

EXTENDING THE EXPRESSIVE POWER OF SEMANTIC NETWORKS

L. K. Schubert
University of Alberta
Edmonton, Alberta

Abstract. "Factual knowledge" used by natural language processing systems can be conveniently represented in the form of semantic networks. Compared to a "linear" representation such as that of the Predicate Calculus however, semantic networks present special problems with respect to the use of logical connectives, quantifiers, descriptions, and certain other constructions. Systematic solutions to these problems will be proposed, in the form of extensions to a more or less conventional network notation. Predicate Calculus translations of network propositions will frequently be given for comparison, to illustrate the close kinship of the two forms of representation.

I. Introduction

Semantic networks (or nets) mean different things to different people. They are variously thought of as diagrams on paper, as abstract sets of n-tuples of some sort, as data structures in computers, or even as information structures in brains. My concern here will be with semantic nets as graphical analogues of data structures representing "facts" in a computer system for understanding natural language. They aid both in the formulation and exposition of the data structures they resemble. Examples of such graphical aids are found in the work of Quillian (1968, 1969), Palme (1971), Schank (1972, 1973), Simmons & Bruce (1971), Anderson & Bower (1973), Hendrix et al. (1973), Rumelhart et al. (1973), Mylopoulos et al. (1973), and many other writers. Semantic nets are also used to advantage in the mechanization of other forms of understanding, particularly scene understanding, e.g., by Winston (1970), Guzman (1971), and Firschein & Fischler (1971).

The informal and disparate ways in which semantic nets have been used preclude their precise definition in a nonrestrictive way. However, they have generally shared the following characteristics:

- (1) Particular as well as general concepts are represented as labeled or unlabeled nodes of a graph.
- (2) Propositions consist of subgraphs with links to a predicative concept and to a suitable number of conceptual arguments for the predicate. Explicit proposition nodes are sometimes introduced as points of attachment for these links, and as units on which propositional operators (e.g., "knows that") can operate.
- (3) Duplication of nodes denoting the same concept is avoided. Thus several arcs associated with several distinct propositions may share the same concept node. Such nodes are usually regarded as corresponding to a unique computer storage location, i.e., the entry point for accessing knowledge about that concept. Similarly proposition nodes are regarded as unique.

In comparison with Predicate Calculus encodings of factual knowledge, semantic nets seem more natural and understandable. This is due to the one-to-one correspondence between nodes and the concepts they denote, to the clustering about a particular node of propositions about a particular thing, and to the visual immediacy of "interrelationships" among concepts, i.e., their

connections via sequences of propositional links. These properties of semantic nets aid in the design of comparison algorithms, such as that of Quillian (1968, 1969) for finding intersection nodes for two related concepts, or that of Winston (1970) for comparing two complex scene descriptions. Certain kinds of deductive inference also appear to be facilitated by the network representation (Sandewall, 1970).

Having acknowledged some advantages of semantic nets over the Predicate Calculus representation, I should like to emphasize that I regard the two forms of representation as closely akin¹. I will often supply Predicate Calculus equivalents of network propositions in order to illustrate their near-isomorphism. Furthermore, semantic networks proposed so far have been expressively weaker than Predicate Calculus, particularly in their handling of quantification and of higher-order statements. In the following sections I will develop a network representation which permits the use of n-ary predicates (n=1,2,3...), logical connectives, unrestricted quantification (including quantification over predicates), lambda abstraction, and nonextensional operators such as belief and counterfactual implication. The representation easily accommodates propositions of the type encoded by Quillian (1968, 1969), Winston (1970), Schank (1972), and Rumelhart et al. (1972) in their networks. Comparison with network representations used by these and other authors are made as far as space permits. Sec. II introduces the basic propositional notation, and Secs. III-V progressively extend the power of the notation.

II. Atomic Propositions

The basic node type in the notation to be developed is the concept node. Concept nodes may denote individuals such as John, Canada, a particular chair, or a particular real number; they may denote sets such as a set of children, a set of numbers, or a set of properties; or they may denote predicative concepts such as (the universal concept) chair, red, honest, virtue, larger than, in front of, between, or gives. Nodes may be labeled with names for the concepts they denote, e.g., John, chair, chair1, chair2; ordinary attributive terms such as "chair" are reserved for the corresponding universal concepts, while numerically suffixed words such as "chair1" are used for particular instances of the concepts.

The smallest unit of information in a semantic net is the atomic proposition. An atomic proposition consists of a proposition node, a PRED link to a predicative node, and links to a suitable number of concept nodes serving as arguments of the predicate. The argument links are marked in some systematic way, e.g., A, B, C, etc., to distinguish the first, second, third, etc., arguments. Examples are shown in Fig. 1(a)-(c), along with their Predicate Calculus representations. All nodes in Fig. 1(a)-(c) are regarded as type nodes in Quillian's (1968) sense and correspond to unique storage locations. Note that the links in a proposition are directed from the proposition node to the components of the proposition. The only significance of this convention is that it ensures nonambiguity of the

¹ Formal logical representations are often wrongly maligned for supposedly committing the designer to the application of syntactically oriented uniform inference procedures. This criticism confuses the language of logic with its calculus. Nothing whatever prevents the application of heuristic or plausible inference routines to Predicate Calculus assertions. Indeed, FLANNER-like systems combine heuristic inference procedures with a restricted form of Predicate Calculus in the data base.

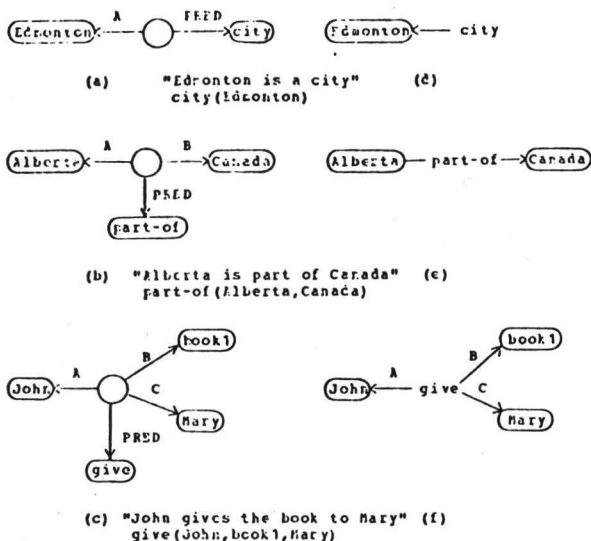


Fig. 1. Atomic propositions, in full and abbreviated.

network syntax. In a computer implementation the links could be reversed or two-way, depending on computational needs.

The propositional diagrams may be simplified as follows. Any explicit proposition node along with its link to the predicative node may be replaced by a predicate token, viz., the (nonencircled) name of the predicate. Since predicate tokens implicitly establish proposition nodes, separate tokens must be used in separate propositions, even if the predicates involved are the same. Another permissible simplification of the diagrams is the omission of link markers when the predicate is monadic (i.e., denotes a property) or dyadic (i.e., denotes a binary relation); in the dyadic case the first and second arguments are then distinguished by omitting the arrowhead on the link to the first argument. The simplified diagrams for the propositions in Fig. 1(a)-(c) are shown in Fig. 1(d)-(f). I will usually opt for the simplified notation in the sequel, except in diagramming certain higher-order constructions.

The proposed propositional notation is closely related to various extant notations. Fig 1(e) is essentially in the style of Winston (1970), although Winston does not introduce proposition nodes as distinct from concept nodes. I regard Fig. 1(d) as the proper monadic analogue of the dyadic notation. Diagrams 1(a)-(c) closely resemble the propositional graphs of Rumelhart et al. (1972). Figs. 2 and 3 indicate how the present conventions relate to those of Quillian (1969) and Schank (1972) respectively. A fuller comparison with a discussion of "cases" can be found in Schubert (1974).

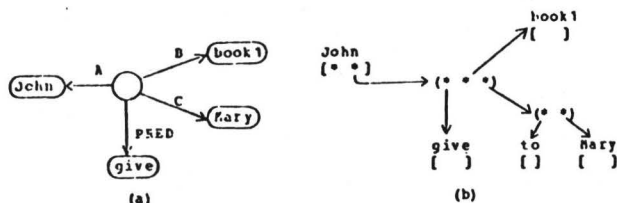


Fig. 2. Comparison with TLC notation "John gives the book to Mary"

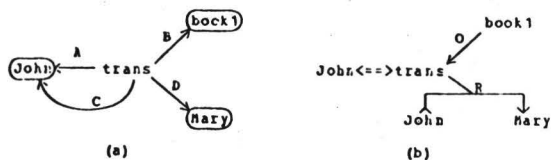


Fig. 3. Comparison with Schank's notation "John transfers the book from John to Mary"

III. Logical Connectives

In most varieties of semantic nets very little use is made of logical connectives. There is little need for conjunction, since usually all propositions in the net are assumed to be asserted, and of course this is equivalent to assertion of their conjunction. That several researchers have chosen to do without disjunction as well is perhaps traceable to the fact that assertion of "pVq" is in a sense only half as informative as assertion of any of the binary conjuncts which imply it (p&q, ~p&q, or p&~q), yet is just as bulky.

Nevertheless disjunction and other connectives are commonplace in ordinary discourse and in any case they are needed for truth-functional completeness. Now everyone who uses semantic nets employs some sort of negation device and of course negation together with conjunction is truth-functionally complete. The problem with most of the negation devices, however, is that they are applicable to atomic sentences only (e.g., putting "not" in front of a predicate, or crossing off a subject-predicate link); and negation of atoms together with conjunction is not truth-functionally complete. It is quite clear what the alternatives are. If we want to restrict negation to atoms, we need to introduce an additional logical connective (e.g., disjunction or implication). If we want to stay with negation and conjunction, we have to extend the negation convention so that it is applicable to conjuncts. In either case we need to create graphical entities which correspond to composite sentences composed of arbitrarily many atomic sentences. The obvious solution lies in the introduction of explicit nodes for logical compounds of propositions (or open sentences), with graphical links to the components. Fig. 4 illustrates the formation of disjunctions by the use of graphical links to tokens of the disjunction operator. The net states "Mary is not at home; she is either at school, or on the playground, or at the zoo; if she is not at school, her mother will be angry". I

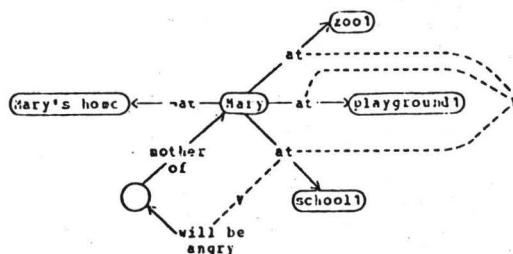


Fig. 4. "Mary is not at home; she is either at school, or on the playground, or at the zoo; if she is not at school, her mother will be angry."

will assume that the operator "V" takes one or more operands. Its general form is shown in Fig. 5.

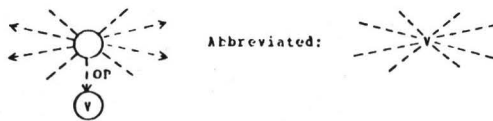


Fig. 5. Generalized disjunction.

Broker lines are used for the operator-operand links to make the logical compounds visually distinguishable. Note that no distinguishing markers are needed on the links, since disjunction is a symmetrical operation. The arrowheads can be dropped when there is no ambiguity, i.e., when the operands are not themselves logical compounds.

The use of "will be" in Fig. 4 as a modifier of "angry" is an evasive manoeuvre, serving to avoid discussion of time. Of course none of the predicates appearing illustratively in this paper are proposed as primitives in an understanding system. The "at" in Fig. 4 is an abbreviation for "at", which shows "at" as a monadic operator on the place-holder "at" for the proposition "at(Mary, Mary's home)".

If desired, other logical connectives can be introduced in exactly the same way. For example, it would have been more natural to render "If Mary is not at school her mother will be angry" by means of implication instead of disjunction, even though this requires the use of an extra negation operator². A generalized implication operator is shown in Fig. 6. This allows for a conjunct of



Fig. 6. Generalized material implication.

arbitrarily many antecedents and a conjunct of arbitrarily many consequents. No labels are needed in the abbreviated notation if consequent and antecedent links are shown emerging fore and aft of the implication symbol respectively. Equivalence is defined analogously (symbol $\Leftrightarrow$), allowing arbitrary sets of conjuncts to be equivalenced.

For a semantic net containing logical compounds, we must revise the usual convention of regarding all propositions in the net as asserted. The convention I will adopt is that the complete semantic net asserts exactly those propositions which are not constituents of compound propositions (i.e., operands of connectives or modal operators). Graphically this means that exactly those propositions are asserted which are not pointed to. Thus in Fig. 4, for example, "Mary is not at home" and "Mary is at school or her mother will be angry" are asserted whereas "Mary is at the zoo" and "Mary is at school" are not. This raises the question of how to assert a proposition which is also a constituent of a compound proposition. First, for logical compounds this need never occur. For example, if a constituent p of a disjunction $p \vee q \vee r$ is known to be assertable, then that entire disjunction can be

replaced by the proposition p and the alternatives $q \vee r$ deleted. The reason is that $p \vee (p \vee q \vee r) \Leftrightarrow p$. Similar simplifications result if a constituent of any logical compound is asserted. For propositional attitudes, causes, intentions and the like, however, we may indeed want to assert a constituent independently of the compound. In this case we can use disjunction with a single operand,

$V \rightarrow p$, as a way of saying "p holds". Since the "compound" proposition established by the token V is not pointed to, it is automatically asserted. Alternatively we could use conjunction of a single proposition (with an explicit "&") or even double negation to the same effect. Examples are shown in Fig. 7. The "beliefs" diagrammed in this figure

(John) — believes —> happy — (Mary)

(a) "John believes that Mary is happy"

(John) — believes —> happy — (Mary)

(b) "John believes that Mary is happy and she is"

(John) — believes —> happy — (Mary)

(c) "John believes that Mary is happy but she isn't"

(John) — believes —> ----> happy — (Mary)

(d) "John believes that Mary isn't happy but she is"

Fig. 7. Asserting propositions by means of monadic disjunction

are examples of "propositional attitudes"; they are governed by modal operators about which I will have a little more to say in Sec. V.

I will conclude this section with some remarks on existing notations. Quillian (1968) used a "hopping arrow" to conjoin or disjoin sets of propositions. However, any such arrow was associated with a particular subject, and as one of the disjunctions in Fig. 4 illustrates, disjoined propositions need not have any subject (or object) in common. Winston (1970) restricted himself to implicit conjunction plus negation of atoms, although he obtained some of the effect of disjunction by means of a "may-be" operator. Rumelhart et al. (1972) state that they allow arbitrary compound propositions in the internal representation, but in their graphical notation they allow only chaining together of propositions making up an "episode". Schank (1972, 1973) makes little attempt to deal graphically with logically connected propositions, as he is usually not concerned with displaying more than 2 or 3 related propositions. He uses negation of atomic propositions and places connectives in the spaces between propositional subgraphs ("conceptualizations") to indicate their logical relations. Anderson & Bower (1973) could easily have introduced unrestricted binary connectives since they use explicit proposition nodes, but it is not clear to me whether or not they did. They discuss disjunction only in connection with checking the semantic net for the presence of either of two propositions, rather than inserting explicit disjunctions. As for implication, it appears that they regard their particular manner of using subset relations as giving the full power of implication. All of their examples, however, involve atomic antecedents, and it is not obvious how a sentence like "The customs official detained all bearded men who were wearing beads" would be

² I am taking a rather literal interpretation of the sentence, ignoring the implicit causal proposition.

represented, in which "man", "bearded", and "wearing beads" are implicative antecedents. Note that the given sentence must be distinguished from both "A number of bearded men wearing beads were detained by the customs official" and "All of those detained by the customs official were bearded men wearing beads".

IV. Quantifiers

It is important to have logical quantifiers within semantic net notation for several reasons: many statements of ordinary discourse involve quantifiers ("He called every day but the phone was always busy"); the representation of general knowledge in declarative form requires quantifiers ("All children like sweets"); the definition of complex concepts requires quantifiers ("At all times when an individual is walking some foot of that individual is touching the ground..."); and definite descriptions of sets require quantifiers ("the people of Canada").

Yet the treatment of quantifiers in semantic nets has generally been rather cursory. Often quantifiers are regarded as monadic modifiers of concept nodes, indicative of "how many there are" of that item (i.e., set cardinality). Universal quantifiers are then attached in the same way, even though the logical operator Λ ("for all") is not at all indicative of cardinality. The only systematic attempts to include quantifiers in semantic nets of which I am aware are those of Palme (1971) and Anderson & Bower (1973). Palme's symbolism is based on Sandewall's (1970) analysis of property-structures. In that approach quantifiers are attached singly or in pairs to predicates, e.g., to symbolize a transformation from a binary relation R on individuals to a binary relation on sets $\Lambda R \Leftrightarrow \lambda X Y [(\Lambda x)(\Lambda y) \text{ member}(x, X) \ \& \ \text{member}(y, Y) \Rightarrow R(x, y)]$. However, this doesn't allow for 3 or more quantifiers in a proposition ("Any politician can fool some of the people all of the time"). Anderson & Bower's treatment is not entirely satisfactory either. First, there is a difficulty about quantified implicative propositions with complex antecedents, which stems from the deficient implicative notation. One way of characterizing the difficulty is that there is no apparent method for distinguishing definite and indefinite set descriptions, such as "the set of all dogs that chase cats" versus "a set of dogs that chase cats", and hence no way of distinguishing statements involving such descriptions antecedently. Second, the rule that quantifiers in the subject position of the propositional tree have the largest scope leads to difficulties. In particular, it is awkward to raise a propositional object to the level of maximum scope, as Anderson & Bower are well aware. For example, they are forced to render "There is a cat that all dogs chase" as "There is a cat distinguished by the fact that all dogs chase it", where "distinguished by" is a pseudo-predicate introduced to allow objects to be raised to subject position. Additional problems are encountered in quantification over time, since in Anderson & Bower's notation the "time context" includes an entire proposition in its scope. For example, there is no direct way to handle the distinction between "There is always someone there" and "There is someone who is always there". Finally Anderson & Bower neglect to supply quantifier precedence rules when the scope of a quantifier extends over logical combinations of propositions, as it certainly may.

The notation I will propose is analogous to quantifier-free normal form in Predicate Calculus. Propositions are expressed in prenex form (i.e., quantifiers have maximum scope), existentially quantified variables are Skolemized, and universal quantification is implicit. This first of all requires a distinction between existentially and universally quantified nodes. A simple method is

the use of solid lines for existentially quantified concept nodes (as in all previous figures), and broken lines for universally quantified nodes. Graphical Skolemization then consists of linking each existentially quantified node to all universally quantified nodes on which it depends (i.e., whose universal quantifiers precede the existential quantifier in prenex form). I shall use dotted lines for these dependency links for easy distinguishability from propositional and logical links. For example, "All dogs chase some cat" is represented as shown in Fig. 8(a). In Predicate Calculus notation this is

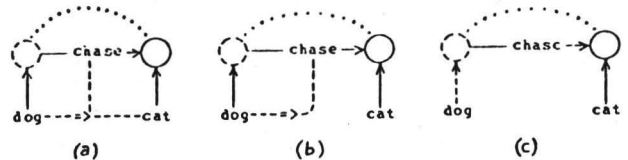


Fig. 8. "All dogs chase some cat"

$(\Lambda x)[\text{dog}(x) \Rightarrow (\exists y)[\text{cat}(y) \ \& \ \text{chase}(x, y)]]$, or $\text{dog}(x) \Rightarrow [\text{cat}(f(x)) \ \& \ \text{chase}(x, f(x))]$, Skolemized. Now if we can assume $(\exists y)\text{cat}(y)$, i.e., there is at least one cat (or alternatively, that there is at least one dog), then this becomes $\text{cat}(f(x)) \ \& \ [\text{dog}(x) \Rightarrow \text{chase}(x, f(x))]$ which corresponds to the slightly simpler diagram shown in Fig. 8(b). Here the "cat" proposition is no longer regarded as a consequent of the "dog" proposition. This type of simplification is often appropriate for encoding natural language statements, since we do not usually communicate in terms of propositions which are trivially true by virtue of the nonexistence of their referents³. A further simplification is indicated in Fig. 8(c), which is based on the implicit notation for implication explained in Schubert (1974). The diagram for the proposition "There is a cat which all dogs chase" differs from Fig. 8 only in the absence of the dependency link between the "cat" and "dog" nodes. As another example consider the proposition "There is always someone there". This might be diagrammed as in Fig. 9(a), after adding the assumption that there is at least one moment of time. Note that a time argument has been added

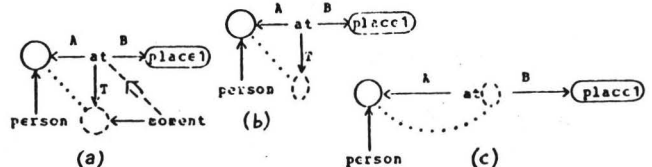


Fig. 9. "There is always someone there"

to the predicate "at". The representation seems a little unnatural because of the need to restrict the universally quantified node to "moments" and the implicative dependence of the main relationship on that restriction. This suggests that it would be more natural to use a many-sorted logic, with each argument of each predicate restricted to a particular subdomain of the domain of discourse, and with time forming a distinct sort. Then quantification over a time argument would automatically be restricted to moments of time. This is the course I will take, at least nominally. Sortal distinctions could be made explicit by using distinct node shapes for distinct sorts, or by using a distinct kind of argument marker on argument pointers to entities of each distinct sort, e.g., always using OBJ_i (i = 1, 2, ...) to point to arguments of the sort "physical object". In fact the latter technique is used by Rumelhart et al. (1972). Rather than

³ Which is not to say that we do not communicate about nonexistent entities.

committing myself to a particular method here, I shall leave sortal distinctions implicit, except in the case of time. Time calls for special treatment because of its central importance in structuring events. I will use pairs of parentheses instead of circles for moments of time and mark pointers to moments of time "T". A name for a moment of time can be placed between the parentheses. Broken parentheses indicate universally quantified time variables. With these conventions Fig. 9(a) can be redrawn as shown in Fig. 9(b). "There is someone who is always there" would merely lack the dependency link of Fig. 9. The representation of time dependence can be simplified further with the aid of two additional conventions. The first is to place the time at which a proposition holds directly alongside the predicate token of that proposition, as in Fig. 9(c). The second is to use time intervals as time arguments (in some suitable sense of interval - e.g., see Bruce, 1972). If T is a time interval, then a proposition of the form $P(x, y, \dots, T)$ is taken as an abbreviation of $(\forall t)[\text{member}(t, T) \Rightarrow P(x, y, \dots, t)]$. In the graphical notation I will use square brackets instead of parentheses for nodes denoting time intervals, and mark pointers (if any) to such nodes "TI" instead of "T". Two equivalent ways of representing "The sun rose" are shown in Fig. 10. In both versions quantification

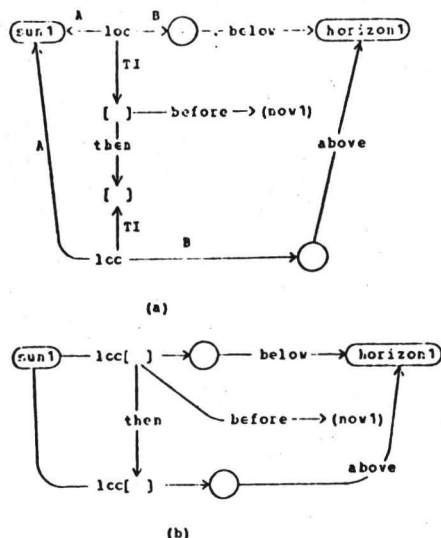


Fig. 10. "The sun rose".

over moments of time has been entirely suppressed by means of the interval notation. "Then" is taken as a relation between adjacent time intervals, and "before" as a relation between moments or intervals of time. For complete sets of time relations see Findler & Chen (1971), Bruce (1972), or Schank et al. (1973).

Many higher-order constructions are easily expressed with the notation already introduced. For example, "John has all of his father's faults, and carelessness is one of them" is represented as shown in Fig. 11. Note that both the abbreviated and unabbreviated notation for propositions have been used here. Three of the proposition nodes are explicit, while "father-of" and the two occurrences of "fault" establish three implicit proposition nodes. The higher-order predicate is of course "fault", and the universally quantified node should be read "for all predicates". Here the implicit restriction of quantification to appropriate sorts has been extended to apply to types as well, i.e., since "fault" is a predicate on predicates, its argument in any proposition is implicitly restricted to the type "predicate".

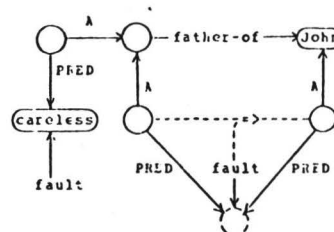


Fig. 11. "John has all of his father's faults, and carelessness is one of them"

Here I should remark that past claims about the equivalence of certain varieties of semantic net notation to second (or higher) order logic have not been backed by adequate quantificational apparatus. Statements about predicates alone do not demonstrate a second-order capability, as they can be made in a many-sorted first-order logic.

Finally I should point out that the logical quantifiers are unsuited for expressing many natural language quantifiers. I believe that natural language quantifiers not readily expressible in terms of the logical quantifiers, such as "several", "many", "most of", "a few more than", etc., can be handled systematically by the use of (fuzzy) properties of set cardinality and relations between set cardinalities, plus standard set relations such as set inclusion. For examples see Schubert (1974) and Cercone & Schubert (1974).

V. Further Extensions

In this section I will briefly illustrate the representation of definite and indefinite descriptions, lambda expressions, and modal constructions. A fuller exposition is given elsewhere (Schubert, 1974).

Russellian descriptions of individuals and sets are illustrated in Figs. 12 and 13 respectively (e.g., Quine, 1960). The descriptions

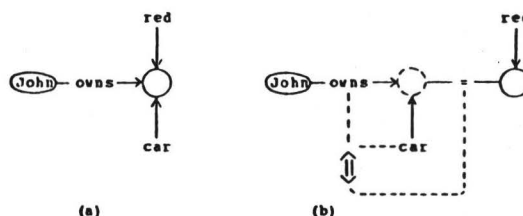


Fig. 12. Indefinite and definite descriptions

- (a) "John owns a red car"
 $(\exists x)[\text{owns}(\text{John}, x) \ \& \ \text{red}(x) \ \& \ \text{car}(x)]$
 (b) "John's car is red"
 $(\exists x)(\forall y)[\text{owns}(\text{John}, y) \ \& \ \text{car}(y) \Leftrightarrow x=y] \ \& \ \text{red}(x)$

are underlined in the figure legends, and Predicate Calculus translations are given for comparison. A nongeneral but frequently useful method of abbreviating such descriptions is proposed in Schubert (1974).

In diagramming Russellian descriptions, I am not doing so as an uncritical advocate of Russell's theory. Certainly it is incorrect to regard referential descriptions as nothing but disguised assertions (see Strawson, 1950). The role I envisage for Russellian descriptions in a semantic net-based language understanding system is best seen by example. Suppose that the language understanding system is told "John's car is red". The system would first look for an existing node to use as referent of "John's car". We need not be concerned with the details of this search here, noting only that if it succeeds, no new

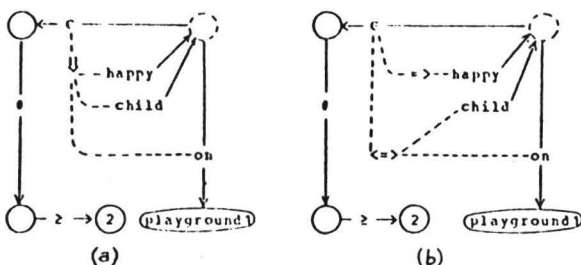


Fig. 13. Indefinite and definite descriptions of sets.

- (a) "There are happy children on the playground"
 $(\exists S) [\exists x \exists y (\text{member}(x, S) \Rightarrow \text{happy}(x) \ \& \ \text{child}(y) \ \& \ \text{on}(y, \text{playground}))]$
- (b) "The children on the playground are happy"
 $(\exists S) [\exists x \exists y (\text{member}(x, S) \Rightarrow \text{child}(x) \ \& \ \text{on}(x, \text{playground})) \ \& \ [\text{member}(y, S) \Rightarrow \text{happy}(y)]]$

description is placed in memory. Only "red" is predicated about the node found (provided this predication is consistent with prior knowledge). If the search fails, however, the system creates a new existentially quantified node with the attached proposition that this is the one and only car John has, possibly in a sense of "has" determined by context. This Russellian existence assertion is placed in memory, provided it is consistent with prior knowledge. If all goes well, the net of Fig. 12(b) is the final result. But suppose that the existence assertion contradicts prior information to the effect that John has no car. Then the attempt to insert a new node in memory is aborted, and thus no referent for the predication "is red" is made available. Seeking a resolution of the difficulty encountered, the system might well respond "But I thought John doesn't have a car". Thus I see "presupposition failure" as an operational phenomenon, rather than as a logical phenomenon calling for complex model-theoretic manoeuvring (e.g., Fisk, 1969).

The problem raised by lambda abstraction and modal operators such as "necessarily", "knows", "wants", and "causes" is that statements involving them cannot in general be converted to prenex form. Thus the scope conventions introduced for quantifiers are inadequate. The conventions can be generalized by allowing (dotted) scope dependency links between proposition nodes and quantified nodes, to indicate that the quantifiers lie within the scopes of the operators which form those propositions. The new links supplement the dotted links already introduced to indicate the relative scopes of quantifiers. Two examples are given in Fig. 14. The lambda pointer in Fig. 14(a) from the

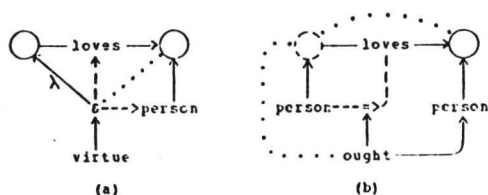


Fig. 14. Lambda abstraction and modal operators.

- (a) "Loving someone is a virtue"
 $\text{virtue}(\lambda x [(\exists y) \text{person}(y) \ \& \ \text{loves}(x, y)])$
- (b) "Everyone ought to love someone"
 $\text{ought}((\exists x) (\exists y) \text{person}(y) \ \& \ [\text{person}(x) \Rightarrow \text{loves}(x, y)])$

conjunctive proposition node to the first argument of "loves" abstracts the predicate "loving someone" from the open sentence "x loves someone".

The operands of the modal operator "ought" in Fig. 14(b) are regarded as implicitly conjoined. Deletion of some of the scope dependency links in the two diagrams yields other (less plausible) readings of the sentences.

VI. Concluding Remarks

I have put forward some views on the proper construction and interpretation of semantic networks, and suggested systematic methods for expressing operations such as logical combination and quantification in these networks. The proposed representation is a fairly direct extension of several quite successful, superficially disparate representations, such as Schank's conceptualizations, Winston's descriptive nets, and Sandewall's property-structures. Consequently the computational processes that create and utilize their data structures can readily be adapted to structures based on the present representation. This indicates that the increased expressive power the suggested representation provides should be of real value in the design of understanding systems.

A variety of problems in the representation of knowledge could raise additional questions of notational adequacy. Examples are the handling of vagueness, events, the lexical meanings of complex concepts, and overall knowledge organization. We are currently studying some of these problems in the context of a practical attempt to design a language understanding system.

Acknowledgements

This work was intended to establish a notational basis for research on a "state-based" conceptual representation by Nick Cercone and myself. A great deal of credit goes to Nick for contributing to the ideas herein. I am indebted to G. Prideaux, C. G. Morgan and F. J. Pelletier for pointing me to some relevant literature, and to the latter two for several very helpful discussions. I am also grateful for J. R. Sampson's and K. V. Wilson's comments on the original manuscript, which hopefully have led to a more readable version. The research was supported by the National Research Council of Canada under Operating Grant no. A8818.

References

- Anderson, J. and Eower, G. (1973). Human Associative Memory, V. H. Winston and Sons, Washington, D.C.
- Bruce, E. C. (1972). "A Model for Temporal Referencing and its Application in a Question Answering Program," Artificial Intelligence 3, pp 1-26.
- Cercone, N. and Schubert, L. K. (1974). "A Sketch of State-Based Conceptual Representation," Department of Computing Science, TF74-19, University of Alberta, Edmonton, Alberta. See also this volume.
- Findler, N. and Chen, D. (1971). "On the Problems of Time, Retrieval of Temporal Relations, Causality, and Co-existence," Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 531-545.
- Firschein, O. and Fischler, M. A. (1971). "A Study in Descriptive Representation of Pictorial Data," Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 258-269.

- Fisk, M. (1969). "A Modal Analogue of Free Logic," in The Logical Way of Doing Things, Lambert, K. (ed.), Yale University Press, New Haven, Connecticut, pp147-184.
- Guzman, A. (1971). "Analysis of Curved Line Drawings Using Context and Global Information," in Machine Intelligence 6, Meltzer, B. and Michie, D. (eds.), American Elsevier Publishing Company, New York, New York, pp 325-375.
- Hendrix, G. G., Thompson, C. W., and Slocum, J. (1973). "Language Processing via Canonical Verbs and Semantic Models," Advance Papers of Third International Joint Conference on Artificial Intelligence, Stanford Research Institute, Menlo Park, California, pp 262-269.
- Mylopoulos, J., Badler, N., Melli, L., and Roussoyculos, N. (1973). "1.PAF: A ASNOBOL-Based Programming Language for Artificial Intelligence Applications," Advance Papers of Third International Joint Conference on Artificial Intelligence, Stanford Research Institute, Menlo Park, California, pp 691-696.
- Palme, J. (1971). "Making Computers Understand Natural Language," in Artificial Intelligence and Heuristic Programming, Findler, N. V. & Meltzer, B. (eds.), American Elsevier Publishing Company, New York, New York, pp 199-244.
- Quillian, M. (1968). "Semantic Memory," in Semantic Information Processing, Minsky, M. (ed.), MIT Press, Cambridge, Massachusetts, pp 227-270.
- Quillian, M. (1969). "The Teachable language Comprehender," Comm. Assoc. Comp. Mach. 12, pp 459-475.
- Quine, W. V. A. (1960). Word and Object, MIT Press, Cambridge, Massachusetts.
- Rumelhart, D., Lindsay, P., and Norman, D. (1972). "A Process Model for Long Term Memory," in Organization of Memory, Tulving, E., and Donaldson, W. (eds.), Academic Press, New York, New York, pp 198-221.
- Sandewall, E. J. (1970). "A Set-Oriented Property-Structure Representation for Binary Relations," in Machine Intelligence 5, Meltzer, B. and Michie, D. (eds.), American Elsevier Publishing Company, New York, New York, pp 237-252.
- Schank, R. (1972). "Conceptual Dependency: A Theory of Natural Language Understanding," Cogn. Psych. 3, pp 552-631.
- Schank, R. (1973). "The Fourteen Primitive Actions and Their Inferences," Stanford AI Project, Memo AIM-183, Stanford University, Stanford, California.
- Schank, R., Goldman, N., Rieger, C., and Riesbeck, C. (1973). "Margie: Memory, Analysis, Response Generation, and Inference on English," Advance Papers of Third International Joint Conference on Artificial Intelligence, Stanford Research Institute, Menlo Park, California, pp 255-261.
- Simmons, R. F., and Bruce, B. C. (1971). "Some Relations Between Predicate Calculus and Semantic Net Representation of Discourse," Advance Papers of Second International Joint Conference on Artificial Intelligence, The British Computer Society, London, England, pp 524-530.
- Strawson, P. F. (1950). "On Referring," Mind 59, pp 320-344.
- Winston, P. (1970). "Learning Structural Descriptions from Examples," PhD Thesis, MIT, MAC-TR-76, Cambridge, Massachusetts.