i\backslash(\varphi \circ R'_i)] \wedge [(R_i \circ \varphi)/R'_i] \wedge [(R'_i \circ \varphi^{-1})/R_i]^{-1} \wedge [R'_i\backslash(\varphi^{-1} \circ R_i)]^{-1} \\ &= \phi^{fs}(\varphi) \wedge [\phi^{bs}(\varphi^{-1})]^{-1} \wedge \phi^{bs}(\varphi) \wedge [\phi^{fs}(\varphi^{-1})]^{-1} = \phi^{fb}(\varphi) \wedge \phi^{bb}(\varphi), \quad (3.19)\end{aligned}$$

for any $\varphi \in \mathscr{R}(W, W')$. Notice that in the expression "$\phi^\theta(\alpha^{-1})$" ($\theta \in \{fs, bs\}$) a function from $\mathscr{R}(W', W)$ into itself is denoted by ϕ^θ.

The following theorem provides alternative forms of the second and third conditions in the definitions of simulations and bisimulations, using initial fuzzy relations π^θ, and the corresponding functions ϕ^θ for $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$. These forms are more suitable for construction of algorithms that will be given in the sequel.

Theorem 3.2. *Let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$ and let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models. A fuzzy relation $\varphi \in \mathscr{R}(W, W')$ satisfies the conditions (θ-2) and (θ-3) if and only if it satisfies*

$$\varphi \leqslant \phi^\theta(\varphi), \qquad \varphi \leqslant \pi^\theta. \quad (3.20)$$

Proof. We will prove only the case $\theta = fs$. The assertion concerning the case $\theta = bs$ follows by the duality, and according to Eqs. (3.12) and (3.15)-(3.19), all other assertions can be obtained by the first two.

Consider an arbitrary $\varphi \in \mathscr{R}(W, W')$. According to Lemma 3.3(b), φ satisfies the condition (*fs-3*) if and only if $\varphi^{-1} \leqslant V'_p/V_p = (V_p\backslash V'_p)^{-1}$, for all $p \in PV$, which is equivalent to $\varphi \leqslant V_p\backslash V'_p$, for all $p \in PV$. Hence, we have

$$\varphi \leqslant \bigwedge_{p\in PV}(V_p\backslash V'_p) = \pi^{fs}.$$

Therefore, φ satisfies (*fs-3*) if and only if $\varphi \leqslant \pi^{fs}$.

On the other hand, φ satisfies (*fs-2*) if and only if

$$\varphi^{-1}(w', w) \wedge R_i(w, u) \leqslant (R'_i \circ \varphi^{-1})(w', u),$$

for all $w, u \in W$, $w' \in W'$ and $i \in I$. According to the adjunction property (1.68), this is equivalent to

$$\varphi^{-1}(w', w) \leqslant \bigwedge_{u\in W}[R_i(w, u) \to (R'_i \circ \varphi^{-1})(w', u))] = ((R'_i \circ \varphi^{-1})/R_i)(w', w),$$

for all $w \in W$, $w' \in W'$ and $i \in I$, which is further equivalent to

$$\varphi(w, w') \leqslant \bigwedge_{i\in I}[(R'_i \circ \varphi^{-1})/R_i]^{-1}(w, w') = (\phi^{fs}(\varphi))(w, w'),$$

for all $w \in W$ and $w' \in W'$. Therefore, φ satisfies (*fs-3*) if and only if $\varphi \leqslant \phi^{fs}(\varphi)$.

Now, we conclude that a fuzzy relation $\varphi \in \mathscr{R}(W, W')$ satisfies (*fs-2*) and (*fs-3*) if and only if it satisfies (3.20) (for $\theta = fs$), which has to be proved. □

3.3 Testing the existence and computing the greatest simulations and bisimulations

In this section we provide a method for testing the existence of simulations and bisimulations between fuzzy Kripke models, and for computing the greatest ones, in the cases when they exist.

Let W and W' be non-empty sets of worlds and let $\phi : \mathscr{R}(W,W') \to \mathscr{R}(W,W')$ be an isotone function, i.e., let $\alpha \leqslant \beta$ implies $\phi(\alpha) \leqslant \phi(\beta)$, for all $\alpha, \beta \in \mathscr{R}(W,W')$. Then, according to the Definition 1.25, a fuzzy relation $\alpha \in \mathscr{R}(W,W')$ is called a post-fixed point of ϕ if $\alpha \leqslant \phi(\alpha)$, and is called a fixed point of ϕ if $\alpha = \phi(\alpha)$. The Knaster-Tarski fixed point theorem 1.6 (stated and proved in a more general context, for complete lattices) asserts that the set of all post-fixed points of ϕ form a complete lattice (for more details see [122]). Moreover, for any fuzzy relation $\pi \in \mathscr{R}(W,W')$ we have that the set of all post-fixed points of ϕ contained in π is also a complete lattice. According to Theorem 3.2, our main task is to find an efficient procedure for computing the greatest post-fixed point of the function ϕ^θ contained in the fuzzy relation π^θ, for each $\theta = \{fs, bs, fb, bb, fbb, bfb, rb\}$.

Note that the set of all post-fixed points of an isotone function on a complete lattice less than or equal to a given element is always non-empty, because it contains the least element of this lattice. However, a trivial case may occur that this set consist only of that single element. In our case, since we are dealing with a complete lattice of fuzzy relations and isotone functions on it of the form ϕ^θ, the empty relation may be the only post-fixed point contained in the corresponding fuzzy relation π^θ, and in this case there is not any simulation/bisimulation of type θ. We remember that we defined simulations and bisimulations, as well as presimulations and prebisimulations, so that they must be non-empty.

If the set of all post-fixed points of the function ϕ^θ contained in π^θ includes at least one non-empty fuzzy relation, then the greatest post-fixed point of ϕ^θ contained in π^θ is non-empty, and we will see that it is the greatest presimulation/prebisimulation of type θ, but it is not necessary a simulation/bisimulation of this type. We will show that it can be easily tested whether this greatest presimulation/prebisimulation of type θ is a simulation/bisimulation of this type, by simply checking if it satisfies the condition (θ-*1*).

Therefore, our task is actually to find an efficient procedure for computing the greatest post-fixed point of ϕ^θ contained in π^θ, and to check if it is non-empty and if it satisfies the condition (θ-*1*).

Let $\phi : \mathscr{R}(W,W') \to \mathscr{R}(W,W')$ be an isotone function and $\pi \in \mathscr{R}(W,W')$. We define a sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ of fuzzy relations from $\mathscr{R}(W,W')$ by

$$\varphi_1 = \pi, \qquad \varphi_{k+1} = \varphi_k \wedge \phi(\varphi_k) \text{ for each } k \in \mathbb{N}. \tag{3.21}$$

The sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is obviously descending. If we denote by $\hat{\varphi}$ the greatest post-fixed point of ϕ contained in π, we can verify that

$$\hat{\varphi} \leqslant \bigwedge_{k\in\mathbb{N}} \varphi_k. \tag{3.22}$$

Now, two questions arise. First, under what conditions do the equality in (3.22) hold? Second, under what conditions this sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite? If this sequence is finite, then it is not hard to show that there exists $k \in \mathbb{N}$ such that $\varphi_k = \varphi_m$, for every $m \geqslant k$, i.e., there exists $k \in \mathbb{N}$ such that the sequence stabilizes on k. We can recognize that the sequence has stabilized when we find the smallest $k \in \mathbb{N}$ such that $\varphi_k = \varphi_{k+1}$. In this case $\hat{\varphi} = \varphi_k$, and we have an algorithm which computes $\hat{\varphi}$ in a finite number of steps. Some conditions under which equality holds in (3.22) or the sequence is finite can be found in [68, 69].

The next two theorems are essentially proved in [69] (see also [26]), but for the sake of completeness we state them here.

A sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ of fuzzy relations from $\mathscr{R}(W,W')$ is called *image-finite* if the set $\bigcup_{k\in\mathbb{N}} \mathrm{Im}(\varphi_k)$ is finite. Clearly, $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite if and only if it is image-finite. Next, a function $\phi : \mathscr{R}(W,W') \to \mathscr{R}(W,W')$ is called *image-localized* if there exists a finite $K \subset H$ such that for each fuzzy relation $\varphi \in \mathscr{R}(W,W')$ we have

$$\mathrm{Im}(\phi(\varphi)) \subseteq \langle K \cup \mathrm{Im}(\varphi)\rangle. \tag{3.23}$$

Such K will be called a *localization set* of the function ϕ.

Theorem 3.3. *Let the function ϕ be image-localized, let K be its localization set, let $\pi \in \mathscr{R}(W,W')$, and let $\{\varphi_k\}_{k\in\mathbb{N}}$ be the sequence of fuzzy relations in $\mathscr{R}(W,W')$ defined by (3.21). Then*

$$\bigcup_{k\in\mathbb{N}} \mathrm{Im}(\varphi_k) \subseteq \langle K \cup \mathrm{Im}(\pi)\rangle. \tag{3.24}$$

If, moreover, $\langle K \cup \mathrm{Im}(\pi)\rangle$ is a finite subalgebra of $\mathscr{H}$, then the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite.

Theorem 3.4. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two image-finite fuzzy Kripke models.*

For any $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$ the function ϕ^θ is isotone and image-localized.

Proof. Let $\varphi_1, \varphi_2 \in \mathscr{R}(W,W')$ be fuzzy relation such that $\varphi_1 \leqslant \varphi_2$, and consider the following systems of fuzzy relation inequations:

$$\chi^{-1} \circ R_i \leqslant R'_i \circ \varphi_1^{-1}, \qquad \text{for every } i \in I, \tag{3.25}$$
$$\chi^{-1} \circ R_i \leqslant R'_i \circ \varphi_2^{-1}, \qquad \text{for every } i \in I, \tag{3.26}$$

where $\chi \in \mathscr{R}(W,W')$ is an unknown fuzzy relation. Using Lemma 3.3(b) and the definition of an inverse relation, it can be easily shown that the set of all solutions to system (3.25) (resp. (3.26)) form a principal ideal of $\mathscr{R}(W,W')$ generated by $\phi^{fs}(\varphi_1)$ (resp. $\phi^{fs}(\varphi_2)$). Since for every $i \in I$ we have that $R'_i \circ \varphi_1^{-1} \leqslant R'_i \circ \varphi_2^{-1}$, we conclude that every solution to (3.25) is a solution to (3.26). Consequently, $\phi^{fs}(\varphi_1)$ is a solution to (3.26), so $\phi^{fs}(\varphi_1) \leqslant \phi^{fs}(\varphi_2)$. Therefore, we proved that ϕ^{fs} is an isotone function.

Next, let $K = \bigcup_{i\in I}(\mathrm{Im}(R_i) \cup \mathrm{Im}(R'_i))$ and let $\varphi \in \mathscr{R}(W,W')$ be an arbitrary fuzzy relation. It is evident that $\mathrm{Im}(\phi^{fs}(\varphi)) \subseteq \langle K \cup \mathrm{Im}(\varphi)\rangle$, and since fuzzy relations R_i and R'_i are image-finite, for every $i \in I$, then K is also finite. This confirms that the function ϕ^{fs} is image-localized. □

Now we are ready for the main result of the chapter. The next theorem provides algorithms for computing the greatest presimulations or prebisimulations of a given type between two fuzzy Kripke models and consequently gain the greatest simulations or bisimulations of a given type, when they exist.

Theorem 3.5. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be fuzzy Kripke models, let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, and let a sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ of fuzzy relations from $\mathscr{R}(W, W')$ be defined by*

$$\varphi_1 = \pi^\theta, \qquad \varphi_{k+1} = \varphi_k \wedge \phi^\theta(\varphi_k) \qquad \textit{for each } k \in \mathbb{N}. \tag{3.27}$$

If $\langle \mathrm{Im}(\pi^\theta) \cup \bigcup_{i\in I}(\mathrm{Im}(R_i) \cup \mathrm{Im}(R_i'))\rangle$ is a finite subalgebra of $\mathscr{H}$, then the following is true:

(a) *the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite and descending, and there is the least natural number k such that $\varphi_k = \varphi_{k+1}$;*
(b) *if φ_k is non-empty, then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-2) and (θ-3), i.e., φ_k is the greatest presimulation/prebisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*
(c) *if φ_k is non-empty and satisfies (θ-1), then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., φ_k is the greatest simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*
(d) *if φ_k is empty or does not satisfy (θ-1), then there is not any fuzzy relation in $\mathscr{R}(W, W')$ satisfying (θ-1), (θ-2), and (θ-3), i.e., there is not any simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. We will prove only the case $\theta = fs$. All other cases can be proved in a similar manner.

So, let $\langle \mathrm{Im}(\pi)^\theta \cup \bigcup_{i\in I}(\mathrm{Im}(R_i) \cup \mathrm{Im}(R_i'))\rangle$ be a finite subalgebra of $\mathscr{H}$.

(a) According to Theorems 3.4 and 3.3, the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite and descending, so there are $k, m \in \mathbb{N}$ such that $\varphi_k = \varphi_{k+m}$, whence $\varphi_{k+1} \leqslant \varphi_k = \varphi_{k+m} \leqslant \varphi_{k+1}$. Thus, there is $k \in \mathbb{N}$ such that $\varphi_k = \varphi_{k+1}$, and consequently, there is the least natural number having this property.

(b) By $\varphi_k = \varphi_{k+1} = \varphi_k \wedge \phi^{fs}(\varphi_k)$ we obtain that $\varphi_k \leqslant \phi^{fs}(\varphi_k)$, and also, $\varphi_k \leqslant \varphi_1 = \pi^{fs}$. Therefore, by Theorem 3.2 it follows that φ_k satisfies (*fs-2*) and (*fs-3*).

Let $\alpha \in \mathscr{R}(W, W')$ be an arbitrary fuzzy relation which satisfies (*fs-2*) and (*fs-3*). As we have already noted, α satisfies (*fs-3*) if and only if $\alpha \leqslant \pi^{fs} = \varphi_1$. Next, suppose that $\alpha \leqslant \varphi_n$, for some $n \in \mathbb{N}$. Then for every $i \in I$ we have that $\alpha^{-1} \circ R_i \leqslant R_i' \circ \alpha^{-1} \leqslant R_i' \circ \varphi_n^{-1}$, and according to Lemma 3.4(b), $\alpha^{-1} \leqslant (R_i' \circ \varphi_n^{-1})/R_i$, i.e., $\alpha \leqslant [(R_i' \circ \varphi_n^{-1})/R_i]^{-1} = \phi^{fs}(\varphi_n)$. Therefore, $\alpha \leqslant \varphi_n \wedge \phi^{fs}(\varphi_n) = \varphi_{n+1}$. Now, by induction we obtain that $\alpha \leqslant \varphi_n$, for every $n \in \mathbb{N}$, and hence, $\alpha \leqslant \varphi_k$. This means that φ_k is the greatest fuzzy relation in $\mathscr{R}(W, W')$ satisfying (*fs-2*) and (*fs-3*).

(c) This follows immediately from (b).

(d) Suppose that φ_k does not satisfy (*fs-1*). Let $\varphi \in \mathscr{R}(W, W')$ be an arbitrary fuzzy relation which satisfies (*fs-1*), (*fs-2*) and (*fs-3*). According to (b) of this theorem, $\varphi \leqslant \varphi_k$, so we have that $R_i \leqslant R_i' \circ \varphi^{-1} \leqslant R_i' \circ \varphi_k^{-1}$. But, this contradicts our starting assumption that φ_k does not satisfy (*fs-1*). Hence, we conclude that there is not any fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (*fs-1*), (*fs-2*) and (*fs-3*). $\square$

Algorithm 3.1. [*Testing the existence and computing the greatest simulations and bisimulation*] The input of this algorithm are two fuzzy Kripke models $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$. The algorithm decides whether there is a simulation or bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ of a given type $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, and when it exists, the output of the algorithm is the greatest simulation/bisimulation of type θ.

The procedure is to construct a sequence of fuzzy relations $\{\varphi_k\}_{k\in\mathbb{N}}$, in the following way:

(A1) In the first step we compute π^θ and we set $\varphi_1 = \pi^\theta$.

(A2) After the kth step let a fuzzy relation φ_k has been constructed.

(A3) In the next step we construct the fuzzy relation φ_{k+1} by means of the formula $\varphi_{k+1} = \varphi_k \wedge \phi^\theta(\varphi_k)$.

(A4) Simultaneously, we check whether $\varphi_{k+1} = \varphi_k$.

(A5) The first time we find a number k such that $\varphi_{k+1} = \varphi_k$, the procedure of constructing the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ terminates, and if φ_k is non-empty, then it is the greatest presimulation/prebisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ of type θ. If φ_k is empty, then there is not any presimulation/prebisimulation nor simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;

(A6) If φ_k is non-empty, in the final step we check whether it satisfies (θ-*1*). If φ_k satisfies (θ-*1*), then it is the greatest simulation/bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ of type θ, and if φ_k does not satisfy (θ-*1*), then there is not any simulation/bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ of type θ.

If the underlying Heyting algebra $\mathscr{H}$ is locally finite, in the sense that each finitely generated subalgebra of $\mathscr{H}$ is finite, then the algorithm terminates in a finite number of steps, for arbitrary finite fuzzy Kripke models over $\mathscr{H}$. Inter alia, examples of locally finite Heyting algebras include Gödel algebras and linearly ordered Heyting algebras. On the other hand, if $\mathscr{H}$ is not locally finite, then the algorithm terminates in a finite number of steps under conditions determined by Theorems 3.3 and 3.5.

However, regardless of the local finiteness of the underlying Heyting algebra and the fulfillment of the conditions of Theorems 3.3 and 3.5, the conditions under which there exists the greatest simulation/bisimulation of a given type and the greatest simulation/bisimulation itself are characterized by the following theorem.

If the underlying Heyting algebra $\mathscr{H}$ satisfies condition (1.77) from Lemma 1.4, we have the following.

Theorem 3.6. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two finite fuzzy Kripke models, let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, let $\{\varphi_k\}_{k\in\mathbb{N}}$ be the sequence of fuzzy relations from $\mathscr{R}(W, W')$ defined by* (3.27)*, and let*

$$\varphi = \bigwedge_{k\in\mathbb{N}} \varphi_k. \tag{3.28}$$

Then the following is true:

(a) *if φ is non-empty, then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-2) and (θ-3), i.e., it is the greatest presimulation/prebisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(b) *if φ is non-empty and satisfies (θ-1), then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., it is the greatest simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(c) *if φ is empty or does not satisfy (θ-1), then there is not any fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., there is not any simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. Only the case $\theta = fs$ will be proved. All other cases can be proved similarly.

(a) For arbitrary $i \in I$, $w \in W$ and $w' \in W'$ we have that

$$\left(\bigwedge_{k\in\mathbb{N}}(R_i' \circ \varphi_k^{-1})\right)(w', w) = \bigwedge_{k\in\mathbb{N}}(R_i' \circ \varphi_k^{-1})(w', w) = \bigwedge_{k\in\mathbb{N}}\left(\bigvee_{u'\in W'} R_i'(w', u') \wedge \varphi_k^{-1}(u', w)\right)$$

$$= \bigvee_{u'\in W'}\left(\bigwedge_{k\in\mathbb{N}} R_i'(w', u') \wedge \varphi_k^{-1}(u', w)\right) \qquad \text{(by (1.78))}$$

$$= \bigvee_{u'\in W'}\left(R_i'(w', u') \wedge \left(\bigwedge_{k\in\mathbb{N}} \varphi_k^{-1}(u', w)\right)\right) \qquad \text{(by (1.73))}$$

$$= \bigvee_{u'\in W'}\big(R_i'(w', u') \wedge \varphi^{-1}(u', w)\big) = (R_i' \circ \varphi^{-1})(w', w),$$

which means that

$$\bigwedge_{k\in\mathbb{N}} R_i' \circ \varphi_k^{-1} = R_i' \circ \varphi^{-1},$$

for every $i \in I$. The use of condition (1.78) is justified by the facts that W' is finite, and that $\{\varphi_k^{-1}(u', w)\}_{k\in\mathbb{N}}$ is a non-increasing sequence, so $\{R_i'(w', u') \wedge \varphi_k^{-1}(u', w)\}_{k\in\mathbb{N}}$ is also a non-increasing sequence.

Now, for all $k \in \mathbb{N}$ we have that

$$\varphi \leqslant \varphi_{k+1} \leqslant \phi^{fs}(\varphi_k) = [(R_i' \circ \varphi_k^{-1})/R_i]^{-1},$$

which is equivalent to

$$\varphi^{-1} \circ R_i \leqslant R_i' \circ \varphi_k^{-1}.$$

As the last inequation holds for every $k \in \mathbb{N}$ we have that

$$\varphi^{-1} \circ R_i \leqslant \bigwedge_{k\in\mathbb{N}} R_i' \circ \varphi_k^{-1} = R_i' \circ \varphi^{-1},$$

for every $i \in I$. Therefore, φ satisfies (*fs-2*). Moreover, $\varphi \leqslant \varphi_1 = \pi^{fs}$, so φ also satisfies (*fs-3*).

Next, let $\alpha \in \mathscr{R}(W, W')$ be an arbitrary fuzzy relation satisfying (*fs-2*) and (*fs-3*). According to Theorem 3.2, $\alpha \leqslant \phi^{fs}(\alpha)$ and $\alpha \leqslant \pi^{fs} = \varphi_1$. By induction, we can easily prove that $\alpha \leqslant \varphi_k$ for every $k \in \mathbb{N}$, therefore, $\alpha \leqslant \varphi$. This means that φ is the greatest fuzzy relation $\mathscr{R}(W, W')$ which satisfies (*fs-2*) and (*fs-3*).

The assertion (b) follows immediately from (a), whereas the assertion (c) can be proved in the same way as the assertion (d) of Theorem 3.5. □

We can formulate the following theorem using block representation and the Definition 2.15 of the disjoint union of models.

Theorem 3.7. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models and $\mathfrak{M}''$ their disjoint union, and let φ be a fuzzy relation on $\mathfrak{M}''$ with the block representation:*

$$\varphi = \begin{bmatrix} \varphi_{W\times W} & \varphi_{W\times W'} \\ \varphi_{W'\times W} & \varphi_{W'\times W'} \end{bmatrix}. \tag{3.29}$$

Then φ is the greatest θ-simulation/bisimulation on $\mathfrak{M}''$ for $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$ if and only if the following statements are true:

(a) $\varphi_{W\times W}$ *is the greatest θ-simulation/bisimulation on $\mathfrak{M}$;*

(b) $\varphi_{W\times W'}$ *is the greatest θ-presimulation/prebisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(c) $\varphi_{W'\times W}$ *is the greatest θ-presimulation/prebisimulation between $\mathfrak{M}'$ and $\mathfrak{M}$;*

(d) $\varphi_{W'\times W'}$ *is the greatest θ-simulation/bisimulation on $\mathfrak{M}'$.*

Proof. We will prove only the case when $\theta = fs$. The other cases can be proved similarly. Also, to avoid unnecessarily complicating the notations, we will omit the symbol p from V_p and index i from R_i whenever it is clear from the context.

Let φ be the greatest forward simulation on $\mathfrak{M}''$. Then, from (*fs-3*) it follows $\varphi^{-1} \circ V''_p \leqslant V''_p$ for every $p \in PV$:

$$\begin{aligned}\varphi^{-1} \circ V''_p &= \begin{bmatrix} \varphi^{-1}_{W\times W} & \varphi^{-1}_{W'\times W} \\ \varphi^{-1}_{W\times W'} & \varphi^{-1}_{W'\times W'} \end{bmatrix} \circ \begin{bmatrix} V_W \\ V'_{W'} \end{bmatrix} \\ &= \begin{bmatrix} \varphi^{-1}_{W\times W} \circ V_W \vee \varphi^{-1}_{W'\times W} \circ V'_{W'} \\ \varphi^{-1}_{W\times W'} \circ V_W \vee \varphi^{-1}_{W'\times W'} \circ V'_{W'} \end{bmatrix} \leqslant \begin{bmatrix} V_W \\ V'_{W'} \end{bmatrix}.\end{aligned}$$

Now, we have:

$$\begin{aligned}\varphi^{-1}_{W\times W} \circ V_W \vee \varphi^{-1}_{W'\times W} \circ V'_{W'} &\leqslant V_W \\ \varphi^{-1}_{W\times W'} \circ V_W \vee \varphi^{-1}_{W'\times W'} \circ V'_{W'} &\leqslant V'_{W'}\end{aligned}$$

and it follows,

$$\begin{aligned}\varphi^{-1}_{W\times W} \circ V_W &\leqslant V_W \\ \varphi^{-1}_{W'\times W} \circ V'_{W'} &\leqslant V_W \\ \varphi^{-1}_{W\times W'} \circ V_W &\leqslant V'_{W'} \\ \varphi^{-1}_{W'\times W'} \circ V'_{W'} &\leqslant V'_{W'}\end{aligned}$$

and it follows that $\varphi_{W\times W}$, $\varphi_{W'\times W}$, $\varphi_{W\times W'}$ and $\varphi_{W'\times W'}$ satisfy condition (*fs-3*) for corresponding models.

From (*fs-2*), it follows:

$$\varphi^{-1} \circ R'' \leqslant R'' \circ \varphi^{-1}$$

$$\begin{bmatrix} \varphi^{-1}_{W\times W} & \varphi^{-1}_{W'\times W} \\ \varphi^{-1}_{W\times W'} & \varphi^{-1}_{W'\times W'} \end{bmatrix} \circ \begin{bmatrix} R_{W\times W} & \mathbf{0}_{W\times W'} \\ \mathbf{0}_{W'\times W} & R'_{W'\times W'} \end{bmatrix} \leqslant \begin{bmatrix} R_{W\times W} & \mathbf{0}_{W\times W'} \\ \mathbf{0}_{W'\times W} & R'_{W'\times W'} \end{bmatrix} \circ \begin{bmatrix} \varphi^{-1}_{W\times W} & \varphi^{-1}_{W'\times W} \\ \varphi^{-1}_{W\times W'} & \varphi^{-1}_{W'\times W'} \end{bmatrix}$$

$$\begin{bmatrix} \varphi^{-1}_{W\times W} \circ R_{W\times W} & \varphi^{-1}_{W'\times W} \circ R'_{W'\times W'} \\ \varphi^{-1}_{W\times W'} \circ R_{W\times W} & \varphi^{-1}_{W'\times W'} \circ R'_{W'\times W'} \end{bmatrix} \leqslant \begin{bmatrix} R_{W\times W} \circ \varphi^{-1}_{W\times W} & R_{W\times W} \circ \varphi^{-1}_{W'\times W} \\ R'_{W'\times W'} \circ \varphi^{-1}_{W\times W'} & R'_{W'\times W'} \circ \varphi^{-1}_{W'\times W'} \end{bmatrix}$$

Now, we have:

$$\varphi^{-1}_{W\times W} \circ R_{W\times W} \leqslant R_{W\times W} \circ \varphi^{-1}_{W\times W}$$
$$\varphi^{-1}_{W'\times W} \circ R'_{W'\times W'} \leqslant R_{W\times W} \circ \varphi^{-1}_{W'\times W}$$
$$\varphi^{-1}_{W\times W'} \circ R_{W\times W} \leqslant R'_{W'\times W'} \circ \varphi^{-1}_{W\times W'}$$
$$\varphi^{-1}_{W'\times W'} \circ R'_{W'\times W'} \leqslant R'_{W'\times W'} \circ \varphi^{-1}_{W'\times W'}$$

and it follows that $\varphi_{W\times W}$, $\varphi_{W'\times W}$, $\varphi_{W\times W'}$ and $\varphi_{W'\times W'}$ satisfy condition (*fs-2*) for corresponding models.

From (*fs-1*) it follows $V''_p \leqslant V''_p \circ \varphi^{-1}$ for every $p \in PV$. By the same procedure as in case (*fs-3*), we obtain:

$$V_W \leqslant V_W \circ \varphi^{-1}_{W\times W} \vee V_W \circ \varphi^{-1}_{W'\times W}$$
$$V'_{W'} \leqslant V'_{W'} \circ \varphi^{-1}_{W\times W'} \vee V'_{W'} \circ \varphi^{-1}_{W'\times W'}.$$

Therefore, we cannot show that (*fs-1*) is valid. However,

$$V_W \leqslant V_W \circ \varphi^{-1}_{W\times W}$$
$$V'_{W'} \leqslant V'_{W'} \circ \varphi^{-1}_{W'\times W'},$$

follows from the reflexivity of φ. The greatest simulation or bisimulation on model $\mathfrak{M}''$ is always reflexive. Hence, $\varphi_{W\times W}$ and $\varphi_{W'\times W'}$ are forward simulations, while $\varphi_{W'\times W}$ and $\varphi_{W\times W'}$ are forward presimulations.

These relations are the greatest ones which can be easily proved by assuming the opposite.

The other direction of the proof is straightforward. □

3.4 Computation of crisp simulations and bisimulations

According to the Theorem 3.6, if there is the greatest presimulation/prebisimulation of type θ, it is equal to the infimum of the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ defined by formula (3.27). Computing that infimum requires computing all members of the sequence, which can only be effectively done when this sequence is finite, in a way described in Algorithm 3.1. However, what to do if this sequence is not finite, i.e., if Algorithm 3.1 fails to terminate in a finite number of steps? In such situations we could "approximate" fuzzy simulations and bisimulations with crisp simulations and bisimulations. We will show how Algorithm 3.1 can be modified to test the existence and compute the greatest crisp simulations and bisimulations. The modified algorithm always terminates in a finite number of steps, independently of the properties of the underlying structure of truth values. Also, in Section 3.5 many interesting examples are given concerning the crisp simulations and bisimulations from which the following conclusions are drawn. First, the greatest crisp simulations and bisimulations cannot be obtained simply by taking the crisp parts of the greatest fuzzy simulations and bisimulations. Second, there are cases in which there is a fuzzy simulation/bisimulation of a given type between two fuzzy Kripke models, but there is not any crisp simulation/bisimulation of this type between them.

Let W and W' be non-empty finite sets of worlds, and let $\mathscr{R}^c(W,W')$ denote the set of all crisp relations from $\mathscr{R}(W,W')$. For each fuzzy relation $\varphi \in \mathscr{R}(W,W')$ we have that $\varphi^c \in \mathscr{R}^c(W,W')$, where φ^c denotes *the crisp part* of a fuzzy relation φ, i.e., a function $\varphi^c : W \times W' \to \{0,1\}$ defined by $\varphi^c(w,w') = 1$ if $\varphi(w,w') = 1$, and $\varphi^c(w,w') = 0$, if $\varphi(w,w') < 1$, for arbitrary $w \in W$ and $w' \in W'$. Equivalently, φ^c is considered as an ordinary crisp relation between W and W' given by $\varphi^c = \{(w,w') \in W \times W' \mid \varphi(w,w') = 1\}$.

Hence, for each function $\phi : \mathscr{R}(W,W') \to \mathscr{R}(W,W')$ we define a function $\phi^c : \mathscr{R}^c(W,W') \to \mathscr{R}^c(W,W')$ by

$$\phi^c(\varphi) = (\phi(\varphi))^c \qquad \text{for any} \qquad \varphi \in \mathscr{R}^c(W,W').$$

If ϕ is an isotone, then it can be easily shown that φ^c is also an isotone function.

Theorem 3.8. *Let W and W' be non-empty finite sets, let $\phi : \mathscr{R}(W,W') \to \mathscr{R}(W,W')$ be an isotone function and let $\pi \in \mathscr{R}(W,W')$ be a given fuzzy relation. A crisp relation $\varrho \in \mathscr{R}^c(W,W')$ is the greatest crisp solution in $\mathscr{R}(W,W')$ to the system*

$$\chi \leqslant \phi(\chi), \qquad \chi \leqslant \pi, \tag{3.30}$$

if and only if it is the greatest solution in $\mathscr{R}^c(W,W')$ to the system

$$\xi \leqslant \phi^c(\xi), \qquad \xi \leqslant \pi^c, \tag{3.31}$$

where χ is an unknown fuzzy relation and ξ is an unknown crisp relation.

Furthermore, a sequence $\{\varrho_k\}_{k\in\mathbb{N}} \subseteq \mathscr{R}(W,W')$ defined by

$$\varrho_1 = \pi^c, \qquad \varrho_{k+1} = \varrho \wedge \phi^c(\varrho_k) \qquad \text{for every } k \in \mathbb{N}, \tag{3.32}$$

is a finite descending sequence of crisp relations, and the least member of this sequence is the greatest solution to the system (3.31) in $\mathscr{R}^c(W,W')$.

Proof. The proof of this theorem can be obtained simply by translating the proof of Theorem 5.8 from [68] to the case of relations between the two sets. □

Taking ϕ to be any of the functions ϕ^θ, for $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, Theorem 3.8 gives algorithms for deciding whether there is a crisp simulation/bisimulation of a given type between two fuzzy Kripke models, and computing the greatest one, when it exists. As it can be seen in Theorem 3.8, these algorithms always terminate in a finite number of steps, independently of the properties of the underlying structure of truth values.

It is worth noting that functions $(\phi^\theta)^c$, for all $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, can be characterized as follows:

$$\begin{aligned}
&(w,w') \in (\phi^{fs})^c(\varrho) \quad \Leftrightarrow \quad (\forall i \in I)(\forall u \in W) R_i(w,u) \leqslant (R'_i \circ \varrho^{-1})(w',u)\\
&(w,w') \in (\phi^{bs})^c(\varrho) \quad \Leftrightarrow \quad (\forall i \in I)(\forall u \in W) R_i(u,w) \leqslant (\varrho \circ R'_i)(u,w')\\
&(\phi^{fb})^c(\varrho) = (\phi^{fs})^c(\varrho) \wedge [(\phi^{fs})^c(\varrho^{-1})]^{-1}\\
&(\phi^{bb})^c(\varrho) = (\phi^{bs})^c(\varrho) \wedge [(\phi^{bs})^c(\varrho^{-1})]^{-1}\\
&(\phi^{fbb})^c(\varrho) = (\phi^{fs})^c(\varrho) \wedge [(\phi^{bs})^c(\varrho^{-1})]^{-1}\\
&(\phi^{bfb})^c(\varrho) = (\phi^{bs})^c(\varrho) \wedge [(\phi^{fs})^c(\varrho^{-1})]^{-1}\\
&(\phi^{rb})^c(\varrho) = (\phi^{fs})^c(\varrho) \wedge [(\phi^{bs})^c(\varrho^{-1})]^{-1} \wedge (\phi^{bs})^c(\varrho) \wedge [(\phi^{fs})^c(\varrho^{-1})]^{-1}
\end{aligned}$$

for all $\varrho \in \mathscr{R}^c(W,W')$, $w \in W$ and $w' \in W'$.

3.5 Computational examples

This section gives examples that demonstrate the application of algorithms and clarify relationships between different types of simulations and bisimulations.

Several examples are on the standard Gödel modal logic over $[0, 1]$, while the last example is on the Boolean algebra of all subsets of some set A.

In the sequel, for any $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, by φ^θ we will denote the greatest simulation/bisimulation of type θ between two given fuzzy Kripke models, if it exists. On the other hand, by φ^θ_* we will denote the greatest fuzzy relation satisfying (θ-2) and (θ-3). It can be empty, but if it is non-empty, it is the greatest presimulation/prebisimulation of type θ. Therefore, in particular, by $\varphi^{fs}, \varphi^{bs}, \varphi^{fb}, \varphi^{bb}, \varphi^{fbb}, \varphi^{bfb}, \varphi^{rb}$ we will denote the greatest forward simulation, backward simulation, forward bisimulation, backward bisimulation, forward-backward bisimulation, backward-forward bisimulation and regular bisimulation, respectively, while by $\varphi^{fs}_*, \varphi^{bs}_*, \varphi^{fb}_*, \varphi^{bb}_*, \varphi^{fbb}_*, \varphi^{bfb}_*, \varphi^{rb}_*$ we will denote corresponding presimulation/prebisimulation. Analogously, ϱ^θ will denote the greatest crisp simulation/bisimulation of type θ, if it exists, and ϱ^θ_* the greatest crisp relation satisfying (θ-2) and (θ-3). If it is non-empty, it is the greatest crisp presimulation/prebisimulation of type θ. Therefore, in particular, by ϱ^{fs}, ϱ^{bs}, ϱ^{fb}, ϱ^{bb}, ϱ^{fbb}, ϱ^{bfb}, ϱ^{rb} we will denote the greatest crisp forward simulation, crisp backward simulation, crisp forward bisimulation, crisp backward bisimulation, crisp forward-backward bisimulation, crisp backward-forward bisimulation and crisp regular bisimulation, respectively, while by ϱ^{fs}_*, ϱ^{bs}_*, ϱ^{fb}_*, ϱ^{bb}_*, ϱ^{fbb}_*, ϱ^{bfb}_*, ϱ^{rb}_* we will denote corresponding crisp presimulation/prebisimulation.

Example 3.1. Let us recall fuzzy Kripke models from Example 2.3. Hence, fuzzy relations R_1, R_1' and fuzzy sets V_p, V_q, V_p' and V_q' are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 1 & 0 & 0.9 \\ 1 & 0.3 & 0.6 \\ 1 & 0 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}, \tag{3.33}$$

$$R_1' = \begin{bmatrix} 1 & 0.4 \\ 1 & 0.4 \end{bmatrix}, \qquad V_p' = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}, \qquad V_q' = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}. \tag{3.34}$$

Using algorithms for testing the existence of simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ and computing the greatest ones, we have:

$$\varphi^{fs}_* = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi^{bs}_* = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix},$$

$$\varphi^{fb}_* = \varphi^{fb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi^{bb}_* = \begin{bmatrix} 0.3 & 0.3 \\ 0.3 & 0.3 \\ 0.3 & 0.3 \end{bmatrix},$$

$$\varphi^{fbb}_* = \begin{bmatrix} 0.4 & 0.4 \\ 0.3 & 0.3 \\ 0.4 & 0.4 \end{bmatrix}, \qquad \varphi^{bfb}_* = \varphi^{bfb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi^{rb}_* = \begin{bmatrix} 0.3 & 0.3 \\ 0.3 & 0.3 \\ 0.3 & 0.3 \end{bmatrix},$$

while φ^{bb}, φ^{fbb} and φ^{rb} do not exist, since φ^{bb}_*, φ^{fbb}_* and φ^{rb}_* do not satisfy (*bb-1*), (*fbb-1*) and (*rb-1*), respectively.

Algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho^{fs}_* = \varrho^{fs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \varrho^{bs}_* = \varrho^{bs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix},$$

while ϱ^{fb}_*, ϱ^{bb}_*, ϱ^{fbb}_*, ϱ^{bfb}_* and ϱ^{rb}_* are empty, so ϱ^{fb}, ϱ^{bb}, ϱ^{fbb}, ϱ^{bfb} and ϱ^{rb} do not exist. Therefore, there are not the greatest crisp fb- and bfb-bisimulations, regardless of the fact that there are the greatest fuzzy bisimulations of these types.

If we consider the reverse fuzzy Kripke models for $\mathfrak{M}$ and $\mathfrak{M}'$, we have the opposite situation. Namely, in this case there are no fb- and bfb-bisimulations, while there are the greatest fs- and bs-simulations, as well as the greatest bb- and fbb-bisimulations. Since regular bisimulations are self-dual, there is not any regular bisimulation even between the reverse fuzzy Kripke models.

The Figure 3.2 graphically represents forward bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$.

According to the Definition 2.15 of disjoint union of Kripke models and Theorem 3.7 we can also compute any of (pre)simulation/(pre)bisimulation. For example, for forward bisimulation on model $\mathfrak{M} \sqcup \mathfrak{M}'$, we have:

$$\varphi^{fb} = \left[\begin{array}{ccc|cc} 1 & 0.4 & 1 & 1 & 0.4 \\ 0.4 & 1 & 0.4 & 0.4 & 1 \\ 1 & 0.4 & 1 & 1 & 0.4 \\ \hline 1 & 0.4 & 1 & 1 & 0.4 \\ 0.4 & 1 & 0.4 & 0.4 & 1 \end{array}\right].$$

Therefore, relations

$$\varphi^{fb}_{W\times W} = \begin{bmatrix} 1 & 0.4 & 1 \\ 0.4 & 1 & 0.4 \\ 1 & 0.4 & 1 \end{bmatrix}, \qquad \varphi^{fb}_{W\times W'} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix},$$

$$\varphi^{fb}_{W'\times W} = \begin{bmatrix} 1 & 0.4 & 1 \\ 0.4 & 1 & 0.4 \end{bmatrix}, \qquad \varphi^{fb}_{W'\times W'} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix},$$

are forward bisimulations for corresponding models.

The following example illustrates the situation where all five types of bisimulations are identical, which also holds for all crisp bisimulations.

Example 3.2. Let us replace R_1, V_p and V_q in (3.33) with

$$R_1 = \begin{bmatrix} 0.8 & 1 & 1 \\ 0.6 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.6 \end{bmatrix}, \tag{3.35}$$

and R'_1, V'_p and V'_q in (3.34) with

$$R'_1 = \begin{bmatrix} 0.8 & 1 \\ 0.6 & 0.5 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}. \tag{3.36}$$

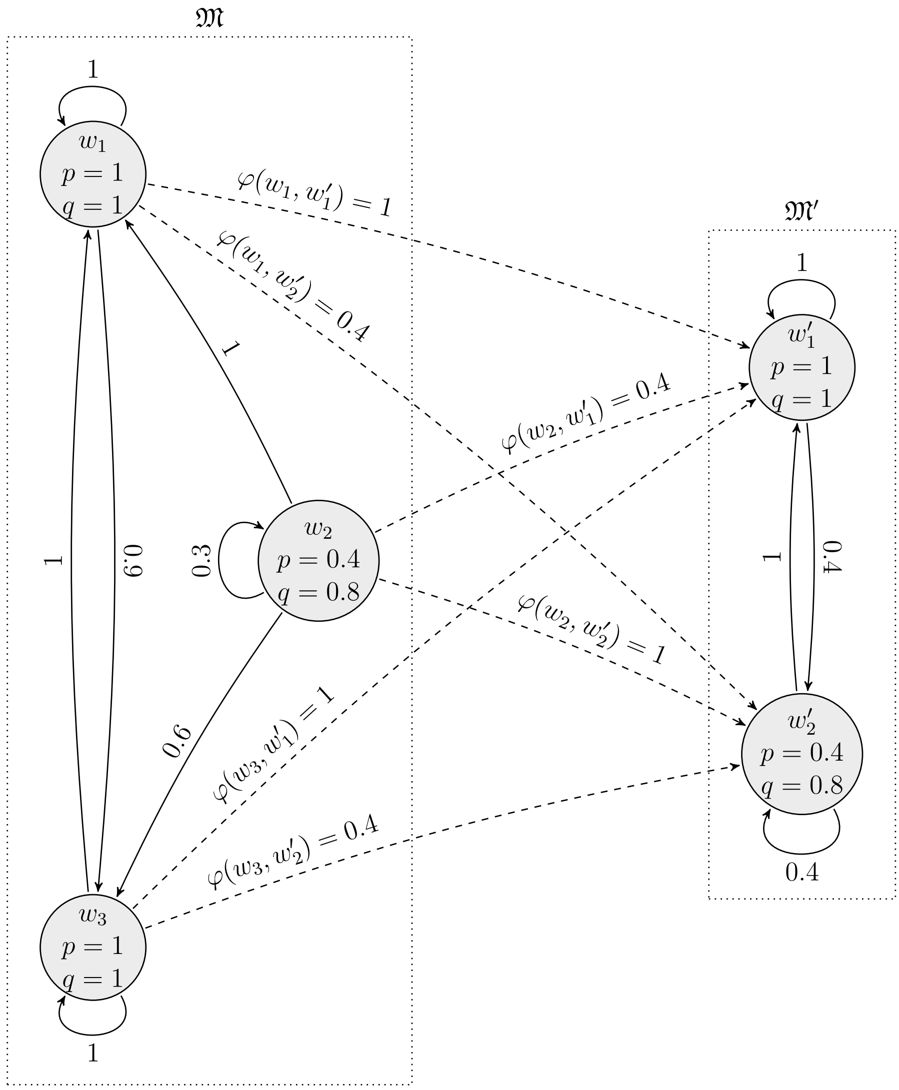

Figure 3.2: Forward bisimulation (dashed arrows) between models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 3.1

Using algorithms for testing the existence of simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ and computing the greatest ones, we have:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.5 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.5 \\ 0.8 & 1 \\ 0.8 & 1 \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \varphi_*^{bb} = \varphi^{bb} = \varphi_*^{fbb} = \varphi^{fbb} = \varphi_*^{bfb} = \varphi^{bfb} = \varphi_*^{rb} = \varphi^{rb} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \\ 0.5 & 1 \end{bmatrix}.$$

Algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho_*^{fs} = \varrho^{fs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad \varrho_*^{bs} = \varrho^{bs} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix},$$

$$\varrho_*^{fb} = \varrho^{fb} = \varrho_*^{bb} = \varrho^{bb} = \varrho_*^{fbb} = \varrho^{fbb} = \varrho_*^{bfb} = \varrho^{bfb} = \varrho_*^{rb} = \varrho^{rb} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

The Figure 3.3 graphically represents crisp forward bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$.

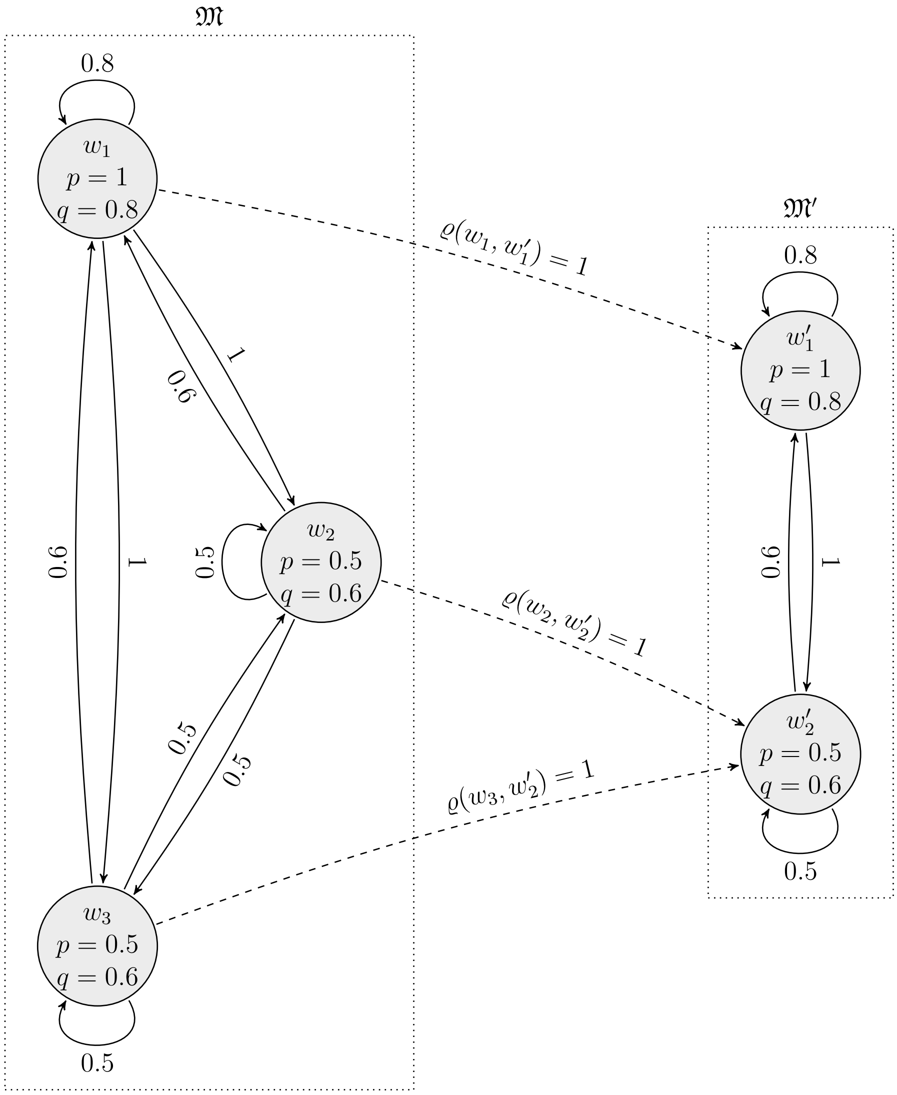

Figure 3.3: Crisp forward bisimulation ϱ^{fb} (dashed arrows) between models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 3.2.

The following example concerns simulations and bisimulations between fuzzy Kripke models with two fuzzy relations, i.e., it concerns a modal language with two quadruples of modal operators.

Example 3.3. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure, where $W = \{u, v, w\}$, $W' = \{u', v'\}$ and set $I = \{1, 2\}$. Fuzzy relations R_1, R_2, R_1', R_2' and fuzzy sets V_p, V_q, V_p' and V_q' are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0.8 & 0.7 & 0.7 \\ 1 & 0.7 & 0.7 \\ 1 & 0.6 & 0.6 \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 0.9 & 0.8 & 0.8 \\ 0.6 & 1 & 1 \\ 0.6 & 1 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.9 \\ 0.9 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.9 \\ 0.4 \\ 0.4 \end{bmatrix},$$

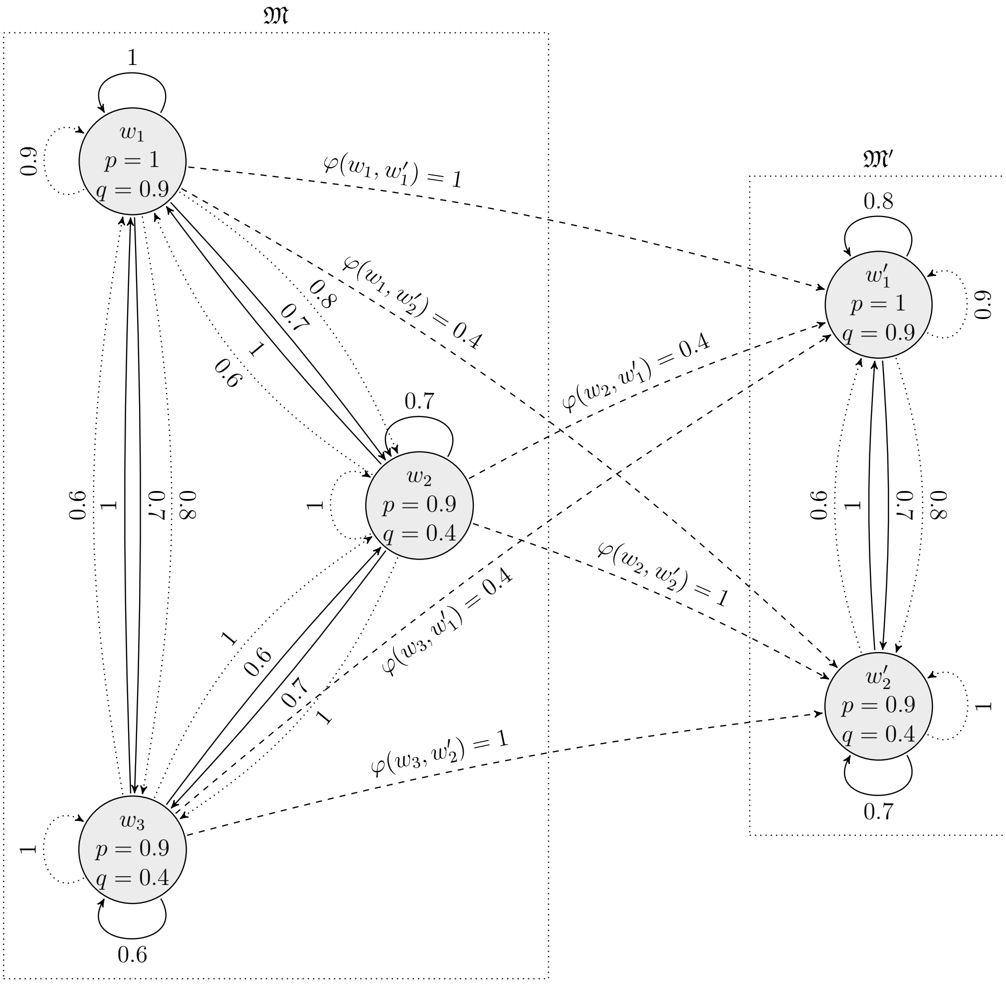

Figure 3.4: Backward bisimulation ϱ^{bb} (dashed arrows) between models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 3.3

$$R'_1 = \begin{bmatrix} 0.8 & 0.7 \\ 1 & 0.7 \end{bmatrix}, \qquad R'_2 = \begin{bmatrix} 0.9 & 0.8 \\ 0.6 & 1 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 0.9 \\ 0.4 \end{bmatrix}.$$

Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 \\ 0.8 & 1 \\ 0.8 & 1 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 \\ 0.9 & 1 \\ 0.9 & 1 \end{bmatrix},$$

$$\varphi_*^{fb} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \\ 0.4 & 0.6 \end{bmatrix}, \qquad \varphi_*^{bb} = \varphi^{bb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 0.4 & 1 \end{bmatrix},$$

$$\varphi_*^{fbb} = \varphi^{fbb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 0.4 & 1 \end{bmatrix}, \qquad \varphi_*^{bfb} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \\ 0.4 & 0.6 \end{bmatrix}, \qquad \varphi_*^{rb} = \begin{bmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \\ 0.4 & 0.6 \end{bmatrix},$$

and φ_*^{fb}, φ_*^{bfb} and φ_*^{rb} do not satisfy (*fb-1*), (*bfb-1*) and (*rb-1*), respectively, which means that φ^{fb}, φ^{bfb} and φ^{rb} do not exist.

The Figure 3.4 graphically represents backward bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$.

On the other hand, algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho_*^{fs} = \varrho^{fs} = \varrho_*^{bs} = \varrho^{bs} = \varrho_*^{bb} = \varrho^{bb} = \varrho_*^{fbb} = \varrho^{fbb} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

In this case, ϱ_*^{fb}, ϱ_*^{bfb} and ϱ_*^{rb} are empty, so there are no ϱ^{fb}, ϱ^{bfb} and ϱ^{rb}.

The following example shows what the simulations and bisimulations look like between a fuzzy Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and itself. We give this example to clearly see all variations and differences between various types of simulations and bisimulations.

Example 3.4. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model over the Gödel structure, where $W = \{u, v, w\}$ and set $I = \{1\}$. A fuzzy relation R_1 and fuzzy sets V_p, V_q, are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0.8 & 1 & 0.7 \\ 0.1 & 0.7 & 0.8 \\ 1 & 0.4 & 0.6 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.2 \\ 0.6 \\ 0.5 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.7 \\ 0.4 \\ 0.3 \end{bmatrix}. \tag{3.37}$$

If we set $\mathfrak{M}' = \mathfrak{M}$, then we have:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 & 0.3 \\ 0.2 & 1 & 0.3 \\ 0.2 & 0.4 & 1 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 & 0.3 \\ 0.2 & 1 & 0.3 \\ 0.2 & 0.7 & 1 \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.1 & 0.2 & 1 \end{bmatrix}, \qquad \varphi_*^{bb} = \varphi^{bb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.3 \\ 0.2 & 0.3 & 1 \end{bmatrix},$$

$$\varphi_*^{fbb} = \varphi^{fbb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.3 \\ 0.2 & 0.2 & 1 \end{bmatrix}, \qquad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.3 & 1 \end{bmatrix},$$

$$\varphi_*^{rb} = \varphi^{rb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.2 \\ 0.2 & 0.2 & 1 \end{bmatrix}.$$

On the other hand, all crisp simulations and bisimulations are equal to the equality relation (identity matrix).

The last example of this section shows what the simulations and bisimulations look like between two fuzzy Kripke models where the underlying structure is a Boolean algebra. Interestingly, the Boolean algebra in this example is not linearly ordered.

Example 3.5. Let us recall the power set algebra of $X = \{x, y, z\}$ from Example 1.7, i.e., $(\mathcal{P}(X), \cap, \cup, \to, \emptyset, X)$. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models over the power set algebra of X, where $W = \{u, v, w\}$, $W' = \{u', v'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R_1' and fuzzy sets V_p, V_q, V_p' and V_q' are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} X & \emptyset & X \\ \{x, y\} & \{y, z\} & \{x, z\} \\ X & \emptyset & X \end{bmatrix}, \qquad V_p = \begin{bmatrix} X \\ \{y\} \\ X \end{bmatrix}, \qquad V_q = \begin{bmatrix} X \\ \{y, z\} \\ X \end{bmatrix},$$

$$R_1' = \begin{bmatrix} X & \{y\} \\ X & \{y,z\} \end{bmatrix}, \qquad V_p' = \begin{bmatrix} X \\ \{y\} \end{bmatrix}, \qquad V_q' = \begin{bmatrix} X \\ \{y,z\} \end{bmatrix}.$$

Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} X & \{y\} \\ X & X \\ X & \{y\} \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} X & \{y\} \\ X & X \\ X & \{y\} \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} X & \{y\} \\ \{y\} & X \\ X & \{y\} \end{bmatrix}, \qquad \varphi_*^{bb} = \begin{bmatrix} \{x,y\} & \{y\} \\ \{y\} & X \\ \{x,y\} & \{y\} \end{bmatrix},$$

$$\varphi_*^{fbb} = \begin{bmatrix} \{x,y\} & \{y\} \\ \{y\} & \{x,y\} \\ \{x,y\} & \{y\} \end{bmatrix}, \quad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} X & \{y\} \\ \{y\} & X \\ X & \{y\} \end{bmatrix}, \quad \varphi_*^{rb} = \begin{bmatrix} \{x,y\} & \{y\} \\ \{y\} & \{x,y\} \\ \{x,y\} & \{y\} \end{bmatrix},$$

and φ_*^{bb}, φ_*^{fbb} and φ_*^{rb} do not satisfy (*bb-1*), (*fbb-1*) and (*rb-1*), respectively, which means that φ^{bb}, φ^{fbb} and φ^{rb} do not exist.

On the other hand, algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho_*^{fs} = \varrho^{fs} = \begin{bmatrix} X & \emptyset \\ X & X \\ X & \emptyset \end{bmatrix}, \qquad \varrho_*^{bs} = \varrho^{bs} = \begin{bmatrix} X & \emptyset \\ X & X \\ X & \emptyset \end{bmatrix},$$

while ϱ_*^{fb}, ϱ_*^{bb}, ϱ_*^{fbb}, ϱ_*^{bfb} and ϱ_*^{rb} are empty, so there are no ϱ^{fb}, ϱ^{bb}, ϱ^{fbb}, ϱ^{bfb} and ϱ^{rb}, similar like in Example 3.1.

3.6 State reduction of fuzzy Kripke models

In this section we present several ways to reduce the number of worlds of a fuzzy Kripke model while preserving its semantic properties. In other words, we provide a construction of a reduced fuzzy Kripke model which is $\Phi_{I,\mathscr{H}}^{+}$-equivalent, $\Phi_{I,\mathscr{H}}^{-}$-equivalent or $\Phi_{I,\mathscr{H}}$-equivalent to the original fuzzy Kripke model.

Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model. It is easy to see that for any $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$ the equality relation on W satisfies (θ-*1*), (θ-*2*) and (θ-*3*), i.e., it is a θ-simulation/bisimulation on $\mathfrak{M}$ (between $\mathfrak{M}$ and itself). It follows that the union of all θ-simulations/bisimulations on $\mathfrak{M}$ is non-empty, and it is also a θ-simulation/bisimulation, i.e., it is *the greatest θ-simulation/bisimulation on $\mathfrak{M}$*. We can also easily verify that the greatest θ-simulation (for $\theta \in \{fs, bs\}$) and the greatest θ-bisimulation (for $\theta \in \{fbb, bfb\}$) are fuzzy quasi-orders, while the greatest θ-bisimulation (for $\theta \in \{fb, bb, rb\}$) is a fuzzy equivalence. This emphasizes the importance of studying θ-simulations that are fuzzy quasi-orders, which will be called *θ-simulation fuzzy quasi-orders* (for $\theta \in \{fs, bs\}$), as well as of studying θ-bisimulations that are fuzzy equivalences, which will be called *θ-bisimulation fuzzy equivalences* (for $\theta \in \{fb, bb, rb\}$).

In the following text, special attention will be paid to forward and backward simulation fuzzy quasi-orders and forward and backward bisimulation fuzzy equivalences on a Kripke model.

The following two theorems establish connections between a model $\mathfrak{M}$ and its afterset model $\mathfrak{M}/Q$, that can be regarded as counterparts of the well-known First Isomorphism Theorem from algebra.

Theorem 3.9. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, let Q be a fuzzy quasi-order on W, and let $\mathfrak{M}/Q = (W/Q, \{R_i^{W/Q}\}_{i\in I}, V^{W/Q})$ be the afterset fuzzy Kripke model with respect to Q. Then the following is valid:*

(A) *A fuzzy relation $\varphi \in \mathscr{R}(W, W/Q)$ defined by*

$$\varphi(u, Q_v) = Q(u, v), \qquad \text{for all } u, v \in W, \tag{3.38}$$

is a backward simulation between $\mathfrak{M}$ and $\mathfrak{M}/Q$.

(B) *If Q is a forward simulation on $\mathfrak{M}$, then φ is a forward simulation between $\mathfrak{M}$ and $\mathfrak{M}/Q$.*

Proof. (A) We first notice that φ is a well-defined function, in the sense that for all $u, v_1, v_2 \in W$ such that $Q_{v_1} = Q_{v_2}$ we have that $\varphi(u, Q_{v_1}) = \varphi(u, Q_{v_2})$. Indeed, according to Theorem 1.13 we have that $Q^{v_1} = Q^{v_2}$ and

$$\varphi(u, Q_{v_1}) = Q(u, v_1) = Q^{v_1}(u) = Q^{v_2}(u) = Q(u, v_2) = \varphi(u, Q_{v_2}).$$

Further, for arbitrary $u, v \in W$, $p \in PV$ and $i \in I$

$$\begin{aligned} V_p(u) &\leqslant (Q \circ V_p \circ Q)(u) = \bigvee_{w\in W} Q(u, w) \wedge V_p \circ Q(w) \\ &= \bigvee_{w\in W} \varphi(u, Q_w) \wedge V_p^{W/Q}(Q_w) = (\varphi \circ V_p^{W/Q})(u), \end{aligned} \tag{3.39}$$

$$\begin{aligned} (R_i \circ \varphi)(u, Q_v) &= \bigvee_{w\in W} R_i(u, w) \wedge \varphi(w, Q_v) \\ &= \bigvee_{w\in W} R_i(u, w) \wedge Q(w, v) = (R_i \circ Q)(u, v) \\ &\leqslant (Q \circ Q \circ R_i \circ Q)(u, v) = \bigvee_{w\in W} Q(u, w) \wedge (Q \circ R_i \circ Q)(w, v) \\ &= \bigvee_{Q_w \in W/Q} \varphi(u, Q_w) \wedge R_i^{W/Q}(Q_w, Q_v) = (\varphi \circ R_i^{W/Q})(u, Q_v), \end{aligned} \tag{3.40}$$

$$\begin{aligned} (V_p \circ \varphi)(Q_v) &= \bigvee_{w\in W} V_p(w) \wedge \varphi(w, Q_v) = \bigvee_{w\in W} V_p(w) \wedge Q(w, v) \\ &= (V_p \circ Q)(v) = V_p^{W/Q}(Q_v). \end{aligned} \tag{3.41}$$

Note that the inequalities in (3.39) and (3.40) follow from the fact that $\alpha \leqslant \alpha \circ S$ and $\alpha \leqslant S \circ \alpha$, for each fuzzy relation or fuzzy set α, and each reflexive fuzzy relation S on a given set. Therefore, φ is a backward simulation between $\mathfrak{M}$ and $\mathfrak{M}/Q$.

(B) For arbitrary $u, v \in W$, $p \in PV$ and $i \in I$ we have

$$\begin{aligned} (V_p^{W/Q} \circ \varphi^{-1})(u) &= \bigvee_{Q_w \in W/Q} V_p^{W/Q}(Q_w) \wedge \varphi^{-1}(Q_w, u) \\ &= \bigvee_{w\in W} (V_p \circ Q)(w) \wedge Q^{-1}(w, u) \end{aligned}$$

$$
\begin{aligned}
&= (V_p \circ Q \circ Q^{-1})(u) \\
&\geqslant V_p(u) \quad \text{(due to the transitivity of } Q \circ Q^{-1}\text{)}, \qquad (3.42)
\end{aligned}
$$

$$
\begin{aligned}
(\varphi^{-1} \circ R_i)(Q_v, u) &= \bigvee_{w \in W} \varphi^{-1}(Q_v, w) \wedge R_i(w, u) \\
&= \bigvee_{w \in W} Q^{-1}(v, w) \wedge R_i(w, u) = (Q^{-1} \circ R_i)(v, u), \qquad (3.43)
\end{aligned}
$$

$$
\begin{aligned}
(R_i^{W/Q} \circ \varphi^{-1})(Q_v, u) &= \bigvee_{Q_w \in W/Q} R_i^{W/Q}(Q_v, Q_w) \wedge \varphi^{-1}(Q_w, u) \\
&= \bigvee_{w \in W} (Q \circ R_i \circ Q)(v, w) \wedge Q^{-1}(w, u) \\
&= (Q \circ R_i \circ Q \circ Q^{-1})(v, u), \qquad (3.44)
\end{aligned}
$$

$$
\begin{aligned}
(\varphi^{-1} \circ V_p)(Q_v) &= \bigvee_{w \in W} \varphi^{-1}(Q_v, w) \wedge V_p(w) = \bigvee_{w \in W} Q^{-1}(v, w) \wedge V_p(w) \\
&= (Q^{-1} \circ V_p)(v) = (V_p \circ Q)(v) = V_p^{W/Q}(Q_v). \qquad (3.45)
\end{aligned}
$$

From (3.42) and (3.45) it immediately follows that φ satisfies (*fs-1*) and (*fs-3*). With the additional assumption that Q is a forward simulation, and due to reflexivity of Q, (3.43) and (3.44) yield

$$
\begin{aligned}
(\varphi^{-1} \circ R_i)(u, Q_v) = (Q^{-1} \circ R_i)(u, v) &\leqslant (R_i \circ Q^{-1})(u, v) \\
&\leqslant (Q \circ R_i \circ Q \circ Q^{-1})(u, v) = (R_i^{W/Q} \circ \varphi^{-1})(u, Q_v).
\end{aligned}
$$

Therefore, φ satisfies (*fs-2*), so it is a forward simulation. □

Remark 3.1. *If we define $V_p^{W/Q} \in \mathscr{F}(W/Q)$ and $\varphi \in \mathscr{R}(W, W/Q)$ by*

$$
V_p^{W/Q}(Q_v) = (Q \circ V_p)(v), \qquad \varphi(u, Q_v) = Q^{-1}(u, v) = Q(v, u), \qquad (3.46)
$$

for all $u, v \in W$, $p \in PV$, then without any additional assumption we have that φ is a forward simulation between $\mathfrak{M}$ and $\mathfrak{M}/Q$, and with the additional assumption that Q^{-1} is a backward simulation on $\mathfrak{M}$ we get that φ is a backward simulation between $\mathfrak{M}$ and $\mathfrak{M}/Q$. This can be easily shown, in a similar way as in the proof of Theorem 3.9.

Theorem 3.10. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ be a fuzzy Kripke model, let E be a fuzzy equivalence on W, and let $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i \in I}, V^{W/E})$ be the afterset fuzzy Kripke model with respect to E.*

(A) *A fuzzy relation $\varphi \in \mathscr{R}(W, W/E)$ defined by*

$$
\varphi(u, E_v) = E(u, v), \qquad \text{for all } u, v \in W, \qquad (3.47)
$$

is both a forward and a backward simulation between $\mathfrak{M}$ and $\mathfrak{M}/E$.

(B) *The following conditions are equivalent:*
  - (i) *E is a forward (resp. backward) bisimulation fuzzy equivalence on $\mathfrak{M}$;*
  - (ii) *φ is a forward (resp. backward) bisimulation between $\mathfrak{M}$ and $\mathfrak{M}/E$;*
  - (iii) *φ is a backward-forward (resp. forward-backward) bisimulation between $\mathfrak{M}$ and $\mathfrak{M}/E$.*

Proof. (A) Since $E = E^{-1}$ and $E \circ V_p = V_p \circ E$, for each $p \in PV$, it follows directly from Theorem 3.9 and Remark 3.1 that φ is both a forward and a backward simulation.

(B) We will prove only the assertions that refer to forward bisimulations. Claims concerning backward bisimulations can be proved similarly.

(i)⇒(ii) and (i)⇒(iii). Suppose that E is a forward bisimulation. This means that $E \circ R_i \leqslant R_i \circ E$ and $E \circ V_p = V_p \circ E \leqslant V_p$, for all $i \in I$ and $p \in PV$. According to (A) we have that φ is a forward and backward simulation, so it remains to prove that φ^{-1} is a forward simulation.

For arbitrary $u, v \in W$, $p \in PV$ and $i \in I$ we have

$$
\begin{aligned}
V_p^{W/E}(E_v) &= (V_p \circ E)(v) = \bigvee_{w \in W} V_p(w) \wedge E(w,v) \\
&= \bigvee_{w \in W} V_p(w) \wedge \varphi(w, E_v) = (V_p \circ \varphi)(v), \qquad (3.48) \\
(\varphi \circ R_i^{W/E})(u, E_v) &= \bigvee_{E_w \in W/E} \varphi(u, E_w) \circ R_i^{W/E}(E_w, E_v) \\
&= \bigvee_{w \in W} E(u,w) \wedge (E \circ R_i \circ E)(w,v) = (E \circ E \circ R_i \circ E)(u,v) \\
&= (E \circ R_i \circ E)(u,v) \leqslant (R_i \circ E \circ E)(u,v) \\
&= (R_i \circ E)(u,v) = \bigvee_{w \in W} R_i(u,w) \wedge E(w,v) \\
&= \bigvee_{w \in W} R_i(u,w) \wedge \varphi(w, E_v) = (R_i \circ \varphi)(u, E_v), \qquad (3.49) \\
(\varphi \circ V_p^{W/E})(u) &= \bigvee_{E_w \in W/E} \varphi(u, E_w) \wedge V_p^{W/E}(E_w) = \bigvee_{w \in W} E(u,w) \wedge (V_p \circ E)(w) \\
&= (E \circ E \circ V_p \circ E)(u) = (E \circ V_p)(u) \leqslant V_p(u). \qquad (3.50)
\end{aligned}
$$

Thus, φ^{-1} is a forward simulation, whence we get that φ is a forward bisimulation, and also a backward-forward bisimulation. In the same way we prove the assertion that refers to backward bisimulations.

(ii)⇒(i) and (iii)⇒(i). Suppose that φ is a forward bisimulation or a backward-forward bisimulation, i.e., that φ^{-1} is a forward simulation. According to (3.50) we get $E \circ V_p = \varphi \circ V_p^{W/E} \leqslant V_p$, for each $p \in PV$, and according to (3.49) we get

$$
(E \circ R_i \circ E)(u,v) = (\varphi \circ R_i^{W/E})(u, E_v) \leqslant (R_i \circ \varphi)(u, E_v) = (R_i \circ E)(u,v),
$$

for all $u, v \in W$ and $i \in I$. From there we conclude that $E \circ R_i \circ E \leqslant R_i \circ E$, which yields

$$
E \circ R_i \leqslant E \circ R_i \circ E \leqslant R_i \circ E.
$$

Therefore, E is a forward bisimulation. □

The following theorems provide conditions under which the factor Kripke models $\mathfrak{M}$ and $\mathfrak{M}/E$ are $\Phi_{I,\mathscr{H}}$-equivalent, $\Phi_{I,\mathscr{H}}^{+}$-equivalent and $\Phi_{I,\mathscr{H}}^{-}$-equivalent, respectively. They are proven under the assumption that the underlying complete Heyting algebra $\mathscr{H}$ is linearly ordered.

Theorem 3.11. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be an image-finite fuzzy Kripke model over a linearly ordered Heyting algebra, let E be a forward bisimulation fuzzy equivalence on $\mathfrak{M}$, and $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i\in I}, V^{W/E})$ be the factor fuzzy Kripke model with respect to E. A fuzzy relation $\varphi \in \mathscr{R}(W, W/E)$ defined by*

$$\varphi(u, E_v) = E(u, v), \text{ for all } u, v \in W, \tag{3.51}$$

is a forward bisimulation and the following is true:

$$\varphi(u, Q_v) \leqslant \bigwedge_{A\in\Phi_{I,\mathscr{H}}^{+}} V_A(u) \leftrightarrow V_A^{W/E}(E_v), \quad \text{for all } u, v \in W. \tag{3.52}$$

Consequently, $\mathfrak{M}$ and $\mathfrak{M}/E$ are $\Phi_{I,\mathscr{H}}^{+}$-equivalent fuzzy Kripke models.

Proof. The fact that φ is a forward bisimulation follows from Theorem 3.10. By induction on complexity of a formula $A \in \Phi_{I,\mathscr{H}}^{+}$ we will prove that

$$\varphi(u, E_v) \leqslant V_A(u) \leftrightarrow V_A^{W/E}(E_v), \quad \text{for all } u, v \in W \text{ and every } A \in \Phi_{I,\mathscr{H}}^{+}. \tag{3.53}$$

Induction basis: If $A = p \in PV$, then from the fact that φ is forward bisimulation we have

$$\varphi^{-1} \circ V_p \leqslant V_p^{W/E}, \qquad \varphi \circ V_p^{W/E} \leqslant V_p,$$

and according to Lemma 3.3, it follows

$$\varphi^{-1} \leqslant V_p^{W/E}/V_p = (V_p \backslash V_p^{W/E})^{-1}, \qquad \varphi \leqslant V_p/V_p^{W/E},$$

whence

$$\varphi \leqslant V_p \backslash V_p^{W/E}, \qquad \varphi \leqslant V_p/V_p^{W/E},$$

i.e.,

$$\varphi \leqslant (V_p \backslash V_p^{W/E}) \wedge (V_p/V_p^{W/E}) = V_p \leftrightarrow V_p^{W/E}.$$

Therefore, (3.53) holds for any propositional variable p, and it trivially holds for any truth constant $\bar{t}$.

Induction step: (a) Let $A = B \wedge C$ and let (3.53) hold for B and C, i.e., $\varphi \leqslant V_B \leftrightarrow V_B^{W/E}$ and $\varphi \leqslant V_C \leftrightarrow V_C^{W/E}$. This yields

$$\varphi \leqslant (V_B \leftrightarrow V_B^{W/E}) \wedge (V_C \leftrightarrow V_C^{W/E}).$$

Using the property of Heyting algebras $(x_1 \leftrightarrow y_1)\wedge(x_2 \leftrightarrow y_2) \leqslant (x_1\wedge x_2) \leftrightarrow (y_1\wedge y_2)$, we get

$$\begin{aligned}
\varphi(u, E_v) &\leqslant (V(u, B) \leftrightarrow V_B^{W/E}(E_v)) \wedge (V(u, C) \leftrightarrow V_C^{W/E}(E_v)) \\
&\leqslant (V(u, B) \wedge V(u, C)) \leftrightarrow (V_B^{W/E}(E_v) \wedge V_C^{W/E}(E_v)) \\
&= V(u, B \wedge C) \leftrightarrow V_{B\wedge C}^{W/E}(E_v) \\
&= V_A(u) \leftrightarrow V_A^{W/E}(E_v),
\end{aligned}$$

for all $u \in W$ and $E_v \in W/E$, and we conclude that (3.53) holds for $A = B \wedge C$.

(b) Let A be of the form $B \to C$ and let (3.53) hold for B and C. In a similar way as (a), using the property of Heyting algebras $(x_1 \leftrightarrow y_1) \wedge (x_2 \leftrightarrow y_2) \leqslant (x_1 \to x_2) \leftrightarrow (y_1 \to y_2)$, we prove that (3.53) also holds for A.

(c) Let $A = \Diamond_i B$ and (3.53) let hold for B, i.e.,

$$\varphi \leqslant V_B \leftrightarrow V_B^{W/E} = (V_B \backslash V_B^{W/E}) \wedge (V_B / V_B^{W/E}).$$

Then it follows that

$$\varphi \leqslant V_B \backslash V_B^{W/E} \quad \text{and} \quad \varphi^{-1} \leqslant \left(V_B \backslash V_B^{W/E}\right)^{-1} = V_B^{W/E} / V_B,$$

and according to Lemma 3.3 we finally get $\varphi^{-1} \circ V_B \leqslant V_B^{W/E}$. Now we have

$$\begin{aligned} \varphi^{-1} \circ V_A = \varphi^{-1} \circ R_i \circ V_B &\leqslant R_i^{W/E} \circ \varphi^{-1} \circ V_B \qquad \text{according to } (\textit{fb-2}) \\ &\leqslant R_i^{W/E} \circ V_B^{W/E} = V_A^{W/E}, \end{aligned}$$

for every $i \in I$. Hence, from $\varphi^{-1} \circ V_A \leqslant V_A^{W/E}$ we can conclude that $\varphi^{-1} \leqslant V_A^{W/E} / V_A = (V_A \backslash V_A^{W/E})^{-1}$, whence $\varphi \leqslant V_A \backslash V_A^{W/E}$. In a similar way we can conclude that $\varphi \leqslant V_A / V_A^{W/E}$, which means that

$$\varphi \leqslant (V_A \backslash V_A^{W/E}) \wedge (V_A / V_A^{W/E}) = V_A \leftrightarrow V_A^{W/E}.$$

Therefore, we have proved that (3.53) holds for $A = \Diamond_i B$.

(d) Suppose that $A = \Box_i B$ and (3.53) holds for B. In a similar way as in (c), from $\varphi \leqslant V_B \leftrightarrow V_B^{W/E}$, we conclude

$$\varphi^{-1} \circ V_B \leqslant V_B^{W/E}, \qquad \varphi \circ V_B^{W/E} \leqslant V_B.$$

Since underlying structure is linearly ordered, values $\varphi(u, Q_v) = \varphi^{-1}(E_v, u)$, $V_A(u)$ and $V_A^{W/E}(E_v)$ can be compared with each other for every $u \in W$, $E_v \in W/E$, therefore, case analysis can be used.

If $\varphi^{-1}(Q_v, u) \leqslant V_A(u) \wedge V_A^{W/E}(E_v)$ and $V_A(u) \neq V_A^{W/E}(E_v)$, then

$$\varphi(u, E_v) = \varphi^{-1}(E_v, u) \leqslant V_A(u) \wedge V_A^{W/E}(E_v) = V_A(u) \leftrightarrow V_A^{W/E}(E_v).$$

In case $V_A(u) = V_A^{W/E}(E_v)$ we have that $V_A(u) \leftrightarrow V_A^{W/E}(E_v) = 1$, which gives $\varphi(u, E_v) \leqslant V_A(u) \leftrightarrow V_A^{W/E}(E_v)$.

Hence, we only need to consider case where $\varphi^{-1}(E_v, u) > V_A(u) \wedge V_A^{W/E}(E_v)$. Without loss of generality, we can assume that $\varphi^{-1}(E_v, u) > V_A(u)$, and then we have:

$$\begin{aligned} V_A(u) &= \varphi^{-1}(E_v, u) \wedge V_A(u) \\ &= \varphi^{-1}(E_v, u) \wedge \bigwedge_{w \in W} (R_i(u, w) \to V_B(w)) \\ &= \bigwedge_{w \in W} \left[\varphi^{-1}(E_v, u) \wedge \left(R_i(u, w) \to V_B(w)\right)\right] \quad \text{(by (1.73))} \\ &= \bigwedge_{w \in W} \left[\varphi^{-1}(E_v, u) \wedge \left(\varphi^{-1}(E_v, u) \wedge R_i(u, w) \to V_B(w)\right)\right] \quad \text{(by (1.69))} \end{aligned}$$

$$= \varphi^{-1}(E_v, u) \wedge \bigwedge_{w\in W} \left[\varphi^{-1}(E_v, u) \wedge R_i(u, w) \to V_B(w)\right] \quad \text{(by (1.73))} \tag{3.54}$$

Since the relation φ is a forward bisimulation, it satisfies (*fb-2*), i.e.

$$\varphi^{-1} \circ R_i \leqslant R_i^{W/E} \circ \varphi^{-1}, \qquad \text{for every } i \in I.$$

Next, since $R_i^{W/E}$ is image-finite, for any $w \in W$ we can find $E_z \in W/E$ such that

$$\varphi^{-1}(E_v, u) \wedge R_i(u, w) \leqslant R_i^{W/E}(E_v, E_z) \wedge \varphi^{-1}(E_z, w),$$

and it follows

$$\big(\varphi^{-1}(E_v, u) \wedge R_i(u, w)\big) \to V_B(w) \geqslant \big(\varphi^{-1}(E_z, w) \wedge R_i^{W/E}(E_v, E_z)\big) \to V_B(w).$$

Now, two cases need to be analyzed. First, if $V_B(w) = V_B^{W/E}(E_z)$, then

$$\begin{aligned}\big(\varphi^{-1}(E_z, w) \wedge R_i^{W/E}(E_v, E_z)\big) \to V_B(w) &\geqslant R_i^{W/E}(E_v, E_z) \to V_B(w)\\ &= R_i^{W/E}(E_v, E_z) \to V_B^{W/E}(E_z).\end{aligned}$$

On the other hand, if $V_B(w) \neq V_B^{W/E}(E_z)$, then by the induction hypothesis we have that

$$\varphi^{-1}(E_z, w) \leqslant (V_B(w) \leftrightarrow V_B^{W/E}(E_z)) \leqslant V_B(w).$$

Thus,

$$\big(\varphi^{-1}(E_z, w) \wedge R_i^{W/E}(E_v, E_z)\big) \to V_B(w) = 1 \geqslant R_i^{W/E}(E_v, E_z) \to V_B^{W/E}(E_z).$$

In both cases, we have shown that for any $w \in W$, we can find E_z such that

$$\big(\varphi^{-1}(E_v, u) \wedge R_i(u, w)\big) \to V_B(w) \geqslant R_i^{W/E}(E_v, E_z) \to V_B^{W/E}(E_z).$$

Therefore,

$$\bigwedge_{w\in W} \big(\varphi^{-1}(E_v, u) \wedge R_i(u, w)\big) \to V_B(w) \geqslant \bigwedge_{z\in W} R_i^{W/E}(E_v, E_z) \to V_B^{W/E}(E_z) = V_A^{W/E}(E_v)$$

and using (3.54) we conclude:

$$V_A(u) \geqslant \varphi^{-1}(E_v, u) \wedge V_A^{W/E}(E_v).$$

Because of the assumption that $\varphi^{-1}(E_v, u) > V_A(u)$, we have

$$V_A(u) \geqslant V_A^{W/E}(E_v) \;\text{ and }\; \varphi^{-1}(E_v, u) > V_A^{W/E}(E_v).$$

Since $\varphi^{-1}(E_v, u) > V_A^{W/E}(E_v)$, by the same reasoning we can prove that $V_A^{W/E}(E_v) \geqslant V_A(u)$. Hence, we have $V_A(u) = V_A^{W/E}(E_z)$, and since $\varphi(u, E_v) = \varphi^{-1}(E_v, u)$ it follows

$$\varphi(u, E_v) \leqslant V_A(u) \leftrightarrow V_A^{W/E}(E_v) = 1$$

when $\varphi^{-1}(E_v, u) > V_A(u) \wedge V_A^{W/E}(E_v)$. This completes the proof of the theorem. □

Similarly we prove the following two theorems.

Theorem 3.12. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a domain-finite fuzzy Kripke model over a linearly ordered Heyting algebra, let E be a backward bisimulation fuzzy equivalence on W, and let $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i\in I}, V^{W/E})$ be the factor fuzzy Kripke model with respect to E. A fuzzy relation $\varphi \in \mathscr{R}(W, W/E)$ defined by*

$$\varphi(u, E_v) = E(u, v), \text{ for all } u, v \in W, \tag{3.55}$$

is a backward bisimulation and the following is true:

$$\varphi(u, E_v) \leqslant \bigwedge_{A\in\Phi_{I,\mathscr{H}}^{-}} V_A(u) \leftrightarrow V_A^{W/E}(E_v). \tag{3.56}$$

Consequently, $\mathfrak{M}$ and $\mathfrak{M}/E$ are $\Phi_{I,\mathscr{H}}^{-}$-equivalent fuzzy Kripke models.

Proof. This follows from the previous theorem since a backward bisimulation between two models is a forward bisimulation between the reverse models. □

Theorem 3.13. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a degree-finite fuzzy Kripke model over a linearly ordered Heyting algebra, let E be a regular bisimulation fuzzy equivalence on W, and let $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i\in I}, V^{W/E})$ be the factor fuzzy Kripke model with respect to E. A fuzzy relation $\varphi \in \mathscr{R}(W, W/E)$ defined by*

$$\varphi(u, E_v) = E(u, v), \text{ for all } u, v \in W, \tag{3.57}$$

is a regular bisimulation and the following is true:

$$\varphi(u, E_v) \leqslant \bigwedge_{A\in\Phi_{I,\mathscr{H}}} V_A(u) \leftrightarrow V_A^{W/E}(E_v). \tag{3.58}$$

Consequently, $\mathfrak{M}$ and $\mathfrak{M}/E$ are $\Phi_{I,\mathscr{H}}$-equivalent fuzzy Kripke models.

Proof. This follows immediately from the previous two theorems. □

3.7 Computational examples for state reductions of fuzzy Kripke models

In this section we provide examples which demonstrate the application of theorems from the previous section in the state reduction of the fuzzy Kripke models. As in Section 3.5, several examples are based on the standard Gödel modal logic over the real unit interval $[0, 1]$, while the last example is on the Boolean algebra.

As we already said in the previous section, the greatest bisimulation of type $\theta \in \{fb, bb, rb\}$ on a fuzzy Kripke model $\mathfrak{M}$ is a fuzzy equivalence, which will be denoted by E^θ, while the greatest bisimulation of type $\theta \in \{fbb, bfb\}$ on $\mathfrak{M}$ is a fuzzy quasi-order, which will be denoted by Q^θ.

The following example illustrates a situation where E^{fb} reduces the states of the model, but none of the other bisimulations do so.

Example 3.6. Let $\mathfrak{M} = (W, \{R_1\}, V)$ be the fuzzy Kripke model from Example 3.1, i.e., let the fuzzy relation R_1 and fuzzy sets V_p, V_q, be represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} 1 & 0 & 0.9 \\ 1 & 0.3 & 0.6 \\ 1 & 0 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}. \tag{3.59}$$

Using algorithms for computing the greatest bisimulations on the fuzzy Kripke model $\mathfrak{M}$, we have:

$$E^{fb} = \begin{bmatrix} 1 & 0.4 & 1 \\ 0.4 & 1 & 0.4 \\ 1 & 0.4 & 1 \end{bmatrix}, \qquad E^{bb} = \begin{bmatrix} 1 & 0.3 & 0.6 \\ 0.3 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{bmatrix}, \qquad E^{rb} = \begin{bmatrix} 1 & 0.3 & 0.6 \\ 0.3 & 1 & 0.3 \\ 0.6 & 0.3 & 1 \end{bmatrix},$$

$$Q^{fbb} = \begin{bmatrix} 1 & 0.4 & 1 \\ 0.3 & 1 & 0.3 \\ 0.6 & 0.4 & 1 \end{bmatrix}, \qquad Q^{bfb} = \begin{bmatrix} 1 & 0.3 & 0.6 \\ 0.4 & 1 & 0.4 \\ 1 & 0.3 & 1 \end{bmatrix}.$$

Hence, E^{fb} is a forward bisimulation fuzzy quasi-order with two different aftersets, and we have:

$$E^{fb} \circ R_1 \circ E^{fb} = \begin{bmatrix} 1 & 0.4 & 1 \\ 1 & 0.4 & 1 \\ 1 & 0.4 & 1 \end{bmatrix}, \qquad V_p \circ E^{fb} = V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q \circ E^{fb} = V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}.$$

Now, from (2.16) and (2.17) we get the related afterset model $\mathfrak{M}/E^{fb} = (W/E^{fb}, \{R_1^{W/E^{fb}}\}, V^{W/E^{fb}})$ where

$$R_1^{W/E^{fb}} = \begin{bmatrix} 1 & 0.4 \\ 1 & 0.4 \end{bmatrix}, \qquad V_p^{W/E^{fb}} = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}, \qquad V_q^{W/E^{fb}} = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix},$$

which is isomorphic to the model $\mathfrak{M}'$ from Example 3.1. According to Theorem 3.11 we have that the models $\mathfrak{M}$ and $\mathfrak{M}/E^{fb}$ are $\Phi^{+}_{I,\mathscr{H}}$-equivalent.

On the other hand, E^{bb}, E^{rb}, Q^{fbb} and Q^{bfb} are fuzzy equivalences and fuzzy quasi-orders whose equivalence classes and aftersets are all different (such fuzzy equivalences and fuzzy quasi-orders are called fuzzy equalities and fuzzy orders, respectively). For that reason, they cannot reduce the states of the model.

What we can also conclude from there is that the greatest forward-backward bisimulation and the greatest backward-forward bisimulation are not necessarily fuzzy equivalences.

If we consider the reverse model $\mathfrak{M}^{-1} = (W, \{R_1\}^{-1}, V)$, then we have that the greatest backward bisimulation on $\mathfrak{M}^{-1}$ reduces the number of states of this model, and in this case the related afterset model is $\Phi^{-}_{I,\mathscr{H}}$-equivalent to $\mathfrak{M}^{-1}$, but other types of bisimulations on $\mathfrak{M}^{-1}$ cannot reduce any states.

Example 3.7. Let $\mathfrak{M} = (W, \{R_1\}, V)$ be the fuzzy Kripke model from Example 3.2, i.e., let the fuzzy relation R_1 and fuzzy sets V_p and V_q be given as follows:

$$R_1 = \begin{bmatrix} 0.8 & 1 & 1 \\ 0.6 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.6 \end{bmatrix}. \tag{3.60}$$

Using algorithms for computing the greatest bisimulations on the fuzzy Kripke model $\mathfrak{M}$, we have:

$$E^{fb} = E^{bb} = E^{rb} = Q^{fbb} = Q^{bfb} = \begin{bmatrix} 1 & 0.5 & 0.5 \\ 0.5 & 1 & 1 \\ 0.5 & 1 & 1 \end{bmatrix}.$$

Let us denote all these fuzzy equivalences by E. Then, we have:

$$E \circ R_1 \circ E = \begin{bmatrix} 0.8 & 1 & 1 \\ 0.6 & 0.5 & 0.5 \\ 0.6 & 0.5 & 0.5 \end{bmatrix}, \qquad V_p \circ E = V_p = \begin{bmatrix} 1 \\ 0.5 \\ 0.5 \end{bmatrix}, \qquad V_q \circ E = V_q = \begin{bmatrix} 0.8 \\ 0.6 \\ 0.6 \end{bmatrix},$$

and from (2.16) and (2.17) we get the related factor fuzzy Kripke model $\mathfrak{M}/E = (W/E, \{R_1^{W/E}\}, V^{W/E})$, where

$$R_1^{W/E} = \begin{bmatrix} 0.8 & 1 \\ 0.6 & 0.5 \end{bmatrix}, \qquad V_p^{W/E} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \qquad V_q^{W/E} = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix},$$

i.e., we get the model with a smaller number of states identical to the model $\mathfrak{M}'$ from Example 3.2.

Also, according to Theorem 3.13, the models $\mathfrak{M}$ and $\mathfrak{M}/E$ are $\Phi_{I,\mathscr{H}}$-equivalent. Clearly, these models are also $\Phi_{I,\mathscr{H}}^{+}$-equivalent and $\Phi_{I,\mathscr{H}}^{-}$-equivalent.

The following example illustrates a situation where no type of bisimulation can reduce the number of states of a model.

Example 3.8. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model over the Gödel structure $[0, 1]$, where $W = \{u, v, w\}$ and set $I = \{1\}$. Fuzzy relation R_1 and fuzzy sets V_p, V_q, are represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} 0.6 & 0.7 & 0.7 \\ 0.9 & 0.9 & 0.5 \\ 1 & 0.3 & 0.8 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.7 \\ 0.3 \\ 0.3 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.8 \end{bmatrix}. \tag{3.61}$$

Using the algorithms for computing the greatest bisimulations on the fuzzy Kripke model $\mathfrak{M}$, we have:

$$E^{fb} = E^{bb} = E^{rb} = Q^{fbb} = Q^{bfb} = \begin{bmatrix} 1 & 0.2 & 0.2 \\ 0.2 & 1 & 0.8 \\ 0.2 & 0.8 & 1 \end{bmatrix}.$$

Clearly, this fuzzy equivalence is a fuzzy equality, i.e., all its equivalence classes are different. This means that the number of states of the related factor fuzzy Kripke model is the same as in the original fuzzy Kripke model $\mathfrak{M}$. The factor fuzzy Kripke model $\mathfrak{M}/E^{fb} = (W/E^{fb}, \{R_1^{W/E^{fb}}\}, V^{W/E^{fb}})$ is represented by the following fuzzy matrix and column vectors:

$$R_1^{W/E^{fb}} = \begin{bmatrix} 0.6 & 0.7 & 0.7 \\ 0.9 & 0.9 & 0.8 \\ 1 & 0.8 & 0.8 \end{bmatrix}, \qquad V_p^{W/E^{fb}} = \begin{bmatrix} 0.7 \\ 0.3 \\ 0.3 \end{bmatrix}, \qquad V_q^{W/E^{fb}} = \begin{bmatrix} 0.2 \\ 0.8 \\ 0.8 \end{bmatrix}.$$

Further, if we recall model from Example 2.5, i.e., model $\mathfrak{M} = (W, \{R_1, R_2\}, V)$ where $W = \{w_1, w_2, w_3, w_4\}$, fuzzy relations R_1, R_2 and fuzzy sets V_p, V_q are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0.7 & 1 & 0.5 & 0.8 \\ 1 & 0.4 & 0.7 & 1 \\ 0.3 & 0.8 & 0.1 & 1 \\ 0.6 & 1 & 0.9 & 0.8 \end{bmatrix}, R_2 = \begin{bmatrix} 1 & 0.1 & 0.2 & 0.6 \\ 0.4 & 0.3 & 0.8 & 1 \\ 0.2 & 0.7 & 0.1 & 1 \\ 0.3 & 0.8 & 0.1 & 0.4 \end{bmatrix}, V_p = \begin{bmatrix} 0.7 \\ 0.8 \\ 1 \\ 1 \end{bmatrix}, V_q = \begin{bmatrix} 0.7 \\ 0.6 \\ 1 \\ 1 \end{bmatrix}.$$

In this case we have:

$$E^{fb} = E^{bb} = E^{rb} = Q^{fbb} = Q^{bfb} = \begin{bmatrix} 1 & 0.6 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 & 0.6 \\ 0.6 & 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 0.6 & 1 \end{bmatrix},$$

and the factor fuzzy Kripke model $\mathfrak{M}/E^{fb} = (W/E^{fb}, \{R_1^{W/E^{fb}}\}, V^{W/E^{fb}})$ is represented by the following fuzzy matrix and column vectors:

$$R_1^{W/E^{fb}} = \begin{bmatrix} 0.7 & 1 & 0.6 & 0.8 \\ 1 & 0.6 & 0.7 & 1 \\ 0.6 & 0.8 & 0.6 & 1 \\ 0.6 & 1 & 0.9 & 0.8 \end{bmatrix}, \qquad R_2^{W/E^{fb}} = \begin{bmatrix} 1 & 0.6 & 0.6 & 0.6 \\ 0.6 & 0.6 & 0.8 & 1 \\ 0.6 & 0.7 & 0.6 & 1 \\ 0.6 & 0.8 & 0.6 & 0.6 \end{bmatrix},$$

$$V_p^{W/E^{fb}} = \begin{bmatrix} 0.7 \\ 0.8 \\ 1 \\ 1 \end{bmatrix}, \qquad V_q^{W/E^{fb}} = \begin{bmatrix} 0.7 \\ 0.6 \\ 1 \\ 1 \end{bmatrix}.$$

Therefore, in these cases, we can not reduce the number of states of the model.

The following example illustrates a situation where all three types of bisimulation fuzzy equivalences can reduce the number of states of a fuzzy Kripke model but provide factor fuzzy Kripke models of different number of states.

Example 3.9. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model over the Gödel structure $[0,1]$, where $W = \{u, v, w, z\}$ and set $I = \{1\}$. Fuzzy relation R_1 and fuzzy sets V_p and V_q are represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0.2 & 0 & 0 \\ 0.8 & 0.5 & 1 & 1 \\ 0.8 & 0.5 & 1 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.3 \\ 1 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.6 \\ 1 \\ 1 \end{bmatrix}.$$

Using algorithms for computing the greatest bisimulations on the fuzzy Kripke model $\mathfrak{M}$, we have:

$$E^{fb} = \begin{bmatrix} 1 & 0.2 & 0.5 & 0.5 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.5 & 0.2 & 1 & 1 \\ 0.5 & 0.2 & 1 & 1 \end{bmatrix}, \qquad E^{bb} = \begin{bmatrix} 1 & 0.3 & 1 & 1 \\ 0.3 & 1 & 0.3 & 0.3 \\ 1 & 0.3 & 1 & 1 \\ 1 & 0.3 & 1 & 1 \end{bmatrix},$$

$$E^{rb} = \begin{bmatrix} 1 & 0.2 & 0.5 & 0.5 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.5 & 0.2 & 1 & 1 \\ 0.5 & 0.2 & 1 & 1 \end{bmatrix},$$

$$Q^{fbb} = \begin{bmatrix} 1 & 0.2 & 0.5 & 0.5 \\ 0.3 & 1 & 0.3 & 0.3 \\ 1 & 0.2 & 1 & 1 \\ 1 & 0.2 & 1 & 1 \end{bmatrix}, \qquad Q^{bfb} = \begin{bmatrix} 1 & 0.3 & 1 & 1 \\ 0.2 & 1 & 0.2 & 0.2 \\ 0.5 & 0.3 & 1 & 1 \\ 0.5 & 0.3 & 1 & 1 \end{bmatrix}.$$

Now, E^{fb} and E^{rb} provide factor fuzzy Kripke models $\mathfrak{M}/E^{fb} = (W/E^{fb}, \{R_1^{W/E^{fb}}\}, V^{W/E^{fb}})$ with 3 worlds, whereas E^{bb} provides the factor fuzzy Kripke model $\mathfrak{M}/E^{bb} = (W/E^{bb}, \{R_1^{W/E^{bb}}\}, V^{W/E^{bb}})$ with 2 worlds.

$$R_1^{W/E^{fb}} = \begin{bmatrix} 1 & 1 & 1 \\ 0.2 & 0.2 & 0.2 \\ 0.8 & 0.5 & 1 \end{bmatrix}, \qquad V_p^{W/E^{fb}} = \begin{bmatrix} 1 \\ 0.3 \\ 1 \end{bmatrix}, \qquad V_q^{W/E^{fb}} = \begin{bmatrix} 1 \\ 0.6 \\ 1 \end{bmatrix},$$

$$R_1^{W/E^{bb}} = \begin{bmatrix} 1 & 1 \\ 0.3 & 0.3 \end{bmatrix}, \qquad V_p^{W/E^{bb}} = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}, \qquad V_q^{W/E^{bb}} = \begin{bmatrix} 1 \\ 0.6 \end{bmatrix}.$$

Note that Q^{fbb} provide afterset fuzzy Kripke model with 2 worlds while Q^{bfb} provide afterset fuzzy Kripke model with 3 worlds.

However, the factor model with respect to E^{fb} cannot be further reduced by the greatest forward bisimulation, but it can be easily verified that it can be reduced by the backward bisimulation, which again provides a factor model with 2 worlds. These type of reduction is known as *alternating reductions*. For alternating reduction on fuzzy automata see [133].

The last example illustrates a situation where the fuzzy Kripke model is over partially ordered Boolean algebra. Hence, none of the Theorems 3.11, 3.12 and 3.13 do not hold. Still, in this example E^{fb} reduces the number of states of the model, but none of the other bisimulations do so.

Example 3.10. Let $\mathfrak{M} = (W, \{R_1\}, V)$ be the fuzzy Kripke model from Example 3.5, i.e., let the fuzzy relation R_1 and fuzzy sets V_p, V_q, be represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} X & \emptyset & X \\ \{x,y\} & \{y,z\} & \{x,z\} \\ X & \emptyset & X \end{bmatrix}, \qquad V_p = \begin{bmatrix} X \\ \{y\} \\ X \end{bmatrix}, \qquad V_q = \begin{bmatrix} X \\ \{y,z\} \\ X \end{bmatrix}. \tag{3.62}$$

Using algorithms for computing the greatest bisimulations on the fuzzy Kripke model $\mathfrak{M}$, we have:

$$E^{fb} = \begin{bmatrix} X & \{y\} & X \\ \{y\} & X & \{y\} \\ X & \{y\} & X \end{bmatrix}, \qquad E^{bb} = \begin{bmatrix} X & \{y\} & \{x,y\} \\ \{y\} & X & \{y\} \\ \{x,y\} & \{y\} & X \end{bmatrix},$$

$$E^{rb} = \begin{bmatrix} X & \{y\} & \{x,y\} \\ \{y\} & X & \{y\} \\ \{x,y\} & \{y\} & X \end{bmatrix},$$

$$Q^{fbb} = \begin{bmatrix} X & \{y\} & \{x,y\} \\ \{y\} & X & \{y\} \\ X & \{y\} & X \end{bmatrix}, \qquad Q^{bfb} = \begin{bmatrix} X & \{y\} & X \\ \{y\} & X & \{y\} \\ \{x,y\} & \{y\} & X \end{bmatrix}.$$

Hence, E^{fb} is a forward bisimulation fuzzy quasi-order with two different aftersets, and we have:

$$E^{fb} \circ R_1 \circ E^{fb} = \begin{bmatrix} X & \{y\} & X \\ X & \{y,z\} & X \\ X & \{y\} & X \end{bmatrix}, V_p \circ E^{fb} = V_p = \begin{bmatrix} X \\ \{y\} \\ X \end{bmatrix}, V_q \circ E^{fb} = V_q = \begin{bmatrix} X \\ \{y,z\} \\ X \end{bmatrix}.$$

Now, from (2.16) and (2.17) we get the related afterset model $\mathfrak{M}/E^{fb} = (W/E^{fb}, \{R_1^{W/E^{fb}}\}, V^{W/E^{fb}})$ where

$$R_1^{W/E^{fb}} = \begin{bmatrix} X & \{y\} \\ X & \{y,z\} \end{bmatrix}, \qquad V_p^{W/E^{fb}} = \begin{bmatrix} X \\ \{y\} \end{bmatrix}, \qquad V_q^{W/E^{fb}} = \begin{bmatrix} X \\ \{y,z\} \end{bmatrix},$$

i.e., we get the model with a smaller number of states identical to the model $\mathfrak{M}'$ from Example 3.5. However, since the underlying structure is not linearly ordered, we cannot apply Theorem 3.11.

On the other hand, E^{bb}, E^{rb}, Q^{fbb} and Q^{bfb} are fuzzy equivalences and fuzzy quasi-orders whose equivalence classes and aftersets are all different, and for that reason, they cannot reduce the number of states of the model.

Chapter 4

Weak simulations and bisimulations

> "So far as laws of mathematics refer to reality, they are not certain. And so far as they are certain, they do not refer to reality."
>
> *Albert Einstein, 1921.*

There are two main types of simulations and bisimulations. The first ones are known as strong simulations and strong bisimulations, or just simulations and bisimulations, and were studied in the previous chapter. The second ones are known as weak simulations and weak bisimulations, and they are used for (bi)simulating internal systems' actions (such as automata languages, transitions in labelled transition systems, formulae in Kripke models, etc.). As we shall see, in the fuzzy modal logic, the greatest weak bisimulation between two models which is fuzzy equivalence represents the "degree of logical equivalence" (in the sense of Definition 2.9).

The notion of bisimilarity was developed in the theory of process algebra by Hennessy and Milner as a formalization of the relation of *observational equivalence* between states of transition systems. Hennessy and Milner developed a logical system today known as Hennessy-Milner logic (HML) to establish the relationship between bisimilarity and logical equivalence (see [63, 64]).

In terms of Kripke models and modal logic, if two models are bisimilar then they are modally equivalent. However, the converse of this assertion is generally not true, i.e. equivalent worlds that satisfy the same set of formulae do not have to be bisimilar. The special class of models to which this applies is said to have the Hennessy-Milner property. Hence, the Hennessy-Milner property in modal logic, i.e., the property when modal equivalence coincides with bisimilarity for the image-finite models, is well-known in modal logic. Also, Hennessy-Milner property is valid for modally saturated models (see [11]).

Still, the question when the Hennessy-Milner property holds for fuzzy modal logic, is not the easy one, and it is mostly unexplored, although there are several papers on the subject. It is significant to mention the work of Fan (see [47]) who defined a fuzzy bisimulation for standard Gödel modal logic and its extension with converse modalities and generalized Hennessy-Milner theorem for these logics. Notion of bisimulations for many-valued modal languages over Heyting algebras was examined in [42] by Eleftheriou et al. They defined notions like *t-invariance*, *t-bisimilarity* and also the notion of *weak bisimulation*. In addition, they showed that for the image-finite models, t-invariance implies t-bisimilarity. We also need to mention other papers dealing with this subject such as [86, 90] where the Hennessy-Milner property was investigated for many-valued logic with a crisp accessibility relation; [9] where the Hennessy-Milner property was investigated via coalgebraic methods; [36] where the Hennessy-Milner property was investigated for many-valued modal logic with a many-valued accessibility relation; as well as the research in fuzzy description logic [96, 98, 102], etc.

The results from this chapter are partially published in [136].

The chapter consists of four sections. In the first Section 4.1, we define weak simulations and weak bisimulations for the non-empty set Ψ of formulae, and provide propositions that give their basic properties. Weak Ψ-bisimulation can be used to express the degree of modal equivalence between worlds w and w' with respect to the formulae from Ψ.

Section 4.2 provides several Hennessy-Milner type theorems for fuzzy multimodal logics over linearly ordered Heyting algebras. We show that the greatest weak $\Phi^+_{I,\mathscr{H}}$-bisimulation between the image-finite Kripke models coincides with the greatest forward bisimulation. Also, we show that the greatest weak $\Phi^-_{I,\mathscr{H}}$-bisimulation between domain-finite Kripke models coincides with the greatest backward bisimulation and that the greatest weak $\Phi_{I,\mathscr{H}}$-bisimulation for the set of all modal formulae between degree-finite Kripke models coincides with the greatest regular bisimulation. This means that in these cases the degrees of modal equivalences for $\Phi^+_{I,\mathscr{H}}$, $\Phi^-_{I,\mathscr{H}}$ and $\Phi_{I,\mathscr{H}}$ can be expressed using the greatest forward, backward and regular bisimulations. We also provide these types of theorems in special cases such as Propositional Modal Logic.

In Section 4.3 we present interesting computational examples which demonstrate applications of the Hennessy-Milner type theorems.

In Section 4.4, we use the concept of uniform fuzzy relation from [24] in conjunction with weak Ψ-simulations and weak Ψ-bisimulation. Therefore, uniform weak Ψ-simulations and uniform weak Ψ-bisimulation are defined and characterized. Further, it is shown that two fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ are weak Ψ-bisimulation equivalent, i.e., there is a uniform weak Ψ-bisimulation between them, if and only if there is a weak Ψ-isomorphism between the afterset Kripke models with respect to the greatest weak Ψ-bisimulation equivalences on $\mathfrak{M}$ and $\mathfrak{M}'$ (Theorem 4.16).

Results from Section 4.4 are closely related to [73, 91] where uniform weak bisimulations between two fuzzy automata are considered.

4.1 Definitions of weak simulations and bisimulations

The motivation for the introduction of weak simulations and bisimulations can be found in the theory of fuzzy automata (cf. [73]). It has been shown that the existence of weak simulation between two automata implies language inclusion between them while the existence of weak bisimulation implies language-equivalence.

Thus, we will define weak simulations and bisimulations to examine formulae inclusion and formulae-equivalence between two fuzzy Kripke models. To make the definitions of weak simulations and bisimulations as general as possible, we will define them on a set of formulae (not necessarily on the set of all formulae). Also, the question arises as to the relationship between strong bisimulations and weak bisimulations for some fragments of logic defined in Section 2.3.

Definition 4.1. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, let $\Psi \subseteq \Phi_{I,\mathscr{H}}$ be a non-empty set of formulae and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. We call φ a *weak forward simulation for the set* Ψ if it is a solution to the system of fuzzy relation inequalities:

$$V_p \leqslant V'_p \circ \varphi^{-1}, \qquad \text{for every } p \in PV, \tag{ws-1}$$
$$\varphi^{-1} \circ V_A \leqslant V'_A, \qquad \text{for every } A \in \Psi, \tag{ws-2}$$

and a *weak forward presimulation for the set* Ψ if it satisfies condition (*ws-2*).

We call φ a *weak backward simulation for the set* Ψ if satisfies

$$V_p \leqslant \varphi \circ V'_p, \qquad \text{for every } p \in PV, \tag{4.1}$$
$$V_A \circ \varphi \leqslant V'_A, \qquad \text{for every } A \in \Psi, \tag{4.2}$$

and a *weak backward presimulation for the set* Ψ if it satisfies condition (4.2). According to (3.1) and (3.2), concepts of *weak forward (pre)simulation* and *weak backward (pre)simulation for the set* Ψ mutually coincide. Therefore, if φ satisfies (*ws-1*) and (*ws-2*) we will simply call it *weak simulation for the set* Ψ, and a *weak presimulation for the set* Ψ if it satisfies only (*ws-2*).

Definition 4.2. We call φ a *weak bisimulation for the set* Ψ if both φ and φ^{-1} are weak simulations for the set Ψ, i.e., if φ satisfies

$$V_p \leqslant V'_p \circ \varphi^{-1}, \quad V'_p \leqslant V_p \circ \varphi, \qquad \text{for every } p \in PV, \tag{wb-1}$$
$$\varphi^{-1} \circ V_A \leqslant V'_A, \quad \varphi \circ V'_A \leqslant V_A, \qquad \text{for every } A \in \Psi, \tag{wb-2}$$

and φ is called a *weak prebisimulation for the set* Ψ if both φ and φ^{-1} are weak presimulations for the set Ψ, i.e., if φ satisfies (*wb-2*).

Let's notice that it is also possible to define four types of weak (pre)bisimulations, but they all mutually coincide. Note that for fuzzy automata, weak forward and weak backward bisimulations do not coincide (cf. [91], p. 68).

To avoid hard phrase *weak (pre)(bi)simulation for the set* Ψ, from now on, we will use the phrase *weak* Ψ*-(pre)(bi)simulation*. Also, analogously to strong simulations and bisimulations, we can define weak crisp simulations and bisimulations here as well.

In the sequel, by $\varphi^{ws}_*, \varphi^{ws}, \varphi^{wb}_*$ and φ^{wb} we will denote the greatest weak Ψ-presimulation, weak Ψ-simulation, weak Ψ-prebisimulation and weak Ψ-bisimulation, respectively.

Again, the meaning of weak simulation and bisimulation can best be explained in the case when $\mathfrak{M}$ and $\mathfrak{M}'$ are crisp Kripke models and φ is an ordinary crisp (Boolean-valued) binary relation. The condition (*ws-1*) is the same as (*fs-1*). The condition (*ws-2*) is very similar to condition (*fs-3*), but it does not refer only to the propositional variables but to all formulae from the set Ψ. Hence, (*ws-2*) means that if w' simulates w and the valuation V assigns the value "true" to the formula $A \in \Psi$ in the world w, then the valuation V' also assigns to this formula the value "true" in the world w' (see Figure 3.1).

Remark 4.1. *When $\Psi = PV$ then condition (wb-2) becomes*

$$\varphi^{-1} \circ V_p \leqslant V'_p, \quad \varphi \circ V'_p \leqslant V_p, \qquad \textit{for every } p \in PV,$$

which is equivalent to (θ-3) condition for $\theta \in \{fb, bb, fbb, bfb, rb\}$ using (3.1) and (3.2).

In this way, the condition (θ-3) is packed in the condition (wb-2) and with (wb-1), it can be said that the concepts of strong bisimulations and weak Ψ-bisimulation coincide on conditions (θ-1) and (θ-3) for $\theta \in \{fb, bb, fbb, bfb, rb\}$ when $PV \subseteq \Psi$.

Regardless of the definitions of weak Ψ-(pre)simulations and Ψ-(pre)bisimulations refer to the arbitrary set of formulae Ψ, we usually want Ψ to contain all propositional variables and also for the set Ψ, we usually take some fragments of $\Phi_{I,\mathscr{H}}$.

Remark 4.2. *Note that condition (ws-2) can be written down in an equivalent form:*

$$\varphi(w, w') \leqslant \bigwedge_{A \in \Psi} V_A(w) \to V'_A(w'), \tag{4.3}$$

for any $w \in W$ and $w' \in W'$. Hence, the greatest weak Ψ-presimulation is

$$\varphi^{ws}_*(w, w') = \bigwedge_{A \in \Psi} V_A(w) \to V'_A(w'), \tag{4.4}$$

for any $w \in W$ and $w' \in W'$. Therefore, the greatest weak Ψ-presimulation between two fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ can be interpreted as a measure of degrees of formulae inclusion between two fuzzy Kripke models on the set Ψ.

In particular, if $\varphi^{ws}_*(w, w') = t$, value t can be interpreted as a measure of formulae inclusion between worlds w and w' on the set Ψ.

On the other hand, condition (wb-2) can be written down in an equivalent form:

$$\varphi(w, w') \leqslant \bigwedge_{A \in \Psi} V_A(w) \leftrightarrow V'_A(w'), \tag{4.5}$$

for any $w \in W$ and $w' \in W'$. Hence, the greatest weak Ψ-prebisimulation is

$$\varphi^{wb}_*(w, w') = \bigwedge_{A \in \Psi} V_A(w) \leftrightarrow V'_A(w'), \tag{4.6}$$

for any $w \in W$ and $w' \in W'$. Therefore, the greatest weak Ψ-prebisimulation between two fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ can be interpreted as a measure of degrees of formulae equality on the set Ψ, i.e., a measure of how much fuzzy Kripke models are Ψ-equivalent.

In particular, if $\varphi^{wb}_*(w, w') = t$, value t can be interpreted as a measure of formulae equality between worlds w and w' on the set Ψ.

Figure 4.1 explains the structural differences between strong and weak bisimulation. Strong bisimulation is represented by dashed lines with an arrow at both ends that makes reference to the local properties of the worlds and to the structure of the models. On the other hand, the lower part of the figure schematically shows the

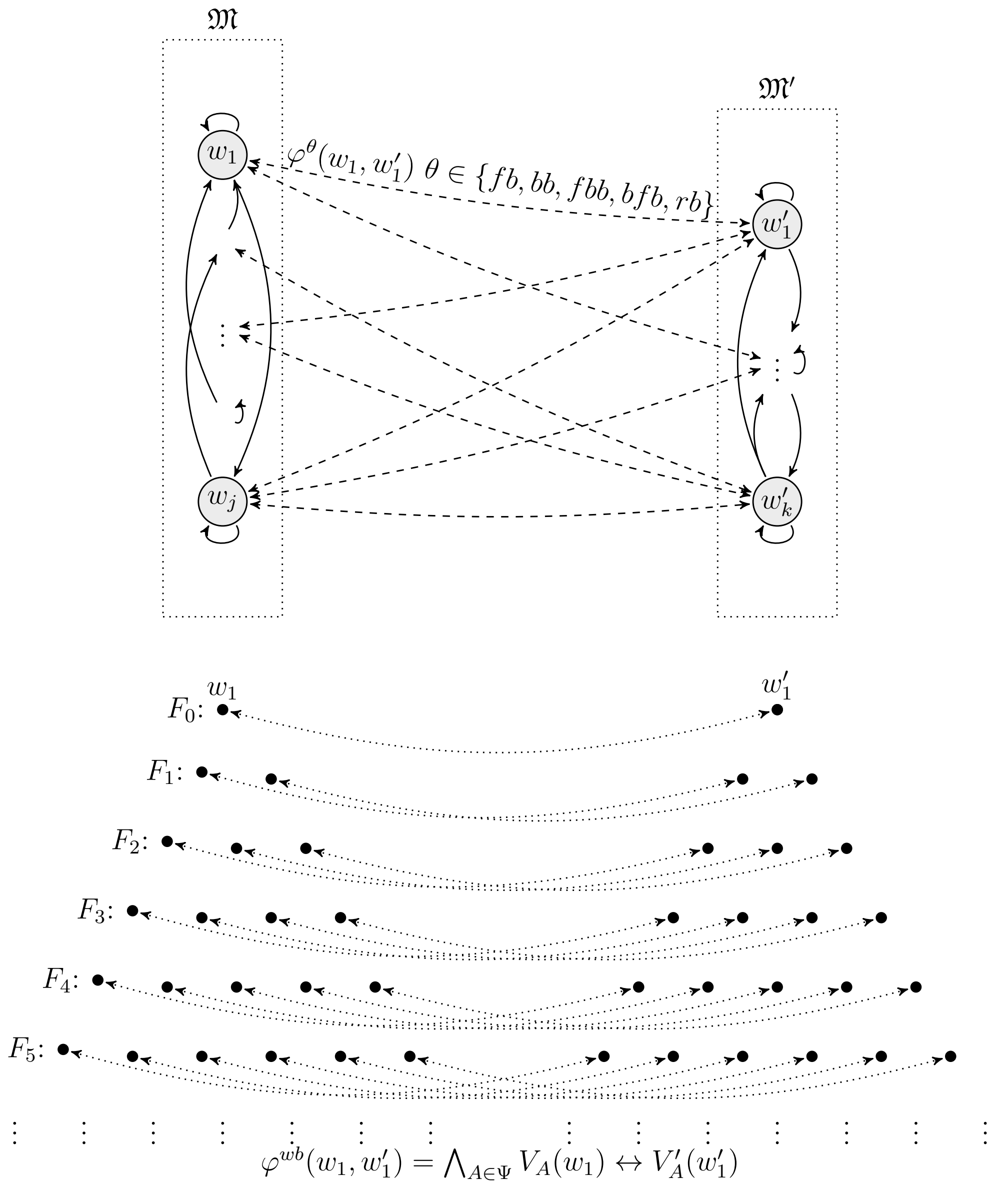

Figure 4.1: Structural differences between strong and weak bisimulation.

formulae in the worlds w_1 and w_1', which are arranged by formulae complexity (see Definition 2.14). Weak bisimulation between some set of formulae Ψ in the worlds w_1 and w_1' is represented by the dotted lines with an arrow at both ends which represents a degree of formulae equality. Finally, weak bisimulation is obtained by taking the infimum over the set of values $V_A(w) \leftrightarrow V_A'(w')$ where $A \in \Psi$.

Now that we have given the characterization of weak bisimulations, the question arises under what conditions are strong and weak bisimulations equal to each other? Do certain types of strong bisimulations correspond to weak ones for certain sets of formulae and under what conditions? The answers to these questions will be given in the next section.

In the previous chapter, we discussed duality between simulations and bisimulations (see Theorem 3.1). Similarly, we can state duality between weak simulations and bisimulations using duality between sets of formulae (2.7).

Now we can state and prove the following:

Theorem 4.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, let $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$ be the reverse fuzzy Kripke models for $\mathfrak{M}$ and $\mathfrak{M}'$, respectively, and let Ψ be an arbitrary set of formulae (usually, $\Psi \in \{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$.*

Then the following is true:

(a) *φ is weak Ψ-simulation/bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ if and only if φ is a weak Ψ^d-simulation/bisimulation between the reverse fuzzy Kripke models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$.*

(b) *The assertion* (a) *remains valid if the terms simulation and bisimulation are replaced by a presimulation and prebisimulation, respectively.*

Proof. The proof is a direct consequence of the definition of formulae, definitions of sets of formulae and reverse model. □

The set of weak (bi)simulations between two models is closed under arbitrary union. Further, the composition of two weak (bi)simulations is also weak (bi)simulation, similar to strong (bi)simulations (see Lemma 3.2). Therefore, we state the following lemma that can be easily proved.

Lemma 4.1. (a) *If $\{\varphi_\alpha\}_{\alpha\in Y}$ are weak simulations/bisimulations between models $\mathfrak{M}$ and $\mathfrak{M}'$, then $\bigvee_{\alpha\in Y}\varphi_\alpha$ is also a weak simulation/bisimulation between these models.*

(b) *If φ_1 is a weak simulation/bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$ and φ_2 is a weak simulation/bisimulation between models $\mathfrak{M}'$ and $\mathfrak{M}''$, then $\varphi_1 \circ \varphi_2$ is a weak simulation/bisimulation between $\mathfrak{M}$ and $\mathfrak{M}''$.*

(c) *The assertions* (a) *and* (b) *remain valid if the terms simulation and bisimulation are replaced by presimulation and prebisimulation, respectively.*

Let us consider an arbitrary fuzzy Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$. A weak Ψ-bisimulation from $\mathfrak{M}$ into itself will be called a *weak Ψ-bisimulation on $\mathfrak{M}$*. Since the equality relation is a weak Ψ-bisimulation on $\mathfrak{M}$ for every set Ψ, the set of all weak Ψ-bisimulations on $\mathfrak{M}$ is non-empty. Moreover, according to Remark 4.2 and (4.6) there is the greatest weak Ψ-bisimulation on $\mathfrak{M}$, and we can easily show that it is a fuzzy equivalence.

Weak Ψ-bisimulations on $\mathfrak{M}$ which are equivalences will be called *weak Ψ-bisimulation equivalences*. The set of all weak Ψ-bisimulation equivalences on $\mathfrak{M}$ will be denoted by $\mathscr{E}^{wb}(\mathfrak{M})$.

Theorem 4.2. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model. A fuzzy equivalence E on W is a weak Ψ-bisimulation on $\mathfrak{M}$ if and only if:*

$$E \circ V_A \leqslant V_A, \qquad A \in \Psi, \tag{4.7}$$

or equivalently,

$$E \circ V_A = V_A, \qquad A \in \Psi.$$

Proof. Let E be a fuzzy equivalence on W, which satisfies (4.7). Obviously, E satisfies (*wb-2*). In special case, when $A = p$ we have $E \circ V_p = V_p$ and by symmetry of E and Lemma 3.1 it follows that (*wb-1*) holds. Therefore, E is a weak Ψ-bisimulation on $\mathfrak{M}$.

Conversely, if E is a weak Ψ-bisimulation equivalence then (4.7) holds. □

Theorem 4.3. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model.*

The set $\mathscr{E}^{wb}(\mathfrak{M})$ of all weak Ψ-bisimulation fuzzy equivalence relations on $\mathfrak{M}$, forms a principal ideal of the lattice $\mathscr{E}(W)$ of all fuzzy equivalences on W generated by the relation E^{wb} on W for the set Ψ defined by

$$E^{wb}(u, v) = \bigwedge_{A\in\Psi} V_A(u) \leftrightarrow V_A(v), \qquad u, v \in W. \tag{4.8}$$

Proof. Apparently, E^{wb} for the set Ψ is a fuzzy equivalence. Also, according to Remark 4.2, E^{wb} is the greatest weak Ψ-bisimulation from $\mathfrak{M}$ into itself. Next, let $E \in \mathscr{E}(W)$ such that $E \leqslant E^{wb}$. Then, for every $A \in \Psi$, we have that $E \circ V_A \leqslant E^{wb} \circ V_A \leqslant V_A$. Hence, $E \in \mathscr{E}^{wb}(\mathfrak{M})$.

Conversely, let $E \in \mathscr{E}^{wb}(\mathfrak{M})$. Hence, E satisfies (4.7). Then,

$$\bigvee_{v\in W} E(u, v) \wedge V_A(v) \leqslant V_A(u), \qquad A \in \Psi,$$

that is

$$E(u, v) \wedge V_A(v) \leqslant V_A(u), \qquad A \in \Psi,$$

for every $u, v \in W$. By adjunction property (1.68), we have

$$E(u, v) \leqslant V_A(v) \to V_A(u), \qquad A \in \Psi,$$

for every $u, v \in A$. Similarly, using Lemma 3.1 we can conclude

$$E(u, v) \leqslant V_A(u) \to V_A(v), \qquad A \in \Psi,$$

and therefore,

$$E(u, v) \leqslant \bigwedge_{A\in\Psi} V_A(u) \leftrightarrow V_A(v), \qquad u, v \in W,$$

i.e.,

$$E(u, v) \leqslant E^{wb}(u, v), \qquad u, v \in W.$$

Thus, a fuzzy equivalence E is a solution to (4.7) if and only if $E \leqslant E^{wb}$. Accordingly, E^{wb} is the greatest solution to (4.7) and therefore $E \leqslant E^{wb}$ if and only if $E \in \mathscr{E}^{wb}(\mathfrak{M})$. $\square$

4.2 Hennessy-Milner Type Theorems

Bisimulations preserve the truth values of formulae. Hence, for basic modal language PML$^+$, bisimilar worlds are formulae equivalent with respect to the set of all formulae.

The converse of this assertion, which means that if worlds are formulae equivalent, they must be bisimilar, generally does not hold, but it is valid for some classes of Kripke models. This is exactly what the Hennessy-Milner theorem specifies.

Theorem 4.4 (Hennessy-Milner Theorem)**.** *Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be two image-finite Kripke models over the basic modal language* PML$^+$*. Then, for any $w \in W$ and $w' \in W'$, w and w' are bisimilar with respect to* PML$^+$ *if and only if w and w' are* PML$^+$*-equivalent.*

In other words, Hennessy-Milner Theorem says that two worlds w and w' are bisimilar with respect to PML$^+$ if and only if the sets of PML$^+$-formulae valid in w and w' coincide. In the context of fuzzy multimodal logics we can make the following generalization.

Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model and let $\Psi \subseteq \Phi_{I,\mathscr{H}}$ be some set of formulae. For each $w \in W$ we define a fuzzy subset V_w of Ψ by $V_w(A) = V(w, A)$, for every $A \in \Psi$. This means that the degree to which a formula A belongs to the fuzzy set V_w is equal to the truth degree of A in the world w. In classical modal logic V_w is simply the set of all formulae valid in the world w, so in the context of fuzzy modal logic we will say that V_w is the fuzzy set of formulae that are true (with a certain degree of truth) in w.

Now, let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let $\Psi \subseteq \Phi_{I,\mathscr{H}}$ be some set of formulae. As we noted in the previous section, the greatest weak (pre)bisimulation for the set Ψ (when it exists) is given by

$$\varphi(w, w') = \bigwedge_{A\in\Psi} V_A(w) \leftrightarrow V_A'(w') = \bigwedge_{A\in\Psi} V_w(A) \leftrightarrow V_{w'}'(A).$$

In fuzzy set theory, the expression far right in this equation is known as the degree of equality of fuzzy sets V_w and $V_{w'}'$, and therefore, the greatest weak Ψ-prebisimulation is the measure of the degree of equality of fuzzy sets of formulae from Ψ valid in two worlds w and w', that is, the measure of the degree of modal equivalence between worlds w and w' with respect to formulae from Ψ.

Note that Hennessy-Milner theorem replaces weak bisimulations by bisimulations, which is important because the greatest bisimulations between finite models can be computed by algorithms of polynomial complexity, in contrast to the greatest weak bisimulations, which are generally computed by algorithms of exponential complexity. An even bigger problem arises when computing the greatest fuzzy weak bisimulations.

Our aim is to prove several Hennessy-Milner type theorems for fuzzy multimodal logics over linearly ordered Heyting algebras. We will show that the degree of modal equivalence with respect to plus-formulae, between two worlds in image-finite Kripke models, can be expressed by the greatest forward (pre)bisimulation, the degree of modal equivalence with respect to minus-formulae, between two worlds in domain-finite Kripke models, can be expressed by the greatest backward (pre)bisimulation, and the degree of modal equivalence with respect to all formulae, between two worlds in degree-finite Kripke models, can be expressed by the greatest regular (pre)bisimulation.

First we prove the following theorem.

Theorem 4.5 (The Hennessy-Milner type theorem for plus-formulae)**.** *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two image-finite fuzzy Kripke models over a linearly ordered Heyting algebra $\mathscr{H}$. The greatest weak $\Phi_{I,\mathscr{H}}^+$-prebisimulation (resp. the greatest $\Phi_{I,\mathscr{H}}^+$-bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$, if it exists, is the greatest forward prebisimulation (resp. the greatest forward bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$.*

The proof is based on the next two lemmas.

Lemma 4.2. *Under the assumptions of Theorem 4.5, any forward prebisimulation (resp. forward bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$ is a weak $\Phi_{I,\mathscr{H}}^+$-prebisimulation (resp. $\Phi_{I,\mathscr{H}}^+$-bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. Let φ be a forward prebisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. To prove that φ is a weak $\Phi^+_{I,\mathscr{H}}$-prebisimulation we will prove that

$$\varphi(u,u') \leqslant V_A(u) \leftrightarrow V'_A(u'), \tag{4.9}$$

for all $u \in W$, $u' \in W'$ and every $A \in \Phi^+_{I,\mathscr{H}}$. This will be proved by induction on the complexity of a formula A.

Induction basis: If $A = p \in PV$, then from the fact that φ is forward bisimulation we have that $\varphi^{-1} \circ V_p \leqslant V'_p$ and $\varphi \circ V'_p \leqslant V_p$, which means that

$$\varphi^{-1}(u',u) \wedge V_p(u) \leqslant V'_p(u'), \quad \varphi(u,u') \wedge V'_p(u') \leqslant V_p(u),$$

for all $u \in W$, $u' \in W'$ and $p \in PV$. Using the adjunction property (1.68) we get

$$\varphi(u,u') \leqslant V_p(u) \to V'_p(u'), \quad \varphi(u,u') \leqslant V'_p(u') \to V_p(u),$$

and therefore,

$$\varphi(u,u') \leqslant V_p(u) \leftrightarrow V'_p(u'),$$

for all $u \in W$, $u' \in W'$ and $p \in PV$. Consequently, (4.9) holds for any propositional variable p. It trivially holds for any truth constant $\bar{t}$.

Induction step: a) Let $A = B \wedge C$ and let (4.9) hold for B and C, i.e.,

$$\varphi(u,u') \leqslant V_B(u) \leftrightarrow V'_B(u'), \quad \varphi(u,u') \leqslant V_C(u) \leftrightarrow V'_C(u'),$$

for all $u \in W$, $u' \in W'$. This yields

$$\varphi(u,u') \leqslant (V_B(u) \leftrightarrow V'_B(u')) \wedge (V_C(u) \leftrightarrow V'_C(u')).$$

Using the property of Heyting algebras (1.70), we get

$$\begin{aligned}
\varphi(u,u') &\leqslant (V_B(u) \leftrightarrow V'_B(u')) \wedge (V_C(u) \leftrightarrow V'_C(u')) \\
&\leqslant (V_B(u) \wedge V_C(u)) \leftrightarrow (V'_B(u') \wedge V'_C(u')) \\
&= V_{B\wedge C}(u) \leftrightarrow V'_{B\wedge C}(u') \\
&= V_A(u) \leftrightarrow V'_A(u'),
\end{aligned}$$

for all $u \in W$ and $u' \in W'$, so we conclude that (4.9) holds for $A = B \wedge C$.

b) Let A be of the form $B \to C$ and let (4.9) hold for B and C. In a similar way as in a), using the property of Heyting algebras (1.72), we prove that (4.9) also holds for A.

c) Let $A = \Diamond_i B$ and (4.9) let hold for B, i.e.,

$$\begin{aligned}
\varphi(u,u') &\leqslant V_B(u) \leftrightarrow V'_B(u') \\
&= (V_B(u) \to V'_B(u')) \wedge (V'_B(u') \to V_B(u)),
\end{aligned}$$

for all $u \in W$ and $u' \in W'$. Then it follows that

$$\varphi(u,u') \leqslant (V_B(u) \to V'_B(u')), \quad \varphi(u,u') \leqslant (V'_B(u') \to V_B(u)),$$

and using the adjunction property (1.68) we conclude

$$\varphi^{-1}(u',u) \wedge V_B(u) \leqslant V'_B(u'), \quad \varphi(u,u') \wedge V'_B(u') \leqslant V_B(u)$$

for all $u \in W$ and $u' \in W'$. Hence,

$$\varphi^{-1} \circ V_B \leqslant V'_B, \quad \varphi \circ V'_B \leqslant V_B,$$

and we have

$$\begin{aligned} \varphi^{-1} \circ V_A = \varphi^{-1} \circ R_i \circ V_B &\leqslant R'_i \circ \varphi^{-1} \circ V_B \qquad \text{(by (\textit{fb-2}))} \\ &\leqslant R'_i \circ V'_B = V'_A, \end{aligned}$$

for every $i \in I$. Now, from $\varphi^{-1} \circ V_A \leqslant V'_A$ we conclude that $\varphi(u, u') \leqslant V_A(u) \to V'_A(u')$. Thus, we conclude that $\varphi(u, u') \leqslant V'_A(u') \to V_A(u)$, for all $u \in W$ and $u' \in W'$, which means that

$$\varphi(u, u') \leqslant V_A(u) \leftrightarrow V'_A(u'),$$

for all $u \in W$ and $u' \in W'$. Therefore, we have proved that 4.9 is also true for $A = \Diamond_i B$.

d) Suppose that $A = \Box_i B$ and (4.9) holds for B. In a similar way as in c), from $\varphi(u, u') \leqslant V_B(u) \leftrightarrow V'_B(u')$, for all $u \in W$ and $u' \in W'$, we obtain

$$\varphi^{-1} \circ V_B \leqslant V'_B, \quad \varphi \circ V'_B \leqslant V_B.$$

Since the underlying Heyting algebra is linearly ordered, values $\varphi(u, u') = \varphi^{-1}(u', u)$, $V_A(u)$ and $V'_A(u')$ can be compared with each other, for all $u \in W$, $u' \in W'$, therefore, case analysis can be used.

If $\varphi^{-1}(u', u) \leqslant V_A(u) \wedge V_A(u')$ and $V_A(u) \neq V'_A(u')$, then

$$\varphi(u, u') = \varphi^{-1}(u', u) \leqslant V_A(u) \wedge V'_A(u') = V_A(u) \leftrightarrow V'_A(u').$$

In case $V_A(u) = V'_A(u')$ we have that $V_A(u) \leftrightarrow V'_A(u') = 1$, which again gives $\varphi(u, u') \leqslant V_A(u) \leftrightarrow V'_A(u')$.

Hence, we need to consider only the case when

$$\varphi^{-1}(u', u) > V_A(u) \wedge V'_A(u').$$

Without loss of generality, we can assume that $\varphi^{-1}(u', u) > V_A(u)$, and then we have:

$$\begin{aligned} V_A(u) &= \varphi^{-1}(u', u) \wedge V_A(u) \\ &= \varphi^{-1}(u', u) \wedge \bigwedge_{v \in W} (R_i(u, v) \to V_B(v)) \\ &= \bigwedge_{v \in W} \left[\varphi^{-1}(u', u) \wedge \big(R_i(u, v) \to V_B(v)\big)\right] \qquad \text{(by (1.73))} \\ &= \bigwedge_{v \in W} \left[\varphi^{-1}(u', u) \wedge \big(\varphi^{-1}(u', u) \wedge R_i(u, v) \to V_B(v)\big)\right] \qquad \text{(by (1.69))} \\ &= \varphi^{-1}(u', u) \wedge \bigwedge_{v \in W} \left[\varphi^{-1}(u', u) \wedge R_i(u, v) \to V_B(v)\right] \qquad \text{(by (1.73))} \end{aligned} \tag{4.10}$$

According to the starting assumption, φ is a forward prebisimulation, so it satisfies (*fb-2*), i.e.

$$\varphi^{-1} \circ R_i \leqslant R'_i \circ \varphi^{-1}, \qquad \text{for every } i \in I.$$

Next, since R'_i is image-finite, for each $v \in W$ we can find $v' \in W'$ such that

$$\varphi^{-1}(u', u) \wedge R_i(u, v) \leqslant R'_i(u', v') \wedge \varphi^{-1}(v', v),$$

and it follows

$$\big(\varphi^{-1}(u', u) \wedge R_i(u, v)\big) \to V_B(v) \geqslant \big(\varphi^{-1}(v', v) \wedge R'_i(u', v')\big) \to V_B(v).$$

Now, two cases need to be analyzed. If $V_B(v) = V'_B(v')$, then

$$R'_i(u', v') \to V_B(v) = R'_i(u', v') \to V'_B(v').$$

Since

$$\big(\varphi^{-1}(v', v) \wedge R'_i(u', v')\big) \to V_B(v) \geqslant R'_i(u', v') \to V_B(v),$$

it follows

$$\big(\varphi^{-1}(v', v) \wedge R'_i(u', v')\big) \to V_B(v) \geqslant R'_i(u', v') \to V'_B(v').$$

On the other hand, if $V_B(v) \neq V'_B(v')$, then by the induction hypothesis we have that

$$\varphi^{-1}(v', v) \leqslant (V_B(v) \leftrightarrow V'_B(v')) \leqslant V_B(v).$$

Thus,

$$\big(\varphi^{-1}(v', v) \wedge R'_i(u', v')\big) \to V_B(v) = 1 \geqslant R'_i(u', v') \to V'_B(v').$$

In both cases, we have shown that for any $v \in W$, we can find v' so that

$$\big(\varphi^{-1}(u', u) \wedge R_i(u, v)\big) \to V_B(v) \geqslant R'_i(u', v') \to V'_B(v').$$

Therefore,

$$\bigwedge_{v \in W} \big(\varphi^{-1}(u', u) \wedge R_i(u, v)\big) \to V_B(v) \geqslant \bigwedge_{v' \in W'} R'_i(u', v') \to V'_B(v') = V'_A(u')$$

and using (4.10) we conclude: $V_A(u) \geqslant \varphi^{-1}(u', u) \wedge V'_A(u')$. Because of the assumption that $\varphi^{-1}(u', u) > V_A(u)$, we have

$$V_A(u) \geqslant V'_A(u') \quad \text{and} \quad \varphi^{-1}(u', u) > V'_A(u').$$

Analogously, by the same reasoning we can prove that $V'_A(u') \geqslant V_A(u)$, since $\varphi^{-1}(u', u) > V'_A(u')$. Hence, we have $V_A(u) = V'_A(v')$, and since $\varphi(u, u') = \varphi^{-1}(u', u)$ it follows

$$\varphi(u, u') \leqslant V_A(u) \leftrightarrow V'_A(u') = 1$$

when $\varphi^{-1}(u', u) > V_A(u) \wedge V'_A(u')$.

This completes the proof of the statement that every forward prebisimulation is a weak $\Phi^+_{I,\mathscr{H}}$-prebisimulation. Also, this brings about that every forward bisimulation is a weak $\Phi^+_{I,\mathscr{H}}$-bisimulation, since the additional conditions (*fb-1*) and (*wb-1*) that distinguish between prebisimulations and bisimulations are the same in both cases. □

Lemma 4.3. *Under the assumptions of Theorem 4.5, the greatest weak $\Phi^+_{I,\mathscr{H}}$-prebisimulation (resp. the greatest $\Phi^+_{I,\mathscr{H}}$-bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$, if it exists, is a forward prebisimulation (resp. a forward bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. Let φ be a weak $\Phi_{I,\mathscr{H}}^{+}$-prebisimulation. According to Remark 4.1, φ satisfies condition (*fb-3*). Hence, it remains to prove that (*fb-2*) is true.

To prove that, we will use proof by a contradiction and the same method used in Lemma 2 from [47]. Namely, we will prove the assumption that (*wb-2*) is true while (*fb-2*) is not true, which leads to a contradiction. Therefore, let us assume that (*fb-2*) does not hold. This means that there exists $i \in I$ so that

$$\varphi^{-1} \circ R_i \not\leqslant R_i' \circ \varphi^{-1} \quad \text{or} \quad \varphi \circ R_i' \not\leqslant R_i \circ \varphi, \tag{4.11}$$

for some $i \in I$. We will handle only the case

$$\varphi^{-1} \circ R_i \not\leqslant R_i' \circ \varphi^{-1}, \tag{4.12}$$

for some $i \in I$, because the second case in (4.11) can be treated similarly. By the hypothesis, the underlying Heyting algebra $\mathscr{H}$ is linearly ordered, so formula (4.12) means that there are $u, v \in W$ and $u' \in W'$ such that

$$\varphi^{-1}(u', u) \wedge R_i(u, v) > \bigvee_{v' \in W'} R_i'(u', v') \wedge \varphi^{-1}(v', v). \tag{4.13}$$

Let $W'_{u'} = \{v' \in W' \mid R_i'(u', v') > 0\}$. By the assumption of the theorem, R_i' is image-finite, which means that $W'_{u'}$ is finite.

To simplify, set

$$\begin{aligned} &x = \varphi^{-1}(u', u), \quad y = R_i(u, v), \\ &x_{v'} = R_i'(u', v'), \quad y_{v'} = \varphi^{-1}(v', v), \end{aligned}$$

for each $v' \in W'_{u'}$. Then, formula (4.13) becomes

$$x \wedge y > \bigvee_{v' \in W'_{u'}} x_{v'} \wedge y_{v'}. \tag{4.14}$$

Due to (4.14), for each $v' \in W'_{u'}$ we have that $x_{v'} \wedge y_{v'} < x \wedge y$, and because of the linearity of the ordering in $\mathscr{H}$, we get that either $x_{v'} < x \wedge y$ or $y_{v'} < x \wedge y$.

<u>*Case $y_{v'} < x \wedge y$:*</u> If $y_{v'} < x \wedge y$, i.e.,

$$\varphi^{-1}(v', v) = \varphi(v, v') < x \wedge y,$$

then by the definition of $\varphi = \varphi_*^{wb}$, for each $v' \in W'_{u'}$ there exists $A_{v'} \in \Phi_{I,\mathscr{H}}^{+}$ such that

$$(V(v, A_{v'}) \leftrightarrow V'(v', A_{v'})) < x \wedge y.$$

In fact, since underlying algebra $\mathscr{H}$ is linearly ordered,

$$(V(v, A_{v'}) \leftrightarrow V'(v', A_{v'})) = V(v, A_{v'}) \wedge V'(v', A_{v'}),$$

for $V(v, A_{v'}) \neq V'(v', A_{v'})$ and then $A_{v'}$ can be any formula such that $V(v, A_{v'}) < x \wedge y$ or $V'(v', A_{v'}) < x \wedge y$.

Set $z_{v'} = V(v, A_{v'})$. Now we define $B_{v'}$, for each $v' \in W'_{u'}$, as follows:

$$B_{v'} = \begin{cases} \overline{1}, & \text{if } x_{v'} < x \wedge y \\ A_{v'} \leftrightarrow \overline{z_{v'}}, & \text{otherwise} \end{cases} \tag{4.15}$$

Note that if $x_{v'} \geqslant x \wedge y$, then we have that

$$\begin{aligned} V'(v', B_{v'}) &= V'(v', A_{v'} \leftrightarrow \overline{z_{v'}}) \\ &= V'(v', A_{v'}) \leftrightarrow V(v, A_{v'}) < x \wedge y \end{aligned}$$

and

$$\begin{aligned} V(v, B_{v'}) &= V(v, A_{v'} \leftrightarrow \overline{z_{v'}}) \\ &= V(v, A_{v'}) \leftrightarrow V(v, A_{v'}) = 1. \end{aligned}$$

Further, set $B = \bigwedge_{v' \in W'_{u'}} B_{v'}$. Then,

$$\begin{aligned} V'(u', \Diamond_i B) &= \bigvee_{v' \in W'_{u'}} R'_i(u', v') \wedge V'(v', B) \\ &= \bigvee_{v' \in W'_{u'}} x_{v'} \wedge V'(v', B). \end{aligned}$$

Thus,

$$V'(u', \Diamond_i B) \leqslant \left(\bigvee_{\substack{v' \in W'_{u'} \\ x_{v'} < x \wedge y}} x_{v'} \right) \vee \left(\bigvee_{\substack{v' \in W'_{u'} \\ x_{v'} \geqslant x \wedge y}} V'(v', B_{v'}) \right) < x \wedge y.$$

On the other hand,

$$\begin{aligned} V(u, \Diamond_i B) &= \bigvee_{v \in W} R_i(u, v) \wedge V(v, B) \\ &\geqslant R_i(u, v) \wedge V(v, B) = y \geqslant x \wedge y. \end{aligned}$$

Now, according to (4.5), we have

$$\begin{aligned} x = \varphi^{-1}(u', u) &\leqslant (V'(u', \Diamond_i B) \leftrightarrow V(u, \Diamond_i B)) \\ &= V'(u', \Diamond_i B) \wedge V(u, \Diamond_i B) \\ &= V'(u', \Diamond_i B) < x \wedge y \end{aligned}$$

which represents a contradiction.

Case $x_{v'} < x \wedge y$: Set $B = \overline{1}$ (B can also be any propositional formula that is a tautology, for example, $p \leftrightarrow p$).

In the same way as in the proof of the previous case, we conclude that

$$V'(u', \Diamond_i B) < x \wedge y, \quad V(u, \Diamond_i B) \geqslant y \geqslant x \wedge y,$$

whence

$$\begin{aligned} x = \varphi^{-1}(u', u) &= V'(u', \Diamond_i B) \wedge V(u, \Diamond_i B) \\ &= V'(u', \Diamond_i B) < x \wedge y, \end{aligned}$$

and again we get a contradiction.

Therefore, in all cases, the assumption that (*wb-2*) is true while (*fb-2*) is not true leads to a contradiction, whence we finally conclude that (*wb-2*) implies (*fb-2*), i.e., that every weak $\Phi^+_{I,\mathscr{H}}$-prebisimulation is a forward prebisimulation. Since the conditions (*fb-1*) and (*wb-1*) are the same, we also conclude that every weak $\Phi^+_{I,\mathscr{H}}$-bisimulation is a forward bisimulation.

This completes the proof of the lemma, as well as the proof of Theorem 4.5. $\square$

Remark 4.3. *Note that the proof of the Lemma 4.3 can be carried out by constructing formula $\Box_i B$ instead of $\Diamond_i B$. Now we give only the part of the proof that needs to be modified.*

Proof. Let $B = \bigwedge_{v' \in W'_{u'}} B_{v'}$. Then

$$\begin{aligned} V'(u', \Box_i B) &= \bigwedge_{v' \in W'_{u'}} R'_i(u', v') \to V'(v', B) = \bigwedge_{v' \in W'_{u'}} x_{v'} \to V'(v', B) \\ &= \bigwedge_{v' \in W'_{u'}} x_{v'} \to V'\left(v', \bigwedge_{v' \in W'_{u'}} B_{v'}\right) \\ &\leqslant \bigwedge_{v' \in W'_{u'}} x_{v'} \to V'(v', B_{v'}). \end{aligned}$$

Thus,

$$\begin{aligned} V'(u', \Box_i B) &= \left(\bigwedge_{\substack{v' \in W'_{u'} \\ x_{v'} < x \wedge y}} x_{v'} \to V'(v', B_{v'}) \right) \wedge \left(\bigwedge_{\substack{v' \in W'_{u'} \\ x_{v'} \geqslant x \wedge y}} x_{v'} \to V'(v', B_{v'}) \right) \\ &\leqslant \left(\bigwedge_{\substack{v' \in W'_{u'} \\ x_{v'} \geqslant x \wedge y}} x_{v'} \to V'(v', B_{v'}) \right) = \left(\bigwedge_{\substack{v' \in W'_{u'} \\ x_{v'} \geqslant x \wedge y}} V'(v', B_{v'}) \right) < x \wedge y. \end{aligned}$$

On the other hand,

$$\begin{aligned} V(u, \Box_i B) &= \bigwedge_{w \in W} R_i(u, w) \to V(w, B) \\ &\geqslant R_i(u, v) \to V(v, B) = y \to V(v, B) = 1. \end{aligned}$$

Hence, by the definition of $\varphi = \varphi_*^{wb}$ and (4.5) we have

$$\begin{aligned} x = \varphi^{-1}(u', u) &\leqslant (V'(u', \Box_i B) \leftrightarrow V(u, \Box_i B)) \\ &= (V'(u', \Box_i B) \leftrightarrow 1 = V'(u', \Box_i B) < x \wedge y, \end{aligned}$$

which again represents a contradiction. $\square$

Similarly, we prove the following two theorems.

Theorem 4.6 (The Hennessy-Milner type theorem for minus-formulae)**.** *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i \in I}, V')$ be two domain-finite fuzzy Kripke models over a linearly ordered Heyting algebra $\mathscr{H}$. The greatest weak $\Phi^-_{I,\mathscr{H}}$-prebisimulation (resp. the greatest $\Phi^-_{I,\mathscr{H}}$-bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$, if it exists, is the greatest backward prebisimulation (resp. the greatest backward bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Theorem 4.7 (The Hennessy-Milner type theorem for the set of all modal formulae)**.** *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i \in I}, V')$ be two degree-finite fuzzy Kripke models over a linearly ordered Heyting algebra $\mathscr{H}$. The greatest weak $\Phi_{I,\mathscr{H}}$-prebisimulation (resp. the greatest $\Phi_{I,\mathscr{H}}$-bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$, if it exists, is the greatest regular prebisimulation (resp. the greatest regular bisimulation) between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Also note that Theorem 4.6 follows from Theorem 4.5 by duality between the Kripke models and their reverse models. Theorem 4.7 is a direct consequence of Theorems 4.5 and 4.6.

Remark 4.4. *Lemma 4.2 generally does not hold in the case of weak Ψ-presimulations, i.e., inequality $\varphi_*^{fs}(w,w') \leqslant \varphi_*^{ws}(w,w')$, does not hold.*

For example, if a formula α is of the form $A \to B$ and the result holds for A and B, using adjunction property (1.68) we have

$$\varphi_*^{fs}(w,w') \leqslant V_A(w) \to V'_A(w'),$$
$$\varphi_*^{fs}(w,w') \leqslant V_B(w) \to V'_B(w'),$$

for every $w \in W$ and $w' \in W'$. Hence, we have

$$\varphi_*^{fs}(w,w') \leqslant (V_A(w) \to V'_A(w')) \wedge (V_B(w) \to V'_B(w')).$$

But, we want to prove $\varphi(w,w') \leqslant (V_A(w) \to V_B(w)) \wedge (V'_A(w') \to V'_B(w'))$ and for that, we need the property

$$(x_1 \to y_1) \wedge (x_2 \to y_2) \leqslant (x_1 \to x_2) \wedge (y_1 \to y_2),$$

which simply does not hold in the linearly ordered Heyting algebra. To make sure, we can take a Gödel $[0,1]$ structure and the following values, $x_1 = 0.7$, $y_1 = 0.8$, $x_2 = 0.6$ and $y_2 = 0.7$.

However, this does not mean that the Hennessy-Milner property is not valid for fuzzy simulations in another logic. For example, in [97] the Hennessy-Milner property for fuzzy simulations was given for Fuzzy Labelled Transition Systems in Fuzzy Propositional Dynamic Logic.

Now, using the fact that weak (pre)bisimulation is logical equivalence on a set of formulae, then the Hennessy-Milner theorems can be reformulated as follows:

Theorem 4.8 (The Hennessy-Milner theorem for PML$^+$)**.** *Let $\mathfrak{M} = (W,R,V)$ and $\mathfrak{M}' = (W',R',V')$ be two image-finite* PML$^+$ *models. Models $\mathfrak{M}$ and $\mathfrak{M}'$ are* PML$^+$*-equivalent if and only if they are forward bisimilar.*

In fact, from theorem it follows that if worlds w and w' are PML$^+$-equivalent then they are forward bisimilar. Thus, we obtain Theorem 2.24 from [11], p. 69.

Theorem 4.9 (The Hennessy-Milner theorem for PML$^-$)**.** *Let $\mathfrak{M} = (W,R,V)$ and $\mathfrak{M}' = (W',R',V')$ be two domain-finite* PML$^-$ *models. Models $\mathfrak{M}$ and $\mathfrak{M}'$ are* PML$^-$*-equivalent if and only if they are backward bisimilar.*

Theorem 4.10 (The Hennessy-Milner theorem for PML)**.** *Let $\mathfrak{M} = (W,R,V)$ and $\mathfrak{M}' = (W',R',V')$ be two degree-finite* PML *models. Models $\mathfrak{M}$ and $\mathfrak{M}'$ are* PML*-equivalent if and only if they are regular bisimilar.*

Also, an analogous statement as Remark 4.4 holds in Propositional Modal Logic, i.e., the Hennessy-Milner property is not valid for simulations.

4.3 Computational examples

This section gives examples that demonstrate the application of the Hennessy-Milner-type Theorems from the previous sections and clarifies relationships between different types of strong and weak bisimulations.

If we recall Example 3.1, according to Theorem 4.5 and Definition 2.9 we conclude that models $\mathfrak{M}$ and $\mathfrak{M}'$ are $\Phi^+_{I,\mathscr{H}}$-equivalent. However, this only confirmed the results from the previous chapter because the model $\mathfrak{M}'$ was created to be $\Phi^+_{I,\mathscr{H}}$-equivalent with models $\mathfrak{M}$. Therefore, in the following examples, we will present some models that are more interesting for the application of Theorems 4.5, 4.6 and 4.7.

Example 4.1. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure, where $W = \{t, u, v, w\}$, $W' = \{u', v', w'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R'_1 and fuzzy sets V_p, V_q, V'_p and V'_q are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0.3 & 1 & 0 & 0.3 \\ 0.8 & 0.4 & 0.8 & 0.9 \\ 0.4 & 0.9 & 0.1 & 0.5 \\ 1 & 0.2 & 1 & 0.3 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.6 \\ 1 \\ 0.6 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.9 \\ 0.4 \\ 0.9 \\ 0.4 \end{bmatrix},$$

$$R'_1 = \begin{bmatrix} 0 & 1 & 0.5 \\ 0.8 & 0.2 & 0.9 \\ 1 & 0.1 & 0.4 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 0.6 \\ 1 \\ 1 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 0.9 \\ 0.4 \\ 0.4 \end{bmatrix}.$$

Using algorithms for testing the existence of bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ and computing the greatest ones, we have:

$$\varphi^{fb}_* = \begin{bmatrix} 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \end{bmatrix}, \qquad \varphi^{bb}_* = \varphi^{bb} = \begin{bmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 1 & 0.4 & 0.4 \\ 0.4 & 0.4 & 1 \end{bmatrix},$$

$$\varphi^{fbb}_* = \varphi^{fbb} = \begin{bmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.4 \\ 1 & 0.4 & 0.4 \\ 0.4 & 0.5 & 1 \end{bmatrix}, \qquad \varphi^{bfb}_* = \varphi^{bfb} = \begin{bmatrix} 1 & 0.4 & 0.4 \\ 0.4 & 1 & 0.5 \\ 0.9 & 0.4 & 0.4 \\ 0.4 & 0.4 & 1 \end{bmatrix},$$

$$\varphi^{rb}_* = \begin{bmatrix} 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \\ 0.4 & 0.4 & 0.4 \end{bmatrix},$$

while φ^{fb} and φ^{rb} do not exist, since φ^{fb}_* and φ^{rb}_* do not satisfy (*fb-1*) and (*rb-1*), respectively.

According to Theorem 4.6, backward bisimulation φ^{bb} and weak $\Phi^-_{I,\mathscr{H}}$-bisimulation φ^{wb} are identical fuzzy relations. Therefore, according to Definition 2.9, it follows that models $\mathfrak{M}$ and $\mathfrak{M}'$ are $\Phi^-_{I,\mathscr{H}}$-equivalent.

If we consider the reverse fuzzy Kripke models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$, we have the opposite situation. Namely, in this case there are no *bb*- and *rb*-bisimulations. In this case, according to Theorem 4.5, and Definition 2.9 it follows that models $\mathfrak{M}^{-1}$ and $\mathfrak{M}'^{-1}$ are $\Phi^{+}_{I,\mathscr{H}}$-equivalent.

However, if we apply method for state reducing of the Kripke model from Section 3.6 on model $\mathfrak{M}$, we obtain model $\mathfrak{M}/E^{bb}$ represented by the following fuzzy matrix and column vectors:

$$R_1^{W/E^{bb}} = \begin{bmatrix} 0.4 & 1 & 0.5 \\ 0.8 & 0.4 & 0.9 \\ 1 & 0.4 & 0.4 \end{bmatrix}, \qquad V_p^{W/E^{bb}} = \begin{bmatrix} 0.6 \\ 1 \\ 1 \end{bmatrix}, \qquad V_q^{W/E^{bb}} = \begin{bmatrix} 0.9 \\ 0.4 \\ 0.4 \end{bmatrix}.$$

Therefore, models $\mathfrak{M}'$ and $\mathfrak{M}/E^{bb}$ are different, but they are both $\Phi^{-}_{I,\mathscr{H}}$-equivalent with model $\mathfrak{M}$.

Example 4.2. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure, where $W = \{u, v, w\}$, $W' = \{u', v', w'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R'_1 and fuzzy sets V_p and V_q are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 1 & 0.8 & 0.3 \\ 0.5 & 0 & 1 \\ 0.7 & 0.2 & 0.4 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.6 \\ 0.8 \\ 1 \end{bmatrix},$$

$$R'_1 = \begin{bmatrix} 0.9 & 0.8 & 0.3 \\ 0.5 & 0.4 & 0.8 \\ 0.7 & 0.5 & 0.4 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 0.6 \\ 0.8 \\ 1 \end{bmatrix}.$$

Using algorithms for testing the existence of bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ and computing the greatest ones, we have:

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 0.9 & 0.6 & 0.6 \\ 0.6 & 0.8 & 0.6 \\ 0.6 & 0.6 & 1 \end{bmatrix}, \qquad \varphi_*^{bb} = \begin{bmatrix} 0.9 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 0.8 \end{bmatrix},$$

$$\varphi_*^{fbb} = \varphi^{fbb} = \begin{bmatrix} 0.9 & 0.6 & 0.6 \\ 0.6 & 0.8 & 0.6 \\ 0.6 & 0.6 & 1 \end{bmatrix}, \qquad \varphi_*^{bfb} = \begin{bmatrix} 0.9 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 0.8 \end{bmatrix},$$

$$\varphi_*^{rb} = \begin{bmatrix} 0.9 & 0.6 & 0.6 \\ 0.6 & 0.8 & 0.6 \\ 0.6 & 0.6 & 0.8 \end{bmatrix},$$

while φ^{bb}, φ^{bfb} and φ^{rb} do not exist, since φ_*^{bb}, φ_*^{bfb} and φ_*^{rb} do not satisfy (*bb-1*), (*bfb-1*) and (*rb-1*), respectively.

According to Theorem 4.5, forward bisimulation φ^{fb} and weak $\Phi^{+}_{I,\mathscr{H}}$-bisimulation φ^{wb} are identical fuzzy relations. Nevertheless, according to Definition 2.9, it follows that models $\mathfrak{M}$ and $\mathfrak{M}'$ are not $\Phi^{+}_{I,\mathscr{H}}$-equivalent.

In fact, we can draw the following conclusion: For models to be logically equivalent, the weak bisimulation for the set Ψ must have at least one element 1 in each row and column. This situation can be interpreted in the following way: "models $\mathfrak{M}$ and $\mathfrak{M}'$ are as $\Phi^{+}_{I,\mathscr{H}}$-equivalent as they are forward bisimilar and vice versa."

According to the Definition 2.9 we can conclude that worlds w and w' are $\Phi^{+}_{I,\mathscr{H}}$-equivalent, while the worlds v and v' are $\Phi^{-}_{I,\mathscr{H}}$-equivalent.

The following example illustrates the situation where fuzzy Kripke models are restricted to crisp values $\{0,1\}$.

Example 4.3. Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be two Kripke models over the two-valued Boolean structure, where $W = \{t, u, v, w\}$, $W' = \{v', w'\}$. Relations R, R' and propositional variables V_p, V_q, V'_p and V'_q are represented by the following matrices and column vectors:

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$
$$R' = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Using algorithms for testing the existence of bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ and computing the greatest ones, we have:

$$\varphi^{fb}_* = \varphi^{fb} = \varphi^{bb}_* = \varphi^{bb} = \varphi^{fbb}_* = \varphi^{fbb} = \varphi^{bfb}_* = \varphi^{bfb} = \varphi^{rb}_* = \varphi^{rb} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

According to Theorem 4.10 and Definition 2.9, it follows that models $\mathfrak{M}$ and $\mathfrak{M}'$ are PML-equivalent. These models are also PML$^+$-equivalent and PML$^-$-equivalent.

4.4 Uniform weak simulations and bisimulations

In this section, we deal with weak simulations and bisimulations which are partial fuzzy functions and uniform fuzzy relations.

Lemma 4.4. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models such there exists at least one weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$. Then there exists the greatest weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$, which is a partial fuzzy function.*

Proof. Let $\{\varphi_\alpha\}_{\alpha\in Y} \in \mathscr{R}(W, W')$ be the family of all weak Ψ-bisimulations between models $\mathfrak{M}$ and $\mathfrak{M}'$. Denote $\varphi = \bigvee_{\alpha\in Y} \varphi_\alpha$. According to Lemma 4.1, φ is also weak Ψ-bisimulation, and it is the greatest one.

To prove that φ is a partial fuzzy function, we will show $\varphi\circ\varphi^{-1}\circ\varphi \leqslant \varphi$. Denote $\eta = \varphi\circ\varphi^{-1}\circ\varphi$. Then, for every $p \in PV$, we have:

$$V'_p \circ \eta^{-1} = V'_p \circ \varphi^{-1} \circ \varphi \circ \varphi^{-1} \geqslant V_p \circ \varphi \circ \varphi^{-1} \geqslant V'_p \circ \varphi^{-1} \geqslant V_p.$$

Hence, $V_p \leqslant V'_p \circ \eta^{-1}$ is proved. The other part $V'_p \leqslant V_p \circ \eta$ can be proved similarly. Therefore, condition (*wb-1*) holds.

Also, for every $A \in \Psi$, we have:

$$\eta^{-1} \circ V_A = \varphi^{-1} \circ \varphi \circ \varphi^{-1} \circ V_A \leqslant \varphi^{-1} \circ \varphi \circ V'_A \leqslant \varphi^{-1} \circ V_A \leqslant V'_A.$$

Hence, $\eta^{-1}\circ V_A \leqslant V'_A$ is proved. The other part $\eta\circ V'_A \leqslant V_A$ can be proved similarly. Therefore, condition (*wb-2*) holds, and η is a weak Ψ-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$. Hence, since $\eta \leqslant \varphi$, it follows that φ is a partial fuzzy function. $\square$

Now, we define uniform weak simulation and bisimulation. According to Lemma 3.1, concepts of *uniform weak forward (pre)simulation* and *uniform weak backward (pre)simulation for the set* Ψ mutually coincide, and we will simply call it *weak (pre)simulation.*

Definition 4.3. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, let $\Psi \subseteq \Phi_{I,\mathscr{H}}$ be a non-empty set of formulae and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. We call φ a *uniform weak (pre)simulation for the set* Ψ if:

(1) φ is a uniform fuzzy relation;
(2) φ is a weak (pre)simulation for the set Ψ.

Equivalently, if $\varphi \in \mathscr{R}(W, W')$ is a non-empty uniform fuzzy relation, then we call φ an *uniform weak simulation for the set* Ψ if it is a solution to the system of fuzzy relation equalities:

$$V_p \circ \varphi \circ \varphi^{-1} = V'_p \circ \varphi^{-1}, \qquad \text{for every } p \in PV \tag{uws-1}$$
$$\varphi^{-1} \circ V_A = V'_A, \qquad \text{for every } A \in \Psi, \tag{uws-2}$$

and φ is called a *uniform weak presimulation for the set* Ψ if it satisfies condition (*uws-2*).

Definition 4.4. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, let $\Psi \subseteq \Phi_{I,\mathscr{H}}$ be a non-empty set of formulae and let $\varphi \in \mathscr{R}(W, W')$ be a non-empty fuzzy relation. We call φ a *uniform weak (pre)bisimulation for the set* Ψ if:

(1) φ is a uniform fuzzy relation;
(2) φ and φ^{-1} are weak (pre)simulations for the set Ψ.

Equivalently, if $\varphi \in \mathscr{R}(W, W')$ is a non-empty uniform fuzzy relation, then we call φ an *uniform weak bisimulation for the set* Ψ if it is a solution to the system of fuzzy relation equalities:

$$V_p \circ \varphi \circ \varphi^{-1} = V'_p \circ \varphi^{-1}, \qquad V'_p \circ \varphi^{-1} \circ \varphi = V_p \circ \varphi, \qquad \text{for every } p \in PV \tag{uwb-1}$$
$$\varphi^{-1} \circ V_A = V'_A, \qquad \varphi \circ V'_A = V_A, \qquad \text{for every } A \in \Psi, \tag{uwb-2}$$

and φ is called a *uniform weak prebisimulation for the set* Ψ if both φ and φ^{-1} are uniform weak presimulations for the set Ψ, i.e. if φ satisfies (*uwb-2*).

Again, it is possible to define four types of uniform weak (pre)bisimulations, but they all mutually coincide. Also, similarly to the study of weak simulation and weak bisimulation, we usually want the set Ψ to contain all propositional variables and for the set Ψ, we usually take some fragments of $\Phi_{I,\mathscr{H}}$. Similarly to the above, we will use the term *uniform weak* Ψ*-(pre)(bi)simulations.* First, we have the following theorem.

Theorem 4.11. *Let* $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ *and* $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ *be two fuzzy Kripke models, let* $\varphi \in \mathscr{R}(W, W')$ *be a uniform fuzzy relation and let* Ψ *be some set of formulae. Then,* φ *is a weak* Ψ*-(pre)(bi)simulation if and only if* φ *is a uniform weak* Ψ*-(pre)(bi)simulation.*

Proof. We will prove the theorem in the case of weak Ψ-bisimulation. Other cases are less difficult.

Let φ be a weak Ψ-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$. We need to check only condition (*uwb-1*). By (*wb-1*) we have that:

$$\begin{aligned} V_p \circ \varphi \circ \varphi^{-1} &\leqslant V'_p \circ \varphi^{-1} \circ \varphi \circ \varphi^{-1} \qquad (\textit{wb-1}) \\ &= V'_p \circ \varphi^{-1}, \qquad (\text{Theorem 1.15, property 3}) \\ &= V_p \circ \varphi \circ \varphi^{-1} \qquad (\textit{wb-1}), \end{aligned}$$

and analogously we get

$$\begin{aligned} V'_p \circ \varphi^{-1} \circ \varphi &\leqslant V_p \circ \varphi \circ \varphi^{-1} \circ \varphi \qquad (\textit{wb-1}) \\ &= V_p \circ \varphi \qquad (\text{Theorem 1.15, property 3}) \\ &= V'_p \circ \varphi^{-1} \circ \varphi \qquad (\textit{wb-1}). \end{aligned}$$

Therefore, φ satisfies both conditions (*wb-1*) and (*wb-2*).

Conversely, let (*wb-1*) and (*wb-2*) hold. According to reflexivity of $\varphi \circ \varphi^{-1}$ and $\varphi^{-1} \circ \varphi$ we have

$$V_p \leqslant V_p \circ \varphi \circ \varphi^{-1} \leqslant V'_p \circ \varphi^{-1}, \qquad V'_p \leqslant V'_p \circ \varphi^{-1} \circ \varphi \leqslant V_p \circ \varphi, \qquad \text{for every } p \in PV.$$

Therefore, (*uwb-1*) holds and (*wb-2*) hold trivially. Thus, φ is a uniform weak Ψ-bisimulation. □

Remark 4.5. *Uniform weak Ψ-(bi)simulations are a special case of weak Ψ-(bi)simulations. Therefore, all results (theorems, lemmas, etc.) also hold for uniform weak Ψ-(bi)simulations.*

According to the definitions, it follows that uniform weak Ψ-pre(bi)simulation is equal to the greatest weak Ψ-pre(bi)simulation.

Lemma 4.5. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, let E be a fuzzy equivalence on $\mathfrak{M}$, and $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i\in I}, V^{W/E})$ be the afterset fuzzy Kripke model concerning E. If E is a weak Ψ-bisimulation, then*

$$V_A^{W/E}(E_w) = V_A(w), \qquad A \in \Psi,\ w \in W. \tag{4.16}$$

Proof. This follows immediately from the definition of $V_A^{W/E}$ and the fact that E is a weak Ψ-bisimulation equivalence on $\mathfrak{M}$, i.e.

$$V_A^{W/E}(E_w) = (E \circ V_A)(w) = V_A(w), \qquad \text{for every } A \in \Psi,\ w \in W.$$

□

The following theorem uses the notion of natural function 1.87 and gives us characterization when natural function φ_E is a weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}/E$ where E is a fuzzy equivalence on $\mathfrak{M}$.

Theorem 4.12. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, E a fuzzy equivalence on $\mathfrak{M}$, φ_E the natural function from W to W/E and $\mathfrak{M}/E = (W/E, \{R_i^{W/E}\}_{i\in I}, V^{W/E})$ be the afterset fuzzy Kripke model concerning E.*

Then, E is a weak Ψ-bisimulation equivalence on $\mathfrak{M}$ if and only if φ_E is a weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}/E$.

Proof. Let E be a weak Ψ-bisimulation equivalence on $\mathfrak{M}$. According to the Lemma 4.5, for arbitrary $w \in W$ and $p \in PV$ we have:

$$
\begin{aligned}
V_p(w) = E \circ V_p(w) &= \bigvee_{u\in W} E(w,u) \wedge V_p(u) = \bigvee_{u\in W} \varphi_E(w,E_u) \wedge V_p^{W/E}(E_u) \\
&= \bigvee_{u\in W} V_p^{W/E}(E_u) \wedge \varphi_E^{-1}(E_u,w) = V_p^{W/E} \circ \varphi_E^{-1}(w).
\end{aligned}
$$

Similarly, $V_p^{W/E} = V_p \circ \varphi_E$ can be proved. Hence, (*wb-1*) holds. Also, for arbitrary $w \in W$ and $A \in \Psi$ we have:

$$
\begin{aligned}
\varphi_E^{-1} \circ V_A(E_w) &= \bigvee_{u\in W} \varphi_E^{-1}(E_w,u) \wedge V_A(u) = \bigvee_{u\in W} E(w,u) \wedge V_A(u) \\
&= E \circ V_A(w) = V_A(w) = V_A^{W/E}(E_w),
\end{aligned}
$$

and similarly $\varphi_E \circ V_A^{W/E} = V_A(w)$ can be proved. Hence, (*wb-2*) holds. Thus, φ_E is a weak Ψ-bisimulation.

Conversely, let φ_E be a weak Ψ-bisimulation. Now, for arbitrary $A \in \Psi$ and $w \in W$ we have:

$$
E \circ V_A(w) = V_A^{W/E}(E_w) = \varphi_E(w,E_w) \wedge V_A^{W/E}(E_w) \leqslant \varphi_E \circ V_A^{W/E}(w) = V_A(w).
$$

Therefore, according to Theorem 4.2, E is a weak Ψ-bisimulation equivalence on $\mathfrak{M}$. $\square$

Definition 4.5. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models. A bijective function ϕ of W onto W' is called a *weak isomorphism for the set* Ψ of fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ if

$$
V(w,p) = V'(\phi(w),p), \text{ for every } p \in PV, \tag{4.17}
$$

$$
V(w,A) = V'(\phi(w),A), \text{ for every } A \in \Psi. \tag{4.18}
$$

Again, we will use the term *weak* Ψ*-isomorphism* instead of *weak isomorphism for the set* Ψ.

The notion of a weak Ψ-isomorphism generalizes the notion of an isomorphism between fuzzy Kripke models, that is, the following is true:

Lemma 4.6. *Any isomorphism between two fuzzy Kripke models is also a weak* Ψ*-isomorphism between these models, for every set* Ψ*.*

Proof. Lemma will be proved by induction on the complexity of formula A.

Induction basis: If $A = p \in PV$, then (4.17) trivially holds. Also, assertion trivially holds for any truth constant $\overline{t}$.

Induction step: a) Let $A = B\wedge C$, and let $V(w,B) = V'(\phi(w),B)$ and $V(w,C) = V'(\phi(w),C)$, then we have:

$$
\begin{aligned}
V(w,A) = V(w,B\wedge C) &= V(w,B) \wedge V(w,C) = V'(\phi(w),B) \wedge V'(\phi(w),C) \\
&= V'(\phi(w),B\wedge C) = V'(\phi(w),A)
\end{aligned}
$$

b) If A is of the form $B \to C$, the proof is analogous as in a).

c) Let $A = \Diamond_i B$, and let $V(w, B) = V'(\phi(w), B)$, and $R_i(u, v) = R'_i(\phi(u), \phi(v))$ for all $u, v, w \in W$. Now, we have:

$$V(w, A) = V(w, \Diamond_i B) = \bigvee_{u \in W} R_i(w, u) \wedge V(u, B)$$
$$= \bigvee_{u \in W} R'_i(\phi(w), \phi(u)) \wedge V'(\phi(u), B) = V'(\phi(w), A)$$

d) If A is of the form $\Box_i B$, the proof is analogous as in c) as well as for $\Diamond_i^- B$ and $\Box_i^- B$. $\square$

In the theory of fuzzy automata, weak forward and weak backward isomorphism can be defined from $\mathscr{A}$ to $\mathscr{B}$ (cf. [73, 91]). Weak forward (backward) bisimulation from fuzzy automata $\mathscr{A}$ to $\mathscr{B}$ can be understood as weak Ψ-bisimulation between Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ for plus (minus) formulae.

Lemma 4.7. *Let $\mathfrak{M}$ and $\mathfrak{M}'$ be two fuzzy Kripke models and Ψ some set of formulae. If there exists a weak Ψ-isomorphism of fuzzy Kripke models, then $\mathfrak{M}$ and $\mathfrak{M}'$ are Ψ-equivalent.*

Proof. Let $\phi : W \to W'$ be a weak Ψ-isomorphism. Then, for every $A \in \Psi$, we have $V(w, A) = V'(\phi(w), A)$, and it follows that the expression

$$\bigwedge_{A \in \Psi} V(w, A) \leftrightarrow V'(\phi(w), A)$$

is equal to 1. Hence, according to Definition 2.9, it follows that $\mathfrak{M}$ and $\mathfrak{M}'$ are Ψ-equivalent models. $\square$

The following lemma can be easily proved.

Lemma 4.8. (a) *If ϕ_1 is a weak Ψ-isomorphism between models $\mathfrak{M}$ and $\mathfrak{M}'$ and ϕ_2 is a weak Ψ-isomorphism between models $\mathfrak{M}'$ and $\mathfrak{M}''$, then $\phi_1 \circ \phi_2$ is a weak Ψ-isomorphism between $\mathfrak{M}$ and $\mathfrak{M}''$.*

(b) *If ϕ is a weak Ψ-isomorphism between models $\mathfrak{M}$ and $\mathfrak{M}'$, then the inverse $\phi^{-1} : W' \to W$ is a weak Ψ-isomorphism between models $\mathfrak{M}'$ and $\mathfrak{M}$.*

Lemma 4.9. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i \in I}, V')$ be two fuzzy Kripke models. If $\mathfrak{M}$ and $\mathfrak{M}'$ are weak Ψ-isomorphic, then there exists a uniform weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. Let $\phi : W \to W'$ be a weak Ψ-isomorphism between $\mathfrak{M}$ and $\mathfrak{M}'$. Define a fuzzy relation $\varphi \in \mathscr{R}(W, W')$ by:

$$\varphi(w, w') = \begin{cases} 1, & \text{if } w' = \phi(w) \\ 0, & \text{otherwise.} \end{cases}$$

It can be easily shown that φ is a surjective $\mathscr{L}$-function. Next, for every $w \in W$ and $p \in PV$, we have the following:

$$V_p(w) = V'_p(\phi(w)) = V'_p(\phi(w)) \wedge \varphi^{-1}(\phi(w), w) \leqslant V'_p \circ \varphi^{-1}(w).$$

Hence, $V_p \leqslant V_p' \circ \varphi^{-1}$ holds and similarly, $V_p' \leqslant V_p \circ \varphi$ can be proved. Therefore, condition (*wb-1*) holds.

Also, according to the definition of φ and property $0 \wedge a = a \wedge 0 = 0$, for every $w' \in W'$ and for every $A \in \Psi$ we have:

$$\begin{aligned}
\varphi^{-1} \circ V_A(w') &= \bigvee_{w \in W} \varphi^{-1}(w', w) \wedge V_A(w) \\
&= \bigvee_{\substack{w \in W, \\ w = \phi^{-1}(w')}} \varphi^{-1}(w', \phi^{-1}(w')) \wedge V_A(\phi^{-1}(w')) \\
&= \varphi^{-1}(w', \phi^{-1}(w')) \wedge V_A(\phi^{-1}(w')) \\
&= V_A(\phi^{-1}(w')) = V_A(\phi(\phi^{-1}(w'))) = V_A'(w').
\end{aligned}$$

Hence, $\varphi^{-1} \circ V_A \leqslant V_A'$ holds and similarly, $\varphi \circ V_A' \leqslant V_A$ can be proved. Therefore, condition (*wb-2*) holds and φ is weak Ψ-bisimulation. Now, by Lemma 4.4 there exists the greatest weak Ψ-bisimulation ξ between $\mathfrak{M}$ and $\mathfrak{M}'$ which is a partial fuzzy function. Since φ is a surjective $\mathscr{L}$-function and $\varphi \leqslant \xi$, then ξ is also a surjective $\mathscr{L}$-function. Whence, ξ is a uniform weak Ψ-bisimulation. $\square$

Theorem 4.13. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i \in I}, V')$ be two fuzzy Kripke models and $\varphi \in \mathscr{R}(W, W')$ be uniform fuzzy relation. Then φ is a weak Ψ-bisimulation if and only if the following is true:*

(1) *E_A^φ is a weak Ψ-bisimulation equivalence on the fuzzy Kripke model $\mathfrak{M}$;*
(2) *E_B^φ is a weak Ψ-bisimulation equivalence on the fuzzy Kripke model $\mathfrak{M}'$;*
(3) *$\widetilde{\varphi}$ is a weak Ψ-isomorphism of afterset fuzzy Kripke models $\mathfrak{M}/E_A^\varphi$ and $\mathfrak{M}'/E_B^\varphi$.*

Proof. To simplify notations, let denote $E = E_A^\varphi$ and $F = E_B^\varphi$. Also, let $\psi \in CR(\varphi)$ be an arbitrary crisp description of φ.

Let φ be a uniform weak Ψ-bisimulation. By (iv) and (v) of Theorem 1.15, we have that $E = \varphi \circ \varphi^{-1}$ and $F = \varphi^{-1} \circ \varphi$. Now,

$$V_A \leqslant \varphi \circ \varphi^{-1} \circ V_A \leqslant \varphi \circ V_A' \leqslant V_A,$$

so $E \circ V_A = V_A$ for every $A \in \Psi$, i.e., by Theorem 4.2, E is a weak Ψ-bisimulation fuzzy equivalence. Similarly, we prove that $F \circ V_A' = V_A'$, for every $A \in \Psi$, therefore (2) also holds. According to Theorem 1.16, $\widetilde{\varphi}$ is a bijective function of W/E onto W'/F. By Theorem 4.11 we have:

$$\begin{aligned}
V_p^{W/E}(E_w) &= V_p \circ E(w) = V_p \circ \varphi \circ \varphi^{-1}(w) = V_p' \circ \varphi^{-1}(w) \\
&= \bigvee_{w' \in W'} V_p'(w') \wedge \varphi(w, w') = \bigvee_{w' \in W'} V_p'(w') \wedge F(\psi(w), w') = V_p'^{W'/F}(F_{\psi(w)})
\end{aligned}$$

and hence, $V_p^{W/E}(E_w) = V_p'^{W'/F}(F_{\psi(w)}) = V_p'^{W'/F}(\widetilde{\varphi}(E_w))$. In the same way, we can show that $V_A^{W/E}(E_w) = V_A'^{W'/F}(F_{\psi(w)}) = V_A'^{W'/F}(\widetilde{\varphi}(E_w))$ for every $A \in \Psi$. Therefore, it follows that $\widetilde{\varphi}$ is a weak Ψ-isomorphism of afterset fuzzy Kripke models $\mathfrak{M}/E_A^\varphi$ and $\mathfrak{M}'/E_B^\varphi$.

Conversely, let (1), (2) and (3) hold. Then,

$$V_p(w) \leqslant V_p \circ E(w) = V_p^{W/E}(E_w) = V_p'^{W'/F}(\widetilde{\varphi}(E_w)) = V_p'^{W'/F}(F_{\psi(w)})$$

$$= \bigvee_{w'\in W'} V'_p(w') \wedge F(\psi(w), w') = \bigvee_{w'\in W'} V'_p(w') \wedge \varphi(w, w') = V'_p \circ \varphi^{-1}(w).$$

On the other hand, according to Lemma 4.8 and (3) it follows that $\widetilde{\varphi}^{-1}$ is a weak Ψ-isomorphism of afterset fuzzy Kripke models $\mathfrak{M}'/E^{\varphi}_B$ and $\mathfrak{M}/E^{\varphi}_A$. From $\widetilde{\varphi}(E_w) = F_{\psi(w)}$ it follows that $\widetilde{\varphi}^{-1}(F_{w'}) = E_{\psi(w')}$ where $\psi \in CR(\varphi^{-1})$. In that case, we have:

$$V'^{W'/F}_p(F_{w'}) = V^{W/E}_p(\widetilde{\varphi}^{-1}(F_{w'})) = V^{W/E}_p(E_{\psi(w')}).$$

Now, we have:

$$V'_p(w') \leqslant V'_p \circ F(w') = V^{W'/F}_p(F_{w'}) = V^{W/E}_p(\widetilde{\varphi}^{-1}(F_{w'})) = V^{W/E}_p(E_{\psi(w')})$$
$$= \bigvee_{w\in W} V_p(w) \wedge E(\psi(w'), w) = \bigvee_{w\in W} V_p(w) \wedge \varphi^{-1}(w', w) = V_p \circ \varphi(w').$$

Hence, (*wb-1*) holds.

Now, for arbitrary $A \in \Psi$ and $w \in W$ we have:

$$\varphi \circ V'_A(w) = \bigvee_{w'\in W'} \varphi(w, w') \wedge V'_A(w') = \bigvee_{w'\in W'} F(\psi(w), w') \wedge V'_A(w')$$
$$= V'^{W'/F}_A(F_{\psi(w)}) = V'^{W'/F}_A(\widetilde{\varphi}(E_w)) = V^{W/E}_A(E_w) = V_A(w).$$

Therefore, $\varphi \circ V'_A = V_A$, which yields $\varphi^{-1} \circ V_A = \varphi^{-1} \circ \varphi \circ V'_A = F \circ V'_A = V'_A$, and hence, φ satisfies (*wb-2*). So it follows that φ is weak Ψ-bisimulation. □

Theorem 4.14. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models, and let E be a weak Ψ-bisimulation fuzzy equivalence on $\mathfrak{M}$ and F weak Ψ-bisimulation fuzzy equivalence on $\mathfrak{M}'$*

Then there exists a uniform weak Ψ-bisimulation $\varphi \in \mathscr{R}(W, W')$ such that

$$E^{\varphi}_W = E \qquad \text{and } E^{\varphi}_{W'} = F, \tag{4.19}$$

if and only if there exists a weak Ψ-isomorphism $\phi : \mathfrak{M}/E \to \mathfrak{M}'/F$ such that for every $w_1, w_2 \in W$ we have

$$\widetilde{E}(E_{w_1}, E_{w_2}) = \widetilde{F}(\phi(E_{w_1}), (\phi(E_{w_2})). \tag{4.20}$$

Proof. The proof of this theorem is similar to the proof of Theorem 6.4 in [25], and it will be omitted. □

Theorem 4.15. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, let E be a weak Ψ-bisimulation equivalence on $\mathfrak{M}$ and F a fuzzy equivalence on $\mathfrak{M}$ such that $E \leqslant F$. Then F is a weak Ψ-bisimulation equivalence on $\mathfrak{M}$ if and only if F/E is a weak Ψ-bisimulation equivalence on $\mathfrak{M}/E$.*

Proof. Let E be a weak Ψ-bisimulation equivalence on W. Then, according to the definition of F/E and Lemma 4.5, for every $w \in W$ and $A \in \Psi$ we have

$$F \circ V_A(w) = F/E \circ V^{W/E}_A(E_w) \leqslant V^{W/E}_A(E_w) = V_A(w).$$

Therefore, we obtain that $(F/E) \circ V^{W/E}_A \leqslant V^{W/E}_A$ if and only if $F \circ V_A \leqslant V_A$, which was to be proved. □

Corollary 4.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be a fuzzy Kripke model, and let E and F be weak Ψ-bisimulation equivalences on W such that $E \leqslant F$. Then F is the greatest weak Ψ-bisimulation equivalence on W if and only if F/E is the greatest weak Ψ-bisimulation equivalence on A/E.*

Proof. This is a direct consequence of the previous theorem and Theorem 2.4. □

4.4.1 Weak bisimulation equivalent Kripke models

Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models. If there exists a complete and surjective weak Ψ-bisimulation from $\mathfrak{M}$ to $\mathfrak{M}'$ then we say that $\mathfrak{M}$ and $\mathfrak{M}'$ are weak Ψ-bisimulation equivalent for the set Ψ, or briefly WΨB-equivalent, and we write $\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}'$. Note that surjectivity and completeness of this Ψ-bisimulation mean that every world of W is Ψ-equivalent to some world of W', and vice versa. It is also worth noting, that if there exists a weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$, which is complete and surjective, then the greatest weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ has the same property, and according to Lemma 4.4, it is a uniform weak Ψ-bisimulation.

For arbitrary fuzzy Kripke models $\mathfrak{M}$, $\mathfrak{M}'$ and $\mathfrak{M}''$ we have the following:

$$\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}; \tag{4.21}$$

$$\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}' \Rightarrow \mathfrak{M}' \sim_{W\Psi B} \mathfrak{M}; \tag{4.22}$$

$$(\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}' \ \wedge\ \mathfrak{M}' \sim_{W\Psi B} \mathfrak{M}'') \Rightarrow \mathfrak{M} \sim_{W\Psi B} \mathfrak{M}''. \tag{4.23}$$

It is clear that (4.21) and (4.22) hold, since the identity function is a weak Ψ-bisimulation between $\mathfrak{M}$ and itself, and the inverse relation of any weak Ψ-bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ is a weak Ψ-bisimulation between $\mathfrak{M}'$ and $\mathfrak{M}$. Further, (4.23) follows from the fact that composition of two weak Ψ-bisimulation is also weak Ψ-bisimulation between corresponding models.

Lemma 4.10. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let ϕ be a weak Ψ-isomorphism from $\mathfrak{M}$ to $\mathfrak{M}'$. Let E and F be the greatest weak Ψ-bisimulation equivalences on $\mathfrak{M}$ and $\mathfrak{M}'$, respectively.*

Then for every $w_1, w_2 \in W$ the following holds:

$$E(w_1, w_2) = F(\phi(w_1), \phi(w_2)).$$

Proof. By the definition of the greatest weak Ψ-(pre)bisimulation equivalences, and a weak Ψ-isomorphism, we have:

$$\begin{aligned} E(w_1, w_2) &= \bigwedge_{A\in\Psi} V_A(w_1) \leftrightarrow V_A(w_2) \\ &= \bigwedge_{A\in\Psi} V_A(\phi(w_1)) \leftrightarrow V_A(\phi(w_2)) = F(\phi(w_1), \phi(w_2)), \end{aligned}$$

for every $w_1, w_2 \in W$. □

Theorem 4.16. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models and let E and F be the greatest weak Ψ-bisimulation equivalence on $\mathfrak{M}$ and $\mathfrak{M}'$. Then $\mathfrak{M}$ and $\mathfrak{M}'$ are WΨB-equivalent if and only if there exists a weak Ψ-isomorphism between afterset Kripke models $\mathfrak{M}/E$ and $\mathfrak{M}'/F$.*

Proof. Let $\mathfrak{M}$ and $\mathfrak{M}'$ be two WΨB-equivalent fuzzy Kripke models, that is, there exists a complete and surjective weak Ψ-bisimulation ϕ from $\mathfrak{M}$ to $\mathfrak{M}'$. Now, by Theorem 4.2, there exists the greatest weak Ψ-bisimulation φ from $\mathfrak{M}$ to $\mathfrak{M}'$, which is a partial fuzzy function. Since ϕ is complete and surjective and $\phi \leqslant \varphi$, then φ is also complete and surjective. Hence, φ is a uniform weak Ψ-bisimulation.

$$
\begin{tikzcd}
 & \mathfrak{M} \arrow[r, "\varphi", "\text{W}\Psi\text{B equivalence}"'] \arrow[d] \arrow[dddl, dotted, bend right] & \mathfrak{M}' \arrow[d] \arrow[dddr, dotted, bend left] & \\
 & \mathfrak{M}/E^{\varphi}_{W} \arrow[r, dashed, "\widetilde{\varphi}", "\text{weak } \Psi\text{-isomorphism}"'] \arrow[d, dotted] & \mathfrak{M}'/E^{\varphi}_{W'} \arrow[d, dotted] & \\
 & (\mathfrak{M}/E^{\varphi}_{W})/P \arrow[r, dashed, "\xi", "\text{weak } \Psi\text{-isomorphism}"'] & (\mathfrak{M}'/E^{\varphi}_{W'})/Q \arrow[dr, dashed, bend left, "\widetilde{\chi_2}^{-1}"] & \\
\mathfrak{M}/E \arrow[ur, dashed, "\widetilde{\chi_1}"] \arrow[rrr, dashed, "\eta = \widetilde{\chi_1} \circ \xi \circ \widetilde{\chi_2}^{-1}", "\text{weak } \Psi\text{-isomorphism}"'] & & & \mathfrak{M}'/F \arrow[ul, dashed, "\widetilde{\chi_2}"]
\end{tikzcd}
$$

Figure 4.2: Proof of the Theorem 4.16.

Now, according to Theorem 4.13, E^{φ}_{W} and $E^{\varphi}_{W'}$ are weak Ψ-bisimulation equivalences on $\mathfrak{M}$ and $\mathfrak{M}'$, respectively, and $\widetilde{\varphi}$ is weak Ψ-isomorphism of the afterset fuzzy Kripke models $\mathfrak{M}/E^{\varphi}_{W}$ and $\mathfrak{M}'/E^{\varphi}_{W'}$.

Let P and Q be respectively the greatest weak Ψ-bisimulation equivalences on $\mathfrak{M}/E^{\varphi}_{W}$ and $\mathfrak{M}'/E^{\varphi}_{W'}$. We define a function

$$\xi : (\mathfrak{M}/E^{\varphi}_{W})/P \to (\mathfrak{M}'/E^{\varphi}_{W'})/Q$$

by $\xi(P_w) = Q_{\widetilde{\varphi}(w)}$ for every $w \in E^{\varphi}_{W}$. Using Lemma 4.10 it is easy to prove that ξ is a well-defined bijective function and according to (4.16), (4.21), (4.22) and (4.23) and the fact that $\widetilde{\varphi}$ is a weak Ψ-isomorphism, we obtain that ξ is a weak Ψ-isomorphism.

By Corollary 4.1 it follows that $P = E/E^{\varphi}_{W}$ and $Q = F/E^{\varphi}_{W'}$, and according to Theorem 2.3, $\mathfrak{M}/E$ is isomorphic to $(\mathfrak{M}/E^{\varphi}_{W})/P$ and $\mathfrak{M}'/F$ is isomorphic to $(\mathfrak{M}'/E^{\varphi}_{W'})/Q$, so $\mathfrak{M}/E$ is isomorphic to $\mathfrak{M}'/F$. According to Lemma 4.6, we obtain that $\mathfrak{M}/E$ is weak Ψ-isomorphic to $\mathfrak{M}'/F$.

Simplified, we have shown the following:

$$\mathfrak{M}/E \cong (\mathfrak{M}/E^{\varphi}_{W})/P \cong (\mathfrak{M}'/E^{\varphi}_{W'})/Q \cong \mathfrak{M}'/F.$$

Conversely, if there exists a weak Ψ-isomorphism from $\mathfrak{M}/E$ to $\mathfrak{M}'/F$, then according to Lemma 4.9, $\mathfrak{M}/E$ is WΨB-equivalent to $\mathfrak{M}'/F$. Also, by Theorem 4.12, $\mathfrak{M}$ and $\mathfrak{M}/E$ are WΨB-equivalent and $\mathfrak{M}'$ and $\mathfrak{M}'/F$ are WΨB-equivalent. Now, according to 4.21, 4.22 and 4.23, $\mathfrak{M}$ is WΨB-equivalent to $\mathfrak{M}'$. $\square$

Figure 4.2 graphically represent the proof of the Theorem 4.16. Functions $\widetilde{\chi_1} : \mathfrak{M}/E \to (\mathfrak{M}/E^{\varphi}_{W})/P$, $\xi : (\mathfrak{M}/E^{\varphi}_{W})/P \to (\mathfrak{M}'/E^{\varphi}_{W'})/Q$ and $\widetilde{\chi_2} : \mathfrak{M}'/F \to (\mathfrak{M}'/E^{\varphi}_{W'})/Q$ are all isomorphisms and the composition given by $\eta = \widetilde{\chi_1} \circ \xi \circ \widetilde{\chi_2}^{-1}$ is also an isomorphism since the inverse and composition of isomorphisms is also an isomorphism.

In the following corollary, for a model $\mathfrak{M}'$ we say that is a *minimal fuzzy Kripke model* in the class $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$ of all Kripke models which are WΨB-equivalent to $\mathfrak{M}$ if the following hold:

(1) the number of states of the model $\mathfrak{M}'$ is less or equal to any other model from the class $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$.
(2) if the number of states of model $\mathfrak{M}'$ is equal to the number of states of model $\mathfrak{M}''$ from $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$, then they are isomorphic.

Corollary 4.2. *Let $\mathfrak{M}$ be a fuzzy Kripke model, let E be the greatest weak Ψ-bisimulation equivalence on $\mathfrak{M}$, and let $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$ be the class of all Kripke models which are $W\Psi B$-equivalent to $\mathfrak{M}$. Then, $\mathfrak{M}/E$ is a minimal fuzzy Kripke model in $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$. Moreover, if $\mathfrak{M}'$ is any minimal fuzzy Kripke model in $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$, then there exists a weak Ψ-isomorphism between $\mathfrak{M}/E$ and $\mathfrak{M}'$.*

Proof. Let $\mathfrak{M}'$ be an arbitrary minimal Kripke model in $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$, and let F be the greatest weak Ψ-bisimulation equivalence on $\mathfrak{M}'$. According to the previous theorem, there exists a weak Ψ-isomorphism between $\mathfrak{M}/E$ and $\mathfrak{M}'/F$, and by Theorem 4.12 and 4.21, 4.22 and 4.23, it follows that $\mathfrak{M}'/F \in \mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$. Now, by minimality of $\mathfrak{M}'$, we have that F is the equality relation on W'. Thus, we obtain $\mathfrak{M}'/F \cong W'$. Hence, there is a weak Ψ-isomorphism between $\mathfrak{M}/E$ and $\mathfrak{M}'$.

Therefore, $\mathfrak{M}/E$ is also a minimal Kripke model in $\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$. □

Chapter 5

Computation of weak simulations and bisimulations

> "All exact science is dominated by the idea of approximation."
>
> *Bertrand Russell*

Weak bisimulation equivalence is one of the most commonly used model checking tools. These concepts are also used in many areas of mathematics and computer science such as formal verification, modal logic, labelled transition systems, etc. There are several algorithms to deal with this problem, but the coarsest refinement of a state partition is the most famous and fastest. It is also known as Paige and Tarjan's algorithm (abbreviated as PTA) (cf. [107]). Later, PTA is improved by Dovier et al. (cf. [37]) which is known as a fast bisimulation algorithm (abbreviated as FBA). Hence, after PTA, several algorithms for LTSs, Automata, Kripke models, etc., have been proposed, based on relational coarsest partition (cf. [56, 60, 75, 113, 119, 123]). These algorithms are based on the crucial equivalence between the greatest bisimulation equivalence and the relational coarsest partition problem.

In the theory of model checking, the notion of a *non-flat* system is well known, and it means a system described implicitly as a synchronized product of elementary subsystems. Attempting to verify such a system usually leads to the infamous *state-space explosion problem*, probably the main limitation in model checking theory. Examples of systems prone to this problem are synchronized transition systems (e.g. Markov chains), various types of systems with Boolean variables, 1-safe Petri nets, etc.

The state explosion problem causes an exponential increase in the number of states, which very quickly surpasses the possibilities of programs for model checking. In the worst case, the number of states can be infinite. Still, over the years, various model checking techniques have been developed to handle state-space explosion problems. For short overviews see [89, 95, 112].

In our structure, i.e., Kripke models for fuzzy multimodal logics, the computation of weak bisimulations inevitably leads to the *formulae explosion problem* (see Example 2.2). That is why we will apply the strategy of rejecting those formulae that we do not need; in fact, we will discard all logically equivalent formulae, except the one that appeared first, and thus get a set that we will call a *reachable fuzzy sets*. As we will see, for fuzzy finite Kripke models over locally finite algebra, the number of reachable fuzzy sets is limited. That way, the number of formulae we work with is also limited and so we will be able to "control the explosion".

In the previous Chapter (Section 4.2), we have shown that computation of weak bisimulation for sets of plus-formulae, minus-formulae and all formulae can be reduced to the problems of computing the greatest forward, backward and regular bisimulations, respectively. Considering that the computation of weak bisimulations is a computationally hard problem, Hennessy-Milner type Theorems represent a great benefit. However, how to compute weak bisimulation for some other set of formulae? Therefore, below we will develop an algorithm for computing weak (pre)simulation/(pre)bisimulation for any set of formulae.

Let us note that the greatest weak presimulation and weak prebisimulation for finite set Ψ can be easily computed by

$$\varphi_*^{ws}(w,w') = \bigwedge_{A\in\Psi} V_A(w) \to V'_A(w'), \qquad \varphi_*^{wb}(w,w') = \bigwedge_{A\in\Psi} V_A(w) \leftrightarrow V'_A(w'),$$

for any $w \in W$ and $w' \in W'$. Hence, we will consider cases when the set Ψ is infinite, especially when Ψ belongs to $\{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$.

The chapter consists of four sections. Section 5.1 provides an algorithm for reachable fuzzy sets for fuzzy Kripke models over locally finite Heyting algebra. Then, in Section 5.2 we determine the computational complexity of the algorithm. Specifically, we roughly determine the upper limit of the computational complexity and consider some ideas that can be used to finer determine the complexity of the algorithm. An algorithm for computation of weak simulation and bisimulation for arbitrary sets is developed in Section 5.3. The algorithm is based on the algorithm for reachable fuzzy sets. The last Section 5.4 provides interesting computational examples for both of the developed algorithms.

5.1 Algorithm for reachable fuzzy sets

To avoid complicated notations, the lines over the truth constants (for example, see Definition 2.5) will be omitted. The meaning is clear from the context, and therefore we will emphasize it only where necessary.

From now on we will treat *fuzzy finite Kripke models*, i.e., fuzzy Kripke models with finite sets W, I and PV, defined over locally finite Heyting algebra $\mathscr{H} = (H, \wedge, \vee, \to, 0, 1)$. Note that fuzzy finite Kripke models are also image-finite, domain-finite and degree-finite models.

Under these assumptions, values from fuzzy finite Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ induce a locally finite subalgebra $\langle K\rangle \subseteq \mathscr{H}$, where

$$\langle K\rangle = \left\langle \bigcup_{i\in I}\{R_i(u,v) \mid u,v\in W\} \cup \bigcup_{w\in W}\{V(w,p) \mid p\in PV\} \cup \{0\}\right\rangle. \tag{5.1}$$

Let's note that the subalgebra $\langle K\rangle$ contains the element 1 because $0 \to 0 = 1$, and therefore we will immediately write element 1 as a constituent element of the subalgebra. For example, if we recall Example 2.6, then model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$

given by the following fuzzy matrix and column vectors

$$R_1 = \begin{bmatrix} 0.8 & 0.1 & 0.9 \\ 0.2 & 0.8 & 1 \\ 0.6 & 0.7 & 0.9 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \end{bmatrix},$$

induces a subalgebra $\langle K \rangle$ from the set of values $\{0, 0.1, 0.2, 0.6, 0.7, 0.8, 0.9, 1\}$.

Under these assumptions, the truth assignment V, i.e., function $V : W \times \Psi \to K \subseteq H$ has a finite codomain, for every set of formulae $\Psi \subseteq \Phi_{I,\mathscr{H}}$. As a consequence, the maximum number of formulae which have different truth assignments in the model $\mathfrak{M}$ can be calculated.

Using formula for variations with repetition, we can conclude that number of formulae which have different truth assignments is less or equal to $|K|^{|W|}$. Therefore, for the above-mentioned model $\mathfrak{M}$, the maximum formulae with different truth assignments are 8^3. The following example further clarifies this statement.

Example 5.1. Let $\mathfrak{M} = (W, R, V)$ be fuzzy Kripke model over two-valued Heyting algebra ($H = \{0, 1\}$), with one relation R, and let p be one propositional variable in the model. Let $W = \{u, v, w\}$, and since $|W| = 3$ and $|H| = 2$, then maximum 2^3 formula with different assignment can be obtained, as follows:

$$A_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix},$$

$$A_5 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad A_6 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad A_7 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \quad A_8 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

However, in the general case, the number of formulae that can be obtained is not always equal to $|H|^{|W|}$. The following example illustrates that situation.

Example 5.2 (Reachable fuzzy sets)**.** Let $\mathfrak{M} = (W, R, V)$ be fuzzy Kripke model over Gödel $[0, 1]$ algebra, where $W = \{u, v\}$. Fuzzy relation R and one fuzzy set V_p, are represented by the following fuzzy matrix and column vector:

$$R_1 = \begin{bmatrix} 1 & 1 \\ 0.5 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}.$$

According to (5.1), values from the model $\mathfrak{M}$ form the set $K = \{0, 0.5, 1\}$, and therefore we have the subalgebra $\langle K \rangle$. By applying all logical operations (unary and binary) on the sets of truth constants and propositional variables, only one new fuzzy set can be obtained:

$$\begin{bmatrix} 0.5 \\ 1 \end{bmatrix}.$$

Hence, in the given model $\mathfrak{M} = (W, R, V)$ only the following fuzzy sets can be reached:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \qquad \begin{bmatrix} 0.5 \\ 1 \end{bmatrix}, \qquad \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \qquad \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

and will be called *reachable fuzzy sets in the model*. The set that contains all reachable fuzzy sets will be denoted by $\mathcal{T}$. Obviously, it follows that $|\mathcal{T}| \leqslant |H|^{|W|}$.

Our goal is to develop an algorithm for reachable fuzzy sets for fuzzy finite Kripke models over a locally finite subalgebra $\langle K\rangle$ generated as in (5.1). Note that with the approach so far, we have eliminated all constants from the set $H \setminus K$. If we did not do that, in the case when the set H is infinite, the practical application of the algorithm would not be possible. However, we will consider later how adding or removing some constants from $\langle K\rangle$ can change the output of the algorithm.

The following algorithm gives us an inefficient procedure on how to get set $\mathcal{T}$ for a given model. We define the sequence of the set of formulae $\mathcal{T}_n$ with the following properties:

$$\mathcal{T}_n = \bigcup_{k=0}^{n} T_k.$$

Algorithm 5.1 (An inefficient procedure for reachable fuzzy sets construction)**.** *The input of this algorithm is a fuzzy finite $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i\in I})$ Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$, over the induced subalgebra $\langle K\rangle$, and the output is the set $\mathcal{T}$ - set of all reachable fuzzy sets in $\mathfrak{M}$.*

(A1) $\mathcal{T}_0 = T_0 = F_0$, *i.e.*, $\mathcal{T}_0 = \{\bar{t} \mid t \in K\} \cup PV$.

(A2) *Then, we compute T_1 and $\mathcal{T}_1$:*

$$T_1 = \{V_A \mid A \in F_1 \wedge V_A \notin \mathcal{T}_0\}$$
$$\mathcal{T}_1 = \bigcup_{k=0}^{1} T_k.$$

If $\mathcal{T}_1 = \mathcal{T}_0$, algorithm terminates and $\mathcal{T}$ is equal to $\mathcal{T}_0$.

(A3) *After the nth step let T_n and $\mathcal{T}_n$ have been constructed, then*

$$T_{n+1} = \{V_A \mid A \in F_{n+1} \wedge V_A \notin \mathcal{T}_n\}$$
$$\mathcal{T}_{n+1} = \bigcup_{k=0}^{n+1} T_k.$$

(A4) *If $\mathcal{T}_{2n+1} = \mathcal{T}_n$, then $\mathcal{T}$ is equal to T_n.*

Figure 5.1 graphically shows how Algorithm 5.1 is performed. In the first step, all values from F_0 are placed in T_0. In the next steps, only new values from F_k are transferred to set T_k, for $1 \leqslant k \leqslant n$. The stop criterion follows from the fact that the algorithm is based on the definition of sets F_k and will be explained in more detail and supported by examples.

If we consider an equivalence class of formula A in the set $\Phi_{I,\mathscr{H}}$, i.e.,

$$[A] = \{B \mid V_B(w) \leftrightarrow V_A(w) = 1,\ w \in W\},$$

then the algorithm can be understood as a process executed by formulae complexity and by which the first representative of the equivalence class is singled out. Therefore, the first representative means to be the first representative encountered by the algorithm during execution. The number of steps ensures that all possible classes are covered by the algorithm. The algorithm can be reformulated to create equivalence classes, but this would further complicate the notation and software implementation.

Theorem 5.1. *Algorithm 5.1 terminates in a finite number of steps and $\mathcal{T}$ is a set of all reachable fuzzy sets for given model $\mathfrak{M}$.*

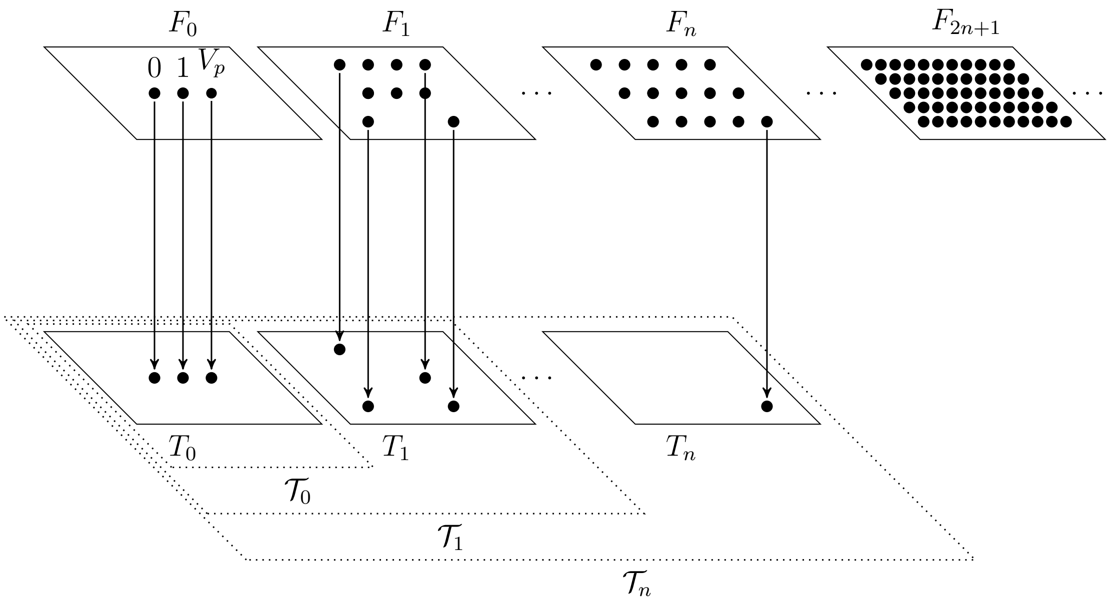

Figure 5.1: Graphical scheme of Algorithm 5.1

Proof. The proof by contradiction will be used to prove the theorem. First, suppose that algorithm does not terminate in a finite number of steps. This is in contradiction with the fact that $\mathcal{T}$ is a finite set.

Second, suppose that $\mathcal{T}$ is not a set of all reachable fuzzy sets in the model. It follows that there exists $k \in \mathbb{N}$ and formula $A \in \mathcal{T}_{2n+k+1}$ such that $V_A \notin \mathcal{T}$. Then, using the definition of sequence $\{\mathcal{T}_n\}$, we get

$$\mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \ldots \subseteq \mathcal{T}_n = \ldots = \mathcal{T}_{2n+1} = \ldots = \mathcal{T}_{2n+k} \subseteq \mathcal{T}_{2n+k+1}.$$

We distinguish the following cases:

(1) If formula A is of the form $*\alpha$ for some $* \in ULC$, it follows that $\alpha \in F_{2n+k}$ which means that $V_\alpha \in \mathcal{T}_{2n+k}$ and since $\mathcal{T}_n = \mathcal{T}_{2n+k}$ it follows that $V_\alpha \in \mathcal{T}_n$ and $V_A \in \mathcal{T}_{n+1}$. Since $\mathcal{T}_n = \mathcal{T}_{n+1}$, it follows that $V_A \in \mathcal{T}_n$ and $V_A \in \mathcal{T}$.

(2) If formula A is of the form $\alpha \star \beta$ for some $\star \in BLC$, it follows that $\alpha \in F_r$ and $\beta \in F_{2n+k-r}$ for some $0 \leqslant r \leqslant 2n+k$. Now, we can consider the following cases:
   - (i) $r < n$; Then, since $\beta \in F_{2n+k-r}$ it follows that $V_\beta \in \mathcal{T}_{2n+k-r}$ and then $V_\beta \in \mathcal{T}_n$. Since $V_\alpha \in \mathcal{T}_r$ and $r < n$, it follows that $V_\alpha \in \mathcal{T}_n$. Hence, $V_\alpha \star V_\beta \in \mathcal{T}_{2n+1}$ and $V_A \in \mathcal{T}_{2n+1}$, i.e., $V_A \in \mathcal{T}$ which give us a contradiction.
   - (ii) $n \leqslant r \leqslant 2n$, similar to the first item.
   - (iii) $2n \leqslant r \leqslant 2n+k$, similar to the first item.

□

Algorithm 5.1 and Theorem 5.1 are formulated for the $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i \in I})$ model, but they are also valid for every set of formulae $\Psi \subseteq \Phi_{I,\mathscr{H}}$.

The algorithm for reachable fuzzy sets construction is based on the definition of sequence $\{F_n\}_{n \in \mathbb{N}_0}$ and that is the reason for its inefficiency. Also, the one can expect that Algorithm will terminate when $\mathcal{T}_{n+1} = \mathcal{T}_n$, but this is not the case. For example, let $n = 6$, and $T_7 = \emptyset$. Therefore, $\mathcal{T}_6 = \mathcal{T}_7$. But, it may happen to exist some formulae $A \in F_3$ and $B \in F_4$ such that $A \wedge B \in F_8$ and $V_{A \wedge B} \notin \mathcal{T}_7$. To prevent this situation, condition $\mathcal{T}_{2n+1} = \mathcal{T}_n$ has been introduced (See Example 5.5).

Since the Algorithm is based on the computation of sequence $\{F_n\}_{n\in\mathbb{N}_0}$, it makes it practically unusable for any *non-trivial* fuzzy Kripke models. Note that we do not have to find all the formulae F_{2n+1} to get $\mathcal{T}_n$. However, it is significant because of the idea that the computation process ends in a finite number of steps.

Therefore, below is an improved version of the algorithm which is not based on the computation of sequence $\{F_n\}_{n\in\mathbb{N}_0}$.

The algorithm can be improved in the following way:

Algorithm 5.2 (Reachable fuzzy sets construction-improved version)**.** *The input of this algorithm is a fuzzy finite $\mathscr{H}(\{\Box_i, \Diamond_i, \Box_i^-, \Diamond_i^-\}_{i\in I})$ Kripke model $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ over the induced subalgebra $\langle K\rangle$, and the output is the set $\mathcal{T}$ - set of all reachable fuzzy sets in $\mathfrak{M}$.*

(A1) $\mathcal{T}_0 = T_0 = F_0$, *i.e.*, $\mathcal{T}_0 = \{\bar{t} \mid t \in K\} \cup PV$.

(A2) *Then, we compute T_1 and $\mathcal{T}_1$:*

$$T_1 = \{V_A \mid A = *\alpha,\ \ * \in ULC,\ \ \alpha \in \mathcal{T}_0,\ \ V_A \notin \mathcal{T}_0\}\cup$$
$$\cup\,\{V_A \mid A = \alpha \star \beta,\ \ \star \in BLC,\ \ \alpha, \beta \in \mathcal{T}_0,\ \ V_A \notin \mathcal{T}_0\},$$
$$\mathcal{T}_1 = \bigcup_{k=0}^{1} T_k.$$

If $\mathcal{T}_1 = \mathcal{T}_0$, the algorithm terminates and $\mathcal{T}$ is equal to T_0.

(A3) *After the nth step let T_n and $\mathcal{T}_n$ have been constructed, then*

$$T_{n+1} = \{V_A \mid A = *\alpha,\ \ * \in ULC,\ \ \alpha \in \mathcal{T}_n,\ \ V_A \notin \mathcal{T}_n\}\cup$$
$$\cup\,\{V_A \mid A = \alpha \star \beta,\ \ \star \in BLC,\ \ \alpha \in T_i,\ \ \beta \in T_j,\ \ i + j = n,\ \ V_A \notin \mathcal{T}_n\},$$
$$\mathcal{T}_{n+1} = \bigcup_{k=0}^{n} T_k.$$

(A4) *If $\mathcal{T}_{2n+1} = \mathcal{T}_n$, then $\mathcal{T} = \mathcal{T}_n$.*

In this way, to determine the set $\mathcal{T}$ it is not necessary to specify a set F_{2n+1} which significantly reduces the computational complexity of the algorithm. An improved version of the algorithm can be understood as an algorithm that works with equivalence classes of formulae instead of all formulae and thus avoids the formulae explosion problem.

The following example illustrates the application of the improved algorithm.

Example 5.3. Recall models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 2.6, i.e., we have the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0.8 & 0.1 & 0.9 \\ 0.2 & 0.8 & 1 \\ 0.6 & 0.7 & 0.9 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \end{bmatrix}, \tag{5.2}$$

$$R_1' = \begin{bmatrix} 0.8 & 0.7 \\ 0.6 & 0.8 \end{bmatrix}, \qquad V_p' = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}, \qquad V_q' = \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix}. \tag{5.3}$$

Model $\mathfrak{M}$ induces subalgebra $\langle K\rangle$ from the set of values $\{0, 0.1, 0.2, 0.6, 0.7, 0.8, 0.9, 1\}$. Using Algorithm 5.2 we can determine reachable fuzzy sets for fuzzy Kripke model $\mathfrak{M}$ and the set of formulae $\Phi_{I,\mathscr{H}}$.

First, we construct reachable fuzzy sets for model $\mathfrak{M}$:

$$T_0 = \left\{ t_{0,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad t_{0,2} = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \quad t_{0,3} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}, \quad t_{0,4} = \begin{bmatrix} 0.6 \\ 0.6 \\ 0.6 \end{bmatrix}, \quad t_{0,5} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.7 \end{bmatrix}, \right.$$
$$\left. t_{0,6} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \end{bmatrix}, \quad t_{0,7} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.9 \end{bmatrix}, \quad t_{0,8} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad t_{0,9} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \end{bmatrix}, \quad t_{0,10} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \end{bmatrix} \right\}$$

$$T_1 = \left\{ t_{1,1} = \Box t_{0,7} = \begin{bmatrix} 1 \\ 0.9 \\ 1 \end{bmatrix}, \quad t_{1,2} = \Box^- t_{0,6} = \begin{bmatrix} 1 \\ 1 \\ 0.8 \end{bmatrix}, \quad t_{1,3} = \Box^- t_{0,7} = \begin{bmatrix} 1 \\ 1 \\ 0.9 \end{bmatrix}, \right.$$
$$t_{1,4} = \Box^- t_{0,9} = \begin{bmatrix} 1 \\ 1 \\ 0.7 \end{bmatrix}, \quad t_{1,5} = \Box^- t_{0,10} = \begin{bmatrix} 1 \\ 0.7 \\ 0.7 \end{bmatrix}, \quad t_{1,6} = \Diamond t_{0,8} = \begin{bmatrix} 0.9 \\ 1 \\ 0.9 \end{bmatrix},$$
$$t_{1,7} = \Diamond t_{0,9} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.7 \end{bmatrix}, \quad t_{1,8} = \Diamond^- t_{0,9} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.9 \end{bmatrix}, \quad t_{1,9} = \Diamond^- t_{0,8} = \begin{bmatrix} 0.8 \\ 0.8 \\ 1 \end{bmatrix},$$
$$t_{1,10} = \Diamond^- t_{0,10} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.8 \end{bmatrix}, \quad t_{1,11} = t_{0,9} \to t_{0,5} = \begin{bmatrix} 0.7 \\ 0.7 \\ 1 \end{bmatrix},$$
$$t_{1,12} = t_{0,10} \to t_{0,5} = \begin{bmatrix} 0.7 \\ 1 \\ 1 \end{bmatrix}, \quad t_{1,13} = t_{0,9} \to t_{0,6} = \begin{bmatrix} 0.8 \\ 1 \\ 1 \end{bmatrix},$$
$$\left. t_{1,14} = t_{0,7} \to t_{0,9} = \begin{bmatrix} 1 \\ 0.8 \\ 0.7 \end{bmatrix}, \quad t_{1,15} = t_{0,9} \to t_{0,10} = \begin{bmatrix} 0.8 \\ 0.7 \\ 1 \end{bmatrix} \right\}$$

$$T_2 = \left\{ t_{2,1} = \Box t_{1,15} = \begin{bmatrix} 1 \\ 0.7 \\ 1 \end{bmatrix}, \quad t_{2,2} = \Box^- t_{1,12} = \begin{bmatrix} 0.7 \\ 1 \\ 0.7 \end{bmatrix}, \quad t_{2,3} = \Diamond^- t_{1,5} \begin{bmatrix} 0.8 \\ 0.7 \\ 0.9 \end{bmatrix}, \right.$$
$$t_{2,4} = \Diamond^- t_{1,11} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.9 \end{bmatrix}, \quad t_{2,5} = \Diamond^- t_{1,12} = \begin{bmatrix} 0.7 \\ 0.8 \\ 1 \end{bmatrix}, \quad t_{2,6} = t_{0,6} \wedge t_{1,11} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.8 \end{bmatrix},$$
$$t_{2,7} = t_{0,6} \wedge t_{1,12} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.8 \end{bmatrix}, \quad t_{2,8} = t_{0,6} \to t_{0,10} = \begin{bmatrix} 1 \\ 0.8 \\ 0.8 \end{bmatrix},$$
$$t_{2,9} = t_{0,7} \wedge t_{1,2} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad t_{2,10} = t_{1,2} \to t_{0,7} = \begin{bmatrix} 0.9 \\ 0.9 \\ 1 \end{bmatrix},$$
$$t_{2,11} = t_{0,7} \wedge t_{1,4} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.7 \end{bmatrix}, \quad t_{2,12} = t_{0,7} \wedge t_{1,5} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.7 \end{bmatrix},$$
$$t_{2,13} = t_{1,5} \to t_{0,7} = \begin{bmatrix} 0.9 \\ 1 \\ 1 \end{bmatrix}, \quad t_{2,14} = t_{0,7} \wedge t_{1,12} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.9 \end{bmatrix},$$

$$t_{2,15} = t_{1,12} \to t_{0,7} = \begin{bmatrix} 1 \\ 0.9 \\ 0.9 \end{bmatrix}, \quad t_{2,16} = t_{0,7} \wedge t_{1,13} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.9 \end{bmatrix},$$

$$t_{2,17} = t_{1,4} \to t_{0,9} = \begin{bmatrix} 0.9 \\ 0.8 \\ 1 \end{bmatrix}, \quad t_{2,18} = t_{0,9} \wedge t_{1,12} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.7 \end{bmatrix} \Bigg\}$$

$$T_3 = \Bigg\{ t_{3,1} = \Box t_{2,14} = \begin{bmatrix} 0.7 \\ 0.9 \\ 1 \end{bmatrix}, \quad t_{3,2} = \Diamond^{-} t_{2,5} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.9 \end{bmatrix}, \quad t_{3,3} = t_{2,1} \to t_{0,6} = \begin{bmatrix} 0.8 \\ 1 \\ 0.8 \end{bmatrix},$$

$$t_{3,4} = t_{2,2} \to t_{0,6} \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}, \quad t_{3,5} = t_{0,7} \wedge t_{2,1} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.9 \end{bmatrix}, \quad t_{3,6} = t_{0,7} \wedge t_{2,2} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.7 \end{bmatrix},$$

$$t_{3,7} = t_{0,7} \wedge t_{2,17} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.8 \end{bmatrix}, \quad t_{3,8} = t_{0,7} \wedge t_{2,17} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.9 \end{bmatrix},$$

$$t_{3,9} = t_{2,1} \to t_{0,9} = \begin{bmatrix} 0.9 \\ 1 \\ 0.7 \end{bmatrix}, \quad t_{3,10} = t_{2,1} \to t_{0,10} = \begin{bmatrix} 0.8 \\ 1 \\ 0.7 \end{bmatrix},$$

$$t_{3,11} = t_{1,1} \wedge t_{1,2} = \begin{bmatrix} 1 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad t_{3,12} = t_{1,1} \wedge t_{1,4} = \begin{bmatrix} 1 \\ 0.9 \\ 0.7 \end{bmatrix},$$

$$t_{3,13} = t_{1,1} \wedge t_{1,13} = \begin{bmatrix} 0.8 \\ 0.9 \\ 1 \end{bmatrix}, \quad t_{3,14} = t_{1,2} \wedge t_{1,6} = \begin{bmatrix} 0.9 \\ 1 \\ 0.8 \end{bmatrix},$$

$$t_{3,15} = t_{1,2} \wedge t_{1,12} = \begin{bmatrix} 0.7 \\ 1 \\ 0.8 \end{bmatrix}, \quad t_{3,16} = t_{1,3} \wedge t_{1,12} = \begin{bmatrix} 0.7 \\ 1 \\ 0.9 \end{bmatrix},$$

$$t_{3,17} = t_{1,3} \wedge t_{1,13} = \begin{bmatrix} 0.8 \\ 1 \\ 0.9 \end{bmatrix}, \quad t_{3,18} = t_{1,8} \to t_{1,10} = \begin{bmatrix} 1 \\ 0.7 \\ 0.8 \end{bmatrix},$$

$$t_{3,19} = t_{1,12} \to t_{1,8} = \begin{bmatrix} 1 \\ 0.8 \\ 0.9 \end{bmatrix} \Bigg\}$$

$$T_4 = \Bigg\{ t_{4,1} = t_{0,7} \wedge t_{3,3} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad t_{4,2} = t_{0,7} \wedge t_{3,10} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.7 \end{bmatrix},$$

$$t_{4,3} = t_{0,7} \wedge t_{3,15} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad t_{4,4} = t_{0,7} \wedge t_{3,18} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.8 \end{bmatrix},$$

$$t_{4,5} = t_{1,3} \wedge t_{2,1} = \begin{bmatrix} 1 \\ 0.7 \\ 0.9 \end{bmatrix}, \quad t_{4,6} = t_{1,4} \to t_{2,12} = \begin{bmatrix} 0.9 \\ 0.7 \\ 1 \end{bmatrix} \Bigg\}$$

And, finally $T_5 = T_6 = T_7 = T_8 = T_9 = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_4$. Note that $|\mathcal{T}| = 10 + 15 + 18 + 19 + 6 = 68$.

Let's also note that the method of determining T_k sets is not uniquely determined. For example, the element $t_{3,3}$ can also be obtained as $t_{1,2} \wedge t_{1,13}$. We can understand this as the fact that the formulae $t_{2,1} \to t_{0,6}$ and $t_{1,2} \wedge t_{1,13}$ belong to the same equivalence class whose representative is the element $t_{3,3}$.

Now, we construct reachable fuzzy sets for model $\mathfrak{M}'$. Model $\mathfrak{M}'$ induces subalgebra $\langle K \rangle$ from the set of values $\{0, 0.6, 0.7, 0.8, 0.9, 1\}$.

$$
\begin{aligned}
T'_0 = \Bigg\{ & t'_{0,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad t'_{0,2} = \begin{bmatrix} 0.6 \\ 0.6 \end{bmatrix}, \quad t'_{0,3} = \begin{bmatrix} 0.7 \\ 0.7 \end{bmatrix}, \quad t'_{0,4} = \begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix}, \quad t'_{0,5} = \begin{bmatrix} 0.9 \\ 0.9 \end{bmatrix}, \\
& t'_{0,6} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad t'_{0,7} = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}, \quad t'_{0,8} = \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix} \Bigg\} \\
T'_1 = \Bigg\{ & t'_{1,1} = \Box t'_{0,8} = \begin{bmatrix} 1 \\ 0.7 \end{bmatrix}, \quad t'_{1,2} = t'_{0,8} \to t'_{0,3} = \begin{bmatrix} 0.7 \\ 1 \end{bmatrix}, \\
& t'_{1,3} = t'_{0,4} \to t'_{0,7} = \begin{bmatrix} 0.8 \\ 1 \end{bmatrix}, \quad t'_{1,4} = t'_{0,5} \to t'_{0,7} = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}, \Bigg\} \\
T'_2 = \Bigg\{ & t'_{2,1} = \Diamond t'_{1,1} = \begin{bmatrix} 0.7 \\ 0.8 \end{bmatrix}, \quad t'_{2,2} = t'_{0,5} \wedge t'_{1,1} = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix}, \\
& t'_{2,3} = t'_{1,1} \to t'_{0,5} = \begin{bmatrix} 0.9 \\ 1 \end{bmatrix}, \quad t'_{2,4} = t'_{0,5} \wedge t'_{1,2} = \begin{bmatrix} 0.7 \\ 0.9 \end{bmatrix}, \\
& t'_{2,5} = t'_{1,2} \to t'_{0,5} = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}, \quad t'_{2,6} = t'_{0,5} \wedge t'_{1,3} = \begin{bmatrix} 0.8 \\ 0.9 \end{bmatrix} \Bigg\}
\end{aligned}
$$

And, finally $T'_3 = T'_4 = T'_5 = \emptyset$. Hence, $\mathcal{T}' = \mathcal{T}'_2$. Note that $|\mathcal{T}'| = 8 + 4 + 6 = 18$.

If we applied the algorithm for model $\mathfrak{M}'$ over the subalgebra $\langle K \rangle = \langle \{0, 0.1, 0.2, 0.6, 0.7, 0.8, 0.9, 1\} \rangle$, only the initial set T_0 would have a plus of these two fuzzy column vectors:

$$\begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}, \quad \begin{bmatrix} 0.2 \\ 0.2 \end{bmatrix}.$$

5.2 Complexity of the algorithm for reachable fuzzy sets

In this section, we will give a rough estimate of the complexity of Algorithm 5.2. We will also suggest some directions of research that could give a finer assessment.

Lemma 5.1. *Let $m_0, m_1, \ldots m_k$ be a sequence of $k+1$ non-negative integers. Then we have:*

$$\sum_{r=1}^{2k+1} \sum_{i+j=r-1} m_i m_j = \left(\sum_{i=0}^{k} m_i \right)^2. \tag{5.4}$$

Proof. The proof follows from the next set of equations (see Figure 5.2):

$$
\begin{aligned}
r = 1: \quad & \sum_{i+j=0} m_i m_j = m_0^2 \\
r = 2: \quad & \sum_{i+j=1} m_i m_j = m_0 m_1 + m_1 m_0
\end{aligned}
$$

$$r = 3: \quad \sum_{i+j=2} m_i m_j = m_0 m_2 + m_1^2 + m_2 m_0$$

$$\vdots$$

$$r = k: \quad \sum_{i+j=k-1} m_i m_j = m_0 m_{k-1} + m_1 m_{k-2} + \ldots + m_{k-2} m_1 + m_{k-1} m_0$$

$$\vdots$$

$$r = 2k: \quad \sum_{i+j=2k-1} m_i m_j = m_{k-1} m_k + m_k m_{k-1}$$

$$r = 2k+1: \quad \sum_{i+j=2k} m_i m_j = m_k^2$$

□

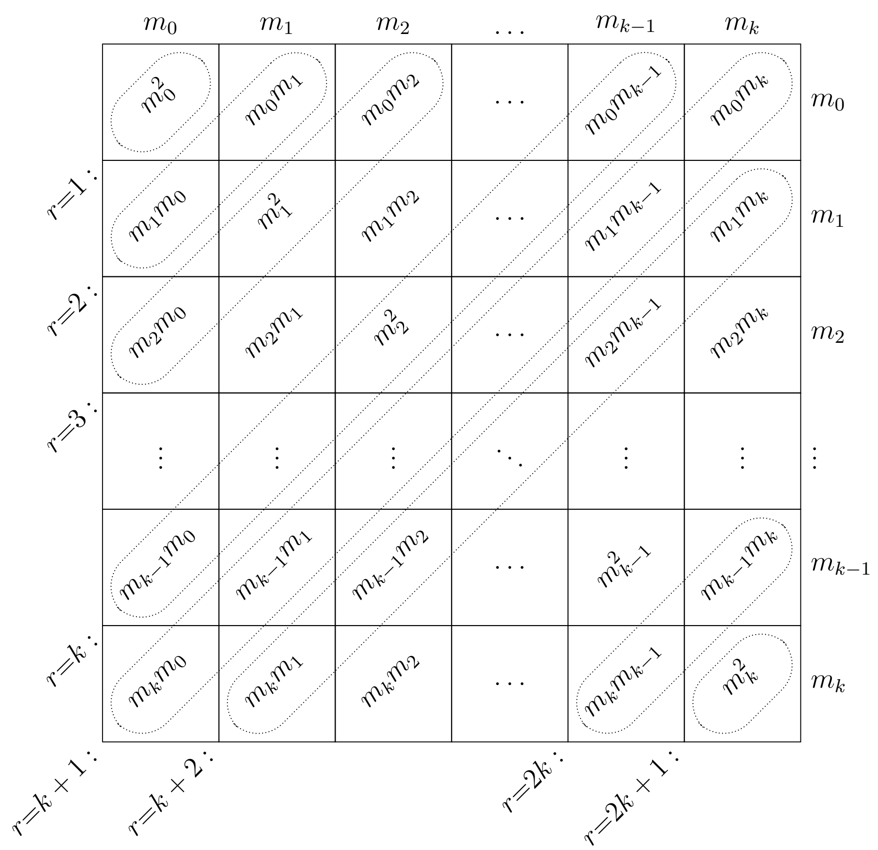

Figure 5.2: Proof of the Lemma 5.1

Let n denote the number of worlds in the model $\mathfrak{M}$ and $l = |K|$ denote the number of elements in subalgebra. Let $c_\vee$, $c_\wedge$ and $c_\rightarrow$ be respectively computational complexity of the operations of $\vee$, $\wedge$, $\rightarrow$ in algebra $\mathscr{H}$. These values may differ depending on the underlying algebra, but we will consider them as constants and omit them from determining complexity.

The complexity of the modal operators. We can now determine the complexity of the modal operators. The modal operator $\Diamond$ has complexity $O(n^2 c_\wedge +$

$n(n-1)c_\vee)$ which is $O(n^2(c_\wedge + c_\vee))$. Similarly, the modal operator $\Box$ has complexity $O(n^2 c_\rightarrow + n(n-1)c_\wedge)$ which is $O(n^2(c_\rightarrow + c_\wedge))$. Therefore, we will consider that the modal operators have complexity of $O(n^2)$.

The complexity of the binary logical operators. The binary logical connectives $\wedge$ and $\rightarrow$ have computational complexity $O(nc_\wedge)$ and $O(nc_\rightarrow)$. Therefore, we will consider that binary operators have complexity of $O(n)$.

The complexity of the comparison of formulae. Since the number of worlds of the model is n, to compare two formulae, n comparisons should be made. Therefore, the complexity of the comparison of one formula is $O(n)$.

The complexity of the formulae computations. Let us suppose that the reachable set $\mathcal{T} = \mathcal{T}_k$ is given and set $m_i = |T_i|$, for every $i \leqslant k$.

Also, let $|ULC|$ be the number of unary operators and $|BLC|$ the number of binary logical operations.

In the first iteration, we have $m_0|ULC|$ computations on formulae with unary operators and $m_0^2|BLC|$ computations on formulae with binary operators.

Further, in the second iteration, we have $m_1|ULC|$ computations on formulae with unary operators and $2m_0m_1|BLC|$ computations on formulae with binary operators. Hence, we can form the Table 5.1 showing the number of unary and binary operations by iteration steps.

iteration	number of the unary computations	number of the binary computations
1	$m_0\|ULC\|$	$m_0^2\|BLC\|$
2	$m_1\|ULC\|$	$2m_0m_1\|BLC\|$
3	$m_2\|ULC\|$	$(m_0m_2 + m_1^2 + m_2m_0)\|BLC\|$
$\vdots$	$\vdots$	$\vdots$
k	$m_{k-1}\|ULC\|$	$\sum_{i+j=k-1} m_im_j\|BLC\|$
$k+1$	$m_k\|ULC\|$	$\sum_{i+j=k} m_im_j\|BLC\|$
$k+2$	0	$\sum_{i+j=k+1} m_im_j\|BLC\|$
$\vdots$	$\vdots$	$\vdots$
$2k$	0	$2m_{k-1}m_k\|BLC\|$
$2k+1$	0	$m_k^2\|BLC\|$

Table 5.1: Number of unary and binary operations by iteration steps.

Since $|\mathcal{T}| \leqslant l^n$, in the worst case scenario, we have:

$$m_0 + m_1 + \ldots + m_k = l^n. \tag{5.5}$$

Now we have the maximal number of unary computations:

$$m_0|ULC| + m_1|ULC| + \ldots + m_k|ULC| = l^n|ULC|. \tag{5.6}$$

Using Lemma 5.1 and the Table 5.1, we have that the maximal number of binary computations is:

$$(m_0 + m_1 + \ldots + m_k)^2\,|BLC| = (l^n)^2\,|BLC| = l^{2n}|BLC|. \tag{5.7}$$

Now, according to the computational complexity of the unary and binary operators, we have the following complexity:

$$O(n^2l^n|ULC|) + O(nl^{2n}|BLC|) = O(nl^n(n|ULC| + l^n|BLC|)). \tag{5.8}$$

The complexity of the comparisons of formulae. When the algorithm computes the formulae, for the last formula, l^n comparisons have to be done. Still, for the formulae before the last one in $\mathcal{T}$, fewer comparisons have to be done. We create Table 5.2 shows the maximal number of the elements to compare with and a number of the comparisons by iteration steps.

iteration	maximal number of the elements to compare with	number of the comparisons
1	$m_0 + m_1$	$(m_0 + m_1)(m_0\lvert ULC\rvert + m_0^2\lvert BLC\rvert)$
2	$m_0 + m_1 + m_2$	$(m_0 + m_1 + m_2)(m_1\lvert ULC\rvert + 2m_0m_1\lvert BLC\rvert)$
$\vdots$	$\vdots$	$\vdots$
k	$\sum_{i=0}^{k} m_i$	$l^n(m_{k-1}\lvert ULC\rvert + \sum_{i+j=k-1} m_im_j\lvert BLC\rvert)$
$k+1$	$\sum_{i=0}^{k} m_i$	$l^n(m_k\lvert ULC\rvert + \sum_{i+j=k} m_im_j\lvert BLC\rvert)$
$k+2$	$\sum_{i=0}^{k} m_i$	$l^n(\sum_{i+j=k+1} m_im_j\lvert BLC\rvert)$
$\vdots$	$\vdots$	$\vdots$
$2k$	$\sum_{i=0}^{k} m_i$	$l^n(2m_{k-1}m_k\lvert BLC\rvert)$
$2k+1$	$\sum_{i=0}^{k} m_i$	$l^n(m_k^2\lvert BLC\rvert)$

Table 5.2: Maximal number of the elements to compare with and the number of the comparisons by iteration steps.

Hence, the sum in the last column is less or equal to

$$l^n(l^n|ULC| + l^{2n}|BLC|).$$

Therefore, computational complexity of the comparisons is:

$$O(nl^{2n}(|ULC| + l^n|BLC|)). \tag{5.9}$$

The complexity of the algorithm.

First, we used the notation of $|BLC|$ to represent the number of binary operators. Still, this value is very limited and in our case, $|BLC| = 2$. We take $|BLC|$ into consideration, because of the precision of the computations, and to leave the possibility of application on the structures where there are more binary operators (for example, $\vee$, $\leftrightarrow$, t-norms, s-norms, etc.). Regardless of all the above, we will still throw out the $|BLC|$ and consider that number of binary operations does not affect the complexity of computations.

According to the complexity of computations and complexity of comparisons we have:

$$\begin{aligned}
&O[n^2l^n|ULC| + nl^{2n}|BLC|] + O[nl^{2n}(|ULC| + l^n|BLC|)] \\
&\quad = O(n^2l^n|ULC| + nl^{2n} + nl^{2n}(|ULC| + l^n)) \\
&\quad = O(nl^n(n|ULC| + l^n + l^n(|ULC| + l^n))) \\
&\quad = O(nl^n(n|ULC| + l^n + l^n|ULC| + l^{2n})) \\
&\quad = O(nl^n(l^n|ULC| + l^{2n})) \\
&\quad = O(nl^{2n}(|ULC| + l^n))
\end{aligned} \tag{5.10}$$

In the fourth line, we use $n|ULC| \leqslant l^n|ULC|$ and $l^n \leqslant l^{2n}$.

Note also that the complexity of the algorithm can be limited and expressed as follows:

$$\begin{aligned} O(nl^{2n}(|ULC| + l^n)) &\leqslant O(nl^{2n}(|ULC| \cdot l^n)) \\ &= O(n|ULC|l^{3n}) \end{aligned}$$

Further, n is the number of worlds and since $|ULC|$ is related to the number of relations in the model and from that perspective, product $n|ULC|$ can be compared with the size of the model $\mathfrak{M}$ in the sense of the Definition 2.11. Hence, $O(n|ULC|) \leqslant O(|\mathfrak{M}|)$. Now, we have limitation from above:

$$O(n|ULC|l^{3n}) \leqslant O(|\mathfrak{M}|)l^{O(n)}, \tag{5.11}$$

since $O(ab^a) \leqslant O(a)b^{O(a)}$.

On the other hand, we can look at things this way. The number of unary operators $|ULC|$ is finite since the set of indices I is finite. Hence, $O(n|ULC|) = O(n)$. So, we have another estimate of the complexity of the algorithm:

$$O(n|ULC|l^{3n}) \leqslant O(n)l^{O(n)}. \tag{5.12}$$

However, even the first estimate of the complexity of the algorithm (5.10) is quite rough. In practice, the algorithm works significantly faster. In the Example 5.2, we can see that for model $\mathfrak{M}$ it follows that number of computational steps $k = 4$, number of elements in algebra is $l = |\{0, 0.1, 0.2, 0.6, 0.7, 0.8, 0.9, 1\}| = 8$, number of worlds in the model $\mathfrak{M}$ is $n = 3$. Still, $|\mathcal{T}| = m_0 + \ldots + m_4 = 68$ which is far less than $l^n = 8^3 = 512$.

Parameterized complexity

Some NP-hard problems of great computational complexity are much easier to solve in practice. Therefore, it is necessary to look deeper into the problems in order to more finely determine the complexity. In practice, it is very often the case that complexity depends to a much greater extent on one variable, while the others variables are limited, or may even be neglected. Hence, the idea of parameterized complexity is to perceive problems according to their inherent difficulty concerning multiple parameters of the input or output. We will give a few definitions and then explain in which ways the complexity of the algorithm can be considered. We will also compare the obtained complexity estimates with some other algorithms.

The following two definitions give some standard terminology. For more details, we refer to [38, 103, 140].

Definition 5.1. A *parameterized problem* is a language $L \subseteq \Sigma^* \times \Sigma^*$, where Σ is a fixed, finite alphabet. The second component of $(x, k) \in \Sigma^* \times \Sigma^*$ is called the *parameter* of the problem.

Usually, the parameter is a non-negative integer and then notation $L \subseteq \Sigma^* \times \mathbb{N}$ is used instead of $L \subseteq \Sigma^* \times \Sigma^*$.

Definition 5.2. A parameterized problem L is *fixed-parameter tractable* if there is an algorithm A to determine if instance (x, k) is in L in time bounded by $f(k) \cdot |x|^\alpha$, where $|x|$ is the size of the first component (x, k), α is a constant independent of x and k and $f : \Sigma^+ \to \mathbb{N}$ is an arbitrary computable function.

Several algorithms are known whose complexity is expressed similarly to (5.11). For example, in [81] Lichtenstein and Pnueli considered a decision-making algorithm for model checking problem for Linear Temporal Logic (LTL) and get the result that the algorithm terminates in time $O(|\mathscr{A}|)2^{O(|\varphi|)}$, where $|\varphi|$ is the length of the formula φ and $|\mathscr{A}|$ is the length of the structure $\mathscr{A}$ whose need to be checked. Emerson and Lei in [43] got the same result for Full Branching Time Logic (CTL*). In practice, it turned out that the length of the formula $|\varphi|$ is a small value, while $|\mathscr{A}|$ can be really huge and therefore, the algorithm terminates in polynomial time. Both of the algorithms are analyzed for several parameterizations such as *temporal depth* and *treewidth* and *pathwidth* in [82].

With that in mind, as well as the fact that we have many more parameters on our structures, we will now suggest some of them and describe how they can affect the complexity of the computation.

Properties of relations $\{R_i\}_{i\in I}$. If relations are Euclidean, reflexive, symmetric, transitive, etc., can certainly affect the computation in Kripke structures (cf. [114]). Also, the presence and arrangement of various elements from the structure can have an impact on computation.

Dominating value. If we analyze underlying algebra and values in model $\mathfrak{M}$ we can see that the values $0.7, 0.8, 0.9$ appear most often, and they are dominant through the reachable set $\mathcal{T}$. On the other hand, the values $0.1, 0.2, 0.6$ appear only once, and they are located in such places in the relation that they do not influence the computation of the set $\mathcal{T}$, except for they appear in T_0. For example, if the values $0.1, 0.2, 0.6$ can be replaced with 0, then the set $\mathcal{T}$ will still be the same, except for values $0.1, 0.2, 0.6$ which would be omitted.

For example, the dominating set of vertices in a graph is analyzed in [140].

The number of elements in the underlying subalgebra. This is closely related to the previous discussion. However, subalgebra can have many elements that appear in models and have no effect on computations, except for they appear in T_0.

The number of elements in the initial set. As we will see in the following examples, reducing truth constants from the initial set can increase the number of steps in the computations. Also, the number of propositional variables can have an impact.

Hence, all of these factors have an impact on the number of steps $2k + 1$ which is hard to predict. Nevertheless, we will see later that on other algebraic structures the number of reachable fuzzy sets can be maximal and that in those cases the complexity of the algorithm is appropriate.

5.3 Computation of weak simulations and weak bisimulations

We will now discuss which formulae affect the computation of weak simulation/bisimulation. We will show that the computation of simulations and bisimulations depends only on the formulae from the modal fragments, that is, from formulae that contain at least one of the modal operators.

Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two Kripke models.

According to Remark 4.2, fuzzy relations

$$\pi^{ws} = \bigwedge_{A\in\Psi} (V_A \backslash V'_A), \tag{5.13}$$

$$\pi^{wb} = \bigwedge_{A\in\Psi} [(V_A \backslash V'_A) \wedge (V_A / V'_A)] = \bigwedge_{A\in\Psi} (V_A \leftrightarrow V'_A), \tag{5.14}$$

are the greatest weak Ψ-simulation, where Ψ is some set of formulae. According to (2.8), i.e., $\Psi = \bigcup_{n=0}^{+\infty} F_n$, and we define

$$\pi_n^{ws} = \bigwedge_{A\in F_n} (V_A \backslash V'_A), \qquad \pi_n^{wb} = \bigwedge_{A\in F_n} (V_A \leftrightarrow V'_A), \tag{5.15}$$

for every $n \in \mathbb{N}_0$.

In this way, we can perform a computation based on the complexity of the formulae.

Remark 5.1. *Note that for computing π_0^{ws} and π_0^{wb} we have to consider only proposition variables PV, because $V_t(w) \to V'_t(w') = 1$ and $V_t(w) \leftrightarrow V'_t(w') = 1$ for every $t \in H$ and every $w \in W$, $w' \in W'$. Hence,*

$$\pi_0^{ws} = \bigwedge_{p\in PV} V_p \to V'_p, \tag{5.16}$$

$$\pi_0^{wb} = \bigwedge_{p\in PV} V_p \leftrightarrow V'_p, \tag{5.17}$$

the same as in (3.11) and (3.12), respectively.

Remark 5.2. *The sequence of fuzzy relations π_n is non-increasing. This is because, for every $A \in F_n$, we can find $B \in F_{n+1}$, such that $V_A = V_B$ (for example, B can be formula $A \wedge 1$).*

Theorem 5.2. *Let the formulae without modal operators be in the form of a union of sets F_n, i.e., $\Phi_{I,\mathscr{H}}^{PF} = \bigcup_{n=0}^{+\infty} F_n$. Then, the sequence of fuzzy relations $\{\pi_n^{wb}\}_{n\in\mathbb{N}_0}$, defined by*

$$\pi_n^{wb} = \bigwedge_{A\in F_n} (V_A \leftrightarrow V'_A), \tag{5.18}$$

is a constant sequence, i.e., for every $n \in \mathbb{N}_0$, it follows $\pi_n = \pi_{n+1}$.

Proof. This will be proved by induction.

Induction basis: $\pi_0 = \pi_1$;

Obviously, according to Remark 5.2, we have that $\pi_0 \geqslant \pi_1$.

Now, we will prove that $\pi_0 \leqslant \pi_1$. First, $\pi_1(w, w') = \bigwedge_{A\in F_1} V(w, A) \leftrightarrow V'(w', A)$, and we distinguish the following cases:

(a) if A is of the form $B \wedge C$, where $B, C \in F_0$, then

$$\begin{aligned} V(w, A) \leftrightarrow V'(w', A) &= V(w, B \wedge C) \leftrightarrow V'(w', B \wedge C) \\ &\geqslant (V(w, B) \leftrightarrow V'(w', B)) \wedge (V(w, C) \leftrightarrow V'(w', C)) \quad \text{by (1.70)} \end{aligned}$$

and it follows $\pi_1(w, w') \geqslant \pi_0(w, w')$.

(b) If A is of the form $B \to C$, the proof is practically the same, just property (1.72) has to be used.

Induction step:

First, we have that

$$\pi_0 = \pi_1 = \pi_2 = \ldots = \pi_n,$$

and let

$$\pi_{n+1} = \bigwedge_{A \in F_{n+1}} V_A \leftrightarrow V'_A.$$

(a) If A is of the form $B \wedge C$ where $B \in F_i$ and $C \in F_{n-i}$, for some i $(0 \leqslant i \leqslant n)$ then

$$\begin{aligned} V(w, A) \leftrightarrow V'(w', A) &= V(w, B \wedge C) \leftrightarrow V'(w', B \wedge C) \\ &\geqslant (V(w, B) \leftrightarrow V'(w', B)) \wedge (V(w, C) \leftrightarrow V'(w', C)) \quad \text{by (1.70)} \end{aligned}$$

and since $\pi_i = \pi_{n-i} = \pi_n$, it follows $\pi_{n+1}(w, w') \geqslant \pi_n(w, w')$.

(b) If A is of the form $B \to C$ the proof is similar like before. □

The previous proof is straightforward for other binary operators such as $\vee$ and $\leftrightarrow$ if they are considered. Also, if one is considered a unary operator $\neg$, the proof follows from the definition $\neg A \equiv A \to 0$.

Now, we have the following remark.

Remark 5.3. *Sequence $\{\pi_n\}_{n \in \mathbb{N}_0}$ is non-increasing when Ψ is some fragment with modal formulae.*

Theorem 5.3. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i \in I}, V')$ be two fuzzy Kripke models and $\mathfrak{M}''$ their disjoint union, and let φ be a fuzzy relation on $\mathfrak{M}''$ with the block representation*

$$\varphi = \begin{bmatrix} \varphi_{W \times W} & \varphi_{W \times W'} \\ \varphi_{W' \times W} & \varphi_{W' \times W'} \end{bmatrix}. \tag{5.19}$$

Then φ is the greatest weak Ψ-simulation/bisimulation on $\mathfrak{M}''$ for $\Psi \in \{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$ if and only if the following statements are true:

(a) *$\varphi_{W \times W}$ is the greatest weak Ψ-simulation/bisimulation on $\mathfrak{M}$;*

(b) *$\varphi_{W \times W'}$ is the greatest weak Ψ-presimulation/prebisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(c) *$\varphi_{W' \times W}$ is the greatest weak Ψ-presimulation/prebisimulation between $\mathfrak{M}'$ and $\mathfrak{M}$;*

(d) *$\varphi_{W' \times W'}$ is the greatest weak Ψ-simulation/bisimulation on $\mathfrak{M}'$.*

Proof. The proof is based on checking conditions (*ws-1*) and (*ws-2*) which is very similar to the proof of Theorem 3.7. □

Theorem 5.4. *Let $\mathfrak{M} = (W, \{R_i\}_{i \in I}, V)$ be a fuzzy Kripke model with the reachable fuzzy set $\mathcal{T}$ for the set $\Psi \in \{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$. Then, the greatest weak Ψ-simulations φ^{ws}, and Ψ-bisimulation φ^{wb} are:*

$$\varphi^{ws} = \bigwedge_{A \in \mathcal{T}} V_A \to V_A, \qquad \varphi^{wb} = \bigwedge_{A \in \mathcal{T}} V_A \leftrightarrow V_A.$$

Proof. This is an immediate consequence of the definition of the algorithm and the definition of weak Ψ-simulations and Ψ-bisimulations. $\square$

Now, based on the previous Theorem, we can compute weak $\Phi_{I,\mathscr{H}}$-simulations and $\Phi_{I,\mathscr{H}}$-bisimulations for models $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ from Example 5.3.

Therefore, for model $\mathfrak{M}$, we have:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \begin{bmatrix} 1 & 0.7 & 0.7 \\ 0.7 & 1 & 0.7 \\ 0.7 & 0.7 & 1 \end{bmatrix},$$

and for model $\mathfrak{M}'$, we have:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \end{bmatrix}.$$

We are now ready to formulate an algorithm for computation of the greatest weak (pre)simulation and (pre)bisimulation.

Algorithm 5.3 (Computation of the greatest weak Ψ-(pre)simulation and (pre)bisimulation)**.** *The input of this algorithm is two fuzzy Kripke models $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$. The algorithm computes the greatest weak Ψ-(pre)simulation/(pre)bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ in the following way:*

(A1) *In the first step, create the disjoint union of $\mathfrak{M}$ and $\mathfrak{M}'$, i.e., $\mathfrak{M}'' = \mathfrak{M} \sqcup \mathfrak{M}'$.*

(A2) *In the second step, using Algorithm 5.2, a reachable fuzzy set $\mathcal{T}$ for model $\mathfrak{M}''$ has been constructed.*

(A3) *Then, the greatest weak simulation and bisimulation for model $\mathfrak{M}''$ can be computed (Theorem 5.4):*

$$\pi^{ws} = \bigwedge_{A\in\mathcal{T}} V_A \to V_A, \qquad \pi^{wb} = \bigwedge_{A\in\mathcal{T}} V_A \leftrightarrow V_A.$$

(A4) *Then, $\varphi_*^\theta = \pi^\theta_{W\times W'}$ is the greatest weak fuzzy Ψ-(pre)simulation/(pre)bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$ (Theorem 5.3).*

(A5) *In the final step we check whether φ_*^θ satisfies (θ−1) for $\theta \in \{ws, wb\}$. If φ_*^θ satisfies (θ−1), then it is the greatest weak Ψ-presimulation/prebisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ of type θ. If φ_*^θ does not satisfy (θ−1), then there is no weak Ψ-simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$.*

The sets of formulae we consider $\Psi \in \{\Phi^{PF}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi_{I,\mathscr{H}}\}$ form a complete lattice (see Figure 2.1), and also the corresponding prebisimulations form the complete lattice shown in Figure 5.3 which is "upside-down" from the one in Figure 2.1.

5.4 Computational examples

This section gives examples that demonstrate the application of the Algorithm for reachable fuzzy sets 5.2 and the Algorithm for computation of the greatest weak Ψ-(pre)simulation and (pre)bisimulation 5.3. In the following examples, the set of binary operators is $BLC = \{\wedge, \to\}$, unless otherwise stated.

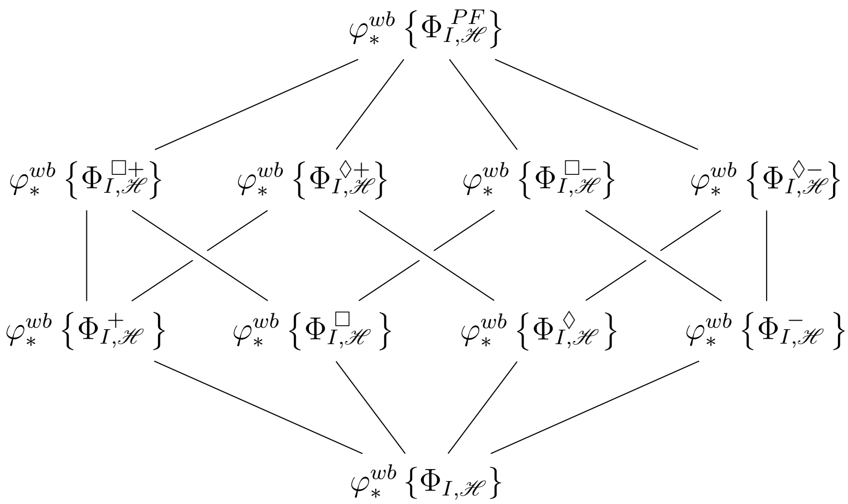

Figure 5.3: Lattice of weak prebisimulations

Example 5.4. Let us recall models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 5.3. First, we compute strong simulations and bisimulations because we will need them to compare with the weak ones.

Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \begin{bmatrix} 0.8 & 0.7 \\ 0.8 & 0.8 \\ 0.8 & 0.8 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.7 \\ 1 & 1 \\ 0.8 & 0.7 \end{bmatrix},$$
$$\varphi_*^{fb} = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix}, \qquad \varphi_*^{bb} = \varphi^{bb} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \\ 0.7 & 0.7 \end{bmatrix},$$
$$\varphi_*^{fbb} = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix}, \qquad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \\ 0.7 & 0.7 \end{bmatrix}, \qquad \varphi_*^{rb} = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix},$$

and φ_*^{fs}, φ_*^{fb}, φ_*^{fbb} and φ_*^{rb} do not satisfy (*fs-1*), (*fb-1*), (*fbb-1*) and (*rb-1*), respectively, which means that φ^{fs}, φ^{fb}, φ^{fbb} and φ^{rb} do not exist.

Then, we create a disjoint union $\mathfrak{M}''$ of models $\mathfrak{M}$ and $\mathfrak{M}'$ as in Example 2.6. After that, we compute reachable fuzzy sets for $\mathfrak{M}''$ for the set of formulae $\Phi_{I,\mathscr{H}}$. We will not list all elements of reachable fuzzy sets $\mathcal{T}$, but we will specify the cardinality of T_k sets. Hence, we have:

$$\begin{array}{ll} |T_0| = 10 & |T_5| = 220 \\ |T_1| = 19 & |T_6| = 202 \\ |T_2| = 48 & |T_7| = 136 \\ |T_3| = 119 & |T_8| = 68 \\ |T_4| = 192 & |T_9| = 14 \end{array}$$

Finally, $T_{10} = T_{11} = \ldots = T_{19} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_9$ and $|\mathcal{T}| = 10 + 19 + 48 + 119 + 192 + 220 + 202 + 136 + 68 + 14 = 1028$.

Of course, other sets of formulae have different reachable sets, but we omit all those details. Then, we compute weak simulations and bisimulations, and we have the following.

Using algorithms for computing weak Ψ-simulations and weak Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}\}$, we have:

$$\varphi^{ws}_{*} = \varphi^{wb}_{*} = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix},$$

and φ^{ws}_{*} and φ^{ws}_{*} do not satisfy (*ws-1*) and (*wb-1*), respectively. Therefore, there are no weak Ψ-simulations and weak Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{H}}, \Phi^{+}_{I,\mathscr{H}}, \Phi^{\Box}_{I,\mathscr{H}}, \Phi^{\Box+}_{I,\mathscr{H}}, \Phi^{\Diamond}_{I,\mathscr{H}}, \Phi^{\Diamond+}_{I,\mathscr{H}}\}$.

Using algorithms for computing weak Ψ-simulations and weak Ψ-bisimulations when $\Psi \in \{\Phi^{-}_{I,\mathscr{H}}, \Phi^{\Box-}_{I,\mathscr{H}}, \Phi^{\Diamond-}_{I,\mathscr{H}}, \Phi^{PF}_{I,\mathscr{H}}\}$, we have:

$$\varphi^{ws} = \varphi^{wb} = \begin{bmatrix} 1 & 0.7 \\ 0.7 & 1 \\ 0.7 & 0.7 \end{bmatrix}.$$

Note that the obtained results are in accordance with the Hennessy-Milner type theorems 4.5, 4.6 and 4.7.

The following example best explains the condition $\mathcal{T}_{2n+1} = \mathcal{T}_n$ in the Algorithm 5.2.

Example 5.5. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be fuzzy Kripke model over the Gödel structure $[0,1]$, where $W = \{w_1, w_2, w_3, w_4, w_5\}$, and set $I = \{1\}$. Fuzzy relation R_1, and fuzzy sets V_p and V_q, are represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5.20}$$

Model $\mathfrak{M}$ induces subalgebra $\langle K \rangle$ from the set of values $\{0,1\}$. Using Algorithm 5.2 we can determine reachable fuzzy sets for fuzzy Kripke model $\mathfrak{M}$ and the set of formulae $\Phi^{\Diamond}_{I,\mathscr{H}}$.

Hence, we have:

$$T_0 = \left\{ t_{0,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad t_{0,2} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad t_{0,3} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad t_{0,4} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \right\}$$

$$T_1 = \left\{ t_{1,1} = \neg t_{0,3} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad t_{1,2} = \neg t_{0,4} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad t_{1,3} = \Diamond t_{0,2} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}, \right.$$

$$t_{1,4} = \Diamond t_{0,4} = \begin{bmatrix}0\\0\\0\\1\\0\end{bmatrix} \Bigg\}$$

$$T_2 = \left\{ t_{2,1} = \Diamond t_{1,2} = \begin{bmatrix}0\\1\\1\\0\\0\end{bmatrix}, \quad t_{2,2} = \Diamond t_{1,4} = \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix}, \quad t_{2,3} = t_{1,2} \to t_{0,3} = \begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix} \right\}$$

$$T_3 = \Bigg\{ t_{3,1} = \neg t_{2,3} = \begin{bmatrix}0\\1\\0\\1\\1\end{bmatrix}, \quad t_{3,2} = \Diamond t_{2,1} = \begin{bmatrix}0\\0\\1\\1\\0\end{bmatrix}, \quad t_{3,3} = t_{1,1} \to t_{1,3} = \begin{bmatrix}1\\1\\1\\1\\0\end{bmatrix},$$

$$t_{3,4} = t_{1,1} \to t_{1,4} = \begin{bmatrix}1\\0\\0\\1\\0\end{bmatrix}, \quad t_{3,5} = t_{1,2} \wedge t_{1,3} = \begin{bmatrix}0\\1\\0\\1\\0\end{bmatrix}, \quad t_{3,6} = t_{1,3} \to t_{1,4} = \begin{bmatrix}1\\0\\0\\1\\1\end{bmatrix} \Bigg\}$$

$$T_4 = \Bigg\{ t_{4,1} = t_{1,1} \to t_{2,1} = \begin{bmatrix}1\\1\\1\\0\\0\end{bmatrix}, \quad t_{4,2} = t_{1,1} \to t_{2,2} = \begin{bmatrix}1\\1\\0\\0\\0\end{bmatrix}, \quad t_{4,3} = t_{1,3} \to t_{2,1} = \begin{bmatrix}1\\1\\1\\0\\1\end{bmatrix},$$

$$t_{4,4} = t_{1,3} \to t_{2,2} = \begin{bmatrix}1\\1\\0\\0\\1\end{bmatrix}, \quad t_{4,5} = t_{2,2} \to t_{1,4} = \begin{bmatrix}1\\0\\1\\1\\1\end{bmatrix} \Bigg\}$$

$$T_5 = \left\{ t_{5,1} = t_{1,1} \to t_{3,2} = \begin{bmatrix}1\\0\\1\\1\\0\end{bmatrix}, \quad t_{5,2} = t_{1,1} \to t_{3,5} = \begin{bmatrix}1\\1\\0\\1\\0\end{bmatrix}, \quad t_{5,3} = t_{1,1} \wedge t_{3,6} = \begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix} \right\}$$

$$T_6 = \left\{ t_{6,1} = t_{1,1} \wedge t_{4,3} = \begin{bmatrix}0\\1\\1\\0\\1\end{bmatrix}, \quad t_{6,2} = t_{1,1} \wedge t_{4,4} = \begin{bmatrix}0\\1\\0\\0\\1\end{bmatrix}, \quad t_{6,3} = t_{1,1} \wedge t_{4,5} = \begin{bmatrix}0\\0\\1\\1\\1\end{bmatrix} \right\}$$

$T_7 = \emptyset$

$$T_8 = \left\{ t_{8,1} = t_{3,6} \wedge t_{4,3} = \begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix} \right\} \qquad T_9 = \left\{ t_{9,1} = t_{4,3} \wedge t_{4,5} = \begin{bmatrix}1\\0\\1\\0\\1\end{bmatrix} \right\}$$

$$T_{10} = \left\{ t_{10,1} = t_{1,1} \wedge t_{8,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\} \quad T_{11} = \left\{ t_{11,1} = t_{1,1} \wedge t_{9,1} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \right\}$$

and finally $T_{12} = \ldots = T_{23} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_{11}$.

Note that set T_7 is empty, but that cannot be a criterion for termination of the algorithm because further execution creates non-empty T-sets. Therefore, we introduce the criterion $\mathcal{T}_{2n+1} = \mathcal{T}_n$ as the stopping criterion in the execution of the algorithm.

For the set of formulae $\Phi^{\Diamond}_{I,\mathscr{H}}$ weak simulation and bisimulation are both equal to the identity matrix.

The following example shows that the computation of weak simulations and bisimulations must not be stopped when the condition $\varphi_k = \varphi_{k+1}$ is met, where φ_k is the corresponding fuzzy matrix for the set $\mathcal{T}_k$.

Example 5.6. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ be fuzzy Kripke model over the Gödel structure $[0,1]$, where $W = \{u, v, w\}$, and set $I = \{1\}$. Fuzzy relation R_1, and fuzzy sets V_p and V_q, are represented by the following fuzzy matrix and column vectors:

$$R_1 = \begin{bmatrix} 0.7 & 1 & 0.2 \\ 0.5 & 0.8 & 1 \\ 1 & 0.3 & 0.8 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 0.6 \\ 0.5 \\ 0.1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 0.3 \\ 0.7 \\ 0.8 \end{bmatrix}. \tag{5.21}$$

Now, we will compute the greatest weak fuzzy $\Phi_{I,\mathscr{H}}$-(pre)simulation and $\Phi_{I,\mathscr{H}}$-(pre)bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$.

After computing reachable fuzzy sets $\mathcal{T} = \mathcal{T}_7$, for every $\mathcal{T}_k$ we have the corresponding fuzzy matrix:

$$\varphi_0^{ws} = \begin{bmatrix} 1 & 0.5 & 0.1 \\ 0.3 & 1 & 0.1 \\ 0.3 & 0.7 & 1 \end{bmatrix}, \quad \varphi_1^{ws} = \begin{bmatrix} 1 & 0.3 & 0.1 \\ 0.3 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}, \quad \varphi_2^{ws} = \begin{bmatrix} 1 & 0.3 & 0.1 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix},$$

$$\varphi_3^{ws} = \begin{bmatrix} 1 & 0.2 & 0.1 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix},$$

and for $4, \ldots, 7$ $\varphi_4^{ws} = \ldots = \varphi_7^{ws}$. Since φ_7^{ws} satisfies condition (*ws-1*), it follows that the greatest weak simulation is:

$$\varphi^{ws} = \begin{bmatrix} 1 & 0.2 & 0.1 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}.$$

For weak bisimulation, we have the following:

$$\varphi_0^{wb} = \begin{bmatrix} 1 & 0.3 & 0.1 \\ 0.3 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}, \quad \varphi_1^{wb} = \begin{bmatrix} 1 & 0.3 & 0.1 \\ 0.3 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}, \quad \varphi_2^{wb} = \begin{bmatrix} 1 & 0.2 & 0.1 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix},$$

and for $3,\dots,7$ $\varphi_3^{wb} = \dots = \varphi_7^{wb}$. Since φ_8^{wb} satisfies condition (*wb-1*), it follows that the greatest weak bisimulation is equal to the greatest weak simulation.

This example confirms the fact that we must take into consideration all reachable fuzzy sets and that we must not stop computation when the condition $\varphi_k^{wb} = \varphi_{k+1}^{wb}$ is satisfied.

Moreover, if we compute weak $\Phi_{I,\mathscr{H}}^{\Box+}$-bisimulation, then we have $\mathcal{T} = \mathcal{T}_8$ and:

$$\varphi_0^{wb} = \varphi_1^{wb} = \varphi_2^{wb} = \begin{bmatrix} 1 & 0.3 & 0.1 \\ 0.3 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}, \qquad \varphi_3^{wb} = \dots = \varphi_8^{wb} = \begin{bmatrix} 1 & 0.2 & 0.1 \\ 0.2 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{bmatrix}.$$

In the previous consideration, we performed computations over locally finite subalgebra $\langle K \rangle$ induced by the Kripke models. However, including new values in subalgebra $\langle K \rangle$ can have an impact on computational results because constants interact with modal operators and produce values that affect the computation. The following example illustrates the impact of including new values in the subalgebra.

Example 5.7. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two fuzzy Kripke models over the Gödel structure $[0,1]$, where $W = \{v, w\}$, $W' = \{v', w'\}$ and set $I = \{1\}$. Fuzzy relations R_1, R'_1 and fuzzy sets V_p, and V'_p are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{5.22}$$

$$R'_1 = \begin{bmatrix} 0.6 & 1 \\ 1 & 1 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \tag{5.23}$$

Hence, models $\mathfrak{M}$ and $\mathfrak{M}'$ induce subalgebra $\langle K \rangle = \langle \{0, 0.6, 1\} \rangle$.

Now, we will compute the greatest weak $\Phi_{I,\mathscr{H}}^{\Diamond}$-(pre)simulation φ_*^{ws} and weak fuzzy $\Phi_{I,\mathscr{H}}^{\Diamond}$-(pre)bisimulation φ_*^{wb} between models $\mathfrak{M}$ and $\mathfrak{M}'$.

First, we construct model $\mathfrak{M}'' = \mathfrak{M} \sqcup \mathfrak{M}'$:

$$R''_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0.6 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \qquad V''_p = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{5.24}$$

Then, we compute reachable fuzzy sets for $\mathfrak{M}''$ for the set of formulae $\Phi_{I,\mathscr{H}}^{\Diamond}$. We will not list all elements of reachable fuzzy sets $\mathcal{T}$, but we will specify the cardinality of T_k sets. Hence, we have:

$$\begin{array}{ll} |T_0| = 4 & |T_6| = 7 \\ |T_1| = 4 & |T_7| = 4 \\ |T_2| = 5 & |T_8| = 3 \\ |T_3| = 3 & |T_9| = 0 \\ |T_4| = 7 & |T_{10}| = 2 \\ |T_5| = 10 & \end{array}$$

Finally, $T_{11} = \ldots = T_{21} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_{10}$ and $|\mathcal{T}| = 4+4+5+3+7+10+7+4+3+0+2 = 49$.

Algorithms for computing weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-simulation and $\Phi^{\Diamond}_{I,\mathscr{H}}$-bisimulations for model $\mathfrak{M}''$ yield:

$$\varphi^{ws}_{*} = \varphi^{ws} = \begin{bmatrix} 1 & 0 & 0.6 & 0 \\ 0 & 1 & 0 & 0.6 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \varphi^{wb}_{*} = \varphi^{wb} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore, there is no weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-simulation and $\Phi^{\Diamond}_{I,\mathscr{H}}$-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$, while weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-presimulation is:

$$\varphi^{ws}_{*} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \end{bmatrix}.$$

Let's now add a new value 0.5 in the subalgebra to see what happens in that case. Hence, let subalgebra be $\langle K \rangle = \langle \{0, 0.5, 0.6, 1\} \rangle$. Again, we compute reachable fuzzy sets for $\mathfrak{M}''$ for the set of formulae $\Phi^{\Diamond}_{I,\mathscr{H}}$. We will not list all elements of reachable fuzzy sets $\mathcal{T}$, but we will specify the cardinality of T_k sets. Hence, we have:

$$\begin{array}{ll} |T_0| = 5 & |T_6| = 29 \\ |T_1| = 6 & |T_7| = 26 \\ |T_2| = 10 & |T_8| = 14 \\ |T_3| = 13 & |T_9| = 15 \\ |T_4| = 25 & |T_{10}| = 5 \\ |T_5| = 21 & \end{array}$$

Finally, $T_{11} = \ldots = T_{21} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_{10}$ and $|\mathcal{T}| = 5+6+10+13+25+21+29+26+14+15+5 = 169$.

Algorithms for computing weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-simulation and $\Phi^{\Diamond}_{I,\mathscr{H}}$-bisimulations for model $\mathfrak{M}''$ yield:

$$\varphi^{ws}_{*} = \varphi^{ws} = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \varphi^{wb}_{*} = \varphi^{wb} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Therefore, there is no weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-simulation and $\Phi^{\Diamond}_{I,\mathscr{H}}$-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$, while weak $\Phi^{\Diamond}_{I,\mathscr{H}}$-presimulation is:

$$\varphi^{ws}_{*} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}.$$

Hence, a new value in the subalgebra increases the number of reachable fuzzy sets and can have an impact on the computation of weak simulations and bisimulations. We can understand the new value as a new propositional variable that has the same values in all worlds.

Note that Algorithm 5.2 for reachable fuzzy sets also terminate in the case when $\mathcal{T}_0$ contains only $\{0, 1\}$ and proposition variables, i.e., $\mathcal{T}_0 = \{0, 1\} \cup PV$. The following Example illustrates this situation.

Example 5.8. Let us recall models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 5.3. We will determine reachable fuzzy sets for models $\mathfrak{M}$ and $\mathfrak{M}'$ when T_0 and T_0' contains only $\{0,1\}$ and PV.

First, we construct reachable fuzzy sets for model $\mathfrak{M}$:

$$T_0 = \left\{ t_{0,1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\quad t_{0,2} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},\quad t_{0,3} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.7 \end{bmatrix},\quad t_{0,4} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.7 \end{bmatrix} \right\}$$

$$T_1 = \left\{ t_{1,1} = \Box t_{0,3} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.7 \end{bmatrix},\quad t_{1,2} = \Box^- t_{0,3} = \begin{bmatrix} 1 \\ 1 \\ 0.7 \end{bmatrix},\quad t_{1,3} = \Box^- t_{0,4} = \begin{bmatrix} 1 \\ 0.7 \\ 0.7 \end{bmatrix}, \right.$$
$$t_{1,4} = \Diamond t_{0,2} = \begin{bmatrix} 0.9 \\ 1 \\ 0.9 \end{bmatrix},\quad t_{1,5} = \Diamond t_{0,3} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.7 \end{bmatrix},\quad t_{1,6} = \Diamond^- t_{0,2} = \begin{bmatrix} 0.8 \\ 0.8 \\ 1 \end{bmatrix},$$
$$\left. t_{1,7} = \Diamond^- t_{0,3} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.9 \end{bmatrix},\quad t_{1,8} = \Diamond^- t_{0,4} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.8 \end{bmatrix},\quad t_{1,9} = t_{0,3} \to t_{0,4} = \begin{bmatrix} 0.8 \\ 0.7 \\ 1 \end{bmatrix} \right\}$$

$$T_2 = \left\{ t_{2,1} = \Box t_{1,4} = \begin{bmatrix} 1 \\ 0.9 \\ 1 \end{bmatrix},\quad t_{2,2} = \Box t_{1,9} = \begin{bmatrix} 1 \\ 0.7 \\ 1 \end{bmatrix},\quad t_{2,3} = \Box^- t_{1,6} \begin{bmatrix} 1 \\ 1 \\ 0.8 \end{bmatrix}, \right.$$
$$t_{2,4} = \Diamond t_{1,4} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.9 \end{bmatrix},\quad t_{2,5} = \Diamond t_{1,8} = \begin{bmatrix} 0.8 \\ 0.8 \\ 0.8 \end{bmatrix},\quad t_{2,6} = \Diamond^- t_{1,3} = \begin{bmatrix} 0.8 \\ 0.7 \\ 0.9 \end{bmatrix},$$
$$t_{2,7} = t_{0,3} \to t_{1,1} = \begin{bmatrix} 0.7 \\ 0.7 \\ 1 \end{bmatrix},\quad t_{2,8} = t_{1,2} \to t_{0,3} = \begin{bmatrix} 0.9 \\ 0.8 \\ 1 \end{bmatrix},$$
$$t_{2,9} = t_{0,3} \wedge t_{1,3} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.7 \end{bmatrix},\quad t_{2,10} = t_{1,3} \to t_{0,3} = \begin{bmatrix} 0.9 \\ 1 \\ 1 \end{bmatrix},$$
$$t_{2,11} = t_{1,4} \to t_{0,3} = \begin{bmatrix} 1 \\ 0.8 \\ 0.7 \end{bmatrix},\quad t_{2,12} = t_{0,3} \to t_{1,5} = \begin{bmatrix} 0.8 \\ 1 \\ 1 \end{bmatrix},$$
$$\left. t_{2,13} = t_{0,4} \to t_{1,1} = \begin{bmatrix} 0.7 \\ 1 \\ 1 \end{bmatrix} \right\}$$

$$T_3 = \left\{ t_{3,1} = \Box^- t_{2,1} = \begin{bmatrix} 1 \\ 1 \\ 0.9 \end{bmatrix},\quad t_{3,2} = \Box^- t_{2,13} = \begin{bmatrix} 0.7 \\ 1 \\ 0.7 \end{bmatrix},\quad t_{3,3} = \Diamond^- t_{2,7} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.9 \end{bmatrix}, \right.$$
$$t_{3,4} = \Diamond^- t_{2,13} \begin{bmatrix} 0.7 \\ 0.8 \\ 1 \end{bmatrix},\quad t_{3,5} = t_{2,2} \to t_{0,3} = \begin{bmatrix} 0.9 \\ 1 \\ 0.7 \end{bmatrix},\quad t_{3,6} = t_{0,3} \wedge t_{2,13} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.7 \end{bmatrix},$$
$$\left. t_{3,7} = t_{2,2} \to t_{0,4} = \begin{bmatrix} 0.8 \\ 1 \\ 0.7 \end{bmatrix},\quad t_{3,8} = t_{1,6} \to t_{1,8} = \begin{bmatrix} 1 \\ 0.7 \\ 0.8 \end{bmatrix} \right\}$$

$$T_4 = \left\{ t_{4,1} = \Diamond^- t_{3,4} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.9 \end{bmatrix}, \quad t_{4,2} = \Diamond^- t_{3,6} = \begin{bmatrix} 0.7 \\ 0.8 \\ 0.8 \end{bmatrix}, \quad t_{4,3} = t_{3,2} \to t_{0,3} = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix}, \right.$$

$$t_{4,4} = t_{1,2} \wedge t_{2,1} = \begin{bmatrix} 1 \\ 0.9 \\ 0.7 \end{bmatrix}, \quad t_{4,5} = t_{1,2} \wedge t_{2,4} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.7 \end{bmatrix}, \quad t_{4,6} = t_{1,2} \to t_{2,4} = \begin{bmatrix} 0.9 \\ 0.9 \\ 1 \end{bmatrix},$$

$$t_{4,7} = t_{1,2} \to t_{2,9} = \begin{bmatrix} 0.9 \\ 0.7 \\ 1 \end{bmatrix}, \quad t_{4,8} = t_{1,4} \wedge t_{2,2} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.9 \end{bmatrix}, \quad t_{4,9} = t_{1,4} \wedge t_{2,3} = \begin{bmatrix} 0.9 \\ 1 \\ 0.8 \end{bmatrix},$$

$$t_{4,10} = t_{1,4} \wedge t_{2,8} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.9 \end{bmatrix}, \quad t_{4,11} = t_{1,4} \wedge t_{2,12} = \begin{bmatrix} 0.8 \\ 1 \\ 0.9 \end{bmatrix},$$

$$t_{4,12} = t_{1,4} \wedge t_{2,13} = \begin{bmatrix} 0.7 \\ 1 \\ 0.9 \end{bmatrix}, \quad t_{4,13} = t_{1,6} \to t_{2,6} = \begin{bmatrix} 1 \\ 0.7 \\ 0.9 \end{bmatrix},$$

$$t_{4,14} = t_{2,12} \to t_{1,7} = \begin{bmatrix} 1 \\ 0.8 \\ 0.9 \end{bmatrix}, \quad t_{4,15} = t_{2,2} \to t_{1,8} = \begin{bmatrix} 0.8 \\ 1 \\ 0.8 \end{bmatrix},$$

$$\left. t_{4,16} = t_{1,8} \wedge t_{2,7} = \begin{bmatrix} 0.7 \\ 0.7 \\ 0.8 \end{bmatrix} \right\}$$

$$T_5 = \left\{ t_{5,1} = \Box t_{4,1} = \begin{bmatrix} 0.7 \\ 0.9 \\ 1 \end{bmatrix}, \quad t_{5,2} = t_{1,4} \wedge t_{3,8} = \begin{bmatrix} 0.9 \\ 0.7 \\ 0.8 \end{bmatrix}, \quad t_{5,3} = t_{2,1} \wedge t_{2,3} = \begin{bmatrix} 1 \\ 0.9 \\ 0.8 \end{bmatrix}, \right.$$

$$t_{5,4} = t_{2,1} \wedge t_{2,12} = \begin{bmatrix} 0.8 \\ 0.9 \\ 1 \end{bmatrix}, \quad t_{5,5} = t_{2,3} \wedge t_{2,4} = \begin{bmatrix} 0.9 \\ 0.9 \\ 0.8 \end{bmatrix}, \quad t_{5,6} = t_{2,3} \wedge t_{2,8} = \begin{bmatrix} 0.9 \\ 0.8 \\ 0.8 \end{bmatrix},$$

$$t_{5,7} = t_{2,3} \wedge t_{2,13} = \begin{bmatrix} 0.7 \\ 1 \\ 0.8 \end{bmatrix}, \quad t_{5,8} = t_{2,10} \to t_{2,4} = \begin{bmatrix} 1 \\ 0.9 \\ 0.9 \end{bmatrix},$$

$$t_{5,9} = t_{2,4} \wedge t_{2,12} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.9 \end{bmatrix}, \quad t_{5,10} = t_{2,4} \wedge t_{2,13} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.9 \end{bmatrix},$$

$$\left. t_{5,11} = t_{2,12} \to t_{2,5} = \begin{bmatrix} 1 \\ 0.8 \\ 0.8 \end{bmatrix} \right\}$$

$$T_6 = \left\{ t_{6,1} = t_{2,1} \wedge t_{3,2} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.7 \end{bmatrix}, \quad t_{6,2} = t_{2,1} \wedge t_{3,7} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.7 \end{bmatrix} \right\}$$

$$T_7 = \left\{ t_{7,1} = t_{2,1} \wedge t_{4,15} = \begin{bmatrix} 0.8 \\ 0.9 \\ 0.8 \end{bmatrix} \right\} \qquad T_8 = \left\{ t_{8,1} = t_{2,1} \wedge t_{5,7} = \begin{bmatrix} 0.7 \\ 0.9 \\ 0.8 \end{bmatrix} \right\}$$

And, finally $T_9 = T_{10} = \ldots = T_{16} = T_{17} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_8$ and $|\mathcal{T}| = 4 + 9 + 13 + 8 + 16 + 11 + 2 + 1 + 1 = 65$. Comparing the results from Example 5.3, we can draw the following conclusions.

In Example 5.3, the number of reachable fuzzy sets was $|\mathcal{T}| = 10 + 15 + 18 + 19 + 6 = 68$, although we added six constants. In this example, the algorithm itself created constants 0.7, 0.8 and 0.9, although they were not in the initial set. Values 0.1, 0.2 and 0.6 do no affect the computation, and they represent the shortcoming of the three formulae. Therefore, omitting (adding) constants in the initial set T_0 can increase (decrease) the number of steps in the execution of the algorithm.

Now, we construct reachable fuzzy sets for model $\mathfrak{M}'$:

$$\begin{aligned}
T'_0 &= \left\{ t'_{0,1} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad t'_{0,2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad t'_{0,3} = \begin{bmatrix} 0.9 \\ 0.8 \end{bmatrix}, \quad t'_{0,4} = \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix} \right\} \\
T'_1 &= \left\{ t'_{1,1} = \Box t'_{0,4} = \begin{bmatrix} 1 \\ 0.7 \end{bmatrix}, \quad t'_{1,2} = \Diamond t'_{0,2} = \begin{bmatrix} 0.8 \\ 0.8 \end{bmatrix} \right\} \\
T'_2 &= \left\{ t'_{2,1} = t'_{0,3} \wedge t'_{1,1} = \begin{bmatrix} 0.9 \\ 0.7 \end{bmatrix}, \quad t'_{2,2} = t'_{1,1} \to t'_{0,3} = \begin{bmatrix} 0.9 \\ 1 \end{bmatrix}, \right. \\
&\qquad \left. t'_{2,3} = t'_{0,3} \to t'_{1,2} = \begin{bmatrix} 0.8 \\ 1 \end{bmatrix} \right\} \\
T'_3 &= \left\{ t'_{3,1} = t'_{2,2} \to t'_{0,3} = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix} \right\}
\end{aligned}$$

And, finally $T'_4 = T'_5 = T'_6 = T'_7 = \emptyset$. Hence, $\mathcal{T}' = \mathcal{T}'_3$ and $|\mathcal{T}'| = 4 + 2 + 3 + 1 = 10$. Note that in Example 5.3 $|\mathcal{T}'| = 20$.

The fact that we have computed the reachable fuzzy sets $\mathcal{T}$ and $\mathcal{T}'$ for the models $\mathfrak{M}$ and $\mathfrak{M}'$, respectively, is not enough to compute weak Ψ-(pre)simulation and Ψ-(pre)bisimulation. The following example shows why the creation of disjoint union model $\mathfrak{M}''$ cannot be avoided because it leads to the wrong result.

Example 5.9. Recall models $\mathfrak{M}$ and $\mathfrak{M}'$ from Example 5.3. We determined the sets $\mathcal{T}$ and $\mathcal{T}'$, but sets T_k and T'_k have different cardinality. So, we come to the first question, and that is how to meaningfully pair formulae to compute corresponding fuzzy relations. Also, the first model has T_0, T_1, T_2, T_3, T_4 sets, while the second has T'_0, T'_1, T'_2 sets. Hence, there is another question: with which formulae to pair the formulae from the sets T_3 and T_4?

When we created the sets T_0, T_1, T_2, T_3, T_4 and T'_0, T'_1, T'_2, we also recorded information about how the formulae were obtained, i.e., information about formulae parents as well as logical operators. Note that there is a $1-1$ correspondence between sets T_0 and T'_0 when they are computed over the same subalgebra. Therefore, for every formula from $\mathcal{T}$ we can create a corresponding formula in model $\mathfrak{M}'$ and for every formula from $\mathcal{T}'$ we can create a corresponding formula in model $\mathfrak{M}$.

If we use this method of pairing formulae and compute weak Ψ-presimulation for $\Psi = \Phi_{I,\mathscr{H}}$ or $\Psi = \Phi^+_{I,\mathscr{H}}$ we will get the results:

$$\varphi^{ws}_* = \varphi^{wb}_* = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.8 \\ 0.7 & 0.7 \end{bmatrix},$$

which differ from the results in Example 5.4:

$$\varphi_*^{ws} = \varphi_*^{wb} = \begin{bmatrix} 0.7 & 0.7 \\ 0.7 & 0.7 \\ 0.7 & 0.7 \end{bmatrix}.$$

Also, obtained results are inconsistent with Hennessy-Milner's theorems. Therefore, by pairing reachable formulae we can lose some fuzzy sets, which are valuable in the computation. In the end, we see that $|\mathcal{T}| = 68$, $|\mathcal{T}'| = 20$, while $|\mathcal{T}''| = 1028$.

Chapter 6

Some generalized results

"Truth is much too complicated to allow anything but approximations."

John von Neumann

In this chapter, we present generalized results from the previous chapters.

The chapter consists of three sections. Section 6.1 provides a short overview of the results concerning strong simulations and bisimulations, which are valid on residuated lattices. Most importantly, algorithms for computing strong simulations and bisimulations are valid on residuated lattices. Section 6.2 provides a generalization of algorithms for computing weak simulations and bisimulations which are valid on locally finite algebras that do not even have to contain a pair of adjoint operations. Section 6.3 provides some interesting computational examples.

6.1 Generalized results for simulations and bisimulations

As already mentioned, the terms concerning the fuzzy sets and the fuzzy relations from Section 1.7 are already defined on residuated lattices. We refer to classical textbooks in the field of fuzzy logic [5, 7]. Also, definitions and terms for fuzzy sets and relations over residuated lattices are used in various papers of Ćirić and his coworkers (see, [26, 28, 68, 70, 73], etc.).

Therefore, we generalize Kripke semantics from Section 2.2 to be defined over residuated lattice. In that case, Definition 2.5 becomes:

Definition 6.1. Let $\mathscr{L} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ be a complete residuated lattice and write $\overline{L} = \{\overline{t} \mid t \in L\}$ for the elements of $\mathscr{L}$ viewed as constants. Let I be some index set. Define the language $\Phi_{I,\mathscr{L}}$ via the grammar

$$A ::= \overline{t} \mid p \mid A \wedge A \mid A \otimes A \mid A \to A \mid \Box_i A \mid \Diamond_i A \mid \Box_i^- A \mid \Diamond_i^- A \tag{6.1}$$

where $\overline{t} \in \overline{L}$, $i \in I$ and p ranges over some set PV of proposition letters.

We omit all the details, just note that the truth assignment function V can be inductively extended to a function $V : W \times \Phi_{I,\mathscr{L}} \to L$ by:

(V1) $V(w, A \otimes B) = V(w, A) \otimes V(w, B)$;

(V2) $V(w, A \wedge B) = V(w, A) \wedge V(w, B)$;
(V3) $V(w, A \to B) = V(w, A) \to V(w, B)$;
(V4) $V(w, \Box_i A) = \bigwedge_{u \in W} R_i(w, u) \to V(u, A)$, for every $i \in I$;
(V5) $V(w, \Diamond_i A) = \bigvee_{u \in W} R_i(w, u) \otimes V(u, A)$, for every $i \in I$;
(V6) $V(w, \Box_i^- A) = \bigwedge_{u \in W} R_i(u, w) \to V(u, A)$, for every $i \in I$;
(V7) $V(w, \Diamond_i^- A) = \bigvee_{u \in W} R_i(u, w) \otimes V(u, A)$, for every $i \in I$.

Again, we omit lines over truth constants. The meaning is clear from the context, and therefore we will emphasize it only where necessary. We also define subsets of $\Phi_{I,\mathscr{L}}$ such as $\Phi^+_{I,\mathscr{L}}, \Phi^-_{I,\mathscr{L}}, \Phi^{\Diamond}_{I,\mathscr{L}}, \Phi^{\Diamond+}_{I,\mathscr{L}}, \Phi^{\Diamond-}_{I,\mathscr{L}}, \Phi^{\Box}_{I,\mathscr{L}}, \Phi^{\Box+}_{I,\mathscr{L}}, \Phi^{\Box-}_{I,\mathscr{L}}, \Phi^{PF}_{I,\mathscr{L}}$ with self-explanatory notation, analogous as in Section 2.3.

Definitions of simulations and bisimulations over residuated lattice are analogous as in Heyting algebra. Therefore, all claims concerning the characterization of simulations and bisimulations, as well as the claims for their computation, have their analogous versions over residuated lattice.

Therefore, we state only the following two theorems, and we omit the other statements. Hence, we first generalize Theorem 3.5.

Theorem 6.1. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be fuzzy Kripke models, let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, and let a sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ of fuzzy relations from $\mathscr{R}(W, W')$ be defined by*

$$\varphi_1 = \pi^\theta, \qquad \varphi_{k+1} = \varphi_k \wedge \phi^\theta(\varphi_k) \qquad \textit{for each } k \in \mathbb{N}. \tag{6.2}$$

If $\langle \mathrm{Im}(\pi^\theta) \cup \bigcup_{i\in I}(\mathrm{Im}(R_i) \cup \mathrm{Im}(R'_i))\rangle$ is a finite subalgebra of $\mathscr{L}$, then the following is true:

(a) *the sequence $\{\varphi_k\}_{k\in\mathbb{N}}$ is finite and descending, and there is the least natural number k such that $\varphi_k = \varphi_{k+1}$;*
(b) *if φ_k is non-empty, then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-2) and (θ-3), i.e., φ_k is the greatest presimulation/prebisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*
(c) *if φ_k is non-empty and satisfies (θ-1), then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., φ_k is the greatest simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*
(d) *if φ_k is empty or does not satisfy (θ-1), then there is not any fuzzy relation in $\mathscr{R}(W, W')$ satisfying (θ-1), (θ-2), and (θ-3), i.e., there is not any simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Generalization of the Algorithm 3.1 for testing the existence and computing the greatest simulations and bisimulation has the same formulation, so we omit it.

If the underlying residuated lattice $\mathscr{L}$ is locally finite, in the sense that each finitely generated subalgebra of $\mathscr{L}$ is finite, then the algorithm terminates in a finite number of steps, for arbitrary finite fuzzy Kripke models over $\mathscr{L}$. On the other hand, if $\mathscr{L}$ is not locally finite, then the algorithm terminates in a finite number of steps under conditions determined by Theorems 3.3 and 3.5.

However, regardless of the local finiteness of the underlying residuated lattice and the fulfillment of the conditions under which there exists the greatest simulation/bisimulation of a given type and the greatest simulation/bisimulation itself are characterized by the following theorem. Conditions under which there exists the greatest simulations and bisimulations is the same for fuzzy automata (cf. [26]).

If the underlying residuated lattice $\mathscr{L}$ satisfies condition (1.65) from Lemma 1.1, and

$$x \otimes \bigwedge_{i\in I} y_i = \bigwedge_{i\in I} (x \otimes y_i) \tag{6.3}$$

for all $x \in L$ and $\{y_i\}_{i\in I} \subseteq L$, then we have the following theorem.

Theorem 6.2. *Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R'_i\}_{i\in I}, V')$ be two finite fuzzy Kripke models, let $\theta \in \{fs, bs, fb, bb, fbb, bfb, rb\}$, let $\{\varphi_k\}_{k\in\mathbb{N}}$ be the sequence of fuzzy relations from $\mathscr{R}(W, W')$ defined by (3.27) (but over residuated lattice), and let*

$$\varphi = \bigwedge_{k\in\mathbb{N}} \varphi_k. \tag{6.4}$$

Then the following is true:

(a) *if φ is non-empty, then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-2) and (θ-3), i.e., it is the greatest presimulation/prebisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(b) *if φ is non-empty and satisfies (θ-1), then it is the greatest fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., it is the greatest simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$;*

(c) *if φ is empty or does not satisfy (θ-1), then there is not any fuzzy relation in $\mathscr{R}(W, W')$ which satisfies (θ-1), (θ-2) and (θ-3), i.e., there is not any simulation/bisimulation of type θ between $\mathfrak{M}$ and $\mathfrak{M}'$.*

Proof. Only the case $\theta = fs$ will be proved. All other cases can be proved similarly.

(a) For arbitrary $i \in I$, $w \in W$ and $w' \in W'$ we have that

$$\begin{aligned}
\left(\bigwedge_{k\in\mathbb{N}} (R'_i \circ \varphi_k^{-1})\right)(w', w) = \bigwedge_{k\in\mathbb{N}} (R'_i \circ \varphi_k^{-1})(w', w) &= \bigwedge_{k\in\mathbb{N}} \left(\bigvee_{u'\in W'} R'_i(w', u') \otimes \varphi_k^{-1}(u', w)\right) \\
&= \bigvee_{u'\in W'} \left(\bigwedge_{k\in\mathbb{N}} R'_i(w', u') \otimes \varphi_k^{-1}(u', w)\right) \qquad \text{(by (1.66))} \\
&= \bigvee_{u'\in W'} \left(R'_i(w', u') \otimes \left(\bigwedge_{k\in\mathbb{N}} \varphi_k^{-1}(u', w)\right)\right) \qquad \text{(by (6.3))} \\
&= \bigvee_{u'\in W'} \left(R'_i(w', u') \otimes \varphi^{-1}(u', w)\right) = (R'_i \circ \varphi^{-1})(w', w),
\end{aligned}$$

which means that

$$\bigwedge_{k\in\mathbb{N}} R'_i \circ \varphi_k^{-1} = R'_i \circ \varphi^{-1},$$

for every $i \in I$. The use of condition (1.66) is justified by the facts that W' is finite, and that $\{\varphi_k^{-1}(u', w)\}_{k\in\mathbb{N}}$ is a non-increasing sequence, so $\{R_i'(w', u') \otimes \varphi_k^{-1}(u', w)\}_{k\in\mathbb{N}}$ is also a non-increasing sequence.

Now, for all $k \in \mathbb{N}$ we have that

$$\varphi \leqslant \varphi_{k+1} \leqslant \phi^{fs}(\varphi_k) = [(R_i' \circ \varphi_k^{-1})/R_i]^{-1},$$

which is equivalent to

$$\varphi^{-1} \circ R_i \leqslant R_i' \circ \varphi_k^{-1}.$$

As the last inequation holds for every $k \in \mathbb{N}$ we have that

$$\varphi^{-1} \circ R_i \leqslant \bigwedge_{k\in\mathbb{N}} R_i' \circ \varphi_k^{-1} = R_i' \circ \varphi^{-1},$$

for every $i \in I$. Therefore, φ satisfies (*fs-2*). Moreover, $\varphi \leqslant \varphi_1 = \pi^{fs}$, so φ also satisfies (*fs-3*).

Next, let $\alpha \in \mathscr{R}(W, W')$ be an arbitrary fuzzy relation satisfying (*fs-2*) and (*fs-3*). According to Theorem 3.2, $\alpha \leqslant \phi^{fs}(\alpha)$ and $\alpha \leqslant \pi^{fs} = \varphi_1$. By induction, we can easily prove that $\alpha \leqslant \varphi_k$ for every $k \in \mathbb{N}$, therefore, $\alpha \leqslant \varphi$. This means that φ is the greatest fuzzy relation $\mathscr{R}(W, W')$ which satisfies (*fs-2*) and (*fs-3*).

The proof of assertion (b), (c) and (d) is analogous as in Theorem 6.2. □

6.2 Generalized results for weak simulations and bisimulations

Definitions of weak simulations and bisimulations are also analogous as in Heyting algebra. However, they do not retain some beautiful properties as linearly ordered Heyting algebra. In particular, Hennessy-Milner type theorems are not valid anymore.

However, Algorithm 5.2 for reachable fuzzy sets and Algorithm 5.3 for computation of weak simulations and bisimulations can also be applied over any locally finite residuated lattice. The underlying structure doesn't even have to be linearly ordered. Therefore, algorithms cannot be applied in Goguen (product) structure (1.15) since it is not locally finite. Also, note that Łukasiewicz algebra is not always locally finite. The following Remark clarifies this.

Remark 6.1. *Lukasiewicz algebra* $\boldsymbol{Ł}$ *(or MV-algebra) is locally finite iff every non-zero element* $x \in \boldsymbol{Ł}$ *there exist the least positive integer* n *such that*

$$nx = \underbrace{x \oplus \cdots \oplus x}_{n-times} = 1.$$

Also, every locally finite MV-algebra is linearly ordered (cf. [21]) and isomorphic to a subalgebra of the real unit interval $[0, 1]$ *(cf. [22]).*

For example, let us consider equidistant subchains of rational numbers in the real unit interval. Therefore, we have

$$\boldsymbol{Ł}_m = \left\{0, \frac{1}{m-1}, \frac{2}{m-2}, \ldots, \frac{m-2}{m-1}, 1\right\}$$

for some positive integer m where lattice operations $\wedge$ and $\vee$ coincide with the operation of minimum and maximum, respectively. In that case, $\boldsymbol{Ł}$ is called an equidistant Lukasiewicz chain. In particular, we have:

$$Ł_3 = \left\{0, \frac{1}{2}, 1\right\},$$
$$Ł_4 = \left\{0, \frac{1}{3}, \frac{2}{3}, 1\right\},$$
$$Ł_5 = \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}.$$

In Lukasiewicz chain, Lukasiewicz operations are special cases of (1.14), and they are given by:

$$\frac{i}{m} \otimes \frac{j}{m} = \max\left(0, \frac{i}{m} + \frac{j}{m} - 1\right), \tag{6.5}$$
$$\frac{i}{m} \to \frac{j}{m} = \min\left(1, 1 - \frac{i}{m} + \frac{j}{m}\right). \tag{6.6}$$

The following remark will give us a clearer picture of the number of elements in the subalgebra. Some elements are created by applying logical operations to already existing ones.

Remark 6.2. *Let consider algebra $\langle K \rangle = \langle \{0, 0.1, 1\} \rangle$ over Łukasiewicz structure. Note that Lukasiewicz operations produce values $\{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Therefore, the cardinality of the algebra $\langle K \rangle$ over Lukasiewicz structure is 11.*

On the other hand, if we consider the same algebra over the Nilpotent Minimum structure, the cardinality of the algebra $\langle K \rangle$ is 4, because only the element 0.9 can be obtained as a negation of element 0.1.

Various pairs of t-norms and fuzzy implications are often used in fuzzy logic applications. So far, we have only mentioned residual implications (R-implications) generated by left-continuous t-norms. However, in addition to R-implications, there are other types of fuzzy implications, such as strong implications (S-implications), quantum logic implications (QL-implications), and reciprocal implications with respect to the negation of QL-implications (NQL-implications) (cf. [87, 139]). For more information and basic properties of t-norms, we refer to [52, 76], while for more information about fuzzy implications, we refer to [3].

Also, Algorithm 5.2 for reachable fuzzy sets and Algorithm 5.3 for computation of weak simulations and bisimulations can be applied over any algebraic structure which is locally finite. For example, we can enrich an ordered lattice with a t-norm and fuzzy implication as long as the underlying structure remains locally finite.

Below are some operations to which algorithms could be applied. Table 6.1 shows some t-norms, while Table 6.2 shows some fuzzy implications.

One of the most commonly used S-implication is the Kleene-Dienes implication. For example, KD implications is applied in the rough set theory (cf. [32, 40, 41, 117]), fuzzy description logic (cf. [15, 137]), etc. A major flaw in this type of implication is that it does not preserve order, which is why it is difficult to interpret it in logical system. On the other hand, residuated implications behave excellently with respect to the order, and there they are at a great advantage when it comes to interpretation

Name	Formula
minimum	$T_{\mathbf{M}}(x,y) = \min(x,y)$
Łukasiewicz	$T_{\mathbf{LK}}(x,y) = \max(x+y-1,0)$
drastic product	$T_{\mathbf{D}}(x,y) = \begin{cases} 0, & \text{if } x,y \in [0,1), \\ \min(x,y), & \text{otherwise.} \end{cases}$
nilpotent minimum	$T_{\mathbf{nM}}(x,y) = \begin{cases} \min(x,y), & \text{if } x+y>1, \\ 0, & \text{otherwise.} \end{cases}$

Table 6.1: T-norms

Name	Formula
Łukasiewicz	$I_{\mathbf{LK}}(x,y) = \min(1, 1-x+y)$
Gödel	$I_{\mathbf{GD}}(x,y) = \begin{cases} 1, & \text{if } x \leqslant y, \\ y, & \text{otherwise.} \end{cases}$
Kleene-Dienes	$I_{\mathbf{KD}}(x,y) = \max(1-x,y)$
Rescher	$I_{\mathbf{RS}}(x,y) = \begin{cases} 1, & \text{if } x \leqslant y, \\ 0, & \text{otherwise.} \end{cases}$
Weber	$I_{\mathbf{WB}}(x,y) = \begin{cases} 1, & \text{if } x < 1, \\ y, & \text{otherwise.} \end{cases}$
Fodor	$I_{\mathbf{FD}}(x,y) = \begin{cases} 1, & \text{if } x \leqslant y, \\ \max(1-x,y), & \text{otherwise.} \end{cases}$
Zadeh	$I_{\mathbf{ZD}}(x,y) = \max(1-x, \min(x,y))$
largest S-impl.	$I_{\mathbf{DP}}(x,y) = \begin{cases} y, & \text{if } x=1, \\ 1-x, & \text{if } y=0, \\ 1, & \text{otherwise.} \end{cases}$
Willmott	$I_{\mathbf{W}}(x,y) = \min(\max(1-x,y), \max(x,1-x), \max(y,1-y))$

Table 6.2: Fuzzy implications

in logical systems. Regardless of the mentioned lack of KD implication, our goal is to show various possible applications of the developed algorithms. That is why we will define below Kleene-Dienes Modal Logic, i.e., fuzzy multimodal logic over complete lattice endowed with minimum t-norm and KD implication.

Definition 6.2. Let $\mathscr{L}_{KD} = (L, \wedge, \vee, \otimes, \to, 0, 1)$ be a complete lattice where $\otimes$ and $\to$ are minimum t-norm and KD implication, respectively, and write $\overline{L} = \{\bar{t} \mid t \in L\}$ for the elements of $\mathscr{L}$ viewed as constants. Let I be some index set. Define the language $\Phi^{KD}_{I,\mathscr{L}}$ via the grammar

$$A ::= \bar{t} \mid p \mid A \wedge A \mid A \otimes A \mid A \to A \mid \Diamond_i A \mid \Diamond_i^- A \tag{6.7}$$

where $\bar{t} \in \overline{L}$, $i \in I$ and p ranges over some set PV of proposition letters.

The following well-known abbreviations will be used:

$\neg A \equiv A \to \overline{0}$ (*negation*),
$A \leftrightarrow B \equiv (A \to B) \wedge (B \to A)$ (*equivalence*),
$A \vee B \equiv \neg(\neg A \wedge \neg B)$ (*disjunction*),
$\Box A \equiv \neg \Diamond \neg A$ (*necessity operator*),

$\Box^- A \equiv \neg\Diamond^- \neg A$ (*inverse necessity operator*).

Note that modal operators are interdefinable, which is generally false in fuzzy modal logic. That's why we don't have such a wealth of fragments but only define the following subsets of $\Phi_{I,\mathscr{L}}^{KD}$. The set of those formulae from $\Phi_{I,\mathscr{L}}^{KD}$ that do not contain any of the modal operator $\Diamond_i^-$, $i \in I$, will be denoted by $\Phi_{I,\mathscr{L}}^{\Diamond^+ KD}$. Analogous, the set of those formulae from $\Phi_{I,\mathscr{L}}^{KD}$ that do not contain any of the modal operator $\Diamond$, $i \in I$, will be denoted by $\Phi_{I,\mathscr{L}}^{\Diamond^- KD}$. Finally, the set of those formulae from $\Phi_{I,\mathscr{L}}^{KD}$ that do not contain any of the modal operators $\Diamond_i$ and $\Diamond_i^-$, $i \in I$, will be denoted with $\Phi_{I,\mathscr{L}}^{PF\,KD}$, where PF denotes *propositional formulae*.

Remark 6.3. *Structure $\mathscr{L}_{KD}$ is not residuated and definitions of weak Ψ-pre(bi)simulations are not equivalent to the relations (4.3) and (4.5) from Remark 4.2. However, with slight modifications to the algorithms 5.2 and 5.3, we can compute the relations*

$$\varphi(w, w') = \bigwedge_{A\in\Psi} V_A(w) \to V'_A(w'), \quad \varphi(w, w') = \bigwedge_{A\in\Psi} V_A(w) \leftrightarrow V'_A(w'), \tag{6.8}$$

for any $w \in W$ and $w' \in W'$.

Figure 6.1 graphically shows Kleene-Dienes implication and bi-implication.

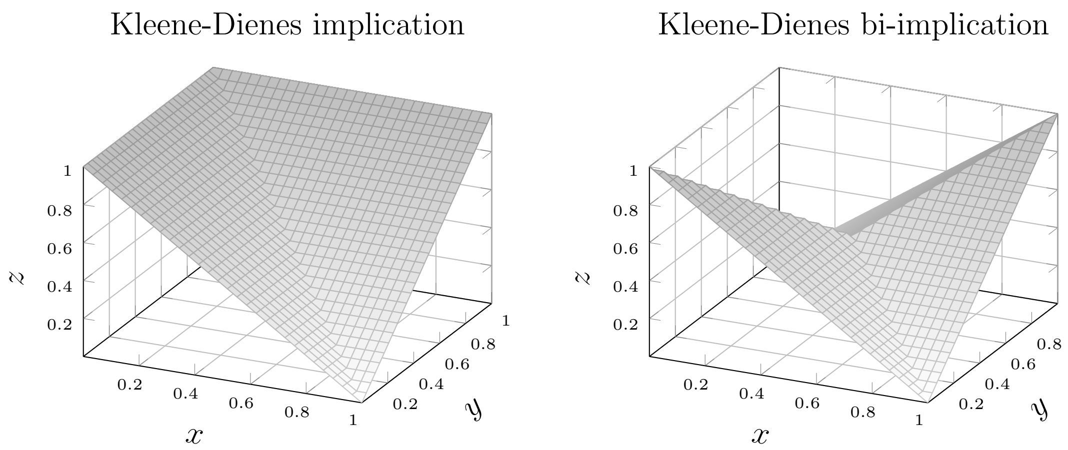

Figure 6.1: Kleene-Dienes implication and bi-implication

6.3 Computational examples

The following example demonstrates testing the existence and computation of strong simulation and bisimulation over the Łukasiewicz, Goguen and Nilpotent Minimum structure.

Example 6.1. Let us recall fuzzy Kripke models from Examples 2.3 and 3.1. Hence, fuzzy relations R_1, R'_1 and fuzzy sets V_p, V_q, V'_p and V'_q are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 1 & 0 & 0.9 \\ 1 & 0.3 & 0.6 \\ 1 & 0 & 1 \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ 0.4 \\ 1 \end{bmatrix}, \qquad V_q = \begin{bmatrix} 1 \\ 0.8 \\ 1 \end{bmatrix},$$

$$R'_1 = \begin{bmatrix} 1 & 0.4 \\ 1 & 0.4 \end{bmatrix}, \qquad V'_p = \begin{bmatrix} 1 \\ 0.4 \end{bmatrix}, \qquad V'_q = \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}.$$

Now, we will compute simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ over the following structures:

(a) Łukasiewicz structure;
(b) Goguen (product) structure;
(c) Nilpotent Minimum structure.

(a) Łukasiewicz structure. Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bb} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.6 \\ 0.2 & 0 \end{bmatrix},$$

$$\varphi_*^{fbb} = \begin{bmatrix} 0.5 & 0.4 \\ 0 & 0.5 \\ 0.4 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix},$$

and φ_*^{bb} and φ_*^{fbb}, do not satisfy (*bb-1*) and (*fbb-1*), respectively, what means that φ^{bb} and φ^{fbb} do not exist. In this example, φ_*^{rb} is an empty relation and therefore φ^{rb} also do not exists.

Algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho_*^{fs} = \varrho^{fs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}, \qquad \varrho_*^{bs} = \varrho^{bs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix},$$

while ϱ_*^{fb}, ϱ_*^{bb}, ϱ_*^{fbb}, ϱ_*^{bfb} and ϱ_*^{rb} are empty, so ϱ^{fb}, ϱ^{bb}, ϱ^{fbb}, ϱ^{bfb} and ϱ^{rb} do not exist. Therefore, there are not the greatest crisp fb-bisimulation, regardless of the fact that there is the greatest fuzzy bisimulations of these type.

(b) Goguen structure. Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{fbb} = \begin{bmatrix} 0.4 & 0.4 \\ 0 & 0 \\ 0.4 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix},$$

and φ_*^{fbb}, do not satisfy (*fbb-1*) which means that φ^{fbb} do not exists. In this example, φ_*^{bb} and φ_*^{rb} are an empty relations and therefore φ^{bb} and φ^{rb} do not exist.

To obtain φ_*^{fbb}, and to conclude that φ_*^{bb} and φ_*^{rb} are empty relations, we applied Theorem 6.2.

Algorithms for testing the existence and computing crisp simulations and bisimulations give the same result as in (a).

(c) Nilpotent Minimum structure. Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & 0.4 \\ 1 & 1 \\ 1 & 0.4 \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bb} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \\ 0.6 & 0 \end{bmatrix},$$

$$\varphi_*^{fbb} = \begin{bmatrix} 0.6 & 0.4 \\ 0 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}, \qquad \varphi_*^{bfb} = \varphi^{bfb} = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \\ 1 & 0.4 \end{bmatrix}, \qquad \varphi_*^{rb} = \begin{bmatrix} 0.6 & 0 \\ 0 & 0.6 \\ 0.6 & 0 \end{bmatrix},$$

and φ_*^{bb}, φ_*^{fbb} and φ_*^{rb}, do not satisfy (*bb-1*), (*fbb-1*) and (*rb-1*), respectively, which means that φ^{bb}, φ^{fbb} and φ^{rb} do not exist.

Algorithms for testing the existence and computing crisp simulations and bisimulations give the same result as in (a).

In the following example, we compute strong and weak bisimulations over the Łukasiewicz, Nilpotent Minimum and Gödel structure. Interestingly, the number of reachable fuzzy sets is maximal for Nilpotent Minimum and Łukasiewicz structures.

Example 6.2. Let $\mathfrak{M} = (W, \{R_i\}_{i\in I}, V)$ and $\mathfrak{M}' = (W', \{R_i'\}_{i\in I}, V')$ be two fuzzy Kripke models with values over the real unit interval $[0,1]$, where $W = \{u, v, w\}$, $W' = \{v', w'\}$ and set $I = \{1, 2\}$. Fuzzy relations R_1, R_2, R_1', R_2' and fuzzy sets V_p, and V_p' are represented by the following fuzzy matrices and column vectors:

$$R_1 = \begin{bmatrix} 1 & \frac{1}{3} & 1 \\ 0 & 1 & 0 \\ 1 & \frac{1}{3} & \frac{1}{3} \end{bmatrix}, \qquad R_2 = \begin{bmatrix} 1 & 0 & \frac{1}{3} \\ \frac{1}{3} & 1 & 1 \\ 1 & 0 & \frac{1}{3} \end{bmatrix}, \qquad V_p = \begin{bmatrix} 1 \\ \frac{1}{3} \\ 1 \end{bmatrix}, \tag{6.9}$$

$$R_1' = \begin{bmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{bmatrix}, \qquad R_2' = \begin{bmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{bmatrix}, \qquad V_p' = \begin{bmatrix} 1 \\ \frac{1}{3} \end{bmatrix}. \tag{6.10}$$

Now, we will compute strong and weak bisimulations between models $\mathfrak{M}$ and $\mathfrak{M}'$ over the following structures:

(a) Łukasiewicz structure;

(b) Nilpotent Minimum structure;

(c) Gödel structure.

Let first compute strong simulations and bisimulations. Interestingly, for all three structures we get the same results. Algorithms for testing the existence and computing simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ yield:

$$\varphi_*^{fs} = \varphi^{fs} = \begin{bmatrix} 1 & \frac{1}{3} \\ 1 & \frac{1}{3} \\ 1 & \frac{1}{3} \end{bmatrix}, \qquad \varphi_*^{bs} = \varphi^{bs} = \begin{bmatrix} 1 & \frac{1}{3} \\ 1 & 1 \\ 1 & \frac{1}{3} \end{bmatrix},$$

$$\varphi_*^{fb} = \varphi^{fb} = \begin{bmatrix} 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ 1 & \frac{1}{3} \end{bmatrix}, \qquad \varphi_*^{bb} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix},$$

$$\varphi_*^{fbb} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix}, \qquad \varphi_*^{bfb} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix}, \qquad \varphi_*^{rb} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix},$$

and $\varphi_*^{bb}, \varphi_*^{fbb}, \varphi_*^{bfb}$ and φ_*^{rb} do not satisfy (*bb-1*), (*fbb-1*), (*bfb-1*) and (*rb-1*), respectively, which means that $\varphi^{bb}, \varphi^{fbb}, \varphi^{bfb}$ and φ^{rb} do not exist.

Algorithms for testing the existence and computing crisp simulations and bisimulations yield:

$$\varrho_*^{fs} = \varrho^{fs} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \qquad \varrho_*^{bs} = \varrho^{bs} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix},$$

while ϱ_*^{fb}, ϱ_*^{bb}, ϱ_*^{fbb}, ϱ_*^{bfb} and ϱ_*^{rb} are empty, so ϱ^{fb}, ϱ^{bb}, ϱ^{fbb}, ϱ^{bfb} and ϱ^{rb} do not exist. Therefore, there are not the greatest crisp fb-bisimulation, regardless of the fact that there is the greatest fuzzy bisimulations of these type.

Now, we will compute weak simulations and bisimulations.

(a) Łukasiewicz structure. Fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ induce subalgebra $\langle K\rangle = \langle\{0, \frac{1}{3}, 1\}\rangle$. Note that Łukasiewicz operations produce value $\frac{2}{3}$, and therefore we get equidistant Łukasiewicz chain $\mathbf{Ł}_4$. Hence, the cardinality of the subalgebra $\langle K\rangle$ is 4.

Now, we will compute the greatest weak $\Phi_{I,\mathscr{L}}$-simulation and $\Phi_{I,\mathscr{L}}$-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$.

First, we construct model $\mathfrak{M}'' = \mathfrak{M} \sqcup \mathfrak{M}'$. Then, we compute reachable fuzzy sets for $\mathfrak{M}''$ for the set of formulae $\Phi_{I,\mathscr{L}}$. We will not list all elements of reachable fuzzy sets $\mathcal{T}$, but we will specify the cardinality of T_k sets. Hence, we have:

$$\begin{aligned} |T_0| &= 4 & \qquad |T_5| &= 247 \\ |T_1| &= 7 & |T_6| &= 268 \\ |T_2| &= 23 & |T_7| &= 169 \\ |T_3| &= 68 & |T_8| &= 64 \\ |T_4| &= 163 & |T_9| &= 11 \end{aligned}$$

Finally, $T_{10} = \ldots = T_{19} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_9$ and $|\mathcal{T}| = 4+7+23+68+163+247+268+169+64+11 = 1024$.

Algorithms for computing weak $\Phi_{I,\mathscr{L}}$-simulation and $\Phi_{I,\mathscr{L}}$-bisimulation for model $\mathfrak{M}''$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Therefore, there is no weak $\Phi_{I,\mathscr{L}}$-(pre)simulation and $\Phi_{I,\mathscr{L}}$-(pre)bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$, while weak $\Phi_{I,\mathscr{L}}$-simulation and weak $\Phi_{I,\mathscr{L}}$-bisimulation on both models are equal to the identity relation.

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \Phi^{\Box+}_{I,\mathscr{L}}, \Phi^{\Diamond+}_{I,\mathscr{L}}, \Phi^{+}_{I,\mathscr{L}}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

We omit other details here and below. Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \Phi^{\Box-}_{I,\mathscr{L}}, \Phi^{\Diamond-}_{I,\mathscr{L}}, \Phi^{-}_{I,\mathscr{L}}, \Phi^{\Box}_{I,\mathscr{L}}, \Phi^{\Diamond}_{I,\mathscr{L}}$ give the same results as in the case when $\Psi = \Phi_{I,\mathscr{L}}$. When $\Psi = \Phi^{PF}_{I,\mathscr{L}}$, we have:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

Let us note that Hennessy-Milner theorems 4.5, 4.6 and 4.7 do not hold for the Łukasiewicz structure. Second, we note that the number of reachable fuzzy sets is maximal, i.e., 1024. We have 4 possibilities in each of the 5 worlds of the disjoint model, i.e., $4^5 = 1024$.

(b) Nilpotent Minimum structure. Similarly as in the Łukasiewicz structure, models $\mathfrak{M}$ and $\mathfrak{M}'$ induce subalgebra $\langle K \rangle = \left\langle \left\{0, \frac{1}{3}, \frac{2}{3}, 1\right\} \right\rangle$ since operations in Nilpotent Minimum structures produce value $\frac{2}{3}$. Therefore, the cardinality of the subalgebra $\langle K \rangle$ is again 4.

Analogously to above, we will only specify the cardinality of T_k sets for the set of formulae $\Phi_{I,\mathscr{L}}$. Hence, we have:

$$\begin{array}{ll} |T_0| = 4 & |T_5| = 239 \\ |T_1| = 7 & |T_6| = 274 \\ |T_2| = 23 & |T_7| = 172 \\ |T_3| = 66 & |T_8| = 72 \\ |T_4| = 155 & |T_9| = 12 \end{array}$$

Finally, $T_{10} = \ldots = T_{19} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_9$ and $|\mathcal{T}| = 4 + 7 + 23 + 66 + 155 + 239 + 274 + 172 + 72 + 12 = 1024$.

For the set $\Phi_{I,\mathscr{L}}$, the result is the same as in the Łukasiewicz structure, i.e., there is no weak $\Phi_{I,\mathscr{L}}$-(pre)simulation and $\Phi_{I,\mathscr{L}}$-(pre)bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$, while weak $\Phi_{I,\mathscr{L}}$-simulation and weak $\Phi_{I,\mathscr{L}}$-bisimulation on both models are equal to the identity relation.

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi^{\Box+}_{I,\mathscr{L}}, \Phi^{\Diamond+}_{I,\mathscr{L}}, \Phi^{+}_{I,\mathscr{L}}\}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & \frac{2}{3} & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & \frac{2}{3} & 0 \\ \hline \frac{2}{3} & 0 & \frac{2}{3} & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

Therefore, there is no weak Ψ-simulation and Ψ-bisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$, while weak Ψ-presimulation and weak Ψ-prebisimulation are

$$\varphi_*^{ws} = \varphi_*^{wb} = \begin{bmatrix} \frac{2}{3} & 0 \\ 0 & 0 \\ \frac{2}{3} & 0 \end{bmatrix}.$$

Also, on both models Ψ-simulation and Ψ-bisimulation are equal to the identity relation.

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Box-}, \Phi_{I,\mathscr{L}}^{\Diamond-}, \Phi_{I,\mathscr{L}}^{-}\}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{2}{3} \\ 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{2}{3} & 0 & 0 & 1 \end{array}\right].$$

We omit other details here and below. Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Box}, \Phi_{I,\mathscr{L}}^{\Diamond}\}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

At the end, for weak $\Phi_{I,\mathscr{L}}^{PF}$-simulation and $\Phi_{I,\mathscr{L}}^{PF}$-bisimulation we have:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{array}\right].$$

(c) Gödel structure. Models $\mathfrak{M}$ and $\mathfrak{M}'$ induce subalgebra $\langle K\rangle = \left\langle \left\{0, \frac{1}{3}, 1\right\} \right\rangle$ and the cardinality of the subalgebra $\langle K\rangle$ is 3.

Also note that in this example we use the notation $\Phi_{I,\mathscr{L}}$ while in the previous chapter we used $\Phi_{I,\mathscr{H}}$.

Analogously to above, we will only specify the cardinality of T_k sets for the set of formulae $\Phi_{I,\mathscr{L}}$. Hence, we have:

$$\begin{array}{ll} |T_0| = 4 & |T_3| = 8 \\ |T_1| = 3 & |T_4| = 8 \\ |T_2| = 6 & |T_5| = 4 \end{array}$$

Finally, $T_6 = \ldots = T_{11} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_5$ and $|\mathcal{T}| = 4+3+6+8+8+4 = 33$.

Algorithms for computing weak $\Phi_{I,\mathscr{L}}$-simulation and $\Phi_{I,\mathscr{L}}$-bisimulation for model $\mathfrak{M}''$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} \\ \hline \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 \end{array}\right].$$

Therefore, weak $\Phi_{I,\mathscr{L}}$-presimulation and $\Phi_{I,\mathscr{L}}$-prebisimulation between models $\mathfrak{M}$ and $\mathfrak{M}'$ are:

$$\varphi_*^{ws} = \varphi_*^{wb} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} \end{bmatrix},$$

while there is no φ^{ws} and φ^{wb}, since φ_*^{ws} and φ_*^{wb} do not satisfy (*ws-1*) and (*wb-1*), respectively. The same result is obtained when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Box}, \Phi_{I,\mathscr{L}}^{\Diamond}\}$.

We omit other details here and below. Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Box+}, \Phi_{I,\mathscr{L}}^{\Diamond+}, \Phi_{I,\mathscr{L}}^{+}\}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \hline 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 \end{array}\right].$$

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Box-}, \Phi_{I,\mathscr{L}}^{\Diamond-}, \Phi_{I,\mathscr{L}}^{-}\}$ yield:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 \\ \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} \\ \hline \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 \end{array}\right].$$

At the end, for $\Phi_{I,\mathscr{L}}^{PF}$-simulation and $\Phi_{I,\mathscr{L}}^{PF}$-bisimulation, we have:

$$\varphi_*^{ws} = \varphi^{ws} = \varphi_*^{wb} = \varphi^{wb} = \left[\begin{array}{ccc|cc} 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 \\ 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \hline 1 & \frac{1}{3} & 1 & 1 & \frac{1}{3} \\ \frac{1}{3} & 1 & \frac{1}{3} & \frac{1}{3} & 1 \end{array}\right].$$

Remark 6.4. *In this example, the Łukasiewicz and NM algebra gave the maximum number of reachable sets.*

In this regard, the high complexity of the algorithm for reachable fuzzy sets (complexity is $O(|\mathfrak{M}|)l^{O(n)}$) certainly makes sense for the Łukasiewicz algebra. According to Remark 6.2, we can see that in Łukasiewicz algebra, values are created by applying the algorithm, which significantly increases the number of operations, the number of comparisons, and thus the complexity of the algorithm.

However, in the practical application, we see that applying the algorithm on Gödel algebra is much faster, and we should consider parameterized evaluation of complexity.

The following example shows application of the algorithms 5.2 and 5.3 over $\mathscr{L}_{KD}$.

Example 6.3. Let us recall fuzzy Kripke models from Examples 2.3 and 3.1. Now, we will demonstrate the application of the algorithm for reachable fuzzy sets and computation of weak simulations and bisimulations between fuzzy Kripke models $\mathfrak{M}$ and $\mathfrak{M}'$ over the Kleene-Dienes structure 6.7 from Definition 6.2. We again emphasize that we cannot compute weak simulation and bisimulation, but we can compute relations (6.8).

Models $\mathfrak{M}$ and $\mathfrak{M}'$ induce subalgebra $\langle K \rangle = \langle \{0, 0.3, 0.4, 0.6, 0.8, 0.9, 1\} \rangle$ since operations in the structure produce values $\{0.1, 0.2, 0.7\}$. Therefore, the cardinality of the subalgebra $\langle K \rangle$ is 10.

Analogously to above, we will only specify the cardinality of T_k sets for the set of formulae $\Phi_{I,\mathscr{L}}^{KD}$. In this case, we consider $ULC = \{\Diamond, \Diamond^-, \Box, \Box^-, \neg\}$, $BLC = \{\wedge, \otimes, \rightarrow, \vee, \leftrightarrow\}$ and we have:

$$\begin{array}{ll} |T_0| = 9 & |T_5| = 49 \\ |T_1| = 25 & |T_6| = 27 \\ |T_2| = 34 & |T_7| = 13 \\ |T_3| = 57 & |T_8| = 3 \\ |T_4| = 63 & \end{array}$$

Finally, $T_9 = \ldots = T_{17} = \emptyset$. Hence, $\mathcal{T} = \mathcal{T}_8$ and $|\mathcal{T}| = 9 + 25 + 34 + 57 + 63 + 49 + 27 + 13 + 3 = 280$.

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{KD}, \Phi_{I,\mathscr{L}}^{\Diamond - KD}\}$ yield that both of the relations from (6.8) on disjoint model $\mathfrak{M}''$ are equal:

$$\varphi = \left[\begin{array}{ccc|cc} 0.6 & 0.3 & 0.6 & 0.6 & 0.4 \\ 0.3 & 0.6 & 0.3 & 0.3 & 0.6 \\ 0.6 & 0.3 & 0.6 & 0.6 & 0.4 \\ \hline 0.6 & 0.3 & 0.6 & 0.6 & 0.4 \\ 0.4 & 0.6 & 0.4 & 0.4 & 0.6 \end{array}\right].$$

Algorithms for computing weak Ψ-simulations and Ψ-bisimulations when $\Psi \in \{\Phi_{I,\mathscr{L}}^{\Diamond + KD}, \Phi_{I,\mathscr{L}}^{PF\,KD}\}$ yield that both of the relations from (6.8) on disjoint model $\mathfrak{M}''$ are equal:

$$\varphi = \left[\begin{array}{ccc|cc} 0.6 & 0.4 & 0.6 & 0.6 & 0.4 \\ 0.4 & 0.6 & 0.4 & 0.4 & 0.6 \\ 0.6 & 0.4 & 0.6 & 0.6 & 0.4 \\ \hline 0.6 & 0.4 & 0.6 & 0.6 & 0.4 \\ 0.4 & 0.6 & 0.4 & 0.4 & 0.6 \end{array}\right].$$

Appendix A

Java codes

> "Science is what we understand well enough to explain to a computer; art is everything else."
>
> *Donald Ervin Knuth*

This appendix chapter presents codes that implement developed algorithms from previous chapters. For that purpose, the COB application was created which is acronym for Computation Of Bisimulations. Codes are written in Java programming language using Eclipse IDE for Enterprise Java Developers (Version: 2019-12 (4.14.0), Build id: 20191212-1212). In the following codes, we omit all methods which can be automatically generated by Eclipse IDE such as `equals()` and `hashCode()`. We also omit some less important parts of the code, such as the method `toString()`. The code is grouped into two packages:

(1) `com.logic.operations` I;

(2) `com.logic` II.

Package `com.logic.operations` consists of 3 interfaces:

(a) `Operation.java` A.1,

(b) `BinaryOperation.java` A.2,

(c) `UnaryOperation.java` A.3,

and 11 classes:

(i) `TNorm.java` A.4,

(ii) `Conjunction.java` A.5,

(iii) `LeftImplication.java` A.6,

(iv) `RightImplication.java` A.7,

(v) `Disjunction.java` A.8,

(vi) `Negation.java` A.9,

(vii) `BiImplication.java` A.10,

(viii) `Necessity.java` A.11,

(ix) `NecessityInv.java` A.12,

(x) `Possibility.java` A.13,

(xi) `PossibilityInv.java` A.14,

that implement those interfaces.

Package `com.logic` consists of 16 classes:

(i) `Algorithms.java` A.15,
(ii) `App.java` A.16,
(iii) `AppFrame.java` A.17,
(iv) `COBFrame.java` A.18,
(v) `Computator.java` A.19,
(vi) `CRLattice.java` A.20,
(vii) `CRLatticeGodel.java` A.21,
(viii) `CRLatticeLukasiewicz.java` A.22,
(ix) `CRLatticeNilMin.java` A.23,
(x) `CRLatticeProduct.java` A.24,
(xi) `FileParser.java` A.25,
(xii) `FSet.java` A.26,
(xiii) `FuzzyFormula.java` A.27,
(xiv) `FuzzyRelation.java` A.28,
(xv) `Model.java` A.30,
(xvi) `TSet.java` A.31.

Package com.logic.operations

Source Code A.1: Interface `Operation.java`

```
package com.logic.operations;

public interface Operation {

}
```

Source Code A.2: Interface `BinaryOperation.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;

public interface BinaryOperation extends Operation {
 FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2, CRLattice
     crLattice);
}
```

Source Code A.3: Interface `UnaryOperation.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;

public interface UnaryOperation extends Operation {
 FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice);
}
```

Source Code A.4: Class `TNorm.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzySet;

public class TNorm implements BinaryOperation {

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
     CRLattice crLattice) {
  FuzzySet fs = FuzzySet.strongConjunction(fuzzyFormula1.getFuzzySet(), fuzzyFormula2.
     getFuzzySet(), crLattice);
  return new FuzzyFormula(fs, this, fuzzyFormula1, fuzzyFormula2);
 }

}
```

Source Code A.5: Class `Conjunction.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzySet;

public class Conjunction implements BinaryOperation {

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
     CRLattice crLattice) {
  FuzzySet fs = FuzzySet.conjunction(fuzzyFormula1.getFuzzySet(), fuzzyFormula2.
     getFuzzySet());
  return new FuzzyFormula(fs, this, fuzzyFormula1, fuzzyFormula2);
 }

}
```

Source Code A.6: Class `LeftImplication.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzySet;

public class LeftImplication implements BinaryOperation {

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
     CRLattice crLattice) {
  FuzzySet fs = FuzzySet.leftImplication(fuzzyFormula1.getFuzzySet(), fuzzyFormula2.
     getFuzzySet(), crLattice);
  return new FuzzyFormula(fs,  this, fuzzyFormula1, fuzzyFormula2);
 }

}
```

Source Code A.7: Class `RightImplication.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzySet;

public class RightImplication implements BinaryOperation {

 @Override
```

```java
    public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
        CRLattice crLattice) {
      FuzzySet fs = FuzzySet.rightImplication(fuzzyFormula1.getFuzzySet(), fuzzyFormula2.
          getFuzzySet(), crLattice);
      return new FuzzyFormula(fs, this, fuzzyFormula1, fuzzyFormula2);
    }

}
```

Source Code A.8: Class `Disjunction.java`

```java
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;

public class Disjunction implements BinaryOperation {

    @Override
    public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
        CRLattice crLattice) {
      LeftImplication leftImplication = new LeftImplication();
      RightImplication rightImplication = new RightImplication();

      FuzzyFormula leftSide = leftImplication.apply(leftImplication.apply(fuzzyFormula1,
          fuzzyFormula2, crLattice), fuzzyFormula2, crLattice);
      FuzzyFormula rightSide = rightImplication.apply(fuzzyFormula1, rightImplication.apply(
          fuzzyFormula1, fuzzyFormula2, crLattice), crLattice);

      return new Conjunction().apply(leftSide, rightSide, crLattice);
    }

}
```

Source Code A.9: Class `Negation.java`

```java
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzySet;

public class Negation implements UnaryOperation {

    @Override
    public FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice) {
      double[] values = new double[fuzzyFormula.getFuzzySet().getNumberOfElements()];
      for (int i = 0; i < values.length; i++) {
        values[i] = crLattice.res(fuzzyFormula.getFuzzySet().getValue(i), 0);
      }
      return new FuzzyFormula(new FuzzySet(values), this, fuzzyFormula, null);
    }

}
```

Source Code A.10: Class `BiImplication.java`

```java
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;

public class BiImplication implements BinaryOperation {

    @Override
    public FuzzyFormula apply(FuzzyFormula fuzzyFormula1, FuzzyFormula fuzzyFormula2,
        CRLattice crLattice) {
      FuzzyFormula leftSide = new LeftImplication().apply(fuzzyFormula1, fuzzyFormula2,
          crLattice);
      FuzzyFormula rightSide = new RightImplication().apply(fuzzyFormula1, fuzzyFormula2,
          crLattice);
```

```java
    return new Conjunction().apply(leftSide, rightSide, crLattice);
  }

}
```

Source Code A.11: Class `Necessity.java`

```java
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzyRelation;
import com.logic.FuzzySet;

public class Necessity implements UnaryOperation {

 private FuzzyRelation fuzzyRelation;

 public Necessity(FuzzyRelation fuzzyRelation) {
  this.fuzzyRelation = fuzzyRelation;
 }

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice) {
  if (fuzzyRelation.getCols() != fuzzyFormula.getFuzzySet().getNumberOfElements())
   throw new IllegalArgumentException("Numbers of relation rows and set elements don't
       match");

  double res[] = new double[fuzzyRelation.getRows()];

  for (int i = 0; i < fuzzyRelation.getRows(); i++) {
   double min = Double.MAX_VALUE;
   for (int k = 0; k < fuzzyRelation.getCols(); k++) {
    min = Math.min(min, crLattice.res(fuzzyRelation.getValue(i, k), fuzzyFormula.
        getFuzzySet().getValue(k)));
   }
   res[i] = min;
  }

  return new FuzzyFormula(new FuzzySet(res), this, fuzzyFormula, null);
 }

}
```

Source Code A.12: Class `NecessityInv.java`

```java
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzyRelation;
import com.logic.FuzzySet;

public class NecessityInv implements UnaryOperation {

private FuzzyRelation fuzzyRelation;

 public NecessityInv(FuzzyRelation fuzzyRelation) {
  this.fuzzyRelation = fuzzyRelation;
 }

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice) {
  if (fuzzyRelation.getCols() != fuzzyFormula.getFuzzySet().getNumberOfElements())
   throw new IllegalArgumentException("Numbers of relation rows and set elements don't
       match");

  double res[] = new double[fuzzyRelation.getRows()];

  for (int i = 0; i < fuzzyRelation.getRows(); i++) {
   double min = Double.MAX_VALUE;
   for (int k = 0; k < fuzzyRelation.getCols(); k++) {
```

```
      min = Math.min(min, crLattice.res(fuzzyRelation.getValue(k, i), fuzzyFormula.
          getFuzzySet().getValue(k)));
    }
    res[i] = min;
   }

   return new FuzzyFormula(new FuzzySet(res), this, fuzzyFormula, null);
  }

}
```

Source Code A.13: Class `Possibility.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzyRelation;
import com.logic.FuzzySet;

public class Possibility implements UnaryOperation {

 private FuzzyRelation fuzzyRelation;

 public Possibility(FuzzyRelation fuzzyRelation) {
  this.fuzzyRelation = fuzzyRelation;
 }

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice) {
  if (fuzzyRelation.getCols() != fuzzyFormula.getFuzzySet().getNumberOfElements())
   throw new IllegalArgumentException("Numbers of relation rows and set elements don't
       match");

  double res[] = new double[fuzzyRelation.getRows()];

  for (int i = 0; i < fuzzyRelation.getRows(); i++) {
   double max = Double.MIN_VALUE;
   for (int k = 0; k < fuzzyRelation.getCols(); k++) {
    max = Math.max(max, crLattice.mult(fuzzyRelation.getValue(i, k), fuzzyFormula.
        getFuzzySet().getValue(k)));
   }
   res[i] = max;
  }

  return new FuzzyFormula(new FuzzySet(res), this, fuzzyFormula, null);
 }

}
```

Source Code A.14: Class `PossibilityInv.java`

```
package com.logic.operations;

import com.logic.CRLattice;
import com.logic.FuzzyFormula;
import com.logic.FuzzyRelation;
import com.logic.FuzzySet;

public class PossibilityInv implements UnaryOperation {

 private FuzzyRelation fuzzyRelation;

 public PossibilityInv(FuzzyRelation fuzzyRelation) {
  this.fuzzyRelation = fuzzyRelation;
 }

 @Override
 public FuzzyFormula apply(FuzzyFormula fuzzyFormula, CRLattice crLattice) {
  if (fuzzyRelation.getCols() != fuzzyFormula.getFuzzySet().getNumberOfElements())
   throw new IllegalArgumentException("Numbers of relation rows and set elements don't
       match");
```

```java

  double res[] = new double[fuzzyRelation.getRows()];

  for (int i = 0; i < fuzzyRelation.getRows(); i++) {
   double max = Double.MIN_VALUE;
   for (int k = 0; k < fuzzyRelation.getCols(); k++) {
    max = Math.max(max, crLattice.mult(fuzzyRelation.getValue(k, i), fuzzyFormula.getFuzzySet().getValue(k)));
   }
   res[i] = max;
  }

  return new FuzzyFormula(new FuzzySet(res), this, fuzzyFormula, null);
 }

}
```

Package com.logic

Class `Algorithms.java` contains algorithms for computing simulations and bisimulations. Table A.1 gives an overview of the algorithms and functions.

Name of the algorithm or function	Line of the code
`forwardSimulation`	10
`backwardSimulation`	62
`forwardBisimulation`	113
`backwardBisimulation`	167
`forwardBackwardBisimulation`	221
`backwardForwardBisimulation`	276
`regularBisimulation`	332
`crispForwardSimulation`	388
`crispBackwardSimulation`	439
`crispForwardBisimulation`	491
`crispBackwardBisimulation`	545
`crispForwardBackwardBisimulation`	600
`crispBackwardForwardBisimulation`	656
`crispRegularBisimulation`	713
`modelReduction`	770

Table A.1: Overview of algorithms and functions from the class `Algorithms.java`

Source Code A.15: Class `Algorithms.java`

```java
1  package com.logic;
2  
3  import java.util.ArrayList;
4  import java.util.List;
5  
6  import com.logic.Computator.Action;
7  
8  public class Algorithms {
9  
10  public static FuzzyRelation forwardSimulation(Model m1, Model m2, int maxIterations) {
11   System.out.println("Forward simulation:");
12  
13   FuzzyRelation oldFuzzyRelation = Model.piFs(m1, m2);
14   System.out.println("Iteration 0");
15   System.out.println(oldFuzzyRelation);
```

```
 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation, Model.
      phiForwardSimulation(m1, m2, oldFuzzyRelation));

  System.out.println("Iteration " + iteration);
  System.out.println(newFuzzyRelation);

  if (newFuzzyRelation.equals(oldFuzzyRelation)) {
   equals = true;
   break;
  }

  oldFuzzyRelation = newFuzzyRelation;
 }


 if (equals) {
  boolean found = true;
  for (int i = 0; i < m1.getFuzzySets().size(); i++) {
   FuzzySet fs1 = m1.getFuzzySets().get(i);
   FuzzySet fs2 = m2.getFuzzySets().get(i);
   boolean lessOrEqual = fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.
       getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }

  if (found) {
   System.out.println(
     String.format("Maximum forward simulation found at iteration %s", iteration));
   return oldFuzzyRelation;
  }
 }

 System.out.println("Maximum forward presimulation found:");
 System.out.println(oldFuzzyRelation);

 return null;

}


public static FuzzyRelation backwardSimulation(Model m1, Model m2, int maxIterations) {
 System.out.println("Backward simulation:");
 FuzzyRelation oldFuzzyRelation = Model.piBs(m1, m2);
 System.out.println("Iteration 0");
 System.out.println(oldFuzzyRelation);

 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation, Model.
      phiBackwardSimulation(m1, m2, oldFuzzyRelation));

  System.out.println("Iteration " + iteration);
  System.out.println(newFuzzyRelation);

  if (newFuzzyRelation.equals(oldFuzzyRelation)) {
   equals = true;
   break;
  }

  oldFuzzyRelation = newFuzzyRelation;
 }


```

```
 if (equals) {
  boolean found = true;
  for (int i = 0; i < m1.getFuzzySets().size(); i++) {
   FuzzySet fs1 = m1.getFuzzySets().get(i);
   FuzzySet fs2 = m2.getFuzzySets().get(i);
   boolean lessOrEqual = fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }

  if (found) {
   System.out.println(
     String.format("Maximum backward simulation found at iteration %s", iteration));
   return oldFuzzyRelation;
  }
 }

 System.out.println("Maximum backward presimulation found:");
 System.out.println(oldFuzzyRelation);

 return null;

}


public static FuzzyRelation forwardBisimulation(Model m1, Model m2, int maxIterations) {
 System.out.println("Forward bisimulation:");
 FuzzyRelation oldFuzzyRelation = Model.piFb(m1, m2);
 System.out.println("Iteration 0");
 System.out.println(oldFuzzyRelation);


 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
    Model.phiForwardBisimulation(m1, m2, oldFuzzyRelation));

  System.out.println("Iteration " + iteration);
  System.out.println(newFuzzyRelation);

  if (newFuzzyRelation.equals(oldFuzzyRelation)) {
   equals = true;
   break;
  }

  oldFuzzyRelation = newFuzzyRelation;
 }


 if (equals) {
  boolean found = true;
  for (int i = 0; i < m1.getFuzzySets().size(); i++) {
   FuzzySet fs1 = m1.getFuzzySets().get(i);
   FuzzySet fs2 = m2.getFuzzySets().get(i);
   boolean lessOrEqual = fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.getCrLattice()))
     && fs2.lessOrEqual(fs1.compose(oldFuzzyRelation, m1.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }

  if (found) {
   System.out.println(
     String.format("Maximum forward bisimulation found at iteration %s", iteration));
   return oldFuzzyRelation;
```

```
    }
  }

  System.out.println("Maximum forward preBisimulation found:");
  System.out.println(oldFuzzyRelation);

  return null;

}


public static FuzzyRelation backwardBisimulation(Model m1, Model m2, int maxIterations) {
  System.out.println("Backward bisimulation:");
  FuzzyRelation oldFuzzyRelation = Model.piBb(m1, m2);
  System.out.println("Iteration 0");
  System.out.println(oldFuzzyRelation);

  int iteration = 0;
  boolean equals = false;
  while (iteration < maxIterations) {
    iteration++;
    FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
        Model.phiBackwardBisimulation(m1, m2, oldFuzzyRelation));

    System.out.println("Iteration " + iteration);
    System.out.println(newFuzzyRelation);

    if (newFuzzyRelation.equals(oldFuzzyRelation)) {
      equals = true;
      break;
    }

    oldFuzzyRelation = newFuzzyRelation;
  }


  if (equals) {
    boolean found = true;
    for (int i = 0; i < m1.getFuzzySets().size(); i++) {
      FuzzySet fs1 = m1.getFuzzySets().get(i);
      FuzzySet fs2 = m2.getFuzzySets().get(i);
      boolean lessOrEqual =fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()))
          && fs2.lessOrEqual(oldFuzzyRelation.transpose().compose(fs1, m1.getCrLattice()));

      if (!lessOrEqual) {
        found = false;
        break;
      }
    }

    if (found) {
      System.out.println(
          String.format("Maximum backward bisimulation found at iteration %s", iteration));
      return oldFuzzyRelation;
    }
  }

  System.out.println("Maximum backward preBisimulation found:");
  System.out.println(oldFuzzyRelation);

  return null;

}


public static FuzzyRelation forwardBackwardBisimulation(Model m1, Model m2, int maxIterations) {
  System.out.println("Forward-backward bisimulation:");
  FuzzyRelation oldFuzzyRelation = Model.piFbb(m1, m2);
  System.out.println("Iteration 0");
  System.out.println(oldFuzzyRelation);
```

```
    int iteration = 0;
    boolean equals = false;
    while (iteration < maxIterations) {
     iteration++;
     FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
       Model.phiForwardBackwardBisimulation(m1, m2, oldFuzzyRelation));

     System.out.println("Iteration " + iteration);
     System.out.println(newFuzzyRelation);

     if (newFuzzyRelation.equals(oldFuzzyRelation)) {
      equals = true;
      break;
     }

     oldFuzzyRelation = newFuzzyRelation;
    }


    if (equals) {
     boolean found = true;
     for (int i = 0; i < m1.getFuzzySets().size(); i++) {
      FuzzySet fs1 = m1.getFuzzySets().get(i);
      FuzzySet fs2 = m2.getFuzzySets().get(i);
      boolean lessOrEqual =fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.
          getCrLattice()))
        && fs2.lessOrEqual(oldFuzzyRelation.transpose().compose(fs1, m1.getCrLattice()));


      if (!lessOrEqual) {
       found = false;
       break;
      }
     }

     if (found) {
      System.out.println(
        String.format("Maximum forward-backward bisimulation found at iteration %s",
            iteration));
      return oldFuzzyRelation;
     }
    }

    System.out.println("Maximum forward-backward preBisimulation found:");
    System.out.println(oldFuzzyRelation);

    return null;

   }


   public static FuzzyRelation backwardForwardBisimulation(Model m1, Model m2, int
       maxIterations) {
    System.out.println("Backward-forward bisimulation:");
    FuzzyRelation oldFuzzyRelation = Model.piBfb(m1, m2);
    System.out.println("Iteration 0");
    System.out.println(oldFuzzyRelation);


    int iteration = 0;
    boolean equals = false;
    while (iteration < maxIterations) {
     iteration++;
     FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
       Model.phiBackwardForwardBisimulation(m1, m2, oldFuzzyRelation));

     System.out.println("Iteration " + iteration);
     System.out.println(newFuzzyRelation);

     if (newFuzzyRelation.equals(oldFuzzyRelation)) {
      equals = true;
      break;
```

```
    }

    oldFuzzyRelation = newFuzzyRelation;
   }


   if (equals) {
    boolean found = true;
    for (int i = 0; i < m1.getFuzzySets().size(); i++) {
     FuzzySet fs1 = m1.getFuzzySets().get(i);
     FuzzySet fs2 = m2.getFuzzySets().get(i);
     boolean lessOrEqual = fs2.lessOrEqual(fs1.compose(oldFuzzyRelation, m1.getCrLattice()))
       && fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));


     if (!lessOrEqual) {
      found = false;
      break;
     }
    }

    if (found) {
     System.out.println(
       String.format("Maximum backward-forward bisimulation found at iteration %s", iteration));
     return oldFuzzyRelation;
    }
   }

   System.out.println("Maximum backward-forward preBisimulation found:");
   System.out.println(oldFuzzyRelation);

   return null;

  }


  public static FuzzyRelation regularBisimulation(Model m1, Model m2, int maxIterations) {
   System.out.println("Regular bisimulation:");
   FuzzyRelation oldFuzzyRelation = Model.piRb(m1, m2);
   System.out.println("Iteration 0");
   System.out.println(oldFuzzyRelation);


   int iteration = 0;
   boolean equals = false;
   while (iteration < maxIterations) {
    iteration++;
    FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
      Model.phiRegularBisimulation(m1, m2, oldFuzzyRelation));

    System.out.println("Iteration " + iteration);
    System.out.println(newFuzzyRelation);

    if (newFuzzyRelation.equals(oldFuzzyRelation)) {
     equals = true;
     break;
    }

    oldFuzzyRelation = newFuzzyRelation;
   }


   if (equals) {
    boolean found = true;
    for (int i = 0; i < m1.getFuzzySets().size(); i++) {
     FuzzySet fs1 = m1.getFuzzySets().get(i);
     FuzzySet fs2 = m2.getFuzzySets().get(i);
     boolean lessOrEqual = fs2.lessOrEqual(fs1.compose(oldFuzzyRelation, m1.getCrLattice()))
       && fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));
```

```java

        if (!lessOrEqual) {
          found = false;
          break;
        }
      }

      if (found) {
        System.out.println(
          String.format("Maximum regular bisimulation found at iteration %s", iteration));
        return oldFuzzyRelation;
      }
    }

    System.out.println("Maximum regular preBisimulation found:");
    System.out.println(oldFuzzyRelation);

    return null;

  }


  public static FuzzyRelation crispForwardSimulation(Model m1, Model m2, int
      maxIterations) {
    System.out.println("Crisp forward simulation:");
    FuzzyRelation oldFuzzyRelation = Model.piFs(m1, m2).crisp();
    System.out.println("Iteration 0");
    System.out.println(oldFuzzyRelation);

    int iteration = 0;
    boolean equals = false;
    while (iteration < maxIterations) {
      iteration++;
      FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
        Model.phiForwardSimulation(m1, m2, oldFuzzyRelation).crisp());

      System.out.println("Iteration " + iteration);
      System.out.println(newFuzzyRelation);

      if (newFuzzyRelation.equals(oldFuzzyRelation)) {
        equals = true;
        break;
      }

      oldFuzzyRelation = newFuzzyRelation;
    }

    if (equals) {
      boolean found = true;
      for (int i = 0; i < m1.getFuzzySets().size(); i++) {
        FuzzySet fs1 = m1.getFuzzySets().get(i);
        FuzzySet fs2 = m2.getFuzzySets().get(i);
        boolean lessOrEqual = fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.
            getCrLattice()));
        if (!lessOrEqual) {
          found = false;
          break;
        }
      }

      if (found) {
        System.out.println(
          String.format("Maximum crisp forward simulation found at iteration %s", iteration)
              );
        return oldFuzzyRelation;
      }
    }

    System.out.println("Maximum crisp forward presimulation found:");
    System.out.println(oldFuzzyRelation);

    return null;
```

```
}


public static FuzzyRelation crispBackwardSimulation(Model m1, Model m2, int maxIterations) {
 System.out.println("Crisp backward simulation:");
 FuzzyRelation oldFuzzyRelation = Model.piBs(m1, m2).crisp();
 System.out.println("Iteration 0");
 System.out.println(oldFuzzyRelation);


 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
    Model.phiBackwardSimulation(m1, m2, oldFuzzyRelation).crisp());

  System.out.println("Iteration " + iteration);
  System.out.println(newFuzzyRelation);

  if (newFuzzyRelation.equals(oldFuzzyRelation)) {
   equals = true;
   break;
  }

  oldFuzzyRelation = newFuzzyRelation;
 }


 if (equals) {
  boolean found = true;
  for (int i = 0; i < m1.getFuzzySets().size(); i++) {
   FuzzySet fs1 = m1.getFuzzySets().get(i);
   FuzzySet fs2 = m2.getFuzzySets().get(i);
   boolean lessOrEqual = fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }

  if (found) {
   System.out.println(
     String.format("Maximum crisp backward simulation found at iteration %s", iteration));
   return oldFuzzyRelation;
  }
 }

 System.out.println("Maximum crisp backward presimulation found:");
 System.out.println(oldFuzzyRelation);

 return null;

}

public static FuzzyRelation crispForwardBisimulation(Model m1, Model m2, int maxIterations) {
 System.out.println("Crisp forward bisimulation:");
 FuzzyRelation oldFuzzyRelation = Model.piFb(m1, m2).crisp();
 System.out.println("Iteration 0");
 System.out.println(oldFuzzyRelation);


 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
    Model.phiForwardBisimulation(m1, m2, oldFuzzyRelation)).crisp();
```

```java

   System.out.println("Iteration " + iteration);
   System.out.println(newFuzzyRelation);

   if (newFuzzyRelation.equals(oldFuzzyRelation)) {
    equals = true;
    break;
   }

   oldFuzzyRelation = newFuzzyRelation;
  }


  if (equals) {
   boolean found = true;
   for (int i = 0; i < m1.getFuzzySets().size(); i++) {
    FuzzySet fs1 = m1.getFuzzySets().get(i);
    FuzzySet fs2 = m2.getFuzzySets().get(i);
    boolean lessOrEqual =fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.getCrLattice()))
      && fs2.lessOrEqual(fs1.compose(oldFuzzyRelation, m1.getCrLattice()));
    if (!lessOrEqual) {
     found = false;
     break;
    }
   }

   if (found) {
    System.out.println(
      String.format("Maximum crisp forward bisimulation found at iteration %s", iteration));
    return oldFuzzyRelation;
   }
  }

  System.out.println("Maximum crisp forward preBisimulation found:");
  System.out.println(oldFuzzyRelation);

  return null;

 }


 public static FuzzyRelation crispBackwardBisimulation(Model m1, Model m2, int maxIterations) {
  System.out.println("Crisp backward bisimulation:");
  FuzzyRelation oldFuzzyRelation = Model.piBb(m1, m2).crisp();
  System.out.println("Iteration 0");
  System.out.println(oldFuzzyRelation);


  int iteration = 0;
  boolean equals = false;
  while (iteration < maxIterations) {
   iteration++;
   FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
     Model.phiBackwardBisimulation(m1, m2, oldFuzzyRelation)).crisp();

   System.out.println("Iteration " + iteration);
   System.out.println(newFuzzyRelation);

   if (newFuzzyRelation.equals(oldFuzzyRelation)) {
    equals = true;
    break;
   }

   oldFuzzyRelation = newFuzzyRelation;
  }


  if (equals) {
   boolean found = true;
   for (int i = 0; i < m1.getFuzzySets().size(); i++) {
```

```
    FuzzySet fs1 = m1.getFuzzySets().get(i);
    FuzzySet fs2 = m2.getFuzzySets().get(i);
    boolean lessOrEqual =fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()
        ))
      && fs2.lessOrEqual(oldFuzzyRelation.transpose().compose(fs1, m1.getCrLattice()));

    if (!lessOrEqual) {
     found = false;
     break;
    }
   }

   if (found) {
    System.out.println(
      String.format("Maximum crisp backward bisimulation found at iteration %s",
          iteration));
    return oldFuzzyRelation;
   }
  }

  System.out.println("Maximum crisp backward preBisimulation found:");
  System.out.println(oldFuzzyRelation);

  return null;

 }


 public static FuzzyRelation crispForwardBackwardBisimulation(Model m1, Model m2, int
     maxIterations) {
  System.out.println("Crisp forward-backward bisimulation:");
  FuzzyRelation oldFuzzyRelation = Model.piFbb(m1, m2).crisp();
  System.out.println("Iteration 0");
  System.out.println(oldFuzzyRelation);


  int iteration = 0;
  boolean equals = false;
  while (iteration < maxIterations) {
   iteration++;
   FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
     Model.phiForwardBackwardBisimulation(m1, m2, oldFuzzyRelation)).crisp();

   System.out.println("Iteration " + iteration);
   System.out.println(newFuzzyRelation);

   if (newFuzzyRelation.equals(oldFuzzyRelation)) {
    equals = true;
    break;
   }

   oldFuzzyRelation = newFuzzyRelation;
  }


  if (equals) {
   boolean found = true;
   for (int i = 0; i < m1.getFuzzySets().size(); i++) {
    FuzzySet fs1 = m1.getFuzzySets().get(i);
    FuzzySet fs2 = m2.getFuzzySets().get(i);
    boolean lessOrEqual =fs1.lessOrEqual(fs2.compose(oldFuzzyRelation.transpose(), m1.
        getCrLattice()))
      && fs2.lessOrEqual(oldFuzzyRelation.transpose().compose(fs1, m1.getCrLattice()));


    if (!lessOrEqual) {
     found = false;
     break;
    }
   }

   if (found) {
    System.out.println(
```

```
            String.format("Maximum crisp forward-backward bisimulation found at iteration %s",
                iteration));
        return oldFuzzyRelation;
      }
    }

    System.out.println("Maximum crisp forward-backward preBisimulation found:");
    System.out.println(oldFuzzyRelation);

    return null;

  }


  public static FuzzyRelation crispBackwardForwardBisimulation(Model m1, Model m2, int
      maxIterations) {
    System.out.println("Crisp backward forward bisimulation:");
    FuzzyRelation oldFuzzyRelation = Model.piBfb(m1, m2).crisp();
    System.out.println("Iteration 0");
    System.out.println(oldFuzzyRelation);


    int iteration = 0;
    boolean equals = false;
    while (iteration < maxIterations) {
      iteration++;
      FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
        Model.phiBackwardForwardBisimulation(m1, m2, oldFuzzyRelation)).crisp();

      System.out.println("Iteration " + iteration);
      System.out.println(newFuzzyRelation);

      if (newFuzzyRelation.equals(oldFuzzyRelation)) {
        equals = true;
        break;
      }

      oldFuzzyRelation = newFuzzyRelation;
    }


    if (equals) {
      boolean found = true;
      for (int i = 0; i < m1.getFuzzySets().size(); i++) {
        FuzzySet fs1 = m1.getFuzzySets().get(i);
        FuzzySet fs2 = m2.getFuzzySets().get(i);
        boolean lessOrEqual =fs2.lessOrEqual(fs1.compose(oldFuzzyRelation, m1.getCrLattice()
            ))
          && fs1.lessOrEqual(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));


        if (!lessOrEqual) {
          found = false;
          break;
        }
      }

      if (found) {
        System.out.println(
            String.format("Maximum crisp backward-forward bisimulation found at iteration %s",
                iteration));
        return oldFuzzyRelation;
      }
    }

    System.out.println("Maximum crisp backward-forward preBisimulation found:");
    System.out.println(oldFuzzyRelation);


    return null;

  }
```

```java

public static FuzzyRelation crispRegularBisimulation(Model m1, Model m2, int
    maxIterations) {
 System.out.println("Crisp regular bisimulation:");
 FuzzyRelation oldFuzzyRelation = Model.piRb(m1, m2).crisp();
 System.out.println("Iteration 0");
 System.out.println(oldFuzzyRelation);


 int iteration = 0;
 boolean equals = false;
 while (iteration < maxIterations) {
  iteration++;
  FuzzyRelation newFuzzyRelation = FuzzyRelation.conjunction(oldFuzzyRelation,
    Model.phiRegularBisimulation(m1, m2, oldFuzzyRelation)).crisp();

  System.out.println("Iteration " + iteration);
  System.out.println(newFuzzyRelation);

  if (newFuzzyRelation.equals(oldFuzzyRelation)) {
   equals = true;
   break;
  }

  oldFuzzyRelation = newFuzzyRelation;
 }


 if (equals) {
  boolean found = true;
  for (int i = 0; i < m1.getFuzzySets().size(); i++) {
   FuzzySet fs1 = m1.getFuzzySets().get(i);
   FuzzySet fs2 = m2.getFuzzySets().get(i);
   boolean equal =fs2.equals(fs1.compose(oldFuzzyRelation, m1.getCrLattice()))
     && fs1.equals(oldFuzzyRelation.compose(fs2, m1.getCrLattice()));


   if (!equal) {
    found = false;
    break;
   }
  }

  if (found) {
   System.out.println(
     String.format("Maximum crisp regular bisimulation found at iteration %s",
         iteration));
   return oldFuzzyRelation;
  }
 }

 System.out.println("Maximum crisp regular preBisimulation found:");
 System.out.println(oldFuzzyRelation);


 return null;

}


public static String modelReduction(Model model, FuzzyRelation fuzzyRelation, Action
    action) {
 Model currentModel = model;
 FuzzyRelation currentRelation = fuzzyRelation;
 Action lastAction = action;
 Model modelReduced = firstOrderReduction(model, fuzzyRelation);

 StringBuilder sb = new StringBuilder();
 sb.append(modelReduced);

 while (!modelReduced.equals(currentModel)) {
  currentModel = modelReduced;
  if (lastAction == Action.BackwardBisimulation) {
```

```
      currentRelation = Algorithms.forwardBisimulation(currentModel, currentModel, Computator.MAX_ITERATIONS);
      lastAction = Action.ForwardBisimulation;
    } else if (lastAction == Action.ForwardBisimulation) {
      currentRelation = Algorithms.backwardBisimulation(currentModel, currentModel, Computator.MAX_ITERATIONS);
      lastAction = Action.BackwardBisimulation;}
    else if (lastAction == Action.ForwardBackwardBisimulation) {
       currentRelation = Algorithms.forwardBackwardBisimulation(currentModel, currentModel, Computator.MAX_ITERATIONS);
       lastAction = Action.BackwardForwardBisimulation;}
    else if (lastAction == Action.BackwardForwardBisimulation) {
      currentRelation = Algorithms.backwardForwardBisimulation(currentModel, currentModel, Computator.MAX_ITERATIONS);
      lastAction = Action.ForwardBackwardBisimulation;}
    else {
      // It's regular and should stay regular
      currentRelation = Algorithms.regularBisimulation(currentModel, currentModel, Computator.MAX_ITERATIONS);
      lastAction = Action.RegularBisimulation;
    }
    modelReduced = firstOrderReduction(currentModel, currentRelation);
    sb.append(System.lineSeparator()).append(modelReduced);
  }

  return sb.toString();
}

private static Model firstOrderReduction(Model model, FuzzyRelation fuzzyRelation) {
  List<FuzzyRelation> newFuzzyRelations = new ArrayList<>();
  List<FuzzySet> newFuzzySets = new ArrayList<>();

  for (FuzzyRelation fr : model.getFuzzyRelations()) {
    FuzzyRelation newFuzzyRelation = fuzzyRelation.compose(fr, model.getCrLattice()).compose(fuzzyRelation, model.getCrLattice());
    newFuzzyRelations.add(newFuzzyRelation);
  }

  for (FuzzySet fs : model.getFuzzySets()) {
    //fuzzyRelation.compose(fs, model.getCrLattice());
    FuzzySet newFuzzySet = fs.compose(fuzzyRelation, model.getCrLattice());
    newFuzzySets.add(newFuzzySet);
  }

  while (true) {
    int[] equalRows = findEqualRows(newFuzzyRelations, newFuzzySets);
    if (equalRows != null && areColsEqual(newFuzzyRelations, equalRows[0], equalRows[1])) {
      int rowToReduce = equalRows[1];
      for (FuzzyRelation r : newFuzzyRelations) {
        r.reduce(rowToReduce);
      }
      for (FuzzySet s : newFuzzySets) {
        s.reduce(rowToReduce);
      }
    } else {
      break;
    }
  }

  return new Model(newFuzzyRelations, newFuzzySets, model.getCrLattice());
}


private static int[] findEqualRows(List<FuzzyRelation> fuzzyRelations, List<FuzzySet> fuzzySets) {
  int dim = fuzzyRelations.get(0).getRows();
  for (int i = 0; i < dim - 1; i++) {
    for (int j = i + 1; j < dim; j++) {

      boolean relationsRowsEqual = true;
      for (FuzzyRelation fr : fuzzyRelations) {
        if (!fr.areRowsEqual(i, j)) {
```

```java
      relationsRowsEqual = false;
      break;
     }
    }
    if (!relationsRowsEqual) {
     continue;
    }

    boolean setsRowsEqual = true;
    for (FuzzySet fs : fuzzySets) {
     if (fs.getValue(i) != fs.getValue(j)) {
      setsRowsEqual = false;
      break;
     }
    }

    if (setsRowsEqual) {
     return new int[] { i, j };
    }
   }
  }

  return null;
 }

 private static boolean areColsEqual(List<FuzzyRelation> fuzzyRelations, int col1, int col2) {
  for (FuzzyRelation r : fuzzyRelations) {
   if (!r.areColsEqual(col1, col2)) {
    return false;
   }
  }
  return true;
 }
}
```

Source Code A.16: Class App.java

```java
package com.logic;

import java.util.ArrayList;
import java.util.List;
import java.util.Random;

public class App {
 public static void main(String[] args) {
 new COBFrame("COB");
}
}
```

Source Code A.17: Class AppFrame.java

```java
package com.logic;

import javax.swing.JFrame;

public class AppFrame extends JFrame {
 private static final long serialVersionUID = -296324805436899545L;

 public AppFrame() {
  setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  pack();
  setVisible(true);
 }

}
```

Source Code A.18: Class COBFrame.java

```java
package com.logic;

```

```java
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;

import com.logic.Computator.Action;
import com.logic.Computator.UnaryModalOperator;
import com.logic.Computator.WeakAction;

public class COBFrame extends JFrame {

 private JTextField txtNumOfWorldsModel1;
 private JTextField txtNumOfWorldsModel2;
 private JTextField txtNumOfPropLetters;
 private JTextField txtNumOfRelations;

 private JLabel lblModel1Path;
 private JLabel lblModel2Path;

 private JRadioButton rbGodel;
 private JRadioButton rbLukasiewicz;
 private JRadioButton rbGoguen;
 private JRadioButton rbNilpotentMinimum;

 private List<JCheckBox> bisimulations = new ArrayList<JCheckBox>();
 private List<JCheckBox> weakBisimulations = new ArrayList<JCheckBox>();
 private List<JCheckBox> unaryModalOperators = new ArrayList<JCheckBox>();

 private static final long serialVersionUID = 1L;

 public COBFrame(String title) {
  super(title);
  setDefaultCloseOperation(EXIT_ON_CLOSE);
  setSize(500, 750);

  Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();
  setLocation(dim.width / 2 - this.getSize().width / 2, dim.height / 2 - this.getSize().
      height / 2);

  setLayout(new BorderLayout());

  JPanel header = new JPanel();
  header.add(new JLabel(""));
  add(header, BorderLayout.NORTH);

  JPanel content = new JPanel();
  content.setLayout(new GridLayout(4, 1));
  content.add(firstRow());
  content.add(secondRow());
  content.add(thirdRow());
  content.add(forthRow());
  add(content, BorderLayout.CENTER);

  JPanel footer = new JPanel();
```

```java
        JButton btnCompute = new JButton("Compute");
        btnCompute.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                compute();
            }
        });
        footer.add(btnCompute);
        add(footer, BorderLayout.SOUTH);

        setVisible(true);
    }

    private void compute() {
        if (txtNumOfWorldsModel1.getText() == null  txtNumOfWorldsModel1.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Number of worlds in Model 1 is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int numOfWorldsModel1 = Integer.parseInt(txtNumOfWorldsModel1.getText());

        if (txtNumOfWorldsModel2.getText() == null  txtNumOfWorldsModel2.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Number of worlds in Model 2 is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int numOfWorldsModel2 = Integer.parseInt(txtNumOfWorldsModel2.getText());

        if (txtNumOfPropLetters.getText() == null  txtNumOfPropLetters.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Number of propositional letters is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int numOfLetters = Integer.parseInt(txtNumOfPropLetters.getText());

        if (txtNumOfRelations.getText() == null  txtNumOfRelations.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Number of relations is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        int numOfRelations = Integer.parseInt(txtNumOfRelations.getText());

        if (lblModel1Path.getText() == null  lblModel1Path.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Path for Model 1 is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        String model1Path = lblModel1Path.getText();

        if (lblModel2Path.getText() == null  lblModel2Path.getText().isBlank()) {
            JOptionPane.showMessageDialog(this, "WARNING. Path for Model 2 is empty.", "Warning", JOptionPane.WARNING_MESSAGE);
            return;
        }
        String model2Path = lblModel2Path.getText();

        CRLattice crLattice = getCRLattice();

        Model model1 = null;
        try {
            model1 = new Model(model1Path, numOfWorldsModel1, numOfRelations, numOfLetters, crLattice);
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, "WARNING. Something wrong with Model 1: " + e.getMessage(), "Warning", JOptionPane.WARNING_MESSAGE);
        }

        Model model2 = null;
        try {
            model2 = new Model(model2Path, numOfWorldsModel2, numOfRelations, numOfLetters, crLattice);
```

```
 } catch (Exception e) {
  JOptionPane.showMessageDialog(this, "WARNING. Something wrong with Model 2: " + e.
      getMessage(), "Warning", JOptionPane.WARNING_MESSAGE);
 }

 List<Action> actions = new ArrayList<Action>();
 for (JCheckBox cb : bisimulations) {
  if (cb.isSelected()) {
   actions.add(Action.valueOf(cb.getName()));
  }
 }

 List<WeakAction> weakActions = new ArrayList<WeakAction>();
 for (JCheckBox cb : weakBisimulations) {
  if (cb.isSelected()) {
   weakActions.add(WeakAction.valueOf(cb.getName()));
  }
 }

 List<UnaryModalOperator> unaryModalOperators = new ArrayList<Computator.
     UnaryModalOperator>();
 for (JCheckBox cb : this.unaryModalOperators) {
  if (cb.isSelected()) {
   unaryModalOperators.add(UnaryModalOperator.valueOf(cb.getName()));
  }
 }

 String result = Computator.compute(model1, model2, actions, unaryModalOperators,
     weakActions);

 JFileChooser fileChooser = new JFileChooser();
       int option = fileChooser.showSaveDialog(this);
 if (option == JFileChooser.APPROVE_OPTION) {
  File file = fileChooser.getSelectedFile();
  FileWriter myWriter = null;
  try {
   myWriter = new FileWriter(file.getAbsoluteFile());
   myWriter.write(result);
  } catch (IOException e) {
   e.printStackTrace();
  } finally {
   if (myWriter != null) {
    try {
     myWriter.close();
    } catch (IOException e) {
     e.printStackTrace();
    }
   }
  }

 }
}

private CRLattice getCRLattice() {
 if (rbGodel.isSelected()) {
  return new CRLatticeGodel();
 }
 if (rbLukasiewicz.isSelected()) {
  return new CRLatticeLukasiewicz();
 }
 if (rbGoguen.isSelected()) {
  return new CRLatticeProduct();
 }
 return new CRLatticeNilMin();
}

private JPanel firstRow() {
 JPanel panel = new JPanel();
 panel.setLayout(new GridLayout(2, 2));

 txtNumOfWorldsModel1 = new JTextField(2);
 JPanel p1 = new JPanel();
 p1.setLayout(new FlowLayout(FlowLayout.LEFT));
```

```
  p1.add(txtNumOfWorldsModel1);
  p1.add(new JLabel("Number of worlds in Model 1"));
  panel.add(p1);

  txtNumOfWorldsModel2 = new JTextField(2);
  JPanel p2 = new JPanel();
  p2.setLayout(new FlowLayout(FlowLayout.LEFT));
  p2.add(txtNumOfWorldsModel2);
  p2.add(new JLabel("Number of worlds in Model 2"));
  panel.add(p2);

  txtNumOfPropLetters = new JTextField(2);
  JPanel p3 = new JPanel();
  p3.setLayout(new FlowLayout(FlowLayout.LEFT));
  p3.add(txtNumOfPropLetters);
  p3.add(new JLabel("Number of propositional letters"));
  panel.add(p3);

  txtNumOfRelations = new JTextField(2);
  JPanel p4 = new JPanel();
  p4.setLayout(new FlowLayout(FlowLayout.LEFT));
  p4.add(txtNumOfRelations);
  p4.add(new JLabel("Number of relations"));
  panel.add(p4);

  return panel;
 }

 private JPanel secondRow() {
  JPanel panel = new JPanel();
  panel.setLayout(new GridLayout(3, 1));

  JPanel p1 = new JPanel();
  p1.setLayout(new FlowLayout(FlowLayout.LEFT));
  p1.add(new JLabel("Model 1:"));
  lblModel1Path = new JLabel();
  JButton btnBrowse1 = new JButton("Browse");
  btnBrowse1.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
     JFileChooser fc = new JFileChooser();
     int i = fc.showOpenDialog(p1);
     if (i == JFileChooser.APPROVE_OPTION) {
      File f = fc.getSelectedFile();
      String filepath = f.getPath();
      lblModel1Path.setText(filepath);
     }
   }
  });
  p1.add(btnBrowse1);
  p1.add(lblModel1Path);
  panel.add(p1);

  JPanel p2 = new JPanel();
  p2.setLayout(new FlowLayout(FlowLayout.LEFT));
  p2.add(new JLabel("Model 2:"));
  lblModel2Path = new JLabel();
  JButton btnBrowse2 = new JButton("Browse");
  btnBrowse2.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
     JFileChooser fc = new JFileChooser();
     int i = fc.showOpenDialog(p1);
     if (i == JFileChooser.APPROVE_OPTION) {
      File f = fc.getSelectedFile();
      String filepath = f.getPath();
      lblModel2Path.setText(filepath);
     }
   }
  });
  p2.add(btnBrowse2);
  p2.add(lblModel2Path);
  panel.add(p2);
```

```java

        JPanel p3 = new JPanel();
        rbGodel = new JRadioButton("Godel structure", true);
        rbLukasiewicz = new JRadioButton("Lukasiewicz structure");
        rbGoguen = new JRadioButton("Goguen structure");
        rbNilpotentMinimum = new JRadioButton("Nilpotent minimum structure");
        ButtonGroup bg = new ButtonGroup();
        bg.add(rbGodel);
        bg.add(rbLukasiewicz);
        bg.add(rbGoguen);
        bg.add(rbNilpotentMinimum);
        p3.add(rbGodel);
        p3.add(rbLukasiewicz);
        p3.add(rbGoguen);
        p3.add(rbNilpotentMinimum);
        panel.add(p3);

        return panel;
    }

    private JPanel thirdRow() {
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(7, 2));

        JCheckBox cb11ForwardSimulation = new JCheckBox("Forward simulation");
        cb11ForwardSimulation.setName(Action.ForwardSimulation.toString());
        this.bisimulations.add(cb11ForwardSimulation);
        panel.add(cb11ForwardSimulation);

        JCheckBox cb12CrispForwardSimulation = new JCheckBox("Crisp forward simulation");
        cb12CrispForwardSimulation.setName(Action.CrispForwardSimulation.toString());
        this.bisimulations.add(cb12CrispForwardSimulation);
        panel.add(cb12CrispForwardSimulation);

        JCheckBox cb21BackwardSimulation = new JCheckBox("Backward simulation");
        cb21BackwardSimulation.setName(Action.BackwardSimulation.toString());
        this.bisimulations.add(cb21BackwardSimulation);
        panel.add(cb21BackwardSimulation);

        JCheckBox cb22CrispBackwardSimulation = new JCheckBox("Crisp backward simulation");
        cb22CrispBackwardSimulation.setName(Action.CrispBackwardSimulation.toString());
        this.bisimulations.add(cb22CrispBackwardSimulation);
        panel.add(cb22CrispBackwardSimulation);

        JCheckBox cb31ForwardBisimulation = new JCheckBox("Forward bisimulation");
        cb31ForwardBisimulation.setName(Action.ForwardBisimulation.toString());
        this.bisimulations.add(cb31ForwardBisimulation);
        panel.add(cb31ForwardBisimulation);

        JCheckBox cb32CrispForwardBisimulation = new JCheckBox("Crisp forward bisimulation");
        cb32CrispForwardBisimulation.setName(Action.CrispForwardBisimulation.toString());
        this.bisimulations.add(cb32CrispForwardBisimulation);
        panel.add(cb32CrispForwardBisimulation);

        JCheckBox cb41BackwardBisimulation = new JCheckBox("Backward bisimulation");
        cb41BackwardBisimulation.setName(Action.BackwardBisimulation.toString());
        this.bisimulations.add(cb41BackwardBisimulation);
        panel.add(cb41BackwardBisimulation);

        JCheckBox cb42CrispBackwardBisimulation = new JCheckBox("Crisp backward bisimulation")
            ;
        cb42CrispBackwardBisimulation.setName(Action.CrispBackwardBisimulation.toString());
        this.bisimulations.add(cb42CrispBackwardBisimulation);
        panel.add(cb42CrispBackwardBisimulation);

        JCheckBox cb51ForwardBackwardBisimulation = new JCheckBox("Forward backward
            bisimulation");
        cb51ForwardBackwardBisimulation.setName(Action.ForwardBackwardBisimulation.toString())
            ;
        this.bisimulations.add(cb51ForwardBackwardBisimulation);
        panel.add(cb51ForwardBackwardBisimulation);

        JCheckBox cb52CrispForwardBackwardBisimulation = new JCheckBox("Crisp forward backward
```

```java
         bisimulation");
    cb52CrispForwardBackwardBisimulation.setName(Action.CrispForwardBackwardBisimulation.toString());
    this.bisimulations.add(cb52CrispForwardBackwardBisimulation);
    panel.add(cb52CrispForwardBackwardBisimulation);

    JCheckBox cb61BackwardForwardBisimulation = new JCheckBox("Backward forward bisimulation");
    cb61BackwardForwardBisimulation.setName(Action.BackwardForwardBisimulation.name());
    this.bisimulations.add(cb61BackwardForwardBisimulation);
    panel.add(cb61BackwardForwardBisimulation);

    JCheckBox cb62CrispBackwardForwardBisimulation = new JCheckBox("Crisp backward forward bisimulation");
    cb62CrispBackwardForwardBisimulation.setName(Action.CrispBackwardForwardBisimulation.name());
    this.bisimulations.add(cb62CrispBackwardForwardBisimulation);
    panel.add(cb62CrispBackwardForwardBisimulation);

    JCheckBox cb71RegularBisimulation = new JCheckBox("Regular bisimulation");
    cb71RegularBisimulation.setName(Action.RegularBisimulation.name());
    this.bisimulations.add(cb71RegularBisimulation);
    panel.add(cb71RegularBisimulation);

    JCheckBox cb72CrispRegularBisimulation = new JCheckBox("Crisp regular bisimulation");
    cb72CrispRegularBisimulation.setName(Action.CrispRegularBisimulation.name());
    this.bisimulations.add(cb72CrispRegularBisimulation);
    panel.add(cb72CrispRegularBisimulation);

    return panel;
  }

  private JPanel forthRow() {
    JPanel mainPanel = new JPanel();
    mainPanel.setLayout(new GridLayout(2, 1));

    JPanel upperPanel = new JPanel();
    upperPanel.setLayout(new GridLayout(2, 2));
    mainPanel.add(upperPanel);

    JCheckBox cb81WeakSimulation = new JCheckBox("Weak simulation");
    cb81WeakSimulation.setName(WeakAction.WeakSimulation.name());
    this.weakBisimulations.add(cb81WeakSimulation);
    upperPanel.add(cb81WeakSimulation);

    JCheckBox cb82CrispWeakSimulation = new JCheckBox("Crisp weak simulation");
    cb82CrispWeakSimulation.setName(WeakAction.CrispWeakSimulation.name());
    this.weakBisimulations.add(cb82CrispWeakSimulation);
    upperPanel.add(cb82CrispWeakSimulation);

    JCheckBox cb91WeakBisimulation = new JCheckBox("Weak bisimulation");
    cb91WeakBisimulation.setName(WeakAction.WeakBisimulation.name());
    this.weakBisimulations.add(cb91WeakBisimulation);
    upperPanel.add(cb91WeakBisimulation);

    JCheckBox cb92CrispWeakBisimulation = new JCheckBox("Crisp weak bisimulation");
    cb92CrispWeakBisimulation.setName(WeakAction.CrispWeakBisimulation.name());
    this.weakBisimulations.add(cb92CrispWeakBisimulation);
    upperPanel.add(cb92CrispWeakBisimulation);

    JPanel bottomPanel = new JPanel();
    bottomPanel.setLayout(new GridLayout(1, 4));
    mainPanel.add(bottomPanel);

    JCheckBox cbNecessity = new JCheckBox("Necessity");
    cbNecessity.setName(UnaryModalOperator.Necessity.name());
    this.unaryModalOperators.add(cbNecessity);
    bottomPanel.add(cbNecessity);

    JCheckBox cbNecessityInv = new JCheckBox("NecessityInv");
    cbNecessityInv.setName(UnaryModalOperator.NecessityInv.name());
    this.unaryModalOperators.add(cbNecessityInv);
    bottomPanel.add(cbNecessityInv);
```

```java

    JCheckBox cbPossibility = new JCheckBox("Possibility");
    cbPossibility.setName(UnaryModalOperator.Possibility.name());
    this.unaryModalOperators.add(cbPossibility);
    bottomPanel.add(cbPossibility);

    JCheckBox cbPossibilityInv = new JCheckBox("PossibilityInv");
    cbPossibilityInv.setName(UnaryModalOperator.PossibilityInv.name());
    this.unaryModalOperators.add(cbPossibilityInv);
    bottomPanel.add(cbPossibilityInv);

    return mainPanel;
  }

}
```

Source Code A.19: Class `Computator.java`

```java
package com.logic;

import java.util.ArrayList;
import java.util.List;

import com.logic.operations.BiImplication;
import com.logic.operations.BinaryOperation;
import com.logic.operations.Conjunction;
import com.logic.operations.Disjunction;
import com.logic.operations.RightImplication;
import com.logic.operations.Necessity;
import com.logic.operations.NecessityInv;
import com.logic.operations.Negation;
import com.logic.operations.Possibility;
import com.logic.operations.PossibilityInv;
import com.logic.operations.LeftImplication;
import com.logic.operations.TNorm;
import com.logic.operations.UnaryOperation;

public class Computator {

  public static final int MAX_ITERATIONS = 100;

  public static String compute(Model model1, Model model2, List<Action> actions,
      List<UnaryModalOperator> unaryModalOperators, List<WeakAction> weakActions) {
    List<String> results = new ArrayList<String>();

    for (Action action : actions) {
      results.add(compute(model1, model2, action));
    }

    Model disjointModel = Model.disjointModel(model1, model2);

    StringBuilder sb = new StringBuilder();
    sb.append("Model 1:").append(System.lineSeparator()).append(model1).append(System.
        lineSeparator())
      .append(System.lineSeparator());
    sb.append("Model 2:").append(System.lineSeparator()).append(model2).append(System.
        lineSeparator())
      .append(System.lineSeparator());
    sb.append(String.join(System.lineSeparator() + System.lineSeparator(), results));
    sb.append(System.lineSeparator() + System.lineSeparator());
    sb.append("Disjoint model:").append(System.lineSeparator()).append(disjointModel).
        append(System.lineSeparator())
      .append(System.lineSeparator());
    sb.append(String.join(System.lineSeparator() + System.lineSeparator()));

    // Computation of weak simulations and bisimulations
    sb.append(computeWeak(disjointModel, unaryModalOperators, weakActions));

    return sb.toString();
  }

  private static String compute(Model model1, Model model2, Action action) {
    FuzzyRelation result = null;
```

```java
String modelReductionResult = null;

switch (action) {
case ForwardSimulation:
 result = Algorithms.forwardSimulation(model1, model2, MAX_ITERATIONS);
 break;
case CrispForwardSimulation:
 result = Algorithms.crispForwardSimulation(model1, model2, MAX_ITERATIONS);
 break;
case BackwardSimulation:
 result = Algorithms.backwardSimulation(model1, model2, MAX_ITERATIONS);
 break;
case CrispBackwardSimulation:
 result = Algorithms.crispBackwardSimulation(model1, model2, MAX_ITERATIONS);
 break;
case ForwardBisimulation:
 result = Algorithms.forwardBisimulation(model1, model2, MAX_ITERATIONS);
 if (model1.equals(model2) && result != null && result.isQuasiOrder(model1.getCrLattice())) {
  modelReductionResult = Algorithms.modelReduction(model1, result, Action.ForwardBisimulation);
 }
 break;
case CrispForwardBisimulation:
 result = Algorithms.crispForwardBisimulation(model1, model2, MAX_ITERATIONS);
 break;
case BackwardBisimulation:
 result = Algorithms.backwardBisimulation(model1, model2, MAX_ITERATIONS);
 if (model1.equals(model2) && result != null && result.isQuasiOrder(model1.getCrLattice())) {
  modelReductionResult = Algorithms.modelReduction(model1, result, Action.BackwardBisimulation);
 }
 break;
case CrispBackwardBisimulation:
 result = Algorithms.crispBackwardBisimulation(model1, model2, MAX_ITERATIONS);
 break;
case ForwardBackwardBisimulation:
 result = Algorithms.forwardBackwardBisimulation(model1, model2, MAX_ITERATIONS);
 if (model1.equals(model2) && result != null && result.isQuasiOrder(model1.getCrLattice())) {
  modelReductionResult = Algorithms.modelReduction(model1, result, Action.ForwardBackwardBisimulation);
 }
 break;
case CrispForwardBackwardBisimulation:
 result = Algorithms.crispForwardBackwardBisimulation(model1, model2, MAX_ITERATIONS);
 break;
case BackwardForwardBisimulation:
 result = Algorithms.backwardForwardBisimulation(model1, model2, MAX_ITERATIONS);
 if (model1.equals(model2) && result != null && result.isQuasiOrder(model1.getCrLattice())) {
  modelReductionResult = Algorithms.modelReduction(model1, result, Action.BackwardForwardBisimulation);
 }
 break;
case CrispBackwardForwardBisimulation:
 result = Algorithms.crispBackwardForwardBisimulation(model1, model2, MAX_ITERATIONS);
 break;
case RegularBisimulation:
 result = Algorithms.regularBisimulation(model1, model2, MAX_ITERATIONS);
 if (model1.equals(model2) && result != null && result.isQuasiOrder(model1.getCrLattice())) {
  modelReductionResult = Algorithms.modelReduction(model1, result, Action.RegularBisimulation);
 }
 break;
case CrispRegularBisimulation:
 result = Algorithms.crispRegularBisimulation(model1, model2, MAX_ITERATIONS);
 break;
default:
 result = null;
}
```

```

 String resStr = action.toString() + ":" + System.lineSeparator() + (result == null ? " Doesn't exist" : result) + System.lineSeparator();

 if (modelReductionResult != null) {
  resStr = resStr + System.lineSeparator() + "Model reduced:" + System.lineSeparator() + modelReductionResult;
 }

 return resStr;
}

private static List<TSet> computeTSetList(Model disjointModel, List<UnaryOperation> unaryOperations, List<BinaryOperation> binaryOperations, CRLattice crLattice) {
 FSet fSet = FSet.generate(disjointModel, disjointModel.getAlgebra());
 List<FSet> fSetList = new ArrayList<FSet>();
 fSetList.add(fSet);
 TSet firstTSet = TSet.generate(fSet);

 List<TSet> tSetList = new ArrayList<TSet>();
 tSetList.add(firstTSet);

 List<TSet> tSetUnion = new ArrayList<TSet>();
 tSetUnion.add(firstTSet);

 boolean done = false;
 int iteration = 1;
 do {
  TSet tSet = TSet.generate(tSetList, unaryOperations, binaryOperations, crLattice);
  tSetList.add(tSet);

  List<FuzzyFormula> formulasUnion = new ArrayList<FuzzyFormula>();
  formulasUnion.addAll(tSetUnion.get(iteration - 1).getFormulaList());
  formulasUnion.addAll(tSet.getFormulaList());
  tSetUnion.add(new TSet(formulasUnion));

  if (iteration % 2 == 1) {
   done = tSetUnion.get(iteration).equals(tSetUnion.get((iteration - 1) / 2));
  }

  iteration++;
 } while (!done);

 return tSetList;
}

public static String computeWeak(Model disjointModel, List<UnaryModalOperator> unaryModalOperators, List<WeakAction> weakActions) {
 if (weakActions.isEmpty()) {
  return "";
 }

 CRLattice crLattice = disjointModel.getCrLattice();

 List<UnaryOperation> unaryOperations = getUnaryOperations(disjointModel, unaryModalOperators);
 List<BinaryOperation> binaryOperations = getBinaryOperations();

 List<TSet> tSetList = computeTSetList(disjointModel, unaryOperations, binaryOperations, crLattice);

 StringBuilder result = new StringBuilder();

 for (int i = 0; i < tSetList.size(); i++) {
  result.append("Iteration " + i + ":").append(System.lineSeparator());
  result.append(tSetList.get(i)).append(System.lineSeparator());
 }

 result.append(System.lineSeparator());

 FuzzyRelation weakSimulation = null;
 FuzzyRelation crispWeakSimulation = null;
 FuzzyRelation weakBisimulation = null;
```

```java
FuzzyRelation crispWeakBisimulation = null;

for (int i = 0; i < tSetList.size(); i++) {
 TSet tSet = tSetList.get(i);

 for (int j = 0; j < tSet.getFormulaList().size(); j++) {
  FuzzyFormula formula = tSet.getFormulaList().get(j);

  for (WeakAction weakAction : weakActions) {
   FuzzyRelation fr;
   switch (weakAction) {
   case WeakSimulation:
    fr = FuzzySet.weakSimulation(formula.getFuzzySet(), formula.getFuzzySet(),
        crLattice);
    weakSimulation = weakSimulation == null ? fr : FuzzyRelation.conjunction(
        weakSimulation, fr);
    break;
   case CrispWeakSimulation:
    fr = FuzzySet.weakSimulation(formula.getFuzzySet(), formula.getFuzzySet(),
        crLattice);
    crispWeakSimulation = crispWeakSimulation == null ? fr
      : FuzzyRelation.conjunction(crispWeakSimulation, fr);
    crispWeakSimulation = crispWeakSimulation.crisp();
    break;
   case WeakBisimulation:
    fr = FuzzySet.weakBisimulation(formula.getFuzzySet(), formula.getFuzzySet(),
        crLattice);
    weakBisimulation = weakBisimulation == null ? fr
      : FuzzyRelation.conjunction(weakBisimulation, fr);
    break;
   case CrispWeakBisimulation:
    fr = FuzzySet.weakBisimulation(formula.getFuzzySet(), formula.getFuzzySet(),
        crLattice);
    crispWeakBisimulation = crispWeakBisimulation == null ? fr
      : FuzzyRelation.conjunction(crispWeakBisimulation, fr);
    crispWeakBisimulation = crispWeakBisimulation.crisp();
    break;
   default:
    break;
   }
  }
 }

 if (weakSimulation != null) {
  result.append("ConjuctionRelation-WS:").append(System.lineSeparator()).append(
      weakSimulation)
    .append(System.lineSeparator());
 }
 if (crispWeakSimulation != null) {
  result.append("ConjuctionRelation-CWS:").append(System.lineSeparator()).append(
      crispWeakSimulation)
    .append(System.lineSeparator());
 }
 if (weakBisimulation != null) {
  result.append("ConjuctionRelation-WB:").append(System.lineSeparator()).append(
      weakBisimulation)
    .append(System.lineSeparator());
 }
 if (crispWeakBisimulation != null) {
  result.append("ConjuctionRelation-CWB:").append(System.lineSeparator()).append(
      crispWeakBisimulation)
    .append(System.lineSeparator());
 }
}

if (weakSimulation != null) {
 boolean found = true;
 for (int i = 0; i < disjointModel.getFuzzySets().size(); i++) {
  FuzzySet fs = disjointModel.getFuzzySets().get(i);
  boolean lessOrEqual = fs.lessOrEqual(fs.compose(weakSimulation.transpose(),
      disjointModel.getCrLattice()));
  if (!lessOrEqual) {
   found = false;
```

```
    break;
   }
  }
  result.append("Maximum weak simulation found: ").append(found).append(System.lineSeparator());
 }
 if (crispWeakSimulation != null) {
  boolean found = true;
  for (int i = 0; i < disjointModel.getFuzzySets().size(); i++) {
   FuzzySet fs = disjointModel.getFuzzySets().get(i);
   boolean lessOrEqual = fs.lessOrEqual(fs.compose(crispWeakSimulation.transpose(), disjointModel.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }
  result.append("Maximum crisp weak simulation found: ").append(found).append(System.lineSeparator());
 }
 if (weakBisimulation != null) {
  boolean found = true;
  for (int i = 0; i < disjointModel.getFuzzySets().size(); i++) {
   FuzzySet fs = disjointModel.getFuzzySets().get(i);
   boolean lessOrEqual = fs.lessOrEqual(fs.compose(weakBisimulation.transpose(), disjointModel.getCrLattice()))
     && fs.lessOrEqual(fs.compose(weakBisimulation, disjointModel.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }
  result.append("Maximum weak bisimulation found: ").append(found).append(System.lineSeparator());
 }
 if (crispWeakBisimulation != null) {
  boolean found = true;
  for (int i = 0; i < disjointModel.getFuzzySets().size(); i++) {
   FuzzySet fs = disjointModel.getFuzzySets().get(i);
   boolean lessOrEqual = fs.lessOrEqual(fs.compose(crispWeakBisimulation.transpose(), disjointModel.getCrLattice()))
     && fs.lessOrEqual(fs.compose(crispWeakBisimulation, disjointModel.getCrLattice()));
   if (!lessOrEqual) {
    found = false;
    break;
   }
  }
  result.append("Maximum crisp weak bisimulation found: ").append(found).append(System.lineSeparator());
 }

 return result.toString();
}

private static List<UnaryOperation> getUnaryOperations(Model model, List<UnaryModalOperator> unaryModalOperators) {
 List<UnaryOperation> unaryOperations = new ArrayList<UnaryOperation>();
 unaryOperations.add(new Negation());
 for (FuzzyRelation fuzzyRelation : model.getFuzzyRelations()) {
  for (UnaryModalOperator operator : unaryModalOperators) {

   if (operator == UnaryModalOperator.Necessity) {
    unaryOperations.add(new Necessity(fuzzyRelation));
   } else if (operator == UnaryModalOperator.NecessityInv) {
    unaryOperations.add(new NecessityInv(fuzzyRelation));
   } else if (operator == UnaryModalOperator.Possibility) {
    unaryOperations.add(new Possibility(fuzzyRelation));
   } else if (operator == UnaryModalOperator.PossibilityInv) {
    unaryOperations.add(new PossibilityInv(fuzzyRelation));
   }

  }
```

```java
    }
    return unaryOperations;
  }

  private static List<BinaryOperation> getBinaryOperations() {
    List<BinaryOperation> binaryOperations = new ArrayList<BinaryOperation>();
    binaryOperations.add(new Conjunction());
    binaryOperations.add(new TNorm());
//    binaryOperations.add(new RightImplication());
    binaryOperations.add(new LeftImplication());
//    binaryOperations.add(new Disjunction());
//    binaryOperations.add(new BiImplication());
    return binaryOperations;
  }

  public enum Action {
    ForwardSimulation, CrispForwardSimulation, BackwardSimulation, CrispBackwardSimulation
        , ForwardBisimulation,
    CrispForwardBisimulation, BackwardBisimulation, CrispBackwardBisimulation,
        ForwardBackwardBisimulation,
    CrispForwardBackwardBisimulation, BackwardForwardBisimulation,
        CrispBackwardForwardBisimulation,
    RegularBisimulation, CrispRegularBisimulation
  }

  public enum WeakAction {
    WeakSimulation, CrispWeakSimulation, WeakBisimulation, CrispWeakBisimulation
  }

  public enum UnaryModalOperator {
    Necessity, NecessityInv, Possibility, PossibilityInv
  }
}
```

Source Code A.20: Class `CRLattice.java`

```java
package com.logic;

public abstract class CRLattice {

  public abstract double mult(double a, double b);

  public abstract double res(double a, double b);

  public double biRes(double a, double b) {
    return Math.min(res(a, b), res(b, a));
  }

}
```

Source Code A.21: Class `CRLatticeGodel`

```java
package com.logic;

public class CRLatticeGodel extends CRLattice {

  @Override
  public double mult(double a, double b) {
    return Math.min(a, b);
  }

  @Override
  public double res(double a, double b) {
    return (a <= b ? 1 : b);
  }

}
```

Source Code A.22: Class `CRLatticeLukasiewicz.java`

```java
package com.logic;

public class CRLatticeLukasiewicz extends CRLattice {

 @Override
 public double mult(double a, double b) {
  return Math.max(a + b - 1, 0);
 }

 @Override
 public double res(double a, double b) {
  return Math.min(1, 1 - a + b);
 }

}
```

Source Code A.23: Class `CRLatticeNilMin.java`

```java
package com.logic;

public class CRLatticeNilMin extends CRLattice {

 @Override
 public double mult(double a, double b) {
  return (a+b <= 1 ? 0 : Math.min(a, b));
 }

 @Override
 public double res(double a, double b) {
  return (a <= b ? 1 : Math.max(1-a, b));
 }

}
```

Source Code A.24: Class `CRLatticeProduct.java`

```java
package com.logic;

public class CRLatticeProduct extends CRLattice {

 @Override
 public double mult(double a, double b) {
  return a*b;
 }

 @Override
 public double res(double a, double b) {
  return (a <= b ? 1 : b / a);
 }

}
```

Source Code A.25: Class `FileParser.java`

```java
package com.logic;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

public class FileParser {

 public static List<Double> parse(String filePath) throws FileNotFoundException,
     IOException {
  List<Double> result = new ArrayList<Double>();

  try (BufferedReader br = new BufferedReader(new FileReader(new File(filePath)))) {
```

```
      String line;
      while ((line = br.readLine()) != null) {
        line = line.trim();
        // Ignoring empty or lines that start with #
        if (line == null  line.isBlank()  line.startsWith("#")) {
         continue;
        }
        String[] tokenArray = line.split("\\s+");
        for (String token : tokenArray) {
         result.add(Double.parseDouble(token));
        }
      }
  }

  return result;
 }


}
```

Source Code A.26: Class `FSet.java`

```
package com.logic;

import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

public class FSet {

 private List<FuzzyFormula> formulaList;

 private FSet(List<FuzzyFormula> formulList) {
  this.formulaList = formulList;
 }

 public static FSet generate(Model model, Set<Double> algebra) {

  List<FuzzyFormula> formulas = new ArrayList<FuzzyFormula>();

  Set<Double> wholeAlgebraSet = new HashSet<Double>();
  wholeAlgebraSet.addAll(model.getAlgebra());
  wholeAlgebraSet.addAll(algebra);

  List<Double> wholeAlgebra = new ArrayList<Double>(wholeAlgebraSet);
  Collections.sort(wholeAlgebra);

  int numOfElements = model.getFuzzySets().get(0).getNumberOfElements();

  for (Double c : wholeAlgebra) {
   double[] values = new double[numOfElements];
   for (int i = 0; i < numOfElements; i++) {
    values[i] = c;
   }
   FuzzyFormula constFormula = new FuzzyFormula(new FuzzySet(values));
   formulas.add(constFormula);
  }

  for (FuzzySet fuzzySet : model.getFuzzySets()) {
   formulas.add(new FuzzyFormula(fuzzySet));
  }

  return new FSet(formulas);
 }

 public List<FuzzyFormula> getFormulaList() {
  return formulaList;
 }
}
```

Source Code A.27: Class `FuzzyFormula.java`

```java
package com.logic;

import com.logic.operations.Operation;

public class FuzzyFormula {

 private FuzzySet fuzzySet;

 private Operation parentOperation;
 private FuzzyFormula firstParent;
 private FuzzyFormula secondParent;

 public FuzzyFormula(FuzzySet fuzzySet) {
  this.fuzzySet = fuzzySet;
 }

 public FuzzyFormula(FuzzySet fuzzySet, Operation parentOperation, FuzzyFormula
     firstParent, FuzzyFormula secondParent) {
  this.fuzzySet = fuzzySet;
  this.parentOperation = parentOperation;
  this.firstParent = firstParent;
  this.secondParent = secondParent;
 }

 public int getOperationCount() {
  int result = 0;
  if (firstParent != null) {
   result += firstParent.getOperationCount() + 1;
  }
  if (secondParent != null) {
   result += secondParent.getOperationCount();
  }
  return result;
 }

 public FuzzySet getFuzzySet() {
  return fuzzySet;
 }

 public Operation getOperation() {
  return parentOperation;
 }

}
```

Source Code A.28: Class `FuzzyRelation.java`

```java
package com.logic;

import java.util.Arrays;
import java.util.Locale;

public class FuzzyRelation {
 public static final double PRECISION = 1e-2;

 private double[][] values;
 private int rows;
 private int cols;

 public FuzzyRelation(int rows, int cols) {
  this.values = new double[rows][cols];
  this.rows = rows;
  this.cols = cols;
 }

 public FuzzyRelation(double[][] values) {
  this.values = values;
  this.rows = values.length;
  this.cols = values[0].length;
 }

```

```
public FuzzyRelation compose(FuzzyRelation other, CRLattice crLattice) {
 if (this.cols != other.rows)
  throw new IllegalArgumentException("Numbers of rows and/or columns don't match");

 double res[][] = new double[this.rows][other.cols];

 for (int i = 0; i < res.length; i++) {
  for (int j = 0; j < res[0].length; j++) {
   double max = Double.MIN_VALUE;
   for (int k = 0; k < this.cols; k++) {
    max = Math.max(max, crLattice.mult(this.values[i][k], other.values[k][j]));
   }
   res[i][j] = max;
  }
 }

 return new FuzzyRelation(res);
}

public FuzzySet compose(FuzzySet fuzzySet, CRLattice crLattice) {
 if (this.cols != fuzzySet.getNumberOfElements())
  throw new IllegalArgumentException("Numbers of relation rows and set elements don't match");

 double res[] = new double[this.rows];

 for (int i = 0; i < this.rows; i++) {
  double max = Double.MIN_VALUE;
  for (int k = 0; k < this.cols; k++) {
   max = Math.max(max, crLattice.mult(values[i][k], fuzzySet.getValue(k)));
  }
  res[i] = max;
 }

 return new FuzzySet(res);
}

public boolean isQuasiOrder(CRLattice crLattice) {
 for (int i = 0; i < rows; i++) {
  if (values[i][i] != 1) {
   return false;
  }
 }

 FuzzyRelation leftSide = this.compose(this, crLattice);

 return leftSide.lessOrEqual(this);
}

public boolean lessOrEqual(FuzzyRelation other) {
 if(other == null)
  throw new IllegalArgumentException("Parameter other must not be null");
 if (this.cols != other.cols  this.rows != other.rows)
  return false;
 for(int i = 0; i < rows; i++) {
  for (int j = 0; j < cols; j++) {
   if (Math.abs(this.values[i][j] - other.values[i][j]) > PRECISION)
    return false;
  }
 }
 return true;
}

public int getRows() {
 return rows;
}

public int getCols() {
 return cols;
}

public double getValue(int row, int col) {
 if (row < 0  row >= this.rows) {
```

```
   throw new IndexOutOfBoundsException(
     String.format("row must be in interval [0, %s], but it's %s", rows - 1, row));
  }
  if (col < 0 col >= this.cols) {
   throw new IndexOutOfBoundsException(
     String.format("col must be in interval [0, %s], but it's %s", cols - 1, col));
  }

  return values[row][col];
 }

 public FuzzyRelation transpose() {
  double[][] res = new double[this.cols][this.rows];
  for(int i=0; i<this.rows; i++) {
   for(int j=0; j<this.cols; j++)
    res[j][i]=this.values[i][j];
  }
  return new FuzzyRelation(res);
 }

 public static FuzzyRelation rightResidual(FuzzyRelation fr1, FuzzyRelation fr2,
     CRLattice crLattice) {
  if (fr1.cols != fr1.rows)
   throw new IllegalArgumentException("fr1 must be a quadratic fuzzy relation");
  if (fr1.cols != fr2.rows)
   throw new IllegalArgumentException("fr1.cols must be equal to fr2.rows");

  double[][] res = new double[fr2.rows][fr2.cols];

  for (int i = 0; i < fr2.rows; i++) {
   for (int j = 0; j < fr2.cols; j++) {
    double min = Double.MAX_VALUE;
    for (int k = 0; k < fr1.cols; k++) {
     min = Math.min(min, crLattice.res(fr1.getValue(k, i), fr2.getValue(k, j)));
    }
    res[i][j] = min;
   }
  }

  return new FuzzyRelation(res);
 }

 public static FuzzyRelation leftResidual(FuzzyRelation fr1, FuzzyRelation fr2,
     CRLattice crLattice) {
  if (fr1.cols != fr1.rows)
   throw new IllegalArgumentException("fr1 must be a quadratic fuzzy relation");
  if (fr1.cols != fr2.cols)
   throw new IllegalArgumentException("fr1.cols must be equal to fr2.rows");

  double[][] res = new double[fr2.rows][fr2.cols];

  for (int i = 0; i < fr2.rows; i++) {
   for (int j = 0; j < fr2.cols; j++) {
    double min = Double.MAX_VALUE;
    for (int k = 0; k < fr2.cols; k++) {
     min = Math.min(min, crLattice.res(fr1.getValue(j, k), fr2.getValue(i, k)));
    }
    res[i][j] = min;
   }
  }

  return new FuzzyRelation(res);
 }

 public static FuzzyRelation conjunction(FuzzyRelation fr1, FuzzyRelation fr2) {
  if (fr1.rows != fr2.rows)
   throw new IllegalArgumentException("fr1 and fr2 rows must be equal");
  if (fr1.cols != fr2.cols)
   throw new IllegalArgumentException("fr1 and fr2 cols must be equal");

  double[][] res = new double[fr1.rows][fr1.cols];

  for (int i = 0; i < fr1.rows; i++) {
```

```
   for (int j = 0; j < fr1.cols; j++) {
    res[i][j] = Math.min(fr1.values[i][j], fr2.values[i][j]);
   }
  }
  return new FuzzyRelation(res);
 }

 public FuzzyRelation crisp() {
  double[][] res = new double[rows][cols];
  for (int i = 0; i < rows; i++) {
   for (int j = 0; j < cols; j++) {
    res[i][j] = Math.abs(values[i][j] - 1) > PRECISION ? 0 : 1;
   }
  }
  return new FuzzyRelation(res);
 }

 public boolean areRowsEqual(int r1, int r2) {
  for (int col = 0; col < cols; col++) {
   //if (Math.abs(values[r1][col] - values[r2][col]) > PRECISION) {
   if (values[r1][col] != values[r2][col]) {
    return false;
   }
  }
  return true;
 }

 public boolean areColsEqual(int c1, int c2) {
  for (int row = 0; row < rows; row++) {
   if (values[row][c1] != values[row][c2]) {
    return false;
   }
  }
  return true;
 }

 public void reduce(int row) {
  double[][] reducedValues = new double[rows - 1][cols - 1];
  for (int i = 0; i < row; i++) {
   for (int j = 0; j < row; j++) {
    reducedValues[i][j] = values[i][j];
   }
   for (int j = row + 1; j < cols; j++) {
    reducedValues[i][j - 1] = values[i][j];
   }
  }
  for (int i = row + 1; i < rows; i++) {
   for (int j = 0; j < row; j++) {
    reducedValues[i - 1][j] = values[i][j];
   }
   for (int j = row + 1; j < cols; j++) {
    reducedValues[i - 1][j - 1] = values[i][j];
   }
  }

  this.values = reducedValues;
  this.cols--;
  this.rows--;
 }

 public boolean isZero() {
  for (int i = 0; i < rows; i++) {
   for (int j = 0; j < cols; j++) {
    if (values[i][j] != 0) {
     return true;
    }
   }
  }
  return false;
 }
}
```

Source Code A.29: Class `FuzzySet.java`

```java
package com.logic;

import java.util.Arrays;
import java.util.Locale;

public class FuzzySet {

 public static final double PRECISION = 1e-2;

 private double[] values;
 private int numberOfElements;

 public FuzzySet(int numberOfElements) {
  this.numberOfElements = numberOfElements;
  this.values = new double[numberOfElements];
 }

 public FuzzySet(double[] values) {
  this.numberOfElements = values.length;
  this.values = values;
 }

 public void setValue(int position, double value) {
  if (position >= numberOfElements) {
   throw new IndexOutOfBoundsException(
     String.format("position must be lower than %s, but it's %s", numberOfElements,
         position));
  }

  values[position] = value;
 }

 public double getValue(int position) {
  if (position >= numberOfElements) {
   throw new IndexOutOfBoundsException(
     String.format("position must be in interval [0, %s], but it's %s",
       numberOfElements - 1, position));
  }

  return values[position];
 }

 /* Check if fuzzy set is zero */
 public boolean isZero() {
  for (int i=0; i<numberOfElements; i++) {
   if(values[i]!=0)
    return false;
  }
  return true;
 }

 /* Check if fuzzy set is made of constant elements */
 public boolean isConstant() {
  double first =values[0];
  for (int i=0; i<numberOfElements; i++) {
   if(values[i]!=first)
    return false;
  }
  return true;
 }

 public double supremum() {
  double s=values[0];
  for(int i=1; i<numberOfElements;i++) {
   if(values[i]>s)
    s=values[i];
  }
  return s;
 }

 public boolean lessOrEqual(FuzzySet other) {
  if(other==null)
```

```
  throw new IllegalArgumentException("Parameter other must not be null");
 if (numberOfElements != other.numberOfElements)
  return false;
 for(int i=0; i<numberOfElements; i++) {
  if(!(Math.abs(this.values[i] - other.values[i]) < PRECISION  this.values[i]<other.values[i]))
   return false;
 }
 return true;
}

public boolean greaterOrEqual(FuzzySet other) {
 if(other==null)
  throw new IllegalArgumentException("Parameter other must not be null");
 if (numberOfElements != other.numberOfElements)
  return false;
 for(int i=0; i<numberOfElements; i++) {
  if(!(Math.abs(this.values[i] - other.values[i]) < PRECISION  this.values[i]>other.values[i]))
   return false;
 }
 return true;
}

public int getNumberOfElements() {
 return numberOfElements;
}

public FuzzySet compose(FuzzyRelation fuzzyRelation, CRLattice crLattice) {
 if (this.numberOfElements != fuzzyRelation.getRows())
  throw new IllegalArgumentException("Numbers of relation rows and set elements don't match");

 double res[] = new double[fuzzyRelation.getCols()];

 for (int j = 0; j < fuzzyRelation.getCols(); j++) {
  double max = Double.MIN_VALUE;
  for (int k = 0; k < fuzzyRelation.getRows(); k++) {
   max = Math.max(max, crLattice.mult(values[k], fuzzyRelation.getValue(k, j)));
  }
  res[j] = max;
 }

 return new FuzzySet(res);
}

public static FuzzySet conjunction(FuzzySet fs1, FuzzySet fs2) {
 if (fs1.numberOfElements != fs2.numberOfElements)
  throw new IllegalArgumentException("fs1 and fs2 number of elements must be equal");

 double[] res = new double[fs1.numberOfElements];

 for (int i = 0; i < res.length; i++) {
  res[i] = Math.min(fs1.values[i], fs2.values[i]);
 }
 return new FuzzySet(res);
}

public static FuzzySet strongConjunction(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 if (fs1.numberOfElements != fs2.numberOfElements)
  throw new IllegalArgumentException("fs1 and fs2 number of elements must be equal");

 double[] res = new double[fs1.numberOfElements];

 for (int i = 0; i < res.length; i++) {
  res[i] = crLattice.mult(fs1.values[i], fs2.values[i]);
 }
 return new FuzzySet(res);
}

public static FuzzySet maximum(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 if (fs1.numberOfElements != fs2.numberOfElements)
```

```
  throw new IllegalArgumentException("fs1 and fs2 number of elements must be equal");

 double[] res = new double[fs1.numberOfElements];

 for (int i = 0; i < res.length; i++) {
  res[i] = Math.max(fs1.values[i], fs2.values[i]);
 }
 return new FuzzySet(res);
}

public static FuzzySet leftImplication(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 if (fs1.numberOfElements != fs2.numberOfElements)
  throw new IllegalArgumentException("fs1 and fs2 number of elements must be equal");

 double[] res = new double[fs1.numberOfElements];

 for (int i = 0; i < res.length; i++) {
  res[i] = crLattice.res(fs1.values[i], fs2.values[i]);
 }
 return new FuzzySet(res);
}

public static FuzzySet rightImplication(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 if (fs1.numberOfElements != fs2.numberOfElements)
  throw new IllegalArgumentException("fs1 and fs2 number of elements must be equal");

 double[] res = new double[fs1.numberOfElements];

 for (int i = 0; i < res.length; i++) {
  res[i] = crLattice.res(fs2.values[i], fs1.values[i]);
 }
 return new FuzzySet(res);
}

public static FuzzyRelation weakSimulation(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 // each relation has the same dimension, so we take it from the first one.
 int rows = fs1.getNumberOfElements(); // the same as getCols()...
 int cols = fs2.getNumberOfElements(); // the same as getRows()...

 double res[][] = new double[rows][cols];
 for (int i = 0; i < rows; i++) {
  for (int j = 0; j < cols; j++) {
   res[i][j] = crLattice.res(fs1.getValue(i), fs2.getValue(j));
  }
 }

 return new FuzzyRelation(res);
}

public static FuzzyRelation weakBisimulation(FuzzySet fs1, FuzzySet fs2, CRLattice crLattice) {
 // each relation has the same dimension, so we take it from the first one.
 int rows = fs1.getNumberOfElements(); // the same as getCols()...
 int cols = fs2.getNumberOfElements(); // the same as getRows()...

 double res[][] = new double[rows][cols];
 for (int i = 0; i < rows; i++) {
  for (int j = 0; j < cols; j++) {
   res[i][j] = crLattice.biRes(fs1.getValue(i), fs2.getValue(j));
  }
 }

 return new FuzzyRelation(res);
}

public void reduce(int row) {
 double[] reducedValues = new double[numberOfElements - 1];
 for (int i = 0; i < row; i++) {
  reducedValues[i] = this.values[i];
 }
```

```
  for (int i = row + 1; i < numberOfElements; i++) {
   reducedValues[i - 1] = this.values[i];
  }

  this.values = reducedValues;
  this.numberOfElements--;
 }
}
```

Class `Model.java` contains some methods for models and implementations of the functions ϕ^{θ}. Table A.1 gives an overview of the algorithms and functions.

Function ϕ^{θ}	Name of the algorithm or function	Line of the code
ϕ^{fs} (see (3.13))	`phiForwardSimulation`	241
ϕ^{bs} (see (3.14))	`phiBackwardSimulation`	256
ϕ^{fb} (see (3.15))	`phiForwardBisimulation`	271
ϕ^{bb} (see (3.16))	`phiBackwardBisimulation`	277
ϕ^{fbb} (see (3.17))	`phiForwardBackwardBisimulation`	283
ϕ^{bfb} (see (3.18))	`phiBackwardForwardBisimulation`	289
ϕ^{rb} (see (3.19))	`phiRegularBisimulation`	295

Table A.2: Overview of algorithms and functions from the class `Model.java`

Source Code A.30: Class `Model.java`

```
1  package com.logic;
2
3  import java.io.FileNotFoundException;
4  import java.io.IOException;
5  import java.util.ArrayList;
6  import java.util.HashSet;
7  import java.util.List;
8  import java.util.Set;
9
10 public class Model {
11
12  private List<FuzzyRelation> fuzzyRelations;
13  private List<FuzzySet> fuzzySets;
14  private CRLattice crLattice;
15  private Set<Double> algebra;
16
17  public Model(List<FuzzyRelation> fuzzyRelations, List<FuzzySet> fuzzySets, CRLattice
        crLattice) {
18   if (fuzzyRelations == null   fuzzyRelations.isEmpty())
19    throw new IllegalArgumentException("fuzzyRelation must not be null");
20   if (fuzzySets == null   fuzzySets.isEmpty())
21    throw new IllegalArgumentException("fuzzySets must not be null or empty");
22
23   int numOfElements = fuzzySets.get(0).getNumberOfElements();
24   for (int i = 1; i < fuzzySets.size(); i++)
25    if (numOfElements != fuzzySets.get(i).getNumberOfElements())
26     throw new IllegalArgumentException(
27      "All fuzzy sets in fuzzySets parameter must have the same number of elements");
28
29   for (int i = 0; i < fuzzyRelations.size(); i++) {
30    if (fuzzyRelations.get(i).getRows() != numOfElements   fuzzyRelations.get(i).getCols()
          != numOfElements)
31     throw new IllegalArgumentException(
32      String.format("fuzzyRelation must be %sx%s quadratic relation", numOfElements,
          numOfElements));
33   }
34
35   if (crLattice == null)
36    throw new IllegalArgumentException("crLattice must not be null");
37
38   Set<Double> algebra = new HashSet<Double>();
```

```java
  // Adding constants 0 and 1
  algebra.add(0.0);
  algebra.add(1.0);


  this.fuzzyRelations = fuzzyRelations;
  this.fuzzySets = fuzzySets;
  this.crLattice = crLattice;
  this.algebra = algebra;
 }

 public Model(String modelPath, int numberOfWorlds, int numOfRelations, int numOfLetters
     , CRLattice crLattice) throws FileNotFoundException, IOException {
  List<Double> fileNumbers = FileParser.parse(modelPath);
  int fileIndex = 0;

  Set<Double> algebra = new HashSet<Double>();
  algebra.add(0.0);
  algebra.add(1.0);

  List<FuzzyRelation> fuzzyRelations = new ArrayList<FuzzyRelation>();
  for (int i = 0; i < numOfRelations; i++) {
   double[][] values = new double[numberOfWorlds][numberOfWorlds];
   for (int r = 0; r < numberOfWorlds; r++) {
    for (int c = 0; c < numberOfWorlds; c++) {
     Double nextNumber = fileNumbers.get(fileIndex++);
     algebra.add(nextNumber);
     values[r][c] = nextNumber;
    }
   }
   fuzzyRelations.add(new FuzzyRelation(values));
  }

  List<FuzzySet> fuzzySets = new ArrayList<FuzzySet>();
  for (int i = 0; i < numOfLetters; i++) {
   double[] values = new double[numberOfWorlds];
   for (int j = 0; j < numberOfWorlds; j++) {
    Double nextNumber = fileNumbers.get(fileIndex++);
    algebra.add(nextNumber);
    values[j] = nextNumber;
   }
   fuzzySets.add(new FuzzySet(values));
  }

  this.fuzzyRelations = fuzzyRelations;
  this.fuzzySets = fuzzySets;
  this.crLattice = crLattice;
  this.algebra = algebra;
 }

 public static Model disjointModel(Model model1, Model model2) {
  if (model1 == null  model2 == null) {
   return null;
  }

  if (model1.equals(model2)) {
   return model1;
  }

  int model1Dim = model1.getFuzzyRelations().get(0).getCols();
  int model2Dim = model2.getFuzzyRelations().get(0).getCols();
  int disjointDim = model1Dim + model2Dim;

  List<FuzzyRelation> fuzzyRelations = new ArrayList<FuzzyRelation>();
  List<FuzzySet> fuzzySets = new ArrayList<FuzzySet>();

  for (int i = 0; i < model1.getFuzzyRelations().size(); i++) {
   FuzzyRelation fr1 = model1.getFuzzyRelations().get(i);
   FuzzyRelation fr2 = model2.getFuzzyRelations().get(i);

   double[][] frValues = new double[disjointDim][disjointDim];
   for (int k = 0; k < model1Dim; k++) {
    for (int l = 0; l < model1Dim; l++) {
```

```
    frValues[k][l] = fr1.getValue(k, l);
   }
  }
  for (int k = 0; k < model2Dim; k++) {
   for (int l = 0; l < model2Dim; l++) {
    frValues[model1Dim + k][model1Dim + l] = fr2.getValue(k, l);
   }
  }

  fuzzyRelations.add(new FuzzyRelation(frValues));
 }

 for (int i = 0; i < model1.fuzzySets.size(); i++) {
  FuzzySet fs1 = model1.getFuzzySets().get(i);
  FuzzySet fs2 = model2.getFuzzySets().get(i);

  double[] fsValues = new double[disjointDim];
  for (int k = 0; k < model1Dim; k++) {
   fsValues[k] = fs1.getValue(k);
  }
  for (int k = model1Dim; k < disjointDim; k++) {
   fsValues[k] = fs2.getValue(k - model1Dim);
  }

  fuzzySets.add(new FuzzySet(fsValues));
 }

 Model result = new Model(fuzzyRelations, fuzzySets, model1.getCrLattice());
 result.getAlgebra().addAll(model1.getAlgebra());
 result.getAlgebra().addAll(model2.getAlgebra());

 return result;
}

public List<FuzzyRelation> getFuzzyRelations() {
 return fuzzyRelations;
}

public List<FuzzySet> getFuzzySets() {
 return fuzzySets;
}

public CRLattice getCrLattice() {
 return crLattice;
}

public Set<Double> getAlgebra() {
 return algebra;
}

public static FuzzyRelation piFs(Model m1, Model m2) {
 if (m1 == null  m2 == null)
  throw new IllegalArgumentException("m1 and m2 must not be null");
 if (m1.fuzzySets.size() != m2.fuzzySets.size())
  throw new IllegalArgumentException(
    "m1.fuzzySets and m2.fuzzySets must have the same number of elements");
 if (!m1.crLattice.getClass().equals(m2.crLattice.getClass()))
  throw new IllegalArgumentException("m1 and m2 must have the same type of CRLattice");

 int fuzzySetsCount = m1.fuzzySets.size();
 // each relation has the same dimension, so we take it from the first one.
 int rows = m1.fuzzyRelations.get(0).getRows(); // the same as getCols()...
 int cols = m2.fuzzyRelations.get(0).getCols(); // the same as getRows()...

 double res[][] = new double[rows][cols];
 for (int i = 0; i < rows; i++) {
  for (int j = 0; j < cols; j++) {
   double min = Double.MAX_VALUE;
   for (int k = 0; k < fuzzySetsCount; k++) {
    FuzzySet fs1 = m1.fuzzySets.get(k);
    FuzzySet fs2 = m2.fuzzySets.get(k);
    min = Math.min(min, m1.crLattice.res(fs1.getValue(i), fs2.getValue(j)));
   }
```

```
    res[i][j] = min;
   }
  }

  return new FuzzyRelation(res);
 }

 public static FuzzyRelation piBs(Model m1, Model m2) {
  return piFs(m1, m2);
 }

 public static FuzzyRelation piFb(Model m1, Model m2) {
  if (m1 == null  m2 == null)
   throw new IllegalArgumentException("m1 and m2 must not be null");
  if (m1.fuzzySets.size() != m2.fuzzySets.size())
   throw new IllegalArgumentException(
     "m1.fuzzySets and m2.fuzzySets must have the same number of elements");
  if (!m1.crLattice.getClass().equals(m2.crLattice.getClass()))
   throw new IllegalArgumentException("m1 and m2 must have the same type of CRLattice");

  int fuzzySetsCount = m1.fuzzySets.size();
  // each relation has the same dimension, so we take it from the first one.
  int rows = m1.fuzzyRelations.get(0).getRows(); // the same as getCols()...
  int cols = m2.fuzzyRelations.get(0).getCols(); // the same as getRows()...

  double res[][] = new double[rows][cols];
  for (int i = 0; i < rows; i++) {
   for (int j = 0; j < cols; j++) {
    double min = Double.MAX_VALUE;
    for (int k = 0; k < fuzzySetsCount; k++) {
     FuzzySet fs1 = m1.fuzzySets.get(k);
     FuzzySet fs2 = m2.fuzzySets.get(k);
     min = Math.min(min, m1.crLattice.biRes(fs1.getValue(i), fs2.getValue(j)));
    }
    res[i][j] = min;
   }
  }

  return new FuzzyRelation(res);
 }

 public static FuzzyRelation piFbb(Model m1, Model m2) {
  return piFb(m1, m2);
 }

 public static FuzzyRelation piBfb(Model m1, Model m2) {
  return piFb(m1, m2);
 }

 public static FuzzyRelation piBb(Model m1, Model m2) {
  return piFb(m1, m2);
 }

 public static FuzzyRelation piRb(Model m1, Model m2) {
  return piFb(m1, m2);
 }

 public static FuzzyRelation phiForwardSimulation(Model m1, Model m2, FuzzyRelation fr)
    {
  FuzzyRelation composition = m2.fuzzyRelations.get(0).compose(fr.transpose(), m1.
      crLattice);
  FuzzyRelation leftR = FuzzyRelation.leftResidual(m1.fuzzyRelations.get(0), composition
      , m1.crLattice);
  FuzzyRelation result = leftR.transpose();

  for (int i = 1; i < m2.getFuzzyRelations().size(); i++) {
   composition = m2.fuzzyRelations.get(i).compose(fr.transpose(), m1.crLattice);
   leftR = FuzzyRelation.leftResidual(m1.fuzzyRelations.get(i), composition, m1.
       crLattice);
   FuzzyRelation iResult = leftR.transpose();

   result = FuzzyRelation.conjunction(result, iResult);
  }
```

```java
        return result;
    }

    public static FuzzyRelation phiBackwardSimulation(Model m1, Model m2, FuzzyRelation fr) {
        FuzzyRelation composition = fr.compose(m2.fuzzyRelations.get(0), m1.crLattice);
        FuzzyRelation rightR = FuzzyRelation.rightResidual(m1.fuzzyRelations.get(0), composition, m1.crLattice);
        FuzzyRelation result = rightR;

        for(int i = 1; i < m2.getFuzzyRelations().size(); i++) {
            composition = fr.compose(m2.fuzzyRelations.get(i), m1.crLattice);
            rightR = FuzzyRelation.rightResidual(m1.fuzzyRelations.get(i), composition, m1.crLattice);
            FuzzyRelation iResult = rightR;

            result = FuzzyRelation.conjunction(result, iResult);
        }
        return result;
    }

    public static FuzzyRelation phiForwardBisimulation(Model m1, Model m2, FuzzyRelation fr) {
        FuzzyRelation phiFs = phiForwardSimulation(m1, m2, fr);
        FuzzyRelation phiFsTransposed = phiForwardSimulation(m2, m1, fr.transpose()).transpose();
        return FuzzyRelation.conjunction(phiFs, phiFsTransposed);
    }

    public static FuzzyRelation phiBackwardBisimulation(Model m1, Model m2, FuzzyRelation fr) {
        FuzzyRelation phiBs = phiBackwardSimulation(m1, m2, fr);
        FuzzyRelation phiBsTransposed = phiBackwardSimulation(m2, m1, fr.transpose()).transpose();
        return FuzzyRelation.conjunction(phiBs, phiBsTransposed);
    }

    public static FuzzyRelation phiForwardBackwardBisimulation(Model m1, Model m2, FuzzyRelation fr) {
        FuzzyRelation phiFs = phiForwardSimulation(m1, m2, fr);
        FuzzyRelation phiBsTransposed = phiBackwardSimulation(m2, m1, fr.transpose()).transpose();
        return FuzzyRelation.conjunction(phiFs, phiBsTransposed);
    }

    public static FuzzyRelation phiBackwardForwardBisimulation(Model m1, Model m2, FuzzyRelation fr) {
        FuzzyRelation phiBs = phiBackwardSimulation(m1, m2, fr);
        FuzzyRelation phiFsTransposed = phiForwardSimulation(m2, m1, fr.transpose()).transpose();
        return FuzzyRelation.conjunction(phiBs, phiFsTransposed);
    }

    public static FuzzyRelation phiRegularBisimulation(Model m1, Model m2, FuzzyRelation fr) {
          FuzzyRelation phiFb = phiForwardBisimulation(m1, m2, fr);
          FuzzyRelation phiFbTransposed = phiForwardBisimulation(m2, m1, fr.transpose()).transpose();

          FuzzyRelation phiBb = phiBackwardBisimulation(m1, m2, fr);
          FuzzyRelation phiBbTransposed = phiBackwardBisimulation(m2, m1, fr.transpose()).transpose();

          return FuzzyRelation.conjunction(FuzzyRelation.conjunction(phiFb, phiFbTransposed), FuzzyRelation.conjunction(phiBb, phiBbTransposed));
    }
}
```

Source Code A.31: Class `TSet.java`

```java
package com.logic;

import java.util.ArrayList;
```

```java
import java.util.List;

import com.logic.operations.BinaryOperation;
import com.logic.operations.UnaryOperation;

public class TSet {

 private List<FuzzyFormula> formulaList;

 public TSet() {
  this.formulaList = new ArrayList<FuzzyFormula>();
 }

 public TSet(List<FuzzyFormula> formulaList) {
  this.formulaList = formulaList;
 }

 public static TSet generate(FSet fSet) {
  List<FuzzyFormula> formulas = new ArrayList<FuzzyFormula>();
  for (FuzzyFormula formula : fSet.getFormulaList()) {
   if (!formulas.contains(formula)) {
    formulas.add(formula);
   }
  }

  return new TSet(formulas);
 }

 public static TSet generate(List<TSet> tSetList, List<UnaryOperation> unaryOperations,
     List<BinaryOperation> binaryOperations, CRLattice crLattice) {
  TSet tSetPrevious = tSetList.get(tSetList.size() - 1);

  List<FuzzyFormula> formulas = new ArrayList<FuzzyFormula>();

  for (int i = 0; i < unaryOperations.size(); i++) {
   for (int j = 0; j < tSetPrevious.getFormulaList().size(); j++) {
    FuzzyFormula formula = tSetPrevious.getFormulaList().get(j);
    FuzzyFormula newFormula = unaryOperations.get(i).apply(formula, crLattice);
    if (shouldAdd(newFormula, tSetList, formulas)) {
     formulas.add(newFormula);
    }
   }
  }

  for (int i = 0; i < tSetList.size(); i++) {
   int j = tSetList.size() - 1 - i;
   TSet tSetI = tSetList.get(i);
   TSet tSetJ = tSetList.get(j);

   List<FuzzyFormula> formulasI = tSetI.getFormulaList();
   List<FuzzyFormula> formulasJ = tSetJ.getFormulaList();

   for (int k = 0; k < formulasI.size(); k++) {
    for (int l = 0; l < formulasJ.size(); l++) {
     for (BinaryOperation binaryOperation : binaryOperations) {
      FuzzyFormula fI1 = formulasI.get(k);
      FuzzyFormula fJ1 = formulasJ.get(l);
      FuzzyFormula newFormula = binaryOperation.apply(fI1, fJ1, crLattice);
      if (shouldAdd(newFormula, tSetList, formulas)) {
       formulas.add(newFormula);
      }
     }
    }
   }
  }

  return new TSet(formulas);
 }

 private static boolean shouldAdd(FuzzyFormula newFormula, List<TSet> tSetList, List<
     FuzzyFormula> currentFormulas) {
  for (TSet tSet: tSetList) {
   if (tSet.getFormulaList().contains((newFormula))) {
```

```
    return false;
   }
  }

  if (currentFormulas.contains(newFormula)) {
   return false;
  }

  return true;
 }

 public List<FuzzyFormula> getFormulaList() {
  return formulaList;
 }
}
```

List of Abbreviations

FLTS Fuzzy Labelled Transition System. 51

fs forward simulation. 53

G

G Idempotency. 18

H

hcf Highest Common Factor. 16

HML Hennessy-Milner logic. 87

I

Inv Involution. 18

L

lcm Least Common Multiple. 16

LTL Linear Temporal Logic. 128

LTS Labelled Transition System. 4

M

ML Monoidal Logic. 17

P

PFF partial fuzzy function; If a fuzzy relation $\varphi \in \mathscr{R}(A, B)$ is extensional with respect to E and F, and it satisfies $\varphi(a, b_1) \wedge \varphi(a, b_2) \leqslant F(b_1, b_2)$, for all $a \in A$ and $b_1, b_2 \in B$, then it is called a *partial fuzzy function* with respect to E and F. 34

Prl Pre-linearity condition. 18

PTA Paige and Tarjan algorithm. 115

R

rb regular bisimulation. 54

U

ULC The set of Unary Logical Connectives. 44

uwb uniform weak bisimulation. 105

uws uniform weak simulation. 105

W

WΨB Weak Ψ-bisimulation. 111

wb weak bisimulation. 89

WFF well-formed formula. 39

WNM Weak Nilpotent Minimum. 18

ws weak simulation. 89

List of Symbols

ϱ^{fs} the greatest crisp forward simulation. 68

ϱ^{fbb} the greatest crisp forward-backward bisimulation. 68

ϱ_*^{fbb} the greatest crisp forward-backward prebisimulation. 68

ϱ^{rb} the greatest crisp regular bisimulation. 68

ϱ_*^{rb} the greatest crisp regular prebisimulation. 68

E

$\mathscr{E}(A)$ the set of all (fuzzy) equivalences on A. 16

F

$\Phi_{I,\mathscr{H}}$ set of formulae defined via the grammar $A ::= \bar{t} \mid p \mid A \wedge A \mid A \to A \mid \Box_i A \mid \Diamond_i A \mid \Box_i^- A \mid \Diamond_i^- A$, where $\bar{t} \in \overline{H}$, i is from some finite set of indices I and p ranges over set PV of proposition variables. Letter $\mathscr{H}$ indicates that the underlying structure is Heyting algebra. 40

$\Phi_{I,\mathscr{H}}^{\Diamond}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i$ and $\Box_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{\Diamond-}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i$, $\Box_i^-$ and $\Diamond_i$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{\Diamond+}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i$, $\Box_i^-$ and $\Diamond_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{-}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i$ and $\Diamond_i$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{+}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without modal operators $\Box_i^-$ and $\Diamond_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{PF}$ fragment of $\Phi_{I,\mathscr{H}}$ without modal operators. 43

$\Phi_{I,\mathscr{H}}^{\Box}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Diamond_i$ and $\Diamond_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{\Box-}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i$, $\Diamond_i$ and $\Diamond_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{H}}^{\Box+}$ fragment of $\Phi_{I,\mathscr{H}}$ defined without operators $\Box_i^-$, $\Diamond_i$ and $\Diamond_i^-$, $i \in I$. 42

$\Phi_{I,\mathscr{L}}$ set of formulae defined via the grammar $A ::= \bar{t} \mid p \mid A \wedge A \mid A \otimes A \mid A \to A \mid \Box_i A \mid \Diamond_i A \mid \Box_i^- A \mid \Diamond_i^- A$, where $\bar{t} \in \overline{L}$, i is from some finite set of indices I and p ranges over set PV of proposition variables. Letter $\mathscr{L}$ indicates that the underlying structure is residuated lattice. 143

$\Phi_{I,\mathscr{L}}^{\Diamond}$ fragment of $\Phi_{I,\mathscr{L}}$ defined without operators $\Box_i$ and $\Box_i^-$, $i \in I$. 144

$\Phi_{I,\mathscr{L}}^{\Diamond-}$ fragment of $\Phi_{I,\mathscr{L}}$ defined without operators $\Box_i$, $\Box_i^-$ and $\Diamond_i$, $i \in I$. 144

$\Phi_{I,\mathscr{L}}^{\Diamond+}$ fragment of $\Phi_{I,\mathscr{L}}$ defined without operators $\Box_i$, $\Box_i^-$ and $\Diamond_i^-$, $i \in I$. 144

$\Phi_{I,\mathscr{L}}^{-}$ fragment of $\Phi_{I,\mathscr{L}}$ defined without operators $\Box_i$ and $\Diamond_i$, $i \in I$. 144

$\Phi_{I,\mathscr{L}}^{+}$ fragment of $\Phi_{I,\mathscr{L}}$ defined without modal operators $\Box_i^-$ and $\Diamond_i^-$, $i \in I$. 144

$\Phi_{I,\mathscr{L}}^{PF}$ fragment of $\Phi_{I,\mathscr{L}}$ without modal operators. 144

$\otimes$ triangular norm (t-norm). 19

U

∇_A universal relation on the set A. 8

W

φ^{wb} the greatest weak bisimulation. 89

φ_*^{wb} the greatest weak prebisimulation. 89

φ_*^{ws} the greatest weak presimulation. 89

$\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}'$ If there exists a complete and surjective weak Ψ-bisimulation from $\mathfrak{M}$ to $\mathfrak{M}'$ then we say that $\mathfrak{M}$ and $\mathfrak{M}'$ are weak Ψ-bisimulation equivalent for the set Ψ, or briefly WΨB-equivalent, and we write $\mathfrak{M} \sim_{W\Psi B} \mathfrak{M}'$. 111

$\mathbb{W}\Psi\mathbb{B}(\mathfrak{M})$ a class of all Kripke models which are WΨB-equivalent to $\mathfrak{M}$. 112

φ^{ws} the greatest weak simulation. 89

Index

Bibliography

[1] R. J. Adillon and V. Verdú, "On a Contraction-Less Intuitionistic Propositional Logic with Conjunction and Fusion," *Studia Logica*, vol. 65, no. 1, pp. 11–30, 2000. DOI: 10.1023/A:1005238908087 (cit. on p. 17).

[2] M. Baaz, "Infinite-valued Gödel logics with 0-1-projections and relativizations," in *Gödel'96: Logical foundations of mathematics, computer science and physics—Kurt Gödel's legacy, Brno, Czech Republic, August 1996, proceedings*, vol. 6, Association for Symbolic Logic, 1996, pp. 23–34 (cit. on p. 38).

[3] M. Baczyński and B. Jayaram, *Fuzzy Implications* (Studies in Fuzziness and Soft Computing). Springer Berlin, Heidelberg, 2008, vol. 231. DOI: 10.1007/978-3-540-69082-5 (cit. on p. 147).

[4] R. Balbes and P. Dwinger, "Distributive lattices," 1974 (cit. on p. 24).

[5] R. Bělohlávek, *Fuzzy Relational Systems: Foundations and Principles*. New York: Kluwer Academic/Plenum Publishers, 2002. DOI: 10.1007/978-1-4615-0633-1 (cit. on pp. 29, 32, 143).

[6] R. Bělohlávek, J. W. Dauben, and G. J. Klir, *Fuzzy Logic and Mathematics: A Historical Perspective*. Oxford University Press, 2017 (cit. on pp. 2, 3).

[7] R. Bělohlávek and V. Vychodil, *Fuzzy Equational Logic*. Berlin Heidelberg New York: Springer, 2005. DOI: 10.1007/b105121 (cit. on pp. 7, 22, 29, 143).

[8] J. van Benthem, "Modal Correspondence Theory," Ph.D. dissertation, Universiteit van Amsterdam, 1976 (cit. on p. 3).

[9] M. Bílková and M. Dostál, "Expressivity of Many-Valued Modal Logics, Coalgebraically," in *International Workshop on Logic, Language, Information, and Computation*, J. Väänänen, Å. Hirvonen, and R. de Queiroz, Eds., Springer Berlin Heidelberg, 2016, pp. 109–124. DOI: 10.1007/978-3-662-52921-8_8 (cit. on p. 88).

[10] G. Birkhoff, *Lattice Theory* (Colloquium publications), 3rd ed. American Mathematical Society, 1967 (cit. on pp. 7, 13).

[11] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*. Cambridge: Cambridge University Press, 2001. DOI: 10.1017/CBO9781107050884 (cit. on pp. 39, 42, 87, 101).

[12] T. S. Blyth, *Lattices and Ordered Algebraic Structures*. Springer, 2005. DOI: 10.1007/b139095 (cit. on pp. 7, 13, 24).

[13] F. Bobillo, M. Delgado, J. Gómez-Romero, and U. Straccia, "Fuzzy description logics under Gödel semantics," *International Journal of Approximate Reasoning*, vol. 50, no. 3, pp. 494–514, 2009, Special Section on Bayesian Modelling. DOI: 10.1016/j.ijar.2008.10.003 (cit. on p. 5).

[14] F. Bobillo, M. Delgado, J. Gómez-Romero, and U. Straccia, "Joining Gödel and Zadeh fuzzy logics in fuzzy description logics," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 20, no. 04, pp. 475–508, 2012. DOI: 10.1142/S0218488512500249 (cit. on p. 5).

[15] F. Bobillo and U. Straccia, "Fuzzydl: An Expressive Fuzzy Description Logic Reasoner," in *2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence)*, 2008, pp. 923–930. DOI: 10.1109/FUZZY.2008.4630480 (cit. on p. 147).

[16] F. Bou, F. Esteva, L. Godo, and R. O. Rodríguez, "On the Minimum Many-Valued Modal Logic over a Finite Residuated Lattice," *Journal of Logic and Computation*, vol. 21, no. 5, pp. 739–790, 2011. DOI: 10.1093/logcom/exp062 (cit. on pp. 3, 37, 38, 43).

[17] S. Burris and H. P. Sankappanavar, *A Course in Universal Algebra (Graduate Texts in Mathematics)*. Springer-Verlag, 1981, The Millennium Edition (cit. on pp. 7, 10, 50).

[18] X. Caicedo and R. O. Rodriguez, "Standard Gödel Modal Logics," *Studia Logica*, vol. 94, no. 2, pp. 189–214, 2010. DOI: 10.1007/s11225-010-9230-1 (cit. on pp. 37, 38).

[19] X. Caicedo and R. O. Rodriguez, "Bi-modal Gödel logic over $[0, 1]$-valued Kripke frames," *Journal of Logic and Computation*, vol. 25, pp. 37–55, 1 2015. DOI: 10.1093/logcom/exs036 (cit. on p. 37).

[20] Y. Cao, G. Chen, and E. E. Kerre, "Bisimulations for Fuzzy-Transition Systems," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 3, pp. 540–552, 2011. DOI: 10.1109/TFUZZ.2011.2117431 (cit. on p. 51).

[21] C. C. Chang, "Algebraic analysis of many valued logics," *Transactions of the American Mathematical Society*, vol. 88, pp. 467–490, 1958. DOI: 10.1090/S0002-9947-1958-0094302-9 (cit. on p. 146).

[22] C. C. Chang, "A New Proof of the Completeness of the Lukasiewicz Axioms," *Transactions of the American Mathematical Society*, vol. 93, no. 1, pp. 74–80, 1959. DOI: 10.2307/1993423 (cit. on p. 146).

[23] B. F. Chellas, *Modal logic: an introduction*. Cambridge University Press, 1980 (cit. on p. 39).

[24] M. Ćirić, J. Ignjatović, and S. Bogdanović, "Uniform fuzzy relations and fuzzy functions," *Fuzzy Sets and Systems*, vol. 160, pp. 1054–1081, 8 2009. DOI: 10.1016/j.fss.2008.07.006 (cit. on pp. 6, 7, 34, 35, 88).

[25] M. Ćirić, J. Ignjatović, N. Damljanović, and M. Bašić, "Bisimulations for fuzzy automata," *Fuzzy Sets and Systems*, vol. 186, no. 1, pp. 100–139, 2012. DOI: 10.1016/j.fss.2011.07.003 (cit. on pp. 51, 52, 55, 110).

[26] M. Ćirić, J. Ignjatović, I. Jančić, and N. Damljanović, "Computation of the greatest simulations and bisimulations between fuzzy automata," *Fuzzy Sets and Systems*, vol. 208, pp. 22–42, 2012. DOI: 10.1016/j.fss.2012.05.006 (cit. on pp. 51, 57, 61, 143, 145).

[27] M. Ćirić, J. Ignjatović, and I. Stanković, "Direct and Indirect Methods for Solving Two-Mode Systems of Fuzzy Relation Equations and Inequalities," *Trends in Mathematics and Computational Intelligence. Studies in Computational Intelligence*, vol. 796, M. E. Cornejo, L. T. Kóczy, J. Medina, and A. E. De Barros Ruano, Eds., pp. 155–165, 2019. DOI: 10.1007/978-3-030-00485-9_18 (cit. on p. 33).

[28] M. Ćirić, A. Stamenković, J. Ignjatović, and T. Petković, "Fuzzy relation equations and reduction of fuzzy automata," *Journal of Computer and System Sciences*, vol. 76, pp. 609–633, 7 2010. DOI: 10.1016/j.jcss.2009.10.015 (cit. on pp. 24, 143).

[29] W. Conradie, D. Della Monica, E. Muñoz-Velasco, and G. Sciavicco, "An approach to fuzzy modal logic of time intervals," in *ECAI 2020*, ser. Frontiers in Artificial Intelligence and Applications, vol. 325, IOS Press, 2020, pp. 696–703. DOI: 10.3233/FAIA200156 (cit. on p. 5).

[30] N. Damljanović, "Multivalued relations over lattices and semirings: Theory and applications," (in Serbian), Ph.D. dissertation, University of Niš, Faculty of Sciences and Mathematics, 2012 (cit. on p. 6).

[31] B. A. Davey and H. A. Priestley, *Introduction to Lattices and Order*, 2nd ed. Cambridge University Press, 2002. DOI: 10.1017/CBO9780511809088 (cit. on pp. 7, 13, 17).

[32] L. D'eer, N. Verbiest, C. Cornelis, and L. Godo, "A comprehensive study of implicator-conjunctor-based and noise-tolerant fuzzy rough sets: Definitions, properties and robustness analysis," *Fuzzy Sets and Systems*, vol. 275, pp. 1–38, 2015, Theme: Fuzzy Relations, ISSN: 0165-0114. DOI: 10.1016/j.fss.2014.11.018 (cit. on p. 147).

[33] M. Demirci, "Fuzzy functions and their applications," *Journal of Mathematical Analysis and Applications*, vol. 252, no. 1, pp. 495–517, 2000. DOI: 10.1006/jmaa.2000.7185 (cit. on p. 35).

[34] M. Demirci, "Foundations of fuzzy functions and vague algebra based on many-valued equivalence relations, part I: Fuzzy functions and their applications," *International Journal of General Systems*, vol. 32, no. 2, pp. 123–155, 2003. DOI: 10.1080/0308107031000090765 (cit. on p. 35).

[35] M. Demirci, "A theory of vague lattices based on many-valued equivalence relations—I: General representation results," *Fuzzy Sets and Systems*, vol. 151, no. 3, pp. 437–472, 2005, ISSN: 0165-0114. DOI: 10.1016/j.fss.2004.06.017 (cit. on p. 35).

[36] D. Diaconescu, "Modal Equivalence and Bisimilarity in Many-valued Modal Logics with Many-valued Accessibility Relations," *Fundamenta Informaticae*, vol. 173, no. 2-3, pp. 177–189, 2020. DOI: 10.3233/FI-2020-1920 (cit. on pp. 37, 42, 88).

[37] A. Dovier, C. Piazza, and A. Policriti, "An efficient algorithm for computing bisimulation equivalence," *Theoretical Computer Science*, vol. 311, no. 1, pp. 221–256, 2004. DOI: 10.1016/S0304-3975(03)00361-X (cit. on p. 115).

[38] R. G. Downey and M. R. Fellows, *Parameterized Complexity* (Monographs in Computer Science). Springer Science & Business Media, 2012. DOI: 10.1007/978-1-4612-0515-9 (cit. on p. 127).

[39] D. Dubois, F. Esteva, L. Godo, and H. Prade, "Fuzzy-set Based Logics–An History-oriented Presentation of their Main Developments," *The Many Valued and Nonmonotonic Turn in Logic*, vol. 8, D. M. Gabbay and J. Woods, Eds., pp. 325–449, 2007 (cit. on p. 17).

[40] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets*," *International Journal of General Systems*, vol. 17, no. 2-3, pp. 191–209, 1990. DOI: 10.1080/03081079008935107 (cit. on p. 147).

[41] D. Dubois and H. Prade, "Putting Rough Sets and Fuzzy Sets Together," in *Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory*, R. Słowiński, Ed. Dordrecht: Springer Netherlands, 1992, pp. 203–232. DOI: 10.1007/978-94-015-7975-9_14 (cit. on p. 147).

[42] P. E. Eleftheriou, C. D. Koutras, and C. Nomikos, "Notions of Bisimulation for Heyting-Valued Modal Languages," *Journal of Logic and Computation*, vol. 22, no. 2, pp. 213–235, Feb. 2010. DOI: 10.1093/logcom/exq005 (cit. on pp. 37, 51, 88).

[43] E. A. Emerson and C.-L. Lei, "Modalities for model checking: Branching time logic strikes back," *Science of Computer Programming*, vol. 8, no. 3, pp. 275–306, 1987. DOI: 10.1016/0167-6423(87)90036-0 (cit. on p. 128).

[44] F. Esteva, J. Gispert, L. Godo, and C. Noguera, "Adding truth-constants to logics of continuous t-norms: Axiomatization and completeness results," *Fuzzy Sets and Systems*, vol. 158, no. 6, pp. 597–618, 2007, The Logic of Soft Computing. DOI: 10.1016/j.fss.2006.11.010 (cit. on p. 38).

[45] F. Esteva and L. Godo, "Monoidal t-norm based logic: Towards a logic for left-continuous t-norms," *Fuzzy Sets and Systems*, vol. 124, no. 3, pp. 271–288, 2001, Fuzzy Logic. DOI: 10.1016/S0165-0114(01)00098-7 (cit. on pp. 17, 21).

[46] F. Esteva, L. Godo, and C. Noguera, "Expanding the propositional logic of a t-norm with truth-constants: Completeness results for rational semantics," *Soft Computing*, vol. 14, no. 3, pp. 273–284, 2010. DOI: 10.1007/s00500-009-0402-8 (cit. on p. 38).

[47] T.-F. Fan, "Fuzzy bisimulation for Gödel modal logic," *IEEE Transactions on Fuzzy Systems*, vol. 23, pp. 2387–2396, 6 Dec. 2015. DOI: 10.1109/TFUZZ.2015.2426724 (cit. on pp. 37, 51, 55, 87, 98).

[48] T.-F. Fan and C.-J. Liau, "Many-Valued Modal Logic and Regular Equivalences in Weighted Social Networks," in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty. ECSQARU 2013.*, L. C. van der Gaag, Ed., ser. Lecture Notes in Computer Science, vol. 7958, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 194–205. DOI: 10.1007/978-3-642-39091-3_17 (cit. on p. 5).

[49] T.-F. Fan and C.-J. Liau, "Logical characterizations of regular equivalence in weighted social networks," *Artificial Intelligence*, vol. 214, pp. 66–88, 2014. DOI: 10.1016/j.artint.2014.05.007 (cit. on p. 5).

[50] M. C. Fitting, "Many-Valued Modal Logics," *Fundamenta Informaticae*, vol. 15, no. 3-4, pp. 235–254, 1991. DOI: 10.3233/FI-1991-153-404 (cit. on pp. 3, 37).

[51] M. C. Fitting, "Many-Valued Modal Logics II," *Fundamenta Informaticae*, vol. 17, no. 1-2, pp. 55–73, 1992. DOI: 10.3233/FI-1992-171-205 (cit. on pp. 3, 37, 43).

[52] J. Fodor, "Left-continuous t-norms in fuzzy logic: An overview," *Acta Polytechnica Hungarica*, vol. 1, no. 2, pp. 35–47, 2004 (cit. on p. 147).

[53] J. C. Fodor, "Nilpotent minimum and related connectives for fuzzy logic," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, IEEE, vol. 4, 1995, pp. 2077–2082. DOI: 10.1109/FUZZY.1995.409964 (cit. on p. 21).

[54] M. Forti and F. Honsell, "Set theory with free construction principles," en, *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, vol. Ser. 4, 10, no. 3, pp. 493–522, 1983 (cit. on p. 3).

[55] N. Galatos, P. Jipsen, T. Kowalski, and H. Ono, *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Amsterdam: Elsevier Science, 2007 (cit. on p. 17).

[56] R. Gentilini, C. Piazza, and A. Policriti, "From Bisimulation to Simulation: Coarsest Partition Problems," *Journal of Automated Reasoning*, vol. 31, no. 1, pp. 73–103, 2003. DOI: 10.1023/A:1027328830731 (cit. on p. 115).

[57] J. Gispert, "Finitary Extensions of the Nilpotent Minimum Logic and (Almost) Structural Completeness," *Studia Logica*, vol. 2106, pp. 789–808, 2018. DOI: 10.1007/s11225-017-9766-4 (cit. on p. 21).

[58] J. A. Goguen, "L-Fuzzy Sets," *Journal of Mathematical Analysis and Applications*, vol. 18, no. 1, pp. 145–174, 1967. DOI: 10.1016/0022-247X(67)90189-8 (cit. on p. 3).

[59] Q.-T. Ha, L. A. Nguyen, T. H. K. Nguyen, and T.-L. Tran, "Fuzzy Bisimulations in Fuzzy Description Logics Under the Gödel Semantics," in *Rough Sets*, H. S. Nguyen, Q.-T. Ha, T. Li, and M. Przybyła-Kasperek, Eds., Cham: Springer International Publishing, 2018, pp. 559–571. DOI: 10.1007/978-3-319-99368-3_44 (cit. on p. 51).

[60] M. Habib, C. Paul, and L. Viennot, "Partition refinement techniques: An interesting algorithmic tool kit," *International Journal of Foundations of Computer Science*, vol. 10, no. 02, pp. 147–170, 1999. DOI: 10.1142/S0129054199000125 (cit. on p. 115).

[61] P. Hájek, "Making fuzzy description logic more general," *Fuzzy Sets and Systems*, vol. 154, no. 1, pp. 1–15, 2005. DOI: 10.1016/j.fss.2005.03.005 (cit. on p. 5).

[62] G. Hansoul and B. Teheux, "Extending Łukasiewicz Logics with a Modality: Algebraic Approach to Relational Semantics," *Studia Logica*, vol. 101, no. 3, pp. 505–545, 2013. DOI: 10.1007/s11225-012-9396-9 (cit. on p. 37).

[63] M. Hennessy and R. Milner, "On observing nondeterminism and concurrency," in *Automata, Languages and Programming. ICALP 1980. Lecture Notes in Computer Science*, J. de Bakker and J. van Leeuwen, Eds., vol. 85, Berlin, Heidelberg: Springer Berlin Heidelberg, 1980, pp. 299–309. DOI: 10.1007/3-540-10003-2_79 (cit. on p. 87).

[64] M. Hennessy and R. Milner, "Algebraic Laws for Nondeterminism and Concurrency," *Journal of the ACM*, vol. 32, no. 1, pp. 137–161, 1985. DOI: 10.1145/2455.2460 (cit. on p. 87).

[65] A. Heyting, "Die formalen regeln der intuitionistischen logik," *Sitzungsbericht PreuBische Akademie der Wissenschaften Berlin, physikalisch-mathematische Klasse II*, pp. 42–56, 1930 (cit. on p. 24).

[66] U. Höhle, "Commutative, residuated l–monoids," in *Non-Classical Logics and their Applications to Fuzzy Subsets: A Handbook of the Mathematical Foundations of Fuzzy Set Theory*, U. Höhle and E. P. Klement, Eds. Dordrecht: Springer Netherlands, 1995, pp. 53–106. DOI: 10.1007/978-94-011-0215-5_5 (cit. on p. 17).

[67] J. Ignjatović, "Fuzzy relations, automata and languages," (in Serbian), Ph.D. dissertation, University of Niš, Faculty of Sciences and Mathematics, 2007 (cit. on p. 32).

[68] J. Ignjatović, M. Ćirić, and S. Bogdanović, "On the greatest solutions to weakly linear systems of fuzzy relation inequalities and equations," *Fuzzy Sets and Systems*, vol. 161, pp. 3081–3113, 24 2010. DOI: 10.1016/j.fss.2010.08.002 (cit. on pp. 58, 61, 67, 143).

[69] J. Ignjatović, M. Ćirić, N. Damljanović, and I. Jančić, "Weakly linear systems of fuzzy relation inequalities: The heterogeneous case," *Fuzzy Sets and Systems*, vol. 199, pp. 64–91, 2012. DOI: 10.1016/j.fss.2011.11.011 (cit. on p. 61).

[70] J. Ignjatović, M. Ćirić, B. Šešelja, and A. Tepavčević, "Fuzzy relational inequalities and equations, fuzzy quasi-orders, closures and openings of fuzzy sets," *Fuzzy Sets and Systems*, vol. 260, pp. 1–24, 2015. DOI: 10.1016/j.fss.2014.05.006 (cit. on pp. 32, 143).

[71] J. Ignjatović, M. Ćirić, and I. Stanković, "Bisimulations in fuzzy social network analysis," in *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*, Atlantis Press, 2015, pp. 404–411. DOI: 10.2991/ifsa-eusflat-15.2015.59 (cit. on pp. 5, 53, 54).

[72] M. Jain, A. Madeira, and M. A. Martins, "A Fuzzy Modal Logic for Fuzzy Transition Systems," *Electronic Notes in Theoretical Computer Science*, vol. 348, pp. 85–103, 2020, 14th International Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2019). DOI: 10.1016/j.entcs.2020.02.006 (cit. on p. 51).

[73] I. Jančić, "Weak bisimulations for fuzzy automata," *Fuzzy Sets and Systems*, vol. 249, pp. 49–72, 2014, Theme: Automata Theory and Algebra. DOI: 10.1016/j.fss.2013.10.006 (cit. on pp. 6, 88, 108, 143).

[74] D. de Jongh and F. Yang, "Jankov's theorems for intermediate logics in the setting of universal models," in *Logic, Language, and Computation - 8th International Tbilisi Symposium on Logic, Language, and Computation, TbiLLC 2009, Bakuriani, Georgia, September 21-25, 2009. Revised Selected Papers*, N. Bezhanishvili, S. Löbner, K. Schwabe, and L. Spada, Eds., ser. Lecture Notes in Computer Science, vol. 6618, Springer, 2009, pp. 53–76. DOI: 10.1007/978-3-642-22303-7_5 (cit. on p. 41).

[75] P. C. Kanellakis and S. A. Smolka, "Ccs expressions, finite state processes, and three problems of equivalence," *Information and Computation*, vol. 86, no. 1, pp. 43–68, 1990. DOI: 10.1016/0890-5401(90)90025-D (cit. on p. 115).

[76] E. P. Klement, R. Mesiar, and E. Pap, "Triangular norms. Position paper I: Basic analytical and algebraic properties," *Fuzzy Sets and Systems*, vol. 143, no. 1, pp. 5–26, 2004, Advances in Fuzzy Logic. DOI: 10.1016/j.fss.2003.06.007 (cit. on p. 147).

[77] B. Knaster, "Un théorème sur les functions d'ensembles," *Annales de la Société Polonaise de Mathématique*, vol. 6, pp. 133–134, 1928 (cit. on p. 17).

[78] S. A. Kripke, "Semantical Analysis of Modal Logic I Normal Modal Propositional Calculi," *Mathematical Logic Quarterly*, vol. 9, no. 5-6, pp. 67–96, 1963. DOI: 10.1002/malq.19630090502 (cit. on p. 1).

[79] S. A. Kripke, "Semantical Considerations on Modal Logic," *Acta Philosophica Fennica*, vol. 16, pp. 83–94, 1963 (cit. on p. 1).

[80] C. I. Lewis, *Collected Papers of Clarence Irving Lewis*, J. D. Goheen and J. L. Mothershead Jr, Eds. Stanford University Press, 1970 (cit. on p. 1).

[81] O. Lichtenstein and A. Pnueli, "Checking That Finite State Concurrent Programs Satisfy Their Linear Specification," in *Proceedings of the 12th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, ser. POPL '85, New York, NY, USA: Association for Computing Machinery, 1985, pp. 97–107. DOI: 10.1145/318593.318622 (cit. on p. 128).

[82] M. Lück, "Parameterized Complexity of Temporal Logics," M.S. thesis, Leibniz Universität Hannover, Fakultät für Elektrotechnik und Informatik, Institut für Theoretische Informatik, Hannover, 2015 (cit. on p. 128).

[83] J. Łukasiewicz, “Interpretacja liczbowa teorii zdań,” *Ruch Filozoficzny*, vol. 7, pp. 92–93, 1923, English translation (‘A numerical interpretation of the theory of propositions’) by O. Wojtasiewicz in Jan Łukasiewicz Selected works, edited by L. Borkowski, 1970, pp. 129-130. (cit. on p. 2).

[84] J. Łukasiewicz and A. Tarski, “Untersuchungen über den aussagenkalkül,” *Comptes rendus des séances de la Société des Sciences et des Lettres de Varsovie, Classe III*, vol. 23, pp. 1–21, 1930, English translation (‘Investigations into the Sentential Calculus’) by J. H. Woodger in Alfred Tarski, Logic, Semantics, Meta-Mathematics, edited by J. H. Woodger, Oxford (Clarendon Press) 1956, pp. 38-59. (cit. on p. 2).

[85] S. Mac Lane and G. Birkhoff, *Algebra*. American Mathematical Society, 1999, vol. 330 (cit. on p. 10).

[86] M. Marti and G. Metcalfe, “Expressivity in chain-based modal logics,” *Archive for mathematical logic*, vol. 57, pp. 361–380, 2018. DOI: 10.1007/s00153-017-0573-4 (cit. on p. 88).

[87] M. Mas, M. Monserrat, and J. Torrens, “On two types of discrete implications,” *International Journal of Approximate Reasoning*, vol. 40, no. 3, pp. 262–279, 2005. DOI: 10.1016/j.ijar.2005.05.001 (cit. on p. 147).

[88] J. C. C. McKinsey and A. Tarski, “On Closed Elements in Closure Algebras,” *Annals of Mathematics*, vol. 47, no. 1, pp. 122–162, 1946. DOI: 10.2307/1969038 (cit. on p. 24).

[89] S. Merz, “Model Checking: A Tutorial Overview,” in *Modeling and Verification of Parallel Processes: 4th Summer School, MOVEP 2000 Nantes, France, June 19–23, 2000 Revised Tutorial Lectures*, F. Cassez, C. Jard, B. Rozoy, and M. D. Ryan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 3–38. DOI: 10.1007/3-540-45510-8_1 (cit. on p. 115).

[90] G. Metcalfe and M. Marti, “A Hennessy-Milner Property for Many-Valued Modal Logics,” in *Advances in Modal Logic*, R. Goré, B. Kooi, and A. Kurucz, Eds., vol. 10, London: College Publications, 2014, pp. 407–420 (cit. on p. 88).

[91] I. Micić, “Bisimulations for fuzzy automata,” Ph.D. dissertation, University of Niš, Faculty of Sciences and Mathematics, 2014 (cit. on pp. 6, 88, 89, 108).

[92] R. Milner, “A calculus of communicating systems,” in *Lecture Notes in Computer Science*, G. Goos and J. Hartmanis, Eds., vol. 92, Springer-Verlag Berlin Heidelberg, 1980. DOI: 10.1007/3-540-10235-3 (cit. on p. 3).

[93] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989 (cit. on p. 55).

[94] R. Milner, *Communicating and Mobile Systems: The π Calculus*. Cambridge University Press, 1999 (cit. on p. 55).

[95] F. Nejati, A. A. A. Ghani, N. K. Yap, and A. B. Jafaar, “Handling State Space Explosion in Component-Based Software Verification: A Review,” *IEEE Access*, vol. 9, pp. 77 526–77 544, 2021. DOI: 10.1109/ACCESS.2021.3081742 (cit. on p. 115).

[96] L. A. Nguyen, “Bisimilarity in Fuzzy Description Logics Under the Zadeh Semantics,” *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 6, pp. 1151–1161, 2019. DOI: 10.1109/TFUZZ.2018.2871004 (cit. on pp. 51, 88).

[97] L. A. Nguyen, “Characterizing fuzzy simulations for fuzzy labeled transition systems in fuzzy propositional dynamic logic,” *International Journal of Approximate Reasoning*, vol. 135, pp. 21–37, 2021. DOI: 10.1016/j.ijar.2021.04.006 (cit. on p. 101).

[98] L. A. Nguyen, “Logical characterizations of fuzzy bisimulations in fuzzy modal logics over residuated lattices,” *Fuzzy Sets and Systems*, vol. 431, pp. 70–93, 2022, Logic and Related Topics. DOI: 10.1016/j.fss.2021.08.009 (cit. on pp. 51, 88).

[99] L. A. Nguyen and N. T. Nguyen, “Bisimulations for Fuzzy Description Logics with Involutive Negation Under the Gödel Semantics,” in *Computational Collective Intelligence*, N. T. Nguyen *et al.*, Eds., Cham: Springer International Publishing, 2019, pp. 16–30. DOI: 10.1007/978-3-030-28377-3_2 (cit. on p. 51).

[100] L. A. Nguyen and N.-T. Nguyen, “Minimizing Interpretations in Fuzzy Description Logics Under the Gödel Semantics by Using Fuzzy Bisimulations,” *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 6, pp. 7669–7678, 2019. DOI: 10.3233/JIFS-179371 (cit. on p. 51).

[101] L. A. Nguyen and D. X. Tran, "Computing Fuzzy Bisimulations for Fuzzy Structures Under the Gödel Semantics," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 7, pp. 1715–1724, 2021. DOI: 10.1109/TFUZZ.2020.2985000 (cit. on p. 51).

[102] L. A. Nguyen *et al.*, "Bisimulation and bisimilarity for fuzzy description logics under the Gödel semantics," *Fuzzy Sets and Systems*, vol. 388, pp. 146–178, 2020, Logic. DOI: 10.1016/j.fss.2019.08.004 (cit. on pp. 51, 88).

[103] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*. OUP Oxford, 2006, vol. 31. DOI: 10.1093/acprof:oso/9780198566076.001.0001 (cit. on p. 127).

[104] R. V. Noorden, B. Maher, and R. Nuzzo, "The top 100 papers," *Nature News*, vol. 514, no. 7524, p. 550, 2014 (cit. on p. 3).

[105] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic* (The Springer International Series in Engineering and Computer Science). Springer New York, NY, 1999, vol. 517. DOI: 10.1007/978-1-4615-5217-8 (cit. on pp. 17, 29).

[106] P. Ostermann, "Many-Valued Modal Propositional Calculi," *Mathematical Logic Quarterly*, vol. 34, no. 4, pp. 343–354, 1988. DOI: 10.1002/malq.19880340411 (cit. on p. 3).

[107] R. Paige and R. E. Tarjan, "Three Partition Refinement Algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987. DOI: 10.1137/0216062 (cit. on p. 115).

[108] D. Park, "Concurrency and automata on infinite sequences," in *Theoretical Computer Science*, P. Deussen, Ed., vol. 104, Berlin, Heidelberg: Springer Berlin Heidelberg, 1981, pp. 167–183. DOI: 10.1007/BFb0017309 (cit. on p. 3).

[109] J. Pavelka, "On Fuzzy Logic I Many-valued rules of inference," *Mathematical Logic Quarterly*, vol. 25, no. 3-6, pp. 45–52, 1979. DOI: 10.1002/malq.19790250304 (cit. on p. 38).

[110] J. Pavelka, "On Fuzzy Logic II. Enriched residuated lattices and semantics of propositional calculi," *Mathematical Logic Quarterly*, vol. 25, no. 7-12, pp. 119–134, 1979. DOI: 10.1002/malq.19790250706 (cit. on p. 38).

[111] J. Pavelka, "On Fuzzy Logic III. Semantical completeness of some many-valued propositional calculi," *Mathematical Logic Quarterly*, vol. 25, no. 25-29, pp. 447–464, 1979. DOI: 10.1002/malq.19790252510 (cit. on p. 38).

[112] R. Pelánek, "Fighting State Space Explosion: Review and Evaluation," in *Formal Methods for Industrial Critical Systems*, D. Cofer and A. Fantechi, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 37–52. DOI: 10.1007/978-3-642-03240-0_7 (cit. on p. 115).

[113] M. Pistore and D. Sangiorgi, "A Partition Refinement Algorithm for the π-Calculus," *Information and Computation*, vol. 164, no. 2, pp. 264–321, 2001. DOI: 10.1006/inco.2000.2895 (cit. on p. 115).

[114] M. Praveen, "Does Treewidth Help in Modal Satisfiability?" *ACM Trans. Comput. Logic*, vol. 14, no. 3, 2013. DOI: 10.1145/2499937.2499939 (cit. on p. 128).

[115] G. Priest, "Many-Valued Modal Logics: A Simple Approach," *The Review of Symbolic Logic*, vol. 1, no. 2, pp. 190–203, 2008. DOI: 10.1017/S1755020308080179 (cit. on p. 3).

[116] A. M. Radzikowska, "Fuzzy Modal Operators and Their Applications," *Journal of Automation, Mobile Robotics and Intelligent Systems*, vol. 11, no. 1, pp. 10–20, 2017. DOI: 10.14313/JAMRIS_1-2017/2 (cit. on p. 38).

[117] A. M. Radzikowska and E. E. Kerre, "A comparative study of fuzzy rough sets," *Fuzzy Sets and Systems*, vol. 126, no. 2, pp. 137–155, 2002. DOI: 10.1016/S0165-0114(01)00032-X (cit. on p. 147).

[118] A. M. Radzikowska and E. E. Kerre, "Duality via Truth for Some Fuzzy Modal Logic," in *Algebraic Techniques and Their Use in Describing and Processing Uncertainty: To the Memory of Professor Elbert A. Walker*, H. T. Nguyen and V. Kreinovich, Eds. Cham: Springer International Publishing, 2020, pp. 129–149. DOI: 10.1007/978-3-030-38565-1_11 (cit. on pp. 37, 43).

[119] F. Ranzato and F. Tapparo, "Generalizing the Paige-Tarjan algorithm by abstract interpretation," *Information and Computation*, vol. 206, no. 5, pp. 620–651, 2008, Special Issue: The 17th International Conference on Concurrency Theory (CONCUR 2006). DOI: 10.1016/j.ic.2008.01.001 (cit. on p. 115).

[120] H. Rasiowa and R. Sikorski, *The Mathematics of Metamathematics*, 1st ed. Warsaw: Polish Scientific Publishers, 1963 (cit. on p. 24).

[121] M. Roggenbach and M. Majster-Cederbaum, "Towards a unified view of bisimulation: A comparative study," *Theoretical Computer Science*, vol. 238, no. 1, pp. 81–130, 2000. DOI: 10.1016/S0304-3975(99)00303-5 (cit. on p. 55).

[122] S. Roman, *Lattices and Ordered Sets*. New York: Springer Science & Business Media, 2008. DOI: 10.1007/978-0-387-78901-9 (cit. on pp. 7, 13, 16, 17, 60).

[123] D. Saha, "An Incremental Bisimulation Algorithm," in *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science*, V. Arvind and S. Prasad, Eds., ser. Lecture Notes in Computer Science, vol. 4855, Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 204–215. DOI: 10.1007/978-3-540-77050-3_17 (cit. on p. 115).

[124] E. Sanchez, "Resolution of composite fuzzy relation equations," *Information and Control*, vol. 30, pp. 38–48, 1 1976. DOI: 10.1016/S0019-9958(76)90446-0 (cit. on pp. 57, 58).

[125] E. Sanchez, "Solutions in composite fuzzy relation equations: Application to medical diagnosis in brouwerian logic," in *Fuzzy Automata and Decision Processes*, M. M. Gupta, G. N. Saridis, and B. R. Gaines, Eds., Amsterdam: North-Holland, 1977, pp. 221–234 (cit. on pp. 57, 58).

[126] E. Sanchez, "Resolution of eigen fuzzy sets equations," *Fuzzy Sets and Systems*, vol. 1, pp. 69–74, 1 1978. DOI: 10.1016/0165-0114(78)90033-7 (cit. on pp. 57, 58).

[127] D. Sangiorgi, "On the origins of bisimulation and coinduction," *ACM Transactions on Programming Languages and Systems*, vol. 31, pp. 1–41, 4 2009. DOI: 10.1145/1516507.1516510 (cit. on p. 3).

[128] D. Sangiorgi, *Introduction to Bisimulation and Coinduction*. Cambridge: Cambridge University Press, 2011. DOI: 10.1017/CBO9780511777110 (cit. on p. 3).

[129] H. S. Santos and B. C. Bedregal, "The Interval Constructor on Some Classes of ML-algebras," *LSFA'08*, 2008 (cit. on p. 17).

[130] P. Savický *et al.*, "On Product Logic with Truth-constants," *Journal of Logic and Computation*, vol. 16, no. 2, pp. 205–225, 2006. DOI: 10.1093/logcom/exi075 (cit. on p. 38).

[131] P. K. Schotch, "Fuzzy Modal Logic," in *Proceedings of the 5th International Symposium on Multiple-Valued Logic*, 1976, pp. 176–182 (cit. on p. 3).

[132] M. Shamsizadeh, M. M. Zahedi, and K. Abolpour, "Bisimulation for BL-general fuzzy automata," *Iranian Journal of Fuzzy Systems*, vol. 13, no. 4, pp. 35–50, 2016 (cit. on p. 51).

[133] A. Stamenković, M. Ćirić, and J. Ignjatović, "Reduction of fuzzy automata by means of fuzzy quasi-orders," *Information Sciences*, vol. 275, pp. 168–198, 2014. DOI: 10.1016/j.ins.2014.02.028 (cit. on pp. 32, 50, 85).

[134] I. Stanković, "Fuzzy relation equations and inequalities and their application in data analysis," (in Serbian), Ph.D. dissertation, University of Niš, Faculty of Sciences and Mathematics, 2017 (cit. on p. 54).

[135] M. Stanković, M. Ćirić, and J. Ignjatović, "Simulations and bisimulations for fuzzy multimodal logics over Heyting algebras," *Filomat*, 2021, accepted for publication (cit. on pp. 38, 52).

[136] M. Stanković, M. Ćirić, and J. Ignjatović, "Hennessy-Milner type Theorems for Fuzzy Multimodal Logics over Heyting Algebras," *Journal of Multiple-Valued Logic and Soft Computing*, 2022 (cit. on pp. 38, 88).

[137] U. Straccia, "Chapter 4 A fuzzy description logic for the semantic web," in *Fuzzy Logic and the Semantic Web*, ser. Capturing Intelligence, E. Sanchez, Ed., vol. 1, Elsevier, 2006, pp. 73–90. DOI: 10.1016/S1574-9576(06)80006-7 (cit. on p. 147).

[138] A. Tarski, "A lattice-theoretical fixpoint theorem and its applications," *Pacific journal of Mathematics*, vol. 5, no. 2, pp. 285–309, 1955. DOI: 10.2140/pjm.1955.5.285 (cit. on p. 17).

[139] J. Tick and J. Fodor, "Fuzzy implications and inference processes," *Computing and Informatics*, vol. 24, pp. 105–109, 2005 (cit. on p. 147).

[140] H. Todd Wareham, "The Parameterized Complexity of Intersection and Composition Operations on Sets of Finite-State Automata," in *Implementation and Application of Automata*, S. Yu and A. Păun, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 302–310. DOI: 10.1007/3-540-44674-5_26 (cit. on pp. 127, 128).

[141] A. Vidal, "On modal expansions of t-norm based logics with rational constants," Ph.D. dissertation, University of Barcelona, 2015. [Online]. Available: http://hdl.handle.net/10803/316575 (cit. on pp. 17, 38).

[142] A. Vidal, F. Esteva, and L. Godo, "A product modal logic," in *35th Linz Seminar on Fuzzy Set Theory*, 2014, pp. 127–130 (cit. on p. 37).

[143] A. Vidal, F. Esteva, and L. Godo, "On modal extensions of Product fuzzy logic," *Journal of Logic and Computation*, vol. 27, no. 1, pp. 299–336, 2017. DOI: 10.1093/logcom/exv046 (cit. on p. 37).

[144] P. Wild, L. Schröder, D. Pattinson, and B. König, "A van Benthem Theorem for Fuzzy Modal Logic," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, ser. LICS '18, Association for Computing Machinery, 2018, pp. 909–918. DOI: 10.1145/3209108.3209180 (cit. on p. 51).

[145] H. Wu and Y. Deng, "Logical characterizations of simulation and bisimulation for fuzzy transition systems," *Fuzzy Sets and Systems*, vol. 301, pp. 19–36, 2016, Theme: Logic and Computer Science. DOI: 10.1016/j.fss.2015.09.012 (cit. on p. 51).

[146] C. Yang and Y. Li, "Approximate bisimulations and state reduction of fuzzy automata under fuzzy similarity measures," *Fuzzy Sets and Systems*, vol. 391, pp. 72–95, 2020, Computer Science. DOI: 10.1016/j.fss.2019.07.010 (cit. on p. 51).

[147] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965. DOI: 10.1016/S0019-9958(65)90241-X (cit. on pp. 2, 3).

Biography of Author

Marko S. Stanković was born in Leskovac, Serbia, on March 29, 1988. He graduated from elementary school “Vuk Karadžić” in Lebane, class 2003, being awarded the “Vuk Karadžić” award and being pronounced the student of the generation. In 2007, he finished high school in Lebane, being awarded the “Vuk Karadžić” award and being pronounced the student of the generation.

In 2007, he started “BA” studies at the Faculty of Mathematics in Belgrade, as a module *Professor of Mathematics and Computer Science*, where he graduated in 2011 with an average grade of 9.00 and thus acquired the title *Bachelor of Mathematics*. He enrolled in Master studies on the same module. He defended his master thesis *Equivalent forms of the axiom of continuity of a set of real numbers* (in Serbian) in 2012 under the mentorship of Professor Zoran Kadelburg and thus acquired a Master degree with an average grade of 10.00, gaining the title *Master of Mathematics*.

In 2012, he started PhD studies at the Faculty of Mathematics in Belgrade, in the module *Mathematics*, but in 2016 he continued his studies at the University of Niš, Faculty of Sciences and Mathematics, PhD School of Mathematics, module: *Algebra and mathematical logic*. He passed all the exams with the highest grades.

In 2012, he started working at the Pedagogical Faculty in Vranje University of Niš as a Teaching Associate in the narrow scientific field of *Mathematics and Informatics*. In the following period, he continued his work at the Pedagogical Faculty as a Teaching Assistant, Junior researcher, as well as in the Research center of the Faculty. He is currently engaged as a Senior researcher at the Pedagogical Faculty.

During his work at the Pedagogical faculty, he held practical work in several subjects, such as Mathematics, Informatics, Elementary Mathematical Concepts, Information Technology, Information Technology Teaching Methods, Mathematical Logic, Programming, etc.

His research interests encompass several areas of mathematics and computer science, such as mathematical logic, fuzzy relations and fuzzy relation equations, automata and formal languages, information technology, mathematics education, etc. He is the author or co-author of several publications and research papers and one textbook. He has participated as a researcher in the following projects funded by the Ministry of Education, Science and Technological Development:

- Development of new information and communication technologies, based on advanced mathematical methods, with their application in medicine, telecommunications, energetics, protection of national heritage and education - III 44006 (2015-2019);
- Quantitative Automata Models: Fundamental Problems and Applications - QUAM 7750185 (2022-).

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом

Бисимулације за Крипкеове моделе фази мултимодалних логика

која је одбрањена на Природно-математичком факултету Универзитета у Нишу:

- резултат сопственог истраживачког рада;
- да ову дисертацију, ни у целини, нити у деловима, нисам пријављивао на другим факултетима, нити универзитетима;
- да нисам повредио ауторска права, нити злоупотребио интелектуалну својину других лица.

Дозвољавам да се објаве моји лични подаци, који су у вези са ауторством и добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу, 14.7.2022.

Потпис аутора дисертације:

Марко С. Станковић

Марко С. Станковић

AUTORSHIP STATEMENT

I declare that doctoral dissertation entitled

Bisimulations for Kripke models of Fuzzy Multimodal Logics

defended at Faculty of Sciences and Mathematics, University of Niš, is

- the result of original personal research;
- has not been submitted, either partially or in a whole, to any other faculty or university;
- not abusing authorship rights or intellectual property of anyone else.

I am allowing publication of my personal data related to the authorship of this dissertation, as well as the data related to gaining the academic title Doctor of Science, such as name, last name, year and place of birth, PhD defending date, to be published in the library catalogue of the University of Niš, digital repositorium of the University of Niš and in publication in general of the University of Niš.

In Niš, 14.7.2022.

Dissertation author's signature:

Marko S. Stanković

Marko S. Stanković

ИЗЈАВА О ИСТОВЕТНОСТИ ШТАМПАНОГ И ЕЛЕКТРОНСКОГ ОБЛИКА ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Наслов дисертације

Бисимулације за Крипкеове моделе фази мултимодалних логика

Изјављујем да је електронски облик моје докторске дисертације, коју сам предао за уношење у **Дигитални репозиторијум Универзитета у Нишу**, истоветан штампаном облику.

У Нишу, 14.7.2022.

Потпис аутора дисертације:

Марко С. Станковић

Марко С. Станковић

STATEMENT ABOUT IDENTICITY OF PRINTED AND ELECTRONIC FORMS OF THE DISSERTATION

Dissertation title:

Bisimulations for Kripke models of Fuzzy Multimodal Logics

I declare that the electronic form of my PhD dissertation, delivered to be uploaded in the **Digital repositorium of University of Niš**, is identical to the printed form of the dissertation.

In Niš, 14.7.2022.

Dissertation author's signature:

Marko S. Stanković

Marko S. Stanković

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла” да у дигитални репозиторијум Универзитета у Нишу унесе моју докторску дисертацију, под насловом:

Бисимулације за Крипкеове моделе фази мултимодалних логика

Дисертацију са свим прилозима предао сам у електронском облику, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons), за коју сам се одлучио.

1. Ауторство (CC BY)
2. Ауторство-некомерцијално (CC BY-NC)
3. **Ауторство-некомерцијално-без прераде (CC BY-NC-ND)**
4. Ауторство-некомерцијално-делити под истим условима (CC BY-NC-SA)
5. Ауторство-без прераде (CC BY-ND)
6. Ауторство-делити под истим условима (CC BY-SA)

У Нишу, 14.7.2022.

Потпис аутора дисертације:

Марко С. Станковић

Марко С. Станковић

USING STATEMENT

I am authorising herewith the library "Nikola Tesla" of the University of Niš to archive my dissertation into Digital repositorium of the University of Niš, entitled:

Bisimulations for Kripke models of Fuzzy Multimodal Logics

I have submitted the dissertation, along with its enclosures, in the electronic form, appropriate for permanent archiving.

My dissertation, archived in the Digital repositorium of the University of Niš, can be used by anyone who respects restrictions imposed by the chosen type of Creative Commons licence.

1. Authorship (CC BY)
2. Noncomercial authorship (CC BY-NC)
3. **Noncomercial authorship-no alterations (CC BY-NC-ND)**
4. Noncomercial authorship-share under the same conditions (CC BY-NC-SA)
5. Authorship-no alterations (CC BY-ND)
6. Authorship-share under the same conditions (CC BY-SA)

In Niš, 14.7.2022.

Dissertation author's signature:

Marko S. Stanković

Marko S. Stanković

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:	
Идентификациони број, **ИБР**:	
Тип документације, **ТД**:	монографска
Тип записа, **ТЗ**:	текстуални / графички
Врста рада, **ВР**:	докторска дисертација
Аутор, **АУ**:	Марко С. Станковић
Ментор, **МН**:	Мирослав Д. Ћирић
Наслов рада, **НР**:	Бисимулације за Крипкеове моделе фази мултимодалних логика
Језик публикације, **ЈП**:	енглески
Језик извода, **ЈИ**:	енглески и српски
Земља публиковања, **ЗП**:	Србија
Уже географско подручје, **УГП**:	Србија
Година, **ГО**:	2022.
Издавач, **ИЗ**:	ауторски репринт
Место и адреса, **МА**:	Ниш, Вишеградска 33
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)	6 поглавља; 227+xvi страна;147 цитата; 1 прилог
Научна област, **НО**:	математика
Научна дисциплина, **НД**:	фази логика и фази скупови
Предметна одредница/Кључне речи, **ПО**:	симулације, бисимулације, Крипкеови модели, фази логика, модална логика
УДК	510.643/.644(043.3) 510.22(043.3)
Чува се, **ЧУ**:	библиотека
Важна напомена, **ВН**:	
Извод, **ИЗ**:	Главни задатак дисертације јесте да пружи детаљну студију више различитих типова симулација и бисимулација за Крипкеове моделе фази мултимодалних логика. Представљена су два типа симулација (директне и повратне) и пет типова бисимулација (директне, повратне, директно-повратне, повратно-директне и регуларне). За сваки тип симулација и бисимулација креиран је алгоритам који тестира постојање симулације или бисимулације и, уколико иста постоји, алгоритам израчунава највећу. У дисертацији је приказана примена бисимулација у редуковању броја светова фази Крипкеових модела уз очување њихових семантичких својстава. Даље, разматране су слабе симулације и бисимулације и испитано је Хенеси-

		Милнерово (Hennessy-Milner) својство. На крају, креиран је алгоритам за израчунавање слабих симулација и бисимулација за фази Крипкеове моделе над локално коначним алгебрама.
Датум прихватања теме, **ДП**:		4.3.2022.
Датум одбране, **ДО**:		
Чланови комисије, **КО**:	Председник:	
	Члан:	
	Члан:	
	Члан:	
	Члан:	
	Члан, ментор:	

Образац Q4.09.13 - Издање 1

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	monograph
Type of record, **TR**:	textual / graphic
Contents code, **CC**:	doctoral dissertation
Author, **AU**:	Marko S. Stanković
Mentor, **MN**:	Miroslav D. Ćirić
Title, **TI**:	Bisimulations for Kripke models of Fuzzy Multimodal Logics
Language of text, **LT**:	English
Language of abstract, **LA**:	English and Serbian
Country of publication, **CP**:	Serbia
Locality of publication, **LP**:	Serbia
Publication year, **PY**:	2022
Publisher, **PB**:	author's reprint
Publication place, **PP**:	Niš, Višegradska 33
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	6 chapters; 227+xvi pages; 147 references; 1 appendix
Scientific field, **SF**:	mathematics
Scientific discipline, **SD**:	fuzzy logic and fuzzy sets
Subject/Key words, **S/KW**:	Simulations, Bisimulations, Kripke models, Fuzzy logic, Modal logic
UC	510.643/.644(043.3) 510.22(043.3)
Holding data, **HD**:	library
Note, **N**:	
Abstract, **AB**:	The main objective of the dissertation is to provide a detailed study of several different types of simulations and bisimulations for Kripke models of fuzzy multimodal logics. Two types of simulations (forward and backward) and five types of bisimulations (forward, backward, forward-backward, backward-forward and regular) are presented hereby. For each type of simulation and bisimulation, an algorithm is created to test the existence of the simulation or bisimulation and, if it exists, the algorithm computes the greatest one. The dissertation presents the application of bisimulations in the state reduction of fuzzy Kripke models, while preserving their semantic properties. Next, weak simulations and bisim-

		ulations were considered and the Hennessy-Milner property was examined. Finally, an algorithm was created to compute weak simulations and bisimulations for fuzzy Kripke models over locally finite algebras.
Accepted by the Scientific Board on, **ASB**:		04.03.2022
Defended on, **DE**:		
Defended Board, **DB**:	President:	
	Member:	
	Member:	
	Member:	
	Member:	
	Member, Mentor:	

Образац Q4.09.13 - Издање 1