

Extending the Expressive Power of Semantic Networks

L. K. Schubert

University of Alberta, Edmonton, Alberta, Canada

Recommended by Bernard Meltzer

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.

1. 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. My point of view is that of a heuristic programmer. For discussions of the place of semantic nets in psychological theories of cognition see, for example, Anderson and Bower [2] and Wilson [42, 43].

Semantic nets aid both in the formulation and exposition of the computer data structures they resemble. Examples of such graphical aids are found in the work of Quillian [23, 24], Palme [22], Schank [32, 33], Simmons and Bruce [37], Anderson and Bower [2], Hendrix et al. [13], Rumelhart et al. [27], Mylopoulos et al. [21] 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 [44], Guzman [12] and Firschein and Fischler [10].

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. Arguments of n -ary predicates may be distinguished by the use of link labels, distinct linkage types, or binary decomposition of the predicates.

(3) Duplication of nodes denoting the same concept is avoided. Thus several arcs associated with several distinct propositions may impinge on a 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" between concepts, i.e., their connections via sequences of propositional links. By virtue of these properties semantic nets lead more readily to "associative" and comparison algorithms of the type described by Quillian [23, 24] for finding intersection nodes for two related concepts, and by Winston [44] for comparing two complex scene descriptions. Certain kinds of deductive inference also appear to be facilitated by the network representation (Sandewall [30]).

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 akin.¹ 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 ($n = 1, 2, 3, \dots$), logical connectives, unrestricted quantification

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 rules to Predicate Calculus assertions. Indeed, PLANNER-like systems combine heuristic inference procedures with a restricted form of Predicate Calculus in the data base.

(including quantification over predicates), lambda abstraction, and modal operators such as belief and counterfactual implication. The representation easily accommodates propositions of the type encoded by Quillian [23, 24], Winston [44], Schank [32] and Rumelhart et al. [27] in their networks. Comparison with network representations used by these and other authors are made as far as space permits. Section 2 introduces the basic propositional notation and includes a necessary digression on the role of "cases" in semantic nets. Sections 3-8 progressively extend the power of the notation. All extensions are analogues of standard notational devices employed in various first-order or higher-order Predicate Calculi. Although no formal semantics are given for the network notation, its correspondence to standard logical notation indicates how such semantics could be formulated.

2. 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 [23] 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 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, *distinct* tokens must be used in distinct propositions, even if the predicates involved are the same.

Artificial Intelligence 7 (1976), 163-198

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.

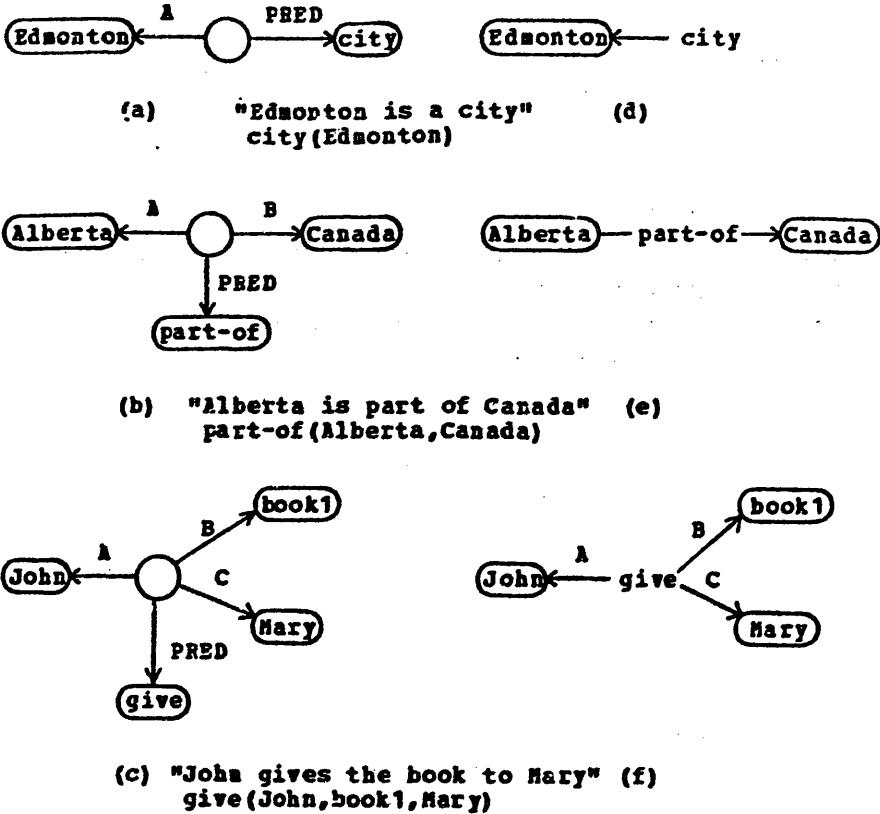


FIG. 1. Atomic propositions, in full and abbreviated.

The proposed propositional notation is closely related to various extant notations. Fig. 1(e) is essentially in the style of Winston [44] except that the two argument links have not been joined into a single directed arc. However, Winston did not introduce proposition nodes; this omission would hamper the addition of logical connectives and propositional operators to his notation. I regard Fig. 1(d) as the proper monadic analogue of the dyadic notation. Most exponents of semantic nets artificially convert monadic to dyadic predicates by introducing such predicates as ISA and HAS-PROPERTY which can be thought of as applying their second arguments to their first arguments. This strikes me as a Procrustean measure for *Artificial Intelligence* 7 (1976), 163–198

enforcing the dyadic propositional format, whose theoretical or practical benefits are unclear. Another method of paraphrasing monadic predication is by means of set membership, e.g., by expressing "Turing is human" as "Turing belongs to the class of human beings". This is less objectionable because set membership and set inclusion predicates have other worthwhile uses that justify their introduction into the formal vocabulary. In particular, the usual set terminology would be sorely missed in the representation of many quite ordinary propositions, such as: "Several members of the audience fell asleep and one walked out." However, while paraphrasing "x has property P" as "x is in the set of P's" is acceptable, paraphrasing "property P" as "the set of P's" is in general unacceptable. Observe, for example, the contrast in the meanings of the two statements (based on Carnap [5]) "The class of human beings is the same as the class of featherless bipeds", and "The property 'human' is the same as the property 'featherless biped'." The lambda abstraction mechanism to be introduced in Section 7 will enable us to represent statements of the latter type.

Diagrams 1(a)–(c) closely resemble the propositional graphs of Rumelhart et al. [27]. Less obviously, diagrams (c) and (f) are quite close to Quillian's [24] and Schank's [32] representations respectively. The similarity can be brought to light by slightly rearranging their representations. First compare Fig. 2(a) (repeated from Fig. 1(c)) with Fig. 2(b). The latter is based on

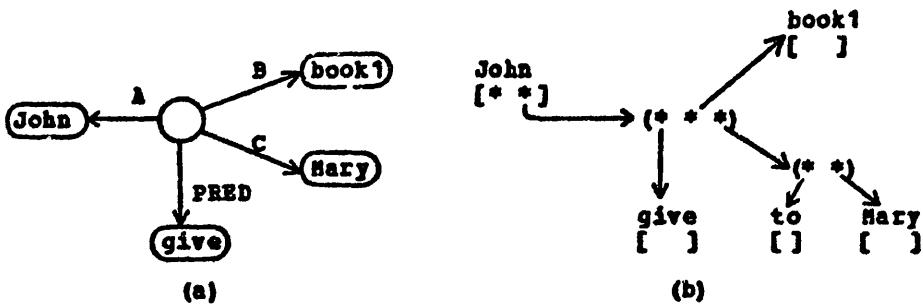


FIG. 2. Comparison with TLC notation "John gives the book to Mary".

Quillian's notation for his Teachable Language Comprehender [24]. Argument order is indicated by the order of the asterisks which are the placeholders for the components of the proposition (except for the first argument of the predicate, which is taken as the subject about which the proposition is made and is linked to the proposition node as a whole). The square brackets contain place-holders for the superset (general category) of the corresponding concept and for any number of properties of (propositions about) the concept. Quillian tried to do away with binary predicates by regarding additional arguments as modifiers. This accounts for the modifying property "to Mary"

in the diagram. This seems to stretch the notion of a property quite inordinately and indicates that the restriction to binary predicates is unnatural. Extension of the TLC notation to allow for n -ary predicates would make it nearly isomorphic with the present notation.

To bring out the relationship of Schank's [32] notation to the present notation, we paraphrase "John gives the book to Mary" as "John transfers the book from himself to Mary", thus replacing the 3-place predicate "give" by the 4-place predicate "transfer".² The representations in the present notation and in Schank's notation (apart from a slight geometrical rearrangement) are shown in Fig. 3. It would not be difficult to devise an algorithm for translating either notation into the other. The special appeal of Schank's notation lies in the significance one may attach to the linkage types and labels: agent, object, and recipient in this example, i.e., conceptual "cases" of the type advocated by Fillmore [8] and used by Schank [32], Simmons [35], Abelson [1] and others.

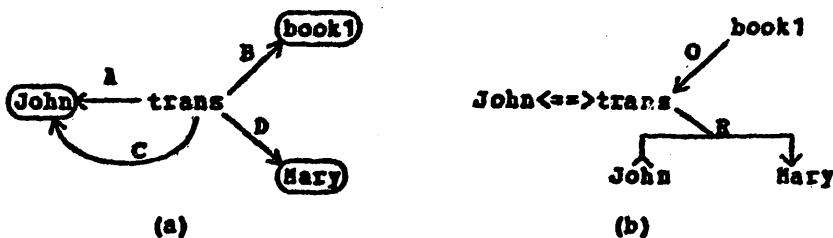


FIG. 3. Comparison with Schank's notation "John transfers the book from John to Mary".

The notion of case derives from the systematic similarities between the roles played by the arguments of many predicates in relation to those predicates. For example, action predicates as different as "give" and "tell" both involve an agent, an entity directly acted on (an object in "give", a message in "tell"), and an indirect object or recipient. From such similarities some researchers influenced by Fillmore have concluded that the cases themselves are *conceptually primitive relations* between the entities participating in an action and the action as a whole. This view might lead one to redraw Fig. 3(a) as shown in Fig. 4. The central node now denotes a specific action or process (a notion that seems a little obscure) modified by *five* distinct atomic propositions: transfer(b), agent(John, b), object(book1, b), donor(John, b) and recipient(Mary, b), where b is the action being modified. Note that cases are here regarded as genuine relational concepts rather than as mnemonic markers for ranking arguments of predicates (hence the lower case spelling). Subscribing to this view of cases, Simmons [35, 36] writes down propositions such as agent(John,run) as explicit instances of case

² None of the predicates appearing illustratively in this paper are proposed as primitives in an understanding system.

relations and argues that the meanings of case relations can be defined by specifying their implications. As examples of implications he suggests that $\text{agent}(a,b)$ implies that a is animate, b is a process, and b has some effect on a . The last of these implications is difficult to evaluate without additional explication. The first and second merely delimit the domain and range of the agent relation respectively and thus cannot go far towards defining its meaning. Moreover, the domain restrictions associated with a particular case vary from one predicate to the next. Schank's [33] objective case requires a physical object in the context of a PTRANS (physical transfer), an abstract relationship such as ownership in the context of an ATRANS (abstract transfer), and a proposition in the context of an MTRANS (mental transfer).

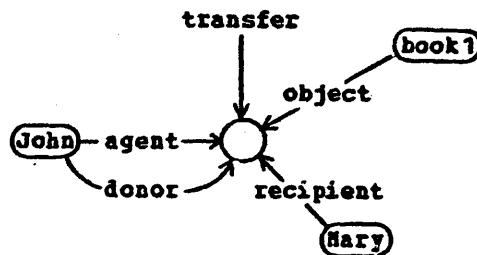


FIG. 4. Case relations for "John gives the book to Mary".

To my mind, a genuine understanding of the structural analogies between different sorts of actions requires analysis of such actions in terms of more elementary events. For example, the proposition "John handed the book to Mary" could be analysed along the following lines. John was in a purposive state, the purpose being "the book is in Mary's hand" (or perhaps "the book is near Mary"). As an eventual (and intentional) result of John's purposive state, the book was in one of John's hands and was moving towards Mary. As an eventual result of this state the book was near Mary. The analysis of "John gave the book to Mary" would be similar but would contain less specific state information. The description of events in terms of successive states is developed in more detail in Cercone and Schubert [6]. Case relations can be understood as complex nonprimitive relations derived from such causally and teleologically related sequences of states. For example, the notion of an agent seems to depend in part on causal priority of a state of the supposed agent in the sequence of states under consideration, in part on the extent to which purposive behaviour can be ascribed to the supposed agent in general, and in part on the extent to which the particular sequence of states which he initiated can be assumed to be intentional on his part.

Thus I claim that there is a level of semantic representation which is semantically "deeper" or more elementary than the level of case-structured action proposition. That is not to say that such action propositions should

be banished from semantic nets; on the contrary it seems plausible that several levels of representation may be required for full language comprehension, ranging from "low-resolution" representations relatively close to linguistic surface structure to "high-resolution" representations that are relatively language-free. What I am calling into question is any claim that cases are the fundamental semantic relations and that Fig. 4 (for example) therefore is the proper interpretation of Fig. 3(a). Instead I regard case labels such as in Fig. 3(a) as mere mnemonic markers, although indicative of complex relations predicable of sequences of states at a more elemental level of analysis.³ To prevent confusion between markers and predicate names I will consistently use capital letters for the former and lower case letters for the latter. I will occasionally use suggestive argument markers such as WHO, WHAT in place of the noncommittal markers A, B,

Finally, Fig. 5 illustrates yet another way of diagramming the sentence "John gives the book to Mary" which is quite unlike any of those of Figs. 1-4. This is based loosely on Anderson and Bower [2]:⁴ the predicate has been

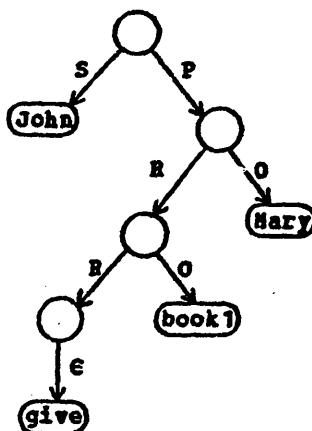


FIG. 5. Binary decomposition of "John gives the book to Mary".

decomposed as a binary tree, i.e., the predicate is applied to one argument at a time. The four unlabeled nodes from the bottom of the tree to the root correspond to the binding of zero, one, two, and all three arguments of the predicate. Links are labeled S (subject), P (predicate), R (relation), O (object) or with set membership or set inclusion symbols.

³ An important exception is Schank's *instrumental* case, which expresses a genuine relation between action propositions. I believe that this relation can be further analyzed in terms of intentional states of agents and causal relations between events.

⁴ Actually Anderson and Bower render this sentence as "John transfers the book, causing Mary to possess it". However, this trades the problem of accommodating the recipient of "give" for that of accommodating the destination of the transfer, which Anderson and Bower neglect.

The first question raised by this notation is whether any semantic significance is to be attached to the link labels. In view of the earlier comments about case labels, I take Anderson and Bower's link labels to be mnemonic distinguishing marks, possibly indicative of complex relations at some deeper level of analysis. Here the indicated relations seem to be dependent not only on elementary sequences of states and properties of the concepts involved but also on discourse context. The notion of a propositional subject, for instance, would appear to depend upon momentary discourse theme or focus to some extent.

A second feature of the notation calling for explanation is the set-theoretic form of predication (e.g., the "membership in give" in Fig. 5). At first glance we might take this as expressing membership in a set of n -tuples, by analogy with the standard paraphrase of monadic predication in terms of membership in a set of 1-tuples (individuals). Let us accept this view for the moment and turn to another distinctive feature of Anderson and Bower's notation, namely the extra nodes corresponding to the binding of some but not all arguments of a predicate. These nodes serve as arguments of adverbial modifiers and as such are perhaps indispensable. However, since they are tied ultimately to relations at the bottom of a propositional tree via set membership, they presumably denote single n -tuples; I cannot see how to interpret adverbial modification of an n -tuple. A possible way of resolving this difficulty may be to regard general concepts such as "give" as *classes* of sets of n -tuples (i.e., classes of relations). In that case the adverbial modifiers would operate on relations (sets of n -tuples) rather than single n -tuples, and could be understood as selecting subsets of these relations (cf. Reichenbach [26]).

Unfortunately Anderson and Bower do not justify or explain these features of their system very thoroughly. I am inclined to think that at a sufficiently detailed level of analysis adverbial modification can be expressed without decomposition of predicates. For example, a manner adverb such as "quickly" applied to an action or activity could ultimately be expressed by a comparative proposition about the time taken for the action or for a certain portion of the activity. However, I should point out that the lambda abstraction technique to be introduced in Section 7 allows selective binding of predicate variables, making it possible in principle to treat modifiers (adjectival as well as adverbial) in a manner similar to that of Anderson and Bower.

3. 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 " $p \vee q$ " is in a sense only half as informative as assertion of any of the binary conjuncts which imply it ($p \& q$, $\neg p \& q$, or $p \& \neg q$), yet is just as bulky.

Nevertheless disjunction and other connectives are commonplace in ordinary discourse and in any case 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.

The simplest logical operator is the negation operator. Fig. 6 shows its use in forming the denial (labeled p2) of the proposition "John loves Mary" (labeled p1). The structure of p2 is entirely analogous to the structure of any

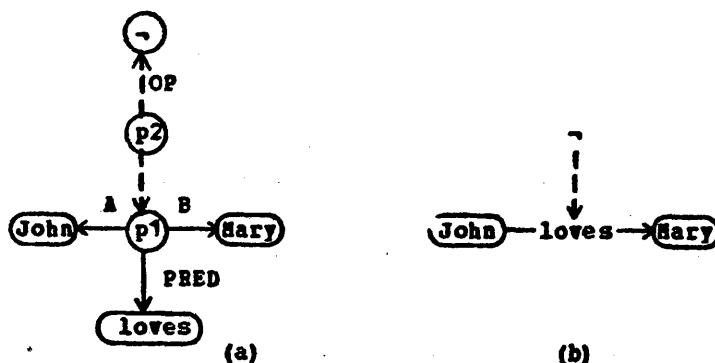


FIG. 6. "John does not love Mary", in full and abbreviated.

monadic predication, such as that in Fig. 1(a). Instead of a PRED link we have an OP (operator) link and instead of a conceptual argument we have a propositional operand. Broken links are used for both links to make the logical compound visually distinguishable. The abbreviated version of p2 is also entirely analogous to the abbreviated form of monadic predication, with an *operator token* implicitly establishing the compound proposition node.

Fig. 7 illustrates disjunctive notation for the proposition "John, Jim, or Artificial Intelligence 7 (1976), 163-198

Joe loves Mary". The compound proposition p4 is formed by application of the operator "v" to the propositional operands p1, p2 and p3. In general it is convenient to assume that the disjunction operator takes one or more operands. Note that no distinguishing markers are needed on the links, since disjunction is a commutative and associative operation.

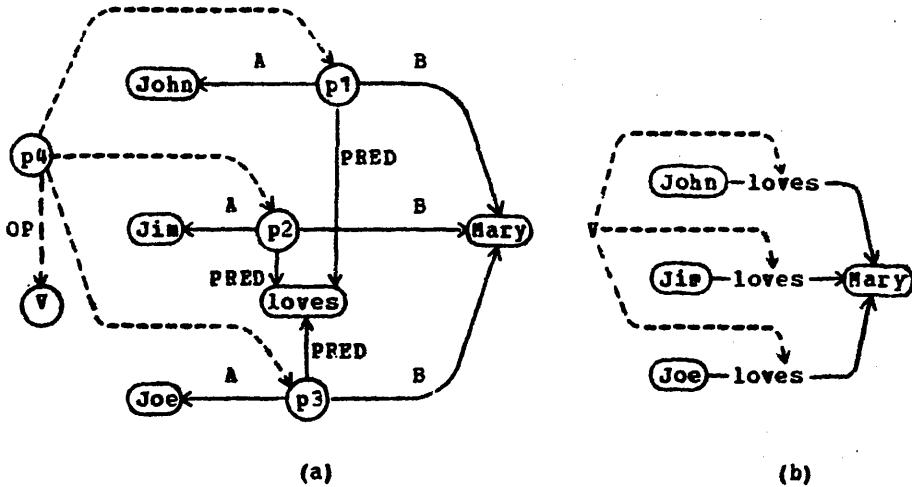


FIG. 7. "John, Jim or Joe loves Mary", in full and abbreviated.

If desired, other logical connectives can be introduced in exactly the same way. For example, a generalized implication operator is shown in Fig. 8. This allows for a conjunct of 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. Examples involving these forms of implication and equivalence will be seen later on.

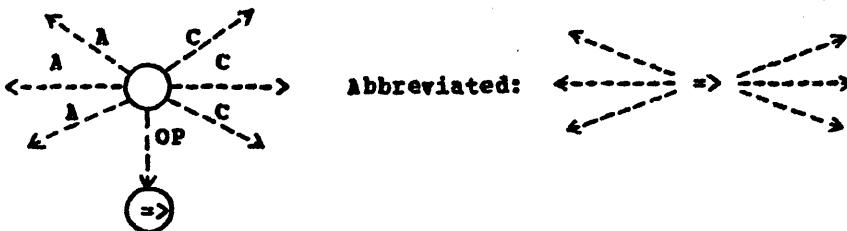


FIG. 8. Generalized material implication.

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. 7 the compound proposition "John, Jim, or Joe loves Mary" is asserted whereas the constituents "John loves Mary", "Jim loves Mary", and "Joe loves Mary" 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 does not arise. 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 $p \& (p \vee q \vee r) \Leftrightarrow p$. 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 previous work on semantic nets this has been accomplished by *repeating* the constituent, as would be necessary in Predicate Calculus. For example, to assert that John believes that Mary is happy and that in fact Mary is happy one would write "believes(John,happy(Mary)) & happy(Mary)". For networks, however, there is a more elegant method which completely avoids multiple occurrences of propositions. It is to use disjunction with a *single* operand,

$$\vee \text{ --- } \rightarrow p,$$

as a way of saying " p holds". Since the "compound" proposition established by the token \vee is not pointed to, it is automatically asserted even though p might be embedded in any number of compound propositions. 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. 9. The "beliefs" diagrammed in this figure are examples of "propositional attitudes"; they are governed by modal operators about which I will have a little more to say in Section 8.

I will conclude this section with some remarks on existing notations. Quillian [23] used a "hopping arrow" to conjoin or disjoin sets of propositions. However, any such arrow was associated with a particular *subject*, and as the disjunction in Fig. 7 illustrates, disjoined propositions need not have a common subject. Indeed they need not share *any* node ("Someone is in the house or Mary forgot to close the gate"). Winston [44] 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. [27] 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 [32, 33] 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.

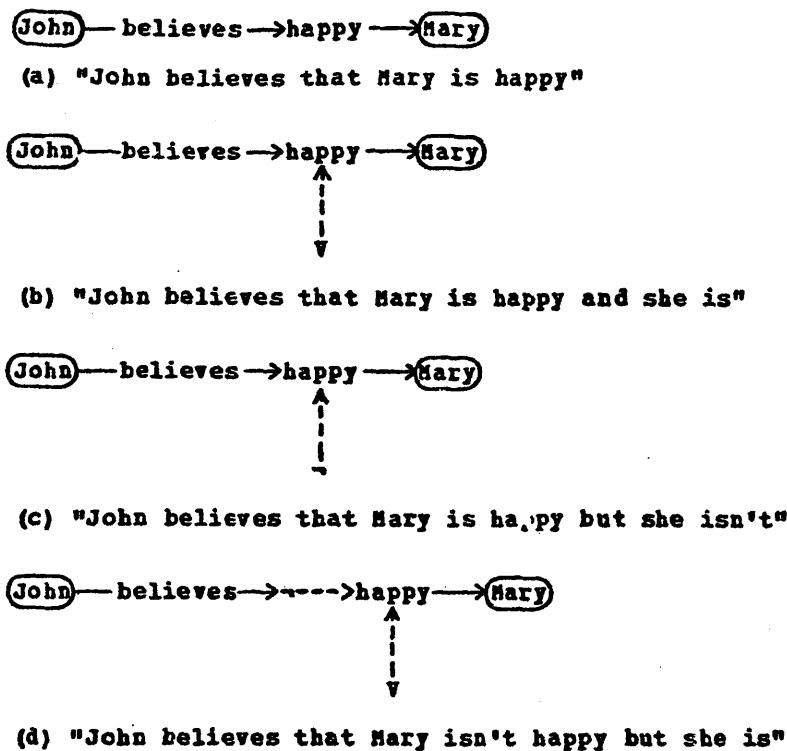


FIG. 9. Asserting propositions by means of monadic disjunction.

He uses negation of atomic propositions and places connectives in the spaces between propositional subgraphs ("conceptualizations") to indicate their logical relations. Anderson and Bower [2] are able to introduce arbitrary monadic and dyadic connectives since they use explicit proposition nodes. The same difficulty of interpretation obscures their use of such connectives, however, as that remarked upon earlier. For example, they represent implication by "membership in imply" without further explanation and give only one illustration. They paraphrase all other implicative propositions in terms of subset relations. These latter examples involve atomic antecedents only, and it is not obvious how a sentence like "The customs official detained all bearded men who were wearing beads" would be 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".

4. 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 \forall ("for all") is not at all indicative of cardinality.

The notation I will propose initially is analogous to quantifier-free normal form in Predicate Calculus. Propositions are 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. Two simple propositions involving universal quantification are shown in Fig. 10. They are "Everyone likes Mary" and "Every child likes anything sweet" and require one and two universal quantifiers respectively.

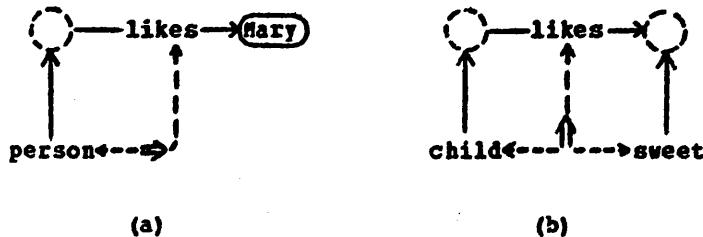


FIG. 10. (a) "Everyone likes Mary" $(\forall x)[\text{person}(x) \Rightarrow \text{likes}(x, \text{Mary})]$; (b) "Every child likes anything sweet" $(\forall x)(\forall y)[\text{child}(x) \ \& \ \text{sweet}(y) \Rightarrow \text{likes}(x, y)]$.

Graphical Skolemization 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. The arrows are directed from the universally quantified nodes to the dependent existentially quantified nodes, i.e., they indicate the direction of *decreasing* quantifier scope. For example, "All dogs chase some cat" is represented as shown in Fig. 11(a). In Predicate Calculus notation this is

$$(\forall x)\{\text{dog}(x) \Rightarrow (\exists y)[\text{cat}(y) \ \& \ \text{chase}(x, y)]\}$$

$$\text{or } \text{dog}(x) \Rightarrow [\text{cat}(f(x)) \ \& \ \text{chase}(x, f(x))],$$

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. 11(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 non-existence of their referents.⁵ The diagram for the proposition "There is a cat

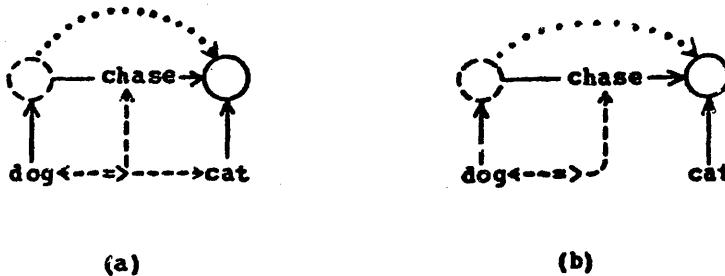


FIG. 11. "All dogs chase some cat".

which all dogs chase" differs from Fig. 11 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. 12(a), after adding the assumption that there is at



FIG. 12. "There is always someone there".

least one moment of time. Note that a time argument has been added 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

⁵ Which is not to say that we do not communicate about nonexistent entities.

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 ($i = 1, 2, \dots$) to point to arguments of the sort "physical object". In fact the latter technique is used by Rumelhart et al. [27]. Rather than 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. 12(a) can be redrawn as shown in Fig. 12(b). "There is someone who is always there" would merely lack the dependency link of Fig. 12(b). Additional conventions for time will be introduced in Section 5.

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. 13. 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.

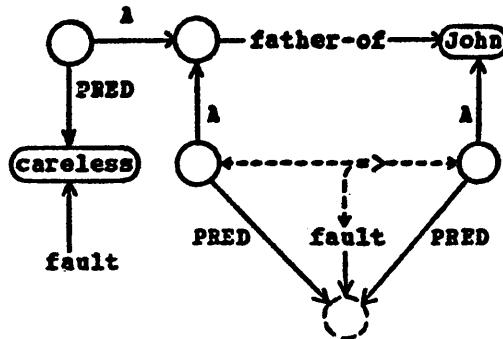


FIG. 13. "John has all of his father's faults, and carelessness is one of them".

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".

The proposed method of representing quantification is applicable only to propositions in prenex form. Since conversion to prenex form sometimes leads *Artificial Intelligence* 7 (1976), 163-198

to loss of clarity, I shall now introduce a generalization of the notation which allows arbitrary embedding of quantifiers within the scopes of logical connectives or other propositional operators. The notation turns out to be more than a mere convenience. It will prove essential in Sections 7 and 8 which deal with lambda abstraction and modal operators.

To see how the prenex restriction can be removed consider the following proposition:

$$(\forall x)(\exists y)(\forall z)\{(\forall s)P(s,x) \Rightarrow (\exists t)[Q(t,y,z) \Leftrightarrow (\forall u)(\forall v)(\exists w)R(u,v,w)]\}.$$

We can completely specify the scopes of all quantifiers as follows. First, for each sequence of adjacent quantifiers, we specify the dependence of the existentially quantified variables on the universally quantified variables as before, i.e., y depends on x and w depends on u and v . Second, we specify which variables have their quantifier scopes nested just inside the scopes of which propositional connectives. Thus s and t depend on \Rightarrow , and u , v and w depend on \Leftrightarrow in this sense. Exactly the same kind of dotted dependency link can be used for this second type of scope relationship as for the first, with the arrow going from the embedding proposition to the embedded node. The representation of the above proposition is shown in Fig. 14. By the transitivity

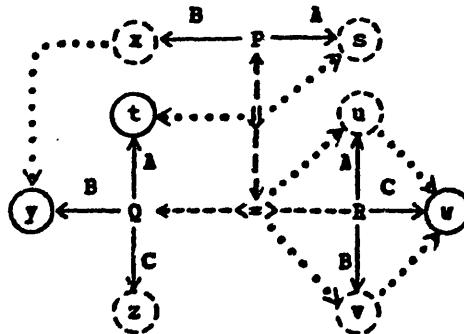


FIG. 14. Indicating quantifier scopes. $(\forall x)(\exists y)(\forall z)\{(\forall s)P(s,x) \Rightarrow (\exists t)[Q(t,y,z) \Leftrightarrow (\forall u)(\forall v)(\exists w)R(u,v,w)]\}$.

of scope inclusion, $(\forall u)$, $(\forall v)$ and $(\exists w)$ lie within the scope of \Rightarrow , since the equivalence proposition is embedded within the implicative proposition. Note that all paths in the net which follow broken and dotted lines in the direction of the arrows are paths of decreasing scope. For example, the path from \Rightarrow to \Leftrightarrow to u to w is a path of decreasing scope. The assumption that all quantifiers have scope compatible with the indicated constraints uniquely determines all scopes.

A more meaningful application of the operator scope conventions is shown in Fig. 15. The diagram expresses the proposition "Mary will receive a scholarship provided that she passes all of the exams". Time has been

ignored for simplicity and the exams in question are assumed to be some predetermined set. "Provided that" has been rendered (correctly, I believe) as "if and only if". The Predicate Calculus version is given in the figure legend, and is seen to be of the logical form $[(\exists x)P(x)] \Leftrightarrow [(\forall y)Q(y)]$. In prenex form this is

$$(\exists v)(\exists w)(\forall x)(\forall y)[[P(x) \Rightarrow Q(y)] \& [Q(w) \Rightarrow P(v)]]$$

which is twice as complicated as the original version and quite incomprehensible when stated in English. Thus the use of the operator scope conventions to retain the original form of the proposition seems justified in this instance.

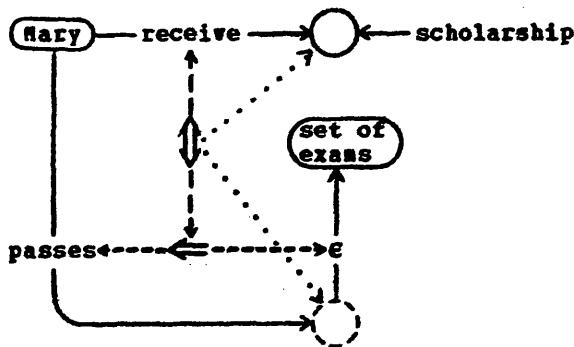


FIG. 15. "Mary will receive a scholarship provided that she passes all of the exams", $[(\exists x)[\text{scholarship}(x) \& \text{receives}(\text{Mary},x)]] \Leftrightarrow [(\forall y)[\text{member}(y,\text{set-of-exams}) \Rightarrow \text{passes}(\text{Mary},y)]]$.

The only prior systematic attempts to include quantifiers in semantic nets of which I am aware are those of Simmons and Bruce [37], Palme [22] and Anderson and Bower [2]. Simmons and Bruce discussed single quantifiers only. Palme's symbolism is based on Sandewall's [30] 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

$$\forall R \forall \lambda \Leftrightarrow \text{lambda } XY[(\forall x)(\forall y) \text{ member}(x,X) \& \text{ member}(y,Y) \Rightarrow R(x,y)].$$

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 and Bower's treatment is not entirely satisfactory either. First, quantified implicative propositions are difficult to formulate and interpret, because of the peculiar way in which connectives and quantifiers are used. For instance, Anderson and Bower's network representation of

$$(\forall x)[\text{dog}(x) \Rightarrow (\exists y)[\text{cat}(y) \& \text{chase}(x,y)]],$$

i.e., all dogs chase some cat, contains 3 universal quantifiers plus subsets of "chase" and "cat". Thus the correspondence to the Predicate Calculus representation is not at all clear. Also there is no apparent method for *Artificial Intelligence* 7 (1976), 163-198

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 implicative statements involving such descriptions antecedently. Further difficulties arise from the rule that quantifiers in the subject position of the propositional tree have the largest scope. In particular, it is awkward to raise a propositional object to the level of maximum scope, as Anderson and 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 and 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 and Bower neglect to supply quantifier precedence rules when the scope of a quantifier extends over logical combinations of propositions, as it certainly may.

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. An example appears in Section 6.

5. Time

Quine [25] deplores the “tiresome bias” shown by ordinary language in the treatment of time and shows how to relegate time to a role on par with that played by position, weight, or colour in the description of events. Thus “It will happen west of here” becomes “It happens after now west of here”, where “happens” is taken *timelessly* and “after”, like “west of”, is regarded as a (timeless) relation between portions of space-time (i.e., events or objects). This formal adjustment, which renders all predications timeless, is in sharp contrast with the approach of McCarthy and Hayes [19] in which all predications about physical objects are *time-dependent* (or rather, situation-dependent, where a “situation” is an instantaneous state of the universe; but this notion virtually coincides with Quine’s conception of a moment of time as an

Artificial Intelligence 7 (1976), 163–198

instantaneous cross-section of the four-dimensional material world). I prefer the “biased” treatment of time, since the “unbiased” treatment leads to rather cumbersome representations of ordinary events and circumstances. For example, the sentence “There is always someone there” becomes “For every moment of time t there is a person x (regarded as a spatiotemporal object) such that the intersection of x with t is located at the intersection of place1 with t ” (where “place1” is the place signified by “there”). The corresponding semantic net is shown in Fig. 16, and should be compared

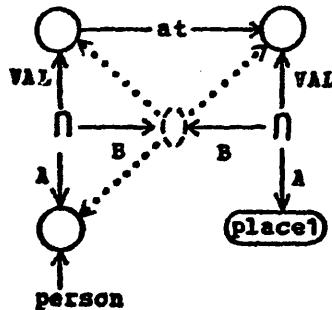


FIG. 16. “There is always someone there”, based on space-like treatment of time.

with Fig. 12 (see also Fig. 17 below). Notice that the new representation directly associates a time with every *participant* in an event or circumstance, rather than associating it with the main relation. Clearly the use of time variables in predicates yields more concise and more natural representations.⁶

Furthermore, I regard time as the only situational or contextual variable that needs to be added to action propositions, unlike Anderson and Bower [2] or Rumelhart et al. [27] for example. Whenever a proposition is regarded as true only within a particular situational context, that context can be made an explicit premise instead of an argument of the predicate. For example, “Mary is livelier with her lovers than with her parents” (an example discussed by Bartsch and Vennemann [3]) can be rephrased as “For all times t and all times s , if Mary is with a lover at time t and if Mary is with her parents at time s , then Mary is livelier at time t than she is at time s ”. Anderson and Bower [2] and also Schank et al. [34] add locale to the time as basic dimensions of events. But locale is not a property of an event as a whole, but a (frequently time-dependent) property of the *participants* in an event. For example, in “John is watching a circling hawk” it is John and the hawk who have locations, not the event. For those who would take a phenomenalist view

⁶ Incidentally, it seems to me that time variables can be taken to denote time in a more intuitive sense (e.g., Newtonian physical time) than cross-sections or states of the universe. McCarthy and Hayes [19] do in fact regard true time as a *function* of states of the universe. As long as this function is one-to-one within any given universe (be it real or hypothetical), states of the universe may be replaced by (true) times without risk of ambiguity.

of John's action, the sentence "the moon is circling the earth" may serve as a better illustration. Thus I give no special consideration to spatial relationships in semantic nets, since the relational notation already introduced can be used to describe the locations of things.

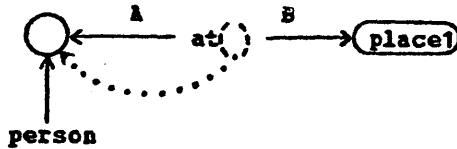
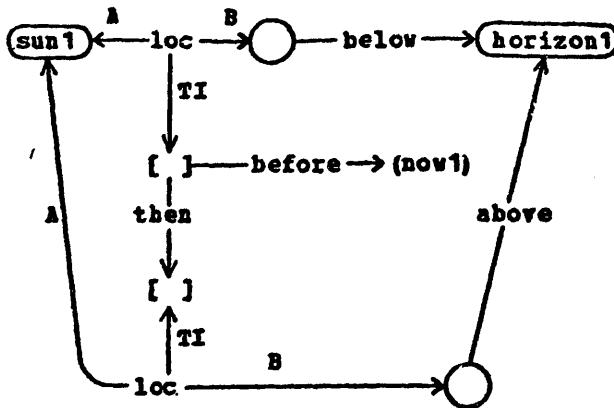
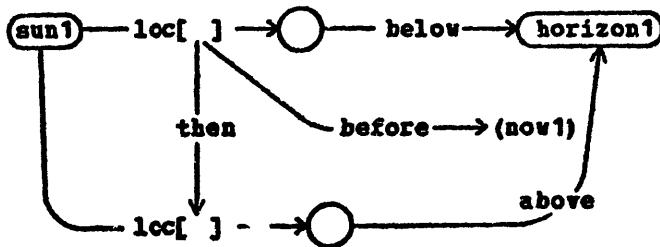


FIG. 17. "There is always someone there".

The following conventions are based on the assignment of a central role to time by the introduction of time variables into all predicates over physical entities. I will permit any time-dependent predicate $P(x, y, \dots, t)$, where t is the time variable, to be used in 3 modes: the instantaneous mode, the interval mode, and the permanent mode.



(a)



(b)

FIG. 18. "The sun rose".

In the *instantaneous mode* any fixed or variable *moment* of time is supplied as temporal argument. Because of the special importance of time arguments I will allow times to be placed immediately next to predicate tokens. For example, Fig. 12(b) can be redrawn as shown in Fig. 17. As before, parentheses are used to symbolize a moment of time and are broken to indicate universal quantification.

In the *interval mode* an interval of time T is supplied as temporal argument (in some suitable sense of "interval", such as Bruce's [4]). The definition in terms of the instantaneous mode is

$$P(x,y, \dots, T) \Leftrightarrow (\forall t)[\text{member}(t,T) \Rightarrow P(x,y, \dots, t)].$$

Argument marker TI and square brackets are used for time intervals to distinguish them from moments of time. In most descriptions of events the interval mode appears to be much more convenient than the instantaneous mode. An example is given in Fig. 18. Fig. 18(b) is the same as Fig. 18(a) except that the time arguments have been placed with their predicate tokens. The specific time "now1" in Fig. 18 is the time of assertion, so that past tense is expressed as "before the time of assertion". "Then" is taken as a relation between contiguous time intervals, and "before" as a relation between moments of intervals of time. For complete sets of time relations see Findler and Chen [9], Bruce [4], or Schank et al. [34].

In the *permanent mode* no temporal argument at all is shown. The illustrations in this paper contain numerous examples. The omission of time arguments from time-dependent predicates is a notational expedient used by many but acknowledged by few. What it means is that the predication is true of the individual concerned over an extensive segment of its life span. For example, "being a girl" is normally a much more enduring property than "being angry", so that the former is usually represented in the permanent mode while the latter is not. Ultimately, however, the time dependence even of relatively enduring properties may need to be taken into account. Consider the following story:

One day Johnny caught a tadpole. He called it "Hugo" and kept it in a jar. A year later, when Johnny set Hugo free, Hugo was a frog.

Question: was Hugo a tadpole or a frog when Johnny caught it? *In a semantic net in which "tadpole" was predicated of Hugo in the permanent mode, this question is not decidable*; Hugo might already have been a frog when Johnny caught it. Thus we need to record that Hugo was a tadpole *at the time Johnny caught it*. More gradual changes linked to growth and aging show that even such terms as "kitten", "cat", "girl", or "woman" are properly regarded as time-dependent. Determining when time-dependence *can* be ignored appears to be a matter of some subtlety. In drawing semantic nets for expository purposes we are of course justified in omitting seemingly unimportant time

Artificial Intelligence 7 (1976), 163-198

dependencies. Such omissions may be justified even in the computerized counterparts of the semantic nets, especially if it is possible to recover most of the missing information through inference. However, the semantic nets must be capable in principle of expression the time limitations of all propositions. This is possible with the proposed method of associating a time with each predicate over physical entities rather than, say, with "episodes" which define more or less complex events.

It would not be difficult to formulate the analogues of the instantaneous, interval, and permanent modes of predication in an approach based entirely on timeless predicates. A time interval would be regarded as a "slice" of the four-dimensional material world perpendicular to the time axis. Its intersection with an individual yields a time-slice of that individual. The analogue of the permanent mode of predication would involve the application of a predicate to a complete four-dimensional individual, on the understanding that the predication might be strictly applicable only to a proper time-slice of that individual.

6. Descriptions

Ordinary discourse abounds with definite and indefinite descriptions such as "the woman who answered the telephone", "John's car", "a big apple", and "several children". In previous work on network representations the distinction between definite and indefinite descriptions has been ignored. The purpose of this section is to propose methods for making this important distinction explicit.

Quine [23] discusses definite and indefinite descriptions of particular things and sets of things, and mentions the use of the Peano-Russell conventions to formalize such descriptions. Then he goes on to show how one can get along in principle with only the two singular terms ($\forall x$) and ($\exists x$), and of course one of these is redundant. Essentially the reductions are based on Russell's theory of descriptions (Russell [28], Whitehead and Russell [41], Russell [29]). I shall rely on analogous reductions in the graphical notation, basing the representation of descriptions on the logical connectives and quantifiers already introduced. I should point out immediately that Russell's theory has come in for much criticism. It is usually faulted for treating reference as a disguised form of assertion, and for arbitrarily filling with falsity the "truth value gaps" that occur when presuppositions conveyed by definite descriptions are violated (Strawson [39]). As I shall shortly indicate, however, these criticisms do not apply to the use of Russell's analysis here proposed. An alternative way of dealing with definite descriptions would be to use a description operator (e.g., Kalish and Montague [16], Sandewall [31] or Moore [20]). The scope of such an operator could be indicated with the generalized scope conventions explained in Section 4. An important advantage

of Russell's analysis, however, is that it allows ambiguities in modal sentences to be explained as quantifier scope ambiguities (Hughes and Cresswell [14]; see also Section 8 below).

The sentences "John's car is red" and "John has a red car" are diagrammed in Fig. 19. The terms "John's car" and "a red car" are examples of definite and indefinite descriptions respectively. A presupposition conveyed by the definite description "John's car" is that John has exactly one car (at least

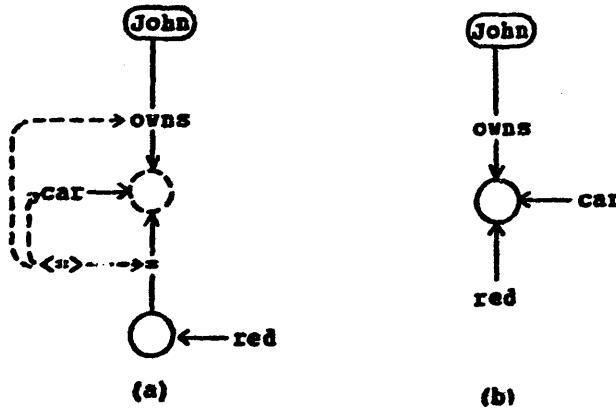


FIG. 19. Definite and indefinite descriptions. (a) "John's car is red"; $(\exists x)(\forall y)[\text{owns}(\text{John}, y) \& \text{car}(y) \leftrightarrow x = y] \& \text{red}(x)$. (b) "John owns a red car"; $(\exists x)[\text{owns}(\text{John}, x) \& \text{car}(x) \& \text{red}(x)]$.

this is true for certain discourse contexts in which the given sentence might occur). This presupposition of unique existence has been made explicit in Fig. 19(a), which asserts that there is a thing which is identical with every car John owns and is red. The diagram in Fig. 19(b) is comparatively simple; it states that there is a thing which John owns, which is red, and which is a car.

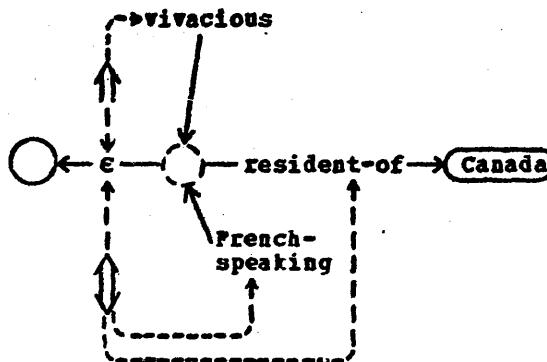


FIG. 20. Definite descriptions of sets. "The French-speaking people of Canada are vivacious" $(\exists S)[(\forall x)[\text{member}(x, S) \leftrightarrow [\text{French-speaking}(x) \& \text{resident-of}(x, \text{Canada})]] \& (\forall x)[\text{member}(x, S) \Rightarrow \text{vivacious}(x)]]$.

Definite and indefinite descriptions of sets are illustrated in Figs. 20 and 21 respectively. The definite description in Fig. 20 is "the French-speaking people of Canada", the existence of whose referent is presupposed. Thus the net states that there is a set such that a thing is in it *if and only if* it is French-speaking and a resident of Canada. The assertion made about this set is that all of its members are vivacious.

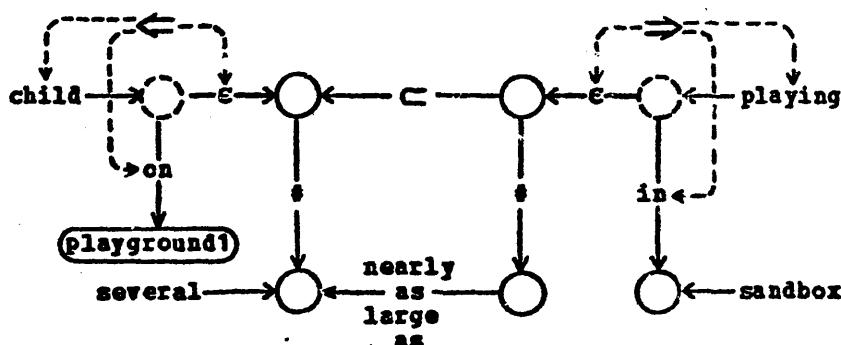


FIG. 21. Indefinite descriptions of sets. "Several children are on the playground; most of them are playing in the sandbox".

The net of Fig. 21 states "Several children are on the playground; most of them are playing in the sandbox". The indefinite descriptions are "several children" and "most of them". In contrast with definite descriptions of sets, indefinite descriptions of sets associate implications, rather than equivalence conditions, with membership in the set. The predicate "#" in the diagram denotes the number of elements of a set, or rather the single-valued relation which holds between a set and the number of its elements. Note that "several" is treated as a (fuzzy?!) monadic predicate on numbers, whereas "most of" is broken down into a subset relation and a (fuzzy?!) relation on pairs of numbers.

I will now indicate by way of an example how definite descriptions might be processed by an understanding system, and how Russell's analysis would be employed in a manner which avoids the usual criticisms levelled against it. The basic idea is that the understanding system would first attempt to interpret definite descriptions *referentially*, and failing that, *attributively*. Referential interpretation means that an *existing* node is selected as the semantic correlate of the description, while attributive interpretation means that a *new* node is created as the semantic correlate of the description and the properties expressed by the description attributed to it. This terminology is based loosely on the distinctions made by Dcnellan [7].

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 description is placed in memory. Only “red” is predicated about the node found (provided that 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 that it is consistent with prior knowledge. If all goes well, the net of Fig. 19(a) 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 place 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., van Fraassen [40] and Fisk [11]). The question of the truth value of propositions with failing presuppositions does not arise, exactly as Strawson [39] requires.

The last kind of definite descriptions, namely definite descriptions of relations, is the subject of the next section.

7. Lambda Abstraction

Certain definite descriptions of predicative concepts such as “(the property which is) John’s only virtue” can already be expressed with the available notation. The diagram for “John’s only virtue is honesty” is shown in Fig. 22. Note that equality is here used as a *second-order* relation, denoting identity of *intension* of the equated predicates (i.e., identity of their truth sets in all possible worlds).

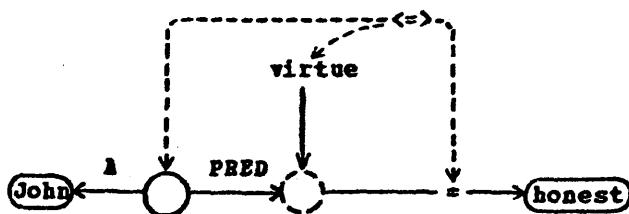


FIG. 22. “John’s only virtue is honesty”; $(\forall P)[P(\text{John}) \ \& \ \text{virtue}(P) \Leftrightarrow P = \text{honest}]$.

The notation so far introduced is inadequate, however, for descriptions of predicative concepts expressed in terms of predicates of the same (rather than higher) type. For example, suppose we wish to say that the property “human” is the same as the property “rational animal”; note that the latter property is of type 1 and is expressed in terms of the type 1 properties “rational” and “animal”. We cannot diagram this statement on the basis of the formula
Artificial Intelligence 7 (1976), 163–198

$(\forall x)[\text{human}(x) \Leftrightarrow \text{rational}(x) \ \& \ \text{animal}(x)]$, since this merely asserts extensional identity (i.e., the set of human beings equals the set of rational animals). The desired statement of intensional identity can be made with the aid of Church's lambda operator. This operator abstracts a predicate from an open sentence by designating certain variables of the sentence as arguments of the predicate. Thus we write

$$\text{human} = \text{lambda } x[\text{rational}(x) \ \& \ \text{animal}(x)].$$

A more interesting example is provided by the sentence "Loving one's neighbours is a virtue", which requires abstraction of the monadic predicate "loves one's neighbours" from the dyadic predicate "loves". This is shown

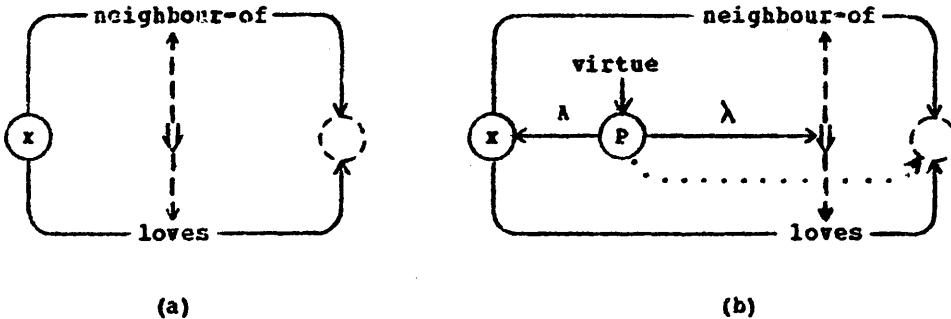


FIG. 23. "Loving one's neighbours is a virtue"; $\text{virtue}\{\text{lambda } x\{(\forall y)[\text{neighbour}(x,y) \Rightarrow \text{loves}(x,y)]\}\}$.

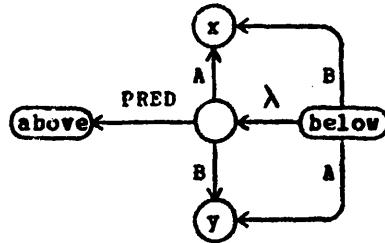


FIG. 24. Definition of "below" as the inverse of "above"; $\text{below} = \text{lambda } xy[\text{above}(y,x)]$.

in Fig. 23, using a graphical analogue of lambda abstraction. In Fig. 23(a) the graph for the proposition "x loves all of his neighbours" is shown. The proposition node for this compound proposition is established by the implication token. In Fig. 23(b) the predicate "loving one's neighbours" (labeled P) has been abstracted from the proposition of Fig. 23(a). In addition the second-order predication "virtue" has been attached to the abstracted predicate. Lambda abstraction is accomplished by means of a *lambda link* from the new predicate node to the proposition from which the predicate is abstracted plus argument links from the predicate node to variables (nodes)

of the proposition, indicating lambda conversion of the variables to arguments of the predicate. The lambda link is labeled "lambda" and the argument links are labeled A, B, . . . (or in some other systematic way) as in propositions. Note in addition the dotted scope dependency link from the abstracted predicate to the universally quantified node. This is an important application of the generalized scope notation introduced towards the end of Section 4. It places the universal quantifier within the scope of the lambda operator. Without the link the net of Fig. 23(b) would incorrectly state⁷

$$(\forall y) \text{virtue}(\text{lambda } x[\text{neighbour}(x,y) \Rightarrow \text{loves}(x,y)]).$$

In general propositions containing lambda abstracts cannot be converted to prenex form, so that the generalized scope notation is a necessity here, rather than a mere convenience. The definition of "below" in terms of "above" by means of lambda abstraction is shown in Fig. 24. The lambda abstraction mechanism provides a general means for defining relations (including *functions*, i.e., single-valued relations) in terms of relations already in the system. The newly defined relations can be used in the same way as those already in the system.

8. Modal Operators and Referential Opacity

Finally we need conventions for representing propositions involving modal operators such as the necessity operator, the belief operator, the causal operator, and the counterfactual conditional. Note that it is only the representation, not the manipulation or formal semantics of modal constructions that is at issue here. However, it is reassuring that the "possible worlds" semantics devised in recent years by modal logicians appears to provide an adequate basis for the formal semantical analysis of modal constructions (e.g., Snyder [38]).

As far as I can see, the only notational problem raised by the introduction of modal operators is that of distinguishing between "opaque" and "transparent" environments generated by such operators. A sentential environment is opaque if replacement of a term by a referentially equivalent term can change the truth value of the sentence, and transparent otherwise. For example, the necessity operator generates an opaque environment in the sentence "9 is necessarily greater than 8", as we cannot replace "9" by the referentially equivalent term "the number of major planets".

In general we can regard a term in an opaque environment as locked into

⁷ Lambda abstraction was handled inadequately in Schubert [46]. First, the proposition node on which lambda operates was itself used to denote the abstracted predicate. This is more concise but unfortunately leads to scope ambiguities if the proposition from which the predicate is abstracted is not in prenex form. Second, the scope dependency links needed to tie quantifiers into the scope of the lambda operator were inadvertently neglected.

the *scope* of the modal operator (necessarily, wants, etc.), as suggested by Isard and Longuet-Higgins [15]. Quantifiers in such terms cannot be extracted to convert the proposition to prenex form. For example, the proposition believes(John, $(\exists x)\text{ghost}(x)$) cannot be paraphrased as $(\exists x)\text{believes}(\text{John}, \text{ghost}(x))$. In words, "John believes that a ghost exists" cannot be paraphrased as "Something exists that John believes to be a ghost". Once again the generalized scope conventions are needed, so that quantifiers can be embedded within the scopes of propositional operators.

The representations of the nonsynonymous sentences "9 is necessarily greater than 8" and "The number of major planets is necessarily greater than 8" are shown in Fig. 25. The necessity operator \square is assumed to take one or more operands. In Fig. 25(b) the dotted line from the necessity operator to

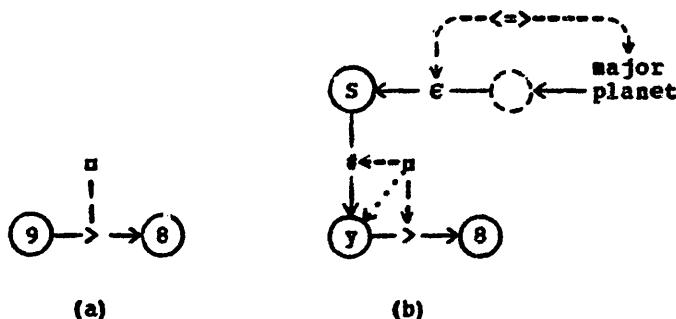


FIG. 25. (a) "9 is necessarily greater than 8"; $\square(9 > 8)$. (b) "The number of major planets is necessarily greater than 8"; $(\exists S)(\forall x)\{\text{member}(x, S) \Leftrightarrow \text{major-planet}(x) \ \& \ \square(\exists y)[\#(S, y) \ \& \ y > 8]\}$.

the node *y* places the existential quantifier for *y* within the scope of the operator. Removal of the dotted line to *y* and of the broken line to the token # would give the (implausible) *transparent* reading of the sentence in which "the number of major planets" is simply taken as a reference to the number 9.

Counterfactual implication can be treated in much the same manner as necessity. In the sentence "If there were a major planet beyond Pluto, the number of major planets would equal 10" the term "the number of major planets" is nonreferential. The representation is shown in Fig. 26. I have borrowed Lewis' symbol $\square \Rightarrow$ to symbolize counterfactual implication (Lewis [17]); however, I am allowing any number of implicitly conjoined antecedents and consequents as operands, much as in the generalized form of material implication.

Many English modal sentences, particularly those involving verbs of propositional attitude such as "wants" or "believes", admit both a transparent and an opaque reading. An example is "John wants to marry the prettiest girl" (a variant taken from Moore [20]). In the transparent reading we can replace "the prettiest girl" by any term referring to the same individual

(e.g., Sally Sunshine), while in the opaque reading we cannot (i.e., John wants to marry the prettiest girl, whoever she might be). The opaque reading of the sentence is shown in Fig. 27. The complexity results mainly from the description “the prettiest girl”, represented as “the most pretty member of the set of all girls”. A ternary second-order relation “most” has been contrived for

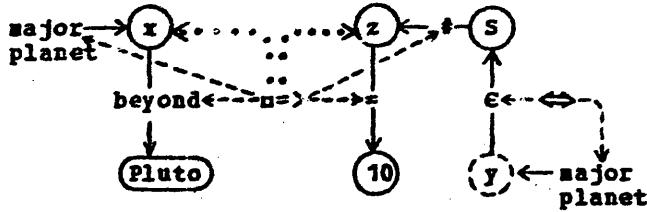


FIG. 26. “If there were a major planet beyond Pluto, the number of major planets would equal 10”; $(\exists S)(\forall y)\{[member(y,S) \leftrightarrow major\text{-planet}(y)] \& [(\exists x) major\text{-planet}(x) \& beyond(x,Pluto)] \square \Rightarrow (\exists z)[\#(S,z) \& z = 10]\}$.

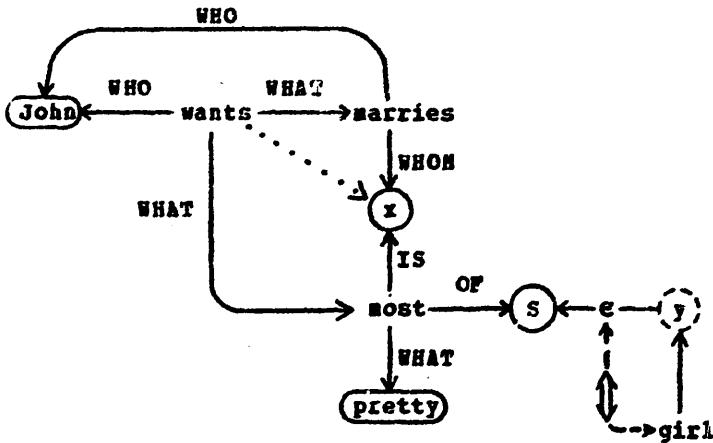


FIG. 27. “John wants to marry the prettiest girl”; $(\exists S)[(\forall x)[member(x,S) \leftrightarrow girl(x)] \& wants(John,(\exists y)[most(pretty,S,y) \& marries(John,y)!])]$.

this purpose which holds for x, P, S if x is the unique member of set S which is superlative with respect to property P . An alternative representation which avoids the use of “most” at the expense of greater complexity can be found in Schubert [46]. The propositional operands of “wants” are regarded as implicitly conjoined, i.e., $wants(w,y,z, \dots)$ is taken to mean x wants y & z & \dots . The dotted line from “wants” is missing in the transparent reading. Quite correctly the diagram then reads “There exists an x such that x is the prettiest girl and John wants to marry x ”.

Propositional attitudes may involve quantification over propositions, as in “John knows everything”, or in “John knows everything Mary knows”. The *Artificial Intelligence* 7 (1976), 163–198

most direct way of representing such propositions is by quantification over propositional variables, as in Fig. 28. In Fig. 28(a) "John knows everything"

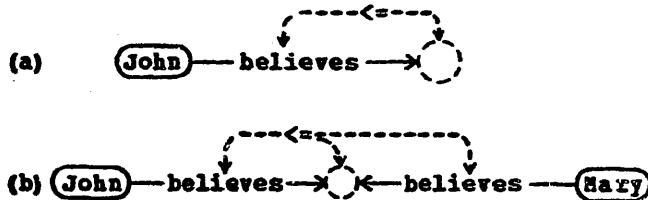


FIG. 28. (a) "John knows everything". (b) "John knows everything Mary knows".

has been paraphrased as "John believes all true propositions". If we accept these representations, we must carefully distinguish between the universally quantified "proposition nodes" in Fig. 28 and the proposition nodes previously introduced as points of attachment for the parts of an explicit proposition. These previous proposition nodes are not quantifiable. The propositions in which they participate "exist" simply by virtue of appearing in the semantic net. The universally quantified nodes in Fig. 28, by contrast, are *concept* nodes denoting *complete* propositions.

The constructions of Figs. 27 and 28 are not intended to sanction unrestricted use of relations with propositional arguments and quantification over propositions. Such lack of restraint could render the network logic inconsistent. Rather, we should restrict ourselves to some carefully chosen set of propositional operators, including the truth-functional operators and some set of modal operators. Also we could avoid the use of propositional variables altogether by paraphrasing "for all propositions" as "for all predicates P and all x such that $P(x)$ ", giving the new version of Fig. 28(a) shown in Fig. 29. This would avoid the use of propositions as concepts. On

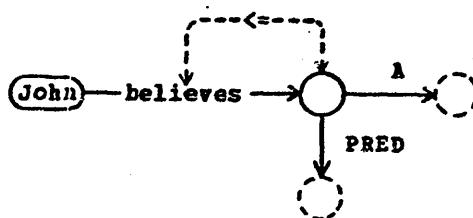


FIG. 29. "John knows everything".

the other hand Fig. 28 has the advantage of being more concise. Note that the replacement of quantification over propositions by quantification over predicates and variables can also be carried through for existential quantification. For example, the sentence "There is a proposition x such that if x is true then mankind is doomed" can be paraphrased as "There is a predicate P such that if $P(x)$ is true for all x then mankind is doomed".

The techniques for representing knowing and believing apply equally to other propositional attitudes such as remembering, supposing, intending, deciding, avoiding, hoping, imagining, pretending and trying. Nonreferential terms within the scopes of such operators (whichever ones are deemed useful independently of the others) can be identified by means of scope dependency links as in Figs. 25–27. The same applies to the deontic modalities such as obligation. It should be obvious, for example, how “John ought to marry the prettiest girl” would be represented.

As a final important modality causal dependency should be mentioned. Causal explanations are rather closely related to counterfactual conditionals, as can be seen from the inference “B would not have happened if A had not happened”, which is often reasonable given that “A caused B”. An example of an opaque context generated by a causal construction is found in the sentence “John asked Mary to dance because she was the only girl left without a partner”. Substitution of the term “Mary” for its referential synonym “the only girl left without a partner” clearly fails. In causal constructions as in other modal constructions, therefore, we may need scope dependency links.

Modal operators have played a prominent role in the work of several network theorists, including Winston [44], Schank [33], Anderson and Bower [2] and others. Winston’s MUST operator, which modifies many relationships in learned structural concepts, can probably be identified with the usual necessity operator. Schank’s causal relation, instrumental relation, and CONC, MTRANS, and MBUILD primitives can all be regarded as modal operators, the latter three being epistemic operators. Anderson and Bower make frequent use of a causal operator, as do most network theorists. What has been lacking in the work of these authors is a recognition of the important distinction between opaque and transparent modal contexts.

9. 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, quantification, and lambda abstraction in these networks.

I hasten to add that I am not urging universal adoption of this notation. It is intended merely as a standard of comparison. What makes it suitable as such a standard is, I think, its clear correspondence to Predicate Calculus notation, which is itself well understood. Certainly a rather different representation might be adopted to meet specific needs in specific applications of semantic nets. For example, a representation which more closely reflects particular computer data structures might be used. Furthermore these computer data structures might themselves take different forms for different

Artificial Intelligence 7 (1976), 163–193

fields of knowledge, rather than conforming with some fixed propositional syntax (see Lindsay [18]). Also additional operators might be formulated to simplify common constructions. For example, to represent simple generalizations of the form "All P are Q" (such as "All crows are black") it would be useful to have an operator, say " \Rightarrow ", with the following definition. If P and Q are n -ary predicates then

$$\textcircled{P} \text{--}\Rightarrow\text{--}\textcircled{Q} \quad \text{iff}$$

$$(\forall x_1)(\forall x_2) \dots (\forall x_n)[P(x_1, \dots, x_n) \Rightarrow Q(x_1, \dots, x_n)].$$

Thus

$$\textcircled{\text{crow}} \text{--}\Rightarrow\text{--}\textcircled{\text{black}}$$

An even more useful generalization operator would be one which expresses "All P are *necessarily* Q", which could be diagrammed as $\square\Rightarrow$ and defined as

$$\textcircled{P} \text{--}\square\Rightarrow\text{--}\textcircled{Q} \quad \text{iff}$$

$$\square(\forall x_1)(\forall x_2) \dots (\forall x_n)[P(x_1, \dots, x_n) \Rightarrow Q(x_1, \dots, x_n)].$$

This would greatly simplify the encoding of *essential* meaning relationships among concepts, such as

$$\textcircled{\text{CROW}} \text{--}\square\Rightarrow\text{--}\textcircled{\text{bird}}$$

(i.e., crows are necessarily birds),

$$\textcircled{\text{drinks}} \text{--}\square\Rightarrow\text{--}\textcircled{\text{ingests}}$$

(i.e., if x drinks y then x necessarily ingests y), etc. The operators would induce a hierarchy on concepts which could be exploited for efficient inferencing about subset-superset relationships.⁸ The point is that the definitions of these operators can be expressed in terms of the basic network notation.

The reader familiar with Predicate Calculus may have noticed the lack of a functional notation for networks analogous to that of Predicate Calculus. Such a notation is easily devised (see Cercone [45]). The notation slightly simplifies the representation of some propositions and complicates that of others. In particular, existential quantification of the *value* of a function $f(x)$ requires

⁸ The operator $\square\Rightarrow$ differs from Quillian's "subset" relation in two respects. First, "property P " is not regarded as synonymous with "the set of P 's" for the reason mentioned in Section 2. Second, Quillian presumably had in mind the usual (contingent) subset relation, rather than one reinforced by necessity.

the introduction of a proposition of the form $(\exists y)f(x) = y$, whereas no such proposition is needed in the relational notation, which already has a "slot" for y .

A variety of issues in the representation of informal knowledge could raise additional notational problems. Examples are the handling of vagueness (already mentioned parenthetically in the comments on natural language quantifiers), events, the lexical meanings of complex concepts, and overall knowledge organization. Beyond these relatively static issues lie the more dynamic issues of actual language interpretation and generation, plausible inference, learning, and the interplay between procedural and factual knowledge. Clearly any questions about representation raised by these problem areas can only be answered in the context of particular approaches toward the solutions of the problems themselves. However, the present representation is a fairly direct extension of several quite successful, superficially disparate representations, such as Schank's conceptualizations, Winston's descriptions, or 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 notation provides should be of real value in the design of understanding systems.

ACKNOWLEDGMENTS

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, as well as for the comments of the referees, all of 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

1. Abelson, R. P. The structure of belief systems. *Computer Models of Thought and Language*, R. Schank and K. Colby, eds., Freeman, San Francisco, Calif. (1973), 465-471.
 2. Anderson, J. and Bower, G. *Human Associative Memory*. Winston, Washington, D.C. (1973).
 3. Bartsch, R. and Vennemann, T. *Semantic Structures*. Athenaeum Verlag, Frankfurt (1972).
 4. Bruce, B. C. A model for temporal references and its application in a question answering program. *Artificial Intelligence* 3 (1972), 1-26.
 5. Carnap, R. *Meaning and Necessity*. Univ. of Chicago Press, Chicago, Ill. (1947).
 6. Cercone, N. and Schubert, L. K. A sketch of state-based conceptual representation. Department of Computing Science, TR74-19, University of Alberta, Edmonton, Alberta (1974).
 7. Donnellan, K. Reference and definite descriptions. *Philosophical Review* 75 (1966), 281-304. Reprinted in *Semantics*, D. D. Steinberg and L. A. Jakobovits, eds., Cambridge Univ. Press, Cambridge (1971), 100-114.
- Artificial Intelligence* 7 (1976), 163-198

8. Fillmore, C. J. The case for case. *Universals in Linguistic Theory*, E. Bach and R. T. Harms, eds., Holt, Rinehart and Winston, New York (1968), 1-88.
9. Findler, N. and Chen, D. On the problems of time, retrieval of temporal relations, causality and coexistence. *Advance Papers of Second International Joint Conference on Artificial Intelligence*, The British Computer Society, London (1971), 531-545.
10. Firschein, O. and Fischler, M. A. A study in descriptive representation of pictorial data. *Advance Papers of Second International Joint Conference on Artificial Intelligence*, The British Computer Society, London (1971), 258-269.
11. Fisk, M. A modal analogue of free logic. *The Logical Way of Doing Things*, K. Lambert, ed., Yale Univ. Press, New Haven, Conn. (1969), 147-184.
12. Guzman, A. Analysis of curved line drawings using context and global information. *Machine Intelligence 6*, B. Meltzer and D. Michie, eds., American Elsevier, New York (1971), 325-375.
13. Hendrix, G. G., Thompson, C. W. and Slocum, J. Language processing via canonical verbs and semantic models. *Advance Papers of Third International Joint Conference on Artificial Intelligence*, Stanford Research Institute, Menlo Park, Calif. (1973), 262-269.
14. Hughes, G. E. and Cresswell, M. J. *An Introduction to Modal Logic*. Methuen, London (1968).
15. Isard, S. and Longuet-Higgins, H. C. Question-answering in English. *Machine Intelligence 6*, B. Meltzer and D. Michie, eds., American Elsevier, New York (1971), 243-254.
16. Kalish, D. and Montague, R. *Logic: Techniques of Formal Reasoning*. Harcourt, Brace, and World, New York (1964).
17. Lewis, D. *Counterfactuals*. Harvard Univ. Press, Cambridge, Mass. (1973).
18. Lindsay, R. K. In defense of ad hoc systems. *Computer Models of Thought and Language*, E. C. Schank and K. M. Colby, eds., Freeman, San Francisco, Calif. (1973), 372-395.
19. McCarthy, J. and Hayes, P. Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence 4*, B. Meltzer and D. Michie, eds., American Elsevier, New York (1969), 463-502.
20. Moore, R. D-SCRIPT: A computational theory of descriptions. Memo No. 278, Artificial Intelligence Laboratory, MIT, Cambridge, Mass.; see also *Advance Papers of Third International Joint Conference on Artificial Intelligence*, Stanford Research Institute, Menlo Park, Calif. (1973), 223-229.
21. Mylopoulos, J., Badler, N., Nelli, L. and Roussopoulos, N. I.PAK: A SNOBOL-based programming language for artificial intelligence applications. *Advance Papers of Third International Joint Conference on Artificial Intelligence*, Stanford Research Institute, Menlo Park, Calif. (1973), 691-696.
22. Palme, J. Making computers understand natural language. *Artificial Intelligence and Heuristic Programming*, N. V. Findler and B. Meltzer, eds., American Elsevier, New York (1971), 199-244.
23. Quillian, M. Semantic memory. *Semantic Information Processing*, M. Minsky, ed., MIT Press, Cambridge, Mass. (1968), 227-270.
24. Quillian, M. The teachable language comprehender. *Comm. ACM* 12 (1969), 459-475.
25. Quine, W. V. A. *Word and Object*. MIT Press, Cambridge, Mass. (1960).
26. Reichenbach, H. *Elements of Symbolic Logic*. Free Press, New York (1966).
27. Rumelhart, D., Lindsay, P. and Norman, D. A process model for long term memory. *Organization of Memory*, E. Tulving and W. Donaldson, eds., Academic Press, New York (1972), 198-221.
28. Russell, B. A. W. On denoting. *Mind* 14 (1905), 479-493. Reprinted in *Readings in Philosophical Analysis*, H. Feigl and W. Sellars, eds., Appleton-Century-Crofts, New York (1949), 103-115.

Artificial Intelligence 7 (1976), 163-198

29. Russell, B. A. W. Descriptions. Chapter XVI of *Introduction to Mathematical Philosophy* B. A. W. Russell, 2nd ed., Allen and Unwin, London (1920). Reprinted in *Semantics and the Philosophy of Language*, L. Linsky, ed., Univ. of Illinois Press, Urbana, Ill. (1952), 95-108.
30. Sandewall, E. J. A set-oriented property-structure representation for binary relations. *Machine Intelligence 5*, B. Meltzer and D. Michie, eds., American Elsevier, New York (1970), 237-252.
31. Sandewall, E. J. Representing natural language information in predicate calculus. *Machine Intelligence 6*, B. Meltzer and D. Michie, eds., American Elsevier, New York (1971), 255-277.
32. Schank, R. Conceptual dependency: A theory of natural language understanding. *Cogn. Psych.* 3 (1972), 552-631.
33. Schank, R. The fourteen primitive actions and their inferences. Stanford AI Project, Memo AIM-182, Stanford University, Stanford, Calif. (1973).
34. Schank, R., Goldman, N., Rieger, C. and Riesbeck, C. Margie: memory, analysis, response generation, and inference on English. Advance Papers of Third International Joint Conference on Artificial Intelligence, Stanford Research Institute, Menlo Park, Calif. (1973), 255-261.
35. Simmons, R. F. Some semantic structures for representing English meanings. *Language Comprehension and the Acquisition of Knowledge*, J. B. Carroll and R. O. Freedle, eds., Winston, Washington, D.C. (1972).
36. Simmons, R. Semantic networks: Their computation and use for understanding English. *Computer Models of Thought and Language*, R. Schank and K. Colby, eds., Freeman, San Francisco, Calif. (1973), 63-113.
37. Simmons, R. F. and Bruce, B. C. 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 (1971), 524-530.
38. Snyder, D. P. *Modal Logic and its Applications*. Van Nostrand Reinold, New York (1971).
39. Strawson, P. F. On referring. *Mind* 59 (1950), 320-344.
40. Van Fraassen, B. C. Singular terms, truth-value gaps, and free logic. *J. Phil.* 63 (1966), 481-495.
41. Whitehead, A. N. and Russell, B. A. W. *Principia Mathematica*, Vol. 1. Cambridge Univ. Press, Cambridge (1910).
42. Wilson, K. V. Memory organization and question answering. *The Psychology of Knowing*, J. R. Royce and W. W. Rozeboom, eds., Academic Press, New York (1972)
43. Wilson, K. V. Formal representation of cognitive processes. Center for the Advanced Study of Theoretical Psychology, University of Alberta, Edmonton, Alberta (1975).
44. Winston, P. Learning structural descriptions from examples. PhD Thesis, MIT, MAC TR-76, Cambridge, Mass. (1970).
45. Cercone, N. Representing natural language in extended semantic networks. Department of Computing Science, TR75-11, University of Alberta, Edmonton, Alberta (1975).
46. Schubert, L. K. Extending the expressive power of semantic nets. Department of Computing Science, TR74-18, University of Alberta, Edmonton, Alberta (1974).

Received April 1975; revised version received August 1975