Annales Univ. Sci. Budapest., Sect. Comp. 15 (1995) 223-231

FIXPONT QUERY IN FUZZY DATALOG PROGRAMS

Á. Achs (Pécs, Hungary)
A. Kiss (Budapest, Hungary)

Abstract. In this paper we will discuss the concept of fuzzy Datalog program. It will be shown that for function- and negationfree fDATALOG program the fixpoint is the least model.

1. Introduction

In the knowledge-base systems there are given some facts which represent certain knowledge and some rules which in general mean that certain kinds of information imply other kinds of information. In classical deductive database theory ([1], [6]) the Datalog-like data model is widely spread. A classical Datalog program is a set of Horn clauses, i.e. a set of the clauses with at most one positive literal ($q \leftarrow q_1, ..., q_n$). The most general type of the programs allows the use of functional symbols and negation as well.

The meaning of a Datalog-like program is the least (if it exists) or a minimal model which contains the facts and satisfies the rules. This model is generally computed by a fixpoint algorithm.

The aim of this paper, which is partially a further developing of [3], is to give a possible extension of Datalog-like languages to fuzzy relational databases using lower bounds of degrees of uncertainty in facts and rules. We get a method for fixpoint query.

This research was supported by the projects OTKA-2149, OTKA-4281 of the National Research Foundation of Hungary.

2. The concept of fuzzy Datalog program

To define the idea of fuzzy Datalog program (fDATALOG) we need some basic concepts.

A *term* is a variable, a constant or a complex term of the form $f(t_1, ..., t_n)$, where f is a function symbol and $t_1, ..., t_n$ are terms. An *atom* is a formula of the form $p(\underline{t})$, where p is a predicate symbol of a finite arity (say n) and $\underline{t}$ is a sequence of terms of length n (arguments). A *literal* is either an atom (positive literal) or its negation (negative literal).

A term, atom, literal is *ground* if it is free of variables.

An *implication operator* is a mapping of the form

$$I(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ f(x, y) & \text{otherwise} \end{cases}$$

where $x, y \in [0, 1]$ and $0 \leq f(x, y) \leq 1$.

Let D be a set. The *fuzzy set* F in D is a function $F : D \rightarrow [0, 1]$. Denote $\mathcal{F}(D)$ the set of all fuzzy sets in D. So $F \in \mathcal{F}(D)$. If $F(d) = 0$ then d does not belong to F.

$$F \cup G(d) \stackrel{def}{=} \max(F(d), G(d)),$$

$$F \cap G(d) \stackrel{def}{=} \min(F(d), G(d)).$$

We can define an ordering relation : $F \leq G$ iff $F(d) \leq G(d)$ $\forall d \in D$. The support of fuzzy set F is a classical set

$$Supp(F) = \{\ d\ |F(d) \neq 0\}.$$

It can be seen that $(\mathcal{F}(D), \leq)$ is a complete lattice. The top element of lattice is $U : D \rightarrow [0, 1] : U(d) = 1\ \forall d \in D$. The bottom element is: $\emptyset : D \rightarrow [0, 1] : \emptyset(d) = 0\ \forall d \in D$.

It is usual to explicitly write down the fuzzy sets as follows:

$$F = \bigcup_{d \in D} (d, \alpha_d),$$

where $(d, \alpha_d) \in D \times [0, 1]$. It is also usual to omit from F the (d, α_d) pairs where $\alpha_d = 0$, or enlarge $Supp(F)$ with $(d, 0)$ pairs, where $d \in D$ but $d \notin Supp(F)$.

Definition 1. An fDATALOG rule is a triplet $(r; I; \beta)$, where r is a formula of the form

$$Q \leftarrow Q_1, ..., Q_n \quad (n \geq 0)$$

where Q is an atom (head of the rule), $Q_1, ..., Q_n$ are literals (body of the rule); I is an implication operator and $\beta \in (0, 1]$ (level of the rule). An fDATALOG rule is safe if

- all variables which occur in the head also occur in the body;

- all variables occuring in a negative literal also occur in a positive literal.

An fDATALOG program is a finite set of safe fDATALOG rules. The rules in the form $(A \leftarrow; I; \beta)$, where A is a ground atom, are facts.

The *Herbrand universe* of a program P (denoted by H_P) is the set of all possible ground terms constructed by using constants and function symbols occurring in P. The *Herbrand base* of P (B_P) is the set of all possible ground atoms whose predicate symbols occur in P and whose arguments are elements of H_P. A *ground instance* of a rule $(r; I; \beta)$ in P is a rule obtained from r by replacing every variable X in r by $\Phi(X)$ where Φ is a mapping from all variables occurring in r to H_P. The set of all ground instances of $(r; I; \beta)$ are denoted by $(ground(r); I; \beta)$. The ground instance of P is

$$ground(P) = \cup_{(r;I;\beta)\in P} \ (ground(r); I; \beta).$$

Definition 2. An interpretation of a program P is a fuzzy set of B_P :

$$N_P \in \mathcal{F}(B_P)$$

Let

$$\alpha_{A_1 \wedge ... \wedge A_n} \stackrel{def}{=} \min(\alpha_{A_1}, ..., \alpha_{A_n}),$$

$$\alpha_{\neg A} \stackrel{def}{=} 1 - \alpha_A.$$

Definition 3. An interpretation is a model of P if for each $(ground(r); I; \beta)$ $\in ground(P)$, $ground(r) = A \leftarrow A_1, ..., A_n$

$$I(\alpha_{A_1 \wedge ... \wedge A_n}, \alpha_A) \geq \beta.$$

A model M is least if for any model N, $M \leq N$. A model M is minimal if there is not any model N, that $N < M$.

To be short we sometime denote $\alpha_{A_1 \wedge ... \wedge A_n}$ by α_{body} and α_A by α_{head}.

3. The fixpoint query

Definition 4. The consequence transformation $T : \mathcal{F}(B_P) \rightarrow \mathcal{F}(B_P)$ is defined as

$$T(X) = \{\cup\{(A, \alpha_A)\} | (A \leftarrow A_1, ..., A_n; I; \beta) \in ground(P),$$

$$(|A_i|, \alpha_{A_i}) \in X \text{ for each } 1 \leq i \leq n,$$

$$\alpha_A = \max(0, \min\{\gamma | I(\alpha_{body}, \gamma) \geq \beta\})\} \cup X.$$

$|A|$ denotes $p(\underline{c})$ either $A = p(\underline{c})$ or $A = \neg p(\underline{c})$ where p is a predicate symbol with arity k and $\underline{c}$ is a list of k ground terms.

Let

$$T_0 = \{\cup\{(A, \alpha_A)\} | (A \leftarrow; I; \beta) \in ground(P), \alpha_A = \min\{\gamma | I(1, \gamma) \geq \beta\}\} \cup$$

$$\cup\{(A, 0) | \exists (B \leftarrow ...\neg A...; I; \beta) \in ground(P)\}.$$

Let

$$T_1 = T(T_0)$$

$$T_n = T(T_{n-1})$$

$$T_\delta = \text{least upper bound}\{T_\gamma | \gamma < \delta\} \text{ if } \delta \text{ is a limit ordinal.}$$

Proposition 1. *T has a fixpoint, i.e. there exists $X \in \mathcal{F}(B_P) : T(X) = X$. If P is positive, then this is the least fixpoint. (That is for any $Y = T(Y) : X \leq Y$.)*

Proof. Since T is inflationary and $\mathcal{F}(B_P)$ is a complete lattice, therefore it has an inflationary fixpoint [1, 2]. If P is positive, then T is monotone, therefore the proposition is true (due to [4]).

Theorem 1. *T_∞ is a model of P.*

Proof. In *ground*(P) there are rules in the next forms:

a) $(A \leftarrow; I; \beta)$;

b) $(A \leftarrow A_1, ..., A_n; I; \beta); (A, \alpha_A) \in T_\infty$ and $(|A_i|, \alpha_{A_i}) \in T_\infty$, $1 \leq i \leq n$;

c) $(A \leftarrow A_1, ..., A_n; I; \beta); \exists i : (|A_i|, \alpha_{A_i}) \notin T_\infty$.

In the cases a), b) because of the construction of α_A $I(\alpha_{body}, \alpha_A) \geq \beta$. In the case c) because of the construction of T_0 A_i is not negative, so $\alpha_{A_i} = 0$, therefore $\alpha_{body} = 0$, so $I(\alpha_{body}, \alpha_A) = 1 \geq \beta$. So T_∞ is a model.

It would be practical to choose the implication operator so that in case of facts $(A \leftarrow; I; \beta)$ the calculated α_A and β be equal.

Proposition 2. *If for the implication operator*

$$I(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ f(x, y) & \text{otherwise} \end{cases}$$

$f(1, y) \equiv y$ holds, then in case of facts $(A \leftarrow; I; \beta) \in ground(P)$ for α_A calculated by consequence transformation $\alpha_A = \beta$.

Proof.

$$I(1, \alpha_A) = \begin{cases} 1, & \text{if } \alpha_A = 1, \\ f(1, \alpha_A) & \text{if } \alpha_A < 1. \end{cases}$$

Because of the construction $I(1, \alpha_A) \geq \beta$, so $f(1, \alpha_A) \geq \beta$. As $f(1, \alpha_A) = \alpha_A$, therefore $\alpha_A \geq \beta$, but α_A is minimal, so $\alpha_A = \beta$.

Proposition 3. The implication operators

$$I_1(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ y & \text{otherwise}, \end{cases}$$

$$I_2(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ 1 - (x - y) & \text{otherwise}, \end{cases}$$

$$I_3(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ y/x & \text{otherwise} \end{cases}$$

satisfy the condition of Proposition 3.

Proposition 4. The implication operator

$$I_4(x, y) = \begin{cases} 1, & \text{if } x \leq y, \\ 0 & \text{otherwise} \end{cases}$$

does not satisfy the condition of Proposition 3. In the case of $(A \leftarrow; I_4; \beta) \in$ $\in ground(P)$ $\alpha_A = 1$.

Proposition 5. If in every rule of P the implication operator is I_4 then we derive the ordinary Datalog (with negation) with inflationary semantics.

The concept of inflationary semantics is detailed in [1].

Theorem 2. *If the program has not any negation then T_∞ is the least model.*

Proof. In case of ordinary Datalog, the least fixpoint is the least model. There may be trouble with the degrees, as it can occur that in a later step we get a smaller α_A, but according to definition the greater is valid. But it could happen only if there was any negation.

Theorem 3. *For function- and negationfree fDATALOG program P there exists some n such that $T_n = T_{n+1} = \ldots = T_\infty$.*

Proof. P is finite and since $\alpha_{head} \leq \alpha_{body}$, therefore the degrees cannot increase infinitely.

Proposition 6. *If the program has any negation then T_∞ is not always a minimal model.*

Proof. Let us see the following examples:

1.

$$r(a) \leftarrow; I_4; \beta_1$$

$$p(x) \leftarrow r(x), \neg q(x); I_4; \beta_2$$

$$q(x) \leftarrow r(x), \neg p(x); I_4; \beta_3$$

Then $T_\infty = \{(r(a),1);(p(a),1);(q(a),1)\}$ It is a model, but it is not a minimal one because

$$M_1 = \{(r(a),1);(p(a),1)\} \quad \text{or} \quad M_2 = \{(r(a),1);(q(a),1)\}$$

is also a model.

2.

$$r(a) \leftarrow; I_1; 0.8$$

$$p(x) \leftarrow r(x), \neg q(x); I_1; 0.6$$

$$q(x) \leftarrow r(x); I_1; 0.5$$

Then $T_\infty = \{(r(a),0.8);(p(a),0.6);(q(a),0.5)\}$. It is a model, but it is not a minimal one because

$$M = \{(r(a), 0.8); (p(a), 0.5); (q(a), 0.5)\}$$

is also a model.

4. Examples

Example 1.
Let us consider the rules :

$$p(a) \leftarrow; I_1; \beta_1$$
$$r(b) \leftarrow; I_2; \beta_2$$
$$q(x, y) \leftarrow p(x), r(y); I_1; \beta_3$$
$$q(x, y) \leftarrow q(y, x); I_4; \beta_4$$
$$s(x) \leftarrow q(x, y); I_3; \beta_5$$

Then

$$T_\infty = \{(p(a), \alpha_1), (r(b), \alpha_2), (q(a, b), \alpha_3), (q(b, a), \alpha_4), (s(a), \alpha_5), (s(b), \alpha_6)\}.$$

Let $\beta_1 = 0.8, \beta_2 = 0.6, \beta_3 = 0.7, \beta_4 = 0.9, \beta_5 = 0.7$. Now we can calculate α_i, $1 \leq i \leq 6$

$$\alpha_1 = \min(\alpha_{body}, \beta_1) = \beta_1 = 0.8$$
$$\alpha_2 = \min(\alpha_{body}, \alpha_{body} + \beta_2 - 1) = \beta_2 = 0.6$$
$$\alpha_3 = \min(\alpha_{body}, \beta_3) = min(min(\alpha_1, \alpha_2), \beta_3) = 0.6$$
$$\alpha_4 = \alpha_{body} = 0.6$$
$$\alpha_5 = \beta_5 \cdot \alpha_{body} = 0.7 \cdot 0.6 = 0.42$$
$$\alpha_6 = \beta_5 \cdot \alpha_{body} = 0.7 \cdot 0.6 = 0.42$$

So

$$T_\infty = \{(p(a), 0.8), (r(b), 0.6), (q(a, b), 0.6), (q(b, a), 0.6), (s(a), 0.42), (s(b), 0.42)\}$$

Obviously T_∞ is a model.

Example 2.

$$r(a) \leftarrow; I_1; \beta_1$$
$$p(x) \leftarrow r(x); I_1; \beta_2$$
$$p(x) \leftarrow \neg p(x); I_1; \beta_3$$
$$q(x) \leftarrow s(x); I_1; \beta_4$$

Let $\beta_1 = 0.9, \beta_2 = 0.7, \beta_3 = 0.3, \beta_4 = 0.6$. Then

$$T_0 = \{(r(a), 0.9), (p(a), 0)\}$$
$$T_1 = \{(r(a), 0.9), (p(a), 0.7)\}$$
$$T_\infty = T_2 = T_1$$

Example 3.

$$r(a) \leftarrow; I_1; \beta_1$$
$$p(a) \leftarrow; I_1; \beta_2$$
$$p(x) \leftarrow r(x), \neg q(x); I_1; \beta_3$$
$$q(x) \leftarrow r(x), \neg p(x); I_2; \beta_4$$

Let $\beta_1 = 0.8, \beta_2 = 0.7, \beta_3 = 0.9, \beta_4 = 0.8$. Then

$$T_0 = \{(r(a), 0.8), (p(a), 0.7), (q(a), 0)\}$$
$$T_1 = \{(r(a), 0.8), (p(a), 0.7), (q(a), 0.1)\}$$
$$T_2 = \{(r(a), 0.8), (p(a), 0.8), (q(a), 0.1)\}$$

$T_3 = T_2$, so $T_\infty = T_2$.

References

[1] **Ceri S., Gottlob G. and Tanca L.,** *Logic Programming and Databases*, Springer Verlag, Berlin, 1990.

[2] **Gurevich Yu. and Shelah S.,** Fixed-point extensions of first-order logic, *IEEE Symp. on FOCS*, 1985, 346-353.

[3] **Kiss A.**, On the least models of fuzzy Datalog programs, *Proc. International Conference on Information Processing and Management of Uncertainty in Knowledge-based system, Mallorca*, 1992, 465-471.

[4] **Lloyd J.W.**, *Foundations of database and knowledge-base systems*, Springer, Berlin, 1987.

[5] **Novák V.**, *Fuzzy sets and their applications*, Adam Hilger, Bristol and Philadelphia, 1987.

[6] **Ullman J.D.**, *Principles of database and knowledge-base systems*, Computer Science Press, Rockville, 1988.

(Received June 22, 1994)

Á. Achs
Inst.of Mathematics and Informatics
Pollack Mihály College
Janus Pannonius University
Rókus u. 2.
H-7624 Pécs, Hungary

A. Kiss
Dept.of General Computer Science
Eötvös Loránd University
VIII. Múzeum krt. 6-8
H-1088 Budapest, Hungary