

# PATH-BASED AND NODE-BASED INFERENCE IN SEMANTIC NETWORKS

Stuart C. Shapiro

Department of Computer Science  
State University of New York at Buffalo  
Amherst, New York 14226

## Abstract

Two styles of performing inference in semantic networks are presented and compared. Path-based inference allows an arc or a path of arcs between two given nodes to be inferred from the existence of another specified path between the same two nodes. Path-based inference rules may be written using a binary relational calculus notation. Node-based inference allows a structure of nodes to be inferred from the existence of an instance of a pattern of node structures. Node-based inference rules can be constructed in a semantic network using a variant of a predicate calculus notation. Path-based inference is more efficient, while node-based inference is more general. A method is described of combining the two styles in a single system in order to take advantage of the strengths of each. Applications of path-based inference rules to the representation of the extensional equivalence of intensional concepts, and to the explication of inheritance in hierarchies are sketched.

## 1. Introduction

Semantic networks have developed since the mid sixties [10;11] as a formalism for the representation of knowledge. Methods have also been developing for performing deductive inference on the knowledge represented in the network. In this paper, we will compare two styles of inference that are used in semantic networks, path-based inference and node-based inference. In sections 2 and 3, these terms will be explained and references to systems that use them will be provided. In sections 4 and 5, the advantages and disadvantages of each will be discussed. Sections 6, 7 and 8 will show how they can be used to complement each other in a single semantic network system, how path-based inference can help represent the extensional equivalence of intensional concepts, and how a formalism for writing path-based inference rules can be used to explicate the notion of "inheritance" in a semantic network.

## 2. Path-Based Inference

Let us refer to a relation (perform binary) that is represented by an arc in a network as an arc-relation. If  $R$  is an arc-relation, an arc labelled  $R$  from node  $a$  to node  $b$  represents that the relationship  $aRb$  holds. It may be that this arc is not present in the network, but  $aRb$  may be inferred from other information present in the network and one or more inference rules. If the other information in the network is a specified path of arcs from  $a$  to  $b$ , we will refer to the inference as *path-based*. The ways in which such paths may be specified will be developed as this paper proceeds.

The two clearest examples of the general use of path-based inference are in SAMENLAQ II [18] and Protosynthex III [13]. Both these systems use what might be called "relational" networks rather than "semantic" networks since arc-relations include conceptual relations as well as structural relations (see [14] for a discussion of the difference). For example, in Protosynthex III there is an arc labelled COMMANDED from the node representing Napoleon to the node representing the French army, and in SAMENLAQ II an arc labelled EAST.OF goes from the node for Albany to the node for Buffalo. Both systems use relational calculus expressions to form path-based inference rules. The following relational operators are employed (we here use a variant of the earlier notations):

1. Relational Converse -- If  $R$  is a relation,  $R^C$  is its converse. So,  $\forall x, y (xR^C y \leftrightarrow yRx)$ .
2. Relational Composition -- If  $R$  and  $S$  are relations,  $R/S$  is  $R$  composed with  $S$ . So,  $\forall x, y (xR/S y \leftrightarrow \exists z (xRz \ \& \ zSy))$ .
3. Domain Restriction -- If  $R$  and  $S$  are relations,  $(S z)R$  is the relation  $R$  with its domain restricted to those objects that bear the relation  $S$  to  $z$ . So,  $\forall x, y, z (x(S z)R y \leftrightarrow (xSz \ \& \ xRy))$ .
4. Range Restriction -- If  $R$  and  $S$

are relations,  $R(S z)$  is the relation  $R$  with its range restricted to those objects that bear the relation  $S$  to  $z$ . So,  
 $\forall x, y, z (xR(S z)y \leftrightarrow (xRy \ \& \ ySz))$ .

5. **Relational Intersection** -- If  $R$  and  $S$  are relations,  $R \& S$  is the intersection of  $R$  and  $S$ . So,  
 $\forall x, y (xR \& Sy \leftrightarrow (xRy \ \& \ xSy))$ .

Notice that  $\forall Q, R, S, x, y, z (xR(Q z)/Sy \leftrightarrow xR/(Q z)Sy)$  so we can use the notation  $R(Q z)S$  unambiguously.

In SAMENLAQ II, path-based inference rules are entered by using the relational operators to give alternate definitions of simple arc labels. For example (again in a variant notation):

EAST.OF + EAST.OF/EAST.OF  
 declares that EAST.OF is transitive  
 SOUTH.OF + NORTH.OF  
 declares that  
 $\forall x, y (yNORTH.OFx \rightarrow xSOUTH.OFy)$   
 FATHER.OF + (GENDER MALE)PARENT.OF  
 declares that a father is a male parent.

SIR [11] is another relational network system that uses path-based inference. Although the original expressed inference rules in the form of general LISP functions, the reproduction in [16, Chap. 7] uses the notion of path grammars. The relation operators listed above are augmented with  $R^*$ , meaning zero or more occurrences of  $R$  composed with itself,  $R^+$ , meaning one or more occurrences of  $R$  composed with itself, and  $R \vee S$ , meaning the union of  $R$  and  $S$ . The following relations are used:

|                         |  |
|-------------------------|--|
| $x$ EQUIV $y$           | means $x$ and $y$ are the same individual                        |
| $x$ SUBSET $y$          | means $x$ is a subset of $y$                                     |
| $x$ MEMBER $y$          | means $x$ is a member of the set $y$                             |
| $x$ POSSESS $y$         | means $x$ owns a member of the set $y$                           |
| $x$ POSSESS-BY-EACH $y$ | means every member of the set $x$ owns a member of the set $y$ . |

To determine if  $x$  POSSESS  $y$ , the network is searched using the following rule:

POSSESS + EQUIV\*  
 /(POSSESS  
 $\vee$  (MEMBER/SUBSET\*/POSSESS-BY-EACH))  
 /SUBSET\*

The widest use of path-based inference is in ISA hierarchies. Fig. 1 is based on probably the most famous ISA hierarchy, that of Collins and Quillian [2]. The two important rules here are

ISA + ISA\*  
 and PROP + ISA\*/PROP

As McDermott [8] points out, ISA hierar-

chies have been abused as well as used. In Section 8, we will propose a method authors can use to describe their hierarchies precisely.

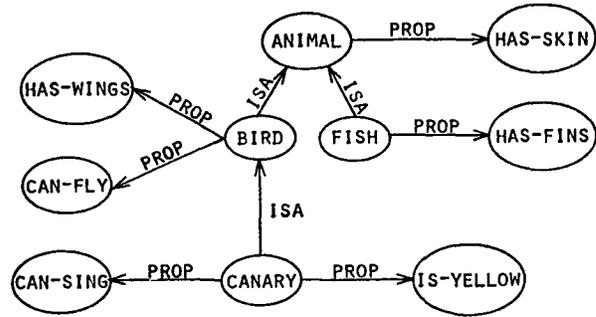


FIGURE 1: ISA hierarchy based on that of Collins and Quillian

### 3. Node-Based Inference

Several semantic network systems incorporate methods of representing general rules in a semantic network version of predicate calculus. Among these systems are those of Shapiro [14;15;17], Kay [7], Hendrix [6], Schubert [12], and Fikes and Hendrix [3]. Figure 2 shows such a network deduction rule representing

$\forall x [x \in \text{MAN} \rightarrow \exists y (y \in \text{WOMAN} \ \& \ x \text{LOVES} y)]$ .

Figure 3 shows a rule for

$\forall r [r \in \text{TRANSITIVE} \rightarrow \forall x, y, z (xry \ \& \ yrz \rightarrow xrz)]$ .

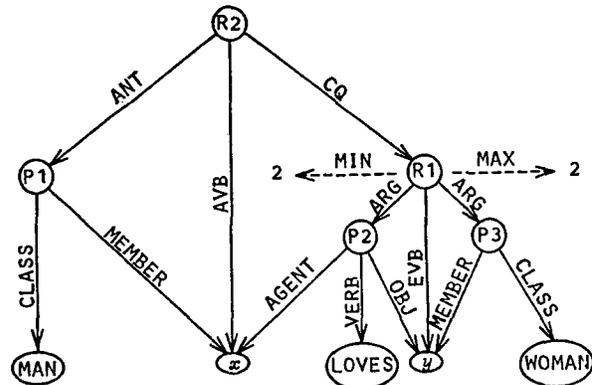


FIGURE 2: A semantic network deduction rule for  $\forall x [x \in \text{MAN} \rightarrow \exists y (y \in \text{WOMAN} \ \& \ x \text{LOVES} y)]$

The network formalism employed is that of Shapiro [15;17]. These deduction rules employ pattern nodes (P1, P2, P3, P4, P5, P6, P7), each one of which represents a pattern of nodes that might occur in the network. We will therefore call this kind of inference rule a *node-based* inference rule. Pattern nodes are related to each other by *rule nodes*, each of which represent a propositional operator, or, equivalently, an inference mechanism. For example, R2 represents the rule that if an instance of P1 occurs in the network, an instance of R1 with the same substitution

for  $x$  may be deduced. Quantification is represented in this notation by an arc-relation between a rule node and the variable nodes bound in the rule. For example,  $x$  is bound by a universal quantifier in R2 and  $y$  is bound by an existential quantifier in R1.

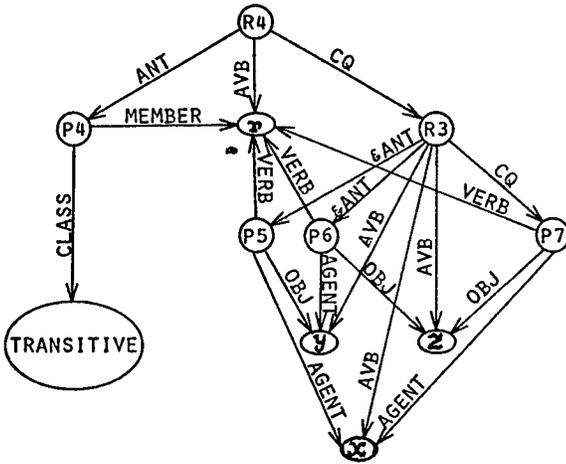


FIGURE 3: A semantic network deduction rule for  $\forall r[r \in \text{TRANSITIVE} \rightarrow \exists x,y,z(xry \ \& \ yrz \rightarrow xrz)]$

To see how a node-based inference proceeds, consider the network of Figure 4 in conjunction with the rule of Figure 3, and say that we wish to decide if A SUPPORTS C. The network that would represent that A SUPPORTS C matches P7 with the variable binding  $\{x/A, r/\text{SUPPORTS}, z/C\}$ . P4 in the binding  $\{r/\text{SUPPORTS}\}$  is matched against the network and is found to successfully match M1. P5  $\{x/A, r/\text{SUPPORTS}, y/y\}$  and P6  $\{y/y, r/\text{SUPPORTS}, z/C\}$  are then both matched against the network and each succeeds with a consistent binding of  $y$  to B. The rule thus succeeds and A SUPPORTS C is deduced. (Details of the bindings and the match routine are given in [15].)

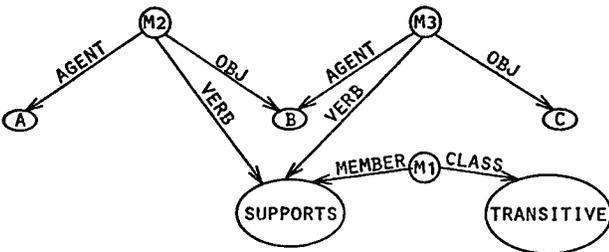


FIGURE 4: A network data base asserting that A SUPPORTS B; B SUPPORTS C and SUPPORTS  $\in$  TRANSITIVE.

It should be noted that set inclusion was represented by an arc (ISA) in Section 2, but set membership is being represented by a node (with a MEMBER, CLASS "case frame") in this section. The nodal representation is required by node-based inference rules and is consistent with the notion that everything that the network

"knows", and every concept to which the network can refer is represented by a node.

#### 4. Advantages of Node-Based Inference

The advantages of node-based inference stem from the generality of the syntax of node-based inference rules. Path-based rules are limited to binary relations, have a restricted quantification structure and require that an arc between two nodes be implied by a path between the same two nodes. Rule R2 of Figure 2 could not be written as a path-based rule, and, although the transitivity of SUPPORTS could be expressed by a path-based rule (SUPPORTS + SUPPORTS<sup>+</sup>), the "second order" rule R4 of Figure 3 could not.

Let us briefly consider how rule R4 is constructed, whether it really is or is not a second order rule, and why it could not be expressed as a path-based rule. Rule R4 supplies a rule for use with transitive relations. In order to assert that a relation is transitive (e.g. assertion node M1 of Figure 4), the relation must be represented as a node, rather than as an arc. This also allows quantification over such relations, since in all node-based inference rule formalisms variables may only be substituted for nodes, not for arcs. Since the relation is a node, another node must be used to show the relationship of the relation to its arguments (e.g. nodes M2 and M3 in Figure 4). Thus, R4 is really a first order rule derived from the second order rule  $\forall r[r \in \text{TRANSITIVE} \rightarrow \exists x,y,z(xry \ \& \ yrz \rightarrow xrz)]$  by reducing  $r$  to an individual variable and introducing a higher order relation, AVO, whose second argument is a conceptual relation and whose other arguments are conceptual individuals. So R4 is more accurately seen as the first order rule

$$\forall r[r \in \text{TRANSITIVE} \rightarrow \forall x,y,z(\text{AVO}(x,r,y) \ \& \ \text{AVO}(y,r,z) \rightarrow \text{AVO}(x,r,z))].$$

In this view, the predicates of semantic networks are not the nodes representing conceptual relations, but the different case frames. Rule R4 cannot be represented as a path-based rule because it is a rule about the relation AVO, and AVO is a trinary, rather than a binary relation.

Although some node-based inference rules cannot be expressed by path-based inference rules, it is easy to see that any path-based inference rule can be expressed by a node-based inference rule, as long as we are willing to replace some arc-relations by nodes and higher order predicates.

#### 5. Advantages of Path-Based Inference

The major advantage of path-based inference is efficiency. Carrying out a path-based inference involves merely checking that a specified path exists in the network between two given nodes (plus,

perhaps, some side paths to specified nodes required by domain and range restrictions). This is a well understood and relatively efficient operation, especially compared to the backtracking, intersection, or unification operations required to check the consistency of variable substitutions in node-based inference rules.

Moreover, path following seems to many people to be what semantic networks were originally designed for. The major search algorithm of Quillian's Semantic Memory is a bi-directional search for a path connecting two nodes [10, p. 249]. Also, the ability to do path tracing is a motivation underlying ISA hierarchies, and is why the Collins and Quillian results [2] gained such attention. These efficiencies are lost by replacing path-based inference rules by node-based inference rules.

### 6. Combining Path-Based and Node-Based Inference

We begin the task of unifying path-based and node-based inferences by noting the formal equivalence between an arc-relation and a two case case frame. Figure 5 illustrates this using ISA vs. SUB-SUP. Figure 5a shows the use of the ISA arc-relation to represent that canaries are birds. Figure 5b represents the same relationship by a SUB-SUP case frame, and has the advantage that the relationship is represented by a node, M4. Figure 5c is a redrawing of 5b, using the arc label SUB- to represent the relation SUBC. (It is generally understood in semantic network formalisms that whenever an arc representing a relation R goes from some node n to some node m, there is also an arc representing RC going from m to n). Figure 5c clarifies the notion that we may think of an instance of a two case case frame (such as M4) as both an arc and a node if we are willing to recalibrate the measurement of time it takes to follow one arc-relation to be the time it takes to follow two arcs. We can replace all instances of ISA in the path-based inference rules of Section 2 by the composition SUB-/SUP and still have valid rules except that we now have paths on the left of the "+" symbol.

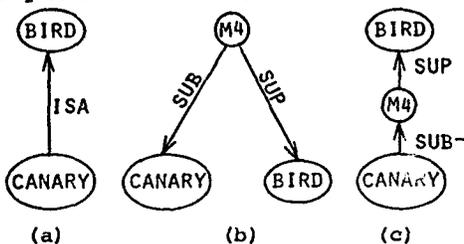


FIGURE 5: An illustration of the equivalence of an arc-relation to a two case case frame. a) Representing set membership as the ISA arc-relation. b) Representing set membership as a SUB-SUP case frame. c) Redrawing (b) so it looks like (a).

Let us, therefore, extend our syntax of path-based inference rules to allow a path of arc compositions on the left of the "+" symbol. The rule  $ISA + ISA^*$  states that whenever there is a path of ISA arcs from node n to node m, we can infer a "virtual" ISA arc directly from n to m which we may, if we wish, actually add to the network. Similarly, let the rule  $SUB-/SUP + (SUB-/SUP)^*$  state that whenever a path of alternating SUB- and SUP arcs goes from node n to node m, we can infer a "virtual" node with SUB to n and SUP to m which we may, if we wish, actually add to the network.

We now have a formalism for specifying path-based inference rules in a network formalism that represents binary conceptual relations by two case case frames. This would allow, for example, for a more unified representation in the SNIFFER system [3], in which node-based inference rules are implemented and built-in path based inference rules are used for set membership and set inclusion, both of which are represented only by arc-relations. The formalism presented here would allow set membership and set inclusion assertions to be represented by nodes, permitting other assertions to reference them, without giving up the efficiency of built-in routines to implement the set inclusion hierarchies.

We desire, however, a more general unification of path-based and node-based inferences. There are two basic routines used to implement node-based inferences (although specific implementations may differ). One is the match routine that is given a pattern node and finds instances of it in the network, and the other is the routine that interprets the quantifiers and connectives to carry out the actual deduction. The match routine can be enhanced to make use of path-based inference rules. Consider a typical match routine used in the deduction in Section 3 of A SUPPORTS C from the network of Figure 4 and the rule of Figure 3, and let us introduce the notation that if P is a path of arcs and n is a node, P[n] represents the set of nodes found by following the path P from the node n. In the example, the match routine was given the pattern P4 to match in the binding [r/SUPPORTS]. It was able to find M1 by intersecting  $CLASS^C[TRANSITIVE]$  with  $MEMBER^C[SUPPORTS]$ . Now, let us suppose that the path-based inference rule  $CLASS + CLASS / (SUB-/SUP)^*$  has been declared in such a way that the match routine could use it. The match routine would intersect  $MEMBER^C[SUPPORTS]$  with  $(CLASS / (SUB-/SUP)^*)^C[TRANSITIVE]$  and be able to find a virtual node asserting that SUPPORTS is TRANSITIVE even if a long chain of set inclusions separated them. The proposal, therefore, is this: any arc-relation in a semantic network may be defined in terms of a path-based inference rule which the match routine is capable of using when finding instances of pattern

nodes. This completes the general unification of path-based and node-based inference we desired. Since path-based inference is embedded in the match routine, while node-based inference requires the quantifier and connective interpreter, the difference is reminiscent of the difference between subconscious inference and conscious reasoning.

### 7. Application to Extensional Equivalence of Intensional Concepts

A basic assumption of semantic networks is that each concept is represented by a single node and that all information about a concept is reachable from its node. Yet, since Woods' discussion [20], most semantic network authors have agreed that a node represents an intensional, rather than an extensional concept. How should we handle the information that two different intensional concepts are extensionally equivalent?

Let us illustrate this by a story (entirely fictional). For the last year we have heard of a renowned surgeon in town, Dr. Smith, known for his brilliance and dexterity, who saved the life of the famous actress Maureen Gelt by a difficult heart transplant operation. Meanwhile, at several social gatherings, we have met someone by the name of John Smith, about five feet, six inches tall, black hair and beard, generally disheveled and clumsy. We now discover, much to our amazement that John Smith and Dr. Smith are one and the same! In our semantic network, we have one node for Dr. Smith connected to his attributes, and another for John Smith connected to his attributes. What are we to do? Although we now know that John Smith saved the life of Maureen Gelt and that Dr. Smith has black hair, surely we cannot retrieve that information as fast as that Dr. Smith is a surgeon and that John Smith is 5'6" tall. If we were to combine the two nodes by taking all the arcs from one node, tying them to the other and throwing away the first, we would lose this distinction. We must introduce an assertion, say an EQUIV-EQUIV case frame, that represents the fact that Dr. Smith and John Smith, different intensional concepts, are extensionally the same.<sup>1</sup> How are we to use this assertion?

Ignoring for the moment referentially opaque contexts ("We didn't know that John Smith was Dr. Smith."), how can we express the rule that if  $n$  EQUIV-EQUIV  $m$ , than anything true of  $n$  is true of  $m$ ? Our node based inference rules cannot express this rule because expressing "anything true of  $n$ " requires quantifying over those higher order case frame predicates such as AVO

<sup>1</sup>The psychological plausibility of this discussion is supported by the experiments of Anderson and Hastie [1] and of McNabb [9].

and MEMBER-CLASS. One possibility is to use lambda abstraction as Schubert does [12]. Each n-ary higher order predicate involving some node becomes a unary predicate by replacing that node by a lambda variable. Thus, "Dr. Smith saved Maureen Gelt's life" becomes an instance of the unary predicate  $\lambda(x)[x \text{ saved Maureen Gelt's life}]$  applied to Dr. Smith. Using a PRED-ARG case frame, it is easy to represent the rule

$$\forall x, y, z [\text{EQUIV-EQUIV}(x, y) \ \& \ \text{PRED-ARG}(x, z) \rightarrow \text{PRED-ARG}(y, z)].$$

The trouble with this solution is, how are we to retrieve this information as a fact about Maureen Gelt? Must we also store

$$\lambda(x)[\text{Dr. Smith saved } x\text{'s life}] \text{ (Maureen Gelt)?}$$

This duplication is unsatisfying. An alternative is to include in the path-based inference rule defining each arc-relation the path (EQUIV-EQUIV)\*. For example, AGENT + AGENT/(EQUIV-EQUIV)\*, and CLASS + CLASS/((EQUIV-EQUIV)\*/(SUB-/SUP)\*). Although this solution requires more rules than the lambda abstraction solution, and the rules look complicated, it avoids the duplication of the same assertion in different forms and the postulation of conceptual predicates such as  $\lambda(x)[x \text{ saved Maureen Gelt's life}]$ .

Hays' cognitive networks [4;5] include a scheme similar to the one proposed here. Each assertion about Dr. Smith would refer to a different node, each with an MST (manifestation) arc to a common node. This node would represent the intension of Dr. Smith, while the others represent Dr. Smith as surgeon, Dr. Smith as saviour of Maureen Gelt, etc. Presumably, when Hays' network learns of the identity of Dr. Smith with John Smith, a new node is introduced with MST arcs from both Dr. Smith and John Smith.<sup>2</sup> Dr. Smith and John Smith are then seen as two manifestations of the newly integrated Dr. John Smith. Hays presumably uses an MST\*/(MST<sup>C</sup>)\* path where we propose an (EQUIV-EQUIV)\* path.

Blocking referentially opaque contexts seems to require introducing *relational complement*. For any path  $P$  and nodes  $x$  and  $y$ , let  $xPy$  hold just in case a path  $P$  from  $x$  to  $y$  does not exist in the network. We might block referentially opaque contexts by including the domain or range restriction (OBJ-/VERB/MEMBER-/CLASS OPAQUE) in the arc definitions.

### 8. Application to the Explication of Inheritance

As was mentioned in Section 2, many

<sup>2</sup>Actually, Hays' networks have not yet been implemented, and I have been warned [R. Fritzson, personal communication] that the implementation may differ from what I have supposed.

semantic networks include inheritance (ISA) hierarchies. Often these are at best vague and at worst inconsistent. We propose that the inheritance properties of these hierarchies be clearly defined by path-based inference rules using the syntax we are presenting here or some other well defined syntax. We do not say that all systems should be able to input and interpret such rules, but only that authors use such rules to explain clearly to their readers how their hierarchies work.

Before this proposal is feasible, we must be able to handle two more situations. The first is the exception principle, first expressed by Raphael [11, p.85] and succinctly stated by Winograd as, "Any property true of a concept in the hierarchy is implicitly true of anything linked below it, unless explicitly contradicted at the lower level" [19, p.197]. To allow for this, let us introduce an *exception operator*. If  $P$  and  $Q$  are paths and  $x$  and  $y$  are nodes, let  $xP\backslash Qy$  hold just in case there is a path described by  $P$  from  $x$  to  $y$  and no path of equal or shorter length described by  $Q$  from  $x$  to  $y$ . To see that this suffices to handle the exception principle, consider the hierarchy of flying penguins. We have also taken the liberty of explicitly representing that CAN-FLY and CAN-NOT-FLY are negations of each other. The rule for inheritance in this hierarchy is

$$\text{PROP} \leftarrow (\text{ISA}^*/\text{PROP}) \backslash (\text{ISA}^*/\text{PROP}/\text{NOT}).$$

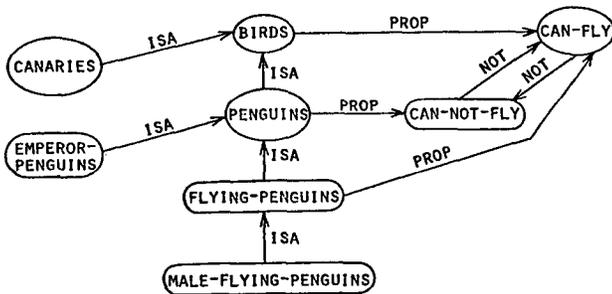


FIGURE 6: An ISA hierarchy illustrating the exception principle.

The other situation we must discuss is "almost transitive" relations such as SIBLING. SIBLING is certainly symmetric, but it cannot be transitive since it is irreflexive. Yet your sibling's sibling is your sibling as long as he/she is not yourself. This is what we mean by "almost transitive." Note that for any relation,  $R$ ,  $R \circ \bar{R} \circ R$  is the identity relation. Let us call it  $I$ . Then for any relation  $P$ , let  $PR$  be  $P \circ \bar{I}$ .  $PR$  is the *irreflexive restriction* of  $P$ . We can use this to define SIBLING as  $\text{SIBLING} \leftarrow (\text{SIBLING} \vee \text{SIBLING}^C) \circ PR$ .

We suggest that the syntax for path-based inference rules is now complete

enough to explicate the inheritance rules of various hierarchies. The complete syntax will be summarized in the next section

## 9. Summary

We have presented and compared two styles of inference in semantic networks, path-based inference and node-based inference. The former is more efficient, while the latter is more general. We showed the equivalence of an arc-relation to a two case case frame, and described how path-based inference could be incorporated into the match routine of a node-based inference mechanism, thereby combining the strengths of the two inference styles. We discussed the use of equivalence paths to represent the extensional equivalence of intensional concepts. Finally, we urged authors of inheritance hierarchies to explicate their hierarchies by displaying the path-based inference rules that govern inheritance in them.

We also presented a syntax for path-based inference rules which can be summarized as follows:

1. A path is either an arc-relation or a path as described in part 2 enclosed in parentheses. Parentheses may be omitted whenever an ambiguity does not result.
2. If  $P$  and  $Q$  are paths and  $x$ ,  $y$ , and  $z$  are nodes, paths may be formed as follows:
  - a. Converse: if  $P$  is a path from  $x$  to  $y$ ,  $\overleftarrow{P}$  is a path from  $y$  to  $x$ .
  - b. Composition: if  $P$  is a path from  $x$  to  $z$  and  $Q$  is a path from  $z$  to  $y$ ,  $P/Q$  is a path from  $x$  to  $y$ .
  - c. Composition zero or more times: If  $P$  composed with itself zero or more times describes a path from  $x$  to  $y$ ,  $P^*$  is a path from  $x$  to  $y$ .
  - d. Composition one or more times: If  $P$  composed with itself one or more times is a path from  $x$  to  $y$ ,  $P^+$  is a path from  $x$  to  $y$ .
  - e. Union: If  $P$  is a path from  $x$  to  $y$  or  $Q$  is a path from  $x$  to  $y$ ,  $P \vee Q$  is a path from  $x$  to  $y$ .
  - f. Intersection: If  $P$  is a path from  $x$  to  $y$  and  $Q$  is a path from  $x$  to  $y$ ,  $P \& Q$  is a path from  $x$  to  $y$ .
  - g. Complement: If there is no path  $P$  from  $x$  to  $y$ ,  $\overline{P}$  is a path from  $x$  to  $y$ .
  - h. Irreflexive restriction: If  $P$  is a path from  $x$  to  $y$  and  $x \neq y$ ,  $P^R$  is a path from  $x$  to  $y$ .
  - i. Exception: If  $P$  is a path from  $x$  to  $y$  and there is no path  $Q$  of length equal to or less than the length of  $P$ ,  $P \backslash Q$  is a path from  $x$  to  $y$ .
  - j. Domain restriction: If  $P$  is a

path from  $x$  to  $y$  and  $Q$  is a path from  $x$  to  $z$ ,  $(Qz)P$  is a path from  $x$  to  $y$ .

k. Range restriction: If  $P$  is a path from  $x$  to  $y$  and  $Q$  is a path from  $y$  to  $z$ ,  $P(Qz)$  is a path from  $x$  to  $y$ .

3. A path-based inference rule is of the form  $\langle \text{defined path} \rangle + \langle \text{defining path} \rangle$  where  $\langle \text{defining path} \rangle$  is any path described by parts 1 or 2 above, and  $\langle \text{defined path} \rangle$  is either a) a single arc-relation, or b) a composition of  $n$  arc relations for some fixed  $n$ , i.e. using only "/", not "\*" or "+". The rule is interpreted to mean that if the  $\langle \text{defining path} \rangle$  goes from some node  $x$  to some node  $y$  then: a) the arc that is the  $\langle \text{defined path} \rangle$  is inferred to exist from  $x$  to  $y$ ; b) the  $n$  arcs that are the  $\langle \text{defined path} \rangle$  and  $n-1$  new intermediate nodes are inferred to exist from  $x$  to  $y$ .

#### References

1. Anderson, J. and Hastie, R. Individuation and reference in memory: proper names and definite descriptions. Cognitive Psychology 6, 4 (October, 1974), 495-514.
2. Collins, A.M. and Quillian, R. Retrieval time from semantic memory. J. of Verbal Learning and Verbal Behavior 8, (1969), 240-247.
3. Fikes, R. and Hendrix, G. A network-based knowledge representation and its natural deduction system. Proc. Fifth Int. Jt. Conf. on Artificial Intelligence, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, 1977, 235-246.
4. Hays, D.G. Cognitive Structures. unpublished ms. Dept. of Linguistics, SUNY at Buffalo, Amherst, NY.
5. Hays, D.G. Types of processes on cognitive networks. In L.S. Olschki, ed. Mathematical Linguistics, Frieze, Pisa, 1977, 523-532.
6. Hendrix, G.G. Expanding the utility of semantic networks through partitioning. Advance Papers of the Fourth Int. Jt. Conf. on Artificial Intelligence, MIT AI Laboratory, Cambridge, MA, 1975, 115-121.
7. Kay, M. The MIND system. In R. Rustin, ed. Natural Language Processing, Algorithmics Press, New York, 1973, 155-188.
8. McDermott, D. Artificial intelligence meets natural stupidity. SIGART Newsletter, 57 (April, 1976), 4-9.
9. McNabb, S.D. The effects of encoding strategies and age on the memory representation for sentences containing proper names and definite descriptions. Report No. 77-3, Indiana Mathematical Psychology Program, Department of Psychology, Indiana University, Bloomington, IN. August, 1977.
10. Quillian, M.R. Semantic memory. In M. Minsky, ed. Semantic Information Processing, MIT Press, Cambridge, MA, 1968, 227-270.
11. Raphael, B. SIR: semantic information retrieval. In M. Minsky, ed. Semantic Information Processing, MIT Press, Cambridge, MA., 1968, 33-145.
12. Schubert, L.K. Extending the expressive power of semantic networks. Artificial Intelligence 7, 2 (Summer, 1976), 163-198.
13. Schwarcz, R.M., Burger, J.F., and Simmons, R.F. A deductive question-answerer for natural language inference. CACM 13, 3 (March, 1970), 167-183.
14. Shapiro, S.C. A net structure for semantic information storage, deduction and retrieval. Proc. Second Int. Jt. Conf. on Artificial Intelligence, The British Computer Society, London, 1971, 512-523.
15. Shapiro, S.C. Representing and locating deduction rules in a semantic network. Proc. Workshop on Pattern-Directed Inference Systems. In SIGART Newsletter, 63 (June, 1977), 14-18.
16. Shapiro, S.C. Techniques of Artificial Intelligence. D. Van Nostrand, New York, 1979.
17. Shapiro, S.C. The SNePS semantic network processing system. In N. Findler, ed. Associative Networks -- The Representation and Use of Knowledge in Computers, Academic Press, New York, in press.
18. Shapiro, S.C. and Woodmansee, G.H. A net structure based relational question answerer: description and examples. Proc. Int. Jt. Conf. on Artificial Intelligence, The MITRE Corp., Bedford, MA., 1969, 325-346.
19. Winograd, T. Frame representations and the declarative/procedural controversy. In D.G. Bobrow and A. Collins, eds. Representation and Understanding, Academic Press, Inc., New York, 1975, 185-210.
20. Woods, W.A. What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. Collins, eds. Representation and Understanding, Academic Press, Inc., New York, 1975, 35-82.