Fuzzy Equilibrium Logic: Declarative Problem Solving in Continuous Domains

STEVEN SCHOCKAERT
Cardiff University
JEROEN JANSSEN
Vrije Universiteit Brussel
DIRK VERMEIR
Vrije Universiteit Brussel

In this paper, we introduce fuzzy equilibrium logic as a generalization of both Pearce equilibrium logic and fuzzy answer set programming. The resulting framework combines the capability of equilibrium logic to declaratively specify search problems, with the capability of fuzzy logics to model continuous domains. We show that our fuzzy equilibrium logic is a proper generalization of both Pearce equilibrium logic and fuzzy answer set programming, and we locate the computational complexity of the main reasoning tasks at the second level of the polynomial hierarchy. We then provide a reduction from the problem of finding fuzzy equilibrium logic models to the problem of solving a particular bilevel mixed integer program (biMIP), allowing us to implement reasoners by reusing existing work from the operations research community. To illustrate the usefulness of our framework from a theoretical perspective, we show that a well-known characterization of strong equivalence in Pearce equilibrium logic generalizes to our setting, yielding a practical method to verify whether two fuzzy answer set programs are strongly equivalent. Finally, to illustrate its application potential, we show how fuzzy equilibrium logic can be used to find strong Nash equilibria, even when players have a continuum of strategies at their disposal. As a second application example, we show how to find abductive explanations from Łukasiewicz logic theories.

Categories and Subject Descriptors: I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving—*Logic programming*; *Uncertainty, "fuzzy", and probabilistic reasoning*; G.1.6 [**Numerical Analysis**]: Optimization—*Integer programming*

General Terms: Theory

Additional Key Words and Phrases: Answer set programming, Equilibrium logic, Fuzzy logics, Łukasiewicz logic

1. INTRODUCTION

Answer set programming (ASP) provides a declarative language which is particularly useful for modeling combinatorial problems [Baral 2003]. A problem instance

Author's address: S. Schockaert, School of Computer Science & Informatics, Cardiff University, 5 The Parade, CF24 3AA Cardiff, UK, s.schockaert@cs.cardiff.ac.uk; J. Janssen and D. Vermeir, Department of Computer Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium, {jeroen.janssen,dvermeir}@vub.ac.be.

is encoded as a set of rules, called a program, of the form $\alpha \leftarrow \beta$. Intuitively, such a rule indicates that the head α should be assumed, whenever the body β is assumed to hold. Typically, α and β are propositional expressions in negation-normal form, although two types of negation may occur in front of atoms, viz. strong negation $\sim$ and negation-as-failure *not*. The intuition of ASP is to apply forward chaining on a set of facts and rules. A strongly negated atom $\sim a$ is then true when the falsity of a can be explicitly derived, whereas $not\, a$ is true when the truth of a cannot be derived. Given a program P, the core idea of the answer set semantics is to designate particular sets of literals as plausible conclusions, i.e., what may be derived using forward chaining. Due to the non-determinism resulting from the use of negation-as-failure, there may be several such sets (or none at all), which are called the answer sets of P. The program P should be such that there is a one-on-one correspondence between the answer sets and the solutions of the considered problem.

Fuzzy answer set programming (FASP) is a generalization of ASP based on the idea of graded truth [Janssen et al. 2009; Medina et al. 2001; Straccia et al. 2009; Van Nieuwenborgh et al. 2007]. A rule of the form $\alpha \leftarrow \beta$ then intuitively means that the truth degree of the expression α must be greater than or equal to the truth degree of β, where α and β are expressions that evaluate to a number in the unit interval $[0, 1]$. By allowing infinitely many truth degrees, it becomes possible to encode problems with variables that range over continuous domains, in a way which is entirely similar to how discrete problems are modeled in classical ASP. It is important to note that, despite referring to the term 'fuzzy', FASP is not about dealing with vagueness or uncertainty. The notion of graded truth, as we use it here, is nothing more than a vehicle to encode certain knowledge in a more compact way; see [Dubois et al. 2000; Dubois and Prade 2001] for a discussion on graded truth and the difference with vagueness and uncertainty. Below, we will often restrict ourselves to rules that are encoded using the connectives from Łukasiewicz logic. From a well-known theorem due to McNaughton [McNaughton 1951], we know that the problems that can be modeled using Łukasiewicz logic are essentially those that can be described as piecewise-linear functions. Hence, when using the Łukasiewicz connectives, FASP is related to mixed integer programming (MIP) in the same way as ASP is related to the boolean SAT problem.

While several authors have already studied FASP, this formalism is by far not as developed as classical ASP. Very little is known about the computational complexity, for example, and almost no techniques are available to calculate the answer sets of a FASP program. One exception is [Janssen et al. 2008], where a translation from a restricted variant of FASP to fuzzy propositional theories is proposed, generalizing the ASSAT technique from classical ASP [Lin and Zhao 2004]. Furthermore, many of the syntactic extensions that have been proposed for ASP, such as e.g., nested rules, have not yet been considered in FASP. In fact, with the exception of some preliminary work on disjunctive FASP [Lukasiewicz and Straccia 2007], most work has been restricted to rules whose head is an atom. Finally, the derivation of theoretical results about FASP is complicated by the fact that no model-theoretic characterization of answer sets has been developed for FASP.

The aim of this paper is to address these issues by applying the idea of graded truth as it is used in FASP to the equilibrium logic of Pearce [Pearce 1997]. Equilibrium logic is one of the most general approaches to ASP, in which programs can be arbitrary propositional theories, with no restrictions on where the two types of negation may occur. When restricted to the syntax of ASP, there is a one-to-one correspondence between the equilibrium models of a program and its answer sets. Equilibrium logic thus achieves two goals at the same time: extending the generality of ASP and providing an elegant model-theoretic characterization of answer sets. Due to its generality, equilibrium logic has proven useful in defining the semantics of various practical extensions to ASP [Ferraris 2005]. Moreover, due to the elegant characterization of answer sets, equilibrium logic has also proven fundamental in establishing important theoretical results [Lifschitz et al. 2001].

We show that the fuzzy equilibrium logic introduced in this paper is a sound extension of both Pearce equilibrium logic and FASP, in the sense that, under the appropriate syntactic restrictions, fuzzy equilibrium logic models correspond to models in Pearce equilibrium logic, and under other syntactic restrictions to answer sets of FASP programs. By developing a fuzzy equilibrium logic, we inherit the generality and intuitive appeal of equilibrium logic. Moreover, due to the way in which fuzzy equilibrium logic models are defined, we can show that reasoning in FASP is located at the second level of the polynomial hierarchy, thus revealing that the computational complexity of FASP is identical to that of ASP in the general case. To the best of our knowledge, this is the first result about the computational complexity of FASP. Moreover, we show how reasoning in fuzzy equilibrium logic can be reduced to bilevel mixed integer programming, which constitutes the first technique for finding answer sets of general FASP programs. Moreover, by making explicit what the intuitive relationship is between FASP and mathematical programming, we can tap into the vast amount of work that has already been done in the operations research community on efficient techniques for mathematical programming. Note that in the same way, many strong ASP solvers rely on the availability of efficient SAT solvers.

We illustrate the usefulness of our approach for both theoretical and practical purposes. First, we show how fuzzy equilibrium logic can be used to verify whether two FASP programs P and Q are strongly equivalent, in the sense that for every FASP program R, the answer sets of $P \cup R$ and $Q \cup R$ coincide. This result is the generalization of a property from [Lifschitz et al. 2001] about the strong equivalence of classical ASP programs. Next, we study two application examples: finding strong (pure) Nash equilibria with continuous strategies, and abductive reasoning in Łukasiewicz logic. Both problems are at the second level of the polynomial hierarchy, and hence, they could in principle be implemented directly in bilevel MIP. Finding such an implementation, and proving its correctness, however, would be far from trivial. In contrast, we show how fuzzy equilibrium logic allows us to encode these problems in a declarative way, similar to how discrete problems are encoded in disjunctive ASP. The actual solutions can then be found by relying on a subsequent compilation to bilevel MIP for finding the solutions.

The paper is structured as follows. In the next section, we provide the relevant background on fuzzy logics, answer set programming, fuzzy answer set programming and Pearce equilibrium logic. Next, Section 3 introduces fuzzy equilibrium logic and discusses its relationship with fuzzy answer set programming and with Pearce equilibrium logic. Section 4 illustrates the usefulness of our framework, discussing strong equivalence of fuzzy answer set programs, finding strong Nash equilibria under continuous strategies, and abductive reasoning in Łukasiewicz logic. In Section 5, we subsequently focus on the computational complexity of the most important reasoning tasks, and we provide an implementation using bilevel MIP. Related work is discussed in Section 6, after which we present our conclusions. The proofs of the main results are provided in the appendix. Finally, note that this paper extends results that appeared in [Schockaert et al. 2009] in a preliminary form.

2. BACKGROUND

Before we introduce fuzzy equilibrium logic, in this section we briefly recall some basic notions about answer set programming, equilibrium logic, fuzzy logic and fuzzy answer set programming. We will also focus on the intuitions underlying these formalisms, which will guide us in developing a fuzzy equilibrium logic.

2.1 Answer Set Programs

In its basic form, an *answer set program* is a collection of rules of the form

$$a \leftarrow b_1, ..., b_n, not\, c_1, ..., not\, c_m \tag{1}$$

where $a, b_1, ..., b_n, c_1, ..., c_m$ are atoms that are taken from a fixed set At. A subset of atoms $V \subseteq At$ is called an *interpretation*. It is important to note that interpretations in ASP encode epistemic states rather than truth, in the sense that $a \in V$ means that a is assumed to be true, while $a \notin V$ means that a is not known to be true (rather than that a is known to be false, as in classical logic). Accordingly, the negation *not* is understood as negation-as-failure: $not\, c_i$ is satisfied unless c_i is known to be true. Intuitively, the body $b_1, ..., b_n, not\, c_1, ..., not\, c_m$ is understood as a conjunction, defining a condition under which the consequent a is assumed to be true. An interpretation V is called a *model* of the rule (1) if it is a classical model of the propositional formula $b_1 \wedge ... \wedge b_n \wedge \neg c_1 \wedge ... \wedge \neg c_m \rightarrow a$; V is called a model of a program if it is a model of every rule in that program. When $m = 0$, the rule (1) is called a *definite rule*, and a collection of definite rules is called a *definite program*. When $n = 0$ and $m = 0$, the rule (1) is called a fact, as it then expresses that a is unconditionally true.

An answer set program encodes a given problem of interest. The core idea of the answer set semantics then is to designate particular models of a program as solutions of that problem; these models are called the answer sets of the program. The *answer set* of a definite program is defined as its unique minimal model. Alternatively, it can be defined as the least fixpoint of an immediate consequence operator, i.e., the atoms that are in the answer set of a definite program are those atoms that can be derived from the rules using forward chaining. To understand the semantics of programs with negation-as-failure, it makes sense to again refer to the idea of forward chaining. During the process of applying forward chaining, at any moment we can distinguish between three situations: (i) an atom a is known to be true, (ii) an atom a is not yet known to be true but it will be when the forward chaining procedure has ended, and (iii) an atom a will not even be known to be true when the procedure has ended. We then have that $not\, c_i$ is satisfied only in the third case, i.e., *not* c_i intuitively means that we will not be able to derive c_i. As a result of this interpretation, the forward chaining procedure is non-deterministic, which is useful to model problems that have more than one solution. Intuitively, to find an answer set of a program with negation-as-failure, we first need to guess what we will be able to derive, and then verify whether the guess is correct.

EXAMPLE 1. *Consider the program* $\{a \leftarrow \mathit{not}\, b, b \leftarrow \mathit{not}\, a\}$. *Initially, we have not yet derived anything, so we may intuitively think that* $\mathit{not}\, a$ *and* $\mathit{not}\, b$ *should be true. This would mean, however, that we could derive both* a *and* b*, which is in conflict with our assumption that* $\mathit{not}\, a$ *and* $\mathit{not}\, b$ *are true. On the other hand, if we assume that we will only be able to derive* a*, we find that* $\mathit{not}\, a$ *is false and* $\mathit{not}\, b$ *is true. Thus we can indeed derive* a *but not* b*, which means that our initial assumption was correct and that* $\{a\}$ *is an answer set. In the same way, we find that* $\{b\}$ *is an answer set of this program.*

The semantics of answer set programs with negation-as-failure can be formally defined using the *Gelfond-Lifschitz reduct* [Gelfond and Lifschitz 1988]. Given an interpretation V, the Gelfond-Lifschitz reduct P^V of a program P can be obtained by replacing every atom of the form $\mathit{not}\, c_i$ by $\top$ if $c_i \notin V$ and by $\bot$ otherwise. This corresponds to removing rules of the form (1) when $\{c_1, ..., c_m\} \cap V \neq \emptyset$, and replacing them by $a \leftarrow b_1, ..., b_n$ otherwise. An interpretation V is then an answer set of a program P if it is an answer set of the definite program P^V. Clearly, this approach directly follows the idea of guessing an interpretation and then verifying whether this guess is consistent with the aforementioned intuition of negation-as-failure. In addition to rules of the form (1), sometimes rules with an empty head are used:

$$\leftarrow b_1, ..., b_n, \mathit{not}\, c_1, ..., \mathit{not}\, c_m \tag{2}$$

Such a rule is called a constraint; it expresses that the body should not be true. An interpretation V satisfies the constraint (2) if V classically satisfies the implication $b_1 \wedge ... \wedge b_n \wedge \neg c_1 \wedge ... \wedge \neg c_m \rightarrow \bot$. An interpretation V is called an answer set of $P \cup C$, with P a set of normal rules and C a set of constraints, if V is an answer set of P and a model of C.

EXAMPLE 2. *The graph coloring problem can easily be described in ASP:*

$$\begin{aligned}
\mathit{green}(X) &\leftarrow \mathit{not}\, \mathit{red}(X), \mathit{not}\, \mathit{blue}(X)\\
\mathit{red}(X) &\leftarrow \mathit{not}\, \mathit{green}(X), \mathit{not}\, \mathit{blue}(X)\\
\mathit{blue}(X) &\leftarrow \mathit{not}\, \mathit{red}(X), \mathit{not}\, \mathit{green}(X)\\
&\leftarrow \mathit{edge}(X, Y), \mathit{red}(X), \mathit{red}(Y)\\
&\leftarrow \mathit{edge}(X, Y), \mathit{green}(X), \mathit{green}(Y)\\
&\leftarrow \mathit{edge}(X, Y), \mathit{blue}(X), \mathit{blue}(Y)
\end{aligned}$$

The first three rules express that every node should have exactly one of the colors green, red, blue. The last three rules are constraints expressing that two nodes which are connected by an edge should have different colors. This pattern of first guessing a solution, using the non-determinism provided by the negation-as-failure, and then using constraints to verify whether the guess was correct is a typical pattern in many answer set programs. Note that these rules contain variables such *as X and Y, which are essentially used to allow for a compact description of the problem. After grounding a program (i.e., instantiating the rules in all meaningful ways), the answer set semantics defines the solutions to the problem. In addition to these six rules, a number of facts of the form $edge(a,b) \leftarrow$ are used to encode which graph we are actually interested in. For example, if we consider the facts $\{(edge(a,b) \leftarrow), (edge(a,c) \leftarrow)\}$, then one of the answer sets is given by $V = \{edge(a,b), edge(a,c), green(a), red(b), red(c)\}$.*

Several syntactic extensions to rules of the form (1) exist, including *disjunctive rules*, in which the head is a disjunction of atoms, and *strong negation*. The strong negation $\sim a$ of an atom a is satisfied if the falsity of a can be established. The semantics of answer set programs with strong negation are essentially defined by treating $\sim a$ as an atom, whose truth is, in effect, independent of the atom a, with the exception that answer sets are required to be consistent in the sense that they cannot contain at the same time a and $\sim a$. Using strong negation, a consistent interpretation V can distinguish between three epistemic states regarding an atom a: (i) nothing is known about the truth of a (i.e., $a \notin V$ and $\sim a \notin V$), (ii) a is known to be true (i.e., $a \in V$ and $\sim a \notin V$), and (iii) a is known to be false (i.e., $a \notin V$ and $\sim a \in V$).

Example 3. *Consider the following program with strong negation:*

$$P = \{(\sim a \leftarrow), (c \leftarrow not\, a), (d \leftarrow \sim a), (e \leftarrow not\, b), (f \leftarrow \sim b)\}$$

The unique answer set of P is given by $V = \{\sim a, c, d, e\}$. Note in particular that we cannot derive f, because we have not explicitly established the falsity of b, while we can derive e because we have not established b's truth.

2.2 Equilibrium Logic

Equilibrium logic was introduced by Pearce with the aim of extending the notion of answer set to general propositional theories [Pearce 1997; 2006]. The formulation of this logic is based on an extension of *the logic of here-and-there* with strong negation. The logic of here-and-there, also known as Smetanich logic, is known to be the strongest intermediate logic that is properly included in classical logic [Chagrov and Zakharyaschev 1997; Pearce 1997]. As will be recalled in Section 2.3, this logic can semantically be characterized as a three-valued logic. Alternatively, however, it can also be characterized in terms of Kripke frames, using a two-valued valuation in two worlds, called h (here) and t (there). The semantics of equilibrium logic is also based on these two worlds, by considering a three-valued valuation in both worlds[1] [Pearce 1997]. In particular, a *valuation* V is defined as a mapping from $\{h,t\} \times At$ to $\{-1,0,1\}$, such that for each atom a for which $V(h,a) \neq 0$, it holds that $V(h,a) = V(t,a)$. The intuition is that $V(w,a) = 1$ means that a is known to be true in world w, $V(w,a) = -1$ means that a is known to be false in world w, and $V(w,a) = 0$ means that the truth of a is unknown in world w. Furthermore, the there-world is assumed to be a refinement of the here-world, i.e.,

[1] Alternatively, it is also possible to use two-valued valuations that assign truth values to literals instead of atoms [Pearce 2006]. The use of three-valued valuations, however, makes the introduction of fuzzy equilibrium logic slightly more intuitive.

atoms whose truth value is unknown 'here' may have a known truth value 'there', but whenever the truth value of a is already known 'here' it has to be the same 'there', hence the requirement that $V(h,a) \neq 0 \Rightarrow V(h,a) = V(t,a)$. The valuation in world t will intuitively play the role of the guess in the Gelfond-Lifschitz reduct. By defining $\leq$ as $\{(h,h),(t,t),(h,t)\}$, valuations are extended to arbitrary formulas as follows:

$$V(w, \sim\alpha) = -V(w,\alpha) \tag{3}$$

$$V(w, \alpha \wedge \beta) = \min(V(w,\alpha), V(w,\beta)) \tag{4}$$

$$V(w, \alpha \vee \beta) = \max(V(w,\alpha), V(w,\beta)) \tag{5}$$

$$V(w, \alpha \rightarrow \beta) = \begin{cases} 1 & \text{if } \forall w' \geq w\,.\,(V(w',\alpha) = 1) \Rightarrow (V(w',\beta) = 1) \\ -1 & \text{if } V(w,\alpha) = 1 \text{ and } V(w,\beta) = -1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

$$V(w, \mathit{not}\,\alpha) = \begin{cases} 1 & \text{if } \forall w' \geq w\,.\,V(w',\alpha) < 1 \\ -1 & \text{if } V(w,\alpha) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The intuition behind the semantics of strong negation $\sim$, conjunction and disjunction is rather straightforward, e.g., a conjunction is known to be true (valuation 1) if both conjuncts are known to be true, it is known to be false (valuation -1) if either of its conjuncts is known to be false, and it is undecided (valuation 0) otherwise. The semantics of implication is such that an implication can only be true 'here' if it is neither violated 'here' nor 'there'. Apart from this, the implication $\alpha \rightarrow \beta$ behaves as material implication, i.e., as $\sim\alpha \vee \beta$. Similarly, $\mathit{not}\,\alpha$ is true, in either world, unless α is true 'there', and essentially behaves as strong negation otherwise. In analogy with logic programming, we will sometimes write the implication $\alpha \rightarrow \beta$ as $\beta \leftarrow \alpha$.

Note that due to the requirement that t is a refinement of h, there are five possibilities for the valuation of an atom a. Hence, the logic defined by (3)–(7) is actually a five-valued logic, which is called $N5$ in [Pearce 2006][2]. When there may be cause for confusion, we will refer to $N5$ valuations to denote $\{h,t\} \times At \rightarrow \{-1,0,1\}$ mappings. An *($N5$) valuation* V is called an ($N5$) model of a set of formulas Θ if for each $\alpha \in \Theta$, it holds that $V(h,\alpha) = V(t,\alpha) = 1$.

Equilibrium logic is obtained from $N5$ logic by restricting attention to particular $N5$ models, which are called *equilibrium models*. Let Lit be the set of all literals, i.e., $Lit = At \cup \{\sim a | a \in At\}$. For a valuation V, let V_h and V_t be the set of literals that are *true* in worlds h and t:

$$V_h = \{l \in Lit | V(h,l) = 1\} \qquad\qquad V_t = \{l \in Lit | V(t,l) = 1\}$$

A model is called *h-minimal* if its *here* world is as little committing as possible, given its particular *there* world[3].

[2]This logic is called $N2$ in [Pearce 1997].
[3]In [Pearce 1997] the notion of h-minimality is defined in a slightly different way. The difference is irrelevant, however, w.r.t. the definition of equilibrium models.

Definition 1. *[Pearce 2006] Let the ordering $\preccurlyeq$ be defined for two $N5$ valuations V and V' as $V \preccurlyeq V'$ iff $V_t = V'_t$ and $V_h \subseteq V'_h$. An $N5$ model V of a set of formulas Θ is then called h-*minimal *if it is minimal w.r.t. $\preccurlyeq$ among all models of Θ, i.e., for every other model V' of Θ it holds that either $V_t \neq V'_t$ or $V_h \not\supset V'_h$.*

Note that minimality refers to the set of *literals* that are verified by a valuation, and not the set of *atoms*. The notion of h-minimality makes the connection with ASP more explicit: what is true 'there' can intuitively be understood as a guess of what can be derived from available knowledge, whereas what is true 'here' can actually be derived. Recall that in ASP we are interested in the case where the guess about what can be derived coincides with what can actually be derived. Accordingly, equilibrium models are h-minimal models whose valuation in h and t coincides.

Definition 2. *[Pearce 1997] A h-minimal model V of a set of formulas Θ is called an* equilibrium model *if $V_h = V_t$.*

The following result shows that equilibrium logic properly extends answer set programming, even when strong negation and disjunctive rules are allowed.

Proposition 1. *[Pearce 1997] Let P be an ASP program and S a consistent set of literals (i.e., a and $\sim a$ cannot be both in S, for any atom in At). Then S is an answer set of P iff there is an equilibrium model V of P such that $S = V_t$.*

For clarity, we will sometimes talk about Pearce equilibrium models, to avoid confusion with the fuzzy equilibrium models introduced below.

Example 4. *Let $\Theta = \{a \leftarrow not\, b, b \leftarrow not\, a\}$, then the model V defined by $V_t = V_h = \{a, b\}$ is not h-minimal, as witnessed by the model V' defined by $V'_t = \{a, b\}$ and $V'_h = \{\}$. Note that the absence of e.g., both a and $\sim a$ in V'_h implies that $V(h, a) = 0$. However, V' is not an equilibrium model because $V'_t \neq V'_h$. It is easy to see that the only equilibrium models are V'' and V''' defined by $V''_t = V''_h = \{a\}$ and $V'''_t = V'''_h = \{b\}$.*

Example 5. *The theory $\Theta = \{(\sim a \leftarrow), (c \leftarrow not\, a), (d \leftarrow \sim a), (e \leftarrow not\, b), (f \leftarrow \sim b)\}$ has a unique equilibrium model V which is defined by $V_t = V_h = \{\sim a, c, d, e\}$.*

2.3 Fuzzy logic

The term *fuzzy logic* is used with different meanings in the literature. Sometimes it refers to a theory of approximate reasoning that was initiated by Zadeh [Zadeh 1975]. Here, however, we use fuzzy logic to denote the class of logics, whose semantics are based on truth degrees that are taken from the unit interval $[0, 1]$. An important subclass of fuzzy logics are residuated t-norm based logics [Hájek 2001]. These latter logics have a syntax which is similar to classical logic. The semantics are based on interpretations that map atoms to values from the unit interval $[0, 1]$. The semantics of logical conjunction are generalized to [0,1] by a class of functions called *t-norms*: symmetric, associative, increasing $[0, 1]^2 \rightarrow [0, 1]$ mappings T that satisfy the boundary condition $T(1, x) = x$ for all $x \in [0, 1]$. Given a left-continuous t-norm T, logical implication can be generalized using the *residuation* I_T of T, defined by

$$I_T(x, y) = \sup\{\lambda | \lambda \in [0, 1] \text{ and } T(x, \lambda) \leq y\}$$

Table I. Semantics of logical connectives in Gödel logic, product logic and Łukasiewicz logic respectively.

t-norm	t-conorm	implicator
$T_m(x,y) = \min(x,y)$	$S_m(x,y) = \max(x,y)$	$I_{T_m}(x,y) = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}$
$T_p(x,y) = x \cdot y$	$S_p(x,y) = x + y - x \cdot y$	$I_{T_p}(x,y) = \begin{cases} 1 & \text{if } x \leq y \\ \frac{y}{x} & \text{otherwise} \end{cases}$
$T_l(x,y) = \max(0, x+y-1)$	$S_l(x,y) = \min(1, x+y)$	$I_{T_l}(x,y) = \min(1, 1-x+y)$

Note that the first argument of I_T corresponds to the antecedent of the implication and the second argument corresponds to the consequent. An important property of such implicators is that $I_T(x,y) = 1$ iff $x \leq y$, which will be important in defining the semantics of rules in FASP. In general, by *implicator* we mean any $[0,1]^2 \to [0,1]$ mapping I which is decreasing in its first argument, increasing in its second argument, and which satisfies $I(0,1) = I(0,0) = I(1,1) = 1$ and $I(1,0)$. Writing $\otimes$ for conjunction[4] and $\to$ for implication, an interpretation V (i.e., a mapping from the set of atoms At to $[0,1]$) defines a valuation $[.]_V$ as follows:

$$[\alpha]_V = \begin{cases} V(\alpha) & \text{if } \alpha = a \text{ for some } a \in At \\ 0 & \text{if } \alpha = 0 \\ T([\alpha_1]_V, [\alpha_2]_V) & \text{if } \alpha = \alpha_1 \otimes \alpha_2 \\ I_T([\alpha_1]_V, [\alpha_2]_V) & \text{if } \alpha = \alpha_1 \to \alpha_2 \end{cases} \tag{8}$$

where α is a propositional formula built from the atoms in At, the constant 0, and the connectives $\otimes$ and $\to$ in the usual way. In addition, logical disjunction $\oplus$ and negation $\neg$ can be introduced as notational abbreviations, choosing $\alpha_1 \oplus \alpha_2 = \neg(\neg\alpha_1 \otimes \neg\alpha_2)$ and $\neg\alpha = \alpha \to 0$. Semantically, disjunction corresponds to a *t-conorm*, i.e., a symmetric, associative, increasing $[0,1]^2 \to [0,1]$ mapping S satisfying the boundary condition $S(0,x) = x$ for all $x \in [0,1]$. If the considered t-norm T is continuous, it is well-known that $[\alpha_1 \otimes (\alpha_1 \to \alpha_2)]_V = \min([\alpha_1]_V, [\alpha_2]_V)$, a property which is called divisibility. In logics based on a continuous t-norm, it is thus possible to define an additional conjunction operator that semantically corresponds to the minimum; we will write this operator as $\wedge$. Similarly, we will write $\vee$ for the maximum. Along the same lines, it is also possible to define a second implication operator by $\neg a \oplus b$; such implications are known as *S-implicators*. Important fuzzy logics are *Gödel logic*, *product logic* and *Łukasiewicz logic*, which are based on the continuous t-norms T_m, T_p and T_l, defined as in Table I. This table also depicts the corresponding t-conorm and residual implicator. Note that in Gödel logic, the two conjunctions $\otimes$ and $\wedge$ coincide, whereas in Łukasiewicz logic, the two implications coincide.

Gödel logic is *intermediate* between intuitionistic logic and classical logic, in the sense that all tautologies from intuitionistic logic are tautologies in Gödel logic, and all tautologies from Gödel logic are classical tautologies, but not the other way around. It is interesting to note that an axiomatization of Gödel logic is obtained by adding to the axioms of intuitionistic logic, the axiom of prelinearity:

[4]Note that [Hájek 2001] uses & for this conjunction instead of $\otimes$.

$(\alpha \rightarrow \beta) \oplus (\beta \rightarrow \alpha)$. Intuitively, this axiom forces the set of truth values to be linearly ordered, which means that we can interpret truth degrees in a numerical fashion. By adding the following axiom to the axiomatization of Gödel logic, the three-valued Gödel logic is obtained [Baaz et al. 2003]:

$$\alpha \vee (\alpha \rightarrow \beta) \vee (\alpha \otimes \beta \rightarrow \gamma) \tag{9}$$

Adding this axiom effectively forces the set of truth values to be at most three. It turns out that this three-valued Gödel logic is exactly the logic of here-and-there that we already encountered in Section 2.2.

Łukasiewicz logic is often used in fuzzy logic applications, because it preserves many desirable properties from classical logic. Moreover, in contrast to Gödel logic and product logic, the residual implicator in Łukasiewicz logic is continuous, which is interesting w.r.t. the robustness of applications. Often, the logic is extended with arbitrary truth constants from $[0, 1] \cap \mathbb{Q}$, in which case it is called *rational Pavelka logic*. Łukasiewicz logic is closely related to *mixed integer programming*, as was first discovered by McNaughton [1951]. Specifically, let α be a formula in Łukasiewicz logic over the variables $\{a_1, \ldots, a_n\}$. For any formula α, there exists a continuous $[0, 1]^n \rightarrow [0, 1]$ function f that maps every $(\lambda_1, \ldots, \lambda_n)$ from $[0, 1]^n$ to the valuation $[\alpha]_V$ of α when V is defined as $V(a_i) = \lambda_i$. McNaughton [1951] showed that a continuous $[0, 1]^n \rightarrow [0, 1]$ function f is definable in this way iff there exists a finite number of linear functions of the form $p_i(x_1, \ldots, x_n) = c^i_1 x_1 + \cdots + c^i_n x_n + c^i$ with $c^i_1, \ldots, c^i_n, c^i$ in $\mathbb{N}$ such that for every $(\lambda_1, ..., \lambda_n) \in [0, 1]^n$ it holds that $f(\lambda_1, ..., \lambda_n) = p_i(\lambda_1, ..., \lambda_n)$ for some i. As a result of this, checking the satisfiability of a Łukasiewicz logic formula corresponds to checking the feasibility of a mixed integer program, i.e., verifying whether a given set of linear inequalities has a solution, given that some designated variables are required to take an integer value. Hähnle [1994] has made this link explicit, by providing an algorithm for translating a Łukasiewicz logic theory into a mixed integer program in a semantics-preserving way. Finally, note that Łukasiewicz logic is also very close to linear logic [Girard 1987] (see e.g., [Ciabattoni and Luchi 1997]).

2.4 Fuzzy Answer Set Programs

FASP combines the ideas of ASP with the notion of graded truth from fuzzy logic. Different variants of FASP have been proposed in the literature; here we consider a variant that is based on [Janssen et al. 2009]. Let $\mathcal{F}_n$ for each $n \in \mathbb{N}$ be a set of $[0, 1]^n \rightarrow [0, 1]$ functions that are monotonic in each argument (either decreasing or increasing), and let $\mathcal{F} = \bigcup_n \mathcal{F}_n$. Typically, the set $\mathcal{F}$ will contain the connectives from a given fuzzy logic, although other choices may be useful as well (e.g., averaging operators). From the set $\mathcal{F}$ and a set of atoms At, formulas are defined inductively as follows. Each atom $a \in At$ is a formula, as well as each constant[5] $\lambda \in [0, 1]$. Furthermore, if $\alpha_1, ..., \alpha_n$ are formulas, and $f \in \mathcal{F}_n$, then $f(\alpha_1, ..., \alpha_n)$ is a formula

[5] When defining the syntax of fuzzy answer set programs, it seems natural to require that constants are taken from $[0, 1] \cap \mathbb{Q}$ to ensure that the language remains recursively enumerable. To define the semantics of fuzzy answer set programs in terms of a generalized reduct operation, however, programs with arbitrary constants from $[0, 1]$ need to be considered from a conceptual point of view. In practice, however, the language can still be restricted to rational constants.

as well. A *(FASP) program* is then defined as a set of rules of the form

$$a \leftarrow \alpha \tag{10}$$

where α is a formula and $a \in At$ is an atom or a constant. In the latter case, the rule is called a *constraint*. If α is a constant from $[0, 1]$, the rule (10) is called a *fact*. Note that in contrast to ASP, the body can contain other connectives than conjunction. In the classical case, this is not needed, since for instance $a \leftarrow b \vee c$ can be expressed by the two rules $a \leftarrow b$ and $a \leftarrow c$. In FASP, this correspondence only holds for the maximum t-norm, because, as will become clear below, e.g., $a \leftarrow b \oplus_l c$ generally allows us to derive a stronger lower bound for a than the two rules $a \leftarrow b$ and $a \leftarrow c$ together.

A *definite (FASP) program* is a FASP program without constraints in which no functions with decreasing arguments are used. An *interpretation* V is a mapping from At to $[0, 1]$; V is called a model of the rule (10) iff $[a]_V \geq [\alpha]_V$, where V is extended to a $[0, 1]$-valued valuation similar to (8). Alternatively, we may interpret the symbol $\leftarrow$ as any residual implicator, and say that V is a model of (10) iff $[a \leftarrow \alpha]_V = 1$. The restriction to residual implicators is important here, to guarantee that $[a \leftarrow \alpha]_V = 1$ iff $V(a) \geq [\alpha]_V$ for an interpretation V. Recall from Section 2.3 that the residual implicator I_T of a left-continuous t-norm T indeed satisfies the property that $I_T(x, y) = 1$ iff $x \leq y$ for all $x, y \in [0, 1]$. Next, V is called a *model* of a program P if it is a model of all rules in P. Intuitively, the fact that $V(a) = \lambda$ for a model V means that the truth value of a is at least λ, i.e., truth values in (F)ASP should be understood as lower bounds on truth values. As in ASP, interpretations thus correspond to epistemic states rather than possible worlds.

The intuition of the notion of an *answer set* is as in classical ASP. The answer set of a definite program is defined as its unique minimal model. Alternatively, it can be defined as the least fixpoint of an immediate consequence operator [Janssen et al. 2009]. In general, when functions with decreasing arguments are used, FASP programs may have several minimal models, not all of which satisfy the intuitive criterion required of an answer set. To define answer sets of general programs, we need the notion of *negative and positive occurrence* of an atom. An atom a by itself is a positive occurrence of that atom. Furthermore, a positive (resp. negative) occurrence of a in e_i is called positive (resp. negative) in $f(e_1, ..., e_n)$ if f is increasing in its i^{th} argument, and it is called negative (resp. positive) if f is decreasing in its i^{th} argument. An interpretation V is an answer set of a FASP program P without constraints iff it is an answer set of the reduct program P^V, which is obtained from P by replacing all negative occurrences of atoms a by their interpretation $V(a)$. Finally, if C is a set of constraints, V is an answer set of $P \cup C$ iff is is an answer set of P and a model of C.

In the following, we use the subscripts m, p, and l to refer to connectives from Gödel logic, product logic and Łukasiewicz logic respectively, e.g., $\rightarrow_l$ is the implication from Łukasiewicz logic. The notations $\wedge$ and $\vee$ will be used for the minimum and maximum, i.e., $\otimes_m = \wedge$ and $\oplus_m = \vee$. In addition, we will use the Kleene-Dienes implication, defined by

$$\alpha \rightarrow_{kd} \beta = \neg\alpha \vee \beta \tag{11}$$

for any interpretation V. The semantics of FASP as it is introduced above treats all

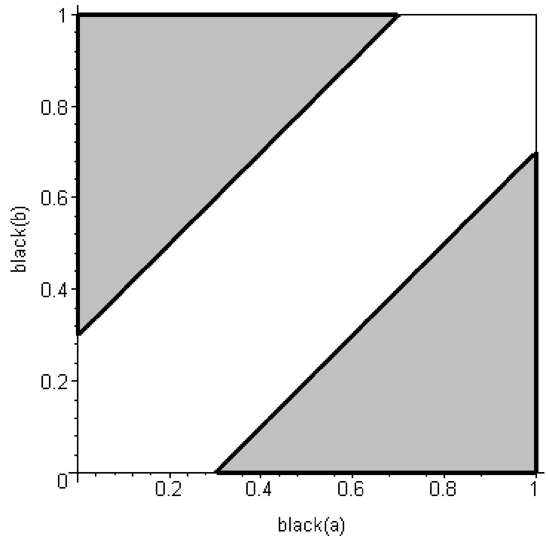

Fig. 1. Visual representation of the answer sets from Example 6.

occurrences of negation as negation-as-failure. We will therefore use the notation *not* for negation, with $[not\,\alpha]_V = 1 - [\alpha]_V$.

Example 6. *Using FASP, a continuous variant of the graph coloring problem could easily be defined. Assume that a weighted graph is given, where edge weights are in* $[0, 1]$ *and specified by facts of the form*

$$edge(a, b) \leftarrow 0.3 \tag{12}$$

The problem now consists of assigning a grey value to each node, such that the distance between the grey values of adjacent nodes is at least as high as the corresponding edge weight. It is not hard to see that for a given interpretation V*, it holds that* $[(x \rightarrow_l y) \otimes_l (y \rightarrow_l x)]_V = 1 - |V(x) - V(y)|$*, where* $\rightarrow_l$ *and* $\otimes_l$ *are the implication and conjunction from Łukasiewicz logic. Indeed, assume for example that* $V(x) \leq V(y)$*, then* $[x \rightarrow_l y]_V = 1$ *and* $[(x \rightarrow_l y) \otimes_l (y \rightarrow_l x)]_V = [y \rightarrow_l x]_V = 1 - V(y) + V(x) = 1 - |V(x) - V(y)|$*, and similar for the case where* $V(x) > V(y)$*. The graph coloring problem could then be expressed as follows:*

$$black(X) \leftarrow not\, white(X) \tag{13}$$

$$white(X) \leftarrow not\, black(X) \tag{14}$$

$$sim(X, Y) \leftarrow (black(X) \rightarrow_l black(Y)) \otimes_l (black(Y) \rightarrow_l black(X)) \tag{15}$$

$$0 \leftarrow edge(X, Y) \otimes_l sim(X, Y) \tag{16}$$

The first two rules express that a node is black to the degree that it is not white. This introduces the required non-determinism, providing the possibility to generate all solutions for which $V(black(n)) = 1 - V(white(n))$ *for all nodes* n*. The third rule then calculates the degree to which two nodes are similar. This degree is used in the last rule, which filters out all solutions in which* $edge(n_1, n_2) \otimes_l sim(n_1, n_2) > 0$*,* *which is equivalent to* $|black(n_1) - black(n_2)| < edge(n_1, n_2)$*. Note the similarity between this FASP program and the ASP program for graph coloring in Example 2. Again the first part of the program generates all possible color assignments, and the second part restricts the allowed assignments to those in which adjacent nodes have a different color. Figure 1 provides a visual representation of the answer sets, in the specific case where a and b are the only nodes, and their edge weight is specified by the fact* (12)*. Note that from the values of* $black(a)$ *and* $black(b)$ *the values of the remaining atoms can easily be found.*

3. FUZZY EQUILIBRIUM LOGIC

We are now ready to present our *fuzzy equilibrium logic*. After introducing the logic itself in Section 3.1, Section 3.2 presents a detailed analysis of the relationship between fuzzy equilibrium logic on the one hand, and Pearce equilibrium logic and FASP on the other. Finally, in Section 3.3, a number of illustrative examples are provided of fuzzy equilibrium logic formulas, clarifying how various types of information can be expressed.

3.1 Definition

In the equilibrium logic of Pearce (and in $N5$), a third truth value, 0, is used to allow for underspecified valuations. This third truth value from equilibrium logic should not be confused with the use of 0 in fuzzy logic, where it stands for complete falsity. When moving from boolean to fuzzy truth degrees, underspecified valuations can be defined by assigning an interval of truth degrees to atoms, as opposed to a precise degree from $[0,1]$. For example, -1 from equilibrium logic corresponds to the degenerate interval $[0,0]$, while 1 corresponds to $[1,1]$ and the third truth value 0 corresponds to $[0,1]$. The restriction that the valuation in the *there*-world should be more specific than the valuation in the *here*-world then translates to the requirement that the interval assigned to an atom in t should be contained in the interval assigned to it in h. Thus we define a *(fuzzy* $N5$*) valuation* V as a mapping from $\{h,t\} \times At$ to (possibly degenerate) subintervals of $[0,1]$ such that $V(h,a) \supseteq V(t,a)$, where At is a set of atoms as before. For $V(w,\alpha) = [u,v]$, we write $V^-(w,\alpha)$ to denote the lower bound u and $V^+(w,\alpha)$ to denote the upper bound v. Next, we propose a semantics for complex formulas. For a constant $\lambda \in [0,1] \cap \mathbb{Q}$, we define $V(h,\lambda) = V(t,\lambda) = [\lambda,\lambda]$. Furthermore, the semantics for strong negation, conjunction, and disjunction follows naturally from the given setting ($w \in \{h,t\}$):

$$V(w, {\sim}\alpha) = [1 - V^+(w,\alpha), 1 - V^-(w,\alpha)] \tag{17}$$

$$V(w, \alpha \otimes \beta) = [V^-(w,\alpha) \otimes V^-(w,\beta), V^+(w,\alpha) \otimes V^+(w,\beta)] \tag{18}$$

$$V(w, \alpha \oplus \beta) = [V^-(w,\alpha) \oplus V^-(w,\beta), V^+(w,\alpha) \oplus V^+(w,\beta)] \tag{19}$$

where we have used the notation $\otimes$ both for the logical connective and the t-norm that implements it, for the ease of presentation (and similar for $\oplus$). For instance, the minimal degree to which the conjunction $\alpha \otimes \beta$ is known to be satisfied in world w is obtained by conjunctively combining the minimal degrees to which α and β are known to be satisfied in that world. Similarly, the truth value of ${\sim}\alpha$ is minimized by considering the maximal value of α, which is indeed the smallest value that ${\sim}\alpha$

can take, given what is known about the possible values of α. In contrast, the idea of negation-as-failure is essentially to take an optimistic attitude, taking as truth value for $not\,\alpha$ the highest value possible. Indeed, in the boolean case, $not\,\alpha$ is true unless α is known to be true. A first idea might be to evaluate $not\,\alpha$ in each world w by the degenerate interval $V(w, not\,\alpha) = [1-V^-(w,\alpha), 1-V^-(w,\alpha)]$. However, this would violate the assumption that t is a refinement of h, as it might be the case that $1 - V^-(h,\alpha) > 1 - V^-(t,\alpha)$. This observation corresponds to the fact that we need to guess a solution in ASP before rules with negation-as-failure can be evaluated. Thus we arrive at:

$$V(h, not\,\alpha) = [1 - V^-(t,\alpha), 1 - V^-(h,\alpha)] \tag{20}$$

$$V(t, not\,\alpha) = [1 - V^-(t,\alpha), 1 - V^-(t,\alpha)] \tag{21}$$

where in both worlds, the lower bound on the truth value of $not\,\alpha$ is based on the valuation in t. This is in accordance with a reading of $not\,\alpha$ as "we will not be able to establish α". The upper bound $V^+(h, not\,\alpha)$ is defined as $1 - V^-(h,\alpha)$, rather than the alternative $1 - V^-(t,\alpha)$, which is in accordance with the fact that in $N5$ logic, a valuation V is a model of $\sim not\,\alpha$ iff it is a model of $\sim\sim\alpha$. Indeed, (20) ensures that $V^-(h, \sim not\,\alpha) = V^-(h, \sim\sim\alpha) = V^-(h,\alpha)$, and in particular that $V^-(h, \sim not\,\alpha) = 1$ iff $V^-(h, \sim\sim\alpha) = 1$.

To generalize the semantics of an implication (i.e., a rule) $\alpha \to \beta$, note that the condition $(V(w',\alpha) = 1) \Rightarrow (V(w',\beta) = 1)$, from $N5$ logic, can be generalized as $V^-(w',\alpha) \to V^-(w',\beta) = 1$ for a suitable implicator $\to$, noting that e.g., $V(w',\alpha) = 1$ in $N5$ logic corresponds to $V(w',\alpha) = [1,1]$ in our setting. Hence, the degree to which the condition $\forall w' \geq h\,.\,(V(w',\alpha) = 1) \Rightarrow (V(w',\beta) = 1)$ is satisfied could be measured by the degree $\min\big(V^-(h,\alpha) \to V^-(h,\beta), V^-(t,\alpha) \to V^-(t,\beta)\big)$, adhering to the common practice of generalizing universal quantification by the infimum in fuzzy logics. Similarly, the condition "$V(w,\alpha) = 1$ and $V(w,\beta) = -1$", from $N5$ logic, is violated iff $V(w,\alpha) = 1 \Rightarrow V(w,\beta) \neq -1$, which can be measured by the degree $V^-(w,\alpha) \to V^+(w,\beta)$. We obtain:

$$\begin{aligned} V(h, \alpha \to \beta) = [&\min\big(V^-(h,\alpha) \to V^-(h,\beta), V^-(t,\alpha) \to V^-(t,\beta)\big), \\ &V^-(h,\alpha) \to V^+(h,\beta)] \end{aligned} \tag{22}$$

$$V(t, \alpha \to \beta) = [V^-(t,\alpha) \to V^-(t,\beta), V^-(t,\alpha) \to V^+(t,\beta)] \tag{23}$$

In analogy with FASP, we will sometimes write a rule $\alpha \to \beta$ as $\beta \leftarrow \alpha$. In contrast to FASP, however, every rule in fuzzy equilibrium logic is tied to a particular residual implicator. When this choice of implicator is important, we will use a subscript (e.g. $\alpha \leftarrow_l \beta$ corresponds to a rule whose semantics is defined in terms of the Łukasiewicz implicator).

As in FASP, in practice it is sometimes convenient to use functions that do not directly correspond to logical connectives. In general, if f is an arbitrary $(m+r)$-ary function which is increasing in its first m arguments and decreasing in its r last arguments, we may define (writing $f(\alpha_1, ..., \beta_r)$ as a shorthand for $f(\alpha_1, ..., \alpha_m, \beta_1, ..., \beta_r)$)

$$V^-(h, f(\alpha_1, ..., \beta_r)) = f(V^-(h,\alpha_1), ..., V^-(h,\alpha_m), V^-(t,\beta_1), ..., V^-(t,\beta_r)) \tag{24}$$

$$V^+(h, f(\alpha_1, ..., \beta_r)) = f(V^+(h,\alpha_1), ..., V^+(h,\alpha_m), V^-(h,\beta_1), ..., V^-(h,\beta_r)) \tag{25}$$

$$V^-(t, f(\alpha_1, ..., \beta_r)) = f(V^-(t, \alpha_1), ..., V^-(t, \alpha_m), V^-(t, \beta_1), ..., V^-(t, \beta_r)) \quad (26)$$
$$V^+(t, f(\alpha_1, ..., \beta_r)) = f(V^+(t, \alpha_1), ..., V^+(t, \alpha_m), V^-(t, \beta_1), ..., V^-(t, \beta_r)) \quad (27)$$

which is in accordance to how the semantics of negation-as-failure *not* is defined in terms of fuzzy logic negation. Note that there is a difference in the semantics of an implication when we see the implication as a rule, in which case the semantics are given by (22)–(23), and when we see it as a function with a decreasing and an increasing argument, in which case the semantics are given by (24)–(27). In this paper, we will always treat occurrences of implications as rules. Alternatively, the semantics of f may be defined as

$$V^-(h, f(\alpha_1, ..., \beta_r)) = f(V^-(h, \alpha_1), ..., V^-(h, \alpha_m), V^+(h, \beta_1), ..., V^+(h, \beta_r))$$
$$V^+(h, f(\alpha_1, ..., \beta_r)) = f(V^+(h, \alpha_1), ..., V^+(h, \alpha_m), V^-(h, \beta_1), ..., V^-(h, \beta_r))$$
$$V^-(t, f(\alpha_1, ..., \beta_r)) = f(V^-(t, \alpha_1), ..., V^-(t, \alpha_m), V^+(t, \beta_1), ..., V^+(t, \beta_r))$$
$$V^+(t, f(\alpha_1, ..., \beta_r)) = f(V^+(t, \alpha_1), ..., V^+(t, \alpha_m), V^-(t, \beta_1), ..., V^-(t, \beta_r))$$

which is in accordance to how the semantics of strong negation $\sim$ is defined in terms of fuzzy logic negation. Unless otherwise specified, we will assume that the semantics given by (24)–(27) are adopted, which is in accordance to the semantics that are used in FASP. In general, however, both types of semantics may be of interest.

A valuation V is a *(fuzzy* $N5$*) model* of a set of formulas Θ if for every α in Θ, $V^-(h, \alpha) = 1$, which also implies $V^+(h, \alpha) = V^-(t, \alpha) = V^+(t, \alpha) = 1$.

Example 7. *We may wonder which of the axioms from $N5$ logic remain tautologies in the setting of fuzzy $N5$ models. Recall that the axiomatization of the logic of here-and-there consists of the axioms of intuitionistic logic and one additional axiom that forces the number of truth degrees to be at most three. Clearly, this latter axiom cannot be satisfied in a fuzzy setting. However, it holds that every fuzzy $N5$ valuation is a model of the following formulas (corresponding to axioms of intuitionistic logic):*

$$(\alpha \otimes_l \beta) \rightarrow_l \alpha \qquad \alpha \rightarrow_l (\alpha \oplus_l \beta)$$
$$(\alpha \otimes_l \beta) \rightarrow_l \beta \qquad \beta \rightarrow_l (\alpha \oplus_l \beta)$$
$$(\alpha \rightarrow_l \mathit{not}\, \beta) \rightarrow_l (\beta \rightarrow_l \mathit{not}\, \alpha) \qquad \alpha \rightarrow_l (\beta \rightarrow_l \alpha)$$
$$\mathit{not}\, (\alpha \rightarrow_l \alpha) \rightarrow_l \beta$$

The validity of these formulas w.r.t. fuzzy $N5$ valuations can be easily shown from the fact that these are all valid formulas in Łukasiewicz logic. However, the following formulas (again corresponding to axioms of intuitionistic logic) are neither valid in Łukasiewicz logic nor in fuzzy equilibrium logic:

$$(\alpha \rightarrow_l (\beta \rightarrow_l \gamma)) \rightarrow_l ((\alpha \rightarrow_l \beta) \rightarrow_l (\alpha \rightarrow_l \gamma))$$
$$(\alpha \rightarrow_l \beta) \rightarrow_l ((\alpha \rightarrow_l \gamma) \rightarrow_l (\alpha \rightarrow_l (\beta \otimes_l \gamma)))$$
$$(\alpha \rightarrow_l \beta) \rightarrow_l ((\alpha \rightarrow_l \gamma) \rightarrow_l ((\alpha \oplus_l \beta) \rightarrow_l \gamma))$$

Hence what we lose w.r.t. intuitionistic logic, from a proof-theoretic point of view, is related to distributivity. When choosing other fuzzy logic connectives, different *axioms from intuitionistic logic will be retained. To get the axioms from $N5$ logic, 6 additional axioms are added, which govern the behavior of strong negation. All of these axioms remain valid w.r.t. the Łukasiewicz connectives, i.e. every fuzzy $N5$ valuation is a model of the following formulas:*

$$\sim(\alpha \rightarrow_l \beta) \leftrightarrow_l \alpha \otimes_l \sim\beta \qquad \sim\sim\alpha \leftrightarrow_l \alpha$$
$$\sim(\alpha \otimes_l \beta) \leftrightarrow_l \sim\alpha \oplus_l \sim\beta \qquad \sim \mathit{not}\,\alpha \leftrightarrow_l \alpha$$
$$\sim(\alpha \oplus_l \beta) \leftrightarrow_l \sim\alpha \otimes_l \sim\beta \qquad \sim a \rightarrow_l \mathit{not}\,a$$

where a is an atom from At and $\alpha \leftrightarrow_l \beta$ is a shorthand for $(\alpha \rightarrow_l \beta) \otimes_l (\beta \rightarrow_l \alpha)$.

Analogous to models in Pearce equilibrium logic, *fuzzy equilibrium models* are models which are in some sense minimal, and which assign the same value to literals in both worlds.

DEFINITION 3. *Let the ordering $\preccurlyeq$ be defined for two fuzzy $N5$ valuations V and V' as $V \preccurlyeq V'$ iff $V(t,a) = V'(t,a)$ and $V(h,a) \supseteq V'(h,a)$ for all $a \in At$. A fuzzy $N5$ model V of a set of fuzzy equilibrium logic formulas Θ is then called* h-minimal *if it is minimal w.r.t. $\preccurlyeq$ among all models of Θ.*

Note that h-minimal fuzzy $N5$ models are those that are least committing, i.e., those whose valuation in the *here*-world corresponds to the largest possible interval.

DEFINITION 4. *An h-minimal fuzzy $N5$ model V of a set of formulas Θ is a* (fuzzy) equilibrium model *if $V(h,a) = V(t,a)$ for all a in At.*

3.2 Relationship to existing frameworks

To see the connection between fuzzy equilibrium models and Pearce equilibrium models, it is useful to observe that the interval $[0,1]$ in fuzzy $N5$ valuations takes the role of 0 (*undecided*) in $N5$ valuations, whereas the degenerate intervals $[0,0]$ and $[1,1]$ take the role of respectively -1 (*false*) and 1 (*true*).

PROPOSITION 2. *Let Θ_1 be a set of equilibrium logic formulas, and let Θ_2 be the set of fuzzy equilibrium logic formulas obtained from Θ_1 by replacing conjunction, disjunction and implication by respectively some t-norm $\otimes$, t-conorm $\oplus$ and implicator $\rightarrow$. Furthermore, let V_1 be an $N5$ valuation, and let V_2 be the fuzzy $N5$ valuation obtained from V_1 by replacing -1, 0 and 1 by respectively $[0,0]$, $[0,1]$ and $[1,1]$. It holds that V_1 is an $N5$ model of Θ_1 iff V_2 is a fuzzy $N5$ model of Θ_2.*

Note that the previous proposition is valid for any t-norm, t-conorm and implicator.

COROLLARY 1. *Let Θ_1, Θ_2, V_1 and V_2 be as in Proposition 2. If V_2 is a fuzzy equilibrium model of Θ_2, then V_1 is a Pearce equilibrium model of Θ_1.*

PROPOSITION 3. *Let Θ be a set of formulas that are built from the constant 0, the atoms from At, and the connectives $\otimes_m$, $\oplus_m$, $\sim$, not and $\rightarrow_{kd}$. If a fuzzy $N5$ valuation V is an h-minimal model of Θ, it holds that $V(h,a) \in \{[0,0],[1,1],[0,1]\}$ for every atom a.*

COROLLARY 2. *Let Θ_1 be a set of equilibrium logic formulas, and let Θ_2 be the set of fuzzy equilibrium logic formulas obtained from Θ_1 by replacing conjunction, disjunction and implication by respectively $\otimes_m$, $\oplus_m$ and $\rightarrow_{kd}$. Furthermore, let*

V_1 be an $N5$ valuation, and let V_2 be the fuzzy $N5$ valuation obtained from V_1 by replacing -1, 0 and 1 by respectively $[0,0]$, $[0,1]$ and $[1,1]$. If V_1 is a Pearce equilibrium model of Θ_1, then V_2 is a fuzzy equilibrium model of Θ_2.

The above results teach us that the set of Pearce equilibrium models of a theory Θ coincides with its set of fuzzy equilibrium models, provided that logical connectives are interpreted in a particular way. As a result, all problems that can be modeled using Pearce equilibrium logic can straightforwardly be modeled in fuzzy equilibrium logic as well, thus showing that fuzzy equilibrium logic is a proper generalization of Pearce equilibrium logic. It is interesting to note that the choice of fuzzy connectives in Proposition 3 and Corollary 2 is important. In particular, when other fuzzy connectives are considered, the counterparts of Pearce equilibrium theories can have additional fuzzy equilibrium models, as illustrated in the following example.

EXAMPLE 8. *Let $\Theta_1 = \{a \leftarrow not\, b, b \leftarrow not\, a\}$ and $\Theta_2 = \{a \leftarrow_l not\, b, b \leftarrow_l not\, a\}$. The set Θ_1 has two equilibrium models V_1 and V_2, defined by $V_1(w,a) = V_2(w,b) = 1$ and $V_1(w,b) = V_2(w,a) = 0$ for $w \in \{h,t\}$. Moreover, a fuzzy $N5$ valuation V is a model of Θ_2 iff*

$$
\begin{aligned}
&(V^-(h, a \leftarrow_l not\, b) = 1) \wedge (V^-(h, b \leftarrow_l not\, a) = 1)\\
&\Leftrightarrow \min(V^-(h,a) \leftarrow_l V^-(h, not\, b), V^-(t,a) \leftarrow_l V^-(t, not\, b)) = 1\\
&\qquad \wedge \min(V^-(h,b) \leftarrow_l V^-(h, not\, a), V^-(t,b) \leftarrow_l V^-(t, not\, a)) = 1\\
&\Leftrightarrow V^-(h,a) \geq 1 - V^-(t,b) \wedge V^-(t,a) \geq 1 - V^-(t,b)\\
&\qquad \wedge V^-(h,b) \geq 1 - V^-(t,a) \wedge V^-(t,b) \geq 1 - V^-(t,a)\\
&\Leftrightarrow V^-(h,a) \geq 1 - V^-(t,b) \wedge V^-(h,b) \geq 1 - V^-(t,a)
\end{aligned}
$$

Thus we find that for every λ in $[0,1]$, the fuzzy $N5$ valuation V defined by $V(t,a) = V(h,a) = [\lambda, 1]$ and $V(t,b) = V(h,b) = [1-\lambda, 1]$ is a fuzzy equilibrium model.

Moreover, when other connectives than the ones from Proposition 3 are used, some of the equilibrium models of an equilibrium logic theory Θ_1 may not be preserved when moving to fuzzy equilibrium logic.

EXAMPLE 9. *Let $\Theta_1 = \{a \vee b \vee b\}$, and $\Theta_2 = \{a \oplus_l b \oplus_l b\}$. The valuation V_2 that was defined in Example 8 is an equilibrium model of Θ_1. In contrast, the corresponding fuzzy $N5$ valuation V defined by $V(w,a) = [0,1]$ and $V(w,b) = [1,1]$ is not a fuzzy equilibrium model of Θ_2. Indeed, it is easy to see that the fuzzy $N5$ valuation W, defined by $W(w,a) = [0,1]$, $W(h,b) = [0.5,1]$ and $W(t,b) = [1,1]$, is a model of Θ_2 as well, which means that V is not h-minimal.*

When translating Pearce equilibrium logic theories into fuzzy equilibrium logic, it is thus important to add constraints that limit valuations to the intervals $[0,0]$, $[1,1]$ and $[0,1]$. One possibility to ensure this is to choose the connectives from Proposition 3 and Corollary 2. Another possibility would be to choose arbitrary t-norms, t-conorms and implicators, but add additional formulas to limit valuations. One possibility would be to add formulas of the form $a \leftarrow (a > 0)$ and $\sim a \leftarrow (\sim a > 0)$ for each $a \in At$ (see Section 3.3 below). Now we turn to the relationship between fuzzy equilibrium logic and fuzzy answer set programming. Recalling that rules in FASP can be modeled using any residual implicator, FASP programs can be seen as special instances of fuzzy equilibrium theories. In general, however, fuzzy equilibrium logic is more sensitive to syntax than FASP. For example, while $a \leftarrow \mathit{not}\,\mathit{not}\,b$ and $a \leftarrow b$ are different in fuzzy equilibrium logic, such a distinction cannot be made in FASP. To show how fuzzy equilibrium logic generalizes FASP, it is thus important to choose the right syntactic encoding. In particular, and without lack of generality, we may assume that a FASP program consists of rules of the form

$$a \leftarrow f(a_1, ..., a_n; b_1, ..., b_m) \tag{28}$$

where f is increasing in the first n arguments and decreasing in the last m arguments. A rule such as (28) can be interpreted as a fuzzy equilibrium logic formula, by replacing $\leftarrow$ by an arbitrary residual implicator, and defining the semantics of f as in (24)–(27).

PROPOSITION 4. *Let P be a FASP program, where rules are of the form* (28) *and are interpreted in fuzzy equilibrium logic by replacing $\leftarrow$ by an arbitrary residual implicator. Assume furthermore that all constants occurring in P are taken from $[0,1] \cap \mathbb{Q}$. Let V be a fuzzy $N5$ valuation, and let W be the interpretation defined by $W(a) = V^-(t,a)$ for all $a \in At$. If V is a fuzzy equilibrium model of P, then W is an answer set of P.*

PROPOSITION 5. *Let P be as in Proposition 4, and let W be an interpretation. Furthermore, let V be the fuzzy $N5$ valuation defined by $V(h,a) = V(t,a) = [W(a), 1]$ for all $a \in At$. If W is an answer set of P then V is a fuzzy equilibrium model of P.*

Hence, whenever the syntax of fuzzy equilibrium theories is restricted to what can be expressed in FASP, the fuzzy equilibrium models coincide with the answer sets. Hence, fuzzy equilibrium logic is also a proper generalization of FASP. Fuzzy equilibrium logic has the practical advantage over FASP that more syntactic constructs can be used, including nested rules, negations in front of complex formulas, etc., as well as the theoretical advantage of having a model-theoretic semantics. In addition to these advantages, which are analogous to the advantages of Pearce equilibrium logic over ASP, fuzzy equilibrium logic has the additional advantage over FASP that it poses less restrictions on the type of connectives that can be adopted. In particular, while rules in FASP are modeled using residual implicators, any type of implicator could be used in fuzzy equilibrium logic.

3.3 Examples of useful constructs

Not all of the syntactic constructs that can be specified in fuzzy equilibrium logic have an intuitive meaning. Practical applications of fuzzy equilibrium logic could result either by restricting attention to those constructs that do have an intuitive meaning, or by using fuzzy equilibrium logic as an interlingua to define non-standard extensions to FASP (e.g., analogous to how the semantics of aggregates in ASP are naturally expressed using Pearce equilibrium logic [Ferraris 2005]). In this section, however, we provide some examples of how fuzzy equilibrium logic formulas may provide a useful generalization of FASP rules. In particular, we first show how two standard techniques for encoding problems in classical answer set programming, viz. *generate-and-test* and *saturation*, can be generalized to the fuzzy equilibrium setting. Then we discuss how techniques for partial rule satisfaction in FASP can easily be implemented using fuzzy equilibrium logic.

3.3.1 *Generate-and-test.* A common approach to encode NP-complete problems in answer set programming consists of writing a program that consists of two parts. In the first part, all possible candidate solutions are generated, while the second part tests which of these candidates constitutes an actual solution. For instance, (13)–(14) are used to guess a grey level assignment, while (15)–(16) are used to verify whether it constitutes a valid graph coloring. The idea of guessing a solution typically means that certain variables may take an arbitrary value.

In the graph coloring example, a candidate solution is guessed using sets of formulas of the form $\Theta = \{a \leftarrow_l \mathit{not}\, b, b \leftarrow_l \mathit{not}\, a\}$. As we know from Example 8, the fuzzy equilibrium models of Θ are such that $V^-(h,a) = V^-(t,a) = 1 - V^-(h,b) = 1 - V^-(t,b) = \lambda$ for an arbitrary $\lambda \in [0,1]$, while $V^+(h,a) = V^+(h,b) = 1 = V^+(t,a) = V^+(t,b) = 1$. By considering $\Theta' = \Theta \cup \{\sim b \leftarrow_l \mathit{not}\,(\sim a), \sim a \leftarrow_l \mathit{not}\,(\sim b)\}$ we obtain a theory whose fuzzy equilibrium models are of the form $V(h,a) = [\lambda,\lambda]$ and $V(h,b) = [1-\lambda, 1-\lambda]$.

Although the previous construction can be extended from 2 formulas involving 2 variables to n formulas involving n variables, for $n \geq 2$, often we just want to assert that a single variable can take an arbitrary truth value. In fuzzy equilibrium logic, we can express this using a formula of the form $a \oplus_l \sim a$. Indeed, the fuzzy equilibrium models of that formula are of the form $V(h,a) = V(t,a) = [\lambda,\lambda]$ for an arbitrary $\lambda \in [0,1]$. Writing $a \oplus_l \sim a$ thus means that a can freely take any value from $[0,1]$. In the graph coloring example, this means that we no longer need the artificial predicate *white*, as we can simply write $\mathit{black}(X) \oplus_l \sim \mathit{black}(X)$ to guess the grey level of each node X. Using the maximum (i.e., the Gödel disjunction) instead of the Łukasiewicz disjunction, we can state that a can take any value from $\{0,1\}$, which is useful when some variables in the problem under consideration are continuous (fuzzy) and others are boolean. Specifically, we have that $V^-(h, a \vee \sim a) = 1$ iff either $V^-(h,a) = 1$ or $V^-(h, \sim a) = 1$, which means $V(h,a) = [0,0]$ or $V(h,a) = [1,1]$. More generally, it is possible in fuzzy equilibrium logic to assert that a takes a value from any given, finite discrete domain. For example, to assert that a takes one of the values $\{0, \frac{1}{3}, \frac{2}{3}, 1\}$, we can use the formula $a \vee \sim a \vee \alpha_{\frac{1}{3}} \vee \alpha_{\frac{2}{3}}$, where $\alpha_{\frac{p}{q}} = (a \leftarrow_l \frac{p}{q}) \wedge (\sim a \leftarrow_l \frac{q-p}{q})$.

To write the second part of the fuzzy equilibrium logic theory, where candidate solutions are tested, it is possible to use constraint rules, as in (F)ASP, or to simply state that certain formulas should be satisfied. In addition to constraints with a logical flavor, in many application domains it is useful to refer to constraints with a numerical flavor. A typical example is verifying whether the truth value of some formula α is greater than some constant, or verifying whether the truth value of some formula α is greater than the truth value of another formula β. To this end, we consider formulas of the form

$$a \leftarrow_l (\alpha > \beta)$$

where $>$ is interpreted as a $\{0,1\}$-valued function which maps to 1 if the inequality holds and to 0 otherwise, i.e., we have $V^-(h, \alpha > \beta) = 1$ iff $V^-(h,\alpha) > V^-(t,\beta)$, and $V^-(t, \alpha > \beta) = 1$ iff $V^-(t, \alpha) > V^-(t, \beta)$. Such strict inequalities are extremely useful in practice, e.g., a rule $w \leftarrow_l (w > 0)$ effectively turns w into a binary atom which evaluates to either 0 or 1.

EXAMPLE 10. *To illustrate the use of generate-and-test in fuzzy equilibrium logic, let us consider the linear assignment problem, where n tasks have to be assigned to n agents such that no two agents are assigned the same task. Let $k_{i,j} \in [0, 1]$ be the cost associated with assigning agent i to task j, and assume that all costs are normalized such that $\sum_{i,j} k_{i,j} \leq 1$. The problem we consider here is deciding whether there exists a matching whose total cost is at most $\lambda \in [0, 1]$. Let us consider the fuzzy equilibrium logic theory, which contains the following formulas for all $1 \leq i, i_1, i_2 \leq n$ with $i_1 \neq i_2$:*

$$\text{in}(i, 1) \vee ... \vee \text{in}(i, n) \tag{29}$$

$$c(i, j) \leftarrow_l \text{in}(i, j) \otimes_l k_{i,j} \tag{30}$$

$$c \leftarrow_l c(1, 1) \oplus_l ... \oplus_l c(1, n) \oplus_l c(2, 1) \oplus_l ... \oplus_l c(n, n) \tag{31}$$

$$0 \leftarrow_l \text{in}(i_1, j) \otimes_l \text{in}(i_2, j) \tag{32}$$

$$0 \leftarrow_l (c > \lambda) \tag{33}$$

The fuzzy equilibrium models of this theory are exactly the matchings whose overall cost is at most λ. The generate-part of the theory are the rules of the form (29)*, where the intuition is that* $\text{in}(i, j)$ *is true iff agent i is assigned task j. Since the maximum is used in these rules, in each fuzzy equilibrium model there will be exactly one j for which $V(h, \text{in}(i, j)) = V(t, \text{in}(i, j)) = [1, 1]$, whereas for all $j' \neq j$, it holds that $V(h, \text{in}(i, j')) = V(t, \text{in}(i, j')) = [0, 1]$. The rules* (30)–(31) *are used to calculate the cost that corresponds to the matching. Finally, incorrect matchings are eliminated by testing in* (32) *that no two agents are assigned the same task, and in* (33) *that the overall cost is indeed at most λ.*

3.3.2 *Saturation.* The generate-and-test technique is useful in situations where a configuration must be found that satisfies certain constraints. Sometimes, however, we need to find a configuration which is optimal in some sense. In Example 10, for instance, we may be interested in finding the matching which minimizes the cost, rather than finding any matching whose cost is below a predefined threshold. This can be accomplished using a saturation technique, which is also commonly used in classical disjunctive logic programming (see e.g., for some typical examples [Eiter et al. 1997]). The idea is to guess a solution, as in the generate-and-test methodology, and then consider a second solution whose values are *saturated* unless its cost is lower than the first solution. We illustrate the technique using the following example.

EXAMPLE 11. *Consider the problem of finding an optimal matching in the setting of Example 10. We consider a fuzzy equilibrium logic theory which consists of the formulas* (29)–(32)*, together with formulas of the following form:*

$$\text{in'}(i, 1) \vee ... \vee \text{in'}(i, n) \tag{34}$$

$$c'(i, j) \leftarrow_l \text{in'}(i, j) \otimes_l k_{i,j} \tag{35}$$

$$c' \leftarrow_l c'(1, 1) \oplus_l ... \oplus_l c'(1, n) \oplus_l c'(2, 1) \oplus_l ... \oplus_l c'(n, n) \tag{36}$$

$$w \leftarrow_l \text{in'}(i_1, j) \otimes_l \text{in'}(i_2, j) \tag{37}$$

$$w \leftarrow_l c' \geq c \tag{38}$$

$$\text{in'}(i, j) \leftarrow_l w \tag{39}$$

$$\text{c'}(i, j) \leftarrow_l w \tag{40}$$

$$c' \leftarrow_l w \tag{41}$$

$$0 \leftarrow_l \mathit{not}\, w \tag{42}$$

The idea is that formulas (29)–(32) *are first used to guess an optimal matching, while formulas* (34)–(38) *are used to find a counterexample to the claim that the first matching were optimal. Due to* (37)–(38)*, we have that w is true (to degree 1) unless a valid counterexample was found. If w is indeed true,* (39)–(41) *cause the atoms occurring in* (34)–(37) *to* saturate*, i.e. to become true to degree 1. As fuzzy equilibrium models are least committing models, by definition, this means that in any fuzzy equilibrium model, the atoms* in'(i, j) *will correspond to a counterexample unless no counterexample exists. Finally,* (42) *encodes the constraint that there should not be a counterexample for the solution encoded by the atoms* in(i, j)*. As a result, in any fuzzy equilibrium model, the valuation of the atoms* in(i, j) *encodes an optimal matching.*

While the saturation technique is perhaps less intuitive at first glance, this same pattern can be used in many application domains. Moreover, the way it is used in fuzzy equilibrium logic is entirely analogous to the way it is used in disjunctive ASP (see e.g., [Eiter et al. 1997] for more illustrations of this technique). In Section 4 the saturation technique will be used to find strong Nash equilibria and to find abductive explanations in Łukasiewicz logic.

3.3.3 *Conditional rules.* Another interesting feature of fuzzy equilibrium logic is the use of nested rules. Nested rules of the form $(\alpha \leftarrow \beta) \leftarrow \gamma$ are useful to encode that the validity of the rule $\alpha \leftarrow \beta$ is conditional on γ. Formally, however, for many types of implication, this formula has the same fuzzy $N5$ models as[6] $\alpha \leftarrow \beta \otimes \gamma$ for a particular conjunction $\otimes$. Indeed, it is not hard to show that

$$\begin{aligned} V^-(h, (\alpha \leftarrow \beta) \leftarrow \gamma) &= \min((V^-(h, \alpha) \leftarrow V^-(h, \beta)) \leftarrow V^-(h, \gamma), \\ &\qquad (V^-(t, \alpha) \leftarrow V^-(t, \beta)) \leftarrow V^-(t, \gamma)) \\ V^-(h, \alpha \leftarrow \beta \otimes \gamma) &= \min(V^-(h, \alpha) \leftarrow V^-(h, \beta) \otimes V^-(h, \gamma), \\ &\qquad V^-(t, \alpha) \leftarrow V^-(t, \beta) \otimes V^-(t, \gamma)) \end{aligned}$$

Moreover, it is well-known that for the residual implicator I_T induced by a left-continuous t-norm T, it holds that $I_T(x, I_T(y, z)) = I_T(T(x, y), z)$ for all $x, y, z \in [0, 1]$, from which we can conclude that $(\alpha \leftarrow \beta) \leftarrow \gamma$ and $\alpha \leftarrow \beta \otimes \gamma$ have the same fuzzy $N5$ models as soon as $\otimes$ and $\leftarrow$ are interpreted by T and I_T respectively. On the other hand, a rule such as $\alpha \leftarrow (\beta \leftarrow \gamma)$ cannot straightforwardly be expressed in FASP. In this latter case, the truth of α is asserted to be conditional on the satisfaction of the rule $\beta \leftarrow \gamma$. The rule $\beta \leftarrow \gamma$ itself is not actually asserted to

[6]Throughout the paper, we let $\otimes$ and $\oplus$ have priority over $\leftarrow$, and we let $\sim$ and *not* have priority over $\otimes$ and $\oplus$; for instance, $\sim a \leftarrow b \oplus \mathit{not}\, c$ is the same as $(\sim a) \leftarrow (b \oplus (\mathit{not}\, c))$.

hold, however. For instance, the only fuzzy equilibrium model V of $\{c, a \leftarrow (b \leftarrow c)\}$ is given by $V(w,a) = V(w,b) = [0,1]$ and $V(w,c) = [1,1]$ for $w \in \{h,t\}$, i.e., the truth of c is not sufficient to derive that b is true.

Along similar lines, we could consider formulas of the form $a \vee (\alpha \leftarrow \beta)$ that essentially express that the rule $\alpha \leftarrow \beta$ is optional, with the atom a encoding whether or not the rule was satisfied. This could be useful to encode constraints such as "at least 4 among the following set of rules should be satisfied", or even "rule r_1 can only be satisfied when rule r_2 is not satisfied". More generally, such a kind of formulas allow us to easily simulate the idea of aggregated fuzzy ASP from [Janssen et al. 2009].

4. ILLUSTRATIVE EXAMPLES

4.1 Strong equivalence

In Pearce equilibrium logic, two forms of equivalence can be considered. First, two sets of equilibrium logic formulas Θ_1 and Θ_2 are called *equivalent* iff Θ_1 and Θ_2 have the same equilibrium models. Second, Θ_1 and Θ_2 are called *strongly equivalent* iff Θ_1 and Θ_2 have the same $N5$ models. Clearly, strong equivalence implies equivalence. The notion of strong equivalence was studied in [Lifschitz et al. 2001], where it was shown that Θ_1 and Θ_2 are strongly equivalent iff $\Theta_1 \cup \Psi$ and $\Theta_2 \cup \Psi$ are equivalent for every set of equilibrium logic formulas Ψ. This characterization makes it clear why strong equivalence might be important for practical applications, e.g., to find sound techniques for rewriting answer set programs into a form that can be more easily implemented. This result can be extended to the setting of fuzzy equilibrium logic. In particular, we have the following proposition.

PROPOSITION 6. *Let Θ_1 and Θ_2 be two sets of fuzzy equilibrium logic formulas. The fuzzy $N5$ models of Θ_1 and Θ_2 coincide iff for every set of fuzzy equilibrium logic formulas Ψ, $\Theta_1 \cup \Psi$ and $\Theta_2 \cup \Psi$ have the same fuzzy equilibrium models.*

As an example of how Proposition 6 could be useful, we consider the use of strict inequalities:

$$\Theta_1 = \{c \leftarrow_l (a > b)\} \tag{43}$$

Such formulas are often very useful in applications, but they are not supported by the implementation that we will introduce in Section 5.4. However, the following theory only uses Łukasiewicz connectives, and is supported by this implementation:

$$\Theta_2 = \{c \leftarrow_l a \otimes_l \mathit{not}\, b, c \leftarrow_l c \oplus_l c\} \tag{44}$$

We have that Θ_2 is strongly equivalent to Θ_1, as any fuzzy $N5$ valuation V is a model of Θ_1 or Θ_2 iff

$$(V^-(h,a) \leq V^-(t,b) \text{ or } V^-(h,c) = 1) \text{ and } (V^-(t,a) \leq V^-(t,b) \text{ or } V^-(t,c) = 1)$$

This means that whenever we want to use a strict inequality in the body of a rule, we may simply replace the rule of the form (43) by rules of the form (44). Alternatively, using the Kleene-Dienes implicator, (43) can even be simulated using only a single rule:

$$c \leftarrow_{kd} a \otimes_l \mathit{not}\, b$$

In the same way, in many applications, we would like to write rules such as $c \leftarrow (a \geq b)$. Unfortunately, such rules cannot straightforwardly be simulated, although formulas of the form $c \vee (a \geq b)$ may be simulated as $c \vee (a \leftarrow \mathit{not}\,\mathit{not}\, b)$. Indeed, both formulas are strongly equivalent as they are satisfied by a valuation V iff either $V^-(h,c) = 1$ or $V^-(h,a) \geq V^-(t,b)$.

4.2 Strong Nash Equilibria

As an illustration of how fuzzy equilibrium logic can be used in the context of declarative problem solving, we present a technique to find strong Nash equilibria, a problem which is known to be Σ_2^P-complete [Gottlob et al. 2005]. Nash equilibria are one of the most fundamental notions from game theory. Assume that a finite set of players $p_1, ..., p_n$ is given, and for each player p_i a set of actions A_i. A choice of actions $(a_1, ..., a_n) \in A_1 \times ... \times A_n$ is called a global strategy. Let μ_i be an $A_1 \times ... \times A_n \to \mathbb{R}$ function representing the utility (or desirability) to player p_i of a certain global strategy. A global strategy $\mathcal{A} = (a_1, ..., a_n)$ is called a (pure) strong Nash equilibrium if there does not exist a non-empty $K \subseteq \{1, ..., n\}$ and a global strategy $\mathcal{A}' = (a'_1, ..., a'_n)$ such that

(1) for all $i \notin K$ it holds that $a_i = a'_i$; and
(2) $\mu_i(\mathcal{A}) < \mu_i(\mathcal{A}')$ for each i in K.

In other words, a global strategy is a strong Nash equilibrium if there cannot be a coalition of players that is able to (strictly) improve the current situation of each of its members, without help of others.

Often, the set of actions that players can choose from is assumed to be finite. Here we allow an infinite number of actions, which can be encoded as a $[0,1]$-valued parameter a. Moreover, we assume that each utility function μ_i can be represented as a formula from Łukasiewicz logic, such that the truth value $[\mu_i(a_1, ..., a_n)]_M$ for a given interpretation M corresponds to the utility of the global strategy $(M(a_1), ..., M(a_n))$. This restriction is rather general, encompassing all scenarios where the number of actions is finite, and essentially those where utility functions are piecewise linear. For the ease of presentation, we rewrite $\mu_i(a_1, ..., a_n)$ using a function of the form (24)–(27). In particular, let U_i be the function which is increasing in all of its arguments and which satisfies $\mu_i(a_1, ..., a_n) = U_i(a_1, ..., a_n; 1 - a_1, ..., 1 - a_n)$.

Now we construct a set of fuzzy equilibrium logic formulas Θ such that the fuzzy equilibrium models of Θ are exactly the strong Nash equilibria that correspond to the given utility functions. For each i in $\{1, ..., n\}$, Θ contains the following formulas:

$$a_i \oplus_l \sim a_i \tag{45}$$

$$c_i^- \oplus_m c_i^+ \tag{46}$$

$$d_i^- \oplus_l d_i^+ \tag{47}$$

$$e_i^+ \leftarrow_l (a_i \otimes_m c_i^-) \oplus_m (d_i^+ \otimes_m c_i^+) \tag{48}$$

$$e_i^- \leftarrow_l (\sim a_i \otimes_m c_i^-) \oplus_m (d_i^- \otimes_m c_i^+) \tag{49}$$

Intuitively, on the first line a strong Nash equilibrium $(a_1, ..., a_n)$ is guessed. The use of the Łukasiewicz t-conorm ensures that in every fuzzy $N5$ model V, it holds that $V(h, a_i) = V(t, a_i) = [\lambda_i, \lambda_i]$ for some $\lambda_i \in [0, 1]$. To verify that we have indeed found a strong Nash equilibrium, a coalition is guessed, defined by the indices c_i^+ and c_i^-; the former intuitively means that player i belongs to the coalition, and the latter that i does not. Note that c_i^- and c_i^+ correspond to crisp properties. By using the maximum, $V(h, c_i^-)$ and $V(h, c_i^+)$ will either be $[0, 1]$ or $[1, 1]$ in h-minimal fuzzy $N5$ models. Next, the strategies of the users in the coalition are guessed: e_i^+ corresponds to the new strategy for player i, whereas e_i^- corresponds intuitively to its negation (i.e., complement with 1). For players outside the coalition this value e_i^+ is simply a_i, whereas for players in the coalition e_i^+ corresponds to a new value d_i^+. To check whether the coalition and strategies that have been guessed constitute a counterexample of the assumption that $(a_1, ..., a_n)$ were a strong Nash equilibrium, the following formulas are added:

$$w \leftarrow_l ((c_1^- \otimes_m ... \otimes_m c_n^-) \leftarrow_{kd} (\alpha_1 \otimes_m ... \otimes_m \alpha_n)) \tag{50}$$

$$w \leftarrow_l e_i^- \otimes_l e_i^+ \tag{51}$$

$$w \leftarrow_l c_i^- \otimes_l c_i^+ \tag{52}$$

$$w \leftarrow_l (w > 0) \tag{53}$$

where α_i is used as an abbreviation of

$$(U_i(e_1^+, ..., e_n^+; e_1^-, ..., e_n^-) > U_i(a_1, ..., a_n; {\sim}a_1, ..., {\sim}a_n)) \oplus_m c_i^- \tag{54}$$

The expression α_i encodes whether player i is either outside the coalition or was able to improve her utility, i.e., α_i is true when player i does not prevent the coalition that was guessed to be a counterexample. Analogous to the simulation of rules with strict inequalities in Section 4.1, (54) can be simulated using Łukasiewicz logic connectives as

$$(U_i(e_1^+, ..., e_n^+; e_1^-, ..., e_n^-) \otimes_l \textit{not}\,(U_i(a_1, ..., a_n; {\sim}a_1, ..., {\sim}a_n))) \oplus_m c_i^-$$

observing that the formulas α_i only appear in the body of the Kleene-Dienes implicator. Intuitively, the atom w is then true whenever the coalition that was guessed does not correspond to a counterexample. The rule (50) ensures that w is true unless none of the players prevent the coalition from being a counterexample, and the coalition is not empty, while (51) and (52) verify whether the coalition itself is valid, e.g., that no player can be simultaneously inside and outside of the coalition. Finally, (53) ensures that w takes a boolean value, i.e., as soon as w can be derived to a non-zero degree, this rule ensures that w is derived to degree 1.

Finally, we need to ensure that all guesses of coalitions and corresponding strategies are tried, rather than just one. In other words, that a fuzzy equilibrium model can only satisfy $V^-(h, w) = 1$ when there are no counterexamples. Specifically, using the saturation technique, the following formulas are added:

$$d_i^- \leftarrow_l w \qquad\qquad d_i^+ \leftarrow_l w \tag{55}$$

$$c_i^- \leftarrow_l w \qquad\qquad c_i^+ \leftarrow_l w \tag{56}$$

$$0 \leftarrow_l \textit{not}\, w \tag{57}$$

Thus, when $(e_1^+, ..., e_n^+)$ does not correspond to a counterexample, all atoms different from a_i are intuitively made true. As a result, in h-minimal models V, we may only have $V^-(h, w) > 0$ if there are no counterexamples. This is the reason why we needed atoms such as c_i^- and d_i^- to simulate the negation of other atoms. Because of this saturation technique, an h-minimal model V may only satisfy $V^-(h, w) = 1$ if there does not exist any counterexample. When there is a counterexample, we will have $V^-(h, w) = 0$, and because of the last rule, V cannot be a fuzzy equilibrium model in such a case. We can show the following proposition.

PROPOSITION 7. *A global strategy* $(\lambda_1, ..., \lambda_n)$ *is a strong Nash equilibrium iff* Θ *has a fuzzy equilibrium model* V *such that* $V^-(t, a_i) = \lambda_i$.

Several relationships between answer set programming and Nash equilibria are already known [Foo et al. 2005; De Vos and Vermeir 2004], although existing approaches do not consider *strong* Nash equilibria nor continuous strategies. Interestingly, an implementation of Nash equilibria using mixed integer programming was proposed in [Sandholm et al. 2005]. Finally note that finding mixed strategy equilibria would require a rather different approach. For instance, rather than using the graded nature of fuzzy equilibrium logic to allow for continuous strategies, we may think of using graded propositions to encode probabilities for a finite number of strategies.

4.3 Fuzzy abductive reasoning

In this section, we consider the problem of logical abduction, which has mainly been studied in classical propositional logic [Eiter and Gottlob 1995] and logic programming [Eiter et al. 1997]. In particular, given a propositional theory T over the set of atoms At, a set of hypotheses (or explanations) $H \subseteq At$, a set of possible manifestations $M \subseteq At$, with $H \cap M = \emptyset$, and a set of observed manifestations $O \subseteq M$, the goal is to find a set $S \subseteq H$ such that $S \cup T$ is consistent and $S \cup T \models O$. In propositional logic, checking the existence of an abductive explanation (existence), as well as deciding whether some $h \in H$ belongs to some abductive explanation (relevance) is Σ_2^P complete, while deciding if some $h \in H$ belongs to all abductive explanations (necessity) is Π_2^P complete. It is interesting to note that these complexity results remain identical when abductive explanations are defined as subset-minimal sets S that satisfy the two aforementioned conditions. Other types of minimality requirements (e.g., subsets of minimal cardinality) give rise to problems of slightly higher complexity [Eiter and Gottlob 1995]. When entailment relations from logic programming are considered, problems that are complete for complexity classes up to the fourth level of the polynomial hierarchy are obtained [Eiter et al. 1997].

Now we consider a natural extension of abductive reasoning to Łukasiewicz logic and, more generally, rational Pavelka logic. Let T be a set of formulas from rational Pavelka logic over the set of atoms At. Let $H \subseteq At$ be a set of possible hypotheses, and let $M \subseteq At$ be a set of possible manifestations, with $H \cap M = \emptyset$. In addition a fuzzy subset O of M is available, where for each m in M, $O(m) \in [0, 1]$ is the intensity to which manifestation m was observed. The problem we consider is to find a fuzzy subset $S \subseteq H$ such that $S \cup T$ is consistent (i.e., has at least one model) and $S \cup T \models O$, where we identify a fuzzy set X of atoms with the set of formulas $\{X(a) \rightarrow a | a \in At, X(a) > 0\}$ for notational convenience; recall that $W \models \lambda \rightarrow a$ in Łukasiewicz logic iff $W(a) \geq \lambda$. For instance, $S \cup T \models O$ means that for every model W of T such that $W(e) \geq S(e)$ for all $e \in H$, it holds that $W(m) \geq O(m)$ for all $m \in M$. Again we call S an abductive explanation for O, and we can consider three important decision problems: checking the existence of an abductive explanation (existence), deciding if there is an abductive explanation S such that $S(h) \geq \lambda$ (λ-relevance), and deciding whether $S(h) \geq \lambda$ for all abductive explanations (λ-necessity).

PROPOSITION 8. *The existence and λ-relevance problems in Łukasiewicz logic and rational Pavelka logic are Σ_2^P hard; the λ-necessity problem is Π_2^P hard.*

Although Łukasiewicz logic provides a natural setting for abductive reasoning, it has not yet been widely considered in this context. However, in [Medina et al. 2001] a procedural semantics is provided for abduction from multi-valued logic programs (which may be based on Łukasiewicz logic among others). In addition, fuzzy and possibilistic versions of the abductive model of Reggia et al. [1985] has already been considered [Dubois and Prade 1995]. Finally, the problem of fuzzy abductive reasoning has also received some attention from the point of view of approximate reasoning [Yamada and Mukaidono 1995; Mellouli and Bouchon-Meunier 2003]. In our setting, a theory T encodes how the input and output variables of a continuous system are related to each other, and the problem of abduction consists of finding possible values of the unknown input parameters, given the values of the observed output parameters.

We now construct a fuzzy equilibrium theory $\Theta = \Theta_1 \cup \Theta_2 \cup \Theta_3$ to find abductive explanations. The set Θ_1 contains the "generate" part of the program, i.e., Θ_1 is used to guess an abductive explanation S for O. To test whether the fuzzy set S effectively is an abductive explanation, two conditions need to be considered which are respectively encoded in Θ_2 and in Θ_3. Specifically, Θ_2 is used to verify that $S \cup T$ is consistent, and Θ_3 is used to verify that $S \cup T \models O$.

4.3.1 *Guessing an abductive explanation.* The set Θ_1 contains the following formulas:

$$\Theta_1 = \{s_e \oplus_l {\sim} s_e | e \in H\} \tag{58}$$
$$\cup \{c_e \leftarrow s_e | e \in H\} \tag{59}$$
$$\cup \{v_e \leftarrow s_e | e \in H\} \tag{60}$$

Intuitively (58) guesses an abductive explanation S, where s_e represents the value of $S(e)$ for $e \in H$. It is not hard to see that any fuzzy $N5$ model of Θ will satisfy $V(h, s_e) = V(t, s_e) = [\lambda_e, \lambda_e]$ for some $\lambda_e \in [0, 1]$ and $e \in H$. The atoms c_e and v_e in (59) and (60) intuitively correspond to two copies of atom e, which will respectively be used to test whether S is consistent with T and whether $S \cup T \models O$.

4.3.2 *Testing consistency.* A set of formulas Θ_2 will be used to ensure that only fuzzy sets S which are consistent with T are considered as abductive explanations. Here and below, we assume that T does not contain any implications and that T is in negation-normal form. This can be accomplished by virtue of the following properties, which are valid in Łukasiewicz logic and rational Pavelka logic

$$[a \rightarrow_l b]_W = [{\sim} a \oplus_l b]_W \tag{61}$$
$$[{\sim}(a \otimes_l b)]_W = [({\sim} a) \oplus_l ({\sim} b)]_W \tag{62}$$

$$[\sim(a \oplus_l b)]_W = [(\sim a) \otimes_l (\sim b)]_W \tag{63}$$

for any interpretation W. Furthermore, the requirements on T can easily be enforced in polynomial time, hence they do not pose any theoretical restrictions. For a formula α in Łukasiewicz logic, we write $\xi(\alpha)$ to denote the corresponding fuzzy equilibrium logic formula, where all occurrences of atom a are replaced by c_a, and occurrences of negation in α are interpreted as strong negation ($\sim$) in $\xi(\alpha)$. The set Θ_2 is then defined by

$$\Theta_2 = \{\xi(\alpha) | \alpha \in T\}$$

LEMMA 1. *Let the fuzzy set S in H be given by $S(e) = \lambda_e$ for all $e \in H$. It holds that S is consistent with T iff $\Theta_1 \cup \Theta_2$ has a fuzzy $N5$ model V such that $V(h, s_e) = V(t, s_e) = [\lambda_e, \lambda_e]$ for all $e \in H$.*

4.3.3 *Testing entailment.* Finally, a set of formulas Θ_3 will be used to verify that only fuzzy sets S for which $S \cup T \models O$ are considered. For $\alpha \in T$, we write $\xi''(\alpha)$ for the formula which is obtained from α by replacing all occurrences of positive literals a by v_a and all occurrences of negative literals $\neg a$ by $\overline{v_a}$. Let us write $T' = \{\xi''(\alpha) | \alpha \in T\}$. Each model of T corresponds to a fuzzy $N5$ model of T' and vice versa, as made explicit in the following lemma.

LEMMA 2. *Let V be a fuzzy $N5$ model of T' such that $V^-(h, v_a) + V^-(h, \overline{v_a}) \leq 1$ for all $a \in At$. The $[0,1]$-valued interpretation W defined by $W(a) = V^-(h, v_a)$ for all $a \in At$ is a model of T. Conversely if W is a model of T, the fuzzy $N5$ valuation V defined by $V(h, v_a) = V(t, v_a) = W(a)$ and $V(h, \overline{v_a}) = V(t, \overline{v_a}) = 1 - W(a)$ is a fuzzy $N5$ model of T'.*

We now define the set Θ_3 as follows

$$\begin{aligned}
\Theta_3 = &\{\xi''(\alpha) | \alpha \in T\} && (64)\\
&\cup \{w \leftarrow \bigwedge_{a \in M} (v_a \geq O(a))\} && (65)\\
&\cup \{w \leftarrow v_a \otimes_l \overline{v_a} | a \in At\} && (66)\\
&\cup \{w \leftarrow (w > 0)\} && (67)\\
&\cup \{v_a \leftarrow w | a \in At\} && (68)\\
&\cup \{\overline{v_a} \leftarrow w | a \in At\} && (69)\\
&\cup \{0 \leftarrow \mathit{not}\, w\} && (70)
\end{aligned}$$

Intuitively, by adding the formulas (64), a model W of $S \cup T$ is guessed. To check whether the entailment $S \cup T \models O$ is valid, we need to verify whether $W(a) \geq O(a)$ in any such model W. To implement this intuition, we again use the saturation method. In particular, an atom w is used which is considered to be true (to degree 1) whenever the model that was guessed does not constitute a valid countermodel for $S \cup T \models O$. In particular, (65) ensures that w is true to degree 1 whenever the model that was guessed is actually a model of O as well. In addition, (66)–(67) ensure that w is true to degree 1 when the condition from Lemma 2 is not satisfied, i.e., when the guess that was made about the atoms v_a and $\overline{v_a}$ does not actually correspond to a model W of T. Finally, (68)–(70) implement the actual saturation, ensuring that only valuations V in which $V^-(h,w) = 1$, corresponding to the scenario where no countermodel for $S \cup T \models O$ exists, can be fuzzy equilibrium models. We thus arrive at the following result.

PROPOSITION 9. *Let H, O, T, and M be defined as before. Let the fuzzy set S in H be defined by $S(e) = \lambda_e$ for all $e \in H$. It holds that S is an abductive explanation for O w.r.t. T iff $\Theta = \Theta_1 \cup \Theta_2 \cup \Theta_3$ has a fuzzy equilibrium model V such that $V(h,s_e) = V(t,s_e) = [\lambda_e, \lambda_e]$ for all $e \in H$.*

5. COMPLEXITY AND IMPLEMENTATION

Throughout this section, we assume that the set of connectives in $\mathcal{F}$ is limited to $\sim$, *not*, $\otimes_l$, $\oplus_l$, $\rightarrow_l$, $\otimes_m$, $\oplus_m$ and $\rightarrow_{kd}$, i.e., the connectives that can be expressed in Łukasiewicz logic (but considering the two types of negation).

5.1 Hardness

We consider three main reasoning tasks: (i) verifying whether a set of fuzzy equilibrium formulas has at least one fuzzy equilibrium model, (ii) verifying whether the truth value of a particular atom is within certain bounds in at least one of these models, and (iii) verifying whether the truth value of a particular atom is within certain bounds in all of these models. These three tasks are the counterparts of the problems known as *existence*, *set-membership* and *set-entailment* in disjunctive logic programming, which are respectively complete for the complexity classes Σ_2^P, Σ_2^P and Π_2^P [Eiter and Gottlob 1993]. Recall that Σ_2^P is the set of problems that can be solved in polynomial time on a non-deterministic Turing machine with an NP oracle, i.e., $\Sigma_2^P = \text{NP}^{\text{NP}}$, while Π_2^P is the set of problems whose complement is in Σ_2^P. In the following, we will show that moving from equilibrium logic (or disjunctive logic programming) to fuzzy equilibrium logic does not increase the computational complexity for the three aforementioned problems. First, however, we establish the following hardness results.

PROPOSITION 10. *Let Θ be a set of fuzzy equilibrium logic formulas. The problem of deciding whether Θ has a fuzzy equilibrium model is Σ_2^P-hard.*

PROPOSITION 11. *Let Θ be a set of fuzzy equilibrium logic formulas, and let $\mu, \lambda \in [0,1] \cap \mathbb{Q}$ with $\mu \leq \lambda$. The problem of deciding whether $V(t,a) \subseteq [\mu, \lambda]$, for $a \in At$, in at least one fuzzy equilibrium model V of Θ is Σ_2^P-hard.*

PROPOSITION 12. *Let Θ be a set of fuzzy equilibrium logic formulas, and let $\mu, \lambda \in [0,1] \cap \mathbb{Q}$ with $\mu \leq \lambda$. If $[\mu, \lambda] \neq [0,1]$, the problem of deciding whether $V(t,a) \subseteq [\mu, \lambda]$, for $a \in At$, in all fuzzy equilibrium models V of Θ is Π_2^P-hard.*

Proving the corresponding membership results is complicated by the fact that, unlike in approaches based on classical logic, not all fuzzy $N5$ models can be guessed in polynomial time on a non-deterministic Turing machine. In some model V, for instance, it might be the case that $V(h,a) = [\frac{p}{q}, 1]$ with p and q co-prime integers such that the binary representation of q requires a number of bits which is exponential in the size of the input (or worse). Indeed, even a simple theory such as $\{a \leftarrow_l not\, b, b \leftarrow_l not\, a\}$ already has an infinite number of answer sets. However, as explained in the following, it is possible to restrict attention to models that can be guessed in polynomial time.

5.2 Structure of the solution space

Next, we analyze the geometrical structure underlying the fuzzy equilibrium models of a given set of formulas Θ, with the aim of gaining further insight into the nature of fuzzy equilibrium logic and of proving the membership results corresponding to the decision problems that were considered in Propositions 10–12. This approach is in line with existing work on (the complexity of) reasoning in Łukasiewicz logic [Mundici 1987; Aguzzoli and Ciabattoni 2000], which is often based on geometrical arguments.

5.2.1 *Characterizing fuzzy $N5$ models using mixed integer programs.* Let Θ be a set of formulas in fuzzy equilibrium logic. Under the given restrictions on the connectives in $\mathcal{F}$, it is straightforward to construct a theory P_Θ in rational Pavelka logic, such that there is a one-on-one correspondence between the fuzzy $N5$ models of Θ and the models of P_Θ. Specifically, for each atom a appearing in Θ, we may consider the atoms a_h^-, a_h^+, a_t^- and a_t^+, which intuitively correspond to the values $V^-(h,a)$, $V^+(h,a)$, $V^-(t,a)$ and $V^+(t,a)$. Let us define $A_h^- = \{a_h^- | a \in At\}$ and similar for A_h^+, A_t^- and A_t^+. For each atom a, the set P_Θ should contain the formulas $a_h^- \to a_t^-$, $a_t^- \to a_t^+$ and $a_t^+ \to a_h^+$, which encode the requirement that $V^-(h,a) \leq V^-(t,a) \leq V^+(t,a) \leq V^+(h,a)$. This already ensures that for every model M of P_Θ, we may define a fuzzy $N5$ valuation V as $V(h,a) = [M(a_h^-), M(a_h^+)]$ and $V(t,a) = [M(a_t^-), M(a_t^+)]$. To restrict attention to models of P_Θ that correspond to fuzzy $N5$ models of Θ, by recursively applying the definitions (17)–(23) to each formula α from Θ, it is straightforward to find a formula f_α in rational Pavelka logic such that whenever V is a fuzzy $N5$ model of Θ, the interpretation M defined by $M(a_h^-) = V^-(h,a)$, $M(a_h^+) = V^+(h,a)$, $M(a_t^-) = V^-(t,a)$ and $M(a_t^+) = V^+(t,a)$ is a model of f_α and vice versa. This procedure is illustrated in the next example.

EXAMPLE 12. *Let $\Theta = \{a \oplus b, c \leftarrow_l not\, a\}$. A fuzzy $N5$ valuation V is a model of $\alpha = a \oplus b$ iff $V^-(h,a) \oplus V^-(h,b) = 1$, and a model of $\beta = c \leftarrow_l not\, a$ iff $((1 - V^-(t,a)) \to V^-(h,c)) \otimes_m ((1 - V^-(t,a)) \to V^-(t,c)) = 1$. We thus obtain*

$$f_\alpha = a_h^- \oplus b_h^- \qquad\qquad f_\beta = (\neg a_t^- \to c_h^-) \otimes_m (\neg a_t^- \to c_t^-)$$

and

$$P_\Theta = \{a_h^- \to a_t^-, a_t^- \to a_t^+, a_t^+ \to a_h^+, b_h^- \to b_t^-, b_t^- \to b_t^+, b_t^+ \to b_h^+, c_h^- \to c_t^-, c_t^- \to c_t^+, c_t^+ \to c_h^+, f_\alpha, f_\beta\}$$

Recall that a linear program is a set of expressions of the form $\lambda_1 x_1 + ... + \lambda_n x_n \leq \lambda_0$, where $\lambda_0, ..., \lambda_n \in \mathbb{Q}$ and $x_1, ..., x_n$ are variables from a given set X, together with a linear objective function of the form $\nu_0 + \nu_1 x_1 + ... + \nu_n x_n$ to be maximized (or minimized). We will call a mapping σ from X to $\mathbb{R}$ a solution of the linear program Γ if it satisfies all inequalities (in the sense that e.g., $\lambda_1\sigma(x_1) + ... + \lambda_n\sigma(x_n) \leq \lambda_0$); σ will be called an optimal solution if it is a solution that maximizes the objective function. A mixed integer program is a linear program, together with additional constraints of the form $x_i \in \mathbb{N}$, i.e., certain variables may be required to take an integer value in any solution. The special case where all variables are constrained to be integers is often called integer programming.

Using a polynomial procedure explained in [Hähnle 1994], it is possible to construct a mixed integer program Γ_Θ over a set of variables $X \supseteq A_h^- \cup A_h^+ \cup A_t^- \cup A_t^+$, such that for every solution σ of Γ_Θ, the restriction of σ to the variables $A_h^- \cup A_h^+ \cup A_t^- \cup A_t^+$ is a model of P_Θ, and conversely, every model of P_Θ can be extended to a solution of Γ_Θ. The number of variables $n = |X|$ is polynomial in the size of P_Θ, and thus also polynomial in the size of Θ.

5.2.2 *Geometrical characterization of fuzzy equilibrium models.* Recall that a polyhedron is the intersection of a finite number of half-spaces. A facet of a polyhedron is the intersection of the polyhedron with one of its bounding hyperplanes; a face of a polyhedron is defined recursively as either the polyhedron itself or one of the faces of its facets. A face of dimension 0 is also called a vertex. Each solution of Γ_Θ can be identified with a point of the n-dimensional Euclidean space, by assigning each of the variables from X to one of the n dimensions. For the ease of presentation, we will identify points from $\mathbb{R}^n$ with the $X \to \mathbb{R}$ mappings they represent. Accordingly, we will write $p(x)$ to denote the value of x in the mapping corresponding to point p.

Geometrically, the solutions of Γ_Θ correspond to the union of a finite number of polyhedra, each of which is bounded by the unit hypercube $[0,1]^n$. Due to a result by Aguzzoli [2000], we know that the coordinates of the vertices of these polyhedra are rational numbers of the form $\frac{q_1}{q_2}$ such that the binary representation of q_1 and q_2 is polynomial in the number of variable occurrences in P_Θ (i.e., the value of q_2 is at most exponential).

To characterize which points in the solution space of Γ_Θ correspond to h-minimal models, let us extend the ordening $\preccurlyeq$ from Definition 3 to points p and q from $[0,1]^n$, by defining $p \preccurlyeq q$ if for all $a \in At$ it holds that $p(a_t^-) = q(a_t^-)$, $p(a_t^+) = q(a_t^+)$, $p(a_h^-) \leq q(a_h^-)$ and $p(a_h^+) \geq q(a_h^+)$. Note that when V_p and V_q are the fuzzy $N5$ valuations corresponding to p and q, it holds that $p \preccurlyeq q$ iff $V_p \preccurlyeq V_q$.

LEMMA 3. *Let $H \subseteq [0,1]^n$ be a polyhedron, let $M \subseteq H$ be the points that are minimal w.r.t. $\preccurlyeq$. Furthermore let F be a face of H and let p be an interior point of F. If $p \in M$ then it holds that $F \subseteq M$.*

From this lemma, it follows that the set of $\preccurlyeq$-minimal points of a polyhedron corresponds to the union of one or more of its faces. Let us denote by $\mathcal{D}(H)$ the set of points that are dominated by a polyhedron H, i.e.

$$p \in \mathcal{D}(H) \Leftrightarrow \exists q \in H \,.\, q \preccurlyeq p$$

Recall that the convex hull $cvx(V)$ of a set of points $V = \{p_1, ..., p_n\}$ is given by $cvx(V) = \{\sum_i \lambda_i p_i \,|\, \lambda_i \geq 0, \sum_i \lambda_i = 1\}$; it is the smallest convex set that contains all the points in V. As the following lemma expresses, the set $\mathcal{D}(H)$ is itself a polyhedron, with vertices that have a representation which is polynomial in the size of Θ.

LEMMA 4. *Let H be a polyhedron and let V be the set of vertices of H. Let V^**

be the set of all points p such that for some $q \in V$, it holds that

$$p(x) \in \begin{cases} \{q(x), 1\} & \textit{if } x \in A_h^- \\ \{q(x), 0\} & \textit{if } x \in A_h^+ \\ \{q(x)\} & \textit{if } x \in A_t^- \cup A_t^+ \\ \{0, 1\} & \textit{otherwise} \end{cases} \tag{71}$$

It holds that $\mathcal{D}(H) \cap [0,1]^n$ is equal to the convex hull of V^*.

In general, the set M of h-minimal fuzzy $N5$ models of Θ corresponds to a finite union of sets of the form $\bigcap_j G_j \setminus \bigcup_i \mathcal{D}(H_i)$ where each G_j is the face of a certain polyhedron, and each H_i is one of the polyhedra in the solution space of Γ_Θ. Indeed, each point which is $\preccurlyeq$-minimal among the points of each of the polyhedra in which it occurs is either h-minimal, or dominated by a point from another polyhedron. The fuzzy equilibrium models of Θ are then geometrically characterized as the intersection of M with the polyhedron E defined by

$$p \in E \Leftrightarrow \forall a \in At \,.\, p(a_h^-) = p(a_t^-) \wedge p(a_h^+) = p(a_t^+)$$

Note that $M \cap E$ (or even M) is not a topologically closed set in general. However, its topological closure $cl(M \cap E)$ corresponds to the finite union of polyhedra, whose vertices have coordinates with a polynomial binary representation. This means that the centre-of-gravity of the vertices of these polyhedra, which belong to $M \cap E$ and thus correspond to fuzzy equilibrium models, are also polynomially representable.

5.3 Membership

From the discussion in Section 5.2, it follows that if Θ has at least one fuzzy equilibrium model, we can guess a fuzzy equilibrium model V in polynomial time on a non-deterministic machine. Moreover, we can verify that V is indeed a fuzzy equilibrium model as follows. First, note that we can verify in polynomial time that V is a fuzzy $N5$ model of Θ, and that $V(t,a) = V(h,a)$ for all a in At. Next, to verify that V is h-minimal, let Γ_Θ^V be the linear relations that are obtained from Γ_Θ by instantiating all variables of the form a_t^- and a_t^+ by respectively $V^-(t,a)$ and $V^+(t,a)$. From the theory of mixed integer programming, we know that a point p minimizing $\sum_a p(a_h^-) - \sum_a p(a_h^+)$ can be found in NP [Schrijver 1998]. Using the fact that $p(a_h^-) \leq V^-(t,a)$ and $p(a_h^+) \geq V^+(t,a)$, it is clear that V is h-minimal iff for this particular point p it holds that $\sum_a p(a_h^-) - \sum_a p(a_h^+) = \sum_a V^-(t,a) - \sum_a V^+(t,a)$. Hence, we have established the following result.

PROPOSITION 13. *Let Θ be a set of fuzzy equilibrium logic formulas. The problem of deciding whether Θ has a fuzzy equilibrium model is in Σ_2^P.*

We can also prove the following membership results.

PROPOSITION 14. *Let Θ be a set of fuzzy equilibrium logic formulas. The problem of deciding whether $V(t,a) \subseteq [\mu, \lambda]$, for $0 \leq \mu \leq \lambda \leq 1$ and $a \in At$, in at least one fuzzy equilibrium model V of Θ is in Σ_2^P.*

PROPOSITION 15. *Let Θ be a set of fuzzy equilibrium logic formulas. The problem of deciding whether $V(t,a) \subseteq [\mu, \lambda]$, for $0 \leq \mu \leq \lambda \leq 1$ and $a \in At$, in all fuzzy equilibrium models V of Θ is in Π_2^P.*

Hence, the main reasoning tasks are in the same complexity class as their counterparts in (disjunctive) answer set programming.

Another problem that might be of interest is determining a fuzzy equilibrium model of a given set Θ which minimizes the value $V^-(h,a)$. However, such a fuzzy equilibrium model might not exist, which is related to the observation that the set $M \cap E$, defined above, is not topologically closed. For instance, V is a fuzzy equilibrium model of the following set of formulas iff $V(h,a) = V(t,a) = [\lambda, \lambda]$ for any $\lambda > 0$, and $V(h,b) = V(t,b) = [1,1]$:

$$\Theta = \{a \oplus_l {\sim} a, b \leftarrow_l a, b \leftarrow_l b \oplus_l b, 0 \leftarrow_l \mathit{not}\, b\}$$

5.4 Implementation

We show in this section how bilevel mixed integer programming (biMIP; [Moore and Bard 1990]) can be used to find a fuzzy equilibrium model of a set of formulas Θ, if one exists. Bilevel mixed integer programming is a form of mathematical programming involving two agents, called the leader and the follower. Let X^l and X^f be two disjoint sets of variables, and let Γ be a set of linear inequalities over the variables in $X^l \cup X^f$. Furthermore, let $F(X^l, X^f)$ and $f(X^l, X^f)$ be two linear expressions in these variables. Intuitively, the leader attempts to minimize $F(X^l, X^f)$ while the follower attempts to minimize $f(X^l, X^f)$. To accomplish this, the leader can only control the variables in X^l whereas the follower controls the variables in X^f. Furthermore, the leader first has to fix the values of the variables in X^l, and the follower subsequently fixes the values of the variables in X^f; note that some variables are designated to be integers and others to be real numbers (as in classical MIP). The leader must thus choose the particular solution of X^l which minimizes $F(X^l, X^f)$, knowing only that the variables X^f will be assigned a value such that $f(X^l, X^f)$ is minimized.

Once the leader has fixed a partial solution σ^l, mapping the variables X^l to their values, finding a partial solution σ^f (which maps the variables X^f to their values), is simply a matter of solving a mixed integer program, called the follower's program:

$$\Phi(\sigma^l) = \operatorname*{argmin}_{\sigma^f} f(\sigma^l(X^l), \sigma^f(X^f)) \qquad \text{with } \sigma^l \cup \sigma^f \text{ a solution of } \Gamma$$

where the set $\Phi(\sigma^l)$ contains all solutions σ^f for which the linear expression $f(X^l, X^f)$ is minimized; note that we write $\sigma^l \cup \sigma^f$ for the solution of Γ defined by σ^l and σ^f, and we write $f(\sigma^l(X^l), \sigma^f(X^f))$ for the value of $f(X^l, X^f)$ under that solution. The actual optimization problem, i.e., the leader's program, is then given by

$$\operatorname*{argmin}_{\sigma^l} F(\sigma^l(X^l), \sigma^f(X^f)) \qquad \text{with } \sigma^f \in \Phi(\sigma^l)$$

When $\Phi(\sigma^l)$ contains more than one solution, different variants may be conceived, including an optimistic variant (in which the solution minimizing $F(\sigma^l(X^l), \sigma^f(X^f))$ is chosen among those in $\Phi(\sigma^l)$) and a pessimistic variant (in which the solution maximizing $F(\sigma^l(X^l), \sigma^f(X^f))$ is chosen). While the issue of developing scalable solvers for biMIP is still an active area of research (see e.g., [Saharidis and Ierapetritou 2009] for a recent contribution), some prototypes are nonetheless available[7].

[7]One example is the YALMIP toolbox for Matlab, available from `http://users.isy.liu.se/`

Now we turn to the problem of finding a fuzzy equilibrium model of a set of formulas Θ. Let Γ_Θ be the corresponding set of linear inequalities whose solutions correspond to the fuzzy $N5$ models of Θ, as before. Let $A_t = \{a_t^-|a \in At\} \cup \{a_t^+|a \in At\}$ and $A_h = \{a_h^-|a \in At\} \cup \{a_h^+|a \in At\}$. The fuzzy equilibrium models of Θ correspond to those solutions σ of Γ_Θ in which $\sigma(a_h^-) = \sigma(a_t^-)$ and $\sigma(a_t^-) = \sigma(a_t^+)$ for all $a \in At$, such that there is no other solution σ' of Γ_Γ for which $\sigma'(a_t^-) = \sigma(a_t^-)$, $\sigma'(a_t^+) = \sigma(a_t^+)$, $\sigma'(a_h^-) \leq \sigma(a_h^-)$ and $\sigma'(a_h^+) \geq \sigma(a_h^+)$ for all a in At, and moreover either (i) $\sigma'(a_h^-) < \sigma(a_t^-)$ and $\sigma'(a_h^+) \geq \sigma(a_t^+)$, or (ii) $\sigma'(a_h^-) \leq \sigma(a_t^-)$ and $\sigma'(a_h^+) > \sigma(a_t^+)$ for some a in At. Hence, fuzzy equilibrium models are found by choosing the values of a_t^- and a_t^+, for each atom a, such that $\sum_{a\in At}(a_t^- - a_h^-) + (a_h^+ - a_t^+)$ is minimized, knowing that the values of the remaining variables are subsequently chosen such that $\sum_{a\in At} a_h^- - a_h^+$ is minimal. This corresponds to a bilevel MIP program, where the leader's program is

$$\operatorname*{argmin}_{\sigma^t} \sum_{a\in At} (\sigma^t(a_t^-) - \sigma^h(a_h^-)) + (\sigma^h(a_h^+) - \sigma^t(a_t^+)) \qquad \text{with } \sigma^h \in \Phi(\sigma^t)$$

and the follower's program is

$$\Phi(\sigma^t) = \operatorname*{argmin}_{\sigma^h} \sum_{a\in At} \sigma^h(a_h^-) - \sigma^h(a_h^+) \qquad \text{with } \sigma^t \cup \sigma^h \text{ a solution of } \Gamma_\Theta$$

and where σ^t defines the value of the variables in A_t and σ^h defines the values of the remaining variables. Note that it does not matter which variant of biMIP is considered, as $\Phi(\sigma^t)$ will be a singleton whenever σ^t defines a fuzzy equilibrium model, in which case σ^h is the unique solution satisfying $\sum_{a\in At} \sigma^h(a_h^-) - \sigma^h(a_h^+) = \sum_{a\in At} \sigma^t(a_t^-) - \sigma^t(a_t^+)$.

PROPOSITION 16. *Let Θ be a set of fuzzy equilibrium logic formulas, and let Λ be the corresponding bilevel program, as defined above. If σ is a solution of Λ such that $\sigma(a_h^-) = \sigma(a_t^-)$ and $\sigma(a_h^+) = \sigma(a_t^+)$ for all $a \in At$, it holds that the fuzzy $N5$ valuation V defined by $V(h,a) = [\sigma(a_h^-), \sigma(a_h^+)]$ and $V(t,a) = [\sigma(a_t^-), \sigma(a_t^+)]$ is a fuzzy equilibrium model of Θ. Moreover, if $\sigma(a_h^-) \neq \sigma(a_t^-)$ or $\sigma(a_h^+) \neq \sigma(a_t^+)$ for some $a \in At$, it holds that Θ has no fuzzy equilibrium models.*

Finally, note that finding a fuzzy equilibrium model V for which $V(h,a) \in [\mu, \lambda]$ can easily be accomplished by adding two formulas to Θ and applying the procedure from Proposition 16; see the proof of Proposition 14 for more details.

Although the syntactic restrictions that we assumed are sufficient for many applications, rules of the form $c \leftarrow (a \geq b)$ cannot be simulated using the Łukasiewicz logic connectives, as we discussed already in Section 4.1, and hence they are not readily supported by the proposed implementation method. On the other hand, it is not hard to see that using rules of the form $c \leftarrow (a \geq b)$ does not increase the computational complexity. Indeed, applying the analysis of Section 5.2 in the presence of such rules yields a set of constraints Γ_Θ which may contain strict inequalities as well as weak inequalities. This has no influence, however, on the proof of the membership results. Although many solvers cannot cope with strict inequalities directly, they can be handled indirectly, e.g., by replacing a linear constraint

`johanl/yalmip/`

such as $a > b$ by $a \geq b + \varepsilon$ for a sufficiently small $\varepsilon > 0$, or by using solvers that can cope with disequalities. In this way, a reasoner for fuzzy equilibrium logic based on Gödel logic may also be implemented. Regarding the product t-norm, recall that reasoning in product logic can be reduced to Mixed Integer Quadratically Constrained Programming (MIQCP) [Bobillo and Straccia 2007]. Hence, we might implement a reasoner for fuzzy equilibrium logic based on product logic using the bilevel version of MIQCP.

6. RELATED WORK

A variety of approaches to fuzzy answer set programming have been proposed in recent years, which mainly differ in terms of the types of connectives they allow, the way in which they handle partial satisfaction of rules, and the truth lattices that are used [Damásio et al. 2004; Damásio and Pereira 2001; Damasio and Pereira 2003; Janssen et al. 2009; Lukasiewicz and Straccia 2007; Straccia 2006; Van Nieuwenborgh et al. 2007]. Most approaches generalize either the fixpoint definition or the minimal model definition of answer sets (stable models), although [Van Nieuwenborgh et al. 2007] generalizes a definition in terms of unfounded sets. Regarding expressive power, typically only rules with literals in the head are considered. One exception is [Lukasiewicz and Straccia 2007] which allows disjunctions of literals in the head of a rule to define a hybridization of FASP with fuzzy description logics. In a quite different context, multi-valued ASP with disjunctions in the head are used in [Sakama and Inoue 1995], as a vehicle to deal with inconsistencies in classical ASP. However, none of these existing approaches can deal with the syntactic flexibility that is provided by (fuzzy) equilibrium logic (e.g., nested rules, negation-as-failure in front of arbitrary formulas, etc.).

The FASP paradigm essentially allows the programmer to encode problems in continuous domains. Hence, fuzzy equilibrium logic and FASP are not about modeling uncertainty, nor about commonsense reasoning. Other extensions of logic programming have been proposed, with the purpose of modeling uncertain rules, based on probability theory [Dekhtyar and Subrahmanian 1997; Lukasiewicz 1998], possibility theory [Nicolas et al. 2006; Bauters et al. 2010] and belief functions [Wan 2009]. Although these approaches are conceptually very different from FASP, due to its capability of modeling continuous phenomena, uncertainty features can often be encoded in FASP [Damásio and Pereira 2000]. In a broader perspective, fuzzy rules have mainly be studied in the areas of fuzzy control and commonsense reasoning [Mamdani and Assilian 1975; Tanaka and Sugeno 1992; Dubois and Prade 1996; Perfilieva et al. 2011]. In this context, fuzzy rules are often called (fuzzy) if-then rules, and they are used for interpolative reasoning, which has proven a fruitful way for implementing controllers of non-linear systems. Despite their reliance on fuzzy set theory, fuzzy if-then rules thus have little in common with FASP, a crucial difference being that reasoning with fuzzy if-then rules is based on manipulating fuzzy sets of more or less plausible values of variables. However, in [Bauters et al. 2010] it is shown how reasoning with fuzzy if-then rules can be approximated in a possibilistic extension of FASP. Furthermore, some authors have looked at formalisms that combine features of logic programming with inference based on fuzzy if-then rules [Baldwin et al. 1995; Cao 2000].

In [Janssen et al. 2008], an implementation of a restricted variant of FASP is presented, in which the heads of rules are constrained to be atoms, and the bodies to be conjunctions of literals. With the exception of [Janssen et al. 2008], to the best of our knowledge, our paper presents the first implementation of a variant of fuzzy answer set programming with an infinite number of truth values. When only a finite number of truth values is considered, the problem of finding answer sets of FASP programs can be reduced to the setting of classical ASP. In [Van Nieuwenborgh et al. 2007], for instance, an implementation of finite-valued FASP is proposed which is based on DLVHEX. When restricted to finite domains, however, FASP is no longer suitable to model phenomena that are inherently continuous; in such a case, FASP programs may still be useful as a compact way of encoding particular kinds of classical ASP programs. Having a finite number of degrees is also enough to deal with degrees of belief that are assigned by an expert. Such degrees occur for instance in possibilistic answer set programming, which is closely related to fuzzy answer set programming under the Gödel semantics [Bauters et al. 2010]. In the special case where the only connective appearing in the bodies of rules is some fixed left-continuous t-norm, FASP programs are similar in spirit to systems of fuzzy relation equations, for which various solution strategies exist [Sanchez 1976; Perfilieva 2004]. Converting a FASP program to a corresponding system of fuzzy relation equations, however, seems only feasible when the dependency graph of the program contains no positive loops (see [Janssen et al. 2008] for more details). To our knowledge, our paper has established the first link between answer set programming (or equilibrium logic) and bilevel programming. Interestingly, however, bilevel programming can be seen as a special case of mathematical programming with equilibrium constraints [Colson et al. 2005], the latter being a version of mathematical programming in which variational inequalities appear as constraints. While mathematical programming with equilibrium constraints is not directly related to logic programming, it is often used to find game theoretic equilibria [Hobbs et al. 2000; Pang and Fukushima 2005].

Another line of related work is the study of formal multi-valued and fuzzy logics. Most work in this area extends classical propositional or predicate logic [Hájek 2001]. One exception is [Takeuti and Titani 1984], where a fuzzy version of intuitionistic logic is proposed, which corresponds, in fact, to the infinite-valued first-order Gödel logic [Baaz and Zach 2000]. It is interesting to note that the fuzzy versions of here-and-there logic and Nelson logic that are proposed in this paper are fuzzy versions of logics that are already multi-valued. However, the intuitive meaning of truth degrees in fuzzy logics and in (extensions) of intermediate logics is entirely different. Indeed, while here-and-there logic and Nelson logic use different truth degrees to encode different attitudes towards the notion of truth (e.g., constructively true vs. not constructively false), fuzzy logics use partial truth to model gradual phenomena.

Finally, equilibrium logic has also been extended in other ways. In [Pearce and Valverde 2004; 2008], for instance, a first-order version of equilibrium logic is developed, while in [Cabalar et al. 2006] a variant of equilibrium logic is introduced to characterize and generalize the well-founded [Van Gelder et al. 1991] and p-stable semantics [Przymusinski 1991] of logic programs. As shown in [Ferraris 2005], it is possible to characterize the equilibrium logic semantics using a reduct operation on propositional theories, which generalizes the Gelfond-Lifschitz reduct. This brings the formulation of equilibrium logic closer to how logic programming semantics are usually defined. It is shown in [Truszczynski 2010] how adapting the aforementioned reduct formulation of equilibrium logic leads to a generalization of the FLP semantics (which deals with recursive aggregates [Faber et al. 2004]) and the supported-model semantics [Marek and Subrahmanian 1992] of logic programs.

7. CONCLUDING REMARKS

We have proposed fuzzy equilibrium logic as a proper generalization of both equilibrium logic in the sense of Pearce and fuzzy answer set programming. The overall computational complexity of the main reasoning tasks was shown to be at the second level of the polynomial hierarchy, which is the same as for Pearce equilibrium logic and for disjunctive ASP. This shows that adding fuzziness to answer set programming does not, in general, imply an increase in the computational complexity. For practical reasoning, we have proposed an implementation based on bilevel mixed integer programming, which is more general and conceptually simpler than existing implementations of fuzzy answer set programming.

From a theoretical point of view, the model-theoretic nature of the fuzzy equilibrium logic semantics provides a convenient vehicle to study the properties of fuzzy answer set programming, and to gain further insight into its behavior (e.g., the meaning of negation-as-failure in a fuzzy setting). In this paper, in addition to establishing the first complexity results for FASP, we have also shown how fuzzy equilibrium logic can be used to determine whether two FASP programs are strongly equivalent.

From an application point of view, fuzzy equilibrium logic seems a suitable candidate for modeling many types of search problems in continuous spaces. This was illustrated using two practical examples: finding strong Nash equilibria, and finding abductive explanations from fuzzy logic theories. In this sense, the relationship between fuzzy equilibrium logic and mathematical programming is analogous to the relationship between answer set programming (or constraint satisfaction) and the boolean SAT problem, where different choices for the fuzzy logic connectives give rise to different variants of mathematical programming. Indeed, while there is a long tradition in building efficient solvers for a variety of mathematical programming techniques, many real-world problems are difficult to directly encode as mathematical optimization problems. Encoding these problems in fuzzy equilibrium logic may provide a more intuitive alternative, while still offering the power of mathematical programming for finding the required solutions. Finally note that the intuitive appeal of fuzzy equilibrium logic, for applications, may be further enhanced by introducing new constructs, on top of the basic syntax, to capture often recurring aspects of problem specification. A similar situation presents itself in classical ASP, where e.g., the introduction of aggregates has allowed a considerably simpler specification of many problems.

Acknowledgments

Steven Schockaert was funded as a postdoctoral fellow of the Research Foundation – Flanders (FWO), and Jeroen Janssen was funded by a joint Research Foundation – Flanders project.

APPENDIX: PROOFS

Proof of Proposition 2

The proof is provided in the online appendix.

Proof of Corollary 1

The proof is provided in the online appendix.

Proof of Proposition 3

The proof is provided in the online appendix.

Proof of Corollary 2

The proof is provided in the online appendix.

Proof of Proposition 4

The proof is provided in the online appendix.

Proof of Proposition 5

The proof is provided in the online appendix.

Proof of Proposition 6

Clearly, if the fuzzy $N5$ models of Θ_1 and Θ_2 coincide, it also holds that the fuzzy equilibrium models of $\Theta_1 \cup \Psi$ and $\Theta_2 \cup \Psi$ coincide, for all Ψ. Conversely, assume that Θ_1 has a fuzzy $N5$ model V which is not a model of Θ_2. We show that there exists some Ψ such that the fuzzy equilibrium models of $\Theta_1 \cup \Psi$ and $\Theta_2 \cup \Psi$ do not coincide.

Let us define the fuzzy $N5$ valuation V' as $V'(h, a) = V'(t, a) = V(t, a)$ for every atom a. Clearly, the fact that V is a model of Θ_1 implies that V' is also a model of Θ_1, which follows from the observation that for any fuzzy equilibrium logic formula α, $V^-(h, \alpha) \leq V^-(t, \alpha)$.

(1) First assume that V' is not a fuzzy $N5$ model of Θ_2. In that case, we may define $\Psi = \{a \leftarrow_l V'^-(t, a) | a \in At\} \cup \{\sim a \leftarrow_l 1 - V'^+(t, a) | a \in At\}$. Clearly, V' is an h-minimal model of Ψ, and thus a fuzzy equilibrium model of $\Theta_1 \cup \Psi$, whereas V' is by assumption not a model of Θ_2 and a fortiori not a fuzzy equilibrium model of $\Theta_2 \cup \Psi$.

(2) Now assume that V' is a fuzzy $N5$ model of Θ_2, which entails that $V \neq V'$. Let us use α as an abbreviation of the following formula:

$$\bigvee_{a \in At} (a \otimes_l (1 - V^-(h, a))) \vee \bigvee_{a \in At} (\sim a \otimes_l V^+(h, a))$$

Note that for any valuation V'', it holds that $V''(h,\alpha) > 0$ iff $V''^-(h,a) > V^-(h,a)$ or $V''^+(h,a) < V^+(h,a)$ for some $a \in At$. Now let us define the program Ψ as follows:

$$\Psi = \{a \leftarrow_l (a \oplus_l \alpha) \wedge V^-(t,a) \mid a \in At\} \tag{72}$$
$$\cup \{\sim a \leftarrow_l (\sim a \oplus_l \alpha) \wedge V^-(t,\sim a) \mid a \in At\} \tag{73}$$
$$\cup \{a \leftarrow_l V^-(h,a) \mid a \in At\} \tag{74}$$
$$\cup \{\sim a \leftarrow_l V^-(h,\sim a) \mid a \in At\} \tag{75}$$

By simple inspection, we can see that V' is a model of Ψ and thus, using the assumption that V' is a model of Θ_2, we find that V' is also a model of $\Theta_2 \cup \Psi$. We show that V' is moreover a fuzzy equilibrium model of $\Theta_2 \cup \Psi$. Suppose that V'' was also a fuzzy $N5$ model with $V'' \preccurlyeq V'$ and $V'' \neq V'$. Since V is not a model of Θ_2 while V'' is a model, it must be the case that $V \neq V''$, which together with the fact that V'' is a model of (74)–(75) means that for some $a \in At$ either $V^-(h,a) < V''^-(h,a)$ or $V^+(h,a) > V''^+(h,a)$. In other words, we have that $V''^-(h,\alpha) > 0$. However, because V'' is a model of (72)–(73), this would mean that $V''^-(h,a) \geq V^-(t,a) = V'^-(h,a)$ and $V''^-(h,\sim a) \geq V^-(t,\sim a) = V'^-(h,\sim a)$ for all a. Hence, we would have $V' \preccurlyeq V''$, contradicting the assumption that $V'' \preccurlyeq V'$ and $V'' \neq V'$. Hence V' is a fuzzy equilibrium model of $\Theta_2 \cup \Psi$.

To complete the proof, it suffices to show that V' cannot be a fuzzy equilibrium model of $\Theta_1 \cup \Psi$. This follows immediately from the fact that $V \preccurlyeq V'$ and $V \neq V'$, together with the observation that V is a model of $\Theta_1 \cup \Psi$ (V is a model of Θ_1 by assumption; the fact that V is a model of Ψ can straightforwardly be checked).

Proof of Proposition 7

The proof is provided in the online appendix.

Proof of Proposition 8

The proof is provided in the online appendix.

Proof of Lemma 1

The proof is provided in the online appendix.

Proof of Lemma 2

The proof is entirely analogously to the proof of Lemma 1.

Proof of Proposition 9

The proof is provided in the online appendix.

Proof of Proposition 10

We show Σ_2^P-hardness by reduction from the existence problem of disjunctive logic programming, which is Σ_2^P-complete [Eiter and Gottlob 1993]. let P_1 be an arbitrary disjunctive logic program. From Corollaries 1 and 2 we know that there exists a fuzzy equilibrium theory P_2 such that there is a one-to-one correspondence of the equilibrium models (i.e., the consistent answer sets) of P_1 and the fuzzy equilibrium models of P_2. Hence P_1 has a consistent answer set iff P_2 has a fuzzy equilibrium model.

Proof of Proposition 11

Let Θ be any set of fuzzy equilibrium logic formulas, and let a be an atom which does not occur in Θ. Now consider the set Θ' which contains all formulas from Θ, together with:

$$\mu \rightarrow_l a \qquad\qquad (1-\lambda) \rightarrow_l \sim\!a$$

It is easy to see that any valuation V is a model of these latter formulas iff $V(h,a) \subseteq [\mu, \lambda]$. Now let V be a fuzzy equilibrium model of Θ. Since a does not occur in Θ, it holds that $V(w,a) = [0,1]$. Clearly, the mapping V' defined by $V'(h,a) = V'(t,a) = [\mu, \lambda]$ and $V'(h,b) = V'(t,b) = V(h,b)$ for all $b \neq a$ is an equilibrium model of Θ'. Conversely, if V' is an equilibrium model of Θ', then $V'(w,a) = [\mu, \lambda]$ and the mapping V defined by $V(h,a) = V(t,a) = [0,1]$ and $V(h,b) = V(t,b) = V'(h,b)$ for $b \neq a$ is a fuzzy equilibrium model of Θ. Hence, we have that Θ has a fuzzy equilibrium model iff Θ' has a fuzzy equilibrium model in which $V(h,a) \subseteq [\mu, \lambda]$. Hence, as deciding whether the former holds is Σ_2^P-hard by Proposition 10, also deciding the latter is Σ_2^P-hard.

Proof of Proposition 12

The proof is analogous to the proof of Theorem 3.7 from [Eiter and Gottlob 1993]. Consider a set of fuzzy equilibrium logic formulas Θ, and let a be an atom which does not occur in Θ. Now consider the set $\Theta' = \Theta \cup \{a \rightarrow_l a\}$. Clearly, in any fuzzy equilibrium model of Θ, it holds that $V(w,a) = [0,1]$, hence any fuzzy equilibrium model of Θ is also a model of Θ'. Moreover, if $\mu > 0$ or $\lambda < 1$, it holds that $V(h,a) = [0,1] \not\subseteq [\mu, \lambda]$. Hence, we have that Θ has a fuzzy equilibrium model iff Θ' has a fuzzy equilibrium model in which $V(h,a) \not\subseteq [\mu, \lambda]$. Since deciding whether Θ has a fuzzy equilibrium model is Σ_2^P-hard, the stated follows.

Proof of Lemma 3

Let $q \neq p$ be a point from F and assume that q were not $\preccurlyeq$-minimal. Then there exists an $r \in M$ such that $r \preccurlyeq q$ with $r \neq q$. This means $r(x) = q(x)$ for all x in $A_t^- \cup A_t^+$, $r(x) \leq q(x)$ for all $x \in A_h^-$ and $r(x) \geq q(x)$ for all $x \in A_h^+$. Hence, for any point s and $\varepsilon > 0$, it also holds that $s + \varepsilon\overrightarrow{qr} \preccurlyeq s$, as adding $\varepsilon\overrightarrow{qr}$ amounts to decrease $s(x)$ for some $x \in A_h^-$ and increase $s(x)$ for some $x \in A_h^+$.

(1) Assume that $r \in F$. As p is an interior point, there would be an $\varepsilon > 0$ such that $p + \varepsilon \cdot \overrightarrow{qr}$ belongs to F, which would contradict the assumption that p corresponds to an h-minimal model.

(2) If $r \notin F$, we find that $p + \varepsilon \cdot \overrightarrow{qr}$ would not belong to F for any $\varepsilon > 0$. Because polyhedra are convex and p is an internal point of F, however, there would still be some $\varepsilon > 0$ for which $p+\varepsilon\cdot\overrightarrow{qr}$ belongs to the polyhedron, again contradicting the assumption that p corresponds to an h-minimal model. Indeed, for each $\varepsilon > 0$ we may a consider the line L_ε through the points r and $p + \varepsilon \cdot \overrightarrow{qr}$. As p is an internal point of F, for ε sufficiently small, the line L_ε intersects the face

F at some point p'. Since p' and q are both in H, we find that $p + \varepsilon \cdot \overrightarrow{qr} \in H$ from the convexity of H.

Proof of Lemma 4

First, note that clearly, the points from $[0,1]^n$ that are dominated by a fixed point q from V are exactly those that are in the convex hull of the points p that satisfy condition (71). We now show that the points from $[0,1]^n$ that are dominated by any point from H are those that are in the convex hull of V^*.

(1) Let $r \in \mathcal{D}(H)$. Then there is a point q from H such that $q \preccurlyeq r$. Since polyhedra are convex by definition, the point q can be written as a convex combination of the vertices of H, i.e., $q = \sum_{i=1}^{m} \lambda_i v_i$, where we assume $V = \{v_1, ..., v_m\}$, and $\lambda_1, ..., \lambda_m \in [0,1]$ are such that $\lambda_1 + ... + \lambda_m = 1$. We need to show that $r = \sum_i \mu_i w_i$ where all points w_i belong to the convex hull of V^* and $\sum_i \mu_i = 1$. If $r = q$, we may simply choose $\lambda_i = \mu_i$ and $v_i = w_i$, as all vertices of H belong to the convex hull of V^*. Now suppose that $q(x) \neq r(x)$ (which means $x \notin X \setminus (A_t^- \cup A_t^+)$). For instance, assume that $x = a_h^-$ (the proof for the cases where $x \in A_h^+$ and $x \in X \setminus (A_h^- \cup A_h^+ \cup A_t^- \cup A_t^+)$ is entirely analogous), in which case we have $q(a_h^-) < r(a_h^-)$. For each vertex v_i of H, we may consider the point v_i^* defined by $v_i^*(a_h^-) = 1$ and $v_i^*(y) = v_i(y)$ for all $y \neq a_h^-$. Clearly the points $v_1^*, ..., v_m^*$ belong to the convex hull of V^*. Now define the point q' as (note that $q(a_h^-) < 1$ follows from $q(a_h^-) < r(a_h^-)$)

$$q'(x) = \frac{1 - r(a_h^-)}{1 - q(a_h^-)} q(x) + \frac{r(a_h^-) - q(a_h^-)}{1 - q(a_h^-)} (\sum_i \lambda_i v_i^*(x))$$

Note that q' is a convex combination of points that are in the convex hull of V^*. Furthermore, using the fact that $\sum_i \lambda_i v_i^*(a_h^-) = \sum_i \lambda_i = 1$, we find

$$q'(a_h^-) = \frac{1 - r(a_h^-)}{1 - q(a_h^-)} \cdot q(a_h^-) + \frac{r(a_h^-) - q(a_h^-)}{1 - q(a_h^-)} = \frac{r(a_h^-) - q(a_h^-) r(a_h^-)}{1 - q(a_h^-)} = r(a_h^-)$$

while for $y \neq a_h^-$, we have $q'(y) = q(y)$. If $q' = r$, the proof is complete. Otherwise, there is an $x \notin X \setminus (A_t^- \cup A_t^+)$ such that $q'(x) \neq r(x)$ and we may simply repeat the argument.

(2) Let q be a point in the convex hull of V^*. It holds that q can be written as a convex combination of the form $\sum \lambda_i w_i$ where $w_i \in V^*$. Moreover, for each $w_i \in V^*$ there is some $w_i' \in V$ such that $w_i' \preccurlyeq w_i$. We immediately find that $q' = \sum \lambda_i w_i'$ is a point of H and $q' \preccurlyeq q$.

Proof of Proposition 14

Membership in Σ_2^P is easily shown, using the following algorithm: guess a fuzzy $N5$ model V of Θ (in polynomial time), verify that it is a fuzzy equilibrium model (using one call to the NP oracle), and that it satisfies the requirement $V(t,a) \subseteq [\mu, \lambda]$ (constant time). The fact that a fuzzy equilibrium model V can be guessed in polynomial time, follows from the fact that the intersection of $cl(M \cap E)$, with M and E defined as in Section 5.3, with the half-spaces defined by $a_t^- \geq \mu$ and $a_t^+ \leq \lambda$ is still the finite union of polyhedra whose vertices have a polynomial binary representation.

Here, we also present an alternative proof, which reduces this problem to the problem from Proposition 13. In particular, it can easily be verified that Θ has some fuzzy equilibrium model V satisfying $V(t,a) \subseteq [\mu, \lambda]$ iff Θ' has at least one answer set, where

$$\Theta' = \Theta \cup \{not\, a \rightarrow_l (1-\mu), not\, (\sim a) \rightarrow_l \lambda\}$$

Indeed, it is straightforward to verify that

$$V^-(h, not\, a \rightarrow_l (1-\mu)) = 1 \Leftrightarrow \mu \leq V^-(t,a)$$
$$V^-(h, not\, (\sim a) \rightarrow \lambda) = 1 \Leftrightarrow V^+(t,a) \leq \lambda$$

Since $\mu \leq V^-(t,a)$ and $V^+(t,a) \leq \lambda$ only constrain the world t, and not h, they act as constraints, requiring that a takes a truth value from $[\mu, \lambda]$, but without providing support for it.

Proof of Proposition 15

To show membership in Π_2^P, we provide a Σ_2^P algorithm which decides whether Θ has at least one fuzzy equilibrium model V for which $V(t,a) \not\subseteq [\lambda, \mu]$. In particular, it suffices to guess a fuzzy $N5$ model V of Θ (in polynomial time), verify that it is a fuzzy equilibrium model (using one call to the NP oracle), and that it satisfies the requirement $V(t,a) \not\subseteq [\mu, \lambda]$ (constant time). The fact that a fuzzy equilibrium model V can be guessed in polynomial time, follows from the fact that the closure of the intersection of $M \cap E$, with M and E defined as in Section 5.3, and the open half-space defined by $a_t^- < \mu$ is still the finite union of polyhedra whose vertices have a polynomial binary representation. Similarly, the closure of the intersection of $M \cap E$ with the open half-space defined by $a_t^+ > \lambda$ is the finite union of polyehdra whose vertices have a polynomial binary representation.

Proof of Proposition 16

Assume that σ is a solution of Λ such that $\sigma(a_h^-) = \sigma(a_t^-)$ and $\sigma(a_h^+) = \sigma(a_t^+)$ for all $a \in At$, but that the corresponding fuzzy $N5$ model V were not a fuzzy equilibrium model of Θ. However, V is a fuzzy $N5$ model of Θ by construction, since $\sigma = \sigma^t \cup \sigma^h$ is solution of Γ_Θ. Moreover, the fact that $\sigma(a_h^-) = \sigma(a_t^-)$ and $\sigma(a_h^+) = \sigma(a_t^+)$ holds teaches us that $V(h,a) = V(t,a)$ for every a in At, which is only possible if V is not h-minimal (observing that V was assumed not to be a fuzzy equilibrium model). Therefore there exists a fuzzy $N5$ model V' of Θ such that $V'(t,a) = V(t,a)$ and $V(h,a) \subseteq V'(h,a)$ for all $a \in At$, while $V(h,a_0) \subset V'(h,a_0)$ for a particular $a_0 \in At$. Now let σ' be the solution of Γ_Θ corresponding to V', with σ'^t and σ'^h respectively the restrictions to A^t and A^h. We immediately find that $\sigma^t = \sigma'^t$ while $\sum_{a \in At} \sigma^h(a_h^-) - \sigma^h(a_h^+) > \sum_{a \in At} \sigma'^h(a_h^-) - \sigma'^h(a_h^+)$, a contradiction, since σ^h was supposed to be a solution minimizing $\sum_{a \in At} a_h^- - a_h^+$.

Assume that there is a solution σ of Λ such that $\sigma(a_h^-) \neq \sigma(a_t^-)$ or $\sigma(a_h^+) \neq \sigma(a_t^+)$ for some $a \in At$, while Θ has some fuzzy equilibrium model V'. Let σ' be the solution of Γ_Θ corresponding to V' and let σ^t, σ^h, σ'^t and σ'^h be as before. Then $\sum_{a \in At}(\sigma'^t(a_t^-) - \sigma'^h(a_h^-)) + (\sigma'^h(a_h^+) - \sigma'^t(a_t^+)) = 0$ while $\sum_{a \in At}(\sigma^t(a_t^-) - \sigma^h(a_h^-)) + (\sigma^h(a_h^+) - \sigma^t(a_t^+)) > 0$, a contradiction since σ is a solution minimizing $\sum_{a \in At}(a_t^- - a_h^-) + (a_h^+ - a_t^+)$.

REFERENCES

AGUZZOLI, S. AND CIABATTONI, A. 2000. Finiteness in infinite-valued Łukasiewicz logic. *Journal of Logic, Language, and Information 9*, 5–29.

BAAZ, M., CIABATTONI, A., AND FERMÜLLER, C. 2003. Hypersequent calculi for Gödel logics — a survey. *Journal of Logic and Computation 13,* 6, 835–861.

BAAZ, M. AND ZACH, R. 2000. Hypersequents and the proof theory of intuitionistic fuzzy logic. In *Computer Science Logic*, P. Clote and H. Schwichtenberg, Eds. Vol. 1862. Springer Berlin / Heidelberg, 187–201.

BALDWIN, J. F., MARTIN, T. P., AND PILSWORTH, B. W. 1995. *Fril - Fuzzy and Evidential Reasoning in Artificial Intelligence.* John Wiley & Sons, Inc., New York, NY, USA.

BARAL, C. 2003. *Knowledge Representation, Reasoning and Declarative Problem Solving.* Cambridge University Press.

BAUTERS, K., SCHOCKAERT, S., DE COCK, M., AND VERMEIR, D. 2010. Possibilistic answer set programming revisited. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence.*

BAUTERS, K., SCHOCKAERT, S., JANSSEN, J., DE COCK, M., AND VERMEIR, D. 2010. Towards possibilistic fuzzy answer set programming. In *Proceedings of the 13th International Workshop on Non-monotonic Reasoning (NMR).*

BOBILLO, F. AND STRACCIA, U. 2007. A fuzzy description logic with product t-norm. In *In Proceedings of the 16th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007).* 652–657.

CABALAR, P., ODINTSOV, S., PEARCE, D., AND VALVERDE, A. 2006. Analysing and extending well-founded and partial stable semantics using partial equilibrium logic. In *Logic Programming.* Lecture Notes in Computer Science, vol. 4079. 346–360.

CAO, T. H. 2000. Annotated fuzzy logic programs. *Fuzzy Sets and Systems 113,* 2, 277 – 298.

CHAGROV, A. AND ZAKHARYASCHEV, M. 1997. *Modal Logic.* Oxford University Press.

CIABATTONI, A. AND LUCHI, D. 1997. Two connections between linear logic and Lukasiewicz logics. In *Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory.* Springer-Verlag, 128–139.

COLSON, B., MARCOTTE, P., AND SAVARD, G. 2005. Bilevel programming: A survey. *4OR: A Quarterly Journal of Operations Research 3*, 87–107.

DAMÁSIO, C. V., MEDINA, J., AND OJEDA-ACIEGO, M. 2004. Sorted multi-adjoint logic programs: termination results and applications. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04).* Springer-Verlag, 260–273.

DAMÁSIO, C. V. AND PEREIRA, L. M. 2000. Hybrid probabilistic logic programs as residuated logic programs. In *Proceedings of the 7th European Workshop on Logics in Artificial Intelligence (JELIA'00).* 57–72.

DAMÁSIO, C. V. AND PEREIRA, L. M. 2001. Antitonic logic programs. In *Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'01).* Springer-Verlag, 379–392.

DAMASIO, C. V. AND PEREIRA, L. M. 2003. An encompassing framework for paraconsistent logic programs. *Journal of Applied Logic 3*, 67–95.

DE VOS, M. AND VERMEIR, D. 2004. Extending answer sets for logic programming agents. *Annals of Mathematics and Artificial Intelligence 42*, 103–139.

DEKHTYAR, A. AND SUBRAHMANIAN, V. S. 1997. Hybrid probabilistic programs. In *Proceedings of the Fourteenth International Conference on Logic Programming (ICLP'97).* 391–405.

DUBOIS, D., OSTASIEWICZ, W., AND PRADE, H. 2000. Fuzzy sets: History and basic notions . In *Fundamentals of Fuzzy Sets* , D. Dubois and H. Prade, Eds. The Handbooks of Fuzzy Sets Series. Kluwer Academic Publishers, 21–124.

DUBOIS, D. AND PRADE, H. 1995. Fuzzy relation equations and causal reasoning. *Fuzzy Sets and Systems 75,* 2, 119 – 134.

DUBOIS, D. AND PRADE, H. 1996. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems 84,* 2, 169–185.

Dubois, D. and Prade, H. 2001. Possibility theory, probability theory and multiple-valued logics: A clarification. *Annals of Mathematics and Artificial Intelligence 32,* 1, 35–66.

Eiter, T. and Gottlob, G. 1993. Complexity results for disjunctive logic programming and application to nonmonotonic logics. In *Proceedings of the International Logic Programming Symposium*. 266–278.

Eiter, T. and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM 42,* 1, 3–42.

Eiter, T., Gottlob, G., and Leone, N. 1997. Abduction from logic programs: Semantics and complexity. *Theoretical computer science 189,* 1-2, 129–177.

Eiter, T., Gottlob, G., and Mannila, H. 1997. Disjunctive datalog. *ACM Transactions on Database Systems 22,* 3, 364–418.

Faber, W., Leone, N., and Pfeifer, G. 2004. Recursive aggregates in disjunctive logic programs: Semantics and complexity. In *Logics in Artificial Intelligence*. Lecture Notes in Computer Science, vol. 3229. 200–212.

Ferraris, P. 2005. Answer sets for propositional theories. In *LPNMR*, C. Baral, G. Greco, N. Leone, and G. Terracina, Eds. Lecture Notes in Computer Science, vol. 3662. Springer, 119–131.

Foo, N., Meyer, T., and Brewka, G. 2005. LPOD answer sets and Nash equilibria. In *Advances in Computer Science - ASIAN 2004*. Lecture Notes in Computer Science, vol. 3321. 3198–3199.

Gelfond, M. and Lifschitz, V. 1988. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference and Symposium on Logic Programming*. 1081–1086.

Girard, J.-Y. 1987. Linear logic. *Theoretical Computer Science 50*, 1–102.

Gottlob, G., Greco, G., and Scarcello, F. 2005. Pure Nash equilibria: hard and easy games. *Journal of Artificial Intelligence Research 24*, 357–406.

Hähnle, R. 1994. Many-valued logic and mixed integer programming. *Annals of Mathematics and Artificial Intelligence 12*, 231–264.

Hájek, P. 2001. *Metamathematics of Fuzzy Logic (Trends in Logic)*. Springer.

Hobbs, B., Metzler, C., and Pang, J.-S. 2000. Strategic gaming analysis for electric power systems: an mpec approach. *IEEE Transactions on Power Systems 15,* 2, 638 –645.

Janssen, J., Heymans, S., Vermeir, D., and De Cock, M. 2008. Compiling fuzzy answer set programs to fuzzy propositional theories. In *Proceedings of the 24th International Conference on Logic Programming (ICLP08)*. 362–376.

Janssen, J., Schockaert, S., Vermeir, D., and De Cock, M. 2009. General fuzzy answer set programs. In *Proceedings of the 8th International Workshop on Fuzzy Logic and Applications (WILF)*. 352–359.

Lifschitz, V., Pearce, D., and Valverde, A. 2001. Strongly equivalent logic programs. *ACM Transactions on Computational Logic 2,* 4, 526–541.

Lin, F. and Zhao, Y. 2004. ASSAT: computing answer sets of a logic program by sat solvers. *Artificial Intelligence 157,* 1-2, 115–137.

Lukasiewicz, T. 1998. Probabilistic logic programming. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*. 388–392.

Lukasiewicz, T. and Straccia, U. 2007. Tightly integrated fuzzy description logic programs under the answer set semantics for the semantic web. In *Proceedings of the First International Conference on Web Reasoning and Rule Systems (RR'07)*. 289–298.

Mamdani, E. and Assilian, S. 1975. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies 7,* 1, 1 – 13.

Marek, W. and Subrahmanian, V. S. 1992. The relationship between stable, supported, default and autoepistemic semantics for general logic programs. *Theoretical Computer Science 103,* 2, 365 – 386.

McNaughton, R. 1951. A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic 16*, 1–13.

Medina, J., Ojeda-Aciego, M., and Vojtáš, P. 2001. Multi-adjoint logic programming with continuous semantics. *Lecture Notes in Computer Science 2173*, 351–364.

Medina, J., Ojeda-Aciego, M., and Vojtáš, P. 2001. A multi-adjoint logic approach to abductive reasoning. In *Logic Programming*. Lecture Notes in Computer Science, vol. 2237. 269–283.

Mellouli, N. and Bouchon-Meunier, B. 2003. Abductive reasoning and measures of similitude in the presence of fuzzy rules. *Fuzzy Sets and Systems 137,* 1, 177 – 188.

Moore, J. and Bard, J. 1990. The mixed integer linear bilevel programming problem. *Operations Research 38,* 5, 911–921.

Mundici, D. 1987. Satisfiability in many-valued sentential logic is NP-complete. *Theoretical Computer Science 52*, 145–153.

Nicolas, P., Garcia, L., Stéphan, I., and Lefèvre, C. 2006. Possibilistic uncertainty handling for answer set programming. *Annals of Mathematics and Artificial Intelligence 47,* 1-2, 139–181.

Pang, J.-S. and Fukushima, M. 2005. Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games. *Computational Management Science 2*, 21–56.

Pearce, D. 1997. A new logical characterisation of stable models and answer sets. In *NMELP '96: Selected papers from the Non-Monotonic Extensions of Logic Programming*. 57–70.

Pearce, D. 2006. Equilibrium logic. *Annals of Mathematics and Artificial Intelligence 47*, 3–41.

Pearce, D. and Valverde, A. 2004. Towards a first order equilibrium logic for nonmonotonic reasoning. In *Logics in Artificial Intelligence*. Lecture Notes in Computer Science, vol. 3229. 147–160.

Pearce, D. and Valverde, A. 2008. Quantified equilibrium logic and foundations for answer set programs. In *Logic Programming*. Lecture Notes in Computer Science, vol. 5366. 546–560.

Perfilieva, I. 2004. Fuzzy function as an approximate solution to a system of fuzzy relation equations. *Fuzzy Sets and Systems 147,* 3, 363 – 383.

Perfilieva, I., Dubois, D., Prade, H., Esteva, F., Godo, L., and Hodáková, P. 2011. Interpolation of fuzzy data: Analytical approach and overview. *Fuzzy Sets and Systems doi:10.1016/j.fss.2010.08.005*.

Przymusinski, T. 1991. Stable semantics for disjunctive programs. *New Generation Computing 9*, 401–424.

Reggia, J., Nau, D., Wang, P., and Peng, H. 1985. A formal model of diagnostic inference. *Information Sciences 37*, 227–285.

Saharidis, G. and Ierapetritou, M. 2009. Resolution method for mixed integer bi-level linear problems based on decomposition technique. *Journal of Global Optimization 44*, 29–51.

Sakama, C. and Inoue, K. 1995. Paraconsistent stable semantics for extended disjunctive programs. *Journal of Logic and Computation 5*, 265–285.

Sanchez, E. 1976. Resolution of composite fuzzy relation equations. *Information and Control 30,* 1, 38 – 48.

Sandholm, T., Gilpin, A., and Conitzer, V. 2005. Mixed-integer programming methods for finding nash equilibria. In *Proceedings of the 20th national conference on Artificial intelligence*. 495–501.

Schockaert, S., Janssen, J., Vermeir, D., and Cock, M. 2009. Answer sets in a fuzzy equilibrium logic. In *Proceedings of the 3rd International Conference on Web Reasoning and Rule Systems*. 135–149.

Schrijver, A. 1998. *Theory of linear and integer programming*. John Wiley & Sons Inc.

Straccia, U. 2006. Annotated answer set programming. In *Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU'06)*.

Straccia, U., Ojeda-Aciego, M., and Damásio, C. V. 2009. On fixed-points of multivalued functions on complete lattices and their application to generalized logic programs. *SIAM Journal on Computing 38,* 5, 1881–1911.

Takeuti, G. and Titani, S. 1984. Intuitionistic fuzzy logic and intuitionistic fuzzy set theory. *The Journal of Symbolic Logic 49,* 3, 851–866.

Tanaka, K. and Sugeno, M. 1992. Stability analysis and design of fuzzy control systems. *Fuzzy Sets and Systems 45,* 2, 135 – 156.

Truszczynski, M. 2010. Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs. *Artificial Intelligence In Press.*

Van Gelder, A., Ross, K. A., and Schlipf, J. S. 1991. The well-founded semantics for general logic programs. *Journal of the ACM 38,* 3, 619–649.

Van Nieuwenborgh, D., De Cock, M., and Vermeir, D. 2007. Computing fuzzy answer sets using DLVHEX. In *Proceedings of the 23rd international conference on Logic programming.* 449–450.

Van Nieuwenborgh, D., De Cock, M., and Vermeir, D. 2007. An introduction to fuzzy answer set programming. *Annals of Mathematics and Artificial Intelligence 50,* 3-4, 363–388.

Wan, H. 2009. Belief logic programming. In *Logic Programming*, P. Hill and D. Warren, Eds. Lecture Notes in Computer Science, vol. 5649. 547–548.

Yamada, K. and Mukaidono, M. 1995. Fuzzy abduction based on lukasiewicz infinite-valued logic and its approximate solutions. In *International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium.* 343–350.

Zadeh, L. A. 1975. The concept of a linguistic variable and its application to approximate reasoning – I. *Information Sciences 8,* 3, 199 – 249.