INFERENCE IN A TOPICALLY ORGANIZED SEMANTIC NET

Johannes de Haan and Lenhart K. Schubert

Department of Computing Science, University of Alberta
Edmonton, Alberta, Canada T6G 2H1

ABSTRACT

A semantic net system in which knowledge is topically organized around concepts has been under development at the University of Alberta for some time. The system is capable of automatic topical classification of modal logic input sentences, concept and topic oriented retrieval, and property inheritance of a general sort. This paper presents an inference method which efficiently determines yes or no answers to relatively simple questions about knowledge in the net. It is a deductive, resolution based method, enhanced by a set of special inference methods, and relies on the classification and retrieval mechanisms of the net to maintain its effectiveness, unencumbered by the volume or diversity of knowledge in the net.

I INTRODUCTION

In 1975, Scott Fahlman and Drew McDermott discussed the so-called "symbol-mapping problem" (Fahlman 1975, McDermott 1975). In essence, this is the problem of making simple inferences quickly in a system with a potentially very large, varied knowledge base. Among the examples they discussed were the inference that Clyde is grey, given that Clyde is an elephant and elephants are grey, and the inference that Clyde does not live in a teacup or play the piano, given standard knowledge about elephants, teacups, and pianos. What makes the problem hard is not the complexity of the requisite knowledge or reasoning, which are quite modest, but the fact that the right knowledge may be very hard to find: we have a "needle-in-a-haystack" problem.

More than a decade later, the problem cannot be considered satisfactorily solved. Progress has been made in "customized" understanding and reasoning systems — systems that can make a wide range of inferences in a circumscribed domain (e.g., the divorce story domain of BORIS, described in Lehnert et al. (1983), or the various domains of expertise of expert systems in medicine, computer configuration, prospecting, and so on). However, the problem of scaling up to systems with a large amount of knowledge about a wide variety of subjects is still very much with us.

The work reported here is part of a continuing effort to develop a question-answering and English conversational system (ECOSYSTEM) which is unencumbered by the volume or diversity of its knowledge. Our approach to efficient question-answering for large knowledge bases was first sketched in Schubert et al. (1979). This sketch motivated the design of a 3-level semantic net organization: at the highest level, knowledge resides in a main net (for real world knowledge) and in arbitrarily nested subnets (primarily for "mental worlds" and "narrative worlds"); the next level is the level of concepts, which are used in each subnet as access points for knowledge directly involving them; and the third level is the level of topics, which hierarchically subdivide the knowledge about a given concept into topically related subsets of facts (Covington & Schubert, 1980). This organization allows highly **selective** retrieval of knowledge relevant to a query. For example, the question "Is the wolf in the story of Little Red Riding Hood grey?" (posed in logical form) would prompt access of the Little Red Riding Hood subnet, followed by access within that subnet of the node for the wolf, followed by access of 'colouring' information about the wolf and its superordinate concepts. (Additional information may be accessed during the inference attempt – see Section III.) The implementation provides for input of modal logic sentences which are automatically converted to clause form, topically classified, and inserted at appropriate concept nodes. In addition, special inference methods have been developed to short-cut taxonomic reasoning and reasoning about colours and time. The new deductive algorithm builds on this work; it efficiently determines yes or no answers to the sorts of questions discussed by Fahlman and McDermott, relying upon the classification and retrieval mechanisms of the knowledge base.

In some respects, the method has goals that are similar to those of automatic theorem provers. However, the domains of natural language understanding and theorem-proving are different, and two fundamental differences distinguish this method from a theorem-prover:

1. Size of the knowledge base. Theorem provers work on problems in well defined logical or mathematical domains. These systems are artificial and are usually axiomatized by some small set of statements. This is not true of the semantic net, which will incorporate a very large body of knowledge, sufficient at least to carry on an intelligent conversation.
2. Deductive ability. Theorem provers are judged mainly by their deductive ability - better provers solve logically more complex problems. People require minutes or even hours to solve these problems, and yet, they are able to perform the inference needed for natural language understanding almost immediately. It is not unreasonable to suppose that natural language inference is shallower (i.e., requires fewer steps) than that required for mathematical theorem proving.

II THE SEMANTIC NET

To understand how the inference method works, it is important to understand how the semantic net represents and organizes knowledge. The representational scheme is essentially that as described in Schubert (1976), incorporating changes described in Schubert et al. (1979) and Covington (1980). The syntax of the net provides for the representation of formulae in higher-order modal logic,* with constants, functions, existentially and universally quantified variables, and the usual truth function connectives (negation, implication, disjunction and conjunction).

* Although the net is able to represent and organize modal propositions, the current inference method is restricted to the first order predicate calculus.

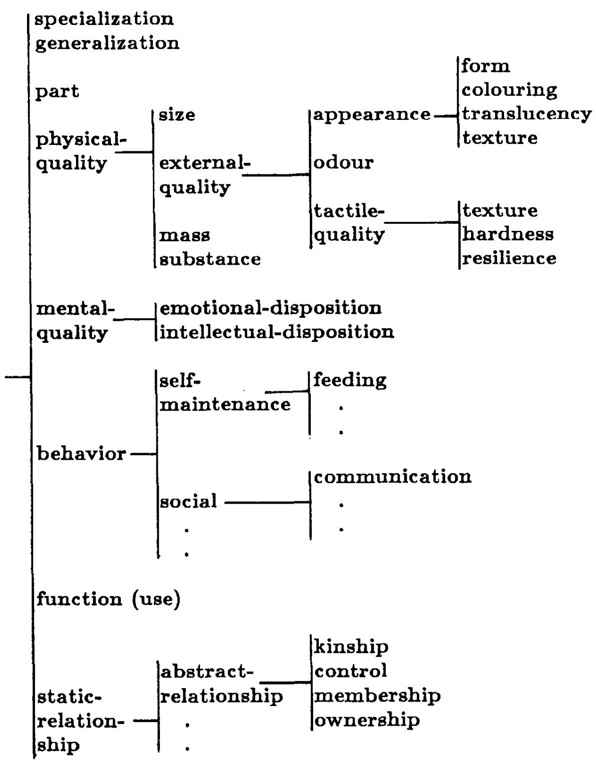

Figure 1. Topic hierarchy (TH)

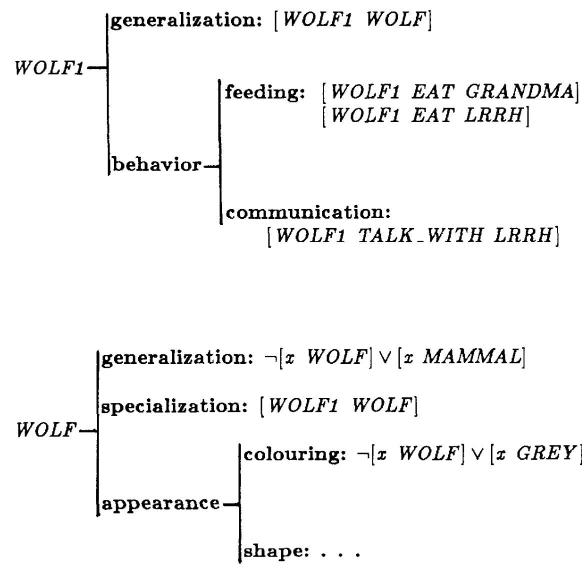

Figure 2. Topic access skeletons (TAS's)

Even though the net has the expressive ability to store propositions in an unrestricted form, the need for automatic classification, storage economy and recognition of syntactic variants of previously inserted information has led to the adoption of conjunctive normal form for the internal storage of propositions (see Covington & Schubert 1980 for a detailed discussion). Thus, input formulae are transformed into sets of clauses. Universal quantification is implicit, and existentially quantified variables become skolem functions of zero or more arguments.

Within the main net and each subnet, a dictionary provides direct (hashed) access to named concepts. Facts about these concepts are then organized using a topical hierarchy (TH). Using the TH, it becomes possible to directly access clauses which topically pertain to a concept and to ignore all the rest (which can potentially be a large number of clauses). The structure of the TH also defines relationships among topics, so it is possible to broaden an access to sub-topics or super-topics of a given topic. Figure 1 illustrates a simplified topical hierarchy for concepts which are physical objects.

It is not likely that we know something about all topics for all concepts, and to duplicate the entire TH for every concept would waste storage space and increase traversal time across empty topics. To solve this problem, topic access skeletons (TAS) are used. Each TAS is a minimal hierarchy based on the complete TH, and only includes topics about which there is some knowledge, or topics which are needed to preserve the structure of the hierarchy.

Figure 2 illustrates simple TAS's for the specific concept *WOLF1* and for the generic concept *WOLF*. Using predefined topical indicators for predicates, a classification algorithm automatically assigns to each asserted clause particular (*concept*, *topic*) pairs. For example, the predicate *EAT* indicates the topic 'feeding' with respect to its first argument and the topic 'consumption' with respect to its second argument. Hence, the clause [*WOLF1 EAT GRANDMA*]* is assigned the pairs (*WOLF1*, *feeding*) and (*GRANDMA*, *consumption*), and is indexed accordingly in the TAS's of the concepts *WOLF1* and *GRANDMA*. Similarly, the predicate *GREY* indicates the topic 'colouring', so the clause $\neg[x\ WOLF] \vee [x\ GREY]$ is assigned the pair (*WOLF*, *colouring*), and the clause is indexed under the *colouring* topic in the TAS for the concept *WOLF*. Subsequent queries about the colouring of wolves would be able to directly access this clause. Queries about the appearance of wolves would also be able to quickly get to the clause, because *colouring* is a sub-topic of *appearance*.

The special topic 'major implication' is used to classify fundamental properties of predicates that characterize their meaning.** For example, a major implication of the predicate *IN* might be that if A is in B, then A is smaller that B, and that B is a container or enclosure of some sort. Similarly, an 'exclusion' topic is used to classify clauses which explicitly define such a relationship (e.g., $\neg[x\ CREATURE] \vee \neg[x\ PLANT]$).

For the main net and all of its subnets, there is also a hierarchical organization of the concepts within each net. Concepts are organized using a structure called a concept hierarchy (CH), which is essentially a type hierarchy for physical objects. It is used in two ways: (1) for quick associative access to groups of concepts with the same type (just as the TH provides quick access to clauses about the same topic); and (2) to guide a property inheritance mechanism in its search for generalizations or specializations of a given concept. Figure 3 presents a simplified CH for physical objects.

To repeat the CH for every subnet is also waste of resources, so a concept access skeleton (CAS) is maintained for each sub-

* To improve readability, the syntax places the first argument of a literal before the predicate.

** Major implications were linguistically motivated in Schubert et al. (1979). They are related to Schank's ACT-based inferences, as well as to "terminological facts" in systems like KRYPTON and KL-ONE.

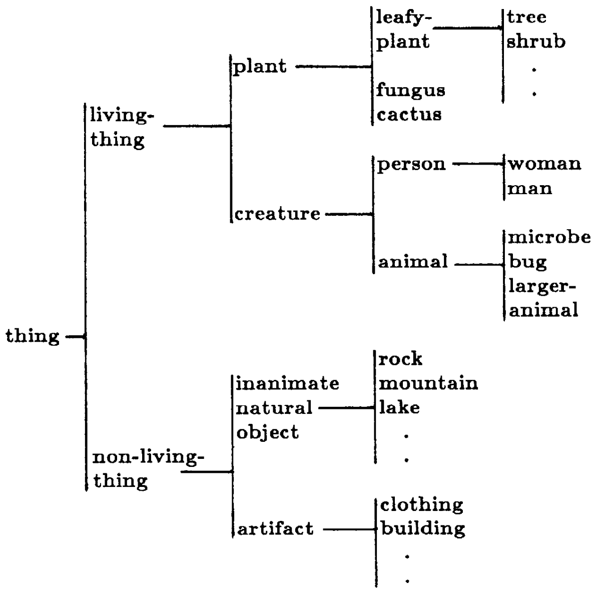

Figure 3. Concept hierarchy (CH)

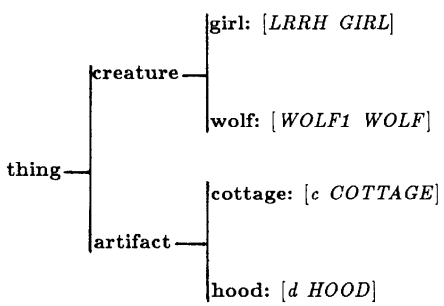

Figure 4. Concept access skeleton (CAS)

net. Just as a TAS is a minimal hierarchy based on the TH, a CAS is a minimal hierarchy based on the CH. Figure 4 presents a CAS which associatively organizes some of the constant concepts in the story of Little Red Riding Hood* (LRRH). The classification algorithm automatically places clauses which are assigned a (*concept, generalization*) pair into a CAS whenever *concept* is an instance.

III THE INFERENCE METHOD

As mentioned above, we were led to adopt clause form as the most convenient logical form for the purposes of automatic topical classification. It was therefore natural to choose resolution as our basic inference method. Our deductive algorithm employs certain familiar strategies, such as set-of-support and preference for simple resolvents. What makes the algorithm unique, however, are the following three features. (1) The set of potential inference steps considered at any one time is severely limited by use of both the concept hierarchy and the topic hierarchy (as reflected in each concept's TAS); (2) there is an ongoing decision-making process which trades off inference steps against retrieval steps; and (3) resolution is "generalized", permitting use of special inference methods for specific domains.

* Most examples in this paper are loosely based on this story.

We will elaborate on the last point first, and then discuss (1) and (2) under the heading "Resolution control".

A. Generalized resolution and evaluation

Resolving two clauses is usually done by resolving on literals from each clause which have the same predicate but opposing signs. For example, $[LRRH\ GIRL]$ resolves against $\neg[LRRH\ GIRL]$. Given the existence of a special inference method which quickly infers relationships among 'type' predicates, it is also possible to reduce long inference chains to a single resolution step. For example, $[LRRH\ GIRL]$ directly resolves against $\neg[LRRH\ CREATURE]$, without using the intermediary clauses $\neg[x\ GIRL] \lor [x\ CHILD]$ ... $\neg[x\ PERSON] \lor [x\ CREATURE]$. Or, given a special inference method for colour, it becomes possible to directly resolve $[WOLF1\ BROWN]$ against $[WOLF1\ GREY]$ (Papalaskaris & Schubert 1982, Schubert et al. 1983, Brachman et al. 1983, Stickel 1983, Vilain 1985). This method of "generalized" resolving can similarly be used for factoring and subsumption (Stickel 1985, Schubert et al. 1986). A crucial advantage of our topical retrieval mechanism is that it allows candidates for generalized resolving to be found as efficiently as candidates for ordinary resolving, on the basis of their classification under the same topic (e.g., $[WOLF1\ GREY]$ and $[WOLF1\ BROWN]$ are both classified as colouring propositions for *WOLF1*).

Evaluation is another means by which the resolution process can be considerably shortened. Every clause from the original question or generated during the proof goes through an evaluative attempt, to try to achieve an immediate proof (if the clause is false), or to remove the clause from consideration (if it is true). If the clause cannot be evaluated, each of its literals is tried, to try to prove the whole clause true (if the literal is true), or to remove it from the clause (if it is false). The simplest evaluative method is to match a clause or literal against previously asserted clauses (the normal form ensures that the input form of a formulae has no affect on the matching process).

Generalized resolution and evaluation can be used for any class of predicates for which there exists a special inference method. Currently, the inference algorithm uses special inference methods for types and colours; however, methods for time and part-of relationships have been developed as well (Schubert 1979, Papalaskaris & Schubert 1982, Schubert et al. 1983, Schubert et al. 1986, Taugher 1983).

B. Resolution control

A great many resolution control strategies have appeared in the literature, but none of them has been completely successful in containing the usual combinatorial explosion of generated clauses and the ensuing difficulty in finding the 'right' ones to resolve with. Nevertheless, resolution proved to be well-suited to our purposes, for two reasons: (1) Proofs required for natural language understanding and ordinary question-answering are generally short (at least when special 'shortcut' methods are available); and (2) The "needle-in-a-haystack" problem is solved in our system by the access organization we have described.

The examination of most resolution proofs that have gone astray soon reveals a large number of resolutions where the unification process made some substitution that did not seem to be semantically valid. For example, it is syntactically possible to resolve $\neg[x\ WOLF] \lor [x\ GREY]$ against $\neg[LRRH\ GREY]$. However, we intuitively realize that the first clause (knowledge about wolves) can not really be applied to LRRH, and that this resolution is fruitless. To express it another way: the universal variable x is typed to represent *WOLF*, and should not by substituted by a concept which is of type *GIRL*.

Briefly, our algorithm avoids fruitless inferences by restricting its search for potential resolving candidates against a given clauses to clauses connected to it by a path (of length 0 or more) in the concept hierarchy, and classified under the same topic. The confinement of resolution to paths in the concept hierarchy is comparable to approaches based on sortal logics (e.g., McSkimin & Minker 1979, Walther 1983). However, our method does not require explicit typing of predicates and 'sort' checks during unification. The topical confinement of resolution readily picks out clause pairs containing resolvable literals, either in the ordinary sense or in the generalized sense.

The inference algorithm maintains an agenda of potential actions. Each action is relevant to a single clause, and is either a possible resolution that can be performed with that clause, or a retrieval action which might lead to possible resolution actions. Retrieval actions are based directly on the classification of a clause, and are specific to the same kind of $(concept, topic)$ pairs that the classification procedure derives for asserted clauses. Six kinds of retrieval actions can appear on the agenda:

(1) $clause \leftrightarrow (conc, topc, super)$ is the notation for the action which would retrieve all clauses stored at concept *conc* and its superconcepts, under topic *topc*, and form all potential resolutions between *clause* and the retrieved clauses, placing them on the agenda.

(2) $clause \leftrightarrow (conc, topc, sub)$ is similar to (1), but uses the subconcepts of *conc* only, excluding instances.

(3) $clause \leftrightarrow (conc, topc, inst)$ is also similar to (1), but uses instances of *conc* only.

(4) $clause \leftrightarrow (conc, major\_imp)$ denotes the action which would retrieve all major-implications of concept *conc*, and form all potential resolutions between *clause* and the retrieved clauses.

(5) $clause \leftrightarrow (conc, excl)$ is similar to (4), but uses exclusion propositions of concept *conc*.

(6) $clause \leftrightarrow (conc, inst)$ denotes the action which would retrieve all clauses specifying instances of *conc* and form potential resolutions between *clause* and these clauses.

The CH is used to quickly determine the super or subconcepts of a given concept, and the CAS is used to quickly find instances of a given concept.

The agenda is ordered by the estimated cost of the actions, and the inference method always chooses to do the action with the least cost first. The cost of a possible resolution is high to the extent that the resolvent is expected to be complex (effectively implementing a 'least complex resolvent' preference strategy). The cost of a possible retrieval, $clause \leftrightarrow (conc, \ldots)$, is high to the extent that *clause* is complex, and the expected number of clauses to be retrieved is high.

Each clause c which is to be considered for possible refutation is "loaded" into the network, as follows:

1. Simplify (evaluate) c, if possible. If c is *true*, discard it; if c is *false*, report a "disproof".
2. Classify c for insertion and, if not yet present, insert it (i.e., as if it were an asserted clause).
3. If c was classified twice w.r.t the same $(conc, topc)$ pair, try to factor it; if successful, load the factor(s).
4. Generate the following possible retrievals relevant to c, placing them on the agenda:
   (a) if c was classified under $(conc, topc)$ generate the retrieval $c \leftrightarrow (conc, topc, super)$, and, if *conc* is not a constant, the additional retrievals $c \leftrightarrow (conc, topc, sub)$ and $c \leftrightarrow (conc, topc, inst)$.
   (b) if c contains a positive predicate P then generate the retrievals $c \leftrightarrow (P, major\_imp)$ and $c \leftrightarrow (P, excl)$.
   (c) if c contains a type predicate P with variable argument and c was not classified under any $(conc, topc)$ pair, generate $c \leftrightarrow (P, inst)$.

The complete inference algorithm can now be described:

1. Load the clauses to be refuted. This might yield an immediate disproof, but more likely it will put a set of potential retrievals on the agenda.
2. Carry out the potential action with the least cost. If it was a retrieval, this results in a set of potential resolutions being placed on the agenda. If the action was a resolution with resolvent c, then load c. If c evaluates to *false* when it is loaded, or if c is the null clause, then report a "disproof".
3. If some predefined resource limit has been exceeded, then return "unknown", else repeat from Step 2.

The method also concurrently searches for a proof, using the dual of the original question. Note that a set-of-support strategy is used, as only clauses from the original set to be refuted, or one of their descendants, is ever considered for a resolution action.

IV EXAMPLES

1. To answer the question "Is there a creature (in the story of LRRH)?", the clause to be refuted for a "yes" answer is $\neg[x\ CREATURE]$ (called 'c' for brevity). Loading this clause generates the retrieval $c \leftrightarrow (CREATURE, inst)$. Using the CAS, the clauses $[LRRH\ GIRL], [WOLF1\ WOLF], \ldots$ are retrieved, any of which gives an immediate null resolvent by generalized resolution.

2. To answer the question "Is the Wolf grey?", the clause to be refuted for a "yes" answer is $\neg[WOLF1\ GREY]$ $(=c)$. Loading this clause generates the retrieval $c \leftrightarrow (WOLF1, colouring, super)$. Using the CH to get from *WOLF1* to the generic *WOLF* concept, the clause $\neg[x\ WOLF] \vee [x\ GREY]$ is retrieved. Resolving against the orginal clause yields $\neg[WOLF1\ WOLF]$, which immediately evaluates to false when it is loaded.

3. To answer the question "Are all creatures pink?" the clause to be refuted for a "no" answer is $\neg[x\ CREATURE] \vee [x\ PINK]$ $(=c)$. Loading this clause generates the retrievals $c \leftrightarrow (CREATURE, colouring, super)$ and $c \leftrightarrow (CREATURE, colouring, sub)$. The *sub* retrieval yields $\neg[x\ WOLF] \vee [x\ GREY]$ $(=c')$. Using generalized resolution (for colours) yields $\neg[x\ CREATURE] \vee \neg[x\ WOLF]$. Generalized factoring on this clause yields $\neg[x\ WOLF]$ $(=c'')$. A retrieval for this clause is $c'' \leftrightarrow (WOLF, inst)$. The CAS is used to find the clause $[WOLF1\ WOLF]$, which resolves against c'' to complete the disproof.

4. To answer the question "Does the Wolf live in LRRH's basket?" (cf. Fahlman's Clyde-in-the-teacup problem), the clause to be refuted is $[WOLF1\ LIVE\_IN\ BASKET1]$ $(=c)$. One of the retrievals generated for this clause is $c \leftrightarrow (LIVE\_IN, major\_imp)$. A major implication of *LIVE_IN* indicates that the *WOLF1* would have to be smaller than *BASKET1* to live in it.* Further inference is then needed to

* Alternatively, a major implication of *LIVE_IN* could merely indicate that if x lives in y, then x is in y (at some time). The knowledge that x is smaller than y would then be retrieved from a major implication for *IN*.

establish that the *WOLF1* is not smaller than the basket, and therefore cannot live in it. The best way of establishing the relative sizes of *WOLF1* and *BASKET1* would be via a special inference method for relationships among physical objects, but the current implementation does not include such a method and instead resorts to explicitly asserting these relationships in the knowledge base.

The above questions were chosen to illustrate technical points. More natural questions are also easily handled, such as "Did an animal eat someone?" and "Is there anything in LRRH's basket that she likes to eat?".

V DISCUSSION AND FUTURE WORK

An implementation of the inference method, written in Berkeley PASCAL, was able to answer a test set of 40 questions in about 15 seconds CPU time on a VAX 11/780, using a knowledge base of over 200 clauses (general and specific knowledge about the story of LRRH). Doubling the size of the knowledge base had no effect on the question-answering time.

The successful implementation of the method vindicates the net organization developed earlier, showing that it provides quick selective access to the knowledge needed for simple question-answering. Furthermore, the organizational structure proved useful in guiding and constraining deduction steps. Proofs are confined to vertical paths through the concept hierarchy, and are topically focused, and as a result are very direct, avoiding "meaningless" deductions, regardless of the amount of knowledge stored. Thus, we have made significant progress towards solving the "symbol-mapping" problem.

Recent knowledge representation systems somewhat similar in aim to ours include KRYPTON (Brachman et al., 1983), KL-TWO (Vilain, 1985) and HORNE (Allen et al., 1984). Like ECOSYSTEM, these systems are intended to provide a domain-independent logical representation, and general and special inference methods (such as taxonomic methods) applicable to a variety of domains. However, ECOSYSTEM's concept-centred, topically focused retrieval mechanism, and its use in guiding deduction, appear to be unique. Further, rather than providing alternative inference "tools", such as forward and backward chaining, we have tried to provide a single, efficient algorithm for deductive question-answering. Also, our overall philosophy has been to provide a perfectly general representation and inference mechanism, which we then seek to <u>accelerate</u> by special methods, as opposed to providing an initially restrictive representation and inference mechanism, to be subsequently extended by special inference methods.

Numerous extensions to ECOSYTEM are planned. These include extensions to handle temporal information (the temporal system is nearly operational), equality, arithmetic, sets, modalities (including causation), generics (using the approach of Pelletier & Schubert, 1984), and special methods for "naive physics". Work on wh-question-answering and on the natural language front end are also under way (Schubert 1984, Schubert & Watanabe 1986).

REFERENCES

Allen, J. F., Giuliano, M., and Frisch A. M. (1984). *The HORNE Reasoning System*, TR 126, Computer Science Department, University of Rochester, Rochester NY. Brachman, R. J., Fikes, R. E., and Levesque, H. J. (1983). Krypton: a functional approach to knowledge representation, *Computer 16*, 67-73.

Covington, A. R. (1980). *Organization and Representation of Knowledge*, M.Sc. Thesis, University of Alberta.

Covington, A. R., and Schubert, L. K. (1980). Organization of modally embedded propositions and of dependent concepts, *Proc. of the 3rd National Conference of the CSCSI/SCEIO*, Victoria, BC, May, 87–94.

Fahlman, S. (1975). *A System for Representing and Using Real World Knowledge*, AI Lab Memo 331, MIT, Cambridge, Massachusetts.

Lehnert, W. G, Dyer, M. G., Johnson, P. N., Yang, C. J., and Harley, S. (1983). BORIS — an experiment in in-depth understanding of narratives, *Artificial Intelligence 20*, 15–62, MIT press.

McDermott, D. (1975). Symbol-mapping: a technical problem in PLANNER-like systems, *SIGART Newsletter 51*, (April), p.4.

McSkimin, J. R., and Minker, J. (1979). A predicate calculus based semantic network for deductive searching, in *Associative Networks*, N. V. Findler (ed.), Academic Press, 205–238

Papalaskaris, M. A. (1982). *Special Purpose Inference Methods* M.Sc. Thesis, University of Alberta.

Papalaskaris, M. A. and Schubert, L. K. (1982). Inference, incompatible predicates and colours. *Proc. of the 4th National Conference of the CSCSI/SCEIO*, Saskatoon, Sask., 97–102.

Pelletier F. J., and Schubert L. K. (1984). Two theorems for computing the logical form of mass expressions, *Proc. COLING-84*, July, Stanford, California, 108-111.

Schubert, L. K. (1976). Extending the expressive power of semantic networks, *Artificial Intelligence 7*, 163–198.

Schubert, L. K. (1979). Problems with parts. *IJCAI–79*, Tokyo, Japan, August, 778–784.

Schubert, L. K., Goebel, R. G., and Cercone, N. J., (1979). The structure and organization of a semantic net for comprehension and inference, in *Associative Networks*, N. V. Findler (ed.), Academic Press, 179–203.

Schubert, L. K., Papalaskaris, M. A., and Taugher, J. (1983). Determining type, part, color, and time relationships, *Computer (USA) 16*, 10, 53–60.

Schubert L. K. (1984). On parsing preferences, *Proc. COLING-84*, July, Stanford, California, 247-250.

Schubert L. K., and Watanabe, L. (1986). What's in an answer: a theoretical perspective on deductive question-answering, *Proc. of the 6th Canadian Conf. on AI (AI-86)*, May, Montreal, Canada, 71-77.

Schubert, L. K., Papalaskaris, M. A., and Taugher, J. (1986 – to appear). Accelerating deductive inference: special methods for taxonomies, colours and times, in *Knowledge Representation*, Cercone, N., and McCalla, G. (eds.), Springer-Verlag, New York.

Stickel, M. E. (1983). Theory resolution: building in nonequational theories, *Proc. AAAI–83*, Washington, D.C. , August, 391–397.

Stickel, M. E. (1985). Automated deduction by theory resolution, *Proc. IJCAI–85*, Los Angeles, California, August, 1181–1186.

Taugher, J. E. (1983). *An Efficient Representation for Time Information*, M.Sc. Thesis, University of Alberta.

Vilain, M. (1985). The restricted language architecture of a hybrid representation system, *Proc. IJCAI–85*, Los Angeles, California, August, 547–551.

Walther, C. (1983). A many-sorted calculus based on resolution and paramodulation, *Proc. IJCAI-83*, Karlsruhe, West Germany, August, 882-891.