Two Fuzzy Logic Programming Paradoxes

⇒

Continuum Hypothesis = “False”
Axiom of Choice = “False”
⇒

ZF\(\subseteq\) is Inconsistent

Rafee Ebrahim Kamouna
Email: rafee102000@yahoo.com
submitted to ACM Transactions on Computational Logic

Abstract

Two different paradoxes of the fuzzy logic programming system of [29] are presented. The first paradox is due to two distinct (contradictory) truth values for every ground atom of \(FLP\), one is syntactical, the other is semantical. The second paradox concerns the cardinality of the valid \(FLP\) formulas which is found to have contradictory values: both \(\aleph_0\) the cardinality of the natural numbers, and \(c\), the cardinality of the continuum. It follows that both \(CH\) & \(AC\) are false. Hence, \(ZF\subseteq\) is inconsistent.

1. Introduction

Fuzzy logic programming and possibilistic logic programming systems in the works of Alsinet and Godo et al. [1-18], Vojtas et al. [45-49] were developed with large number of soundness and completeness results with interesting properties. Variations as the multi-adjoint logic programming was developed by Medina et al. [33-36]. The first use of truth constants in the language syntax first appeared in Pavelka’s logic [41] as early as 1979. Before that, truth was expressed only in the language semantics as in Lukasiewicz and Kleene many-valued logics. Pavelka extended Lukasiewicz logic with rational truth constants. Novak [37-40], in his weighted inference systems developed a syntax of pairs: (formula, truth value). Expansions of other logics with truth constants in Esteva et al. 2000, and recently in Esteva et al. 2006 [24-27], and Savicky et al. 2006 [43]. In 2007, truth constants appeared in Esteva et al. [28]. The work of Bobillo and Straccia et al. [19,20,42,44] in fuzzy description logics employed truth constants as well. So, the
idea of having a truth constant in the language syntax is well-established. This paper presents two different paradoxes as properties of the fuzzy logic programming system presented in [29].

In an attempt to re-organize the XXth significant negative results of mathematics and computation, the present (poor) author introduced the “Syntactico-Semantical Bi-Polar Disorder Taxonomy”:

1. Self-referential SySBPD:
   (a) Russell’s paradox.
   (b) The Liar’s paradox.

2. Gödel Completeness/Incompleteness SySBPD; note the relationship between the proof of his celebrated incompleteness theorem and the Liar’s paradox.

3. Turing Decidability/Undecidability SySBPD.

4. Finiteness/Infiniteness SySBPD: results in finite model theory that succeed infinitely and fail finitely. Most importantly, Gödel’s completeness theorem which is:
   (a) Positive: Completeness/Incompleteness SySBPD.
   (b) Negative: Finiteness/Infiniteness SySBPD.

All these SySBPD’s are instances of the “Syntactico-Semantical Precedence/Principality Bi-Polar Disorder”.

1. Precedence: syntax definition precedes semantics:
   [Syntax < Semantics]_{Precedence}.

2. Principality: during computation the input takes various syntactic forms where semantics is principal over syntax in every computation step:
   [Semantics < Syntax]_{Principality}.

3. (1) & (2) \implies [Syntax] <> [Semantics], i.e. Bi-Polar Disorder.

The question: “Are the XXth the only SySBPD’s” led to the discovery of those two FLP SySBPD’s. Any more SySBPD’s? An open question.

2. The First SySBPD FLP Paradox
An atom in fuzzy logic programming in [29] looks like:

\[ p(t_1, t_2, \ldots, t_n, \mu) \]
Consider a program consisting only of a ground fact in FLP, e.g.:

\[ \text{AgeAbout21}(John, 0.9) \]

running this program with any ground goal results in answers either: “1” or “0” (semantical truth value), which is in contradiction with the truth constant: “0.9” - (syntactical truth value). This is because atoms in FLP are classical even when the weight is attached to them. Now this paradox is formalized rigorously in the following theorem.

First, the classical definition of an Herbrand interpretation and an Herbrand model are recalled. Second, it is shown that if truth constants are allowed in the language syntax in the sense of [29], then every Herbrand interpretation of any FLP language is a model iff it is not a model, except for the case when FLP collapses to classical logic, i.e. \( \mu = "0" \) or \( \mu = "1" \). This is the first “Syntactico-Semantical Bi-Polar Disorder FLP Paradox”.

**Definition 1:** Let \( L \) be a language over an alphabet \( \Sigma \) containing at least one constant symbol. The set \( U_L \) of all ground terms constructed from functions and constants in \( L \) is called the Herbrand universe of \( L \). The set \( B_L \) of all ground atomic formulas over \( L \) is called the Herbrand base of \( L \).

**Definition 2:** The Herbrand interpretation \( I_L \) for a language \( L \) is a structure \( I_L \equiv < I_c, I_f, I_p > \) whose domain of discourse is \( U_L \) where:

1. \( \forall c \in L : c \) is a constant:
   \[ I_c(c) = c \]

2. \( \forall f \in L : f \) is a function symbol of arity \( n \), and \( t_1, t_2, \ldots, t_n \) are terms:
   \[ I_f(f)(t_1, t_2, \ldots, t_n) = f(I(t_1), \ldots, I(t_n)) \]

3. \( \forall p \in L : p \) is a predicate of arity \( n \):
   \[ I_p(p) : B_L \to \{0, 1\} \]

**Definition 3:** The Herbrand interpretation \( I_L \) for a language \( L \) is a model iff \( I_L : B_L \to \{1\} \land B_L \not\to \{0\} \)

Let \( L \) be the classical logic program consisting of the single (ground) fact:

\[ p(c_1, c_2, \ldots, c_n) \leftarrow \]
and let \( c_n = \mu \in C \subseteq [0, 1] \) be a truth constant. If \( I_L \) is an Herbrand interpretation for \( L \), then \( I_L \) is a model iff it is not a model (unless \( \mu = "0" \) or "1", i.e. \( FLP \) collapses into classical logic). \( I_L \) interprets the predicate symbol \( p \) (classically) as a relation between the domains from which the n-tuple \((c_1, c_2, \ldots, c_n)\) is extracted. The last member of the tuple \( c_n \) is a real number in a countable \( C \subseteq [0, 1] \). When constant symbols are interpreted in classical semantics, it banishes an argument of a predicate to be the truth constant of the same predicate. \( FLP \) non-classical semantics enforces an argument of a predicate to be a truth constant of the same predicate. Semantics of formal languages are enforced in the same way as in natural languages. Since the string “main” over the Latin alphabet is interpreted differently in English and French (the word “main” in French means “hand”). Obviously,

\[
Oxford(main) \neq Larousse(main) \\
I_{L_{\text{Classical}}} (p) \neq I_{L_{\text{FLP}}} (p)
\]

Neither the English people may ask the French to follow Oxford dictionary, nor the French may ask the English to follow Larousse. Forbidding arguments of a predicate to be the truth constant of the same predicate is equally unacceptable. Moreover, if someone attempts to attack the \( P \) vs. \( NP \) question by examining the properties of any language, he may do so. The entire scientific community is pre-occupied with ANY set of strings (a language) that may separate the two classes. Usually, a set of strings in \( NP \) and not in \( P \), hence the question is settled. Let alone the self-referential nature of the question, i.e. \( P \) vs. \( NP \) is a question in \( NP \). So, if \( X \) is the decision problem \( X \equiv P \Rightarrow NP \), then \( X \in NP \). But classes are (forbidden) to be elements, so such an argument is a metamathematical/philosophical one (\( X \) is not a valid mathematical object). Just consider an analogy of the question: \( x? = y, x \in N, y \in R \). Obviously, this latter question is an ill-posed one. This situation encourages researchers to investigate any family of languages for possible potential important implications.

For the above considerations, the author is not deterred to enforce such semantics on the same syntax of classical logic, then examine the consequences. Forbidding such semantics won’t help because both classes contain infinite number of languages. Any method to forbid such semantics can obviously be eliminated with a counter-part to enforce whatever semantics to examine its implications to this long outstanding question. In other words, a counter-argument against \( FLP \) non-classical semantics should prove that such languages don’t exist at all. The fact that it leads to paradoxical and inconsistent computations never means that these computations are wrong or meaningless. \( FLP \) meta-interpreters have been implemented and work quite well meaningfully from a practical engineering point-of-view. The reason for this is that in a logic programming system, the user is
interested in answer substitutions rather than logical consequences as in automatic theorem proving. Cantor’s set theory has its famous paradoxes, one can never argue it is wrong, though initially it was controversial. The following theorem proves that languages written in FLP can have interpretations consisting of paradoxical structures.

**Theorem 1:** Let $L$ be the classical logic program consisting of the single (ground) fact:

$$p(c_1, c_2, \ldots, c_n) \leftarrow$$

and let $c_n = \mu \in [0, 1]$ be a truth constant (with the countability restriction). If $I_L$ is an Herbrand interpretation for $L$, then $I_L$ is a model iff it is not a model (unless $\mu = \text{“0”}$ or “1”, i.e. FLP collapses into classical logic).

**Proof:**

1. $I_L \equiv < I_c, I_f, I_p > \equiv < I_c, I_p >$.
2. $\Rightarrow I_c(c_1) = c_1$.
3. $\Rightarrow I_c(c_2) = c_2$.
4. \ldots
5. \ldots
6. \ldots
7. $\Rightarrow I_{c_{n-1}} = c_{n-1}$.
8. $\Rightarrow I_c(\mu) = \mu \in [0, 1]$.
9. $\Rightarrow I_p \in \{0, 1\}$.
10. $\Rightarrow I_L \equiv < I_c, I_p >$.
11. $\Rightarrow I_L \equiv < I_c \in [0, 1], I_p \in \{0, 1\} >$
12. $\Rightarrow \forall I_c \in [0, 1], I_L$ is a model iff it is not a model $\blacksquare$

**3. The Second SySBPD FLP Paradox**

For a program $L$ written in FLP, what is the cardinality of $\text{Valid}(L)$, the valid statements of $L$. Considering Gödel completeness theorem for predicate calculus one has:

$$|\text{Valid}(L)|_{L,\text{Classical}} = \aleph_0$$
where \( \aleph_0 \) is the cardinality of the natural numbers. But after lifting the countability restriction on \( \mu \in C \subseteq [0,1] \), one has:

\[
|\text{Valid}(L)|_{L:Fuzzy} = c
\]

where \( c \) is the cardinality of the continuum. So the second SySBP DFLP paradox can be expressed as:

\[
[|\text{Valid}(L)| = \aleph_0]_{\text{Classical}} \iff [|\text{Valid}(L)| = c]_{Fuzzy}
\]

The implications of those paradoxes would be considered by computer scientists/mathematicians in general and computational complexity theorists in particular. Let \( P, Q, R \) and \( S \) be as follows:

1. \( P = I(p) \in \{0,1\} \).
2. \( Q = I(p) \in [0,1] \).
3. \( R = |\text{Valid}(L)| = \aleph_0 \).
4. \( S = |\text{Valid}(L)| = c \), then:

5. Paradox I: (Theorem 1)

\[
I(p) \in \{0,1\} \iff I(p) \in [0,1] \equiv [P \iff Q]
\]

6. Paradox II: (Conjecture)

\[
|\text{Valid}(L)| = \aleph_0 \iff |\text{Valid}(L)| = c \equiv [R \iff S]
\]

**Theorem 2:** Paradox I \( \implies \) Paradox II.

**Proof:**

1. \( P \iff Q \).
2. \([I(p) \in \{0,1\} \iff |\text{Valid}(L)| = \aleph_0] \equiv [P \iff R] \).
3. \([I(p) \in [0,1] \iff |\text{Valid}(L)| = c] \equiv [Q \iff S] \).
4. \( P \iff S \), (1) & (3).
5. \( R \iff S \equiv \text{Pardox II}, (2) \& (4) \), Q.E.D. \( \blacksquare \)

Let \( L \) be a fuzzy program written in \( FLP \) whose fuzzy atom \( p(t_1, t_2, \ldots, t_n; \mu) \) is fuzzy if it is not fuzzy. The notation \( |\text{Valid}(L)| \) denotes the cardinality of the class of valid formulas of \( L \). Since \( FLP \) is classical, we are sure that \( |\text{Valid}(L)| = \aleph_0 \). But we know that \( FLP \) is (paradoxically) fuzzy, then \( |\text{Valid}(L)| = \aleph_0 \) only if \( \mu \in C \subseteq [0,1] \) where \( C \) is countable. If this condition is lifted (which is
the interesting case here), i.e. \([0, 1]\) is taken to be uncountable, we would have \(|Valid(L)| = c\). So, (paradoxically):

\[
|Valid(L)| = \aleph_0 \iff |Valid(L)| = c
\]

where \(c\) is the cardinality of the continuum. The following proof directly demonstrates that the Continuum Hypothesis is “False” and \(ZFC\) is inconsistent.

**Theorem 3**: Paradox I \(\implies\) Paradox II \(\implies\) \(ZFC\) is inconsistent.

1. \(\exists \alpha = |Valid(L)| : \aleph_0 < \alpha < c\).
2. \([\aleph_0 < \alpha < c \implies \neg CH] \implies ZFC \) is Inconsistent.

**Proof**:

The proof is as follows:

1. \(p\) is fuzzy \(\iff\) \(p\) is not fuzzy.
2. \(\implies [(\alpha = \aleph_0) \iff (\alpha = c)]\).
3. \([(\alpha = \aleph_0) \implies (\alpha = c)] \implies \alpha \neq \aleph_0\).
4. \([(\alpha < \aleph_0) \land (\neg \alpha < \aleph_0)] \implies \alpha \neq \aleph_0\).
5. \([(\alpha \neq \aleph_0) \land (\alpha \neq \aleph_0)] \implies \alpha > \aleph_0\).
6. \([(\alpha = c) \implies (\alpha = \aleph_0)] \implies \alpha \neq c\).
7. \([(\alpha > c) \land (\neg \alpha > c)] \implies \alpha < c\).
8. \(\implies \aleph_0 < \alpha < c\).
9. \(\implies \neg CH\).
10. \(ZFC\) is consistent \(\implies CH\) is formally independent (Gödel [133] & Cohen [48]).
11. \(\neg CH \implies CH\) is formally dependent.
12. \(\implies ZFC\) is Inconsistent!

Let \(\Sigma\) be the alphabet of the \(FLP\) system and \(\Sigma^*\) be the set of all finite strings over \(\Sigma\), then the set of all languages \(L \subseteq \Sigma^*\) is uncountable. For each \(L\), a paradoxical cardinal \(\alpha\) is associated with the set of all valid formulas of \(L\).

**Theorem 4**: Axiom of Choice = “False”.

1. \(\forall L \exists \alpha : \alpha\) is countable iff it is uncountable.
2. The set of all paradoxical cardinals CANNOT be well-ordered.
3. \(\implies\) Axiom of Choice = “False”. ■
The following proof emphasizes and clarifies the above results.

**Theorem 5:** Let \( p(t, \mu) \) be an atomic formula written in \( FLP \), there are four cases:

1. \( t: \text{countable}, \mu: \text{countable} \), then irrelevant to CH & AC, i.e. only Paradox I.

2. \( t: \text{countable}, \mu: \text{uncountable} \): uncountable pairs.

   (a) \( \forall \mu \exists p. \)
   
   (b) \( \implies |p| = c. \)

   (c) \( \forall \mu \in ]0, 1[ \ \exists \text{Valid}(p)_{\text{classical}}. \)

   (d) \( \forall \mu \in \{1\} \ \exists \text{Valid}(p)_{\text{classical}}. \)

   (e) \( \implies |\text{Valid}(p)_{\text{classical}}| \neq c. \)

   (f) \( \forall \mu \in ]0, 1[ \ \exists \text{Valid}(p)_{\text{fuzzy}}. \)

   (g) \( \implies |\text{Valid}(p)_{\text{fuzzy}}| = c. \)

   (h) \( (e) \& (g) \implies \text{the set of valid formulas is countable iff it is uncountable, Paradox II.} \)

3. \( t: \text{uncountable}, \mu: \text{countable} \): uncountable pairs

   (a) \( \forall \mu \exists p. \)

   (b) \( \implies |p| = c. \)

   (c) \( \forall \mu \in \{1\} \ \exists \text{Valid}(p)_{\text{classical}}. \)

   (d) \( \implies |\text{Valid}(p)_{\text{classical}}| \neq c. \)

   (e) \( \forall \mu \in ]0, 1[ \ \exists \text{Valid}(p)_{\text{fuzzy}}. \)

   (f) \( \implies |\text{Valid}(p)_{\text{fuzzy}}| \neq c. \)

   (g) \( (e) \& (g) \implies \text{No Paradox II.} \)

4. \( t: \text{uncountable}, \mu: \text{uncountable} \): uncountable pairs, i.e. same as (2) above.
References:

1. T. Alsinet, L. Godo: “Adding similarity-based reasoning capabilities to a Horn fragment of possibilistic logic with fuzzy constants”. Fuzzy Sets and Systems 144(1): 43-65 (2004) 2003.

2. T. Alsinet, C. Ansótegui, R. Bjar, C. Fernández, F. Many: “Automated monitoring of medical protocols: a secure and distributed architecture”. Artificial Intelligence in Medicine 27(3): 367-392 (2003).

3. T. Alsinet, R. Bjar, A. Cabiscol, C. Fernández, F. Many: “Minimal and Redundant SAT Encodings for the All-Interval-Series Problem. CCIA 2002: 139-144.

4. T. Alsinet, L. Godo, S. Sandri: “Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description”. Electr. Notes Theor. Comput. Sci. 66(5): (2002).

5. T. Alsinet, L. Godo: “Towards an automated deduction system for first-order possibilistic logic programming with fuzzy constants”. Int. J. Intell. Syst. 17(9): 887-924 (2002) 2001.

6. T. Alsinet, L. Godo: “A Proof Procedure for Possibilistic Logic Programming with Fuzzy Constants”. ECSQARU 2001: 760-771 2000.

7. T. Alsinet, R. Bjar, C. Fernandez, F. Many: “A Multi-agent system architecture for monitoring medical protocols”. Agents 2000: 499-505.

8. T. Alsinet, L. Godo: “A Complete Calculus for Possibilistic Logic Programming with Fuzzy Propositional Variables”. UAI 2000: 1-10 1999.

9. T. Alsinet, L. Godo, S. Sandri: “On the Semantics and Automated Deduction for PLFC, a Logic of Possibilistic Uncertainty and Fuzziness”. UAI 1999: 3-12.

10. T. Alsinet, F. Many, J. Planes: “Improved Exact Solvers for Weighted Max-SAT”. SAT 2005: 371-377 2004.

11. T. Alsinet, F. Many, J. Planes: “A Max-SAT Solver with Lazy Data Structures”. IBERAMIA 2004: 334-342.

12. T. Alsinet, C. I. Chesnevar, L. Godo, G. R. Simari: “A logic programming framework for possibilistic argumentation: Formalization and logical properties”. Fuzzy Sets and Systems 159(10): 1208-1228 (2008).

13. T. Alsinet, C. I. Chesnevar, L. Godo, G. R. Simari: “A logic programming framework for possibilistic argumentation: Formalization and logical properties”, Fuzzy Sets and Systems 159(10): 1208-1228 (2008).
14. T. Alsinet, L. Godo: “Adding similarity-based reasoning capabilities to a Horn fragment of possibilistic logic with fuzzy constants”. Fuzzy Sets and Systems 144(1): 43-65 (2004)

15. T. Alsinet, L. Godo, S. Sandri: Two formalisms of extended possibilistic logic programming with context-dependent fuzzy unification: a comparative description. Electr. Notes Theor. Comput. Sci. 66(5): (2002)

16. T. Alsinet, L. Godo: Towards an automated deduction system for first-order possibilistic logic programming with fuzzy constants. Int. J. Intell. Syst. 17(9): 887-924 (2002)

17. T. Alsinet, L. Godo: A Proof Procedure for Possibilistic Logic Programming with Fuzzy Constants. ECSQARU 2001: 760-771

18. T. Alsinet, L. Godo: A Complete Calculus for Possibilistic Logic Programming with Fuzzy Propositional Variables. UAI 2000: 1-10

19. F. Bobillo and U. Straccia: “On Qualified Cardinality Restrictions in Fuzzy Description Logics under Lukasiewicz semantics”. In Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-08), 2008.

20. F. Bobillo and U. Straccia: “fuzzyDL: An Expressive Fuzzy Description Logic Reasoner”. In Proceedings of the 2008 International Conference on Fuzzy Systems (FUZZ-08).

21. C. I. Chesnevar, Guillermo Ricardo Simari, Lluis Godo, Teresa Alsinet: “Argument-Based Expansion Operators in Possibilistic Defeasible Logic Programming: Characterization and Logical Properties”. ECSQARU 2005: 353-365.

22. C. I. Chesnevar, Guillermo Ricardo Simari, Lluis Godo, Teresa Alsinet: Expansion Operators for Modelling Agent Reasoning in Possibilistic Defeasible Logic Programming. EUMAS 2005: 474-475

23. C. I. Chesnevar, Guillermo Ricardo Simari, Teresa Alsinet, Lluis Godo: “A Logic Programming Framework for Possibilistic Argumentation with Vague Knowledge”. UAI 2004: 76-84.

24. F. Esteva, L. Godo: “Putting together Lukasiewicz and product logic”, Math.ware and Soft Computing 6:219:234, 1999.

25. F. Esteva, L. Godo, P. Hajek and M. Navara: “Residuated Fuzzy Logics with an Involution Negation”, Archive for Math. Log., 39: 103-124.
26. F. Esteva, L. Godo: “Monoidal t-norm Based Logic”, Fuzzy Sets and Systems, 124:271-288, 2001.

27. F. Esteva, L. Godo and C. Noguera: “On Rational Weak Nilpotent Minimum Logics”, J. Multiple-Valued Logic and Soft Computing, 2006.

28. F. Esteva, J. Gispert, L. Godo, C. Noguera: “Adding Truth-Constants to Logics of Continuous t-norms: Axiomatization and Completeness Results”, Fuzzy Sets and Systems, 158:597-618, 2007.

29. R. E. Kamouna: “Fuzzy Logic Programming”, Fuzzy Sets and Systems, 1998.

30. S. Krajci, R. Lencses, P. Vojtás: “A comparison of fuzzy and annotated logic programming”. Fuzzy Sets and Systems, 144 (2004) 173192

31. T. Lukasiewicz and U. Straccia: “Managing Uncertainty and Vagueness”, in Description Logics for the Semantic Web In Journal of Web Semantics.

32. S. Krajci, R. Lencses, P. Vojtás: “A comparison of fuzzy and annotated logic programming”. Fuzzy Sets and Systems, 144 (2004) 173192

33. J. Medina, M. Ojeda, P. Vojtás: “Multi-adjoint logic programming with continuous semantics”. In Proc. LPNMR’01. Th. Eiter et al eds. Lecture Notes in Artificial Intelligence 2173, Springer Verlag 2001, 351-364

34. J. Medina, M. Ojeda, P. Vojtás: “A procedural semantics for multi-adjoint logic programming. In Proc. EPIA’01, P. Brazdil and A. Jorge eds. Lecture Notes in Artificial Intelligence 2258, Springer Verlag 2001, 290-297

35. J. Medina, M. Ojeda, P. Vojás: “A completeness theorem for multi-adjoint logic programming”. In Proc. 10th IEEE Internat. Conf. Fuzzy Systems, IEEE 2001, 1031-1034,

36. J. Medina, M. Ojeda-Aciego, A. Valverde, P. Vojtás: “Towards Biresiduated Multi-adjoint Logic Programming”. R. Conejo et al Eds. Revised Selected Papers of CAEPIA 2003. Lecture Notes in Computer Science 3040 Springer 2004, 608-617,

37. V. Novak, I. Perfilieva and J. Mockor: “Mathematical principles of fuzzy logic”, Kluwer, Boston/Dordrecht, 1999.

38. V. Novak: “Weighted inference systems”, in J. C. Bezdek, D. Dubois and H. Prade (eds.): Fuzzy Sets in Approximate Reasoning and Information Systems. Handbooks of Fuzzy Sets Series, Vol. 3. Kluwer, Boston, 191-241, 1999.
39. V. Novak and I.Perfilieva (eds.): “Discovering the World with Fuzzy Logic; Studies in fuzziness and soft computing”, Heidelberg, New York: Physica-Verlag, Vol. 57, 302-304, 2000.

40. V. Novák, S. Gottwald, P. Hájek: Selected papers from the International Conference ”The Logic of Soft Computing IV” and Fourth workshop of the ERCIM working group on soft computing. Fuzzy Sets and Systems 158(6): 595-596 (2007)

41. J. Pavelka: “On Fuzzy Logic I-III. Zeit”, Math Logik Grund. Math. 25, 45-52, 119-134, 447-464, 1979.

42. A. Ragone, U. Straccia, T. Di Noia, E. Di Sciascio and F. M. Domini: “Fuzzy Description Logics for Bilateral Matchmaking in e-Marketplaces”. In Proceedings of the 16th Italian Symposium on Advanced Database Systems (SEBD-08), 2008.

43. P. Savicky, R. Cignoli, F. Esteva, L. Godo, C. Noguera: “On Product Logic with Truth-constants, Journal of Logic and Computation, Volume 16, Number 2, pp. 205-225(21), Oxford University, 2006.

44. U. Straccia: “Fuzzy Description Logic Programs”, in Uncertainty and Intelligent Information Systems, B. Bouchon-Meunier, R.R. Yager, C. Marsala, and M. Rifqi eds., 2008.

45. P. Vojtás: “Fuzzy logic programming”. Fuzzy Sets and Systems. 124,3 (2001) 361-370

46. P. Vojtás, T. Alsinet, Ll. Godo: “Different models of fuzzy logic programming with fuzzy unification (towards a revision of fuzzy databases)”. In Proc. IFSA’01 Vancouver, IEEE, 2001, 1541-1546,

47. P. Vojtás: “Tunable fuzzy logic programming for abduction under uncertainty”. In Proc. Workshop Many Valued Logic for Computer Science Applications. European Conference on Artificial Intelligence 98, University of Brighton, 1998, 7 pages

48. P. Vojtás, L. Paulak: “Soundness and completeness of non-classical extended SLD-resolution”, in Proc. ELP’96 Extended logic programming, Leipzig, ed. R. Dyckhoff et al., Lecture Notes in Comp. Sci. 1050 Springer Verlag, 1996, 289-301.

49. P. Vojtás, M. Vomlelová: “Transformation of deductive and inductive tasks between models of logic programming with imperfect information”, In Proc. IPMU 2004, B. Bouchon-Meunier et al. eds. Editrice Universita La Sapienza, Roma, 2004, 839-846