# RESOLUTION PRINCIPLE IN FUZZY PREDICATE LOGIC

**Hashim Habiballa***

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

**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 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 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 [No99]. 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 [Le95] 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) [Ba01]. It requires more complex unification algorithm based on the polarity criteria and the quantifier mapping.

## 2. Background from first-order logic

For the purposes of ($RRTP_{DL}$) we will use generalized resolution principle defined in [Ba01].

**General resolution**

$$\frac{F[G] \qquad F'[G]}{F[G/\bot] \vee F'[G/T]} \tag{1}$$

where the first-order formulas F and F′ are the premises of inference and G is an occurrence of a subformula of both F and F′. The expression $F[G/\bot] \vee F'[G/T]$ is the resolvent of the premises on G. Every occurrence of G is replaced by false in the first formula and by true in the second one. It is called F the positive, F' the negative premise, G the resolved subformula.

**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. $[a \wedge \bot] \vee [a \wedge T]$ (resolvent from (2), (2) on c) $\Rightarrow$ a

* Supported by research grant of Czech Ministry of Education - MSM 6198898701

5. $[a \wedge \bot] \vee [a \wedge T \leftrightarrow b \wedge d]$ ((2), (1) on c) $\Rightarrow a \leftrightarrow b \wedge d$
6. $\bot \vee [T \leftrightarrow b \wedge d]$ ((4), (5) on a) $\Rightarrow b \wedge d$
7. $\bot \wedge d \vee T \wedge d$ ((6), (6) on b) $\Rightarrow d$
8. $b \wedge \bot \vee b \wedge T$ ((6), (6) on d) $\Rightarrow b$
9. $\bot \vee \neg [T \wedge d]$ ((8), (3) on b) $\Rightarrow \neg d$
10. $\bot \vee \neg T$ ((7), (9) on d) $\Rightarrow \bot$ (refutation)

When trying to refine the general resolution rule for first-order logic and description logic, it is important to devise a sound and complete unification algorithm. Standard unification algorithms require variables to be treated only as universally quantified ones. We will present a more general unification algorithm, which can deal with existentially quantified variables without the need for those variables be eliminated by skolemization. It should be stated that the following unification process doesn't allow an occurrence of the equivalence connective. It is needed to remove equivalence by the following rewrite rule:

$A \leftrightarrow B \Leftrightarrow [\, A \rightarrow B \,] \wedge [\, B \rightarrow A \,]$.

We assume that the language and semantics of 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 ($\bot$, T). We also work with standard notions of logical and special axioms (sets LAx, SAx), logical consequence, consistency etc. as they are used in mathematical logic.

**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 $F = G \vee H$ | $Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ |
| | $Pol(G) = Pol(F)$, $Pol(H) = Pol(F)$ |
| $F = G \rightarrow H$ | $Sub(F) = \{G, H\}$, $Sup(G) = F$, $Sup(H) = F$ |
| | $Pol(G) = -Pol(F)$, $Pol(H) = Pol(F)$ |
| $F = \neg G$ | $Sub(F) = \{G\}$, $Sup(G) = F$ |
| | $Pol(G) = -Pol(F)$ |
| $F = \exists\alpha G$ or $F = \forall\alpha G$ | $Sub(F) = \{G\}$, $Sup(G) = F$ |
| ($\alpha$ is a variable) | $Pol(G) = Pol(F)$ |

$Sup(F) = \emptyset \Rightarrow Lev(F) = 0,\ Pol(F) = 1,$

$Sup(F) \neq \emptyset \Rightarrow Lev(F) = Lev(Sup(F)) + 1$

*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 framework for assignment of quantifiers to variable occurrences. It is needed for the correct simulation of skolemization (the information about a variable quantification in the prenex form). 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 also 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. We will introduce notions of the corresponding quantifier for a variable occurrence, substitution mapping and significance mapping (we have to distinguish between original variables occurring in special axioms and newly introduced ones in the proof sequence).

**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 $\alpha$ a variable occurring in $t_i$. Variable mappings Qnt(quantifier assignment), Sbt (variable substitution) and Sig(significance) are defined as follows:*

$Qnt(\alpha) = Q\alpha H$, *where* $Q = \exists \vee Q = \forall$, $H,I \in Sub^*(F)$, $Q\alpha H \in Sup^*(G)$, $\forall Q\alpha I \in Sup^*(G) \Rightarrow Lev(Q\alpha I) < Lev(Q\alpha H)$.

$F[\alpha/t']$ *is a substitution of term* $t'$ *into* $\alpha$ *in* $F \Rightarrow Sbt(\alpha) = t'$.

*A variable* $\alpha$ *occurring in* $F \in LAx \cup SAx$ *is significant w.r.t. existential substitution,* $Sig(\alpha) = 1$ *iff variable is significant,* $Sig(\alpha) = 0$ *otherwise.*

Note that with Qnt mapping (assignment of first name matching quantifier variable in a formula hierarchy from 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 $\alpha$ 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 it).
Before we can introduce the standard unification algorithm, we should formulate the notion of global universal and global existential variable (it simulates conversion into prenex normal form).

**Definition 3 Global quantification**

*Let F be a formula without free variables and* $\alpha$ *be a variable occurrence in a term of F.*
$\alpha$ *is a global universal variable* ($\alpha \in Var_\forall(F)$) *iff* ($Qnt(\alpha) = \forall\alpha H \wedge Pol(Qnt(\alpha)) = 1$) *or* ($Qnt(\alpha) = \exists\alpha H \wedge Pol(Qnt(\alpha)) = -1$)
$\alpha$ *is a global existential variable* ($\alpha \in Var_\exists(F)$) *iff* ($Qnt(\alpha) = \exists\alpha H \wedge Pol(Qnt(\alpha)) = 1$) *or* ($Qnt(\alpha) = \forall\alpha H \wedge Pol(Qnt(\alpha)) = -1$)
$Var_\forall(F)$ *and* $Var_\exists(F)$ *are sets of global universal and existential variables.*

It is clear w.r.t. skolemization technique that an existential variable can be substituted into an universal one only if all global universal variables over the scope of the existential one have been already substituted by a term. Skolem functors function in the same way. Now we can define the most general unification algorithm based on recursive conditions (extended unification in contrast to standard MGU).

**Definition 4 Most general unifier algorithm**

*Let* $G = p(t_1, ..., t_n)$ *and* $G' = r(u_1, ..., u_n)$ *be atoms. Most general unifier (substitution mapping)* $MGU(G, G') = \sigma$ *is obtained by following atom and term unification steps or the algorithm returns fail-state for unification. For the purposes of the algorithm we define the Variable Unification Restriction (VUR).*

***Variable Unification Restriction***

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

*1. $\alpha$ is a global universal variable and $t \neq \beta$, where $\beta$ is a global existential variable and $\alpha$ not occurring in t (non-existential substitution)*

*2. $\alpha$ is a global universal variable and $t = \beta$, where $\beta$ is a global existential variable and $\forall F \in Sup^*(Qnt(\beta))$, $F = Q\, \gamma G$, $Q \in \{\forall, \exists\}$, $\gamma$ is a global universal variable, $Sig(\gamma)=1 \Rightarrow (Sbt(\gamma) = r') \in \sigma$, $r'$ is a term (existential substitution).*

***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 pairs $(t_1, u_1), \ldots, (t_n, u_n)$; If for every pair unifier exists then $MGU(G,G') = \sigma$ obtained during term unification (success state).*

*3. In any other case unifier doesn't exist (fail-state).*

***Term unification $(t',u')$***

*1. 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).*

*2. if $t' = \alpha$ is a variable and $(Sbt(\alpha) = v') \in \sigma$ then perform term unification for $(v',u')$; The unifier for $(t',u')$ exists iff it exists for $(v',u')$ (success-state for an already substituted variable).*

*3. if $u' = \alpha$ is a variable and $(Sbt(\alpha) = v') \in \sigma$ then perform term unification for $(t',v')$; The unifier for $(t',u')$ exists iff it exists for $(t',v')$ (success-state for an already substituted variable).*

*4. if $t' = a$, $u' = b$ are constants and $a = b$ then for $(t',u')$ unifier exists (success-state).*

*5. if $t' = f(t'_1, \ldots, t'_m)$, $u' = g(u'_1, \ldots, u'_n)$ are function symbols with arguments and $f = g$ then unifier for $(t', u')$ exists iff unifier exists for every pair $(t'_1, u'_1), \ldots, (t'_n, u'_n)$ (success-state).*

*6. if $t' = \alpha$ is a variable and VUR for $(t', u')$ holds then unifier exists for $(t',u')$ holds and $\sigma = \sigma \cup (Sbt(\alpha) = u')$ (success-state).*

*7. if $u' = \alpha$ is a variable and VUR for $(u', t')$ holds then unifier exists for $(t',u')$ holds and $\sigma = \sigma \cup (Sbt(\alpha) = t')$ (success-state).*

*8. In any other case unifier doesn't exist (fail-state).*

*$MGU(A) = \sigma$ for a set of atoms $A = \{ G_1, \ldots, G_k \}$ is computed by atom unification for $(G_1, G_i)$, $\sigma_i = MGU(G_1, G_i)$, $\forall i, \sigma_0 = \emptyset$, where before every atom unification $(G_1, G_i)$, $\sigma$ is set to $\sigma_{i-1}$.*

With above defined notions it is simple to state the general resolution rule for FOL (without the equivalence connective). It conforms to the definition from [Ba97].

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

$$\frac{F[G_1,,\ldots,G_k] \quad F'[G'_1,\ldots,G'_n]}{F\sigma[G / \perp] \vee F'\sigma[G / T]} \tag{2}$$

*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 $\alpha$ in F or F', $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow Sig(\alpha) = 1$ in F or F' iff $Sig(\alpha) = 1$ in $F\sigma[G / \perp] \vee F'\sigma[G / T]$. 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.*

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 (it is unique). 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.

**Variable Unification Restriction**

We would try to prove if from $\forall X \exists Y\ p(X, Y)$ is provable $\exists Y \forall X\ p(X,Y)$? We will use refutational proving and therefore we will construct a special axiom from the first formula and negation of the second formula: $F_0 : \forall X \exists Y\ p(X, Y)$. $F_1$ ($\neg$query) :$\neg \exists Y \forall X\ p(X, Y)$.

There are 2 trivial and 2 non-trivial combinations how to resolve $F_0$ and $F_1$ (combinations with the same formula as the positive and the negative premise could not lead to refutation since they are consistent): Trivial cases: $R[F_1\&F_1] : \bot \vee T$ and $R[F_0\&F_0] : \bot \vee T$. Both of them lead to T and the atoms are simply unifiable since the variables are the same.

Non-trivial cases:$[F_1\&F_0]$ : no resolution is possible.$Y \in Var_\forall(F_1)$ and $Y \in Var_\exists(F_0)$ can't unify since VUR for (Y,Y) doesn't hold - there is a variable $X \in Sup^*(Qnt(Y))$(over the scope), $X \in Var_\forall(F_0)$, $Sbt(X) = \varnothing$); the case with variable X is identical.

$[F_0\&F_1]$ : no resolution is possible (the same reason as above).

No refutation could be derived from $F_0$ and $F_1$ due to VUR.

Further we would like to prove from $\exists Y \forall X\ p(X, Y)$ is provable $\forall X \exists Y\ p(X,Y)$.

F0 : $\exists Y \forall X\ p(X, Y)$. F1 ($\neg$query) : $\neg \forall X \exists Y p(X, Y)$

In this case we can simply derive a refutation: $R[F_1\&F_0] : \bot \vee \neg T$(refutation)$X \in Var_\forall(F_0)$ and $X \in Var_\exists(F_1)$ can unify since VUR for (X,X) holds - there is no global universal variable over the scope of X in $F_1$; Sbt(X)=X and Sbt(Y)=Y.

Now we entail above presented definitions by introduction of refutational theorem proving with the rule $GR_{FOL}$. We assume standard notions and theorems stated in [Ba01].

**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 $\bot$, T (standard equivalencies for logical constants). A refutational proof of the goal G from the set of formulas (special axioms) $U = \{ A_1, ..., A_m \}$ is a sequence of formulas $F_1, F_2, \ldots, F_n, \bot$, 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. If there is a refutational proof of G from U then we write the $\bot$ is provable U $\cup\{ \neg G \}$ \vdash $\bot$). 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.*

We presented only its essential ideas since the main intent of the article is to present RRTP for Fuzzy Predicate Logic. The non-clausal theorem prover for FOL is already implemented in the form of an experimental application [Ha00], [Ha05].

## 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 [No99]. For the purposes of fuzzy extension the Modus ponens rule was considered as an inspiration [Ha02]. 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), $\nabla$(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

$$a\ /\ A$$

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**

*Inference rule*

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

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

*using which the evaluated formulas* $a_1/ A_1, \ldots, a_n/ A_n$ *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 8 Formal fuzzy theory**

*A formal fuzzy theory T in the language J is a triple*

$$T = \langle LAx, SAx, R \rangle$$

*where* $LAx \subset F_J$ *is a fuzzy set of logical axioms,* $SAx \subset F_J$ *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 \mathrel* $\subset \sim F_J$ *is a finite sequence of evaluated formulas*

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

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

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

*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.*

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

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

The fuzzy modus ponens rule could be formulated:

**Definition 11 Fuzzy modus ponens**

$$r_{MP}: \quad \frac{a / A, \ b / A \Rightarrow B}{a \otimes b / B} \tag{6}$$

*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 ($GR_{FPL}$)**

$$r_{GR}: \quad \frac{a / F[G_1,,\ldots,G_k], \ b / F'[G'_1,\ldots,G'_n]}{a \otimes b / F\sigma[G / \bot] \nabla F'\sigma[G / T]} \tag{7}$$

*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 $\alpha$ in F or F′, $(Sbt(\gamma) = \alpha) \cap \sigma = \emptyset \Rightarrow Sig(\alpha) = 1$ in F or F′ iff $Sig(\alpha) = 1$ in $F\sigma[G / \bot] \vee F'\sigma[G / T]$. F is called positive and F′ is called negative premise, G represents an occurrence of an atom. The expression $F\sigma[G / \bot] \vee F'\sigma[G / T]$ is the resolvent of the 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 simplification rules for $\bot$, 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(α) = 1 for ∀α in F ∈ N ∪¬G formula, every formula in a proof has no free variable and has no quantifier for a variable not occurring in the formula.*

**Lemma 1** ***Soundness of $r_R$***
*The inference $r_R$ rule for FPL based on L is sound i.e. for every truth valuation D,*

$$D(r^{syn}(A_1, ..., A_n)) \geq r^{evl}(D(A_1), ...,D(A_n)) \tag{8}$$

*holds true.*

**Definition 14 Simplification rules for ∇, ⇒**

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

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

**Lemma 2 Provability and refutation degree for $GR_{FPL}$**
*T |-$_a$ A iff a = ∨{Val(w) /w is a refutational proof of A from LAx∪SAx}*
*Proof:* If T |-$_a$ A then a = ∨{Val(w) | w is a proof of A from LAx∪SAx} and for every such a proof of we can construct refutational proof as follows (Val(w) ≤ a): w : = a/ A {a proof A}, 1/ ¬A {a member of a refutational proof}, a ⊗1/ ⊥ {$r_{GR}$}
If a = ∨{Val(w) | w is a refutational proof of A from LAx∪SAx}(Val(w) ≤ a):
w : = $a_0$/ $A_0$ , ..., $a_i$/ $A_i$ , 1/ ¬A ,... , a/ ⊥ , where $A_0$,..., $A_i$ are axioms. There is a proof w′: = $a_0$/ $A_0$ , ..., $a_i$/ $A_i$ , 1/ ¬A ∇A , $a_{i+2}$/ $A_{i+2}$ ∇A , ... , a/ ⊥∇A . All the schemes of the type $A_j$ ∇A , j > i could be simplified by sound simplification rules and the formula ¬A ∇A may be removed. The proof w′′: = $a_0$/ $A_0$ , ..., $a_i$/ $A_i$ , $a_{i+2}$/ $A_{i+2}$ ∇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. ◊

**Theorem 1 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 |-$_a$ A iff T |=$_a$ 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 ⇒B {a proof $w_{A\Rightarrow B}$ }, a ⊗b/ B {$r_{MP}$ }. Then we can replace it by the proof: w : = a/ A {a proof $w_a$ }, b/ A ⇒ B {a proof $w_{A\Rightarrow B}$ }, a ⊗b/ ⊥∇ [T⇒B]{$r_R$ },a ⊗b/ T⇒B{$r_{s\nabla}$, a⊗b/ B{$r_{s\Rightarrow}$ }

**An example**
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.

Common proof members (axioms):

| | | | |
|---|---|---|---|
| 1. 0.8 | / | ∀X[∃Y[child(X, Y)& female(Y)]<br>&∃Y[child(X, Y) & male(Y)] | (happy with parents - 0.8) |

| | | |
|---|---|---|
| | ⇒happy(X)] | |
| 2. 0.5 | / ∀X[toys(X) ⇒happy(X)] | (happy with toys - 0.5) |
| 3. 1 | / child(johana, hashim) | (clear crisp fact) |
| 4. 1 | / child(johana, lucie) | (clear crisp fact) |
| 5.1 | / male(hashim) | (clear crisp fact) |
| 6.1 | / female(lucie) | (clear crisp fact) |
| 7.0.9 | / toys(johana) | (johana has a lot of toys - 0.9) |
| 8.1 | / ¬happy(johana) | (negated goal - is johana happy?) |

Proof 1:

| | | |
|---|---|---|
| 9. 0.9 ⊗0.5 | / ⊥∇[ T⇒happy(johana)] | |
| 9a. 0.4 | / happy(johana) | ($r_{GR}$ on 7.,2., Sbt(X)=johana) |
| 10. 1 ⊗0.4 | / ⊥∇¬T | |
| 10a. 0.4 | / ⊥ | ($r_{GR}$ on 9.,8.) |

(happy(johana) is provable in 0.4)

Proof 2:

| | | |
|---|---|---|
| 9. 0.8 ⊗1 | / [∃Y[child(johana, Y)& female(Y)] &∃Y[child(johana, Y)&male(Y)]⇒⊥]∇¬T | |
| 9a. 0.8 | / ¬[∃Y[child(johana, Y)& female(Y)]& ∃Y[child(johana, Y)& male(Y)]] | ($r_{GR}$ on 1.,8., Sbt(X)=johana) |
| 10. 0.8 ⊗1 | / ¬[[child(johana, lucie)& T]&∃Y[child(johana, Y)&male(Y)]]∇⊥ | |
| 10a. 0.8 | / ¬[child(johana, lucie)&∃Y[child(johana, Y)& male(Y)]] | ($r_{GR}$ on 6.,9., Sbt(Y)=lucie) |
| 11. 0.8 ⊗1 | / ¬[child(johana, lucie) &[child(johana, hashim)&T]]∇⊥ | |
| 11a. 0.8 | / ¬[child(johana, lucie)& child(johana, hashim)] | ($r_{GR}$ on 5.,10., Sbt(Y)=hashim) |
| 12. 0.8 ⊗1 | / ¬[T&child(johana, hashim)]∇⊥ | |
| 12a. 0.8 | / ¬[child(johana, hashim)] | ($r_{GR}$ on 4.,11.) |
| 13. 0.8 ⊗1 | / ¬T∇⊥ | |
| 13a. 0.8 | / ⊥ | ($r_{GR}$ 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.

## 4. Conclusion

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 [Ha05a]. The recent idea of fuzzy description logic is naturally suitable for further extensions with the presented inference rules [Hj05] 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 [Ha05]. This implementation called GERDS (GEneralised Resolution Deductive System) will be extended for usage in fuzzy logic and description logic.

## References

[Ba97] Bachmair, L., Ganzinger, H. A theory of resolution. Technical report: Max-Planck-Institut für Informatik, 1997

[Ba01] Bachmair, L., Ganzinger, H. Resolution theorem proving. In Handbook of Automated Reasoning, MIT Press, 2001

[Ha00] Habiballa, H. Non-clausal resolution - theory and practice. Research report: University of Ostrava, 2000,
http://www.volny.cz/habiballa/files/gerds.pdf

[Ha02] 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

[Ha05] Habiballa, H. Non-clausal Resolution Theorem Prover. Research report, No.64: University of Ostrava, 2005,
http://ac030.osu.cz/irafm/ps/rep64.ps.gz

[Ha05a] 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

[Hj00] Hájek, P. Metamathematics of fuzzy logic. Kluwer Academic Publishers, 2000.

[Hj05] Hájek, P. Making fuzzy description logic more general. Research report: Institute of Computer Science, Czech Academy of Sciences, 2005

[Le95] Lehmke, S. On resolution-based theorem proving in propositional fuzzy logic with bold connectives. University of Dortmund, 1995

[No99] Novák, V., Perfilieva, I., Mockor, J. Mathematical principles of fuzzy logic. Kluwer Academic Publishers, 1999