# PARALLEL REALIZATION OF FUZZY AUTOMATON BASED ON FUZZY NEURONS

Vladimír OLEJ, Department of Computer Science, Technical University, 031 19 Liptovský Mikuláš, Czechoslovakia

Abstract: This paper presents parallel algorithm for realization of fuzzy automaton which is realized by fuzzy neurons.

## 1. Introduction

Finite deterministic automaton (FDA) is based on classical binary logic. Language input string enters the FDA and after processing the FDA the latter is in the final state (FDA accepts the given string), or occurs in some of nonfinal states (FDA does not accept the given string). By means of fuzzy set theory [1] the fuzzy acceptance of regular fuzzy language input string (RFLIS) can be determined after processing by finite fuzzy automaton (FFA). The fuzzy acceptance acquires value from the interval [0,1]. This FFA can be realized by an adaptive fuzzy neural net.

Parallel algorithm of RFLIS processing (written in parallel programming language OCCAM2) represents the possibility of technical realization of an analyzer on the basis of transputers. Technical realization of the design of RFLIS analyzer architecture on the basis of the transputers can be considered as highly perspective in the given field, due to their high performance, simplicity of multitransputer systems, simple possibility of expansion and possibility of unified approach to the design of technical and programming means.

## 2. Fuzzy Language and Finite Fuzzy Automaton

Fuzzy language $\mathbf{L}$ is given by the generalization of formal language. It is expressed by a set of ordered couples

$$\mathbf{L} = \{x,\ u(x)\}$$

where: - x is string of fuzzy language, $x \in \mathbf{L}$,

- $u(x)$ is value of membership function.

Regular fuzzy language **L** can be described by fuzzy production rules. The value from the interval [0,1] is assigned to each rule. On the basis of these rules the fuzzy acceptance $D(x)$ RFLIS is determined. As in the case of the theory of formal languages for each regular fuzzy language there is an equivalent FFA. The input of FFA is RFLIS, its output is the fuzzy acceptance $D(x)$. Finite fuzzy automaton thus realizes the membership function of regular fuzzy language. Each FFA is defined by an ordered 5-tuple

$$FFA = \langle \mathbf{I}, \mathbf{S}, \mathbf{M}, \mathbf{S}_0, \mathbf{F} \rangle$$

where:
- **I** is finite input alphabet,
- **S** is finite set of states,
- **M**: $\mathbf{S} \times \mathbf{I} \times [0,1] \longrightarrow \mathbf{S}$ is transition function, represented by transition fuzzy matrix (TFM),
- $\mathbf{S}_0$ is vector of membership function values (VMFV) of initial states, $(\mathbf{S}_0 \subset \mathbf{S})$,
- **F** is VMFV of final states, $(\mathbf{F} \subset \mathbf{S})$.

The producing procedure is realized by a matrix describing the interconnection of n neurons, where each neuron represents a state of FFA.

Let RFLIS be $x = x_1, x_2, \ldots, x_n$. Then its fuzzy acceptance $D(x)$ can be expressed in the following way

$$D(x) = \mathbf{S}_0 \circ \mathbf{S}(x) \circ \mathbf{F}^T = \mathbf{S}_0 \circ \mathbf{S}(x_1) \circ \ldots \circ \mathbf{S}(x_n) \circ \mathbf{F}^T$$

where:
- symbol $\circ$ denotes **MIN-MAX** operation,
- $\mathbf{S}(x_i)$ is transition function, represented by TFM,
- $\mathbf{F}^T$ is the transposed VMFV of final states.

## 3. Parallel Algorithm of Processing of Regular Fuzzy Language Input String

Development of the present computing systems is characterized by increasing their speed on the basis of parallel realization of the computing process. The necessity of parallelization follows from the need to realize the response of a computing system in real time. This requirement is often so demanding that it cannot be met even by computing means designed on the basis of the most advanced technology, providing the computing process is realized sequentially. The present development of the production technology of elements with high-scale integration (mainly transputers) enables to design parallel computing systems at relatively available prices in various architectures so that they may suit particular applications to their best.

Algorithm for processing RFLIS can be realized by the parallel programming language OCCAM2 [2]. The structure of programming language OCCAM2 corresponds to the hardware architecture of transputers. It enables programming of parallel computing transputer **systems**. The input quantities of algorithm of processig RFLIS are:

- number s TFM **S**,
- number r of components $\mathbf{S}_0(i)$ VMFV of initial states $\mathbf{S}_0$, components $\mathbf{F}(i)$ VMFV of final states **F** and of elements $\mathbf{S}(h,i,j)$ TFM **S**.

Parallel processing of RFLIS is given by this algorithm [3]:

```
PROC Comp (REAL X, Y, Z)
  REAL P:
  SEQ
    IF
      X < Y
        P = X
      TRUE
        P = Y
    IF
      P > Z
        Z = P
      TRUE
        SKIP
:
```

realization **MIN-MAX** operation between VMFV initial states $\mathbf{S}_0$ and TFM **S**

```
  PAR h = 1 FOR s
```

```
PAR i = 1 FOR r
  SEQ
    Q[i] = 1
    - - assigning of  initial value to resulting VMFV Q
    PAR j = 1 FOR r
      Comp (S0[j], S[h][j][i], Q[i])
      - - comparing of j-th components S0(j) VMFV of
      - - initial states S0 with elements S(h,j,i) TFM S
realization of MIN-MAX operation between resulting VMFV Q
and VMFV of final states F
 SEQ
   D = 1
   - - assigning of initial value to resulting value fuzzy
   - - acceptance D
   PAR i = 1 FOR r
     Comp (Q[i], F[i], D)
     - - comparing of i-th components Q(i) of resulting
     - - VMFV Q and i-th components F(i) of VMFV of final
     - - states F
```

4. Summary

These fuzzy neuron networks are powerful in computing fuzzy acceptance of fuzzy automaton, and thus well suited to tasks like pattern or language recognition. Algorithm of the processing of RFLIS can be applied on signal recognition.

References

[1] SHIUE,L.C.-GRONDIN,R.O.: On Designing Fuzzy Learning Neural-Automata. Proc. of the IEEE First International Conference on Neural Networks, Vol.2, 1987, San Diego, California, pp.299-307.

[2] MAY,D.:OCCAM2. Product Definition. Bristol, INMOS, 1986.

[3] OLEJ,V.-CHMÚRNY,J.-LEHOTSKÝ,M.: Fuzzy Automaton Based on Fuzzy Neurons. Proc. of the International Symposium on Neural Networks and Neural Computing, Neuronet 90, Prague, 1990, pp.264-266.