Magic Sets Method with Fuzzy Logic

Karel Ježek and Martin Zíma

Department of Computer Science and Engineering, University of West Bohemia,
Univerzitní 8, 306 14 Plzeň, Czech Republic
Tel.: +420-19-7491212
Fax.: +420-19-7491213
jezek_ka@kiv.zcu.cz, zima@kiv.zcu.cz

Abstract. This paper describes a method of the efficient query evaluation when uncertainty is involved in a deductive database system. A deductive system enriched with fuzzy logic is able to serve better as a knowledge system. Speeding up its execution makes this system practically useful.

1 Introduction

Deductive database systems make it possible to deduce new facts not contained among the facts of the original (extensional) database. These new facts are derived on the base of deduction rules (intensional database). In case that the facts of the extensional database and/or the rules describing the intensional database are vague, the evaluation with uncertainty has to be used. Uncertain information may arise in databases and knowledge bases in various ways. Imprecise information may be expressed at the attribute value level, at the level of the predicate applicability or at the tuple (fact) level. Only the last two are considered in this paper.

The efficiency of a query evaluation in database systems (including deductive ones) is crucial. Sophisticated query optimization techniques, such as magic sets and counting methods, have existed in the deductive database literature for many years [1][3] but have yet to be studied in detail for fuzzy deductive databases. Introducing uncertainty into a deductive program enforces changes in the program evaluation as well as changes in the common used Magic Sets Method.

The structure of this paper is as follows. The fundamental Magic Sets Method [3] is shortly reminded in Section 2. Section 3 defines used principles of fuzzy logic and the evaluation of deductive programs with fuzzy logic. The goal queries optimization is described in Section 4, where we introduce the extended Magic Sets Method. It enables the efficient evaluation of logic programs when uncertainty is involved in the deduction process. The additional supposed extensions of the Magic Sets Method and some experimental results are summarized in the Conclusions.

2 The Magic Sets Method

Let us shortly mention the principle of the Magic Sets Method. This method transforms the original program into a program called magic program. The magic program is equivalent to (gives the same result as) the original one but its evaluation is usually substantially shorter. Constant arguments of the goal query are utilized for its efficient evaluation.

Note 1. Suppose P denotes a set of rules, D denotes a set of facts belonging to a logical program $\mathcal{P} = P \cup D$.

Definition 1. *Let us have two different logical programs $P_1 \cup D$, $P_2 \cup D$ and a query Q. We say that P_1 and P_2 are* ***equivalent sets of rules*** *if for every possible extensional database D the programs $P_1 \cup D$ and $P_2 \cup D$ produce the same answer to the query Q. We call the two programs equivalent as well.*

The first step of the Magic Sets Method is the adornment of the original program. It excludes all the rules of the program which do not participate in the goal query evaluation. Only derived predicates are adorned for in fact only they have to be computed. The adornment strings become part of the predicate names. The so-called sideways information passing graph is created for each rule together with the adornment. It describes the binding of arguments among individual predicates and consequently the flow of restrictive information among the predicates of the rule.

Definition 2. ***An adornment*** *for an n-ary predicate p is a string a of length n over the alphabet $\{b, f\}$, where b stands for bound and f stands for free. We assume a fixed order of the arguments of the predicate. Intuitively, an adorned occurrence of the predicate, p^a, corresponds to a computation of the predicate with some arguments bound to constants, and the other arguments being free, where the bound arguments are those indicated by the adornment.*

Example 1. Let us suppose there is some intensional predicate `p` having 3 arguments in the head of rule r. In this case the adornment string of length 3 is attached to the predicate `p`. Let us now suppose the string `bff` is attached to the occurrence of the predicate `p(X, Y, Z)` when the rule r is adorned. Then the resulting literal shall be `p_bff(X, Y, Z)`. Its first argument (the variable `X`) is bound and the next two arguments are free.

In the second step the so-called magic program is constructed. This step consists of three parts: initialization, construction of magic rules and construction of modified rules.

Definition 3. ***A magic predicate*** *m_p^a is created as a projection of an original p^a predicate on its bound arguments. The number of b characters in an adornment string (it has to be at least one) indicates the arity of the created magic predicate. The prefix $m\_$ is used to identify the magic predicate.*

Example 2. Let us suppose the body of the adorned rule r^{ad} contains the adorned predicate `p_bfb(X, Y, Z)`. Then we can create the magical version of the predicate `p`. The resulting magic predicate will have two arguments, since the corresponding adorned string has just `b` characters. That's why the resulting literal has the form `m_p_bfb(X, Z)`.

In the course of initialization the magic fact (seed) is created. It contains the constants of the goal query. These constants shall be propagated through the resulting program with the help of magic predicates. The magic predicates are described in magic rules and determine which values shall participate in the query evaluation. Therefore they are consequently used in the bodies of modified original rules to restrict the number of computed tuples. We obtain the modified rules by putting the magic predicates into the bodies of the original rules. Every modified rule can contain a different number of magic predicates in its body.

3 The Fuzzy Logic

When working with the uncertainty model, we can use the fuzzy logic approach. Possible ways of the uncertainty implementation are described in [5][9] in detail. Out of various fuzzy logics, we have investigated Gödel, Łukasiewicz and product fuzzy logic [4][8]. We use the real number from the interval (0, 1] as a confidence factor (CF).

Definition 4. *Let p, q be predicates, $\overline{u}$ and $\overline{v}$ their argument value vectors, $\overline{u} = u_1, u_2, \ldots, u_N$, $\overline{v} = v_1, v_2, \ldots, v_M$, where $u_i, i = 1 \ldots N$ and $v_j, j = 1 \ldots M$ have to be instantiated with constants. The semantics of* ***Gödel*** *fuzzy conjunction "$\wedge_G$", fuzzy disjunction "$\vee_G$" and fuzzy negation "not_G" is as follows:*

$$
\begin{aligned}
CF\left(p\left(\overline{u}\right) \wedge_G q\left(\overline{v}\right)\right) &= \min\left(CF\left(p\left(\overline{u}\right)\right), CF\left(q\left(\overline{v}\right)\right)\right) \\
CF\left(p\left(\overline{u}\right) \vee_G q\left(\overline{v}\right)\right) &= \max\left(CF\left(p\left(\overline{u}\right)\right), CF\left(q\left(\overline{v}\right)\right)\right) \\
CF\left(not_G\ p\left(\overline{u}\right)\right) &= \begin{cases} 1 & if\ \ CF\left(p\left(\overline{u}\right)\right) = 0 \\ 0 & otherwise \end{cases}
\end{aligned}
$$

The following semantics of ***Łukasiewicz*** *fuzzy conjunction "$\wedge_L$", fuzzy disjunction "$\vee_L$" and fuzzy negation "not_L" is as follows:*

$$
\begin{aligned}
CF\left(p\left(\overline{u}\right) \wedge_L q\left(\overline{v}\right)\right) &= \max\left(0, CF\left(p\left(\overline{u}\right)\right) + CF\left(q\left(\overline{v}\right)\right) - 1\right) \\
CF\left(p\left(\overline{u}\right) \vee_L q\left(\overline{v}\right)\right) &= \min\left(1, CF\left(p\left(\overline{u}\right)\right) + CF\left(q\left(\overline{v}\right)\right)\right) \\
CF\left(not_L\ p\left(\overline{u}\right)\right) &= 1 - CF\left(p\left(\overline{u}\right)\right)
\end{aligned}
$$

The last semantics of ***product*** *fuzzy conjunction "$\wedge_P$", fuzzy disjunction "$\vee_P$" and fuzzy negation "not_P" is as follows:*

$$
\begin{aligned}
CF\left(p\left(\overline{u}\right) \wedge_P q\left(\overline{v}\right)\right) &= CF\left(p\left(\overline{u}\right)\right) * CF\left(q\left(\overline{v}\right)\right) \\
CF\left(p\left(\overline{u}\right) \vee_P q\left(\overline{v}\right)\right) &= CF\left(p\left(\overline{u}\right)\right) + CF\left(q\left(\overline{v}\right)\right) - CF\left(p\left(\overline{u}\right)\right) * CF\left(q\left(\overline{v}\right)\right) \\
CF\left(not_P\, p\left(\overline{u}\right)\right) &= \begin{cases} 1 & if\ CF\left(p\left(\overline{u}\right)\right) = 0 \\ 0 & otherwise \end{cases}
\end{aligned}
$$

Note 2. We will use $CF(p)$ instead of $CF(p(\overline{u}))$ for brevity in the equations introduced below.

Proposition 1. *If $p_1, p_2, \ldots, p_n$ are predicates, $CF(p_1), CF(p_2), \ldots, CF(p_n)$ their confidence factors, then Gödel fuzzy conjunction and fuzzy disjunction of these predicates are:*

$$CF\left(\bigwedge_{i=1}^{n}{}_{G}\; p_i\right) = \min\left(CF(p_1), CF(p_2), \ldots, CF(p_n)\right)$$
$$CF\left(\bigvee_{i=1}^{n}{}_{G}\; p_i\right) = \max\left(CF(p_1), CF(p_2), \ldots, CF(p_n)\right)$$

and the Łukasiewicz connectives are introduced as:

$$CF\left(\bigwedge_{i=1}^{n}{}_{L}\; p_i\right) = \max\left(0, \sum_{i=1}^{n} CF(p_i) - n + 1\right)$$
$$CF\left(\bigvee_{i=1}^{n}{}_{L}\; p_i\right) = \min\left(1, \sum_{i=1}^{n} CF(p_i)\right)$$

The last formula represents the semantics of the product fuzzy conjunction:

$$CF\left(\bigwedge_{i=1}^{n}{}_{P}\; p_i\right) = \prod_{i=1}^{n} CF(p_i)$$

Proof. Proof is obvious. $\square$

Note 3. In the case of product fuzzy disjunction it is not possible to derive a simple universal formula as in the case of the other introduced fuzzy logics. The resulting formula should be endless.

Definition 5. *Let a rule r be given with a head predicate p and body literals $L_1, L_2 \ldots, L_m$. Let v be the explicitly assigned confidence factor to r. Then the resulting confidence factor of p w.r.t. r shall be the product of v and the confidence of the rule r body.*

Example 3. In the case of Gödel fuzzy logic, the head predicate p of a rule r has the confidence factor value:

$$CF(p) = CF\left(\bigwedge_{i=1}^{m}{}_{G}\; L_i\right) * v = \min\left(CF(p_1), CF(p_2), \ldots, CF(p_n)\right) * v$$

Definition 6. *Let rules $r_1, r_2, \ldots, r_m$ contain the predicate p in their heads. The resulting confidence factor of the predicate p w.r.t. rules $r_1, r_2, \ldots, r_m$ is given by the respective fuzzy disjunction of the following values:*

- *confidence factor of p w.r.t. r_1,*

- *confidence factor of* p *w.r.t.* r_2,
- $\ldots$,
- *confidence factor of* p *w.r.t.* r_m.

Example 4. Let us suppose the predicate p has received three confidence factors v_1, v_2 and v_3 from three rules. If the Łukasiewicz fuzzy logic is used the resulting confidence factor of p is:

$$CF\left(p\right) = CF\left(\bigvee_{i=1}^{3}{}_{\text{Ł}}\ v_i\right) = \min\left(1, v_1 + v_2 + v_3\right)$$

4 Adaptation of the Magic Sets Method

Introducing uncertainty into a program requires a modified definition of the program equivalency. The extended Magic Sets Method has to produce a magic program which is equivalent to the original one.

Definition 7. *Let* P_1 *and* P_2 *be the different sets of rules.* P_1 *and* P_2 *are equivalent, if for any extensional database* D *the programs* $P_1 \cup D$ *and* $P_2 \cup D$ *evaluated by means of an arbitrary fuzzy logic produce the same answer, including uncertainties of resulting tuples for any given goal query* Q.

The introduction of a program equivalence for an arbitrary fuzzy logic in Definition 7 requires to define the general fuzzy conjunction.

Definition 8. *Let* p, q *and* r *be predicates,* $CF\left(p\right)$, $CF\left(q\right)$ *and* $CF\left(r\right)$ *their respective uncertainties and* κ *is a function of two variables defined the following way:*

1. κ *is commutative*
2. κ *is assocoiative*
3. $\kappa\left(CF\left(p\right), 0\right) = 0$
4. $\kappa\left(CF\left(p\right), 1\right) = CF\left(p\right)$
5. $\max\left(0, CF\left(p\right) + CF\left(q\right) - 1\right) \leq \kappa\left(CF\left(p\right), CF\left(q\right)\right) \leq \min\left(CF\left(p\right), CF\left(q\right)\right)$

Let us we call κ *is a function of arbitrary fuzzy conjunction "*$\wedge_K$*".*

Definition 9. *Let* p_1, p_2, $\ldots$, p_{n-1}, p_n *be predicates and* $CF\left(p_1\right)$, $CF\left(p_2\right)$, $\ldots$, $CF\left(p_{n-1}\right)$, $CF\left(p_n\right)$ *their uncertainties and* $\mathcal{K}_n$ *the function of an arbitrary fuzzy conjunction of* n *variables "*$\bigwedge_K$*". We define "*$\bigwedge_K$*" as follows:*

$$CF\left(\bigwedge_{i=1}^{n}{}_{K}\ p_i\right) = \mathcal{K}_n\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_{n-1}\right), CF\left(p_n\right)\right) =$$
$$= \kappa\left(CF\left(p_1\right), \kappa\left(CF\left(p_2\right), \ldots, \kappa\left(CF\left(p_{n-1}\right), CF\left(p_n\right)\right)\ldots\right)\right)$$

Note 4. We do not define the general properties of fuzzy disjunction, because it is not used in this text.

Theorem 1 (Soundness). *Let r be a rule with uncertainty and m_p_1, m_p_2, $\ldots$, m_p_m be magic predicates. The modified rule r_{mod} shall be created with the aid of inserting of the mentioned magic predicates into the body of the rule r. If all tuples of magic predicates m_p_1, m_p_2, $\ldots$, m_p_m satisfy the condition $CF\left(m\_p_j\right) = 1$, then the rules r and r_{mod} are equivalent (i.e. produce the same tuples including CF coefficients) for any fuzzy logic.*

Proof. The proof shows that rules r and r_{mod} produce the same tuples (inclusive of CF coefficients) if all tuples of the magic predicates m_p_1, m_p_2, $\ldots$, m_p_m satisfy the equality $CF\left(m\_p_j\right) = 1$. Suppose the rules r and r_{mod} have the forms:

$$\begin{aligned}
r:&\ p\left(\overline{X}\right)\ :-\ p_1\left(\overline{X_1}\right),\ p_2\left(\overline{X_2}\right), \ldots, p_n\left(\overline{X_n}\right) \quad \text{CF } v. \\
r_{mod}:&\ p\left(\overline{X}\right)\ :-\ p_1\left(\overline{X_1}\right), \ldots, p_n\left(\overline{X_n}\right), m_p_1\left(\overline{X_1^b}\right), \ldots, m_p_m\left(\overline{X_m^b}\right) \quad \text{CF } v.
\end{aligned}$$

The following two formulas show the evaluation of head predicates of rules r and r_{mod} if the Gödel fuzzy conjunction is used. The double underscored results of both formulas indicate the Theorem validity in the case of Gödel fuzzy logic.

$$CF\left(p\right) = CF\left(\bigwedge_{i=1}^{n}{}_{\mathrm{G}}\ p_i\right) * v = \underline{\underline{\min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) * v}}$$

$$\begin{aligned}
CF\left(p\right) &= CF\left(\left(\bigwedge_{i=1}^{n}{}_{\mathrm{G}}\ p_i\right) \wedge_{\mathrm{G}} \left(\bigwedge_{j=1}^{m}{}_{\mathrm{G}}\ m_p_j\right)\right) * v = \\
&= \min\left(CF\left(p_1\right), \ldots, CF\left(p_n\right), CF\left(m_p_1\right), \ldots, CF\left(m_p_n\right)\right) * v = \\
&= \min\left(CF\left(p_1\right), \ldots, CF\left(p_n\right), 1, \ldots, 1\right)\right) * v = \\
&= \underline{\underline{\min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) * v}}
\end{aligned}$$

If the Łukasiewicz fuzzy conjunction is used the next formulas indicate the CF evaluation. The double underscored results show the Theorem validity in the case of Łukasiewicz fuzzy logic.

$$CF(p) = CF\left(\bigwedge_{i=1}^{n}{}_{\mathrm{Ł}}\ p_i\right) * v = \underline{\underline{\max\left(0, \sum_{i=1}^{n} CF(p_i) - n + 1\right) * v}}$$

$$\begin{aligned}
CF\left(p\right) &= CF\left(\left(\bigwedge_{i=1}^{n}{}_{\mathrm{Ł}}\ p_i\right) \wedge_{\mathrm{Ł}} \left(\bigwedge_{j=1}^{m}{}_{\mathrm{Ł}}\ m_p_j\right)\right) * v = \\
&= \max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) + \sum_{j=1}^{m} CF\left(m_p_j\right) - (n+m) + 1\right) * v =
\end{aligned}$$

$$= \max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) + \sum_{j=1}^{m} 1 - (n+m) + 1\right) * v =$$

$$= \max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) + m - n - m + 1\right) * v =$$

$$= \underline{\underline{\max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) - n + 1\right) * v}}$$

The next two formulas show the evaluation of the head predicates of rules r and r_{mod} if arbitrary fuzzy conjunction (see Definitions 8 and 9) is used. The double underscored results of both formulas indicate the Theorem validity in the case of arbitrary fuzzy logic.

$$CF\left(p\right) = CF\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\ p_i\right) * v = \underline{\underline{\mathcal{K}_n\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) * v}}$$

$$\begin{aligned} CF\left(p\right) = CF&\left(\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\ p_i\right) \wedge_{\mathrm{K}} \left(\bigwedge_{j=1}^{m}{}_{\mathrm{K}}\ m\_p_j\right)\right) * v = \\ &= \mathcal{K}_{n+m}\left(CF\left(p_1\right), \ldots, CF\left(p_n\right), CF\left(m\_p_1\right), \ldots, CF\left(m\_p_m\right)\right) * v = \\ &= \mathcal{K}_{n+m}\left(CF\left(p_1\right), \ldots, CF\left(p_n\right), 1, \ldots, 1\right)\right) * v = \\ &= \kappa\left(\mathcal{K}_n\left(CF\left(p_1\right), \ldots, CF\left(p_n\right)\right), \mathcal{K}_m\left(1, \ldots, 1\right)\right) * v = \\ &= \kappa\left(\mathcal{K}_n\left(CF\left(p_1\right), \ldots, CF\left(p_n\right)\right), 1\right) * v = \\ &= \underline{\underline{\mathcal{K}_n\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) * v}} \end{aligned}$$

The last formula summarizes the result of the whole proof.

$$CF\left(p\right) = CF\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\ p_i\right) * v = CF\left(\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\ p_i\right) \wedge_{\mathrm{K}} \left(\bigwedge_{j=1}^{m}{}_{\mathrm{K}}\ m\_p_j\right)\right) * v$$

$\square$

Theorem 2 (Completeness). *Let r be a rule with uncertainty and m_p_1, m_p_2, …, m_p_m be magic predicates. The modified rule r_{mod} shall be created by means of inserting of the mentioned magic predicates into the body of the rule r. Then there exists such uncertainty (CF coefficient) for every magic predicate that rules r and r_{mod} produce the same tuples including CF coefficients for any fuzzy logic.*

Proof. The proof determines values of CF coefficients for all tuples of magic predicates m_p_1, m_p_2, …, m_p_m when the above introduced fuzzy logics are used. We assume that the rule r with uncertainty has the form:

$$p\left(\overline{X}\right)\ :-\ p_1\left(\overline{X_1}\right),\ p_2\left(\overline{X_2}\right), \ldots, p_n\left(\overline{X_n}\right)\quad \mathrm{CF}\ v.$$

The resulting value of the confidence factor for predicate p is

$$CF\left(p\right) = CF\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\; p_i\right) * v \tag{1}$$

The body of the modified rule contains the predicates p_1, p_2, $\ldots$, p_n and may contain the magic predicates m_p_1, m_p_2, $\ldots$, m_p_m. The resulting value of the confidence factor for the modified rule is

$$CF\left(p\right) = CF\left(\left(\bigwedge_{i=1}^{n}{}_{\mathrm{K}}\; p_i\right) \wedge_{\mathrm{K}} \left(\bigwedge_{j=1}^{m}{}_{\mathrm{K}}\; m\_p_j\right)\right) * v \tag{2}$$

The original program and its corresponding magic program have to be equivalent (including the resulting values of CF coefficients). This implies that the values of equations (1) and (2) must be identical. We reach this equivalence easily for Gödel fuzzy logic where the fuzzy conjunction is expressed as the minimum.

$$\begin{aligned}&\min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) = \\ &= \min\left(CF\left(p_1\right), \ldots, CF\left(p_n\right), CF\left(m\_p_1\right), \ldots, CF\left(m\_p_m\right)\right)\end{aligned}$$

The solution of this equation is the following system of inequalities:

$$\begin{aligned}\min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) &\leq CF\left(m\_p_1\right) \leq 1 \\ \min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) &\leq CF\left(m\_p_2\right) \leq 1 \\ &\ldots \\ \min\left(CF\left(p_1\right), CF\left(p_2\right), \ldots, CF\left(p_n\right)\right) &\leq CF\left(m\_p_m\right) \leq 1\end{aligned}$$

where one solution may be e.g.: $\forall i \in [1..m]$ holds $CF(m\_p_i) = 1$.

We specify possible values of CF coefficients of magic predicates in the body of the modified rule r_{mod} if Łukasiewicz fuzzy logic is used. The resulting CF coefficient value of the original rule head predicate p in the case of Łukasiewicz fuzzy logic has the form:

$$CF\left(p\right) = \max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) - n + 1\right) * v \tag{3}$$

and the resulting CF coefficient value of the modified rule head predicate p in the case of Łukasiewicz fuzzy logic has the form:

$$CF\left(p\right) = \max\left(0, \sum_{i=1}^{n} CF\left(p_i\right) + \sum_{j=1}^{m} CF\left(m\_p_j\right) - n - m + 1\right) * v \tag{4}$$

In this case the resulting values of CF coefficients of equations (3) and (4) have to be identical, too. The solution of identity of these equations is a simple equation the unambiguous solution results from.

$$\sum_{j=1}^{m} CF\,(m\_p_j) = m \quad \Longleftrightarrow \quad \begin{cases} CF\,(m\_p_1) = 1 \\ CF\,(m\_p_2) = 1 \\ \quad \ldots \\ CF\,(m\_p_m) = 1 \end{cases}$$

We assume that κ resp. $\mathcal{K}_n$ is an arbitrary fuzzy conjunction of two resp. n variables (see Definitions 8 and 9). The CF coefficient of the predicate p w.r.t. original rule r has the resulting value:

$$CF\,(p) = \mathcal{K}_n\,(CF\,(p_1)\,, CF\,(p_2)\,, \ldots, CF\,(p_n)) * v \tag{5}$$

and the CF coefficient of the predicate p w.r.t. modified rule r_{mod} of the magic program has the resulting value:

$$CF\,(p) = \mathcal{K}_{n+m}\,(CF\,(p_1)\,, \ldots, CF\,(p_n)\,, CF\,(m\_p_1)\,, \ldots, CF\,(m\_p_m)) * v \tag{6}$$

CF coefficients produced by the equations (5) and (6) must have identical values.

$$\begin{aligned} &\mathcal{K}_n\,(CF\,(p_1)\,, CF\,(p_2)\,, \ldots, CF\,(p_n)) = \\ &= \mathcal{K}_{n+m}\,(CF\,(p_1)\,, \ldots, CF\,(p_n)\,, CF\,(m\_p_1)\,, \ldots, CF\,(m\_p_m)) = \\ &= \kappa\,(\mathcal{K}_n\,(CF\,(p_1)\,, \ldots, CF\,(p_n))\,, \mathcal{K}_m\,(CF\,(m\_p_1)\,, \ldots, CF\,(m\_p_m))) \end{aligned}$$

If we utilize the 4th property of an arbitrary fuzzy conjunction (see Definition 8) we obtain a simple equation the unambiguous solution results from.

$$\mathcal{K}_m\,(CF\,(m\_p_1)\,, \ldots, CF\,(m\_p_m)) = 1 \quad \Longleftrightarrow \quad \begin{cases} CF\,(m\_p_1) = 1 \\ CF\,(m\_p_2) = 1 \\ \quad \ldots \\ CF\,(m\_p_m) = 1 \end{cases}$$

□

Now we can look at the consequences of the proofs of Theorems 1 and 2: When we wish to apply the Magic Sets Method together with a fuzzy logic we must set the CF values to 1 for all tuples of magic predicates. This requirement can be easily achieved if we use classic two-valued logic for evaluation of the magic rules, i.e. we do not respect CF coefficients of the non-magic predicates from the bodies of magic rules.

Theorem 3. *Let $\mathcal{P}$ be a logic program with uncertainty and $\mathcal{P}^m$ be its corresponding magic program. If we use classic two-valued logic for the magic rules evaluation and any fuzzy logic for modified rules evaluation then programs $\mathcal{P}$ and $\mathcal{P}^m$ are equivalent according to Definition 7.*

Proof. The proof results from the proofs of Theorems 1 and 2. □

5 Conclusions

The Magic Sets Method application in the environment of fuzzy logic programs was described. A correct processing of fuzzy operation resulting in a magic program which is equivalent to the original one was entered up. We focus on fuzzy conjunction as it has direct influence on the evaluation of modified rules.

The theory introduced in the paper was experimentally verified on the deductive database system EDD [7] and applied on the database of bank clients. The logic program appraises clients of banks according to several aspects. These aspects are described by the help of rules and their CFs. The program separates clients to safe and unsafe ones. The original program consists of 62 rules and it takes more than 30 GB of disk memory. Its running time might exceed one year. The resulting magic program consists of 125 rules and during the evaluation the program generates 500 KB of data into the database. The running time of this magic program was less than 2 minutes.

The next challenging task of the Magic Sets Method extensions concerns the negation. A program with negation has to be stratified [6]. A non-stratified program can generate wrong results or its evaluation is infinite. The Magic Sets Method applied to a stratified program does not always produce the stratified resulting magic program. The general conditions for producing stratified or non-stratified magic programs were not yet formulated.

Acknowledgement. This work was supported by the Research Project of Czech Ministry of Education No. MSM235200005 “Information Systems and Technology”.

References

1. Bancilhon, F., Maier, D., Sagiv, Y., Ullman, J.D.: Magic Sets and Other Strange Ways to Implement Logic Programs. Proc. 5th ACM Symp. Principles of Database Systems (1986) 1–15
2. Berka, P.: PKDD ’99 Discovery Challenge: Guide to the Financial Set. http://lisp.vse.cz/pkdd99/Challenge/berka.htm
3. Beeri, C., Ramakrishnan,R.: On the Power of Magic. Proc. 6th ACM Symp. Principles of Database Systems (1987) 269–283
4. Hájek, P., Godo, L.: Deductive Systems of Fuzzy Logic. Tatra Mountains Mathematical Publications **13** (1997) 37–68
5. Lakshmanan, L.V.S., Shiri, N.: A Parametric Approach to Deductive Database with Uncertainty. IEEE Transactions on Knowledge and Data Engineering **13** (2001) 554–570
6. Meskesh, M.: Extending the Stratification Approach for Deductive Database Systems. PhD Thesis, Technischen Hochschule, Aachen (1996)
7. Toncar, V.: Datalog Extensions and the Use of Datalog for Large Data Analysis. PhD Thesis, University of West Bohemia, Pilsen (2000)
8. Vojtáš, P.: Uncertain Reasoning with Floating Connectives. Proc. 3rd International Workshop on Artificial Intelligence and Technics (1996) 31–40
9. Wagner, G.: Foundations of Knowledge Systems with Applications to Databases and Agents. Kluwer Academic Publishers (1998)