

Summary of Scientific Progress
Supported by N.S.F. Grant No. MCS80-06314
1980-81

SNeRG Technical Note 3

Stuart C. Shapiro
Department of Computer Science
SUNY at Buffalo
Amherst, New York 14226

April, 1981

1 A Belief Revision System Based on Relevance Logic and
Heterarchical Contexts

Belief revision is important in systems which have to make conclusions based on partial evidence and might have to revise such conclusions if an unexpected condition occurs. One of the most primitive forms of belief revision is chronological backtracking, which consists of changing the most recent decision taken. Dependency-directed backtracking was proposed as an alternative by [Stallman and Sussman, 1977]. In this system, each derived proposition has stored with it the set of all propositions in its derivation -- its dependency record. When a contradiction is found, dependency records are traced to find all hypotheses upon which the contradiction relies, and heuristics are used to rule out one of them. Doyle [Doyle, 1979] adds to the dependency record of a proposition the set of other propositions in the data base known to be incompatible with it. This latter set is used to prevent two propositions known to be incompatible from being believed at the same time. When new incompatibilities are recognized, dependency-directed backtracking is used.

All these systems maintain one set of currently believed propositions, the "in" set, all other propositions are equally "out". When backtracking causes a currently in hypothesis to be put out, dependency records are used to decide which other propositions must now be out, and which can be brought in.

We have worked out the underlying theory of a more flexible belief revision system based on the relevance logics of [Anderson and Belnap, 1975] as discussed by [Shapiro and Wand, 1976]. In this system, each proposition is of the form A, t, c, r , where A is some formula (wff), t is an origin tag used to distinguish hypotheses from derived propositions, and c and r are sets of hypotheses known as the core context (CC) of A , and the restriction set (RS) of A , respectively. Instead of marking each proposition in or out, a current context of beliefs B will be maintained. B is simply a set of hypotheses. Each proposition A, t, c, r is currently believed just in case c is a subset of B and r is disjoint from B . Reasoning may be done on propositions whether or not they are currently believed, because the rules of inference specify how to calculate the origin tag, CC and RS of a

child proposition from its parent propositions. The rules of inference also specify when two propositions may not be used because their origin tags, CCs or RSs are incompatible.

The rules of inference of this system are presented in [Martins and Shapiro, 1981], along with further discussions of motivations and examples. Implementation of this system as part of the SNePS deductive semantic network processing system will be carried out in the next stage of this research project.

2 Bi-directional Inference

Over the past year, we have added a forward reasoning facility to SNIP, the SNePS Inference Package, which previously only had backward reasoning and a restricted forward reasoning that was limited to making use of new information which is relevant to previously asked questions (see [Shapiro, 1979]).

There is still only one class of rules -- any rule can be used for forward as well as backward reasoning. There are two ways of adding information to the network: BUILD adds the information, but does not trigger forward inference; ADD adds the information and triggers forward inference. There are also two ways of asking for information: FIND finds only explicitly stored information; DEDUCE triggers backward inference to help answer the question.

SNIP is implemented in MULTI [McKay and Shapiro, 1980], a multi-processing system that provides a producer-consumer model [Kaplan, 1973] of inference. If forward inference finds a consumer interested in its new information, no other potential consumers (rule antecedents) are looked for. Similarly, if backward inference finds a producer with the required information, no other potential producers (rule consequents) are looked for. Thus, our combined forward/backward inference provides the same savings of fan-out as bi-directional search provides over uni-directional search. We therefore term it bi-directional inference. It is discussed more fully in [Martins, McKay and Shapiro, 1981].

Forward, backward and bi-directional inference are discussed in a more general context in [Shubin, 1981], where they are compared with data-flow, lazy evaluation and a proposed "bi-directional computation."

3 Active Connection Graphs

The producer-consumer model of inference mentioned above is implemented by a set of communicating MULTI processes which may be viewed as an active version of predicate connection graphs [Kowalski, 1975; Sickel, 1976]. This active connection graph may also be viewed as an AND/OR problem reduction graph in which the root node represents a query and rules are problem reduction

operators. The system is designed so that if a node representing some problem is about to create a node for a subproblem, and there is another node already representing that subproblem, or some more general instance of it, the parent node can make use of the extant node and so avoid solving the same problem again. If, instead, the extant node is a more specific instance of the proposed subproblem, the results already produced by it are immediately available to the parent, and the new more general node can supplant the older more specific node. This method enables SNIP to handle recursive rules easily with no additional mechanism.

.Active connection graphs are described more fully in [McKay and Shapiro, 1981] in a terminology abstracted from the specifics of the SNePS representation to make the results of this research more available to the general theorem-proving and deductive question-answering community.

4 Combining Path-Based and Node-Based Inference

In [Shapiro, 1978], we discussed the difference between path-based and node-based inference in semantic networks. Essentially, node-based inference involves deducing a structure of nodes according to a rule which is, itself, a structure of nodes used as a node-structure pattern. Path-based inference involves deducing an arc between two nodes based on the existence of a path of arcs between those same two nodes according to a rule expressed as a path grammar.

We suggested that path-based inferences could be added to a SNePS-like system by incorporating it into the network match routine. This has now been done by Rohini Srihari as a Master's Project. In this system, path-based and node-based inferences are integrated. A node-structure pattern might match a virtual structure some of whose arcs were deduced to exist by a path-based inference performed by the match routine. Sections of the path used by the path-based inference might have been constructed by an earlier node-based inference.

Motivations, the syntax of path-based inference rules, examples, and implementation notes are in [Srihari, 1981].

5 Intensional Concepts in Propositional Semantic Networks

In [Maida and Shapiro, 1981], we discuss the proposition that all nodes in a propositional semantic network represent intensional concepts rather than extensions. This follows from the Uniqueness Principle -- that every concept represented in a semantic network is represented by a unique node -- and from our desire to use semantic network to represent human knowledge structures rather than events occurring in the real world.

This proposition has interesting implications for knowledge

representation and reasoning in deductive semantic networks. Among these are a nice solution to McCarthy's "telephone problem" [McCarthy, 1979] -- knowing that Mary's telephone number is the same as Mike's telephone number, from "Pat dialled Mike's telephone number" we can infer that "Pat dialled Mary's telephone number"; but from "Pat knows Mike's telephone number", we cannot infer that "Pat knows Mary's telephone number". From the intensional concept proposition, we conclude that Mary's telephone number and Mike's telephone number are different intensional concepts and are, therefore, represented by different nodes in the network. The system also has a node representing the (intensional) proposition that "Mary's telephone number" and "Mike's telephone number" are co-referential. The system can also have a node-based (or path-based) inference rule that if a referentially transparent predicate applies to some intensional concept, it also applies to any co-referential concept. That rule and the information that "dial" is referentially transparent will allow the dialling inference to be made, but since "know" is not referentially transparent, the knowing inference will not be made.

The intensional concept proposition makes referential opacity the norm -- referential transparency is easily performed where explicitly permitted. Systems that use extensional representation must assume that referential transparency is the norm and must block it in opaque contexts. This is considerably more difficult.

References

1. Anderson, A. and Belnap, N. [1975] Entailment. Princeton University Press, Princeton, NJ.
2. Doyle, J. [1979] A truth maintenance system. Artificial Intelligence 12, 3, 231-272.
3. Kaplan, R.M. [1973] A multi-processing approach to natural language understanding. Proc. NCC, AFIPS Press, Montvale, NJ, 435-440.
4. Kowalski, R. [1975] A proof procedure using connection graphs. JACH 22, 4, 572-595.
5. Maida, A.S. and Shapiro, S.C. [1981] Intensional concepts in propositional semantic networks. Technical Report No. 171, Department of Computer Science, SUNY at Buffalo, Amherst, NY.
6. Martins, J.; McKay, D.P.; and Shapiro, S.C. [1981] Bi-directional inference. Technical Report No. 174, Department of Computer Science, SUNY at Buffalo, Amherst, NY.
7. Martins, J., and Shapiro, S.C. [1981] A belief revision system based on relevance logic and heterarchical contexts. Technical Report No. 175, Department of Computer Science,

SUNY at Buffalo, Amherst, NY.

8. McKay, D.P., and Shapiro, S.C. [1980] MULTI - A LISP based multiprocessing system. Proc. 1980 LISP Conference, Stanford University, Stanford, CA, 29-37.
9. McKay, D.P., and Shapiro, S.C. Using active connection graphs for reasoning with recursive rules. Department of Computer Science, SUNY at Buffalo, Amherst, NY.
10. Shapiro, S.C. [1979] The SNePS semantic network processing system. In N.V. Findler, ed. Associative Networks: The Representation and Use of Knowledge by Computers. Academic Press, New York, 179-203.
11. Shapiro, S.C. [1978] Path-based and node-based inference in semantic networks. In D. Waltz, ed. TINLAP-2: Theoretical Issues in Natural Language Processing. ACM, New York, 219-225.
12. Shapiro, S.C. and Wand, M. [1976] The relevance of relevance. Technical Report No. 46, Computer Science Department, Indiana University, Bloomington, IN.
13. Shubin, H. [1981] Inference and Control in Multiprocessing environments. M.S. Thesis, Department of Computer Science, SUNY at Buffalo, Amherst, NY.
14. Sickel, S. [1976] A search technique for clause interconnectivity graphs. IEEE Trans. on Computers C-25, 8, 823-835.
15. Srihari, R. [1981] Combining Path-based and node-based reasoning in SNePS. M.S. Project Report, Department of Computer Science, SUNY at Buffalo, Amherst, NY.
16. Stallman, R. and Sussman, G. [1977] Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. Artificial Intelligence 9, 2, 135-196.