SOME PROBLEMS WITH THE CURRENT IMPLEMENTATION OF FORWARD INFERENCE

Joao P. Martins

December 5, 1980

During this week two problems with the current implementation of forward inference were found. One relates to the use of function nodes and the other to the network matching function. I will try to explain these problems hoping to receive comments and/or suggestions from the other members of the group.

1. Problems with function nodes

Consider the rules represented in Figure 1 and the following two runs of the inference system, one using forward inference and the other backward inference.

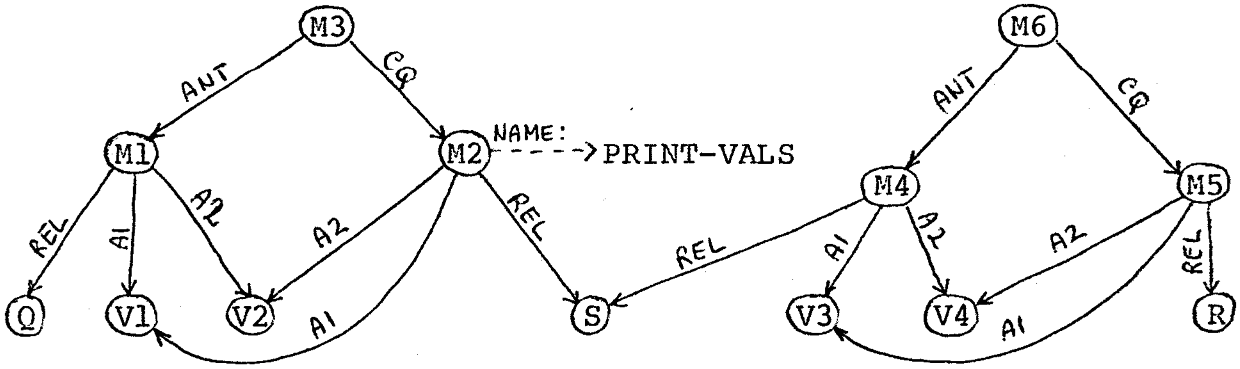

Figure 1
Network with rule containing a function node

1. Forward Inference run: Assume that having the network shown in Figure 1 we execute the function

(ADD A1 A REL Q A2 B). The node ADDed matches M1 with binding [A/V1,B/V2]. This fact triggers forward inference and after some processing is done the IMPLY process responsible for rule M3 gets a message stating that an instance of the antecedent of M3 (M1) was found with binding [A/V1,B/V2]. IMPLY looks at the consequent of M3 and since it is a function node, executes it. The function PRINT-VALS has been defined as:

```
(DP PRINT-VALS (A1 REL A2)
               (PRIN3 *A1 *REL *A2)
               (ADD-NODES (BUILD A1 (^ A1)
                                 REL S
                                 A2 (^ A2))))
```

The function ADD-NODES triggers further forward inference on the node A1 A REL S A2 B (this function has just been added to the file INFER). This node matches M4 with binding [A/V3,B/V4] and the forward inference system deduces the consequent M5 with binding [A/V3,B/V4]: A1 A REL R A2 B.

2. Backward Inference run (BACKWARD-FLAG ()): Assume that having the network shown in Figure 1 we build the node A1 A REL Q A2 B and ask the question A1 A REL R A2 B? This node matches M5 with binding [A/V3,B/V4] and the backward inference system attempts to prove the antecedent M4 in the same binding. There is no constant node in the network wich matches M4 with such a binding but there is a rule (M3) which may enable the deduction of such a node. The backward inference system starts working on rule M3 and finds an instance of its antecedent, the node A1 A REL Q A2 B. The consequent M2 can therefore be

deduced, a message stating this fact is passed to the process looking for instances of M4 and so an instance of M5 with binding [A/V3,B/V4] can be deduced.

After studying these two runs one may think that everything seems to be working just fine. In the first case we ADDed a node that matched M1 and the effect of this was propagated until it reached node M5; in the second case we built a node which matched M1 and asked for an instance of M5, getting back an affirmative answer.

The flaw in all this process, however, is that both deductions did not follow the same mechanism. In forward inference the system went from rule M3 to rule M6 via the function PRINT-VALS (in particular, through the execution of the function ADD-NODES) and in the backward inference the system went from rule M6 to rule M3 via a SWITCH process.

The question being raised here is "what should be done with function nodes to take care of this problem?".

2.Problems with MATCH

MATCH is the function that matches a given node (called source node) against the network. MATCH is given the source node and returns a list of the matched nodes (called target nodes) and the corresponding bindings. The way MATCH is implemented allows that a target node may immediately dominate more nodes than the source node, and as [Shapiro 77] pointed out "this seems appropriate when looking for rules to apply in a consequent fashion, but inappropriate when looking for rules to apply in an antecedent fashion". In fact, suppose that we have in the network the rule represented in Figure 2 and ask for an instance of its consequent. If such an instance is not explicitly stored in the network the inference system tries to find an instance of its antecedent in the same binding. With such a goal in mind it does a match of the antecedent against the network. Suppose that the node represented in Figure 3 exists in the network. Should this node be found by the matching function? If we recall that a node represents the conjunction of <u>all</u> the binary relations expressed by its arcs the answer to the question above will be YES since node M4 asserts that the following expression is true A1(M4,B1)&A2(M4,B2)&...&An(M4,Bn)&...&Am(M4,Bm), the definition

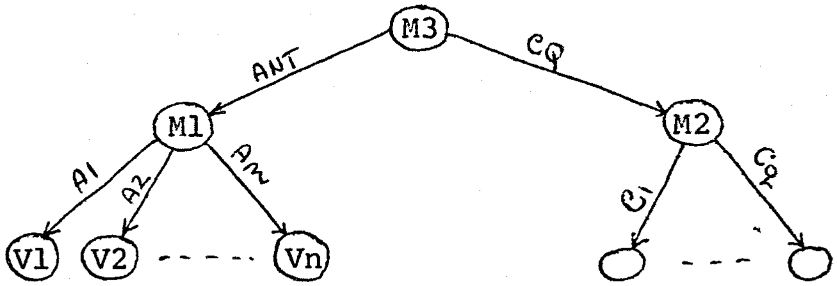

Figure 2
Example of rule

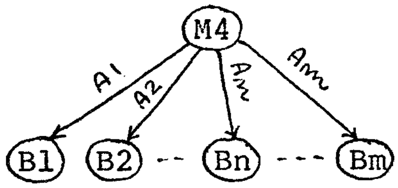

Figure 3
Node in the network

of conjunction entails that A1(M4,B1)&A2(M4,B2)&...&An(M4,Bn) is true, which means that M4 is an instance of M1 with binding [B1/V1,B2/V2,...,Bn/Vn]. We conclude that if the source node is the antecedent of a rule we should allow target nodes to have additional arcs.

Now, suppose that we are doing forward inference. In this case the source node is some node that has been ADDed to the network and the candidates for target nodes are the nodes in antecedent position of rules. If we allow the target node to have more arcs that the source node what would happen? Consider the example presented in Figure 4. If M1 is the source node (node just ADDed) should M3 be found by the matching function? Of course NOT. Node M1 asserts that the following conjunction holds A1(M1,X)&A2(M1,Y) and from this fact we can by no means infer A1(M1,X)&A2(M1,Y)&A3(M1,?Z). Therefore, in forward inference, the target nodes should not have more arcs than the source nodes. The case of their having the same arcs is obvious, so let us look at the case where the target node has less arcs that the source node. Again let us look at an example (Fig.5). Node M1 asserts that the following conjuction holds

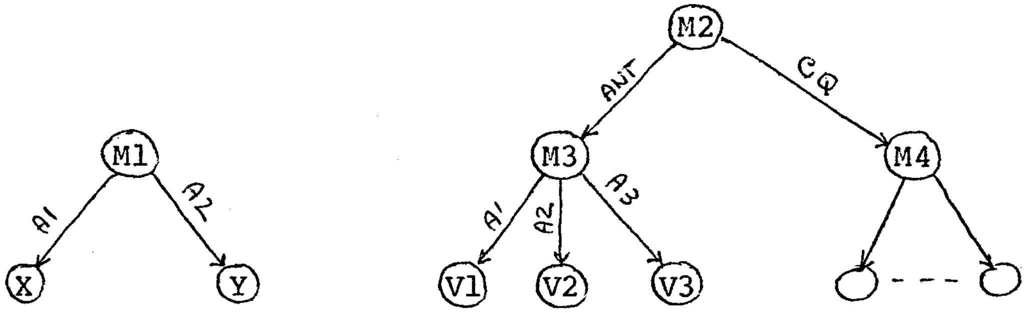

Figure 4
Sample network

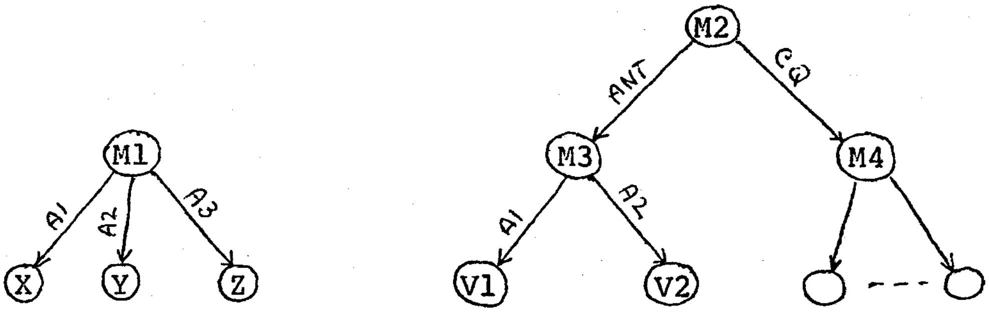

Figure 5
Sample network

A1(M1,X)&A2(M1,Y)&A3(M1,Z). Therefore the conjunction A1(M1,X)&A2(M1,Y) also holds and so there is in the network an instance of the node M3 with binding [X/V1,Y/V2]. We conclude that we should allow, in forward inference, target nodes having less arcs than source nodes.

From the preceding discussion it is easy to see that using the backward inference MATCH when doing forward inference is doubly wrong as it allows some matches that should not be done and does not allow other matches that should be done.

The point of all this? A new matching function has to be written for forward inference. Anyone interested in writing such a matching function???

3.References

1. Shapiro S.C., "Representing and locating deduction rules in a semantic network", in Proc. Workshop on Pattern-Directed Inference Systems, SIGART Newsletter, No.63, June 77, pp.14-18.