FUZZY DEDUCTIVE DATABASES BASED ON RESOLUTION PRINCIPLE

Hashim Habiballa, Alena Lukasová, Zdenka Telnarová

Institute for Research and Applications of Fuzzy Modeling
and Department of Computer Science,
University of Ostrava, 30. dubna 22, Ostrava, Czech Republic
e-mail: Hashim.Habiballa@osu.cz

Abstract. The article presents the possibilities of usage of the resolution principle for Fuzzy Predicate Logic with Evaluated Syntax as theoretical and inference background for fuzzy deductive databases. We start on explaining the problem of the classical deductive databases, further describing possibilities of fuzzyfication of deductive database on the mentioned fuzzy logic. The key part shows how to utilize resolution principle as an inference engine of the fuzzy deductive database together with some efficient resolution strategies devised originally for the refutational resolution theorem proving system for FPL.

Key words: Fuzzy inference systems, Non-classical logics, Fuzzy deductive databases.

1. Introduction

Hardly any other area of informatics has undergone such an extensive development of theoretical conceptions as database systems. The first commercial software applications of database systems were available to users already at the end of the 60s (IMS and IBM firms). It might seem that enough time has passed and that the entire theoretical development has already reached its accomplishment. Yet, after more than forty years that is not true, and the current development still brings a number of new views and conceptions the only aim of which is more exactly to seize a reality to be modelled.

The recent development in the area of database systems has progressed from the oldest hierarchical conception characterised by a quite rigid structure of hierarchic relations among data elements of entity nature, through network conception that was actually the improvement, generalisation of hierarchical conception, to relation conception.

After the commercial defeat of network paradigm by which the theoretical dispute between Bachmann and Codd had climaxed, the relation conception became prevalent worldwide in the field of data modelling. Its greatest preference was of course the simplicity of work with relations. On the other hand, this simplicity was accompanied by a number of shortcomings as for example the absence of time dimension, refusal of derived data modelling, problem of weak entities, decomposition of relations into elementary structures due to the introduction of normalisation procedures, isolation from the functional side of modelling, etc. These facts led to further theoretical development accentuating the above mentioned particular or group shortcomings. Without any claim to the completeness, the following can be stated:

- Time databases
- Multidimensional databases
- Object relational databases
- Object databases
- Derived data in databases

This chapter will at least briefly concentrate on the problem of modelling of derived data in databases.

The pure relational theory does not deal with derived data. There is a notion that derived data will always come into existence only on the basis of selection of atomic data from the database and that they are presented by users as a reflection of the immediate state of the database. In practice, however, the application of such approach brings a considerable set of problems. At frequent selections of the same nature from extensive databases, the overall efficiency of processing is decreasing. An ordinary requirement from the area of business is a requirement for the derived data to be filed in some way and analysed later (time series, overall surveys and the like). A natural reaction of almost all commercial database systems based on the relational conception is in this direction a concept of views and snapshots. Views provide a dynamic structure and snapshots a static structure that can be used for capturing derived data in the dynamic (actual) or static (to the specified date) shape. Views explicitly or physically do not occur in the database. A data model may contain relations and attributes whose instances or values are derivable from other instances and thus they are obtainable from the data stored in the database. Derived instances can be represented as additional information. Rules for derivation of additional information are being created in the phase of designing the information system.

Analogically object or compromise object-relational conception by comprising methods to object-classes easily solves the creation of derived data defined by the method. Another, probably more sophisticated, but also more specifically orientated conception modelling the derived data is a conception of deductive databases. This conception will be later dealt with in greater detail.

Deductive databases are based on the support of theory of proving and they are able to deduct additional facts from the database. Specified deductive axioms and deductive rules have been built in them. Deductive axioms together with integrity constraints are usually indicated as intensional databases. Therefore, the deductive database consists of two elements, extensional database and intensional database. Extensional database then corresponds to relations in the relational data model or classes in the object data model. Derived conceptions need not be stored in the database and they are usually temporal.

Deductive rules.

Conventional database systems do not work with a term deductive rules and its partial function is exercised by queries. In deductive database systems the rules are the basic concepts used in order to obtain information from the database. In addition to it, the rules are used as means for maintaining database consistency. From a formal standpoint, the rules are declarative expressions. By their evaluation (or an appropriate interpretation) it is possible to obtain additional information from the database.

Introduction to deductive rules

Deductive rule is an expression in the form:

$$\boxed{\text{Conclusion} \longleftarrow \text{Premise}}$$

Where conclusion is an atomic formula (head of the rule) and premise is a formula (body of the rule). Theory of deductive rules is based on deductive axioms; deductive axiom is a rule by which we are able to deduce additional facts from the given facts. Deductive axioms together with integrity constraints form a so called deductive database.

Deductive rules can be used to express the subsumption relationship between concepts, to define intensional predicates, to express different types of knowledge, to represent causal relationship between causes and effects [1].

Data model based on deductive rules we used to call „Datalog“ model, and it is an extended relational model. Predicate symbols in Datalog denote relations. Deductive relational approach can be transferred into deductive object approach by using of a logical framework for objects. Example of deductive - object model can be Telos or Chimera model.

2. Fuzzy Predicate Logic with Evaluated Syntax and relational deductive databases

The fuzzy predicate logic with evaluated syntax is a flexible and fully complete formalism, which will be used for below presented theoretical system of fuzzy deductive databases [9]. We will suppose that set of truth values is Lukasiewicz algebra. Therefore we will assume standard notions of conjunction, disjunction etc. to be bound with Lukasiewicz operators.

We will assume Lukasewicz algebra to be

$$L_L = \langle [\,0,1\,], \wedge, \vee, \otimes, \rightarrow, 0, 1\,\rangle$$

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:

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

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 $\perp$ and instead of 1 we write T, connectives - &(Lukasiewicz conjunction), ∇(Lukasiewicz disjunction), $\Rightarrow$(implication), $\neg$(negation), $\forall$X(universal quantifier), $\exists$X(existential quantifier). FPL formulas have the following semantic interpretations (D is the universe): Interpretation of terms is equivalent to FOL, $D(p_i(t_{i1},...,t_{in})) = P_i(D(t_{i1}),...,D(t_{in}))$ where P_i is a fuzzy relation assigned to p_i, $D(A \,\&\, B) = D(A) \otimes D(B)$, $D(A \nabla B) = D(A) \oplus D(B)$, $D(A \Rightarrow B) = D(A) \rightarrow D(B)$, $D(\neg A) = \neg D(A)$, $D(\forall X\,(A)) = \wedge\, D(A[x/d] \| d \in D)$, $D(\exists X\,(A)) = \vee\, D(A[x/d] \| d \in D)$.

For every subformula defined above Sub, Sup, Pol, Lev, Qnt, Sbt, Sig and other derived properties defined above hold (where the classical FOL connective is presented the Lukasiewicz one has the same mapping value).

Graded fuzzy predicate calculus assigns grade to every axiom, in which the formula is valid. It will be written as

$$a \qquad / \qquad A$$

where A is a formula and a is a syntactic evaluation. We will need to introduce several notions from fuzzy logic, in order to give the reader more exact definition of fuzzy theory. Since these definitions rather address inference process we will state them in the next section concerning resolution principle as a reasoning principle for fuzzy deductive databases according to the presented theory.

The main advantages of using Fuzzy Predicate Logic with Evaluated Syntax are the following:

- standard definition of this logic does not work with fuzzy terms (i.e. interpretation structures of individual constants are members of classical sets and interpretation structures of functors are classical crisp functions); this issue lead to standard unification algorithm within inference process; this is not bringing any loss of generality since fuzzy functions may be simulated by fuzzy relations (predicates).

- this logic explicitly differentiates between syntactical and semantic truth values; it works with syntactic evaluation.

Basic principle of integration relation data model and fuzzy predicate logic lies in the encoding of standard notion of relation into the syntactical framework of predicates and its semantic counterpart – fuzzy relation [11]. Then we can refer to standard logic programming notion like facts statement and rules statements.

Fuzzy facts statements

Any member of a fuzzy relation P_i is expressed as an atomic formula

$p_i(t_{i1},...,t_{in})$ / a, where p_i is a predicate name and t_{i1}, ...,t_{in} are constant terms and a is syntactic evaluation of the fuzzy formula.

Example: For a relation expressing employee's language skills – skill with attributes – person and language we can express the fuzzy facts:

Skills(john, czech) / 0.2 (john is speaking czech a little),
Skills(john, english) / 0.95 (john is speaking english excellent),
Skills(vaclav, czech) / 0.95 (vaclav is speaking czech excellent),
Skills(vaclav, english) / 0.01 (vaclav is not speaking english - almost),
Skills(juraj, english) / 0.7, Skills(juraj, slovak) / 1 and other skills
Skills(john, computers) / 0.95, Skills(vaclav, electronics) / 0.9, Skills(juraj, metalurgy) / 1, Skills(juraj, computers) / 0.4

Fuzzy rules statements

Fuzzy rule is a formula of Fuzzy predicate logic with a syntactic evaluation.

Example: We express a rule describing an ability to communicate in Slovakia according to speaking czech (any czech speaking person is able to communicate fluently in Slovakia very well).

$\forall X$ (skills(X, czech) $\Rightarrow$ fluent_communication(X, slovakia)) / 0.95
$\forall X$ (skills(X, slovak) $\Rightarrow$ fluent_communication(X, slovakia)) / 1

and we can also express a rule that anyone with good skills in electronics has also relatively good skills in computers

$\forall X$ (skills(X, electronics) $\Rightarrow$ skills(X, computers)) / 0.8

Then we can formulate a rule describing suitability of a person to be a representative in Slovakia.

$\forall X$((skills(X,electronics)&fluent_communication(X,slovakia))$\Rightarrow$suitable(X, slovakia))

Fuzzy queries statements

Fuzzy query is a formula of fuzzy predicate logic which is transformed into goal.

Example: Following our previous examples we could ask (formulate a query) about a suitable person being a representative.

$\exists$X suitable(X, slovakia).

2.1. Resolution principle in Fuzzy Logic and deduction in databases

For a successful reasoning in fuzzy deductive databases we have utilize effective inference theory. The theory of non-clausal resolution seems to be very useful for fuzzy logic programming and that's why also it should be good candidate for deductive databases. We recall some notions already published concerning resolution principle for fuzzy logic [4].

Evaluated proof, refutational proof and refutation degree

An evaluated formal proof of a formula A from the fuzzy set $X \subset \sim F_J$ *is a finite sequence of evaluated formulas*

$$w := \ a_0 / A_0 , \ \ a_1 / A_1 ,... , \ \ a_n / A_n \qquad (1)$$

such that $A_n := A$ *and for each* $i \le n$*, either there exists an m-ary inference rule r such that*

$$a_i / A_i \ := \ r^{evl}(a_{i1}, ..., a_{im}) / r^{syn}(A_{i1}, ..., A_{im}) , \quad i_1, ..., i_m < n$$

or

$$a_i / A_i \ := \ X(A_i) / A_i$$

We will denote the value of the evaluated proof by $Val(w) = a_n$*, which is the value of the last member in (1).*

An evaluated refutational formal proof of a formula A form X is w, where additionally

$$a_i / A_i \ := \ 1 / \neg A$$

and $A_n := \perp$*.* $Val(w) = a_n$ *is called refutation degree of A.*

General resolution for fuzzy predicate logic (GR_{FPL})

$$r_{GR}: \frac{a / F[G_1,,...,G_k], \ \ b / F'[G'_1,...,G'_n]}{a \otimes b / F\sigma[G / \perp] \nabla F'\sigma[G / T]}$$

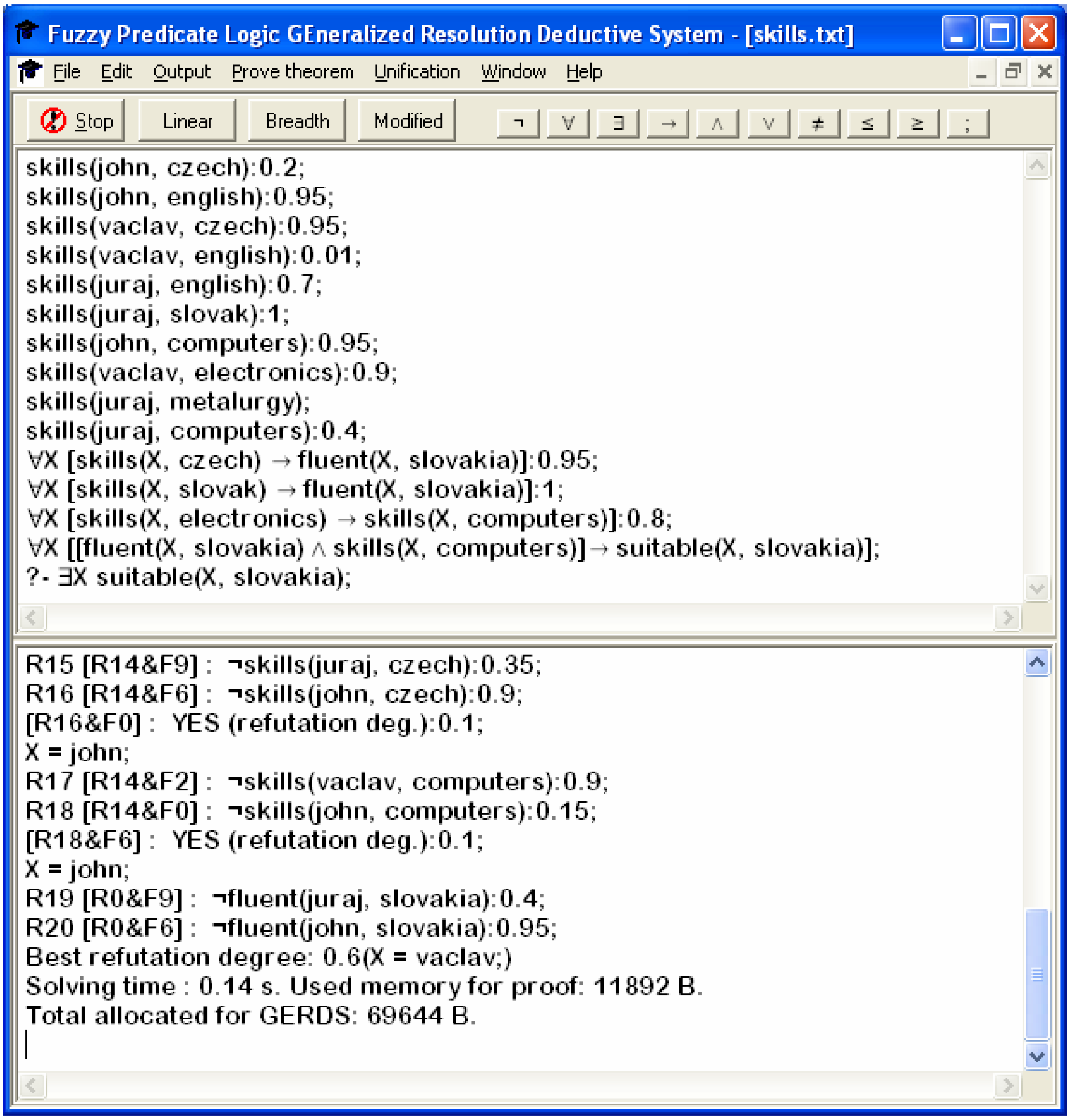
where $\sigma = MGU(A)$ *is the most general unifier (MGU) of the set of the atoms* $A = \{ G_1, \dots, G_k ,G'_1, \dots,G'_n \} , G = G_1\sigma$*. For every variable* α *in F or F', (*$Sbt(\gamma) = \alpha) \cap \sigma = \varnothing \Rightarrow Sig(\alpha) = 1$ *in F or F' iff* $Sig(\alpha) = 1$ *in* $F\sigma[G / \perp] \vee F'\sigma[G / T]$*. F is called positive and F' is called negative premise, G represents an occurrence of an atom. The expression* $F\sigma[G / \perp] \vee F'\sigma[G / T]$ *is the resolvent of the premises on G.*

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 simplification rules for* $\perp$*, T (equivalencies for logical constants). A refutational proof represents a proof of a formula G (goal) from the set* *of special axioms N. It is assumed that $Sig(\alpha) = 1$ for $\forall\alpha$ in $F \in N \cup \neg G$ formula, every formula in a proof has no free variable and has no quantifier for a variable not occurring in the formula.*

For any fuzzy deductive database we can encode fuzzy rules and facts statements into special axioms and formulate a query as a goal G.

We can use existing theorem prover for fuzzy predicate logic FPLGERDS as an inference engine [5]. It enables to edit knowledge bases of FPL with evaluated syntax and performing deduction on required goals. The fig. 1 shows GERDS's GUI.

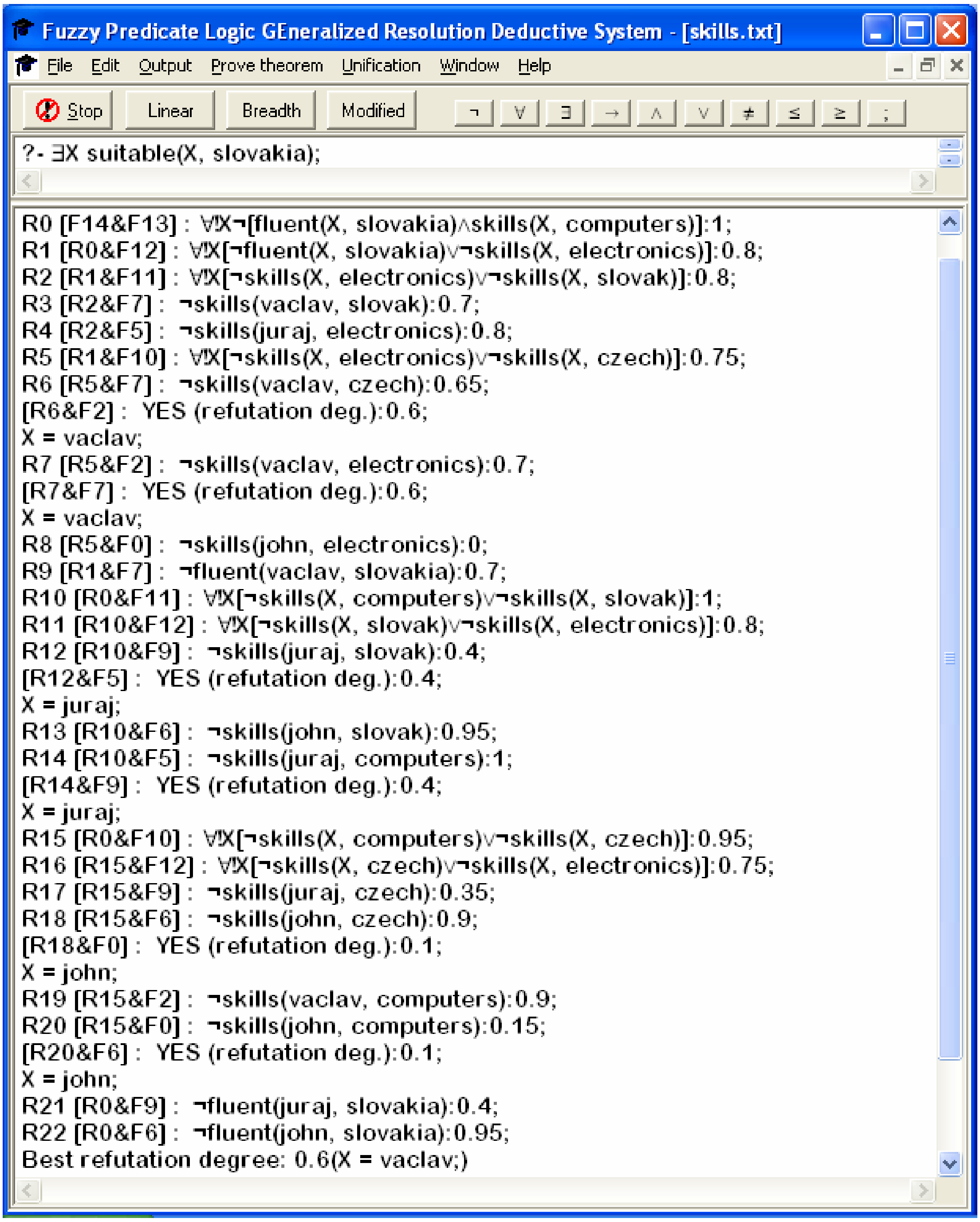


```
skills(john, czech):0.2;
skills(john, english):0.95;
skills(vaclav, czech):0.95;
skills(vaclav, english):0.01;
skills(juraj, english):0.7;
skills(juraj, slovak):1;
skills(john, computers):0.95;
skills(vaclav, electronics):0.9;
skills(juraj, metalurgy);
skills(juraj, computers):0.4;
∀X [skills(X, czech) → fluent(X, slovakia)]:0.95;
∀X [skills(X, slovak) → fluent(X, slovakia)]:1;
∀X [skills(X, electronics) → skills(X, computers)]:0.8;
∀X [[fluent(X, slovakia) ∧ skills(X, computers)]→ suitable(X, slovakia)];
?- ∃X suitable(X, slovakia);
```

```
R15 [R14&F9] : ¬skills(juraj, czech):0.35;
R16 [R14&F6] : ¬skills(john, czech):0.9;
[R16&F0] : YES (refutation deg.):0.1;
X = john;
R17 [R14&F2] : ¬skills(vaclav, computers):0.9;
R18 [R14&F0] : ¬skills(john, computers):0.15;
[R18&F6] : YES (refutation deg.):0.1;
X = john;
R19 [R0&F9] : ¬fluent(juraj, slovakia):0.4;
R20 [R0&F6] : ¬fluent(john, slovakia):0.95;
Best refutation degree: 0.6(X = vaclav;)
Solving time : 0.14 s. Used memory for proof: 11892 B.
Total allocated for GERDS: 69644 B.
```

Fig. 1. Fuzzy Predicate Logic Generalized Resolution Deductive System

We can observe results of inference on Fig. 2, solving our example for query about suitability of a person being a representative in Slovakia. We can see that according to the semantics of Lukasiewicz operators the best refutation degree is given for "Vaclav", which attains the refutation/provability degree of 0.6. There is also a positive proof for "Juraj" at the degree 0.4 and "John" gives 0.1.



```
?- ∃X suitable(X, slovakia);
```

```
R0 [F14&F13] : ∀X¬[fluent(X, slovakia)∧skills(X, computers)]:1;
R1 [R0&F12] : ∀X[¬fluent(X, slovakia)∨¬skills(X, electronics)]:0.8;
R2 [R1&F11] : ∀X[¬skills(X, electronics)∨¬skills(X, slovak)]:0.8;
R3 [R2&F7] : ¬skills(vaclav, slovak):0.7;
R4 [R2&F5] : ¬skills(juraj, electronics):0.8;
R5 [R1&F10] : ∀X[¬skills(X, electronics)∨¬skills(X, czech)]:0.75;
R6 [R5&F7] : ¬skills(vaclav, czech):0.65;
[R6&F2] : YES (refutation deg.):0.6;
X = vaclav;
R7 [R5&F2] : ¬skills(vaclav, electronics):0.7;
[R7&F7] : YES (refutation deg.):0.6;
X = vaclav;
R8 [R5&F0] : ¬skills(john, electronics):0;
R9 [R1&F7] : ¬fluent(vaclav, slovakia):0.7;
R10 [R0&F11] : ∀X[¬skills(X, computers)∨¬skills(X, slovak)]:1;
R11 [R10&F12] : ∀X[¬skills(X, slovak)∨¬skills(X, electronics)]:0.8;
R12 [R10&F9] : ¬skills(juraj, slovak):0.4;
[R12&F5] : YES (refutation deg.):0.4;
X = juraj;
R13 [R10&F6] : ¬skills(john, slovak):0.95;
R14 [R10&F5] : ¬skills(juraj, computers):1;
[R14&F9] : YES (refutation deg.):0.4;
X = juraj;
R15 [R0&F10] : ∀X[¬skills(X, computers)∨¬skills(X, czech)]:0.95;
R16 [R15&F12] : ∀X[¬skills(X, czech)∨¬skills(X, electronics)]:0.75;
R17 [R15&F9] : ¬skills(juraj, czech):0.35;
R18 [R15&F6] : ¬skills(john, czech):0.9;
[R18&F0] : YES (refutation deg.):0.1;
X = john;
R19 [R15&F2] : ¬skills(vaclav, computers):0.9;
R20 [R15&F0] : ¬skills(john, computers):0.15;
[R20&F6] : YES (refutation deg.):0.1;
X = john;
R21 [R0&F9] : ¬fluent(juraj, slovakia):0.4;
R22 [R0&F6] : ¬fluent(john, slovakia):0.95;
Best refutation degree: 0.6(X = vaclav;)
```

Fig. 2. Results for query

2.2. Resolution strategies for practically successful reasoning

Further we have to ask about more effective proof searching resolution strategies than blind breadth-first search. Experiments concerning prospective inference strategies can be performed with 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:

$$(R_{old} \mid\!- R_{new}) \wedge (U, R_{new} \mid\!- \bot) \Rightarrow (U, R_{old} \mid\!- \bot)$$

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

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:

$$R_{old} \mid\!- R_{new} \Leftrightarrow \neg(R_{old} \rightarrow R_{new}) \mid\!- \bot$$

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: $Sbt(x) = a$, $Res = \neg(\ \bot \rightarrow \bot) \vee \neg(\ T \rightarrow T) \Rightarrow \bot$

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 $U \mid\!-_a R_{old} \wedge U \mid\!-_b R_{new} \wedge b \leq a$ 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 $U \mid\!-_a R_{old} \wedge U \mid\!-_b R_{new} \wedge b \geq a$ then Kill R_{old} (resolvent is removed).

We have shown possibilities of combining the idea of fuzzy logic, database technology and the resolution principle. Additionally we have presented some preliminary results concerning usage of efficient strategies like DCF. Very important issue lies in the computer application FPLGERDS that implements the above presented ideas. Its inference engine may serve as a background for database engine of proposed theoretical ideas or for further experiments with time and space efficiency of the resolution principle.

3. Conclusion

We have shown possibilities of combining the idea of fuzzy logic, database technology and the resolution principle. Additionally we have presented some preliminary results concerning usage of efficient strategies like DCF. Very important issue lies in the computer application FPLGERDS that implements the above presented ideas. Its inference engine may serve as a background for database engine of proposed theoretical ideas or for further experiments with time and space efficiency of the resolution principle.

References

[1] ABITEBOUL, S., HULL, R., VIANU, V. Foundations of Databases. Addison-Wesley Pubs. Co., 1995

[2] BACHMAIR, L., GANZINGER, H. A theory of resolution. *Technical report*: Max-Planck-Institut für Informatik, 1997

[3] BACHMAIR, L., GANZINGER, H. Resolution theorem proving. In *Handbook of Automated Reasoning*, MIT Press, 2001

[4] HABIBALLA, H. Non-clausal Resolution Theorem Proving for Fuzzy Description Logic. In *SOFSEM 2006 : Theory and practice of computer science – 32nd conference on current trends in theory and practice of computer science*. 21.1.2006-27.1.2006 Měřín. Praha : Institute of Computer Science, Czech Academy of Sciences, 2006, pp. 1-12.

[5] HABIBALLA, H. *Fuzzy Predicate Logic Generalized Resolution Deductive System*. Technical Report, Institute for Research and Application of Fuzzy Modeling, University of Ostrava, 2006.

[6] HÁJEK, P. *Metamathematics of fuzzy logic*. Kluwer Academic Publishers, 2000.

[7] HÁJEK, P. Making fuzzy description logic more general. *Research report*: Institute of Computer Science, Czech Academy of Sciences, 2005

[8] LEHMKE, S. On resolution-based theorem proving in propositional fuzzy logic with bold connectives. University of Dortmund, 1995

[9] NOVÁK, V., PERFILIEVA, I., MOCKOR, J. *Mathematical principles of fuzzy logic*. Kluwer Academic Publishers, 1999

[10] ULLMAN, D. J. Principles of Database and Knowledge-Base Systems. Volume I, II. Computer Sc. Press, 1988

[11] VOTJAS, P., ALSINET, T., AND GODO, L. Different models of fuzzy logic programming with fuzzy unification (towards a revision of fuzzy database model). In *Proceedings of Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, 2001.