Vrije Universiteit Brussel
Faculty of Science and
Bio-Engineering Sciences
Department of Computer Science
Theoretical Computer Science

Universiteit Gent
Faculty of Science
Department of Applied Mathematics,
Computer Science & Statistics
Computational Web Intelligence

Nonmonotonic Reasoning in Multivalued Logics

Dissertation submitted in fulfilment of the requirements for the degree of Doctor of Science: Computer Science

Marjon Blondeel

19 March 2014

Promotors: Prof. Dr. Dirk Vermeir (Vrije Universiteit Brussel)
Prof. Dr. Martine De Cock (Universiteit Gent)
Dr. Steven Schockaert (Cardiff University)

Abstract

Inference in classical logic is monotonic: if a conclusion can be derived from a set of premises, then no additional premises will ever invalidate this conclusion. However, commonsense reasoning has a nonmonotonic component. Human beings draw sensible conclusions from what they know, making default assumptions where needed. And if there is new information, we might reconsider previous conclusions. Depending on how one deals with statements as "typically it holds that" and "in the absence of information to the contrary", different nonmonotonic logics can be considered, which have been studied since the 1970s. Two important formalisms for nonmonotonic reasoning are autoepistemic logic and negation-as-failure in logic programming.

Answer set programming (ASP) is a declarative programming language based on the stable model semantics that is used to model complex combinatorial problems. Its strength lies in the use of the negation-as-failure operator which allows retracting previously made conclusions when new information is available. Moreover, there is a clear connection between ASP and autoepistemic logic: ASP programs can be translated to a set of formulas in autoepistemic logic such that the answer sets are in one-to-one correspondence with the so-called stable expansions in autoepistemic logic.

Although ASP has been successfully applied to model combinatorial problems in a concise and declarative manner, it is not directly suitable for expressing problems in continuous domains. Fuzzy answer set programming (FASP) is a generalisation of ASP based on fuzzy logic that is capable of modelling continuous systems by using an infinite number of truth degrees corresponding to intensities of properties. Since it is a relatively new concept, little is known about the computational complexity of FASP and almost no techniques are available to compute answer sets of FASP programs. Furthermore, the connections of FASP to other paradigms for nonmonotonic reasoning with continuous values are largely unexplored. In our dissertation, we contribute to the ongoing research on FASP on several levels.

First, we will pin down the complexity of the direct syntactical generalisation of classical ASP to FASP, and we will develop an implementation based on bilevel linear programming for this type of programs.

Second, we will combine the paradigms of fuzzy logic and autoepistemic logic into fuzzy autoepistemic logic, and show that the latter generalises FASP. Since the language of (fuzzy) autoepistemic logic is much more expressive than the theories we need to represent (fuzzy) answer set programs, this could serve as a useful basis for defining or comparing extensions to the basic language of (F)ASP. Moreover, we show that many important properties from classical autoepistemic logic remain valid when generalising to fuzzy autoepistemic logic.

Finally, we will investigate relationships between fuzzy autoepistemic logic and fuzzy modal logics, generalising well-known links beween autoepistemic logic and several classical modal logic systems. In particular we will generalise Levesque's logic of only knowing to the many-valued case, and show that the correspondence with autoepistemic logic is preserved under this generalisation; stable expansions (and hence answer sets) correspond to a particular type of valid sentences in this logic. Moreover we will provide a sound and complete axiomatisation for this many-valued logic of only knowing.

To summarise, in this thesis we will introduce several systems of nonmonotonic reasoning when properties may be graded and investigate their properties, complexity of appropriate decision problems and relations among them.

Contents

1 Introduction

1.1 Logic

The term "logic" comes from the Greek word *logos*, originally meaning "speech", "reason", "plea", "opinion", "ground", ... It first became a technical term in philosophy due to Heraclitus (ca. 535-475 BC). Heraclitus argued that there was an objective truth about everything which he called *Logos* [Audi 1995]. Ancient philosophers used the term in different ways; e.g. the sophists (5th century BC) used it as "discourse" and Aristotle (384-322 BC) used it to refer to "reasoned discourse" [Audi 1995]. The earliest study of formal logic is accredited to Aristotle. The so-called Aristotelian logic was the dominant form of Western logic until the 19th century advances in mathematical logic [Corcoran 2009]. One of the important contributions Aristotle made to the study of logic is the idea of *deductive reasoning*, e.g. suppose we know that

All men are mortal and Socrates is a man.

By simple syllogistic reasoning it then follows that

Socrates is mortal.

In the 19th century, Boole presented his work *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*. He proposed to express logical propositions as algebraic expressions, providing a fail-safe method of logical deduction. This work can be seen as the starting point of mathematical logic [Corcoran

2003]. From then onwards, logic also became a mathematical study rather than a purely philosophical one.

Until the first half of the 20th century, logic was primarily a "tool" to understand the foundations of mathematics. After Boole, great advances in mathematical logic were made by Frege. He was the first to propose an axiomatisation of propositional and first-order logic, where he "invented" the last one himself. He was the first important proponent of logicism; the theory that mathematics is reducible to logic. In his work *Grundgesetze der Arithmetik* he attempted to derive the laws of arithmetic from logic. After publication of the first part in 1893, Russell proved that the laws (grundgesetze) of Frege led to a contradiction which is now known as *Russell's paradox*. An important method to solve this paradox was proposed by Zermelo and this led to the first axiomatic set theory which later developed into the well-known *Zermelo-Fraenkel set theory (ZF)*. An important work on the foundations of mathematics is *Principia Mathematica* (published in three volumes during 1910-1913). In this work, Russell and Whitehead attempted to define a set of axioms and inference rules in a symbolic, propositional logic from which all mathematical truths could be proven. However, in 1913, *Gödel's incompleteness theorem* proved that this goal could never be achieved. Also during this period, first-order logic was being further developed. In 1929, Gödel proved a correspondence between semantics and proof theory in first-order logic, known as *Gödel's completeness theorem*.

From the 1930s onwards, computability theory became an important part of logic. Computability theory as we know it now is mostly influenced by the work in the 1930s and 40s of Turing, Church, Kleene and Post. Church is best known for his proof that the "Entscheidungsproblem" is undecidable [Church 1936]. This is a problem that asks for an algorithm that given a statement of a first-order logic answers yes or no depending on whether the statement is a tautology. As part of this research in computability theory he also introduced lambda calculus. Independently from Church, Turing also showed in [Turing 1936] that a general solution to the Entscheidungsproblem does not exist. In his original proof, Turing formalised the concept of algorithm by introducing Turing machines which became the standard model for computing devices. Post and Kleene then extended the scope of computability theory and introduced the concept of degrees of unsolvability [Post 1944, Kleene and Post 1954].

The development of digital computers, the interest in machine simulation of human intelligence and the development of mathematical linguistics led to many new applications of classical logic. In particular, new problems suggested by computer science and artificial intelligence led to the requirement of non-traditional logics. Some of the resulting logics are extensions of classical logics with new operators. One example is the development of modal logics, which are extensions of formal logic with modal operators to include e.g. possibility, necessity, belief and knowledge. Modal logic had already been studied by e.g. Aristotle in his *Prior Analytics*, but modal logic as we know it today was first studied (syntactically) by Lewis in [Lewis 1918, Lewis and Langford 1932]. In [Lewis and Langford 1932], five axiomatic systems were formulated; "*S1*", "*S2*", "*S3*", "*S4*" and "*S5*". Gödel [Gödel 1933] then suggested a new way to axiomatise the systems of modal logic, the first step in obtaining the well-known systems *K*, *T*, *S4* and *S5*. The use of semantics started later [Carnap 1947] and became increasingly noticeable at the beginning of the 1960s [Hintikka 1962] and was greatly influenced by Kripke when he introduced the *Kripke Semantics* in 1959 [Kripke 1959]. Besides extensions, also restrictions of classical logic have been investigated. One example is intuitionistic logic which was introduced and axiomatised by Heyting (1930). It is based on the idea of Brouwer (1907) that you should not count a proof of the form "there exists x such that $Q(x)$ holds" valid unless the proof gives a method for constructing such an x, and similar for a proof of the form "R or S holds". Hence you cannot assert the statement "R or $not\ R$" (law of the excluded middle) unless you have a proof for R or for $not\ R$.

One property of classical logics and many of their extensions and restrictions is that they are *monotonic*: if a formula can be inferred from a set Γ of premises, then it can also be inferred from any set of premises Λ with $\Gamma \subseteq \Lambda$. However, in daily life, we do not use this kind of inference. One usually draws conclusions tentatively and when obtaining more information one might change the conclusions. Such inferences are called *nonmonotonic* since, as opposed to monotonic reasoning, the set of conclusions does not necessarily increase as the set of premises increases.

1.2 Nonmonotonic reasoning

Historically, the need for nonmonotonic reasoning was inspired by knowledge representation problems in several areas of artificial intelligence [Brewka et al. 1997]. A typical motivating example is the development of a database about airline flights. One usually just stores only positive facts in a database, like "there is a flight from Brussels to Barcelona at 10am, August 19, 2014". Obviously it is not possible to store all the negative facts, e.g. "there is no flight from Brussels to Barcelona at 11am, August 20, 2014" in the database. Hence the unstated assumption is that if a flight is not listed in the database, it does not exist. Reiter [Reiter 1978] formalised the assumption that what is not currently known to be true is false: the *closed world assumption* (CWA). Intuitively, the CWA means that any information not mentioned in (or that cannot be inferred from) a database DB is taken to be false.

A second knowledge representation problem introduced by McCarthy [McCarthy 1980] consists of formalising puzzles such as the Missionaries and Cannibals problem:

Three missionaries and three cannibals come to a river. A row boat that seats two is available. If the cannibals ever outnumber the missionaries on either bank of the river, the missionaries will be eaten. How shall they cross the river?

As was the case with databases, it is easy to state all required positive facts but more difficult to state negative assumptions that are implicit, e.g. "the only way to cross the river is by boat". McCarthy proposed a formal solution to deal with such unstated assumptions called *circumscription*. The idea of circumscription is that things are as expected unless otherwise specified. Since its first formulation in [McCarthy 1980], circumscription has taken several forms. The most popular and useful form is *parallel predicate circumscription* [McCarthy 1980, McCarthy 1986, Lifschitz 1985].

CWA and circumscription have a lot in common. Both work on the principle of preferring interpretations in which positive facts are minimised. CWA also has a strong correspondence with logic programming semantics, which will be discussed in Section 1.2.1.

We will now discuss several nonmonotonic logics that differ from the previous logics, in the sense that they explicitly use the notion of consistency or (dis-)belief. A common property of consistency-based logics is that they sometimes generate multiple "solutions". In these logics, solutions are sets of formulas representing beliefs a reasoner can adopt based on premises and defaults. Consider for instance the following premises and defaults.

Quakers (typically) are pacifists. Republicans (typically) are not pacifists. Nixon was quaker and republican.

From the first default rule one might conclude that Nixon was a pacifist. On the other hand, there are also reasons to believe that Nixon was not a pacifist since he was a republican. A consistency-based logic would generate two so-called extensions. There does not seem to be a specific correct way of using these extensions, e.g. considering intersections or introducing a preference relation. Consistency-based logics just define the extensions and leave open what to do with them.

Reiter's *default logic* (DL) [Reiter 1980] is probably the most prominent consistency-based logic and has been used to formalise a number of different reasoning taks, e.g. diagnosis from first principles [Reiter 1987] and inheritance [Etherington 1987]. Together with the closely related autoepistemic logic, which we will discuss in detail in Section 2.1, it has a greater expressive power than for instance circumscription.

DL assumes that knowledge is represented in terms of a default theory, i.e. a pair (D, W) with W a set of first-order formulas representing facts that are true and D a set of defaults of the form

$$A : B_1, \dots, B_n/C.$$

This default means that if A is provable then C should be derived if for all i, $\sim B_i$ (i.e. the negation of B_i) is not provable. For example, consider $W = \{Bird(Tweety)\}$ and D a single default

$$Bird(Tweety) : Flies(Tweety)/Flies(Tweety).$$

Hence if Tweety is a bird and it is not provable that Tweety does not fly, then we should derive that Tweety flies. Reiter defines the extensions of a pair (D, W) as fixpoints.

DL's expressive power is mainly due to the representation of defaults as inference rules. Unfortunately this also has problems and many researchers have considered this as a serious drawback [Brewka et al. 1997]. For instance, the existence of extensions is not guaranteed and sometimes DL does not give the results one would intuitively expect. These difficulties have led to a number of modifications of DL, e.g. [Łukasiewicz 1988], [Brewka 1991], [Schaub 1991], [Delgrande and Jackson 1991]. Another variant of DL defined in [Baral and Subrahmanian 1991] can be viewed as a generalisation of the well-founded semantics for logic programs, which will be discussed in Section 1.2.1.

Modal nonmonotonic logics [McDermott and Doyle 1980], [Moore 1985], [Marek et al. 1991] use a modal operator to express explicitly that a formula is consistent (or believed). The most widely studied logic of this class is Moore's *autoepistemic logic* [Moore 1985]. It has interesting links with DL and logic programming. Moore introduced a modal operator B in the logical language where $\mathrm{B}\alpha$ stands for "α is believed". The idea of autoepistemic logic is to model the reasoning of an ideally rational agent about his own beliefs. Ideally rational means that the agent knows completely what he believes as well as what he does not believe. Hence if a formula α belongs to the set S of beliefs of the agent, then $\mathrm{B}\alpha$ should also be an element of S. And if α does not belong to S, then $\sim\mathrm{B}\alpha$ must be in S. Stalnaker [Stalnaker 1993][1] describes the state of belief characterised by such a theory as *stable*: no further conclusions can be drawn by an ideally rational agent in such a state. Theories satisfying these conditions are called *stable autoepistemic theories*. As will be discussed in Section 2.1, Moore introduced the so-called *stable expansions* for a set of premises. These are stable autoepistemic theories defined as particular fixpoints.

Moore's work about autoepistemic logic was originally based on ideas from McDermott and Doyle ([McDermott 1982], [McDermott and Doyle 1980]). In [McDermott and Doyle 1980] an operator M was introduced which has to be read as "it is consistent to believe that" and where we have the inference rule that "MP is derivable if $\sim P$ is not derivable."

[1] Article based on the unpublished manuscript (1980) to which Moore referred in [Moore 1985].

However, as pointed out in [McDermott and Doyle 1980], $\mathrm{M}P$ is not inconsistent with $\sim P$. In [McDermott 1982] the standard modal logics T, S4 and S5 were extended to nonmonotonic logics, although in the case of S5 it was shown that introducing nonmonotonicity is redundant: the theories of nonmonotonic S5 are exactly those of the standard version of S5. Moore [Moore 1985] showed why some of the problems McDermott and Doyle encountered arised and how they can be avoided. The language Moore defined is much like McDermott and Doyle's, a propositional logical language augmented by an epistemic modal operator B interpreted as "is believed".

In [Moore 1984], Moore proposed an alternative possible world semantics. In this sense, autoepistemic logic is closely related to the work of Halpern and Moses [Halpern and Moses 1984]. The major difference is that in [Halpern and Moses 1984] a logic of knowledge rather than belief is considered. It then follows that $\mathrm{B}p \to p$ (with B here interpreted as "is known") is an axiom. It was obtained independently by Moore and by Halpern and Moses that the sets of formulas that are true in every world of some S5 structure are exactly the stable autoepistemic theories.

Although the intuitions underlying default and autoepistemic logic seem very different, they are much more related then one might expect. In [Konolige 1988], Konolige proposed the following translation from default logic to autoepistemic logic:

$$A : B_1, \ldots, B_n / C \quad \text{is translated to} \quad \mathrm{B}A \wedge {\sim}\mathrm{B}B_1 \wedge \ldots \wedge {\sim}\mathrm{B}B_n \to C.$$

Konolige showed that extensions of a default theory (D, W) correspond exactly to the sets of formulas in the stable expansions not containing the operator B.

There are also nonmonotonic systems using two independent modal operators ([Lin and Shoham 1990],[Lifschitz 1991], [Lifschitz 1994]). Lin and Shoham showed that DL and autoepistemic logic, as well as the minimal belief logic of [Halpern and Moses 1984] can be embedded in their system. Lifschitz's logic uses an epistemic operator B representing minimal belief and an operator "not" to represent the notion of negation-as-failure. The difference between (strong) negation "$\sim$" and "not" is that $\sim a$ is true if we can derive $\sim a$, whereas $\mathrm{not}\, a$ is true if we fail to derive a. Lifschitz showed that DL and autoepistemic logic can be embedded in his logic as well. As we will discuss later in this section, there are also interesting links with logic programming.

Finally, another modal approach to default reasoning has been introduced and investigated in [Levesque 1990]. Levesque defined a logic containing a modal operator O, where $\mathrm{O}\alpha$ stands for "α is *all* that is known". Hence there are no other relevant beliefs about α. As will be discussed in Section 2.1.3, the operator O has an intuitive possible world semantics and the stable expansions of a formula α correspond to those worlds that satisfy $\mathrm{O}\alpha$.

In [Kraus et al. 1990], general patterns of nonmonotonic reasoning have been studied. The approach is based on the work of Gabbay [Gabbay 1985] who suggested to focus the study of nonmonotonic logics on their consequence relation, although some of the systems mentioned earlier were not meant to define a consequence relation. He proposed that such a consequence operator $\mid\!\sim$ should have the following properties:

- Reflexivity: $\alpha \mid\!\sim \alpha$
- Cut: from $\alpha \wedge \beta \mid\!\sim \gamma$ and $\alpha \mid\!\sim \beta$, infer $\alpha \mid\!\sim \gamma$
- Cautious (or weak) monotonicity: from $\alpha \mid\!\sim \beta$ and $\alpha \mid\!\sim \gamma$ infer $\alpha \wedge \beta \mid\!\sim \gamma$

In [Makinson 1988], Makinson proposed a semantics for Gabbay's logic, but only for a rather limited syntax. Independently, Shoham [Shoham 1987], [Shoham 1988], proposed a general framework for nonmonotonic reasoning. He suggested models that are described as a set of worlds with a preference relation. He assumed a more expressive language, containing all classical connectives. This idea also appears in [Halpern and Moses 1984] in relation with epistemic logic. In [Kraus et al. 1990], *system* $\mathbf{P}$ was introduced as a variation on the semantics proposed in [Shoham 1987]. The key aspect is that the consequence relation $\mid\!\sim$ satisfies cautious monotonicity. It expresses that a new conclusion should not invalidate previous conclusions. Intuitively, you add only what you expect. This system does not cover everything in the area of nonmonotonic reasoning. It can be shown that there cannot be such a consequence relation for e.g. DL [Makinson 1988] and autoepistemic logic [Makinson 2005].

1.2.1 Nonmonotonicity in logic programming

In general, most of the programming languages that are used are imperative programming languages[2]. The programmer has to describe what the program should do and also how the program should do this. On the other hand, using declarative programming languages the programmer should describe what the program should do but not how this should be done. Most programming languages based on logic are declarative. Historically, semantics for logic programs have been considered since the 1970s with [Colmerauer et al. 1973], [Kowalski 1974], [Van Emden and Kowalski 1976]. This work led to the logic programming language *PROLOG*. Originally, PROLOG was restricted to horn clauses but it was quickly extended to include negation-as-failure. As mentioned previously, negation-as-failure is a special construct "not" where $\text{not}\, a$ is true if we cannot prove that a is true. For example, consider the following program P.

$$\begin{array}{lll} r_1 : & \text{beach} & \leftarrow \text{sunny} \wedge \text{not raining} \\ r_2 : & \text{sunny} & \leftarrow \end{array}$$

[2]http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Rule r_1 informally means that we will go to the beach if there is no reason to think that it is raining and if we are sure that it is sunny. A rule such as r_2 is called a fact; it encodes that it is unconditionally true that it is sunny.

Defining the semantics for logic programs containing negation-as-failure turned out to be a challenge. The most well-known ideas are *Clark's completion* [Clark 1978], the *stable model semantics* [Gelfond and Lifschitz 1988] and the *well-founded semantics* [van Gelder et al. 1988]. Clark's completion has various drawbacks, see [Apt and Bol 1994], [Stepherdson 1991] for more details. The stable model semantics refine the conclusions of the Clark completion and in [Baral and Subrahmanian 1993] it has been shown that the well-founded semantics is an approximation of the stable model semantics. Over the last two decades, many other definitions of the stable model semantics have been formulated, with each of them giving new insights [Lifschitz 2008].

The close relation between these semantics and circumscription, DL and autoepistemic logic has been shown by translating programs into these logics and proving that the semantics of the program corresponds to the meaning of the translation in the logic. Marek and Truszczyńki [Marek and Truszczyński 1989] and Bidoit and Froidevaux [Bidoit and Froidevaux 1991b] considered translations from a logic program P into default logic. They proved that the stable models of P correspond to Reiter's extensions of the corresponding default theory. In [Marek and Truzsczyński 1993] a summary of similar results is given. In combination with the relation between stable expansions in autoepistemic logic and extensions in default logic it is then obtained that stable models of a program correspond to the stable expansions of a corresponding autoepistemic theory. In [Gelfond et al. 1989], translations from circumscriptive theories into logic programs were considered but the conditions under which these translations work well are very strong.

In this thesis we will focus on the relationship between answer set programming – logic programming based on the stable model semantics – and autoepistemic logic. The research oriented towards the relationship between answer set programming and autoepistemic logic has its roots in the problem of defining semantics for programs with negation-as-failure. Consider for instance the rule

$$a \leftarrow b \wedge \mathrm{not}\, c.$$

Gelfond [Gelfond 1987] observed that in autoepistemic logic, this rule can be expressed naturally as the formula

$$b \wedge \sim(\mathrm{B}c) \rightarrow a.$$

Gelfond and Lifschitz [Gelfond and Lifschitz 1988] showed that the answer sets of programs P without strong negation and only one atom in the head of a rule correspond to the models of the autoepistemic logic theory $\lambda(P)$ obtained from P by interpreting rules as material implication and replacing all expressions of the form $\mathrm{not}\, a$ by $\sim(\mathrm{B}a)$. Unfortunately this translation does not work for programs with more than one atom in the head of rules. For instance the program consisting of the single rule

$$a \vee b \leftarrow$$

would correspond to the autoepistemic formula $a \vee b$ which has exactly one model. The program itself however has two answer sets $\{a\}$ and $\{b\}$. If strong negation is allowed and disjunction in the head is not allowed, problems arise since material implication is contrapositive and rules in ASP are not. This observation led Gelfond and Lifschitz [Gelfond and Lifschitz 1991] to reject autoepistemic logic as a tool for the study of logical programming, using a semantics based on default logic instead. In [Lifschitz and Schwarz 1993] however, Lifschitz and Schwarz showed that programs with strong negation and disjunction in the head can easily be represented by autoepistemic theories. In this translation, a rule of the form

$$a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \mathrm{not}\, c_1 \wedge \ldots \wedge \mathrm{not}\, c_k$$

is transformed into the autoepistemic formula

$$(b_1 \wedge \mathrm{B}b_1) \wedge \ldots \wedge (b_m \wedge \mathrm{B}b_m) \wedge \sim(\mathrm{B}c_1) \wedge \ldots \wedge \sim(\mathrm{B}c_k) \rightarrow (a_1 \wedge \mathrm{B}a_1) \vee \ldots \vee (a_n \wedge \mathrm{B}a_n).$$

The results in [Lifschitz and Schwarz 1993] are based on the logic of minimal belief and negation as failure (MNBF). This correspondence was independently found by Chen [Chen 1993], also using MBNF as a starting point. The negation-as-failure modality in MBNF exactly corresponds to negative introspection in autoepistemic logic. MBNF is thus an extension of autoepistemic logic with the "minimal knowledge operator" due to Halpern and Moses [Halpern and Moses 1984]. A simplified version of MBNF (from [Lifschitz 1994]), which will also be used in this thesis, can be used to simulate some forms of default logic and circumscription, as well as some logic programming languages.

1.3 Many-valued logic

Besides monotonicity, other aspects of classical logic have been questioned throughout history. In particular, classical logic is not suitable to deal with situations in which properties are gradual, such as temperature and size. One of the most radical and fruitful attempts for augmenting the representational capabilities of classical logic was made by Zadeh in 1965 [Zadeh 1965] by introducing *fuzzy sets*. Fuzzy set theory is a generalisation of classical set theory with many-valued membership functions; an element belongs to a *fuzzy* set to a certain degree. Therefore, fuzzy sets are seen as functions. Fuzzy set theory is closely related to the many-valued logics that appeared in the 1930s; many-valued logic is to fuzzy set theory what classical logic is to set theory. The idea is that one could extend the usual notions of derivability and entailment from (usual) sets of premises to fuzzy sets of premises. These extensions work most naturally in these many-valued logics, hence also called fuzzy logics. Fuzzy or many-valued logics are extensions and generalisations of classical logic. They can have $k \in \mathbb{N} \setminus \{0, 1\}$ or even infinitely many truth degrees in $[0, 1]$. The only assumptions are that each sentence in such a logic has exactly one truth degree and that there exist always at least two truth degrees, $\overline{0}$ and $\overline{1}$, which behave exactly like "true" and "false" in classical (two-valued) logic. Compared to classical logic, a large set of truth degrees then allows more operations with truth degrees.

Historically, the era of many-valued logic started in the 1920s with Łukasiewicz [Łukasiewicz 1920] and Post [Post 1921] although many-valued logic may be traced back to Aristotle who discussed the problem of contingent statements about the future. This problem of future contingents is closely tied with the philosophical problems of determinism and the understanding of modalities. Indeed, a future event may be seen as "possible" or "undetermined". This problem inspired Łukasiewicz to consider a three-valued logic, where there is – besides true and false – a not determined third truth degree. Independently, Post introduced the idea of additional truth degrees and applied it to problems of representability of functions and he proved that every sentence of an m-valued logic can be interpreted as an ordered set of $m - 1$ sentences of classical logic. In 1922, Łukasiewicz [Łukasiewicz 1922] generalised his previous work to an infinite number of truth degrees.

This initial phase of many-valued logic was followed by basic theoretical results for systems of many-valued logic. Wajsberg [Wajsberg 1931] proposed an axiomatisation for the three-valued system of Łukasiewicz. The latter system was extended to a functionally complete one and an axiomatisation was given in [Słupecki 1936]. Gödel [Gödel 1932] used multiple truth degrees to understand intuitionistic logic, leading to the well-known family of Gödel systems. These systems were extended to infinitely many truth degrees in [Jaskowski 1936]. The work of Gödel and Jaskowski clarified mutual relations of intuitionistic and many-valued logic: it was shown that there does not exist a many-valued system whose set of logically valid formulas coincides with the set of logically valid formulas of intuitionistic logic. Furthermore, besides Łukasiewicz' 1930s papers, the monograph [Rosser and Turquette 1952] was the standard reference for years. It is a collection of papers from the 1940s in which they generalise basic approaches and proved essential results. They emphasise on the development of Hilbert style axiomatic calculi for systems of many-valued logic.

In the 1950s, McNaughton [McNaughton 1951] showed that every piecewise linear function $[0, 1]^n \rightarrow [0, 1]$ can be represented by a sentence in the infinitely-valued calculus of

Łukasiewicz logic. A completeness proof for the system was given by Chang introducing the notion of an MV-algebra and showing that they form the algebraic semantics for this logic ([Chang 1958],[Chang 1959]). In the same period, Dummet [Dummet 1959] gave a completeness proof for the infintely-valued Gödel logic by showing that the corresponding algebraic structure is the class of all Heyting algebras.

The 1970s was a period of rather restricted activity in pure many-valued logics. However, there was a lot of work in the closely related area of applications of fuzzy sets, defined by Zadeh. Fuzzy logic – where derivability and entailment are extended from sets to fuzzy set of premises – was first studied by Pavelka [Pavelka 1979]. He was concerned with fuzzy propositional logic with sets of truth degree that are as general as possible. He presented an axiomatisation but his proof of completeness is only valid for the case where the set of truth degrees is (isomorphic to) the set of truth degrees from one of the Łukasiewicz systems. An important special case of fuzzy logic was to consider only truth degrees taken in $[0,1]\cap\mathbb{Q}$. Simplifications of the axiomatisation and other interesting results on this fuzzy logic where given more recently by Hájek, e.g. [Hájek 1995, Hájek 1998]. In particular, it turns out that the main results hold without forcing the language to be uncountable.

A family of infinitely-valued logics with truth degrees from the whole unit interval $[0,1]$ is the class of t-norm based fuzzy logics. A systematic study of these particular logics started with [Hájek 1998]. Hájek presents the notion of the logic of a continuous t-norm which is a continuous function $[0,1]\times[0,1]\to[0,1]$ that can be seen as a generalisation of the truth table of the classical conjunction. He presents the logics of the three basic continuous t-norms: Łukasiewicz, Gödel and product logic, and the 'basic" fuzzy logic of all continuous t-norms. As mentioned before, Łukasiewicz and Gödel logic have been introduced and investigated long before the family of t-norm based logics was recognised. Since then many other t-norm fuzzy logics have been introduced and investigated, e.g. *monodical t-norm based logic* (MTL) [Esteva and Godo 2001] which is the logic of left-continuous t-norms. An overview can be found in [Gottwald and Hájek 2005].

An important remark that should be made is the fact that fuzzy logic deals with many-valuedness in a logical format and not with uncertainty or probability. De Finetti [De Finetti 1936] pointed out that uncertainty is a meta concept: it is related to agents not being totally sure whether a proposition is true or false, without questioning the fact that this proposition can only be true or false. Carnap [Carnap 1945] points out the difference between truth values and probability values in the sense that "true" is not the same as "verified". We may have a probability that a sentence is true but it is not a degree of truth and a sentence is often neither verified or falsified but it is either true or false, whether anybody knows it or not. See also [Dubois and Prade 2001] for a discussion on the difference between uncertainty, probability and truth degrees.

1.4 Outline of the thesis

In this thesis we will combine the ideas of nonmontonic reasoning and many-valued logic to obtain a framework which is useful to deal with situations for nonmontonic reasoning when propositions are graded. One of the existing tools for nonmonotonic reasoning in such a context is fuzzy answer set programming (FASP), which is a combination of answer set programming (which we will recall in Section 3.1) and fuzzy logic (which we will discuss in Section 2.2). FASP inherits the declarative nonmonotonic reasoning capabilities from answer set programming (ASP), while fuzzy logic adds the power to model continuous problems. FASP can be tailored towards different applications since fuzzy logic gives a great flexibility, e.g. by the possibility to use different generalisations of the classical connectives. In recent years a variety of approaches to FASP have been proposed (e.g. [Damásio and Pereira 2001], [Janssen et al. 2009], [Łukasiewicz and Straccia 2007], [Van Nieuwenborgh et al. 2007]). Although it has been studied by several authors, FASP is by far not as developed as ASP. For example, very little is known about its computational complexity and few techniques are known to compute the answer sets of FASP programs. Also, many extensions proposed for ASP have not yet been considered in FASP. With the exceptions of [Łukasiewicz and Straccia 2007], [Schockaert et al. 2012], [Straccia et al. 2009] and this thesis, most work is even restricted to FASP programs with exactly one atom in the head. We will introduce the FASP framework we want to study in Section 3.2 and provide motivating examples in Section 3.3.

The main purpose of this thesis is to study fundamental properties of FASP and introduce other types of systems for nonmonotonic reasoning when propositions are graded. We will investigate their properties and study the computational complexity and relationships between several of these graded versions of formalisms for nonmonotic reasoning.

- In Chapter 4 we will study the computational complexity of FASP under Łukasiewicz semantics. Łukasiewicz logic (Section 2.2.3) is often used in applications because it preserves many desirable properties from classical logic and it is closely related to mixed integer programming. An overview of the complexity results that we can establish is provided in Tables 4.1 and 4.2. Moreover we will provide a reduction from reasoning with such FASP programs to bilevel linear programming.
- In Chapter 5 we will combine autoepistemic logic (Section 2.1) and fuzzy logic (Section 2.2) and show that the classical results about the equivalence of answer sets and stable expansions (Theorems 3.1 and 3.2) remain valid in the resulting fuzzy autoepistemic logic. This logic is useful to reason about one's beliefs about the degrees to which properties are satisfied and we show that important properties from classical autoepistemic logic remain valid in this generalisation. As in the classical case, the language of fuzzy autoepistemic logic is much more expressive

than the theories we need to represent the FASP programs. This could serve as as useful basis to define extensions for FASP.

- In Chapter 6 we will study relationships between fuzzy autoepistemic logic based on finitely-valued Łukasiewicz logic and fuzzy modal logics of belief. We will provide sound and complete axiomatisations for these fuzzy modal logics of belief and show interesting links with fuzzy autoepistemic logic. In particular we will generalise Levesque's logic of only knowing (Section 2.1.3), provide a sound and complete axiomatisation and show that the stable expansions correspond to valid sentences in this fuzzy logic of only knowing, generalising Theorem 2.1.

A summarising diagram of embeddings and generalisations with references can be found in Figure 1.1.

The results in this thesis have been published, or submitted for publication, in international journals and the proceedings of international conferences with peer review. Specifically, fuzzy autoepistemic was first introduced in [Blondeel et al. 2011b] and the relations with FASP were investigated. This was extended to more general FASP programs in [Blondeel et al. 2014b]. First results on the complexity of FASP under Łukasiewicz logic were presented in [Blondeel et al. 2011a]. These results were extended to more subclasses of FASP in [Blondeel et al. 2012] and [Blondeel et al. 2014c] where the latter also contains an overview of all results. In [Blondeel et al. 2013a] a first attempt was given to study the relation between fuzzy autoepistemic logic and fuzzy modal logics of belief. These results were bundled and studied in more detail in a paper that is submitted to an international journal [Blondeel et al. 2014a]. Finally, an introductory book chapter on FASP can be found in [Blondeel et al. 2013c] and an article [Blondeel et al. 2013b] containing the highlights of this thesis was presented at the International Joint Conference on Artificial Intelligence (IJCAI).

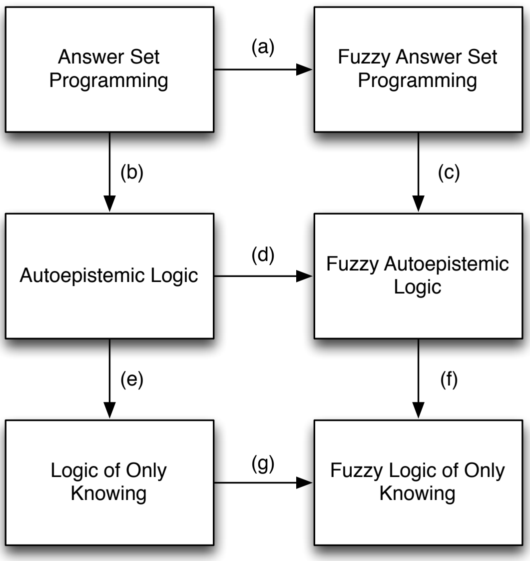


Figure 1.1: (a): e.g. [Van Nieuwenborgh et al. 2007], (b): [Gelfond and Lifschitz 1988], (c)-(d): Chapter 5, (e): [Levesque 1990], (f)-(g): Chapter 6

2 Reasoning about beliefs

In this chapter we introduce some preliminary notions on autoepistemic logic, fuzzy logic and the minimal modal logic over finitely-valued Łukasiewicz logic with a finite set of truth constants.

2.1 Autoepistemic logic

In this section we will introduce autoepistemic logic, a logic that was originally intended to model the beliefs of an ideally rational agent [Moore 1985]. Moreover it is well-known to generalise the stable model semantics of answer set programming which we will recall in Section 3.1. In Section 2.2 we will discuss fuzzy logic and we will use this to propose a generalisation of autoepistemic logic in chapter 5 and we will show that many properties from classical autoepistemic logic remain valid under this generalisation. We will also show that the relation between fuzzy answer set programming – a combination of answer set programming and fuzzy logic – which will be recalled in Section 3.2 and fuzzy autoepistemic logic remains valid as well. In this section we will also recall logic of only knowing, a logic in which autoepistemic logic can be embedded. The so-called stable expansions from autoepistemic logic then occur in this logic as valid sentences. In Chapter 6 we will generalise this result when considering the particular semantics of finitely-valued Łukasiewicz logic with a finite set of truth constants (see Section 2.3).

2.1.1 Formalizing autoepistemic reasoning

Autoepistemic logic was originally intended to model the beliefs of an ideally rational agent reflecting upon his own beliefs [Moore 1985]. Formulas in this logic are used to represent the beliefs of such agents. In particular, the formulas of autoepistemic logic are built from a countable set of variables or atoms A , the constants true ($\overline{1}$) and false ($\overline{0}$), the usual classical connectives ($\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\sim$[1]) and a modal operator B, interpreted as "is believed". For example, if φ is a formula, then $\mathrm{B}\varphi$ indicates that φ is believed. Note that $\mathrm{B}(\sim\varphi)$ indicates that $\sim\varphi$ is believed and $\sim\mathrm{B}\varphi$ that φ is not believed. Recall that in propositional logic, the set $\{\overline{0}, \rightarrow\}$ is a minimal functionally complete set. This means that these operators are sufficient to express all possible propositional formulas. For autoepistemic formulas α and β we thus have

$$\begin{array}{ll} \sim\alpha = \alpha \rightarrow \overline{0} & \alpha \vee \beta = \sim\alpha \rightarrow \beta \\ \alpha \wedge \beta = \sim(\sim\alpha \vee \sim\beta) & \alpha \leftrightarrow \beta = (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha) \end{array}$$

We write $\mathcal{L}^c$ for the language of all propositional formulas over A and $\mathcal{L}^c_{\mathrm{B}}$ for the extension of $\mathcal{L}^c$ with the operator B.

Definition 2.1

The language $\mathcal{L}^c_{\mathrm{B}}$ is recursively defined as follows.

- $a \in A$ is a formula.
- $\overline{0}$ is a formula.
- If α is a formula, then $\mathrm{B}\alpha$ is a formula.
- If α, β are formulas, then $\alpha \rightarrow \beta$ is a formula.

A set of formulas in $\mathcal{L}^c_{\mathrm{B}}$ is called an *autoepistemic theory* and formulas not containing the operator B are called *objective* .

We define

$$A' = A \cup \{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}^c_{\mathrm{B}}\},$$

which is an infinite set, even if A is finite. For technical reasons, we sometimes treat A' as a set of atoms, and consider evaluations $I' \in \mathcal{P}(A') = \{B \mid B \subseteq A'\}$. In such a case, expressions of the form $\mathrm{B}(a \wedge \mathrm{B}b)$ and $\mathrm{B}(\mathrm{B}a)$ are atoms but $a \wedge \mathrm{B}b$ is not. This trick allows

[1] We do not use the usual notation "$\neg$" to make a clear distinction with the strong negation $\neg$ we will use in ASP.

us to deal with autoepistemic theories in a purely propositional fashion and soundness and completeness theorems from propositional logic are inherited, i.e. a formula α is true in all propositional models of an autoepistemic theory T iff it is a tautological consequence of T iff it is derivable from T by the usual rules of propositional logic.

Moore [Moore 1985] originally defined the semantics for autoepistemic logic by considering autoepistemic interpretations $I \in \mathcal{P}(A')$. He defined an *autoepistemic interpretation* I of an autoepistemic theory T as a subset of A' that is a propositional evaluation of T such that a formula $\mathrm{B}\alpha$ is true (in I) iff $\alpha \in T$. Note that the theory T completely determines the truth of formulas of the form $\mathrm{B}\alpha$, independently of the truth assignments of the propositional atoms in T. Finally, an *autoepistemic model* of T is an autoepistemic interpretation of T in which all formulas in T are true. Hence the autoepistemic models of T are the propositional models that correspond to the intended meaning of the operator B. Indeed, suppose that the beliefs of an agent are represented by some autoepistemic theory T, then an autoepistemic interpretation I is an autoepistemic model of T if it is a propositional model of T and all the beliefs of the agents are true in I, i.e. $\mathrm{B}\alpha$ is true in I iff $\alpha \in T$.

Given this formal semantics we also want a notion of inference: which set of beliefs T' should an agent adopt on the basis of a set of initial premises T? First of all, since we are dealing with rational agents, the beliefs should be true provided that the premises are true. Moreover we want these beliefs to contain everything that the agent could semantically conclude from his/her beliefs and from the knowledge that these are his/her beliefs. Formally, we want T' to be *sound* w.r.t. T and *semantically complete*:

Definition 2.2

An autoepistemic theory T' is *sound* w.r.t. to an initial set of premises T iff every autoepistemic interpretation of T' that is an autoepistemic model of T is an autoepistemic model of T' as well. T' is called *semantically complete* iff it contains all of its logical consequences, i.e. T' contains every formula that is true in every autoepistemic model of T'.

To give a more intuitive definition of the set of beliefs T' an agent should adopt based on a set of initial premises T, we will now discuss a syntactical characterisation. To do so, let us specify the closure conditions we would expect the beliefs of an ideally rational agent to possess. Intuitively, they should include whatever the agent can infer by classical logic and by reflecting on what he/she believes. Formally, an autoepistemic theory T representing the beliefs of an ideally rational agent should satisfy the following conditions:

(1) If $\alpha_1, \ldots, \alpha_n$ are in T and β is a propositional logical consequence of $\alpha_1, \ldots, \alpha_n$, then β is also in T.

(2) If α is in T, then $\mathrm{B}\alpha$ is in T.

(3) If α is not in T, then $\sim\mathrm{B}\alpha$ is in T.

Stalnaker [Stalnaker 1993][2] describes the state of belief characterised by such a theory as *stable*: no further conclusions can be drawn by an ideally rational agent in such a state. Theories satisfying these conditions are called *stable autoepistemic theories*. For such theories T that are also consistent, i.e. there exists no formula α such that α and $\sim\alpha$ are elements of T, one can show that two additional intuitive conditions are satisfied:

(4) If $\mathrm{B}\alpha$ is in T, then α is in T.

(5) If $\sim\mathrm{B}\alpha$ is in T, then α is not in T.

For a consistent stable theory T we then have that $\alpha \in T$ iff $\mathrm{B}\alpha \in T$ and $\alpha \notin T$ iff $\sim\mathrm{B}\alpha \in T$.

In [Moore 1985] it is shown that the stable autoepistemic theories are exactly those that are semantically complete. Stability alone does not tell us what an agent should not believe. It is still possible that the agent believes propositions that are not sound w.r.t. his initial premises. However, by imposing syntactical constraints on the theory, the notion of soundness can be captured. In particular, we need constraints imposing that the agent only believes his initial premises T and those required by the stability conditions (1)-(3). In [Moore 1985], it is shown that T' is sound w.r.t. T iff every formula of T' is included in the propositional consequences of

$$T \cup \{\mathrm{B}\varphi \mid \varphi \in T'\} \cup \{\sim\mathrm{B}\varphi \mid \varphi \notin T'\}.$$

Hence, given a set of premises T the sets of beliefs $T' \supseteq T$ a rational agent may have should be such that

1. T' is semantically complete, i.e. T' is a stable autoepistemic theory, and
2. T' is sound w.r.t. T, i.e. every formula of T' is included in the propositional consequences of $T \cup \{\mathrm{B}\varphi \mid \varphi \in T'\} \cup \{\sim\mathrm{B}\varphi \mid \varphi \notin T'\}$.

We will call such sets T' *stable expansions* of T. They can be equivalently defined as follows.

Definition 2.3: [Moore 1985]

Consider autoepistemic theories T and E. Then E is a *stable expansion* of T iff

$$E = \mathsf{Cn}(T \cup \{\mathrm{B}\varphi \mid \varphi \in E\} \cup \{\sim\mathrm{B}\varphi \mid \varphi \notin E\}),$$

[2] Article based on the unpublished manuscript (1980) to which Moore referred in [Moore 1985].

where Cn(X) denotes the set of propositional consequences of X.

Example 2.1

Consider an initial set of premises

$$T = \{{\sim}\mathrm{B}a \rightarrow b, {\sim}\mathrm{B}b \rightarrow a\}.$$

The first formula has to be read as "if a is not believed, then b is true" and the second as "if b is not believed, then a is true". Suppose E is a stable expansion of T. If $a \notin E$ then we have ${\sim}\mathrm{B}a \in E$. Since $({\sim}\mathrm{B}a \rightarrow b) \in T$ it follows that $b \in E$. Similarly, $b \notin E$ implies $a \in E$. On the other hand, if $a \in E$ there is no basis to have $b \in E$ as well, and vice versa. Hence if T has a stable expansion E it will contain either a or b, but not both. It follows that there are two stable expansions E_1 and E_2, where E_1 (resp. E_2) contains a (resp. b) and all formulas a ideally rational agent can derive from the set $\{a\}$ (resp. $\{b\}$) using the reasoning provided by conditions (1)-(3).

Using Definition 2.3, the following proposition can be shown.

Proposition 2.1: [Marek 1989]

If all formulas in an autoepistemic theory T are objective, then T has exactly one stable expansion.

Example 2.2

Consider the autoepistemic theory $Q = \{a\}$. Since Q is a set of objective formulas, by Proposition 2.1 it has exactly one stable expansion E :

$$E = \mathsf{Cn}(Q \cup C)$$

with

$$C = \{\mathrm{B}\varphi \mid \varphi \in E\} \cup \{{\sim}\mathrm{B}\varphi \mid \varphi \notin E\}.$$

We will show that E is also a stable expansion of $T = \{{\sim}\mathrm{B}a \rightarrow b, {\sim}\mathrm{B}b \rightarrow a\}$ by proving that

$$\mathsf{Cn}(T \cup C) = \mathsf{Cn}(Q \cup C).$$

First, we prove that $a \in \mathsf{Cn}(T \cup C)$; it then follows that $\mathsf{Cn}(Q \cup C) \subseteq \mathsf{Cn}(T \cup C)$. Because $b \notin Q$, it must hold that $b \notin E$ and hence we have $\sim\mathrm{B}b \in C$. Since $(\sim\mathrm{B}b \rightarrow a) \in T$ it then follows that $a \in \mathsf{Cn}(T \cup C)$.

Conversely, since $a \in E$, we have $\mathrm{B}a \in C$ and thus $(\sim\mathrm{B}a \rightarrow b) \in \mathsf{Cn}(Q \cup C)$. We also have $(\sim\mathrm{B}b \rightarrow a) \in \mathsf{Cn}(Q \cup C)$, which implies that $\mathsf{Cn}(T \cup C) \subseteq \mathsf{Cn}(Q \cup C)$.

By symmetry, it follows that the unique stable expansion of $\{b\}$ is also a stable expansion of T.

2.1.2 Possible world semantics and syntactical characterisation

In [Moore 1984], Moore proposes to characterise stable autoepistemic theories by a Kripke-style possible world semantics. Truth is now defined relative to a structure (I, S) with $I \in \mathcal{P}(A) = \{B \mid B \subseteq A\}$ representing the actual world and $S \subseteq \mathcal{P}(A)$ representing all worlds considered possible i.e. the possible beliefs of an agent. An atom a is *true* in (I, S), written as $(I, S) \models a$, iff $a \in I$. This can be extended to formulas as follows:

- $(I, S) \not\models \overline{0}$,
- $(I, S) \models (\alpha \rightarrow \beta)$ iff $(I, S) \not\models \alpha$ or $(I, S) \models \beta$,
- $(I, S) \models \mathrm{B}\alpha$ iff for every $J \in S$ it holds that $(J, S) \models \alpha$.

with α and β autoepistemic formulas. Intuitively, a formula α is *believed*, i.e. $\mathrm{B}\alpha$ is true, if α is true in every interpretation which is considered possible.

Definition 2.4

A set $S \subseteq \mathcal{P}(A)$ is a *possible world autoepistemic model* of an autoepistemic theory T iff

$$S = \{I \in \mathcal{P}(A) \mid \forall \varphi \in T : (I, S) \models \varphi\}.$$

In other words, the set of possible worlds w.r.t. the beliefs of the agent is a possible world autoepistemic model of T if it is exactly the set of worlds in which all formulas of T are true.

Definition 2.5

An autoepistemic theory T is called the *belief set* of $S \subseteq \mathcal{P}(A)$ iff

$$T = \{\varphi \in \mathcal{L}_{\mathrm{B}}^{\mathrm{c}} \mid \forall I \in S : (I, S) \models \varphi\} = \{\varphi \in \mathcal{L}_{\mathrm{B}}^{\mathrm{c}} \mid \forall I \in \mathcal{P}(A) : (I, S) \models \mathrm{B}\varphi\},$$

We will write $\mathsf{Th}(S)$ to denote this set of formulas.

The set $\mathsf{Th}(S)$ thus contains exactly those formulas that are true in every world that is considered possible w.r.t. the beliefs of an agent and hence this set contains exactly the beliefs of the agent.

The following proposition describes the relation between stable expansions and possible world autoepistemic models.

Proposition 2.2: [Moore 1984], [Halpern and Moses 1984]

An autoepistemic theory E is a stable expansion of an autoepistemic theory T iff $E = \mathsf{Th}(S)$ for some possible world autoepistemic model S of T.

Example 2.3

Let us explicitely compute the stable expansions of the autoepistemic theory $Q = \{a\}$. By Proposition 2.1 it follows that there is exactly one stable expansion E. Thus by Proposition 2.2 there is also exactly one possible world autoepistemic model S such that $E = \mathsf{Th}(S)$:

$$S = \{I \in \mathcal{P}(A) \mid (I, S) \models a\} = \{I \in \mathcal{P}(A) \mid a \in I\}\,.$$

Hence, the unique possible world autoepistemic model of $Q = \{a\}$ is the set of all propositional interpretations that contain a and it follows that

$$E = \mathsf{Th}(S) = \{\varphi \in \mathcal{L}_{\mathrm{B}}^{c} \mid \forall I \in \mathcal{P}(A) : a \in I \Rightarrow (I, S) \models \varphi\}\,.$$

Finally, we recall a syntactical characterisation in terms of stable sets (see Section 2.1.1) for stable expansions.

Proposition 2.3: [Levesque 1990]

Consider an autoepistemic theory T. There exists $S \subseteq \mathcal{P}(A)$ such that $T = \mathsf{Th}(S)$ iff T is a stable autoepistemic theory.

2.1.3 Embedding autoepistemic logic into logic of only knowing

In [Levesque 1990], the language of $\mathcal{L}_{\mathrm{B}}^{c}$ is augmented with an operator O[3]. A formula of the form $\mathrm{O}\varphi$ is read as "φ is all that is believed" or "only φ is believed", i.e. there are no

[3] In [Levesque 1990], the language considered is more expressive. There a first-order language is augmented with operators B and O.

other (relevant) beliefs. The semantics for O is defined as follows

$$(I, S) \models \mathrm{O}\phi \quad \text{iff} \quad \forall J \in \mathcal{P}(A) : J \in S \Leftrightarrow (J, S) \models \phi.$$

Notice the difference with the modal operator B which can be stated as

$$(I, S) \models \mathrm{B}\phi \quad \text{iff} \quad \forall J \in \mathcal{P}(A) : J \in S \Rightarrow (J, S) \models \phi.$$

Informally the only difference is that for "believing" every possible world, i.e. the worlds in S, must satisfy ϕ and for "only believing" all possible worlds satisfy ϕ but no other world satisfies ϕ. The resulting logic is called *logic of only knowing*.

An important result is that stable expansions occur in this logic as valid sentences. Intuitively, knowing a sentence means that what is believed is a stable expansion of that sentence.

Theorem 2.1: [Levesque 1990]

Suppose α is a formula in $\mathcal{L}_{\mathrm{B}}^{\mathrm{c}}$, $I \in \mathcal{P}(A)$ is an arbitrary interpretation and $S \subseteq \mathcal{P}(A)$ is a set of evaluations. Then $(I, S) \models \mathrm{O}\alpha$ iff Th(S) is a stable expansion of $\{\alpha\}$.

One can easily prove Theorem 2.1 using the definitions of possible world autoepistemic models and the semantics of the operator O. The result, although slightly adapted, remains valid when considering a first-order language, see [Levesque 1990] for details.

In [Levesque 1990] a sound and complete axiomatisation for logic of only knowing is provided. This is done by noticing that only knowing can be broken up into two parts. On the one hand we have that α is believed. This can be expressed by $\mathrm{B}\alpha$. On the other hand we have that at most α is believed. We will express this using a new operator N, where $\mathrm{N}\varphi$ means that at most φ is believed to be false. We obtain that $\mathrm{O}\alpha$ can be stated as $\mathrm{B}\alpha \wedge \mathrm{N}(\sim\alpha)$. Hence α is believed and at most α is believed; exactly α is believed. The precise definition of the semantics for N is as follows:

$$(I, S) \models \mathrm{N}\alpha \text{ iff } \forall J \in \mathcal{P}(A) : (J, S) \not\models \alpha \Rightarrow J \in S.$$

Notice that we could rewrite this definition a follows.

$$(I, S) \models \mathrm{N}\alpha \text{ iff } \forall J \in \mathcal{P}(A) : J \in \mathcal{P}(A) \setminus S \Rightarrow (J, S) \models \alpha.$$

So N is like a belief operator but the complement of S is used as the set of possible worlds. It can be shown that both B and N are K45 operators (see e.g. [Halpern and Moses 1992] for basic notions about modal logics). By adding K45 axioms for both B and N to an axiomatisation for propositional logic and the following cross-axioms, a sound and complete axiomatisation for logic of only knowing is obtained.

- $\phi \rightarrow \mathrm{B}\phi$, where all variables and constants in ϕ occur in the scope of an operator N or B,
- $\phi \rightarrow \mathrm{N}\phi$, where all variables and constants in ϕ occur in the scope of an operator N or B,
- $\sim\mathrm{B}\phi \vee \sim\mathrm{N}\phi$, if $\sim\phi$ is satisfiable and does not contain any modal operators,
- $\mathrm{O}\phi \equiv \mathrm{B}\phi \wedge \mathrm{N}(\sim\phi)$.

2.2 Fuzzy logic

Fuzzy logics (e.g. [Hájek 1998]) form a class of logics whose semantics are based on truth degrees taken from the unit interval $[0, 1]$. In this section we will present common operators of fuzzy logic (see e.g. [Klement et al. 2000]), i.e. generalisations of the classical logical operators, and we will discuss formal fuzzy logics. We will conclude with a detailed discussion on a particular type of fuzzy logic, Łukasiewicz logic. This is a logic which is often used in applications because it preserves many desirable properties from classical logic and because inference in this logic can naturally be reduced to mixed integer programming, for which efficient solvers are available. In Chapter 4 we will thoroughly discuss the complexity of fuzzy answer set programming under Łukasiewicz semantics.

2.2.1 Logical operators

Conjunction and disjunction are usually generalised by *triangular norms* and *triangular conorms* respectively .

Definition 2.6

- A *triangular norm* (short t-norm) is an increasing, associative and commutative mapping $T : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfying $T(x, 1) = x$ for each $x \in [0, 1]$.
- A *triangular conorm* (short t-conorm) is an increasing, associative and commutative mapping $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$ satisfying $S(x, 0) = x$ for each $x \in [0, 1]$.

Since triangular norms and conorms are associative and commutative, they can be extended to $n \in \mathbb{N}$ arguments.

Note that a function $S : [0, 1] \times [0, 1] \rightarrow [0, 1]$ is a t-conorm iff there exists a t-norm T such that for all $x, y \in [0, 1]$

$$S(x, y) = 1 - T(1 - x, 1 - y).$$

The t-conorm S is then called the dual t-conorm of T and T is called the dual t-norm of S.

There exist infinitely many t-norms [Klement et al. 2000]. The most well-known are the minimum T_M, the product T_P and the Łukasiewicz t-norm T_L:

- $T_M(x, y) = \min(x, y)$
- $T_P(x, y) = x \cdot y$
- $T_L(x, y) = \max(x + y - 1, 0)$

for $x, y \in [0, 1]$.

For the t-norms described above we get the following dual t-conorms: the maximum S_M, the probabilistic sum S_P and the Łukasiewicz t-conorm S_L:

- $S_M(x, y) = \max(x, y)$,
- $S_P(x, y) = x + y - x \cdot y$,
- $S_L(x, y) = \min(x + y, 1)$

for $x, y \in [0, 1]$.

Example 2.4

It is easy to show that every t-norm generalises the truth table of the classical conjunction. Indeed, using the fact that a t-norm is commutative and has to satisfy $T(x, 1) = x$ for each $x \in [0, 1]$ it follows that $0 = T(0, 1) = T(1, 0)$ and $T(1, 1) = 1$. Since T is increasing we also have $0 \leq T(0, 0) \leq T(0, 1) = 0$ and hence $T(0, 0) = 0$. Similarly, one can show that every t-conorm generalises the truth table of the classical disjunction.

For intermediate truth values we obtain different results depending on the particular t-(co)norm that is used, e.g.

$$\begin{array}{rclrcl} T_M(0.5, 0.5) & = & 0.5 & S_M(0.5, 0.5) & = & 0.5 \\ T_P(0.5, 0.5) & = & 0.25 & S_P(0.5, 0.5) & = & 0.75 \\ T_L(0.5, 0.5) & = & 0 & S_L(0.5, 0.5) & = & 1 \end{array}$$

Logical implication can be generalised by an *implicator*.

Definition 2.7

An *implicator* is a mapping $I : [0,1] \times [0,1] \to [0,1]$ such that $I(0,0) = I(0,1) = I(1,1) = 1$ and $I(1,0) = 0$ and I is decreasing in the first component and increasing in the second.

For a continuous t-norm, the following proposition can be shown.

Proposition 2.4: [Hájek 1998]

Given a continuous t-norm T, there is a unique function $I^* : [0,1] \times [0,1] \to [0,1]$ such that

$$T(z,x) \leq y \text{ iff } z \leq I^*(x,y)$$

for $x, y, z \in [0,1]$. This function is defined as

$$I^*(x,y) = \max\{z \in [0,1] \mid T(x,z) \leq y\}.$$

Example 2.5

For the Łukasiewicz t-norm T_L the corresponding function I^* coincides with

$$I_L : [0,1] \times [0,1] \to [0,1] : (x,y) \mapsto \min(1 - x + y, 1).$$

Indeed, if $x \leq y$, then

$$I^*(x,y) = \max\{z \in [0,1] \mid T_L(x,z) \leq y\} = 1.$$

Otherwise, if $x > y$ then the maximal value z such that $\max(x + z - 1, 0) \leq y$ is $z = 1 - x + y$. Hence

$$I^*(x,y) = \max\{z \in [0,1] \mid T_L(x,z) \leq y\} = 1 - x + y.$$

Notice that all such functions I^* are implicators and they generalise classical modus ponens in the sense that $T(x,z) \leq y$ iff $z \leq I^*(x,y)$: from a (lower bound of) the truth degree x of a formula ϕ and a (lower bound of) the truth degree $I^*(x,y)$ of a formula $\phi \to \psi$ one can derive the (lower bound of) the truth degree y of ψ. For this reason and also because of other nice properties such as Proposition 2.5 we will often consider such implicators and call them residual implicators.

Definition 2.8

Given a continuous t-norm T, the *residual implicator* I_T is defined as

$$I_T(x, y) = \max\{z \in [0, 1] \mid T(x, z) \leq y\}$$

for $x, y \in [0, 1]$.

This residual implicator has a nice property:

Proposition 2.5: [Hájek 1998]

If T is a continuous t-norm, then for all $x, y \in [0, 1]$ it holds that

$$x \leq y \text{ iff } I_T(x, y) = 1.$$

Example 2.6

Consider the mapping

$$I : [0, 1] \times [0, 1] \to [0, 1] : (x, y) \mapsto \max(1 - x, y).$$

It is easy to see that I is an implicator. Moreover it generalises the classical tautology $p \to q \leftrightarrow {\sim}p \vee q$ using the t-conorm S_M and the negator N_L (see Definition 2.10):

$$I(x, y) = S_M(N_L(x), y).$$

However Proposition 2.5 does not hold since e.g. $I(0.1, 0.8) = 0.9 \neq 1$. Hence it is not a residual implicator.

For the continuous t-norms minimum, product and Łukasiewicz t-norm, we obtain the following residual implicators:

- $I_M(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}$
- $I_P(x, y) = \begin{cases} 1 & \text{if } x \leq y \\ \frac{y}{x} & \text{otherwise} \end{cases}$
- $I_L(x, y) = \min(1 - x + y, 1)$

for $x, y \in [0, 1]$. Note that I_L is a continuous function, but I_M and I_P are not.

Logical equivalence can be generalised by defining a *biresiduum*.

Definition 2.9

Given a t-norm T and an implicator I, the *biresiduum* of T and I is a mapping $E_{T,I} : [0,1] \times [0,1] \to [0,1]$ defined as

$$E_{T,I}(x,y) = T(I(x,y), I(y,x))$$

for $x, y \in [0,1]$.

Finally, negation can be generalised by a *negator*.

Definition 2.10

A *negator* is a decreasing mapping $N : [0,1] \to [0,1]$ such that $N(1) = 0$ and $N(0) = 1$.

Every implicator I induces a negator N_I as follows

$$N_I : [0,1] \to [0,1] : x \mapsto I(x,0).$$

For the minimum, product and Łukasiewicz t-norms, we obtain the following negators

- $N_{I_M}(x) = N_{I_P}(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$
- $N_{I_L}(x) = 1 - x$

for $x \in [0,1]$. Notice that the function N_{I_L} is involutive, i.e. $N_L(N_L(x)) = x$ for all $x \in [0,1]$ and $N_{I_M} = N_{I_P}$ are not.

2.2.2 The basic many-valued logic

We will now introduce the propositional fuzzy logic L^K, the logic corresponding to a set of continuous t-norms K. Its language is built from a countable set of atoms A, the connectives $\otimes$ and $\to$ and the truth constant $\overline{0}$. Further connectives are defined as follows.

$$\phi \wedge \psi = \phi \otimes (\phi \to \psi) \qquad \phi \vee \psi = ((\phi \to \psi) \to \psi) \wedge ((\psi \to \phi) \to \phi)$$
$$\sim\!\phi = \phi \to \overline{0} \qquad \phi \leftrightarrow \psi = (\phi \to \psi) \otimes (\psi \to \phi)$$
$$\overline{1} = \sim\!\overline{0}$$

A T-*evaluation* for some $T \in K$, is a mapping $e : A \to [0, 1]$ such that

$$e(\overline{0}) = 0$$
$$e(\phi \to \psi) = I_T(e(\phi), e(\psi))$$
$$e(\phi \otimes \psi) = T(e(\phi), e(\psi))$$

This evaluation e can be uniquely extended to the evaluation of all formulas:

$$e(\phi \wedge \psi) = \min(e(\phi), e(\psi)) \qquad e(\phi \vee \psi) = \max(e(\phi), e(\psi))$$
$$e(\sim\phi) = N_{I_T}(e(\phi)) \qquad e(\phi \leftrightarrow \psi) = E_{T,I_T}(e(\phi), e(\psi))$$
$$e(\overline{1}) = 1$$

A formula ϕ is called a T- *tautology* if $e(\phi) = 1$ for each T- evaluation e. Formulas that are T-tautologies for every $T \in K$ are called *tautologies* of the logic L^K. A T-evaluation e is called a T-*model* of a set of formulas Γ if $e(\varphi) = 1$ for every $\varphi \in \Gamma$. A formula φ is a *semantic consequence* of Γ in L^K if for each $T \in K$ we have that all T-models of Γ are T-models of $\{\varphi\}$.

The logic of all continuous t-norms is called *basic logic* and is denoted by BL. The logics of $\{T_M\}$, $\{T_P\}$ and $\{T_L\}$ are respectively called *Gödel*, *product* and *Łukasiewicz* logic. Note that in Łukasiewicz logic we can also define disjunction $\oplus$ which corresponds to the t-conorm S_L: $\phi \oplus \psi = \sim(\sim\phi \otimes \sim\psi) : e(\phi \oplus \psi) = S_L(e(\phi), e(\psi))$. This can only be done for Łukasiewicz logic because of the particular definition of the negator based on T_L. Indeed,

$$\begin{aligned}
e(\phi \oplus \psi) &= e(\sim(\sim\phi \otimes \sim\psi)) \\
&= N_{I_L}(e(\sim\phi \otimes \sim\psi)) \\
&= 1 - e(\sim\phi \otimes \sim\psi) \\
&= 1 - T_L(e(\sim\phi), e(\sim\psi)) \\
&= 1 - T_L(1 - e(\phi), 1 - e(\psi)) \\
&= S_L(e(\phi), e(\psi))
\end{aligned}$$

Example 2.7

Consider the formula

$$\phi = (a \wedge b) \otimes (\sim a \vee \sim b).$$

Seen as a classical formula $(a \wedge b) \wedge (\sim a \vee \sim b)$, this formula can never be satisfied since it is of the form $\psi \wedge \sim\psi$. In Łukasiewicz logic we have a similar result. For any evaluation $e : A \to [0,1]$ such that $e(a) \leq e(b)$ we obtain

$$\begin{aligned} e(\phi) &= \max(\min[e(a), e(b)] + \max[1 - e(a), 1 - e(b)] - 1, 0) \\ &= \max(e(a) + 1 - e(a) - 1, 0) \\ &= 0 \end{aligned}$$

A similar computation holds for the cases where $e(a) > e(b)$.

For Gödel and product logic, the truth degree of ϕ varies. For instance if $e(a) = e(b) = 1$, then in Gödel logic we have

$$\begin{aligned} e(\phi) &= \max(\min[e(a), e(b)], \max[N_{I_M}(e(a)), N_{I_M}(e(b))]) \\ &= \max(1, 0) \\ &= 1 \end{aligned}$$

On the other hand, for $e(a) = 0.5$ and $e(b) = 0.5$ we obtain

$$\begin{aligned} e(\phi) &= \max(\min[e(a), e(b)], \max[N_{I_M}(e(a)), N_{I_M}(e(b))]) \\ &= \max(0.5, 0) \\ &= 0.5 \end{aligned}$$

and using the same evaluation $e(a) = 0.5$ and $e(b) = 0.5$, in product logic we have

$$\begin{aligned} e(\phi) &= \min[e(a), e(b)] \cdot \max[N_{I_P}(e(a)), N_{I_P}(e(b))] \\ &= 0.5 \cdot 0 \\ &= 0 \end{aligned}$$

The following set of axioms together with the deduction rule modus ponens (from ϕ and $\phi \to \psi$, infer ψ) is a sound and complete axiomatisation of the logic BL. In [Chvalovský 2012] it is shown that these axioms form a minimal independent set.

(BL1) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$

(BL4) $\varphi \otimes (\varphi \to \psi) \to \psi \otimes (\psi \to \varphi)$

(BL5a) $(\varphi \otimes \psi \to \chi) \to (\varphi \to (\psi \to \chi))$

(BL5b) $(\varphi \to (\psi \to \chi)) \to (\varphi \otimes \psi \to \chi)$

(BL6) $((\varphi \to \psi) \to \chi) \to (((\psi \to \varphi) \to \chi) \to \chi)$

(BL7) $\overline{0} \to \varphi$

The numbering of the axioms is inherited from the original numbering in [Hájek 1998] which included two more axioms which later proved to be redundant [Chvalovský 2012].

By adding axioms to the axiomatic system of BL we then obtain sound and complete axiomatisations for Gödel, product and Łukasiewicz logic.

Proposition 2.6: [Hájek 1998]

- BL and $\varphi \to \varphi \otimes \varphi$ is a sound and complete axiomatisation for Gödel logic
- BL and $\sim\varphi \vee ((\varphi \to \varphi \otimes \psi) \to \psi)$ is a sound and complete axiomatisation for product logic
- BL and $\sim\sim\varphi \to \varphi$ is a sound and complete axiomatisation for Łukasiewicz logic

For any axiomatic extension L of BL, not restricted to Gödel, product or Łukasiewicz logic, we have the following deduction theorem. For a set of formulas $\Gamma \cup \{\varphi, \psi\}$ we have

$$\Gamma, \varphi \vdash_L \psi \text{ iff there exists a natural number } n \text{ such that } \Gamma \vdash_L \varphi^n \to \psi$$

where φ^n is a short notation for the conjunction of $\varphi \otimes \ldots \otimes \varphi$ (n times) and where $\vdash_L$ denotes the notion of proof in L. In particular it holds that $\Upsilon \vdash_L \alpha$ if there exists a finite sequence of formulas whose last member is α and for which every element in the sequence is (i) an axiom in L (ii) an element in Υ or (iii) is derived from previous elements in the sequence by modus ponens. Gödel logic is the only case for which the classical deduction theorem, i.e. where $n = 1$, holds.

Example 2.8

One can show that in Łukasiewicz logic we have the following theorems:

- $\sim(\varphi \otimes \psi) \leftrightarrow (\sim\varphi \oplus \sim\psi)$
- $\sim(\varphi \oplus \psi) \leftrightarrow (\sim\varphi \otimes \sim\psi)$
- $\sim(\varphi \wedge \psi) \leftrightarrow (\sim\varphi \vee \sim\psi)$
- $\sim(\varphi \vee \psi) \leftrightarrow (\sim\varphi \wedge \sim\psi)$

Notice that these are the de Morgan laws. The last two are also theorems in Gödel and product logic. On the other hand

- $\sim\sim\varphi \leftrightarrow \varphi$ is a theorem in Łukasiewicz logic but not in Gödel nor product logic.

Finally we remark that classical (Boolean) logic is an extension of BL by adding the single axiom

$$\varphi \vee \sim\varphi$$

2.2.3 Łukasiewicz logic

Since in this thesis we will often use Łukasiewicz logic, we will explicitly write down its connectives and corresponding semantics. Its formulas are built from a countable set of atoms A and the connectives conjunction $\otimes$, disjunction $\otimes$, minimum $\wedge$, maximum $\vee$, implication $\rightarrow$, equivalence $\leftrightarrow$ and negation $\sim$. Then for an evaluation $e : A \rightarrow [0,1]$ we have

$$\begin{aligned} e(\phi \otimes \psi) &= \max(e(\phi) + e(\psi) - 1, 0) & e(\phi \oplus \psi) &= \min(e(\phi) + e(\psi), 1) \\ e(\phi \wedge \psi) &= \min(e(\phi), e(\psi)) & e(\phi \vee \psi) &= \max(e(\phi), e(\psi)) \\ e(\phi \rightarrow \psi) &= \min(1 - e(\phi) + e(\psi), 1) & e(\phi \leftrightarrow \psi) &= e((\phi \rightarrow \psi) \otimes (\psi \rightarrow \phi)) \\ e(\sim\phi) &= 1 - e(\phi) \end{aligned}$$

with ϕ and ψ formulas. Note that we also have $e(\phi \leftrightarrow \psi) = \min((e(\phi \rightarrow \psi), e(\psi \rightarrow \phi))$ and for formulas $\phi_1, \ldots, \phi_n$ we have

$$\begin{aligned} e(\bigotimes_{i=1}^{n} \phi_i) &= e(\phi_1 \otimes \ldots \otimes \phi_n) = \max(\sum_{i=1}^{n} e(\phi_i) - (n-1), 0), \\ e(\bigoplus_{i=1}^{n} \phi_i) &= e(\phi_1 \oplus \ldots \oplus \phi_n) = \min(\sum_{i=1}^{n} e(\phi_i), 1) \end{aligned}$$

An important logic is the addition to Łukasiewicz logic of truth constants for all rational numbers in $[0,1]$. The idea of using truth constants denoting truth degrees from $[0,1]$ goes back to Pavelka [Pavelka 1979]. He used truth constants for all reals in $[0,1]$ but later (e.g. [Hájek 1998]) it turned out that for the main results to hold without forcing the language to be uncountable, it is sufficient to introduce only truth constants for the rationals.

Definition 2.11: [Hájek 1998]

The *Rational Pavelka logic* is the expansion of Łukasiewicz logic with truth constants $\overline{c}$ for each $c \in [0,1] \cap \mathbb{Q}$ and the "bookkeeping axioms" for all $c, d \in [0,1] \cap \mathbb{Q}$

$$\overline{c} \otimes \overline{d} \leftrightarrow \overline{T_L(c,d)} \quad (\overline{c} \rightarrow \overline{d}) \leftrightarrow \overline{I_{T_L}(c,d)}$$

Semantically, we obtain that $e(\overline{c}) = c$ for an evaluation $e : A \to [0,1]$ and $c \in [0,1] \cap \mathbb{Q}$.

Important properties of this logic are that it maintains the same deduction theorem as Łukasiewicz logic and a property that is called *Pavelka-style completeness* [Hájek 1998]. Let $T \cup \{\varphi\}$ be a (possibly infinite) set of formulas then

$$\inf\{e(\varphi) \mid e \models T\} = \sup\{c \mid T \vdash_{RPL} \overline{c} \to \varphi\}$$

where we write $\vdash_{RPL}$ to denote the provability relation in Rational Pavelka logic.

2.2.4 Finitely-valued Łukasiewicz logic with a finite set of truth constants

Finally, we will formally introduce a variant of Rational Pavelka logic, i.e. the addition of a finite set of truth constants to Łukasiewicz logic.

Consider the propositional language $\mathcal{L}_k^c$ whose formulas are built from a countable set of propositional variables or atoms A, the connective $\to$ (implication) and truth constants $\overline{c}$ for each $c \in S_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$ for some fixed $k \in \mathbb{N}$. Further connectives are defined as follows:

$$\begin{aligned}
{\sim}\phi &= \phi \to \overline{0} & \phi \wedge \psi &= \phi \otimes (\phi \to \psi)\\
\phi \otimes \psi &= {\sim}(\phi \to {\sim}\psi) & \phi \oplus \psi &= {\sim}({\sim}\phi \otimes {\sim}\psi)\\
\phi \vee \psi &= ((\phi \to \psi) \to \psi) & \phi \leftrightarrow \psi &= (\phi \to \psi) \otimes (\psi \to \phi)
\end{aligned}$$

with ϕ and ψ arbitrary formulas. A *propositional evaluation* is a mapping $e : A \to S_k$ that is extended to formulas as follows. If ϕ and ψ are formulas and c is an element in S_k, then

$$e(\phi \to \psi) = I_L(e(\phi), e(\psi)) \qquad\qquad e(\overline{c}) = c.$$

The set of all such evaluations will be denoted by Ω_k. Notice that, in particular, for every formula ϕ and ψ and for every $e \in \Omega_k$, we have

$$\begin{aligned}
e({\sim}\phi) &= 1 - e(\phi) & e(\phi \wedge \psi) &= \min(e(\phi), e(\psi))\\
e(\phi \otimes \psi) &= \max(e(\phi) + e(\psi) - 1, 0) & e(\phi \oplus \psi) &= \min(1, e(\phi) + e(\psi))\\
e(\phi \vee \psi) &= \max(e(\phi), e(\psi)) & e(\phi \leftrightarrow \psi) &= 1 - |(e(\phi) - e(\psi)|
\end{aligned}$$

A formula ϕ is said to be *satisfiable* if there exists $e \in \Omega_k$ such that $e(\phi) = 1$. In such a case we say that e is a *model* of ϕ. A *tautology* is a formula ϕ such that $e(\phi) = 1$ for each propositional evaluation $e \in \Omega_k$. A formula ϕ is a *semantic consequence* of a set of formulas Γ, written as $\Gamma \models \phi$ iff it holds that if $e \in \Omega_k$ is a model of each formula in Γ, then e is also a model of ϕ.

This logic based on the language $\mathcal{L}_k^c$, which we will denote by $Ł_k^c$, has a sound and a strongly complete axiomatisation, see e.g. [Cignoli et al. 2000] for details. In particular, the axioms of $Ł_k^c$ are

(Ł1) $\varphi \to (\psi \to \varphi)$,

(Ł2) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$,

(Ł3) $((\varphi \to \overline{0}) \to (\psi \to \overline{0})) \to (\psi \to \varphi)$,

(Ł4) $((\varphi \to \psi) \to \psi) \to ((\psi \to \varphi) \to \varphi)$,

(Ł5) $(k-1)\varphi \leftrightarrow k\varphi$,

(Ł6) $(l\varphi^{l-1})^k \leftrightarrow k\varphi^l$ for each natural number $l \in \{2, \dots, k-2\}$ that does not divide $k-1$

(Q1) $(\overline{c_1} \to \overline{c_2}) \leftrightarrow \overline{\min\{1, 1-c_1+c_2\}}$

and the only deduction rule is modus ponens (from φ and $\varphi \to \psi$ infer ψ). Axioms (Ł1)-(Ł4) form an alternative axiomatisation for (infinitely-valued) Łukasiewicz logic without truth-constants, and in axioms (Ł5) and (Ł6), $n\varphi$ is an abbreviation for $\varphi \oplus \dots \oplus \varphi$ ($n \in \mathbb{N}$ times) and φ^l for $\varphi \otimes \dots \otimes \varphi$ ($l \in \mathbb{N}$ times). Axiom (Q1) is a bookkeeping axiom for truth-constants. So if $\vdash$ denotes the notion of proof defined from the set of axioms of $Ł_k^c$ and modus ponens, then for any (possibly infinite) set of formulas $T \cup \{\psi\}$, it holds that $T \vdash \psi$ iff $T \models \psi$. A formula ψ that can be proven using only axioms and modus ponens is called a *theorem* and we will write this as $\vdash \psi$.

2.3 Minimal modal logic over finitely-valued Łukasiewicz logic with a finite set of truth constants

In this section we formally introduce the minimal modal logic over Łukasiewicz logic with a finite set of truth constants. In Chapter 6 we will extend this logic to fuzzy modal logics of belief and investigate the relationships with the fuzzy autoepistemic logic which will be introduced in Chapter 5. Moreover, we will use these results to provide an axiomatisation for fuzzy logic of only knowing.

The modal language $\mathcal{L}_{\mathrm{B}}^k$ which we will consider is the expansion of $\mathcal{L}_k^c$ by the modal operator B denoting "belief".

Definition 2.12

The language $\mathcal{L}_{\mathrm{B}}^k$ is recursively defined as follows

- $a \in A$ is a formula.
- $\overline{c}$ with $c \in S_k$ is a formula.
- If α is a formula, then $\mathrm{B}\alpha$ is a formula.
- If α and β are formulas, then $\alpha \to \beta$ with $\to$ the Łukasiewicz implication is a formula.

In [Bou et al. 2011b], where fuzzy modal logics with truth-values forming a finite residuated lattice are discussed, the authors introduce the minimal modal logic over $Ł_k^c$ (see Section 2.2.4). Its axioms are all the axioms of $Ł_k^c$ and

(B2) $(\mathrm{B}\varphi \wedge \mathrm{B}\psi) \to \mathrm{B}(\varphi \wedge \psi)$,

(B3) $\mathrm{B}(\overline{c} \to \varphi) \leftrightarrow (\overline{c} \to \mathrm{B}\varphi)$, for each $c \in S_k$,

(B4) $(\mathrm{B}\varphi \oplus \mathrm{B}\varphi) \leftrightarrow \mathrm{B}(\varphi \oplus \varphi)$.

The rules are modus ponens (from ϕ and $\phi \to \psi$ infer ψ) and monotonicity for B (if $\phi \to \psi$ is a theorem then $\mathrm{B}\phi \to \mathrm{B}\psi$ is a theorem as well).

In [Bou et al. 2011b], the authors show that this is a sound and complete axiomatisation with respect to the class of *Kripke models* [4] $M = (W, e, R)$ where W is a set of possible worlds, $e : W \times A \to S_k$ is a mapping giving an evaluation $e(w, \cdot) : A \to S_k$ for each possible world w and $R : W \times W \to S_k$ is a S_k-valued binary relation on possible worlds. Given a Kripke model $M = (W, e, R)$ and a world $w \in W$, the truth value of a formula in $\mathcal{L}_{\mathrm{B}}^k$ is inductively defined as follows:

- $\|p\|_{M,w} = e(w, p)$,
- $\|\overline{c}\|_{M,w} = c$,
- $\|\mathrm{B}\alpha\|_{M,w} = \inf\{I_L(R(w, w'), \|\alpha\|_{M,w'}) \mid w' \in W\}$,
- $\|\alpha \to \beta\|_{M,w} = I_L(\|\alpha\|_{M,w}, \|\beta\|_{M,w})$.

for an atom p, a truth constant $\overline{c}$ and formulas α and β.

The definition of $\|\mathrm{B}\alpha\|_{M,w}$ intuitively expresses that α is believed in a world $w \in W$ to the degree that α is "at least" true in all worlds w' that are accessible (related to) from w taking into account the degree of the accessibility. A formula ϕ is said to be *satisfiable* if there exists a Kripke model $M = (W, e, R)$ and a $w \in W$ such that $\|\phi\|_{M,w} = 1$. In such

[4] In modal logics it is a convention to use the term models for such structures even if there is no "evaluation to truth value 1".

a case we say that M is a model of ϕ. A set of formulas T is satisfied by a Kripke model M if every formula in T is satisfied by M. It is called *a tautology* if for each Kripke model $M = (W, e, R)$ and for each $w \in W$ we have $\|\phi\|_{M,w} = 1$. A formula ϕ is a *semantic consequence* of a set of formulas Γ, written as $\Gamma \models_{\mathrm{B}} \phi$, if $\|\psi\|_{M,w} = 1$ for all $\psi \in \Gamma$, for all Kripke models $M = (W, e, R)$ and for all $w \in W$ implies $\|\phi\|_{M,w} = 1$ for all Kripke models $M = (W, e, R)$ and for all $w \in W$.

As was shown in [Bou et al. 2011b], the well-known (classical) axiom

(K) $\mathrm{B}(\phi \to \psi) \to (\mathrm{B}\phi \to \mathrm{B}\psi)$

is not generally sound in the above Kripke models. Axiom (K) is only sound in Kripke models $M = (W, e, R)$ where R is a crisp relation on M (i.e. when $R(w, w') \in \{0, 1\}$ for all $w, w' \in W$). Notice that in such Kripke models, the truth evaluation of $\mathrm{B}\alpha$ in a world $w \in W$ reduces to

$$\|\mathrm{B}\alpha\|_{M,w} = \inf\{\|\alpha\|_{M,w'} \mid R(w, w') = 1\}.$$

In the remainder of the thesis we will be interested in this class of Kripke models with crisp accessibility relations. We will denote this class by $\mathbb{M}$. Moreover we will denote by $\mathrm{BŁ}_k^c$ the axiomatic extension of the minimal modal logic over $\mathrm{Ł}_k^c$ with axiom (K). Due to the presence of axiom (K), the monotonicity rule can be replaced by the usual necessitation rule: if ϕ is a theorem then $\mathrm{B}\phi$ is a theorem as well. Indeed, if $\phi \to \psi$ is a theorem then by necessitation it follows that $\mathrm{B}(\phi \to \psi)$ is a theorem as well. Using axiom (K) and modus ponens we then obtain that $\mathrm{B}\phi \to \mathrm{B}\psi$ is a theorem as well.

For each formula ϕ we define a formula $\Delta\phi = \phi \otimes \ldots \otimes \phi$ (k times). Since we only have $k + 1$ truth values this formula is Boolean. Indeed, it is easy to show that

$$\|\Delta\phi\|_{M,w} = \begin{cases} 1 & \text{if } \|\phi\|_{M,w} = 1 \\ 0 & \text{if } \|\phi\|_{M,w} < 1 \end{cases}$$

Notice that in this way Δ corresponds to the well-known Baaz-Monteiro projection operator, which was introduced independently by Monteiro [Monteiro 1980] and Baaz [Baaz 1996].

3 | Fuzzy answer set programming

In this chapter we introduce some preliminary notions on answer set programming (ASP) in Section 3.1 and present relationships with autoepistemic logic. We will then discuss a combination of ASP and fuzzy logic: fuzzy answer set programming (FASP) in Section 3.2. Finally, in Section 3.3, we will introduce some motivating examples for FASP.

3.1 Answer set programming (ASP)

In this section we will introduce answer set programming (ASP) [Gelfond and Lifschitz 1988]. We will define the syntax and semantics and discuss complexity results for various classes of ASP. In Chapter 4 we will then study the complexity of fuzzy answer set programming (FASP) under Łukasiewicz semantics (Section 2.2.3). We will also recall that answer set programming can be embedded in autoepistemic logic (Section 2.1). Indeed, each ASP program can be seen as a set of formulas in autoepistemic logic and the answer sets of the program correspond to the stable expansions of this set of formulas. In Chapter 5 we will propose a generalisation of autoepistemic logic based on fuzzy logic (see Section 2.2) and we will show that the relation between autoepistemic logic and answer set programming remains valid when generalising both using fuzzy logic.

3.1.1 Syntax and semantics

Suppose A is a countable set of propositional atoms. A *literal* is an atom $a \in A$ or its *strong negation* $\neg a$. Such a literal $\neg a$ is essentially seen as a new literal, which has no connection to a, except for the fact that sets of literals L such that $a \in L$ and $\neg a \in L$ will be called *inconsistent*. *(Strongly) negated literals* are defined as follows: $\neg l := \neg a$ if $l = a$ and $\neg l := a$ if $l = \neg a$ (with $a \in A$). An expression of the form $\text{not}\, l$ with l a literal will be called a *negation-as-failure literal* where "not" is the *negation-as-failure operator*. Intuitively, the expression $\text{not}\, l$ is true if there is no proof that supports l. On the other hand an expression $\neg l$ is true if it can be established that $\neg l$ is true. An ASP program (e.g. [Baral 2003]) is then syntactically defined as follows.

Definition 3.1

A *disjunctive ASP program* is a finite set of rules of the form

$$r : a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \text{not}\, c_1 \wedge \ldots \wedge \text{not}\, c_k,$$

with a_i, b_j, c_l literals and/or the constants $\overline{1}$ (true) or $\overline{0}$ (false) with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$.

The expression

$$a_1 \vee \ldots \vee a_n$$

is called the *head* r_h of r and

$$b_1 \wedge \ldots \wedge b_m \wedge \text{not}\, c_1 \wedge \ldots \wedge \text{not}\, c_k$$

is the *body* r_b of r. In a constraint, i.e. a rule of the form "$\overline{0} \leftarrow \alpha$", the body is unconditionally false. In a fact, i.e. a rule of the form "$\alpha \leftarrow \overline{1}$", the head is unconditionally true.

Different classes of ASP programs are often considered, depending on the type of rules they contain.

Definition 3.2

- If each rule in a disjunctive ASP program has one literal or one constant in the head, it is called a *normal* ASP program.
- A normal ASP program not containing negation-as-failure is called a *definite* ASP program.

- A definite ASP program not containing strong negation with exactly one atom in the head of each rule is called a *simple* ASP program.

We will now define the semantics for ASP. Intuitively, we want to derive information from the program by using forward chaining. For example, the program

$$\begin{array}{rrcl} r_1: & \text{sunny} & \leftarrow & \overline{1} \\ r_2: & \text{raining} & \leftarrow & \overline{1} \\ r_3: & \text{rainbow} & \leftarrow & \text{sunny} \wedge \text{raining} \end{array}$$

has to interpreted as follows. In rules r_1 and r_2 the body "$\overline{1}$" is unconditionally true, hence from rule r_1 we can derive "sunny" and from rule r_2 we can derive "raining". Now in rule r_3 the body becomes "true" and hence we can also derive "rainbow".

We will start by defining so-called interpretations of ASP programs.

Definition 3.3

The set of all atoms appearing in a program P is called the *Herbrand base* $\mathcal{B}_P$. An *interpretation* I of P is any consistent set of literals $I \subseteq \mathcal{L}_P$ with

$$\mathcal{L}_P = \{a \mid a \in \mathcal{B}_P\} \cup \{\neg a \mid a \in \mathcal{B}_P\}$$

and where we say that I is *consistent* if there does not exist an $l \in \mathcal{L}_P$ such that $l \in I$ and $\neg l \in I$. The set of all consistent interpretations $I \subseteq \mathcal{L}_P$ will be denoted by $\mathcal{P}(\mathcal{L}_P)$ and the set of all (consistent) interpretations $I \subseteq \mathcal{B}_P$ will be denoted by $\mathcal{P}(\mathcal{B}_P)$.

Example 3.1

Consider the following ASP program P.

$$\begin{array}{rcl} a & \leftarrow & \neg b \wedge c \\ \neg c & \leftarrow & \text{not}\, a \wedge \text{not}\, \neg d \end{array}$$

Then $\mathcal{B}_P = \{a, b, c, d\}$ and $\mathcal{L}_P = \{a, b, c, d, \neg a, \neg b, \neg c, \neg d\}$.

A literal l is *true in* I, written as $I \models l$, iff $l \in I$. An interpretation $I \in \mathcal{P}(\mathcal{L}_P)$ can be extended to rules as follows:

- $I \models \overline{1}$, $I \not\models \overline{0}$,
- $I \models \text{not}\, l$ iff $I \not\models l$,

- $I \models (\alpha \wedge \beta)$ iff $I \models \alpha$ and $I \models \beta$,
- $I \models (\alpha \vee \beta)$ iff $I \models \alpha$ or $I \models \beta$,
- $I \models (\alpha \leftarrow \beta)$ iff $I \models \alpha$ or $I \not\models \beta$,

with l a literal and α and β relevant expressions.

Intuitively, we want to derive the minimal knowledge from an ASP program, i.e. we are looking for minimal (consistent) sets of literals in the program that should be true in order to model the rules. Hence we are interested in the so-called minimal models of the program.

Definition 3.4

An interpretation $I \in \mathcal{P}(\mathcal{L}_P)$ is called a *model* of a disjunctive ASP program P if $I \models r$ for each rule $r \in P$. A model I of P is *minimal* if there exists no model J of P such that $J \subset I$, i.e. $J \subseteq I$ and $J \neq I$.

An interpretation $I \in \mathcal{P}(\mathcal{L}_P)$ is called an *answer set* of a disjunctive ASP program P without negation-as-failure if it is a minimal model of P.

Example 3.2

Consider the disjunctive ASP program consisting of the rules

$$\begin{aligned} \text{beach} \vee \text{park} \vee \text{forest} &\leftarrow \text{sunny} \\ \text{sunny} &\leftarrow \overline{1} \\ \overline{0} &\leftarrow \text{forest} \end{aligned}$$

This first rule states that if it is sunny then we will either go to the beach, the park or the forest. The second rule is a fact implying that it is sunny and the last rule is a constraint implying that we cannot go to the forest.

This program is a negation-as-failure free disjunctive ASP program that has two minimal models, and hence two answer sets, $I_1 = \{\text{sunny}, \text{beach}\}$ and $I_2 = \{\text{sunny}, \text{park}\}$.

It can be shown that a simple ASP program always has exactly one answer set. It equals the least fixpoint of the *immediate consequence operator* Π_P [Van Emden and Kowalski 1976] which maps interpretations to interpretations and is defined as

$$\Pi_P(I) = \{a \mid (a \leftarrow \beta) \in P \text{ and } I \models \beta\}$$

for an interpretation $I \in \mathcal{P}(\mathcal{L}_P)$. From results in [Tarski 1955] it follows that this fixpoint can be computed by iteratively applying Π_P starting from the empty interpretation $I_0 = \emptyset$ until a fixpoint is found.

Example 3.3

Consider the following simple ASP program P:

$$\begin{array}{rcll} r_1: & a & \leftarrow & \overline{1} \\ r_2: & c & \leftarrow & a \\ r_3: & b & \leftarrow & c \wedge b \end{array}$$

- We start from the empty interpretation $I_0 = \emptyset$.
- After one application of Π_P we obtain the interpretation $I_1 = \Pi_P(I_0) = \{a\}$.
- A second application of Π_P gives us $I_2 = \Pi_P(I_1) = \{a, c\}$.
- A third application of Π_P gives us $I_3 = \Pi_P(I_2) = \{a, c\} = I_2$. This is the least fixpoint of Π_P.

As Example 3.4 shows, for programs containing negation-as-failure, some minimal models may not correspond to the intuition of negation-as-failure.

Example 3.4

Consider the normal ASP program consisting of the rules

$$\begin{array}{lcl} \text{beach} & \leftarrow & \text{sunny} \wedge \mathrm{not}\,\text{rainy} \\ \text{sunny} & \leftarrow & \overline{1} \end{array}$$

This first rule states that if it is sunny and there is no indication that it is raining, then we will go to the beach. The second rule implies that it is sunny.

This program has two minimal models $I_1 = \{\text{sunny}, \text{rainy}\}$ and $I_2 = \{\text{sunny}, \text{beach}\}$. Both models contain knowledge that is not explicitly present in the program. Model I_1 assumes that “rainy” is true and model I_2 assumes that there is no evidence to support that “rainy” is true. Since intuitively, “$\mathrm{not}\,$rainy” is true if there is no proof that “rainy” is true we are only interested in I_2.

To define the semantics for disjunctive ASP programs P that contain negation-as-failure, one starts from a candidate answer set $I \in \mathcal{P}(\mathcal{L}_P)$ and computes the Gelfond-Lifschitz reduct P^I [Gelfond and Lifschitz 1988]. The intuition behind this reduct is to guess an interpretation I such that negation-as-failure is removed from the original program and then to check if I is indeed a minimal model of the reduct. The reduct of P w.r.t. I is defined as follows. All rules in P that are trivially satisfied by the guess I, i.e. the rules containing negation-as-failure literals “$\mathrm{not}\, l$” such that $l \in I$, are discarded. In the remaining rules, all negation-as-failure literals "$\text{not}\, l$" are removed, i.e. replaced by the constant $\overline{1}$, since in these cases I does not provide any proof that l is true.

Definition 3.5

An interpretation $I \in \mathcal{P}(\mathcal{L}_P)$ is called an *answer set* of a disjunctive ASP program P if it is a minimal model of P^I where

$$P^I = \{a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \mid \{c_1, \ldots, c_k\} \cap I = \emptyset,$$

$$(a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \text{not}\, c_1 \wedge \ldots \wedge \text{not}\, c_k) \in P\}\,.$$

An answer set of a disjunctive ASP program P is always a minimal model of P [Baral 2003]. The converse does not necessarily hold as Example 3.4 shows. Also note that, as Examples 3.5 and 3.6 show, a disjunctive ASP can have multiple answer sets or even no answer sets at all.

Example 3.5

Consider the following normal ASP program P.

$$\begin{aligned} a &\leftarrow \text{not}\, b \\ b &\leftarrow \text{not}\, a \end{aligned}$$

For the interpretation $I_1 = \{a\}$, we have that P^{I_1} is equal to

$$a \leftarrow \overline{1}$$

Since I_1 is a minimal model of P^{I_1}, we conclude that I_1 is an answer set of P. Similarly, $I_2 = \{b\}$ is also an answer set of P. One can easily check that I_1 and I_2 are the only answer sets of P.

Example 3.6

Consider the following normal ASP program P.

$$a \leftarrow \text{not}\, a$$

The only interpretation that models P is $I = \{a\}$. But $P^I = \emptyset$ which has $\emptyset$ as its unique minimal model. Hence P has no answer sets.

The following example shows how the graph colouring problem can be translated to an ASP program such that the answer sets of the program correspond to the solutions of the original problem. In Section 3.3 we will show how fuzzy answer set programming which will be defined in Section 3.2 can be used to model a continuous variant of this problem.

Example 3.7

Consider the problem of colouring the vertices of a graph in either red, green or blue such that adjacent nodes are coloured differently. This search problem can be modeled by a normal ASP program P consisting of a generating part

$$\begin{aligned} \mathsf{red}(a) &\leftarrow \text{not}\,\mathsf{green}(a) \wedge \text{not}\,\mathsf{blue}(a) \\ \mathsf{green}(a) &\leftarrow \text{not}\,\mathsf{red}(a) \wedge \text{not}\,\mathsf{blue}(a) \\ \mathsf{blue}(a) &\leftarrow \text{not}\,\mathsf{red}(a) \wedge \text{not}\,\mathsf{green}(a) \end{aligned}$$

for each node a. These rules express that each node should be either red, green or blue (cfr. Example 3.5). The following constraints, which we add for each pair of nodes a and b, then express that two nodes connected by an edge should have a different colour.

$$\begin{aligned} \overline{0} &\leftarrow \mathsf{edge}(a,b) \wedge \mathsf{red}(a) \wedge \mathsf{red}(b) \\ \overline{0} &\leftarrow \mathsf{edge}(a,b) \wedge \mathsf{green}(a) \wedge \mathsf{green}(b) \\ \overline{0} &\leftarrow \mathsf{edge}(a,b) \wedge \mathsf{blue}(a) \wedge \mathsf{blue}(b) \end{aligned}$$

Finally a number of facts, a defining part, is added to the program. If there is an edge between nodes a and b, then we have a rule

$$\mathsf{edge}(a,b) \leftarrow \overline{1}.$$

The answer set semantics then defines the solutions to the program. For instance, if there are three nodes a, b and c such that there is an edge between a and b and one between b and c, then one of the answer sets is

$$\{\mathsf{edge}(a,b), \mathsf{edge}(b,c), \mathsf{red}(a), \mathsf{green}(b), \mathsf{red}(c)\}.$$

Finally, we remark that for each disjunctive ASP program there always exists a disjunctive ASP program not containing strong negation or constraints such that answer sets of these programs are in one-to-one correspondence with each other. This implies that without loss of generality we may assume that a disjunctive ASP program does not contain strong negation or constraints.

Remark 3.1

A disjunctive ASP program P with strong negation can be translated to a disjunctive ASP program P' without strong negation, by replacing each literal of the form $\neg a$ with a new atom a' and adding the constraint $\overline{0} \leftarrow a \wedge a'$. An interpretation $I \in \mathcal{P}(\mathcal{L}_P)$ is an answer set of P iff there exists an answer set $I' \in \mathcal{P}(\mathcal{L}_{P'})$ of P' such that $b \in I$ iff $b \in I'$ and $\neg b \in I$ iff $b' \in I'$ for each atom $b \in \mathcal{B}_P$.

Moreover, constraints can be removed by replacing every constraint $\overline{0} \leftarrow r_h$ by a rule $p \leftarrow r_h \wedge \text{not}\, p$ with p a new atom.

3.1.2 Complexity of ASP

When investigating the computational complexity of answer set programming, we are mainly interested in the following reasoning tasks.

Definition 3.6

Given a disjunctive ASP program P and a literal l, we define the following decision problems.

1. **Existence**: Does P have an answer set?
2. **Set-membership**: Does there exist an answer set I of P such that $l \in I$?
3. **Set-entailment**: Does $l \in I$ hold for each answer set I of P?

The complexity class P is defined as the set of decision problems, i.e. those problems for which the answer is either "yes" or "no", that can be solved in polynomial time on a deterministic Turing machine [Papadimitriou 1994], where the polynomial time bound is a function of the input size. The complexity class NP is defined as the class of decision problems that can be solved in polynomial time on a non-deterministic Turing machine or equivalently the set of the decision problems for which the proof that the answer is "yes" can be verified in polynomial time by a deterministic Turing machine [Papadimitriou 1994]. From these classes, other classes can be defined as follows [Papadimitriou 1994]:

$$\Sigma_0^{\mathsf{P}} = \Pi_0^{\mathsf{P}} = \mathsf{P}$$

$$\Sigma_{i+1}^{\mathsf{P}} = \mathsf{NP}^{\Sigma_i^{\mathsf{P}}}$$

$$\Pi_{i+1}^{\mathsf{P}} = \mathsf{co}(\Sigma_{i+1}^{\mathsf{P}})$$

where $\mathsf{NP}^{\Sigma_i^{\mathsf{P}}}$ is the class of decision problems that can be solved in polynomial time on a non-deterministic Turing machine with a Σ_i^{P}-oracle. This means that one assumes that the Turing machine can call an oracle that is able to solve decision problems that are in Σ_i^{P} in constant time. Finally, the class co($\Sigma_{i+1}^{\mathsf{P}}$) is the class of problems for which the complement, i.e. the problem resulting from reversing the "yes" and "no" answers, is in $\Sigma_{i+1}^{\mathsf{P}}$.

For a complexity class C, a decision problem is called *C-hard*, if every problem in C can be reduced to this problem in polynomial time, i.e. each problem in C can be "translated" to this problem in polynomial time such that the solutions correspond to each other. A problem is said to be *C-complete* if the problem is in C and it is C-hard.

Example 3.8

1. The Boolean satisfiability problem (SAT) is the decision problem of determining for some Boolean formula ϕ whether an assignment of true or false to the variables exists that makes ϕ true. For instance consider

$$\phi = (a_1 \vee a_2) \wedge (\sim a_1 \vee \sim a_2)$$

where $\vee$, $\wedge$ and $\sim$ denote classical disjunction, conjunction and negation, respectively. An assignment that makes ϕ true is for example a_1 false and a_2 true. On the other hand, the formula

$$\psi = (\sim a_1 \vee \sim a_2 \vee a_3) \wedge a_1 \wedge a_2 \wedge \sim a_3$$

is not satisfiable, i.e. there does not exist an assignment of truth values to make ψ true.

SAT is a NP-complete decision problem [Cook 1971]. The unSAT problem is the complementary problem of SAT. This is the problem of verifying, given a Boolean formula ϕ, whether ϕ has no assignment that makes the expression true. This problem is coNP-complete [Cook 1971]. A variant of the SAT problem that remains NP-complete is the 3SAT problem. The instances of this problem are Boolean formulas in conjunctive normal form with three variables in each clause:

$$(a_{11} \vee a_{12} \vee a_{13}) \wedge (a_{21} \vee a_{22} \vee a_{23}) \wedge \ldots \wedge (a_{n1} \vee a_{n2} \vee a_{n3}),$$

where each a_{ij} is a (negated) variable.

2. A generalisation of SAT is the Quantified Boolean Formula (QBF) problem. Here existential and universal quantifiers can be applied to each variable. Let $p(x_1, \ldots, x_n)$ be a propositional formula defined over the variables $x_1, \ldots, x_n$. For instance

$$\phi = \exists x_1 \forall x_2 ((x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2))$$

is not satisfiable. Indeed, if x_1 is true, then $(x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$ is not satisfiable when x_2 is true. On the other hand, if x_1 is false, then $(x_1 \vee x_2) \wedge (\sim x_1 \vee \sim x_2)$ is not satisfiable when x_2 is false. The QBF

$$\psi = \exists x_1 \forall x_2 \exists x_3 ((x_1 \vee x_2 \vee x_3) \wedge (\sim x_1 \vee \sim x_2))$$

is satisfiable since, when x_1 is false and x_3 is true, then $(x_1 \vee x_2 \vee) \wedge (\sim x_1 \vee \sim x_2)$ is true regardless of the assignment of x_2.

Deciding the satisfiability of a QBF

$$\exists x_1 \forall x_2 \ldots \clubsuit x_n p(x_1, \ldots, x_n)$$

with $\clubsuit = \exists$ if n is odd and $\clubsuit = \forall$ if n is even is a Σ_n^{P}-complete problem. Deciding the satisfiability of a QBF

$$\forall x_1 \exists x_2 \ldots \clubsuit x_n p(x_1, \ldots, x_n)$$

with $\clubsuit = \forall$ if n is odd and $\clubsuit = \exists$ if n is even is a Π_n^{P}-complete problem.

Depending on the type of ASP program one considers (see Definition 3.2), the computational complexity of the reasoning tasks varies:

Proposition 3.1: [Baral 2003, Eiter and Gottlob 1993]

Existence is

- Σ_2^{P}-complete for the class of disjunctive ASP programs.
- NP-complete for the class of negation-as-failure free disjunctive ASP programs.
- NP-complete for the class of normal ASP programs.
- P-complete for the class of simple ASP programs.
- P-complete for the class of definite ASP programs.

Proposition 3.2: [Baral 2003, Eiter and Gottlob 1993]

Set-membership is

- Σ_2^{P}-complete for the class of disjunctive ASP programs.
- Σ_2^{P}-complete for the class of negation-as-failure free disjunctive ASP programs.
- NP-complete for the class of normal ASP programs.
- P-complete for the class of simple ASP programs.
- P-complete for the class of definite ASP programs.

Proposition 3.3: [Baral 2003, Eiter and Gottlob 1993]

Set-entailment is

- Π_2^{P}-complete for the class of disjunctive ASP programs.
- coNP-complete for the class of negation-as-failure free disjunctive ASP programs.
- coNP-complete for the class of normal ASP programs.
- P-complete for the class of simple ASP programs.
- P-complete for the class of definite ASP programs.

A summary of the computational complexity for ASP is given in Table 3.1.

Table 3.1: Complexity of inference in ASP [Baral 2003, Eiter and Gottlob 1993]

	existence	set-membership	set-entailment
disjunctive	Σ_2^{P}-complete	Σ_2^{P}-complete	Π_2^{P}-complete
normal	NP-complete	NP-complete	coNP-complete
definite	in P	in P	in P

In Chapter 4 we will investigate whether these complexity results still hold when generalising to fuzzy answer set programming.

3.1.3 Embedding answer set programming into autoepistemic logic

We will now discuss the relationship between answer set programming and autoepistemic logic (Section 2.1). Gelfond [Gelfond and Lifschitz 1988] proposed the following transformation from a normal ASP program P without constraints or strong negation to an autoepistemic theory $\lambda(P)$. For each rule

$$r : a \leftarrow b_1, \ldots, b_m, \text{not}\, c_1, \ldots, \text{not}\, c_k$$

in P, the autoepistemic formula $\lambda(r)$

$$b_1 \wedge \ldots \wedge b_m \wedge {\sim}\mathrm{B}c_1 \wedge \ldots \wedge {\sim}\mathrm{B}c_k \rightarrow a$$

is added to $\lambda(P)$. The following result clarifies the relationship between the answer sets of P and the stable expansions of $\lambda(P)$.

Theorem 3.1: [Gelfond and Lifschitz 1988]

Consider a normal ASP program P without constraints or strong negation. M is an answer set of P iff $\lambda(P)$ has a stable expansion E such that $M = E \cap \mathcal{B}_P$.

By Remark 3.1 and Theorem 3.1 it follows that for each normal ASP program P, even if it contains strong negation and/or constraints, that there exists some autoepistemic theory T such that M is an answer set of P iff T has a stable expansion E such that $M = E \cap \mathcal{B}_P$.

Example 3.9

Consider the normal ASP program P from Example 3.5 with $\mathcal{B}_P = \{a, b\}$:

$$\begin{aligned} b &\leftarrow \text{not}\, a \\ a &\leftarrow \text{not}\, b \end{aligned}$$

This program has two answer sets $M_1 = \{a\}$ and $M_2 = \{b\}$. By Examples 2.2 and 2.3, we know that the sets

$$E_1 = \{\varphi \in \mathcal{L}_{\mathrm{B}}^{c} \mid \forall I \in \mathcal{P}(A) : a \in I \Rightarrow (I, S) \models \varphi\}$$

and

$$E_2 = \{\varphi \in \mathcal{L}_{\mathrm{B}}^{c} \mid \forall I \in \mathcal{P}(A) : b \in I \Rightarrow (I, S) \models \varphi\}$$

are the two stable expansions of $\lambda(P)$. We find $E_1 \cap \mathcal{B}_P = M_1$ and $E_2 \cap \mathcal{B}_P = M_2$.

As Example 3.10 shows, Theorem 3.1 is not valid for more general programs.

Example 3.10

Consider the following disjunctive ASP program P.

$$a \vee b \quad \leftarrow \quad \overline{1}$$

One can easily show that P has two answer sets $\{a\}$ and $\{b\}$. Now consider the autoepistemic theory $\lambda(P) = \{\overline{1} \rightarrow a \vee b\}$. By Proposition 2.1 it follows that $\lambda(P)$ has exactly one stable expansion. Hence the stable expansions of $\lambda(P)$ do not coincide with the answer sets of P.

Now consider the following normal ASP programs. The program P'

$$\begin{array}{rcl} a & \leftarrow & \overline{1} \\ b & \leftarrow & a \end{array}$$

has a unique answer set $\{a, b\}$ and is mapped to the autoepistemic theory $\lambda(P') = \{a, a \rightarrow b\}$. On the other hand, suppose we would translate the program P''

$$\begin{array}{rcl} a & \leftarrow & \overline{1} \\ \neg a & \leftarrow & \neg b \end{array}$$

directly, without using Remark 3.1 to translate it to a normal ASP program without constraints or strong negation, to an autoepistemic theory $\{a, {\sim} b \rightarrow {\sim} a\}$. The program P'' has a unique answer set $\{a\}$ which is different from the answer set of P'. However the corresponding autoepistemic theory is also $\lambda(P') = \{a, a \rightarrow b\}$ since ${\sim} b \rightarrow {\sim} a$ and $a \rightarrow b$ are equivalent formulas in autoepistemic logic. Hence for the latter program the answer sets do not coincide with the stable expansions of the corresponding autoepistemic theory.

Lifschitz and Schwarz [Lifschitz and Schwarz 1993], showed that disjunctive ASP programs (even with strong negation) can also be modelled in autoepistemic logic: For each rule

$$r : a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \text{not}\, c_1 \wedge \ldots \wedge \text{not}\, c_k,$$

in a disjunctive ASP program P, the formula $\sigma(r)$

$$(b_1 \wedge \mathrm{B}b_1) \wedge \ldots \wedge (b_m \wedge \mathrm{B}b_m) \wedge {\sim}\mathrm{B}c_1 \wedge \ldots \wedge {\sim}\mathrm{B}c_k \rightarrow (a_1 \wedge \mathrm{B}a_1) \vee \ldots \vee (a_n \wedge \mathrm{B}a_n)$$

is added to the autoepistemic theory $\sigma(P)$.

Theorem 3.2: [Lifschitz and Schwarz 1993]

Consider a disjunctive ASP program P. M is an answer set of P iff Th(Mod(M)) is a stable expansion of $\sigma(P)$ where Mod$(M) = \{I \in \mathcal{P}(A) \mid M \subseteq I\}$. Equivalently, M is an answer set of P iff Mod(M) is a possible world autoepistemic model of $\sigma(P)$.

Example 3.11

Consider the disjunctive ASP program P from Example 3.10:

$$a \vee b \quad \leftarrow \quad \overline{1}$$

The corresponding autoepistemic theory $\sigma(P)$ is the singleton

$$\{\overline{1} \rightarrow (a \wedge \mathrm{B}a) \vee (b \wedge \mathrm{B}b)\}.$$

One can show that $M = \{a\}$ is an answer set of P by showing that

$$S = \text{Mod}(M) = \{I \mid a \in I\}$$

is a possible world autoepistemic model of $\sigma(P)$. To do this, we have to prove that

$$\text{Mod}(M) = S = \{I \in \mathcal{P}(A) \mid (I, S) \models (\overline{1} \rightarrow (a \wedge \mathrm{B}a) \vee (b \wedge \mathrm{B}b))\}.$$

To do so consider $I \in \mathcal{P}(A)$ and observe the following equalities.

$$\begin{aligned}
& (I, S) \models (\overline{1} \rightarrow (a \wedge \mathrm{B}a) \vee (b \wedge \mathrm{B}b)) \\
\Leftrightarrow \quad & (I, S) \models (a \wedge \mathrm{B}a) \text{ or } (I, S) \models (b \wedge \mathrm{B}b)) \\
\Leftrightarrow \quad & ((I, S) \models a \text{ and } (I, S) \models \mathrm{B}a) \text{ or } ((I, S) \models b \text{ and } (I, S) \models \mathrm{B}b) \\
\Leftrightarrow \quad & (a \in I \text{ and for all } J \in S : a \in J) \text{ or } (b \in I \text{ and for all } J \in S : b \in J) \\
\Leftrightarrow \quad & a \in I \\
\Leftrightarrow \quad & I \in S
\end{aligned}$$

Similar, one can show that $\{b\}$ is an answer set of P as well.

3.2 Fuzzy answer set programming (FASP)

In this section we will define a generalisation of ASP based on [Janssen et al. 2009] which is obtained by combining ASP (Section 3.1) and fuzzy logics (Section 2.2). The resulting framework is more general than discussed in [Janssen et al. 2009].

Suppose A is a countable set of propositional atoms. A *literal* is an atom $a \in A$ or its *strong negation* $\neg a$. Such a literal $\neg a$ is essentially seen as a new literal, which has no connection to a, except for the fact that answer sets containing both a and $\neg a$ "to a sufficiently high degree" will be designated as *inconsistent*. As for ASP, *(strongly) negated literals* are defined as follows: $\neg l := \neg a$ if $l = a$ and $\neg l := a$ if $l = \neg a$ (with $a \in A$). An expression of the form $\text{not}\, l$ (with l a literal) will be called a *negation-as-failure literal* where "not" is the *negation-as-failure operator*. A FASP program is then syntactically defined as follows.

Definition 3.7

A *regular FASP program* is a finite set of rules of the form

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \text{not}_1\, c_1, \ldots, \text{not}_\text{k}\, c_k),$$

with a_i, b_j, c_l literals and/or truth constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$. The function g is a prefixnotation for an n-ary connective and f represents an $(m+k)$-ary connective corresponding to increasing functions $\mathtt{g} : [0,1]^n \to [0,1]$ and $\mathtt{f} : [0,1]^{m+k} \to [0,1]$. The negation-as-failure operators not_l correspond to negators N_L and $\leftarrow$ corresponds to some residual implicator I_T.

The expression $g(a_1, \ldots, a_n)$ is called the *head* r_h of r and $f(b_1, \ldots, b_m, \text{not}_1\, c_1, \ldots,$ $\text{not}_\text{k}\, c_k)$ is the *body* r_b of r. A rule of the form $\overline{c} \leftarrow r_b$ with $\overline{c}$ a truth constant is called a *constraint* and a rule of the form $r_h \leftarrow \overline{c}$ a *fact*. As will become clear when defining the semantics, in a constraint $\overline{c} \leftarrow r_b$ the truth value of r_b cannot be greater than c and in a fact $r_h \leftarrow \overline{c}$, the truth value of r_h has to be greater than or equal to c. Typically, in FASP rules, we will use connectives from a given fuzzy logic (see Section 2.2), but other choices e.g. averaging operators can be useful as well. The only requirement is that f and g correspond to increasing functions.

As for ASP, different classes of regular FASP programs are often considered, depending on the type of rules they contain.

Definition 3.8

- If each rule in a regular FASP program has one literal or one constant in the head, it is called a *regular normal* FASP program.
- A regular normal FASP program not containing negation-as-failure is called a *regular definite* FASP program.
- A regular definite FASP program not containing strong negation with exactly one atom in the head of each rule is called a *regular simple* FASP program.

In Chapter 4 we will consider a special subclass of regular FASP. As Definition 3.9 shows, these programs have the same syntax as ASP programs but use the Łukasiewicz connectives. For this reason we will call these types of FASP programs "strict" FASP programs. When, as in Definition 3.7, more syntactical freedom is allowed we call them "regular" FASP programs.

Definition 3.9

A *strict disjunctive FASP program* is a finite set of rules of the form

$$r : a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m \otimes \text{not}\, c_1 \otimes \ldots \otimes \text{not}\, c_k$$

with a_i, b_j, c_l literals and/or truth constants corresponding to truth values in $[0, 1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$ where $\otimes$ is the Łukasiewicz conjunction, $\oplus$ the Łukasiewicz disjunction and not corresponds to the Łukasiewicz negation.

- If each rule in a strict disjunctive FASP program has one literal or one constant in the head, it is called a *strict normal* FASP program.
- A strict normal FASP program not containing negation-as-failure is called a *strict definite* FASP program.
- A strict definite FASP program not containing strong negation with exactly one atom in the head of each rule is called a *strict simple* FASP program.

We will now define the semantics for FASP. We will start by defining so-called fuzzy interpretations of FASP programs.

Definition 3.10

The set of all atoms appearing in a regular FASP program P is called the *Herbrand base* $\mathcal{B}_P$. A *fuzzy interpretation* I of P is an element of $\mathcal{F}(\mathcal{L}_P)$, where $\mathcal{F}(\mathcal{L}_P)$ is the set of all mappings $I : \mathcal{L}_P \to [0,1] \cap \mathbb{Q}$ such that $I(l) + I(\neg l) \leq 1$ for each $l \in \mathcal{L}_P$ with

$$\mathcal{L}_P = \{a \mid a \in \mathcal{B}_P\} \cup \{\neg a \mid a \in \mathcal{B}_P\}.$$

The set $\mathcal{F}(\mathcal{B}_P)$ contains all fuzzy interpretations $I : \mathcal{B}_P \to [0,1] \cap \mathbb{Q}$.

Note that this definition of consistency $(I(l) + I(\neg l) \leq 1$ for all $l \in \mathcal{L}_P)$ generalises the classical definition from Section 3.1 and coincides with the approach in [Madrid and Ojeda-Aciego 2011]. Moreover, for the Łukasiewicz t-norm it holds that $T_L(x, y) = 0$ iff $x + y \leq 1$.

A fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ can be extended to rules as follows:

- $[\overline{c}]_I = c$,
- $[\mathrm{not_j}\, l]_I = N_j(l)$ for the corresponding negator N_j,
- $[f(\alpha_1, \ldots, \alpha_n)]_I = \mathtt{f}([\alpha_1]_I, \ldots, [\alpha_n]_I)$ where f is a prefixnotation for some n-ary connective and $\mathtt{f} : [0,1]^n \to [0,1]$ is the corresponding increasing function (see e.g. Section 2.2),
- $[\alpha \leftarrow \beta]_I = I_T([\beta]_I, [\alpha]_I)$ for the corresponding residual implicator I_T,

with l a literal, $c \in [0,1] \cap \mathbb{Q}$ and $\alpha, \alpha_1, \ldots, \alpha_n$ and β relevant expressions.

Definition 3.11

A fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ is called a *fuzzy model* of a regular FASP program P if $[r]_I = 1$ for each rule $r \in P$. A fuzzy model I of P is *minimal* if there exists no fuzzy model J of P such that $J < I$, i.e. $J(l) \leq I(l)$ for all $l \in \mathcal{L}_P$ and $J \neq I$.

Note that if I is a fuzzy model of a regular FASP program P, then by Proposition 2.5 for each rule $r_h \leftarrow r_b$ in P it holds that $[r_h]_I \geq [r_b]_I$.

Example 3.12

Consider the following strict normal FASP program P.

$$\begin{aligned} r_1 :&\quad \text{open} \leftarrow \mathrm{not}\, \text{closed} \\ r_2 :&\quad \text{closed} \leftarrow \mathrm{not}\, \text{open} \end{aligned}$$

where "$\leftarrow$" and "not" correspond to resp. the Łukasiewicz implicator and the Łukasiewicz negator. The properties "open" and "closed" can be given a value in $[0, 1] \cap \mathbb{Q}$ depending on the extent, e.g. the angle, to which a door is opened resp. closed. Each combination of degrees of "open" and "closed", also those which are not meaningful, is represented by a fuzzy interpretation. The rule r_1 intuitively means that the door is open to a degree greater than or equal to the extent to which the door is not closed. Rule r_2 implies the opposite property. Specifically, a fuzzy interpretation I models P iff

$$\begin{aligned} I(\text{open}) &\geq 1 - I(\text{closed}) \\ I(\text{closed}) &\geq 1 - I(\text{open}). \end{aligned}$$

By considering for example the rule

$$r_3 : \text{open} \quad \leftarrow \quad 0.6$$

we add the information that the door must be open to at least degree 0.6. The minimal fuzzy model of the program only containing rule r_3 is the fuzzy interpretation I such that $I(\text{open}) = 0.6$. A minimal fuzzy model for the program consisting of rules r_1, r_2 and r_3 is for instance $I(\text{open}) = 0.7$ and $I(\text{closed}) = 0.3$. Another possibility is $J(\text{open}) = 0.6$ and $J(\text{closed}) = 0.4$.

We will now introduce the semantics for regular FASP programs. Similar as for ASP, we want to implement the intuition of forward chaining. Hence we want to assign minimal truth values to the literals in a program and not include any more information than what is needed to satisfy the rules. For negation-as-failure free programs this means we are looking for minimal fuzzy models.

Definition 3.12

A fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ is called an *answer set* of a regular FASP program P without negation-as-failure if it is a minimal fuzzy model of P.

Using a result in [Tarski 1955] and the fact that the function that interprets the connectives in the body of a rule is increasing, it can be shown that a regular simple FASP program P always has a minimal fuzzy model. Moreover this answer set is unique.

Proposition 3.4: [Damásio and Pereira 2001]

The unique answer set of a regular simple FASP program P equals the least fixpoint of the immediate consequence operator Π_P which is defined as

$$\Pi_P(I)(a) = \sup\{[r_b]_I \mid (a \leftarrow r_b) \in P\},$$

for an atom $a \in \mathcal{B}_P$ and $I \in \mathcal{F}(\mathcal{B}_P)$.

Due to results in [Tarski 1955], this least fixpoint can be computed using an iterated fixpoint computation which is illustrated in the following example.

Example 3.13

Consider the following regular simple FASP program P.

$$\begin{array}{rccl} r_1: & a & \leftarrow & \overline{0.1} \\ r_2: & b & \leftarrow & \overline{0.8} \\ r_3: & c & \leftarrow & a \oplus b \\ r_4: & a & \leftarrow & b \otimes_P c \end{array}$$

where $\oplus$ is the Łukasiewicz disjunction and $\otimes_P$ is the conjunction from product logic.

- We start from the fuzzy interpretation $I_0 : \mathcal{B}_P \to [0,1] : a \mapsto 0$.
- After one application of Π_P we obtain the fuzzy interpretation $I_1 = \Pi_P(I_0)$ defined as $I_1(a) = 0.1$, $I_1(b) = 0.8$ and $I_1(c) = 0$.
- A second application gives us $I_2 = \Pi_P(I_1)$ defined as $I_2(a) = 0.1$, $I_2(b) = 0.8$ and $I_2(c) = 0.9$.
- A third application gives us $I_3 = \Pi_P(I_2)$ defined as $I_3(a) = 0.72$, $I_3(b) = 0.8$ and $I_3(c) = 0.9$.
- A fourth application gives us $I_4 = \Pi_P(I_3)$ defined as $I_4(a) = 0.72$, $I_4(b) = 0.8$ and $I_4(c) = 1$.
- A final application gives us the least fixpoint $I_5 = \Pi_P(I_4)$ of P defined as $I_5(a) = 0.8$, $I_5(b) = 0.8$ and $I_5(c) = 1$.

Note that, unlike ASP, it is not always possible to compute this fixpoint in a finite number of steps. Consider for instance the program containing the single rule

$$a \leftarrow f(a),$$

where f is interpreted by the function $\mathtt{f}(x) = \frac{x+1}{2}$. It will take infinitely many steps taken by the immediate consequence operator to find the least fixpoint I which is such that $I(a) = 1$ [Straccia 2006]. However, from [Tarski 1955] it follows that the number of iterations needed will always be countable.

For regular definite FASP programs it can be shown that the least fixpoint of the immediate consequence exists, but there exist programs which have no answer set or even no fuzzy model.

Example 3.14

Consider the normal FASP program P:

$$\begin{aligned} a &\leftarrow \overline{1} \\ \overline{0} &\leftarrow a \end{aligned}$$

The least fixpoint of Π_P is the fuzzy interpretation I with $I(a) = 1$. However, P has no fuzzy model since there cannot exist a fuzzy interpretation J such that $J(a) \geq 1$ and $0 \geq J(a)$.

For a negation-as-failure free regular FASP program which is not a regular normal FASP program, there can be several minimal fuzzy models or none at all. In any case, the answer sets represent the minimal knowledge that can be derived from the program.

Example 3.15

Consider the following strict disjunctive program P.

$$\begin{aligned} a \oplus b &\leftarrow \overline{0.3} \\ a &\leftarrow b \\ b &\leftarrow \overline{0.1} \end{aligned}$$

A minimal fuzzy model of P is the fuzzy interpretation I with $I(a) = 0.2$ and $I(b) = 0.1$. However, for instance I' with $I'(a) = 0.15$ and $I'(b) = 0.15$ is a minimal fuzzy model as well. On the other hand, the program

$$\begin{aligned} a \wedge \overline{0.5} &\leftarrow b \\ b &\leftarrow \overline{0.1} \oplus a \end{aligned}$$

has no (minimal) fuzzy models. Indeed, for a fuzzy interpretation I to model this program it must hold that $\min(0.1 + I(a), 1) \leq \min(I(a), 0.5)$. If $0.1 + I(a) \leq 1$, this would imply that $0.1 + I(a) \leq I(a)$ and if $0.1 + I(a) \geq 1$, then it would follow that $1 \leq 0.5$. In both cases, we have a contradiction.

As Example 3.16 shows, for programs containing negation-as-failure, the minimal fuzzy models do not necessarily correspond to the intuition of forward chaining.

Example 3.16

Consider the strict FASP program consisting of the rules

$$\begin{array}{rcl} a & \leftarrow & a \\ \overline{0} & \leftarrow & \text{not}\, a \end{array}$$

The only minimal fuzzy model is I such that $I(a) = 1$. However, the justification for deriving a truth value for a only depends on a itself, so this fuzzy model does not correspond to the intuition of forward chaining.

For programs with negation-as-failure, a generalisation of the Gelfond-Lifschitz reduct [Janssen et al. 2009] is used. In particular, for a program P and a fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ the reduct P^I of P w.r.t. I is obtained by replacing in each rule $r \in P$ all expressions of the form $\text{not}\, l$ by the interpretation $\overline{[\text{not}\, l]_I}$ [1]. For a literal l, we write $l^I = l$ and $(\text{not}\, l)^I = \overline{[\text{not}\, l]_I}$ and we write α^I for a head or a body α of a rule and r^I for a rule r in which all expressions of the form $\text{not}\, l$ have been replaced by the interpretation $\overline{[\text{not}\, l]_I}$. This new program $P^I = \{r^I \mid r \in P\}$ is a regular negation-as-failure free FASP program. I is called an answer set of P if I is an answer set of P^I.

We will assume that for each $l \in \mathcal{L}_P$ in some regular FASP program P, we have that l is in the head of some rule in P. This a reasonable assumption since literals only appearing in the body of rules will have truth value 0 or 1 in each answer set. We can now formulate the following definition.

Definition 3.13

A fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ is called an *answer set* of a regular FASP program P if it is a minimal fuzzy model of $P^I = \{r^I \mid r \in P\}$.

Example 3.17

Intuitively, we "guess" an answer set I and replace all negation-as-failure literals $\text{not}\, l$ by the truth constant $\overline{[\text{not}\, l]_I}$ representing their fuzzy interpretation. For the program

[1] Expressions corresponding to truth values $c \in [0, 1] \cap \mathbb{Q}$ will always be written as $\overline{c}$

from Example 3.12

$$\begin{array}{rcl} \text{open} & \leftarrow & \text{not}\,\text{closed} \\ \text{closed} & \leftarrow & \text{not}\,\text{open} \end{array}$$

a suitable guess would be I with $I(\text{open}) = 0.6$ and $I(\text{closed}) = 0.4$; a door is closed to the degree 0.4 if it is opened to the degree $1 - 0.4 = 0.6$. Let us now consider the same program, but we replace "$\text{not}\,$closed" and "$\text{not}\,$open" by their fuzzy interpretations under I:

$$\begin{array}{rcl} \text{open} & \leftarrow & \overline{0.6} \\ \text{closed} & \leftarrow & \overline{0.4} \end{array}$$

The minimal fuzzy model of this program is exactly I. Hence, I was a "stable" guess and we say that it is an answer set of the program.

Note that I_x with $I_x(\text{open}) = x$ and $I_x(\text{closed}) = 1 - x$ is a stable guess for any $x \in [0, 1] \cap \mathbb{Q}$.

A regular FASP program can have several answer sets as in Example 3.17, or none at all, as in Example 3.18.

Example 3.18

Consider the regular FASP program P consisting of the one rule

$$p \leftarrow \text{not}_M\, p$$

where "not_M" is interpreted by the Gödel negator $N_{I_M} : [0, 1] \to [0, 1]$ with $N_{I_M}(x) = 0$ if $x > 0$ and $N_{I_M}(0) = 1$. For each fuzzy interpretation I with $I(p) > 0$ we have that P^I is is the negation-as-failure free program consisting of the rule

$$p \leftarrow \overline{0}.$$

The unique minimal fuzzy model of P^I is J with $J(p) = 0$, hence our original guess I is clearly not a minimal fuzzy model of P^I. If, on the other hand, we start with the fuzzy interpretation $J(p) = 0$, then we obtain for P^J the rule

$$p \leftarrow \overline{1}.$$

J is not a fuzzy model of P^J, let alone a minimal fuzzy model. We conclude that P has no answer sets.

However, whether or not this program has an answer set depends on the choice of the negator. For example, if "not_M" is interpreted by the Łukasiewicz negator, then P has an answer set. Indeed, for a guess $M(p) = x$ with $x \in [0, 1] \cap \mathbb{Q}$, we now obtain for P^M the rule

$$p \leftarrow \overline{1 - x}.$$

Hence, M is the minimal fuzzy model of P^M if $x = 1 - x$ or $x = 0.5$.

By Definition 3.13 it follows that an answer set of a regular FASP program P is also a fuzzy model of P. One can even prove that it must be a minimal fuzzy model of P:

Proposition 3.5

Let P be a regular FASP program and $I \in \mathcal{F}(\mathcal{L}_\mathcal{P})$. If I is an answer set of P, then I is a minimal fuzzy model of P.

Proof. Suppose I is an answer set of P. By Definition 3.13, it follows that I is a fuzzy model of P^I and hence of P. Now suppose there exists a fuzzy model J of P such that $J \leq I$. We show that J is a fuzzy model of P^I. Let

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \text{not}_1\, c_1, \ldots, \text{not}_\text{k}\, c_k)$$

be a rule in P. It holds that J models r^I since f is an increasing function and for $j \in \{1, \ldots, k\}$ it holds that $[\text{not}_\text{j}\, c_j]_I = N_j(I(c_j)) \leq N_j(J(c_j)) = [\text{not}_\text{j}\, c_j]_J$ since N_j is a decreasing function. Indeed:

$$[f(b_1, \ldots, b_m, \overline{[\text{not}_1\, c_1]_I}, \ldots, \overline{[\text{not}_\text{k}\, c_k]_I})]_J \leq$$

$$[f(b_1, \ldots, b_m, \overline{[\text{not}_1\, c_1]_J}, \ldots, \overline{[\text{not}_\text{k}\, c_k]_J})]_J \leq [g(a_1, \ldots, a_n)]_J$$

where the last inequality follows from the fact that J is a fuzzy model of P and hence of P^J. Since I is a minimal fuzzy model of P^I, it then follows that $I = J$. □

As the following example shows, the converse of Proposition 3.5 does not hold. A minimal fuzzy model does not necessarily correspond to an answer set.

Example 3.19

Recall the strict normal FASP program from Example 3.16.

$$\begin{aligned} a &\leftarrow a \\ \overline{0} &\leftarrow \text{not}\, a \end{aligned}$$

with "not" interpreted by the Łukasiewicz negator. The only minimal fuzzy model of the program is the fuzzy interpretation I such that $I(a) = 1$. However it is not an answer set since I is not a minimal fuzzy model of

$$\begin{array}{rcl} a & \leftarrow & a \\ \overline{0} & \leftarrow & \overline{0} \end{array}$$

3.3 Motivating examples

In this section we present some motivating and illustrating examples for FASP. The first example shows how strict disjunctive FASP can be used to model sensor networks. This is followed by an example showing how strict simple FASP can be used to compute transitive closures of proximity relations. We will also define a regular normal FASP program to tackle a version of the ATM location selection problem followed by a regular normal FASP program that can be used to solve the fuzzy graph colouring problem. In all these examples we will use Łukasiewicz logic.

3.3.1 Sensor networks

Forest fires cause massive loss of vegetation and animal life. If a fire is detected on time, suppression units are able to reach the fire in its initial stages which is important to avoid huge losses. Moreover suppression costs will be considerably reduced in such a case. Wireless sensor networks can be effectively used for this purpose [Yu et al. 2005]. These networks consist of a number of devices that can sense their environment and communicate wirelessly. Consider such a wireless sensor network consisting of sensors measuring temperature. Since there could be sensors that are defective, one should not blindly draw conclusions based on the measurements of a single sensor. We will tackle this as follows. Sensors located near to each other should measure similar temperatures. Hence, if a pair of nearby sensors displays significantly different temperatures, we can assume there must be something wrong with at least one of these sensors. We will use FASP to determine whether there are sensors which are not working optimally and if so, within what range we can assume the real temperature to be.

Suppose we have n sensors. By assuming an appropriate linear rescaling, we can see temperature as a value in $[0, 1] \cap \mathbb{Q}$. Although we might not be able to derive an exact temperature, we will try to find a subinterval of $[0, 1]$ in which we could assume the temperature to be. More specifically, for each sensor $i \in \{1, \ldots, n\}$, we denote the lower bound on the exact temperature as the variable t_i. The temperature measured by sensor i

is a fixed value $t'_i \in [0,1] \cap \mathbb{Q}$. If e_i is the variable representing the error on the measured temperature then the actual temperature must be in the interval $[t'_i - e_i, t'_i + e_i]$. In our setting, the fixed value t'_i corresponds to a truth constant $\overline{t'_i}$ in the FASP program and e_i is a variable for which we will infer a value, i.e. the measured temperature is considered given, and we learn a value for the measurement error that is potentially caused by a sensor that is not functioning as it should.

The sensor network defines a weighted graph G as follows. The vertices are the sensors and there is an edge with weight $w_{ij} \in [0,1] \cap \mathbb{Q}$ between the vertices corresponding to sensor i and sensor j, indicating how near these sensors are to each other. The fixed value $w_{ij} \in [0,1] \cap \mathbb{Q}$ is such that we can reasonably assume, based on the locations of sensors i and j that the difference of the exact temperature between these locations should be less than w_{ij}. So the degree to which we can assume that there is something wrong with sensors i and/or j is equal to the degree to which $d(t'_i, t'_j) = |t'_i - t'_j|$ is greater than or equal to w_{ij}.

We can now write the following program P for given (fixed) values $t'_i, w_{ij} \in [0,1] \cap \mathbb{Q}$ and variables t_i, e_i $(i, j \in \{1, \ldots, n\})$.

$$
\begin{array}{lrcl}
r_1: & \neg t_i \oplus \overline{t'_i} & \leftarrow & \text{not}\, e_i \\
r_2: & \overline{1 - t'_i} \oplus t_i & \leftarrow & \text{not}\, e_i \\
r_3: & t_i & \leftarrow & \text{not}\, \neg t_i \\
r_4: & \neg t_i & \leftarrow & \text{not}\, t_i \\
r_5: & e_i \oplus e_j & \leftarrow & d(\overline{t'_i}, \overline{t'_j}) \otimes \overline{1 - w_{ij}}
\end{array}
$$

Rules r_1 and r_2 are obtained as follows. We want to model that (the lower bound on) the actual temperature t_i and the measured temperature t'_i are similar to the degree that we do not know that there is something wrong with sensor i. Hence we want to model the formula $(t_i \leftrightarrow \overline{t'_i}) \leftarrow \text{not}\, e_i$, where $\overline{t'_i}$ is the constant representing the measured temperature in the program. Note that this "rule" does not adhere to the syntax of (strict) disjunctive FASP. However it can be easily rewritten as the two syntactically correct FASP rules r_1 and r_2. Indeed, notice that in Łukasiewicz logic we have

$$t_i \leftrightarrow \overline{t'_i} = \min(\sim t_i \oplus \overline{t'_i}, \overline{1 - t'_i} \oplus t_i).$$

Hence for a fuzzy interpretation I such that $I(\neg t_i) = 1 - I(t_i)$, we have that I models $(t_i \leftrightarrow \overline{t'_i}) \leftarrow \text{not}\, e_i$ iff $[\min(\neg t_i \oplus \overline{t'_i}, \overline{1 - t'_i} \oplus t_i)]_I \geq [\text{not}\, e_i]_I$ iff $[\neg t_i \oplus \overline{t'_i}]_I \geq [\text{not}\, e_i]_I$ and $[\overline{1 - t'_i} \oplus t_i]_I \geq [\text{not}\, e_i]_I$ and hence iff I models r_1, r_2, r_3 and r_4 where rules r_3 and r_4 are needed to obtain that $I(t_i) + I(\neg t_i) = 1$. Recall that, contrary to Łukasiewicz logic, the latter equation is not automatically valid in FASP.

Rule r_5 is justified by that fact that for a fuzzy interpretation I it holds that

$$[d(\overline{t'_i}, \overline{t'_j}) \otimes \overline{1 - w_{ij}}]_I = \max(d(t'_i, t'_j) - w_{ij}, 0).$$

If $d(t'_i, t'_j) \leq w_{ij}$ and no other rules imply $I(e_1) > 0$ or $I(e_2) > 0$, then for an answer set I we obtain $[e_i \oplus e_j]_I = 0$, i.e. there is nothing wrong with the sensors. Otherwise, if $d(t'_i, t'_j) > w_{ij}$, then $[e_i \oplus e_j]_I \geq d(t'_i, t'_j) - w_{ij}$, i.e. there is something wrong with the sensors at least to the degree to which $d(t'_i, t'_j)$ is greater than or equal to w_{ij}.

Consider as a concrete example a network with three sensors as depicted in Figure 3.1. Suppose we have the measurements $t'_1 = 0.4$, $t'_2 = 0.9$ and $t'_3 = 0.5$ and we have $w_{1,2} = w_{1,3} = w_{2,3} = 0.2$, i.e. all the sensors are fairly far apart from each other.

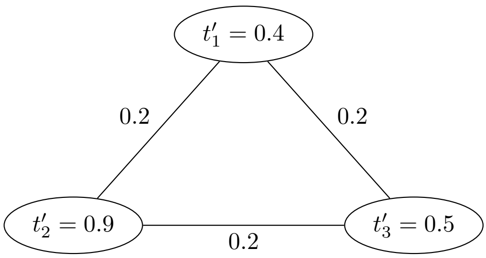


Figure 3.1: Example of a sensor network

The three rules of type r_5 are the following:

$$\begin{array}{rcl} e_1 \oplus e_2 & \leftarrow & 0.3 \\ e_2 \oplus e_3 & \leftarrow & 0.2 \\ e_1 \oplus e_3 & \leftarrow & 0 \end{array}$$

These rules impose lower bounds on $e_i \oplus e_j$ and by computing reducts P^I w.r.t. fuzzy interpretations I meeting these conditions and verifying that I is a minimal fuzzy model of P^I, we obtain for instance the following answer set I.

$$\begin{array}{rclrclrcl} I(e_1) & = & 0.3 & I(e_2) & = & 0 & I(e_3) & = & 0.2 \\ I(t_1) & = & 0.1 & I(t_2) & = & 0.9 & I(t_3) & = & 0.3 \\ I(\neg t_1) & = & 0.9 & I(\neg t_2) & = & 0.1 & I(\neg t_3) & = & 0.7 \end{array}$$

In particular, in this example, P^I is the program containing the following rules.

$$
\begin{array}{rcl@{\qquad}rcl}
\neg t_1 \oplus \overline{0.4} & \leftarrow & \overline{0.7} & \overline{0.6} \oplus t_1 & \leftarrow & \overline{0.7} \\
\neg t_2 \oplus \overline{0.9} & \leftarrow & \overline{1} & \overline{0.1} \oplus t_2 & \leftarrow & \overline{1} \\
\neg t_3 \oplus \overline{0.5} & \leftarrow & \overline{0.8} & \overline{0.5} \oplus t_3 & \leftarrow & \overline{0.8} \\
t_1 & \leftarrow & \overline{0.1} & \neg t_1 & \leftarrow & \overline{0.9} \\
t_2 & \leftarrow & \overline{0.9} & \neg t_2 & \leftarrow & \overline{0.1} \\
t_3 & \leftarrow & \overline{0.3} & \neg t_3 & \leftarrow & \overline{0.7} \\
e_1 \oplus e_2 & \leftarrow & \overline{0.3} & e_1 \oplus e_3 & \leftarrow & \overline{0} \\
e_2 \oplus e_3 & \leftarrow & \overline{0.2} & & &
\end{array}
$$

One can easily check that I is indeed a minimal fuzzy model of P^I.

Another answer set J is defined as follows.

$$
\begin{array}{rcl@{\qquad}rcl@{\qquad}rcr}
J(e_1) & = & 0 & J(e_2) & = & 0.3 & J(e_3) & = & 0 \\
J(t_1) & = & 0.4 & J(t_2) & = & 0.6 & J(t_3) & = & 0.5 \\
J(\neg t_1) & = & 0.6 & J(\neg t_2) & = & 0.4 & J(\neg t_3) & = & 0.5
\end{array}
$$

Notice that there are several answer sets, each corresponding to a possible explanation.

3.3.2 Transitive closure

A proximity relation on a universe X is a mapping $R : X \times X \to [0,1] \cap \mathbb{Q}$ that is reflexive ($R(x,x) = 1$ for each $x \in X$) and symmetric ($R(x,y) = R(y,x)$ for all $x, y \in X$). A proximity relation R is not necessarily transitive ($T_L(R(x,y), R(y,z) \leq R(x,z))$ for all $x, y, z \in X$ where T_L is the Łukasiewicz t-norm). A strict simple FASP program can be used to find the transitive closure of R. This is the minimal mapping $\hat{R} : X \times X \to [0,1] \cap \mathbb{Q}$ that is reflexive, symmetric and transitive such that $R(x,y) \leq \hat{R}(x,y)$ for all $(x,y) \in X \times X$. Finding such "transitive approximations" is useful in many artificial intelligence areas, e.g. in fuzzy clustering [Castro et al. 1998] and analogous problems need to be solved in fuzzy spatial reasoning [Schockaert et al. 2009].

The corresponding FASP program simply consists of the rules

$$\hat{R}(x,z) \leftarrow \hat{R}(x,y) \otimes \hat{R}(y,z)$$

for all $x, y, z \in X$ and facts of the form

$$\hat{R}(x,y) \leftarrow \overline{R(x,y)},$$

where $\overline{R(x,y)}$ is the symbol representing the fixed value $R(x,y)$.

3.3.3 ATM location selection problem

The FASP program presented below is based on the ATM location selection problem for which a corresponding FASP program is given and discussed in [Janssen 2011]. Here, the problem is slightly modified and the resulting program is more concise. This problem is often referred to as the k-center problem and it is shown to be NP-hard [Megiddo and Tamir 1983].

Suppose we want to place k ATM machines $\{a_1, \ldots, a_k\}$ on roads connecting m towns such that the distance between each town and the closest ATM machine is less than a particular distance. Schematically, this can be seen as an undirected weighted graph $G = (\mathit{Towns}, \mathit{Edges})$ where $\mathit{Towns} = \{t_1, \ldots, t_m\}$ is the set of towns and $e_{t_i t_j}$ is an edge if there is a road connecting towns t_i and t_j. Note that $e_{t_i t_j} \in \mathit{Edges}$ iff $e_{t_j t_i} \in \mathit{Edges}$. A weight is given to an edge $e_{t_i t_j}$ in function of the distance between towns t_i and t_j. To obtain a weight that is an element in $[0, 1] \cap \mathbb{Q}$, one can use a normalised distance $d : \mathit{Towns} \times \mathit{Towns} \rightarrow [0, 1] \cap \mathbb{Q}$, e.g. the actual distance between two towns divided by the sum of all distances between all possible pairs of towns[2]. Suppose such a normalised distance function d is given, then we can define a normalised nearness function $n = 1 - d$. By using the Łukasiewicz conjunction we can perform summations of degrees of nearness. Indeed, suppose $n_1 = 1 - d_1 \in [0, 1] \cap \mathbb{Q}$ and $n_2 = 1 - d_2 \in [0, 1] \cap \mathbb{Q}$, then for a fuzzy interpretation I we have

$$
\begin{aligned}
[\overline{n_1} \otimes \overline{n_2}]_I &= \max(n_1 + n_2 - 1, 0) \\
&= \max(1 - d_1 + 1 - d_2 - 1, 0) \\
&= \max(1 - (d_1 + d_2), 0) \\
&= 1 - \min(d_1 + d_2, 1) \\
&= 1 - [\overline{d_1} \oplus \overline{d_2}]_I
\end{aligned}
$$

where $\overline{n_1}, \overline{n_2}, \overline{d_1}, \overline{d_2}$ are the symbols representing the values $n_1, n_2, d_1, d_2 \in [0, 1] \cap \mathbb{Q}$.

We can now specify a program whose answer sets correspond to those configurations of ATMs such that the distance from each town to the nearest ATM is at most a particular degree $d' \in [0, 1] \cap \mathbb{Q}$.

The first part of the program consists of the facts that define the graph. Specifically, we have a set of rules denoting which towns are connected by a single road and how near they are to each other:

$$\mathsf{edge}(e_{t_i t_j}) \leftarrow \overline{n(t_i, t_j)}$$

for each edge $e_{t_i t_j} \in \mathit{Edges}$. Secondly, for each edge one can (arbitrarily) designate one of the towns to be the starting point and the other one to be the ending point. This choice

[2] If two cities are not connected directly, the distance of the shortest path is taken.

has no influence on the outcome of the program:

$$\begin{array}{lcl} \mathsf{start}(t_i, e_{t_i t_j}) & \leftarrow & \overline{1} \\ \mathsf{end}(t_j, e_{t_i t_j}) & \leftarrow & \overline{1} \end{array}$$

for each edge $e_{t_i t_j} \in$ *Edges*.

The second part of the program consists of rules generating eligible solutions. For each $a \in \{a_1, \ldots, a_k\}$ and each edge $e \in$ *Edges* we add the following rules

$$\begin{array}{llcl} r_1: & \mathsf{loc}(a,e) & \leftarrow & \mathsf{loc}(a,e) \oplus \mathsf{loc}(a,e) \\ r_2: & \mathsf{loc}(a,e) & \leftarrow & \otimes\{\mathrm{not}\,\mathsf{loc}(a,e') \mid e' \in \mathit{Edges}, e' \neq e\} \\ r_3: & \mathsf{locnearend}(a,e) & \leftarrow & (\mathsf{edge}(e) \oplus \mathrm{not}\,\mathsf{locnearstart}(a,e)) \otimes \mathsf{loc}(a,e) \\ r_4: & \mathsf{locnearstart}(a,e) & \leftarrow & (\mathsf{edge}(e) \oplus \mathrm{not}\,\mathsf{locnearend}(a,e)) \otimes \mathsf{loc}(a,e) \end{array}$$

Rule r_1 is used to ensure that the truth degree of $\mathsf{loc}(a,e)$ is in $\{0,1\}$, i.e. an ATM a is located on an edge e or not. Indeed, a fuzzy interpretation I models this rule if and only if $\min(2I(\mathsf{loc}(a,e)),1) \leq I(\mathsf{loc}(a,e))$, i.e. $I(\mathsf{loc}(a,e)) \leq 0$ or $I(\mathsf{loc}(a,e)) \geq 1$. By using negation-as-failure, rule r_2 then generates all possible configurations (cfr. Example 3.17). If an ATM a is located on an edge e, then rules r_3 and r_4 determine how near to the start and the end of e it is located. Indeed, in terms of distances we want to model the following. Suppose e is the edge between t_i and t_j on which an ATM a is located, then it should hold that

$$d(t_i,a) + d(a,t_j) = d(t_i,t_j).$$

Hence in terms of nearness we want

$$1 - n(t_i,a) + 1 - n(a,t_j) = 1 - n(t_i,t_j)$$

or

$$n(t_i,a) + n(a,t_j) - 1 = n(t_i,t_j).$$

Thus for a fuzzy interpretation I we want

$$I(\mathsf{locnearend}(a,e)) + I(\mathsf{locnearstart}(a,e)) - 1 = I(\mathsf{edge}(e)).$$

This can be modelled by rules r_3 and r_4 since, assuming that $\mathsf{loc}(a,e)$ has truth value 1, a fuzzy interpretation I models r_3 if

$$I(\mathsf{locnearend}(a,e)) \geq \min(I(\mathsf{edge}(e)) + 1 - I(\mathsf{locnearstart}(a,e)),1).$$

Analogously for rule r_4 we obtain

$$I(\mathsf{locnearstart}(a,e)) \geq \min(I(\mathsf{edge}(e)) + 1 - I(\mathsf{locnearend}(a,e)),1).$$

Hence for an answer set I we obtain by the minimality condition that

$$I(\mathsf{locnearend}(a,e)) + I(\mathsf{locnearstart}(a,e)) - 1 = I(\mathsf{edge}(e)).$$

Note that if $\min(I(\mathsf{edge}(e)) + 1 - I(\mathsf{locnearstart}(a,e)), 1) = 1$, then by rule r_3 we would have $I(\mathsf{locnearend}(a,e)) = 1$ and hence by rule r_4 that $I(\mathsf{locnearstart}(a,e)) = I(\mathsf{edge}(e))$ and we obtain the same result.

The following rules define the maximal nearness and hence the shortest distance to an ATM for a town $t \in$ *Towns*. In particular, r_7 and r_8 define the shortest distance to an ATM if the town is not the start or end point of an edge that contains an ATM.

$$\begin{array}{llll}
r_5: & \mathsf{ATMnear}(t) & \leftarrow & \mathsf{start}(t,e) \otimes \mathsf{locnearstart}(a,e) \otimes \mathsf{loc}(a,e) \\
r_6: & \mathsf{ATMnear}(t) & \leftarrow & \mathsf{end}(t,e) \otimes \mathsf{locnearend}(a,e) \otimes \mathsf{loc}(a,e) \\
r_7: & \mathsf{ATMnear}(t) & \leftarrow & \mathsf{edge}(e) \otimes \mathsf{ATMnear}(t') \otimes \mathsf{start}(t,e) \otimes \mathsf{end}(t',e) \\
r_8: & \mathsf{ATMnear}(t) & \leftarrow & \mathsf{edge}(e) \otimes \mathsf{ATMnear}(t') \otimes \mathsf{end}(t,e) \otimes \mathsf{start}(t',e)
\end{array}$$

for each $t, t' \in$ *Towns*, $e \in$ *Edges* and $a \in \{a_1, \ldots, a_k\}$.

Finally constraints are needed to indicate the minimal nearness $n' = 1 - d'$ allowed in a valid configuration of ATMs.

$$r_9 : \overline{0} \leftarrow \mathrm{not}\, \mathsf{ATMnear}(t) \otimes \overline{n'}$$

for each $t \in$ *Towns*. Indeed, a fuzzy interpretation I models r_9 if

$$I(\mathsf{ATMnear}(t)) \geq n'.$$

The explanations above show that each answer set has the properties the solutions of the original search problem must have. On the other hand, we also have that each solution, seen as a fuzzy interpretation I, corresponds to an answer set. It has to be checked that I is a minimal fuzzy model of P^I. Rules r_1^I and r_2^I in P^I are modelled by I since these rules generate all possible placements of the ATMs on the roads and I corresponds to one particular configuration of ATMs. Rules r_3^I and r_4^I are also modelled since these rules compute the exact location on the edge of each ATM. An explanation similar as above ensures that I is minimal such that these rules are modelled. Rules $r_5^I - r_8^I$ compute the shortest distance to an ATM for each town, hence these rules must be modelled in a minimal way by I. Since I corresponds to a configuration such that for each town the distance to the closest ATM is less then $d' = 1 - n'$, rule r_9^I is also modelled.

Consider as a concrete example the following setting. Suppose there are two ATMs a_1 and a_2 and three towns t_1, t_2 and t_3 such that $n(t_1, t_2) = 0.8$, $n(t_1, t_3) = 0.3$ and $n(t_2, t_3) = 0.1$. Suppose we are interested in placing these ATMs such that the minimal nearness n' is equal to 0.3. One of the answer sets is given in Figure 3.2. ATM a_1 is placed in t_3 and hence a_1 is near t_1 with degree 0.3 and near t_3 with degree 1. ATM a_2 is placed on the road between t_2 and t_3 with nearness degree 0.8 to t_2 and nearness degree 0.3 to t_3. Indeed the answer set I corresponding to this setting is a minimal model of P^I.

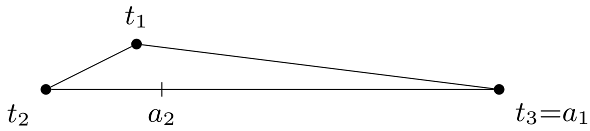


Figure 3.2: Configuration of ATMs

Another possible solution (Figure 3.3) would be to place a_1 in town t_3 and a_2 on the road connecting towns t_1 and t_2, for instance such that a_2 is near t_1 and near t_2 with degree 0.9.

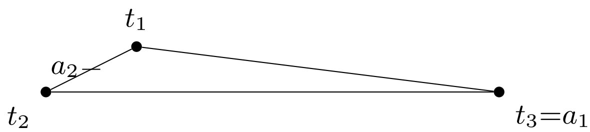


Figure 3.3: Configuration of ATMs

Note that we can also impose different degrees of nearness for different towns.

3.3.4 Fuzzy graph colouring problem

Using FASP, a continuous variant of the graph colouring problem can easily be defined. Recall that the classical graph colouring problem (Example 3.7) consists of colouring the nodes of a graph using a finite number of colors such that two nodes which are connected by an edge have a different color.

Now assume a weighted graph is given, specified by rules

$$\text{edge}(a, b) \leftarrow \overline{c}$$

with $c \in [0,1] \cap \mathbb{Q}$ and where edge(a, b) represents the weight of the edge between nodes a and b. The problem consists of assigning grey values to each node in the graph such that the difference between the grey values is at least as large as the corresponding edge weight. Besides the above facts, we also need a generating part: in each answer set we want to have that a node is black to the degree that is not white (cfr. Example 3.17). Hence for each node a we add the rules.

$$\begin{aligned} r_1 : \quad \mathsf{black}(a) &\leftarrow \text{not}\,\mathsf{white}(a) \\ r_2 : \quad \mathsf{white}(a) &\leftarrow \text{not}\,\mathsf{black}(a) \end{aligned}$$

We also need a connective that denotes how similar the color of two nodes is. An obvious choice would be $\leftrightarrow$. Moreover note that for a fuzzy interpretation I we have

$$[\alpha \leftrightarrow \beta]_I = 1 - |[\alpha]_I - [\beta]_I|.$$

By the definitions of the Łukasiewicz connectives, we have that

$$\alpha \leftrightarrow \beta = (\sim\alpha \oplus \beta) \otimes (\sim\beta \oplus \alpha).$$

By rules r_1 and r_2 we have that for each answer set I it must hold that $I(\mathsf{white}(a)) = 1 - I(\mathsf{black}(a))$ for each node a. Hence we can write the following rules for each pair of nodes a and b

$$r_3 : \quad \mathsf{sim}(a, b) \quad \leftarrow \quad (\mathsf{white}(a) \oplus \mathsf{black}(b)) \otimes (\mathsf{white}(b) \oplus \mathsf{black}(a))$$

Finally, we need constraints to filter out the unwanted assignments:

$$r_4 : \overline{0} \leftarrow \mathsf{edge}(a, b) \otimes \mathsf{sim}(a, b)$$

for each pair of nodes a and b. Indeed, for a fuzzy interpretation I we have that I models r_4 iff $I(\mathsf{edge}(a, b)) + I(\mathsf{sim}(a, b)) - 1 \leq 0$ iff $I(\mathsf{edge}(a, b)) \leq 1 - I(\mathsf{sim}(a, b))$.

Consider as an example the graph consisting of four nodes a, b, c and d with edge weights as depicted in Figure 3.4. Edges with weight 0 have been omitted. One possible colouring, as shown in the picture, is an answer set I such that

$$\begin{array}{llll} I(\mathsf{black}(a)) = 0 & I(\mathsf{black}(b)) = 0.1 & I(\mathsf{black}(c)) = 0.5 & I(\mathsf{black}(d)) = 0.8 \\ I(\mathsf{white}(a)) = 1 & I(\mathsf{white}(b)) = 0.9 & I(\mathsf{white}(c)) = 0.5 & I(\mathsf{white}(d)) = 0.2 \end{array}$$

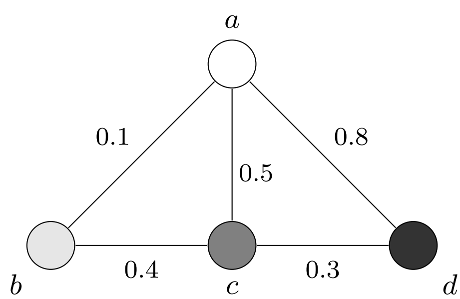


Figure 3.4: Fuzzy graph colouring

4 | Complexity of fuzzy answer set programming under Łukasiewicz semantics

4.1 Introduction

In Section 3.2 we have recalled a general framework for fuzzy answer set programming (FASP), a generalisation of ASP (Section 3.1) in which propositions are allowed to be graded. Little is known about the computational complexity of FASP and almost no techniques are available to compute the answer sets of a FASP program. In this chapter we will introduce results on the computational complexity of FASP under Łukasiewicz semantics (Section 2.2.3) and show a reduction from reasoning with such FASP programs to bilevel linear programming, thus opening the door to practical applications. Łukasiewicz logic is often used in applications because it preserves many desirable properties from classical logic. It is closely related to mixed integer programming, as was first shown by McNaughton [McNaughton 1951] in a non-constructive way. Later, Hähnle [Hähnle 1997] gave a concrete, semantics-preserving, translation from a set of formulas in Łukasiewicz logic into a mixed integer program. Checking the satisfiability of a Łukasiewicz logic formula thus essentially corresponds to checking the feasibility of a mixed integer program. In particular, given a regular FASP program P, a literal l and a value $\lambda_l \in [0,1] \cap \mathbb{Q}$, we are interested in the following decision problems.

1. **Existence**: Does there exist an answer set I of P?
2. **Set-membership**: Does there exist an answer set I of P such that $I(l) \geq \lambda_l$?
3. **Set-entailment**: Does $I(l) \geq \lambda_l$ hold for each answer set I of P?

Note that these are generalisations of the decision problems for ASP for which the complexity is given in Table 3.1.

Recall that, under Łukasiewicz semantics, a *strict disjunctive FASP program* is a set of rules of the form

$$r : a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m \otimes \text{not}\, c_1 \otimes \ldots \otimes \text{not}\, c_k$$

with a_i, b_j, c_l literals and/or truth constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$, "not" the negation-as-failure operator corresponding to the Łukasiewicz negator, $\oplus$ and $\otimes$ resp. the Łukasiewicz disjunction and conjunction and $\leftarrow$ the Łukasiewicz implicator (see Section 2.2.3). By our particular choice of semantics, strict FASP relates to Łukasiewicz logic as ASP does to classical logic. For Łukasiewicz logic, satisfiability is an NP-complete problem [Mundici 1987]. Since satisfiability checking has the same complexity for classical logic, one would perhaps expect ASP and FASP to have the same complexity as well. This expectation is reinforced by the fact that in the case of probabilistic ASP, the complexity of the existence problem has been shown to be Σ_2^{P}-complete [Łukasiewicz 1999], i.e. the same as the complexity of the existence problem in classical ASP. On the other hand,there are fuzzy description logics that, unlike the classical case, do not have the finite model property under Łukasiewicz logic or under product logic [Bobillo et al. 2011] and there are description logics whose classical counterparts are decidable but that are undecidable under Łukasiewicz logic [Cerami and Straccia 2013].

Although existence and set-membership are Σ_2^{P}-complete for disjunctive ASP, in this chapter we will show NP-completeness for strict disjunctive and strict normal FASP. We will also show that the existence of an answer set for a strict normal FASP program without constraints and without strong negation is always guaranteed and hence that the complexity of the existence problem for this class of FASP programs is "constant". However, for strict disjunctive FASP without constraints and without strong negation we are only able to show membership in NP for the existence problem.

Table 4.1: Complexity of inference in strict FASP

strict FASP		existence	set-membership	set-entailment
no restrictions	disjunctive	NP-complete	NP-complete	coNP-complete
	normal	NP-complete	NP-complete	coNP-complete
	definite	in P	in P	in P
no constraints, no strong negation	disjunctive	in NP	NP-complete	in coNP
	normal	in P	NP-complete	in coNP
	definite	in P	in P	in P

An overview of the complexity results that we can establish for strict FASP under Łukasiewicz semantics is provided in Table 4.1.

Besides strict FASP programs (under Łukasiewicz semantics), we will also discuss the computational complexity for regular FASP programs (under Łukasiewicz semantics), i.e. sets of rules of the form

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not}\, c_1, \ldots, \mathrm{not}\, c_k),$$

with a_i, b_j, c_l literals and/or truth constants corresponding to truth values in $[0, 1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$. The connectives f and g are compositions of the Łukasiewicz connectives $\otimes$, $\oplus$, $\vee$ and $\wedge$. The negation-as-failure operator not and the implication $\leftarrow$ correspond resp. to the Łukasiewicz negator and implicator. In this sense, fuzzy equilibrium logic is a proper generalisation of regular FASP [Schockaert et al. 2012]. By using its complexity results we can derive that existence and set-membership for regular FASP are in Σ_2^{P} and that set-entailment is in Π_2^{P}. By reducing the decision problems for disjunctive ASP to regular FASP we will also derive resp. Σ_2^{P}-hardness and Π_2^{P}-hardness. Hence for regular FASP without any restrictions, we obtain Σ_2^{P}-completeness for existence and set-membership and Π_2^{P}-completeness for set-entailment (see Proposition 4.10). However, if we restrict ourselves to programs with at most one literal in the head of each rule, then we can only show Σ_2^{P}-membership and NP-hardness for set-membership and existence and Π_2^{P}-membership and coNP-hardness for set-entailment. If, in addition, we do not allow "not" in the rules we can only find a pseudo-polynomial time algorithm to compute answer sets based on computing least fixpoints. Although in general we can only show membership in NP $\cap$ coNP, for several subclasses of regular definite FASP programs we can show P-membership. In particular, for regular definite FASP programs with only conjunction and maximum or only disjunction in the body of rules, we can provide a polynomial time algorithm to compute answer sets. This is also the case for regular definite FASP programs with a cycle free dependency graph or with only polynomially bounded constants.

An overview of the complexity results that we can establish for regular FASP under Łukasiewicz sematics is provided in Table 4.2.

Table 4.2: Complexity of inference in regular FASP

regular FASP		existence	set-membership	set-entailment
no restrictions	disjunctive	Σ_2^P-complete	Σ_2^P-complete	Π_2^P-complete
	normal	NP-hard, in Σ_2^P	NP-hard, in Σ_2^P	coNP-hard, in Π_2^P
	definite	in NP $\cap$ coNP	in NP $\cap$ coNP	in NP $\cap$ coNP
only $\otimes$ and $\vee$ in body	definite	in P	in P	in P
only $\oplus$ in body	definite	in P	in P	in P
cycle free	definite	in P	in P	in P
polynomially bounded constants	definite	in P	in P	in P

As mentioned before, we will also provide an implementation into bilevel linear programming for strict disjunctive FASP. Intuitively, in a bilevel linear programming problem there are two agents: the leader and the follower. The leader goes first and attempts to optimize his/her objective function. The follower observes this and subsequently makes his/her decision. Since it caught the attention in the 1970s, there have been many algorithms proposed for solving bilevel linear programming problems (e.g. [Bard and Falk 1982, Candler and Townsley 1982, Shi et al. 2006]). A popular way to solve such a problem, e.g. in [Bard and Falk 1982], is to translate the bilevel linear programming problem into a nonlinear programming problem using Kuhn-Tucker constraints. This new program is a standard mathematical program and relatively easy to solve because all but one constraint is linear. In a later study [Bard and Moore 1990], an implicit approach to satisfying the nonlinear complementary constraint was proposed, which proved to be more efficient than the other strategies that were known at the time. By showing a reduction of strict disjunctive FASP into bilevel programming we thus provide a basis to build solvers for FASP.

This chapter is structured as follows. In Section 4.3 resp. Section 4.4 we will present complexity results for strict FASP, resp. regular FASP using propositions and lemmas introduced in Section 4.2. In Section 4.5 we will show that there is a reduction from reasoning with strict disjunctive FASP to bilevel linear programming followed by some concluding remarks in 4.6.

4.2 Preliminaries

In the remainder of this chapter, we will study the complexity of the decision problems discussed in the introduction for regular and strict FASP under Łukasiewicz logic[1]. In this section we present some results that will be often applied to show other results throughout the chapter.

[1] In this chapter we will omit "under Łukasiewicz logic " since we are only dealing with these types of FASP programs.

First of all, using the fact that $I(a) + I(\neg a) \leq 1$ iff $[\overline{0} \leftarrow a \otimes \neg a]_I = 1$, we will show in Lemma 4.1 that a regular FASP program can be rewritten as a regular FASP program without strong negation, hence generalising Remark 3.1.

Lemma 4.1

Let P be a regular FASP program. There exists a regular FASP program P' without strong negation such that a fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ is an answer set of P iff there exists an answer set $I' \in \mathcal{F}(\mathcal{L}_{P'})$ of P' such that for each atom $a \in \mathcal{B}_P$ we have $I(a) = I'(a)$ and $I(\neg a) = I'(a')$ for a fresh atom $a' \in \mathcal{B}_{P'}$.

Proof. For each atom a in P, introduce a fresh atom a'. The program P' is then obtained by replacing all negated atoms $\neg a$ in P by their corresponding atom a' and for each couple of atoms (a, a') adding the constraint $\overline{0} \leftarrow a \otimes a'$. The program P' then has the required properties since $[\overline{0} \leftarrow a \otimes a']_I = 1$ iff $I(a) + I(a') \leq 1$ for each pair of atoms (a, a'). □

Secondly, we present a lemma that combined with Lemma 4.1 implies that a regular FASP program P can be rewritten as the union of a regular FASP program without constraints and without strong negation P' and a set of constraints C without strong negation such that the answer sets of P correspond to the answer sets of P'. Without loss of generality, in this lemma we will assume that all literals in C are also literals in P', and hence that $\mathcal{L}_P = \mathcal{L}_{P'}$. We may assume this since if there exists a literal l in C such that $l \notin \mathcal{L}_{P'}$, then for each answer set I of P we must have that $I(l) = 0$ or $I(l) = 1$, depending on the fact whether l is part of a negation-as-failure literal $\text{not}\, l$ or not. Further, in this lemma we will denote by $g_{|B}$ the restriction of a function $g : C \to D$ to the domain $B \subseteq C$, i.e. the function $g_{|B} : B \to D : x \mapsto g(x)$.

Lemma 4.2

Let P be a regular FASP program such that $P = P' \cup C$ where C is a set of constraints in P, $\mathcal{L}_P = \mathcal{L}_{P'}$ and $I \in \mathcal{F}(\mathcal{L}_P)$. It holds that I is an answer set of P iff I is an answer set of P' and a fuzzy model of C.

Proof. ($\Rightarrow$) Suppose I is an answer set of P. By assumption, I is a fuzzy model of C. It remains to be shown that I is an answer set of P'. I is a fuzzy model of $(P')^I$ since it is a fuzzy model of P^I. Now suppose there exists a fuzzy model $J \in \mathcal{F}(\mathcal{L}_{P'})$ of $(P')^I$ such that $J \leq I$. We show that $J = I$, from which it then follows that I is a minimal fuzzy model of $(P')^I$ and hence an answer set of P'. Note that J is a fuzzy model of P^I. Indeed, let r be an arbitrary rule in $P = P' \cup C$. If $r \in P'$, then J

models r^I by assumption. If $r : \overline{c} \leftarrow \alpha$ is a constraint in C, then $[\alpha^I]_J \leq [\alpha^I]_I$ since $J \leq I$ and $[\alpha^I]_I \leq c$ since I is a fuzzy model of $C^I \subseteq P^I$. Hence $[\alpha^I]_J \leq c$, i.e. J models r^I. We obtain that J is a fuzzy model of P^I such that $J \leq I$. Together with the fact that I is a minimal fuzzy model of P^I, this implies that $J = I$.

($\Leftarrow$) Suppose I is an answer set of P' and a fuzzy model of C. Then I is a fuzzy model of $P^I = (P' \cup C)^I$ as well. Now suppose there exists a fuzzy model J of P^I such that $J \leq I$, then it follows that J is a model of $(P')^I \subseteq P^I$. Hence, since I is an answer set of P' and thus by definition a minimal fuzzy model of $(P')^I$, it follows that $J = I$.

□

Remark 4.1

Recall that a regular simple FASP program, i.e. a regular definite FASP program with exactly one atom in the head of each rule and no strong negation, has a unique answer set (Proposition 3.4). Hence the complexity of the set-membership problem and the set-entailment problem are equal and the complexity of the existence problem is "constant" for regular simple FASP. Moreover, note that by Lemmas 4.1 and 4.2, it follows that if the answer set of a certain type of regular simple FASP programs can be determined in polynomial time, then the complexity of the decision problems for the corresponding types of regular definite FASP is polynomial as well. Indeed, each regular definite FASP program P can be rewritten as $P' \cup C$ where P' is a regular simple FASP program and C is a set of constraints such that I is a answer set of P iff I is an answer set of P' and a fuzzy model of C. To check if P has an answer set, we can compute the answer set of P' and check if it is a fuzzy model of C.

Without loss of generality, we may assume that in each rule of a general FASP program, the body has exactly two arguments. Indeed, from the lemmas below it follows that a program can be rewritten as a program with only rules of the form $\alpha \leftarrow f(a,b)$ with a and b (negation-as-failure) literals and/or truth constants, $f(a,b)$ equal to either $a \otimes b$, $a \oplus b$, $a \vee b$ or $a \wedge b$ and α an arbitrary head. The idea is to substitute expressions in the body by adding new rules with fresh atoms in the head of these rules. This can be done since an answer set is a minimal fuzzy model of the program (Proposition 3.5) and the functions representing the connectives allowed in the bodies of rules are increasing.

Lemma 4.3

Let $P = P_1 \cup \{r\}$ be a regular FASP program where

$$r : \beta \leftarrow f(l_1, \ldots, l_n)$$

with l_i (negation-as-failure) literals and/or constants, β an arbitrary head and $f(l_1, \ldots, l_n)$ denotes either $\otimes_{i=1}^n l_i$, $\oplus_{i=1}^n l_i$, $\bigvee_{i=1}^n l_i$ or $\bigwedge_{i=1}^n l_i$.

For a fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$, it holds that I is an answer set of P iff there exists a fuzzy interpretation $I' \in \mathcal{F}(\mathcal{L}_{P'})$ such that $I'_{|\mathcal{L}_P} = I$ and I' is an answer set of P' where $P' = P_1 \cup P_2$ and P_2 is the program consisting of the rules

$$\begin{aligned} b_1 &\leftarrow f(l_1, l_2) \\ b_2 &\leftarrow f(b_1, l_3) \\ &\vdots \\ b_{n-2} &\leftarrow f(b_{n-3}, l_{n-1}) \\ \beta &\leftarrow f(b_{n-2}, l_n) \end{aligned}$$

with $b_1, \ldots, b_{n-2}$ atoms which are not used in P.

Proof. ($\Rightarrow$) Suppose I is an answer set of P. We expand I to a fuzzy interpretation I' on $\mathcal{L}_{P'}$ as follows. Define $I'(b_1) = [f(l_1, l_2)]_{I'}$ and $I'(b_i) = [f(b_{i-1}, l_{i+1})]_{I'}$ for $i \neq 1$. It is easy to see that I' is a fuzzy model of $(P')^{I'}$. Next, we show that I' is a minimal fuzzy model of $(P')^{I'}$. Suppose that $J' \leq I'$ is a fuzzy model of $(P')^{I'}$. One can show that $J = J'_{|\mathcal{L}_P}$ is a fuzzy model of P^I, hence it follows that $J = I$. We prove by induction on $i = 1, \ldots, n-2$ that $J' = I'$:

$$\begin{aligned} I'(b_1) &= [f(l_1, l_2)]_{I'} && \text{(definition } I'\text{)} \\ &= [f(l_1^{I'}, l_2^{I'})]_{I'} && \text{(definition reduct)} \\ &= [f(l_1^{I'}, l_2^{I'})]_{J'} && (J'_{|\mathcal{L}_P} = J = I = I'_{|\mathcal{L}_P}) \\ &\leq J'(b_1) && (J' \text{ fuzzy model of } (P')^{I'}) \\ &\leq I'(b_1) && (J' \leq I') \end{aligned}$$

Suppose $I'(b_{i-1}) = J'(b_{i-1})$.

$$\begin{aligned} I'(b_i) &= [f(b_{i-1}, l_{i+1})]_{I'} && \text{(definition } I'\text{)} \\ &= [f(b_{i-1}, l_{i+1}^{I'})]_{I'} && \text{(definition reduct)} \\ &= [f(b_{i-1}, l_{i+1}^{I'})]_{J'} && \text{(induction hypothesis and } J'_{|\mathcal{L}_P} = I'_{|\mathcal{L}_P}) \end{aligned}$$

$$\begin{aligned}
&\leq J'(b_i) && (J' \text{ fuzzy model of } (P')^{I'}) \\
&\leq I'(b_i) && (J' \leq I')
\end{aligned}$$

($\Leftarrow$) Suppose there is a fuzzy interpretation $I' \in \mathcal{L}_{P'}$ such that I' is an answer set of P' and $I'_{|\mathcal{L}_P} = I$. We show that I is an answer set of P.

First note that since I' is a minimal fuzzy model of $(P')^{I'}$, it must hold that $I'(b_1) = [f(l_1, l_2)]_{I'}$ and $I'(b_i) = [f(b_{i-1}, l_{i+1})]_{I'}$ for $i \neq 1$. A straightforward proof then shows that I is a fuzzy model of P^I. Now suppose there exists a fuzzy model $J \leq I$ of P^I. We show that there exists $J' \in \mathcal{L}_{P'}$ which is a fuzzy model of $(P')^{I'}$ such that $J' \leq I'$ and $J'_{|\mathcal{L}_P} = J$. Since I' is a minimal fuzzy model of $(P')^{I'}$, it then follows that $J' = I'$ and hence $J = I$. Define J' as follows: for $l \in \mathcal{L}_P$ define $J'(l) = J(l)$ and $J'(b_1) = [f(l_1^{I'}, l_2^{I'})]_{J'}$ and $J'(b_i) = [f(b_{i-1}, l_{i+1}^{I'})]_{J'}$ for $i \neq 1$. We prove by induction on $i = 1, \ldots, n-2$ that $J' \leq I'$:

$$\begin{aligned}
J'(b_1) &= [f(l_1^{I'}, l_2^{I'})]_{J'} && (\text{definition } J') \\
&= [f(l_1^I, l_2^I)]_J && (I = I'_{|\mathcal{L}_P}, J = J'_{|\mathcal{L}_P}) \\
&\leq [f(l_1^I, l_2^I)]_I && (J \leq I) \\
&= [f(l_1^{I'}, l_2^{I'})]_{I'} && (I = I'_{|\mathcal{L}_P}) \\
&= I'(b_1)
\end{aligned}$$

If $J'(b_{i-1}) \leq I'(b_{i-1})$, then

$$\begin{aligned}
J'(b_i) &= [f(b_{i-1}, l_{i+1}^{I'})]_{J'} && (\text{definition } J') \\
&= \mathtt{f}(J'(b_{i-1}), J'(l_{i+1}^{I'})) \\
&= \mathtt{f}(J'(b_{i-1}), I'(l_{i+1}^{I'})) && (I'_{|\mathcal{L}_P} = J'_{|\mathcal{L}_P}) \\
&\leq \mathtt{f}(I'(b_{i-1}), I'(l_{i+1}^{I'})) && (\text{induction and } \mathtt{f} \text{ increasing}) \\
&= I'(b_i)
\end{aligned}$$

It is easy to show that J' is a fuzzy model of $(P')^{I'}$.

□

Lemma 4.4

Let $P = P_1 \cup \{r\}$ be a regular FASP program where

$$r : \beta \leftarrow f(\alpha_1, \ldots, \alpha_n)$$

with α_i formulas built from (negation-as-failure) literals and/or constants, $\otimes$, $\oplus$, $\vee$, $\wedge$ and β an arbitrary head and $f(\alpha_1, \ldots, \alpha_n)$ denotes either $\otimes_{i=1}^n \alpha_i$, $\oplus_{i=1}^n \alpha_i$, $\bigvee_{i=1}^n \alpha_i$ or $\bigwedge_{i=1}^n \alpha_i$.

For a fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$, it holds that I is an answer set of P iff there exists a fuzzy interpretation $I' \in \mathcal{F}(\mathcal{L}_{P'})$ such that $I'_{|\mathcal{L}_P} = I$ and I' is an answer set of P' where $P' = P_1 \cup P_2$ and P_2 is the program consisting of the rules

$$\begin{array}{rcl} a_1 & \leftarrow & \alpha_1 \\ a_2 & \leftarrow & \alpha_2 \\ & \vdots & \\ a_n & \leftarrow & \alpha_n \\ \beta & \leftarrow & f(a_1, \ldots, a_n) \end{array}$$

with $a_1, \ldots, a_n$ atoms which are not used in P.

Proof. ($\Rightarrow$) Suppose that I is an answer set of P. We expand I to a fuzzy interpretation $I' \in \mathcal{F}(\mathcal{L}_{P'})$ as follows: $I'(a_i) = [\alpha_i]_{I'} = [\alpha_i]_I$ for $i \in \{1, \ldots, n\}$. It is easy to show that I' is a fuzzy model of $(P')^{I'}$. Next, we show that I' is a minimal fuzzy model of $(P')^{I'}$. Suppose there exists a fuzzy model $J' \leq I'$ of $(P')^{I'}$. We show that $J' = I'$. First remark that $J = J'_{|\mathcal{L}_P}$ is a fuzzy model of P^I. Since I is a minimal fuzzy model of P^I, it follows that $J'_{|\mathcal{L}_P} = J = I = I'_{|\mathcal{L}_P}$. It remains to be shown that $J'(a_i) = I'(a_i)$ for $i \in \{1, \ldots, n\}$. But this follows easily:

$$\begin{aligned} J'(a_i) &\leq I'(a_i) && (J' \leq I') \\ &= [\alpha_i]_{I'} \\ &= [\alpha_i^{I'}]_{I'} \\ &= [\alpha_i^{I'}]_{J'} && (J'_{|\mathcal{L}_P} = I'_{|\mathcal{L}_P}) \\ &\leq J'(a_i) && (J' \text{ fuzzy model of } (P')^{I'}) \end{aligned}$$

($\Leftarrow$) Suppose that I' is a minimal fuzzy model of $(P')^{I'}$. We show that $I = I'_{|\mathcal{L}_P}$ is a minimal fuzzy model of P^I. Remark that, since I' is a minimal fuzzy model of $(P')^{I'}$, it must hold that $I'(a_i) = [alpha_i]_{I(} = [\alpha_i]_I$. It follows easily that I is a fuzzy model of P^I. Now suppose there exists a fuzzy model $J \leq I$ of P^I. We expand J to a fuzzy interpretation $J' \in \mathcal{L}_{P'}$ as follows: $J'(a_i) = [\alpha_i^I]_J$. It can be shown that J' is a fuzzy model of $(P')^{I'}$. Moreover, for each $i \in \{1, \ldots, n\}$, we have that

$$
\begin{aligned}
J'(a_i) &= [\alpha_i^I]_J \\
&\leq [\alpha_i^I]_I && (J \leq I) \\
&= [\alpha_i]_I \\
&= I'(a_i)
\end{aligned}
$$

Hence $J' \leq I'$ and since I' is a minimal fuzzy model of $(P')^{I'}$, it follows that $J' = I'$ and thus $J = I$.

□

Combining Lemmas 4.3 and 4.4, one can prove the following proposition; a FASP program can be rewritten as a set of rules with only two (negation-as- failure) literals or constants in the body such that the answer sets remain the same.

Proposition 4.1

Let P be a regular FASP program. P can be reduced (in polynomial time) to a regular FASP program P' such that $\mathcal{L}_P \subseteq \mathcal{L}_{P'}$ and each rule in P' has at most two arguments in the body and I is an answer set of P iff there exists a fuzzy interpretation $I' \in \mathcal{F}(\mathcal{L}_{P'})$ such that $I'_{|\mathcal{L}_P} = I$ and I' is an answer set of P'.

Proof. Suppose there exists a rule $r \in P$ with more than 2 arguments. We show by induction on the number of connectives n, written in prefix notation, that this rule can be rewritten as a set of rules with at most two arguments and one connective in the body such that the answer sets remain the same.

If $n = 1$, then r is of the form

$$r : \beta \leftarrow f(l_1, \ldots, l_m).$$

By Lemma 4.3 the assertion holds. Now suppose the assertion holds for $n < k$. We prove that it also holds for $n = k$. Rule r is now of the form

$$r : \beta \leftarrow f(\alpha_1, \ldots, \alpha_n),$$

where the number of connectives in α_i is strictly smaller than k. By Lemma 4.4, the assertion follows. □

4.3 Complexity of strict FASP

In this section we will study the complexity for strict disjunctive FASP, i.e. regular FASP programs with rules of the form

$$a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m \otimes \text{not}\, c_1 \otimes \ldots \otimes \text{not}\, c_k$$

with a_i, b_j, c_k literals and/or truth constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$. A summary of the results in this section can be found in Table 4.1.

We will first show that set-membership for strict disjunctive FASP is NP-complete. We will do this by showing NP-membership in Proposition 4.2 and by showing in Proposition 4.3 that it is already NP-hard for strict normal FASP. We will use these results to derive complexity results for the remaining decision problems for strict disjunctive FASP and strict normal FASP. We will then use these results to show that set-membership remains NP-complete in strict normal (and disjunctive) FASP even if constraints and strong negation are not allowed.

Proposition 4.2

Set-membership for strict disjunctive FASP is in NP.

Proof. From the analysis of the geometrical structure underlying fuzzy equilibrium models, which is a proper generalisation of regular FASP [Schockaert et al. 2012], it follows that a FASP program P has an answer set I such that $I(l) \geq \lambda_l$ for some $l \in \mathcal{L}_\mathcal{P}$ and $\lambda_I \in [0,1] \cap \mathbb{Q}$ iff there is such an answer set that can be encoded using a polynomial number of bits. Given a strict disjunctive program P and an answer set I, we show that we can check in polynomial time that I is an answer set of P. Note that checking if $I(l) \geq \lambda_l$ for a literal l can be done in constant time. By definition, we need to check that I is a minimal fuzzy model of P^I and that for each $l \in \mathcal{L}_P$ we have $I(l) + I(\neg l) \leq 1$. The latter is straightforward. To check whether I is a minimal fuzzy model of P^I, we can use linear programming. Indeed a rule $r : a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m$ from P^I is satisfied iff

$$\begin{aligned}
&[a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m]_I = 1\\
&\Leftrightarrow v_I((\sim b_1) \oplus \ldots \oplus (\sim b_m) \oplus a_1 \oplus \ldots \oplus a_n) = 1\\
&\Leftrightarrow v_I(\sim b_1) + \ldots + v_I(\sim b_m) + v_I(a_1) + \ldots + v_I(a_n) \geq 1\\
&\Leftrightarrow 1 - v_I(b_1) + \ldots + 1 - v_I(b_m) + v_I(a_1) + \ldots + v_I(a_n) \geq 1\\
&\Leftrightarrow 1 - I(b_1) + \ldots + 1 - I(b_m) + I(a_1) + \ldots + I(a_n) \geq 1
\end{aligned}$$

with $v_I : A \rightarrow [0, 1]$ the evaluation defined as $v_I(a) = I(a)$ if $a \in \mathcal{L}_P$ and $v_I(a) = 0$ otherwise. Hence, to check whether I is a minimal fuzzy model of P^I we use the following linear program M. The function to be minimized is the sum $\sum_{a \in \mathcal{L}_{P^I}} a'$ where for each literal $a \in \mathcal{L}_{\mathcal{P}^\mathcal{I}}$ we introduce a variable a' and the constraints in M are the following. For each literal $a \in \mathcal{L}_{P^I}$ we have $0 \leq a' \leq 1$ and for each rule

$$r : a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m$$

in P^I we have

$$1 \leq 1 - b'_1 + \ldots + 1 - b'_m + a'_1 + \ldots + a'_n$$

or equivalently

$$1 - m \leq -b'_1 - \ldots - b'_m + a'_1 + \ldots + a'_n.$$

If $a' = I(a)$ for each literal a is a solution of M, then I is a minimal fuzzy model of P^I. Indeed, since $I(a) = a'$ fulfills the constraints of M, it is a fuzzy model of P^I. Now suppose there exists a fuzzy model J such that $J < I$. Since it is a fuzzy model of P^I, the assignments $a'' = J(a)$ for each literal a satisfy the constraints of M but $\sum_{a \in \mathcal{L}_{P^I}} a'' < \sum_{a \in \mathcal{L}_{P^I}} a'$, a contradiction. Hence I is a minimal fuzzy model of P^I. □

Next, we show that the set-membership problem is also NP-hard by showing a reduction from 3SAT, which is NP-complete [Cook 1971], to (a subclass of) strict disjunctive FASP. Recall from Example 3.8 that instances of the 3SAT problem are Boolean expressions written in conjunctive normal form with 3 variables in each clause:

$$(a_{11} \vee a_{12} \vee a_{13}) \wedge (a_{21} \vee a_{22} \vee a_{23}) \wedge \ldots \wedge (a_{n1} \vee a_{n2} \vee a_{n3}),$$

where each a_{ij} is an atom or a strongly negated atom. The problem consists of deciding whether there exists a propositional interpretation that makes the Boolean expression true.

Proposition 4.3

Set-membership for strict normal FASP is NP-hard.

Proof. Consider an arbitrary instance of the 3SAT problem

$$\alpha = (a_{11} \vee a_{12} \vee a_{13}) \wedge (a_{21} \vee a_{22} \vee a_{23}) \wedge \ldots \wedge (a_{n1} \vee a_{n2} \vee a_{n3})$$

We translate each clause $a_{i1} \vee a_{i2} \vee a_{i3}$ to the rule

$$\overline{0} \leftarrow \neg a_{i1} \otimes \neg a_{i2} \otimes \neg a_{i3} \tag{4.1}$$

and for each literal x in α we add the rules

$$\neg x \leftarrow \mathrm{not}\, x \tag{4.2}$$
$$x \leftarrow \mathrm{not}(\neg x) \tag{4.3}$$
$$x' \leftarrow x \tag{4.4}$$
$$x' \leftarrow \neg x \tag{4.5}$$
$$\overline{0} \leftarrow \mathrm{not}(x') \tag{4.6}$$

where x' is a fresh atom not used in α. We denote the resulting strict normal FASP program by P.

1. First suppose that I is an answer set of P. By Lemma 4.2 we know that I is an answer set of P_1 and a fuzzy model of C where P_1 is the set of all rules in P of the form (4.2)-(4.5) and C is the set of all constraints of the form (4.1) and (4.6).

   Since I is a minimal fuzzy model of $(P_1)^I$ we know that for each literal x it holds that $I(x) = 1 - I(\neg x)$ by rules (4.2) and (4.3) and $I(x') = \max(I(x), I(\neg x))$ by rules (4.4) and (4.5). Since I must be a fuzzy model of the constraints in C, it follows that $1 - I(x') = 0$ by rule (4.6). If $I(x') = I(x)$, then $I(x) = 1$ and $I(\neg x) = 0$. Otherwise, if $I(x') = I(\neg x)$, then $I(\neg x) = 1$ and $I(x) = 0$. Hence, I is a consistent Boolean interpretation.

   We can now define the (consistent) propositional interpretation G as follows. For each literal x in α we have $G(x) =$ "true" if $I(x) = 1$ and $G(x) =$ "false" if $I(x) = 0$. We check that this assignment evaluates α to "true". This follows easily by (4.1) and the following equations:

$$
\begin{aligned}
&[\overline{0} \leftarrow \neg a_{i1} \otimes \neg a_{i2} \otimes \neg a_{i3}]_I = 1 \\
&\Leftrightarrow v_I(\overline{0} \oplus \sim(\neg a_{i1} \otimes \neg a_{i2} \otimes \neg a_{i3})) = 1 \\
&\Leftrightarrow v_I(\overline{0} \oplus \sim(\neg a_{i1}) \oplus \sim(\neg a_{i2}) \oplus \sim(\neg a_{i3})) = 1 \\
&\Leftrightarrow 0 + 1 - v_I(\neg a_{i1}) + 1 - v_I(\neg a_{i2}) + 1 - v_I(\neg a_{i3}) \geq 1 \\
&\Leftrightarrow 1 - I(\neg a_{i1}) + 1 - I(\neg a_{i2}) + 1 - I(\neg a_{i3}) \geq 1
\end{aligned}
$$

with $v_I : A \to [0,1]$ the evaluation defined as $v_I(a) = I(a)$ if $a \in \mathcal{L}_P$ and $v_I(a) = 0$ otherwise. Since for I it holds that $I(x) = 1 - I(\neg x)$ for each literal x, we obtain that

$$[\overline{0} \leftarrow \neg a_{i1} \otimes \neg a_{i2} \otimes \neg a_{i3}]_I = 1 \Leftrightarrow I(a_{i1}) + I(a_{i2}) + I(a_{i3}) \geq 1$$

It must hold that $I(a_{ij}) = 1$ for at least one literal a_{ij} in each clause. Hence, G is an assignment that evaluates each clause $a_{i1} \vee a_{i2} \vee a_{i3}$, and thus the whole expression α, to "true".

2. Consider a propositional interpretation G such that each clause $a_{i1} \vee a_{i2} \vee a_{i3}$ evaluates to "true". We define a fuzzy interpretation in $\mathcal{F}(\mathcal{L}_P)$ by $I(x) = 1$ if $G(x) =$ "true", $I(x) = 0$ if $G(x) =$ "false", $I(x') = \max(I(x), I(\neg x))$. Note that $I(\neg x) = 1 - I(x)$ since G is a propositional interpretation . We show that I is an answer set of P, or by Lemma 4.2 that it is a minimal fuzzy model of $(P_1)^I$ and a fuzzy model of C. It is clear that I is a fuzzy model of $(P_1)^I$. Now suppose there exists a fuzzy model J of $(P_1)^I$ such that $J < I$. Since I is such that $I(\neg x) + I(x) = 1$, by rules (4.2) and (4.3) in P_1 it follows that

$$J(\neg x) \geq [\text{not}\, x]_I = 1 - I(x) = I(\neg x) \geq J(\neg x)$$

and

$$J(x) \geq [\text{not}(\neg x)]_I = 1 - I(\neg x) = I(x) \geq J(x).$$

Hence we have for each literal x that $J(x) = I(x)$ and $J(\neg x) = I(\neg x)$. Since $J < I$, there must exist a literal x such that $J(x') < I(x')$ which implies by rules (4.4) and (4.5) in P_1 that

$$I(x') > J(x') \geq J(x) = I(x) \text{ and } I(x') > J(x') \geq J(\neg x) = I(\neg x).$$

This is impossible since either $I(x) = 1$ or $I(\neg x) = 1$ and then $I(x') > 1$.

It remains to be shown that I is a fuzzy model of C. Since

$$I(x') = \max(I(x), I(\neg x)) = 1$$

we have that I models the rule $\overline{0} \leftarrow \text{not}(x')$ for each literal x. As before, we obtain

$$[\overline{0} \leftarrow \neg a_{i1} \otimes \neg a_{i2} \otimes \neg a_{i3}]_I = 1 \Leftrightarrow I(a_{i1}) + I(a_{i2}) + I(a_{i3}) \geq 1$$

Since each clause $a_{i1} \vee a_{i2} \vee a_{i3}$ is satisfied by G, we know that for least one a_{ij} it must hold that $I(a_{ij}) = 1$. Hence $I(a_{i1}) + I(a_{i2}) + I(a_{i3}) \geq 1$.

□

The following corollary follows directly from Propositions 4.2 and 4.3.

Corollary 4.1

1. Set-membership for strict normal FASP is NP-complete.
2. Set-membership for strict disjunctive FASP is NP-complete.

The proofs of Propositions 4.2 and 4.3 do not exploit the fact that we want to find an answer set I such that $I(l) \geq \lambda_l$ for a particular λ_l. Hence these proofs can also be used to show NP-completeness for the existence problem.

Proposition 4.4

1. Existence for strict normal FASP is NP-complete.
2. Existence for strict disjunctive FASP is NP-complete.

Finally, by the proofs of Propositions 4.2 and 4.3 and the results in Proposition 4.4 we can show the following proposition.

Proposition 4.5

1. Set-entailment for strict normal FASP is coNP-complete.
2. Set-entailment for strict disjunctive FASP is coNP-complete.

Proof. To show coNP-membership for set-entailment in strict normal (disjunctive) FASP, we show that the complementary decision problem, i.e. "Given a strict normal (disjunctive) FASP program P, a literal $l \in \mathcal{L}_P$ and a value $\lambda_l \in [0,1] \cap \mathbb{Q}$; is there an answer set I of P such that $I(l) < \lambda_l$?" is in NP by a straightforward adaption of the proof of Proposition 4.2.

To show coNP-hardness, we reduce the NP-hard problem "existence" to the complement of the set-entailment problem. Consider a strict normal (disjunctive) FASP program P. Define $P' = P \cup \{a \leftarrow a\}$ with a a fresh atom. We show that P has an answer set iff it is not the case that all answer sets I' of P' are such that $I'(a) \geq 0.5$. First suppose that P has an answer set I. Then there exists an answer set I' of P' with $I'(a) < 0.5$. Indeed, define $I'(a) = 0$ and $I'(x) = I(x)$ otherwise. Next, suppose that there exists an answer set I' of P' such that $I'(a) < 0.5$. Then $I = I'_{|\mathcal{L}_P}$ is an answer set of P.

□

We will now show that set-membership remains NP-complete for strict normal and disjunctive FASP even if strong negation and constraints are not allowed. This result is based on Proposition 4.6 which uses the following lemma that enables us to simulate constraints in a regular FASP program. Note that this lemma is valid for more general FASP programs (under Łukasiewicz semantics) and not only for strict disjunctive FASP programs.

Lemma 4.5

Consider a regular FASP program $P = P_1 \cup C$ where P_1 is a regular FASP program and C is a set of constraints of the form $\overline{0} \leftarrow \alpha$. Let $P' = P_1 \cup C' \cup \{z \leftarrow \text{not}\, y\}$ where z and y are fresh atoms and $C' = \{y \leftarrow \alpha \mid (\overline{0} \leftarrow \alpha) \in C\}$. A fuzzy interpretation $I \in \mathcal{F}(\mathcal{L}_P)$ is an answer set of P iff there exists an answer set $I' \in \mathcal{F}(\mathcal{L}_{P'})$ of P' such that $I'_{|\mathcal{L}_P} = I$ and $I'(z) \geq 1$.

Proof. (⇒) Suppose that $I \in \mathcal{F}(\mathcal{L}_P)$ is an answer set of P. Define $I' \in \mathcal{F}(\mathcal{L}_{P'})$ as $I'(a) = I(a)$ if $a \in \mathcal{L}_P$, $I'(z) = 1$ and $I'(y) = 0$. We show that I' is an answer set of P'.

First, we prove that I' is a fuzzy model of P' and thus of $(P')^{I'}$. Clearly, I' is a fuzzy model of P_1 and it models the rule $z \leftarrow \text{not}\, y$. If $y \leftarrow \alpha$ is a rule in C', then by assumption we have that $I = I'_{|\mathcal{L}_P}$ models the rule $\overline{0} \leftarrow \alpha$. Thus $[\overline{0} \leftarrow \alpha]_{I'} = 1$ and $[\alpha]_{I'} = 0 = I'(y)$. Hence I' models $y \leftarrow \alpha$.

Next, we show that I' is a minimal fuzzy model of $(P')^{I'}$. Suppose there exists a fuzzy model $J' \in \mathcal{F}(\mathcal{L}_{P'})$ of $(P')^{I'}$ such that $J' \leq I'$. We show that $J = J'_{|\mathcal{L}_P}$ is a fuzzy model of P^I. Clearly, J is a fuzzy model of $(P_1)^I$. Since $J' \leq I'$ we have that $J(y) = J'(y) \leq I'(y) = 0$, thus given a rule $r : \overline{0} \leftarrow \alpha$ in C we have that for the corresponding rule $y \leftarrow \alpha$ in C' it holds that $0 = J(y) \geq [\alpha^{I'}]_J = [\alpha^I]_J$, with α^I the reduct of the expression α w.r.t. I. Hence $[r^I]_J = 1$ and J is a fuzzy model of P^I. Because I is a minimal fuzzy model of P^I and $J \leq I$, it follows that $I = J$. As mentioned before, we have $J'(y) = I'(y)$ and since $[z \leftarrow \overline{[\text{not}\, y]_{I'}}]_{J'} = 1$, we also have $J'(z) \geq 1 - I'(y) = I'(z) \geq J'(z)$. Hence $I' = J'$, which shows that I' is a minimal fuzzy model of $(P')^{I'}$.

(⇐) Suppose that $I' \in \mathcal{F}(\mathcal{L}_{P'})$ is an answer set of P' such that $I'(z) = 1$. We show that $I = I'_{|\mathcal{L}_P}$ is an answer set of P. By Lemma 4.2 it is sufficient to show that I is an answer set of P_1 and a fuzzy model of C.

First, we show that I is a fuzzy model of C. Since I' is a minimal fuzzy model of $(P')^{I'}$, it must hold that $I'(z) = 1 - I'(y)$ and thus that $I'(y) = 0$. Given a rule $r : \overline{0} \leftarrow \alpha$ in C we have that for the corresponding rule $y \leftarrow \alpha$ in C' it holds that $0 = I'(y) \geq [\alpha]_{I'}$, and thus $[r]_I = [r]_{I'} = 1$.

Next, note that I is a fuzzy model of $(P_1)^I$ since I' is a fuzzy model of $(P_1)^{I'}$. Now suppose there exists a fuzzy model $J \in \mathcal{F}(\mathcal{L}_{P_1})$ of $(P_1)^I$ such that $J \leq I$. Define $J' \in \mathcal{F}(\mathcal{L}_{P'})$ as follows: $J'(a) = J(a)$ if $a \in \mathcal{L}_P$, $J'(y) = 0$ and $J'(z) = 1$. We show that J' is a fuzzy model of $(P')^{I'}$. By assumption, J' is a fuzzy model of $(P_1)^{I'}$. For the rule $r : z \leftarrow \text{not}\, y$ in P' we have $J'(z) = 1 = I'(z) \geq [\text{not}\, y]_{I'}$, hence J' models $r^{I'}$. Finally, given a rule $r : y \leftarrow \alpha$ in C' we have for the corresponding rule $\overline{0} \leftarrow \alpha$ in C that $J'(y) = 0 \geq [\alpha^{I'}]_{I'} \geq [\alpha^{I'}]_{J'}$. Hence J' models $r^{I'}$. Since $J' \leq I'$ and I' is a minimal fuzzy model of $(P')^{I'}$ it follows that $J' = I'$ and thus $J = I$.

□

Proposition 4.6

Set-membership for strict normal FASP is NP-hard even if constraints and strong negation are not allowed.

Proof. Consider an instance of the 3SAT problem

$$\alpha = (a_{11} \vee a_{12} \vee a_{13}) \wedge (a_{21} \vee a_{22} \vee a_{23}) \wedge \ldots \wedge (a_{n1} \vee a_{n2} \vee a_{n3})$$

As shown in the proof of Proposition 4.3, α is satisfied by an assigment G iff the propositional interpretation I, with $I(x) = 1$ if $G(x) =$ "true" and $I(x) = 0$ if $G(x) =$ "false" is an answer set of P with P the program obtained in the proof of Proposition 4.3.

By Lemma 4.1 it follows that P can be rewritten as a strict normal FASP program P' without strong negation and in which the head of each rule contains exactly one atom or the constant $\overline{0}$ such that there is a one-on-one correspondence between the answer sets. By Lemma 4.5, it follows that we can define a strict normal FASP program P'' without constraints and without strong negation such that the answer sets of P' correspond to the answer sets of P'' for which a certain atom has at least truth value 1. □

Finally, we can derive the following corollaries:

Corollary 4.2

1. Set-membership for strict normal FASP is NP-complete, even if constraints and strong negation are not allowed.

2. Set-membership for strict disjunctive FASP is NP-complete, even if constraints and strong negation are not allowed.

Proof. Follows by the reduction in the proof of Proposition 4.6 and by Proposition 4.2. □

By Theorem 3.1 from [Madrid and Ojeda-Aciego 2012] we can derive that a strict normal FASP program without constraints and without strong negation always has an answer set.

Proposition 4.7

A strict normal FASP program without constraints and without strong negation always has an answer set. Hence existence for strict normal FASP program without constraints and without strong negation is in P.

By Theorem 4.10 in [Łukasiewicz 2008] it follows that stratified fuzzy description logic programs have at most one answer set. Since strict normal FASP program for which the dependency graph does not contain cycles (see Section 4.4.1 for the exact definitions) are programs of this form, it follows from Proposition 4.7 that such programs have exactly one answer set.

As the following example shows, the result from Proposition 4.7 is not valid for regular FASP programs in which disjunction is allowed in the body of rules. For these types of FASP programs, the existence of an answer set is not guaranteed, even if constraints and strong negation are not allowed.

Example 4.1

Consider the following regular normal FASP program P.

$$
\begin{array}{lcll}
p & \leftarrow & p \oplus p & (1) \\
q & \leftarrow & q \oplus q & (2) \\
p & \leftarrow & \text{not}\, p \otimes q & (3) \\
q & \leftarrow & \overline{c} & (4)
\end{array}
$$

with $c > 0$.

Suppose P has an answer set I. Since I must be a fuzzy model of rule (1) we have that $I(p) \geq \min(2I(p), 1)$ and hence either $I(p) \geq 2I(p)$, i.e. $I(p) = 0$, or $I(p) = 1$. The same reasoning holds for rule (2) and it follows that $I(p), I(q) \in \{0, 1\}$. By rule (4) it follows that $I(q) > 0$ and thus that $I(q) = 1$. If $I(p) = 0$, then P^I is the regular simple FASP program

$$\begin{aligned} p &\leftarrow p \oplus p \\ q &\leftarrow q \oplus q \\ p &\leftarrow q \\ q &\leftarrow \overline{c} \end{aligned}$$

which has as minimal fuzzy model $J(p) = J(q) = 1$. Thus $I \neq J$ is not an answer set of P. If $I(p) = 1$, then P^I is the regular simple FASP program

$$\begin{aligned} p &\leftarrow p \oplus p \\ q &\leftarrow q \oplus q \\ p &\leftarrow \overline{0} \\ q &\leftarrow \overline{c} \end{aligned}$$

which has as minimal fuzzy model $J(p) = 0$, $J(q) = 1$. Thus $I \neq J$ is not an answer set of P.

For the class of strict disjunctive FASP, in which constraints and strong negation are not allowed we can only show NP-membership for the existence problem (follows from the proof of Proposition 4.2). For set-entailment we can only show coNP-membership for both strict normal and disjunctive FASP where constraints and strong negation are not allowed (see the proof of Proposition 4.5).

Finally, we will discuss the complexity for strict definite FASP. We will show that the decision problems are in P. To do this, by Remark 4.1 it is sufficient to prove that the unique answer set of a strict simple FASP program can be determined in polynomial time. In particular, we will show that for such programs, the unique answer set can be found in polynomial time using linear programming, which is known to be in P. Moreover, the complexity remains the same if the connective maximum is allowed in the body of rules. Indeed, rules of the form $a \leftarrow b \otimes c$ are modelled by a fuzzy interpretation I iff $I(b)+I(c)-1 \leq I(a)$. Rules of the form $d \leftarrow e \vee f$ are modelled by a fuzzy interpretation I iff $I(e) \leq I(d)$ and $I(e) \leq I(f)$. Hence such a program can be efficiently translated to a linear program. Remark that from results in [Schockaert et al. 2012] it follows that a linear program always has a solution consisting of rational numbers.

Example 4.2

Consider the following program P.

$$\begin{array}{rcl} a & \leftarrow & b \otimes \overline{\frac{1}{2}} \\ b & \leftarrow & c \vee \overline{\frac{1}{3}} \end{array}$$

The corresponding linear program contains the following constraints.

$$\begin{array}{rcl} a' & \geq & b' - \frac{1}{2} \\ b' & \geq & c' \\ b' & \geq & \frac{1}{3} \\ 1 & \geq & a', b', c' \\ a', b', c' & \geq & 0 \end{array}$$

and the function to be minimised is $f(a', b', c') = a' + b' + c'$. The solution $a' = 0$, $b' = \frac{1}{3}$, $c' = 0$ given by the linear program then corresponds to the answer set I of P: $I(a) = a'$, $I(b) = b'$ and $I(c) = c'$.

Proposition 4.8

The unique answer set of a regular simple FASP program with only the connectives conjunction and maximum in the body of the rules can be found in polynomial time.

Proof. Consider a regular simple FASP program P with only rules of the form $a \leftarrow b \otimes c$ and $d \leftarrow e \vee f$. The answer set P can be found by solving the following linear program L_P. The function to be minimized is $f(a'_1, \ldots a'_n) = \sum_{i=1}^{n} a'_i$ with $\mathcal{B}_P = \{a_1, \ldots, a_n\}$ and a'_i is the corresponding variable for a_i and for each rule $a \leftarrow b \otimes c$ we add the constraints

$$\begin{aligned} a' &\geq b' + c' - 1 \\ 1 &\geq a', b', c' \\ a', b', c' &\geq 0. \end{aligned}$$

and for each rule $d \leftarrow e \vee f$, we add the constraints

$$\begin{aligned} d' &\geq e' \\ d' &\geq f' \\ 1 &\geq d', e', f' \\ d', e', f' &\geq 0. \end{aligned}$$

Suppose that $I : \mathcal{B}_P \rightarrow [0,1] \cap \mathbb{Q}$ is the answer set of P, i.e. I is the unique minimal fuzzy model of P. We show that if L_P has a solution $J' : \{a'_1, \ldots, a'_n\} \rightarrow \mathbb{Q}$, that $J : \mathcal{B}_P \rightarrow$

$[0,1] \cap \mathbb{Q} : a_i \mapsto J'(a'_i)$ is a minimal fuzzy model of P. Since P has a unique minimal fuzzy model, we then obtain $J = I$ and J' is the unique rational solution of L_P. Clearly, since J' satisfies the constraints in L_P we obtain that J is a fuzzy model of P. Suppose J is not a minimal fuzzy model of P, i.e. there exists a fuzzy model $M : \mathcal{B}_P \to [0,1] \cap \mathbb{Q}$ of P such that $M < J$, then $M' : \{a'_1, \ldots, a'_n\} \to \mathbb{Q} : a'_i \mapsto M(a_i)$ satisfies the constraints of L_P and it holds that $\sum_{i=1}^{n} M'(a'_i) < \sum_{i=1}^{n} J'(a'_i)$, a contradiction.

□

Corollary 4.3

The complexity of existence, set-membership and set-entailment for strict simple and strict definite FASP is polynomial.

Proof. Follows from Remark 4.1 and Proposition 4.8. □

4.4 Complexity of regular FASP

In this section, we will investigate the complexity of the decision problems for regular FASP (under Łukasiewicz semantics). Recall that these FASP programs are sets of rules of the form

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not}\, c_1, \ldots, \mathrm{not}\, c_k),$$

with a_i, b_j, c_l literals and/or truth constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$. The connectives f and g are compositions of the Łukasiewicz connectives $\otimes$, $\oplus$, $\vee$ and $\wedge$ and not and $\leftarrow$ correspond resp. to the Łukasiewicz negator and implicator. A summary of the results in this section can be found in Table 4.2.

From the complexity results for fuzzy equilibrium logic, which is a proper generalisation of regular FASP [Schockaert et al. 2012], we can derive that existence and set-membership for regular FASP are in Σ_2^{P} and that set-entailment is in Π_2^{P}. By reducing the decision problems for disjunctive ASP to regular FASP (see Proposition 4.9), one can also derive resp. Σ_2^{P}-hardness and Π_2^{P}-hardness. Hence for regular FASP without any restrictions, we obtain Σ_2^{P}-completeness for existence and set-membership and Π_2^{P}-completeness for set-entailment (see Proposition 4.10).

Proposition 4.9

Let P be a disjunctive ASP program and let $I \in \mathcal{P}(\mathcal{L}_P)$. Define the regular FASP program P' as follows:

$$P' = \{a_1 \oplus \ldots \oplus a_n \leftarrow b_1 \otimes \ldots \otimes b_m \otimes \text{not}\, c_1 \otimes \ldots \otimes \text{not}\, c_k \mid$$
$$(a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \text{not}\, c_1 \wedge \ldots \wedge \text{not}\, c_k) \in P\}$$
$$\cup \{a \leftarrow a \oplus a \mid a \in \mathcal{L}_P\}.$$

Then I is an answer set of P iff I is an answer set of P'.

Proof. First note that $I \in \mathcal{F}(\mathcal{L}_P)$ models a rule of the form $a \leftarrow a \oplus a$ iff $\min(2I(a), 1) \leq I(a)$. This is only possible if $I(a) = 0$ or $I(a) = 1$. The proposition then follows from the fact that the Łukasiewicz connectives restricted to values in $\{0, 1\}$ agree with the corresponding classical connectives, and the semantics of ASP and FASP coincide in such a case. □

We will use Proposition 4.9 to reduce disjunctive ASP to regular FASP and normal ASP to regular normal FASP.

Proposition 4.10

1. Set-membership and existence for regular FASP are Σ_2^{P}-complete. Set-entailment is Π_2^{P}-complete.
2. Set-membership and existence for regular normal FASP are NP-hard and in Σ_2^{P}. Set-entailment is coNP-hard and in Π_2^{P}.

Proof. Since fuzzy equilibrium logic is a proper generalisation of FASP, we can use its complexity results [Schockaert et al. 2012] to obtain Σ_2^{P}-membership for set-membership and existence and Π_2^{P}-membership for set-entailment. This result holds for regular FASP as well as for regular normal FASP. The hardness results are obtained by reducing the decision problems for normal resp. disjunctive ASP (see Table 3.1) to regular normal resp. regular FASP which is possible due to Proposition 4.9. □

In Section 4.4.1, we will discuss the complexity of regular simple and definite FASP programs. We show that characterizing the complexity for regular simple FASP programs is equivalent to an open problem about integer equations [Gawlitza and Seidle 2007]. However, we can provide a pseudo-polynomial algorithm and show P-membership for several subclasses of regular definite FASP. We will then use these results in Section 4.4.2 to characterize the computational complexity for regular normal FASP in a more fine grained manner than in Proposition 4.10.

4.4.1 Complexity of regular definite FASP programs

In this section we will discuss the complexity results for programs consisting of rules of the form

$$a \leftarrow f(b_1, \dots b_m)$$

with $a, b_1, \dots b_m$ literals and/or constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$ and f a composition of $\otimes$, $\oplus$, $\vee$ and $\wedge$. By Proposition 4.1, we can restrict ourselves to programs in which each rule has at most two arguments in the body.

Satisfiability checking in Łukasiewicz logic can be polynomially reduced to checking the feasibility of a mixed integer program [Hähnle 1994]. As will be shown in Proposition 4.11, the NP-completeness of the latter decision problem [Hähnle 1994] and the fact that each rule in a regular definite FASP program can be seen as a formula in Łukasiewicz logic, it follows that the decision problems for regular simple and thus also for regular definite FASP are all in NP. Moreover, since the answer set of a regular simple FASP program is unique, we can also prove coNP-membership for regular definite FASP programs:

Proposition 4.11

Set-membership, existence and set-entailment for regular definite FASP is in NP $\cap$ coNP.

Proof. Each regular simple FASP program can be seen as a set of formulas in Łukasiewicz logic. Checking if such a set of formulas has a minimal fuzzy model can be polynomially reduced to checking the feasibility of a mixed integer program which is an NP-complete problem. Hence we obtain NP-membership for regular simple FASP for all decision problems. Moreover, since the answer set of a regular simple FASP program is unique, we obtain NP-membership for the complementary decision problems. By Remark 4.1, it follows that it can be checked whether a fuzzy interpretation is an answer set of a regular definite FASP program by checking if it is the unique answer set of a particular regular simple FASP program and checking if a set of constraints is satisfied, hence we obtain the same results for regular definite FASP. □

In general, to find the unique minimal fuzzy model of a regular simple FASP program P, one could use the immediate consequence operator Π_P (see Proposition 3.4). The minimal fuzzy model of P then equals the least fixpoint of Π_P. This least fixpoint can be found by repeatedly applying the immediate consequence operator starting from the fuzzy interpretation $I_0 : \mathcal{B}_P \to [0,1] \cap \mathbb{Q} : a \mapsto 0$. Unfortunately, the number of iterations that is needed to arrive at the least fixpoint can be exponential in the number of bits needed to represent the rules. Consider for example the program consisting of the following rule, where n is equal to the "size" of the problem.

$$a \leftarrow a \oplus \overline{\left(\frac{1}{2^n}\right)}$$

In that case 2^n iterations of the immediate consequence operator are needed to conclude that a should have truth value 1. Indeed, one starts with the fuzzy interpretation $I_0 : \mathcal{B}_P \to [0,1] \cap \mathbb{Q}$ such that $I_0(a) = 0$. The next applications give us $I_1(a) = \frac{1}{2^n}$, $I_2(a) = \frac{2}{2^n}$, $I_3(a) = \frac{3}{2^n}$, ..., $I_{2^n}(a) = \frac{2^n}{2^n} = 1$. Hence 2^n iterations are needed. However, the number of iterations of the immediate consequence operator is polynomial in the size of the largest integer occurring in the program. As the following proposition shows, this will always be the case, i.e. we can find the unique answer set of any regular simple FASP program in pseudo-polynomial time.

Proposition 4.12

The unique answer set of a regular simple FASP program can be found in pseudo-polynomial time.

Proof. Suppose m is the largest integer occurring in the program and n is equal to the size of the program. Then all constants $\overline{c}$ in the program are such that $c \in T = \{0, \frac{1}{k}, \ldots, \frac{k}{k}\}$ with k polynomial in m. After each application of Π_P, either the least fixpoint is found and the procedure terminates or the truth value of at least one atom is increased to a new value in T; hence there are at most $n \cdot k$ such iterations and the number of iterations is polynomial in m and n.

□

Proposition 4.13

The complexity of the decision problems for regular simple and regular definite FASP is polynomial if all constants are polynomially bounded, i.e. all constants $\overline{c}$ in the program are such that $c \in \{0, \frac{1}{k}, \ldots, \frac{k}{k}\}$ with k polynomial in the size of the program.

Proof. Follows from the proof of Proposition 4.12. □

For the above program with rule

$$a \leftarrow a \oplus \overline{\left(\frac{1}{2^n}\right)}$$

we could improve the immediate consequence operator by assigning to a immediately truth value 1. It remains unclear, however, whether a general method could be found that always finds the answer set in polynomial time. The connection of this question to a well-known open problem on the feasibility of systems of integer equations suggests that there is not likely to be a straightforward solution. More precisely, the unique minimal fuzzy model I of a regular simple FASP program P can be found by computing the least solution of a system of equations over the integers in which each equation is of the form $x_i = \alpha_i$ with the variables x_i on the left hand side pairwise distinct, i.e. x_i can only occur once as the left hand side of an equation, and the expressions α_i are built from variables, integers, addition, multiplication with positive constants, maximum and minimum. The translation from P to such a system is defined as follows. First, create a set $\hat{P}$ of Łukasiewicz formulas:

$$\hat{P} = \{r_b \to r_h \mid (r_h \leftarrow r_b) \in P\} \cup \{a \vee \overline{0} \to a \mid a \in \mathcal{B}_P\},$$

where we add tautologies of the form $a \vee \overline{0} \to a$ to ensure that each a obtains a positive value after translating to a system of equations over the integers. Next, create a new set $\hat{P}_1$ of Łukasiewicz formulas by replacing for each atom a in $\hat{P}$ the set of formulas with the same "head" a, $\alpha_1 \to a, \ldots \alpha_n \to a$ by the formula $\alpha_1 \vee \ldots \vee \alpha_n \to a$. Finally, define the set $\hat{P}_2$ of Łukasiewicz formulas:

$$\hat{P}_2 = \{\alpha \leftrightarrow \beta \mid (\alpha \to \beta) \in \hat{P}_1\}.$$

We can now transform the set $\hat{P}_2$ to a set S of equations over the integers. First, define $\hat{S}$:

$$\hat{S} = \{\alpha = \beta \mid (\alpha \leftrightarrow \beta) \in \hat{P}_2\}$$

This is justified by the fact that $[\alpha \leftrightarrow \beta]_I = 1$ iff $[\alpha]_I = [\beta]_I$. Each constant in some equation in $\hat{S}$ can be assumed to be of the form $\overline{\left(\frac{i}{k}\right)}$ for a fixed k. Each such constant $\overline{\left(\frac{i}{k}\right)}$

is then replaced by i, $a \otimes b$ becomes $\max(a + b - k, 0)$ and $a \oplus b$ becomes $\min(a + b, k)$. This gives us the set S of equations over the integers.

There is a positive integer solution $J(a), J(b), J(c)$ for $a = \max(b + c - k, 0)$ iff the fuzzy interpretation I defined by $I(a) = \frac{J(a)}{k}, I(b) = \frac{J(b)}{k}, I(c) = \frac{J(c)}{k}$ is a fuzzy model of $a \leftrightarrow (b \otimes c)$:

$$\begin{aligned}
[a \leftrightarrow (b \otimes c)]_I = 1 &\Leftrightarrow \frac{J(a)}{k} = T_L\left(\frac{J(b)}{k}, \frac{J(c)}{k}\right) \\
&\Leftrightarrow \frac{J(a)}{k} = \max\left(\frac{J(b)}{k} + \frac{J(c)}{k} - 1, 0\right) \\
&\Leftrightarrow \frac{J(a)}{k} = \max\left(\frac{J(b)}{k} + \frac{J(c)}{k} - \frac{k}{k}, 0\right) \\
&\Leftrightarrow J(a) = \max(J(b) + J(c) - k, 0)
\end{aligned}$$

Similarly, one obtains that there is a positive integer solution $J(a), J(b), J(c)$ for $a = \min(b + c, k)$ iff $I(a) = \frac{J(a)}{k}, I(b) = \frac{J(b)}{k}, I(c) = \frac{J(c)}{k}$ models $a \leftrightarrow (b \oplus c)$. The unique least solution of S then corresponds to the unique minimal model of P in this sense. In [Gawlitza and Seidle 2007], an algorithm is presented for computing least solutions of such systems of integer equations. Although in practice it turns out that the algorithm is very efficient, it is still an open problem (e.g. [Bjorklund et al. 2003]) whether it has polynomial time complexity.

Example 4.3

As an illustration, consider the regular simple FASP program consisting of the rules

$$\begin{aligned}
a &\leftarrow b \oplus \overline{\tfrac{1}{4}} \\
b &\leftarrow a \wedge \overline{\tfrac{1}{3}} \\
a &\leftarrow \overline{\tfrac{1}{2}}
\end{aligned}$$

The corresponding set $\hat{P}_2$ is

$$\begin{aligned}
a &\leftrightarrow ((b \oplus \overline{\tfrac{1}{4}}) \vee \overline{\tfrac{1}{2}}) \vee 0 \\
b &\leftrightarrow (a \wedge \overline{\tfrac{1}{3}}) \vee 0
\end{aligned}$$

We then have that I is the minimal fuzzy model of P iff I' is the least solution of

$$\begin{aligned}
a &= \max(\min(b + 3, 12), 6, \overline{0}) \\
b &= \max(\min(a, 4), \overline{0})
\end{aligned}$$

with $I(a) = \frac{I'(a)}{12}$ and $I(b) = \frac{I'(b)}{12}$.

However, as we will show in the following subsections, for several subclasses of regular simple (and definite) FASP programs, we can show P-membership, even if the constants in the program are not polynomially bounded. A summary of the complexity results for regular simple FASP can be found in Table 4.2 in Section 4.1.

Directed graphs and cycles

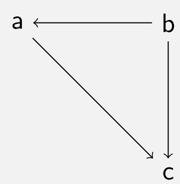
For a regular simple FASP program P we define the dependency graph $G(P)$ as follows. The vertices are the atoms occurring in the program and there is a directed edge from a to b if a occurs in the body of a rule with head b.

Example 4.4

Program P

$$\begin{array}{rcl} a & \leftarrow & b \\ a & \leftarrow & \overline{0.2} \\ c & \leftarrow & a \otimes b \\ c & \leftarrow & \overline{0.1} \\ b & \leftarrow & \overline{0.6} \end{array}$$

has the following dependency graph $G(P)$.



A path in a directed graph is a sequence of vertices such that from each vertex there is an edge to the next vertex in the sequence. A cycle is a path that begins and ends at the same vertex. If there are no cycles in the dependency graph of a regular simple FASP program, the immediate consequence operator will only need a polynomial number of steps to compute the least fixpoint:

Proposition 4.14

Consider a regular simple FASP program P such that the dependency graph $G(P)$ has no cycles and the longest path in $G(P)$ has length m. Then, the immediate consequence operator will only need m iterations to compute the answer set of P.

Proof. Start with the fuzzy interpretation that maps all atoms to 0. Define A_0 as the set of all truth constants in P. Define A_1 as the set of all atoms a that only depend on constants: each rule with head $a \in A_1$ is of the form $a \leftarrow \overline{c}$. After one application of Π_P, each $a \in A_1$ is given a truth value

$$I_1(a) = \sup\{[r_b]_{I_0} \mid (a \leftarrow r_b) \in P\} = \sup\{c \mid (a \leftarrow \overline{c}) \in P\}.$$

In further applications of Π_P, the truth value of $a \in A_1$ will not increase since it only depends on constants.

Next, define A_2 as the set of all atoms $a \notin A_1$ such that for each rule $a \leftarrow f(b,d)$ in P, we have that $b, d \in A_0 \cup A_1$. After two applications of Π_P, each $a \in A_2$ is assigned a truth value

$$I_2(a) = \sup\{[r_b]_{I_1} \mid (a \leftarrow r_b) \in P\} = \sup\{[f(b,d)]_{I_1} \mid (a \leftarrow f(b,d)) \in P\}.$$

In further applications of Π_P, the truth value of $a \in A_2$ will not increase since it only depends on atoms for which we already know the truth value will not increase anymore.

Continuing this procedure, after k iterations of Π_P, we get fixed truth values for all atoms $a \in A_k$: atoms $a \notin \cup_{i=1}^{k-1} A_i$ such that for each rule $a \leftarrow f(b,d)$ in P we have that $b, d \in \cup_{i=0}^{k-1} A_i$. Another application of Π_P will give fixed values for the atoms in A_{k+1}. Since there are no cycles, we have that $A_k = \emptyset$ for $k > m$. Hence, after m iterations, the least fixpoint of Π_P has been found. □

Example 4.5

Reconsider the following FASP program P from Example 4.4 with a cycle free dependency graph and longest path of length 3.

$$\begin{aligned} a &\leftarrow b \\ a &\leftarrow \overline{0.2} \\ c &\leftarrow a \otimes b \\ c &\leftarrow \overline{0.1} \\ b &\leftarrow \overline{0.6} \end{aligned}$$

We apply the immediate consequence operator Π_P to find the unique answer set. We start from the fuzzy interpretation $I_0 : \mathcal{B}_\mathcal{P} \to [0,1] \cap \mathbb{Q} : l \mapsto 0$. After one application of Π_P, we obtain the fuzzy interpretation $I_1 = \Pi_P(I_0)$ which is such that $I_1(a) = 0.2$, $I_1(b) = 0.6$ and $I_1(c) = 0.1$. After one more application, we have $I_2 = \Pi_P(I_1)$ which is such that $I_2(a) = 0.6$, $I_2(b) = 0.6$ and $I_2(c) = 0.1$. After the 3th application, the least fixpoint $I_3 = \Pi_P(I_2)$ of P has been found: $I_3(a) = 0.6$, $I_3(b) = 0.6$ and $I_3(c) = 0.2$.

Only disjunction in the body

For regular simple FASP programs with only disjunctions in the bodies of rules, we can always find the answer set in a polynomial number of steps, even if the dependency graph contains cycles and the constants are not polynomially bounded. In particular, suppose there is a cycle in the dependency graph of such a program such that there is a rule $c \leftarrow a \oplus b$ with a and c elements of the cycle and where b must have a truth value that is strictly positive. It then follows that the truth values of all atoms in that cycle will saturate to 1.

Proposition 4.15

Consider a regular simple FASP program P and its unique answer set I. Suppose P contains the following set of rules

$$\begin{aligned} a_2 &\leftarrow a_1 \oplus b_1 \\ a_3 &\leftarrow a_2 \oplus b_2 \\ &\vdots \\ a_1 &\leftarrow a_n \oplus b_n \end{aligned}$$

and we have $I(b_j) > 0$ for at least one $j \in \{1, \ldots, n\}$. Then for each $i \in \{1, \ldots, n\}$ we have $I(a_i) = 1$.

Proof. If for all $i \in \{1, \ldots, n\}$ we have $I(a_i) + I(b_i) \leq 1$ and thus $[a_i \oplus b_i]_I = I(a_i) + I(b_i)$, then we have $I(a_1) < I(a_1)$ a contradiction. Indeed,

$$I(a_1) \leq I(a_1) + I(b_1) \leq I(a_2) \leq I(a_2) + I(b_2) \leq \ldots \leq I(a_j) < I(a_j) + I(b_j)$$

$$\leq I(a_{j+1}) \leq \ldots \leq I(a_n) + I(b_n) \leq I(a_1).$$

Thus, there has to be some $k \in \{1, \ldots, n\}$ such that $I(a_k) + I(b_k) > 1$, but then $I(a_{k+1}) = 1$ (or $I(a_1) = 1$ if $k = n$). This implies that $I(a_i) = 1$ for each $i \in \{1, \ldots, n\}$. □

Remark 4.2

If we have that $I(b_i) = 0$ for each $i \in \{1, \ldots, n\}$, then we still have $I(a_1) = \ldots = I(a_n) = c$ for some $c \in [0, 1]$. Moreover, if there are no other rules in P that have one of the atoms a_i in the head of a rule, then $c = 0$.

We can define an equivalence relation on the set of vertices of an arbitrary directed graph G as follows. Two vertices u and v are equivalent if there is a cycle in G containing both u and v. The corresponding equivalence classes V_i lead to subgraphs G_i which are defined as the restrictions of G to the vertices in V_i and the edges between the vertices in V_i. Each of these subgraphs G_i is strongly connected, i.e. for each two vertices u and v in G_i, there is a path from u to v. Moreover, no G_i is a proper subgraph of another strongly connected subgraph of G. The graphs G_i are called the *strongly connected components* of G and can be seen as generalisations of cycles. Using Proposition 4.15 we can prove the following proposition.

Proposition 4.16

Consider a regular simple FASP program with only disjunctions in the bodies of rules and its unique answer set I. Suppose one of the rules in the program is of the form $a \leftarrow b \oplus d$ such that $I(b) > 0$ and a and d are atoms in the same strongly connected component S in $G(P)$. Then for all $s \in S$ we have $I(s) = 1$.

Proof. Suppose $S = \{a_1, \ldots, a_n\}$ with $a = a_1$ and $d = a_2$. By the definition of a strongly connected component there must be path from a_1 to a_2 and so on until we reach a_n. Similary, one can also find a path from a_n to a_1. If we consider all corresponding rules in P, we have a cycle consisting of the elements in S that contains a with $a \leftarrow b \oplus d$ and $I(b) > 0$. By Proposition 4.15, we can conclude that $I(a_i) = 1$ for all $i \in \{1, \ldots, n\}$. □

Remark 4.3

Similar as in Remark 4.2, we obtain that all the atoms in a strongly connected component must have the same truth value in the answer set.

Given a regular simple FASP program P with only disjunctions in the bodies of rules, we can find the answer set in polynomial time as follows. Using the algorithm of Tarjan [Tarjan 1972] the strongly connected components in $G(P)$ can be identified in polynomial time. Next, for each strongly connected component S a fresh atom a_S is introduced and each atom from S is replaced by a_S. By doing these substitutions it is possible that duplicate rules arise. A program P' is obtained by removing all rules that are already in the program, i.e. such that each rule occurs only once. Finally a program P'' is defined as follows.

- Rules of the form $a \leftarrow b \oplus c$ in P' where a, b and c are different atoms or constants remain unchanged.
- A rule of the form $a \leftarrow a \oplus b$ in P' where $b \neq a$ with b an atom or a constant is replaced by $a \leftarrow (b > 0)$.
- A rule of the form $a \leftarrow a \oplus a$ in P' is replaced by $a \leftarrow (a' > 0)$ with a fresh atom a'. In all other rules every occurrence of a is replaced by a'.

where the semantics for formulas of the form $(a > 0)$ is defined as $I(a > 0) = 1$ if $I(a) > 0$ and $I(a > 0) = 0$ otherwise.

The new program P'' is then a cycle free program. Since the semantics for formulas $(a > 0)$ is characterized by an increasing function, the immediate consequence operator can be used to compute the minimal model I'' of P'' (see [Janssen 2011]) which coincides with the answer set I of P in the sense that for each $a \in S$ we have $I(a) = I''(a_S)$. Since the proof of Proposition 4.14 does not rely on the fact that the FASP programs adhere to the Łukasiewicz semantics, we can use a very similar proof to show that the answer set of programs containing formulas of the form $(a > 0)$ in the body of rules will be found in polynomial time.

Corollary 4.4

The unique answer set of a regular simple FASP program with only disjunction in the body of the rules can be found in polynomial time.

Proof. Consider a regular simple FASP program P with only disjunctions in the bodies of rules. We can find the answer set I in polynomial time as follows. Using the algorithm of Tarjan [Tarjan 1972] the strongly connected components in $G(P)$ can be identified in polynomial time. Next, for each strongly connected component S we introduce a fresh atom a_S. This is followed by defining a cycle free program P' that is obtained from P by replacing each atom from S by a_S and this for each strongly connected component S. Moreover, superfluous rules are removed in the sense that no rule appears more than once in P'. From P', we then obtain a program P'' as follows.

- Rules of the form $a \leftarrow b \oplus c$ where a, b and c are different atoms or constants remain unchanged.
- A rule of the form $a \leftarrow a \oplus b$ where $b \neq a$ with b an atom or a constant is replaced by $a \leftarrow (b > 0)$.

- A rule of the form $a \leftarrow a \oplus a$ is replaced by $a \leftarrow (a' > 0)$ with a fresh atom a'. In all other rules every occurrence of a is replaced by a'.

Note that these are the only possible types of rules in P'. The semantics for formulas of the form $(a > 0)$ are defined by increasing functions and hence the immediate consequence operator can be used to compute the minimal model of P'', see [Janssen 2011] for details. Since Proposition 4.14 does not exploit the fact that we have FASP programs under Łukasiewicz semantics, we can use the same proof to show that if formulas $(a > 0)$ are allowed in the bodies of rules that the immediate consequence operator will only need a polynomial number of steps to compute the minimal model I'' of P''. Finally we choose $I(a) = I''(a_S)$ for each $a \in S$. By Remark 4.3 and Proposition 4.16 it follows that I is the answer set of P.

□

Example 4.6

Consider the following program P

$$\begin{array}{rcl} a & \leftarrow & b \\ a & \leftarrow & a \oplus \overline{\frac{1}{2^n}} \\ b & \leftarrow & a \oplus c \\ c & \leftarrow & b \end{array}$$

with n an integer larger than the size of the program.

The program P has exactly one strongly connected component $S = \{a, b, c\}$. The corresponding program P' is

$$\begin{array}{rcl} a_S & \leftarrow & a_S \\ a_S & \leftarrow & a_S \oplus \overline{\frac{1}{2^n}} \\ a_S & \leftarrow & a_S \oplus a_S \end{array}$$

From P' we obtain the corresponding cycle free program P''

$$\begin{array}{rcl} a'_S & \leftarrow & (\overline{0} > 0) \\ a'_S & \leftarrow & (\overline{\frac{1}{2^n}} > 0) \\ a_S & \leftarrow & (a'_S > 0) \end{array}$$

To obtain the answer set of P, we then apply the immediate consequence operator to the program P''. We start from a fuzzy interpretation $I_0 : \mathcal{B}_{P''} \rightarrow [0, 1] \cap \mathbb{Q} : a' \mapsto 0$. After one application of $\Pi_{P''}$ we obtain $I_1 = \Pi_{P''}(I_0)$ which is defined as follows:

$I_1(a'_S) = 1$ and $I_1(a_S) = 0$. After one more application we obtain the least fixpoint $I_2 = \Pi_{P''}(I_1)$ where $I_2(a'_S) = I_2(a_S) = 1$. This fixpoint then coincides with the unique answer set I of P: $I(a) = I(b) = I(c) = I_2(a_S) = 1$.

Example 4.7

Consider the following program P

$$\begin{array}{lcl} a & \leftarrow & b \oplus c \\ b & \leftarrow & a \\ d & \leftarrow & c \\ c & \leftarrow & \overline{0.3} \\ c & \leftarrow & d \oplus e \\ a & \leftarrow & c \oplus e \end{array}$$

The program P has three strongly connected components $S_1 = \{a, b\}$, $S_2 = \{c, d\}$ and $S_3 = \{e\}$. Hence the corresponding program P' is

$$\begin{array}{lcl} a_{S_1} & \leftarrow & a_{S_1} \oplus a_{S_2} \\ a_{S_1} & \leftarrow & a_{S_1} \\ a_{S_2} & \leftarrow & a_{S_2} \\ a_{S_2} & \leftarrow & \overline{0.3} \\ a_{S_2} & \leftarrow & a_{S_2} \oplus a_{S_3} \\ a_{S_1} & \leftarrow & a_{S_2} \oplus a_{S_3} \end{array}$$

From P' we obtain the corresponding cycle free program P''

$$\begin{array}{lcl} a_{S_1} & \leftarrow & (a_{S_2} > 0) \\ a_{S_1} & \leftarrow & (\overline{0} > 0) \\ a_{S_2} & \leftarrow & (\overline{0} > 0) \\ a_{S_2} & \leftarrow & \overline{0.3} \\ a_{S_2} & \leftarrow & (a_{S_3} > 0) \\ a_{S_1} & \leftarrow & a_{S_2} \oplus a_{S_3} \end{array}$$

To obtain the answer set of P, we apply the immediate consequence operator to program P''. We start from a fuzzy interpretation $I_0 : \mathcal{B}_{P''} \to [0,1] \cap \mathbb{Q} : a' \mapsto 0$. After one iteration of $\Pi_{P''}$ we obtain $I_1 = \Pi_{P''}(I_0)$ which is defined as follows: $I_1(a_{S_1}) = I_1(a_{S_3}) = 0$ and $I_1(a_{S_2}) = 0.3$. After one more iteration we obtain the least fixpoint $I_2 = \Pi_{P''}(I_1)$ where $I_2(a_{S_1}) = 1$, $I_2(a_{S_2}) = 0.3$ and $I_2(a_{S_3}) = 0$. This fixpoint then coincides with the unique answer set I of P: $I(a) = I(b) = I_2(a_{S_1}) = 1$, $I(c) = I(d) = I_2(a_{S_2}) = 0.3$ and $I(e) = I_2(a_{S_3}) = 0$.

4.4.2 Complexity of regular normal FASP programs

For normal programs in classical ASP, NP-membership follows straightforwardly from the fact that we can guess an answer set and verify that the guess is an answer set in polynomial time. In contrast, due to the infinite number of possible truth values, in FASP not every answer set can be guessed in polynomial time. To address this issue, by analyzing the geometrical structure of fuzzy equilibrium models, [Schockaert et al. 2012] shows that whenever there is an answer set I such that $I(l) \geq \lambda$ for a literal l, there always is an answer set J such that $J(l) \geq \lambda$ and such that for each literal l, $J(l)$ can be encoded using a polynomial number of bits. This means that we can verify whether $I(l) \geq \lambda$ for a regular normal FASP program by guessing an answer set in polynomial time and verifying that the guess is an answer set. As a result, several of the P-membership results for regular definite programs directly translate to NP-membership results for regular normal programs. The only exception is the class of regular normal FASP programs with polynomially bounded constants. Indeed, to check whether I is an answer set of such a program P it has to be verified that I is an answer set of P^I but P^I does not necessarily belong to the class of regular normal FASP programs with polynomially bounded constants. We also obtain the same results for the existence problem since it is the special case of the membership problem with $\lambda = 0$. For set-entailment we obtain coNP-membership if set-membership is in NP. Indeed, the complement of the set-entailment problem is "Given a program P, a literal l and a value λ_l, does there exist an answer set I of P such that $I(l) < \lambda_l$?" By similar results that can be found in [Schockaert et al. 2012], we know that if such an answer set exists, there is always one that can be encoded using a polynomial number of bits.

Proposition 4.17

1. The set-membership and the existence problem for the class of regular normal FASP programs with only disjunction in the bodies of rules is in NP. Set-entailment for this class of programs is in coNP.

2. The set-membership and the existence problem for the class of regular normal FASP programs with cycle free dependency graphs is in NP. Set-entailment for this class of programs is in coNP.

Proof. Let us first show that set-membership and existence are in NP. Suppose P is a regular normal FASP program in one of the subclasses subscribed in the statement of the proposition. From the analysis of the geometrical structure underlying fuzzy equilibrium models which is a proper generalisation of regular FASP [Schockaert et al. 2012], it follows that a FASP program P has an answer set I such that $I(l) \geq \lambda_l$ for some $l \in \mathcal{L}_\mathcal{P}$ and $\lambda_I \in [0,1] \cap \mathbb{Q}$ iff there is such an answer set that can be encoded using a polynomial number of bits. Hence we can guess such an answer set I in polynomial time. The reduct P^I then belongs to the corresponding subclass of regular definite FASP programs. For set-membership and existence (special case with $\lambda_l = 0$) it then remains to be verified that I is an answer set of P^I. But this follows easily from the fact that the answer set of P^I can be determined in polynomial time. To show that set-entailment is in coNP, we need to show that the complement of the decision problem "Given a FASP program P, a literal l and a value λ_l, does there exist an answer set I of P such that $I(l) < \lambda_l$?" is in NP. By a similar result from [Schockaert et al. 2012], it follows that such an answer set, if it exists, can be guessed in polynomial time. The reduct then belongs to the corresponding subclass of regular definite FASP programs for which the unique answer set can be determined in polynomial time. □

Moreover, as shown in Proposition 4.9, we can reduce the considered decision problems for normal ASP to regular normal FASP.

Corollary 4.5

Existence and set-membership for regular normal FASP with polynomially bounded constants is NP-hard. Set-entailment is coNP-hard.

Proof. By Proposition 4.9, it follows that a normal ASP program can be reduced to a regular normal FASP program with polynomially bounded constants. Since set-membership and existence for normal ASP are NP-complete (Table 3.1), it then follows that these decision problems are NP-hard for regular normal FASP with polynomially bounded constants. The fact that set-entailment is coNP-hard follows from the coNP-completeness for normal ASP (Table 3.1). □

4.5 Reduction to bilevel linear programming

In this section, we will show that we can translate strict disjunctive FASP programs into bilevel linear programs such that there is a one-to-one correspondence between particular solutions of the bilevel linear program and the answer sets of the FASP program.

This implementation into bilevel linear programming can then be used as a basis to build solvers for FASP.

Bilevel linear programming problems are optimization problems in which the set of all variables is divided into two disjoint sets $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_m\}$. An assignment to the variables will be denoted by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ for X and by a vector $\mathbf{y} = (y_1, \ldots, y_m)$ for Y. Intuitively, there are two agents, a leader who is responsible for the variables in X and a follower responsible for the variables in Y. Each vector $\mathbf{y}$ has to be chosen by the follower in function of the choice by the leader $\mathbf{x}$ as an optimal solution of the so-called *lower level problem* or the *follower's problem*. Knowing that the follower will react in that way, the leader wants to optimize his objective function in the so-called *upper level problem* or the *leader's problem*.

In a bilevel linear program all objective functions and constraints are linear. In particular, the type of bilevel linear programming problem in which we are interested is given by Bard [Bard 1998]:

$$\begin{aligned}
&\mathbf{x}^* = \arg\min_{\mathbf{x}} c_1\mathbf{x} + d_1\mathbf{y}^* \\
&\text{s.t. } A_1\mathbf{x} + B_1\mathbf{y}^* \leq b_1 \\
&\qquad\qquad \mathbf{y}^* = \arg\min_{\mathbf{y}} c_2\mathbf{x} + d_2\mathbf{y} \\
&\qquad\qquad \text{s.t. } A_2\mathbf{x} + B_2\mathbf{y} \leq b_2
\end{aligned}$$

where $c_1, c_2 \in \mathbb{R}^n$, $d_1, d_2 \in \mathbb{R}^m$, $b_1 \in \mathbb{R}^p$, $b_2 \in \mathbb{R}^q$, $A_1 \in \mathbb{R}^{p\times n}$, $B_1 \in \mathbb{R}^{p\times m}$, $A_2 \in \mathbb{R}^{q\times n}$ and $B_2 \in \mathbb{R}^{q\times m}$.

Now consider a strict disjunctive FASP program P. Without loss of generality we may assume that this program contains no strong negation (see Lemma 4.1). We will translate P to a bilevel linear program Q such that the solutions of Q correspond to the answer sets of P. By definition, I is an answer set of P iff I is an answer set of P^I. Informally, a guess I needs to be made first and then it has to be checked whether this guess corresponds to an answer set of P. If $\mathcal{B}_P = \{a_1, \ldots, a_n\}$, then we will define the vector $\tilde{\mathbf{a}} = (\tilde{a}_1, \ldots, \tilde{a}_n)$ and the vector $\tilde{\mathbf{a}}' = (\tilde{a}'_1, \ldots, \tilde{a}'_n)$ where the vector $\tilde{\mathbf{a}}$ represents the truth values of the atoms in $\{a_1, \ldots, a_n\}$ and the vector $\tilde{\mathbf{a}}'$ intuitively represents the truth values of the guesses for the atoms. For each such guess I, represented by $\tilde{\mathbf{a}}'$, we want to check if it is a minimal fuzzy model of P^I. Note that P^I is a positive FASP program in which each rule is of the form

$$r : l_1 \oplus \ldots \oplus l_n \leftarrow x_1 \otimes \ldots \otimes x_m, \tag{4.7}$$

with l_i, x_j atoms and/or truth constants. As in the proof of Proposition 4.2, a fuzzy interpretation $J \in \mathcal{F}(\mathcal{L}_P)$ is a model of r iff

$$J(l_1) + \ldots + J(l_n) \geq J(x_1) + \ldots + J(x_m) - (m - 1).$$

Thus for each rule $r \in P^I$ we have a constraint $x_1 + \ldots + x_m - m + 1 \leq l_1 + \ldots + l_n$.

Hence, for each guess $\tilde{\mathbf{a}}'$, i.e. an interpretation I, we check if there is a minimal model J of P^I such that $J(a_i) \leq I(a_i)$ by minimizing all elements in the vector $\tilde{\mathbf{a}}$ subject to the constraints arising from P^I. This is the follower's problem. Finally, the guess is chosen such that the differences between $J(a_i)$ and $I(a_i)$ are as small as possible. This can be done by minimizing the function $\sum_{i=1}^{n}(\tilde{a}'_i - \tilde{a}_i)$. If this sum is equal to 0, we have found an answer set. If this sum is not equal to 0, there cannot be an answer set. From the results in [Schockaert et al. 2012], it follows that if a bilevel linear program has a solution, then it also has a rational solution.

More structured, we have the following proposition.

Proposition 4.18

Given a strict disjunctive FASP program P not containing strong negation such that $\mathcal{B}_P = \{a_1, \ldots, a_n\}$. Define the following bilevel linear program Q_P.

$$
\begin{aligned}
&\tilde{\mathbf{a}}'^{*} = \arg\min_{\tilde{\mathbf{a}}'} \textstyle\sum_{i=1}^{n}(\tilde{a}'_i - \tilde{a}^{*}_i)\\
&\text{s.t. } 0 \leq \tilde{a}'_i \leq 1\\
&\qquad\qquad \tilde{\mathbf{a}}^{*} = \arg\min_{\tilde{\mathbf{a}}} \textstyle\sum_{i=1}^{n} \tilde{a}_i\\
&\qquad\qquad \text{s.t. } \tilde{a}_i \leq \tilde{a}'_i, 0 \leq \tilde{a}_i \leq 1 \text{ and}\\
&\qquad\qquad\quad (\textstyle\sum_{j=1}^{m} x_j) - m + 1 \leq \textstyle\sum_{i=1}^{n} l_i \text{ for each rule } (4.7)\\
&\qquad\qquad\quad \text{in the reduct of } P \text{ w.r.t. } \tilde{\mathbf{a}}'
\end{aligned}
$$

Then

1. If Q_P has a rational solution $\tilde{\mathbf{a}}^{*} = (\tilde{a}_1, \ldots, \tilde{a}_n)$, $\tilde{\mathbf{a}}'^{*} = (\tilde{a}'_1, \ldots, \tilde{a}'_n)$ such that the objective function of the upper level problem is evaluated to 0, then $I : \mathcal{B}_P \to [0, 1] \cap \mathbb{Q} : a_i \mapsto \tilde{a}_i$ is an answer set of P.

2. If I is an answer set of P, then $\tilde{\mathbf{a}}^{*} = (\tilde{a}_1, \ldots, \tilde{a}_n)$, $\tilde{\mathbf{a}}'^{*} = (\tilde{a}_1, \ldots, \tilde{a}_n)$ where $I(a_i) = \tilde{a}_i$ for each $i \in \{1, \ldots, n\}$ is a solution of Q_P such that the objective function of the upper level problem is evaluated to 0.

Proof. 1. Suppose Q_P has a rational solution $\tilde{\mathbf{a}}^{*} = (\tilde{a}_1, \ldots, \tilde{a}_n)$, $\tilde{\mathbf{a}}'^{*} = (\tilde{a}'_1, \ldots, \tilde{a}'_n)$ such that the objective function of the upper level problem is evaluated to 0. We show that $I : \mathcal{B}_P \to [0, 1] \cap \mathbb{Q} : a_i \mapsto \tilde{a}_i$ is a minimal fuzzy model of P^I. First note that if $\sum_{i=1}^{n}(\tilde{a}'_i - \tilde{a}_i) = 0$ it must hold that $\tilde{a}'_i = \tilde{a}_i$ for all $i \in \{1, \ldots, n\}$ since we have the constraints $\tilde{a}_i \leq \tilde{a}'_i$. By the constraints in the lower level problem it then follows that I is a fuzzy model of P^I. Now suppose there exists a fuzzy model J of P^I such that $J < I$. Then $\tilde{\mathbf{a}}^{*} = (\tilde{a}_1, \ldots, \tilde{a}_n)$, $\hat{\mathbf{a}}'^{*} = (\hat{a}'_1, \ldots, \hat{a}'_n)$ where $J(a_i) = \hat{a}'_i$ is a solution of Q_P with $\sum_{i=1}^{n}(\hat{a}'_i - \tilde{a}_i) < \sum_{i=1}^{n}(\tilde{a}'_i - \tilde{a}_i)$, a contradiction.

2. Suppose I is an answer set of P. We need to show that $\tilde{\mathbf{a}}^* = (\tilde{a}_1, \ldots, \tilde{a}_n)$, $\tilde{\mathbf{a}}'^* = (\tilde{a}_1, \ldots, \tilde{a}_n)$ is a solution of Q_P. As in the proof of Proposition 4.8, we can show that if the leader makes a choice $\tilde{\mathbf{a}}'^* = (a'_1, \ldots, a'_n)$ which can be seen as a fuzzy interpretation $I'(a_i) = a'_i$ of P, that $\tilde{\mathbf{a}}^* = (a^*_1, \ldots, a^*_n)$ where $J(a_i) = a^*_i$ is a minimal fuzzy model of $P^{I'}$ are the possible optimal solutions of the lower level problem. Since $\tilde{\mathbf{a}}^* = \tilde{\mathbf{a}}'^*$ and the fact that if the leader makes the choice $\tilde{\mathbf{a}}'^* = (\tilde{a}_1, \ldots, \tilde{a}_n)$, that $\tilde{\mathbf{a}}^* = (\tilde{a}_1, \ldots, \tilde{a}_n)$ is an optimal solution of the lower level problem, we have found a solution of Q_P.

□

Example 4.8

Consider the following strict normal FASP program P.

$$
\begin{aligned}
a &\leftarrow \text{not}\, b \\
b &\leftarrow \text{not}\, a
\end{aligned}
$$

The corresponding bilevel linear program is

$$
\begin{array}{l}
\arg\min_{\tilde{a}',\tilde{b}'}[(\tilde{a}' - \tilde{a}) + (\tilde{b}' - \tilde{b})] \\
\text{s.t. } 0 \leq \tilde{a}', \tilde{b}' \leq 1 \\
\qquad\qquad \arg\min_{\tilde{a},\tilde{b}}[\tilde{a} + \tilde{b}] \\
\qquad\qquad \text{s.t. } 0 \leq \tilde{a}, \tilde{b} \leq 1, \tilde{a} \leq \tilde{a}', \tilde{b} \leq \tilde{b}' \\
\qquad\qquad\quad 1 - \tilde{a}' \leq \tilde{b}, 1 - \tilde{b}' \leq \tilde{a}
\end{array}
$$

The only assignments to the variables for which the objective function of the leader is 0 are the ones with $\tilde{a}' = \tilde{a}$, $\tilde{b}' = \tilde{b}$ and $\tilde{a}' = 1 - \tilde{b}'$. This coincides with the answer sets of P (see Example 3.17): I_x with $I_x(a) = x$ and $I_x(b) = 1 - x$ for any $x \in [0, 1] \cap \mathbb{Q}$.

Remark 4.4

A similar construction can be used if ASP is combined with other fuzzy logics, e.g. product logic, but the resulting bilevel program will not necessarily be linear.

4.6 Conclusion

In this chapter, we presented an overview of the computational complexity of FASP under Łukasiewicz semantics. The main contributions in this chapter are the following:

- Although existence and set-membership are Σ_2^{P}-complete for disjunctive ASP, for strict disjunctive and strict normal FASP we were able to show NP-completeness. Moreover, we showed that not allowing constraints and strong negation does not affect the complexity for set-membership.
- We showed that the existence of an answer set for a strict normal FASP program without constraints and without strong negation is always guaranteed and hence that the complexity of the existence problem for this class of FASP programs is "constant". However, for strict disjunctive FASP without constraints and without strong negation we were only able to show membership in NP for the existence problem.
- If more syntactic freedom is allowed, i.e. for regular FASP programs, then we can show Σ_2^{P}-completeness for set-membership and existence and Π_2^{P}-completeness for set-entailment by using known complexity results about fuzzy equilibrium logic [Schockaert et al. 2012]. However, if we restrict ourselves to programs with at most one literal in the head of each rule, then we can only show Σ_2^{P}-membership and NP-hardness for set-membership and existence and Π_2^{P}-membership and coNP-hardness for set-entailment. If in addition, we do not allow "not" in the rules we can only find a pseudo-polynomial time algorithm to compute answer sets based on computing least fixpoints.
- Although in general we can only show membership in NP $\cap$ coNP, for several subclasses of the class of regular definite FASP programs we can show P-membership. In particular, for regular definite FASP programs with only conjunction and maximum or only disjunction in the body of rules we can provide a polynomial time algorithm to compute answer sets. This is also the case for regular definite FASP programs with a cycle free dependency graph or with only polynomially bounded constants.

An overview of the complexity results can be found in Tables 4.1 and 4.2. Finally, we have proposed an implementation of strict disjunctive FASP using bilevel linear programming.

Some open problems remain:

- Does there exist a polynomial time algorithm to compute the answer set of a regular simple FASP program?
- Is existence NP-hard for strict disjunctive FASP if constraints and strong negation are not allowed?

- Is regular normal FASP in NP?

5 | Embedding fuzzy answer set programming in fuzzy autoepistemic logic

5.1 Introduction

Logic programming, which contains answer set programming (ASP) (Section 3.1) as a special case, has had a significant impact on the development of nonmonotonic logics and vice versa [Baral and Gelfond 1994]. In particular, as discussed in Section 3.1.3, ASP can be embedded in autoepistemic logic (Section 2.1). In this chapter we combine autoepistemic logic and fuzzy logics (Section 2.2) and show that the answer sets of a FASP program (Section 3.2) can be equivalently described as stable expansions in the resulting fuzzy autoepistemic logic.

Besides autoepistemic logic, other translations of logic programming to various nonmonotonic logics have been investigated as well, e.g. circumscription ([Lifschitz 1988], [Przymusinski 1988]) and default logic ([Bidoit and Froidevaux 1991a], [Marek and Truszczyński 1989]). Next to autoepistemic logic, reflexive autoepistemic logic has also been used to characterise the semantics of ASP [Marek and Truszczyński 1993]. Reflexive autoepistemic logic [Schwarz 1992] has several almost identical semantic characterisations of expansions as autoepistemic logic but it models knowledge rather than belief. In reflexive autoepistemic logic, a formula is believed (known) if it is true in all possible worlds w.r.t.

the beliefs of the agent and in the actual world. The major difference with autoepistemic logic is that belief allows cyclic arguments and knowledge does not; if you believe a statement ϕ, then it is justified to include ϕ in some "belief set". For knowledge this is not the case. The observation that a rule $p \leftarrow p$ in some ASP program does not justify the inclusion of p into an answer set led the authors in [Marek and Truszczyński 1993] to use reflexive autoepistemic logic. Reflexive autoepistemic logic turns out to be equivalent to autoepistemic logic in the sense that there exist translation from each logic to the other preserving the notion of expansion.

The fuzzy autoepistemic logic we will define in this section is useful to reflect on one's beliefs about the *degrees* to which some properties are satisfied. Consider for example my reason for not believing that my brother smokes *a lot*. If he smoked *a lot*, his breath would smell *often*. Since I do not smell it *often*, I do not believe he smokes *a lot*. Intuitively, if the truth value of $\mathrm{B}\varphi$ is equal to c, this means that it is believed that φ is true *at least* to degree c. Hence, from one point of view φ is believed to the degree c and from another point of view, there is a Boolean form of belief that the truth value of φ is at least c. Furthermore, note how these views generalise the notion of belief from classical autoepistemic logic, in the sense that having $\mathrm{B}\varphi$ false corresponds to having φ true to at least degree 0, i.e. being completely ignorant about φ, and having $\mathrm{B}\varphi$ true corresponds to having φ true at least to degree 1, i.e. believing φ to be true. We show that many important properties from classical autoepistemic logic remain valid when generalising to fuzzy autoepistemic logic.

For regular normal FASP programs, i.e. regular FASP programs with exactly one literal or constant in the head, we show that the answer sets correspond to the models of an associated fuzzy autoepistemic theory. Specifically, it turns out that the translation from normal ASP to classical autoepistemic logic (Theorem 3.1) can be generalised in a straightforward way. However, similar as for classical ASP, this correspondence is not valid for programs with more complex formulas in the head of the rules. To deal with such FASP programs, we observe that for ASP the results on the equivalence between answer sets of an ASP program and a corresponding autoepistemic theory (Theorem 3.2) are based on the logic of minimal belief and negation as failure (MNBF) ([Lin and Shoham 1992], [Lifschitz 1994]). The (classical) logic MNBF uses two independent modal operators, corresponding to resp. a "minimal belief" modality and negation-as-failure. MBNF is thus an extension of autoepistemic logic with the "minimal knowledge operator" due to Halpern and Moses [Halpern and Moses 1984]. A simplified version of MBNF (from [Lifschitz 1994]), which will also be used in this paper, can be used to simulate some forms of default logic and circumscription, as well as some logic programming languages. In this chapter, we will introduce a fuzzy version of MNBF. As for ASP, this will provide us with a tool to show that for regular FASP the answer sets correspond to particular models of an associated fuzzy autoepistemic theory. The fact that this important relationship is preserved provides further insight into the nature of FASP, and at the same time serves as a justification for the particular fuzzy autoepistemic logic we introduce in this chapter.

Note that the language of (fuzzy) autoepistemic logic is much more expressive than the theories we need to represent the (fuzzy) answer set programs. Among others, this could serve as a useful basis for defining or comparing extensions to the basic language of ASP since the computational complexity does not increase when moving from ASP to autoepistemic logic [Gottlob 1992]. This might open doors to define extensions for FASP. With the exceptions of e.g. [Łukasiewicz and Straccia 2007], [Schockaert et al. 2012] and [Straccia et al. 2009], most work on FASP is restricted to programs with exactly one atom in the head. In this manuscript we consider a rather general form of FASP programs; the heads of rules are not restricted to single atoms and connectives can in principal be interpreted by arbitrary $[0, 1]^n \to [0, 1]$-mappings. The fuzzy equilibrium logic introduced in [Schockaert et al. 2012], another generalisation of FASP, also allows such constructs. In [Schockaert et al. 2012] a correspondence between fuzzy equilibrium logic models and answer sets of FASP programs was shown. Apart from these exceptions, it appears that little work has been done on nonmonotonic fuzzy logics and their relationship with fuzzy answer set programming.

As will become clear in Chapter 6, fuzzy autoepistemic logic is closely related to fuzzy modal logics. Finitely many-valued modal logics with graded accessibility relations have been studied in for example [Fitting 1992a] and [Fitting 1992b]. In [Fitting 1992c] Fitting's previous work is extended to finitely many-valued nonmonotonic modal logics. In particular autoepistemic logic is generalised by allowing a finite number of truth values. In [Koutras and Zachos 2000], reflexive autoepistemic logic is generalised by allowing a finite set of truth values. All these generalisations use finitely many truth values whereas we introduce a continuous generalisation for autoepistemic logic.

In the next section we will introduce fuzzy autoepistemic logic, investigate some of its properties and in Section 5.3 we give a motivating example. In Section 5.4 we analyse the relationship between regular normal FASP and fuzzy autoepistemic logic. In Section 5.5, we show that fuzzy autoepistemic logic generalises regular FASP. To do so, we will define fuzzy MBNF. A conclusion is given in Section 5.6.

5.2 Fuzzy autoepistemic logic

In this section we will formally define fuzzy autoepistemic logic, combining the ideas of autoepistemic logic (Section 2.1) and fuzzy logics (Section 2.2).

The language $\mathcal{L}_B$ of fuzzy autoepistemic logic is defined as follows. Formulas are built from a countable set of atoms A, the set of truth constants $\{\overline{c} \mid c \in [0,1] \cap \mathbb{Q}\}$, the set of connectives $F = \bigcup_{n \in \mathbb{N}} F_n$ with F_n the set of n-ary connectives and a modal operator B.

Definition 5.1

The language $\mathcal{L}_B$ is recursively defined as follows

- $a \in A$ is a formula.
- $\overline{c}$ with $c \in [0,1] \cap \mathbb{Q}$ is a formula.
- If α is a formula, then $\mathrm{B}\alpha$ is a formula.
- If $\alpha_1, \ldots, \alpha_n$ are formulas, then $f(\alpha_1, \ldots, \alpha_n)$ is a formula for every $f \in F_n$ with $n \in \mathbb{N}$.

A set of formulas in $\mathcal{L}_B$ is called a *fuzzy autoepistemic theory* and formulas not containing the operator B are called *objective*.

Similar to FASP, typically we will use connectives f from a given fuzzy logic (Section 2.2). As for classical autoepistemic logic we define

$$A' = A \cup \{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}_B\}.$$

We will sometimes treat A' as a set of atoms and consider formulas recursively built from A', the set of truth constants $\{\overline{c} \mid c \in [0,1] \cap \mathbb{Q}\}$ and the set of connectives $F = \bigcup_{n \in \mathbb{N}} F_n$ with F_n the set of n-ary connectives. For a formula $\alpha \in \mathcal{L}_B$ we will denote by α^* the corresponding formula in this non modal language $\mathcal{L}_B^*$:

- $a^* = a$ for a variable a,
- $\overline{c}^* = \overline{c}$ for a truth constant $\overline{c}$,
- $(f(\alpha_1, \ldots, \alpha_n))^* = f(\alpha_1^*, \ldots, \alpha_n^*)$ for $f \in F_n$ and $\alpha_1, \ldots, \alpha_n \in \mathcal{L}_B$,
- $(\mathrm{B}\alpha)^* = p_{B\alpha}$ with $p_{B\alpha}$ a fresh variable for $\alpha \in \mathcal{L}_B$.

We write $T^* = \{\alpha^* \mid \alpha \in T\}$ for a fuzzy autoepistemic theory T. We consider evaluations $v : A' \to [0,1]$ such that $v(\overline{c}) = c$ for truth constants in $\{\overline{c} \mid c \in [0,1] \cap \mathbb{Q}\}$ and $v(f(\alpha_1, \ldots, \alpha_n)) = \mathtt{f}(v(\alpha_1), \ldots, v(\alpha_n))$ where $f \in F_n$ is interpreted by $\mathtt{f} : [0,1]^n \to [0,1]$. We will denote this set of evaluations by Ω^*.

Using this trick, we will now define stable fuzzy expansions, generalising Definition 2.3.

Definition 5.2

Consider a fuzzy autoepistemic theory T in $\mathcal{L}_B$ and a mapping $E : \mathcal{L}_B \to [0,1]$. Then E is a *stable fuzzy expansion* of T if for each $\phi \in \mathcal{L}_B$

$$E(\phi) = \inf \{v(\phi^*) \mid v \in \Omega^*, \forall \alpha \in T : v(\alpha^*) = 1 \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = E(\varphi)\} .$$

Note that Definition 5.2 generalises the definition of a stable expansion in the following sense. A classical stable expansion of T is a set E which can be seen as a mapping $\overline{E} : \mathcal{L}_B \to \{0,1\}$ where $\overline{E}(\alpha) = 1$ iff $\alpha \in E$. It then follows that $\overline{E}(\phi) = 1$ iff $v(\phi^*) = 1$ for all $v \in \Omega^*$ such that $v(\alpha^*) = 1$ for all $\alpha \in T$ and $v((\mathrm{B}\varphi)^*) = E(\varphi)$ for all $\varphi \in \mathcal{L}_B$. Now suppose we only consider classical (two valued) evaluations $v \in \Omega^*$, i.e. evaluations $v : A' \to \{0,1\}$, then $\varphi \in E$ implies $v((\mathrm{B}\varphi)^*) = 1$ and $\varphi \notin E$ implies $1 - v(\sim(\mathrm{B}\varphi)^*) = v((\mathrm{B}\varphi)^*) = 0$ or $v(\sim(\mathrm{B}\varphi)^*) = 1$.

Remark 5.1

Suppose E is a stable fuzzy expansion of $\{\alpha\}$ with $\alpha \in \mathcal{L}_B$. By Definition 5.2 we thus have that $w((\mathrm{B}\alpha)^*) = E(\alpha) \leq v(\alpha^*)$ for all $v, w \in \Omega^*$ such that $w(\alpha) = v(\alpha) = 1$ for all $\alpha \in T^*$ and $w((\mathrm{B}\varphi)^*) = v((\mathrm{B}\varphi)^*) = E(\varphi)$ for all $\varphi \in \mathcal{L}_B$. Thus $v((\mathrm{B}\alpha)^*)$ determines a lower bound on the truth degree of α^*.

We will now generalise Definitions 2.4 and 2.5 and show that the correspondence between stable expansions and possible world autoepistemic models remains valid when generalising to the many-valued case. First of all, we will generalise the Kripke style possible world semantics for autoepistemic formulas.

Definition 5.3

Truth for fuzzy autoepistemic formulas is defined relative to structures (v, S) where $v \in \Omega$ and $S \subseteq \Omega$ with Ω the set of all mappings $w : A \to [0,1]$ such that

- $w(\overline{c}) = c$ for truth constants in $\{\overline{c} \mid c \in [0,1] \cap \mathbb{Q}\}$
- $w(f(\alpha_1, \ldots, \alpha_n)) = \mathtt{f}(w(\alpha_1), \ldots, w(\alpha_n))$ for autoepistemic formulas $\alpha_1, \ldots, \alpha_n$ and where $f \in F_n$ is interpreted by $\mathtt{f} : [0,1]^n \to [0,1]$ with $n \in \mathbb{N}$.

Truth evaluations for fuzzy autoepistemic formulas are then recursively defined as follows:

- $\|a\|_{(v,S)} = v(a)$ for $a \in A$,

- $\|\overline{c}\|_{(v,S)} = c$ for truth constants in $\{\overline{c} \mid c \in [0,1] \cap \mathbb{Q}\}$,
- $\|\mathrm{B}\alpha\|_{(v,S)} = \inf_{w \in S} \|\alpha\|_{(w,S)}$ for a fuzzy autoepistemic formula α,
- $\|f(\alpha_1, \ldots, \alpha_n)\|_{(v,S)} = \mathtt{f}(\|\alpha_1\|_{(v,S)}, \ldots, \|\alpha_n\|_{(v,S)})$ where $f \in F_n$ is interpreted by $\mathtt{f} : [0,1]^n \to [0,1]$ with $n \in \mathbb{N}$.

Remark 5.2

Note that the following are equivalent expressions for a formula α, $c \in [0,1] \cap \mathbb{Q}$, a residual implicator $\to$ and $S \subseteq \Omega$:

$$\begin{aligned} \forall v \in S : \|\mathrm{B}\alpha\|_{(v,S)} \geq c \quad &\Leftrightarrow \quad \forall v \in S : \inf_{w \in S} \|\alpha\|_{(w,S)} \geq c \\ &\Leftrightarrow \quad \forall w \in S : \|\alpha\|_{(w,S)} \geq c \\ &\Leftrightarrow \quad \forall w \in S : \|\overline{c} \to \alpha\|_{(w,S)} = 1 \\ &\Leftrightarrow \quad \forall v \in \Omega : \|\mathrm{B}(\overline{c} \to \alpha)\|_{(v,S)} = 1 \end{aligned}$$

Hence on the one hand, we believe α to degree c, and on the other hand, we have a Boolean belief in the formula $\overline{c} \to \alpha$.

Remark 5.3

In possibilistic logic, the semantics are defined in terms of a possibility distribution over propositional interpretations, i.e. by mappings $\pi : \mathcal{P}(A) \to [0,1]$ where $\mathcal{P}(A)$ is the set of all propositional interpretations over A (see also in Section 2.1). Such a mapping encodes for each interpretation or possible world $I \in \mathcal{P}(A)$ to what extent it is possible that it refers to the real world, or in other words, to what extent available knowledge does not exclude I from being the real world.

Syntactically, a formula in possibilistic logic corresponds to a propositional formula, encapsulated by a graded modality. In particular, formulas are of the form (α, λ), with α a formula in classical propositional logic and $\lambda \in [0,1]$, with the intended meaning that $\sup\{\pi(I) \mid I \text{ does not model } \alpha\} \leq 1 - \lambda$.

Hence, there is a clear duality between the semantics of fuzzy autoepistemic logic as we have defined it here and the semantics of possibilistic logic. Indeed, whereas we have defined the semantics of fuzzy autoepistemic logic in terms of a classical set of fuzzy interpretations in Ω, possibilistic logic is defined in terms of a fuzzy set of classical interpretations in $\mathcal{P}(A)$. This duality also reflects the different way in which the modality should be interpreted. In possibilistic logic, and in a number of graded modal logics, the strength by which an agent believes a proposition can be expressed. Degrees of belief are then used to express that some propositions are considered to be more plausible than others. In contrast, our approach does not deal with such strengths of belief; believing a proposition α to degree λ is interpreted as a Boolean belief in the proposition $\overline{\lambda} \to \alpha$, i.e. degrees of belief are used to express that some propositions are true to a greater extent than others. Of course, one can also imagine a logic based on fuzzy sets of fuzzy interpretations, as was proposed for example in [Alsinet and Godo 2000].

Definition 5.4

A set $S \subseteq \Omega$ is a *fuzzy possible world autoepistemic model* of a fuzzy autoepistemic theory T iff

$$S = \{v \in \Omega \mid \forall \varphi \in T : \|\varphi\|_{(v,S)} = 1\}.$$

Similar as for classical autoepistemic logic (Definition 2.4), the set of possible worlds w.r.t. the beliefs of the agent is a fuzzy possible world autoepistemic model of T if it is exactly the set of worlds in which T is true.

Example 5.1

Consider the fuzzy autoepistemic theory

$$T = \{{\sim}\mathrm{B}a \to b, {\sim}\mathrm{B}b \to a\}$$

with $a, b \in A$ and the negation $\sim$ and implication $\to$ from Łukasiewicz logic (Section 2.2.3). Consider $v \in \Omega$ and $S \subseteq \Omega$. For the first formula of T we have

$$\begin{aligned} \|{\sim}\mathrm{B}a \to b\|_{(v,S)} = 1 \quad &\Leftrightarrow \quad \|{\sim}\mathrm{B}a\|_{(v,S)} \leq \|b\|_{(v,S)} \\ &\Leftrightarrow \quad 1 - \|\mathrm{B}a\|_{(v,S)} \leq v(b) \\ &\Leftrightarrow \quad 1 - \inf_{w \in S} w(a) \leq v(b) \\ &\Leftrightarrow \quad 1 - v(b) \leq \inf_{w \in S} w(a) \end{aligned}$$

By symmetry we also have $\|{\sim}\mathrm{B}b \to a\|_{(v,S)} = 1 \Leftrightarrow 1 - v(a) \leq \inf_{w \in S} w(b)$. It follows that if S is a fuzzy possible world autoepistemic model of T, then

$$S = \left\{ v \in \Omega \mid 1 - v(b) \leq \inf_{w \in S} w(a) \text{ and } 1 - v(a) \leq \inf_{w \in S} w(b) \right\}.$$

For such a set S, let $x = \inf_{w \in S} w(a)$ and $y = \inf_{w \in S} w(b)$. We show that $y = 1-x$. For each $w \in S$, we have $1 - w(b) \leq x$ and thus

$$1 - y = 1 - \inf_{w \in S} w(b) = \sup_{w \in S}(1 - w(b)) \leq x.$$

Hence $x+y \geq 1$. S contains all $v \in \Omega$ such that $1-v(b) \leq x$ and $1-v(a) \leq y$, thus there exists a $v_0 \in S$ with $v_0(a) = 1 - y$. We obtain $x = \inf_{w \in S} w(a) \leq v_0(a) = 1 - y$, hence $x + y \leq 1$. From $x + y \leq 1$ and $x + y \geq 1$, it follows that $x + y = 1$. Thus, if S is a fuzzy possible world autoepistemic model of T, then

$$S = \{v \in \Omega \mid v(b) \geq 1 - x \text{ and } v(a) \geq x\}$$

for some $x \in [0, 1]$. Moreover, we can prove that each set of this form is a fuzzy possible world autoepistemic model of T, such that we can conclude that the fuzzy possible world autoepistemic models of T are exactly all sets of this form. Indeed, define for each $x \in [0, 1]$

$$S_x = \{v \in \Omega \mid v(b) \geq 1 - x \text{ and } v(a) \geq x\}.$$

To conclude that S_x is a fuzzy possible world autoepistemic model of T, we consider $v \in S_x$ and observe that for all $w \in S_x$ we have $1 - v(b) \leq x \leq w(a)$, thus $1 - v(b) \leq \inf_{w \in S_x} w(a)$. Similary, $1 - v(a) \leq \inf_{w \in S_x} w(b)$.

Definition 5.5

A mapping $T : \mathcal{L}_B \to [0, 1]$ is called the *fuzzy belief set* of $S \subseteq \Omega$ iff

$$T(\varphi) = \inf_{v \in S} \|\varphi\|_{(v,S)} = \|\mathrm{B}\varphi\|_{(w,S)},$$

for all $\varphi \in \mathcal{L}_B$ with $w \in \Omega$ arbitrary. We will write $\mathsf{Th}(S)$ to denote this fuzzy set of formulas.

The fuzzy belief set of S is thus a fuzzy set assigning to each formula φ a truth degree which coincides with the truth value of $\mathrm{B}\varphi$, i.e. the belief in the truth value of φ.

We now present some lemmas that will help us to prove generalisations of Propositions 2.1 and 2.2 in respectively Proposition 5.2 and 5.1. To prove Proposition 5.1 we will use the result from Proposition 5.2. In these lemmas we will use the following notation. Since $A \subseteq A'$ we can define restrictions for evaluations $v \in \Omega^*$. We define by $v_{|A}$ the element in Ω such that $v_{|A}(a) = v(a)$ for all $a \in A$.

Lemma 5.1

Consider a set $Q \subseteq \Omega^*$ such that for each $v \in Q$ and $\varphi \in \mathcal{L}_B$ we have that $v((\mathrm{B}\varphi)^*) = \inf_{w \in Q} w(\varphi^*)$. Then, for $\alpha \in \mathcal{L}_B$, $S = \{v_{|A} \mid v \in Q\} \subseteq \Omega$ and $v \in Q$ we have

$$\|\alpha\|_{(v_{|A},S)} = v(\alpha^*).$$

Proof. We will prove this lemma by induction on the structure of the formulas.

- If α is an objective formula, then we obtain $\|\alpha\|_{(v_{|A},S)} = v_{|A}(\alpha) = v(\alpha^*)$.
- Consider the formula $f(\varphi_1, \ldots, \varphi_n)$ such that $\|\varphi_i\|_{(w_{|A},S)} = w(\varphi_i^*)$ for all $w \in Q$ and this for all $i \in \{1, \ldots, n\}$. For $v \in Q$ it then holds that

$$\begin{aligned}\|f(\varphi_1, \ldots, \varphi_n)\|_{(v_{|A},S)} &= \mathtt{f}(\|\varphi_1\|_{(v_{|A},S)}, \ldots, \|\varphi_n\|_{(v_{|A},S)}) \\ &= \mathtt{f}(v(\varphi_1^*), \ldots, v(\varphi_n^*)) \\ &= v((f(\varphi_1, \ldots, \varphi_n))^*)\end{aligned}$$

- Consider $\mathrm{B}\varphi$ such that $\|\varphi\|_{(w_{|A},S)} = w(\varphi^*)$ for all $w \in Q$. We will now show that $\|\mathrm{B}\varphi\|_{(v_{|A},S)} = v(\varphi^*)$ for all $v \in Q$. For each $v \in Q$ it holds by definition of Q and by the induction hypothesis that

$$v((\mathrm{B}\varphi)^*) = \inf_{w \in Q} w(\varphi^*) = \inf_{w \in Q} \|\varphi\|_{(w_{|A},S)}.$$

  Finally by the definition of S and by Definition 5.3 we have

$$\inf_{w \in Q} \|\varphi\|_{(w_{|A},S)} = \inf_{z \in S} \|\varphi\|_{(z,S)} = \|\mathrm{B}\varphi\|_{(v_{|A},S)}.$$

□

Lemma 5.2

Consider a set $S \subseteq \Omega$ and an evaluation $v \in \Omega^*$ such that for each $\varphi \in \mathcal{L}_B$ it holds that $v((\mathrm{B}\varphi)^*) = \mathsf{Th}(S)(\varphi)$. Then, for $\alpha \in \mathcal{L}_B$ we have

$$\|\alpha\|_{(v_{|A},S)} = v(\alpha^*).$$

Proof. We will prove this lemma by induction on the structure of the formulas.

- If α is an objective formula, then we obtain $\|\alpha\|_{(v_{|A},S)} = v_{|A}(\alpha) = v(\alpha^*)$.
- Consider the formula $f(\varphi_1, \ldots, \varphi_n)$ such that $\|\varphi_i\|_{(v_{|A},S)} = v(\varphi_i^*)$ for all $i \in \{1, \ldots, n\}$. It follows that

$$\begin{aligned}\|f(\varphi_1, \ldots, \varphi_n)\|_{(v_{|A},S)} &= \mathtt{f}(\|\varphi_1\|_{(v_{|A},S)}, \ldots, \|\varphi_n\|_{(v_{|A},S)}) \\ &= \mathtt{f}(v(\varphi_1^*), \ldots, v(\varphi_n^*)) \\ &= v((f(\varphi_1, \ldots, \varphi_n))^*)\end{aligned}$$

- Consider $\mathrm{B}\varphi$ such that $\|\varphi\|_{(v_{|A},S)} = v(\varphi^*)$. It follows that $\|\mathrm{B}\varphi\|_{(v_{|A},S)} = v(\varphi^*)$. Indeed, by Definition 5.5, we have

$$v((\mathrm{B}\varphi)^*) = \mathsf{Th}(S)(\varphi) = \|\mathrm{B}\varphi\|_{(v_{|A},S)}.$$

□

Proposition 5.1

Consider a fuzzy autoepistemic theory T.

1. If $E : \mathcal{L}_B \to [0,1]$ is a stable fuzzy expansion of T, then there exists a fuzzy possible world autoepistemic model S of T such that $E = \mathsf{Th}(S)$.
2. If $S \subseteq \Omega$ is a fuzzy possible world autoepistemic model of T, then $E = \mathsf{Th}(S)$ is a stable fuzzy expansion of T.

Proof. We prove both statements separately.

1. Suppose E is a stable fuzzy expansion of T. We define

$$Q = \{v \in \Omega^* \mid \forall \alpha \in T : v(\alpha^*) = 1 \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = E(\varphi)\}$$

and

$$S = \left\{v_{|A} \mid v \in Q\right\}.$$

We prove that S is a fuzzy possible world autoepistemic model of T and that $E = \mathsf{Th}(S)$. By Lemma 5.1 we have for $\alpha \in \mathcal{L}_B$ and $v \in Q$ that

$$\|\alpha\|_{(v_{|A},S)} = v(\alpha^*).$$

We can now prove that S is a fuzzy possible world autoepistemic model of T. For $w \in S$ we have that $w = v_{|A}$ for some $v \in Q$. For every $\varphi \in T$ it then holds that

$$\|\varphi\|_{(w,S)} = \|\varphi\|_{(v_{|A},S)} = v(\varphi^*) = 1,$$

where the last equality follows from the fact that $\varphi \in T$. We have now shown

$$S \subseteq \left\{ w \in \Omega \mid \forall \varphi \in T : \|\varphi\|_{(w,S)} = 1 \right\}.$$

For the converse inclusion, consider an interpretation $w \in \Omega$ such that $\|\varphi\|_{(w,S)} = 1$ for all $\varphi \in T$. We will now construct $v \in Q$ such that $v_{|A} = w$. For $a \in A$, we define $v(a) = w(a)$ and for a formula $\alpha \in \mathcal{L}_B$, we define $v((\mathrm{B}\alpha)^*) = E(\alpha)$. It is then clear that $v_{|A} = w$. By Lemma 5.1 it follows that for each $\alpha \in T$ we have $v(\alpha^*) = \|\alpha\|_{(v_{|A},S)} = \|\alpha\|_{(w,S)} = 1$. It follows that $v \in Q$ and hence

$$S = \left\{ w \in \Omega \mid \forall \varphi \in T : \|\varphi\|_{(w,S)} = 1 \right\}.$$

Finally, we prove that $E(\alpha) = \mathsf{Th}(S)(\alpha)$ for each formula $\alpha \in \mathcal{L}_B$. For every $v \in Q$ we have by definition that $E(\alpha) = v((\mathrm{B}\alpha)^*)$ and by Lemma 5.1 that $v((\mathrm{B}\alpha)^*) = \|\mathrm{B}\alpha\|_{(v_{|A},S)}$. We also have $\|\mathrm{B}\alpha\|_{(v_{|A},S)} = \mathsf{Th}(S)(\alpha)$. We conclude

$$E(\alpha) = \mathsf{Th}(S)(\alpha).$$

2. Now suppose we have a fuzzy possible world autoepistemic model S of T. For each formula $\alpha \in \mathcal{L}_B$, we define $E(\alpha)$ as follows

$$E(\alpha) = \inf_{v \in M} v(\alpha^*),$$

with

$$M = \left\{ v \in \Omega^* \mid v_{|A} \in S \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = \mathsf{Th}(S)(\varphi) \right\}.$$

We show that $E = \mathsf{Th}(S)$ and that E is a stable fuzzy expansion of T. By Lemma 5.2, it follows that for $v \in M$ and $\alpha \in \mathcal{L}_B$

$$\|\alpha\|_{(v_{|A},S)} = v(\alpha^*).$$

Note that, similar as earlier in the proof, for each $w \in S$ we can find $v \in M$ such that $v_{|A} = w$. Thus by definition of M we have $S = \left\{ v_{|A} \mid v \in M \right\}$. These two observations lead to the equality $E = \mathsf{Th}(S)$. Indeed, for $\alpha \in \mathcal{L}_B$, we have

$$\mathsf{Th}(S)(\alpha) = \inf_{w \in S} \|\alpha\|_{(w,S)} = \inf_{v \in M} \|\alpha\|_{(v_{|A},S)} = \inf_{v \in M} v(\alpha^*) = E(\alpha).$$

To see that E is a stable fuzzy expansion of T, it is sufficient to prove that M is equal to

$$Q = \{ v \in \Omega^* \mid \forall \alpha \in T : v(\alpha^*) = 1 \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = E(\varphi) \}.$$

First, we show $M \subseteq Q$. Let $v \in M$. By Lemma 5.2, it follows that for $\alpha \in T$ we have $v(\alpha^*) = \|\alpha\|_{(v_{|A},S)} = 1$, where the last equality holds since S is a fuzzy possible world autoepistemic model of T. It remains to be shown that $v((\mathrm{B}\varphi)^*) = E(\varphi)$ for each $\varphi \in \mathcal{L}_B$. This follows easily:

$$E(\varphi) = \mathsf{Th}(S)(\varphi) = \|\mathrm{B}\varphi\|_{(v_{|A},S)} = v((\mathrm{B}\varphi)^*),$$

where the last equality follows from Lemma 5.2.

To conclude the proof, we show that $Q \subseteq M$. From $E = \mathsf{Th}(S)$ it follows that

$$Q \subseteq \{v \in \Omega^* \mid \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = \mathsf{Th}(S)(\varphi)\}.$$

To show that $Q \subseteq M$, it is sufficient to show that for $v \in Q$ it holds that $v_{|A} \in S$. From Lemma 5.2 and the fact that $v(\alpha^*) = 1$ for all $\alpha \in T$, we have that

$$\|\alpha\|_{(v_{|A},S)} = v(\alpha^*) = 1,$$

for $\alpha \in T$. This means that $v_{|A} \in S$ since S is a fuzzy possible world autoepistemic model of T.

□

Example 5.2

Reconsider the fuzzy autoepistemic theory

$$T = \{{\sim}\mathrm{B}a \to b, {\sim}\mathrm{B}b \to a\}$$

from Example 5.1. All fuzzy possible world autoepistemic models are of the form

$$S_x = \{v \in \Omega \mid v(b) \geq 1 - x \text{ and } v(a) \geq x\},$$

with $x \in [0,1]$. Hence, all stable fuzzy expansions of T are of the form E_x with $x \in [0,1]$ defined by $E_x(a) = \mathsf{Th}(S)(a) = \inf_{v \in S} v(a) = x$ and $E_x(b) = \mathsf{Th}(S)(b) = \inf_{v \in S} v(b) = 1 - x$.

Proposition 5.2

Every set of objective formulas in $\mathcal{L}_B$ has a unique stable fuzzy expansion.

Proof. By Definition 5.4, S is a fuzzy possible world autoepistemic model S of T iff

$$S = \left\{ v \in \Omega \mid \forall \varphi \in T : \|\varphi\|_{(v,S)} = 1 \right\}.$$

Since T only contains objective formulas, we have for all $\varphi \in T$ that $\|\varphi\|_{(v,S)} = v(\varphi)$. Thus

$$S = \{v \in \Omega \mid \forall \varphi \in T : v(\varphi) = 1\}$$

is the unique fuzzy possible world autoepistemic model of T. By Proposition 5.1 there is exactly one stable fuzzy expansion.

□

Remark 5.4

Every inconsistent set of objective formulas has the empty set as a unique fuzzy possible world autoepistemic model. Indeed, suppose T is an objective fuzzy autoepistemic theory, such that T contains α and $\sim\alpha$ with α an objective formula and $\sim$ interpreted by a negator. By Proposition 5.2, it follows that T has a unique fuzzy possible world autoepistemic model S. It follows that

$$S = \{v \in \Omega \mid \forall \varphi \in T : v(\varphi) = 1\} = \emptyset.$$

5.3 Motivating example

In this section we will revisit the sensor network example from Section 3.3.1. Recall that we have a wireless sensor network consisting of a set of devices that can sense their environment and communicate wirelessly. Suppose this network contains n sensors and for each sensor i ($i \in \{1, \ldots, n\}$), we denote the exact temperature at its location as the variable t_i and the measured temperature as the truth value $t'_i \in [0, 1] \cap \mathbb{Q}$. As in Section 3.3.1, a weighted graph is defined as follows. The vertices are the sensors and there is an edge between the vertices corresponding to sensor i and sensor j with weight $w_{ij} \in [0, 1] \cap \mathbb{Q}$ if we can reasonably assume, based on the locations of sensors i and j, that the temperature difference $|t'_i - t'_j|$ between these locations should be at most w_{ij}. We will now present two strategies to determine bounds on the actual temperatures, given the values w_{ij} and t'_i ($i, j \in \{1, \ldots, n\}$). We will use the connectives from Łukasiewicz logic and the Rescher implicator $\rightarrow_R$ which is defined as $\|\alpha \rightarrow_R \beta\|_{(v,S)} = 1$ if $\|\alpha\|_{(v,S)} \leq \|\beta\|_{(v,S)}$ and $\|\alpha \rightarrow_R \beta\|_{(v,S)} = 0$ otherwise for $\alpha, \beta \in \mathcal{L}_B$ and $v \in \Omega$, $S \subseteq \Omega$.

5.3.1 Is the sensor broken or not?

First we suppose the variable b_i represents the Boolean property "sensor i is broken". The formula $b_i \vee {\sim} b_i$ can be used to impose that the truth value of b_i is a binary value. Indeed, for a structure (v, S) we have $\|b_i \vee {\sim} b_i\|_{(v,S)} = 1$ iff $\max(v(b_i), 1 - v(b_i)) = 1$ iff $v(b_i) \in \{0, 1\}$. An alternative way of imposing that b_i is Boolean is to use the formula $b_i \oplus b_i \to b_i$ which can again only be satisfied if $v(b_i) \in \{0, 1\}$. If b_i has truth value 1, it means that sensor i is broken. If it has truth value 0, the sensor works normally. Suppose that each sensor can only display a temperature in $[0, 1] \cap \mathbb{Q}$ with a limited granularity of one decimal but we have no idea how the grounding of decimal numbers is defined. And that for a sensor that is not broken the maximum measurement error is 0.01. If for example the actual temperature is $t_i = 0.095$ and the sensor is not broken, then the measured temperature will be between 0.085 and 0.105. Since we do not know how the grounding works it is possible that the displayed temperature t'_i is equal to 0, 0.1 or 0.2. In formulas (2)-(9) we will provide bounds on the actual temperatures that are large enough to take into account all possible scenarios. We can now write the following formulas with $i, j \in \{1, \ldots, n\}$:

(1) $b_i \vee {\sim} b_i$

(2) $(\overline{0.2} \to_R \overline{t'_i}) \to (\mathrm{B}b_i \oplus (\overline{0.09} \to_R t_i))$

(3) $(\overline{0.3} \to_R \overline{t'_i}) \to (\mathrm{B}b_i \oplus (\overline{0.19} \to_R t_i))$

(4) $\ldots$

(5) $(\overline{1} \to_R \overline{t'_i}) \to (\mathrm{B}b_i \oplus (\overline{0.89} \to_R t_i))$

(6) $(\overline{t'_i} \to_R \overline{0}) \to (\mathrm{B}b_i \oplus (t_i \to_R \overline{0.11}))$

(7) $(\overline{t'_i} \to_R \overline{0.1}) \to (\mathrm{B}b_i \oplus (t_i \to_R \overline{0.21}))$

(8) $\ldots$

(9) $(\overline{t'_i} \to_R \overline{0.8}) \to (\mathrm{B}b_i \oplus (t_i \to_R \overline{0.91}))$

(10) $\overline{B_{ij}} \to (b_i \oplus b_j)$

(11) ${\sim}\mathrm{B}b_i \to {\sim} b_i$

Formulas (2) to (9) define the relationship between the measured and the actual temperature based on what is believed about sensors being broken or not and taking into account the granularity and the maximum measurement error. For instance, suppose you believe that sensor i is not broken and it is given that $t'_i = 0.4$. This means that there is a structure (v, S) such that $\|\mathrm{B}b_i\|_{(v,S)} = 0$ and $v(\overline{t'_i}) = 0.4$. Formulas (2) to (5) then impose that $0.29 \leq v(t_i)$ and formulas (6) to (9) that $v(t_i) \leq 0.51$. If you believe that sensor i is broken, you cannot conclude anything about t_i.

In formula (10), the truth constant $\overline{B_{ij}}$ is a short notation for $\overline{w_{ij}} \rightarrow_R \sim(\overline{t'_i} \leftrightarrow \overline{t'_j})$. Thus

$$\|\overline{B_{ij}}\|_{(v,S)} = 1 \text{ iff } w_{ij} \leq |t'_i - t'_j|.$$

The formula imposes that if the difference between t'_i and t'_j is too large with respect to the weight w_{ij}, then at least one of the sensors must be broken. Formula (11) captures the connection between broken sensors and what you believe about them. It is needed to ensure that a sufficient number of sensors is believed to be broken, which in turn ensures that we do not derive more about the actual temperatures t_i than is warranted. In other words, (11) enforces some form of minimality. It also ensures that we obtain a minimal set of broken sensors.

Consider as a concrete example a forest with three sensors. Suppose we have $t'_1 = 0.4$, $t'_2 = 0.9$ and $t'_3 = 0.5$ and $w_{1,2} = 0.2$, $w_{1,3} = 0.2$ and $w_{2,3} = 0.2$.

We obtain the following degrees of similarity. For an arbitrary structure (v, S) we have

- $\|\overline{t'_1} \leftrightarrow \overline{t'_2}\|_{(v,S)} = 0.5$
- $\|\overline{t'_1} \leftrightarrow \overline{t'_3}\|_{(v,S)} = 0.9$
- $\|\overline{t'_2} \leftrightarrow \overline{t'_3}\|_{(v,S)} = 0.6$

For a structure (v, S) to model formulas (10) and (11) for each sensor i, it must hold that

(a) $1 \leq v(b_1) + v(b_2)$

(b) $1 \leq v(b_2) + v(b_3)$

(c) $v(b_1) \leq \|\mathrm{B}b_1\|_{(v,S)}$

(d) $v(b_2) \leq \|\mathrm{B}b_2\|_{(v,S)}$

(e) $v(b_3) \leq \|\mathrm{B}b_3\|_{(v,S)}$

Suppose we want to find a fuzzy possible world autoepistemic model $S \subseteq \Omega$ of the set T consisting of formulas (1)-(11) for all sensors i. If there exists such a set S, then we have

$$S = \{v \in \Omega \mid \forall \alpha \in T : \|\alpha\|_{(v,S)} = 1\}.$$

Every $v \in S$ should satisfy inequalities (c)-(e) which implies that $v(b_i) \leq \inf_{z \in S} z(b_i) \leq v(b_i)$ and hence that $v(b_i) = \inf_{z \in S} z(b_i)$. It follows that there is a unique evaluation

$v \in S$. Moreover it is a minimal evaluation such that inequalities (a) and (b) are satisfied. Indeed, take $w \notin S = \{v\}$ such that $w < v$, $1 \leq w(b_1) + w(b_2)$ and $1 \leq w(b_2) + w(b_3)$. Then we have have

$$w(b_i) < v(b_i) = \inf_{z \in S} z(b_i) = \|\mathrm{B}b_i\|_{(w,S)}$$

This is a contradiction since v was the unique evaluation satisfying (c)-(e). It follows that there are at most 2 fuzzy possible world autoepistemic models, one in which $v(b_1) = 0$, $v(b_2) = 1$ and $v(b_3) = 0$. And another in which $v(b_1) = 1$, $v(b_2) = 0$ and $v(b_3) = 1$.

Finally, by including the formulas (2)-(9), we get two sets S_1 and S_2:

$$S_1 = \{v \in \Omega \mid v(b_1) = 0, v(b_2) = 1, v(b_3) = 0,$$

$$0.29 \leq v(t_1) \leq 0.51, 0 \leq v(t_2) \leq 1 \text{ and } 0.39 \leq v(t_3) \leq 0.61\}$$

and

$$S_2 = \{v \in \Omega \mid v(b_1) = 1, v(b_2) = 0, v(b_3) = 1,$$
$$0 \leq v(t_1) \leq 1, 0.79 \leq v(t_2) \leq 1 \text{ and } 0 \leq v(t_3) \leq 1\}.$$

Using Definition 5.4, one can show that S_1 and S_2 are indeed fuzzy possible world autoepistemic models of T. These are the only fuzzy possible world autoepistemic models of T.

In S_1 we have the model in which sensor 2 is broken and the others are not. There are lower and upper bounds given for the temperatures t_i. In S_2 we have the case in which sensors 1 and 3 are broken and sensor 2 is not. Corresponding intervals for the temperatures are given. There are no other possibilities since S_1 and S_2 are the unique fuzzy possible world autoepistemic models of T.

5.3.2 How big is the error on the measurement?

Now we suppose that the variable e_i represents the error in $[0, 1]$ on the temperature measured by sensor i. We can then write the following formulas with $i, j \in \{1, \ldots, n\}$:

(1) $\overline{t'_i} \rightarrow (\mathrm{B}e_i \oplus t_i)$

(2) $\sim\overline{t'_i} \rightarrow (\mathrm{B}e_i \oplus \sim t_i)$

(3) $\overline{E_{ij}} \rightarrow (e_i \oplus e_j)$

(4) $\sim\mathrm{B}e_i \rightarrow \sim e_i$

Formulas (1) and (2) define the relationship between the measured and the actual temperature. For a structure (v, S) we have that $v(\overline{t'_i}) - \|\mathrm{B}e_i\|_{(v,S)} \leq v(t_i) \leq v(\overline{t'_i}) + \|\mathrm{B}e_i\|_{(v,S)}$. In formula (3), $\overline{E_{ij}}$ is a short notation for $\sim(\overline{t'_i} \leftrightarrow \overline{t'_j}) \otimes \sim\overline{w_{ij}}$. Thus

$$\|\overline{E_{ij}}\|_{(v,S)} = \max(v(\sim(\overline{t'_i} \leftrightarrow \overline{t'_j})) - v(\overline{w_{ij}}), 0) = \max(|t'_i - t'_j| - w_{ij}, 0).$$

Formula (3) thus imposes that if the difference between t'_i and t'_j is too big with respect to the weight w_{ij}, then there must be something wrong with the sensors. The size of the error depends on how big the difference between t'_i and t'_j is.

Reconsider the previous example: $t'_1 = 0.4$, $t'_2 = 0.9$ and $t'_3 = 0.5$ and $w_{1,2} = 0.2$, $w_{1,3} = 0.2$ and $w_{2,3} = 0.2$.

For a structure (v, S) to model formulas (3) and (4) for each sensor i, it must hold that

(a) $0.3 \leq v(e_1) + v(e_2)$
(b) $0.2 \leq v(e_2) + v(e_3)$
(c) $v(e_1) \leq \|\mathrm{B}e_1\|_{(v,S)}$
(d) $v(e_2) \leq \|\mathrm{B}e_2\|_{(v,S)}$
(e) $v(e_3) \leq \|\mathrm{B}e_3\|_{(v,S)}$

Suppose we want to find a fuzzy possible world autoepistemic model $S \subseteq \Omega$ of the set T consisting of formulas (1)-(4) for all sensors i. If there exists such a set S, then we have

$$S = \{v \in \Omega \mid \forall \alpha \in T : \|\alpha\|_{(v,S)} = 1\}.$$

As before since every $v \in S$ should satisfy inequalities (c)-(e), it follows that there is a unique evaluation $v \in S$. Moreover it is a minimal evaluation such that inequalities (a) and (b) are satisfied.

There are infinitely many fuzzy possible world autoepistemic models S, i.e. there are infinitely many possibilities how the total error can be "divided" among the sensors. However, we know by the minimality of $v \in S$ that $v(e_1)$ must be less than or equal to 0.3, and $v(e_3)$ less than or equal to 0.2.

Let us consider some examples.

$$S_1 = \{v \in \Omega \mid v(e_1) = 0.01, v(e_2) = 0.29, v(e_3) = 0,$$
$$0.39 \leq v(t_1) \leq 0.41, 0.6 \leq v(t_2) \leq 1 \text{ and } v(t_3) = 0.5\}$$

and

$$S_2 = \{v \in \Omega \mid v(e_1) = 0.29, v(e_2) = 0.01, v(e_3) = 0.19,$$
$$0.11 \leq v(t_1) \leq 0.69, 0.89 \leq v(t_2) \leq 0.91 \text{ and } 0.31 \leq v(t_3) \leq 0.69\}.$$

For all such scenarios, exact errors on the measurement for each sensor and corresponding intervals for the temperature are given.

5.4 Relation between regular normal FASP and fuzzy autoepistemic logic

We will now show that the important relation between autoepistemic logic and ASP (Theorem 3.1) is preserved: fuzzy autoepistemic logic generalises regular normal FASP. By Lemmas 4.1 and 4.2 we can restrict ourselves without loss of generality to regular normal FASP programs without constraints and without strong negation. Hence we are interested in programs P in which each rule is of the form

$$r : a \leftarrow f(b_1, \ldots, b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_\mathrm{k}\, c_k),$$

with a an atom, b_i, c_j atoms and/or truth constants ($i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, k\}$) and f an $(m+k)$-ary connective such that the corresponding function $\mathtt{f} : [0,1]^{m+k} \to [0,1]$ is increasing in each of its arguments and where $\leftarrow$ is interpreted by a residual implicator I_T. The negation-as-failure operators not_j are interpreted by negators N_j.

We use a similar transformation as for normal ASP. For rule r we define the associated fuzzy autoepistemic formula $\lambda(r)$ as

$$f(b_1, \ldots, b_m, \sim_1 \mathrm{B}c_1, \ldots \sim_k \mathrm{B}c_k) \to a,$$

where $\sim_j$ is the negation that is interpreted by the same negator N_j as for not_j. The resulting fuzzy autoepistemic theory is $\lambda(P) = \{\lambda(r) \mid r \in P\}$. We will show in Theorem 5.1 that the stable fuzzy expansions of $\lambda(P)$ correspond to the answer sets of P. First we provide a lemma that characterises the relationship between stable fuzzy expansions of $\lambda(P)$ and stable fuzzy expansions of the autoepistemic theory corresponding to a specific reduct of the program P.

Lemma 5.3

Consider a mapping $E : \mathcal{L}_B \to [0,1]$ such that $E(x) \in \mathbb{Q}$ for all $x \in \mathcal{B}_P$ and a regular normal FASP program P without constraints and without strong negation. Then E is a stable fuzzy expansion of $\lambda(P)$ iff E is a stable fuzzy expansion of $\lambda(P^{\hat{E}})$ with $\hat{E} \in \mathcal{F}(\mathcal{B}_P)$ such that $\hat{E}(x) = E(x)$ for all $x \in \mathcal{B}_P$.

Proof. We need to prove that for each $\phi \in \mathcal{L}_B$ it holds that

$$\begin{aligned} &\inf \{v(\phi^*) \mid v \in \Omega^*, \forall \alpha \in \lambda(P) : v(\alpha^*) = 1 \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = E(\varphi)\} \\ &= \inf \left\{v(\phi^*) \mid v \in \Omega^*, \forall \alpha \in \lambda(P^{\hat{E}}) : v(\alpha^*) = 1 \text{ and } \forall \varphi \in \mathcal{L}_B : v((\mathrm{B}\varphi)^*) = E(\varphi)\right\}. \end{aligned}$$

We will show this by proving that $v(\alpha^*) = 1$ for all $\alpha \in \lambda(P)$ iff $v(\alpha^*) = 1$ for all $\alpha \in \lambda(P^{\hat{E}})$ as soon as $v((\mathrm{B}\varphi)^*) = E(\varphi)$ for all $\varphi \in \mathcal{L}_B$. We will show this by proving that for each

$$r : a \leftarrow f(b_1, \ldots, b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not_k}\, c_k)$$

in P we have

$$v\left((\lambda(r))^*\right) = v\left((\lambda(r^{\hat{E}}))^*\right)$$

for all mappings $v \in \Omega^*$ for which it holds that $v((\mathrm{B}\varphi)^*) = E(\varphi)$ for all $\varphi \in \mathcal{L}_B$. Indeed, for such a mapping v, an atom a and a negator N (with corresponding negation-as-failure operator not) we have

$$[\mathrm{not}\, a]_{\hat{E}} = N(\hat{E}(a)) = N(E(a)) = N(v((\mathrm{B}a)^*)) = N(v(p_{\mathrm{B}a}))$$

and we obtain

$$\begin{aligned} v\left((\lambda(r^{\hat{E}}))^*\right) &= v\left(f(b_1, \ldots, b_m, \overline{[\mathrm{not}_1\, c_1]_{\hat{E}}}, \ldots, \overline{[\mathrm{not_k}\, c_k]_{\hat{E}}}) \rightarrow a\right) \\ &= I_T\Big(\mathtt{f}(v(b_1), \ldots, v(b_m), [\mathrm{not}_1\, c_1]_{\hat{E}}, \ldots, [\mathrm{not_k}\, c_k]_{\hat{E}}), v(a)\Big) \\ &= I_T\Big(\mathtt{f}(v(b_1), \ldots, v(b_m), N_1(v(p_{\mathrm{B}c_1})), \ldots, N_k(v(p_{\mathrm{B}c_k})), v(a)\Big) \\ &= v\left(f(b_1, \ldots, b_m, \sim_1 p_{\mathrm{B}c_1}, \ldots, \sim_k p_{\mathrm{B}c_k}) \rightarrow a\right) \\ &= v\left((\lambda(r))^*\right) \end{aligned}$$

□

Now we can prove the generalisation of Theorem 3.1.

Theorem 5.1

Consider a regular normal FASP program P without constraints and without strong negation and $M \in \mathcal{F}(\mathcal{B}_P)$. M is an answer set of P iff $\lambda(P)$ has a stable fuzzy expansion $E : \mathcal{L}_B \rightarrow [0, 1]$ such that $E(a) = M(a)$ for all $a \in \mathcal{B}_P$.

Proof. ($\Rightarrow$) First suppose M is an answer set of P. Since P^M is a regular simple FASP program, M is the unique minimal fuzzy model of P^M. Since $\lambda(P^M)$ is a set of objective formulas, by Proposition 5.2 it has exactly one stable fuzzy expansion E. By Proposition 5.1 we know that $E = \mathrm{Th}(S)$ with S the unique fuzzy possible world autoepistemic model of $\lambda(P^M)$:

$$S = \left\{v \in \Omega \mid \forall \alpha \in \lambda(P^M) : \|\alpha\|_{(v,S)} = 1\right\}$$

$$= \left\{v \in \Omega \mid \forall \alpha \in \lambda(P^M) : v(\alpha) = 1\right\}.$$

Note that, since P^M does not contain negation-as-failure, the mapping $v_M \in \Omega$ such that $v_M(a) = M(a)$ for all $a \in \mathcal{B}_P$ and $v_M(a) = 0$ otherwise, is an element of S. Since M is the unique minimal fuzzy model of P^M, it must hold that v_M is the unique minimal element in S and hence that

$$E(a) = \mathsf{Th}(S)(a) = \inf_{v \in S} v(a) = v_M(a) = M(a)$$

for all $a \in \mathcal{B}_P$.

($\Leftarrow$) Now suppose that $\lambda(P)$ has a stable fuzzy expansion E such that $M(a) = E(a)$ for all $a \in \mathcal{B}_P$. We show that M is an answer set of P. Since P^M is a regular simple FASP program it has a unique minimal model I. We show that M equals I. Consider the following set of evaluations.

$$\begin{aligned} Q &= \left\{v \in \Omega \mid \forall \alpha \in \lambda(P^M) : v(\alpha) = 1\right\} \\ &= \left\{v \in \Omega \mid \forall \alpha \in \lambda(P^M) : \|\alpha\|_{(v,Q)} = 1\right\}. \end{aligned}$$

where the last equality follows from that fact that $\lambda(P^M)$ is a set of objective formulas. Note that $v_I \in \Omega$ with $v_I(a) = I(a)$ for all $a \in \mathcal{B}_P$ and $v_I(a) = 0$ otherwise is the unique minimal element of Q since P^M is a regular simple FASP program. Hence it remains to be shown that $v_I(a) = M(a)$ for all $a \in \mathcal{B}_P$. By Lemma 5.3 we know that E is a stable fuzzy expansion of $\lambda(P^M)$. This implies by Proposition 5.1 that $E = \mathsf{Th}(Q)$ which finishes the proof:

$$M(a) = E(a) = \mathsf{Th}(Q)(a) = \inf_{z \in Q} z(a) = v_I(a).$$

□

Example 5.3

Consider the following regular normal FASP program P under Łukasiewicz semantics from Example 3.12:

$$\begin{aligned} b &\leftarrow \text{not}\, a \\ a &\leftarrow \text{not}\, b \end{aligned}$$

We compute the answer sets by using the characterisation from Theorem 5.1. We compute the stable fuzzy expansions of

$$\lambda(P) = \{\sim\mathrm{B}a \rightarrow b, \sim\mathrm{B}b \rightarrow a\}$$

with $\sim$ the Łukasiewicz negation. Note that this is the fuzzy autoepistemic theory T we encountered in Examples 5.1 and 5.2. All stable fuzzy expansions of T are thus of the form E_x with $x \in [0,1]$ such that $E_x(a) = x$, $E_x(b) = 1 - x$. Hence all answer sets are of the form M_x with $x \in [0,1] \cap \mathbb{Q}$ such that $M_x(a) = x$ and $M_x(b) = 1 - x$. Note that for $x = 0$ and $x = 1$ we get the answer sets from Example 3.5.

Remark 5.5

Theorem 5.1 cannot be generalised to programs in which more complex formulas are allowed in heads of rules. For example, consider the regular FASP program P containing the single rule

$$a \oplus b \leftarrow \overline{1}$$

with a and b atoms and $\oplus$ and $\leftarrow$ connectives from Łukasiewicz logic. The corresponding fuzzy autoepistemic theory would be

$$\lambda(P) = \left\{\overline{1} \rightarrow a \oplus b\right\}.$$

All formulas in $\lambda(P)$ are objective, thus by Proposition 5.2, $\lambda(P)$ has only 1 stable fuzzy expansion. The program P however has infinitely many answer sets: for $x \in [0,1] \cap \mathbb{Q}$, one can easily verify that $M_x(a) = x$ and $M_x(b) = 1 - x$ defines an answer set of P.

In Section 5.5 however, we will show that regular FASP can be embedded in fuzzy autoepistemic logic. We will return to this example in Example 5.4.

5.5 Relation between regular FASP and fuzzy autoepistemic logic

In this section we will investigate the relationship between regular FASP, the fuzzy logic of minimal belief and negation-as-failure (FMBNF) which we will define in Section 5.5.2 and fuzzy autoepistemic logic which we introduced in Section 5.2. In particular, in Section 5.5.3 we will show how the answer sets of a regular FASP program correspond to the models of a corresponding FMBNF theory. In Section 5.5.4, we will then use this result to embed regular FASP in fuzzy autoepistemic logic.

By Lemma 4.1, we can restrict ourselves to regular FASP programs without strong negation. Hence we are interested in programs in which each rule is of the form

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \text{not}_1\, c_1, \ldots, \text{not}_k\, c_k),$$

with a_i, b_j, c_l atoms and/or truth constants corresponding to truth values in $[0,1] \cap \mathbb{Q}$ with $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, k\}$. The function g is a prefixnotation for an n-ary connective and f represents an $(m+k)$-ary connective corresponding to increasing functions $\texttt{g} : [0,1]^n \to [0,1]$ and $\texttt{f} : [0,1]^{m+k} \to [0,1]$. The negation-as-failure operators not_l correspond to negators N_L and $\leftarrow$ corresponds to some residual implicator I_T.

First we provide some background on the classical logic of minimal belief and negation-as-failure.

5.5.1 Logic of minimal belief and negation-as-failure (MBNF)

Lin and Shoham [Lin and Shoham 1992] defined a propositional nonmonotonic logic which uses two independent modal operators. One of them represents minimal belief and the other is related to the ideas of justification and of negation-as-failure. In this manuscript we consider a special case of this system: the logic of minimal belief and negation-as-failure (MBNF) [Lifschitz 1994]. It extends the logic of grounded knowledge of Lin and Shoham [Lin and Shoham 1992] with the theories of epistemic queries by Levesque [Levesque 1984] and Reiter [Reiter 1991].

Formulas of the propositional logic of MBNF are built from a countable set of atoms A, the constants true and false, the standard propositional connectives and two modal operators B and "not". If a formula contains neither B nor "not" it is called *objective*.

Truth is defined relative to a triple (I, S^b, S^n) with $I \in \mathcal{P}(A)$, $S^b \subseteq \mathcal{P}(A)$ and $S^n \subseteq \mathcal{P}(A)$. S^b is the set of interpretations for defining the meaning of the operator B and S^n for "not". If a formula is true in each interpretation in S^b, it is believed by the agent. If there exists an interpretation in S^n in which a formula is not true, then the agent does not believe it. Using this intuition, then in autoepistemic logic the sets S^n and S^b coincide. Note that for MBNF there is not necessarily a relation between S^b and S^n. This implies that there can exist formulas the agent believes and does not believe at the same time. The interpretation of formulas is defined as follows:

- $(I, S^b, S^n) \models p$ iff $p \in I$
- $(I, S^b, S^n) \models \overline{0}$ iff $\overline{0} \in I$
- $(I, S^b, S^n) \models (\alpha \to \beta)$ iff $(I, S^b, S^n) \not\models \alpha$ or $(I, S^b, S^n) \models \beta$
- $(I, S^b, S^n) \models \text{B}\alpha$ iff for every $J \in S^b$ it holds that $(J, S^b, S^n) \models \alpha$
- $(I, S^b, S^n) \models \text{not}\, \alpha$ iff for some $J \in S^n$ it holds that $(J, S^b, S^n) \not\models \alpha$

for $p \in A$ and formulas α and β.

Definition 5.6

Consider $I \in \mathcal{P}(A)$, $S \subseteq \mathcal{P}(A)$ and a MBNF theory T. The structure (I, S) is a *MBNF model* of T iff

1. for each $\alpha \in T$: $(I, S, S) \models \alpha$ and
2. there is no structure (I', S') such that $S \subset S'$ and $(I', S', S) \models \alpha$ for all $\alpha \in T$.

The maximality of S in Definition 5.6 expresses the idea of minimal belief: if the set of possible worlds is larger, then fewer propositions are believed. As pointed out in [Lin and Shoham 1992], minimizing knowledge is not sufficient to model a rational agent's knowledge or beliefs. Intuitively, the agent's conclusions based on negation-as-failure should be supported by his knowledge.

Disjunctive ASP programs can be simulated by theories in MBNF. For each rule

$$r : a_1 \vee \ldots \vee a_n \leftarrow b_1 \wedge \ldots \wedge b_m \wedge \mathrm{not}\, c_1 \wedge \ldots \wedge \mathrm{not}\, c_k,$$

in a disjunctive ASP program P, the formula $\mu(r)$

$$\mathrm{B}b_1 \wedge \ldots \wedge \mathrm{B}b_m \wedge \mathrm{not}\, c_1 \wedge \ldots \wedge \mathrm{not}\, c_k \rightarrow \mathrm{B}a_1 \vee \ldots \vee \mathrm{B}a_n$$

is added to the theory $\mu(P)$ in MBNF. Lifschitz [Lifschitz 1994] showed the following theorem with $\mathrm{Mod}(M) = \{I \in \mathcal{P}(A) \mid M \subseteq I\}$ for an interpretation $M \in \mathcal{P}(A)$.

Theorem 5.2: [Lifschitz 1994]

Consider a disjunctive ASP program P and $M \in \mathcal{P}(\mathcal{L}_P)$. M is an answer set of P iff there exists a structure (I, S) which is a MBNF model of $\mu(P)$ such that $S = \mathrm{Mod}(M)$.

Note that in Theorem 5.2, the interpretation I is arbitrary. This follows easily from the fact that all occurrences of literals in $\mu(P)$ are encapsulated by either B or "not". Using Theorem 5.2, Lifschitz and Schwarz [Lifschitz and Schwarz 1993], showed Theorem 3.2, i.e. that disjunctive ASP programs can also be modelled in autoepistemic logic.

5.5.2 Fuzzy logic of minimal belief and negation-as-failure (FMBNF)

In this section we will generalise the definitions from Section 5.5.1 and use these to establish the relationship between regular FASP and fuzzy autoepistemic logic in Section 5.5.4. The language of fuzzy logic of minimal belief and negation-as-failure (FMBNF) is recursively defined as follows

- $a \in A$ is a formula.
- $\bar{c}$ with $c \in [0,1] \cap \mathbb{Q}$ is a formula.
- If α is a formula, then $\mathrm{B}\alpha$ is a formula.
- If α is a formula, then $\mathrm{not_j}\, \alpha$ is a formula (for $j \in J$).
- If $\alpha_1, \ldots, \alpha_n$ are formulas, then $f(\alpha_1, \ldots, \alpha_n)$ is a formula for every $f \in F_n$ with $n \in \mathbb{N}$.

A *theory in FMBNF* is a set of formulas in FMBNF. If a formula contains neither B nor $\mathrm{not_j}$ it is called *objective*. The semantics are defined relative to triples (v, S^b, S^n) with $v \in \Omega$, $S^b \subseteq \Omega$ and $S^n \subseteq \Omega$. S^b is the set of evaluations for defining the meaning of B and S^n for the operators $\mathrm{not_j}$. The evaluation of formulas is defined as follows:

- $\|p\|_{(v,S^b,S^n)} = v(p)$
- $\|\bar{c}\|_{(v,S^b,S^n)} = c$
- $\|\mathrm{B}\alpha\|_{(v,S^b,S^n)} = \inf_{w \in S^b} \|\alpha\|_{(w,S^b,S^n)}$
- $\|\mathrm{not_j}\, \alpha\|_{(v,S^b,S^n)} = N_j \left(\inf_{w \in S^n} \|\alpha\|_{(w,S^b,S^n)}\right)$, where $\mathrm{not_j}$ is interpreted by the negator N_j
- $\|f(\alpha_1, \ldots, \alpha_n)\|_{(v,S^b,S^n)} = \mathtt{f}(\|\alpha_1\|_{(v,S^b,S^n)}, \ldots, \|\alpha_n\|_{(v,S^b,S^n)})$ where f is interpreted by $\mathtt{f} : [0,1]^n \to [0,1]$

for $p \in A$, $c \in [0,1] \cap \mathbb{Q}$, formulas $\alpha, \alpha_1, \ldots, \alpha_n$ and a connective $f \in F_n$.

Definition 5.7

Consider $v \in \Omega$ and $S \subseteq \Omega$. The structure (v, S) is a *FMBNF model* of a theory T in FMBNF iff

1. for each $\alpha \in T$: $\|\alpha\|_{(v,S,S)} = 1$ and
2. there is no structure (v', S') such that $S \subset S'$ and $\|\alpha\|_{(v',S',S)} = 1$ for all $\alpha \in T$.

For a mapping $v \in \Omega$, we define $\mathrm{Mod}(v)$ as the set of all mappings $w \in \Omega$ such that $v(a) \leq w(a)$ for all $a \in A$, i.e. $v \leq w$. With this definition, we have the following property.

Lemma 5.4

Consider $v, v' \in \Omega$. Then $v \leq v'$ iff $\mathrm{Mod}(v') \subseteq \mathrm{Mod}(v)$.

Proof. First suppose that $v \leq v'$, we show that $\mathrm{Mod}(v') \subseteq \mathrm{Mod}(v)$. Let $w' \in \mathrm{Mod}(v')$, then $v \leq v' \leq w'$, i.e. $w' \in \mathrm{Mod}(v)$. Now suppose that $\mathrm{Mod}(v') \subseteq \mathrm{Mod}(v)$, we show that $v \leq v'$. This follows easily since it holds that $v' \leq v'$ and thus $v' \in \mathrm{Mod}(v') \subseteq \mathrm{Mod}(v)$, i.e. $v \leq v'$.

□

One can now easily see that the following lemma must hold.

Lemma 5.5

Consider $v, v' \in \Omega$. Then $v < v'$ iff $\mathrm{Mod}(v') \subset \mathrm{Mod}(v)$.

Proof. First suppose that $v < v'$, we show that $\mathrm{Mod}(v') \subset \mathrm{Mod}(v)$. By Lemma 5.4 it follows that $\mathrm{Mod}(v') \subseteq \mathrm{Mod}(v)$. Now suppose that $\mathrm{Mod}(v') = \mathrm{Mod}(v)$, then in particular we have $\mathrm{Mod}(v) \subseteq \mathrm{Mod}(v')$ and by Lemma 5.4 it follows that $v' \leq v$, a contradiction. Next, suppose that $\mathrm{Mod}(v') \subset \mathrm{Mod}(v)$, we show that $v < v'$. By Lemma 5.4 it follows that $v \leq v'$. Now suppose that $v = v'$, then in particular we have $v' \leq v$ and by Lemma 5.4 it follows that $\mathrm{Mod}(v) \subseteq \mathrm{Mod}(v')$, a contradiction. □

Remark 5.6

For fuzzy interpretations $I \in \mathcal{F}(\mathcal{B}_P)$ one can also define $\mathrm{Mod}(I)$ since $\mathcal{B}_P \subseteq A$. Indeed, let $\mathrm{Mod}(I) = \mathrm{Mod}(v_I)$ with $v_I \in \Omega$ defined as follows: $v_I(x) = I(x)$ if $x \in \mathcal{B}_P$ and $v_I(x) = 0$ if $x \notin \mathcal{B}_P$. Lemma 5.5 then also holds for $I \in \mathcal{F}(\mathcal{B}_P)$.

5.5.3 Embedding regular FASP in FMBNF

First, we investigate the relationship between FMBNF and regular FASP. A regular FASP program is translated to a theory in FMBNF as follows. Consider a regular FASP program

P. By Lemma 4.1 we may assume without loss of generality that P does not contain strong negation. For each rule

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \text{not}_1\, c_1, \ldots, \text{not}_k\, c_k),$$

the formula $\mu(r)$

$$f(\text{B}b_1, \ldots, \text{B}b_m, \text{not}_1\, c_1, \ldots, \text{not}_k\, c_k) \rightarrow g(\text{B}a_1, \ldots, \text{B}a_n)$$

is added to the theory $\mu(P)$ in FMBNF.

To prove the correspondence between the answer sets of P and the models of $\mu(P)$, we define for a regular FASP program P and a fuzzy interpretation $M \in \mathcal{F}(\mathcal{B}_P)$, a set of evaluations $\Pi(P, M) \subseteq \Omega$.

Definition 5.8

Consider a regular FASP program P without strong negation and $M \in \mathcal{F}(\mathcal{B}_P)$. Define

$$\pi_P^M = \{w \in \Omega \mid \forall \alpha \in \mu(P), \forall v \in \Omega : \|\alpha\|_{(v, \text{Mod}(w), \text{Mod}(M))} = 1\}$$

and

$$\Pi(P, M) = \{w \in \Omega \mid w \text{ is a minimal element of } \pi_P^M\}.$$

Note that π_P^M may not contain any minimal elements, and may even be empty. Also, note that in Definition 5.8, since all atoms in $\mu(P)$ are preceded by a modal operator, the choice of the evaluation v is irrelevant.

Lemma 5.6

Consider a regular FASP program P without strong negation and without negation-as-failure and $M \in \mathcal{F}(\mathcal{B}_P)$. Suppose $I \in \mathcal{F}(\mathcal{B}_P)$, then $v_I \in \Pi(P, M)$ iff I is an answer set of P.

Proof. We need to prove that $\{I \in \mathcal{F}(\mathcal{B}_P) \mid v_I \in \Pi(P, M)\}$ is exactly the set of the minimal elements of $\{I \in \mathcal{F}(\mathcal{B}_P) \mid I \text{ fuzzy model of } P\}$. It is thus sufficient to prove that $v_I \in \pi_P^M$ iff I is a fuzzy model of P.

A rule r in P is of the form

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m).$$

Note that for $I \in \mathcal{F}(\mathcal{B}_P)$ we have $I(a) = \inf_{K \in \mathrm{Mod}(I)} K(a)$ for each $a \in \mathcal{B}_P$. We then obtain

$$
\begin{aligned}
& I \text{ fuzzy model of } P \\
\Leftrightarrow\ & \forall r \in P : \mathtt{f}(I(b_1), \ldots, I(b_m)) \leq \mathtt{g}(I(a_1), \ldots, I(a_n)) \\
\Leftrightarrow\ & \forall r \in P : \mathtt{f}(\inf_{K \in \mathrm{Mod}(I)} K(b_1), \ldots, \inf_{K \in \mathrm{Mod}(I)} K(b_m)) \\
& \leq \mathtt{g}(\inf_{K \in \mathrm{Mod}(I)} K(a_1), \ldots, \inf_{K \in \mathrm{Mod}(I)} K(a_n)) \\
\Leftrightarrow\ & \forall r \in P, \forall v \in \Omega : \\
& \mathtt{f}(\|\mathrm{B}b_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, \|\mathrm{B}b_m\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}) \\
& \leq \mathtt{g}(\|\mathrm{B}a_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, \|\mathrm{B}a_n\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}) \\
\Leftrightarrow\ & \forall r \in P, \forall v \in \Omega : \\
& \|f(\mathrm{B}b_1, \ldots, \mathrm{B}b_m) \to g(\mathrm{B}a_1, \ldots, \mathrm{B}a_n)\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1 \\
\Leftrightarrow\ & \forall \alpha \in \mu(P), \forall v \in \Omega : \|\alpha\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1 \\
\Leftrightarrow\ & v_I \in \pi_P^M
\end{aligned}
$$

□

For a regular FASP program without strong negation we have the following result:

Lemma 5.7

Consider a regular FASP program without strong negation P and $I, M \in \mathcal{F}(\mathcal{B}_P)$. Then

$$v_I \in \Pi(P^M, M) \text{ iff } v_I \in \Pi(P, M).$$

Proof. By Definition 5.8, it is sufficient to check that for $I \in \mathcal{B}_{\mathcal{P}}$:

$$\forall \alpha \in \mu(P), \forall v \in \Omega : \|\alpha\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1$$

is equivalent with

$$\forall \alpha \in \mu(P^M), \forall v \in \Omega : \|\alpha\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1$$

Hence for a rule in P

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_\mathrm{k}\, c_k)$$

it is sufficient to prove that for each $v \in \Omega$ we have

$$\|\mu(r)\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1 \Leftrightarrow \|\mu(r^M)\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = 1.$$

Or, in other words, it has to be shown that

$$\begin{aligned}&\mathtt{f}(\|\mathrm{B}b_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, \|\mathrm{B}b_m\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))},\\ &\|\operatorname{not_1} c_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, \|\operatorname{not_k} c_k\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))})\\ &\leq \mathtt{g}(\|\mathrm{B}a_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots \|\mathrm{B}a_n\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))})\end{aligned}$$

iff

$$\begin{aligned}&\mathtt{f}(\|\mathrm{B}b_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, \|\mathrm{B}b_m\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))},\\ &[\mathrm{B}(\overline{[\operatorname{not_1} c_1]_M})]_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots, [\mathrm{B}(\overline{[\operatorname{not_k} c_k]_M})]_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))})\\ &\leq \mathtt{g}(\|\mathrm{B}a_1\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}, \ldots \|\mathrm{B}a_n\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))})\end{aligned}$$

Thus we need to show that for each atom c_j ($j \in \{1, \ldots, k\}$) we have

$$\|\operatorname{not_j} c_j\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} = \|\mathrm{B}(\overline{[\operatorname{not_j} c_j]_M})\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}.$$

But this follows easily by the definitions and since $\overline{[\operatorname{not_j} c_j]_M}$ is a truth constant:

$$\begin{aligned}\|\mathrm{B}(\overline{[\operatorname{not_j} c_j]_M})\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))} &= \inf_{w\in\mathrm{Mod}(I)} \|\overline{[\operatorname{not_j} c_j]_M}\|_{(w,\mathrm{Mod}(I),\mathrm{Mod}(M))}\\ &= [\operatorname{not_j} c_j]_M\\ &= N_j(M(c_j))\\ &= N_j\left(\inf_{w\in\mathrm{Mod}(M)} w(c_j)\right)\\ &= \|\operatorname{not_j} c_j\|_{(v,\mathrm{Mod}(I),\mathrm{Mod}(M))}\end{aligned}$$

□

Next, we will define a notion of equivalence for subsets of Ω.

Definition 5.9

Consider a regular FASP program without strong negation P and $S_1, S_2 \subseteq \Omega$. We say that S_1 and S_2 are *inf-equivalent* w.r.t. P if $\forall x \in \mathcal{B}_P$

$$\inf_{v\in S_1} v(x) = \inf_{v\in S_2} v(x).$$

For inf-equivalent sets we have the following result.

Lemma 5.8

Consider a regular FASP program without strong negation P and $S_1, S_2 \subseteq \Omega$ inf-equivalent w.r.t. P. Then for each $r \in P$ we have

$$\|\mu(r)\|_{(v,S_1,S)} = \|\mu(r)\|_{(v,S_2,S)},$$

with $v \in \Omega$ and $S \subseteq \Omega$ arbitrary.

Proof. A rule $r \in P$ is of the form

$$g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not_1}\, c_1, \ldots, \mathrm{not_k}\, c_k)$$

and the corresponding formula $\mu(r)$ is

$$f(\mathrm{B}b_1, \ldots, \mathrm{B}b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_k\, c_k) \rightarrow g(\mathrm{B}a_1, \ldots, \mathrm{B}a_n).$$

It follows that

$$\begin{aligned}
& \|\mu(r)\|_{(v,S_1,S)} \\
= \quad & I_T\left(\mathtt{f}(\|\mathrm{B}b_1\|_{(v,S_1,S)}, \ldots, \|\mathrm{B}b_m\|_{(v,S_1,S)}, \|\,\mathrm{not_1}\, c_1\|_{(v,S_1,S)}, \ldots, \|\,\mathrm{not_k}\, c_k\|_{(v,S_1,S)}),\right. \\
& \left.\mathtt{g}(\|\mathrm{B}a_1\|_{(v,S_1,S)}, \ldots, \|\mathrm{B}a_n\|_{(v,S_1,S)})\right) \\
= \quad & I_T\Bigg(\mathtt{f}(\inf_{w \in S_1} w(b_1), \ldots, \inf_{w \in S_1} w(b_m), \sup_{w \in S} N_1(w(c_1)), \ldots, \sup_{w \in S} N_k(w(c_k))), \\
& \mathtt{g}(\inf_{w \in S_1} w(a_1), \ldots, \inf_{w \in S_1} w(a_n))\Bigg) \\
= \quad & I_T\Bigg(\mathtt{f}(\inf_{w \in S_2} w(b_1), \ldots, \inf_{w \in S_2} w(b_m), \sup_{w \in S} N_1(w(c_1)), \ldots, \sup_{w \in S} N_k(w(c_k))), \\
& \mathtt{g}(\inf_{w \in S_2} w(a_1), \ldots, \inf_{w \in S_2} w(a_n))\Bigg) \\
= \quad & I_T\left(\mathtt{f}(\|\mathrm{B}b_1\|_{(v,S_2,S)}, \ldots, \|\mathrm{B}b_m\|_{(v,S_2,S)}, \|\,\mathrm{not_1}\, c_1\|_{(v,S_2,S)}, \ldots, \|\,\mathrm{not_k}\, c_k\|_{(v,S_2,S)}),\right. \\
& \left.\mathtt{g}(\|\mathrm{B}a_1\|_{(v,S_2,S)}, \ldots, \|\mathrm{B}a_n\|_{(v,S_2,S)})\right) \\
= \quad & \|\mu(r)\|_{(v,S_2,S)}
\end{aligned}$$

□

We need one more lemma.

Lemma 5.9

Consider a regular FASP program without strong negation P, $v \in \Omega$ and $S \subseteq \Omega$. Then (v, S) is a FMBNF model of $\mu(P)$ iff there exists $w \in \Pi(P, M)$ such that $S = \mathrm{Mod}(w)$.

Proof. ($\Rightarrow$) First assume that (v, S) is a FMBNF model of $\mu(P)$. By Definition 5.7, this means that

(a) for each $\alpha \in \mu(P)$: $\|\alpha\|_{(v,S,S)} = 1$ and

(b) there is no structure (v', S') such that $S \subset S'$ and $\|\alpha\|_{(v',S',S)} = 1$ for all $\alpha \in \mu(P)$.

Define $w \in \Omega$ as follows:

$$w : A \to [0, 1] : x \mapsto \inf_{z \in S} z(x).$$

For $z \in S$ we obtain $w(x) \leq z(x)$ for all $x \in A$, hence $S \subseteq \mathrm{Mod}(w)$. Note that S and $\mathrm{Mod}(w)$ are inf-equivalent w.r.t P:

$$\inf_{z \in S} z(x) = w(x) = \inf_{z \in \mathrm{Mod}(w)} z(x)$$

for all $x \in \mathcal{B}_P$. By Lemma 5.8, it follows that

$$\|\alpha\|_{(v,\mathrm{Mod}(w),S)} = \|\alpha\|_{(v,S,S)} = 1$$

for all $\alpha \in \mu(P)$. Now suppose that $S \subset \mathrm{Mod}(w)$, then we have a contradiction since S is maximal under all S' such that $\|\alpha\|_{(v,S',S)} = 1$ for all $\alpha \in \mu(P)$. Thus $S = \mathrm{Mod}(w)$ and $\|\alpha\|_{(v,\mathrm{Mod}(w),\mathrm{Mod}(w))} = 1$ for all $\alpha \in \mu(P)$. To prove that $w \in \Pi(P, M)$ it remains to be shown that w is minimal under all $w' \in \Omega$ such that for all $z \in \Omega$ and for all $\alpha \in \mu(P)$ it holds that $\|\alpha\|_{(z,\mathrm{Mod}(w'),\mathrm{Mod}(w))} = 1$. Suppose this is not the case and there exists an w' such that $w' < w$ and for all $z \in \Omega$ and for all $\alpha \in \mu(P)$ it holds that $\|\alpha\|_{(z,\mathrm{Mod}(w'),\mathrm{Mod}(w))} = 1$. By Lemma 5.5, it follows that $S = \mathrm{Mod}(w) \subset \mathrm{Mod}(w')$. This is in contradiction with the maximality of S.

($\Leftarrow$) Now assume $S = \mathrm{Mod}(w)$ with $w \in \Pi(P, M)$. By Definition 5.8, this means that w is minimal under all $w' \in \Omega$ such that for all $\alpha \in \mu(P)$ and for all $z \in \Omega$ it holds that $\|\alpha\|_{(z,\mathrm{Mod}(w'),\mathrm{Mod}(w))} = 1$. Thus, we already know that $\|\alpha\|_{(v,S,S)} =$

$\|\alpha\|_{(v,\mathrm{Mod}(w),\mathrm{Mod}(w))} = 1$ for all $\alpha \in \mu(P)$. To show that (v, S) is a FMBNF model of $\mu(P)$, it remains to be shown that there is no structure (v', S') such that $\mathrm{Mod}(w) \subset S'$ and $\|\alpha\|_{(v',S',\mathrm{Mod}(w))} = 1$ for all $\alpha \in \mu(P)$. Suppose there exists such a structure (v', S'). Define $w' \in \Omega$ as follows:

$$w' : A \to [0,1] : x \mapsto \inf_{v \in S'} v(x).$$

Similar as in the first part of the proof we have that $S' \subseteq \mathrm{Mod}(w')$ and that S' and $\mathrm{Mod}(w')$ are inf-equivalent w.r.t. P. By Lemma 5.8 we then have that

$$\|\alpha\|_{(v',\mathrm{Mod}(w'),\mathrm{Mod}(w))} = \|\alpha\|_{(v',S',\mathrm{Mod}(w))} = 1$$

for all $\alpha \in \mu(P)$. Since $\mu(P)$ only contains atoms preceded by a modal operator, it also follows that for all $z \in \Omega$ and for all $\alpha \in \mu(P)$ we have $\|\alpha\|_{(z,\mathrm{Mod}(w'),\mathrm{Mod}(w))} = 1$. But this contradicts the minimality of w. Indeed, since $\mathrm{Mod}(w) \subset S' \subseteq \mathrm{Mod}(w')$, it follows by Lemma 5.5 that $w' < w$.

□

Combining Lemmas 5.6-5.9, we get the following generalisation of Theorem 5.2.

Theorem 5.3

Consider a regular FASP program without strong negation P and $M \in \mathcal{F}(\mathcal{B}_P)$. M is an answer set of P iff $(v, \mathrm{Mod}(M))$ is a FMBNF model of $\mu(P)$ with $v \in \Omega$ arbitrary.

Proof.

M is an answer set of P

$\Leftrightarrow M$ is an answer set of P^M (definition answer set)

$\Leftrightarrow v_M \in \Pi(P^M, M)$ (Lemma 5.6)

$\Leftrightarrow v_M \in \Pi(P, M)$ (Lemma 5.7)

$\Leftrightarrow (v, \mathrm{Mod}(v_M))$ is an FMBNF model of $\mu(P)$ (Lemma 5.9)

$\Leftrightarrow (v, \mathrm{Mod}(M))$ is an FMBNF model of $\mu(P)$ (Remark 5.6)

□

5.5.4 Embedding regular FASP in fuzzy autoepistemic logic

Using the result from Theorem 5.3, we will now generalise Theorem 3.2: fuzzy autoepistemic logic generalises regular FASP. The translation is defined as follows. For each rule

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_\mathrm{k}\, c_k)$$

in a regular FASP program P we add the formula $\sigma(r)$:

$$f\left(b_1 \wedge \mathrm{B}b_1, \ldots, b_m \wedge \mathrm{B}b_m, \sim_1\mathrm{B}c_1, \ldots, \sim_k\mathrm{B}c_k\right) \to g\left(a_1 \wedge \mathrm{B}a_1), \ldots, (a_n \wedge \mathrm{B}a_n)\right),$$

to the fuzzy autoepistemic theory $\sigma(P)$ where $\sim_j$ is the negation that is interpreted by the same negator N_j as for not_j.

First we provide some useful lemmas.

Lemma 5.10

Consider a regular FASP program without strong negation P and $S \subseteq \Omega$. Then for $r \in P$, $v, w \in \Omega$ and $S' = S \cup \{w\}$ we have

$$\|\mu(r)\|_{(v,S',S)} = \|\sigma(r)\|_{(w,S)}.$$

Proof. Consider a rule $r \in P$

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_\mathrm{k}\, c_k).$$

Since $S' = S \cup \{w\}$ we have that $\inf_{z \in S'} z(x) = \min(w(x), \inf_{z\in S} z(x))$ for each $x \in A$. It follows that

$$\begin{aligned}
\|\mu(r)\|_{(v,S',S)} &= \|f(\mathrm{B}b_1, \ldots, \mathrm{B}b_m, \mathrm{not}_1\, c_1, \ldots, \mathrm{not}_\mathrm{k}\, c_k) \to g(\mathrm{B}a_1, \ldots, \mathrm{B}a_n)\|_{(v,S',S)} \\
&= I_T\left(\mathtt{f}\left(\inf_{z\in S'} z(b_1), \ldots, \inf_{z\in S'} z(b_m), N_1(\inf_{z\in S} z(c_1)), \ldots, N_k(\inf_{z\in S} z(c_k))\right),\right. \\
&\qquad \left.\mathtt{g}\left(\inf_{z\in S'} z(a_1), \ldots, \inf_{z\in S'} z(a_n)\right)\right) \\
&= I_T\left(\mathtt{f}\left(\min\left(w(b_1), \inf_{z\in S} z(b_1)\right), \ldots, \min\left(w(b_m), \inf_{z\in S} z(b_m)\right)\right.\right. \\
&\qquad \left. N_1(\inf_{z\in S} z(c_1)), \ldots, N_k(\inf_{z\in S} z(c_k))\right), \\
&\qquad \left.\mathtt{g}\left(\min\left(w(a_1), \inf_{z\in S} z(a_1)\right), \ldots, \min\left(w(a_n), \inf_{z\in S} z(a_n)\right)\right)\right)
\end{aligned}$$

$$
\begin{aligned}
&= \|f((b_1 \wedge \mathrm{B}b_1), \ldots, (b_m \wedge \mathrm{B}b_m), \sim_1 \mathrm{B}c_1, \ldots, \sim_k \mathrm{B}c_k) \\
&\quad \rightarrow g((a \wedge \mathrm{B}a_1), \ldots, (a_n \wedge \mathrm{B}a_n))\|_{(w,S)} \\
&= \|\sigma(r)\|_{(w,S)}
\end{aligned}
$$

□

Lemma 5.11

Consider a regular FASP program without strong negation P, $v' \in \Omega$ and $w, w' \in \Omega$ such that $w' \leq w$. Then for $r \in P$ and for $S, S' \subseteq \Omega$ such that S and $\mathrm{Mod}(w)$ are inf-equivalent and S' and $\mathrm{Mod}(w')$ are inf-equivalent w.r.t. P, we have that

$$\|\sigma(r)\|_{(w',S)} = \|\mu(r)\|_{(w',S',S)}.$$

Proof. Consider a rule $r \in P$:

$$r : g(a_1, \ldots, a_n) \leftarrow f(b_1, \ldots, b_m, \mathrm{not_1}\, c_1, \ldots, \mathrm{not_k}\, c_k).$$

Since S and $\mathrm{Mod}(w)$ are inf-equivalent w.r.t. P (Definition 5.9), we have

$$w(a) = \inf_{z \in \mathrm{Mod}(w)} z(a) = \inf_{z \in S} z(a)$$

for each $a \in \mathcal{B}_P$. By the inf-equivalence of S' and $\mathrm{Mod}(w')$, we have

$$w'(a) = \inf_{z \in \mathrm{Mod}(w')} z(a) = \inf_{z \in S'} z(a)$$

for each $a \in \mathcal{B}_P$. It follows that

$$
\begin{aligned}
&\|\sigma(r)\|_{(w',S)} \\
=&\|f((b_1 \wedge \mathrm{B}b_1), \ldots, (b_m \wedge \mathrm{B}b_m), \sim_1 \mathrm{B}c_1, \ldots, \sim_k \mathrm{B}c_k) \\
&\rightarrow g((a_1 \wedge \mathrm{B}a_1), \ldots, (a_n \wedge \mathrm{B}a_n))\|_{(w',S)} \\
=&I_T\Bigg(\mathtt{f}\bigg(\min\Big(w'(b_1), \inf_{z \in S} z(b_1)\Big), \ldots, \min\Big(w'(b_m), \inf_{z \in S} z(b_m)\Big), \\
&N_1(\inf_{z \in S} z(c_1)), \ldots, N_k(\inf_{z \in S} z(c_k))\bigg), \\
&\mathtt{g}\bigg(\min\Big(w'(a_1), \inf_{z \in S} z(a_1)\Big), \ldots, \min\Big(w'(a_n), \inf_{z \in S} z(a_n)\Big)\bigg)\Bigg)
\end{aligned}
$$

$$
\begin{aligned}
=& I_T\Bigg(\mathtt{f}\Big(\min\big(w'(b_1), w(b_1)\big), \ldots, \min\big(w'(b_m), w(b_m)\big),\\
& N_1(\inf_{z\in S} z(c_1)), \ldots, N_k(\inf_{z\in S} z(c_k))\Big),\\
& \mathtt{g}\Big(\min\big(w'(a_1), w(a_1)\big), \ldots, \min\big(w'(a_n), w(a_n)\big)\Big)\Bigg)\\
=& I_T\Bigg(\mathtt{f}\Big(w'(b_1), \ldots, w'(b_m), N_1(\inf_{z\in S} z(c_1)), \ldots, N_k(\inf_{z\in S} z(c_k))\Big),\\
& \mathtt{g}\Big(w'(a_1), \ldots, w'(a_n)\Big)\Bigg)\\
=& I_T\Bigg(\mathtt{f}\Big(\inf_{z\in S'} z(b_1), \ldots, \inf_{z\in S'} z(b_m), N_1(\inf_{z\in S} z(c_1)), \ldots, N_k(\inf_{z\in S} z(c_k))\Big),\\
& \mathtt{g}\Big(\inf_{z\in S'} z(a_1), \ldots, \inf_{z\in S'} z(a_n)\Big)\Bigg)\\
=& \|f(\mathrm{B}b_1, \ldots, \mathrm{B}b_m, \mathrm{not_1}\, c_1, \ldots, \mathrm{not_k}\, c_k) \to g(\mathrm{B}a_1, \ldots, \mathrm{B}a_n)\|_{(v',S',S)}\\
=& \|\mu(r)\|_{(v',S',S)}
\end{aligned}
$$

□

We will now use Lemmas 5.10 and 5.11 to prove the main theorem:

Theorem 5.4

Consider a regular FASP program without strong negation P and $M \in \mathcal{F}(\mathcal{B}_P)$. M is an answer set of P iff $\mathrm{Mod}(M)$ is a fuzzy possible world autoepistemic model of $\sigma(P)$.

Proof. ($\Rightarrow$) First suppose that M is an answer set of P. By Theorem 5.3, it follows that (v, S) with $S = \mathrm{Mod}(M)$ and $v \in \Omega$ arbitrary is a FMBNF model of $\mu(P)$. By Definition 5.7 this means that

(a) for each $\alpha \in \mu(P)$: $\|\alpha\|_{(v,S,S)} = 1$ and

(b) there is no structure (v', S') such that $S \subset S'$ and $\|\alpha\|_{(v',S',S)} = 1$ for all $\alpha \in \mu(P)$.

We need to prove that S is a fuzzy possible world autoepistemic model of $\sigma(P)$, or by Definition 5.4 that

$$S = \left\{z \in \Omega \mid \forall \varphi \in \sigma(P) : \|\varphi\|_{(z,S)} = 1\right\}.$$

Let $w \in S$, thus $S \cup \{w\} = S$. By Lemma 5.10 it follows that $\|\sigma(r)\|_{(w,S)} = \|\mu(r)\|_{(v,S,S)} = 1$ for all $r \in P$. Hence

$$w \in \left\{z \in \Omega \mid \forall \varphi \in \sigma(P) : \|\varphi\|_{(z,S)} = 1\right\}$$

as soon as $w \in S$. Conversely, consider $w \in \Omega$ such that $\|\varphi\|_{(w,S)} = 1$ for every $\varphi \in \sigma(P)$. If $w \notin S$ define $S' = S \cup \{w\}$. By Lemma 5.10 it follows that $\|\mu(r)\|_{(w,S',S)} = \|\sigma(r)\|_{(w,S)} = 1$ for each $r \in P$. This contradicts the maximality of S.

($\Leftarrow$) Now suppose that $S = \mathrm{Mod}(M)$ is a fuzzy possible world autoepistemic model of $\sigma(P)$. By Theorem 5.3, it is sufficient to show that (v, S) with $v \in \Omega$ arbitrary is a FMBNF model of $\mu(P)$. By Definition 5.7 we need to show that

(a) for each $\alpha \in \mu(P)$: $\|\alpha\|_{(v,S,S)} = 1$ and

(b) there is no structure (v', S') such that $S \subset S'$ and $\|\alpha\|_{(v',S',S)} = 1$ for all $\alpha \in \mu(P)$.

Let $w \in S$, thus $S \cup \{w\} = S$. By Lemma 5.10, it follows that $\|\mu(r)\|_{(v,S,S)} = \|\sigma(r)\|_{(w,S)}$ for each $r \in P$. Since S is a fuzzy possible world autoepistemic model of $\sigma(P)$ and thus

$$S = \left\{z \in \Omega \mid \forall \varphi \in \sigma(P) : \|\varphi\|_{(z,S)} = 1\right\}$$

we have $\|\mu(r)\|_{(v,S,S)} = \|\sigma(r)\|_{(w,S)} = 1$ for all $\mu(r) \in \mu(P)$. Now suppose there is a structure (v', S') such that $S \subset S'$ and $\|\alpha\|_{(v',S',S)} = 1$ for all $\alpha \in \mu(P)$. Define $w' \in \Omega$ as follows:

$$w' : A \to [0,1] : a \mapsto \inf_{z \in S'} z(a).$$

For $z' \in S'$ it holds that $w'(a) = \inf_{z \in S'} z(a) \leq z'(a)$ for all $a \in A$, hence $S' \subseteq \mathrm{Mod}(w')$ and thus $\mathrm{Mod}(w) = S \subset S' \subseteq \mathrm{Mod}(w')$. By Lemma 5.5 it follows that $w' < w$. Note that S' and $\mathrm{Mod}(w')$ are inf-equivalent w.r.t. P by definition of w'. By Lemma 5.11 it then follows that $\|\sigma(r)\|_{(w',S)} = \|\mu(r)\|_{(v',S',S)} = 1$ for all $\sigma(r) \in \sigma(P)$ which implies that $w' \in S = \mathrm{Mod}(w)$ since S is a fuzzy possible world autoepistemic model of $\sigma(P)$. Hence $w \leq w'$, a contradiction.

□

Example 5.4

Reconsider the regular FASP program P from Remark 5.5.

$$a \oplus b \leftarrow \overline{1}$$

with a and b atoms and $\oplus$ and $\leftarrow$ connectives from Łukasiewicz logic. The corresponding fuzzy autoepistemic theory is

$$\sigma(P) = \left\{\overline{1} \rightarrow (a \wedge \mathrm{B}a) \oplus (b \wedge \mathrm{B}b)\right\}.$$

We will use Theorem 5.4 to calculate all answer sets of P. Suppose $I \in \mathcal{F}(\mathcal{B}_P)$ is an arbitrary answer set of P. Rename $I(a) = x$ and $I(b) = y$. By Theorem 5.4, it then follows that

$$S = \mathrm{Mod}(I) = \{w \in \Omega \mid w(a) \geq x, w(b) \geq y\}$$

is a fuzzy possible world autoepistemic model of $\sigma(P)$. Thus, what we want to know is for which values x and y we have that S is a fuzzy possible world autoepistemic model, or in other words for which values x and y does it hold that

$$S = \left\{w \in \Omega \mid 1 \leq \min(w(a), \inf_{z \in S} w(a)) + \min(w(b), \inf_{z \in S} w(b))\right\}.$$

For an arbitrary $w \in S$ we have

$$\begin{aligned}\min(w(a), \inf_{z \in S} z(a)) + \min(w(b), \inf_{z \in S} z(b)) &= \inf_{z \in S} z(a) + \inf_{z \in S} z(b) \\ &= x + y\end{aligned}$$

Hence $x + y \geq 1$ is a necessary condition. Now suppose that $x + y > 1$. We show that

$$\left\{w \in \Omega \mid 1 \leq \min(w(a), \inf_{z \in S} z(a)) + \min(w(b), \inf_{z \in S} z(b))\right\} \not\subseteq S.$$

Indeed, consider $w \in \Omega$ defined by $w(a) = x$ and $w(b) = 1 - x$. If $x + y > 1$ then

$$\begin{aligned}\min(w(a), \inf_{z \in S} z(a)) + \min(w(b), \inf_{z \in S} z(b)) &= \min(x, x) + \min(1 - x, y) \\ &= x + (1 - x) \\ &= 1\end{aligned}$$

Thus w is an element of

$$\left\{w \in \Omega \mid 1 \leq \min(w(a), \inf_{z \in S} z(a)) + \min(w(b), \inf_{z \in S} z(b))\right\}.$$

If it would hold that $w \in S$, then $1 - x = w(b) \geq y$ or $x + y \leq 1$, a contradiction. Hence $w \notin S$.

5.6 Conclusion

In this chapter we have introduced a fuzzy version of autoepistemic logic, which can be used to reason about one's (lack of) beliefs about the degrees to which properties are satisfied. We have shown that important properties of classical autoepistemic logic are preserved and that the relation between answer set programming and autoepistemic logic remains valid when generalising to the many-valued case. Moreover, we have presented two different but equivalent characterisations of answer sets in fuzzy autoepistemic logic and in a fuzzy logic of minimal belief and negation-as-failure. These results lead to a better comprehension of how to interpret fuzzy answer sets. Since the language of (fuzzy) autoepistemic logic is much more expressive than the theories we need to represent the (fuzzy) answer set programs, this could serve as a useful basis for defining or comparing extensions to the basic language of ASP since the computational complexity does not increase when moving from ASP to autoepistemic logic. This might open doors to define extensions for FASP.

In future work, it would be interesting to see whether the implementation of classical autoepistemic logic by using Quantified Boolean Formulas [Egly et al. 2000] can be extended to fuzzy logics using multi-level linear programming. If this is indeed the case, it could be used as a basis to implement fuzzy autoepistemic logic reasoners, as well as fuzzy answer set programming solvers.

6 Relating fuzzy autoepistemic logic and fuzzy modal logics of belief

6.1 Introduction

In this chapter we will discuss relationships between fuzzy autoepistemic logic and fuzzy modal logics, generalising well-known links beween autoepistemic logic and several classical modal logic systems. In particular we will generalise Levesque's logic of only knowing [Levesque 1990] to the many-valued case, and show that the correspondence with autoepistemic logic is preserved under this generalisation. Moreover we provide a sound and complete axiomatisation for this many-valued logic of only knowing.

Autoepistemic logic, discussed in Section 2.1, has been one of the main formalisms for nonmonotonic reasoning. It extends propositional logic by offering the ability to reason about an agent's (lack of) beliefs. Recall that, given a set of initial premises, the (closed) set of beliefs an agent should adopt is given by the so-called stable expansions. In [Levesque 1990], autoepistemic logic is extended to expressions of the form "φ is all that is believed", i.e. there are no other relevant beliefs, but φ. To this end, a second modal operator O is used where $\mathrm{O}\varphi$ has to be read as "φ is *all* that is believed" or "*only* φ is believed". In [Levesque 1990] it is then shown that stable expansions correspond to a particular type of valid sentences in this logic, see also Theorem 2.1. Finally, a sound and complete axiomatisation based on classical K45 modal logic is provided by pointing out that $\mathrm{O}\varphi$ can be rewritten as $\mathrm{B}\varphi \wedge \mathrm{N}(\sim\varphi)$ where the modal operators N and B are both K45 operators. In particular, $\mathrm{B}\varphi$ corresponds to "φ is believed" and $\mathrm{N}(\sim\varphi)$ to "at most $\sim\varphi$ is believed to be false". Hence $\mathrm{O}\varphi$ corresponds to at least and at most φ is believed, i.e. "exactly φ is believed". By Theorem 3.1 it follows that the answer sets of answer set programs also correspond to valid sentences in logic of only knowing.

In Chapter 5 we introduced a generalisation of autoepistemic logic using fuzzy logic and we showed that the relation between the answer sets of a fuzzy answer set program and the stable fuzzy expansions of a corresponding fuzzy autoepistemic logic theory remains valid. In this chapter, we introduce generalisations of the main classical propositional modal logics of belief (K45, KD45, S5) based on finitely-valued Łukasiewicz logic with truth constants in order to model the notion of belief on fuzzy propositions, in the sense of admitting partial degrees of truth between 0 (fully false) and 1 (fully true). Similar as in the classical case, we show soundness and completeness w.r.t. appropriate Kripke style semantics. We will also show NP-completeness for two variants of the satisfiability problem. Note in particular that generalising to the many-valued case does not imply an increase in computational complexity.

Many-valued modal logics have appeared in the literature under different forms and contexts. In [Fitting 1992a, Fitting 1992b], a modal logic with truth values in finite Heyting algebras is introduced. These modal systems are then used for dealing with opinions of experts with a dominance relation among them. Other papers mainly offer theoretical frameworks. For instance, in the last years there has been some work on fuzzy modal logics with generalised Kripke semantics, see e.g. [Bou et al. 2011b]. In particular, [Bou et al. 2011b] systematically investigates many-valued modal logics over a residuated lattice, dealing with accessibility relations that can take values in this lattice as well as relations taking only values $\top$ and $\bot$ in the lattice. In this chapter we will focus on modal systems based on a finite set of linearly ordered truth values with Łukasiewic logic semantics for connectives which generate the class of finite MV-algebras [Mundici 1987]. These systems represent a good compromise between expressive power and nice logical properties. The infinitely-valued case offers some problems, see e.g. [Hansoul and Teheux 2013] where extending infinitely-valued Łukasiewicz logic with a modality results in an infinitary deduction rule. Another example is the case of S5 with total accessibility relations [Hájek 2010]. In this paper the fact that formulas are in correspondence with formulas of the monadic fuzzy predicate calculus is used to propose a sound and complete axiomatisation by interpreting the modality as "$\forall$". However, for KD45 and K45 this trick fails. On the other hand, a closely related work is [Maruyama 2011] which considers modal logics for belief based on a finitely-valued Heyting algebra of truth values. Recall that in a Heyting algebra, the truth values are not necessarily linearly ordered The formalisation is very similar as in our work, but he also deals with common belief. Here we rather focus on providing a formal basis for the fuzzy generalisation of autoepistemic logic as introduced in Chapter 5.

Then we show how fuzzy autoepistemic logic can be characterised using the possible worlds semantics corresponding to these many-valued modal logics. Finally, we also consider the extension of many-valued autoepistemic logic with an "only knowing" operator O and show that the relationship between stable expansions, belief sets and "only knowing" operators [Levesque 1990] naturally extends to our framework. As in the classical case we show that formulas of the form $\mathrm{O}\varphi$ can be rewritten as $\mathrm{B}\varphi \wedge \mathrm{N}(\sim\varphi)$ where B and N are now many-valued K45 structures. We provide a sound and complete axiomatization for this finitely-valued Łukasiewicz logic of only knowing and show that stable fuzzy expansions correspond to valid sentences in this logic. In particular this implies that the answer sets of a fuzzy answer set program correspond to valid sentences in this logic.

A summarising diagram of embeddings and generalisations with references can be found in Figure 6.1.

This chapter is structured as follows. After this introduction, in Section 6.2 we define proper generalisations of the classical modal systems K45, KD45 and S5 and prove soundness and completeness with respect to appropriate Kripke-style semantics, and we provide a (possibly exponential) reduction of satisfiability to classical modal logics. We also analyse the complexity of these logics and prove NP-completeness for two variants of the satisfiability problem. Then in Section 6.3 we consider possible world semantics for the fuzzy autoepistemic logic and provide a characterisation of stable fuzzy expansions in terms of many-valued K45 belief sets, and also in terms of proper generalisations of stable sets. In Section 6.4, we generalise the propositional fragment of Levesque's "only knowing" logic, provide a sound and complete axiomatisation and show that there is a characterisation of stable fuzzy expansions in terms of the belief sets involving the "only knowing" operator O. We conclude with some final remarks about related work.

6.2 Fuzzy modal logics of belief: Extensions of BŁ_k^c

In this section we will introduce fuzzy modal logics of belief. We will do this by extending the minimal modal logic over finitely-valued Łukasiewicz logic with a finite set of truth constants BŁ_k^c; see Section 2.3. Recall that the language of the logic BŁ_k^c is $\mathcal{L}_{\mathrm{B}}^k$, see Definition 2.12. When defining logics of belief, it is common to require a number of properties of the

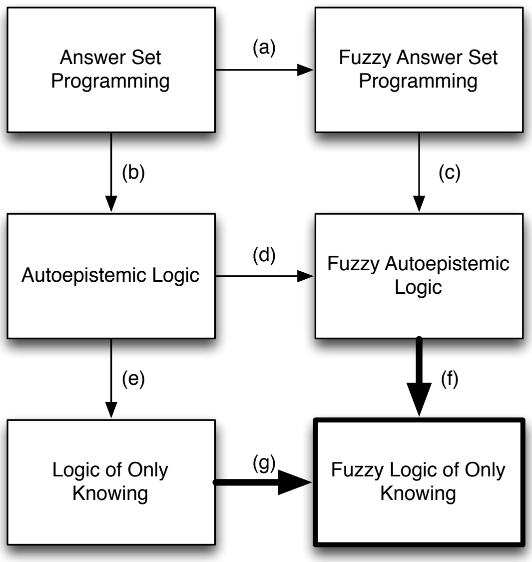


Figure 6.1: Summarising diagram of embeddings and generalisations
(a): e.g. [Van Nieuwenborgh et al. 2007], (b): [Gelfond and Lifschitz 1988], (c)-(d): Chapter 5, (e): [Levesque 1990], (f)-(g) Chapter 6

modal operator B which do not follow from the axioms of $\mathrm{BŁ}_k^c$. For instance, it is common to presume that the agent has both positive and negative introspective capabilities. This is captured in the classical case by the well-known axioms (4) and (5). Moreover, sometimes belief consistency is required which is captured by axiom (D). Finally, when dealing with knowledge instead of beliefs modal axiom (T) can be added. In particular, we will consider extensions of $\mathrm{BŁ}_k^c$ which are obtained by adding some or all of these classical axioms:

(D) $\sim\mathrm{B}\sim\bar{1}$

(4) $\mathrm{B}\phi \to \mathrm{BB}\phi$

(5) $\sim\mathrm{B}\phi \to \mathrm{B}\sim\mathrm{B}\phi$

(T) $\mathrm{B}\phi \to \phi$

As in the classical case [Chellas 1980, Fagin et al. 1994], we consider the following extensions of $\mathrm{BŁ}_k^c$:

- *K45*($Ł_k^c$): $\mathrm{BŁ}_k^c$ extended with axioms (4) and (5),
- *KD45*($Ł_k^c$): $\mathrm{BŁ}_k^c$ extended with axioms (D), (4) and (5)
- *S5*($Ł_k^c$): $\mathrm{BŁ}_k^c$ extended with axioms (T), (4) and (5)

We will denote by $\vdash_L$ the notion of proof for any of the logics

$$L \in \{\mathit{K45}(Ł_k^c), \mathit{KD45}(Ł_k^c), \mathit{S5}(Ł_k^c)\}.$$

The first task is to define the corresponding class of multi-valued Kripke models and to show soundness and completeness w.r.t. these models[1].

Similar as in Chapter 5 we define the set

$$A' = A \cup \left\{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}_{\mathrm{B}}^k\right\}.$$

In this chapter, we will sometimes treat A' as a set of atoms and consider formulas recursively built from A', the set of truth constants $S_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$ for a fixed $k \in \mathbb{N}$ and the connectives of Łukasiewicz logic (see Section 2.2.4). For a formula $\alpha \in \mathcal{L}_{\mathrm{B}}^k$ we will denote by α^* the corresponding formula in this non modal language $(\mathcal{L}_{\mathrm{B}}^k)^*$

- $p^* = p$ for a variable p,
- $\overline{c}^* = \overline{c}$ for $c \in S_k$,
- $(\phi \to \psi)^* = \phi^* \to \psi^*$ for $\phi, \psi \in \mathcal{L}_{\mathrm{B}}^k$ and $\to$ the Łukasiewicz implication,
- $(\mathrm{B}\alpha)^* = p_{B\alpha}$ with $p_{B\alpha}$ a fresh variable for $\alpha \in \mathcal{L}_{\mathrm{B}}^k$.

We write $T^* = \{\alpha^* \mid \alpha \in T\}$ for a set of formulas T in $\mathcal{L}_{\mathrm{B}}^k$. The set of all propositional evaluations $e : A' \to S_k$ will be denoted by Ω_k^*.

As a first step, we show a relation between proving a formula $\psi \in \mathcal{L}_{\mathrm{B}}^k$ in one of the extensions of $\mathrm{BŁ}_k^c$ and proving the corresponding B-free formula $\psi^* \in (\mathcal{L}_{\mathrm{B}}^k)^*$ from a suitable theory in the propositional logic $Ł_k^c$ (but over the set of variables A').

Lemma 6.1

Let L be any of the logics *K45*($Ł_k^c$), *KD45*($Ł_k^c$), *S5*($Ł_k^c$). Suppose $T \cup \{\psi\}$ is a set of formulas from $\mathcal{L}_{\mathrm{B}}^k$ and let $\Lambda_L = \{\phi^* \mid \vdash_L \phi\}$. Then it holds that

$$T \vdash_L \psi \quad \text{iff} \quad T^* \cup \Lambda_L \vdash \psi^*.$$

Proof. Suppose a proof for ψ in L from T has the form $\Gamma = (\gamma_1, \ldots, \gamma_m)$. A proof for ψ^* in $Ł_k^c$ from $T^* \cup \Lambda_L$ is then easily obtained by replacing all formulas γ_i in Γ by γ_i^*. Indeed, notice that for all i it holds that γ_i is one of the following

[1]We restrict ourselves to the logics *K45*($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$), but completeness results could be obtained in a similar way for any of the logics resulting from other combinations of the above axioms.

- an element in T, and then γ_i^* is an element of T^*;
- an instance of an axiom of L, and then γ_i^* is an element of Λ_L;
- a formula obtained by modus ponens from $\gamma_j = \gamma_k \to \gamma_i$ and γ_k, with $j, k < i$. Then γ_i^* is also obtained by modus ponens from γ_j^* and γ_k^*;
- a formula $\mathrm{B}\gamma_j$ obtained by necessitation from a theorem γ_j $(j < i)$ of L. Then γ_j is a theorem in L and γ_i^* is an element of Λ_L.

Conversely, suppose there is a proof $\Phi = (\phi_1, \ldots, \phi_n)$ for ψ^* in $Ł_k^c$ from $T^* \cup \Lambda_L$. The sequence Φ can then be converted to a proof for ψ in L from T as follows.

- If $\phi_i \in \Lambda_L$, i.e. $\phi_i = \psi^*$ with $\vdash_L \psi$, then replace ϕ_i by ψ and add a proof for ψ.
- Otherwise, replace $\phi_i = \psi^*$ by ψ.

□

The proof of Lemma 6.1 is based on the fact that a proof in L can be converted into a proof in $Ł_k^c$ and vice versa. Below we illustrate this idea with an example.

Example 6.1

Consider $T = \{\mathrm{B}a\}$. Then $T \vdash_L \mathrm{B}(b \to a)$ with b an arbitrary atom. Indeed, by axiom (Ł1) it follows that $\vdash_L a \to (b \to a)$ and thus by necessitation it follows that $\vdash_L \mathrm{B}(a \to (b \to a))$. By axiom (K) we have

$$\vdash_L \mathrm{B}(a \to (b \to a)) \to (\mathrm{B}a \to \mathrm{B}(b \to a)).$$

and by modus ponens we can then infer $\vdash_L \mathrm{B}a \to \mathrm{B}(b \to a)$. Another application of modus ponens using $\mathrm{B}a \in T$ then implies $T \vdash_L \mathrm{B}(b \to a)$.

On the other hand, if we consider $T^* = \{p_{\mathrm{B}a}\}$, then we have $T^* \cup \Lambda_L \vdash p_{\mathrm{B}(b \to a)}$. Indeed, since $\vdash_L \mathrm{B}a \to \mathrm{B}(b \to a)$, it follows that

$$p_{\mathrm{B}a} \to p_{\mathrm{B}(b \to a)} = (\mathrm{B}a \to \mathrm{B}(b \to a))^* \in \Lambda_L$$

and by modus ponens and $p_{\mathrm{B}a} \in T^*$ we derive that $T^* \cup \Lambda_L \vdash p_{\mathrm{B}(b \to a)}$.

We now define the canonical Kripke model for a given fuzzy modal logic L. We will use this particular Kripke model to show completeness in Theorem 6.1. The following definition applies to any logic L obtained by adding to the axioms of $\mathrm{B}Ł_k^c$ combinations of the axioms (D), (4), (5) and (T) and hence in particular for $L \in \{\mathit{K45}(Ł_k^c), \mathit{KD45}(Ł_k^c), \mathit{S5}(Ł_k^c)\}$.

Definition 6.1

Let L be any of the logics *K45*(Ł_k^c), *KD45*(Ł_k^c), *S5*(Ł_k^c). The L-canonical Kripke model is defined as the Kripke model $M_{can}^L = (W_{can}^L, e_{can}^L, R_{can}^L)$, where

- $W_{can}^L = \{w \in \Omega_k^* \mid \forall \phi \in \Lambda_L : w(\phi) = 1\}$ with $\Lambda_L = \{\phi^* \mid \vdash_L \phi\}$
- $R_{can}^L = \{(w_1, w_2) \in \Omega_k^* \times \Omega_k^* \mid \forall \phi \in \mathcal{L}_{\mathrm{B}}^k :$ if $w_1((\mathrm{B}\phi)^*) = 1,$ then $w_2(\phi^*) = 1\}$,
- $e_{can}^L(w, p) = w(p)$ for each variable p.

We now introduce some subclasses of the class $\mathbb{M}$ of Kripke models (W, e, R) with two-valued accessibility relations, depending on which properties R satisfies. Recall that a relation[2] $R : X \times X \to \{0, 1\}$ is

- Euclidean if $R(x, y) = R(x, z) = 1$ implies $R(y, z) = 1$,
- serial if for every $x \in X$ there exists $y \in X$ such that $R(x, y) = 1$.
- transitive if $R(x, y) = R(y, z) = 1$ implies $R(x, z) = 1$.
- reflexive if $R(x, x) = 1$ for all $x \in X$.
- symmetric if $R(x, y) = R(y, x)$ for all $x, y \in X$.

We can then define the following subclasses of $\mathbb{M}$.

Definition 6.2

- $\mathbb{M}_{\mathrm{et}}$: class of Kripke models (W, e, R) with R Euclidean and transitive.
- $\mathbb{M}_{\mathrm{est}}$: class of Kripke models (W, e, R) with R Euclidean, serial and transitive.
- $\mathbb{M}_{\mathrm{rsyt}}$: class of Kripke models (W, e, R) with R reflexive, symmetric and transitive.

In Theorem 6.1 we will show that the extensions of BŁ_k^c defined above are sound and complete axiomatisations for these subclasses of $\mathbb{M}$. To show completeness, we need the following truth lemma.

[2]Note that a relation $R \subseteq X \times X$ can be seen as a mapping $R : X \times X \to \{0, 1\}$ where $R(x, y) = 1$ iff $(x, y) \in R$. In this text will use both notations to denote two-valued relations.

Lemma 6.2

(Truth lemma) Suppose ϕ is a formula in $\mathcal{L}_{\mathrm{B}}^{k}$ and $L \in \{$*K45*(Ł_k^c), *KD45*(Ł_k^c), *S5*$(\text{Ł}_k^c)\}$ with M_{can}^{L} its canonical Kripke model. Then it holds that $v(\phi^*) = \|\phi\|_{M_{can}^{L},v}$, for every $v \in W_{can}^{L}$.

Proof. By using the monotonicity of B and the distributivity of $\vee$ and $\wedge$, the claim follows by an easy adaptation from Lemma 4.20 in [Bou et al. 2011b]. □

We can now show the following properties for the canonical Kripke models.

Proposition 6.1

Let $L \in \{$*K45*(Ł_k^c), *KD45*(Ł_k^c), *S5*$(\text{Ł}_k^c)\}$, then the following conditions hold

1. If L contains axiom (T) then R_{can}^{L} is reflexive.
2. If L contains axiom (4) then R_{can}^{L} is transitive.
3. If L contains axiom (5) then R_{can}^{L} is Euclidean.
4. If L contains axiom (D) then R_{can}^{L} is serial.

Proof. In this proof we frequently use the result from Lemma 6.2: for every $v \in W_{can}^{L}$ and every formula $\phi \in \mathcal{L}_{\mathrm{B}}^{k}$ it holds that $v(\phi^*) = \|\phi\|_{M_{can}^{L},v}$.

1. Let $w \in W_{can}^{L}$ and suppose that $w((\mathrm{B}\phi)^*) = 1$. We show $w(\phi^*) = 1$. Then it follows by the construction of the canonical model that $R_{can}^{L}(w,w) = 1$. Since L contains axiom (T), we have $(\mathrm{B}\phi \rightarrow \phi)^* \in \Lambda_L$ and it follows that

$$1 = w((\mathrm{B}\phi \rightarrow \phi)^*) = \|\mathrm{B}\phi \rightarrow \phi\|_{M_{can}^{L},w}.$$

Hence

$$1 = w((\mathrm{B}\phi)^*) = \|\mathrm{B}\phi\|_{M_{can}^{L},w} \leq \|\phi\|_{M_{can}^{L},w} = w(\phi^*).$$

2. Let $w_1, w_2, w_3 \in W_{can}^{L}$ such that $R_{can}^{L}(w_1,w_2) = R_{can}^{L}(w_2,w_3) = 1$. We show that $R_{can}^{L}(w_1,w_3) = 1$. Suppose that $w_1((\mathrm{B}\phi)^*) = 1$, we show $w_3(\phi^*) = 1$. Since L contains axiom (4), we have $(\mathrm{B}\phi \rightarrow \mathrm{B}\mathrm{B}\phi)^* \in \Lambda_L$ and it follows that

$$1 = w_1((\mathrm{B}\phi \rightarrow \mathrm{B}\mathrm{B}\phi)^*) = \|\mathrm{B}\phi \rightarrow \mathrm{B}\mathrm{B}\phi\|_{M_{can}^{L},w_1},$$

and hence that

$$1 = w_1((\mathrm{B}\phi)^*) = \|\mathrm{B}\phi\|_{M^L_{can},w_1} \leq \|\mathrm{BB}\phi\|_{M^L_{can},w_1} = w_1((\mathrm{BB}\phi)^*).$$

Since $R^L_{can}(w_1, w_2) = 1$, we then have that $w_2((\mathrm{B}\phi)^*) = 1$ and subsequently, since $R^L_{can}(w_2, w_3) = 1$, that $w_3(\phi^*) = 1$.

3. Let $w_1, w_2, w_3 \in W^L_{can}$ such that $R^L_{can}(w_1, w_2) = R^L_{can}(w_1, w_3) = 1$. We show that $R^L_{can}(w_2, w_3) = 1$. Suppose that $w_2((\mathrm{B}\phi)^*) = 1$. We show $w_3(\phi^*) = 1$. By definition of B,

$$\|\mathrm{B}{\sim}\mathrm{B}\phi\|_{M^L_{can},w_1} = \inf\{\|{\sim}\mathrm{B}\phi\|_{M^L_{can},w} \mid R^L_{can}(w_1, w) = 1\},$$

hence in particular

$$\|\mathrm{B}{\sim}\mathrm{B}\phi\|_{M^L_{can},w_1} \leq \|{\sim}\mathrm{B}\phi\|_{M^L_{can},w_2}.$$

Now since

$$\|{\sim}\mathrm{B}\phi\|_{M^L_{can},w_2} = 1 - \|\mathrm{B}\phi\|_{M^L_{can},w_2} = 1 - w_2((\mathrm{B}\phi)^*) = 0,$$

we obtain $\|\mathrm{B}{\sim}\mathrm{B}\phi\|_{M^L_{can},w_1} = 0$. But since $({\sim}\mathrm{B}{\sim}\mathrm{B}\phi \to \mathrm{B}\phi)^* \in \Lambda_L$, because of axiom (5), it follows that

$$1 = w_1(({\sim}\mathrm{B}{\sim}\mathrm{B}\phi \to \mathrm{B}\phi)^*) = \|{\sim}\mathrm{B}{\sim}\mathrm{B}\phi \to \mathrm{B}\phi\|_{M^L_{can},w_1}$$

and hence

$$1 = \|{\sim}\mathrm{B}{\sim}\mathrm{B}\phi\|_{M^L_{can},w_1} \leq \|\mathrm{B}\phi\|_{M^L_{can},w_1} = w_1((\mathrm{B}\phi)^*).$$

Finally, since $R^L_{can}(w_1, w_3) = 1$, it then follows that $w_3(\phi^*) = 1$.

4. Let $w_1 \in W^L_{can}$. We show that there exists $w_2 \in W^L_{can}$ such that $R^L_{can}(w_1, w_2) = 1$. Since by axiom (D) we have $({\sim}\mathrm{B}{\sim}\overline{1})^* \in \Lambda_L$, it follows that

$$1 = w_1(({\sim}\mathrm{B}{\sim}\overline{1})^*) = \|{\sim}\mathrm{B}{\sim}\overline{1}\|_{M^L_{can},w_1},$$

and thus

$$0 = \|\mathrm{B}{\sim}\overline{1}\|_{M^L_{can},w_1} = \inf\{\|\overline{0}\|_{M^L_{can},w} \mid R^L_{can}(w_1, w) = 1\}.$$

Therefore the latter set must be non-empty, and hence there must exist $w_2 \in W^L_{can}$ such that $R^L_{can}(w_1, w_2) = 1$. □

Using Proposition 6.1, we can now show the following theorem.

Theorem 6.1

K45($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$) are sound and complete w.r.t. the classes $\mathbb{M}_{\text{et}}$, $\mathbb{M}_{\text{est}}$ and $\mathbb{M}_{\text{rsyt}}$ respectively.

Proof. Soundness is straightforward. We can show the completeness by proving that if there is a formula $\phi \in \mathcal{L}_{\text{B}}^k$ such that $\nvdash_L \phi$ with

$$L \in \{\mathit{K45}(Ł_k^c), \mathit{KD45}(Ł_k^c), \mathit{S5}(Ł_k^c)\},$$

then there must exist a Kripke model $M = (W, e, R)$ in the corresponding subclass of Kripke models and a $w \in W$ with $\|\phi\|_{M,w} < 1$. We show that the L-canonical Kripke model meets this condition. The fact that each of these canonical Kripke models belongs to the correct subclass of $\mathbb{M}$ follows from Proposition 6.1 and from the fact that all reflexive and Euclidean relations are symmetric. By Lemma 6.1 it follows, independent of the choice of L, that $\Lambda_L \nvdash \phi^*$ and by the strong completeness of $Ł_k^c$ that $\Lambda_L \nvDash \phi^*$, i.e. there exists a $v \in W_{can}^L$ such that $v(\phi^*) < 1$. By Lemma 6.2 we obtain that $\|\phi\|_{M_{can}^L, v} < 1$. □

As in the classical case (see e.g. [Halpern and Moses 1992]), the logics *K45*($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$) admit simpler semantics while preserving soundness and completeness. Consider the following classes of Kripke models.

Definition 6.3

- $\mathbb{M}_{\text{et}}^s$: the subclass of Kripke models (W, e, R) with $R = W \times E$ for some fixed $E \subseteq W$
- $\mathbb{M}_{\text{est}}^s$: the subclass of Kripke models (W, e, R) with $R = W \times E$ for some fixed and non-empty $E \subseteq W$
- $\mathbb{M}_{\text{rsyt}}^s$: the subclass of Kripke models (W, e, R) with $R = W \times W$

Notice that $\mathbb{M}_{\text{et}}^s$, $\mathbb{M}_{\text{est}}^s$ and $\mathbb{M}_{\text{rsyt}}^s$ are subclasses of resp. $\mathbb{M}_{\text{et}}$, $\mathbb{M}_{\text{est}}$ and $\mathbb{M}_{\text{rsyt}}$.

Proposition 6.2

K45($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$) are sound and complete w.r.t. the classes $\mathbb{M}_{\text{et}}^s$, $\mathbb{M}_{\text{est}}^s$ and $\mathbb{M}_{\text{rsyt}}^s$ respectively.

Proof. We only prove the case of *KD45*($Ł_k^c$), the other cases being easy variations. By Theorem 6.1, it is sufficient to show that $\mathbb{M}_{\text{est}}$ and $\mathbb{M}_{\text{est}}^s$ have the same tautologies. Since

$\mathbb{M}^s_{\mathrm{est}}$ is a subclass of $\mathbb{M}_{\mathrm{est}}$, we only have to show that if for a formula $\phi \in \mathcal{L}^k_{\mathrm{B}}$ there exists an $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}$ and $w \in W$ such that $\|\phi\|_{M,w} < 1$, then there exists an $M' = (W', e', R') \in \mathbb{M}^s_{\mathrm{est}}$ and $w' \in W'$ such that $\|\phi\|_{M',w'} < 1$.

Suppose such a Kripke model $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}$ and $w \in W$ are given. Define $E = \{v \in W \mid R(w, v) = 1\}$. By the seriality and symmetry of R we have $E \neq \emptyset$. We define M' as follows: $W' = \{w\} \cup E$, $e' : W' \times V \to S_k : (w, p) \mapsto e(w, p)$ and $R' = W' \times E$.

Notice that for any $v \in E$ we have $E \subseteq \{z \in W \mid R(z, v) = 1\}$. Indeed, for every $z \in E$ we have $R(w, z) = 1$ and since R is Euclidean and $R(w, v) = 1$ (because $v \in E$) it follows that $R(z, v) = 1$. Since R is transitive and symmetric we also have $\{z \in W \mid R(z, v) = 1\} \subseteq E$. Indeed, if $R(z, v) = 1$, then since $R(v, w) = R(w, v) = 1$ it follows $R(w, z) = R(z, w) = 1$ and thus $z \in E$. Hence

$$E = \{z \in W \mid R(z, v) = 1\} = \{z \in W \mid R(v, z) = 1\}$$

for all $v \in E$.

We will now use this result to show by structural induction that for each $\psi \in \mathcal{L}^k_{\mathrm{B}}$ it holds that $\|\psi\|_{M,v} = \|\psi\|_{M',v}$ for every $v \in E$. The only notable case is when $\psi = \mathrm{B}\alpha$, but the result then follows by the fact that $E = \{z \in W \mid R(v, z) = 1\}$ for all $v \in E$ and by the hypothesis:

$$\begin{aligned}
\|\mathrm{B}\alpha\|_{M,v} &= \inf\{\|\alpha\|_{M,z} \mid R(v, z) = 1\} \\
&= \inf\{\|\alpha\|_{M,z} \mid z \in E\} \\
&= \inf\{\|\alpha\|_{M',z} \mid z \in E\} \\
&= \inf\{\|\alpha\|_{M',z} \mid R'(v, z) = 1\} \\
&= \|\mathrm{B}\alpha\|_{M',v}
\end{aligned}$$

where we use the fact that that $R' = W' \times E$.

We will use this last result to show that $\|\psi\|_{M,w} = \|\psi\|_{M',w}$ for all $\psi \in \mathcal{L}^k_{\mathrm{B}}$. In particular, it then follows that $\|\phi\|_{M',w} = \|\phi\|_{M,w} < 1$. We will show this by structural induction. Again, the only notable case is when $\psi = \mathrm{B}\alpha$:

$$\begin{aligned}
\|\mathrm{B}\alpha\|_{M',w} &= \inf\{\|\alpha\|_{M',z} \mid R'(w, z) = 1\} \\
&= \inf\{\|\alpha\|_{M',z} \mid z \in E\} \\
&= \inf\{\|\alpha\|_{M,z} \mid z \in E\} \\
&= \inf\{\|\alpha\|_{M,z} \mid R(w, z) = 1\} \\
&= \|\mathrm{B}\alpha\|_{M,w}
\end{aligned}$$

where we use the fact that $E = \{v \in W \mid R(w, v) = 1\}$. □

Finally, let us show that any of our many-valued modal logics can be reduced to classical modal logics. We will use this result in Section 6.4 to obtain a sound and complete axiomatisation for the many-valued logic of only knowing we will define.

For each formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$ and truth value $r \in S_k$ consider the formula $\phi_r \in \mathcal{L}_{\mathrm{B}}^k$ defined as $\Delta(\phi \leftrightarrow \overline{r})$. This formula is a Boolean formula and for each Kripke model $M = (W, e, R)$ in $\mathbb{M}$ and each $w \in W$ we have $\|\phi_r\|_{M,w} = 1$ iff $\|\phi\|_{M,w} = r$.

We define $(\mathcal{L}_{\mathrm{B}}^k)' \subseteq \mathcal{L}_{\mathrm{B}}^k$ as the set of formulas constructed from the set of formulas

$$\{\Delta(p \leftrightarrow \overline{r}) \mid p \in V, r \in S_k\}.$$

In particular we have

- $\Delta(p \leftrightarrow \overline{r}) \in (\mathcal{L}_{\mathrm{B}}^k)'$ for every $p \in V$ and $r \in S_k$
- $(\phi \rightarrow \psi) \in (\mathcal{L}_{\mathrm{B}}^k)'$ if $\phi, \psi \in (\mathcal{L}_{\mathrm{B}}^k)'$
- $\mathrm{B}\phi \in (\mathcal{L}_{\mathrm{B}}^k)'$ if $\phi \in (\mathcal{L}_{\mathrm{B}}^k)'$.

In the following lemma we will use the short notation ϕ_r for a formula $\Delta(\phi \leftrightarrow \overline{r})$. We will show that for each ϕ_r, there exists some $(\phi_r)' \in (\mathcal{L}_{\mathrm{B}}^k)'$ such that the truth values remain the same.

Lemma 6.3

For any formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$ and truth value $r \in S_k$, there exists a formula $(\phi_r)' \in (\mathcal{L}_{\mathrm{B}}^k)'$ such that for each Kripke model $M = (W, e, R) \in \mathbb{M}$ and each world $w \in W$ it holds that

$$\|\phi_r\|_{M,w} = \|(\phi_r)'\|_{M,w}.$$

Proof. We show this lemma by induction on the structure of the formula.

- If $\phi = p \in A$, then we can choose

$$(\phi_r)' = \Delta(p \leftrightarrow \overline{r}).$$

- Suppose that $\phi = \overline{c}$ with $c \in S_k$. Then $\|\phi_r\|_{M,w} = 1$ iff $r = c$. If $c = r$, then we can choose any tautology, e.g.

$$(\phi_r)' = \Delta(p \leftrightarrow \overline{1}) \leftrightarrow \Delta(p \leftrightarrow \overline{1})$$

  with $p \in A$ arbitrary. If $c \neq r$, then we can choose any contradiction, e.g.

$$(\phi_r)' = \sim(\Delta(p \leftrightarrow \overline{1}) \leftrightarrow \Delta(p \leftrightarrow \overline{1}))$$

  with $p \in A$ arbitrary.

- Suppose the claim holds for formulas ϕ and ψ, i.e. for every $s \in S_k$ there exist some $(\phi_s)'$ and $(\psi_s)'$ in $(\mathcal{L}_{\mathrm{B}}^k)'$ such that $\|\phi_s\|_{M,w} = \|(\phi_s)'\|_{M,w}$ and $\|\psi_s\|_{M,w} = \|(\psi_s)'\|_{M,w}$ for every M and every w. We show that the lemma also holds for $\alpha = \phi \to \psi$. Indeed, for $r \in S_k$ we choose

$$(\alpha_r)' = \bigvee_{s,t \in S_k} \{(\phi_s)' \wedge (\psi_t)' \mid r = I_L(s,t)\}.$$

Then we have

$$\begin{array}{rcl} \|\alpha_r\|_{M,w} = 1 & \Leftrightarrow & I_L(\|\phi\|_{M,w}, \|\psi\|_{M,w}) = r \\ & \Leftrightarrow & \exists s,t \in S_k : \|\phi\|_{M,w} = s, \|\psi\|_{M,w} = t, r = I_L(s,t) \\ & \Leftrightarrow & \exists s,t \in S_k : \|\phi_s\|_{M,w} = 1, \|\psi_t\|_{M,w} = 1, r = I_L(s,t) \\ & \Leftrightarrow & \exists s,t \in S_k : \|(\phi_s)'\|_{M,w} = 1, \|(\psi_t)'\|_{M,w} = 1, r = I_L(s,t) \\ & \Leftrightarrow & \|(\alpha_r)'\|_{M,w} = 1 \end{array}$$

Since α_r and $(\alpha_r)'$ are both Boolean formulas it follows that $\|\alpha_r\|_{M,w} = \|(\alpha_r)'\|_{M,w}$.

- Suppose the claim holds for a formula ϕ, i.e. for every $s \in S_k$ there exist some $(\phi_s)'$ in $(\mathcal{L}_{\mathrm{B}}^k)'$ such that $\|\phi_s\|_{M,w} = \|(\phi_s)'\|_{M,w}$ for every M and every w. We show that it also holds for $\alpha = \mathrm{B}\phi$. Suppose that $r \in S_k$. If $r = 1$, then consider $(\alpha_1)' = \mathrm{B}(\phi_1)' \in (\mathcal{L}_{\mathrm{B}}^k)'$:

$$\begin{array}{rcl} \|\alpha_1\|_{M,w} = 1 & \Leftrightarrow & \|\mathrm{B}\phi\|_{M,w} = 1 \\ & \Leftrightarrow & \inf\{\|\phi\|_{M,v} \mid R(w,v) = 1\} = 1 \\ & \Leftrightarrow & \forall v \in W \text{ such that } R(w,v) = 1 : \|\phi\|_{M,v} = 1 \\ & \Leftrightarrow & \forall v \in W \text{ such that } R(w,v) = 1 : \|\phi_1\|_{M,v} = 1 \\ & \Leftrightarrow & \forall v \in W \text{ such that } R(w,v) = 1 : \|(\phi_1)'\|_{M,v} = 1 \\ & \Leftrightarrow & \inf\{\|(\phi_1)'\|_{M,v} \mid R(w,v) = 1\} = 1 \\ & \Leftrightarrow & \|\mathrm{B}(\phi_1)'\|_{M,w} = 1 \\ & \Leftrightarrow & \|(\alpha_r)'\|_{M,w} = 1 \end{array}$$

If $r < 1$, then let r^+ be the successor of r, i.e. if $r = \frac{i}{k}$, then $r^+ = \frac{i+1}{k}$. Define formulas $\phi_{\geq r}$, $\phi_{>r}$, $\phi_{\leq r}$ and $\phi_{<r}$ as disjunctions[3] of formulas ϕ_s, i.e. $\phi_{\geq r} = \bigvee_{s \geq r} \phi_s$ which means that $\|\phi_{\geq r}\|_{M,w} = 1$ iff $\|\phi\|_{M,w} \geq r$ and similar for $\phi_{>r}$, $\phi_{\leq r}$ and $\phi_{<r}$. Then we have that α_r is equivalent to $\alpha_{\geq r} \wedge {\sim}(\alpha_{\geq r^+})$.

[3]Note that for Boolean formulas, $\oplus$, $\vee$ and the classical disjunction coincide. The same result holds for $\otimes$, $\wedge$ and the classical conjunction.

First note that for each $s \in S_k$ and each formula ψ we have

$$\begin{aligned}
\|(\mathrm{B}\psi)_{\geq s}\|_{M,w} = 1 \quad &\Leftrightarrow \quad \|\mathrm{B}\psi\|_{M,w} \geq s \\
&\Leftrightarrow \quad \inf\{\|\psi\|_{M,v} \mid R(w,v) = 1\} \geq s \\
&\Leftrightarrow \quad \forall v \in W \text{ such that } R(w,v) = 1 : \|\psi\|_{M,v} \geq s \\
&\Leftrightarrow \quad \forall v \in W \text{ such that } R(w,v) = 1 : \|\psi_{\geq s}\|_{M,v} = 1 \\
&\Leftrightarrow \quad \forall v \in W \text{ such that } R(w,v) = 1 : \|\textstyle\bigvee_{t \geq s} \psi_t\|_{M,v} = 1 \\
&\Leftrightarrow \quad \|\mathrm{B}(\textstyle\bigvee_{t \geq s} \psi_t)\|_{M,w} = 1
\end{aligned}$$

Hence we obtain

$$\begin{aligned}
\|\alpha_r\|_{M,w} = 1 \quad &\Leftrightarrow \quad \|\alpha_{\geq r} \wedge \sim(\alpha_{\geq r^+})\|_{M,w} = 1 \\
&\Leftrightarrow \quad \|(\mathrm{B}\phi)_{\geq r}\|_{M,w} = 1 \text{ and } \|\sim((\mathrm{B}\phi)_{\geq r^+})\|_{M,w} = 1 \\
&\Leftrightarrow \quad \|\mathrm{B}(\textstyle\bigvee_{t \geq r} \phi_t)\|_{M,w} = 1 \text{ and } \|\sim(\mathrm{B}(\textstyle\bigvee_{t \geq r^+} \phi_t))\|_{M,w} = 1 \\
&\Leftrightarrow \quad \|\mathrm{B}(\textstyle\bigvee_{t \geq r} \phi_t) \wedge \sim\mathrm{B}(\textstyle\bigvee_{t \geq r^+} \phi_t)\|_{M,w} = 1
\end{aligned}$$

By the induction hypothesis it follows that there exists $(\alpha_r)' \in (\mathcal{L}_\mathrm{B}^k)'$ such that $\|\alpha_r\|_{M,w} = \|(\alpha_r)'\|_{M,w}$.

□

Example 6.2

Consider $S_2 = \{0, \frac{1}{2}, 1\}$, $\phi = a \rightarrow b$, $\psi = \mathrm{B}a$ and $r = \frac{1}{2}$. Then the following formulas are the ones that are constructed in the proof of Lemma 6.3

$$(\phi_{\frac{1}{2}})' = (a_{\frac{1}{2}} \wedge b_0) \vee (a_1 \wedge b_{\frac{1}{2}})$$

and

$$(\psi_{\frac{1}{2}})' = \mathrm{B}(a_{\frac{1}{2}} \vee a_1) \wedge \sim\mathrm{B}(a_1).$$

Using Lemma 6.3 we then have the following reduction from $\mathrm{BŁ}_k^c$ to classical modal logic. Note that this reduction is not necessarily polynomial since for a formula $\phi \in \mathcal{L}_\mathrm{B}^k$ and truth value $r \in S_k$, the length of the constructed formula $(\phi_r)' \in (\mathcal{L}_\mathrm{B}^k)'$ in Lemma 6.3 can be exponential in the length of $\phi \in \mathcal{L}_\mathrm{B}^k$.

Proposition 6.3

For each formula $\phi \in \mathcal{L}_\mathrm{B}^k$ and $r \in S_k$ there exists some formula $\psi \in (\mathcal{L}_\mathrm{B}^k)'$ such that

$$\|\phi\|_{M,w} = r \text{ iff } \|\psi\|_{M,w} = 1$$

for every Kripke model $M = (W, e, R) \in \mathbb{M}$ and every world $w \in W$.

In the remainder of this section we will discuss the complexity of two satisfiability problems for *KD45*($Ł_k^c$). As before, the same results can be obtained for *K45*($Ł_k^c$) and *S5*($Ł_k^c$). More precisely, we will discuss the following decision problems.

- 1-SAT: Given a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, does there exist an $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}^s$ and a $w \in W$ such that $\|\phi\|_{M,w} = 1$?
- pos-SAT: Given a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, does there exist an $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}^s$ and a $w \in W$ such that $\|\phi\|_{M,w} > 0$?

We will show that these problems are NP-complete, which is the same complexity class as the corresponding decision problem for classical KD45. See [Halpern and Moses 1992] for more details on the complexity of classical modal logics.

Note that 1-SAT and pos-SAT can be polynomially reduced to each other. Indeed, a formula ϕ is "pos-SAT" if $\sim\Delta(\sim\phi)$ is "1-SAT" and a formula ϕ is "1-SAT" if $\Delta\phi$ is "pos-SAT". Hence it is sufficient to show the NP-completeness of 1-SAT.

For any formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, we denote by $\#\phi$ its *length*:

- $\#\overline{c} = 1$ for each $c \in S_k$ and $\#p = 1$ for every $p \in A$
- $\#(\phi \to \psi) = 1 + \#\phi + \#\psi$ and similar for the other connectives
- $\#(\mathrm{B}\phi) = 1 + \#\phi$.

For a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, we denote by $d(\phi)$ its *depth* which is defined as usual by counting the nested occurrences of the modality B. In particular, for a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, we have

- $d(\overline{c}) = 0$ for each $c \in S_k$ and $d(p) = 0$ for every $p \in A$
- $d(\phi \to \psi) = \max\{d(\phi), d(\psi)\}$ and similar for the other connectives
- $d(\mathrm{B}\phi) = 1 + d(\phi)$.

Example 6.3

Consider the formula $\phi = \mathrm{B}a \oplus \sim b$ with a and b atoms. Then

$$\#(\mathrm{B}a \oplus \sim b) = 1 + \#(\mathrm{B}a) + \#(\sim b) = 1 + 1 + \#(a) + 1 + \#(b) = 5.$$

For φ and ψ propositional (B-free) formulas, we have: $d(\varphi) = d(\psi) = 0$, $d(\mathrm{B}(\varphi)) = d(\mathrm{B}(\varphi \oplus \psi)) = d(\varphi \oplus \mathrm{B}(\psi)) = 1$, $d(B(\varphi \oplus B(\psi))) = 2$, $d(\mathrm{B}\varphi \otimes (\mathrm{B}(\mathrm{B}(\mathrm{B}\psi)))) = 3$ and so forth.

We can then show the following finite model property:

Lemma 6.4

Let ϕ be a formula in $\mathcal{L}^k_{\mathrm{B}}$. Then for every model $M = (W, e, R) \in \mathbb{M}^s_{\mathrm{est}}$, and for every $w \in W$, there exists a finite model $M' = (W', e', R') \in \mathbb{M}^s_{\mathrm{est}}$ and a world $w' \in W'$ such that $|W'| \leq \#\phi$ and $\|\phi\|_{M,w} = \|\phi\|_{M',w'}$.

Proof. Consider a Kripke model $M = (W, e, R)$ with $R = W \times E$ ($\emptyset \neq E \subseteq W$) and $w \in W$. The aim is to find a finite set W', a non empty subset $E' \subseteq W'$, a mapping $e' : W' \times V \to S_k$ and some $w' \in W'$ for which the claim holds.

Trivially, if ϕ is B-free, then take $W' = E' = \{w\}$, $R' = W' \times E'$, let e' be defined by restriction of $e : W \times V \to S_k$ to $W' \times V$ and let $w' = w$.

In general, if $d(\phi) \geq 1$ we proceed as follows. Let $\mathrm{B}\psi^1_1, \ldots, \mathrm{B}\psi^1_{i_1}$ be the subformulas of ϕ of depth 1, which means that each ψ^1_j is B-free. Since the set of truth-values S_k is finite, clearly for each ψ^1_j, there exists a world w^1_j such that

$$\|\mathrm{B}\psi^1_j\|_{M,w} = \|\psi^1_j\|_{M,w^1_j} = w^1_j(\psi^1_j).$$

Letting $r_{1j} = \|\mathrm{B}\psi^1_j\|_{M,w}$, now replace each subformula $\mathrm{B}\psi^1_j$ by the corresponding truth constant $\overline{r_{1j}}$ and, if $d = d(\phi) > 1$, repeat the process for all levels $2, \ldots, d$. Let $E' = \{w^l_j \mid 1 \leq l \leq d, 1 \leq j \leq i_l\}$, $W' = \{w\} \cup E'$, $w' = w$ and let e' be defined by restriction of $e : W \times V \to S_k$ to $W' \times V$. Then, by construction, $\|\phi\|_{M,w} = \|\phi\|_{M',w'}$. Moreover, $|W'| = 1 + \sum_{l=1}^{d} i_l \leq \#\phi$. □

Observe that, as in the proof of Lemma 6.4, given a formula $\phi \in \mathcal{L}^k_{\mathrm{B}}$ and a Kripke model $M = (W, e, R) \in \mathbb{M}^s_{\mathrm{est}}$, we can construct a B-free formula ϕ^M. Indeed, given this Kripke model M every subformula of the form $\mathrm{B}\psi$ can be substituted by a truth constant. We will use this construction in the following theorem.

Theorem 6.2

The problems 1-SAT and pos-SAT for *KD45*(Ł^c_k) are NP-complete.

Proof. Since each formula of Łukasiewicz logic is in particular a formula of $\mathcal{L}_{\mathrm{B}}^k$, and since 1-SAT (as well as pos-SAT) for $Ł_k^c$ is NP-complete, the NP-hardness of our problems follow. In order to prove NP-membership, recall that from Lemma 6.4 a formula ϕ is 1-SAT in a model M iff ϕ is 1-SAT in a finite model M' whose cardinality is polynomial in the length of ϕ. Let us guess the model $M' = (W', e', R')$. Since $|W'| \leq \#\phi$, the guess is polynomial in $\#\phi$. Let $\phi^{M'}$ be the formula of $Ł_k^c$ obtained from M' and ϕ by applying the procedure described above, and notice that $\#\phi^{M'}$ is polynomial in $\#\phi$. Moreover, since $|W'| \leq \#\phi$ the formula $\#\phi^{M'}$ is obtained in a number of steps which is polynomial in $\#\phi$. From [Mundici 1987] it follows that checking 1-SAT (as well as pos-SAT) for $\phi^{M'}$ in $Ł_k^c$ is in NP. Hence NP-membership follows. □

Remark 6.1

It follows that for each $r \in S_k$, the following decision problem, which is called r-SAT, is NP-complete as well:

"Given a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, does there exist an $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}^s$ and a $w \in W$ such that $\|\phi\|_{M,w} = r$?"

Indeed, this problem is equivalent to 1-SAT for the formula $\psi = \Delta(\phi \leftrightarrow \overline{r})$ where ψ is polynomial in ϕ. Hence NP-membership follows. NP-hardness follows from the fact that for each $r > 0$, 1-SAT can be reduced to r-SAT since a formula $\|\phi\|_{M,w} = 1$ iff $\|\phi \otimes \overline{r}\|_{M,w} = r$. For $r = 0$, we can reduce 1-SAT as follows: $\|\phi\|_{M,w} = 1$ iff $\|{\sim}\phi\|_{M,w} = 0$.

Note that in [Bou et al. 2011a], it was shown that 1-SAT and pos-SAT for the minimal modal logic over $Ł_k$ are PSPACE-complete when the relation R in Kripke models (W, e, R) is many-valued.

6.3 Relating fuzzy modal logic and fuzzy autoepistemic logic

In this section we will discuss the relation between the fuzzy modal logic introduced in the previous section and the fuzzy autoepistemic logic from Section 5.2 but restricting to finitely-valued Łukasiewicz logic with truth constants. In this setting, the language of fuzzy autoepistemic logic is $\mathcal{L}_{\mathrm{B}}^k$, see Definition 2.12. The basic construct is the notion of a *stable fuzzy expansion* E of a set of premises T, introduced in a more general setting in

Definition 5.2. The following definition coincides with Definition 5.2 but for the case of finitely-valued Łukasiewicz logic with truth constants $Ł_k^c$.

Definition 6.4

A *stable fuzzy expansion* of a set of $\mathcal{L}_{\mathrm{B}}^k$-formulas T is a fuzzy set $E : \mathcal{L}_{\mathrm{B}}^k \to S_k$ that satisfies the following fix-point condition:

$$E(\phi) = \min\left\{v(\phi^*) \mid v \in \Omega_k^*, \forall\alpha \in T^* \cup \{(\mathrm{B}\psi)^* \leftrightarrow \overline{E(\psi)} \mid \psi \in \mathcal{L}_{\mathrm{B}}^k\} : v(\alpha) = 1\right\}$$

for all $\phi \in \mathcal{L}_{\mathrm{B}}^k$.

Recall that $\overline{E_A(\psi)}$ denotes the truth constant corresponding to the truth value $E_A(\psi) \in S_k$, and that for a formula $\phi \in \mathcal{L}_{\mathrm{B}}^k$, the formula ϕ^* is the corresponding B-free formula in the language $(\mathcal{L}_{\mathrm{B}}^k)^*$. As in the previous section we will denote by Ω_k^* the set of evaluations $w : A' \to S_k$ with

$$A' = A \cup \{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}_{\mathrm{B}}^k\}.$$

Using the strong completeness of $Ł_k^c$, in particular the fact that for any set of $Ł_k^c$ formulas $T \cup \{\phi\}$ one has

$$\min\{v(\phi) \mid v(\psi) = 1 \text{ for all } \psi \in T\} = \max\{r \in S_k \mid T \vdash \overline{r} \to \phi\},$$

one can rewrite the above definition of stable fuzzy expansion of A as a fuzzy set E_A satisfying the following fix-point condition:

$$E_A(\phi) = \max\left\{r \in S_k \mid A^* \cup \{(\mathrm{B}\psi)^* \leftrightarrow \overline{E_A(\psi)} \mid \psi \in \mathcal{L}_{\mathrm{B}}\} \vdash \overline{r} \to \phi^*\right\}.$$

Finally, this condition can be rewritten as the following two joint conditions:

$$\underline{E_A} = \left\{\phi \in \mathcal{L}_{\mathrm{B}}^k \mid A^* \cup \{(\mathrm{B}\psi)^* \leftrightarrow \overline{E_A(\psi)} \mid \psi \in \mathcal{L}_{\mathrm{B}}\} \vdash \phi^*\right\}, \text{ and}$$
$$E_A(\phi) = \max\{r \in S_k \mid \overline{r} \to \phi \in \underline{E_A}\}.$$

Notice that $\phi \in \underline{E_A}$ if and only if $E_A(\phi) = 1$.

Definition 6.5

Truth for fuzzy autoepistemic formulas is defined relative to structures (v, S) where $v \in \Omega_k$ and $S \subseteq \Omega_k$ with Ω_k the set of all propositional evaluations $w : A \to S_k$. The class of these structures will be denoted by $\mathbb{M}^{\mathrm{ae}}$.

Truth evaluations for fuzzy autoepistemic formulas w.r.t. $\mathbb{M}^{\mathrm{ae}}$ are then recursively defined as follows:

- $\|a\|_{(v,S)} = v(a)$ for $a \in A$,
- $\|\overline{c}\|_{(v,S)} = c$ for truth constants in $\{\overline{c} \mid c \in S_k\}$,
- $\|\mathrm{B}\alpha\|_{(v,S)} = \inf_{w \in S} \|\alpha\|_{(w,S)}$ for $\alpha \in \mathcal{L}_{\mathrm{B}}^k$,
- $\|\phi \to \psi\|_{(v,S)} = \min(1 - \|\phi\|_{(v,S)} + \|\psi\|_{(v,S)}, 1)$ for $\phi, \psi \in \mathcal{L}_{\mathrm{B}}^k$.

We consider the following subclasses of $\mathbb{M}^{\mathrm{ae}}$:

- the class $\mathbb{M}_e^{\mathrm{ae}}$, where only pairs (v, S) with S non-empty are considered,
- the class $\mathbb{M}_{in}^{\mathrm{ae}} \subseteq \mathbb{M}_e^{\mathrm{ae}}$, where only pairs (v, S) with $v \in S$ are considered.

Intuitively, one can think of S as a set of “sources" (worlds) and we define the truth value of $\mathrm{B}\varphi$ in S as the minimal value of φ such that each source supports it at least to this degree. Since the truth evaluation of formulas of the form $\mathrm{B}\varphi$ in a structure (w, S) does not depend on the actual world w, we will also write $\|\mathrm{B}\varphi\|_S$ to denote $\|\mathrm{B}\varphi\|_{(w,S)}$. Note that if $S = \emptyset$, then $\|\mathrm{B}\varphi\|_S = 1$. Also note that, conversely, the interpretation w in (w, S) is needed to evaluate non-modal formulas.

It can be shown that *K45*($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$) are still sound and complete with respect to the classes $\mathbb{M}^{\mathrm{ae}}$, $\mathbb{M}_e^{\mathrm{ae}}$ and $\mathbb{M}_{in}^{\mathrm{ae}}$ respectively.

Theorem 6.3

K45($Ł_k^c$), *KD45*($Ł_k^c$) and *S5*($Ł_k^c$) are sound and complete w.r.t. to $\mathbb{M}^{\mathrm{ae}}$, $\mathbb{M}_e^{\mathrm{ae}}$ and $\mathbb{M}_{in}^{\mathrm{ae}}$, respectively.

Proof. We only show the case of *KD45*($Ł_k^c$). The other cases are obtained by slight adaptations of the proof. By Proposition 6.2 it is sufficient to show that $\mathbb{M}_{\mathrm{est}}^s$ and $\mathbb{M}_e^{\mathrm{ae}}$ have the same tautologies.

First suppose there exists an $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}^s$, i.e. $R = W \times E$ and $\emptyset \neq E \subseteq W$, and $w \in W$ such that $\|\phi\|_{M,w} < 1$. We show that for $(e(w, \cdot), S) \in \mathbb{M}_e^{\mathrm{ae}}$ where $S = \{e(w', \cdot) \mid w' \in E\}$, we have $\|\phi\|_{(e(w,\cdot),S)} < 1$. We will do this by showing that $\|\phi\|_{M,w} = \|\phi\|_{(e(w,\cdot),S)}$. To obtain this result we will show by structural induction that for each formula γ we have

$$\|\gamma\|_{M,z} = \|\gamma\|_{(e(z,\cdot),S)} \text{ for each } z \in W.$$

The only notable case is $\gamma = \mathrm{B}\alpha$ for which it holds that $\|\alpha\|_{M,v} = \|\alpha\|_{(e(v,\cdot),S)}$ for all $v \in W$ (induction hypothesis). Now consider $z \in W$, we show that $\|\mathrm{B}\alpha\|_{M,z} = \|\mathrm{B}\alpha\|_{(e(z,\cdot),S)}$.

$$\begin{aligned}\|\mathrm{B}\alpha\|_{M,z} &= \inf\{\|\alpha\|_{M,v} \mid R(z,v) = 1\} = \inf\{\|\alpha\|_{M,v} \mid v \in E\} \\ &= \inf\{\|\alpha\|_{(e(v,\cdot),S)} \mid v \in E\} = \inf\{\|\alpha\|_{(v',S)} \mid v' \in S\}\end{aligned}$$

$$= \|\mathrm{B}\alpha\|_{(e(z,\cdot),S)}$$

To show the other direction, suppose we have $(v, S) \in \mathbb{M}_e^{\mathrm{ae}}$ such that $\|\phi\|_{(v,S)} < 1$. Define $M = (W, e, R) \in \mathbb{M}_{\mathrm{est}}^s$ as follows. Let W be a set of worlds such that W has the same cardinality as $S' = \{v\} \cup S$. Hence there exists a bijection $h : W \to \{v\} \cup S' : w \to w'$. The mapping $e : W \times V \to S_k$ is defined as $e(w, \cdot) = h(w)$ and we define $R = W \times E$ with $E = \{w \in W \mid h(w) \in S\}$. We show by structural induction that for each formula γ and each $w \in W$ we have

$$\|\gamma\|_{M,w} = \|\gamma\|_{(h(w),S)}.$$

The only non-trivial case is $\gamma = \mathrm{B}\alpha$ such that $\|\alpha\|_{M,w} = \|\alpha\|_{(h(w),S)}$ for each $w \in W$ (induction hypothesis). But then we have for $z \in W$ by the induction hypothesis that

$$\begin{aligned}\|\mathrm{B}\alpha\|_{M,z} &= \inf\{\|\alpha\|_{M,w} \mid R(z,w) = 1\} = \inf\{\|\alpha\|_{M,w} \mid w \in E\} \\ &= \inf\{\|\alpha\|_{M,w} \mid h(w) \in S\} = \inf\{\|\alpha\|_{(h(w),S)} \mid h(w) \in S\} \\ &= \|\mathrm{B}\alpha\|_{(h(z),S)}\end{aligned}$$

In particular it then follows that $\|\phi\|_{M,h^{-1}(v)} = \|\phi\|_{(v,S)} < 1$. □

To summarise, Table 6.1 provides a list of the languages and corresponding logics and semantics that have been introduced and used so far, as well as the one we will introduce in the next section.

We also modify Definition 5.4 to the setting of $Ł_k^c$.

Definition 6.6

A set $S \subseteq \Omega_k$ is a *fuzzy possible world autoepistemic model* of a fuzzy autoepistemic theory $T \subseteq \mathcal{L}_{\mathrm{B}}^k$ iff

$$S = \{v \in \Omega_k \mid \forall \varphi \in T : \|\varphi\|_{(v,S)} = 1\}.$$

Recall that from Proposition 5.1 we obtain that $E : \mathcal{L}_{\mathrm{B}}^k \to S_k$ is a stable fuzzy expansion of a set of formulas T iff it is the fuzzy belief set for some fuzzy possible world autoepistemic model S of T, i.e.

$$E(\phi) = \|\mathrm{B}\phi\|_S$$

for all $\phi \in \mathcal{L}_{\mathrm{B}}^k$.

On the other hand, as in the classical case, we can also characterise fuzzy belief sets, or equivalently stable fuzzy expansions, by the syntactic notion of fuzzy stable sets (cfr. [Halpern and Moses 1992]).

Table 6.1: List of languages, logics and semantics used in this chapter.

	Syntax	Logic	Semantics
finitely-valued Łukasiewicz logic with truth constants over variables A	$\mathcal{L}_k^c$	$Ł_k^c$	Ω_k
minimal modal logic over $Ł_k^c$ with crisp accessibility relations over variables A	$\mathcal{L}_{\mathrm{B}}^k$	$\mathrm{B}Ł_k^c$	$\mathbb{M}$
K45 over $Ł_k^c$	$\mathcal{L}_{\mathrm{B}}^k$	*K45*($Ł_k^c$)	$\mathbb{M}_{\mathrm{et}}$
	$\mathcal{L}_{\mathrm{B}}^k$	*K45*($Ł_k^c$)	$\mathbb{M}_{\mathrm{et}}^s$
	$\mathcal{L}_{\mathrm{B}}^k$	*K45*($Ł_k^c$)	$\mathbb{M}^{\mathrm{ae}}$
KD45 over $Ł_k^c$	$\mathcal{L}_B$	*KD45*($Ł_k^c$)	$\mathbb{M}_{\mathrm{est}}$
	$\mathcal{L}_{\mathrm{B}}^k$	*KD45*($Ł_k^c$)	$\mathbb{M}_{\mathrm{est}}^s$
	$\mathcal{L}_{\mathrm{B}}^k$	*KD45*($Ł_k^c$)	$\mathbb{M}_e^{\mathrm{ae}}$
S5 over $Ł_k^c$	$\mathcal{L}_{\mathrm{B}}^k$	*S5*($Ł_k^c$)	$\mathbb{M}_{\mathrm{rsyt}}$
	$\mathcal{L}_{\mathrm{B}}^k$	*S5*($Ł_k^c$)	$\mathbb{M}_{\mathrm{rsyt}}^s$
	$\mathcal{L}_{\mathrm{B}}^k$	*S5*($Ł_k^c$)	$\mathbb{M}_{in}^{\mathrm{ae}}$
"only knowing" over $Ł_k^c$	$\mathcal{L}_{\mathrm{O}}^k$	$\mathrm{O}(Ł_k^c)$	$\mathbb{M}^{\mathrm{ae}}$

Definition 6.7

Let $\Gamma : \mathcal{L}^k_{\mathrm{B}} \rightarrow S_k$ be a fuzzy set of modal formulas and let $\hat{\Gamma} = \{\overline{\Gamma(\varphi)} \rightarrow \varphi^* \mid \varphi \in \mathcal{L}^k_{\mathrm{B}}\}$. We say that Γ is a *fuzzy stable set* if the following conditions hold:

(1) $\hat{\Gamma}$ is propositionally consistent, i.e. $\hat{\Gamma} \not\vdash \overline{0}$.

(2) If $\hat{\Gamma} \vdash \bar{c} \rightarrow \varphi^*$, then $\Gamma(\varphi) \geq c$.

(3) $\Gamma(\varphi) = \Gamma(\mathrm{B}\varphi)$

(4) $1 - \Gamma(\varphi) = \Gamma({\sim}\mathrm{B}\varphi)$

Proposition 6.4

Γ is a fuzzy stable set iff Γ is a fuzzy belief set.

Proof. ($\Leftarrow$) First we show that a fuzzy belief set Γ is a fuzzy stable set. By definition of a fuzzy belief set we know that there exists a $S \subseteq \Omega_k$ such that $\Gamma(\varphi) = \|\mathrm{B}\varphi\|_S$ for each formula $\varphi \in \mathcal{L}^k_{\mathrm{B}}$. In order to show that $\hat{\Gamma}$ is propositionally consistent, by the strong completeness of Ł^c_k, it is sufficient to show that there exists a $v \in \Omega^*_k$ such that for each formula φ we have $\Gamma(\varphi) \leq v(\varphi^*)$. Indeed, it then follows that for each φ we have $v(\overline{\Gamma(\varphi)} \rightarrow \varphi^*) = 1$ and thus $v(\alpha) = 1$ for all $\alpha \in \Gamma$. Let $w \in S$ be arbitrary but fixed and define v such that $v(\varphi^*) = \|\varphi\|_{(w,S)}$ for each φ. It follows that

$$\Gamma(\varphi) = \inf_{z \in S} \|\varphi\|_{(z,S)} \leq \|\varphi\|_{(w,S)} = v(\varphi^*)$$

which proves (1). Next, assume that $\hat{\Gamma} \vdash \bar{c} \rightarrow \varphi^*$, or by the strong completeness of Ł^c_k that $\hat{\Gamma} \models \bar{c} \rightarrow \varphi^*$. We show that $\Gamma(\varphi) \geq c$. Note, similar as above, that for each $w \in S$ we have that v with $v(\psi^*) = \|\psi\|_{(w,S)}$ for all $\psi \in \mathcal{L}^k_{\mathrm{B}}$ is a model of $\hat{\Gamma}$ and hence of $\bar{c} \rightarrow \varphi^*$. Therefore $c \leq \|\varphi\|_{(w,S)}$ for each $w \in S$ and $c \leq \inf_{w \in S} \|\varphi\|_{(w,S)} = \Gamma(\varphi)$ which proves (2). Proving (3) follows easily by noting that

$$\Gamma(\varphi) = \|\mathrm{B}\varphi\|_S = \|\mathrm{BB}\varphi\|_S = \Gamma(\mathrm{B}\varphi).$$

Finally, to show (4), observe that $\Gamma({\sim}\mathrm{B}\varphi)$ equals

$$\|\mathrm{B}{\sim}\mathrm{B}\varphi\|_S = \inf_{w \in S} \|{\sim}\mathrm{B}\varphi\|_{(w,S)} = \inf_{w \in S}(1 - \|\mathrm{B}\varphi\|_{(w,S)}) = \inf_{w \in S}(1 - \Gamma(\varphi))$$

which is equal to $1 - \Gamma(\varphi)$.

($\Rightarrow$) Now, let Γ be a fuzzy stable set. Define

$$S^* = \{u \in \Omega_k^* \mid u(\alpha) = 1, \forall \alpha \in \hat{\Gamma}\}.$$

Note that S^* is non-empty by (1). We show that for each φ and $S = \{u_{|V} \mid u \in S^*\} \neq \emptyset$ with $u_{|V} : V \to S_k : p \mapsto u(p)$ we have

$$\Gamma(\varphi) = \|\mathrm{B}\varphi\|_S.$$

First, note that for a formula φ we have by definition of S^* that $w(\overline{\Gamma(\mathrm{B}\varphi)} \to (\mathrm{B}\varphi)^*) = 1$, or $w((\mathrm{B}\varphi)^*) \geq \Gamma(\mathrm{B}\varphi)$, for each $w \in S^*$. By (3) it then follows that $w((\mathrm{B}\varphi)^*) \geq \Gamma(\mathrm{B}\varphi) = \Gamma(\varphi)$ for each $w \in S^*$. We show that also $w((\mathrm{B}\varphi)^*) \leq \Gamma(\varphi)$ from which it then follows that

$$w((\mathrm{B}\varphi)^*) = \Gamma(\varphi)$$

for each $w \in S^*$. Indeed, since $\overline{\Gamma(\sim\mathrm{B}\varphi)} \to (\sim\mathrm{B}\varphi)^*$ is in $\hat{\Gamma}$ and thus $\Gamma(\sim\mathrm{B}\varphi) \leq w((\sim\mathrm{B}\varphi)^*) = w(\sim(\mathrm{B}\varphi)^*)$ we have

$$w((\mathrm{B}\varphi)^*) = 1 - w(\sim(\mathrm{B}\varphi)^*) \leq 1 - \Gamma(\sim\mathrm{B}\varphi).$$

Hence by (4) it follows that $w((\mathrm{B}\varphi)^*) \leq \Gamma(\varphi)$. We will now use the fact that

$$w((\mathrm{B}\varphi)^*) = \Gamma(\varphi)$$

for each $w \in S^*$ to show that for each formula α and each $w \in S^*$ we have $w(\alpha^*) = \|\alpha\|_{(w_{|V},S)}$ from which we can conclude that

$$\Gamma(\varphi) = w((\mathrm{B}\varphi)^*) = \|\mathrm{B}\varphi\|_{(w_{|V},S)} = \|\mathrm{B}\varphi\|_S$$

for each formula φ and an arbitrary $w \in S^*$. We show $w(\alpha^*) = \|\alpha\|_{(w_{|V},S)}$ by structural induction.The only notable case is where $\alpha = \mathrm{B}\psi$. By the definition of S^* we have $\Gamma(\psi) \leq u(\psi^*)$ for all $u \in S^*$ and hence

$$\Gamma(\psi) \leq \inf_{u \in S^*} u(\psi^*).$$

Now suppose that $\Gamma(\psi) < \inf_{u \in S^*} u(\psi^*)$, i.e. for each $u \in S^*$ we have $\Gamma(\psi) < u(\psi^*)$. Since the set of truth values is finite there exists a $v \in S^*$ such that $v(\psi^*) = \min_{u \in S^*} u(\psi^*)$. Hence for all $u \in S^*$ we have $v(\psi^*) \leq u(\psi*)$ and thus that $u(\overline{v(\psi^*)} \to \psi^*) = 1$. By the strong completeness of $Ł_k^c$ and the definition of S^* it follows that

$$\hat{\Gamma} \vdash \overline{v(\psi^*)} \to \psi^*.$$

By (2), it then follows that $\Gamma(\psi) \geq v(\psi^*)$. Since $v \in S^*$ it follows that $\Gamma(\psi) < v(\psi^*)$ and thus that $\Gamma(\psi) > \Gamma(\psi)$, a contradiction. Hence $\Gamma(\psi) = \inf_{u \in S^*} u(\psi^*)$ and by the induction hypothesis we conclude that

$$\begin{aligned} w(\alpha^*) = w((\mathrm{B}\psi)^*) = \Gamma(\psi) &= \inf_{u \in S^*} u(\psi^*) \\ &= \inf_{u \in S^*} \|\psi\|_{(u_{|V}, S)} = \inf_{z \in S} \|\psi\|_{(z,S)} = \|\alpha\|_{(w_{|V}, S)}. \end{aligned}$$

□

By combining Propositions 5.1 and 6.4, we can derive the following list of properties stable fuzzy expansions satisfy:

Proposition 6.5

Let A be a set of $\mathcal{L}_{\mathrm{B}}^k$-formulas. If E_A is a stable fuzzy expansion of A, then the following properties with $\underline{E_A}$ the alternative definition for a stable fuzzy expansion that was introduced previously in this section, hold:

(i) $\{\phi \in \mathcal{L}_{\mathrm{B}}^k \mid A^* \vdash \phi^*\} \subseteq \underline{E_A}$;
(ii) $\overline{r} \to \phi \in \underline{E_A}$ iff $E_A(\phi) \geq r$;
(iii) If $\overline{r} \to \phi \in \underline{E_A}$ and $\overline{s} \to \phi \notin \underline{E_A}$ for $s > r$, then $E_A(\phi) = r$;
(iv) $E_A(\phi) = E_A(\mathrm{B}\phi)$, and hence $\phi \in \underline{E_A}$ iff $\mathrm{B}\phi \in \underline{E_A}$;
(v) $1 - E_A(\phi) = E_A(\sim\mathrm{B}\phi)$.

Proof. First of all, by Propositions 5.1 and 6.4 it follows that E_A is a fuzzy stable set. Then, (i) is the trivial consequence of $\underline{E_A}$ being (propositionally) deductively closed and containing A, (ii) and (iii) directly follows from the definition of E_A, while (iv) and (v) are just translations from analogous properties of fuzzy stable sets from Proposition 6.4. □

6.4 "Only knowing" operators and stable fuzzy expansions

In this section we will extend the language of $\mathcal{L}_{\mathrm{B}}^k$ with a modal operator O (as well as an operator N) where a formula $\mathrm{O}\psi$ will be interpreted as "ψ is all that is believed". We denote this language by $\mathcal{L}_{\mathrm{O}}^{k}$. As discussed in Section 2.1.3, in the classical case [Levesque 1990], the semantics for a corresponding logic is defined as follows. Given an epistemic state S consisting of a set of classical evaluations, a formula $\mathrm{O}\psi$ is true in S when ψ is true in any structure (z, S) with $z \in S$, and false in any structure (z', S) with $z' \notin S$.

We can straightforwardly generalise this condition to the many-valued case by defining

$$\|\mathrm{O}\psi\|_{(w,S)} = \min(\inf_{z \in S} \|\psi\|_{(z,S)}, \inf_{z \notin S} \|{\sim}\psi\|_{(z,S)}),$$

where now $w \in \Omega_k$ and $S \subseteq \Omega_k$. Other formulas are evaluated as in fuzzy autoepistemic logic. If we then use the modal operator N whose truth evaluation in a pair (w, S) is

$$\|\mathrm{N}\psi\|_{(w,S)} = \inf_{z \notin S} \|\psi\|_{(z,S)},$$

then it is easy to see that the semantics of $\mathrm{O}\psi$ is exactly that of $\mathrm{B}\psi \wedge \mathrm{N}{\sim}\psi$. Notice that $\|\mathrm{N}\psi\|_{(w,S)} = \|\mathrm{B}\psi\|_{(w,\Omega_k \setminus S)}$. Hence by Theorem 6.3 it follows that N is another K45 "operator". Again, since the truth value of $\mathrm{O}\psi$ and $\mathrm{N}\psi$ in a structure (w, S) does not depend on w, we will also write $\|\mathrm{O}\psi\|_S$ and $\|\mathrm{N}\psi\|_S$ to denote $\|\mathrm{O}\psi\|_{(w,S)}$ and $\|\mathrm{N}\psi\|_{(w,S)}$ respectively.

To summarise, the language $\mathcal{L}_{\mathrm{O}}^{k}$ we consider is built as follows.

Definition 6.8

The language $\mathcal{L}_{\mathrm{O}}^{k}$ is recursively defined as follows

- $a \in A$ is a formula.
- $\overline{c}$ with $c \in S_k$ is a formula.
- If α is a formula, then $\mathrm{B}\alpha$ and $\mathrm{N}\alpha$ are formulas.
- If α and β are formulas, then $\alpha \rightarrow \beta$ with $\rightarrow$ the Łukasiewicz implication is a formula.

Truth is defined w.r.t. the structures in $\mathbb{M}^{\mathrm{ae}} = \{(w, S) \mid w \in \Omega_k, S \subseteq \Omega_k\}$ (Definition 6.5).

Definition 6.9

Truth evaluations for formulas in $\mathcal{L}_{\mathrm{O}}^{k}$ are recursively defined as follows:

- $\|a\|_{(v,S)} = v(a)$ for $a \in A$,
- $\|\overline{c}\|_{(v,S)} = c$ for truth constants in S_k,
- $\|\mathrm{B}\alpha\|_{(v,S)} = \inf_{w \in S} \|\alpha\|_{(w,S)}$ for $\alpha \in \mathcal{L}_{\mathrm{O}}^{k}$,
- $\|\mathrm{N}\alpha\|_{(v,S)} = \inf_{w \notin S} \|\alpha\|_{(w,S)}$ for $\alpha \in \mathcal{L}_{\mathrm{O}}^{k}$,

- $\|\phi \to \psi\|_{(v,S)} = \min(1, 1 - \|\phi\|_{(v,S)} + \|\psi\|_{(v,S)})$ for $\phi, \psi \in \mathcal{L}^k_{\mathrm{O}}$.

As in the classical case [Levesque 1990], we will provide a sound and complete axiomatisation for the fuzzy logic of "only knowing". This axiomatisation is similar to axiomatisations for multi-agent extensions of K45. The difference is that we have one agent but two separate modalities for belief and not two "separate believers". Morevover, we have a partition of the worlds in two subsets of worlds: some set S is used for the semantics of B and $\Omega_k \setminus S$ for N. Specifically, we have the following axioms.

(i) axioms of $Ł^c_k$,
(ii) axioms of *K45*($Ł^c_k$) for both B and N,
(iii) $\phi \to \mathrm{B}\phi$, where all variables and constants in ϕ occur in the scope of an operator N or B,
(iv) $\phi \to \mathrm{N}\phi$, where all variables and constants in ϕ occur in the scope of an operator N or B,
(v) $\sim\mathrm{B}\phi \vee \sim\mathrm{N}\phi$, if $\sim\phi$ is satisfiable and does not contain any modal operator,
(vi) $\mathrm{O}\phi \equiv \mathrm{B}\phi \wedge \mathrm{N}\sim\phi$.

The rules are modus ponens and necessitation for N and B. We will denote this logic as $\mathrm{O}(Ł^c_k)$.

First, note that all these axioms are tautologies in our fuzzy framework and rules preserve tautologies in every $(w, S) \in \mathbb{M}^{\mathrm{ae}}$. In particular, we have the following result.

Lemma 6.5

The axioms (i) - (vi) are sound w.r.t. the class of structures $\mathbb{M}^{\mathrm{ae}}$.

Proof. Axioms (i) and (ii) follow from previous results.

Axioms (iii) and (iv) are easy to check. It only has to be shown that these axioms are tautologies for formulas of the form $\phi = \mathrm{B}\alpha$ and $\phi = \mathrm{N}\alpha$. Indeed by truth functionality we then obtain that these axioms are tautologies for formulas where all variables and constants occur in the scope of an operator N or B. For axiom (iii), $(w, S) \in \mathbb{M}^{\mathrm{ae}}$ and $\phi = \mathrm{B}\alpha$ we have

$$\|\mathrm{B}\phi\|_{(w,S)} = \inf_{v \in S} \|\mathrm{B}\alpha\|_{(v,S)} = \inf_{z \in S} \|\alpha\|_{(z,S)} = \|\phi\|_{(w,S)}.$$

The other cases follow similarly.

For condition (v), suppose $\sim\phi$ is satisfiable, i.e. there exists $w^* \in \Omega^*_k$ such that $w^*(\phi) = 0$. For a structure $(w, S) \in \mathbb{M}^{\mathrm{ae}}$ we then have

$$\|\sim\mathrm{B}\phi \vee \sim\mathrm{N}\phi\|_{(w,S)} = 1$$

iff

$$\max(\|{\sim}\mathrm{B}\phi\|_{(w,S)}, \|{\sim}\mathrm{N}\phi\|_{(w,S)}) = 1$$

iff

$$\|\mathrm{B}\phi\|_{(w,S)} = 0 \text{ or } \|\mathrm{N}\phi\|_{(w,S)} = 0$$

iff there exists $z \in S$ such that $z(\phi) = \|\phi\|_{(z,S)} = 0$ or there exists $z \notin S$ such that $z(\phi) = \|\phi\|_{(z,S)} = 0$. This is satisfied by the fact that there exists $w' \in \Omega_k = S \cup (\Omega_k \setminus S)$ such that $w'(\phi) = 0$. Indeed let $w' = w^*_{|V}$. □

Similarly as in Section 6.2, we can show that there exists a reduction of the satisfiability problem for the fuzzy logic of only knowing to the classical counterpart. For every structure $(w, S) \in \mathbb{M}^{\mathrm{ae}}$ we then have $\|\Delta(\phi \leftrightarrow \overline{r})\|_{(w,S)} = 1$ iff $\|\phi\|_{(w,S)} = r$. Define $(\mathcal{L}^k_{\mathrm{O}})' \subseteq \mathcal{L}^k_{\mathrm{O}}$ as the set of formulas built from atomic propositions of the form $\{\Delta(p \leftrightarrow \overline{r}) \mid p \in V, r \in S_k\}$, Łukasiewicz connectives and operators B and N:

- $\Delta(p \leftrightarrow \overline{r}) \in (\mathcal{L}^k_{\mathrm{O}})'$ for every $p \in A$ and $r \in S_k$
- $(\phi \rightarrow \psi) \in (\mathcal{L}^k_{\mathrm{O}})'$ if $\phi, \psi \in (\mathcal{L}^k_{\mathrm{O}})'$
- $\mathrm{B}\phi \in (\mathcal{L}^k_{\mathrm{O}})'$ if $\phi \in (\mathcal{L}^k_{\mathrm{O}})'$
- $\mathrm{N}\phi \in (\mathcal{L}^k_{\mathrm{O}})'$ if $\phi \in (\mathcal{L}^k_{\mathrm{O}})'$

Lemma 6.6

Given a formula $\phi \in \mathcal{L}^k_{\mathrm{O}}$ and a truth value $r \in S_k$, there exists a formula $(\phi_r)' \in (\mathcal{L}^k_{\mathrm{O}})'$ such that for each structure $(w, S) \in \mathbb{M}^{\mathrm{ae}}$ it holds that

$$\|\phi\|_{(w,S)} = r \text{ iff } \|(\phi_r)'\|_{(w,S)} = 1,$$

where ϕ_r is a short notation for the formula $\Delta(\phi \leftrightarrow \overline{r})$ with $\phi \in \mathcal{L}^k_{\mathrm{O}}$.

Proof. This lemma can be shown in a similar way as in Lemma 6.3. In particular the claim now also has to be checked for formulas of the form $\mathrm{N}\phi$ which can be done entirely analogously as for formulas of the form $\mathrm{B}\phi$. □

We will use this lemma together with the following proposition to show that the proposed axiomatisation is complete w.r.t. the proposed semantics. In Proposition 6.6 and Theorem 6.4, besides the languages $\mathcal{L}^k_{\mathrm{O}}$ and $(\mathcal{L}^k_{\mathrm{O}})'$ and structures $\mathbb{M}^{\mathrm{ae}}$, we also consider the following languages:

- $(\mathcal{L}_{\mathrm{O}}^k)^+$, which is an extension of $\mathcal{L}_{\mathrm{O}}^k$ with an additional set of variables $\hat{V} = \{p_r \mid p \in V, r \in S_k\}$, i.e. the language built from variables $V \cup \hat{V}$, truth constants from S_k, the Łukasiewicz implication and operators B and N;
- $(\mathcal{L}_{\mathrm{O}}^k)^C$, built from variables $\hat{V}$, classical connectives ($\wedge$, $\vee$, $\sim$, $\rightarrow$) and operators B and N

and the following classes of semantic structures:

- $\mathbb{M}_+^{\mathrm{ae}} = \{(w, S) \mid w \in \Omega_k^+, S \subseteq \Omega_k^+\}$ where Ω_k^+ is the set of all evaluations $w : V \cup \hat{V} \rightarrow S_k$,
- $\mathbb{M}_C^{\mathrm{ae}} = \{(w, S) \mid w \in \Omega_k^C, S \subseteq \Omega_k^C\}$ where Ω_k^C is the set of all evaluations $w : \hat{V} \rightarrow \{0, 1\}$

Proposition 6.6

Suppose $\phi \in (\mathcal{L}_{\mathrm{O}}^k)'$ is a tautology w.r.t. $\mathbb{M}^{\mathrm{ae}}$, then ϕ is provable in $\mathrm{O}(\text{Ł}_k^c)$.

Proof. Suppose $\phi \in (\mathcal{L}_{\mathrm{O}}^k)'$ is a tautology w.r.t. $\mathbb{M}^{\mathrm{ae}}$. Define ϕ'' as the formula in $(\mathcal{L}_{\mathrm{O}}^k)^+$ obtained by replacing each subformula $\Delta(p \leftrightarrow \overline{r})$ by the atom $p_r \in \hat{V}$. We show that $\Psi'' \rightarrow \phi''$, with

$$\Psi'' = \Delta \left[\bigwedge_{p \in V, r \in S_k} (\Delta(p \leftrightarrow \overline{r}) \leftrightarrow p_r) \right],$$

is a tautology in $\mathbb{M}_+^{\mathrm{ae}}$. Indeed, consider a structure (w, S) in $\mathbb{M}_+^{\mathrm{ae}}$. Note that $\|\Psi''\|_{(w,S)} = 1$ if $\|\Delta(p \leftrightarrow \overline{r})\|_{(w,S)} = \|p_r\|_{(w,S)}$ for all $p \in V$ and $\|\Psi''\|_{(w,S)} = 0$ otherwise. If $\|\Psi''\|_{(w,S)} = 1$, then $\|\phi''\|_{(w,S)} = 1$ since ϕ is a tautology in $\mathbb{M}^{\mathrm{ae}}$ and in this case we have $\|\Delta(p \leftrightarrow \overline{r})\|_{(w,S)} = \|p_r\|_{(w,S)}$ for all $p \in V$. If $\|\Psi''\|_{(w,S)} = 0$, then trivially $\|\Psi''\|_{(w,S)} \leq \|\phi''\|_{(w,S)}$. Hence, in both cases we obtain

$$\|\Psi'' \rightarrow \phi''\|_{(w,S)} = 1.$$

Next we show that $\Sigma'' \rightarrow \phi''$, with

$$\Sigma'' = \bigwedge_{p \in V} \left(\bigvee_{r \in S_k} p_r \ \wedge \bigwedge_{s,t \in S_k, s \neq t} \sim(p_r \wedge p_s) \right),$$

is a tautology in $\mathbb{M}_C^{\mathrm{ae}}$. Indeed, consider a structure $(\hat{w}, \hat{S})$ in $\mathbb{M}_C^{\mathrm{ae}}$. We show that $\|\Sigma''\|_{(\hat{w},\hat{S})} = 1$ implies $\|\phi''\|_{(\hat{w},\hat{S})} = 1$. Suppose that $\|\Sigma''\|_{(\hat{w},\hat{S})} = 1$. By the definition of Σ'', it then holds that for each $p \in V$ there exists exactly one $r_p \in S_k$ such that

$\hat{w}(p_{r_p}) = \|p_{r_p}\|_{(\hat{w},\hat{S})} = 1$. We will now show that there exists a $(w, S) \in \mathbb{M}^{\text{ae}}_{+}$ such that $\|\phi''\|_{(\hat{w},\hat{S})} = \|\phi''\|_{(w,S)}$ and such that $\|\Psi''\|_{(w,S)} = 1$. Since $\Psi'' \to \phi''$ is a tautology in $\mathbb{M}^{\text{ae}}_{+}$ we then obtain that $\|\phi''\|_{(\hat{w},\hat{S})} = 1$. To do this, we first define a mapping $f : \Omega^C_k \to \Omega^+_k$ as follows. For $\hat{v} \in \Omega^C_k$, let $f(\hat{v}) : V \cup \hat{V} \to S_k$ be the evaluation such that $f(\hat{v})(p_r) = \hat{v}(p_r)$ for all $p_r \in \hat{V}$ and for each $p \in V$ let $f(\hat{v})(p) = r_p$ where r_p is the unique element in S_k such that $\hat{v}(p_{r_p}) = \|p_{r_p}\|_{(\hat{v},\hat{S})} = 1$. We define $(w, S) \in \mathbb{M}^{\text{ae}}_{+}$ as follows: $w = f(\hat{w})$ and $S = \{f(\hat{v}) \mid \hat{v} \in \hat{S}\}$. It then holds that $\|\Psi''\|_{(w,S)} = 1$ and $\|\phi''\|_{(\hat{w},\hat{S})} = \|\phi''\|_{(w,S)}$.

Since $\Sigma'' \to \phi''$ is a tautology in $\mathbb{M}^{\text{ae}}_C$ and the classical logic of only knowing is sound and complete w.r.t. $\mathbb{M}^{\text{ae}}_C$ [Levesque 1990], there is a corresponding proof Γ for $\Sigma'' \to \phi''$. We will transform this proof to a proof for ϕ' in $\mathrm{O}(\text{Ł}^c_k)$. First note that Σ', which is obtained from Σ'' by replacing each p_r by $\Delta(p \leftrightarrow \overline{r})$, is a theorem in Ł^c_k and hence also a theorem in $\mathrm{O}(\text{Ł}^c_k)$. By modus ponens, it is now sufficient to show that $\Sigma' \to \phi'$ is a theorem as well. This follows trivially by substituting in every formula in Γ expressions of the form p_r by $\Delta(p \leftrightarrow \overline{r})$. $\square$

In the following theorem we will show completeness for $\mathrm{O}(\text{Ł}^c_k)$ w.r.t. $\mathbb{M}^{\text{ae}}$.

Theorem 6.4

Every tautology in $\mathbb{M}^{\text{ae}}$ is a theorem in $\mathrm{O}(\text{Ł}^c_k)$.

Proof. Suppose ϕ is a tautology in $\mathbb{M}^{\text{ae}}$. Trivially, $\phi_1 = \Delta\phi$ is a tautology as well and by Lemma 6.6, it follows that $(\phi_1)' \in (\mathcal{L}^k_{\mathrm{O}})'$ constructed analogously as in Lemma 6.3 is also a tautology. By Proposition 6.6 it then follows that there is a proof for $(\phi_1)'$ in $\mathrm{O}(\text{Ł}^c_k)$. We will now show that this implies that there also exists a proof for ϕ in $\mathrm{O}(\text{Ł}^c_k)$. We will do this by showing that for each $\varphi \in \mathcal{L}^k_{\mathrm{O}}$ and for each $r \in S_k$ it holds that $\varphi_r = \Delta(\varphi \leftrightarrow \overline{r})$ is provably equivalent to $(\varphi_r)'$ as constructed in Lemma 6.3, i.e. that $\mathrm{O}(\text{Ł}^c_k)$ proves $\varphi_r \leftrightarrow (\varphi_r)'$. In particular, this implies that $(\phi_1)'$ and $\phi_1 = \Delta\phi$ are provably equivalent formulas. Since by Proposition 6.6 $(\phi_1)'$ is a theorem in $\mathrm{O}(\text{Ł}^c_k)$ it then follows that $\Delta\phi$ and hence ϕ is a theorem as well.

Let us show by induction that for each $\varphi \in \mathcal{L}^k_{\mathrm{O}}$ and for each $r \in S_k$ it holds that $\varphi_r = \Delta(\varphi \leftrightarrow \overline{r})$ is provably equivalent to $(\varphi_r)'$. The only non trivial step in this proof is to show that there exists a $(\mathrm{B}\alpha_r)' \in (\mathcal{L}^k_{\mathrm{O}})'$ provably equivalent to $\Delta(\mathrm{B}\alpha \leftrightarrow \overline{r})$ given that the claim holds for α. The case of N can be proved analogously. First note that if $r < 1$,

$Ł_k^c$ proves that $\Delta(B\alpha \leftrightarrow \overline{r})$ is equivalent to

$$\left(\bigvee_{t\geq r} \Delta(B\alpha \leftrightarrow \overline{t})\right) \wedge \left(\sim \bigvee_{t\geq r^+} \Delta(B\alpha \leftrightarrow \overline{t})\right)$$

where r^+ is the successor of r. Now using axioms $(B2)-(B4)$ and the fact that $B(\varphi\otimes\varphi) \leftrightarrow B\varphi \otimes B\varphi$, and hence $B\Delta\varphi \leftrightarrow \Delta B\varphi$ is a theorem of *K45*($Ł_k^c$), the following expression can be derived from $\Delta(B\alpha \leftrightarrow \overline{r})$.

$$B\left(\bigvee_{t\geq r} \Delta(\alpha \leftrightarrow \overline{t})\right) \wedge {\sim}B\left(\bigvee_{t\geq r^+} \Delta(\alpha \leftrightarrow \overline{t})\right).$$

By the induction hypothesis, there exists for each $t \geq r$ a formula $(\alpha_t)' \in (\mathcal{L}_O^k)'$ that is equivalent to $\Delta(\alpha \leftrightarrow \overline{t})$. Therefore, $\Delta(B\alpha \leftrightarrow \overline{r})$ is equivalent to

$$B\left(\bigvee_{t\geq r} (\alpha_t)'\right) \wedge {\sim}B\left(\bigvee_{t\geq r^+} (\alpha_t)'\right) \in (\mathcal{L}_O^k)'.$$

If, on the other hand $r = 1$, then we have to show that $\Delta B\alpha$ is equivalent to $B\Delta\alpha$. As previously mentioned, this is a theorem of *KD45*($Ł_k^c$).

□

Finally, we show that the relationship between the "only knowing" operator O and Moore's stable expansions proved in [Levesque 1990] naturally extends to our framework. The next proposition shows that the belief set Th(S) (Definition 5.5) for an epistemic state defined by a set of $Ł_k$-evaluations S is indeed a stable fuzzy expansion of a premise φ whenever $\Delta\varphi$ is all what is fully believed in the epistemic state S.

This proposition can easily be generalised to stable fuzzy expansions of sets of formulas.

Proposition 6.7

Suppose $A = \{\phi_1, \ldots, \phi_n\}$ is a set of formulas in $\mathcal{L}_B^k$. Then Th(S) is a stable fuzzy expansion of A iff $\|O(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n)\|_S = 1$.

Proof. Since for a formula ψ we have $\|O\psi\|_S = \min(\|B\psi\|_S, \|N({\sim}\psi)\|_S)$, we have the following chain of equivalences:

$\|\mathrm{O}(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n)\|_S = 1$
$\Leftrightarrow \|\mathrm{B}(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n)\|_S = 1$ and $\|\mathrm{N}(\sim(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n))\|_S = 1$
$\Leftrightarrow \forall v \in S : \|\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n\|_{(v,S)} = 1$ and $\forall v \notin S : \|\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n\|_{(v,S)} = 0$
$\Leftrightarrow \forall v \in S, \forall i \in \{1, \ldots, n\} : \|\Delta\phi_i\|_{(v,S)} = 1$ and $\forall v \notin S, \exists j \in \{1, \ldots, n\} :$
$\quad \|\Delta\phi_j\|_{(v,S)} = 0$
$\Leftrightarrow \forall v \in S, \forall i \in \{1, \ldots, n\} : \|\phi_i\|_{(v,S)} = 1$ and $\forall v \notin S, \exists j \in \{1, \ldots, n\} : \|\phi_j\|_{(v,S)} < 1$

Thus, assuming that

$$\|\mathrm{O}(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n)\|_S = 1,$$

we can show that

$$S = \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1 \text{ for all } i \in \{1, \ldots, n\}\}.$$

Indeed, by the chain of equivalences we have

$$S \subseteq \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1 \text{ for all } i \in \{1, \ldots, n\}\}.$$

If $w \in \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1$ for all $i \in \{1, \ldots, n\}\}$ and $w \notin S$, then there would exists a $j \in \{1, \ldots, n\}$ such that $\|\phi_j\|_{(w,S)} < 1$, a contradiction. Hence

$$S = \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1 \text{ for all } i \in \{1, \ldots, n\}\}.$$

Similar, one can also show that

$$S = \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1 \text{ for all } i \in \{1, \ldots, n\}\}$$

implies that for all $v \in S$ we have $\|\phi_i\|_{(v,S)} = 1$ for all $i \in \{1, \ldots, n\}$ and for all $v \notin S$ we have that there exists a $j \in \{1, \ldots, n\}$ such that $\|\phi_j\|_{(v,S)} < 1$. Hence

$\|\mathrm{O}(\Delta\phi_1 \wedge \ldots \wedge \Delta\phi_n)\|_S = 1$
$\Leftrightarrow S = \{v \in \Omega_k \mid \|\phi_i\|_{(v,S)} = 1$ for all $i \in \{1, \ldots, n\}\}$
$\Leftrightarrow S$ is a fuzzy possible world autoepistemic model of A
$\Leftrightarrow$ Th(S) is a stable fuzzy expansion of A.

where the last equivalence follows from Proposition 5.1. □

Example 6.4

Recall the sensor network example from Section 5.3. We will now present an alternative method to find the stable fuzzy expansions.

Let t_i be the variable representing the temperature measured by sensor i. Suppose we have an appropriate rescaling to assure that all variables take values in Ω_k and let e_i be the variable representing the degree to which sensor i is faulty.

Let us define a new connective $d(\varphi, \psi)$ as $\sim(\varphi \leftrightarrow \psi)$ for which the semantics is given by the Euclidean distance $\hat{d}$: for all $x, y \in [0,1]$, $\hat{d}(x,y) = |x - y|$. The connective d is well known in the literature of many-valued logics and it is usually called *Chang distance function* [Cignoli et al. 2000]. The fact that d can be defined in a many-valued logical setting is a peculiarity of MV-algebras and also for this reason we believe these structures to be a suitable algebraic setting.

Then, for $i \neq j$, the formulas

$$d(t_i, t_j) \rightarrow (e_i \vee e_j)$$

capture the idea that if the sensed values of two sensors t_i and t_j are different, this provides a reason to believe that at least one of those two sensors is faulty. Note that we assume that all sensors are physically close to each other and thus they should measure the same temperature.

On the other hand, the formula

$$\sim\mathrm{B}e_i \rightarrow \sim e_i$$

represents the default information that sensors are not faulty.

As a concrete case, we further assume we are told sensor e_2 is very reliable, and hence we believe that sensor e_2 is not faulty, i.e. $\mathrm{B}(\sim e_2)$, and that the three measured values are $t'_1 = 0.2, t'_2 = 0.9, t'_3 = 0.5$.

The formalisation of this scenario then amounts to consider the following theory:

$$A = \{\sim\mathrm{B}e_1 \rightarrow \sim e_1, \sim\mathrm{B}e_2 \rightarrow \sim e_2, \sim\mathrm{B}e_3 \rightarrow \sim e_3, \mathrm{B}(\sim e_2), \overline{0.7} \rightarrow (e_1 \vee e_2), \\ \overline{0.4} \rightarrow (e_2 \vee e_3), \overline{0.3} \rightarrow (e_1 \vee e_3)\}$$

Let us define the formula ϕ_A as follows

$$\phi_A = \Delta(\mathrm{B}(\sim e_2)) \wedge \Delta(\sim\mathrm{B}e_1 \rightarrow \sim e_1) \wedge \Delta(\sim\mathrm{B}e_2 \rightarrow \sim e_2) \wedge \Delta(\sim\mathrm{B}e_3 \rightarrow \sim e_3) \wedge \\ \Delta(\overline{0.7} \rightarrow (e_1 \vee e_2)) \wedge \Delta(\overline{0.4} \rightarrow (e_2 \vee e_3)) \wedge \Delta(\overline{0.3} \rightarrow (e_1 \vee e_3))$$

Let us now use Proposition 6.7 to find the stable fuzzy expansion of A. Suppose there exists a fuzzy possible world autoepistemic model S of A, then by Proposition 6.7 it should hold that $\|\mathrm{O}\phi_A\|_S = 1$. By the semantics of the operator O, it follows that

$$\|\mathrm{B}\left(\Delta(\sim\mathrm{B}e_1 \rightarrow \sim e_1) \wedge \Delta(\sim\mathrm{B}e_2 \rightarrow \sim e_2) \wedge \Delta(\sim\mathrm{B}e_3 \rightarrow \sim e_3) \wedge \Delta(\mathrm{B}(\sim e_2)) \wedge \\ \Delta(\overline{0.7} \rightarrow (e_1 \vee e_2)) \wedge \Delta(\overline{0.4} \rightarrow (e_2 \vee e_3)) \wedge \Delta(\overline{0.3} \rightarrow (e_1 \vee e_3))\right)\|_S = 1$$

and hence by the semantics of the operator B we obtain for all $v \in S$ and for all $\phi \in A$ that $\|\Delta\phi\|_{(v,S)} = 1$ and hence that $\|\phi\|_{(v,S)} = 1$. For every $v \in S$ it must hold that $\|\mathrm{B}(\sim e_2)\|_{(v,S)} = 1$ and thus that $\inf_{w\in S}(1 - w(e_2)) = 1$, i.e. that for all $w \in S$ we have $w(e_2) = 0$. Moreover since $\|\sim\mathrm{B}e_i\|_{(v,S)} \leq \|\sim e_i\|_{(v,S)}$ it follows that

$$v(e_i) \leq \inf_{w\in S} w(e_i) \leq v(e_i)$$

for $i \in \{1, 2, 3\}$. If there exists $v_1, v_2 \in S$, then we have that $v_1(e_i) = \inf_{w\in S} w(e_i) = v_2(e_i)$ for all $i \in \{1, 2, 3\}$. It follows that $S = \{v\}$ is a singleton such that $v(e_2) = 0$. Taking into account the remaining formulas in A we obtain that it must hold that $0.7 \leq v(e_1)$, $0.4 \leq v(e_3)$ and $0.3 \leq \max(v(e_1), v(e_3))$. Actually, we can check that v is the minimal element in Ω_k such that these inequalities are satisfied, i.e. $v(e_1) = 0.7$, $v(e_2) = 0$ and $v(e_3) = 0.4$. Indeed, let $z \in \Omega_k$ such that $z \leq v$ and such that z satisfies the above inequalities. Then we obtain $z(e_i) \leq v(e_i) = \inf_{w\in S} w(e_i)$ and thus $\|\sim\mathrm{B}e_i \to \sim e_i\|_{(z,S)} = \|e_i \to \mathrm{B}e_i\|_{(z,S)} = 1$ for $i \in \{1, 2, 3\}$. This implies that $z \in S$, a contradiction if $z \neq v$. Hence if there exists a fuzzy possible world autoepistemic model S of A it has to be

$$S = \{v \in \Omega_k \mid v(e_1) = 0.7, v(e_2) = 0, v(e_3) = 0.4\}.$$

To obtain that S is a fuzzy possible world autoepistemic model of A, it remains to be shown that

$$\begin{aligned}&\|\mathrm{N}\!\sim(\Delta(\sim\mathrm{B}e_1 \to \sim e_1) \wedge \Delta(\sim\mathrm{B}e_2 \to \sim e_2) \wedge \Delta(\sim\mathrm{B}e_3 \to \sim e_3) \wedge \Delta(\mathrm{B}(\sim e_2))\wedge\\ &\Delta(\overline{0.7} \to (e_1 \vee e_2)) \wedge \Delta(\overline{0.4} \to (e_2 \vee e_3)) \wedge \Delta(\overline{0.3} \to (e_1 \vee e_3)))\,\|_S = 1\end{aligned}$$

Or in other words, it has to be shown that for all $w \notin S$ there exists $\phi \in A$ such that $\|\phi\|_{(w,S)} < 1$. Suppose this is not the case and there exists $w \notin S$ such that for all $\phi \in A$ it holds that $\|\phi\|_{(w,S)} = 1$. Since for all $i \in \{1, 2, 3\}$ we have $\|\sim\mathrm{B}e_i \to \sim e_i\|_{(w,S)} = 1$ it follows that $w(e_i) \leq \inf_{z\in S} z(e_i) = v(e_i)$. Since $w \neq v$ there exists $j \in \{1, 3\}$ such that $w(e_j) < v(e_j)$ and $w(e_2) = 0$. If $j = 1$, since $\|\overline{0.7} \to e_1 \vee e_2\|_{w,S} = 1$, we obtain

$$w(e_1) < v(e_1) = 0.7 \leq w(e_1 \vee e_2) = w(e_1),$$

a contradiction. For $j = 3$ we obtain a similar contradiction.

The next proposition proves interesting properties about graded beliefs the only knowing operator O captures inside the many-valued modal logic $\mathrm{O}(\text{Ł}_k^c)$. These are related to similar features of the fuzzy autoepistemic logic. We recall that a propositional formula φ is r-satisfiable, for $r \in S_k$, if there exists an evaluation $w \in \Omega_k$ such that $w(\varphi) \geq r$.

Proposition 6.8

If φ is a propositional (B-free) formula which is r-satisfiable for some $r \in S_k$, then $\mathrm{O}(\mathrm{Ł}_k^c)$ proves the following formulas:

(i) $\mathrm{N}(\sim\Delta(\overline{r} \to \varphi)) \to (\mathrm{B}\varphi \to \overline{r})$
(ii) $\mathrm{O}(\Delta(\overline{r} \to \varphi)) \to (\mathrm{B}\varphi \leftrightarrow \overline{r})$

Proof. Let $\phi = \Delta(\overline{r} \leftrightarrow \varphi)$.

(i) Obviously, φ is r-satisfiable iff ϕ is 1-satisfiable, and since ϕ is Boolean, by axiom (v) of the logic $\mathrm{O}(\mathrm{Ł}_k^c)$ we have that

$$\mathrm{N}(\sim\Delta(\overline{r} \leftrightarrow \varphi)) \to \sim\mathrm{B}\sim\Delta(r \leftrightarrow \varphi)$$

is a theorem. Since $\Delta(r \leftrightarrow \varphi) \to (\varphi \to \overline{r})$ is a theorem in $\mathrm{Ł}_k^c$ we can derive that the following formula is a theorem in $\mathrm{O}(\mathrm{Ł}_k^c)$:

$$\sim\mathrm{B}(\sim\Delta(r \leftrightarrow \varphi)) \to \sim\mathrm{B}(\sim(\varphi \to \overline{r}))$$

Hence $\mathrm{O}(\mathrm{Ł}_k^c)$ proves

$$\mathrm{N}(\sim\Delta(\overline{r} \leftrightarrow \varphi)) \to \sim\mathrm{B}(\sim(\varphi \to \overline{r})).$$

Since by axiom (B3),

$$\sim\mathrm{B}(\sim(\varphi \to \overline{r})) \to (\mathrm{B}\varphi \to \overline{r})$$

is a theorem, (i) is proved.

(ii) follows by observing that $\mathrm{O}\phi$ is equivalent to $\mathrm{B}\phi \wedge \mathrm{N}\sim\phi$, and hence $\mathrm{O}\phi \to \mathrm{B}\phi$, i.e.

$$\mathrm{O}\Delta(r \leftrightarrow \varphi) \to \mathrm{B}\Delta(r \leftrightarrow \varphi)$$

is a theorem. Since $\Delta(r \leftrightarrow \varphi) \to (\overline{r} \to \varphi)$ is a theorem in $\mathrm{Ł}_k^c$ we obtain that

$$\mathrm{B}(\Delta(r \leftrightarrow \varphi)) \to \mathrm{B}(\overline{r} \to \varphi)$$

is a theorem in $\mathrm{O}(\mathrm{Ł}_k^c)$. Since $\mathrm{B}(\overline{r} \to \varphi)$ is equivalent to $\overline{r} \to \mathrm{B}\varphi$ by axiom (B3), we obtain that

$$\mathrm{B}(\Delta(r \leftrightarrow \varphi)) \to (\overline{r} \to \mathrm{B}\varphi)$$

is a theorem in $\mathrm{O}(\mathrm{Ł}_k^c)$. Hence we obtain that $\mathrm{O}(\mathrm{Ł}_k^c)$ proves

$$\mathrm{O}(\Delta(\overline{r} \to \varphi)) \to (\mathrm{B}\varphi \leftrightarrow \overline{r}).$$

□

6.5 Concluding remarks and related work

In this chapter we have introduced Hilbert-style axiomatisations for fuzzy modal logics of belief as well as for a "logic of only knowing" based on finitely-valued Łukasiewicz logic with truth constants. In particular, we have introduced generalisations of the main classical propositional modal logics of belief (K45, KD45, S5) in order to model the notion of belief for fuzzy propositions, in the sense of admitting partial degrees of truth between 0 (fully false) and 1 (fully true). We have shown that their Kripke style semantics can also be used to characterise fuzzy autoepistemic logic using a possible world semantics, in line with the original work of Moore [Moore 1984]. We have also developed a fuzzy version of Levesque's (propositional) logic of "only knowing", proving soundness and completeness, and in particular we have generalised bridges with autoepistemic logic established in [Levesque 1990] for the classical case by characterising stable fuzzy expansions in terms of models of suitable "only knowing" formulas.

Conclusions

In this thesis we have thoroughly studied nonmonotonic reasoning in many-valued logic and provided a theoretical basis for fuzzy answer set programming. This was accomplished by investigating the foundations of nonmonotonic reasoning when properties may be graded. In particular, we have defined graded generalisations of autoepistemic logic and logic of only knowing and discussed relationships among them and with fuzzy answer set programming. To do this we have also introduced fuzzy modal logics of belief.

Answer set programming (ASP) is a declarative programming language that allows us to model combinatorial optimisations problems in a concise and elegant way. To solve a problem, we translate it to an ASP program and particular models of the program, the answer sets, then correspond to the solutions of the original program. An important component of ASP is the negation-as-failure operator "not" providing a framework for nonmonotonic reasoning.

While ASP provides a rich language, it is not directly suitable for modelling problems with continuous domains. Hence in Section 3.2 we defined a particular – but very general – form of fuzzy answer set programming (FASP) as a combination of ASP and fuzzy logic. In Section 3.3 we have presented some motivating and illustrating examples for FASP. The first example showed how strict disjunctive FASP can be used to model sensor networks. This was followed by an example showing how strict simple FASP can be used to compute transitive closures of proximity relations. Finally, we defined two regular normal FASP programs: one for tackling a version of the ATM location selection problem and a second one that can be used to solve the fuzzy graph colouring problem.

In Chapter 4, we presented an overview of the computational complexity of FASP under Łukasiewicz semantics, a fuzzy logic discussed in Section 2.2.3. In particular we discussed the following decision problems: Given a FASP program P, a literal l and a value $\lambda_l \in [0,1] \cap \mathbb{Q}$:

1. **Existence**: Does there exist an answer set I of P?
2. **Set-membership**: Does there exist an answer set I of P such that $I(l) \geq \lambda_l$?
3. **Set-entailment**: Does $I(l) \geq \lambda_l$ hold for each answer set I of P?

For the programs with the most syntactic freedom, i.e. regular FASP programs, we showed Σ_2^{P}-completeness for set-membership and existence and Π_2^{P}-completeness for set-entailment by using known complexity results about fuzzy equilibrium logic [Schockaert et al. 2012]. However, if we restrict ourselves to programs with at most one literal in the head of each rule, then we could only show Σ_2^{P}-membership and NP-hardness for set-membership and existence and Π_2^{P}-membership and coNP-hardness for set-entailment. If in addition, we do not allow "not" in the rules we could only find a pseudo-polynomial time algorithm to compute answer sets based on computing least fixpoints. If we restrict to regular definite FASP programs, we could only show membership in NP ∩ coNP, but for several subclasses we can show P-membership. In particular, for regular definite FASP programs with only conjunction and maximum or only disjunction in the body of rules we have provided a polynomial time algorithm to compute answer sets. This is also the case for regular definite FASP programs with a cycle free dependency graph or with only polynomially bounded constants.

Although existence and set-membership are Σ_2^{P}-complete for disjunctive ASP, for strict disjunctive and strict normal FASP we were able to show NP-completeness. Moreover, we showed that not allowing constraints and strong negation does not affect the complexity for set-membership. We showed that the complexity of the existence problem for this class of strict normal FASP programs without constraints and without strong negation is "constant" since the existence of an answer set for such a program is always guaranteed. However, for strict disjunctive FASP programs without constraints and without strong negation we were only able to show membership in NP for the existence problem.

An overview of the complexity results that we have established can be found in Tables 4.1 and 4.2. Finally, we have also proposed an implementation of strict disjunctive FASP using bilevel linear programming, providing a basis to build solvers for FASP. However, some open problems remain: the existence of a polynomial time algorithm to compute the answer set of a regular simple FASP program, the complexity of the decision problems for regular normal FASP and existence for strict disjunctive FASP if constraints and strong negation are not allowed.

In Chapter 5, we considered a more general form of FASP programs, in which the connectives can in principal be interpreted by arbitrary mappings not restricted to the connectives of a particular fuzzy logic. We introduced a fuzzy version of autoepistemic logic which can be used to reason about one's (lack of) beliefs about the degrees to which properties are satisfied. In this chapter, we have shown that, when generalising to the many-valued case, important properties of classical autoepistemic logic are preserved and that the relation between answer set programming and autoepistemic logic remains valid. Moreover, we have presented two different but equivalent characterisations of answer sets in fuzzy autoepistemic logic and in a fuzzy logic of minimal belief and negation-as-failure. These results lead to a better comprehension of how to interpret fuzzy answer sets. Since the language of fuzzy autoepistemic logic is much more expressive than the theories we need to represent the fuzzy answer set programs, this could also serve as a useful basis for defining or comparing extensions to the basic language of FASP.

Finally, in Chapter 6 we discussed relationships between fuzzy autoepistemic logic and fuzzy modal logics, generalising well-known links beween autoepistemic logic and several classical modal logic systems. In particular we have generalised Levesque's logic of only knowing and showed that when generalising to the many-valued case the correspondence to autoepistemic logic remains valid. Moreover we have provided a sound and complete axiomatisation for this logic of only knowing based on finitely-valued Łukasiewicz logic with truth constants. To obtain this axiomatisation, we have introduced generalisations of the main classical propositional modal logics of belief (K45, KD45, S5). We have shown that their Kripke style semantics are closely related to fuzzy autoepistemic logic. Moreover, we have generalised correspondences with autoepistemic logic by characterising stable fuzzy expansions in terms of models of suitable "only knowing" formulas.

Overall we have presented many results that enhance our understanding of the complexity of FASP and its relationship to other forms of nonmonotonic reasoning with degrees, but along the way also new questions have emerged that would be interesting avenues of future work. To conclude this dissertation, below we list those questions and problems that we consider the most interesting and intriguing:

- Studying the remaining open problems w.r.t. the complexity of FASP under Łukasiewicz semantics:
  - Providing a polynomial time algorithm to compute the answer set of a regular simple FASP program or showing that such an algorithm cannot exist.
  - Investigating whether existence for strict disjunctive FASP is NP-hard when constraints and strong negation are not allowed.
  - Investigating whether the decision problems for regular normal FASP are in NP (or co-NP for set-entailment).

- Establishing the computational complexity of FASP under product logic and Gödel logic.
- Studying the implementation of fuzzy answer set programming and fuzzy autoepistemic logic using multi-level linear programming, generalising the implementation of classical autoepistemic logic using Quantified Boolean Formulas.
- Investigating fuzzy autoepistemic logic and fuzzy logic of only knowing based on infinitely-valued Łukasiewicz logic with truth constants, as well as using semantics of other fuzzy logics.

Samenvatting

In deze thesis hebben we een grondige studie van niet-monotoon redeneren in meerwaardige logica's uitgevoerd en hebben we een theoretische basis gelegd voor vaag answer set programmeren. Hiertoe hebben we de fundamenten onderzocht van niet-monotoon redeneren wanneer eigenschappen gradueel kunnen zijn. In het bijzonder hebben we meerwaardige veralgemeningen van autoepistemische logica en van logica van "only knowing" geïntroduceerd en hebben we relaties tussen deze logica's onderling en relaties met vaag answer set programmeren bestudeerd. Om dit te doen hebben we ook modale vaaglogica's van geloof geïntroduceerd.

Answer set programmeren (ASP) is een declaratieve programmeertaal die ons toelaat om combinatorische optimalisatieproblemen te modelleren op een beknopte en elegante manier. Om zo'n probleem op te lossen vertalen we het eerst naar een ASP programma. Bepaalde modellen van het programma, de answer sets, komen dan overeen met de oplossingen van het oorspronkelijke probleem. Een belangrijke component van ASP is de negatie-door-falen operator "not" die een kader voor niet-monotoon redeneren schept.

Hoewel ASP een rijke taal biedt, leent het zich niet onmiddellijk tot het modelleren van problemen met continue domeinen. In Sectie 3.2 hebben we daarom een bepaalde – heel algemene – vorm van vaag answer set programmeren (FASP) gedefinieerd, die ASP en vaaglogica combineert. In Sectie 3.3 hebben we vervolgens enkele motiverende voorbeelden voor FASP gepresenteerd. Het eerste voorbeeld illustreerde hoe FASP gebruikt kan worden om sensornetwerken te modelleren. Dit werd gevolgd door een voorbeeld dat toonde hoe we FASP kunnen aanwenden om transitieve sluitingen van nabijheidsrelaties te berekenen. Ten slotte hebben we een FASP-programma gedefinieerd voor het probleem van het plaatsen van bancontactautomaten op strategische plaatsen (het zogenaamde

"ATM-selectieprobleem") evenals een FASP-programma dat gebruikt kan worden om het vaag graafkleuringsprobleem op te lossen.

In Hoofdstuk 4 hebben we een overzicht gegeven van de computationele complexiteit van FASP onder de Łukasiewicz semantiek (d.i. een vaaglogica die we besproken hebben in Sectie 2.2.3). In het bijzonder hebben we volgende beslissingsproblemen behandeld: gegeven een FASP-programma P, een literaal l en een waarde $\lambda_l \in [0,1] \cap \mathbb{Q}$:

1. **Bestaan**: Bestaat er een answer set I van P?
2. **Lidmaatschap**: Bestaat er een answer set I van P zodat $I(l) \geq \lambda_l$?
3. **Gevolg**: Geldt $I(l) \geq \lambda_l$ voor elke answer set I van P?

Voor de programma's met de grootste syntactische vrijheid, nl. reguliere FASP-programma's, hebben we Σ_2^{P}-compleetheid aangetoond voor *lidmaatschap* en *existentie*, en Π_2^{P}-compleetheid voor *gevolg* door gebruik te maken van gekende complexiteitsresultaten voor een meerwaardige versie van equilibrium logica [Schockaert et al. 2012]. Als we ons echter beperkten tot programma's met maximaal één literaal in de head van elke regel, dan konden we enkel Σ_2^{P}*-lidmaatschap* en NP-hardheid aantonen voor *lidmaatschap* en *existentie* en Π_2^{P}*-lidmaatschap* en coNP-hardheid voor *gevolg*. Voor dergelijke programma's die bovendien geen negatie-door-falen bevatten, hebben we een algoritme gegeven dat – gebaseerd op het bepalen van kleinste fixpunten – de answer sets kan berekenen in pseudo-polynomiale tijd. In het bijzonder, voor de deelklasse van zulke FASP-programma's met enkel conjunctie en maximum of enkel disjunctie in de body van regels hebben we een algoritme gevonden dat in polynomiale tijd de answer sets kan berekenen. Dit was ook het geval voor de deelklasse van programma's met een afhankelijksheidgraaf zonder cykels of met enkel polynomiaal gebonden constanten.

Hoewel *existentie* en *lidmaatschap* Σ_2^{P}-compleet zijn voor disjunctieve ASP-programma's, hebben we voor strikt disjunctieve FASP-programma's NP-compleetheid aangetoond. Strikt FASP omvat de programma's die syntactisch overeenkomen met ASP, maar gebruik maken van de semantiek van Łukasiewicz logica. Bovendien hebben we aangetoond dat het niet toestaan van beperkingen en sterke negatie de complexiteit van *lidmaatschap* niet aantast. Daarnaast hebben we aangetoond dat de complexiteit van het *existentie* probleem voor de deelklasse van programma's van maximaal één literaal in de head van elke regel en zonder beperkingen en zonder sterke negatie "constant" is omdat het bestaan van een answer set voor zulke programma's steeds gegarandeerd is. We hebben echter enkel *lidmaatschap* in NP kunnen aantonen voor de programma's zonder restricties op de heads van de regels maar zonder beperkingen en zonder sterke negatie.

Een overzicht van de complexiteitsresultaten kan gevonden worden in Tabellen 4.1 en 4.2. Ten slotte hebben we ook een implementatie van strikt disjunctief FASP voorgesteld, gebruikmakend van lineair programmeren met twee niveaus. Zo bekomen we een basis voor solvers voor FASP.

In Hoofdstuk 5 hebben we een algemenere vorm van FASP-programma's beschouwd waarin de connectieven geïnterpreteerd worden door willekeurige functies die niet beperkt zijn tot de connectieven van een bepaalde vaaglogica. We hebben vervolgens een meerwaardige versie van autoepistemische logica ingevoerd. Deze logica kan gebruikt worden om te redeneren over iemands (gebrek aan) geloof over de graad waarin bepaalde eigenschappen waar zijn. In dit hoofdstuk hebben we aangetoond dat na veralgemenen naar een meerwaardige versie belangrijke eigenschappen van klassieke autoepistemische logica en het verband tussen answer set programmeren en autoepistemische logica bewaard blijven. Bovendien hebben we twee verschillende maar equivalente karakterisaties van answer sets in autoepistemische vaaglogica en in vaaglogica van minimaal geloof en negatie-door-falen aangetoond. Deze resultaten leiden tot een beter begrip van hoe we vage answer sets moeten interpreteren. Omdat de taal van autoepistemische vaaglogica veel expressiever is dan de theorieën die we nodig hebben om de vage answer set programma's voor te stellen kan dit een nuttige basis zijn voor het definiëren of vergelijken van uitbreidingen van de basistaal van FASP.

Ten slotte, in Hoofdstuk 6 hebben we relaties tussen autoepistemische vaaglogica en modale vaaglogica's bestudeerd en hebben we gekende verbanden tussen autoepistemische logica en verschillende klassieke modale logica systemen veralgemeend. In het bijzonder hebben we Levesque's logica van "only knowing" veralgemeend en we hebben aangetoond dat het verband met autoepistemic logic bewaard blijft onder deze veralgemening. Bovendien hebben we een correcte en volledige axiomatisatie gegeven. Om deze axiomatisatie te bekomen hebben we veralgemeningen van de standaard propositionele modale logica's van geloof (K45, KD45, S5) gedefinieerd. We hebben aangetoond dat hun semantiek in de stijl van Kripke gerelateerd is aan autoepistemische vaaglogica en dat het klassieke verbrand tussen autoepistemische logica en logica van "only knowing" bewaard blijft.

Doorheen deze thesis hebben we een waaier van resultaten gepresenteerd die ons begrip over de complexiteit van FASP en het verband met andere vormen van niet-monotoon redeneren met waarheidswaarden heb verrijkt en uitgediept. Hierbij zijn echter ook weer nieuwe vragen en open problemen aan het licht gekomen. Ter afsluiting sommen we die vragen en problemen op die we het meest interessant en intrigerend vinden:

- Het bestuderen van de overgebleven open problemen i.v.m. de complexiteit van FASP onder Łukasiewicz semantiek:
  - Het vinden van een polynomiaal algoritme om de answer sets te bepalen van een regulier FASP-programma zonder sterke negatie, zonder negation-as-failure en met exact één atoom in de head van elke regel, of aantonen dat zo'n algoritme niet kan bestaan.

  - Onderzoeken of *existentie* voor strikt disjunctief FASP NP-hard is wanneer beperkingen en sterke negatie niet toegelaten zijn.
  - Onderzoeken of de beslissingsproblemen voor regulier FASP – als de head van elke regel maximaal één literaal of constante bevat – in NP zijn (of co-NP voor *gevolg*).
- De computationele complexiteit van FASP onder product logica en Gödel logica in kaart brengen.
- Het bestuderen van de implementatie van vaag answer set programmeren en auto-epistemische vaaglogica gebruik makend van multi-level lineair programmeren en op die manier de implementatie van klassieke autoepistemische logics m.b.v. Quantified Boolean Formulas veralgemenen.
- Het onderzoeken van vaagautoepistemische logica en vaaglogica van "only knowing" gebaseerd op oneindigwaardige Łukasiewicz logica met constanten, evenals voor semantieken van andere vaaglogica's.

List of Publications

International Journals

1. M. Blondeel, T. Flaminio, S. Schockaert, LL. Godo, M. De Cock. *Relating Fuzzy Autoepistemic Logic to Fuzzy Modal Logics of Belief.* Submitted.

2. M. Blondeel, S. Schockaert, D. Vermeir, M. De Cock. *Complexity of Fuzzy Answer Set Programming under Łukasiewicz Semantics.* Accepted for publication in International Journal of Approximate Reasoning.
http://dx.doi.org/10.1016/j.ijar.2013.10.011

3. M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir. *Fuzzy Autoepistemic Logic and its Relation to Fuzzy Answer Set Programming.* Fuzzy Sets and Systems, vol. 239, 2014, p 51-80.

Book Chapters

1. M. Blondeel, S. Schockaert, D. Vermeir, M. De Cock. *Fuzzy Answer Set Programming: An Introduction.* Soft Computing: State of the Art Theory and Novel Applications, Studies in Fuzziness and Soft Computing vol 291, Yager R. et al. (ed.), Springer, 2013, p 209-222.

Proceedings of Conferences with International Referees

1. M. Blondeel, T. Flaminio, LL. Godo, M. De Cock. *Relating Fuzzy Autoepistemic Logic to Fuzzy Modal Logics of Belief.* In Working Papers of the IJCAI-2013 (23rd International Joint Conference on Artificial Intelligence) Workshop on Weighted Logics for Artificial Intelligence (WL4AI), 2013, p 18-25.
2. M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir. *Towards a Deeper Understanding of Nonmonotonic Reasoning with Degrees.* In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), 2013, p 3205-3206.
3. M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir. *NP-completeness of Fuzzy Answer Set Programming under Łukasiewicz Semantics.* In Working Papers of the ECAI-2012 (20th European Conference on Artificial Intelligence) Workshop on Weighted Logics for Artificial Intelligence WL4AI, 2012, p 43-50.
4. M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir. *Complexity of Fuzzy Answer Set Programming under Łukasiewicz Semantics: First Results.* In Poster Proceedings of the 5th International Conference on Scalable Uncertainty Management (SUM), 2011, p 7-12.
5. M. Blondeel, S. Schockaert, M. De Cock, D. Vermeir. *Fuzzy Autoepistemic Logic: Reflecting about Knowledge of Truth Degrees.* In Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU), 2011, Lecture Notes in Computer Science 6717, 2011, p 616-627.

Extended abstracts

1. M. Blondeel, T. Flaminio, S. Schockaert, LL. Godo, M. De Cock. *On a Graded Version of "Only Knowing" and its Relation to Fuzzy Autoepistemic Logic and Fuzzy Modal Logics.* Book of Abstracts of the 35th Linz Seminar on Fuzzy Set Theory: Graded Logical Approaches and their Applications, 2014, p 22-24.
2. M. Blondeel, T. Flaminio, M. De Cock. *Complexity of Finitely Valued Łukasiewicz Possibilistic Modal Logics* In Handbook of the 4th World Congress and School on Universal Logic (Unilog), 2013, p 112-113.
3. M. Blondeel, T. Flaminio, LL. Godo. *Relating Fuzzy Autoepistemic Logic and Łukasiewicz KD45 Modal Logic*. In Book of Abstracts of Logic, Algebras and Truth Degrees (LATD), 2012, p 35-39.

Bibliography

[Alsinet and Godo 2000] T. Alsinet and LL. Godo (2000). *A Complete Calculus for Possibilistic Logic Programming with Fuzzy Propositional Variables*. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 1–10.

[Apt and Bol 1994] K.R. Apt and R.N. Bol (1994). *Logic Programming and Negation: A Survey*. Journal of Logic Programming, 19-20, 9–71.

[Audi 1995] R. Audi (1995). *The Cambridge Dictionary of Philosophy*: Cambridge University Press.

[Baaz 1996] M. Baaz (1996). *Infinite-Valued Gödel Logic with 0-1-Projections and Relativisations*. In Petr Hájek (ed.): *Gödel '96: Logical Foundations of Mathematics, Computer Science and Physics*, vol. 6 of *Lecture Notes in Logic*, pp. 23–33: Springer-Verlag.

[Baral 2003] C. Baral (2003). *Knowledge Representation, Reasoning and Declarative Problem Solving*: Cambridge University Press.

[Baral and Gelfond 1994] C. Baral and M. Gelfond (1994). *Logic Programming and Knowledge Representation*. Journal of Logic Programming, 19, 73–148.

[Baral and Subrahmanian 1991] C. Baral and V. Subrahmanian (1991). *Dualities between Alternative Semantics for Logic Programming and Non-monotonic Reasoning*. In *Proceedings of the 1st International Workshop on Non-Monotonic Reasoning*, pp. 69–86.

[Baral and Subrahmanian 1993] C. Baral and V.S. Subrahmanian (1993). *Duality Between Alternative Semantics of Logic Programs and Nonmonotonic Formalisms*. Journal of Automated Reasoning, 10, 399–420.

[Bard 1998] J. Bard (1998). *Practical Bilevel Optimization: Algorithms and Applications*: Kluwer Academic Publishers: USA.

[Bard and Falk 1982] J. Bard and J. Falk (1982). *An Explicit Solution to the Multi-Level Programming Problem*. Computers and Operations Research, 9, 77–100.

[Bard and Moore 1990] J. Bard and J.T. Moore (1990). *A Branch and Bound Algorithm for the Bilevel Programming Problem*. SIAM Journal on Scientific and Statistical Computation, 11, 281–292.

[Bidoit and Froidevaux 1991a] N. Bidoit and C. Froidevaux (1991a). *General Logical Databases and Programs: Default Logic Semantics*. Information and computation, 91(1), 15–54.

[Bidoit and Froidevaux 1991b] N. Bidoit and C. Froidevaux (1991b). *Negation by Default and Unstratifiable Logic Programs*. Theoretical Computer Science, 78, 85–112.

[Bjorklund et al. 2003] H. Bjorklund, S. Sandberg, and S. Vorobyov (2003). *Complexity of Model Checking by Iterative Improvement: The Pseudo-Boolean Framework*. In *Proceedings of the 5th Andrei Ershov Memorial Conference "Perspectives of System Informatics"*, pp. 381–394.

[Blondeel et al. 2013a] M. Blondeel, T. Flaminio, LL. Godo, and M. De Cock (2013a). *Relating Fuzzy Autoepistemic Logic to Fuzzy Modal Logics of Belief*. In *Working Papers of the IJCAI Workshop on Weighted Logics for Artificial Intelligence*, pp. 18–25.

[Blondeel et al. 2014a] M. Blondeel, T. Flaminio, S. Schockaert, LL. Godo, and M. De Cock (2014a). *Relating Fuzzy Autoepistemic Logic to Fuzzy Modal Logics of Belief*. Submitted.

[Blondeel et al. 2011a] M. Blondeel, S. Schockaert, M. De Cock, and D. Vermeir (2011a). *Complexity of Fuzzy Answer Set Programming under Łukasiewicz Semantics: First Results*. In *Poster Proceedings of the 5th International Conference on Scalable Uncertainty Management*, pp. 7–12.

[Blondeel et al. 2011b] M. Blondeel, S. Schockaert, M. De Cock, and D. Vermeir (2011b). *Fuzzy Autoepistemic Logic: Reflecting about Knowledge of Truth Degrees*. In *Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 616–627.

[Blondeel et al. 2012] M. Blondeel, S. Schockaert, M. De Cock, and D. Vermeir (2012). *NP-completeness of Fuzzy Answer Set Programming under Łukasiewicz Semantics*. In *Working Papers of the ECAI Workshop on Weighted Logics for Artificial Intelligence*, pp. 43–50.

[Blondeel et al. 2013b] M. Blondeel, S. Schockaert, M. De Cock, and D. Vermeir (2013b). *Towards a Deeper Understanding of Nonmonotonic Reasoning with Degrees*. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 3205–3206.

[Blondeel et al. 2014b] M. Blondeel, S. Schockaert, M. De Cock, and D. Vermeir (2014b). *Fuzzy Autoepistemic Logic and Its Relation to Fuzzy Answer Set Programming*. Fuzzy Sets and Systems, 239, 51–80.

[Blondeel et al. 2013c] M. Blondeel, S. Schockaert, D. Vermeir, and M. De Cock (2013c). *Fuzzy Answer Set Programming: An Introduction*. In *Soft Computing: State of the Art Theory and Novel Applications*, Studies in Fuzziness and Soft Computing: Springer.

[Blondeel et al. 2014c] M. Blondeel, S. Schockaert, D. Vermeir, and M. De Cock (2014c). *Complexity of Fuzzy Answer Set Programming under Łukasiewicz Semantics*. Accepted for publication in International Journal of Approximate Reasoning.

[Bobillo et al. 2011] F. Bobillo, F. Bou, and U. Straccia (2011). *On the Failure of the Finite Model Property in Some Fuzzy Description Logics*. Fuzzy Sets and Systems, 172(23), 1–12.

[Bou et al. 2011a] F. Bou, M. Cerami, and F. Esteva (2011a). *Finite-Valued Łukasiewicz Modal Logic is PSPACE-complete*. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pp. 774–779.

[Bou et al. 2011b] F. Bou, F. Esteva, LL. Godo, and R. Rodríguez (2011b). *On the Minimum Many-Valued Modal Logic over a Finite Residuated Lattice*. Journal of Logic and Computation, 21(5), 739–790.

[Brewka 1991] G. Brewka (1991). *Cumulative Default Logic: In Defense of Nonmonotonic Inference Rules*. Artificial Intelligence, 50, 183–205.

[Brewka et al. 1997] G. Brewka, J. Dix, and K. Konolige (1997). *Nonmonotonic Reasoning: An Overview*, vol. 73 of *CSLI Lecture notes*: CSLI Publications, Stanford, CA.

[Candler and Townsley 1982] W. Candler and R. Townsley (1982). *A Linear Two-Level Programming Problem*. Computers and Operations Research, 9, 59–76.

[Carnap 1945] R. Carnap (1945). *The Two Concepts of Probability: The Problem of Probability*. Philosophe and Phenomenological Research, 5(4), 513–532.

[Carnap 1947] R. Carnap (1947). *Meaning and Necessity*: University of Chicago Press.

[Castro et al. 1998] J.L. Castro, E. Trillas, and J.M. Zurita (1998). *Non-Monotonic Fuzzy Reasoning*. Fuzzy Sets and Systems, 94, 217–225.

[Cerami and Straccia 2013] M. Cerami and U. Straccia (2013). *On the (Un)decidability of Fuzzy Description Logics under Łukasiewicz t-norm*. Information Sciences, 227, 1–21.

[Chang 1958] C. Chang (1958). *Algebraic Analysis of Many Valued Logics*. Transactions of the American Mathematical Society, 88, 476–490.

[Chang 1959] C. Chang (1959). *A New Proof of the Completeness of the Łukasiewicz Axioms*. Transactions of the American Mathematical Society, 93, 74–80.

[Chellas 1980] B.F. Chellas (1980). *Modal Logic*: Cambridge University Press.

[Chen 1993] J. Chen (1993). *Minimal Knowledge + Negation as Failure = Only Knowing (Sometimes)*. In *Proceedings of the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning*, pp. 132–150.

[Church 1936] A. Church (1936). *A Note on the Entscheidungsproblem*. The Journal of Symbolic Logic, 1(1), 40–41.

[Chvalovský 2012] Karel Chvalovský (2012). *On the Independence of Axioms in BL and MTL*. Fuzzy Sets and Systems, 197, 123–129.

[Cignoli et al. 2000] R Cignoli, I.M.L. D'Ottaviano, and D. Mundici (2000). *Algebraic Foundations of Many-Valued Reasoning*: Kluwer, Dordrecht.

[Clark 1978] K.L. Clark (1978). *Negation as Failure*. In *Logic and Data Bases*, vol. 1: Minker, J.

[Colmerauer et al. 1973] A. Colmerauer, H. Kanoui, R. Pasero, and P. Roussel (1973). *Un Système de Communication Homme-Machine en Français*. Technical Report, Groupe de Intelligence Artificielle Université de Aix-Marseille II.

[Cook 1971] S.A. Cook (1971). *The Complexity of Theorem-Proving Procedures*. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pp. 151–158.

[Corcoran 2003] J. Corcoran (2003). *Aristotle's Prior Analytics and Boole's Laws of Thought*. History and Philosophy of Logic, 24(4), 261–288.

[Corcoran 2009] J. Corcoran (2009). *Aristotle's Demonstrative Logic*. History and Philosophy of Logic, 30, 1–20.

[Damásio and Pereira 2001] C.V. Damásio and L.M. Pereira (2001). *Antitonic Logic Programs*. In *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning*, pp. 379–392.

[De Finetti 1936] B. De Finetti (1936). *La Logique de la Probabilité*. In *Actes du Congrès Internationale de Philosphie Scientifique*.

[Delgrande and Jackson 1991] J.P. Delgrande and W.K. Jackson (1991). *Default Logic Revisited*. In *Proceedings of the 2nd Conference on Principles of Knowledge Representation and Reasoning*, pp. 118–127.

[Dubois and Prade 2001] D. Dubois and H. Prade (2001). *Possibility Theory, Probability Theory and Multiple-Valued Logics: A Clarification*. Annals of Mathematics and Artificial Intelligence, 32(1-4), 35–66.

[Dummet 1959] M. Dummet (1959). *A Propositional Calculus with Denumerable Matrix*. Journal of Symbolic Logic, 24, 97–106.

[Egly et al. 2000] Uwe Egly, Thomas Eiter, Hans Tompits, and Stefan Woltran (2000). *Solving Advanced Reasoning Tasks Using Quantified Boolean Formulas*. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp. 417–422.

[Eiter and Gottlob 1993] T. Eiter and G. Gottlob (1993). *Complexity Results for Disjunctive Logic Programming and Application to Nonmonotonic Logics*. In *Proceedings of the 10th International Logic Programming Symposium*, pp. 266–278.

[Esteva and Godo 2001] F. Esteva and LL. Godo (2001). *Monoidal t-norm Based Logic: Towards a Logic for Let-Continuous t-norms*. Fuzzy Sets and Systems, 124, 271–288.

[Etherington 1987] D.W Etherington (1987). *Formalizing Nonmonotonic Reasoning Systems*. Artificial Intelligence, 31, 41–85.

[Fagin et al. 1994] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi (1994). *Reasoning about Knowledge*: MIT Press.

[Fitting 1992a] M.C. Fitting (1992a). *Many-Valued Modal Logics*. In *Fundamenta Informaticae*, pp. 365–448.

[Fitting 1992b] M.C. Fitting (1992b). *Many-Valued Modal Logics II*. Fundamenta Informaticae, 17.

[Fitting 1992c] M.C. Fitting (1992c). *Many-Valued Non-Monotonic Modal Logics*. In *Logic Foundations of Computer Science*, pp. 139–150.

[Gabbay 1985] D. M. Gabbay (1985). *Theoretical Foundations for Non-Monotonic Reasoning in Expert Systems*. In *Proceedings of the NATO Advanced Study Institute on Logics and Models of Concurrent Systems*, pp. 439–457.

[Gawlitza and Seidle 2007] T. Gawlitza and H. Seidle (2007). *Precise Fixpoint Computation through Strategy Iteration*. In *Proceedings of the 16th European Symposium on Programming*, pp. 300–315.

[Gelfond 1987] M. Gelfond (1987). *On Stratified Autoepistemic Theories*. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pp. 207–211.

[Gelfond and Lifschitz 1988] M. Gelfond and V. Lifschitz (1988). *The Stable Model Semantics for Logic Programming*. In *Proceedings of the 5th International Conference and Symposium on Logic Programming*, pp. 1070–1080.

[Gelfond and Lifschitz 1991] M. Gelfond and V. Lifschitz (1991). *Classical Negation in Logic Programs and Disjunctive Databases*. New Generation Computing, 9, 365–385.

[Gelfond et al. 1989] M. Gelfond, H. Przymusinska, and T.C. Przymusinski (1989). *On the Relationship between Circumscription and Negation as Failure*. Artificial Intelligence, 38, 75–94.

[Gödel 1932] K. Gödel (1932). *Zum Intuitionistischen Aussagenkalkül*. Anzeiger Akadamie der Wissenschaften Wien, 69, 65–66.

[Gödel 1933] K. Gödel (1933). *Eine Interpretation des Intuitionistischen Aussagenkalküls*. Ergebnisse eines Mathematischen Kolloquiums, 4(39-40).

[Gottlob 1992] G. Gottlob (1992). *Complexity Results for Nonmonotonic Logics*. Journal of Logic and Computation, 2(3), 397–425.

[Gottwald and Hájek 2005] S. Gottwald and P. Hájek (2005). *T-norm Based Mathematical Fuzzy Logics*. In E.P. Klement and R. Mesiar (eds.): *Logical, Algebraic, Analytic, and Probabilistic Aspects of Triangualar Norms*: Elsevier, Dordrecht.

[Hähnle 1994] R. Hähnle (1994). *Many-Valued Logic and Mixed Integer Programming*. Annals of Mathematics and Artificial Intelligence, 12, 231–264.

[Hähnle 1997] R. Hähnle (1997). *Proof Theory of Many-Valued Logic - Linear Optimization - Logic Design: Connections and Interactions*. Soft Computing, 1, 107–119.

[Hájek 1995] P. Hájek (1995). *Fuzzy Logic and Arithmetical Hierarchy*. Fuzzy Sets and Systems, 73(359-363).

[Hájek 1998] P. Hájek (1998). *Metamathematics of Fuzzy Logic*, vol. 4 of *Trends in Logic*: Kluwer.

[Hájek 2010] P. Hájek (2010). *On Fuzzy Modal Logics S5(C)*. Fuzzy Sets and Systems, 161(18), 2389–2396.

[Halpern and Moses 1984] J.Y. Halpern and Y. Moses (1984). *Towards a Theory of Knowledge and Ignorance: Preliminary Report*. In *Proceedings of the 1st International Workshop on Nonmonotonic Reasoning*, pp. 125–143.

[Halpern and Moses 1992] J.Y. Halpern and Y. Moses (1992). *A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief*. Artificial Intelligence, 54(3), 319–279.

[Hansoul and Teheux 2013] G. Hansoul and B. Teheux (2013). *Extending Łukasiewicz Logics with a Modality: Algebraic Approach to Relational Semantics*. Studia Logica, 101(3), 505–545.

[Hintikka 1962] J. Hintikka (1962). *Knowledge and Belief*: Cornell University Press.

[Janssen 2011] J. Janssen (2011). *Foundations of Fuzzy Answer Set Programming*. PhD thesis, Ghent University and Vrije Universiteit Brussel.

[Janssen et al. 2009] J. Janssen, S. Schockaert, D. Vermeir, and M. De Cock (2009). *General Fuzzy Answer Set Programs*. In *Proceedings of the 8th International Workshop on Fuzzy Logic and Applications*, pp. 353–359.

[Jaskowski 1936] S. Jaskowski (1936). *Recherches sur le Système de la Logique Intuitionistic*. In *Actes du Congrès Internationale de Philosphie Scientifique*, pp. 58–61.

[Kleene and Post 1954] S.C. Kleene and E.L. Post (1954). *The Upper Semi-Lattice of Degrees of Recursive Unsolvability*. Annals of Mathematics, 59(3), 379–407.

[Klement et al. 2000] E.P. Klement, R. Mesiar, and E. Pap (2000). *Triangular Norms*, vol. 8 of *Trends in Logic*: Springer.

[Konolige 1988] K. Konolige (1988). *On the Relation between Default and Autoepistemic Logic*. Artificial Intelligence, pp. 343–382.

[Koutras and Zachos 2000] C.D. Koutras and S. Zachos (2000). *Many-Valued Reflexive Autoepistemic Logic*. Logic Journal of the IGPL, 8(1), 403–418.

[Kowalski 1974] R.A. Kowalski (1974). *Predicate Logic as a Programming Language*. In *Proceedings of the 6th International Federation for Information Processing World Computer Congress*, pp. 569–574.

[Kraus et al. 1990] S. Kraus, D.J. Lehmann, and M. Magidor (1990). *Nonmonotonic Reasoning, Preferential Models and Cumulative Logics*. Artificial Intelligence, 44(1-2), 167–207.

[Kripke 1959] S. Kripke (1959). *Semantical Analysis of Modal Logic (abstract)*. The Journal of Symbolic Logic, 24, 323–324.

[Levesque 1984] H.J. Levesque (1984). *Foundations of a Functional Approach to Knowledge Representation*. Artificial Intelligence, 23(2), 155–212.

[Levesque 1990] H.J. Levesque (1990). *All I Know: A Study in Autoepistemic Logic*. Artificial Intelligence, pp. 263–309.

[Lewis 1918] C.I. Lewis (1918). *A Survey of Symbolic Logic*: University of California Press.

[Lewis and Langford 1932] C.I. Lewis and C.H. Langford (1932). *Symbolic Logic*: Century Company.

[Lifschitz 1985] V. Lifschitz (1985). *Computing Circumscription*. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pp. 121–127.

[Lifschitz 1988] V. Lifschitz (1988). *On the Declarative Semantics of Logic Programs with Negation*. In J. Minker (ed.): *Foundations of Deductive Databases and Logic Programming*, pp. 177–192: Kaufmann.

[Lifschitz 1991] V. Lifschitz (1991). *Nonmonotonic Databases and Epistemic Queries*. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pp. 381–386.

[Lifschitz 1994] V. Lifschitz (1994). *Minimal Belief and Negation as Failure*. Artificial Intelligence, 70, 53–72.

[Lifschitz 2008] V. Lifschitz (2008). *Twelve Definitions of a Stable Model*. In *Proceedings of the 24th International Conference on Logic Programming*, pp. 37–51.

[Lifschitz and Schwarz 1993] V. Lifschitz and G. Schwarz (1993). *Extended Logic Programs as Autoepistemic Theories*. In *Proceedings of the 2nd International Workshop on Logic Programming and Nonmonotonic Reasoning*, pp. 101–114.

[Lin and Shoham 1990] F. Lin and Y. Shoham (1990). *Epistemic Semantics for Fixed-Points Nonmonotonic Logics*. In *Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 111–200.

[Lin and Shoham 1992] F. Lin and Y. Shoham (1992). *A Logic of Knowledge and Justified Assumptions*. Artificial Intelligence, 57, 271–289.

[Łukasiewicz 1920] J. Łukasiewicz (1920). *O logice trójwartościowej*. Ruch Filozoficzny, 5, 170–171.

[Łukasiewicz 1922] J. Łukasiewicz (1922). *A Numerical Interpretation of the Theory of Propositions (Polish)*. Ruch Filozoficzny, 7, 92–93.

[Łukasiewicz 1999] T. Łukasiewicz (1999). *Many-Valued Disjunctive Logic Programs with Probabilistic Semantics*. In *Proceedings of the 5th International Conference on Logic Programming and Nonmonotonic Reasoning*, pp. 277–289.

[Łukasiewicz 2008] T. Łukasiewicz (2008). *Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web*. Fundamenta Informaticae, 82(3), 289–310.

[Łukasiewicz and Straccia 2007] T. Łukasiewicz and U. Straccia (2007). *Tightly Integrated Fuzzy Description Logic Programs under the Answer Set Semantics for the Semantic Web*. In *Proceedings of the 1st International Conference on Web Reasoning and Rule Systems*, pp. 289–298.

[Łukasiewicz 1988] W. Łukasiewicz (1988). *Consideration on Default Logic*. Computational Intelligence, 4(1), 1–16.

[Madrid and Ojeda-Aciego 2011] N. Madrid and M. Ojeda-Aciego (2011). *Measuring Inconsistency in Fuzzy Answer Set Semantics*. IEEE Transactions on Fuzzy Systems, 19(4), 605–622.

[Madrid and Ojeda-Aciego 2012] N. Madrid and M. Ojeda-Aciego (2012). *On the Existence and Unicity of Stable Models in Normal Residuated Logic Programs*. International Journal of Computer Mathematics, 89(3), 310–324.

[Makinson 1988] D. Makinson (1988). *General Theory of Cumulative Inference*. In *Proceedings of the 2nd International Workshop on Non-Monotonic Reasoning*, pp. 1–18.

[Makinson 2005] D. Makinson (2005). *Bridges from Classical to Nonmonotonic Logic*: King's College Publications.

[Marek 1989] W. Marek (1989). *Stable Theories in Autoepistemic Logic*. Fundamenta Informaticae, 12(254).

[Marek et al. 1991] W. Marek, G. Shvarts, and M. Truszczyński (1991). *The Relationship Between Logic Program Semantics and Non-monotonic Reasoning*. In *Proceedings of the 6th International Conference on Logic Programming*, pp. 276–288.

[Marek and Truszczyński 1989] W. Marek and M. Truszczyński (1989). *Stable Semantics for Logic Programs and Default Reasoning*. In *Proceedings of the 6th North American Conference on Logic Programming*, pp. 243–257.

[Marek and Truszczyński 1993] W. Marek and M. Truszczyński (1993). *Reflexive Autoepistemic Logic and Logic Programming*. In *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning*, pp. 115–131.

[Marek and Truzsczyński 1993] W. Marek and M. Truzsczyński (1993). *Nonmonotonic Logic: Context-Dependent Reasoning*. Artificial Intelligence: Springer.

[Maruyama 2011] Y. Maruyama (2011). *Reasoning about Fuzzy Belief and Common Belief: With Emphasis on Incomparable Beliefs*. In *Proceedings of the 22th International Joint Conference on Artificial Intelligence*, pp. 1008–1013.

[McCarthy 1980] J. McCarthy (1980). *Circumscription - A Form of Nonmonotonic Reasoning*. Artificial Intelligence, 13(1+2), 89–116.

[McCarthy 1986] J. McCarthy (1986). *Applications of Circumscription to Formalizing Commonsense Knowledge*. Artificial Intelligence, 28, 89–116.

[McDermott 1982] D.V. McDermott (1982). *Nonmonotonic Logic 2: Nonmonotonic Modal Theories*. Journal of the Association for Computing Machinery, 29(1), 33–57.

[McDermott and Doyle 1980] D.V. McDermott and J. Doyle (1980). *Nonmonotonic Logic 1*. Artificial Intelligence, 13(1-2), 41–72.

[McNaughton 1951] R. McNaughton (1951). *A Theorem about Infinite-Valued Sentential Logic*. The Journal of Symbolic Logic, 16(1), 1–13.

[Megiddo and Tamir 1983] N. Megiddo and A. Tamir (1983). *New Results on the Complexity of p-Center Problems*. SIAM Journal on Computing, 12(4), 751–758.

[Monteiro 1980] A.A. Monteiro (1980). *Algebrés de Heyting Symétriques*. Portugaliae Mathematica, 39(1-4), 1–239.

[Moore 1984] R. Moore (1984). *Possible-World Semantics in Autoepistemic Logic*. In *Proceedings of the 1st International Workshop on Nonmonotonic Reasoning*, pp. 344–354.

[Moore 1985] R. Moore (1985). *Semantical Considerations on Nonmonotonic Logic*. Artificial Intelligence, 25(1), 75–94.

[Mundici 1987] D. Mundici (1987). *Satisfiability in Many-Valued Sentential Logic is NP-complete*. Theoretical Computer Science, 52(5), 145–153.

[Papadimitriou 1994] C. M. Papadimitriou (1994). *Computational Complexity*: Addison-Wesley.

[Pavelka 1979] Jan Pavelka (1979). *On Fuzzy Logic I, II, III*. Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 25, 45–52, 119–134, 447–464.

[Post 1921] E.L. Post (1921). *Introduction to a General Theory of Elementary Propositions*. American Journal of Mathematics, 43, 163–185.

[Post 1944] E.L. Post (1944). *Recursively Enumerable Sets of Positive Integers and their Decision problems*. Bulletin of the American Mathematical Society, 50(5), 284–316.

[Przymusinski 1988] T.C. Przymusinski (1988). *On the Declarative Semantics of Deductive Databases and Logic Programs*. In *Foundations of Deductive Databases and Logic Programming*, pp. 193–216: Kaufmann.

[Reiter 1978] R. Reiter (1978). *On Closed World Databases*. Logic and Data Bases, pp. 55–76.

[Reiter 1980] R. Reiter (1980). *A Logic for Default-Reasoning*. Artificial Intelligence, 13, 81–132.

[Reiter 1987] R. Reiter (1987). *A Theory of Diagnosis from First Principles*. Artificial Intelligence, 32, 57–97.

[Reiter 1991] R. Reiter (1991). *What Should a Database Know?*. Journal of Logic Programming, 14, 127–153.

[Rosser and Turquette 1952] J.B. Rosser and A.R. Turquette (1952). *Many-Valued Logics*: Nort Holland.

[Schaub 1991] T. Schaub (1991). *On Commitment and Cumulativity in Default Logics*. In *Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Uncertainty*, pp. 304–309.

[Schockaert et al. 2009] S. Schockaert, M. De Cock, and E. Kerre (2009). *Spatial Reasoning in a Fuzzy Region Connection Calculus*. Artificial Intelligence, 173, 258–298.

[Schockaert et al. 2012] S. Schockaert, J. Janssen, and D. Vermeir (2012). *Fuzzy Equilibrium Logic: Declarative Problem Solving in Continuous Domains*. ACM Transactions on Computational Logic, 13(4), 111–155.

[Schwarz 1992] G. Schwarz (1992). *Reflexive Autoepistemic Logic*. Fundamenta Informaticae, 17, 157–173.

[Shi et al. 2006] C. Shi, J. Lu, G. Zhang, and H. Zhou (2006). *An Extended Branch and Bound Algorithm for Linear Bilevel Programming*. Applied Mathematics and Computation, 180(2), 529–537.

[Shoham 1987] Y. Shoham (1987). *A Semantical Approach to Nonmonotonic Logics*. In *Proceedings of the 2nd Symposium on Logic in Computer Science*, pp. 275–279.

[Shoham 1988] Y. Shoham (1988). *Reasoning about Change*: The MIT Press.

[Słupecki 1936] J. Słupecki (1936). *Der Volle Dreiwertige Aussagenkalkül*. Comptes Rendus Séances Société des Sciences et Lettres Varsovie, 3(29), 9–11.

[Stalnaker 1993] R. Stalnaker (1993). *A Note on Non-Monotonic Modal Logic*. Artificial Intelligence, 64(2), 183–196.

[Stepherdson 1991] J.C. Stepherdson (1991). *Unsolvable Problems for SLDNF-Resolution*. Journal of Logic Programming, 10, 19–22.

[Straccia 2006] U. Straccia (2006). *Annotated Answer Set Programming*. In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 1212–1219.

[Straccia et al. 2009] U. Straccia, M. Ojeda-Aciego, and C. V. Damásio (2009). *On Fixed-Points of Multi-Valued Functions on Complete Lattices and their Application to Generalized Logic Programs*. SIAM Journal on Computing, (5), 1881–1911.

[Tarjan 1972] R.E Tarjan (1972). *Depth-First Search and Linear Graph Algorithms*. SIAM Journal on Computing, 1(2), 146–160.

[Tarski 1955] A. Tarski (1955). *A Lattice Theoretical Fixpoint Theorem and its Applications*. Pacific Journal of Mathematics, 5(285-309).

[Turing 1936] A.M. Turing (1936). *On Computable Numbers, with an Application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, 42, 230–265.

[Van Emden and Kowalski 1976] M.H. Van Emden and R.A. Kowalski (1976). *The Semantics of Predicate Logic as a Programming Language*. Journal of the ACM, 23(4), 733–742.

[van Gelder et al. 1988] A. van Gelder, K.A. Ross, and J.S. Schlipf (1988). *Unfounded Sets and Well-founded Semantics for General Logic Programs*. In *Proceedings of the 7th Symposion on Principles of Databases Systems*, pp. 221–230.

[Van Nieuwenborgh et al. 2007] D. Van Nieuwenborgh, M. De Cock, and D. Vermeir (2007). *An Introduction to Fuzzy Answer Set Programming*. Annals of Mathematics and Artificial Intelligence, 50(3-4), 363–388.

[Wajsberg 1931] M. Wajsberg (1931). *Aksjomatyzacja trówartosciowego rachunku zdań*. Comptes Rendus Séances Société des Sciences et Lettres Varsovie, 3(24), 126–148.

[Yu et al. 2005] L. Yu, N. Wang, and X. Meng (2005). *Real-Time Forest Fire Detection with Wireless Sensor Networks*. In *Proceedings of the Wireless Communication Networking and Mobile Computing International Conference*, pp. 1214–1217.

[Zadeh 1965] L.A. Zadeh (1965). *Fuzzy Sets*. Information and Control, 8(3), 338–353.

List of Symbols

A'	denotes the set $A \cup \{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}^{c}_{\mathrm{B}}\}$, page 24
A'	denotes the set $A \cup \{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}_{B}\}$, page 122
A'	denotes the set $A \cup \left\{\mathrm{B}\varphi \mid \varphi \in \mathcal{L}^{k}_{\mathrm{B}}\right\}$, page 161
P^I	reduct of an ASP program P w.r.t. to an interpretation I, page 49
P^I	reduct of a FASP program P w.r.t. to a fuzzy interpretation I, page 65
T^*	denotes the set $\{\alpha^* \mid \alpha \in T\}$ with $T \subseteq \mathcal{L}_B$, page 122
T^*	denotes the set $\{\alpha^* \mid \alpha \in T\}$ with $T \subseteq \mathcal{L}^{k}_{\mathrm{B}}$, page 161
$\mathcal{L}^*_B$	non modal language corresponding to $\mathcal{L}_B$, page 122
Π_P	immediate consequence operator of an ASP program P, page 48
Π_P	immediate consequence operator of a FASP program P, page 63
α^*	corresponding formula in $\mathcal{L}^*_B$ for $\alpha \in \mathcal{L}_B$, page 122
α^*	corresponding formula in $(\mathcal{L}^{k}_{\mathrm{B}})^*$ for $\alpha \in \mathcal{L}^{k}_{\mathrm{B}}$, page 161
$\lambda(P)$	corresponding autoepistemic theory for a normal ASP program P without constraints or strong negation, page 56

$\lambda(P)$	corresponding fuzzy autoepistemic theory for a normal FASP program P without constraints or strong negation, page 136
$\mathcal{B}_P$	set of all atoms in an ASP program P, page 47
$\mathcal{B}_P$	set of all atoms in a FASP program P, page 61
$\mathcal{L}_P$	denotes the set $\{a \mid a \in \mathcal{B}_P\} \cup \{\neg a \mid a \in \mathcal{B}_P\}$ for an ASP program P, page 47
$\mathcal{L}_P$	denotes the set $\{a \mid a \in \mathcal{B}_P\} \cup \{\neg a \mid a \in \mathcal{B}_P\}$ for a FASP program P, page 61
$\mu(P)$	corresponding theory in MBNF for a disjunctive ASP program P, page 141
$\mu(P)$	corresponding theory in FMBNF for a regular FASP program P, page 144
not	negation-as-failure operator in an ASP program, page 46
not	negation-as-failure operator in a FASP program, page 59
$\neg a$	strong negation of a in an ASP program, page 46
$\neg a$	strong negation of a in a FASP program, page 59
ϕ_r	denotes the formula $\Delta(\phi \leftrightarrow \overline{r}) \in \mathcal{L}_{\mathrm{B}}^k$ for $\phi \in \mathcal{L}_{\mathrm{B}}^k$ and $r \in S_k$, page 168
ϕ_r	denotes the formula $\Delta(\phi \leftrightarrow \overline{r}) \in \mathcal{L}_{\mathrm{O}}^k$ for $\phi \in \mathcal{L}_{\mathrm{O}}^k$ and $r \in S_k$, page 183
$\sigma(P)$	corresponding autoepistemic theory for a disjunctive ASP program, page 57
$\sigma(P)$	corresponding fuzzy autoepistemic theory for a regular FASP program, page 150
r_b	body of an ASP rule r, page 46
r_b	body of a FASP rule r, page 59
r_h	head of an ASP rule r, page 46
r_h	head of a FASP rule r, page 59
$v_{\vert A}$	evaluation $v_{\vert A} : A \to [0,1]$ for $v \in \Omega^*$, page 126
$(\mathcal{L}_{\mathrm{B}}^k)'$	subset of $\mathcal{L}_{\mathrm{B}}^k$, set of formulas constructed from $\{\Delta(p \leftrightarrow \overline{r}) \mid p \in V, r \in S_k\}$, page 168

(Ł1)	axiom (Ł1), page 41
(Ł2)	axiom (Ł2), page 41
(Ł3)	axiom (Ł3), page 41
(Ł4)	axiom (Ł4), page 41
(Ł5)	axiom (Ł5), page 41
(Ł6)	axiom (Ł6), page 41
$(\mathcal{L}^k_{\mathrm{B}})^*$	non modal language corresponding to $\mathcal{L}^k_{\mathrm{B}}$, page 161
$(\mathcal{L}^k_{\mathrm{O}})'$	subset of $\mathcal{L}^k_{\mathrm{O}}$, set of formulas constructed from $\{\Delta(p \leftrightarrow \overline{r}) \mid p \in V, r \in S_k\}$, page 183
$(\mathcal{L}^k_{\mathrm{O}})^+$	extension of $\mathcal{L}^k_{\mathrm{O}}$, page 184
$(\mathcal{L}^k_{\mathrm{O}})^C$	two valued $\mathcal{L}^k_{\mathrm{O}}$, page 184
$\mathrm{BŁ}^c_k$	axiomatic extension of the minimal model logic over Ł^c_k with axiom (K), page 43
B	modal operator, page 24
Δ	$\Delta\phi = \phi \otimes \ldots \otimes \phi$ (k times), page 43
KD45(Ł^c_k)	$\mathrm{BŁ}^c_k$ extended with axioms (D), (4) and (5), page 161
K45(Ł^c_k)	$\mathrm{BŁ}^c_k$ extended with axioms (4) and (5), page 161
Λ_L	denotes the set $\{\phi^* \mid \vdash_L \phi\}$, page 163
$\mathcal{L}_B$	language of fuzzy autoepistemic logic, page 122
$\mathcal{L}^k_{\mathrm{B}}$	expansion of $\mathcal{L}^c_k$ with modal operator B, page 42
$\mathcal{L}^c$	language of all propositional formulas over A, page 24
$\mathcal{L}^c_{\mathrm{B}}$	extension of $\mathcal{L}^c$ with modal operator B, page 24
L^K	propositional fuzzy logic corresponding to a set of continuous t-norms K, page 35
$\mathcal{L}^c_k$	language of finitely-valued Łukasiewicz logic with a finite set of truth constants, page 40

$\mathbb{M}^s_{\mathrm{rsyt}}$	class of Kripke models (W, e, R) with $R = W \times W$, page 166
N	modal operator, page 30
Ω	set of propositional evaluations $A \to [0,1]$, page 123
Ω^*	set of propositional evaluations $A' \to [0,1]$, page 122
Ω^*_k	set of propositional evaluations $A' \to S_k$, page 161
Ω_k	set of propositional Ł^c_k evaluations, page 40
O	modal operator, page 29
$\mathrm{O}(\text{Ł}^c_k)$	logic of only knowing under Ł^c_k, page 182
$\Pi(P, M)$	denotes the set $\{w \in \Omega \mid w$ is a minimal element of $\pi^M_P\}$ for a FASP program P and $M \in \mathcal{F}(\mathcal{B}_P)$, page 144
Π^{P}_{i+1}	class of decision problems co$(\Sigma^{\mathsf{P}}_{i+1})$, page 53
S5(Ł^c_k)	$\mathrm{B}\text{Ł}^c_k$ extended with axioms (T), (4) and (5), page 161
$\Sigma^{\mathsf{P}}_{i+1}$	class of decision problems $\mathsf{NP}^{\Sigma^{\mathsf{P}}_i}$, page 52
$\vdash_L$	notion of proof in a logic L, page 38
A	set of atoms, page 24
e^L_{can}	$e^L_{can}(w, p) = w(p)$ for a variable p and $w \in W^L_{can}$, page 163
F_n	the set of n-ary connectives, page 122
I_L	residual implicator for T_L, page 34
I_M	residual implicator for T_M, page 34
I_P	residual implicator for T_P, page 34
M^L_{can}	L-canonical Kripke model $M^L_{can} = (W^L_{can}, e^L_{can}, R^L_{can})$ for a fuzzy modal logic L, page 163
N_{I_L}	negator for I_L, page 35
N_{I_M}	negator for I_M, page 35

N_{I_P}	negator for I_P, page 35
R^L_{can}	denotes the set $\{(w_1, w_2) \in \Omega^*_k \times \Omega^*_k \mid \forall \phi \in \mathcal{L}^k_{\mathrm{B}} :$ if $w_1((\mathrm{B}\phi)^*) = 1,$ then $w_2(\phi^*) = 1\}$, page 163
S_k	denotes the set $\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$, page 40
S_L	Łukasiewicz t-conorm, page 32
S_M	maximum t-conorm, page 32
S_P	probabilistic sum t-conorm, page 32
T_L	Łukasiewicz t-norm, page 32
T_M	minimum t-norm, page 32
T_P	product t-norm, page 32
v_I	$v_I(x) = I(x)$ if $x \in \mathcal{B}_P$ and $v_I(x) = 0$ otherwise with $I \in \mathcal{F}(\mathcal{B}_P)$, page 143
W^L_{can}	denotes the set $\{w \in \Omega^*_k \mid \forall \phi \in \Lambda_L : w(\phi) = 1\}$, page 163
(4)	axiom (4), page 160
(5)	axiom (5), page 160
(B2)	axiom (B2), page 42
(B3)	axiom (B3), page 42
(B4)	axiom (B4), page 42
(D)	axiom (D), page 160
(K)	axiom (K), page 43
(Q1)	axiom (Q1), page 41
(T)	axiom (T), page 160
BL	basic logic, page 36
NP	class of decision problems that can be solved in polynomial time on a non-deterministic Turing machine, page 52

P	class of decision problems that can be solved in polynomial time on a deterministic Turing machine, page 52
Mod(M)	denotes the set $\{I \in \mathcal{P}(A) \mid M \subseteq I\}$ for an ASP program P, page 58
Mod(M)	denotes the set $\{I \in \mathcal{P}(A) \mid M \subseteq I\}$ for $M \in \mathcal{P}(A)$, page 141
Th(S)	belief set of S, page 28
Th(S)	fuzzy belief set of S, page 126

Index