Says Who? – Incorporating Source Credibility Issues into Belief Revision[1]

CSE Technical Report 99-08

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
Buffalo, NY 14260
{flj|shapiro}@cse.buffalo.edu

July 31, 1999

Abstract

We discuss some belief revision theories and the implementation of some of them in SNeBR, the belief revision subsystem of the SNePS knowledge representation and reasoning system. The following guidelines are important to belief revision:

1. minimize information removal
2. retract less-plausible beliefs before more-plausible ones.

Alterations to SNeBR that provide users with information to help them follow these guidelines include the incorporation of sources into the knowledge base, as well as ordering of both sources and individual propositions, allowing partially ordered propositional epistemic entrenchment. We also developed an automatic belief revision option that performs the retraction of a belief if one culprit can be singled out based on the guidelines above.

1. 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 an inconsistent knowledge base to restore consistency. This is especially important to information fusion, where information is combined from multiple sources which might contradict each other.

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

We explore some belief revision theories and the implementation of some of them into the SNePS knowledge representation and reasoning system used at UB (Shapiro and the SNePS Implementation Group, 1999; Shapiro and Rapaport, 1987; Shapiro and Rapaport, 1992). SNePS Belief Revision (SNeBR) is triggered when the system perceives the knowledge base to be inconsistent. It is possible for the system to be inconsistent, but not yet be aware of that fact; however, the underlying logic allows this without causing any runaway logical inferences (Martins and Shapiro, 1988, sec.2; Shapiro, 1992). Therefore, for the purposes of this paper, comments regarding the consistency of sets, contexts, or knowledge bases should include the understanding that in practice (as opposed to theory) "consistent" means *not known to be inconsistent.*

The following guidelines are important to belief revision (Alchourron, Gärdenfors, and Makinson, 1985; Gärdenfors and Rott, 1995):

1. minimize information removal
2. retract less-plausible beliefs before more-plausible ones.

How to properly combine these guidelines is not clear, since they can often seem to conflict.: E.g., how should one resolve a contradiction that would require removal of one well-believed hypothesis or three less-plausible ones? This issue of culprit selection remains an open topic in the area of belief revision. This literature and some implementations using epistemic entrenchment are discussed in section 2.

The initial implementation of these guidelines consisted of giving the user information helpful in minimizing the amount of information lost during belief revision. We also made some improvements to the SNeBR algorithms and interface at the same time. These changes have been included in the newest version of SNePS (SNePS 2.5) and are discussed briefly in section 3.

Section 4 describes the more recent changes. These include the incorporation of source information and ordering options. Ordering sources by their reliability so that beliefs can be ordered and revised based on *their sources'* reliability is an issue central to information fusion, where observations from multiple sources are combined and contradictions arise. Beliefs can also be ordered directly. The resulting epistemic entrenchment (ordering by credibility) of the propositions is used by the automatic belief revision system discussed in section 4.3.

Section 5 discusses future work and issues to consider as we explore belief revision for information fusion.

2. Background

2.1 Coherence vs. Foundations Theories

One of the main differences in the literature on belief revision is whether a theory requires that justifications be maintained for the beliefs held in the knowledge base. Foundations theory states that justifications ***should*** be maintained – requiring that all believed propositions must be justified; and, conversely, those that lose their justification should no longer be believed. By contrast, coherence theory focuses on whether the belief space is consistent – i.e. whether a belief coheres with the other beliefs in the current belief space without regard to its justification.

In other words, the Foundation Theory supports retraction of a belief if there is no longer any reason to believe it, whereas Coherence Theory only retracts a belief when given a reason to specifically disbelieve it. (See Doyle (1992) for a more extensive discussion of the distinctions between these two approaches.)

2.2 Basic Constraints for Belief Revision (Coherence Approach)

Gärdenfors and Rott (1995) discuss postulates for belief revision using a coherence approach. These postulates (first presented by Alchourron, Gärdenfors and 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.

Regarding constraint 1, SNePS always alerts the user when an inconsistency is found and (although the user might opt to continue reasoning in the inconsistent context) offers an opportunity to regain consistency through revision. Inconsistency is dealt with only when it is discovered (during the addition of new beliefs or inference); and the system can reason even when the knowledge base is inconsistent (Martins and Shapiro, 1988, sec.2; Shapiro, 1992). However, the user always has the option to maintain consistency "whenever possible".

Regarding constraint 2, SNePS propositions are derived when needed or asked for. No implemented system can *ever* promise that *all* propositions that *can* be derived *are* present in the network, because there are an *infinite* number of them.

This paper focuses on the incorporation of constraints 3 and 4 into SNeBR. With the activation of automatic belief revision, constraint 1 will also be supported, as the automatic mode assumes the user wants to perform revision to preserve consistency *whenever* a contradiction is detected.

SNePS 2.4 supported ranking hypotheses based on the number of propositions they supported with minimal additional code. In the case of a contradiction, the system also could determine the minimal origin set(s) of hypotheses that underlie the inconsistency (ATMS style) and, therefore, could order these hypotheses by how many of the inconsistent sets they are in. These two orderings can be used in the interest of adhering to constraint 3 (as best as possible in an implemented system[2]). Constraint 4 required new concepts.

2.3 Research on Ordering (from Foundations Approaches)

Cravo and Martins (1993) introduced an altered version of SNePS, called SNePSwD (SNePS with Defaults), that incorporates default reasoning. It also offers automatic belief

[2] Recall that the information is only complete regarding the propositions that the system is currently aware of and believes (hypotheses and 'limited' inferences), not all the propositions in the logical closure of the knowledge base.

revision based on ordered wffs and specificity. The system allows the user not only to order wffs but to also order the orders. Ordering large amounts of information is tedious, however, and any new additions require updating relevant orderings.

Ehrlich (Ehrlich, 1995, Ehrlich and Rapaport, 1997; Rapaport and Ehrlich, 2000) improved a version of SNePSwD that had this ordering capability. She chose to eliminate the laborious hand ordering by coding "knowledge categories" into the structural information of her propositions. She then used the SNePSwD code to order all her propositions based on their knowledge categories and a *hard-coded hierarchy*. Although this saved her the hand ordering, there were several drawbacks. She had to predetermine the knowledge categories that would be used, no new ones could be added, and the hierarchy of the categories was fixed.

The orderings used by Cravo and Martins and by Ehrlich in their automated belief revisions use constraint 4 as their sole source of culprit selection. How to properly combine and weight *both* constraints 3 *and* 4 remains an open topic in belief revision. Our initial attempt at combining these offers a more dynamic interaction with the system and is detailed in section 4.3, below.

3. Changes to SNeBR

The alterations to the SNeBR code falls into three categories:

1. interface changes;
2. algorithm alterations (improvements to the existing functions and adding information);
3. source-related functions, including ordering sources by credibility and passing that ordering on to their beliefs (detailed in section 4).

3.1 Interface Changes

When a contradiction is detected and the user chooses to revise the belief space, the system shows all known minimally inconsistent sets that underlie the contradiction. A minimally inconsistent set can be made consistent by removing any one of its hypotheses. To make the knowledge base consistent (or no longer "known to be inconsistent"), at least one hypothesis from each set must be removed. Although each set is dealt with individually, the new interface first alerts the user to any hypotheses common to more than one set, since removing one of those will make more than one set consistent.

3.1.1 Changes implemented in the new release of SNePS 2.5

The original interface in SNeBR consisted of cycling through the hypotheses of an inconsistent set and asking the user what they wished to do with each hypothesis: keep, discard, or undecided – the last choice returning the hypothesis to the end of the cycle to be considered later. The new interface allows more direct access to the hypotheses and offers more information in the initial windows. It begins its set revision with a warning if any hypothesis in the set is common to both contradicting beliefs. This is because a removal of this hypothesis would remove both beliefs and would probably be considered too extreme.

The interface then offers:

1. all the hypotheses of that set in a numbered list, with menu options allowing direct access to any hypothesis,
2. a heading statement declaring the list of hypotheses to be either consistent or inconsistent depending on the revisions,
3. a full description of the hypotheses listed – with the number of, and a list of, supported wffs,
4. access to hypotheses removed from the set as well as the option to see the entire context, and
5. a quit option on every screen that allows exiting from this set revision.

Here is an example of this manual belief revision interface:[3]

```
This set of hypotheses is known to be inconsistent:

1:  WFF22:  FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)
      (7 supported propositions:
            (WFF25 WFF24 WFF22 WFF21 WFF20 WFF19 WFF18) )
2:  WFF16:  all(X)(OLD(X) => SMART(X))
      (2 supported propositions: (WFF24 WFF16) )
3:  WFF12:  all(X)(FEMALE(X) => (~SMART(X)))
      (2 supported propositions: (WFF25 WFF12) )

Enter the list number of a hypothesis to examine or
   [d] to discard some hypothesis from this list,
   [a] to see ALL the hypotheses in the full context,
   [r] to see what you have already removed,
   [q] to quit revising this set, or
   [i] for instructions

(please type a number OR d, a, r, q or i)
```

This interface change results in the user being able to make more informed decisions using fewer keystrokes,[4] which will result in quicker and (hopefully) wiser revisions.

3.1.2 Changes since SNePS 2.5

When choosing from the menu to print the entire context, the user is now warned that the context might be large and is given the number of hypotheses contained in it. The user is then given the option to cancel the request.

The message about hypotheses common to more than one inconsistent set has been improved to include the expanded description of each hypothesis, as well as the number of inconsistent sets it is included in. This is shown in the example below:

```
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 hypotheses listed below are included in more than one set. Removing one
of these will make more than one set consistent.
```

[3] An edited sample from the manual BR version of the demonstration in the Appendix -- most samples will be from the same.

[4] The demonstration that was being used to test the messages inserted showed a keystroke reduction from 36 to 15.

```
Common to 2 sets:
WFF22: FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)
WFF10:  all(X)(JOCK(X) => (~SMART(X)))
```

This last bit of information is also used to determine the most common hypotheses – to aid in culprit selection – keeping in mind that the information might be incomplete, since we can only deal with the inconsistent sets *that the system is aware of*. Likewise, when using the number of supported nodes for each hypothesis to select a possible culprit – that with the fewest nodes supported – the user also must consider that the information comes from the number of supported nodes *that the system is aware of*. I.e. we are not dealing with the theory and the logical closure of the knowledge base: implemented systems contain incomplete knowledge.

3.2 Algorithm Alterations

3.2.1 Completed and Implemented in SNePS 2.5

As described above, there are two added information messages to aid the user in choosing a hypothesis (culprit) for retraction:

1. a message alerting the user to any hypotheses common to more than one inconsistent set (refer to the example above to see this message in its improved version)
2. a message alerting the user if any hypotheses in an inconsistent set are common to *both* contradictory beliefs – since removing any such a hypotheses might be considered too extreme a revision:

```
WARNING: the following hypothesis supports BOTH contradictory WFFS,
so removing it might be too extreme a revision.
WFF22:  FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)
```

3.2.2 Changes Since SNePS 2.5

Since the SNePS 2.5 release, automatic belief revision has been implemented in SNePS. Users get notified about the three sets of propositions to be combined to produce a culprit list. This list can be used in automatic-belief-revision mode to select a culprit, or as a recommendation in manual mode. An example of the three sets and the system's recommendation is:

```
The least believed hypothesis:
(WFF10)
The most common hypotheses:
(WFF22 WFF10)
The hypotheses supporting the fewest nodes:
(WFF14 WFF10)
You are advised to remove from the following list of hypotheses
(WFF10)
```

How the set of least believed hypotheses is determined is explained in section 4 below.

4. Epistemic Entrenchment Additions to SNeBR

4.1 Sources and their information

Ehrlich's knowledge categories were hard-coded (determined off-line) and included structurally into the proposition, forcing a static treatment and implementation. Assertional source information allows dynamic interaction, where the user can add, remove, and change source information without affecting or entirely rewriting the actual proposition. New sources can also be added to the knowledge base. Propositions can even have more than one source, though our initial code assumes a single source at this time. This source information serves the same purpose as Ehrlich's knowledge categories by allowing groups of hypotheses to be ordered automatically. For further explanation of structural vs. assertional information, see Appendix 2.

4.2 Ordering propositions (epistemic entrenchment) and sources

Propositions and sources can be ordered directly by the user. This allows the system to determine which hypotheses are more reliable (more believed, more epistemically entrenched) than others. It is assumed that the hypotheses of a more credible source are more entrenched than those of a less credible source. Orderings are qualitative and transitive. Soon to be added are the restrictions that orderings cannot be either symmetric or reflexive. This ordering information, like the source information, is stored in the knowledge base assertionally.

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). At this point, the user determines the credibility of the sources, although an interesting issue to explore in the future would be the system dynamically establishing and adjusting source credibility information (see section 5).

4.3 Recommendations and Automatic Belief Revision

The user can set the belief revision mode at any time from the top level of the SNePSLOG interface – an interface to SNePS which allows the user to input propositions in a style that uses "predicate calculus augmented with SNePS logical connectives."(McKay and Martins, 1981) The commands are:

- manual — offers recommendations, but requires the user to revise (this is the default)
- auto — activates automatic belief revision, autoBR

The automatic belief revision option will perform the retraction of a belief if one culprit is singled out. If the system cannot reduce its removal choices to a single hypothesis, it reverts to the manual mode and gives a list of hypotheses that are to be considered as the most likely culprits recommended for removal. This culprit list is a combination of the following sets:

1. the least believed hypotheses
2. the most common hypotheses (underlying multiple inconsistencies)
3. the hypotheses that support the fewest beliefs in the knowledge base.

The first set provides possible culprits that support constraint 4 (remove least important or credible beliefs). Both the second and the third sets will provide possible culprits that support constraint 3 (minimal loss of information during belief revision). [5] In manual mode, the user must revise manually after the advice is given. If there are multiple inconsistent sets, the autoBR culprit choice might not make *all* of them consistent (i.e. the chosen culprit might not be present in every inconsistent set). Any remaining inconsistent sets get revised manually, although future work will include automatically revising the remaining sets as a group (remembering the earlier retraction as well as avoiding the elimination of *both* contradictory propositions).

If autoBR can narrow down its culprit choice to a singleton hypothesis, it removes that culprit from the context after notifying the user with the following message:[6]

```
I will remove the following node:

   WFF12:  all(X)(FEMALE(X) => (~SMART(X)))
   {<HYP,{WFF12},{{WFF16,WFF22}}>}

   because it is the best choice given the previous information.
```

From the intersection of all the inconsistent sets underlying the contradiction, three subsets are found, one for each one of the numbered guidelines above:

LB = least believed hypotheses

MC = most common hypotheses

FS = hypotheses with the fewest supported nodes.

Hypotheses in LB are least believed. If one of these is removed, constraint 4 is being followed (by removing the least important or credible beliefs first).

MC contains those hypotheses that are common to more than one inconsistent set. Removing one of these would make *more* than one set consistent (rather than removing multiple hypotheses – one from one set, a different one from the next, and so on). This would partially support constraint 3 (minimizing damage).

FS contains those hypotheses least used in the derivation of other nodes. Removing one of these also partially supports constraint 3.

The user is informed of the contents of each of these sets after which they are combined to form the recommended culprit list (CL). This combination is the smallest, non-empty intersection set (or partial intersection set):

$$CL = \mathit{Min\text{-}not\text{-}}\varnothing((LB \cap MC \cap FS), (LB \cap MC), (LB \cap FS), (MC \cap FS), LB, MC, FS),$$

where *Min-not-*$\varnothing$ is a function that chooses the smallest non-empty set from a list of sets where the order of the arguments indicates the tie-breaking choice (e.g. choose LB over MC).

In automatic belief revision, if CL contains a single hypothesis, then that hypothesis is removed from the knowledge base, and the user is notified; otherwise, a recommendation is made for the user to remove one or more hypotheses from CL and manual revision is required.

[5] Recall that the information is only complete regarding the propositions that the system currently believes (hypotheses and limited inferences), not all the propositions in the logical closure of the knowledge base.

[6] See the demonstration run in Appendix I

In manual mode, the recommendation is given even if CL is a singleton set (see example in section 3.2.2).

Other factors being equal, *low credibility* (constraint 4) is currently considered a more important factor than *least damage* (constraint 3). Regarding the information used to support constraint 3, *most common* is favored over *fewest supported nodes*, since both use incomplete data, but the former "kills two birds with one stone" (resolving more than one inconsistent set with one retraction). The incomplete data comment refers to the system not deriving all possible propositions, which results in (a) the inconsistent sets not necessarily reflecting all possible inconsistent sets that might underlie the contradiction, and (b) the list of supported nodes possibly being incomplete. Even if forward inference were called with every assertion, the first detected contradiction triggers SNeBR, which might happen before all possible propositions are derived or before all contradictions (or inconsistent sets) are detected.

It should be emphasized, however, that the final determination of CL is the *smallest*, non-empty nodeset found. In this sense, condition 3 regains some of its lost status.

> Example: If (LB ∩ FS) contained 3 hypotheses and (MC ∩ FS) contained 2, the latter would be chosen over the former, even though only the former contained credibility information.

5. Conclusion and Future Work

The issue of incorporating source information and credibility ordering into a knowledge base is key to information fusion. By using this information to order the hypotheses by credibility, an automatic belief revision system can consider less reliable information when it is performing a retraction. Selection of a well-believed hypothesis as the culprit (due to other considerations like minimizing damage to the knowledge base) might indicate a need to verify the reliability of its source.

Future research will include investigating other implemented systems, the techniques used to include this information, and how they deal with the credibility vs. least damage issues. Included in these investigations will be comparisons of these systems (foundations approach) to the coherence theories.

Another issue for future work is the implementation of default reasoning into our current version of SNePS.[7] When this is done, belief revision might also consider weakening a rule over removing it (i.e. reduce it to default status).

One immediate plan is to incorporate orders-of-orders (Cravo and Martins, 1993) and to develop multiple levels of automatic belief revision to follow up on inconsistent sets unaffected by the initial revision. The recommended revisions for these sets should take into account:

- which of the conflicting propositions was removed by the initial revision (avoid removing the other one --- this goal will also be incorporated in the initial revision as well)
- whether these new revisions should cause reconsideration of the earlier ones .

[7] This effort would closely follow the work done by Cravo and Martins (1993) which altered an earlier version of SNePS combined with a major alteration to the underlying logic. We would maintain the underlying logic as it is.

Some of the issues and questions raised as we implemented source credibility into our system are:

1. Dealing with multiple sources for a single proposition. Should a weak three-source belief be more or less credible than a strong single-source belief?
2. Dealing with source nesting: e.g. Bill said that John said that Mary said that…
3. Should the system dynamically adjust source credibility hierarchies based on current track records? i.e. If a source has had its propositions retracted more than any other source, should it become the least believed source? Should manual retraction carry the same (or more, or less) weight as automatic belief revision in altering a sources credibility? What if the manual retraction is being performed by a competing source (see issue 5, below)?
4. Ordering information from a source to allow back-up beliefs to replace retracted former beliefs. This might be helpful when a source can order its query replies. Example: An Image Processor might identify an object as a tank with 90% reliability followed by a ship with 73% reliability. Therefore, it is a tank; but if that proposition is rejected, assert that the object is a ship.
5. How does the system finally decide who's right? An omniscient user can bail the system out regarding whether to believe Bill or Stu, but what happens if all input to the system is from either Bill or Stu, and is tagged with its source? Who decides credibility hierarchies? Should the current user have total control? Then every conflict might be resolved by the user saying, "I'm the one to believe." If the system is given some control (e.g. using sight to determine reliability – Bill told a lie, therefore Stu is more reliable than Bill.), might that lead to the system rejecting input or direction from the less credible source?
6. How should a belief system adjust when source credibility changes? Should it remember previous revisions based on credibility? This brings up the issue of whether credibility is an eternal, permanent, or temporary state.[8] If a source's credibility changes, does that mean that its earlier credibility was wrong *at that time*? If so, previous revisions (and revisions on those revisions?) might need to be altered. If not, do we need to time tag all propositions and inferences?
7. Issue 6, above, awakens another issue - putting algorithms for reasoning about credibility changes into the system, itself, thus allowing the system to develop meta-beliefs. The history of its beliefs *should* affect its future beliefs. One example: If it believed A > B (to be read as "A is more credible than B") followed by a revision to B > A, it might be "easier" for the system to revert to believing A > B (having believed it previously) than if it were the first time to believe that.

6. Acknowledgements

The authors thank Bill Rapaport, Haythem Ismail, and the SNePS Research Group for their input and support.

[8] We thank Haythem Ismail, who discussed these distinctions in his SNePS Research Group (SNeRG) presentations.

Appendix 1

Below is a demonstration showing the features of auto belief revision using source credibility. The demonstration has been edited so that extraneous or duplicated information is not shown. Some formatting liberties have also been taken to improve readability. In some cases the input information is not shown, only the system's representation of that information – that is because the input information was identical to the output but lacking the `WFF#` designations that will be helpful later. Inputs follow the SNePSLOG prompt (the colon- `:`). Author's comments are in ***bold italics***. Some comments that are within the demonstration code have been retained as they appeared for clarity – they follow triple or double semi-colons - `;;;` or `;;`. Anything else is system output.

```
: clearkb
Knowledge Base Cleared
: expert                    ;; show expanded node
: auto                      ;; do auto belief revision when possible

:
;;; source info
;;; Sources:  HolyBook, Professor, Fran, Nerd, Sexist
;;; Source Orders:   HolyBook  >  Prof  >  Nerd  >  Sexist
;;;                                Fran  >  Nerd  >  Sexist
```

Source info added. These are the results:

```
WFF1:  GREATER(HOLYBOOK,PROF)  {<HYP,{WFF1},{}>}
WFF2:  GREATER(PROF,NERD)  {<HYP,{WFF2},{}>}
WFF3:  GREATER(NERD,SEXIST)  {<HYP,{WFF3},{}>}
WFF4:  GREATER(FRAN,NERD)  {<HYP,{WFF4},{}>}
WFF5:  GREATER(FRAN,SEXIST)  {<HYP,{WFF5},{}>}
```

Source ordering is transitive (followed by forward inferencing):

```
: all(x,y,z)({greater(x,y),greater(y,z)} &=> {greater(x,z)})!
```

Resulting in:

```
 WFF9:  GREATER(PROF,SEXIST)  {<DER,{WFF2,WFF3,WFF6},{}>}
 WFF8:  GREATER(HOLYBOOK,SEXIST)  {<DER,{WFF1,WFF2,WFF3,WFF6},{}>}
 WFF7:  GREATER(HOLYBOOK,NERD)  {<DER,{WFF1,WFF2,WFF6},{}>}
 WFF6:  all(Z,Y,X)({GREATER(Y,Z),GREATER(X,Y)} &=> {GREATER(X,Z)})  {<HYP,{WFF6},{}>,<DER,{WFF6},{}>}
```

Sources say:

;;; Nerd: Jocks aren't smart.

```
: all(x) (jock(x) => ~smart(x))            ; wff10
  WFF10:  all(X)(JOCK(X) => (~SMART(X)))  {<HYP,{WFF10},{}>}
: source(nerd,wff10)
  WFF11:  SOURCE(NERD,all(X)(JOCK(X) => (~SMART(X))))  {<HYP,{WFF11},{}>}
```

;;; Sexist: Females aren't smart.

```
: all(x) (female(x) => ~smart(x))          ; wff12
  WFF12:  all(X)(FEMALE(X) => (~SMART(X)))  {<HYP,{WFF12},{}>}
: source(sexist,wff12)
  WFF13:  SOURCE(SEXIST,all(X)(FEMALE(X) => (~SMART(X))))  {<HYP,{WFF13},{}>}
```

;;; Prof: Grads are smart.

```
: all(x) (grad(x) => smart(x))             ; wff14
  WFF14:  all(X)(GRAD(X) => SMART(X))  {<HYP,{WFF14},{}>}
: source(prof,wff14)
  WFF15:  SOURCE(PROF,all(X)(GRAD(X) => SMART(X)))  {<HYP,{WFF15},{}>}
```

;;; HolyBook: Old people are smart.

```
: all(x) (old(x) => smart(x))              ; wff16
  WFF16:  all(X)(OLD(X) => SMART(X))  {<HYP,{WFF16},{}>}
: source(holybook,wff16)
  WFF17:  SOURCE(HOLYBOOK,all(X)(OLD(X) => SMART(X)))  {<HYP,{WFF17},{}>}
```

What the system currently believes:

: list-asserted-wffs
WFF17: SOURCE(HOLYBOOK,all(X)(OLD(X) => SMART(X))) {<HYP,{WFF17},{}>}
WFF16: all(X)(OLD(X) => SMART(X)) {<HYP,{WFF16},{}>}
WFF15: SOURCE(PROF,all(X)(GRAD(X) => SMART(X))) {<HYP,{WFF15},{}>}
WFF14: all(X)(GRAD(X) => SMART(X)) {<HYP,{WFF14},{}>}
WFF13: SOURCE(SEXIST,all(X)(FEMALE(X) => (~SMART(X)))) {<HYP,{WFF13},{}>}
WFF12: all(X)(FEMALE(X) => (~SMART(X))) {<HYP,{WFF12},{}>}
WFF11: SOURCE(NERD,all(X)(JOCK(X) => (~SMART(X)))) {<HYP,{WFF11},{}>}
WFF10: all(X)(JOCK(X) => (~SMART(X))) {<HYP,{WFF10},{}>}
WFF9: GREATER(PROF,SEXIST) {<DER,{WFF2,WFF3,WFF6},{}>}
WFF8: GREATER(HOLYBOOK,SEXIST) {<DER,{WFF1,WFF2,WFF3,WFF6},{}>}
WFF7: GREATER(HOLYBOOK,NERD) {<DER,{WFF1,WFF2,WFF6},{}>}
WFF6: all(Z,Y,X)({GREATER(Y,Z),GREATER(X,Y)} &=> {GREATER(X,Z)}) {<HYP,{WFF6},{}>,<DER,{WFF6},{}>}
WFF5: GREATER(FRAN,SEXIST) {<HYP,{WFF5},{}>,<DER,{WFF5},{}>,<DER,{WFF3,WFF4,WFF6},{}>}
WFF4: GREATER(FRAN,NERD) {<HYP,{WFF4},{}>,<DER,{WFF4},{}>}
WFF3: GREATER(NERD,SEXIST) {<HYP,{WFF3},{}>,<DER,{WFF3},{}>}
WFF2: GREATER(PROF,NERD) {<HYP,{WFF2},{}>,<DER,{WFF2},{}>}
WFF1: GREATER(HOLYBOOK,PROF) {<HYP,{WFF1},{}>,<DER,{WFF1},{}>}

Fran is the source of some info about her (info not yet asserted):
;;; Fran: I'm an old, female, jock who's a grad.
: source(fran,andor (4,4) {jock(fran),grad(fran),old(fran),female(fran)})
WFF23: SOURCE(FRAN,FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)) {<HYP,{WFF23},{}>}

Now, assert that info about Fran with forward inferencing:
: andor (4,4) {jock(fran),grad(fran),old(fran),female(fran)}!

It is the case that FEMALE(FRAN)
It is the case that OLD(FRAN)
It is the case that GRAD(FRAN)
It is the case that JOCK(FRAN)

Since WFF12: all(X)(FEMALE(X) => (~SMART(X))) {<HYP,{WFF12},{}>}
and WFF21: FEMALE(FRAN) {<DER,{WFF22},{}>}
I infer ~SMART(FRAN)

Since WFF16: all(X)(OLD(X) => SMART(X)) {<HYP,{WFF16},{}>}
and WFF20: OLD(FRAN) {<DER,{WFF22},{}>}
I infer SMART(FRAN)

A contradiction was detected within context DEFAULT-DEFAULTCT.
The contradiction involves the newly derived proposition:
WFF24: SMART(FRAN) {<DER,{WFF16,WFF22},{{WFF12}}>}

and the previously existing proposition:
WFF25: ~SMART(FRAN) {<DER,{WFF12,WFF22},{{WFF16}}>}

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} ∪{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))) {<HYP,{WFF12},{{WFF16,WFF22}}>}
because it is the best choice given the previous information.

And the forward inferencing continues:
It is the case that WFF21: FEMALE(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}
It is the case that WFF20: OLD(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}
It is the case that WFF19: GRAD(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}

It is the case that WFF18: JOCK(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}

Since WFF16: all(X)(OLD(X) => SMART(X)) {<HYP,{WFF16},{{WFF12,WFF22}}>}
and WFF20: OLD(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}
I infer WFF24: SMART(FRAN) {<DER,{WFF16,WFF22},{{WFF12}}>}

Since WFF14: all(X)(GRAD(X) => SMART(X)) {<HYP,{WFF14},{{WFF12,WFF16,WFF22}}>}
and WFF19: GRAD(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}
I infer WFF24: SMART(FRAN) {<DER,{WFF16,WFF22},{{WFF12}}>}

Since WFF10: all(X)(JOCK(X) => (~SMART(X))) {<HYP,{WFF10},{{WFF12,WFF16,WFF22}}>}
and WFF18: JOCK(FRAN) {<DER,{WFF22},{{WFF12,WFF16}}>}
I infer WFF25: ~SMART(FRAN) {<DER,{WFF12,WFF22},{{WFF16}}>}

A contradiction was detected within context DEFAULT-DEFAULTCT.
The contradiction involves the newly derived proposition:
WFF25: ~SMART(FRAN) {<DER,{WFF12,WFF22},{{WFF14},{WFF16}}>, <DER,{WFF10,WFF22},{{WFF14},{WFF16}}>}

and the previously existing proposition:
WFF24: SMART(FRAN) {<DER,{WFF16,WFF22},{{WFF10},{WFF12}}>, <DER,{WFF14,WFF22},{{WFF10},{WFF12}}>}

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 hypotheses listed below are included in more than
one set. Removing one of these will make more than one
set consistent.

Common to 2 sets:
WFF22: FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)
{<HYP,{WFF22},{{WFF12,WFF16},{WFF10,WFF14},{WFF10,WFF16},{WFF12,WFF14}}>}
WFF10: all(X)(JOCK(X) => (~SMART(X))) {<HYP,{WFF10},{{WFF16,WFF22},{WFF14,WFF22}}>}

The three sets aiding culprit selection:
The least believed hypothesis:
(WFF10)
The most common hypotheses:
(WFF22 WFF10) ***reported above as common to both sets***
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))) {<HYP,{WFF10},{{WFF16,WFF22},{WFF14,WFF22}}>}
because it is the best choice given the previous information.

And forward inferencing continues, resulting in the final list of beliefs (edited with bold to show new additions):
: list-asserted-wffs
WFF27: ~(all(X)(JOCK(X) => (~SMART(X)))) {<EXT,{WFF14,WFF22},{{WFF10},{WFF12}}>}
WFF26: ~(all(X)(FEMALE(X) => (~SMART(X)))) {<EXT,{WFF16,WFF22},{{WFF10},{WFF12}}>}
WFF24: SMART(FRAN) {<DER,{WFF16,WFF22},{{WFF10},{WFF12}}>, <DER,{WFF14,WFF22},{{WFF10},{WFF12}}>}
WFF23: SOURCE(FRAN,FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN))
WFF22: FEMALE(FRAN) and OLD(FRAN) and GRAD(FRAN) and JOCK(FRAN)
WFF21: FEMALE(FRAN)
WFF20: OLD(FRAN)
WFF19: GRAD(FRAN)
WFF18: JOCK(FRAN)
WFF17: SOURCE(HOLYBOOK,all(X)(OLD(X) => SMART(X)))
WFF16: all(X)(OLD(X) => SMART(X))
WFF15: SOURCE(PROF,all(X)(GRAD(X) => SMART(X)))
WFF14: all(X)(GRAD(X) => SMART(X))
WFF13: SOURCE(SEXIST,all(X)(FEMALE(X) => (~SMART(X))))
WFF11: SOURCE(NERD,all(X)(JOCK(X) => (~SMART(X))))
WFF9: GREATER(PROF,SEXIST)

WFF8: GREATER(HOLYBOOK,SEXIST)
WFF7: GREATER(HOLYBOOK,NERD)
WFF6: all(Z,Y,X)({GREATER(Y,Z),GREATER(X,Y)} &=> {GREATER(X,Z)})
WFF5: GREATER(FRAN,SEXIST)
WFF4: GREATER(FRAN,NERD)
WFF3: GREATER(NERD,SEXIST)
WFF2: GREATER(PROF,NERD)
WFF1: GREATER(HOLYBOOK,PROF)

All belief revision in this example was done automatically, the user gave no input after Fran's info was asserted with forward inferencing until the final request to see all the beliefs (list-asserted-wffs).

Appendix 2: Structural vs. Assertional Information

To represent the source information *structurally* would be to represent it as an additional argument of the predicates. 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)`. There are several problems with this:

1. It is not immediately clear that `Smart(Fran, Prof)` and `Smart(Fran, Nerd)` represent two sources agreeing about *the same belief.*
2. 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.
3. It is not clear whether `~Smart(Fran, Sexist)` represents 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.
4. It is not clear how to ascribe a source to a rule, such as

   `all(x)(Grad(x) => Smart(x)).`

To represent the source information *assertionally*, however, would be 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))`.[9] This solves the four problems cited above:

1. It is clear that `Source(Prof, Smart(Fran))` and `Source(Nerd, Smart(Fran))` represent two sources agreeing about the same belief, because the second arguments of both are identical – representing a single proposition.
2. 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))`.
3. 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))`.
4. 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)))`.

[9] See Shapiro, S. C. Belief Spaces as Sets of Propositions, *Journal of Experimental and Theoretical Artificial Intelligence 5* (1993), 225-235 for the semantics of proposition-valued terms.

References

Alchourron, C.E., Gärdenfors, P. and Makinson, D. (1985), "On the Logic of Theory Change: Partial Meet Contraction and Revision Functions," *Journal of Symbolic Logic*, 20, 510-530.

Cravo M.R. and Martins J.P. (1993), "SNePSwD: A Newcomer to the SNePS Family", *Journal of Experimental and Theoretical Artificial Intelligence* 5: 135-148.

Doyle, J. (1992), "Reason Maintenance and Belief Revision: Foundations vs. Coherence Theories", in: P. Gärdenfors (Ed.), Belief Revision. Cambridge University Press, Cambridge, U.K. 29-51

Ehrlich, K. (1995), "Automatic Vocabulary Expansion Through Narrative Context," Ph.D. Dissertation, Tech. Rept. 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", *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates, Mahwah, NJ. 205-210.

Gärdenfors, P. and Rott, H. (1995), Belief Revision, in: Gabbay, Hogger and Robinson (Eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol 4 Epistemic and Temporal Reasoning. Clarendon Press, Oxford. 35-131.

Martins, J.P. and Shapiro, S.C. (1988), A Model for Belief Revision, *Artificial Intelligence* 35 25-79.

McKay, D. P. and Martins, J. (1981) "SNePSLOG User's Manual." Department of Computer Science, University at Buffalo.

Rapaport, W.J., and Ehrlich, K (2000, forthcoming), ``A Computational Theory of Vocabulary Acquisition'', In Lucja Iwanska and Stuart C. Shapiro, Eds., Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language, AAAI Press/The MIT Press, Menlo Park, CA.

Shapiro, S.C. (1992), Relevance logic in computer science, in A.R. Anderson, N.D. Belnap, Jr., M. Dunn, et al. (eds.) Entailment, Volume II. Princeton University Press, Princeton, NJ. 553-563.

Shapiro, S.C. and Rapaport, W.J. (1987), "SNePS considered as a fully intensional propositional semantic network". In N. Cercone and G. McCalla, editors, The Knowledge Frontier, Springer-Verlag, New York. 263-315.

Shapiro, S.C. and Rapaport, W.J. (1992), "The SNePS family." *Computers & Mathematics with Applications*, 23(2-5):243-275. Reprinted in F. Lehmann, Ed. Semantic Networks in Artificial Intelligence. Pergamon Press, Oxford. 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. *http://www.cse.buffalo.edu/sneps/Manuals/index.html*