The SNePS Family*

Stuart C. Shapiro and William J. Rapaport
Department of Computer Science
and Center for Cognitive Science
226 Bell Hall
Buffalo, NY 14260-7022
(716) 636-3180
shapiro@cs.buffalo.edu rapaport@cs.buffalo.edu

September 11, 1990

1 Motivation

SNePS, the Semantic Network Processing System [45, 54], has been designed to be a system for representing the beliefs of a natural-language-using intelligent system (a "cognitive agent"). It has always been the intention that a SNePS-based "knowledge base" would ultimately be built, not by a programmer or knowledge engineer entering representations of knowledge in some formal language or data entry system, but by a human informing it using a natural language (NL) (generally supposed to be English), or by the system reading books or articles that had been prepared for human readers. Because of this motivation, the criteria for the development of SNePS have included: it should be able to represent anything and everything expressible in NL; it should be able to represent generic, as well as specific information; it should be able to use the generic and the specific information to reason and infer information implied by what it has been told; it cannot count on any particular order among the pieces of information it is given; it must continue to act reasonably even if the information it is given includes circular definitions, recursive rules, and inconsistent information.

*This is a preprint version (missing some figures) of S. C. Shapiro and W. J. Rapaport, The SNePS family. *Computers & Mathematics with Applications 23,* 2–5 (January–March, 1992), 243–275. Reprinted in F. Lehmann, Ed. *Semantic Networks in Artificial Intelligence.* Pergamon Press, Oxford, 1992, 243–275. All quotes should be taken from and all citations should be to one of these published versions.

2 Main Concepts

The entities to be represented in SNePS include all entities a cognitive agent can think about or have beliefs about. This includes individual objects, classes of objects, people, properties, abstractions, actions, times, and propositions, both specific and generic (generic propositions being rules). This set of entities is not the same as the set of objects in the world, nor can the entities be mapped one-to-one onto the set of objects in the world. An agent may believe in the existence of two entities that, in fact, are the same object in the world; an agent may believe that what are, in fact, two objects in the world are one entity; an agent may have beliefs about non-existent, fictional, and even impossible entities. In the past, we have called such entities "intensions," "intensional entities," "intensional objects," and even "concepts" and "intensional concepts." Henceforth, we will call them "entities," or, for emphasis, "mental entities" (Cf. [54, 55]).

The SNePS representation formalism consists of *nodes* and labeled, directed *arcs.* Nodes comprise the terms of the formalism; arcs are grammatical punctuation like the parentheses and commas in the standard syntax of predicate calculus. Every entity represented in SNePS is represented by a node. Even nodes that represent propositions and nodes that represent rules are terms, so that SNePS can represent beliefs about beliefs and rules about rules without limit.

There are four types of nodes in SNePS: base nodes, variable nodes, molecular nodes, and pattern nodes. Base nodes and variable nodes have no arcs emanating from them. A base node represents some particular entity that is conceptually different from the entity represented by any other node. Additional properties of the entity are assertionally determined (see [66]) by the rest of the network connected to its node. (Isolated nodes cannot be created.) Variable nodes also have no arcs emanating from them, and represent arbitrary individuals, propositions, etc., again as determined by the rest of the network. Molecular and pattern nodes have arcs emanating from them and are structurally determined (see [66]) by those arcs, the nodes they go to, the arcs emanating from them, etc. Molecular nodes represent propositions, including rules, or "structured individuals." Pattern nodes are similar to open sentences or functional terms with free variables in standard predicate logic. Every node has an *identifier,* which uniquely identifies it. Base nodes may have identifiers determined by the user. All other nodes have identifiers created by the system.

The current version of SNePS, SNePS 2.1, contains SNeBR, the SNePS Belief Revision system [19] as a standard feature. Therefore, there is always a "current context" specified, which consists of a set of hypotheses asserted to the system by the user, and a "current belief space," consisting of the current context and all propositions so far derived from them. Nodes representing propositions currently asserted in the current belief space (either hypotheses or derived propositions), are indicated by the system's printing an exclamation mark at the end of their identifiers. These represent the current beliefs of the cognitive agent.

The set of arc labels used to structure the network is determined by the user, so that SNePS can be used to experiment with different conceptual structures. However, since SNIP, the SNePS Inference Package, is a standard part of SNePS, and SNIP must be able to interpret rules properly, the arc labels used to represent rules, and the specific representation of rules have been determined by the designers.

Examples of the 2D notation of SNePS are postponed until additional concepts are introduced in the next section.

3 History and Variants

3.1 SAMENLAQ

The earliest antecedent of SNePS was SAMENLAQ [59, 58]. The key ideas are listed in the abstract of [59]:

> The system is capable of three types of learning: by being explicitly told facts; by deducing facts implied by a number of previously stored facts; by induction from a given set of examples. The memory structure is a net built up of binary relations, a relation being a label on the edge joining two nodes. The relations, however, are also nodes and so can be related to other nodes which may also be used as relations. Most significantly, a relation can be related to one or more alternate definitions in terms of compositions of other relations and restrictions on intermediate nodes.

As indicated by this quote, both entities and relations were represented in SAMENLAQ by nodes. The information stored about relations was mostly information putting them in relational classes, and a connection to rules by which instances of the relationship could be deduced. The rules were not as general as allowed by predicate calculus, rather being from the relational calculus and forming a subset of the "path-based" inference rules later allowed in SNePS [50].

SAMENLAQ II [58] differed from the earlier version mainly in allowing the rules to be recursive, and allowing all instances of a relationship to be found even when some of them were explicitly stored in the net while others had to be deduced.

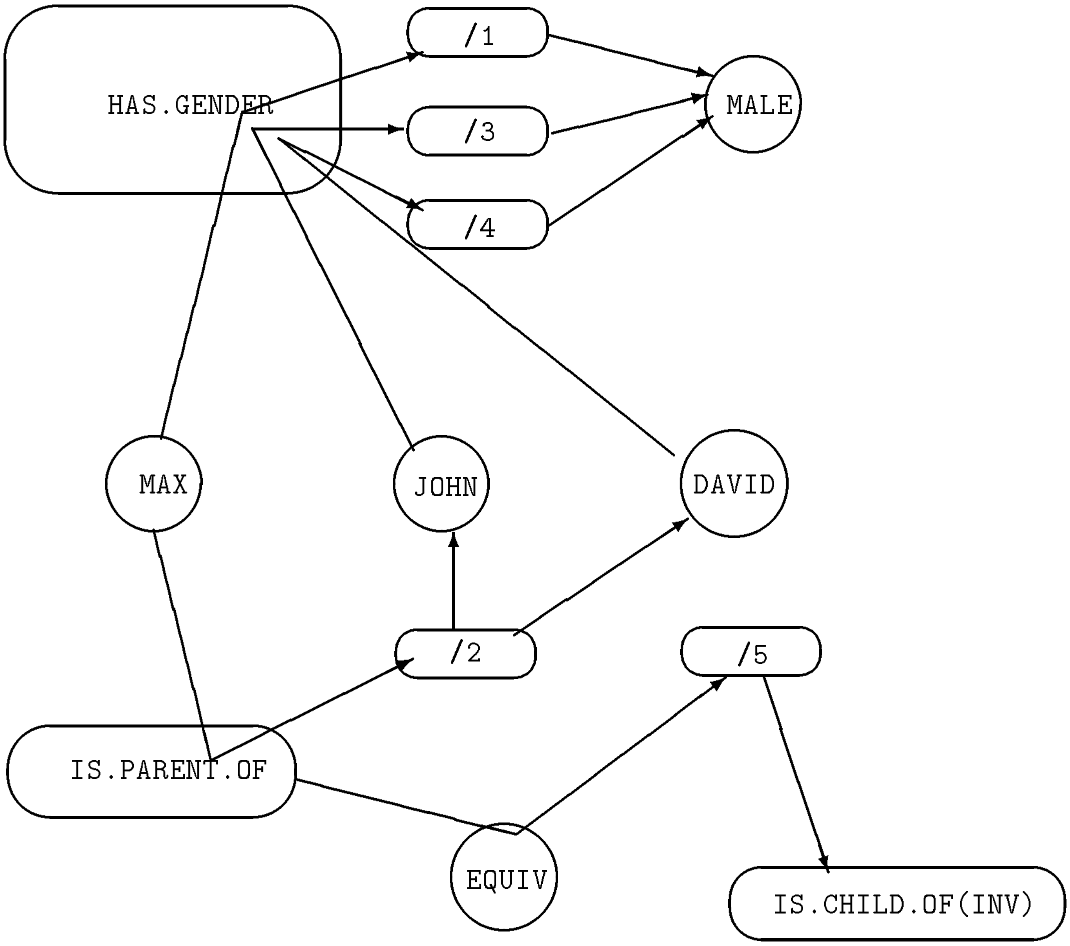

Figure 1: A 2D picture of a SAMENLAQ network showing the information that Max, John, and David are male, that Max is the parent of John and David, and that the parent-of relation is the converse of the child-of relation. Based on [59, p. 49].

The 2D notation for SAMENLAQ is shown in Figure 1. The nodes are `MAX, JOHN, DAVID, MALE, EQUIV, HAS.GENDER, IS.PARENT.OF,` and `IS.CHILD.OF(INV),` the first three representing individuals, `MALE` representing a property, and the last four representing relations. The ovals labeled `/1, /2, /3, /4,` and `/5` are just collectors of sets of nodes, so that, for example, the arc going from `MAX` through `IS.PARENT.OF` to `/2,` together with the arcs from `/2` to `JOHN` and `DAVID` represent the two relationships, Max is the parent of John and Max is the parent of David. The arc from `IS.PARENT.OF` through `EQUIV` to `/5` together with the arc from `/5` to `IS.CHILD.OF(INV)` represents the fact that the parent-of relation is the converse of the child-of relation. Actually, this piece of information was only used in the direction $x \, IS.CHILD.OF \, y \Rightarrow y \, IS.PARENT.OF \, x$. Therefore, in SAMENLAQ II, `EQUIV` was changed to `IMPLBY.` There was no 2D notation for the relationship between `IS.CHILD.OF(INV)` and `IS.CHILD.OF.` Neither was there a 2D notation for the rest of the rule syntax.

SAMENLAQ and SAMENLAQ II were implemented in SNOBOL3.

3.2 MENTAL

3.2.1 Shapiro's Version

MENTAL [40, 41] was a question answering system that used the MENS knowledge representation formalism, and was part of the MIND system [12]. The two major advances of MENS over SAMENLAQ were the representation of statements (relationships, propositions) by nodes, and the representation of rules by nodes.

SAMENLAQ had statements represented by three nodes and an arc that tied them together in the correct order. The move that allowed the representation of statements in MENS was to use one node for the statement, and three differently labeled arcs from the statement node to the three SAMENLAQ nodes. Following to a suggestion from Martin Kay, Fillmore's Case Grammar Theory [6] was used as a model, and the number of labeled arcs was increased from three to an arbitrary number, one for the verb and one for each case.

Figure 2 shows an example MENS network for the information in the sentences

(1) Jane saw (2) John hit Henry.

(3) Henry loves Jane.

(4) John loves Jane.

(5) Narcissus loves himself.

(6) Jane loves John.

The nodes (which were called "items" in the original papers) labeled with numbers, a simplification of the original node labeling scheme, represent the corresponding propositions. Each arc has two labels, one for each direction. For example, the arc from node `1` to node `2` is labeled `O`, while the arc from node `2` to node `1` is labeled `*O`.

Since one proposition can be the object of another, it is necessary to have some means of noting which nodes represent propositions that are asserted (as opposed to merely used) in the system. This was done by the "independent statement flag", which "is a single pointer which, when present, always points to the same item ...[it] indicates that the item it is attached to represents some information that has been alleged to be true. Two other cases are possible—the information may have been alleged to be false ...or no allegation

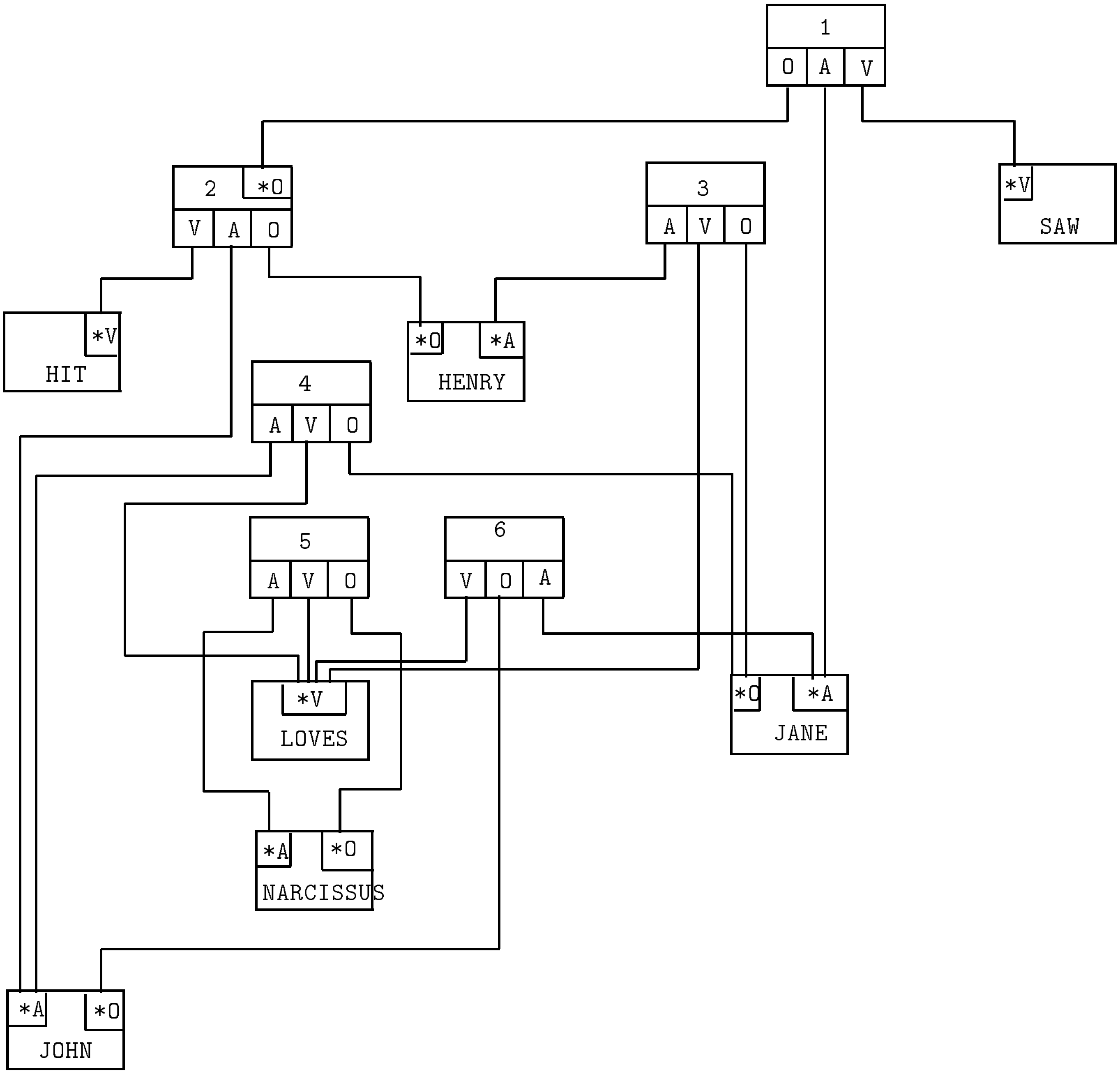

Figure 2: A MENS network based on [41, Figs. 2–4].

may have been made as to its truth value" [40, p 106].

A conscious decision was made not to use class membership or any other particular relation for the basic structure of the network, but to allow rules as representationally complete as the predicate calculus to be represented and used. This required techniques for representing variables, logical connectives, and quantifiers. It was decided to use nodes to represent variables, even though this would seem counter to the notion that every node represents some conceptual entity. (The representation and semantics of variables in SNePS remains a research topic to this day.) It was also decided to use nodes to represent quantifiers and connectives, even though their conceptual status was unclear. (This was changed in the next version. See below.)

Variables were allowed to range over all entities represented by nodes, but not over relations represented by arcs, because, "As Quine says, 'The ontology to which one's use of language commits him comprises simply the objects that he treats as falling ... within the range of values of his variables.' [33, p. 118]" [41, p. 518], and, although this allows the representation of Russell's and other paradoxes, "this possibility will be accepted. We make no type distinctions among the items and impose no restraints in item existence, leaving the avoidance of paradoxes the responsibility of the human informant" [*ibid.*].

The logical connectives included were: NOT, IMPLIES, IFF, AND, OR, and MUTIMP. "MUTIMP stands for mutual implication. It is a predicate with an arbitrary number of arguments and says that its arguments mutually imply each other by pairs (are pairwise equivalent)" [41, footnote, p. 519]. A node representing a non-atomic proposition had an arc labeled OP to the connective node, and one of the following sets of arcs: an arc labeled ARG to the one argument of NOT; arcs labeled ARG1 and ARG2 to the two arguments of IMPLIES and IFF; an arc labeled MARG to each of the arbitrary number of arguments of AND, OR, and MUTIMP.

Restricted quantification was used to represent quantified propositions. A node representing a quantified proposition had an arc labeled Q to the quantifier node, an arc labeled VB to the variable being bound, an arc labeled R to the restriction, and an arc labeled S to the scope. Both the restriction and the scope could be arbitrary propositions with the variable free. Thus, this was more general than the usual notion of restricted quantification, which allows restriction only by class (a unary predicate). Restricted quantification actually obviates most uses of IMPLIES.

Propositions containing either quantifiers or connectives were termed "deduction rules" because they act like derived rules of inference in other inference schemes. Figure 3 shows the MENS representation of the

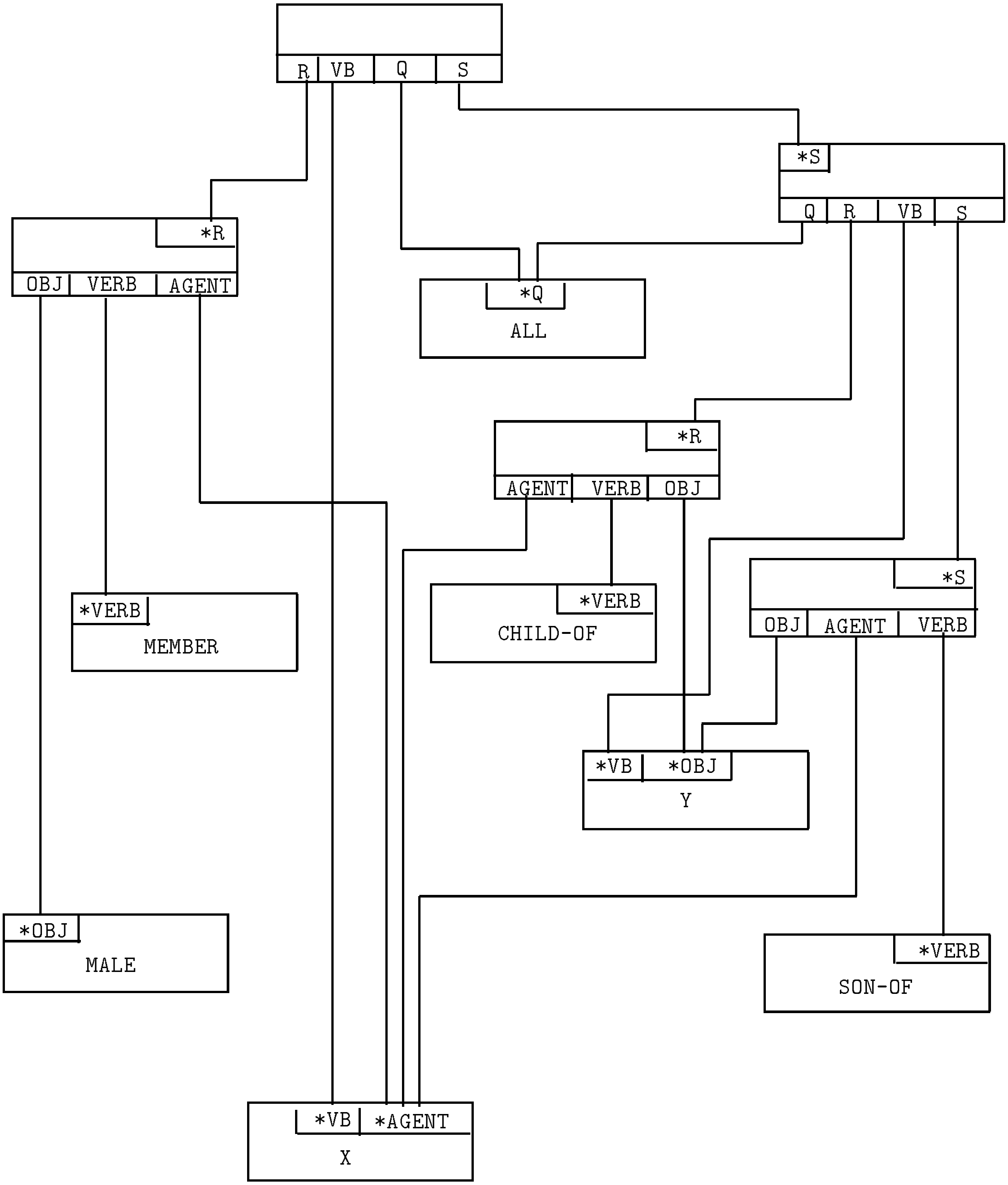

Figure 3: MENS representations of "If a male is a child of someone, he is the son of that person" (Based on [41, p. 519–520].)

deduction rule, "If a male is a child of someone, he is the son of that person."

MENTAL was partially implemented in PL/I, and ran interactively on an IBM System/360. The implementation included "all the procedures for storing information into the data structure, as well as all those for explicit retrieval and some of those for implicit retrieval [inference]" [41, p 512].

3.2.2 Kay's Version

Although discussing the same project, Kay [12] gave a description of the semantic network of the MIND system that differed in some details from that given above, and differed even more in the look of the 2D notation. Figure 4 shows this notation for the representation of the information in the sentence "Jane saw John hit Henry." The arcs labeled TV pointing to node 4 are the independent statement flags mentioned above. Note that Kay distinguished the proposition that John hit Henry from the act of John's hitting Henry, and made the latter the object of Jane's seeing.

Instead of representing quantifiers explicitly, Kay used a Skölem function technique where free variables were assumed to be universally quantified, and existentially quantified variables had an arc labeled F to the universally quantified variables they depended on. For example, Figure 5 shows how he would represent the rule, "Every boy owns a dog." Notice that both the dog and the particular owning act are represented as Skölem functions of the boy.

3.3 SNePS 79

When a revised version of MENS/MENTAL was implemented in Lisp, its name was changed to SNePS (Semantic Network Processing System), and, when SNePS was later revised, the original version was retroactively named SNePS 79 in honor of the date of publication of [45], the main reference for SNePS.

SNePS 79 had "three kinds of arcs: *descending, ascending,* and *auxiliary.* For each relation represented by descending arcs, there is a converse relation represented by ascending arcs and vice versa ... Auxiliary arcs are used for hanging nonnodal information on nodes and for typing the nodes" [45, p 180]. Besides base, variable, molecular, and pattern nodes, SNePS 79 had *auxiliary* nodes, which "are connected to each other and to other nodes only by auxiliary arcs. Auxiliary nodes do not represent concepts but are used by the SNePS system or the SNePS user to type nonauxiliary nodes or to maintain a reference to one or more nonauxiliary nodes" [*ibid*]. Additionally, there were *temporary* variable, molecular, and pattern nodes.

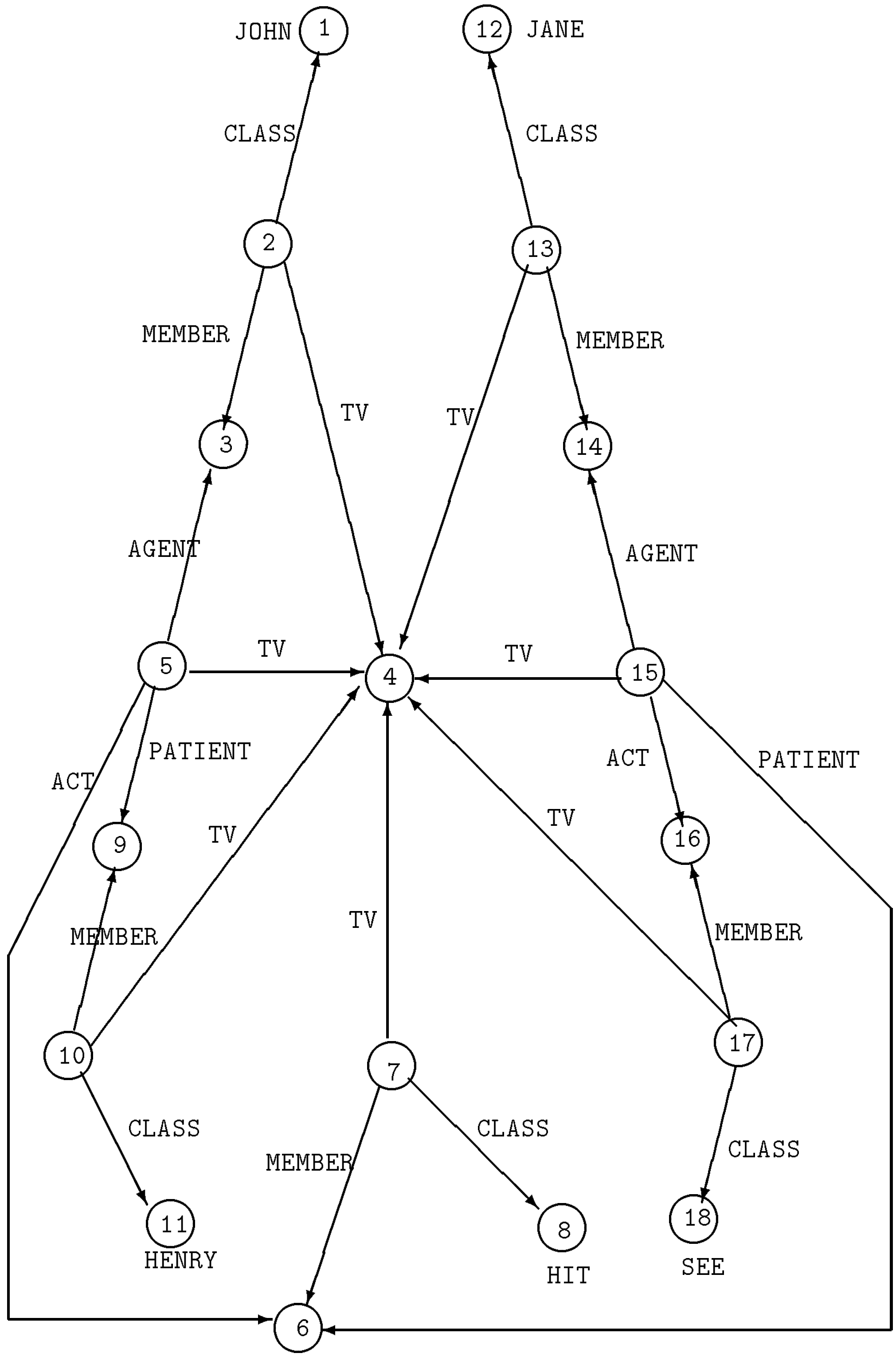

Figure 4: The MIND semantic network representation of "Jane saw John hit Henry" as based on [12, Figures 6,8].

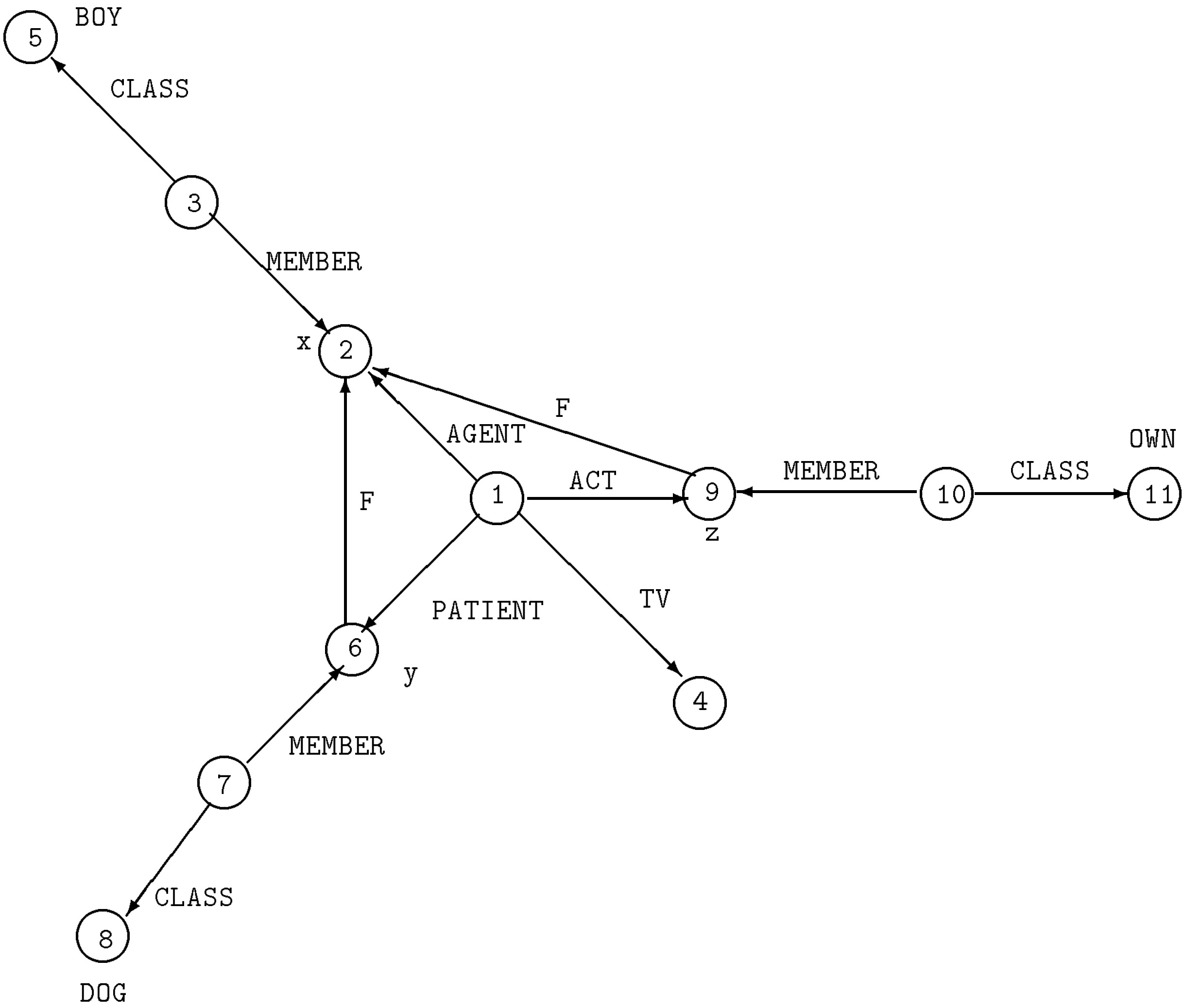

Figure 5: The MIND semantic network representation of “Every boy owns a dog” as based on [12, p 182–184].

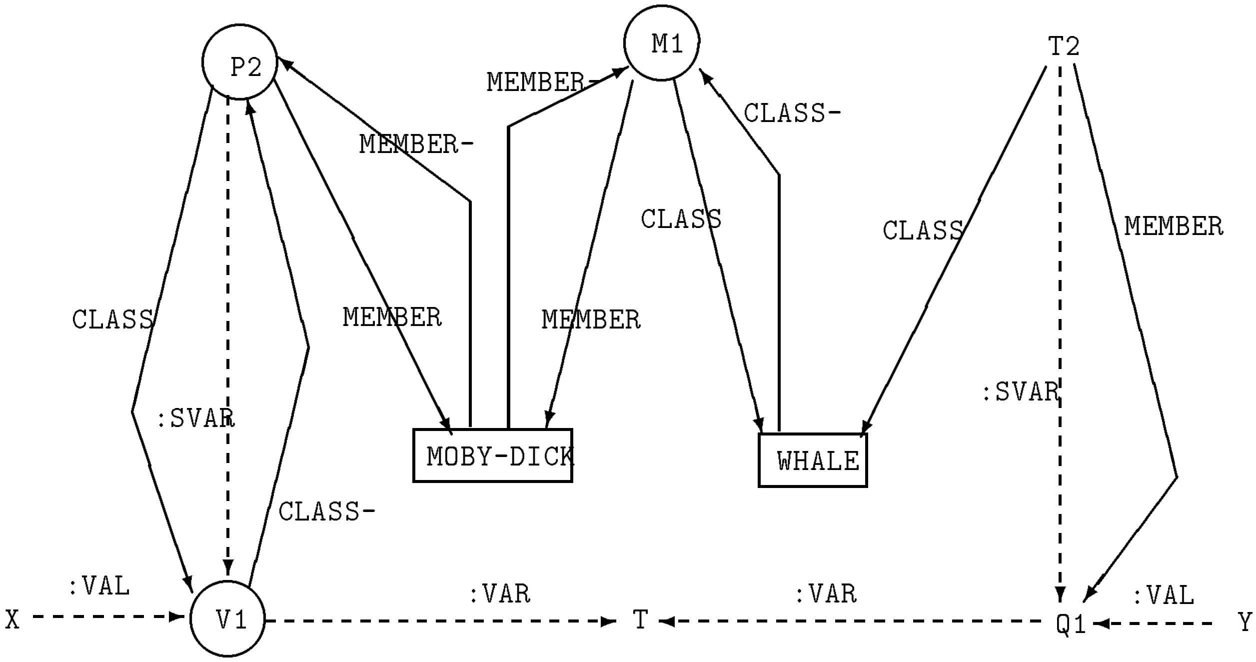

Figure 6: 2D notation of SNePS 79, showing various kinds of nodes and arcs. Based on [45, Figure 1].

Temporary molecular and pattern nodes "have no ascending arcs coming into them from the nodes they dominate. Temporary nodes are not placed on any permanent system list and are garbage-collected when no longer referenced. They are invisible to all the semantic network retrieval operations . . . Temporary nodes are used to build patterns of network structures, which can be matched against the network but do not match themselves" [45, p 181]. Figure 6 shows various kinds of nodes and arcs. In that figure, the arcs labeled `MEMBER` and `CLASS` are descending arcs, those labeled `MEMBER-` and `CLASS-` are ascending arcs, and those labeled `:VAL, :VAR,` and `:SVAR` are auxiliary arcs. The nodes `MOBY-DICK` and `WHALE` are base nodes, `V1` is a variable node, `M1` is a molecular node, `P2` is a pattern node, `Q1` is a temporary variable node, `T2` is a temporary pattern node, and `X, Y,` and `T` are auxiliary nodes. All variable nodes in the network have `:VAR` auxiliary arcs to the auxiliary node `T.` Pattern nodes have `:SVAR` auxiliary arcs to the free variable nodes they dominate. These auxiliary arcs and nodes are not normally shown in SNePS 79 diagrams. Neither are ascending arcs, since there is an ascending arc for every descending arc except when it would go to a temporary node. The SNePS 79 user could name the ascending arc anything, but we typically used the name of the descending arc with "`-`" appended.

In SNePS, we no longer represented quantifiers and logical connectives explicitly as nodes. Instead, a quantifier was represented as an arc from the node representing the scope of quantification to the node representing the bound variable, and logical connectives were represented implicitly by particular structures of arcs emanating from the node representing the non-atomic formula. The logical connectives used in SNePS 79 all took one or two sets of propositions as arguments. They are:

or-entailment: $\{A_1, \ldots, A_n\} \vee\!\!\rightarrow \{C_1, \ldots, C_m\}$ is true just in case each A_i entails each C_j. It is represented by an `ANT` arc to each A_i and a `CQ` arc to each C_j.

and-entailment: $\{A_1, \ldots, A_n\} \wedge\!\!\rightarrow \{C_1, \ldots, C_m\}$ is true just in case each C_j is entailed by the conjunction of all the A_i. It is represented by an `&ANT` arc to each A_i and a `CQ` arc to each C_j.

and-or: ${}_n\text{ⅩⅩ}_i^j\{P_1, \ldots, P_n\}$ is true just in case at least i and at most j of the P are true. It is represented by a `MIN` auxiliary arc to the auxiliary node `i`, a `MAX` auxiliary arc to the auxiliary node `j`, and `ARG` arcs to each of the Ps.

thresh: ${}_n\Theta_i\{P_1, \ldots, P_n\}$ is true just in case either fewer than i of the P are true, or else they all are. It is represented by a `THRESH` auxiliary arc to the auxiliary node `i`, and `ARG` arcs to each of the Ps.

The Universal (Existential) Quantifier was represented by an arc labeled `AVB` (`EVB`) from the node representing the scope of the quantifier to the variable node being bound. Restricted quantification was eliminated because it was felt to be unnecessary, an opinion since reversed.

SNIP [20, 42, 53], the SNePS Inference Package included as a part of SNePS 79, implemented the elimination rules for all of the quantifiers and connectives (deriving either a positive or a negative proposition), and the introduction rule for and-or. SNIP could use any rule either for backwards chaining, or for forward chaining, depending on whether the user asked a question or added new data (see [52]).

Later, the numerical quantifier (see [44]), path-based inference (see [43, 61, 50]), function nodes (see [47]), and default rules (see [46] and [54, p 285–287]) were added. The numerical quantifier is a way of storing and reasoning with statements of the form "Between 3 and 5 people are in the meeting" or "Every person has exactly one mother and exactly one father." Path-based inference is a way of specifying and using rules that infer an arc between two nodes from the existence of a path of arcs between those two nodes, and allows inheritance rules to be specified. Function nodes were a form of procedural (semantic) attachment, where the instances of a proposition were computed rather than being inferred, and where forward inference could trigger action. Default rules were representing by allowing default consequences on and-entailment and or-entailment rules, and were implemented by having SNIP try to derive the negation of the default consequent, and assert the consequent only if the negation were not derivable.

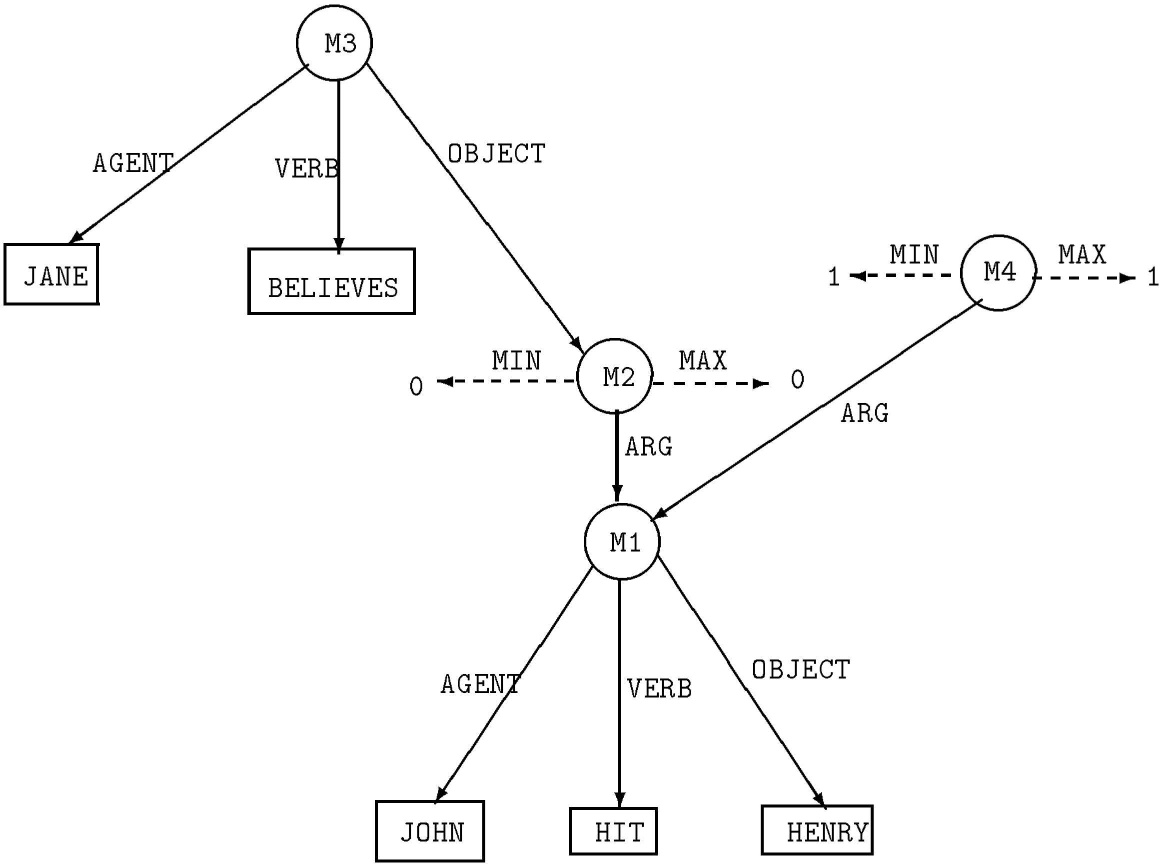

Figure 7: A SNePS 79 representation of "Jane believes that John didn't hit Henry, but he did," based on [45, Figure 17]

Since ${}_1\!\bigvee\!\!\!\!\bigwedge_1^1\{P\}$ means the assertion of P, the independent statement pointer was no longer needed. Instead, every propositional node (a molecular node representing a proposition) with no descending arc coming into it was taken to represent a proposition asserted in the network (believed by the cognitive agent). For example, Figure 7 shows the SNePS 79 representation of "Jane believes that John didn't hit Henry, but he did," using the representation of atomic propositions used in [45]. Node `M1` represents the proposition that John hit Henry, node `M2` uses and-or to represent the proposition that John didn't hit Henry, while node `M4` uses and-or to represent the proposition that, indeed, John hit Henry. Node `M3` represents the proposition that Jane believes the proposition represented by node `M2`. The technique whereby the asserted nodes are precisely the non-dominated nodes specifies that the cognitive agent believes the propositions represented by nodes `M3` and `M4`.

In [45], it was explicitly stated that one could reuse a molecular node, as node `M1` is being used both by node `M2` and node `M4`, or one could duplicate it by having another node with the same set of arcs to the same set of nodes. Later experience and thought convinced us that such node duplication should not be done, but SNePS 79 never enforced that decision.

SNePS 79 was implemented in a series of dialects of Lisp running on a series of different computers at Indiana University and SUNY at Buffalo.

3.4 SNePS 2.0

SNePS 2.0 was a complete redesign and reimplementation of SNePS in Common Lisp using an abstract data type approach explicated in [21]. It also incorporated theoretical decisions that had been made during experiments with SNePS 79. Some of the prominent differences of SNePS 2.0 from SNePS 79 were:

- Auxiliary nodes and function nodes were eliminated as not making conceptual sense. Function nodes are being replaced by a new theory of acting [13, 14, 51].
- An assertion tag was added to each node. The assertions of the system (beliefs of the cognitive agent) were now the set of nodes with the assertion tags set on. This eliminated much use of the and-or assertion operator, which had the additional problem of it's not being clear what the conceptual difference was between, *e.g.,* nodes `M1` and `M4` in Figure 7. Figure 8 shows the Sneps 2.0 version of Figure 7. Asserted nodes are indicated by their identifiers being printed with "`!`" appended. Because of the elimination of auxiliary nodes, the parameters of the and-or are now regular nodes, but this will be changed in the future.
- Temporary nodes were eliminated, since an unasserted permanent node could be used to represent a proposition the cognitive agent is contemplating.
- The system will not build a new node that dominates the same structure as an already extant node, thus enforcing the Uniqueness Principle.
- The Unique Variable Binding Rule, according to which two variables in one rule cannot be instantiated to the same term (see [48]), was implemented and enforced by SNIP 2.0.
- The Universal Quantifier is only supported if on a proposition whose main connective is one of the entailments, thus enforcing a kind of restricted quantifier. (The Existential Quantifier has not yet been reimplemented.) The arc used to represent the Universal Quantifier is `FORALL` instead of `AVB`.

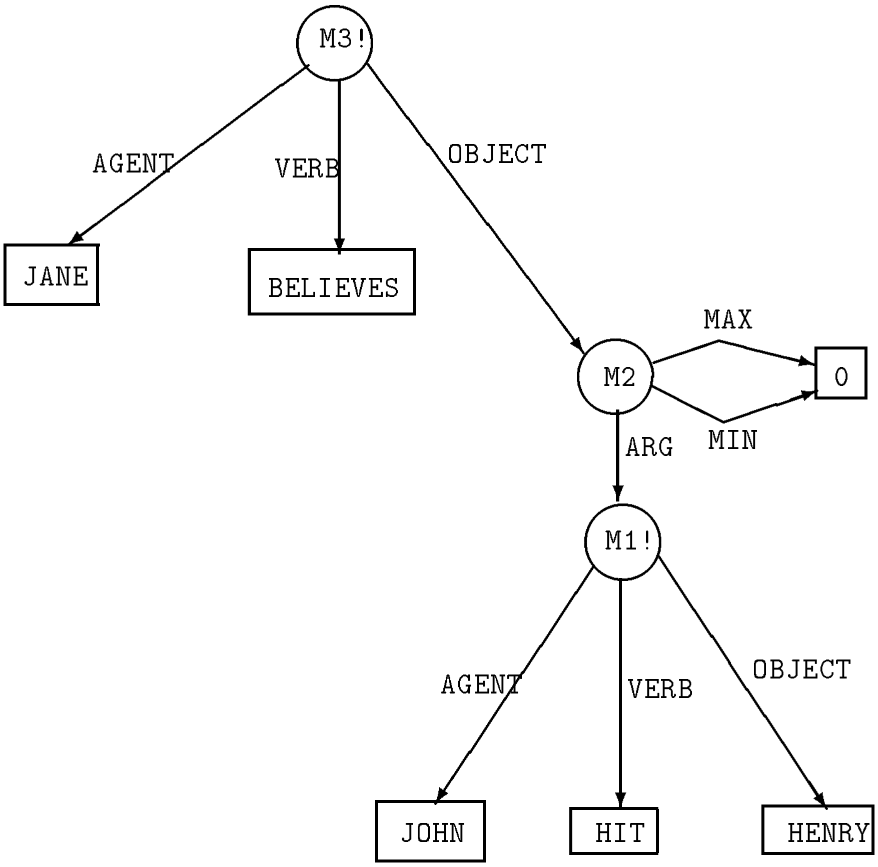

Figure 8: The Sneps 2.0 version of Figure 7. Asserted nodes are indicated by "!" appended to their identifiers.

- Thresh was given an additional parameter. ${}_n\Theta_i^j\{P_1, \ldots, P_n\}$ is true just in case either fewer than i or more than j of the P are true, The arc **THRESHMAX** goes to j.
- Auxiliary arcs were eliminated. Ascending arcs are no longer discussed—they are just considered to be following descending arcs backwards.
- Default consequences, the Existential Quantifier, and the Numerical Quantifier were not yet reimplemented.

3.5 SNePS 2.1

SNePS 2.1 is SNePS 2.0 with the inclusion of SNeBR, the SNePS Belief Revision System [19]. The significance of this was discussed above in Section 2. In brief summary, the assertion flag was replaced by a pointer to the context in which the proposition is either an hypothesis or a derived proposition, so the appearance of the "!" suffix becomes dependent on the current context, which can be changed by the user.

If a contradiction ever arises in SNePS 2.1, by the assertion of a proposition and its negation in the same context, the system will engage the user in a dialogue to identify the culprit and remove it from the context.

SNePS 2.1 can use the hypothetical reasoning facilities of SNeBR [17] to implement the entailment introduction rules and the rule of Universal Generalization.

4 Examples of Use

4.1 "Standard Tricky" Example

Probably because so much linguistic work was done by missionaries working to translate the Christian Bible into "native" languages, the Christian "Lord's Prayer" is often taken to be a "standard tricky" example of linguistic analysis. In this section, we show a possible representation of that example using SNePS/CASSIE case frames. For those unfamiliar with it, one version of the "Lord's Prayer" is the following:

> Our Father which art in heaven,
>
> Hallowed be thy name.
>
> Thy kingdom come.

Thy will be done in earth, as it is in heaven.

Now, CASSIE is not yet a poet, so let us instead consider the following, more prosaic, version:

Our Father, who is in heaven,

[I pray that] your name [will] be hallowed,

[I pray that] your kingdom [will] come,

[I pray that] your will [will] be done on earth [in the same way that] it is [done] in heaven.

Note that in this version we have interpolated some phrases to clarify (or make more understandable) the syntax. Clearly, some of these interpolations are, in fact, interpretations. An even more streamlined version, which is the one we choose to represent, is this:

I pray that our Father (who is in heaven)'s name will be hallowed, kingdom will come, and will will be done on earth in the same way that it is done in heaven.

The representation is shown in Figure 9, and a key to the figure is given in Figure 10. The representation is incomplete: we have used the linguist's triangles to hide the structure of node **m16**, about which we shall say more, and there would also be rules such as those in Figure 11. Also, the representation of 'in heaven' is a vastly oversimplified version of the representation described in Yuhan (forthcoming). Yuhan's theory would interpret the sentence 'My Father is in heaven' by a semantic network containing the following information:

Due to the existing act of my Father,

the Figure Object, individual **b2**

where

b2 is the Father of **b1**

and

b2 is named Jehovah

was BE-LOCated

at a place that has a Spatial Relation of "in"

to the Ground Object, individual **m8**

where **m8** is named heaven.

In Figure A, we have represented the phrase 'thy will be done on earth as it is in heaven' as if it were interpreted to mean "the manner in which thy will is done on earth be the same as the manner in which

Figure 9: SNePS/CASSIE representation of the Lord's Prayer.

Figure 10: Interpretation of the nodes in Figure 9.

Figure 11: Sample rule node for Lord's Prayer network: For all v1, v2, v3, if v3 is a v2 of v1, then v3 is a v2.

thy will is done in heaven" (and we have omitted the details of the representation of "the manner in which thy will is done ⟨at place X⟩"). There are a myriad of hermeneutic, representational, and methodological issues here. On the hermeneutic side, there is the question of how best to interpret the original phrase. For instance, the interpretation we have outlined is that the phrase asserts an equivalence. But that loses a certain asymmetry in the original. Perhaps a better alternative would be one of the following:

1. ... that all properties of the way in which our Father's will is done in heaven be properties of the way in which it is done on earth.
2. ... that $\forall x[x$ is our Father's will in heaven $\rightarrow x$ is our Father's will on earth$]$
3. it is the case that our Father's will is done in heaven, and I pray that it be done on earth, and I pray that ⟨one of the above⟩.

But this leads to the methodological issue, which is the most important: it is not the job of the knowledge-representation researcher to do Biblical exegesis. Once an interpretation is chosen, however, the KR researcher can decide how best to represent it. Perhaps it can be represented using already existing case frames; but perhaps new case frames, and attendant rules characterizing their inferential behavior, need to be developed (with the help of linguists and philosophers).

5 Some Current Uses of SNePS

In this section, we briefly summarize some current research projects that are using SNePS.

5.1 Planning and Acting.

SNACTor, the SNePS acting system [14, 49, 51, 60], is being developed to model in SNePS rational cognitive agents who can also plan and act. The modeled agent should be able to understand natural language; reason about beliefs; act rationally based on its beliefs; plan; recognize plans; and discuss its plans, acts, and beliefs. Doing all these tasks in a single coherent framework poses severe constraints. We have designed and implemented intensional propositional representations for plans.

We treat acts and plans as mental objects. This enables the modeled agent to discuss, formulate, use, recognize, and reason about acts and plans, which is a major advance over operator-based descriptions of plans. Operator-based formulations of actions tend to alienate the discussion of operators themselves. Operators are usually specified in a different language from that used for representing beliefs about states. Moreover, plans (or procedural networks) constructed from these operators can only be accessed by specialized programs (critics, executors) and, like operators, are represented in still another formalism. Our representations for acts, actions, goals, and plans build upon and add to the intensional propositional representations of SNePS. This framework enables us to tackle various tasks mentioned above in a uniform and coherent fashion. Figure 5.1 shows an example of our representations of plans in SNePS. The node M19 represents the proposition "M18 is a plan to perform the act represented by M17." M17 represents the act of performing the action "stack" on A (which is represented by node M15) and B (which is represented by node M12). M18 is the act ("snsequene") of sequencing two acts M11 followed by M14. M11 represents the act of performing "put" on B and the TABLE, and M14 is the act of performing "put" on A and B. Thus M19 is believed by the agent as "A plan to stack A on B is to first put B on the table and then put A on B."

Beliefs are stored as SNePS propositions in the agent's belief space (called a SNeBR context, see [19]). SNeBR (the SNePS system for Belief Revision), an assumption-based truth maintenance system [18, 17, 19], ensures that the agent's belief space is always consistent. In SNACTor, all interaction with the agent is done

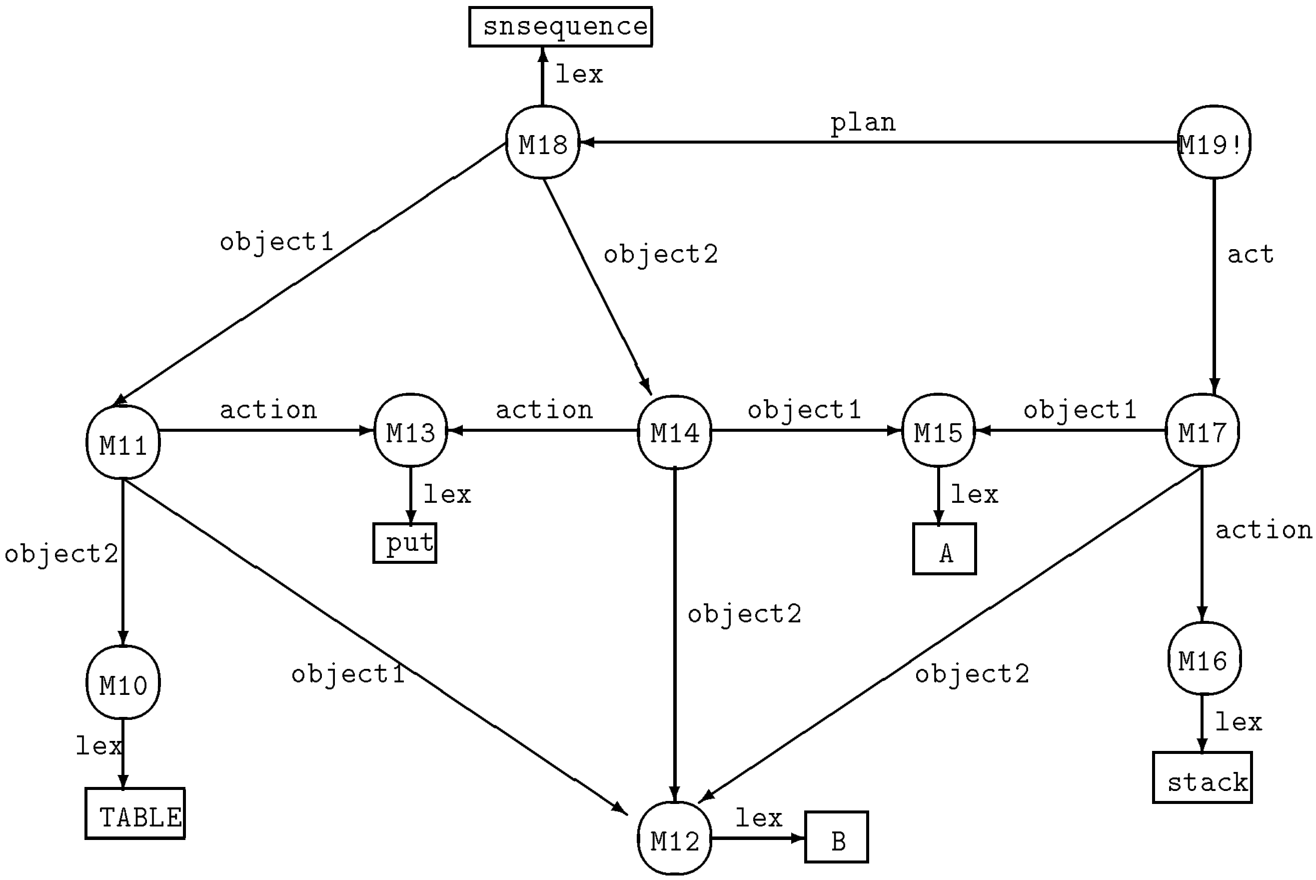

Figure 12: A representation for, "A plan to stack A on B is to first put B on the table and then put A on B"

using a domain-specific (blocks-world) natural language component. Sentences are parsed by a grammar (written in an ATN) and translated into SNePSUL (the SNePS User Language) commands that form beliefs in the agent's belief space. World-model rules for reasoning in the agent's belief space are also translated and represented as agent's beliefs.

The system is currently being advanced in several directions. See [13, 15] for further information.

5.2 Combining Linguistic and Pictorial Information.

There are many situations where words and pictures are combined to form a communicative unit; examples in the print media include pictures with captions, annotated diagrams, and weather charts. In order for a computer system to synthesize the information from these two diverse sources of information, it is necessary to perform the preliminary operations of natural-language processing of the text and image interpretation of the associated picture. This would result in an initial interpretation of the text and image, following which an attempt at consolidation of the information could be made. Although vision and natural-language processing are challenging tasks, since they are severely under-constrained, natural-language processing can more easily exploit constraints posed by the syntax of the language than vision systems can exploit constraints about the physical world. This fact, combined with the observation that the text often describes salient features of the accompanying picture in joint communicative units, leads to the idea of using the information contained in the text as a guide to interpreting the picture. This research focuses on a method of extracting visual information from text, which results in a relational graph describing the hypothesized structure of the accompanying picture (in terms of the objects present and their spatial relationships). The relational graph is subsequently used by a vision system to guide the interpretation of the picture. A system has been implemented that labels human faces in a newspaper photograph, based on information obtained from parsing the caption. A common representation in SNePS is used for the knowledge contained in both the picture and the caption. The theory is general enough to permit construction of a picture when given arbitrary descriptive text (without an accompanying picture).

Newspaper photographs have all the elements required for a true integration of linguistic and visual information. Accompanying captions usually identify objects and provide background information which the photograph alone cannot. Photographs, on the other hand, provide visual detail which the captions do not. Newspaper captions often identify people in a picture through visual detail such as "Tom Jones, wearing sunglasses ...". In order for a computer system to be able to identify Tom Jones, it is necessary to understand the visual implication of the phrase "wearing sunglasses". The face satisfying all the implied visual constraints could then be labeled accordingly.

The system uses a three-stage process to identify human faces in newspaper photographs. Only those photographs whose captions are factual but sometimes incomplete in their description of the photograph are considered. In the first stage, information pertaining to the story is extracted from the caption, and a structure of the picture in terms of the objects present and spatial relationships between them is predicted. The information contained in this structure would be sufficient for generating a picture representing the meaning of the caption. Using this information to label faces in an existing picture, however, entails further processing. The second stage, which constitutes the vision component, calls on a procedure to locate human faces in photographs when the number of faces and their approximate sizes are known. Although the second stage locates faces, it does not know whose they are. The last stage establishes a unique correlation between names mentioned in the caption and their corresponding areas in the image. These associations are recorded in a SNePS network and enable us to selectively view human faces as well as obtain information about them. Input to the system is a digitized image of a newspaper photograph with a caption, as in Figure 14.

Figure 13 illustrates the partial output of the parser on the caption of Figure 14. It postulates that four humans, namely Joseph Crowley, Paul Cotter, John Webb, and David Buck, are present in the picture (nodes **m38**, **m42**, **m46**, and **m50**). Furthermore, it postulates that Joseph Crowley appears above the other three in the picture (since he is "standing"), as represented by nodes **m51**, **m52**, and **m53**. The left to right ordering of the remaining three members is represented by the "left-of" relationship in nodes **m54** and **m55**. Factual information obtained from the caption (**m31**) is separated from derived visual information (**b12**). The hypothesized presence of an object in the picture is represented by a node (e.g., **m38**) with three arcs: COLLECTION, referring to the visual model it is part of; TYPE, indicating whether the object is explicitly mentioned in the caption or inferred to be present; and MEMBER, pointing to a detailed model of this object (e.g., **b10**). A node such as **m37** provides the link between the visual model of an object and the proper name it is associated with (in this case, 'Paul Cotter'). Hypothesized spatial relations between objects are represented by a node (e.g., **m52**) with 3 arcs pointing to (a) the type of spatial relation and (b) the nodes representing the two arguments to this binary relation. The system returns a labeling of parts of the image corresponding to the faces of the people mentioned in the caption, as in Figures 15a and b. See [62] and [63] for further details.

Figure 13: *Partial output of the parser on caption of Figure 14.*

Figure 14: *A newspaper photograph with caption "Wearing their new Celtics sunglasses are Joseph Crowley, standing with pennant, and seated from left, Paul Cotter, John Webb and David Buck."*

5.3 Graphical Deep Knowledge.

During work on the interface of the Versatile Maintenance Expert System (VMES; cf. [56, 57]), a theory of Graphical Deep Knowledge was developed. This representation is not pixel-oriented but combines iconic, meaningful, graphical primitives with propositional information about classes, attributes, positions, inheritability, reference frames, etc. Of special interest in this research is the use of part hierarchies. In [9, 11], a system of three different part hierarchies was developed, namely *real parts, subassemblies,* and *subclusters.*

This system of part hierarchies was extended in [10]. In this latest version, parts may either be *additive, constituting,* or *replacing.* These different part hierarchies differ in their behavior when a graphical representation containing a small amount of detail is extended to contain a large amount of detail. An additive part is simply added to the previous picture. A set of replacing parts forces the removal of the original picture and replaces it with a diagram of all its parts. Finally, constituting parts describe an object that has no form icon of its own. In this case, there is no distinction between the two pictures. For more details of this analysis and for details of the SNePS representation of these different part hierarchies, the reader is referred to [10].

Figure 15: *(a) output of system when asked to display Joseph Crowley (b) output of system when asked to display Paul Cotter*

Figure 16: The ship.

The power of Graphical Deep Knowledge is derived from the fact that most of the graphical information is integrated in the normal propositional representation mode of SNePS, so that it is possible to perform reasoning tasks about the graphical appearance of the represented objects. On the other hand, because the system generates diagrams from the propositional representation (and the icons), most changes to the knowledge state of the system can be immediately reflected in changes of the graphical appearance of the described objects on the screen. Graphical Deep Knowledge therefore permits the easy integration of natural language processing and graphics, because a common knowledge representation is used.

Figure 16 shows an example from [9]. This figure was generated from a knowledge base containing the SNePS network shown in Figure 17. This picture of the network, in turn, was generated according to natural language commands (minimally edited for better understanding).

The structure under **m1** asserts that ship-1 is a ship in the functional display modality. (Objects can be displayed in different modalities, e.g., functional or structural. For details, see [9].) The structure under **m2** asserts that a ship has (in the functional modality) the form ship-form. The node ship-form has a link to a LISP function that, if executed, draws an icon that looks like a ship.

The node **m12** defines a coordinate system that is Cartesian, is on the screen, is called system-1, and

Figure 17: Knowledge base required to generate Figure 16

has two axes called s-x and s-y. Node **m7** and all the nodes that are logically under it (even if they are drawn above it!) define the position of ship-1 in the coordinate system system-1 as (200 pixels, 300 pixels) in the directions defined by s-x and s-y. The system makes the simplifying assumption that any "first axis" specified by the user is parallel to the screen x axis.

Node **m3** and the nodes (logically) under it assert that ship-1 is faulty. Node **m5** and the nodes (logically) under it assert that faultiness is expressed by a rotation of 180 degrees. This is the reason why the ship is displayed upside down. The node **rotate-jg**, similar to the node **ship-form**, has a functional attachment that rotates an icon by the specified angle.

It should be pointed out that except for the special meaning of the node **SCREEN** and for the predefined rotation function, all information in this network has been created on the fly, either by natural language or by a simple, built-in, icon editor. Display requests addressed to this system in natural language are not parsed into a network representation, but immediately executed, resulting in a figure like Figure 16.

5.4 Knowledge Based Lexicons.

AI systems that involve natural language usually have at least three different knowledge bases: one for domain information, one for language rules, and one for information about words. Especially in comparison with domain information, knowledge about words tends to be isolated, fragmented, and impoverished. It is isolated, in that much of the information lies in specialized structures to which the system's reasoning mechanisms have little or no access, and which must be manipulated by specialized (and usually very limited) algorithms. It is fragmented, in that most on-line lexicons represent primarily syntactic information. Semantic information, when it is present at all, is usually in a separate representation scheme, accessible by different techniques. Information which requires both syntax and semantics for its representation usually falls through the cracks, as does information which is at least in part semantic, but which is less about the world than about the words. It is impoverished, in that all kinds of information tend to be represented sketchily, with little apparent concern for giving systems the kinds of information about words to which humans have routine access.

The work discussed in this section involves developing representation schemes for lexical information, in which all lexical information is represented in a single, unified SNePS network, accessible to the same retrieval and inference mechanisms as domain information. The lexicons under discussion are being built using semi-automated techniques from machine readable dictionaries; the representation is intended to support medium scale (semi-realistic) lexicons for a wide range of purposes. (Full, realistic sized lexicons require substantial back-end support because of problems of scale.)

This research is based on the linguistic theory of the relational lexicon [1]. Lexical relations provide a formal mechanism for expressing relations among concepts. Traditionally, lexical relations have been approached as a means for representing semantic information about words. Our approach extends the theory of lexical relations to embrace syntactic and morphological as well as semantic relations. The resulting hierarchy of lexical relations may be thought of as a kind of structural primitives, which represent cognitive relations among lexical items. Information about the lexical relation hierarchy is represented in a SNePS network. Information about specific words is integrated into the hierarchy network, resulting in a large network containing a (potentially dynamic) lexicon, with syntactic, semantic, and morphological information all readily available. The resulting representation distinguishes among homonym labels, words, strings representing word spellings, and word senses, allowing the kinds of reasoning about words that people routinely make. The representation used has several unusual features, among them the clear importance of path-based inference.

The relational hierarchy is a fairly straightforward instance of a hierarchical knowledge structure. At the present level of development, the most common hierarchical relationships involved are most obviously viewed as set membership and subset/superset relations. The one trap to avoid lies in representing the hierarchical information in the same way as one would represent, for instance, the information (derived from whatever source, but construed as part of the lexicon) that dogs are mammals.

There is, however, a relatively simple problem of recursion in representation. Since lexical relations are themselves objects of knowledge, within the SNePS formalism, they should be represented by nodes, not arcs. Now say that we represent the relation between 'Mt. Rushmore' and 'mountain' by something like an *arg1-arg2-rel* frame, with a *rel* arc to a node called something like 'member', which represents the lexical relation which holds between 'Mt. Rushmore' and 'mountain'. Now how do we say that the lexical relation 'member' belongs to the class of taxonomic lexical relations? If we try to use the same 'member' node as the first argument and also as the relation, we wind up with a pathological proposition node.

The reason this is a problem, and not a blunder, is that the lexical relation is not really the same thing that holds between itself and its parent class. The problem here is one of distinguishing metalevel information about the lexical relation hierarchy from object level information which we use it to represent. We can see the importance of distinguishing them if we look at the consequences of trying to use path-based inference to implement inheritance down hierarchical subtrees. When we construct a path-based rule to make all object

Figure 18: Representation of Hierarchical Facts.

property markers behave alike in some respect, we don't want to have to filter for instances of membership like 'Mt. Rushmore' to 'mountain', or of subset like 'dog' to 'mammal'.

Figure 5.4 shows the SNePS representation for a fragment of the lexical relation hierarchy. Throughout the examples, we adopt several node naming conventions for simplicity. Node names in all capitals represent members of the lexical relations hierarchy. Proposition nodes are numbered beginning with **m1** as in other SNePS networks. Other node names in lower case represent word sense nodes.

Information about words takes several forms. The first, and simplest, is lexical relation instances. This information encodes everything from taxonomic classification relations (including relations like TAXON, which closely corresponds to the traditional IS-A link, but also synonymy relations, antonymy relations, and others) to generic role relations (for instance the Act/Actor relation, which holds, e.g., between "bake" and "baker") to morphological and syntactic relations such as the State/Verb relation which holds between a nominalized verb and its verb. The second kind of information could be viewed as collapsed lexical relations, and covers such relatively simple markers as part of speech (associated with sense nodes), alternate spellings (relation between a headword node and a string node), and the like. The third reflects information about the sense-subsense hierarchy. This is the familiar dictionary definition hierarchy, under which a word may have more specific senses which represent refinements on more general ones.

We now integrate all three kinds of information about words into the hierarchy network in the obvious way. Figure 5.4 represents a digested form of the information about some words related to the most common sense of "sheep" as derived from definitions in *Webster's Seventh Collegiate Dictionary.* Note that this sense of "sheep" has a subsense, which covers only a single species. We have omitted all nodes and arcs above the

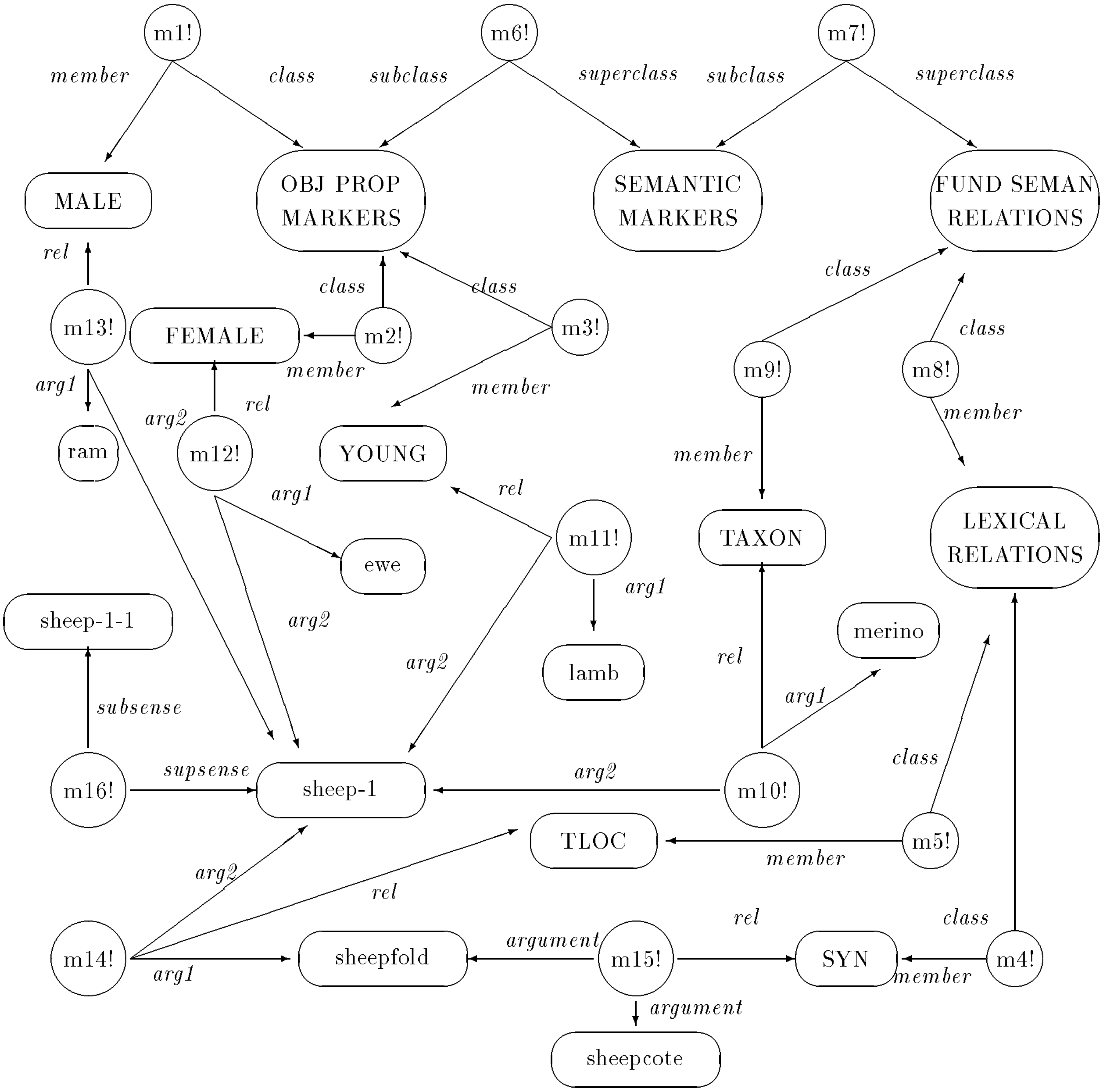

Figure 19: Representation of Lexical Information.

sense level (i.e., headword and string nodes and their associated frames) for simplicity, and collapsed parts of the lexical relation hierarchy (especially in nodes **m4** and **m5**) to make the figure more readable.

The resulting lexicon can be used either as stand-alone support for some external activity (such as automated query expansion for information retrieval) or as an integral part of the knowledge base for a natural language system. For the second option, which is the more interesting, the remaining information in the knowledge base is integrated into the lexicon representation, with semantic information relating to senses hanging off the appropriate nodes. The result is a knowledge base which not only has ideas and knows what words are typically used to represent them (the effect of using something like *lex* arcs in a more traditional SNePS-ish representation), but knows about those words and can reason on the basis of its knowledge. (More detailed descriptions of this research are [7, 25, 24, 27], and [26].

5.5 Kinds of Opacity and Their Representations.

Examinations of referential opacity (the failure of certain linguistic contexts to be "transparent" to substitution of equivalents) have almost invariably focused on propositional attitudes, and even more specifically, on belief sentences. This focus has obscured the fact that there are a number of other kinds of sentences in which the logical-linguistic phenomenon of referential opacity occurs. In addition to the propositional attitudes, which express an agent's attitude towards some proposition or another, such as

(1) John believes that the Morning Star is blue,

there are other kinds of referentially opaque sentences. One of these is attributive adverbs, like *quickly* in

(2) John swam the river quickly.

We have proposed a new treatment of attributive adverbs as generating opacity resting on the general theory of action proposed by the philosopher Donald Davidson [5], though he himself holds that such adverbs must be excluded from his theory.

According to Davidson, a verb of action refers to an event, an act. In (2), John performed a certain act, a swimming of the river. As far as *swimmings* of the river go, his swimming of it was quick. If the time it took John to swim the river is compared to the times it has taken others to *swim* the river, John's time was among the fastest. John's act of swimming the river may, however, also be described as a *crossing* of the river. Thus, we may add the identity statement

(3) John's swimming of the river is a crossing of the river.

But though his swimming is a crossing, it does not follow that

(4) John crossed the river quickly.

On the contrary, it is very likely that he *crossed* the river slowly. If the time it took John to cross the river is compared to the times it has taken others to cross the river, which will perhaps include such modes of crossing as rowing, swinging (on a rope), and driving (on a bridge), John's time would no doubt be among the *slowest*. In other words, when we describe John's act as a swimming, it is quick, but when we describe it as a crossing, it is slow. If we accept that the one *act* cannot be both quick and slow, then the swimming and the crossing cannot be identified with the act. In an intensional system, a natural approach to explore is to treat the swimming and the crossing as different *intensional* objects.

The intensional objects to be used in our representations are called *aspects*, which are reminiscent of Frege's *senses* [8]. Given a description, **d**, the Fregean sense associated with it is the *meaning* of **d**. The aspect associated with **d**, however, is the result of conceiving of the thing **d** refers to as, or *qua*, a thing to which the description **d** applies. Aspects are *objects*, albeit intensional, abstract objects. The general form of an aspect is:

aspect := *object*, qua *description true of the object*

In [67] this intuitive notion of "qua" is replaced formally using Church's λ–abstraction [4].

The sentence containing an attributive adverb

(2) John swam the river quickly,

is understood as having the logical form

(5) There is someone b1 named *John,* there is an action `b2` that is a swimming, and there is an object `b3` that is a member of the class of rivers, such that `b1` is the agent of `b2`, and `b2` has `b3` as its direct object, and there is an aspect `a1` — `b2`, qua `m1` — which is quick.

Assuming that John's swimming of the river is the *same* event as his crossing of the river, and that

(6) John crossed the river slowly,

the representations of (2) and (6) are as in Figure 20

No inferences from the properties of an aspect to properties of the object the aspect is an aspect of, are permitted. Thus, nothing is said to be both quick and slow, since `a1` (i.e., `b2`, qua being a swimming) is a different aspect from `a2` (i.e., `b2`, qua being a crossing). (Details are in [67]; cf. also [68].)

Figure 20: Representation of (2) and (6).

5.6 Representing Fiction in SNePS.

As part of the SUNY Buffalo Center for Cognitive Science's project on Cognitive and Computer Systems for Understanding Narrative Text, we are constructing a computational cognitive agent, Cassie, implemented in SNePS, who will be able to read a narrative and comprehend the indexical information in it, specifically, *where* the events in the narrative are taking place (in the world of the narrative), *when* they take place (in the time-line of the narrative), *who* the participants in these events are (the characters in the world of the narrative), and *from whose point of view* the events and characters are described [3, 38, 39].

In order to do this, Cassie has to be able to (1) read a narrative (in particular, a fictional narrative), (2) build a mental-model representation of the story and the story-world, and (3) use that mental model to understand and to answer questions about the narrative. To build the mental model, she will need to contribute something to her understanding of the story. One contribution is in the form of a "deictic center"—a data structure that contains the indexical information needed to track the who, when, and where.

Another contribution is background knowledge about the real world. For instance, if Cassie is reading a novel about the Civil War, she would presumably bring to her understanding of it some knowledge of the Civil War, such as that Abraham Lincoln was the 16th president and was assassinated in 1865, even if that information is not explicitly stated in the novel. The novel might go on to make other claims about Lincoln, such as that he had a particular conversation with General Grant on a particular day in 1860 (even if, in fact, they never talked on that day—this is a novel, after all). Such a claim would probably not be inconsistent with anything Cassie antecedently believed about Lincoln. But some claims in the novel might be thus inconsistent, e.g., if she read that Lincoln was re-elected to a third term in 1868. So Cassie has to be able to represent the information presented in the narrative, keep it suitably segregated from her background knowledge, yet be able to have information from her antecedent beliefs "migrate" into her model of the story world as well as have information from the story world "migrate" back into her store of beliefs about the real world.

There have been a number of theories in philosophy about the nature of fictional objects. All of these are *ontological* theories concerned with such questions as: What are fictional objects? How can they have properties? How are they related to non-fictional entities? However, for the purposes of our project, we need to be more concerned with "epistemological" or processing/computational/interpretive issues: How does a reader understand a (fictional) narrative? How does a reader decide whether and to what extent it *is* fictional? How does a reader construct a mental model of the story world? How does a reader represent fictional entities and their properties? How does a reader integrate his or her knowledge of the real world with what s/he reads in the narrative? And so on. Some of these are, indeed, ontological issues, but they are what we have elsewhere termed issues in "epistemological ontology" [34]: Corresponding to the purely or *metaphysically* ontological question, "What *are* fictional objects?", we ask the *epistemologically* ontological question, "How does a cognitive agent *represent* fictional objects?". And corresponding to the purely ontological question, "How are properties *predicated* of fictional objects?", we ask the epistemologically ontological question, "How does a cognitive agent *represent* the properties of fictional objects?"

In order for Cassie to read a narrative, the knowledge representations she should construct will include a story operator (like [16] or [64]), only one mode of predication of properties to (fictional) objects (like [28]), and only one kind of property (like [22, 23]). It must be kept in mind that all entities represented in Cassie's mind are just that—entities in her mind—*not* entities some of which are real and some of which are fictional.

The story operator will set up a "story space" that is formally equivalent to a belief space (cf. [35, 55, 65]). It will allow Cassie to distinguish her own beliefs about London from (her beliefs about) claims made about London in a story in precisely the same way that belief spaces allow Cassie to distinguish her own beliefs about Lucy from her beliefs about John's beliefs about Lucy (cf. [35, 54]).

But how should this be handled? Consider Figure 21. Suppose that one of Cassie's background beliefs is that Lincoln died in 1865 and that she reads in a narrative that Lincoln was re-elected in 1868. There is a processing problem: Cassie is faced with an inconsistency. There are two solutions. First, the SNePS Belief Revision system (SNeBR; [19]) can be invoked. The detection of the inconsistency will cause a split to be made into two (consistent) contexts. But note that the net effect of this is to embed the second statement (the re-election in 1868) in a story operator. So why not start with a story operator in the first place? This is the second solution, as shown in Figure 22.

But now let's complicate the data a bit. Consider Figure 23. Suppose that Cassie's background beliefs include both that Lincoln was the 16th president and that Lincoln died in 1865, and suppose once again that Cassie reads in a narrative that Lincoln was re-elected in 1868. The processing "problem" here (it is not really a problem) is that we want the first of Cassie's two background beliefs to "migrate into" the story world. The reason that this is not a problem is that those first two background beliefs are *Cassie's* beliefs and the third is not. The first one (that Lincoln was 16th president) is both believed by Cassie *and* is in the story world.

Consider Figure 24. If Cassie knows that she is reading a narrative, we want it to be the case that she believes (1) (that Washington was the first president), and we want both (1) and (2) (that he chopped down the cherry tree) to be in the story world. How do we accomplish this? By *starting* with a story operator on (2). In general, *we will put a story operator on* **all** *narrative predications.*

But then we face two problems: Background beliefs of the reader are normally brought to bear on understanding the story, as we saw in Figure 21 and Figure 22 (cf. [38]). And we often come to learn (or, at least, come to have beliefs) about the real world from reading fictional narratives. Thus, we need to have two rules:

(R1) Propositions *outside* the story space established by the story operator (i.e., antecedently believed by the reader) are assumed, *when necessary*, to hold *within* that story space *by default* and *defeasibly*.

(R2) Propositions *inside* the story space are assumed, *when necessary*, to hold *outside* the that story space *by default* and *defeasibly*.

Some comments: The "when necessary" clause is there to prevent an explosion in the size of belief and story spaces; the migrations permitted by these two rules would only take place on an as-needed basis for

Figure 21: Example 1a

Figure 22: Example 1b

Figure 23: Example 2

Figure 23: Example 2

Figure 24: Example 3

understanding the story or for understanding the world around us. The "by default" clause is there for obvious reasons: we wouldn't want to have Lincoln's dying in 1865 migrate into a narrative in which he is re-elected in 1868. The "defeasibly" clause is there to undo any damage that might be done at a later point in the narrative if such a migration had taken place, innocently, at an earlier point. Rule (R1) aids in our understanding of the story. Rule (R2) allows us to enlarge our views of the world from reading literature, while also allowing us to segregate our real-world beliefs from our story-world beliefs.

In the Examples, we have used the linguist's triangle to hide irrelevant details, but it is worth showing how the story operator looks in detail. This is shown in Figure 25.

For more details, see [36] and [37].

5.7 Natural Category Systems.

We have developed representations for natural category systems based on a Roschian model of categories that has been extended to accommodate the recent categorization research of Barsalou (1982, 1987, 1988), Keil (1987), Medin (1986, 1987), Murphy (1986, 1988), Neisser (1987), and Lakoff (1987). We take issue with the assumption, implicit in most artificial intelligence (AI), natural language processing (NLP) systems that generic concepts can be viewed simply as collections of attributes. Rather, richer representations are needed to explain conceptual coherence and the richness of conceptual structure (Barsalou 1988; Lakoff 1987; Keil 1987; Medin 1987; Murphy 1987, 1988). I.e., categories are further structured by deeper conceptual relations and organized by core principles; lists of attributes fail to capture interproperty (intraconcept) and interconcept relations. We believe that these deeper conceptual relations encode commonsense knowledge about the world necessary to support natural language understanding.

Our system uses default generalizations to represent facts about the typical exemplars or members of a category. Thus, a basic level category in our semantic network is, in part, a collection of default generalizations about part/whole structure, image schematic structure, additional percepts, and functional and interactional properties. Figure 26 shows the default rule that can be paraphrased as *For all x, if x is a car, then typically x has an engine* or more simply as *Typically, cars have engines.*

We build many such default generalizations about the basic level category *car*; i.e., generalizations about typical parts and other attributes. It may seem cumbersome to build such a default rule to represent a generic sentence such as: *cars have engines*; why not just build a prototypical car that has an engine? Although we want to capture prototype effects in our representations, we agree with Rosch (1978) and Lakoff (1987)

Figure 25: The story operator

Figure 26: The following defines a path to find all the parts of basic-level objects: `(def-path parts (compose arg2- arg1 part- whole forall- ant class))`

Figure 27: *Engines are interior parts of cars*

that the existence of these effects merely indicates that prototypes must have some place in theories of representation, processing, and learning of categories.

As people's knowledge increases, they come to reject mere collections of surface attributes and other typical features as being adequate to specify concepts (Barsalou 1988; Keil 1987; Medin & Wattenmaker 1987; Murphy & Medin 1985). Our current hypothesis is that basic level categories start out as perceptual categories; in this stage of development the knowledge associated with categories consists of default generalizations about surface parts and other perceptual attributes. As learning and development proceed, additional conceptual relations based on theories and causal mental models further structure these categories and attributes. I.e., our knowledge becomes organized by core principles.

We build additional default rules for these "deeper" conceptual relations. Thus, in addition to part-whole relations (**m5** in Figure 26) and relations about other percepts, we structure basic level categories such as *car* with enabling, functional, and spatial relations such as those shown in Figures 27–29. (We have not shown the entire default rules, just the additional conceptual relations. I.e., **m8, m9,** and **m11** would replace **m5**, the part-whole relation, in the default rule of Figure 26, creating three additional, similar default rules.) Figure 27 shows a spatial relation that further structures the parts of *car*, which can be paraphrased as *engines are inside (or interior parts) of cars.* We structure the external parts of *car* similarly. Figure 28 is used to further structure or cluster mechanical parts of *cars*, such as the *brakes* and *engine*, and can be paraphrased as *engines are mechanical parts of cars* (together with **m7**). Figure 29 shows an enabling relation: *engines enable cars to run/go.* Thus, in our system, there will be many assertions linking *car* and *engine*: the knowledge associated with a basic level category such as *car* is highly interconnected and

Figure 28: *Engines are mechanical parts of cars*

Figure 29: *Engines enable cars to run*

organized by spatial, temporal, causal, explanatory, and enabling relations.

In our system, concepts are not invariant structures retrieved intact from long-term memory (LTM), but rather are constructed in working memory (WM), tailored to a particular linguistic context on a particular occasion. I.e., different information associated with a category in LTM is incorporated in the temporary concept constructed in WM in different contexts. Categorizing an entity provides access to a large amount of information; however, only a small subset of the information associated with a category in LTM is incorporated in a temporary concept constructed in WM. Category knowledge in our system is relatively unorganized, interrelated knowledge that can be used to construct temporary concepts in WM, appropriate to the current task and context.

When a new individual identified by its basic level name (e.g., a car) or a generic basic level category (e.g., the type car) is encountered in input, the context-independent and context-dependent satellite entities implicitly evoked by hearing/reading the category name are placed in WM. We believe that these reflex, or subconscious, inferences are made at the time of reading/hearing the central basic level category name. The SNePS path-based inference package provides the subconscious reasoning that is required for implicit focusing of satellite entities. The definition of appropriate paths in the network enables the automatic retrieval of the relevant satellite concepts of basic level concepts. Thus, we use the additional structure provided by the intraconcept and interconcept relations, defining paths in the network that retrieve external parts after processing input such as Lucy washed her car, interior, mechanical parts after processing input such as The mechanic repaired the car, and enabling satellite entities (e.g., a mortgage) after processing Lucy bought a new house.

For additional information on this topic, see [29, 30, 31, 32].

6 Current Availability and Requirements

SNePS 2.1 is implemented in Common Lisp, and has run on a variety of platforms, including Symbolics Lisp Machines, Texas Instruments Explorer Lisp Machines, SUN workstations, an Aliant Multimax, and DEC VAXes running VMS.

SNePS 2.1 is currently available via a license agreement with the Research Foundation of State University of New York. There is a minimal handling fee for non-profit research labs and educational institutions. The license fee for for-profit organizations in negotiable. For information and a copy of the license agreement, send name, address and phone number to the first author of this article.

7 Acknowledgments

We wish to thank James Geller, Deepak Kumar, João Martins, J. Terry Nutter, Sandra L. Peters, Rohini K. Srihari, and Richard W. Wyatt for supplying us with descriptions of their projects, and the members of SNeRG (the SNePS Research Group) for comments and discussion. We especially wish to thank Deepak Kumar for technical assistance at the last minute.

References

[1] Y. D. Apresyan, I. A. Mel'čuk, and A. K. Zolkovsky. Semantics and lexicography: Towards a new type of unilingual dictionary. In F. Kiefer, editor, *Studies in Syntax and Semantics*, pages 1–33. D. Reidel, Dordrecht, Holland, 1969.

[2] R. J. Brachman and H. J. Levesque, editors. *Readings in Knowledge Representation.* Morgan Kaufmann, San Mateo, CA, 1985.

[3] G. A. Bruder, J. F. Duchan, W. J. Rapaport, E. M. Segal, S. C. Shapiro, and D. A. Zubin. Deictic centers in narrative: An interdisciplinary cognitive-science project. Technical Report 86–20, Department of Computer Science, SUNY at Buffalo,, 1986.

[4] A. Church. *The Calculus of Lambda Conversions.* Princeton University Press, Princeton, 1944.

[5] D. Davidson. The logical form of action sentences. In D. Davidson, editor, *Essays on Actions and Events*, pages 105–148. Clarendon Press, Oxford, 1980.

[6] C. J. Fillmore. The case for case. In E. Bach and R. T. Harms, editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart and Winston, New York, 1968.

[7] E. A. Fox, J. T. Nutter, T. E. Ahlswede, M. W. Evens, and J. A. Markowitz. Building a large thesaurus for information retrieval. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 101–108. Association for Computational Linguistics, Feb. 1988.

[8] G. Frege. On sense and reference. In P. Geach and M. Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*, pages 56–78. Blackwell, Oxford, 1970. Originally published in 1892.

[9] J. Geller. *A Knowledge Representation Theory for Natural Language Graphics.* PhD thesis, Department of Computer Science, SUNY at Buffalo, Buffalo, NY, 1988. Available as Technical Report #88-15.

[10] J. Geller. A graphics-based analysis of part-whole relations. *Cognitive Science*, in press.

[11] J. Geller. Propositional representation for graphical knowledge. *International Journal of Man-Machine Studies*, in press.

[12] M. Kay. The MIND system. In R. Rustin, editor, *Natural Language Processing*, pages 155–188. Algorithmics Press, New York, 1973.

[13] D. Kumar. An integrated model of acting and inference. In D. Kumar, editor, *Current Trends in SNePS*, pages 55–65. Springer-Verlag Lecture Notes in Artificial Intelligence, No. 437, Berlin, 1990.

[14] D. Kumar, S. Ali, and S. C. Shapiro. Discussing, using and recognizing plans in SNePS preliminary report—SNACTor: An acting system. In P. V. S. Rao and P. Sadanandan, editors, *Modern Trends in Information Technology: Proceedings of the Seventh Biennial Convention of South East Asia Regional Computer Confederation*, pages 177–182. Tata McGraw-Hill, New Delhi, India, 1988.

[15] D. Kumar, S. S. Ali, S. C. Shapiro, and J. Haas. The SNePS acting system. In K. E. Bettingr and G. Srikanthan, editors, *Proceedings of the Fifth Annual University at Buffalo Graduate Conference in Computer Science*, pages 91–100. Department Of Computer Science, SUNY at Buffalo, Buffalo, NY, 1990.

[16] D. Lewis. Truth in fiction. *American Philosophical Quarterly*, 15:37–46, 1978.

[17] J. P. Martins and S. C. Shapiro. Hypothetical reasoning. In *Applications of Artificial Intelligence to Engineering Problems: Proceedings of The 1st International Conference*, pages 1029–1042. Springer–Verlag, Berlin, 1986.

[18] J. P. Martins and S. C. Shapiro. Theoretical foundations for belief revision. *Theoretical Aspects of Reasoning About Knowledge*, pages 383–398, 1986.

[19] J. P. Martins and S. C. Shapiro. A model for belief revision. *Artificial Intelligence*, 35:25–79, 1988.

[20] D. P. McKay and S. C. Shapiro. Using active connection graphs for reasoning with recursive rules. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 368–374. Morgan Kaufmann, San Mateo, CA, 1981.

[21] 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.

[22] H.-N. C. neda. Thinking and the structure of the world. *Philosophia*, 4:3–40, 1974. Reprinted in 1975 in *Critica* 6 (1972) 43–86.

[23] H.-N. C. neda. Fiction and reality: Their fundamental connections; an essay on the ontology of total experience. *Poetics*, 8:31–62, 1979.

[24] J. T. Nutter. A lexical relation hierarchy. Department of Computer Science Technical Report 89-6, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0106, 1989.

[25] J. T. Nutter. Representing knowledge about words. In *Proceedings of the Second National Conference of AI and Cognitive Science*, Sept. 1989.

[26] J. T. Nutter. Knowledge based lexicons. In D. Kumar, editor, *Current Trends in SNePS*, pages 97–106. Springer-Verlag Lecture Notes in Artificial Intelligence, No. 437, Berlin, 1990.

[27] J. T. Nutter, E. A. Fox, and M. W. Evens. Building a lexicon from machine-readable dictionaries for improved information retrieval. In *The Dynamic Text: 16th ALLC and 9th ICCH International Conferences*, June 1989.

[28] T. Parsons. *Nonexistent Objects*. Yale University Press, New Haven, 1980.

[29] S. L. Peters and W. J. Rapaport. Superordinate and basic level categories in discourse: Memory and context. In *Proceedings of the Twelvth Annual Conference of the Cognitive Science Society*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.

[30] S. L. Peters and S. C. Shapiro. A representation for natural category systems. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, pages 379–390. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[31] S. L. Peters and S. C. Shapiro. A representation for natural category systems. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 140–146. Morgan Kaufmann, San Mateo, CA, 1987.

[32] S. L. Peters, S. C. Shapiro, and W. J. Rapaport. Flexible natural language processing and Roschian category theory. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, pages 125–131. Lawrence Erlbaum Associates, Hillsdale, NJ, 1988.

[33] W. V. O. Quine. Notes on existence and necessity. *Journal of Philosophy*, 40:113–127, 1943.

[34] W. J. Rapaport. Non-existent objects and epistemological ontology. *Grazer Philosophische Studien*, 25/26:61–95, 1985/1986.

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

[36] W. J. Rapaport. Representing fiction in sneps. In D. Kumar, editor, *Current Trends in SNePS*, pages 107–121. Springer-Verlag Lecture Notes in Artificial Intelligence, No. 437, Berlin, 1990.

[37] W. J. Rapaport. Predication, fiction, and artificial intelligence. *Topoi*, 10:79–111, 1991.

[38] W. J. Rapaport, E. M. Segal, S. C. Shapiro, D. A. Zubin, G. A. Bruder, J. F. Duchan, M. J. Almeida, J. H. Daniels, M. M. Galbraith, J. M. Wiebe, and A. H. Yuhan. Deictic centers and the cognitive structure of narrative comprehension. Technical Report 89-01, Department of Computer Science, SUNY at Buffalo,, March 1989.

[39] W. J. Rapaport, E. M. Segal, S. C. Shapiro, D. A. Zubin, G. A. Bruder, J. F. Duchan, and D. M. Mark. Cognitive and computer systems for understanding narrative text. Technical Report 89-07, Department of Computer Science, SUNY at Buffalo,, August 1989.

[40] S. C. Shapiro. The MIND system: A data structure for semantic information processing. Technical Report R-837-PR, The Rand Corporation, Santa Monica, CA, August 1971.

[41] S. C. Shapiro. A net structure for semantic information storage, deduction and retrieval. In *Proceedings of the Second International Joint Conference on Artificial Intelligence*, pages 512–523. Morgan Kaufmann, San Mateo, CA, 1971.

[42] S. C. Shapiro. Compiling deduction rules from a semantic network into a set of processes. In *Abstracts of Workshop on Automatic Deduction*, Cambridge, MA, 1977. MIT. Abstract only.

[43] 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.

[44] S. C. Shapiro. Numerical quantifiers and their use in reasoning with negative information. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 791–796. Morgan Kaufmann, San Mateo, CA, 1979.

[45] 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.

[46] S. C. Shapiro. Using non-standard connectives and quantifiers for representing deduction rules in a semantic network, 1979. Invited paper presented at Current Aspects of AI Research, a seminar held at the Electrotechnical Laboratory, Tokyo.

[47] S. C. Shapiro. COCCI: A deductive semantic network program for solving microbiology unknowns. Technical Report 173, Department of Computer Science, SUNY at Buffalo, 1981.

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

[49] 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.

[50] S. C. Shapiro. Cables, paths and "subconscious" reasoning in propositional semantic networks. In J. Sowa, editor, *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 137–156. Morgan Kaufmann, Los Altos, CA, 1991.

[51] S. C. Shapiro, D. Kumar, and S. Ali. A propositional network approach to plans and plan recognition. In A. Maier, editor, *Proceedings of the 1988 Workshop on Plan Recognition*. Morgan Kaufmann, San Mateo, CA, 1990.

[52] S. C. Shapiro, J. P. Martins, and D. P. McKay. Bi-directional inference. In *Proceedings of the Fourth Annual Meeting of the Cognitive Science Society*, pages 90–93, Ann Arbor, MI, 1982.

[53] S. C. Shapiro and D. P. McKay. Inference with recursive rules. In *Proceedings of the First Annual National Conference on Artificial Intelligence*, pages 151–153. Morgan Kaufmann, San Mateo, CA, 1980.

[54] 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.

[55] S. C. Shapiro and W. J. Rapaport. Models and minds: Knowledge representation for natural-language competence. In R. Cummins and J. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 215–259. MIT Press, Cambridge, MA, 1991.

[56] S. C. Shapiro, S. N. Srihari, J. Geller, and M. Taie. A fault diagnosis system based on an integrated knowledge base. *IEEE Software*, 3(2):48–49, 1986.

[57] S. C. Shapiro, S. N. Srihari, M.-R. Taie, and J. Geller. VMES: a network-based versatile maintenance expert system. In *Applications of Artificial Intelligence to Engineering Problems: Proceedings of The 1st International Conference*, pages 925–936. Springer–Verlag, Berlin, 1986.

[58] S. C. Shapiro and G. H. Woodmansee. A net structure for semantic information storage, deduction and retrieval. In *Proceedings of the First International Joint Conference on Artificial Intelligence*, pages 325–346. Morgan Kaufmann, San Mateo, CA, 1969.

[59] S. C. Shapiro, G. H. Woodmansee, and M. W. Kreuger. A semantic associational memory net that learns and answers questions (SAMENLAQ). Technical Report 8, Computer Sciences Department, University of Wisconsin, Madison, WI, 1968.

[60] S. C. Shapiro, B. Woolf, D. Kumar, S. S. Ali, P. Sibun, D. Forster, S. Anderson, J. Pustejovesky, and J. Haas. Discussing, using, and recognizing plans–Project Report. Technical report, North-East Artificial Intelligence Consortium, 1989.

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

[62] R. K. Srihari and W. J. Rapaport. Extracting visual information from text: Using captions to label human faces in newspaper photographs. In *Proceedings of the 11th Annual Conference of the Cognitive Society*, pages 364–371. Lawrence Erlbaum Associates, 1989.

[63] R. K. Srihari and W. J. Rapaport. Combining linguistic and pictorial information: Using captions to interpret newspaper photographs. In D. Kumar, editor, *Current Trends in SNePS*, pages 85–95. Springer-Verlag Lecture Notes in Artificial Intelligence, No. 437, Berlin, 1990.

[64] P. van Inwagen. Creatures of fiction. *American Philosophical Quarterly*, 74:299–308, 1977.

[65] J. M. Wiebe and W. J. Rapaport. Representing *de re* and *de dicto* belief reports in discourse and narrative. *Proceedings of the IEEE*, 74(10):1405–1413, October 1986.

[66] W. A. Woods. What's in a link. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding*, pages 35–82. Academic Press, New York, 1975. Reprinted in [2, pp. 218–241].

[67] R. W. Wyatt. The representation of opaque contexts. Technical Report 89-14, State University of New York at Buffalo, Buffalo, NY, 1989.

[68] R. W. Wyatt. Kinds of opacity and their representations. In D. Kumar, editor, *Current Trends in SNePS*, pages 123–144. Springer-Verlag Lecture Notes in Artificial Intelligence, No. 437, Berlin, 1990.