

Kostadin D. Kratchanov; Ivan N. Stanev; Desislava S. Sabotinova
A fuzzy computational model implementation

Kybernetika, Vol. 28 (1992), No. Suppl, 97--102

Persistent URL: <http://dml.cz/dmlcz/124204>

Terms of use:

© Institute of Information Theory and Automation AS CR, 1992

Institute of Mathematics of the Academy of Sciences of the Czech Republic provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This paper has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library*
<http://project.dml.cz>

A FUZZY COMPUTATIONAL MODEL IMPLEMENTATION

KOSTADIN D. KRATCHANOV, IVAN N. STANEV AND DESISLAVA S. SABOTINOVA

The main lines of a theory of the interpreters (the inference engines) in fuzzy controlled rule-based systems are sketched. The theory is applied for the case of a system for fuzzy natural language robot control.

During recent years we developed the foundations of the theory of a kind of rule-based systems which we called controlled rule-based systems (CRBS) (e.g. [1] - [3]), and gained some experience in applying this approach in building instrumental tools and particular systems (earlier works cited in [4]). In fact, we consider the CRBS's as a convenient universal computational model, capable of representing any particular algorithm (see also [5]).

Moreover, the whole theory developed so far deals with fuzzy CRBS's (FCRBS) actually. As stated as early as in [6], fuzzification is possible in many points, and equivalent transformations may be considered. The theoretical model allows arbitrary fuzzifications but in any particular case only few of these opportunities are really necessary. One has also to keep in mind that any fuzzification yields big losses of memory and computational time.

For those reasons in [2], where algorithms of the interpreters in FCRBS's have been considered, the following line of exposition has been adopted. Firstly, the general case is studied when all the composite fuzzy relations of the behaviour are essentially fuzzy. Secondly, a universal algorithm is given when some of these composites may be in fact partial or total functions. Finally, based on this general but non-effective algorithm, one can easily form efficient special algorithms for any particular case.

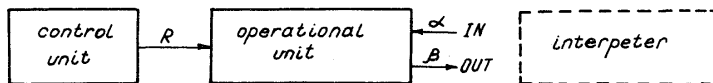


Fig. 1.

A FCRBS has the general form shown in Figure 1. The control unit is a recursive fuzzy automaton [3] called control network. As shown in [3] it can be represented by an ordinary fuzzy (mono)automaton $A_c = \langle Z, R, h_c : Z \rightarrow Z \times R, Z_i, Z_f \rangle$ but with an infinite

state set $Z = S \times T$. By D we denote the set of states of the operational unit. It can be considered as an infinite automaton without outputs, $A_o = \langle D, R, h_o = \delta_o : D \times R \rightarrow R \rangle$. R is a set of fuzzy rules [3, 5] on D . If $b : D \times Z \rightarrow D \times Z$ is the one-step behaviour of the FCRBS then the essential part of its (many-step) behaviour $B' : D \times Z \rightarrow D \times Z$ is some sort of reflective transitive closure of b (cf. [2]). The purpose of the interpreter is just to "compute" B' .

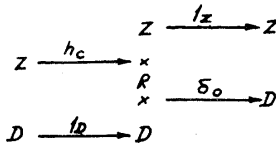


Fig. 2.

The one-step behaviour b of the fuzzy CRBS is by definition the composition of Figure 2.¹

Replacing h_c in Figure 2 by an equivalent composition we get the diagram of Figure 3 representing the one-step behaviour b . Here W is the set of the so called semi-arrows; h_c^R is a total function.

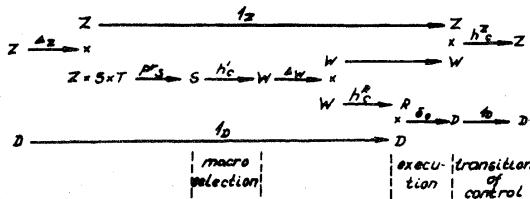


Fig. 3.

The main phases of the process are shown in the figure. The other composites are trivially "computed". An interpreter based on this decomposition of the one-step behaviour, is called the interpreter with first phase macroselection. Other variants based on different representations of b , are also possible.

An interesting, and with practical value system where we came across the necessity of dealing with fuzziness, was a system for fuzzy natural language robot control reported

¹Figure 3 of [3] is an inaccurate analogue of Figure 2. The reason is that the dependence between δ_c and λ_c has not been taken into consideration in [3]. In more general terms, a monoautomaton (called h -automaton in [9]) determines uniquely a Mealy-automaton (δ - λ -automaton), but not *vice versa* [9].

in [7, 8]. The system understands natural language (which includes fuzzy notions) tasks for two-dimensional motion of a robot and, using the map of the area, plans the robot actions working off the fuzziness. It includes a linguistic analyzer, a fuzzy compiler (FC), subsystems for generating the output assembler program or for simulating the robot's motion on the screen, as well as a number of editors. The linguistic analyzer transforms the task, typed in Bulgarian language, into a formal fuzzy program, consisting of both deterministic and fuzzy instructions. Using the map of the area, the FC produces from the fuzzy program one or several deterministic programs which correspond to a certain valuation degree to the source fuzzy program.

Two earlier (rather complex) implementations of the FC have been discussed in [7, 8]. Then we realized that a fuzzy program may be actually considered as a special case of a fuzzy control network, more precisely, a fuzzy (non-recursive) automaton with fuzzy instructions as outputs and with a linear structure (because the source natural language task consists of consecutive imperative sentences). This point of view automatically solved the problems about the FC algorithm – now we only had to use the general theory and methodology of [2] for this rather simple case (and the particular data structures). We must report that the practical result was a small and efficient computer program.

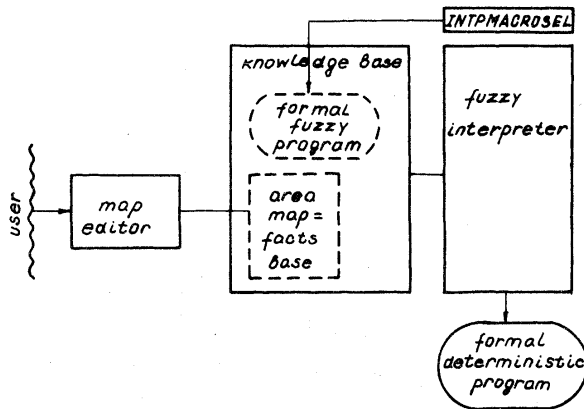


Fig. 4.

The fragment of Figure 1 of [8] corresponding to the FC now takes the form of Figure 4.

For the application under consideration by a state d of the operational unit we mean a pair $d = \langle pos, DP \rangle$. On its hand, the position pos has the form $pos = (x, y, dir)$ where x and y are the current coordinates of the robot on the map, and dir is the current

direction. DP is the currently formed fragment of the object deterministic program (showing how the robot can come to position pos).

Analysing Figure 3 for our particular application we see that $h'_c : S \rightarrow W$ is a mapping (not a fuzzy relation) because the fuzzy program has a linear structure. The only fuzzy relation in the diagram is $\delta_o : D \times R \rightarrow D$. Hence, in accordance with the general theory of [2] and taking into consideration the characteristics of the data structures, we obtain an algorithm of the fuzzy interpreter shown in Figure 5.

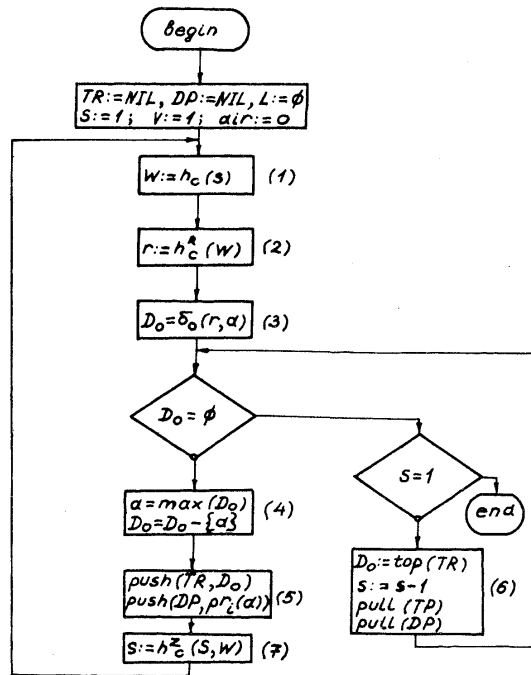


Fig. 5.

Here TR is a stack of reserves. Stacks TP and T (cf. [2]) are not necessary; as a result $z = s$. The nature of DP makes a representation of it as a stack very convenient. By L we denote the set of solutions, i.e., a fuzzy set of deterministic programs. The inclusion of a new solution into L takes place during the execution (block (3)) of an instruction r

for end of the fuzzy program. Initially the current valuation v equals 1, and the direction is fixed to north (0^0).

Blocks (1) and (2) reduce actually to just reading the current fuzzy instruction. The result of its execution (block (3)) is a fuzzy set D_0 of new states. During the phase of selection (block (4)) a state with a maximal valuation is chosen. The set of reserves is pushed into the stack TR (block (5)), and the deterministic instruction $pr_i(d)$ corresponding to the transition to the selected new state d , is pushed into stack DP . A version of the interpreter is chosen in which the stack of reserves stores not $\langle D_0, z \rangle$ but D_0 only. When necessary (block (6)), s is restored not from the stack but by execution of $s := s - 1$. Because of the linear form of the program, block (7) reduces, in fact, to $s := s = 1$.

Of considerable interest is the way the various types of fuzzy instructions are processed in block (3). Details of the approach adopted in the system have been described in [10]. The reader can find also a discussion on the possibilities to influence the process of fuzzy interpretation, as well as a list of directions of possible extensions of the system capabilities, in particular manipulation of recursive non-linear fuzzy programs, usage of quantifiers, modifiers and fuzzy qualifiers [11], etc.

Based on a quite simple interpretation of fuzziness, the system proved to be very efficient and showed, at the same time, quite satisfactory results in finding the possible ways on the map. Some of its current limitations are actually nullified by the basic "fuzzy" idea of searching for a number of solutions instead of a single one.

REFERENCES

- [1] K. Kratchanov: On the Foundations of Rule-Based Systems. *CSTR 34/85* (1985), Dept. Comp. Sci., Brunel Univ., Uxbridge, UK, 47 pages.
- [2] K. Kratchanov and I. Stanev: "Computing" the Reflective Transitive Closure of a Fuzzy Graph: Applications to Fuzzy Rule-Based System Interpreters. 11-th Poznan Mathematical Session, Poznan, Poland, 1987 (in press).
- [3] K. Kratchanov and I. Stanev: Fuzzy rule-based systems, fuzzy algorithms, fuzzy reasoning. In: First Joint IFSA-EC and EURO-WG Workshop on Progress in Fuzzy Sets in Europe, Warsaw, Poland, 1986 (J. Kacprzyk and A. Straszak, eds.). *Prace IBS PAN*, 167, 1988, Warszawa, pp. 155-166.
- [4] K. Kratchanov et al.: Rule networks: a style of programming and an expert system toolkit. In: *Artificial Intelligence III. Methodology, Systems, Applications* (T.O. Shea and V. Sgurev, eds.), North-Holland, Amsterdam 1988, pp. 265-272.
- [5] K. Kratchanov: Computations, Problem Solving and Rule-Based Systems. *CSTR 33/85* (1985), Dept. Comp. Sci., Brunel Univ., Uxbridge, UK, 1985, 203 pages.
- [6] K. Kratchanov: Towards the Fundamentals of Fuzzy Rule-Based Systems. *CSTR 35/85* (1985), Dept. Comp. Sci., Brunel Univ., Uxbridge, UK, 1985, 19 pages.
- [7] I. Stanev, K. Kratchanov et al.: Intelligent system for fuzzy robot control (in Bulgarian). *Techn. Univ. of Rousse Annuals 28* (1986), ser. 6, 43-48.
- [8] K. Kratchanov and I. Stanev: A rule-based system for fuzzy natural language robot control. In: *Proc. 4-th Internat. Conf. Artif. Intelligence and Information-Control Systems of Robots*, Smolence, Czechoslovakia 1987 (I. Plander, ed.), North-Holland, Amsterdam 1987, pp. 304-309.

- [9] K. Kratchanov: Mealy- automata and Moore-automata, δ - λ -Automata and h -Automata. Techn. Univ. of Rouse Annuals 29 (1981), 6, 109-114. In Bulgarian.
- [10] D. Sabotinoва and I. Stanev: Interpretation of fuzzy programs for robot control (in Bulgarian). Techn. Univ. of Rouse Annuals 32 (1989), 9 (in press).
- [11] D. Shapiro: Decision-Making in Organisation Control Systems: Applications of Fuzzy Categories (in Russian). Energoatomizdat, Moscow 1983.

Doc. Dr. Konstantin A. Kratchanov, Head of the Centre of Electronics and Electronic Technologies, Technical University, 61 Leningrad Blvd., Plovdiv 40000. Bulgaria.

Dr. Ivan Stanev, Dr. Desislava S. Sabotinoва, Dept. of Computer Engineering, Technical University, 8 Studentska Str., Rouse 70-17. Bulgaria.