Incremental Inference:

Getting Multiple Agents to Agree on What to Do Next

Gary C. Borchardt
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

Abstract

This paper presents a symbolic reasoning algorithm for use in the construction of *mixed-initiative interfaces;* that is, interfaces allowing several human or machine agents to share collectively the control of an ongoing, real-time activity. The algorithm, called Incremental Inference, is based on propositional logic and is related in structure to the Truth Maintenance System; however, the notion of *justifications* in the Truth Maintenance System is replaced with a simpler notion of *recency*. Basic properties of the Incremental Inference mechanism are described and compared with those of the Truth Maintenance System, and an example is provided drawn from the domain of SPECTRUM, a knowledge-based system for the geological interpretation of imaging spectrometer data.

I. Introduction

The scenario addressed by this research involves several agents simultaneously in control of a subordinate activity. A flexible partitioning of duties allows any one agent temporarily to control the entire activity or all agents to control various aspects of the activity. A centralized interface serves to coordinate the requests of the agents. This scenario has applications in the shared remote/local control of robot mechanisms in space or undersea, in data interpretation or diagnosis tasks managed jointly by a human operator and one or more expert systems, and in the coordinated supervision of an activity by several expert systems with possibly overlapping areas of expertise.

Several published results relate to this problem, yet few true "mixed-initiative" systems exist to date. Probably the most directly relevant work lies in the area of non-monotonic reasoning, especially Truth Maintenance Systems [Doyle, 1979; McDermott and Doyle, 1980; McAllester, 1980; McAllester, 1982] and the Assumption-based Truth Maintenance System of de Kleer [de Kleer, 1986]. In particular, Doyle discusses the use of the Truth Maintenance System as a medium for dialectic argumentation between two or more agents. De Kleer discusses the utility of the ATMS approach in tracking multiple contexts (e.g., those applying to each of several agents) simultaneously. Also relevant is the work concerning focus in human dialog (e.g., [Grosz, 1977]) and the work in distributed problem solving (e.g., [Davis and Smith, 1983; Corkill and Lesser, 1983; Rosenschein and Genesereth, 1985]).

While the Truth Maintenance System is apparently well-suited to the mixed-initiative interface task, there are also drawbacks to this approach. This is best illustrated by a simple example. If in the course of mixed-initiative control of some process one of the agents brings a particular activity into consideration, and from this it is inferred that a particular plan is now active, then if later that activity is taken out of consideration (perhaps due to its completion), we are obligated within a Truth Maintenance System to retract the inference concerning the associated plan, as there is no longer a measure of *well-founded support* for this conclusion. This is, of course, merely the process of carrying out truth maintenance. In the context of mixed-initiative interfaces, however, this actually gets in the way, as we would rather keep the plan "active" until we are forced to change its status, thereby minimizing the number of changes to be endorsed by the interested parties. In general, this amounts to a process of taking up inferred values *as new default assumptions,* retaining these as long as they do not conflict with other values of greater "recency."

The Incremental Inference algorithm amounts to a restructuring of the Truth Maintenance System model to fit the above criteria, replacing the notion of *justifications* with a simpler notion of *recency*.[1] This effects a tradeoff in which the ability to reason based on well-founded support is exchanged for a heuristic capability in fluidly tracking a dynamically changing understanding between several parties.

The following sections outline the organization and properties of the Incremental Inference algorithm, discussing the nature of "inference" which may be drawn based on the notion of "recency" and examining parallels between the mechanism and that of the Truth Maintenance System, including a process of conflict resolution for the Incremental Inference mechanism analogous to that of dependency-directed backtracking in Truth Maintenance Systems.

II. The Algorithm

Structurally, the Incremental Inference algorithm is similar to the truth maintenance algorithm used by McAllester in his Reasoning Utility Package [McAllester, 1980; McAllester, 1982]. Inference is based on propositional logic expressions converted to conjunctive normal form. Propositions in the Incremental Inference algorithm amount to binary state variables for control of the subordinate process, however. These

[1]A preliminary description of the Incremental Inference algorithm appears in [Borchardt, 1987].

variables describe the the pursuit of particular goals, the implementation of plans, the execution of activities, the utilization of individual objects and types of objects, the achievement of milestones in the completion of tasks, and so forth. Logical constraints between the propositions involve, for example, the applicability of various plans to various goals, the compatibility of objects and types of objects with particular activities and the precondition restrictions placed on activities by individual milestones.

As a matter of convenience, no distinction is made between propositions and *literals* in this representation: the negation of a proposition is itself considered a proposition, with certain restrictions placed on the values held by complementary propositions. Thus, each clause may be considered to contain only propositions.

Each proposition is associated with two values: a *truth value,* which may be either *true, false* or *retracted,* and a *time value,* which is a number.[2] The truth value indicates whether or not the designated goal, plan, activity, etc. is currently considered "active" or "favored" by a consensus of all agents involved in the mixed-initiative environment. The time value indicates a measure of recency or strength of support for the truth value. Complementary propositions are constrained by the mechanism to have compatible truth values (that is, both *retracted* or one *true* and one *false*) and equal time values. A global *current time* is maintained, which is always greater than or equal to the most recent time values in the system.

Agents communicate with the system in two distinct ways, the first of which involves the submission of *requests*. A request is carried out by incrementing the global current time, assigning a new truth value to a selected proposition and updating the time value for the proposition to the new current time. In this case, the time value corresponds to a "timestamp," recording the point in time at which the requested truth value was assumed. The time value does *not* serve this function in the case of propositions with truth values derived from those of other propositions, however. This is somewhat clearer in the examples which follow.

Propogation of effects resulting from a request are computed locally at the level of individual clauses by a process of *stabilization.* Stabilization of a clause attempts to satisfy its implied logical disjunction by modifying the truth and time values of propositions where necessary, always protecting the current status of propositions with newer time values over those with older time values. This is the "heuristic" of the algorithm: while it is not guaranteed that modification of the proposition with least recency is the *best* choice, in many cases this is indeed a *good* choice, and at worst it produces a broad search for a new, consistent state, starting with modifications of the least recent values and working back toward the most recent values.

The stabilization process may result in *inference,* updating propositions to *true* or *false,* or *retraction,* updating propositions to the *retracted* state. Where inference occurs, the affected proposition is given a time value equal to the minimum time value among the remaining propositions in the clause. Thus, the strength of the inferred truth value is no greater than the weakest strength among the propositions which have made it so. This is somewhat analogous to the recording of justifications for inferred values in a Truth Maintenance System such as that of McAllester. The general rule for stabilization of an individual clause is given below.

STABILIZATION OF A CLAUSE *C*:

(If *C* contains only one proposition, assume the existence of an additional proposition within *C,* set to *false* at the current time.)

1.) *(Possible inference.)* If there is a single proposition *P* having the oldest time value in *C,* and none of the remaining propositions in *C* are *true,* update *P* to *true* at the minimum time value among the remaining propositions in *C*.

2.) *(Possible retraction.)* If several propositions *P1, P2, ..., PN (N > 1)* share the oldest time value within *C* and there are no propositions in *C* which are *true,* modify those of *P1, P2, ..., PN* which are *false* to *retracted,* leaving their time values unchanged.

A reasonably efficient algorithm for the stabilization of a clause performs an initial scan through the clause, computing the oldest time value among its propositions, the number of propositions having this time value, the second oldest time value among the propositions and the newest time value for a *true* proposition within the clause. Following the determination of these quantities, it is a simple matter to decide which case applies and to perform the appropriate action.

The other means of interaction between the agents and the mechanism involves a process of *refreshing.* Since the rule for stabilization of a clause is guided by the heuristic of "recency," propositions become more and more susceptible to change as their time values become less and less current. As a counteractive force, each agent is allowed to specify an *interest* in the propositions of any clause, and, given such an interest, the agent is queried prior to all modifications involving propositions within the designated clause (even if generated as a result of stabilizing other clauses). When queried concerning a tentative modification, an "interested" agent may attempt to block the modification by increasing the time value of the targeted proposition.[3]

In the simplest case, the refreshing process updates a proposition's time value to the global current time. The refreshing device forces the mechanism to reconsider the clause generating the tentative inference or retraction, selecting an alternative action. The refreshing process thus serves to allow agents to protect the status of various goals, plans, activities, etc., of current importance to them. The interests specified by the agents may be changed whenever desired and constitute a means by which the agents may partition the

[2]The time values may be arbitrary as long as they increase monotonically with actual "clock time." Integers are used here for simplicity.

[3]As a matter of "streamlining," the querying of interested agents is bypassed where a proposition to be modified already has a time value equal to the current time, is to be updated in time value only or has an initial truth value of *retracted.*

responsibilities for various aspects of the overall task, independent of the requests made by each of the agents.

III. A Simple Example

As an example, consider the set of two clauses:

(A v B v C) and
(NOT_A v D v E),

with initial truth and time values as follows.

A: (false,5)	*NOT_A: (true,5)*
B: (true,3)	*D: (false,1)*
C: (false,2)	*E: (false,1)*

If one of the participating agents submits a request to update proposition *B* to *false* at a new current time of *7*, a value of *true* is inferred for *C*. The time value assigned to *C* is the minimum of the time values for *A* and *B*. Assuming no refreshing of values occurs, the following state results.

A: (false,5)	*NOT_A: (true,5)*
B: (false,7)	*D: (false,1)*
C: (true,5)	*E: (false,1)*

On the other hand, if an agent "interested" in the first clause blocks the inference by refreshing *C* to the current time, a reevaluation of the first clause results in an inference of *true* for proposition *A*, giving it a time value of *7*. This value is echoed in a value of *false* at *7* for *NOT_A*, and a stabilization of the second clause results in the retraction of *D* and *E*.

A: (true,7)	*NOT_A: (false,7)*
B: (false,7)	*D: (retracted,1)*
C: (false,7)	*E: (retracted,1)*

If all of the above occurs, plus one of the agents interested in the second clause blocks the retraction of *E*, the following results.

A: (true,7)	*NOT_A: (false,7)*
B: (false,7)	*D: (true,7)*
C: (false,7)	*E: (false,7)*

Finally, if the initial inference for *C* and the subsequent retractions of *D* and *E* are all blocked, the resultant state contains values of *retracted* at time *7* for all of the propositions. In this case, an intermediate state with all propositions of the second clause set to *false* at *7* resolves to a state in which these are all *retracted* at *7*. Subsequent stabilization of the first clause then forces a retraction of *B* and *C*. As the refreshing of values has overturned even the initially requested value, a suitable action to take is to retreat to a previous "safe" point agreed upon by all agents.

IV. Properties of the Algorithm

Despite its lack of the usual apparatus for performing truth maintenance, the Incremental Inference mechanism does retain a limited capability of reasoning based on the notion of well-founded support. This reasoning capability is actually of the *monotonic* variety; that is, it does not tolerate changes in antecedents. This can be seen by considering the well-foundedness of propositional values starting at the current time and working backwards.

Following a request submitted by one of the agents in the mixed-initiative environment, all propositions whose time values stabilize at the new current time may be taken to have well-founded support, based on the truth value of the proposition designated in the request and all propositions refreshed to the current time, provided these propositions themselves have retained their designated values. Likewise, if the proposition designated in the previous request cycle plus all propositions refreshed during that cycle have retained their designated values, then all propositions with time value equal to the previous current time may be taken to have well-founded support, based on the collective requested and refreshed propositions of the last two cycles, and so on.

In general, if we take care to note the time of the most recent request cycle for which either the designated proposition or one of the propositions refreshed during that cycle has been overridden, we may conclude that all propositions with time values newer than this time do indeed have well-founded support, based on the collective propositions designated and refreshed in all cycles since the noted time.

In the context of mixed-initiative interfaces, however, the notion of well-founded support is of lesser concern. Here, in all cases, one may consider a derived truth value for a proposition to be an indication that, in order to protect the *status quo* for "some other" proposition with equal time value, it was necessary to update the proposition under consideration as indicated.

One noteworthy aspect of the inference/retraction process in the Incremental Inference mechanism involves the nature of the *retracted* truth value. This value may be thought of as signaling the presence of a contradiction regarding the proposition in question.[4] In fact, the *retracted* value serves as a medium through which a process analogous to that of dependency-directed backtracking in Truth Maintenance Systems is carried out. Inspection of the rule for stabilization of a clause reveals that a *retracted* value assists in the generation of inferences and retractions much as would a value of *false*. The negation of a retracted proposition, however, also behaves as if it were *false* in all clauses containing it. The net effect is that inferences and retractions are propogated among the propositions with time values older than a retracted proposition as if the retracted proposition were both *true* and *false*.[5] This continuance of inference/retraction may often result in a resolution of the conflict causing a retraction. This occurs when one or more agents refresh values, blocking either the "true" or "false" aspects of the retracted proposition. In such cases, the refreshed values propogate back toward the retracted proposition in a sort of "reflected wave" motion, forcing the retracted proposition to assume either the *true* or *false* state.

Regarding the heuristic nature of the mechanism in tracking the "shifts of attention" generated by requests, it

[4]It is for this reason that a simpler designation of *unknown* as in McAllester's system was not used.

[5]In a similar manner, de Kleer's ATMS also continues to perform inference based on individual facts involved in contradictions, as long as they do not combine in support with their contradictory counterparts.

may be noted that increased intricacy of logical constraints tends to promote the retraction process, as it is more likely that multiple propositions within a clause end up with the same time value. In such cases, the mechanism relies more heavily upon the interested agents for direction through the refreshing of values. Where the logical constraints are fairly "loose," the time values tend to be more widely distributed; thus, inference prevails over retraction.

Two additional properties of the mechanism involve questions of completeness for the inference produced and eventual termination of the stabilization process following a new request. Similar to McAllester's clause-based reasoning mechanism, the inference produced by the Incremental Inference system is logically incomplete. That is, in some situations, usually involving "case analysis," the mechanism will fail to make inferences which logically should be made. This can result in global states in which certain propositional truth values are inconsistent with other propositional truth values. As each new focus of attention for the system may be achieved by a sequence of several requests, however, it is possible to work around the incompleteness by gradually approaching a desired global state, resolving conflicts due to previously undetected inconsistencies as they appear until a global state with all propositions set to *true* or *false* exists, for which there can be no inconsistencies. As well, areas subject to incomplete inference may often be "bridged" through the inclusion of additional clauses in the system.

Regarding the termination of the stabilization process, the mechanism is guaranteed to converge upon a new, globally stable state following a new request. This can be seen in the nature of the rule for stabilization of a single clause. A proposition, when updated, is normally given a newer time value. The only exception involves the modification of a proposition from *true* or *false* to *retracted,* in which case the time value remains unchanged. It is thus possible for at most two truth values (*true* or *false,* then *retracted*) to be associated with a proposition before the time value *must* be incremented. The global current time sets an upper bound on the increase of time values. The example in Section III illustrates this, as the final remaining option is a retraction of all propositions in the clauses at the current time.

V. Extensions to the Algorithm

A useful extension of the Incremental Inference mechanism is to represent the interests of the participating agents with respect to individual clauses not as external parameters, but as propositions in the mechanism itself. If a proposition representing an interest is *true* or *retracted,* the agent is considered to be interested in the specified clause; if it is *false,* the agent is not interested. This allows the construction of multilayer Incremental Inference reasoning systems, where a higher-level system is used to reach a consensus regarding interests in a lower-level system describing the current state of affairs. This approach has been taken in the SPECTRUM system, as illustrated in the next section. Using such a device, it is possible to add an additional layer of "interests in interests," so that one agent may, for instance, block another agent's attempt to relinquish interest in a particular area. As well, logical constraints may be set up such that if one agent relinquishes interest in an area of the decision making, some other agent is then forced to take up an equivalent interest.

A second extension, also employed in the SPECTRUM system, is to include a set of higher-level structures, called *decision sets,* representing groups of related clauses. This device springs from the fact that, whereas the clausal constraint is of the form *at least one,* a corresponding constraint of the form *at most one* has an equivalently simple rule for stabilization, as follows.

STABILIZATION OF A DECISION SET *D* OF TYPE *AT MOST ONE*:

1.) *(Possible inference.)* Modify all propositions with time values older than the newest *true* or *retracted* proposition(s) in *D* to *false* at the time of the newest *true* or *retracted* proposition(s).

2.) *(Possible retraction.)* If several propositions *P1, P2, ..., PN* $(N > 1)$ share the status of being the newest *true* or *retracted* values in *D,* modify those of *P1, P2, ..., PN* which are *true* to *retracted,* leaving their time values unchanged.

In this case, inference and retraction may both occur during the same stabilization. The above rule is equivalent in effect to the stabilization of the N!/(N-2)!2! clauses implied by the *at most one* constraint (e.g., for three propositions *A, B* and *C,* this is equivalent to the clauses *(NOT_A* v *NOT_B), (NOT_A* v *NOT_C)* and *(NOT_B* v *NOT_C)*). Choosing combinations of the above rule and that given previously for clauses, four types of decision sets are produced: *unconstrained, at least one, at most one* and *exactly one*. Decision sets behave in most respects like ordinary clauses; that is, agents may specify interests in particular decision sets, and the decision sets are stabilized as single entities. The rules for stabilization vary according to the type, however. For the *exactly one* constraint, the stabilization rule for *at most one* is applied, followed by the rule for *at least one*.[6] The *exactly one* constraint is extremely useful in building compact representations of logical constraints and has been employed in the area of resolution-based inference [Tenenberg, 1985], in the SNePS system [Shapiro, 1979] and in the ATMS model [de Kleer, 1986].

A third useful extension to the Incremental Inference algorithm concerns a variation of the refreshing process. In some cases, especially when the inference or retraction process attempts to update a proposition using a time value much less than the current time, it is convenient to be able to merely "resist" the change instead of unconditionally "rejecting" it. This amounts to a partial refreshing of the time value: just enough to prevent the tentative inference or retraction, but no further.[7] In this case, the "resisting" agent may

[6]This ordering is necessary due to an occasional interaction of the rules.

[7]For a proposition subject to inference, this is the time value for the inference. For a proposition subject to retraction, this is the same value plus a small increment.

again be queried at a later time concerning the modification of the same proposition given support of greater recency.

VI. An Example Drawn from the SPECTRUM Domain

The Incremental Inference mechanism has been incorporated within SPECTRUM, a knowledge-based system for geological interpretation of imaging spectrometer data. An initial overview of the SPECTRUM application appears in [Borchardt, 1986]. The example described below has been simplified somewhat from the SPECTRUM domain and involves the Incremental Inference algorithm in a mixed-initiative user/system interface for control of a particular segment of the analysis, involving a variant of the *Isodata* algorithm [Duda and Hart, 1973] for the clustering of multidimensional data points into uniform, distinct classes.

The Isodata algorithm consists of a preliminary activity, *initialize,* followed by a cyclic repetition of three activities, *cluster, extract* and *merge,* with *merge* occurring zero, one or many times before each subsequent return to the *cluster* activity. Individual propositions in the Incremental Inference mechanism are used to represent each of the four activities. Two additional propositions, *using_a_map* and *using_some_plots* describe data quantities associated with the activities. A special proposition, *handshake* is used by either agent to signal a desire to execute the currently specified activity. Critical factors for mixed-initiative control are the determination of whether or not to perform one or more merging operations prior to each successive iteration and when to halt the process. A number of decision sets for the interface are thus set up as indicated below. Each entry corresponds to a function call defining a decision set (name, set of elements) or specifying the interest of an agent in a particular decision set (decision set, agent, proposition or constant truth value representing the designated interest).[8]

exactly_one(ds1
[not_isodata_plan initialize cluster extract merge])

at_most_one(ds2
[not_using_a_map initialize cluster extract])

at_most_one(ds3 [using_a_map merge])

at_most_one(ds4 [not_using_some_plots cluster merge])

at_most_one(ds5 [using_some_plots initialize extract])

at_least_one(ds6
[not_isodata_plan user_merge_interest
spectrum_merge_interest])
associated_interest(ds6 spectrum true)

at_least_one(ds7 [not_cluster forego_merge])
associated_interest(ds7 user user_merge_interest)
associated_interest(ds7
spectrum spectrum_merge_interest)

unconstrained(ds8 [using_a_map using_some_plots])
associated_interest(ds8 user user_quantity_interest)

unconstrained(ds9 [handshake])
associated_interest(ds9 user true)
associated_interest(ds9 spectrum true)

The following truth and time values serve as a point of departure for the example.

current time = 16		*isodata_plan:*	*(true,15)*
handshake:	*(false,16)*	*initialize:*	*(false,15)*
using_a_map:	*(true,15)*	*cluster:*	*(false,15)*
using_some_plots:	*(false,15)*	*extract:*	*(true,15)*
forego_merge:	*(false,14)*	*merge:*	*(false,15)*

user_merge_interest:	*(true,12)*
spectrum_merge_interest:	*(false,12)*
user_quantity_interest:	*(false,13)*

In this state of affairs, the *extract* activity is currently "in focus" and the user has specified an interest in protecting the opportunity to perform a *merge* operation at the appropriate time. The following is then a possible mixed-initiative control sequence continuing from this point.

1.) Following a request by the user, the current time is incremented to *17* and *using_some_plots* is updated to *true* at the new current time. This results in an update of *initialize* and *extract* to *false* at *17* plus an attempted retraction of the three remaining propositions in *ds1, not_isodata_plan, cluster* and *merge.* SPECTRUM, having an interest in the proposition *isodata_plan,* blocks its retraction. The user accepts the retraction of *cluster,* but when queried about a subsequent retraction of *forego_merge,* blocks this (that is, the user does not wish to forego the *merge* activity). Thus, *cluster* returns to a status of *false* at a newer time *17.* Due to the refreshing operations by SPECTRUM and the user, the third retracted proposition, *merge,* is then updated to *true* at time *17.* The user requests an update of *handshake* to *true,* SPECTRUM allows this update, and the merge operation is executed, followed by a reset of *handshake* to *false.*

2.) Next, the user relinquishes interest in protecting *merge* by requesting that *user_merge_interest* be set to *false.* SPECTRUM is queried regarding this change and regarding the subsequent update of *spectrum_merge_interest* to *true.* SPECTRUM accepts both updates, completing the exchange of this interest responsibility to SPECTRUM. As a separate request, the user then expresses a new interest in the types of data quantities associated with the activities. Thus, *user_quantity_interest* receives a new truth value of *true.*

3.) At this point, SPECTRUM takes initiative in requesting consideration of the *cluster* activity. As the user is at present interested only in data quantity types, the user acknowledges only the update of *using_a_map* to *true.*

[8] The syntax here draws from the STAR language used in the implementation of the SPECTRUM system [Borchardt, 1986].

SPECTRUM, now posed as the responsible party for *forego_merge,* must grant permission for the update of this proposition to *true*. SPECTRUM then issues a request for *handshake* to be updated to *true*. The user accepts this update, and the *cluster* activity is executed, followed by a subsequent reset of *handshake* to *false*.

VII. Conclusion

The Incremental Inference algorithm provides a general framework supporting a variety of mixed-initiative interface applications. Alternative configurations may be constructed in which particular agents provide only requests or only react to the requests of other agents. The partitioning of duties may be "lateral," that is, dividing interests according to relatively disjoint portions of the decision space, or they may be more or less "vertical," with certain agents taking an interest in more general issues while other agents take an interest in the specific issues underlying these general issues. The boundaries of responsibility assigned to various agents may change dynamically, allowing sudden shifts in control in unpredicted circumstances.

As a commonsense reasoning mechanism, the Incremental Inference algorithm is interesting due to its retention of inferred values as new default assumptions, held as long as they are consistent with future values. This paper has attempted to explore some of the differences associated with reasoning based on *recency* as contrasted with the standard notion of well-founded support. This style of reasoning provides a useful heuristic in the realm of mixed-initiative interfaces and may apply equally well in other, related domains.

Acknowledgements

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The author would like to thank Marc Vilain, Jerry Solomon and Steven Vere for insightful comments regarding the algorithm and its use. Steven Groom implemented a large portion of the code and discovered many nuances in its behavior.

References

[Borchardt, 1986] Gary C. Borchardt. STAR: a computer language for hybrid AI applications. In J. Kowalik, editor, *Coupling Symbolic and Numerical Computing in Expert Systems,* pages 169-177, North-Holland, Amsterdam, 1986.

[Borchardt, 1987] Gary C. Borchardt. Mixed-initiative control of intelligent systems. In *Proceedings NASA Space Telerobotics Workshop,* Pasadena, California, January 1987.

[Corkill and Lesser, 1983] Daniel D. Corkill and Victor R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings IJCAI-83*, pages 748-756, Karlsruhe, West Germany, August 1983.

[Davis and Smith, 1983] Randall Davis and Reid G. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20(1):63-109, January 1983.

[de Kleer, 1986] Johan de Kleer. An assumption-based TMS. *Artificial Intelligence* 28(2):127-162, March 1986.

[Doyle, 1979] Jon Doyle. A truth maintenance system. *Artificial Intelligence* 12(3):231-272, 1979.

[Duda and Hart, 1973] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, New York, 1973.

[Grosz, 1977] Barbara J. Grosz. The representation and use of focus in a system for understanding dialogs. In *Proceedings IJCAI-77*, pages 67-76, Cambridge, Massachusetts, August 1977.

[McAllester, 1980] David A. McAllester. *An Outlook on Truth Maintenance*. AI Memo No. 551, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, August 1980.

[McAllester, 1982] David A. McAllester. *Reasoning Utility Package User's Manual, Version One.* AI Memo No. 667, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, April 1982.

[McDermott and Doyle, 1980] Drew McDermott and Jon Doyle. Non-monotonic logic I. *Artificial Intelligence* 13(1-2):41-72, 1980.

[Rosenschein and Genesereth, 1985] Jeffrey S. Rosenschein and Michael R. Genesereth. Deals among rational agents. In *Proceedings IJCAI-85,* pages 91-99, Los Angeles, California, August 1985.

[Shapiro, 1979] Stuart C. Shapiro. The SNePS semantic network processing system. In N. V. Findler, editor, *Associative Networks,* pages 179-203, Academic Press, New York, 1979.

[Tenenberg, 1985] Josh D. Tenenberg. Taxonomic reasoning. In *Proceedings IJCAI-85,* pages 191-193, Los Angeles, California, August 1985.