University of Ostrava

Institute for Research and Applications of Fuzzy Modeling

Non-clausal Resolution Theorem Proving for Fuzzy Predicate Logic

Hashim Habiballa

Research report No. 70

2005

Submitted/to appear:

Fuzzy sets and systems

Supported by:

Research grant of Czech Ministry of Education - MSM 6198898701

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
30. dubna 22, 701 03 Ostrava 1, Czech Republic

tel.: +420-69-6160234 fax: +420-69-6120 478
e-mail: Hashim.Habiballa@osu.cz

Non-clausal Resolution Theorem Proving for Fuzzy Predicate Logic

Hashim Habiballa [1],

Institute for Research and Applications of Fuzzy Modeling, University of Ostrava, 30. dubna 22, Ostrava, Czech Republic

Abstract

The article presents refutational resolution theorem proving system for the Fuzzy Predicate Logic of First-Order (FPL) based on the general (non-clausal) resolution rule. There is also presented an unification algorithm handling existentiality without the need of skolemization. Its idea follows from the general resolution with existentiality for the first-order logic. When the prover is constructed it provides the deductive system, where existing resolution strategies and its implementations may be used with some limitations arising from specific properties of the FPL.

Key words: Fuzzy inference systems, Non-classical logics, Automated theorem proving, Non-clausal resolution, General resolution, Unification

1 Introduction

The Fuzzy Predicate Logic of First-Order (FPL) forms a powerful generalization of the classical two-valued logic [10]. This generalization brings several hard problems with automated theorem proving especially when utilizing the widely used resolution principle. The resolution-based reasoning in its usually preferred way of application uses the clausal form formulas. In the FPL the standard properties related to the clausal form transformation do not hold. Although there are some attempts to apply the resolution principle in the fuzzy propositonal calculus [9] we will present more general and more straightforward way. We will present the refutational resolution theorem proving system for FPL ($RRTP_{FPL}$) based on general (non-clausal) resolution principle in first-order logic (FOL) [1]. It requires

Email address: `Hashim.Habiballa@osu.cz` (Hashim Habiballa).
URL: `http://www.volny.cz/habiballa/index.htm` (Hashim Habiballa).
[1] Supported by research grant of Czech Ministry of Education - MSM 6198898701

more complex unification algorithm based on the polarity criteria and the quantifier mapping. The below presented idea has its origin in implementation of non-clausal resolution theorem prover [5].

2 Background from first-order logic

For the purposes of ($RRTP_{FPL}$) we will use generalized principle of resolution, which is defined in the handbook [2].

General resolution

$$\frac{F[G] \quad F'[G]}{F[G/\bot] \vee F'[G/\top]} \tag{1}$$

where F and F' are formulas - premises of first-order logic and G represents an occurrence of a subformula of F and F'. The expression below the line means the resolvent of premises on G. Every occurrence of G is replaced by false in the first formula and by true in the second one. It is also called F the positive, F' the negative premise.

Example 1 ***General resolution with equivalence***
1. $a \wedge c \leftrightarrow b \wedge d$ *(axiom) 2.* $a \wedge c$ *(axiom) 3.*$\neg[b \wedge d]$ *(axiom) - negated goal*
4. $[a \wedge \bot] \vee [a \wedge \top]$ *(resolvent from (2), (2) on c)* $\Rightarrow a$
5. $[a \wedge \bot] \vee [a \wedge \top \leftrightarrow b \wedge d]$ *((2), (1) on c)* $\Rightarrow a \leftrightarrow b \wedge d$
6. $\bot \vee [\top \leftrightarrow b \wedge d]$ *((4), (5) on a)* $\Rightarrow b \wedge d$
7. $\bot \wedge d \vee \top \wedge d$ *((6), (6) on b)* $\Rightarrow d$*, 8.* $b \wedge \bot \vee b \wedge \top$ *((6), (6) on d)* $\Rightarrow b$
9. $\bot \vee \neg[\top \wedge d]$ *((8), (3) on b)* $\Rightarrow \neg d$*, 10.* $\bot \vee \neg\top$ *((7), (9) on d)* $\Rightarrow \bot$ *(refutation)*

Standard unification algorithms require variables to be treated only as universally quantified ones. We will present more general unification algorithm, which simulates skolemization. It should be stated that the following unification process doesn't allow an occurrence of the equivalence connective. It is needed to remove the equivalence by the rewrite rule: $A \leftrightarrow B \Rightarrow [A \rightarrow B] \wedge [B \rightarrow A]$. We assume that the language and semantics of first-order logic (FOL) is standard. We use terms - individuals $(a, b, c, ...)$, functions (with n arguments) $(f, g, h, ...)$, variables $(X, Y, Z, ...)$, predicates(with n arguments) $(p, q, r, ...)$, logical connectives $(\wedge, \vee, \rightarrow, \neg)$, quantifiers $(\exists, \forall)$ and logical constants $(\bot, \top)$.

Definition 1 ***Structural notions of a FOL formula***
Let F be a formula of FOL then the structural mappings Sub (subformula), Sup (superformula), Pol (polarity) and Lev (level) are defined as follows:

$F = G \wedge H$ *or* $F = G \vee H \Rightarrow$	$Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ $Pol(G) = Pol(F)$, $Pol(H) = Pol(F)$
$F = G \rightarrow H \Rightarrow$	$Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ $Pol(G) = -Pol(F)$, $Pol(H) = Pol(F)$
$F = \neg G \Rightarrow$	$Sub(F) = \{G\}$, $Sup(G) = F$ $Pol(G) = -Pol(F)$
$F = \exists \alpha G$ *or* $F = \exists \alpha G \Rightarrow$ *(α is a variable)*	$Sub(F) = \{G\}$, $Sup(G) = F$ $Pol(G) = Pol(F)$
F *is a formula*	$Sup(F) = \emptyset \Rightarrow Lev(F) = 0$, $Pol(F) = 1$ $Sup(F) \neq \emptyset \Rightarrow Lev(F) = Lev(Sup(F)) + 1$

For mappings Sub and Sup reflexive and transitive closures Sub^* and Sup^* are defined recursively as follows:
1. $Sub^*(F) \supseteq \{F\}$, $Sup^*(F) \supseteq \{F\}$
2. $Sub^*(F) \supseteq \{H | G \in Sub^*(F) \wedge H \in Sub(G)\}$, $Sup^*(F) \supseteq \{H | G \in Sup^*(F) \wedge H \in Sup(G)\}$

These structural mappings provide a framework for the assignment of the quantifiers to the variable occurrences. The subformula and superformula mappings and its closures encapsulate essential hierarchical information of a formula structure. Level gives the ordering with respect to the scope of variables (which is essential for skolemization simulation - unification is restricted for existential variables). Polarity enables to decide the global meaning of a variable (e.g. globally an existential variable is universal if its quantification subformula has negative polarity). Sound unification requires further definitions on variable quantification.

Definition 2 ***Variable assignment, substitution and significance***
Let F be a formula of FOL, $G = p(t_1, ..., t_n) \in Sub^*(F)$ atom in F and α a variable occurring in t_i. Variable mappings Qnt(quantifier assignment), Sbt (variable substitution) and Sig(significance) are defined as follows:

$Qnt(\alpha) = \{Q\alpha H | Q = \exists \vee Q = \forall, H, I \in Sub^*(F), Q\alpha H \in Sup^*(G),$
$\forall Q\alpha I \in Sup^*(G) \Rightarrow Lev(Q\alpha I) < Lev(Q\alpha H)\}$
For substitution of term t' into α in F ($F[\alpha/t']$) it holds:
$Sbt(Qnt(\alpha)) = t'$
$Sig(\alpha) = 1$ *iff variable is significant,* $Sig(\alpha) = 0$ *iff variable is not significant w.r.t. existential substitution.*

Note that with the Qnt mapping (assignment of first name matching quantifier variable in a formula hierarchy from its bottom) we are able to distinguish between variables of the same name and there is no need to rename any variable. Sbt mapping holds substituted terms in a quantifier and there is no need to rewrite all occurrences of a variable when working with this mapping within unification. It is also clear that if $Qnt(\alpha) = \emptyset$ then α is a free variable. These variables could be simply avoided by introducing new universal quantifiers to F. Significance mapping is important for differentiating between original formula universal variables and newly introduced ones during proof search (an existential variable can't be bounded with that ones).

Lemma 1 ***Free variables quantification***
Let F be a formula of FOL and α be a free variable in F ($Qnt(\alpha) = \emptyset$), then the formula $F' = \forall \alpha F$ is equivalent to F.

Before we can introduce standard unification algorithm, we have to formulate the variable unification restriction for existential and universal variable through the notion of globally universal and globally existential variable (it simulates conversion into the prenex normal form). It is clear w.r.t. skolemization technique that an existential variable can be substituted into an universal one only if all globally universal variables over its scope have been already substituted by a term. Skolem functors function in the same way.

Definition 3 ***Global quantification and variable unification restriction***
Let F be a formula without free variables and α be a variable occurring in a term of F.

- *(1) α is globally universal variable iff* $(Qnt(\alpha) = \forall \alpha H \wedge Pol(Qnt(\alpha)) = 1)$ *or* $(Qnt(\alpha) = \exists \alpha H \wedge Pol(Qnt(\alpha)) = -1)$
- *(2) α is globally existential variable iff* $(Qnt(\alpha) = \exists \alpha H \wedge Pol(Qnt(\alpha)) = 1)$ *or* $(Qnt(\alpha) = \forall \alpha H \wedge Pol(Qnt(\alpha)) = -1)$

Let F_1 be a formula and α be a variable occurring in F_1, F_2 be a formula, t be a term occurring in F_2 and β be a variable occurring in F_2. Variable Unification Restriction (VUR) for (α,t) holds if one of the conditions 1. and 2. holds:

- *(1) α is a globally universal variable and $t \neq \beta$, where β is a globally existential variable and α not occurring in t (non-existential substitution)*
- *(2) α is a globally universal variable and $t = \beta$, where β is a globally existential variable and $\forall F \in Sup^*(Qnt(\beta))$, $F = Q\gamma G$, $Q \in \{\forall, \exists\}$, γ is a globally universal variable, $Sig(\gamma) = 1 \Rightarrow Sbt(\gamma) \neq \emptyset$ (existential substitution)*

Further we define the most general unification algorithm based on recursive conditions (extended unification in contrast to the standard MGU).

Definition 4 ***Most general unifier algorithm***
Let F_1, F_2 be formulas of FOL, $G = p(t_1, ..., t_n)$ be an atom occurring in F_1 and $G' = r(u_1, ..., u_n)$ be an atom occurring in F_2. Most general unifier (MGU)(substitution mapping) σ is obtained by following atom and term unification steps or the algorithm returns fail-state for unification.
Atom unification

(1) *if* $n = 0$ *and* $p = r$ *then* $\sigma = \emptyset$ *and the unifier exists (success-state).*
(2) *if* $n > 0$ *and* $p = r$ *then perform term unification for every pair* (t_1, u_1)*; if for every pair unifier exists then* $MGU(G, G') = \sigma$ *(success state).*
(3) *In any other case unifier doesn't exist (fail-state).*

Term unification (t', u')

(1) *if* $t' = a$, $u' = b$ *are individual constants and* $a = b$ *then for (t',u') unifier exist (success-state).*
(2) *if* $t' = f(t'_1, ..., t'_m)$, $u' = g(u'_1, ..., u'_n)$ *are function symbols with arguments and* $f = g$ *then unifier for* (t', u') *exists iff unifier exists for every pair* (t'_1, u'_1), ..., (t'_n, u'_n) *(success-state).*
(3) *if* $t' = \alpha$ *is a variable and VUR for* (t', u') *holds then unifier exists for* (t', u') *holds and* $(Sbt(\alpha) = u') \in \sigma$ *(success-state).*
(4) *if* $u' = \alpha$ *is a variable and VUR for* (u', t') *holds then unifier exists for* (t', u') *holds and* $(Sbt(\alpha) = t') \in \sigma$ *(success-state).*
(5) *if* $u' = \alpha$, $t' = \beta$ *are variables and* $Qnt(\alpha) = Qnt(\beta)$ *then unifier exists for* (t', u') *(success-state) (occurrence of the same variable).*
(6) *In any other case unifier doesn't exist (fail-state).*

With the above-defined notions it is simple to state the general resolution rule for FOL (without the equivalence connective). It extends the definition from [1].

Definition 5 ***General resolution for first-order logic*** (GR_{FOL})

$$\frac{F[G_1, , ..., G_k] \quad F'[G'_1, ..., G'_n]}{F\sigma[G/\bot] \vee F'\sigma[G/\top]} \tag{2}$$

where σ *is the union of the most general unifiers (mgu) of the atom pairs* (G_1, G_i) *and* (G_1, G'_j), $G_1, ..., G_k, G'_1, ..., G'_n, G = G_1\sigma$. *For every variable* α *in* F *or* F', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow Sig(\alpha) = 1$ *in* F *or* F' *iff* $Sig(\alpha) = 1$ *in* $F\sigma[G/\bot] \vee F'\sigma[G/\top]$. *F is called positive and F' is called negative premise, G represents an occurrence of a subformula (mgu applies to all atoms occurring in* F, F'*). The expression below the line represents the resolvent of premises on G.*

Note that with Qnt mapping we are able to distinguish variables not only by its name (which may not be unique), but also with this mapping. Sig property enables to separate variables, which were not originally in the scope of an existential variable. When utilizing the rule it should be set the Sig mapping for every variable in axioms and negated goal to 1. We present a very simple example of existential variable unification before we introduce the refutational theorem prover for FOL.

Example 2 *It doesn't hold* $\forall X \exists Y p(X, Y) \models \exists Y \forall X p(X, Y)$
and it holds $\exists Y \forall X p(X, Y) \models \forall X \exists Y p(X, Y)$.
(General Y for all X can't be deduced from Y specific for X but contrary it holds)
F0 : $\forall X \exists Y p(X, Y)$. *F1 (*$\neg$*query) :* $\forall Y \exists X \neg p(X, Y)$.
R[F1&F1] : $\bot \vee \top$. *R[F0&F0] :* $\bot \vee \top$.

In this example F0 and F1 can't resolve, since $\forall X \exists Y p(X,Y)$ and $\forall Y \exists X \neg p(X,Y)$ have no unifier. It is impossible to substitute universal X from F0 with X from F1, because X from F1 is existential and its superior variable Y is not assigned with a value. Counter-example with variable Y is the same instance and it is not allowed to substitute anything into an existential variable. However, in the next example a unifier exists.
F0 : $\exists Y \forall X p(X,Y)$. F1 ($\neg$query) : $\exists X \forall Y \neg p(X,Y)$.
R[F1&F0] : YES. R[F0&F1] : YES.
In the second case the universal variable X from F0 could be assigned with existential Y because Y in F1 has no superior variable. Then existential Y from F0 can substitute the universal Y from F1 for the same reason.

Now we entail above presented definitions by introduction of the refutational theorem proving with the rule GR_{FOL}. We assume standard notions and theorems stated in [2].

Definition 6 ***Refutational resolution theorem prover for FOL***
Refutational non-clausal resolution theorem prover for FOL ($RRTP_{FOL}$) is the inference system with the inference rule GR_{FOL} and sound auxiliary simplification rules for $\bot$, $\top$ (standard equivalencies for logical constants). A refutational proof of the goal G from the set of axioms $N = \{A_1, ..., A_m\}$ is a sequence of formulas $F_1, F_2, ..., F_n, \bot$, where F_i is an axiom from N, $\neg G$ or a resolvent from premises F_k and F_l $(k, l < i)$, where simplification rules may be applied to the resolvent. 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.

Theorem 1 *$RRTP_{FOL}$ system is a sound and complete inference system.*

Proofs for both the properties will be proved in the section concerning the $RRTP_{FPL}$. Since FOL is a special case of FPL where the truth value set is a set $\{0, 1\}$ and in this truth valuation doesn't matter if we use classical or Łukasiewicz operators, the proof for FPL holds also for FOL.

3 General resolution for Fuzzy Predicate Logic

The fuzzy predicate logic with evaluated syntax is a flexible and fully complete formalism, which will be used for below presented extension [10]. For the purposes of fuzzy extension the Modus ponens rule was considered as an inspiration [4]. We will suppose that set of truth values is Łukasiewicz algebra. Therefore we will assume standard notions of conjunction, disjunction etc. to be bound with Łukasiewicz operators.

We will assume Łukasewicz algebra to be

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

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 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 $\mathcal{L}_Ł$ will be used in the sequel:

$$a \otimes 1 = a \quad a \otimes 0 = 0 \quad a \oplus 1 = 1 \quad a \oplus 0 = a$$

$$a \rightarrow 1 = 1 \quad a \rightarrow 0 = \neg a \quad 1 \rightarrow a = a \quad 0 \rightarrow a = 1$$

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, logical constants – $\{\mathbf{a}\,|a \in [0,1]\}$. Instead of 0 we write $\perp$ and instead of 1 we write $\top$, connectives - **&** (Łukasiewicz conjunction), $\wedge$ (conjunction), $\boldsymbol{\nabla}$ (Łukasiewicz disjunction), $\vee$ (disjunction), $\Rightarrow$ (implication), $\Leftrightarrow$ (equivalence), $\neg$ (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, $\mathcal{D}(\mathbf{a}) = a$ for $a \in [0,1]$, $\mathcal{D}(A \,\&\, B) = \mathcal{D}(A) \otimes \mathcal{D}(B)$, $\mathcal{D}(A \wedge B) = \mathcal{D}(A) \wedge \mathcal{D}(B)$, $\mathcal{D}(A \boldsymbol{\nabla} B) = \mathcal{D}(A) \oplus \mathcal{D}(B)$, $\mathcal{D}(A \vee B) = \mathcal{D}(A) \vee \mathcal{D}(B)$, $\mathcal{D}(A \Rightarrow B) = \mathcal{D}(A) \rightarrow \mathcal{D}(B)$, $\mathcal{D}(A \Leftrightarrow B) = \mathcal{D}(A) \leftrightarrow \mathcal{D}(B)$, $\mathcal{D}(\neg A) = \neg\mathcal{D}(A)$, $\mathcal{D}(\forall X(A)) = \bigwedge \mathcal{D}(A[x/d]|d \in D)$,$\mathcal{D}(\exists X(A)) = \bigvee \mathcal{D}(A[x/d]|d \in D)$
- for every subformula defined above $Sub, Sup, Pol, Lev, Qnt, Sbt, Sig$ and other derived properties defined in section 2 hold (where the classical FOL connective is presented the Łukasiewicz 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

$$a \big/ A$$

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.

Definition 7 *Inference rule*
An n-ary inference rule r in the graded logical system is a scheme

$$r : \frac{a_1 \big/ A_1, ..., \; a_n \big/ A_n}{r^{evl}(a_1, ..., a_n) \big/ r^{syn}(A_1, ..., A_n)} \tag{3}$$

using which the evaluated formulas $a_1 \big/ A_1, ..., \; a_n \big/ A_n$ *are assigned the evaluated formula* $r^{evl}(a_1, ..., a_n) \big/ r^{syn}(A_1, ..., A_n)$*. 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 8 *Formal fuzzy theory*
A formal fuzzy theory T in the language J is a triple

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

where $\mathrm{LAx} \underset{\sim}{\subseteq} F_J$ *is a fuzzy set of logical axioms,* $\mathrm{SAx} \underset{\sim}{\subseteq} F_J$ *is a fuzzy set of special axioms, and R is a set of sound inference rules.*

Definition 9 ***Evaluated proof, refutational proof and refutation degree***
An evaluated formal proof of a formula A from the fuzzy set $X \underset{\sim}{\subseteq} F_J$ is a finite sequence of evaluated formulas

$$w := a_0/A_0,\ a_1/A_1, ...,\ a_n/A_n \tag{4}$$

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

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

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 (4).
An evaluated refutational formal proof of a formula A form X is w, where additionally

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

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

Definition 10 *Provability and truthfulness*
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.

$$T \vdash_a A \text{ iff } a = \bigvee \{Val(w) \mid w \text{ is a proof of } A \text{ from } \mathrm{LAx} \cup \mathrm{SAx}\} \tag{5}$$

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

$$T \models_a A \text{ iff } a = \bigwedge \{\mathcal{D}(A) \mid \mathcal{D} \models T\}, \text{ where the condition } \mathcal{D} \models T \text{ holds}$$
$$\text{if for every } A \in \mathrm{LAx} : \mathrm{LAx}(A) \leq \mathcal{D}(A),\ A \in \mathrm{SAx} : \mathrm{SAx}(A) \leq \mathcal{D}(A) \tag{6}$$

The fuzzy modus ponens rule could be formulated:

Definition 11 *Fuzzy modus ponens*

$$r_{MP} : \frac{a/A,\ b/A \Rightarrow B}{a \otimes b/B} \tag{7}$$

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

$$r_{GR} : \frac{{}^{a}\big/ F[G_1, , ..., G_k],\ {}^{b}\big/ F'[G'_1, ..., G'_n]}{a \otimes b \big/ F\sigma[G/\bot] \nabla F'\sigma[G/\top]} \tag{8}$$

where σ is the union of the most general unifiers (mgu) of the atom pairs (G_1, G_i) and (G_1, G'_j), $G_1, ..., G_k, G'_1, ..., G'_n, G = G_1\sigma$. For every variable α in F or F', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow Sig(\alpha) = 1$ in F or F' iff $Sig(\alpha) = 1$ in $F\sigma[G/\bot] \vee F'\sigma[G/\top]$. F is called positive and F' is called negative premise, G represents an occurrence of a subformula (mgu applies to all atoms occurring in F, F'). The expression below the line represents the resolvent of premises on G.

Definition 13 ***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 from definition 9 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.

Example 3 ***Proof of child's happiness by*** r_{GR}
Consider the following knowledge (significantly simplified in contrast to the reality) about child's happiness. We suppose that a child is happy in the degree 0.8 if it has mother and father. Further we suppose that a child is happy in the degree 0.5 if it has a lot of toys (we suppose parents are a bit more important for children). We will present several proofs and then we mark the best provability degree from the following axioms. It was used the automated theorem prover of the author for classical logic [5]. Xa. steps represent application of simplification rules for $\bot$ and $\top$.

Common proof members (axioms):

1. $^{0.8}\Big/ \forall X[\exists Y[child(X,Y)\,\&\,female(Y)]$
$\&\,\exists Y[child(X,Y)\,\&\,male(Y)] \Rightarrow happy(X)]$ *(happy with parents - 0.8)*

2. $^{0.5}\Big/ \forall X[toys(X) \Rightarrow happy(X)]$ *(happy with toys - 0.5)*

3. $^{1}\Big/ child(johana, hashim)$ *(clear crisp fact)*

4. $^{1}\Big/ child(johana, lucie)$ *(clear crisp fact)*

5. $^{1}\Big/ male(hashim)$ *(clear crisp fact)*

6. $^{1}\Big/ female(lucie)$ *(clear crisp fact)*

7. $^{0.9}\Big/ toys(johana)$ *(johana has a lot of toys - 0.9)*

8. $^{1}\Big/ \neg happy(johana)$ *(negated goal - is johana happy?)*

Proof 1:

9. $^{0.9 \otimes 0.5}\Big/ \bot \nabla [\top \Rightarrow happy(johana)]$

9a. $^{0.4}\Big/ happy(johana)$ *(r_{GR} on 7.,2., $Sbt(X) = johana$)*

10. $^{1 \otimes 0.4}\Big/ \bot \nabla \neg \top$

10a. $^{0.4}\Big/ \bot$ *(r_{GR} on 9.,8.)*

(happy(johana) is provable in 0.4)

Proof 2:

9. $0.8 \otimes 1 \Big/ [\exists Y[child(johana, Y) \,\&\, female(Y)]$

$\&\, \exists Y[child(johana, Y) \,\&\, male(Y)] \Rightarrow \bot] \nabla \neg\top$

9a. $0.8 \Big/ \neg[\exists Y[child(johana, Y) \,\&\, female(Y)]$

$\&\, \exists Y[child(johana, Y) \,\&\, male(Y)]]$ *(r_{GR} on 1.,8., $Sbt(X) = johana$)*

10. $0.8 \otimes 1 \Big/ \neg[[child(johana, lucie) \,\&\, \top]$

$\&\, \exists Y[child(johana, Y) \,\&\, male(Y)]] \nabla \bot$

10a. $0.8 \Big/ \neg[child(johana, lucie)$

$\&\, \exists Y[child(johana, Y) \,\&\, male(Y)]]$ *(r_{GR} on 6.,9., $Sbt(Y) = lucie$)*

11. $0.8 \otimes 1 \Big/ \neg[child(johana, lucie)$

$\&[child(johana, hashim) \,\&\, \top]] \nabla \bot$

11a. $0.8 \Big/ \neg[child(johana, lucie)$

$\&\, child(johana, hashim)]$ *(r_{GR} on 5.,10., $Sbt(Y) = hashim$)*

12. $0.8 \otimes 1 \Big/ \neg[\top \,\&\, child(johana, hashim)] \nabla \bot$

12a. $0.8 \Big/ \neg[child(johana, hashim)]$ *(r_{GR} on 4.,11.)*

13. $0.8 \otimes 1 \Big/ \neg\top \nabla \bot$

13a. $0.8 \Big/ \bot$ *(r_{GR} on 3.,12.)*

(happy(johana) is provable in 0.8)

We have stated two different proofs and it is clear that several other proofs could be constructed. Let us note that these proofs either consist of redundant steps or they are variants of Proof 1 and Proof 2, where only the order of resolutions is different. So we can conclude that it is effectively provable that Johana is a happy child in the degree 0.8.

Lemma 2 ***Soundness of*** r_R
The inference r_R rule for FPL based on $\mathcal{L}_Ł$ is sound i.e. for every truth valuation $\mathcal{D}$,

$$\mathcal{D}(r^{syn}(A_1, ..., A_n)) \geq r^{evl}(\mathcal{D}(A_1), ..., \mathcal{D}(A_n)) \tag{9}$$

holds true.

PROOF: Before we solve the core of GR_{FPL} we should prove that the unification algorithm preserves soundness. But it could be simply proved since in the classical FPL with the rule of Modus-Ponens [10] from the axiom $\vdash (\forall x)A \Rightarrow A[x/t]$ and $\vdash (\forall x)A$ we can prove $A[x/t]$. For r_R we may rewrite the values of the left and right parts of equation (9):

$$\mathcal{D}(r^{syn}(A_1, ..., A_n)) = \mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])]$$

$$r^{evl}(\mathcal{D}(A_1), ..., \mathcal{D}(A_n)) = \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$$

It is sufficient to prove the equality for $\Rightarrow$ since all other connectives could be defined by it. By induction on the complexity of formula $|A|$, defined as the number of connectives, we can prove:

Let premises F_1 and F_2 be atomic formulas. Since they must contain the same subformula then $F_1 = F_2 = G$ and it holds

$$\mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] = D(\bot\boldsymbol{\nabla}\top) = 0 \oplus 1 = 1 \geq \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$$

Induction step: Let premises F_1 and F_2 be complex formulas and let A and B are subformulas of F_1, C and D are subformulas of F_2 and G is an atom where generally $F_1 = (A \Rightarrow B)$ and $F_2 = (C \Rightarrow D)$. The complexity of $|F_1| = |A| + 1$ or $|F_1| = |B| + 1$ and $|F_2| = |C| + 1$ or $|F_2| = |D| + 1$. Since they must contain the same subformula and for A, B, C, D the induction presupposition hold it remain to analyze the following cases:

(1) $F_1 = A \Rightarrow G \quad F_2 = G \Rightarrow D : \mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] = \mathcal{D}([A \Rightarrow \bot]\boldsymbol{\nabla}[\top \Rightarrow D]) = \mathcal{D}(\neg A\boldsymbol{\nabla}D) = 1 \wedge (1 - a + d)$
We have rewritten the expression into Łukasiewicz interpretation. Now we will try to rewrite the right side of the inequality, which has to be proven.
$\mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G]) = \mathcal{D}(A \Rightarrow G) \otimes \mathcal{D}(G \Rightarrow D) = 0 \vee ((1 \wedge (1 - a + g)) + (1 \wedge (1 - g + d)) - 1) = 1 \wedge (1 - a + d)$ The left and right side of the equation (9) are equal and therefore

$$\mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] \geq \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$$

for this case holds.

(2) $F_1 = A \Rightarrow G \quad F_2 = C \Rightarrow G : \mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] = \mathcal{D}([A \Rightarrow \bot]\boldsymbol{\nabla}[C \Rightarrow \top]) = 1 \geq \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$

(3) $F_1 = G \Rightarrow B \quad F_2 = G \Rightarrow D : \mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] = \mathcal{D}([\bot \Rightarrow B]\boldsymbol{\nabla}[\top \Rightarrow D]) = 1 \geq \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$

(4) $F_1 = G \Rightarrow B \quad F_2 = C \Rightarrow G : \mathcal{D}[\mathcal{D}(F_1[G/\bot])\boldsymbol{\nabla}\mathcal{D}(F_2[G/\top])] = \mathcal{D}([\bot \Rightarrow B]\boldsymbol{\nabla}[C \Rightarrow \top]) = 1 \geq \mathcal{D}(F_1[G]) \otimes \mathcal{D}(F_2[G])$

By induction we have proven that the inequality holds and the r_R is sound. The induction of the case where only one of the premises has greater complexity is included in the above solved induction step. $\diamond$

From this result we can conclude the completeness theorem. We will need two additional simplification rules for purposes of the proof:

Definition 14 ***Simplification rules for*** $\nabla, \Rightarrow$

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

The soundness of $r_{s\nabla}$ and $r_{s\Rightarrow}$ is straightforward.

Lemma 3 ***Provability and refutation degree for*** GR_{FPL}
$T \vdash_a A$ iff $a = \bigvee\{Val(w)|\ w$ is a refutational proof of A from $\mathrm{LAx} \cup \mathrm{SAx}\}$

PROOF: *If $T \vdash_a A$ then $a = \bigvee\{Val(w)|\ w$ is a proof of A from* $\mathrm{LAx} \cup \mathrm{SAx}\}$ *and for every such a proof of we can construct refutational proof as follows ($Val(w) \leq a$):*
$w := a/A$ *{a proof A}*, $1/\neg A$ *{a member of a refutational proof}*, $a \otimes 1/\bot$ $\{r_{GR}\}$
If $a = \bigvee\{Val(w)|\ w$ is a refutational proof of A from $\mathrm{LAx} \cup \mathrm{SAx}\}$*($Val(w) \leq a$):*
$w :=\ a_0/A_0, ..., a_i/A_i, 1/\neg A, ..., a/\bot$, *where $A_0, ..., A_i$ are axioms.*
There is a proof $w' :=\ a_0/A_0, ..., a_i/A_i, 1/\neg A \nabla A, a_{i+2}/A_{i+2} \nabla A, ..., a/\bot \nabla A$.
All the schemes of the type $A_j \nabla A$, $j > i$ could be simplified by sound simplification rules and the formula $\neg A \nabla A$ may be removed.
The proof $w'' :=\ a_0/A_0, ..., a_i/A_i, a_{i+2}/A_{i+2} \nabla A, ..., a/A$ *is a correct proof of A in the degree a since the formulas are either axioms or results of application of resolution.* $\diamond$

Theorem 2 ***Completeness for fuzzy logic with*** r_R***,*** $r_{s\nabla}$***,*** $r_{s\Rightarrow}$
Formal fuzzy theory, where r_{MP} is replaced with r_R, $r_{s\nabla}$, $r_{s\Rightarrow}$, is complete i.e. for every A from the set of formulas $T \vdash_a A$ iff $T \models_a A$.

PROOF: The left to right implication (soundness of such formal theory) could be easily done from the soundness of the resolution rule. Conversely it is sufficient to prove that the rule r_{MP} can be replaced by r_R, $r_{s\nabla}$, $r_{s\Rightarrow}$. Indeed, let w be a proof:
$w := a/A$ {a proof w_a }, $b/A \Rightarrow B$ {a proof $w_{A\Rightarrow B}$ }, $a \otimes b/B$ $\{r_{MP}\}$. Then we can replace it by the proof:

$w := a/A\{aproof w_a\}, b/A \Rightarrow B\{aproof w_{A\Rightarrow B}\}, a \otimes b/\bot \nabla[\top \Rightarrow B]\{r_R\}$,
$a \otimes b/\top \Rightarrow B\{r_{s\nabla}, a \otimes b/B\{r_{s\Rightarrow}\}$

Using the last sequence we can easily make a proof with r_{MP} also with the proposed r_R and simplification rules. Since usual formal theory with r_{MP} is complete as it is proved in [10], every fuzzy formal theory with these rules is also complete. Note that the non-ground case (requiring unification) could be simulated in the same way like in the proof of soundness. $\diamond$

4 Conclusion, further research and applications

The *Non-clausal Refutational Resolution Theorem Prover* forms a powerful inference system for automated theorem proving in fuzzy logic, which is significantly less discovered area in contrast with classical logic. At first it was recalled the notion of the ground non-clausal resolution and it was extended for use with existential variables (through *Variable Unification Restriction* based on structural formula mappings). The main contribution lies in the application into fuzzy logic, which gives a formalization of the refutational proving with the resolution principle and therefore it is essential for practically successful theorem proving in such areas like logic programming in fuzzy logic. Theoretical solution of the prover needed also some new notions to be defined especially the notion of the *refutational proof* and consequent notion of the *refutation degree*. We have established the equivalence property of the provability degree and the refutation degree. The next interesting area for the presented formalism is the field of semantic web and especially *description logic*, in which the author proposed also the usage of the resolution principle [6]. The recent idea of fuzzy description logic is naturally suitable for further extensions with the presented inference rules [8] and also reflects real situations as it could be observed from the last example. The last but not least further application relates to the previous author's works in the implementation of the non-clausal resolution principle [3]. This implementation called GERDS (GEneralised Resolution Deductive System) will be extended for usage in fuzzy logic and description logic.

References

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

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

[3] Habiballa, H. Non-clausal resolution - theory and practice. Research report: University of Ostrava, 2000,
http://www.volny.cz/habiballa/files/gerds.pdf

[4] Habiballa, H., Novák, V. Fuzzy general resolution. Research report: Institute for research and applications of fuzzy modeling, University of Ostrava, 2002,
http://ac030.osu.cz/irafm/ps/rep47.ps

[5] Habiballa, H. Non-clausal Resolution Theorem Prover. Research report, No.64: University of Ostrava, 2005,
http://ac030.osu.cz/irafm/ps/rep64.ps.gz

[6] Habiballa, H. Non-clausal Resolution Theorem Proving for Description Logic. Research report, No.66: University of Ostrava, 2005,
http://ac030.osu.cz/irafm/ps/rep66.ps.gz

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

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

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

[10] Novák, V., Perfilieva, I., Močkoř, J. Mathematical principles of fuzzy logic. Kluwer Academic Publishers, 1999