Formal Foundations of an Intensional Propositional Semantic Network*

Stuart C. Shapiro
Department of Computer Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-7022
(716) 636-3182
shapiro@cs.buffalo.edu

February 2, 1989

1 Introduction

In this paper, I will attempt to lay out the formal foundations of the SNePS knowledge representation formalism [17] as an intensional propositional semantic network (see [8, 21]). In so doing, I try to respond to [6, p. 5]:

> in taking SNePS seriously, in particular in attempting to present a semantic account which honors some of the intuitions of its creator(s), one is led into rather interesting, if slightly forbiding, logical terrain. Of course, enough has been said to prove beyond a shadow of a doubt that there are significant open problems to be solved before a fully adequate account can be given.

The discussion in this paper will be compatible with that in [21], but will not dwell on the specific representations used to implement SNePS/CASSIE, a particular computational (model of a) mind. Rather, this paper will deal with the logical foundations underlying SNePS/CASSIE and any other agent representable in SNePS. The issues to be addressed include: what are the semantics of SNePS nodes; how can we understand the principle that the entire network connected to a node determines what the node represents; what do

*This work was supported in part by the National Science Foundation under Grant IRI-8610517, and in part by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB DC 20332 under Contract No. F30602-85-C-0008, which supports the Northeast Artificial Intelligence Consortium (NAIC).

variable nodes represent; what is the logic of the Belief predicate on propositions represented by nodes; how is the logic affected by the presence of multiple arcs with the same label emanating from a single node.

In discussing the intensionality of representation in SNePS two notions should particularly be kept in mind. First, as has been noted by several authors (*e.g.* [1, 13, 15]), the objects represented in a mind are not limited to a fixed, finite domain—when reading or talking, new entities (sometimes called "discourse entities"), possibly fictional or even logically impossible, can be introduced and beliefs about them can be formed. Second, arguments between people often come down, not to which statements are true, and which false, but to how one person's concepts differ from the other's.

2 The Agent

I will present this analysis in terms of an "agent." The agent has beliefs and performs actions (see [11]). Such an agent is (a model of) a cognitive agent. SNePS/CASSIE [21] is one such agent, but others could be built using SNePS as the knowledge representation system.

Among the actions the agent can perform is the new believing of a previously not believed proposition. I do not want to formalize the agent in such a way that it is logically omniscient [5]. Instead, at any time, the agent will believe only those logical consequents of its beliefs that it has come to believe by "consciously" performing the act of believing them (or those it has come to believe after "subconsciously" believing them, and then thinking of them, see Section 8).

3 The Domain of Interpretation

SNePS nodes are terms of a formal language. The interpretation of a node is an object in the domain of interpretation, $\mathcal{D}$. What we mean $\mathcal{D}$ to be has caused some confusion. We mean it to be the set of all possible entities that the agent can have beliefs about. We have previously characterized $\mathcal{D}$ to be the set of all possible "concepts" or "intensional objects." Rapaport [14] has pointed out that $\mathcal{D}$ is what Meinong referred to as "Aussersein." Certainly, $\mathcal{D}$ is not the real world, since cognitive agents can have beliefs about fictional, hypothetical, and even logically impossible entities.

In understanding $\mathcal{D}$, we want to avoid the apparant contradictions of seemingly compatible researchers like Appelt and Kronfeld [1, p. 643]. They say,

> It is important to note that an individuating set is the result of an agent's beliefs, not a mirror of what is actually the case. It is certainly possible to have an individuating set of terms that do not denote anything real (*e.g.*, a child's representation of Santa Claus) ... If all is well, and there actually is an object corresponding to all the terms in the set, we say that the referent of the latter is that object. If all is not well, the individuating set has no referent and the agent is simply mistaken or confused.

By "individuating set," they mean the same thing as a set of SNePS nodes all connected with *EQUIV* relations (see [8]). I certainly agree with the first two sentences quoted above. However, I don't think that having an individuating set denoting Santa Claus means that all is not well, and that the agent is "mistaken or confused." The domain of interpretation $\mathcal{D}$ must include Santa Claus, and even the square circle and Russell's Set.

In this paper, I shall term the members of $\mathcal{D}$ "entities": "A being; esp., a thing which has reality and distinctness of being either in fact or for thought; as, to view the state as an entity" [9, p. 275]. Every SNePS node denotes an entity. If n is a SNePS node, $[[n]]$ will denote the entity represented by n.

4 Types of Entities

We distinguish four types of entities: individuals, propositions, acts, and rules. Propositions are characterized by being the kind of entities an agent may or may not believe. Acts (see [20]) are characterized by being the kind of enities an agent may or may not intend to perform. Rules are both propositions and acts. In order for a rule to "fire," it must be believed, the agent must intended to apply it, and its (appropriate) antecedents must be believed. When a rule fires, the agent forms the intention of believing its consequents. Intending to apply a rule is what is called "activating" a rule in [19]. These notions will be formalized below.

Individuals include everything that is neither a proposition nor an act, *i.e.* that is neither the kind of entity that can be believed, nor the kind of entity that an agent could intend to perform. Thus, individuals include not only traditional individuals, but also classes, properties, relations, etc.

SNePS nodes are typed according to the type of entities they represent. Thus, there are four types of nodes—individual nodes, proposition nodes, act nodes, and rule nodes. As research proceeds, there may be a need to distinguish other types of entities, but at this time propositions, acts, rules, and individuals are the only types we have found a need for.

5 Meta-Predicates

In formalizing SNePS, we need a set of meta-predicates. These will not necessarily be represented in SNePS, although if the agent, itself, were engaged in the appropriate philosophical reflection, it could conceive of them. The meta-predicates we will need include: *Conceive*, *Believe*, *Intend*, *WonderIf*, *IntendToBelieve*, and =. Others will be introduce subsequently.

Letting n, n_1, and n_2 be meta-variables ranging over nodes, p be a meta-variable ranging over proposition nodes, and a be a meta-variable ranging over act nodes, the semantics of the meta-predicates listed above are:

$Conceive(n)$ means that the node n is actually constructed in the SNePs network, and that the agent has conceived of, or thought of, or thought about $[[n]]$. *Conceive* is similar to, but different than Fagin and Halpern's awareness functions [2]. They gloss $A_i\phi$ as, "'i is aware of ϕ,' 'i is able to figure out the truth of ϕ,' or even (when reasoning about knowledge bases) 'i is able to compute the truth of ϕ within time T.'" Here, $Conceive(n)$ may be true without the agent's being able to figure out the truth of $[[n]]$.

$Believe(p)$ means that the agent believes the proposition $[[p]]$. (In which case, we say that p is an *asserted* node.)

$Intend(a)$ means that the agent will attempt to perform the act $[[a]]$. (*I.e.* the agent will perform the appropriate motor functions, but might be blocked by the world.)

$WonderIf(p)$ means that the agent is wondering or has wondered if $[[p]]$. *WonderIf* will be used to formalize backward-chaining.

$IntendToBelieve(p)$ means that at some time in the future, $Believe(p)$ will be true.

$n_1 = n_2$ means that n_1 and n_2 are the same, identical, node.

Only conceived of entities may be wondered about, believed, or intended. This is captured in the following Axioms:

Axiom 1 $Believe(p) \Rightarrow Conceive(p)$

Axiom 2 $Intend(a) \Rightarrow Conceive(a)$

Axiom 3 $WonderIf(p) \Rightarrow Conceive(p)$

Axiom 4 $IntendToBelieve(p) \Rightarrow Conceive(p)$

6 Arcs and Relations

SNePS nodes are connected to each other by labelled, directed arcs. The labels are drawn from the set of *SNePS Relations*, which can be added to by the user of SNePS in the design of a particular agent. Isolated nodes cannot be constructed in SNePS; neither can cycles of arcs.

7 Types of Nodes

Besides the categorization of nodes into individual nodes, proposition nodes, act nodes, and rule nodes, nodes can also be categorized into base nodes, molecular nodes, variable nodes, and pattern nodes. The two categorizations of nodes are orthogonal, so there are sixteen types of nodes. As a heuristic aid to understanding, base nodes approximately correspond to individual constants in a standard Predicate Logic, variable nodes to variables, molecular nodes to closed sentences, and pattern nodes to open sentences. However, remember that all nodes are terms in SNePS.

7.1 Base Nodes

Base nodes have no arcs emanating from them. Each base node represents some entity of the appropriate type. An individual base node represents an individual entity, a propositional base node represents a proposition, an act base node represents an act, and a rule base node represents a rule. No two base nodes represent the same entity. This is the Uniqueness Principle of [8] for base nodes.

Since no isolated nodes are allowed in SNePS, every base node must have at least one arc pointing to it. The entity represented by a base node is determined by the structure of the network it is connected to. For example, if bi_1 is a base individual node, and it is connected to other nodes in such a way that the agent believes that $[[bi_1]]$ is a dog, then bi_1 represents some dog. If the agent believes nothing else about $[[bi_1]]$, and it also has some individual base node bi_2, and its only belief about $[[bi_2]]$ is that it is a dog, then $[[bi_1]]$ and $[[bi_2]]$ are two intensionally different dogs. Note, however, that nodes are what [1] calls intensional object representations. If the agent believed that the $EQUIV$ relation of [8] held between $[[bi_1]]$ and $[[bi_2]]$

(equivalently, if bi_1 and bi_2 were in the same individuating set), then the agent would believe that $[[bi_1]]$ and $[[bi_2]]$ are the same dog, though conceived of slightly differently.

Axiom 5 (Contrapositive of Uniqueness Principle) $n_1 \neq n_2 \Rightarrow [[n_1]] \neq [[n_2]]$[1]

As additional beliefs are added to the agent about some node, the entity the node represents changes slightly. For example, if the agent's only belief about $[[bi_1]]$ is that it is a dog, then bi_1 represents some specific, but vague dog. If the agent now performs the act of believing that $[[bi_1]]$ is white, bi_1 now represents some specific, but otherwise vague, white dog.

7.2 Variable Nodes

Like base nodes, variable nodes have no arcs emanating from them, and come in four types—individual variables, propositional variables, act variables, and rule variables. In addition, a variable must have a restriction, which is a propositional pattern node for which the variable node occurs free (to be formally explained below). If v is a variable node, and its restriction represents the proposition $P([[v]])$, then $[[v]]$ is the arbitrary individual satisfying P (see [4]). For example, if P is the proposition that its argument is a dog, then v represents the arbitrary dog. If another variable, v_2, also has a restriction representing $P([[v_2]])$, then $[[v_2]]$ is an arbitrary dog other than $[[v]]$ (see [18] and [3]).

For every restriction in the network, there is an ordered sequence of variable nodes. Rules must use these restricted variables in order. If some rule uses some restriction, P, twice (*i.e.* the rule is about two arbitrary individuals satisfying P), and the variables of that restriction are $v_1^P, v_2^P, \ldots$, then the first, or outermost use of P must restrict v_1^P, and the second must restrict v_2^P, etc.

7.3 Binding Relations

One or more relations in the set of SNePS Relations (which label arcs) are *binding relations*. In this paper, I will discuss two: `forall` and `exists`[2]. These approximately correspond to quantifiers in standard Predicate Logic. A binding relation labels an arc that goes from a proposition node to a variable node, that, thereby, is bound in the proposition node. We will say that an arc labelled with a binding relation is a binding arc.

[1] Note that "=" is overloaded to represent both identity of nodes and of entities.

[2] The sense of `exists` is "exists in $\mathcal{D}$," not in the real world. This is also its meaning in, "...when eight years of Reagan rule produces 3,000,000 homeless people who didn't exist before" [Howard Fast interviewed on Spoken Arts, WBFO Radio, Jan. 26, 1989].

7.4 Molecular and Pattern Nodes

A molecular node has one or more labelled, directed arcs emanating from it, each labelled by a relation in the set of SNePS Relations, and each going to another node. Two or more arcs may go from one node to one other node, as long as each arc is labelled with a different label. A molecular node, n, must satisfy the condition that any path from n to a variable node, v, must go through a node, m (possibly n itself), from which a binding arc goes to v.

Alternatively (see [10]), we may say that a molecular node is a non-empty set of *cables*, where each cable is an ordered pair of a SNePS Relation and a non-empty set of other nodes. More precisely, if $r_1, \ldots, r_k$ are distinct relations and $ns_1, \ldots, ns_k$ are non-empty sets of nodes, then $\{\langle r_1, ns_1\rangle, \ldots, \langle r_k, ns_k\rangle\}$ is a *cableset*, and is either a molecular node or a pattern node.

If $\langle r, ns\rangle$ is a cable in a cableset cs, n is a node, and $n \in ns$, then we say that $\langle r, n\rangle$ is a *wire* in the cable $\langle r, ns\rangle$, and in the cableset cs, and write $\langle r, n\rangle \in \langle r, ns\rangle$ and $\langle r, n\rangle \in cs$, overloading "$\in$". A *relation-path* from the node n to the node m is a sequence, $n_1, r_1, \ldots, n_k, r_k, n_{k+1}$ where the n_i are nodes (cablesets), the r_i are SNePS Relations, $n = n_1$, $m = n_{k+1}$, and for each i, $\langle r_i, n_{i+1}\rangle$ is a wire in n_i. We say that the relation-path $n_1, r_1, \ldots, n_k, r_k, n_{k+1}$ goes *through* $n_i, 1 \leq i \leq k$.

If a node m contains a wire $\langle r, v\rangle$ such that r is a binding relation and v is a variable node, then we say that m *binds* v.

If a relation-path goes from a node n to a variable node v, then if every relation-path from n to v goes through a node that binds v, we say that v is *bound* in n. However, if there is at least one relation-path from n to v that does not go through a node that binds v, then we say that v is *free* in n.

A cableset, then, is a molecular node if and only if no variable node is free in it. A cableset with at least one free variable in it is a pattern node.

The use of *sets* of cables and *sets* of nodes is significant, *e.g.*

$$\{\langle r_1, \{n_1, n_2\}\rangle, \langle r_2, \{n_3, n_4\}\rangle\} = \{\langle r_2, \{n_4, n_3\}\rangle, \langle r_1, \{n_2, n_1\}\rangle\}.$$

However, a cableset and a proper subset of it are different nodes, and if two cablesets differ only in that one contains the cable $\langle r, ns_1\rangle$ while the other contains the cable $\langle r, ns_2\rangle$ and the sets ns_1 and ns_2 are different, then the two cablesets are non-identical nodes. Notice that this means that it makes no sense to add a new arc emanating from an existing node (*i.e.*, a new wire to a cableset, while having it remain the same cableset). Also notice that a node is determined by the arcs emanating from it, not by the arcs pointing into it.

Molecular and pattern nodes may represent either propositions, acts, or individuals. Which of these three a given cableset represents depends on, and is determined by, the set of relations in the cableset.[3] Proposition cablesets roughly correspond to propositions in standard Predicate Logic, while act and individual cablesets (which we sometimes call "structured individuals") roughly correspond to functional terms. (Since proposition nodes are also terms, they all roughly correspond to functional terms.) Like their counterparts, cablesets get their semantics from the user—the person who designs a particular SNePS agent. The semantics also depends on the set of relations in the cableset, which, therefore roughly corresponds to a predicate or function.

As examples we will use throughout the rest of this paper, in SNePS/CASSIE [21], `member`, `class`, `subclass`, and `superclass` are SNePS Relations, in [12], `pred` and `word` are, and in [20] `action` and `object1` are. The semantics given in those papers include (paraphrased):

- a node of the form $\{\langle \texttt{member}, \{i_1\}\rangle, \langle \texttt{class}, \{i_2\}\rangle\}$ represents the proposition that the entity $[[i_1]]$ is a member of the class $[[i_2]]$.
- a node of the form $\{\langle \texttt{subclass}, \{i_3\}\rangle, \langle \texttt{superclass}, \{i_4\}\rangle\}$ represents the proposition that the class $[[i_3]]$ is a subclass of the class $[[i_4]]$.
- a node of the form $\{\langle \texttt{pred}, \{s\}\rangle, \langle \texttt{word}, \{w\}\rangle\}$ represents the individual surface string consisting of the word $[[w]]$ concatenated onto the end of the surface string $[[s]]$.
- a node of the form $\{\langle \texttt{action}, \{a\}\rangle, \langle \texttt{object1}, \{o\}\rangle\}$ represents the act of performing the action $[[a]]$ on the entity $[[o]]$.

The Uniqueness Principle for molecular and pattern nodes is enforced in virtue of the fact that different cablesets are different nodes and represent different entities.

8 Path-Based Inference

Although different cablesets represent different entities, an asserted node may give rise to several beliefs depending on the rest of the network it is connected with.

Informally, *path-based* inference [16, 22] is a means of inferring a virtual arc from a node n to a node m when there is a certain path from n to m.

[3] SNePS, as currently implemented, does not actually type nodes as representing propositions, acts, or individuals, but a cableset can be so characterized, as stated, as long as the user supplies a consistent semantics to various sets of relations.

For example, using the relations mentioned above, we may specify the inheritance of class membership with the SNePS User Language (SNePSUL) command,

```
(define-path class (compose class (kstar (compose subclass- ! superclass))))
```

Informally, this says that a virtual `class` arc may be inferred from a node n to a node m whenever a path of arcs consisting of a `class` arc, followed by zero or more occurrences of the path consisting of a `subclass` arc (followed backwards) followed by a `superclass` arc goes from n to m, as long as each `superclass` arc emanates from an asserted node (one representing a believed proposition). There are twelve path formation operators like `compose` and `kstar` in SNePSUL including `converse`, `kplus`, `or`, and `and`. Path-based reasoning was described in [16] as being a kind of "subconscious" reasoning. This is captured in the formalization of path-based reasoning which follows.

For a relation r and a path p, let $Pbr(r,p)$ mean that the path based inference rule `(define-path` r p`)` has been entered into the system. For nodes n and m, let $HavePath(n,p,m)$ mean that the path p is in the network going from n to m. For a node n, let $Vbelieve(n)$ mean that the agent acts as if it believes $[[n]]$, although $Conceive(n)$ is not necessarily true. $Vbelieve$ is a kind of subconscious belief required in order to make path-based reasoning subconscious.[4] Finally, for a cableset cs and a wire w, let $cs \cup w$ be the cableset that contains all wires that cs contains, plus w also. The following axioms specify when $Vbelieve(n)$ holds:

Axiom 6 $Believe(n) \Rightarrow Vbelieve(n)$

Axiom 7 $Vbelieve(n)\&Pbr(r,p)\&HavePath(n,p,m) \Rightarrow Vbelieve(n \cup \langle r,m\rangle)$

Let $Reduce(m_1, m_2)$ be a predicate that holds if m_1 and m_2 are cablesets and the set of wires in m_2 is a subset of the set of wires in m_1. In that case, we will say that m_2 is a *reduction* of m_1.

Axiom 8 $Reduce(m_1, m_2) \Leftrightarrow \forall(w)[w \in m_2 \Rightarrow w \in m_1]$

Other subconscious beliefs of the agent are in propositions represented by reductions of nodes representing believed propositions.

Axiom 9 $VBelieve(p_1)\&Reduce(p_1,p_2) \Rightarrow Vbelieve(p_2)$

[4] $Vbelieve$ is a kind of implicit belief, but it is not as powerful as Levesque's implicit belief predicate L [7]. $L\alpha$ is true whenever α logically follows from the agent's explicit or implicit beliefs, but as will be seen, $Vbelieve(n)$ is true only when n follows from explicitly believed propositions, conceived of entities, and explicitly entered path-based inference rules, which are a very restricted form of rule.

A subconscious belief in some proposition can lead to a conscious belief in the proposition if the agent conceives of the proposition:

Axiom 10 $Vbelieve(p) \& Conceive(p) \Rightarrow Believe(p)$

Let $Pbclosure(n, m)$ mean that n contains all the wires in m and all the virtual wires that can be inferred to be in m by virtue of path-based inference rules:

Axiom 11

$$\begin{aligned} Pbclosure(n, m) \quad \Leftrightarrow \quad & Reduce(n, m) \\ & \& \forall(r, p)[Pbr(r, p) \& HavePath(m, p, m\prime) \Rightarrow \langle r, m\prime \rangle \in n] \\ & \& \forall(w)[w \in n \Rightarrow w \in m \vee \exists(r, p)[Pbr(r, p) \& HavePath(m, p, m\prime) \& w = \langle r, m\prime \rangle]] \end{aligned}$$

If the agent believes (at least subconsciously) a proposition, it will subconsciously believe the proposition represented by the Pbclosure of the node that represents that proposition.

Lemma 1 $Vbelieve(m) \& Pbclosure(n, m) \Rightarrow Vbelieve(n)$

Proof Follows by induction from Axioms 7 and 11.

If the agent conceives of a proposition represented by a reduction of the pbclosure of an asserted node, the agent will believe that proposition:

Theorem 1 $Believe(p_1) \& Pbclosure(p_2, p_1) \& Reduce(p_2, p_3) \& Conceive(p_3) \Rightarrow Believe(p_3)$

Proof: Follows from Axiom 6, Lemma 1, Axiom 9, and Axiom 10.

For example, assume again the SNePS Relations `member`, `class`, `subclass`, and `superclass`, and the path-based inference rule shown above. Then, $\{\langle \texttt{member}, \{\texttt{rover}, \texttt{snoopy}\}\rangle, \langle \texttt{class}, \{\texttt{dog}, \texttt{male}\}\rangle\}$ represents the proposition that [[`rover`]] and [[`snoopy`]] are [[`dog`]]s and [[`male`]]s, and $\{\langle \texttt{subclass}, \{\texttt{dog}\}\rangle, \langle \texttt{superclass}, \{\texttt{animal}\}\rangle\}$ represents the proposition that [[`dog`]]s are [[`animal`]]s. In that case, belief in the two propositions:

$$[[\{\langle \texttt{member}, \{\texttt{rover}, \texttt{snoopy}\}\rangle, \langle \texttt{class}, \{\texttt{dog}, \texttt{male}\}\rangle\}]]$$
$$[[\{\langle \texttt{subclass}, \{\texttt{dog}\}\rangle, \langle \texttt{superclass}, \{\texttt{animal}\}\rangle\}]]$$

entails belief in any of the following (different) propositions that the agent conceives of:

$$[[\{\langle \texttt{member}, \{\texttt{rover}\}\rangle, \langle \texttt{class}, \{\texttt{dog}\}\rangle\}]],$$

$$[[\{\langle \texttt{member}, \{\texttt{rover}\}\rangle, \langle \texttt{class}, \{\texttt{male}\}\rangle\}]],$$

$$[[\{\langle \texttt{member}, \{\texttt{rover}\}\rangle, \langle \texttt{class}, \{\texttt{animal}\}\rangle\}]],$$

$$[[\{\langle \texttt{member}, \{\texttt{snoopy}\}\rangle, \langle \texttt{class}, \{\texttt{dog}\}\rangle\}]],$$

$$[[\{\langle \texttt{member}, \{\texttt{snoopy}\}\rangle, \langle \texttt{class}, \{\texttt{male}\}\rangle\}]],$$

$$[[\{\langle \texttt{member}, \{\texttt{snoopy}\}\rangle, \langle \texttt{class}, \{\texttt{animal}\}\rangle\}]].$$

Example Run

The following is the output of an interaction with SNePS, edited only to eliminate extra blank lines and the list of nodes returned by the `describe` command, and to add comments (in italics). Before this section of the interaction, the relations were declared and the path-based inference rule shown above was entered. The SNePSUL prompt is "*". `build` is the command to construct a node in the network, and thereby to make the agent conceive of the entity represented by the built node. `assert` builds a node and makes it asserted, thereby causing the agent to believe the proposition represented by the node. `describe` is a command to print a Lisp-like description of a node, so the reader can see its cableset. Symbols of the form M*n*, where *n* is an integer, are the names of the nodes. The names of asserted nodes are printed with "!" appended.

```
*(describe (build subclass dog superclass animal))
(M1 (SUBCLASS DOG) (SUPERCLASS ANIMAL)) ; M1 is built, but not asserted.
CPU time :  0.20 GC time :  0.00

*(describe (assert superclass animal subclass dog)) ; order of cables doesn't matter.
(M1!  (SUBCLASS DOG) (SUPERCLASS ANIMAL)) ; This is M1 again, now asserted.
CPU time :  0.08 GC time :  0.00

*(describe (assert member (rover snoopy) class (dog male)))
(M2!  (CLASS DOG MALE) (MEMBER ROVER SNOOPY)) ; built and asserted.
CPU time :  0.22 GC time :  0.00

*(describe (build member rover class male))
(M3!  (CLASS MALE) (MEMBER ROVER)) ; A restriction of M2!, therefore asserted
CPU time :  0.13 GC time :  0.00

*(describe (build member snoopy class animal))
(M4!  (CLASS ANIMAL) (MEMBER SNOOPY)) ; restriction of pbclosure of M2!, therefore asserted
CPU time :  0.18 GC time :  0.00
```

9 Concluding Remarks

This is a paper in progress. I have discussed the semantics of SNePS nodes, various types of entities and nodes, the cableset definition of nodes, and the logic of path-based inference. Still to come are discussions of acts, substitutions, subsumption, node-based rules, node-based inference, and acting rules.

References

[1] D. Appelt and A. Kronfeld. A computational model of referring. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 640–647, Los Altos, CA, 1987. Morgan

Kaufmann.

[2] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34(1):39–76, December 1987.

[3] S. Fahlman. *NETL: A System for Representing and Using Real-World Knowledge.* MIT Press, Cambridge, MA, 1979.

[4] K. Fine. A defence of arbitrary objects. *Proceedings of the Aristotelian Society*, Supp. Vol. 58:55–77, 1983.

[5] J. Hintikka. Impossible possible worlds vindicated. *Journal of Philosophical Logic*, 4:475–484, 1975.

[6] D. J. Israel. Interpreting network formalisms. In N. Cercone, editor, *Computational Linguistics*, pages 1–13. Pergammon Press, Oxford, 1983.

[7] H. J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the National Conference on Artificial Intelligence*, pages 198–202, Los Altos, CA, 1984. AAAI, Morgan Kaufmann.

[8] A. S. Maida and S. C. Shapiro. Intensional concepts in propositional semantic networks. *Cognitive Science*, 6(4):291–330, October–December 1982. Reprinted in R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, CA, 1985, 170–189.

[9] Merriam-Webster. *Webster's New Collegiate Dictionary.* G. & C. Merriam Co., Springfield, MA, 1961.

[10] E. J. M. Morgado. *Semantic Networks as Abstract Data Types.* PhD thesis, Technical Report 86–19, Department of Computer Science, SUNY at Buffalo, Buffalo, NY, 1986.

[11] E. J. M. Morgado and S. C. Shapiro. Believing and acting: A study of meta-knowledge and meta-reasoning. In *Proceedings of EPIA-85 "Encontro Portugues de Inteligencia Artificial"*, pages 138–154, Oporto, Portugal, 1985.

[12] J. G. Neal and S. C. Shapiro. Knowledge-based parsing. In L. Bolc, editor, *Natural Language Parsing Systems*, pages 49–92. Springer-Verlag, Berlin, 1987.

[13] W. J. Rapaport. Meinongian theories and a russellian paradox. *Noûs*, 12:153–80, 1978.

[14] W. J. Rapaport. Meinongian semantics for propositional semantic networks. In *Proc. 23rd Annual Meeting of the Association for Computational Linguistics*, pages 43–48, 1985.

[15] W. J. Rapaport. Logical foundations for belief representation. *Cognitive Science*, 10:371–422, 1986.

[16] S. C. Shapiro. Path-based and node-based inference in semantic networks. In D. L. Waltz, editor, *Tinlap-2: Theoretical Issues in Natural Languages Processing*, pages 219–225. ACM, New York, 1978.

[17] S. C. Shapiro. The SNePS semantic network processing system. In N. V. Findler, editor, *Associative Networks: The Representation and Use of Knowledge by Computers*, pages 179–203. Academic Press, New York, 1979.

[18] S. C. Shapiro. Symmetric relations, intensional individuals, and variable binding. *Proceedings of the IEEE*, 74(10):1354–1363, October 1986.

[19] S. C. Shapiro. Processing, bottom-up and top-down. In S. C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 779–785. John Wiley & Sons, New York, 1987.

[20] S. C. Shapiro. Representing plans and acts. In *Proceedings of the Third Annual Workshop on Conceptual Graphs*, pages 3.2.7-1–3.2.7-6, Menlo Park, CA, 1988. AAAI.

[21] S. C. Shapiro and W. J. Rapaport. SNePS considered as a fully intensional propositional semantic network. In N. Cercone and G. McCalla, editors, *The Knowledge Frontier*, pages 263–315. Springer-Verlag, New York, 1987.

[22] R. Srihari. Combining path-based and node-based reasoning in SNePS. Technical Report 183, Department of Computer Science, SUNY at Buffalo, Buffalo, NY, 1981.