Reports of Enlarged Session of the
Seminar of I. Vekua Institute
of Applied Mathematics
Volume 23, 2009

DISCRETE FUZZY PROLOG

Rogava I.

Abstract. Discrete Fuzzy Prolog is an optimal extension of pure Logic Programming. It is constructed by using the syntax and semantics of Logic Programming but some part of semantics also consists of concepts of Fuzzy Logic. In this paper there is started to describe and formalize these fuzzy concepts through mathematics that makes programming language able to be extensible and capable which in itself transforms the language more applicable. Here are presented some mathematical definitions and notations to describe the syntax and semantics of Discrete Fuzzy Prolog. One theorem and an auxiliary lemma are introduced that are about the interpretations and models of any particular program of Discrete Fuzzy Prolog.

Keywords and phrases: Discrete-interval, aggregation operator, fuzzy program, Herbrand universe, Herbrand base.

AMS subject classification (2000): 03B52; 90C70.

1. Introduction. This paper is about Discrete Fuzzy Prolog. Discrete Fuzzy Prolog is an extension of Logic Programming using Fuzzy Logic. Our aim is to see and analyze some important definitions and theorems of Discrete Fuzzy Prolog through mathematics and describe them. There will be introduced "mathematical versions" of syntax and semantics. One lemma and one theorem which are presented here are the main results of our work. Paper is divided into four paragraphs.

2. Some Useful Definitions.

Definition 2.1 (Interval Inclusion): Given two intervals $I_1 = [a, b]$ and $I_2 = [c, d]$. We will say that I_1 is sub-interval of $I_2, I_1 \subseteq_{II} I_2$ iff $c \leq a$ and $b \leq d$.

Definition 2.2 (Borel Inclusion): Given two unions of intervals $U_1 = I_1 \cup \ldots I_n$ and $U_2 = J_1 \cup \ldots J_m$; $U_1 \subseteq_{BI} U_2$ iff $\forall I_\imath \in U_1 \imath = 1, \ldots, n \exists J_\jmath \in U_2 \jmath = 1, \ldots, m$ s.t. $I_\imath \subseteq_{II} J_\jmath$.

Definition 2.3 (Discrete-Interval): Given a sub-interval $[X_1, X_n]$ of $[0, 1]$. A discrete-interval $[X_1, X_n]_\delta$ is the set $X_1, \ldots, X_n$ s.t. $X_{\imath+1} = X_\imath + \delta, \imath = 1, \ldots, n$.

δ has the most important role to define a discrete-interval. Thereby $[X_1, X_n]$ is a sub-interval of $[0, 1]$ it is clear that $0 \leq X_1 \leq \ldots \leq X_n \leq 1$ and $0 \leq \delta \leq 1$. In addition we can also formalize a value of δ for any particular discrete-interval $[X_1, X_n]_\delta$. It is trivial to see that $\delta = \frac{X_n - X_1}{n-1}$.

Definition 2.4 (Aggregation Operator): Given a function $f : [0, 1]^n \to [0, 1]$. f is called an aggregation operator if the following holds:

(i) $f(0, \ldots, 0) = 0$ and $f(1, \ldots, 1) = 1$.

(ii) f is monotonic and continuous.

Definition 2.5 (Discrete-Aggregation Operator): f is a discrete-aggregation-operator if it has all the properties of aggregation-operator except continuity.

Definition 2.6 (Discrete-Interval-Aggregation Operator): F is called a discrete-interval-aggregation operator if $F : \varepsilon_\delta([0,1])^n \to \varepsilon_\delta([0,1])$ and $F([l_1,u_1]_\delta,\ldots,[l_n,u_n]_\delta) = [f(l_1,\ldots,l_n), f(u_1,\ldots,u_n)]_\delta$, where $\varepsilon_\delta([0,1])$ is a family of all the closed intervals of the real numbers in $[0,1]$ and f is discrete-aggregation operator ($f : [0,1]^n \to [0,1]$).

Definition 2.7 (Discrete-Union-Aggregation-Operator): ψ is called a discrete-union-aggregation operator if $\psi : B([0,1])^n \to B([0,1])$ and $\psi(B_1,\ldots,B_n) = \cup\{F(\varepsilon_{\delta_1},\ldots,\varepsilon_{\delta_n})|\varepsilon_{\delta_i} \in B_i\}$, where $B([0,1])$ is a countable union of sub-intervals of $[0,1]$ and F is a discrete-interval-aggregation-operator($F : \varepsilon_\delta([0,1])^n \to \varepsilon_\delta([0,1])$).

3. Syntax and Semantics.

Definition 3.1 (Fuzzy Fact): $A \to v$ is called a fuzzy fact if the following holds:

(i) A is an atom;

(ii) v is truth value, element of $B([0,1])$.

Definition 3.2 (Fuzzy Clause): $A \to_F B_1,\ldots,B_n$ is called a fuzzy clause if the following holds:

(i) $A, B_1,\ldots,B_n$ are the atoms;

(ii) F is an interval-aggregation operator of truth values in $B([0,1])$.

Definition 3.3 (Fuzzy Query): A tuple $v \to A?$ is called a fuzzy query if the following holds:

(i) A is an atom;

(ii) v is some element(truth value) in $B([0,1])$.

A fuzzy program is a finite set containing fuzzy facts and fuzzy clauses. To be more comfortable in describing of discrete fuzzy prolog we will use some useful notions of first order logic.

Definition 3.4 (Terms): A set of terms T is inductively defined as following:

(i) $V \subseteq T$, variables are terms;

(ii) $C \subseteq T$, constant symbols are terms;

(iii) If $t_1,\ldots,t_n \in T$ and f is a function symbol then $f(t_1,\ldots,t_n) \in T$.

Definition 3.5 (Signature): A set Σ is called a signature if it is defined as following:

(i) Function symbols are included there.

(ii) Predicate symbols are included there.

Remark: Numbers are considered as constants and constants are considered as zeroary functions, so numbers are also zeroary functions and they are members of Σ as function symbols.

Definition 3.6 (Herbrand Universe): Let CS be a set of all constant symbols and FS be a set of all function symbols; then *Herbrand Universe* H is defined as following:

$$H_0 = CS$$
$$\vdots$$
$$H_\imath = H_{\imath-1} \cup \{f(t_1, \ldots, t_n) | f \in FS,\ t_1, \ldots, t_n \in H_{\imath-1},\ n \in N\};$$

$H = \bigcup_{i=0}^{\infty} H_\imath$ is called *Herbrand universe.*

Herbrand Universe is also known as set of all ground terms.

Definition 3.7 (Herbrand Base): Let PS be a set of all predicate symbols; then *Herbrand Base* $H(B)$ is a set of all ground atoms which can be formalized in the following way: $H(B) = \{P(t_1, \ldots, t_n) | P \in PS,\ t_1, \ldots, t_n \in H,\ n \in N\}$.

Definition 3.8 (Fuzzy Interpretation): $I = (B_I, V_I)$ is a fuzzy interpretation if the following holds:

(i) B_I is a subset of Herbrand base$(B_I \subseteq H(B))$;

(ii) V_I is a mapping that assigns truth values in $B([0, 1])$ to elements of B_I.

Definition 3.9 (Interpretation Inclusion): Given two interpretations $I = (B_I, V_I)$ and $I = (B_J, V_J)$; $I \subseteq_{InIn} J$ iff:

(i) $B_I \subseteq B_J$;

(ii) $\forall B \in B_I\ V_I(B) \subseteq V_J(B)$.

Definition 3.10 (Valuation): σ is called *a valuation* of an atom A if it assigns the ground terms to variables of A.

Definition 3.11 (Model): An interpretation $I = (B, V)$ is a model for a fuzzy fact $A \to v$ if:

(i) $\forall \sigma\ \sigma(A) \in B$;

(ii) $v \in_{BI} V(\sigma(A))$.

An interpretation $I = (B, V)$ is *a model* for a fuzzy clause $A \to_F B_1, \ldots, B_n$ if the following holds:

(i) $\forall \sigma \sigma(A) \in B$;

(ii) $v \in_{BI} V(\sigma(A))$, where $v = \psi(V(\sigma(B_1)), \ldots, V(\sigma(B_n)))$ and ψ is discrete-union-aggregation operator.

An interpretation $I = (B, V)$ is a model for a fuzzy program P if I is a model for all fuzzy facts and fuzzy clauses including in P.

Definition 3.12 (Least Model): *A least model* for a fuzzy program P is an intersection of all models of P. Let take M as a set of all models of P, then $lm(P) = \bigcap I_i$, $I_i \in M$.

Lemma 1. *Assume $I = (B_I, V_I)$ is a model for a fuzzy program P, then any sub-interpretation $J = (B_J, V_J)$, $(J \subseteq_{InIn} I)$ of I is also model for P.*

Proof.

(i) For any valuation σ and for any fuzzy fact $A \to v$ in P,

- if $\sigma \in B_J$ then $\sigma \in B_I$;
- if $v \subseteq_{BI} V_J(\sigma(A))$ then $v \subseteq_{BI} V_I(\sigma(A))$.

So J is a model for all fuzzy facts of P.

(ii) For any valuation σ and for any fuzzy clause $A \to_F B_1, \ldots, B_n$ in P,

- if $\sigma \in B_J$ then $\sigma \in B_I$;
- if $v = \psi(V(\sigma(B_1)), \ldots, V(\sigma(B_n)))$ and $v \subseteq_{BI} V_J(\sigma(A))$ then $v \subseteq_{BI} V_I(\sigma(A))$.

So J is a model for all fuzzy clauses of P.

(iii) (i) and (ii) imply that J is a model for the given program P. □

Actually Lemma 1 is direct consequence of definition of interval inclusion but now it will be easier to prove the following theorem.

Theorem 1. *Given a fuzzy program P and its two models $I = (B_I, V_I)$ and $J = (B_J, V_J)$; then*

(i) $I \cap J$ is a model for P;

(ii) $I \cup J$ is a model for P;

(iii) $I \setminus J$ is a model for P.

Proof.

(i) $I \cap J$ is a sub-interpretation of I and J. From the Lemma1 $I \cap J$ is a model for P as a sub-interpretation of a model of P.

(ii) Let $I \cup J$ be $M = (B_M, VM)$;

  (a) For any valuation σ and for any fuzzy fact $A \to v$ in P:
  - if $\sigma(A) \in B_I$ or $\sigma(A) \in B_J$ then $\sigma(A) \in B_I \cup B_J = B_M$;
  - if $v \subseteq_{BI} V_I(\sigma(A))$ or $v \subseteq_{BI} V_J(\sigma(A))$ then $v \subseteq_{BI} V_I(\sigma(A)) \cup V_J(\sigma(A)) = V_M(\sigma(A))$.

  (b) For any valuation σ and for any fuzzy clause $A \to_F B_1, \ldots, B_n$ in P:

- if $\sigma(A) \in B_I$ or $\sigma(A) \in B_J$ then $\sigma(A) \in B_I \cup B_J = B_M$;
- if $v = \psi((V_M(\sigma(B_1)), \ldots, V_M(\sigma(B_n)))$ and $v \subseteq_{BI} V_I(\sigma(A))$ or $v \subseteq_{BI} V_J(\sigma(A))$ then $v \subseteq_{BI} V_I(\sigma(A)) \cup V_J(\sigma(A)) = V_M(\sigma(A))$.

So, M is a model for P.

(iii) It is a trivial case because $I \setminus J$ is a sub-interpretation of I and $J \setminus I$ is a sub-interpretation of J. □

4. Conclusion. An important aim and intention of this short essay is to present and describe the branch and one of the most prevalent component of computational science - Discrete Fuzzy Prolog using mathematics. I have used rules of elementary Set Theory and definitions of First Order Logic. Here are represented very general approaches which give me more ideas for the future and stimulus to work on these subjects. I think nothing is here new and very special for the experts of Discrete Fuzzy Prolog but for me it is beginning and I hope it is short essay which will be extensible and increasable by me.

REFERENCES

1. Munoz-Hernandez S., Vaucheret C. Incomplete Fuzzy Information in Prolog. *In Proceedings of the IEEE International Conference on Fuzzy Systems, Vancouver, Canada,* 2006.

2. Munoz-Hernandez S., Gomez-Perez J.M. Solving collaborative fuzzy agents problems with CLP(FD). *In Manuel Hermenegildo and Daniel Cabeza, editors, International Symposium on Practical Aspects of Declarative Languages, PADL **3350** of LNCS (2005), 187-202. Long Beach, CA (USA)* (2005), *Springer-Verlag.*

Received 18.05.2009; revised 10.10.2009; accepted 12.11.2009.

Author's address:

I. Rogava
Technical University of Madrid (UPM)
Campus de Montegancedo
28660 Boadilla del Monte
Madrid, Spain.
E-mail: ir55@drexel.edu