Belief Revision and Truth Maintenance Systems: An Overview and a Proposal

CSE Technical Report 98-10

Stuart C. Shapiro
Department of Computer Science and Engineering
and Center for Multisource Information Fusion
and Center for Cognitive Science
State University of New York at Buffalo
226 Bell Hall
Buffalo, NY 14260-2000
shapiro@cse.buffalo.edu

December 31, 1998

Abstract

This paper presents a very high-level introduction to Belief Revision and Truth Maintenance Systems, including explanations of basic terminology, a brief history, and a comment on the two literatures: the TMS literature, and the AGM Belief Revision literature. More extensive surveys in each of the two literatures are cited. The paper concludes with a proposal to continue the investigation of the two traditions, and to see how they might be brought together.

1 Introduction

"Belief Revision" and "Truth Maintenance" are essentially equivalent terms referring to operations of subsystems of knowledge-based systems (KBSs). We will assume that a KBS is a computer system consisting of:

- a knowledge-base (KB) of "assertions";
- a set of inference methods for inferring additional assertions from the ones already in the KB.

We will assume that the assertions satisfy the syntax and semantics of some well-defined logic, that the inference methods are sound with respect to those semantics, and that we can use standard terminology from logic to refer to all aspects of the KBSs. The assertions may be:

- ground atomic propositions, often referred to as "facts";
- closed, non-atomic propositions, containing propositional connectives and/or quantified variables, often referred to as "rules";
- supplied by a user or some outside system, referred to as "assumptions" or "hypotheses";
- inferred via the inference methods from other assertions, referred to as "inferences" or "derived assertions".

Although the terms "truth", "fact", and "knowledge" are used, no KBS can guarantee that any assertion it contains is true, and therefore a fact in the strict sense of that term, nor that any set of them constitutes knowledge. "Belief" would be a more accurate term than "knowledge", and this is why some researchers prefer "belief revision" to "truth maintenance." In this document, we will use these two terms interchangeably.

KBSs may need belief revision either because:

- assertions are entered by/from multiple sources that may contradict one another;
- new assertions may contradict old ones because the world the assertions are about has changed;
- old assertions might be retracted either because the world which it is about has changed, or because the source of the assertion no longer wants it in the KB.

A belief revision system (BRS) (or truth maintenance system (TMS)) is a subsystem of a KBS that is intended to solve one or more of the following problems:

- Notice when the KBS contains a contradiction.
- If the KBS is found to contain a contradiction, identify the assertions that were used to derive the contradiction. Call these "possible culprits."
- From among the possible culprits, choose one as the culprit.
- Delete the culprit from the KB.
- Delete from the KB all assertions derived from the deleted culprit.
- Prevent the reintroduction of deleted contradictions.

Even ignoring the issue of contradictions, a BRS should be able to perform the following tasks:

1. Given an assertion, find the assertions used to derive the given one.
2. Given an assumption, find the assertions derived from it.

3. Delete an assertion and all assertions derived from it from the KB.

Some BRSs are designed only for propositional logics, and some are designed for predicate logics. Independently of this distinction, some are designed only for monotonic logics, while others are designed for non-monotonic (or "default") logics. We will assume familiarity with the propositional *vs.* predicate distinction. Monotonic logics satisfy the property that if an assertion A is derivable from some set of assertions Φ, then A is also derivable from any superset of Φ. Non-monotonic logics use rules of the form "If $A_1, \ldots, A_n$ are in the KB and $B_1, \ldots, B_m$ are not in the KB then C may be added to the KB." (Such a rule will be written below as $\{A_1, \ldots, A_n\}\{B_1, \ldots, B_m\} \Rightarrow C$.) Thus, in a non-monotonic logic, A might be derivable from Φ, but not from some superset of Φ. The significance of this for BRSs is that in a monotonic KBS, all the assertions from which some assertion was derived must be in the KB, but in a non-monotonic KBS, some derived assertion may depend on some other assertion's (the B_i) not being in the KB.

In some literature, assertions that might be retracted are called "defeasible," and reasoning that produces defeasible assertions is called "defeasible reasoning."

Recently published overviews of TMSs and BRSs include (Martins, 1991), (Gärdenfors, 1992), (Martins, 1992a), (Martins, 1992b), (Gärdenfors and Rott, 1995), and (Friedman and Halpern, 1996).

2 Dependency-Directed Backtracking

The first idea leading to TMSs was to use dependency-directed backtracking instead of chronological backtracking. Consider a KB to which assertions $A_1, \ldots, A_{100}$, have been inserted in order, following which, the addition of A_{101} reveals a contradiction. In chronological backtracking, assertion A_{100} would be retracted, and A_{101} would be reasserted. If the contradiction recurred, A_{99} would be retracted and A_{101} added again, *etc.* until A_{101} is added without a contradiction. Then all but the last retracted assertions could be reasserted. This is clearly inefficient. It would be more efficient if the contradiction somehow pointed directly to the assertions it depended upon. Then, one of them could be retracted immediately. This basic idea to use dependencies to order backtracking was called "dependency-directed backtracking" (Stallman and Sussman, 1977).

3 JTMSs

The first notion of a dependency to use in a KBS is to have each inferred assertion, C, depend on the assertions directly used by the reasoning step that produced C. For example, if C were produced by a single step of reasoning from the assertions A and B, and the rule $A \wedge B \Rightarrow C$, then C would depend on A and B. It was said that C was "justified" by A and B, and the TMSs based on this notion of dependency were called "Justification-Based TMSs" (Doyle, 1979). The actual assumptions on which C depends could be found by tracing back through A, B, their justifications, their justifications, etc.

JTMSs typically deal only with propositional logic, and since JTMSs only include atomic propositions and negations of atomic propositions (that is, in clause terminology, "literals") in their justification graphs, the rule $A \wedge B \Rightarrow C$ is not considered one of the justifications of C in the above example. JTMSs are implemented using a kind of dependency graph with two kinds of vertices: literals, called "nodes"; and "justifications". Arcs go from nodes to justifications and from justifications to nodes. In the above example, there would be arcs from A and from B going to a justification that had an arc to C, showing that C is justified by A and B. Assumptions are represented by nodes whose justifications have no arcs coming into them. Nodes are also labeled "in" or "out", indicating whether the literal represented by the node is in the KB or not.

JTMSs handle the three BRS tasks as follows.

1. Given an assertion, find the assertions used to derive the given one: Follow the chain of arcs backwards from the assertion to its justification, to the nodes with arcs going to those justifications, to their justifications, etc., until coming to the set of assumptions. All assertions found along this search were used in the derivation(s) of the original assertion.

2. Given an assumption, find the assertions derived from it: Follow the chain of arcs forwards from the assumption to the justifications it points to, to nodes those justifications point to, etc. The set of all nodes thus found were derived from the assumption.

3. Delete an assertion and all assertions derived from it from the KB: If the assertion is an assumption: Change the label of the assertion node from "in" to "out," and follow the chain of arcs forwards, as above, changing the label of all nodes found from "in" to "out"; however, if an encountered node has another justification or is already labeled "out", do not change its label and do not continue following arcs emanating from it. If the assertion is not an assumption: Find the assumptions used to derive it (task 1); choose one of these and delete it and all assertions derived from it from the KB (task 3, first clause); repeat this until the assertion's label is "out."

JTMSs for non-monotonic logics have "inverter" arcs to justifications from literals that must be out for the conclusion to hold. For example, if a rule were $\{D\}\{E\} \Rightarrow F$, there would be a justification with an arc to F, a normal arc from D to the justification, and an inverter arc from E to the justification. If D were labeled "in" and E were labeled "out," F would be labeled "in", but if D were "out" or F were "in", F would be "out." A non-monotonic JTMS works like a monotonic JTMS, but in task 3, it needs to handle the more general problem of changing a label either from "in" to "out" or from "out" to "in," and propagating the change. For example, an assertion may be deleted from the KB by changing the label of one of its assumptions from "out" to "in."

Following chains of justifications could be inefficient. Moreover, it was found that chains of justifications might contain cycles. These were among the motivations for ATMSs (see below). Nevertheless, justifications are still used because they are excellent for use by explanation systems.

4 Logic-Based TMSs

Logic-based TMSs, or LTMSs (McAllester, 1978), add the rule, in clause form, to the justification, and eliminate the directedness of the arcs. In the first example above, the nodes for A, B, and C would have arcs to a justification containing the clause $\{\neg A, \neg B, C\}$. The label of any one of the literals A, B, and C could then be considered to depend on the label of the other two. LTMSs also are generally based on propositional logic, but allow three labels: "true", "false", and "unknown".

5 TMSs as Services

JTMSs are generally implemented as service systems separate from the reasoning systems. Each time the reasoning system produces a new assertion, it sends the new assertion and its justifications to the JTMS, which records it. If belief revision needs to be performed, the information recorded by the JTMS is used. In such a JTMS, there is no logical principle connecting an assertion to its justifications.

LTMSs associate the reasoner more closely with the TMS by using the logic of the justification clauses for revision.

6 ATMSs

The problems with JTMSs (see above) motivated systems in which each inferred assertion is associated directly with the assumptions it depends on. These are, therefore called Assumption-Based TMSs (de Kleer, 1986). If a contradiction occurs, the assumptions associated with the contradiction form the set of possible culprits. Moreover, that set of assumptions may be recorded as being contradictory, and any operation that might form a set of assumptions that is a superset of a contradictory set may be blocked.

ATMSs handle the three BRS tasks as follows.

1. Given an assertion, find the assertions used to derive the given one: Every assertion points directly to the set or sets of assumptions it depends on, but there is no record of the intermediate assertions.

2. Given an assumption, find the assertions derived from it: Assumptions point directly to the assertions derived from them.

3. Delete an assertion and all assertions derived from it from the KB: The current KB is determined by a "current context" consisting of a set of assumptions. The current KB is the current context, plus all assertions at least one of whose assertion sets is a subset of the current context. To delete an assumption from the KB, remove it from the current context. All derived assertions with no remaining assumption set being a subset of the current context are no longer in the KB. To delete a derived assertion from the KB, at least one assumption from each assumption set must be deleted from the current context.

As for JTMSs, de Kleer's ATMS is a service separate from the reasoner. SNeBR, the ATMS of (Martins and Shapiro, 1988), however, unites the ATMS with the reasoner by calculating the assumptions of each inferred assertion directly from its parents according to a calculation rule associated with the rule of inference used to infer it. In the most common cases, the assumptions of an inferred assertion are the same as the parent(s) (for example, for the rule of and-elimination), or are the union of the parents' assertions (for example, for the rule of modus ponens). SNePS handles predicate logic using a variant of relevance logic (Anderson and Belnap, 1975; Anderson et al., 1992), which is a paraconsistent logic, meaning that a contradiction does not imply everything.

SNePSwD (Cravo and Martins, 1993) extends SNeBR to handle non-monotonic logic, but uses classical rather than relevance logic.

7 Choosing the Culprit

As described so far, JTMSs, LTMSs, and ATMSs are all capable of, given a contradiction, finding the possible culprits—the set of assumptions underlying the contradiction. How to choose the culprit to blame the contradiction on from the set of possible culprits, however, is much less clear. SNeBR (Martins and Shapiro, 1988), for example, presents all the possible culprits to the user for him or her to choose.

8 AGM Belief Revision

In parallel with the work on TMSs has been a body of work on belief revision, or "theory change", taken as having its start with the postulates for belief revision of (Alchourrón et al., 1985), though with roots at least as far back as (Gärdenfors, 1978). Following the literature, I will refer to this body of work as "AGM belief revision" (or AGM-BR), though it is probably the case that when people distinguish "belief revision" from "truth maintenance systems", what they mean by "belief revision" is AGM-BR.

AGM-BR is explicitly concerned with formulating postulates to characterize three operations of belief revision: adding a new assertion to a knowledge base ("expansion"); removing an assertion from a knowledge base ("contraction"); adding a new assertion to a knowledge base it is inconsistent with, and adjusting the result to restore consistency ("revision"). Revision can be accomplished by a step of contraction followed by a step of expansion. Among the basic constraints AGM-BR apply to these operations are

> "(i) The amount of information lost in a belief change should be kept minimal.
> (ii) In so far as some beliefs are considered more important or entrenched than others, one should retract the least important ones."
> (Gärdenfors and Rott, 1995, p. 38)

It can be seen that these two constraints give some advice on the issue of which of the possible culprits of a contradiction should be removed from the knowledge base. Nevertheless, the authors in the AGM-BR tradition express their postulates for belief revision in a very high-level way, and assume a KB that is logically closed, a potentially infinite set of assertions. They are not concerned, as are the authors in the TMS tradition, with describing how to implement actual systems. Most amazing is the fact that these two traditions seem to ignore each other. Almost no publication in one tradition cites any publication in the other tradition, a notable exception being (Gärdenfors and Rott, 1995). An interesting example of this is that according to (Friedman and Halpern, 1996, p. 426) the postulates of (Boutilier, 1993) and of (Freund and Lehmann, 1994) both amount to chronological backtracking—the very "problem" the TMS tradition was launched to solve—although Friedman and Halpern do not use the term "chronological backtracking" and do not refer to any publications in the TMS tradition.

9 Implementing AGM Constraints

The AGM constraints listed in the previous section provide some general advice for implementors who want their TMSs to choose the culprit from the set of possible culprits automatically. The advice of constraint (i) is to choose the culprit whose removal from the KB causes the fewest other assertions to be removed. I do not know of any implemented TMSs that do this. The advice of constraint (ii) is to rank the assertions by "importance" or "entrenchment", and to choose the culprit whose removal from the KB causes only the less important or less entrenched assertions to be removed. SNePSwD (Cravo and Martins, 1993) allows the user to give a partial importance ordering among the assertions, and uses that ordering for automatic culprit identification when possible.

10 Proposed Future Work

We propose to continue investigating belief revision and truth maintenance systems, especially for the domain of data fusion systems for enhanced situation assessment. A fruitful direction for this investigation is to assess the extent to which existing BRSs operate in accord with the AGM-BR constraints and postulates, and, to the extent they do not, investigating the design and implementation of one that does, and assessing the usefulness of such a design to the domain of data fusion systems for enhanced situation assessment. This will be useful because, as stated above, very few investigators have even cited the literature from both traditions, let alone analyzed one in terms of the other.

In implementing an experimental BRS that follows the AGM-BR constraints and postulates, we will build on SNePS 2.4 (Shapiro and Group, 1998), which is a knowledge representation and reasoning system with SNeBR (Martins and Shapiro, 1988) as its built-in BRS. As mentioned above, SNeBR is an ATMS that uses a version of predicate logic and is tied in closely with the SNePS reasoner. SNeBR performs the three basic tasks of a BRS, but presents all the possible culprits to the user for him/her to choose the culprit to be eliminated. In doing this, we will experiment with, and use ideas from SNePSwD (Cravo and Martins, 1993), which adds a BRS to an earlier version of SNePS. The BRS of SNePSwD allows for non-monotonic rules, and allows the user to specify a partial importance ordering among assertions, which it uses during belief revision. Thus, SNePSwD is sensitive to at least one of the AGM-BR constraints. We have already obtained and started experimenting with the latest version of SNePSwD, which seems to require the user to specify an explicit order using specific assertions already in the KB. However, Ehrlich (Ehrlich, 1995; Ehrlich and Rapaport, 1997) reports that in using an earlier version of SNePSwD she was able to put assertions into various categories, give an importance ordering to the categories, and have this ordering used in belief revision. This seems much easier to use than an explicit ordering among assertions. We have not yet determined if this feature is in the current version of SNePSwD, was in the earlier version of SNePSwD that Ehrlich used, or was added by Ehrlich on top of SNePSwD. Making this determination is among the very next steps we will take.

11 Acknowledgments

The author thanks James Llinas and Frances L. Johnson for conversations and assistance on this topic. This work was supported in part by the U.S. Army Communications and Electronics Command (CECOM), Ft. Monmouth, NJ.

References

Alchourrón, C. E., Gärdenfors, P., and Makinson, D. (1985). On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530.

Anderson, A. R. and Belnap, Jr., N. D. (1975). *Entailment*, volume I. Princeton University Press, Princeton.

Anderson, A. R., Belnap, Jr., N. D., and Dunn, M. (1992). *Entailment*, volume II. Princeton University Press, Princeton.

Boutilier, C. (1993). Revision sequences and nested conditionals. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 519–525, San Mateo, CA. Morgan Kaufmann.

Cravo, M. R. and Martins, J. P. (1993). SNePSwD: a newcomer to the SNePS family. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, 5(2&3):135–148.

de Kleer, J. (1986). An assumption-based truth maintenance system. *Artificial Intelligence*, 28(2):127–162.

Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, 12(3):231–272.

Ehrlich, K. (1995). *Automatic Vocabulary Expansion through Narrative Context*. Technical Report 95-09, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY.

Ehrlich, K. and Rapaport, W. J. (1997). A computational theory of vocabulary expansion. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 205–210, Mahwah, NJ. Lawrence Erlbaum Associates.

Freund, M. and Lehmann, D. (1994). Belief revision and rational inference. Technical Report TR 94-16, Hebrew University.

Friedman, N. and Halpern, J. Y. (1996). Belief revision: A critique. In Aiello, L. C., Doyle, J., and Shapiro, S. C., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, pages 421–431. Morgan Kaufmann, San Francisco.

Gärdenfors, P. (1978). Conditionals and changes of belief. In Niiniluoto, I. and Tuomela, R., editors, *The Logic and Epistemology of Scientific Change*, pages 381–404. North-Holland, Amsterdam.

Gärdenfors, P. (1992). *Belief Revision*. Cambridge Computer Tracts. Cambridge University Press, Cambridge.

Gärdenfors, P. and Rott, H. (1995). Belief revision. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Clarendon Press, Oxford.

Martins, J. P. (1991). The truth, the whole truth, and nothing but the truth: An indexed bibliography to the literature of truth maintenance systems. *AI Magazine*, 11(5):7–25.

Martins, J. P. (1992a). Belief revision. In Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, pages 110–116. John Wiley &Sons, New York, second edition.

Martins, J. P. (1992b). Truth maintenance systems. In Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, pages 1613–1622. John Wiley &Sons, New York, second edition.

Martins, J. P. and Shapiro, S. C. (1988). A model for belief revision. *Artificial Intelligence*, 35:25–79.

McAllester, D. A. (1978). A three valued truth maintenance system. AI Memo 473, Massachusetts Institute of Technology AI Lab, Cambridge, MA.

Shapiro, S. C. and The SNePS Implementation Group (1998). *SNePS 2.4 User's Manual*. Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY.

Stallman, R. and Sussman, G. J. (1977). Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, 9(2):135–196.