MODELLING AND REASONING WITH FUZZY LOGIC REDUNDANT KNOWLEDGE BASES

Hashim Habiballa
Eva Volná
Michal Janošek
Martin Kotyrba
Department of Computer Science
Faculty of Science, University of Ostrava
30. dubna 22, Ostrava, Czech republic
Email: hashim.habiballa@osu.cz

KEYWORDS

Fuzzy logic; Knowledge base; Resolution reasoning

ABSTRACT

Fuzzy Predicate Logic with evaluated syntax together with resolution principle is presented. The paper focuses mainly on modelling and treating redudancy in knowledge bases. It presents original resolution rule together with algorithm DCF - Detection of Consequent Formulas developed especially for fuzzy logic with evaluated syntax.

REASONING IN FUZZY PREDICATE LOGIC WITH EVALUATED SYNTAX

The problem of effective modelling and reasoning on knowledge bases rises especially when dealing with many-valued logics. We would like to recall previously devised notions and methods of fuzzy resolution principle and then show original efficient methods making such a reasoning tractable.

Fuzzy Predicate Logic with Evaluated Syntax (FPL) [Novák, V., Perfilieva, I., Močkoř, J., 1999] is a well-studied and wide-used logic capable of expressing vagueness. It has a lot of applications based on robust theoretical background. It also requires an efficient formal proof theory. However the most widely applied resolution principle [Dukić, N., Avdagić, Z., 2005] brings syntactically several obstacles mainly arising from normal form transformation. There are also recent attempts of similarity based resolution [Mondal, B., Raha, S., 2012], but our approach is based on classical proof theory of FPL. FPL is associating with even harder problems when trying to use the resolution principle. Solutions to these obstacles based on the non-clausal resolution [Bachmair, L., Ganzinger, H., 1997] were already proposed in [Habiballa, H., 2006].

In this article it would be presented a natural integration of these two formal logical systems into fully functioning inference system with effective proof search strategies. It leads to the refutational resolution theorem prover for FPL ($RRTP_{FPL}$). Another issue addressed in the paper concerns to the efficiency of presented inference strategies developed originally for the proving system. It is showed their perspectives in combination with standard proof-search strategies. The main problem for the fuzzy logic theorem proving lies in the large amount of possible proofs with different degrees and there is presented an algorithm (Detection of Consequent Formulas - DCF) solving this problem. The algorithm is based on detection of such redundant formulas (proofs) with different degrees.

The article presents the method which is the main point of the work on any automated prover. There is a lot of strategies which make proofs more efficient when we use refutational proving. We consider well-known strategies - orderings, filtration strategy, set of support etc. One of the most effective strategies is the elimination of consequent formulas. It means the check if a resolvent is not a logical consequence of a formula in set of axioms or a previous resolvent. If such a condition holds it is reasonable to not include the resolvent into the set of resolvents, because if the refutation can be deduced from it, then so it can be deduced from the original resolvent, which it implies of.

Resolution and Fuzzy Predicate Logic

The fuzzy predicate logic with evaluated syntax is a flexible and fully complete formalism, which will be used for the below presented extension [Novák, V., Perfilieva, I., Močkoř, J., 1999]. In order to use an efficient form of the resolution principle we have to extend the standard notion of a proof (provability value and degree) with the notion of refutational proof (refutation degree). Propositonal version of the fuzzy resolution principle has been already presented in [Habiballa, H., 2002]. We suppose that set of truth values is Łukasiewicz algebra. Therefore we assume standard notions of conjunction, disjunction etc. to be bound with Łukasiewicz operators.

We will assume Łukasewicz algebra to be

$$\mathcal{L}_{\text{Ł}} = \langle [0,1], \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$$

Proceedings 27th European Conference on Modelling and Simulation ©ECMS Webjørn Rekdalsbakken, Robin T. Bye, Houxiang Zhang (Editors)
ISBN: 978-0-9564944-6-7 / ISBN: 978-0-9564944-7-4 (CD)

where [0, 1] is the interval of reals between 0 and 1, which are the smallest and greatest elements respectively. Basic and additional operations are defined as follows:

$$a \otimes b = 0 \vee (a + b - 1) \quad a \rightarrow b = 1 \wedge (1 - a + b)$$

$$a \oplus b = 1 \wedge (a + b) \quad \neg a = 1 - a$$

The biresiduation operation $\leftrightarrow$ could be defined $a \leftrightarrow b =_{df} (a \rightarrow b) \wedge (b \rightarrow a)$, where $\wedge$ is infimum operation. The following properties of $\mathcal{L}_{\text{L}}$ will be used in the sequel:
$a \otimes 1 = a$, $a \otimes 0 = 0$, $a \oplus 1 = 1$, $a \oplus 0 = a$, $a \rightarrow 1 = 1$, $a \rightarrow 0 = \neg a$, $1 \rightarrow a = a$, $0 \rightarrow a = 1$
The syntax and semantics of fuzzy predicate logic is following:

- terms $t_1, ..., t_n$ are defined as in FOL
- predicates with $p_1, ..., p_m$ are syntactically equivalent to FOL ones. Instead of 0 we write $\bot$ and instead of 1 we write $\top$, connectives - **&** (Łukasiewicz conjunction), $\boldsymbol{\nabla}$ (Łukasiewicz disjunction), $\Rightarrow$ (implication), $\neg$ (negation), $\forall X$ (universal quantifier),$\exists X$ (existential quantifier) and furthermore by F_J we denote set of all formulas of fuzzy logic in language J
- FPL formulas have the following semantic interpretations (D is the universe): Interpretation of terms is equivalent to FOL, $\mathcal{D}(p_i(t_{i_1}, ..., t_{i_n})) = P_i(\mathcal{D}(t_{i_1}), ..., \mathcal{D}(t_{i_n}))$ where P_i is a fuzzy relation assigned to p_i, $\mathcal{D}(\mathfrak{a}) = a$ for $a \in [0,1]$, $\mathcal{D}(A \,\&\, B) = \mathcal{D}(A) \otimes \mathcal{D}(B)$, $\mathcal{D}(A \,\boldsymbol{\nabla}\, B) = \mathcal{D}(A) \oplus \mathcal{D}(B)$, $\mathcal{D}(A \Rightarrow B) = \mathcal{D}(A) \rightarrow \mathcal{D}(B)$, $\mathcal{D}(\neg A) = \neg \mathcal{D}(\mathbf{A})$,
$\mathcal{D}(\forall X(A)) = \bigwedge \mathcal{D}(A[x/d] | d \in D)$, $\mathcal{D}(\exists X(A)) = \bigvee \mathcal{D}(A[x/d] | d \in D)$

Graded fuzzy predicate calculus assigns grade to every axiom, in which the formula is valid. It will be written as a/A where A is a formula and a is a syntactic evaluation. We use several standard notions defined in [Novák, V., Perfilieva, I., Močkoř, J., 1999] namely: inference rule, formal fuzzy theory with set of logical and special axioms, evaluated formal proof.

Definition 1: Inference rule
An n-ary inference rule r in the graded logical system is a scheme

$$r : \frac{a_1/A_1, ..., \; a_n/A_n}{r^{evl}(a_1, ..., a_n)/r^{syn}(A_1, ..., A_n)} \qquad (1)$$

using which the evaluated formulas $a_1/A_1, ..., \; a_n/A_n$ are assigned the evaluated formula
$r^{evl}(a_1, ..., a_n)/r^{syn}(A_1, ..., A_n)$. The syntactic operation r^{syn} is a partial n-ary operation on F_J and the evaluation operation r^{evl} is an n-ary lower semicontinous operation on L (i.e. it preserves arbitrary suprema in all variables).

Definition 2: Formal fuzzy theory
A formal fuzzy theory T in the language J is a triple

$$T = \langle \text{LAx}, \text{SAx}, R \rangle$$

where LAx $\subsetneqq F_J$ is a fuzzy set of logical axioms, SAx $\subsetneqq F_J$ is a fuzzy set of special axioms, and R is a set of sound inference rules.

Definition 3: **Evaluated proof, refutational proof and refutation degree**
An evaluated formal proof of a formula A from the fuzzy set $X \subsetneqq F_J$ is a finite sequence of evaluated formulas $w := \; a_0/A_0, \; a_1/A_1, ..., \; a_n/A_n$ such that $A_n := A$ and for each $i \leq n$, either there exists an m-ary inference rule r such that
$a_i/A_i := \; r^{evl}(a_{i_1}, ..., a_{i_m})/r^{syn}(A_{i_1}, ..., A_{i_m})$,
$i_1, ..., i_m < n$ or $a_i/A_i := \; X(A_i)/A_i$.
We will denote the value of the evaluated proof by $Val(w) = a_n$.
An evaluated refutational formal proof of a formula A from X is w, where additionally $a_0/A_0 := \; 1/\neg A$ and $A_n := \bot$. $Val(w) = a_n$ is called refutation degree of A.

Definition 4: **Provability and truth**
Let T be a fuzzy theory and $A \in F_J$ a formula. We write $T \vdash_a A$ and say that the formula A is a theorem in the degree a, or provable in the degree a in the fuzzy theory T.

$$T \vdash_a A \text{ iff}$$
$$a = \bigvee \{Val(w) | \text{ w is a proof of A from LAx} \cup \text{SAx}\}$$

We write $T \models_a A$ and say that the formula A is true in the degree a in the fuzzy theory T.

$$\mathcal{D} \models T \text{ if } \forall A \in \text{LAx} :$$
$$\text{LAx}(A) \leq \mathcal{D}(A), \; A \in \text{SAx} : \; \text{SAx}(A) \leq \mathcal{D}(A)$$

$$T \models_a A \text{ iff } a = \bigwedge \{\mathcal{D}(A) \,|\, \mathcal{D} \models T\}$$

The fuzzy modus ponens rule could be formulated (we use special notion of most general unifier as defined in [Habiballa, H., 2012]):
Definition 5: Fuzzy modus ponens

$$r_{MP} : \frac{a/A, \; b/A \Rightarrow B}{a \otimes b/B} \qquad (2)$$

where from premise A holding in the degree a and premise $A \Rightarrow B$ holding in the degree b we infer B holding in the degree $a \otimes b$.

In classical logic r_{MP} could be viewed as a special case of the resolution. The fuzzy resolution rule presented below is also able to simulate fuzzy r_{MP}. From this fact the completeness of a system based on resolution can be deduced. It will only remain to prove the soundness. It is possible to introduce following notion of resolution w.r.t. the modus ponens.

Definition 6: **General resolution for fuzzy predicate logic** (GR_{FPL})

$$r_{GR} : \frac{a/F[G_1, , ..., G_k], \; b/F'[G'_1, ..., G'_n]}{a \otimes b/F\sigma[G/\bot] \,\boldsymbol{\nabla}\, F'\sigma[G/\top]} \qquad (3)$$

where $\sigma = MGU(A)$ is the most general unifier (MGU) of the set of the atoms
$A = \{G_1, \ldots, G_k, G'_1, \ldots, G'_n\}$, $G = G_1\sigma$. For every variable α in F or F', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow$

F is called positive and F' is called negative premise, G represents an occurrence of an atom. The expression $F\sigma[G/\bot] \vee F'\sigma[G/\top]$ is the resolvent of the premises on G.

Lemma 1: **Soundness of r_{GR}**
The inference rule r_{GR} for FPL based on $\mathcal{L}_{\text{Ł}}$ is sound i.e. for every truth valuation $\mathcal{D}$,

$$\mathcal{D}(r^{syn}(A_1, ..., A_n)) \geq r^{evl}(\mathcal{D}(A_1), ..., \mathcal{D}(A_n)) \qquad (4)$$

holds true.

Definition 7: **Refutational resolution theorem prover for FPL**
Refutational non-clausal resolution theorem prover for FPL ($RRTP_{FPL}$) is the inference system with the inference rule GR_{FPL} and sound simplification rules for $\bot$, $\top$ (standard equivalencies for logical constants). A refutational proof by definition 3 represents a proof of a formula G (goal) from the set of special axioms N.

Definition 8: **Simplification rules for $\nabla, \Rightarrow$**

$$r_{s\nabla} : \frac{a/\bot \nabla A}{a/A} \quad and \quad r_{s\Rightarrow} : \frac{a/\top \Rightarrow A}{a/A}$$

Lemma 2: **Provability and refutation degree for GR_{FPL}**
$T \vdash_a A$ iff $a =$
$\bigvee\{Val(w)|$ w is a refutational proof of A from LAx$\cup$ SAx$\}$

Theorem 1: **Completeness for fuzzy logic with r_{GR}, $r_{s\nabla}$, $r_{s\Rightarrow}$ instead of r_{MP}**
Formal fuzzy theory, where r_{MP} is replaced with r_{GR}, $r_{s\nabla}$, $r_{s\Rightarrow}$, is complete i.e. for every A from the set of formulas $T \vdash_a A$ iff $T \models_a A$.

MODELLING OF REDUNDANCY

The author also currently implements the non-clausal theorem prover into fuzzy logic as an extension of previous prover for FOL (GEneralized Resolution Deductive System - GERDS) [Habiballa, H., 2006]. Experiments concerning prospective inference strategies can be performed with this extension. The prover called Fuzzy Predicate Logic GEneralized Resolution Deductive System (Fig. 1) - FPLGERDS provides standard interface for input (knowledge base and goals) and output (proof sequence and results of fuzzy inference, statistics).

There are already several efficient strategies proposed by author (mainly Detection of Consequent Formulas (DCF) adopted for the usage also in FPL). With these strategies the proving engine can be implemented in real-life applications since the complexity of theorem proving in FPL is dimensionally harder than in FOL (the need to search for all possible proofs - we try to find the best refutation degree). The DCF idea is to forbid the addition of a resolvent which is a logical consequence of any previously added resolvent. For refutational theorem proving it is a sound and complete strategy and it is emipirically very effective. Completeness of such a strategy is also straight-forward in FOL:

$$(R_{old} \vdash R_{new}) \wedge (U, R_{new} \vdash \bot) \Rightarrow (U, R_{old} \vdash \bot)$$

Example: $R_{new} = p(a)$, $R_{old} = \forall x(p(x))$, $R_{old} \vdash R_{new}$.

DCF could be implemented by the same procedures like General Resolution (we may utilize self-resolution). Self-resolution has the same positive and negative premise and needs to resolve all possible combinations of an atom. It uses the following scheme:

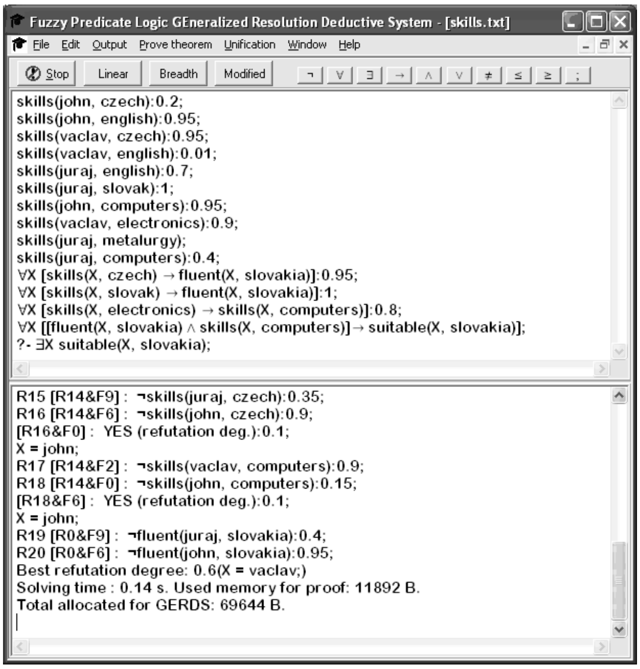


Fig. 1. Fuzzy Predicate Logic GEneralized Resolution Deductive System

$$R_{old} \vdash R_{new} \Leftrightarrow \neg(R_{old} \rightarrow R_{new}) \vdash \bot$$

Even the usage of this teachnique is a semidecidable problem, we can use time or step limitation of the algorithm and it will not affect the completeness of the $RRTP_{FOL}$.

Example: $R_{new} = p(a)$, $R_{old} = \forall x(p(x))$, $\neg(\forall x(p(x)) \rightarrow p(a))$
MGU: $Sbt(x) = a$, $Res = \neg(\bot \rightarrow \bot) \vee \neg(\top \rightarrow \top) \Rightarrow \bot$
We have proved that R_{new} is a logical consequence of R_{old}.

In FPL we have to enrich the DCF procedure by the limitation on the provability degree. if $U \vdash_a R_{old} \wedge U \vdash_b R_{new} \wedge b \leq a$ then we can apply DCF. DCF Trivial check performs a symbolic comparison of R_{old} and R_{new} we use the same provability degree condition. In other cases we have to add R_{new} into the set of resolvents and we can apply DCF Kill procedure. DCF Kill searches for every R_{old} being a logical consequence of R_{new} and if $U \vdash_a R_{old} \wedge U \vdash_b R_{new} \wedge b \geq a$ then Kill R_{old} (resolvent is removed).

We will now show some efficiency results concerning many-valued logic both for Fuzzy Predicate Logic. We have used the above mentioned application FPLGERDS and originally developed DCF strategy for FPL. It is clear that inference in $RRTP_{FPL}$ and $RRTP_{FDL}$ on general knowledge bases is a problem solved in exponential time. Nevertheless as we would like to demonstrate the need to search for every possible proof (in contrast to the two-valued logic) will not necessarily in particular cases lead to the inefficient theory. We have devised knowledge bases (KB) on the following typical problems related to the use of fuzzy logic.

We have performed experimental measurements concerning efficiency of the presented non-clausal resolution principle and also DCF technique. These measurements were done using the FPLGERDS application [Habiballa, H., 2006a]. Special testing knowledge bases were prepared and several types of inference were tested on a PC with standard Intel Pentium 4 processor as described below.

FUZZY PREDICATE LOGIC REDUNDANCY-BASED INEFFICIENT KNOWLEDGE BASES

As it was shown above in the theorem proving example the problem of proof search is quite different in FPL and FDL in comparison with the two-valued logic. We have to search for the best refutation degree using refutational theorem proving in order to make sensible conclusions from the inference process. It means we cannot accept the **first successful** proof, but we have to check **"all possible proofs"** or we have to be sure that every omitted proof is **worse** that some another one. The presented DCF and DCF Kill technique belong to the third sort of proof search strategies, i.e. they omit proofs that are really worse than some another (see the explication above). Proofs and formulas causing this could be called redundant proofs and redundant formulas. Fuzzy logic makes this redundancy dimensionally harder since we could produce not only equivalent formulas but also equivalent formulas of different evaluation degree.

Example 1: **Redundant knowledge base**
Consider the following knowledge base (fragment):

...,
$0.51/a \wedge b_1 \Rightarrow z$,
$0.61/a \wedge b_1 \wedge b_1 \Rightarrow z$,
$0.71/a \wedge b_1 \wedge b_1 \wedge b_1 \Rightarrow z$,
$0.81/a \wedge b_1 \wedge b_1 \wedge b_1 \wedge b_1 \Rightarrow z$,
$0.91/a \wedge b_1 \wedge b_1 \wedge b_1 \wedge b_1 \wedge b_1 \Rightarrow z$, $1/b_1$,
...,
$0.52/a \wedge b_2 \Rightarrow z$,
$0.62/a \wedge b_2 \wedge b_2 \Rightarrow z$,
$0.72/a \wedge b_2 \wedge b_2 \wedge b_2 \Rightarrow z$,
$0.82/a \wedge b_2 \wedge b_2 \wedge b_2 \wedge b_2 \Rightarrow z$,
$0.92/a \wedge b_2 \wedge b_2 \wedge b_2 \wedge b_2 \wedge b_2 \Rightarrow z$, $1/b_2$,
...,
Goal: $? - a \Rightarrow z$

Searching for the best proof of a goal will produce a lot of logically equivalent formulas with different degrees. These resolvents make the inference process inefficient and one of the essential demands to the presented refutational theorem prover is a reasonable inference strategy with acceptable time complexity.

We have compared efficiency of the standard **breadth-first search**, **linear search** and **modified linear search** (starting from every formula in knowledge base) and also combinations with DCF and DCF-kill technique [Habiballa, H., 2006a]. We have prepared knowledge bases of the size 120, 240, 360, 480 and 600 formulas. It has been compared the time and space efficiency on the criterion of 2 redundancy levels. This level represents the number of redundant formulas to which the formula is equivalent (including the original formula). For example the level 5 means the knowledge base contain 5 equivalent redundant formulas for every formula (including the formula itself). The basic possible state space search techniques and DCF heuristics and their combinations are presented in the following tables.

Search		**Description**
Breadth	B	Level order generation
Linear	L	Resolvent $\Rightarrow$ premise
Mod-Linear	M	Resolvent $\Rightarrow$ premise, goal+SAx

Table 1. Proof search algorithms

We use standard state space search algorithms in the FPLGERDS application - Breadth-first and Linear search. Breadth-first method searches for every possible resolvent from the formulas of the level 0 (goal and special axioms). These resolvents form formulas of the level 1 and we try to combine them with all formulas of the same and lower level and continue by the same procedure until no other non-redundant resolvent could be found. Linear search performs depth-first search procedure, where every produced resolvent is used as one of the premises in succeeding step of inference. The first produced resolvents arises from the goal formula. Modified linear search method posses the same procedure as linear one, but it starts from goal and also from all the special axioms.

DCF type		**Description**
Trivial	T	Exact symbolic comparison
DCF	DC	New consequent resol. dispose
DCF Kill	DK	DCF + all old consequent res.

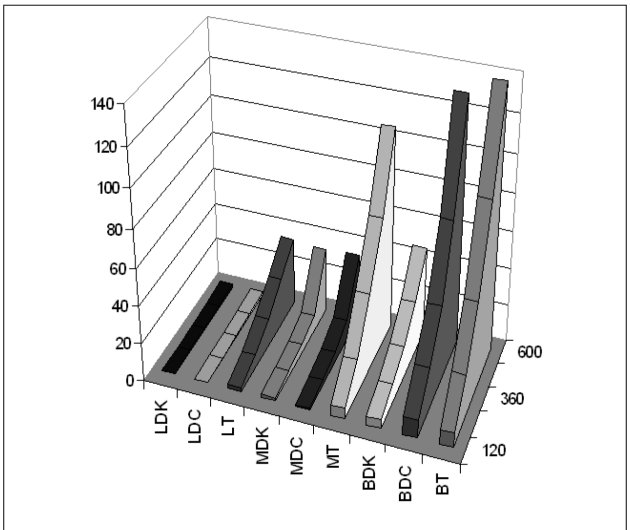
Table 2. DCF heuristics

DCF methods for reduction of resolvent space are basically three. The simplest is trivial DCF method, which detects redundant resolvent only by its exact symbolic comparison, i.e. formulas are equivalent only if the are syntactically the same. Even it is a very rough method, it is computationally very simple and forms necessary essential restriction for possibly infinite inference process. The next method of DCF technique enables do detect the equivalency of a formula (potential new resolvent) by the means described above. DCF Kill technique additionally tries to remove every redundant resolvent from the set of resolvents. The important aspect of the theorem DCF lies in its simple implementation into an automated theorem prover based on general resolution. The prover handles formulas in the form of syntactical tree. It is programmed a procedure performing general resolution with two formulas on an atom. This procedure is also used for the implementation of the theorem. A "virtual tree" is created from candidate and former resolvent (axiom) connected by negated implication. Then it remains to perform self-resolution on such formula until a logical value is obtained. Let us compare the efficiency of standard strategies and the above-defined one.

Search	**DCF**	Code	**Description**
Breadth	Trivial	BT	Complete
Breadth	DCF	BDC	Complete
Breadth	DCF Kill	BDK	Complete
Mod. Linear	Trivial	MT	Incomplete (+)
Mod. Linear	DCF	MDC	Incomplete (+)
Mod. Linear	DCF Kill	MDK	Incomplete (+)
Linear	Trivial	LT	Incomplete
Linear	DCF	LDC	Incomplete
Linear	DCF Kill	LDK	Incomplete

Table 3. Inference strategies

We have built-up 9 combinations of inference strategies from the mentioned proof search and DCF heuristics. They have different computational strength, i.e. their completeness is different for various classes of formulas. Fully complete (as described above) for general formulas of FPL and FDL are only breadth-first search combinations. Linear search strategies are not complete even for two-valued logic and horn clauses. Modified linear search has generally bad completeness results when an infinite loop is present in proofs, but for guarded knowledge bases it can assure completeness preserving better space efficiency than breadth-first search.

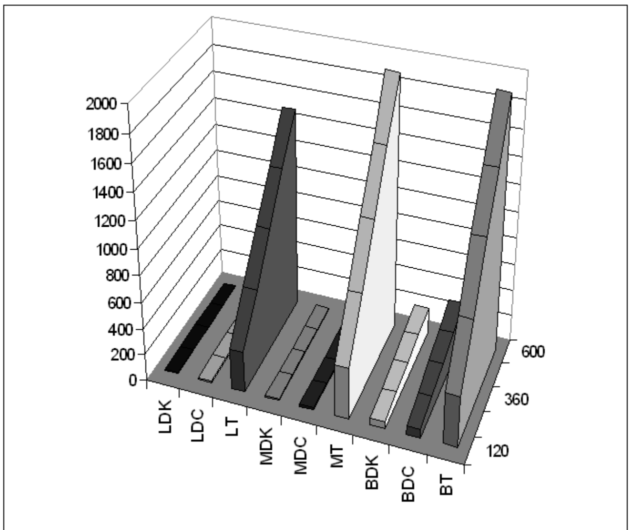


	120	240	360	480	600
BT	7,44	25,09	50,55	90,91	138,67
BDC	9,52	23,73	43,76	76,63	129,94
BDK	4,68	10,1	17,64	29,54	47,31
MT	5,86	18,85	41,46	70,01	106,68
MDC	0,93	3,09	5,14	10,54	35,14
MDK	1,45	3,2	5,03	8,84	34,32
LT	2,17	7,24	14,76	24,98	35,77
LDC	0,26	0,52	0,57	0,8	1,96
LDK	0,28	0,5	0,57	0,78	1,99

Fig. 2. Time complexity for redundancy level 5 (seconds)

We tested presented inference strategies on sample knowledge bases with redundancy level 5 with 20, 40, 60, 80 and 100 groups of mutually redundant formulas (total number of formulas in knowledge base is 120, 240, 360, 480 and 600). At first we have tested their time efficiency for inference process. As it could be observed from figure 2, the best results have **LDK and LDC** strategies. For simple guarded knowledge bases (not leading to an infinite loop in proof search and where the goal itself assures the best refutation degree) these two methods are **very efficient**. DCF strategies significantly reduces the proof search even in comparison with LT strategy (standard), therefore the usage of any non-trivial DCF heuristics is significant. Next important result concludes from the comparison of BDK and MDK, MDC strategies. We can conclude that MDK and MDC strategies are relatively comparable to BDK and moreover BDK preserves completeness for general knowledge bases.

Space complexity is even more significantly affected by the DCF heuristics. There is an interesting comparison of trivial and non-trivial DCF heuristics in figure 3. Even BDK strategy brings significant reduction of resolvents amount, while LDK, LDC, MDK, MDC strategies have minimal necessary amount of kept resolvents during inference process.



	120	240	360	480	600
BT	385	765	1185	1525	1905
BDC	65	125	185	245	306
BDK	46	86	126	166	205
MT	385	765	1145	1525	1905
MDC	14	14	14	14	12
MDK	14	14	14	14	12
LT	304	604	904	1204	1504
LDC	12	12	12	12	10
LDK	12	12	12	12	10

Fig. 3. Space complexity for redundancy level 5 (resolvents)

Performed experiments shows the significance of originally developed DCF strategies in combination with standard breadth-first search (important for general knowledge bases - **BDK**). We also outlined high efficiency for linear search based strategies (mainly **LDK**). Even this strategy is not fully complete and could be used only for guarded fragment of FDL, this problem is already known in classical (two-valued) logic programming and automated theorem proving. We also use these highly efficient linear search strategies, even they are not complete.

CONCLUSIONS

The *Non-clausal Refutational Resolution Theorem Prover* forms a powerful inference system for automated theorem proving in fuzzy predicate logic. The main advantage in contrast with other inference systems lies in the possibility to utilize various inference strategies for effective reasoning. Therefore it is essential for practically successful theorem proving.

The Detection of Consequent Formulas algorithms family brings significant improvements in time and space efficiency for the best proof search. It has been shown results indicating specific behavior of some combinations of the DCF and standard proof search (breadth-first and linear search). DCF strategies (BDC, BDK) have interesting results even for fully general fuzzy predicate logic with evaluated syntax, where the strategy makes the inference process practically manageable (in contrast to unrestricted blind proof-search). However it seems to be more promising for practical applications to utilize incomplete strategies with high time efficiency like LDK (even for large knowledge bases it has very short solving times). It conforms to another successful practical applications in two-valued logic like logic programming or deductive databases where there are also used efficient incomplete strategies for fragments of fully general logics.

It has been briefly presented some efficiency results for the presented automated theorem prover and inference strategies. They show the significant reduction of time and space complexity for the DCF technique. Experimental application FPLGERDS can be obtained from URL:
http://irafm.osu.cz/en/c104_fplgerds/
The package contains current version of the application, source codes, examples and documentation.

ACKNOWLEDGEMENT

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and also by University of Ostrava grant SGS/PŘF/2013.

REFERENCES

[Bachmair, L., Ganzinger, H., 1997] Bachmair, L., Ganzinger, H. (1997). A theory of resolution. Technical report: Max-Planck-Institut, 1997.

[Bachmair, L., Ganzinger, H., 2001] Bachmair, L., Ganzinger, H. (2001). Resolution theorem proving. In Handbook of Automated Reasoning, MIT Press, 2001.

[Dukić, N., Avdagić, Z., 2005] Dukić, N., Avdagić, Z. (2005). Fuzzy Functional Dependency and the Resolution Principle. In Informatica, Vilnius: Lith. Acad. Sci. (IOSPRESS), 2005, Vol.16, No. 1, pp. 45 - 60, 2005.

[Habiballa, H., 2002] Habiballa, H., Novák, V. (2002). Fuzzy General Resolution. In Proc. of Intl. Conf. Aplimat 2002. Bratislava, Slovak Technical University, 2002. pp. 199-206.

[Habiballa, H., 2006] Habiballa, H. (2006). Resolution Based Reasoning in Description Logic. In Proc. of Intl. Conf. ZNALOSTI 2006, Univ. of Hradec Kralove, 2006.

[Habiballa, H., 2006a] Habiballa, H.(2006a). Fuzzy Predicate Logic Generalized Resolution Deductive System. Technical Report, Institute for Research and Application of Fuzzy Modeling, University of Ostrava, 2006.

[Habiballa, H., 2012] Habiballa, H.(2012).Resolution principle and fuzzy logic. In Fuzzy Logic algorithms, techniques and implementations. InTech:Rijeka, 2012. ISBN 978-953-51-0393-6.

[Hájek, P., 2000] Hájek, P. (2000). Metamathematics of fuzzy logic. Kluwer Academic Publishers - Dordrecht, 2000.

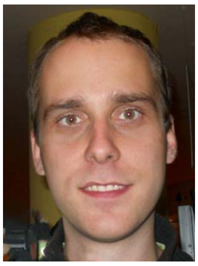
[Hájek, P., 2005] Hájek, P. (2005). Making fuzzy description logic more general. Fuzzy Sets and Systems 154(2005),pp. 1-15.

[Mondal, B., Raha, S., 2012] Mondal, B., Raha, S., Approximate reasoning in fuzzy resolution. Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American, pp.1-6, 6-8 Aug. 2012. ISBN 978-1-4673-2336-9.

[Novák, V., Perfilieva, I., Močkoř, J., 1999] Novák, V., Perfilieva, I., Močkoř, J. (1999). Mathematical principles of fuzzy logic. Kluwer, 1999.

HASHIM HABIBALLA
(*1976) Associate Professor at University of Ostrava, Dept. of Informatics and Computers and researcher of Centre of Excelence IT4Innovations - Institute for Research and Application of Fuzzy Modeling. Holds MSc.(1999) and Ph.D.(2007) in Information Systems, holds also PhD. and EdD. in CS Education (2004). He focuses to artificial intelligence, logic, formal languages and is author of 100 publications.

EVA VOLNÁ
Associate professor at the Department of Computer Science at University of Ostrava, Czech Republic. Her interests include artificial intelligence, artificial neural networks, evolutionary algorithms, and cognitive science. She is an author of more than 50 papers in technical journals and proceedings of conferences.



MICHAL JANOŠEK
He is a doctor at the Department of Computer Science at University of Ostrava, Czech Republic. His interests include artificial intelligence, multi-agent systems, modelingand simulations. He is an author of more than 20 papers in proceedings of conferences.

MARTIN KOTYRBA
He is a doctor at the Department of Computer Science at University of Ostrava, Czech Republic. His interests include artificial intelligence, formal logic, soft computing methods and fractals. He is an author of more than 30 papers in proceedings of conferences.