Implementing Integrity Constraints in an Existing Belief Revision System*
CSE Technical Report 2000-03

Frances L. Johnson and **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, USA
{*flj* | *shapiro*}*@cse.buffalo.edu*

Abstract

SNePS is a mature knowledge representation, reasoning, and acting system that has long contained a belief revision subsystem, called SNeBR. SNeBR is triggered when an explicit contradiction is introduced into the SNePS belief space, either because of a user's new assertion, or because of a user's query. SNeBR then makes the user decide what belief to remove from the belief space in order to restore consistency, although it provides information to help the user in making that decision. We have recently added automatic belief revision to SNeBR, by which, under certain circumstances, SNeBR decides by itself which belief to remove, and then informs the user of the decision and its consequences. We have used the well-known belief revision integrity constraints as a guide in designing automatic belief revision, taking into account, however, that SNePS's belief space is not deductively closed, and that it would be infeasible to form the deductive closure in order to decide what belief to remove. This paper briefly describes SNeBR both before and after this revision, discusses how we adapted the integrity constraints for this purpose, and gives an example of the new SNeBR in action.

Introduction

Belief revision, or belief change, is the term used to describe any change in a knowledge base. The form of belief revision discussed in this paper is removal of propositions from a knowledge base that is known to be inconsistent in order to restore consistency. This is especially important to information fusion, where information is combined from multiple sources which might contradict each other. This paper describes some belief revision theories and the considerations that arose when they were implemented and added to the existing belief revision subsystem of a mature knowledge representation and reasoning system, SNePS(Shapiro & The SNePS Implementation Group 1999; Shapiro & Rapaport 1992).

These considerations center around the impossibility of implementing deductive closure and the proper weighting of belief revision guidelines. We address the need to formalize theories that take into account the fact that deductive closure cannot be guaranteed in a real-world, need-based, implemented system. We also explore one technique for following the belief revision guideline of minimizing damage to the belief space while retaining the most credible beliefs.

The next section provides the background necessary to understand the alterations we made to SNePS. Included are brief descriptions of the following: (a) the different groups doing belief revision research, (b) the integrity constraints we intend to implement, (c) SNePS and its belief revision sub-system, (d) the status of constraint adherence before system alterations, and (e) previous research to improve adherence.

The following two sections discuss the current changes made to improve adherence and implement automatic belief revision (autoBR). The latter gives the user a sense of non-monotonicity, because it is possible to add a belief to the belief space and, consequently, lose a previously held belief. However the underlying relevance-style, paraconsistent logic remains monotonic.

The final section contains conclusions and plans for future work.

Background

Theory vs. Implementation

Belief revision research can be divided into two groups: Theoretical vs. Implementations. These two groups differ in the amount of information assumed to be in a knowledge base. The theoretical researchers develop postulates about how a knowledge base should react during revision based upon a number of guidelines. One of these is the assumption of deductive closure, which means that everything derivable is contained in the knowledge base—a deductively closed belief space (DCBS), which is infinite in size and impossible to implement, although it can be simulated with limitations.

Implemented belief spaces must be finite and of reasonable size, and the reasoning operations performed on them must be done in a finite and reasonable time. Beliefs must be added gradually over time and not all

*This work was supported in part by the US Army Communications and Electronics Command (CECOM), Ft. Monmouth, NJ through a contract with CACI Technologies.

derivable beliefs can be guaranteed to be present in the knowledge base. We refer to this kind of belief space as a deductively open belief space, or DOBS(Johnson & Shapiro 2000). In a DOBS, it is possible for a proposition to be derivable from the existing belief space without it being present in that belief space. If that proposition is the negation of another proposition derivable from the belief space, the belief space would be inconsistent, but not known to be inconsistent—i.e. the system would be unaware of the inconsistency. The normal meaning of the word *consistent* cannot, therefore, be applied to a DOBS. For the purpose of this paper, however, the term consistent will be used to describe a belief space or belief set that has no known contradictions (unless otherwise noted).

Even research on finite belief bases still refers to deductive closure in its determination of consistency or when considering belief revision postulates (Hansson 1993a; 1993b; Nebel 1989; Alchourrón & Makinson 1985). Consistency is determined by the presence of a contradiction in the *implicit* beliefs of a belief base. How can this reliably be implemented? Even if it can be simulated on a small scale knowledge base, the theories developed might be suspect if applied to a very large system. Nebel voiced these concerns as well (Nebel 1989).

We address the need to formalize theories that take into account the fact that deductive closure cannot be guaranteed in a real-world, need-based, implemented system—even if the system is restricted by something as simple as the user needing a response within one minute. These theories need to suggest a belief revision technique that:

- takes time and complexity limitations into account
- recognizes that adhering to these limitations might result in revision choices that are poor in hindsight
- catches and corrects these poor choices as efficiently as possible.

Integrity Constraints for Belief Revision

Gärdenfors and Rott (Gärdenfors & Rott 1995) discuss postulates for belief revision using a coherence approach. These postulates, first presented in (Alchourrón, Gärdenfors, & Makinson 1985), are based on four integrity constraints (paraphrased below):

1. a knowledge base should be kept consistent whenever possible;
2. if a proposition can be derived from the beliefs in the knowledge base, then it should be included in that knowledge base;
3. there should be a minimal loss of information during belief revision;
4. if some beliefs are considered more important or entrenched than others, then belief revision should retract the least important ones.

These constraints come from the Theorist group, so strict adherence to constraints 1 and 2 in an implemented system is impossible. The proper weighting and combining of constraints 3 and 4 remains an open question in belief revision research and a challenge for both the Theorists and the Implementers.

SNePS and SNeBR Before Alteration

SNePS SNePS is a logic- and network-based knowledge representation, reasoning, and acting system designed to constitute the mind of a natural language competent cognitive agent. The underlying logic of SNePS is a monotonic, relevance-style, paraconsistent logic (Martins & Shapiro 1988). One way that users can interact with SNePS is through the SNePSLOG interface—an interface which allows the user to input propositions in a style that uses "predicate calculus augmented with SNePS logical connectives" (Mckay & Martins 1981; Shapiro & The SNePS Implementation Group 1999)—where propositions are expressed as well-formed formulas, or wffs.

Propositions added to the knowledge base by the user are called hypotheses. Propositions derived from those existing in the belief space are called derived propositions. The system records a justification for each proposition (whether it is a hypothesis or derived) by associating it with an *origin set* consisting of the hypotheses used in its derivation. An origin set for a belief is a set of hypotheses which is known to *minimally derive* that belief—i.e. no subset of a belief's origin set is known to derive that belief.

This is along the style of an ATMS, short for "assumption-based" truth maintenance system —a term "introduced by (de Kleer 1984; 1986), although similar ideas had been investigated earlier by (Martins & Shapiro 1983)." (Martins 1990) also mentions that (Sandewall 1967) presented the "first description of as ATMS-like system."

A hypothesis has a singleton origin set, containing only itself, but might also be derivable from other hypotheses. Multiple derivations of a single proposition can result in its having multiple origin sets. If the hypotheses in a proposition's origin set are asserted, or believed, then the proposition is also believed and is part of the belief space. This situates SNePS firmly on the foundations side of the coherence/foundations belief revision divide, but a coherence approach can be simulated by additionally asserting each derived belief as a hypothesis.

In the SNePS terminology, the belief space is the set of believed propositions—both hypotheses and derived beliefs—which are supported by the current context. The current context is, intensionally, a named structure that contains a set of hypotheses. That set is the extensional context. When a new hypothesis is added, the intensional context now contains a different extensional set of hypotheses. When we refer to adding and removing hypotheses from "the context", we are referring to the intensional context.

Unlike theoretical knowledge bases, implemented ones cannot promise deductive closure for a knowledge base, because of the space and time limitations of the real world. SNePS attempts to derive propositions as they are asked for—either by the user or by the system as it performs backward chaining or forward inference. This is typical of a DOBS as it is formalized in (Johnson & Shapiro 2000).

SNeBR The SNePS belief revision sub-system, SNeBR (Martins & Shapiro 1988), is activated when a derived proposition or a hypothesis is added to the belief space, *and* it explicitly contradicts a pre-existing belief. The following are examples of explicit contradictions in SNePS:

- `P` and `~P`
- `P` and `~(P` $\vee$ `Q)`

but *not*

- `Q` and `Q=>P` and `~P`

because, in this last case, `P` is only an implicit belief and must be derived before that contradiction can be detected.

The detection of an explicit contradiction is almost instantaneous, even in a knowledge base with thousands of nodes, due to the Uniqueness Principle (Maida & Shapiro 1982), which states that no two SNePS terms denote the same entity.

> [Therefore an] explicit contradiction ... in the belief space, is easily recognized by the system because ... the data structure representing `P` is directly pointed to by the negation operator in the data structure representing `~P`. (Shapiro & Johnson 2000)

The user has the option of letting the belief space remain inconsistent or activating a manual version of belief revision to restore consistency. The latter is performed by forming a minimally-inconsistent set of hypotheses, which can be made consistent upon the removal of any one of its members. This set is the union of the origin sets for the contradicting beliefs—multiple origin sets for a belief result in multiple inconsistent sets to be revised. For example: If the contradictory propositions `P` and `~P` had one (α) and two (β and γ) origin sets respectively, then there would be two minimally-inconsistent sets formed for belief revision: ($\alpha \cup \beta$) and ($\alpha \cup \gamma$).

After forming the inconsistent sets, SNeBR prompts the user to remove at least one proposition from each set to restore consistency. It is up to the user to decide which beliefs should be removed (retracted, become unasserted).

How SNeBR Adheres to Constraints 1 and 2

Adherence to Constraint 1 Because a SNePS belief space is a DOBS, it can only claim consistency in terms of not knowing of any contradictions. Immediately upon discovery of a contradiction, however, the system activates SNeBR to restore consistency. Therefore, the user always has the option to maintain consistency "whenever possible".

Adherence to Constraint 2 Although SNePS cannot promise that all derivable beliefs are in the knowledge base, it does derive a proposition upon query if that proposition is derivable from the existing knowledge base. Therefore, SNePS follows an altered version of constraint 2: If a proposition can be derived from the beliefs in the knowledge base, then the system will produce it if it is asked for.

Previous Attempts at Ordering Beliefs and AutoBR

The researchers described in this section chose to order beliefs based on relative credibility *as determined by the user*—i.e. the user decides which beliefs (or types of beliefs) are more credible than others. This not the only way to order beliefs as "more important or entrenched," but, due to our interest in information fusion, this epistemic entrenchment is the way we, also, have chosen to order our beliefs. To this end, any reference in this paper to orderings of beliefs or sources should be assumed to mean ordering based on credibility (unless otherwise stated). This ordering was then used in the implementation of automatic belief revision (autoBR) which allowed the systems described to perform belief revision without user interaction.

Cravo and Martins (Cravo & Martins 1993) introduced an altered version of SNePS, called SNePSwD (SNePS with Defaults), that incorporated default reasoning. It also offered automatic belief revision based on ordering beliefs by credibility and specificity. The system allowed the user not only to order beliefs but to also order the orders. Ordering large amounts of information was tedious, however, and any new additions required updating relevant orderings.

Ehrlich (Ehrlich 1995; Ehrlich & Rapaport 1997; Rapaport & Ehrlich 2000) altered a version of SNePSwD that had this ordering capability. She chose to eliminate the laborious hand ordering by defining a predetermined group of "knowledge categories," with a preset ordering. The category for a proposition was added as another argument of the proposition. She then ordered all her propositions based on the relative order of their knowledge categories. Although this saved her the manual ordering, there were several drawbacks: (a) she had to predetermine the knowledge categories that would be used, (b) no new ones could be added, and (c) the ordering hierarchy of the categories was fixed.

Both Cravo and Martins and Ehrlich developed their automatic belief revision processes (autoBR) to remove hypotheses based on credibility orderings—adhering to constraint 4. How to properly combine and weight both constraints 3 and 4 remains an open topic in belief revision. Our initial attempt to combine and weight them is detailed later.

Alterations to Aid Adherence to Constraint 3

To minimize information loss, the total number of beliefs removed from the system during revision must be considered. Removal of some belief, `P`, will also remove any propositions that have `P` in all of their currently active origin sets. The revised SNeBR system orders the hypotheses in the inconsistent sets based on the number of derived propositions that they support.

It is also possible to make multiple inconsistent sets consistent with a single retraction in the case that a hypothesis common to those sets is the one chosen for removal. The revised system creates an ordered list of the hypotheses based on how many of the minimally-inconsistent sets they are in.

From these two orderings, two lists are formed: (a) the hypotheses supporting the fewest number of derived propositions, and (b) the hypotheses common to the largest number of inconsistent sets. These two lists are now considered during belief revision in the interest of adherence to constraint 3.

Unlike belief spaces created by deductive closure, our DOBS system builds beliefs as they are queried about, thus we can consider those beliefs to be of high interest to the user. This somewhat validates the connection between the cardinality of a belief set and the information it holds. Other researchers (Sims & Kucera 1998) also choose culprits based on set cardinality combined with credibility issues.

Our current approach to minimizing information loss involves counting believed propositions without regard to their internal form. For example, `A` and `B∧C∧D` are each counted as one proposition, as are both `Q(a)` and `all(x)(P(x) => Q(x))`, even thought the latter one of each pair clearly "encodes" more information. Especially in the context of a DOBS, it would be important to try to assess the usefulness of a proposition in terms of the number of other propositions that might in the future be derived from it. We leave this assessment for future work.

Epistemic Entrenchment Additions to SNeBR

Sources and their information

As mentioned earlier, Ehrlich's knowledge categories have to be determined and ordered off-line before running the system. This, plus the source information being included in each proposition as additional argument, forced a static treatment and implementation. Our revised SNePS system uses meta-propositions to assign sources, allowing dynamic source addition and ordering.

Problems with Source Information as an Added Argument Representing the source information as an additional argument of the predicates has several problems associated with it. For example, if "Fran is smart." were represented as `Smart(Fran)`, "The prof says that Fran is smart." could be represented as `Smart(Fran, Prof)`, and the problems are:

1. The source of `Smart(Fran, Prof)` cannot be removed or changed without also removing or changing the belief that Fran is smart. Although that might immediately be reintroduced with `Smart(Fran, Nerd)`, belief revision may have had to be performed in the interim, wasting time and effort.
2. The proposition `~Smart(Fran, Sexist)` might represent either the belief that the sexist is the source of the information that Fran is not smart or the belief that the sexist is not the source of the information that Fran is smart. No matter which one it does represent, there is no obvious way to represent the other.
3. It is not clear how to ascribe a source to a rule, such as `all(x)(Grad(x) => Smart(x))`.

Benefits of Source Information in a Meta-Proposition Representing the source information in a meta-proposition, is to represent it as a belief about the belief. For example, "The prof says that Fran is smart." would be represented as `Source(Prof, Smart(Fran))`.[1] This solves the three problems cited above:

1. The source of the belief that Fran is smart can be removed or changed, without removing or changing the belief that Fran is smart, by removing `Source(Prof, Smart(Fran))` without removing `Smart(Fran)`, and then, perhaps, introducing a different source, e.g. `Source(Nerd, Smart(Fran))`.
2. The belief that the sexist is the source of the information that Fran is not smart would be represented as `Source(Sexist, ~Smart(Fran))`, whereas the belief that the sexist is not the source of the information that Fran is smart would be represented as `~Source(Sexist, Smart(Fran))`.
3. The belief that the prof is the source of the rule that all grads are smart would be represented by `Source(Prof, all(x)(Grad(x) => Smart(x)))`.

As shown, this allows dynamic interaction, where the user can add, remove, and change source information about a proposition while the system is running and without affecting or entirely rewriting the actual proposition. Source orderings are also stored as propositions, such as `Greater(Prof,Nerd)`, which can be interacted with and reasoned about dynamically. New sources as well as their credibility orderings can be added at any time to the knowledge base. Propositions can even have more than one source, although SNeBR assumes a single source at this time.

[1] This is syntactically and semantically correct, because propositions like `Smart(Fran)` are, in SNePS, functional terms denoting propositions (Shapiro 1993).

Ordering Propositions (Epistemic Entrenchment) and Sources

Our system currently depends on the user to determine the credibility of sources or beliefs directly. The user inputs information to the system declaring source credibility orders and/or belief credibility orders. These are partial orders that are qualitative and transitive.

The system currently assumes beliefs have, at most, one source, but future research will explore multiple source situations and their implementation. We are also assuming at this time that a more credible source delivers more credible information. E.g. Given the following:

- Lisa is more credible than Bart.
- Lisa tells us, "It is snowing."
- Bart tells us, "Homer is fat."

we consider "It is snowing" more credible than "Homer is fat."

Since ordering can never be assumed complete or unchangeable, the system works with what it has—including propositions whose sources are unknown (assumed at this time to be more credible than beliefs that have recorded sources). An interesting issue to explore in the future would be to have the system dynamically establishing and adjusting source credibility information based on revision experiences.

Recommendations and Automatic Belief Revision

The user can set the belief revision mode at any time from the top level of the SNePSLOG interface. The two modes are:

manual offers recommendations, but requires the user to revise (this is the default)

auto activates automatic belief revision, autoBR

From the union of all the inconsistent sets underlying the contradiction, SNeBR produces three lists that are used to create a recommended culprit list:

LB the least believed hypotheses

MC the hypotheses that are the most common (to the largest number of inconsistent sets)

FS the hypotheses that support the fewest beliefs in the knowledge base.

The first set provides possible culprits that support constraint 4 (remove the least important or least credible beliefs). Both the second and the third sets will provide possible culprits that support constraint 3 (minimal loss of information during belief revision). The culprit list is created by combining these lists using a method described in the next section.

Once the recommended culprit list is formed, the user is notified of the three lists as well as the culprit list. In manual mode, the user must then decide (with the help of those lists) which hypotheses to remove from the context. In auto mode, the system will perform an automatic retraction if the culprit list contains a single hypothesis. Otherwise it reverts to manual. In either case, any unresolved inconsistencies are dealt with manually.

Culprit List Algorithm

Processing the three lists above to create a culprit list (CL) results in the smallest, non-empty intersection: $CL = Min-not-\emptyset((LB \cap MC \cap FS), (LB \cap MC), (LB \cap FS), (MC \cap FS), LB, MC, FS)$, where $Min-not-\emptyset$ is a function that chooses the smallest non-empty set from a list of sets in decreasing order of importance (for tie-breaking purpose—e.g. choose LB over MC).

Other factors being equal, removing a belief with low credibility (constraint 4) is currently preferred over one whose removal does the least damage (constraint 3), because credibility orders are not weakened by the absence of deductive closure. Regarding the information used to support constraint 3, the system prefers to remove a more common belief over one with the fewest supported nodes, because that will protect beliefs that have been derived in more ways. For example, given `A, B, D, A->C, B->C, and D->~C`, and the derivations of `C` (both ways) and `~C`, the minimally inconsistent sets would be `{A, A->C, D, D->~C}` and `{B, B->C, D, D->~C}` and both `D` and `D->~C` would be the most common hypotheses. Removal of either results in the loss of `~C` and the retention of `C`, which was derived two ways.

It should be emphasized, however, that the final determination of CL is the smallest, non-empty set found. In this sense, condition 3 regains some of its lost status. For example: if $(LB \cap FS)$ contained three hypotheses and $(MC \cap FS)$ contained two, the latter would be chosen over the former, even though only the former contained credibility information.

An edited sample of the autoBR output for a belief revision exercise is available in Appendix 1.

Conclusion and Future Work

By considering the work of coherence and theoretical researchers, we were able to improve our existing belief revision system by adding information essential to improving culprit selection during revision. The revised system combines the constraints of minimal information loss and maximal credibility in its development of a culprit list during belief revision to return consistency to a knowledge base. It considers (a) the number of hypotheses and derived propositions that will be affected by the revision as well as (b) the relative credibility orderings of the hypotheses under consideration. These orderings are determined by both source credibility orderings and credibility orders directly assigned between hypotheses. All source and credibility information can be retracted, altered, added to, and reasoned about while the system is running.

Automatic belief revision is possible when the culprit list contains a single hypothesis. In this case, the system appears to the user to be non-monotonic: i.e. the user can add a proposition to an existing context, but end up with a belief space that is not a superset of the original belief space.

The issue of incorporating source information and credibility ordering into a knowledge base is key to maintaining the credibility of an information fusion system. Selection of a well-believed hypothesis as the culprit (due to other considerations like minimizing damage to the knowledge base) might also indicate a need to re-evaluate the reliability of its source. This might lead to development of a system that dynamically adjusts source and propositional credibility orders based on past performance.

Work for the immediate future will include dealing with (1) a proposition having multiple sources and (2) revising the inconsistent sets as a group. For the latter, we might first partition the inconsistent sets based on which of the contradictory nodes each hypothesis supports (or if it supports both), then analyzing the groups by their inconsistent set as well as by their partition. Then the system could better determine which of the contradictory nodes should be contracted and remove its supports efficiently. We hope that one result will be an improvement on safe contraction (Alchourrón & Makinson 1985). For example: If retracting a proposition `P` with the two origin sets of `{A,B,C}` and `{B,D}` ordered in increasing credibility, then retracting the least-believed in each set (as per safe-contraction) would remove both `A` and `B`, when removal of `B` would be sufficient. This improvement might show up as the union of the least believed and the most common sets.

Acknowledgements

The authors appreciate the insights and feedback of Bill Rapaport, Haythem Ismail, and the SNePS Research Group.

Appendix 1

Below is a description of the information given to the knowledge base. `WFF22` is added last with forward inferencing, which produces the contradictory propositions that trigger SNeBR. The five sources with their credibility orderings are:

$Holybook > Prof$	WFF1:	GREATER(HOLYBOOK,PROF)
$Prof > Nerd$	WFF2:	GREATER(PROF,NERD)
$Nerd > Sexist$	WFF3:	GREATER(NERD,SEXIST)
$Fran > Nerd$	WFF4:	GREATER(FRAN,NERD)

Source: statement

Nerd: Jocks aren't smart.
```
WFF10:  all(X)(JOCK(X) => (~SMART(X)))
WFF11:  SOURCE(NERD,WFF10)
```

Sexist: Females aren't smart.
```
WFF12:  all(X)(FEMALE(X) => (~SMART(X)))
WFF13:  SOURCE(SEXIST,WFF12)
```

Prof: Grads are smart.
```
WFF14:  all(X)(GRAD(X) => SMART(X))
WFF15:  SOURCE(PROF,WFF14)
```

HolyBook: Old people are smart.
```
WFF16:  all(X)(OLD(X) => SMART(X))
WFF17:  SOURCE(HOLYBOOK,WFF16)
```

Fran: I'm an old, female, jock who's a grad.
```
WFF22:  FEMALE(FRAN) and OLD(FRAN) and
GRAD(FRAN) and JOCK(FRAN)
WFF23:  SOURCE(FRAN,WFF22)
```

The following code is an edited version of the system output showing the inconsistencies found and the hypotheses removed through autoBR. Author's comments are in *italics.*

After all the information is in, one contradiction is detected:
```
 The contradiction involves the newly
derived proposition:

     WFF24:  SMART(FRAN)

and the previously existing proposition:

     WFF25:  ~SMART(FRAN)
```

To resolve the contradiction, SNePS Belief Revision (SNeBR) analyzes the inconsistent set formed by the union of the two Origin Sets for the contradictory propositions:

$$(WFF16, WFF22) \cup (WFF12, WFF22) = (WFF22, WFF16, WFF12)$$

The three sets aiding culprit selection:
```
The least believed hypothesis:
   (WFF12)
The most common hypotheses:
   (WFF22 WFF16 WFF12)
The hypotheses supporting the fewest nodes:
   (WFF12 WFF16)
```

The system informs the user of its decision:
```
I will remove the following node:
   WFF12:  all(X)(FEMALE(X) => (~SMART(X)))
```

The system continues reasoning and discovers the same contradiction (derived in new ways):
```
 The contradiction involves the newly
derived proposition:

     WFF25:  ~SMART(FRAN)

 and the previously existing proposition:

     WFF24:  SMART(FRAN)
```

There are two known-to-be-inconsistent sets in the context, now:

```
The following sets are known to be
inconsistent.  To make the context
consistent, remove at least one hypothesis
from each of the sets:
     (WFF22 WFF16 WFF10) (WFF22 WFF14 WFF10)
```

The three sets aiding culprit selection:

```
The least believed hypothesis:
   (WFF10)
The most common hypotheses:
   (WFF22 WFF10)
The hypotheses supporting the fewest nodes:
   (WFF14 WFF10)
```

The system's decision in this case:

```
I will remove the following node:
   WFF10:  all(X)(JOCK(X) => (~SMART(X)))
```

References

Alchourrón, C. E., and Makinson, D. 1985. On the logic of theory change: Safe contraction. *Studia Logica* 44:405–422.

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.

Brachman, R. J., and Levesque, H. J., eds. 1985. *Readings in Knowledge Representation.* 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. 1984. Choices without backtracking. In *Proceedings of the Fourth National Conference on AI*, 79–85. Menlo Park,CA: AAAI.

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

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*, 205–210. Mahwah, NJ: Lawrence Erlbaum Associates.

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.

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

Hansson, S. O. 1993a. Reversing the levi identity. *Journal of Philosophical Logic* 22:637–669.

Hansson, S. O. 1993b. Thoery contraction and base contraction unified. *Journal of Symbolic Logic* 58:602–625.

Johnson, F. L., and Shapiro, S. C. 2000. Formalizing a deductively open belief space. Technical Report 2000-02, Department of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY.

Lehmann, F., ed. 1992. *Semantic Networks in Artificial Intelligence.* Oxford: Pergamon Press.

Maida, A. S., and Shapiro, S. C. 1982. Intensional concepts in propositional semantic networks. *Cognitive Science* 6(4):291–330. Reprinted in (Brachman & Levesque 1985, pp. 170–189).

Martins, J. P., and Shapiro, S. C. 1983. Reasoning in multiple belief spaces. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* San Mateo, CA: Morgan Kaufmann. 370–373.

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

Martins, J. P. 1990. 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.

Mckay, D. P., and Martins, J. 1981. SNePSLOG user's manual. SNeRG Technical Note 4, Department of Computer Science, SUNY at Buffalo.

Nebel, B. 1989. A knowledge level analysis of belief revision. In Brachman, R. J.; Levesque, H. J.; and Reiter, R., eds., *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89)*, 301–311.

Rapaport, W. J., and Ehrlich, K. 2000. A computational theory of vocabulary acquisition. In Iwańska, Ł. M., and Shapiro, S. C., eds., *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language.* Menlo Park, CA: AAAI Press/The MIT Press. In press.

Sandewall, E. 1967. Ambiguity logic as a basis for an incremental computer. Technical Report 9, Department of Computer Sciences, Uppsala University, Uppsala, Sweden.

Shapiro, S. C., and Johnson, F. L. 2000. Automatic belief revision in sneps. In *Papers from the System Descriptions and Demonstration Track of NMR2000.* In press.

Shapiro, S. C., and Rapaport, W. J. 1992. The SNePS family. *Computers & Mathematics with Applications* 23(2–5):243–275. Reprinted in (Lehmann 1992, pp. 243–275).

Shapiro, S. C., and The SNePS Implementation Group. 1999. *SNePS 2.5 User's Manual.* Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY.

Shapiro, S. C. 1993. Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* 5(2&3):225–235.

Sims, A., and Kucera, S. 1998. Recovering a consistent theory from an inconsistent belief set: Theory extraction. In *Papers from the Belief Change Workshop at NMR98*.