Parallel Architectures to compute Fuzzy Inferences based on the Goedel Implication

C. Moraga  
Department of Computer Science  
University of Dortmund  
Germany

J. Cañas, J. Atkinson, M. Gómez  
Department of Computer Science  
University “Federico Santa María”  
Valparaíso, Chile

Abstract

The main result of this paper is to show the speed-up possibilities of fuzzy inferences through parallel processing. It is shown that if a Goedel-type implication is used, it is possible to apply the compositional rule of inference to the Aggregation of elementary if-then-(else) rules. Both a systolic and standard multiprocessor architecture are discussed.

1 Introduction

Fuzzy Logic [1] finds important application in the realisation of rule based systems [2]. This is however computationally very demanding. Several studies have been done to develop parallel architectures to alleviate this problem [3,4,5] All these studies, including our own [6] use the so called Mamdani implication, which has found mainly applications in the important area of fuzzy control [7]. In this paper we develop a parallel architecture to speed-up the processing of fuzzy inferences using a Goedel-type implication.

The paper is structured as follows. In the next section we give the necessary background in fuzzy logic, particularly with respect to the realisation of implications by means of fuzzy associative memories as well as a detailed numerical example. In section 3 we discuss different parallel architecture approaches.

2 Fuzzy Logic

Let $U$ be a non empty set, called universe of discourse. A fuzzy subset $A$ of $U$ is defined by means of the membership function $\mu$ as follows:

$$\mu_A : U \rightarrow [0,1]$$

Preliminary work leading to this paper was done meanwhile C. Moraga was visiting the University “Federico Santa María” under the sponsorship of the Fundación Andes, Chile.
\( \mu_A \) associates to each element of \( A \) (which is also an element of \( U \)) a degree of membership.

Let \( U, V \) and \( W \) be non-necessarily disjoint universes. Moreover let \( X \) and \( A_i \) be fuzzy subsets of \( U \), \( Y \) and \( B_i \) fuzzy subsets of \( V \), and \( Z \) and \( C_i \) fuzzy subsets of \( W \).

The following notation will be used: \( A_i \in \mathcal{F}(U) \), \( B_i \in \mathcal{F}(V) \), \( C_i \in \mathcal{F}(W) \), where \( \mathcal{F}(U) \) denotes the set of all fuzzy subsets on \( U \).

A fuzzy inference system is structured as a set of rules \( R_i \) like the following:

\[
R_i : \text{if } X \text{ is } A_i \text{ and } Y \text{ is } B_i \text{ then } Z \text{ is } C_i
\]

\( R_i \) may be expressed as a fuzzy relation on the cartesian product of the background universes.

\[
R_i \subset (U \times V \times W)
\]

where \( \forall u \in U, \ v \in V \) and \( w \in W \).

\[
\mu_{R_i}(u, v, w) = \mu_{(A_i \cap B_i)} - C_i(u, v, w)
\]

The intersection of fuzzy sets will be computed as usual by taking the minimum between \( \mu_{A_i}(u) \) and \( \mu_{B_i}(v) \).

The "implication" between fuzzy sets has been widely studied (see e.g. [8]). In this paper we use the Goedel implication, which is defined as follows:

\[
\forall x, y \in [0, 1] \quad x \rightarrow y = \begin{cases} 
1 & \text{if } x \leq y \\
 y & \text{if } x > y
\end{cases}
\]

It is simple to see that if \( x, y \in \{0, 1\} \) then the Goedel implication is equivalent to the classical implication in binary logic.

**The Associative Memory Model**

In order to simplify the notation in the rest of the paper we consider only finite fuzzy subsets (which is not an unusual restriction for most practical applications) and describe them as vectors of membership degrees.

Let \( A = [a_1, a_2, \cdots, a_p] \); \( a_i = \mu_A(u_i) \ i = 1, 2, \cdots, p \)

\( B = [b_1, b_2, \cdots, b_q] \); \( b_j = \mu_B(v_j) \ j = 1, 2, \cdots, q \)
Lemma: Consider the fuzzy rule

\[
\text{if } X \text{ is } A \text{ then } Y \text{ is } B
\]

or simplified: \( A \rightarrow_G B \), with \( A \in \mathcal{F}(U) \), \( B \in \mathcal{F}(V) \)

(where \( \rightarrow_G \) denotes the Goedel-implication)

Let \( R = A^T \circ_G B = \begin{bmatrix} a_1 \rightarrow b_1 & \cdots & a_1 \rightarrow b_q \\ \vdots & & \vdots \\ a_p \rightarrow b_1 & \cdots & a_p \rightarrow b_q \end{bmatrix} \)

Then holds: \( (A \circ R = B) \iff H(A) \geq H(B) \)

where \( H(A) \) denotes the value of the largest membership degree in \( A \). Similarly for \( H(B) \). Moreover the symbols "\( \circ_G \)" and "\( \circ \)" denote the Goedel implication as outer vector product and the min-max vector-matrix product, respectively.

Proof:

\[
\text{if } A \circ R = \begin{bmatrix} a_1 & \cdots & a_p \end{bmatrix} \circ \begin{bmatrix} a_1 \rightarrow b_1 & \cdots & a_1 \rightarrow b_q \\ \vdots & & \vdots \\ a_p \rightarrow b_1 & \cdots & a_p \rightarrow b_q \end{bmatrix} = B
\]

then \( \max (\min(a_1, a_1 \rightarrow b_1), \ldots, \min(a_p, a_p \rightarrow b_1)) = b_1 \)

\[
\vdots
\]

\( \max (\min(a_1, a_1 \rightarrow b_q), \ldots, \min(a_p, a_p \rightarrow b_q)) = b_q \)

Let \( \max (\min(a_1, a_1 \rightarrow b_j), \ldots, \min(a_p, a_p \rightarrow b_j)) = \min(a_i, a_i \rightarrow b_j) \)

for some \( i \in \{1, 2, \ldots, p\} \)

if \( a_i > b_j \) then \( \min(a_i, a_i \rightarrow b_j) = \min(a_i, b_j) = b_j \)

or if \( a_i = b_j \) then \( \min(a_i, a_i \rightarrow b_j) = \min(a_i, 1) = a_i = b_j \)

Moreover

\( \text{if } \min(a_i, a_i \rightarrow b_j) = b_j \text{ then } \)

either (1) \( a_i \geq a_i \rightarrow b_j \) from where \( \min(a_i, a_i \rightarrow b_j) = a_i \rightarrow b_j \)

but if \( a_i \rightarrow b_j = b_j \) then \( a_i > b_j \)
or (2) \( a_i < a_i \to b_j \) (from where \( \min(a_i, a_i \to b_j) = a_i \))

It follows \( a_i = b_j \)

Since this holds without restrictions on \( i \) or \( j \), the assertion follows.

**Definition: Consistence**

Given a set of fuzzy rules \( \{ R_i : A_i \to C_i \} \), define \( R = \bigcap_i R_i \). A system with the property that for all \( i \) \( B_i = A_i \circ R \) is called consistent.

**Example:**

Let \( U, V \) and \( W \) be the universes of discourse. Moreover let \( A_i \in \mathcal{F}(U) \), \( B_i \in \mathcal{F}(V) \) and \( C_i \in \mathcal{F}(W) \), with \( i = 1, 2, 3 \).

The fuzzy sets have the following structure:

\[ \forall i = 1, 2, 3 \quad A_i := (u_1/\mu_{A_i}(u_1), u_2/\mu_{A_i}(u_2), u_3/\mu_{A_i}(u_3), u_4/\mu_{A_i}(u_4)), \]

which will be written in the following simplified way:

\[ A_i := [\mu_{A_i}(u_1), \mu_{A_i}(u_2), \mu_{A_i}(u_3), \mu_{A_i}(u_4)] \]

similarly for \( B_i \) and \( C_i \).

These fuzzy sets build fuzzy rules as follows:

\[ R_i : \text{if } X \text{ is } A_i \text{ and } Y \text{ is } B_i \text{ then } Z \text{ is } C_i \]

or \[ R_i : \text{if } X \text{ is } A_i \text{ then if } Y \text{ is } B_i \text{ then } Z \text{ is } C_i \]

In the case of the Goedel implication the above rules are equivalent and may be expressed by the following Relation

\[ R_i \Rightarrow \mu_{R_i}(u, v, w) = \mu_{(A_i \land B_i) \rightarrow C_i}(u, v, w) \]

**Numerical example:**

\[
\begin{align*}
A_1 &:= [0 \ 0 \ 0.7 \ 0.8] & A_2 &:= [0 \ 0.8 \ 0.1 \ 0] & A_3 &:= [0.9 \ 0.2 \ 0 \ 0] \\
B_1 &:= [0.9 \ 0.1 \ 0 \ 0] & B_2 &:= [0 \ 0 \ 0.7 \ 0.7] & B_3 &:= [0 \ 0.7 \ 0.8 \ 0] \\
C_1 &:= [0 \ 0.6 \ 0.8 \ 0] & C_2 &:= [0.7 \ 0.6 \ 0.2 \ 0.2] & C_3 &:= [0.2 \ 0.2 \ 0.7 \ 0.8]
\end{align*}
\]
\[ A_1^T \land B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.7 & 0.1 & 0 & 0 \\ 0.8 & 0.1 & 0 & 0 \end{bmatrix} \]

The three dimensional relation will be given the following two dimensional representation:

\[
\begin{align*}
\mu_{C_1}(w_1) &= 0 & \mu_{C_1}(w_2) &= 0.6 & \mu_{C_1}(w_3) &= 0.8 & \mu_{C_1}(w_4) &= 0 \\
R_1 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \ \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 0.6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0.6 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{bmatrix}
\end{align*}
\]

similarly

\[ A_2^T \land B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.7 \\ 0 & 0 & 0.1 & 0.1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

leading to

\[
\begin{align*}
\mu_{C_2}(w_1) &= 0.7 & \mu_{C_2}(w_2) &= 0.6 & \mu_{C_2}(w_3) &= 0.2 & \mu_{C_2}(w_4) &= 0.2 \\
R_2 &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0.6 & 0.6 & 1 & 1 & 0.2 & 0.2 & 1 & 1 & 0.2 & 0.2 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]

and

\[ A_3^T \land B_3 = \begin{bmatrix} 0 & 0.7 & 0.8 & 0 \\ 0 & 0.2 & 0.2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]

leading to

\[
\begin{align*}
\mu_{C_3}(w_1) &= 0.2 & \mu_{C_3}(w_2) &= 0.2 & \mu_{C_3}(w_3) &= 0.7 & \mu_{C_3}(w_4) &= 0.8 \\
R_3 &= \begin{bmatrix} 1 & 0.2 & 0.2 & 1 & 1 & 1 & 1 & 1 & 0.7 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \ \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\end{align*}
\]
With \( R = \cap (R_i) = \min_{i} (R_1, R_2, R_3) \) the global relation is given by

\[
R = \begin{bmatrix}
1 & 0.2 & 0.2 & 1 & 1 & 0.2 & 0.7 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0.6 & 0.6 & 1 & 1 & 0.2 & 0.2 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1
\end{bmatrix}
\]

a) Test of Consistency

\[
\mu_Z(w) = \max_u T(\mu_x(u), \max_v T(\mu_Y(v), \mu_R(u, v, w)))
\]

\[
I(w, u) = \def \max_v T(\mu_Y(v), \mu_R(u, v, w))
\]

Define \( R \circ^* B^T \) to be the max-min matrix-vector product computed successively for the restrictions (or blocks) of \( R \) with respect to \( \mu_C(w_1), \mu_C(w_2), \mu_C(w_3) \) and \( \mu_C(w_4) \). Then holds:

\[
I_1(w, u) = (R \circ^* Y^T)^T
\]

i) \( X = A_1, Y = B_1 \)

\[
I_1(w, u) = (R \circ^* Y^T)^T = \begin{bmatrix}
0.9 & 0.9 & 0 & 0 \\
0.9 & 0.9 & 0.6 & 0.6 \\
0.9 & 0.9 & 0.9 & 0.9 \\
0.9 & 0.9 & 0 & 0
\end{bmatrix}
\]

\[
Z^T = (R \circ^* Y^T)^T \circ X^T = \begin{bmatrix}
0.9 & 0.9 & 0 & 0 \\
0.9 & 0.9 & 0.6 & 0.6 \\
0.9 & 0.9 & 0.9 & 0.9 \\
0.9 & 0.9 & 0 & 0
\end{bmatrix} \circ \begin{bmatrix}
0 \\
0 \\
0.7 \\
0.8
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0.6 \\
0.8
\end{bmatrix} = C_1^T
\]

where "\( \circ \)" is the simple max-min matrix-vector product.

ii) \( X = A_2, Y = B_2 \)

\[
Z^T = \begin{bmatrix}
0.2 & 0.7 & 0.7 & 0.7 \\
0.7 & 0.6 & 0.7 & 0.7 \\
0.7 & 0.2 & 0.7 & 0.7 \\
0.7 & 0.2 & 0.7 & 0.7
\end{bmatrix} \circ \begin{bmatrix}
0 \\
0.8 \\
0.1 \\
0
\end{bmatrix} = \begin{bmatrix}
0.7 \\
0.6 \\
0.2 \\
0.2
\end{bmatrix} = C_2^T
\]
iii) $X = A_3, Y = B_3$

$$Z^T = \begin{bmatrix} 0.2 & 0.8 & 0.8 & 0.8 \\ 0.2 & 0.7 & 0.8 & 0.8 \\ 0.7 & 0.7 & 0.8 & 0.8 \\ 0.8 & 0.8 & 0.7 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.9 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.7 \\ 0.8 \end{bmatrix} = C_3^T$$

b) Test of robustness

i) $X = \begin{bmatrix} 0 & 0 & 0.6 & 0.9 \end{bmatrix}$ (similar to $A_1$)

$Y = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \end{bmatrix}$ (similar to $B_1$)

$$Z^T = \begin{bmatrix} 0.8 & 0.8 & 0 & 0 \\ 0.8 & 0.8 & 0.6 & 0.6 \\ 0.8 & 0.8 & 0.8 & 0.8 \\ 0.8 & 0.8 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 \\ 0 \\ 0.6 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0.6 \\ 0 \end{bmatrix} = C_1^T$$

ii) $X = \begin{bmatrix} 0.8 & 0.3 & 0 & 0 \end{bmatrix}$ (similar to $A_3$)

$Y = \begin{bmatrix} 0 & 0.6 & 0.9 & 0 \end{bmatrix}$ (similar to $B_3$)

$$Z^T = \begin{bmatrix} 0.2 & 0.9 & 0.9 & 0.9 \\ 0.2 & 0.6 & 0.9 & 0.9 \\ 0.7 & 0.6 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 \end{bmatrix} \circ \begin{bmatrix} 0.8 \\ 0.3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.7 \\ 0.8 \end{bmatrix}$$

$Z$ is similar to $C_3$

3. A systolic architecture

Systolic systems [9] are multiprocessor architectures with nearest neighbour communication structure and decentralized control.

Systolic systems have shown to provide very efficient support for the computation of classes of problems, particularly those problems related to operations with matrices [10].

From the former numerical example it becomes apparent that the elementary Relations may be computed in a way similar to the outer product of three vectors, but with appropriate replacement of the required operations.
It is simple to see that the computation of the partial relation $R'_i = A_i \cap B_i$ can be done as shown in Fig 1.

Synchronized by a global clock, both vectors move from elementary processor to elementary processor. If each elementary processor computes and saves the minimum of the first visiting pair of elements of $A_i$ and $B_i$ before transferring these elements to the corresponding neighbors and after that only transfers elements, then after $(p+q)$ clock pluses the array of processors constitutes a memory containing $A_i \cap B_i$.

Recall the former numerical example. A complete elementary Relation may be computed by blocks. Consider now only the first block. The required computation at the $j$-th, $k$-th position is $\min(a_{ij}, b_{ik}) \rightarrow_G c_1$ for all $i = 1, \ldots, p$ and $j = 1, \ldots, q$. In order to do this with a systolic array, the value of $c_1$ has to propagate as a wave-front parallel to the positive diagonal of the Block synchronized with the vectors $A_i$ and $B_i$ in order to meet each pair $(a_{ij}, b_{ik})$ at the right place and the right time. (A wave-front propagation of $c_1$ may be obtained with a dedicated communication structure as shown in Fig. 2)

Similarly, the $m$-th block of an elementary Relation computes $\min(a_{ij}, b_{ik}) \rightarrow_G c_m$ (at the $j$-th, $k$-th position). It becomes apparent that every block needs the information $\min(a_{ij}, b_{ik})$ $j = 1, \ldots, p$; $k = 1, \ldots, q$ before computing the (local) Goedel implication. If we dedicate a systolic array to each block of the Relation and order these arrays to build a rectangular prism, then the elementary processors of the first array should pass the value of $\min(a_{ij}, b_{ik})$ to the corresponding neighbor in the 3rd. dimension before computing the Goedel implication. Every following array will do the same. This leads to a skewed format for the vector $C_i$ in order to compensate for the required propagation delay of $\min(a_{ij}, b_{ik})$. 
Abbildung 2: systolic wave-front propagation

The above analysis may be extended to matrices (with skewed format) comprising the corresponding vectors $A_1, \ldots, A_N; B_1, \ldots, B_N$ and $C_1, \ldots, C_N$ respectively [6]. (see Fig 3). If the program of the elementary processors is made to include the computation of the cumulative minimum of the locally calculated elementary Goedel implications, then the 3D systolic system will compute (and save in-place) the relation $R = \bigcap_i R_i$ in $O(p + q + r + N)$ clock pluses. (A sequential system would require $O(p \cdot q \cdot r \cdot N)$ computation steps.)

As explained in the numerical example, once the global Relation has been computed, i.e. once the set of fuzzy rules has been given a hardware structure, the inferences may be calculated by evaluating the following equations:

$$I(w, u) = (R \circ^* Y^T)^T = Y \circ^* R^T$$

and

$$Z^T = (R \circ^* Y^T)^T \circ X^T = (Y \circ^* R^T) \circ X^T$$

Since both $\circ^*$ and $\circ$ are operations, which exhibit a matrix-vector-product structure, they are quite simple to realize systolically.

Since every plane of the 3D systolic system has saved one block of the Relation the computation of $Y \circ^* R^T$ may be carried in all planes in parallel. The skewed vector $Y$ will be input at the first plane (as it was the case with the matrix $A$ for the computation of $R$) and propagates systolically both along the first plane as well as to the planes along the third dimension. The result of the elementary min-max operations will be propagated from left to right in every plane. It is assumed that the processors of the last column of every plane have a special register to save the final result, and at the same time provide the required array-representation of $Y \circ^* R^T$ (at the east surface of the 3D systolic system) to compute in-place $Z^T = (Y \circ^* R^T) \circ X^T$ driven by the skewed vector $X$. A detaillierte discussion on the structure and programming of a similar
A 3-D systolic system to compute $R$

system for Mamdami-type fuzzy systems may be found in [6]. The time complexity for the computation of inferences (once the Relation has been calculated) is $O(p+q+r+1)$.

It becomes apparent, that even though a systolic realization is theoretically possible and computationally very efficient, the fact that the architecture is three dimensional is a burden, which may be solved by using e.g., a back plane supported architecture. More severe is however the unflexibility of the system which is typical of all 100% hardware solutions: the size parameters of the problem cannot be changed.

The existence of a systolic solution is however a proof of a high degree of potential parallelism. The above problem can be efficiently solved by means of a SIMD architecture, where every processor would take over the processing of the corresponding input data to compute one block of the Relation, and later, the blockwise calculation of the inference. This would speed up the computation in a linear fashion with respect to a SISD architecture. Moreover if changes are required in the size parameters of the problem - (e.g. increased number of datapoints of the fuzzy sets)- only changes in the program of the components of the multiprocessor system are needed.

4 Conclusions

We have shown, that a consistent rule-based system operating with Goedel-type implications may be given efficient parallel realizations.
If the size of the problem is known and is expected not to change, then a full systolic realization provides the most efficient architecture. A SIMD architecture would provide close to linear-speed-up with respect to SISD systems, but keeping flexibility in the program of the processors.

Literatur