# Redefining Belief Change Terminology for Implemented Systems*

**Frances L. Johnson** and **Stuart C. Shapiro**
Department of Computer Science and Engineering and Center for Multisource Information Fusion and Center for Cognitive Science
University at Buffalo, The State University of New York
226 Bell Hall, Buffalo, NY 14260-2000, USA
{*flj* | *shapiro*}*@cse.buffalo.edu*

## Abstract

Knowledge representation and reasoning systems run into danger when they attempt to implement traditional belief change theories intended for ideal reasoning agents. Resource limitations can cause a system that guarantees consistency to fail. We present a belief space whose formalization incorporates the fact that it is not guaranteed to be deductively closed (e.g. unknown implicit beliefs might indicate an undetected inconsistency). Using this formalization, we then define belief maintenance terminology that applies to this deductively open belief space. This new terminology is used to alter some traditional (ideal agent) belief change guidelines and postulates so that they are implementable. We complete our discussion by comparing some current KRR systems and theories in terms of these new concepts and guidelines.

## Introduction

Implemented knowledge representation and reasoning, KRR, systems run into danger when they attempt to implement traditional belief change theories intended for ideal reasoning agents [Alchourrón *et al.*, 1985; Nebel, 1989; Hansson, 1993; 1999]. Resource limitations can cause a system that guarantees consistency to fail. This is especially true with commonsense reasoning which requires a large knowledge base and a complex reasoning system. Due to this size and complexity, implemented commonsense systems cannot guarantee deductive closure, completeness, or decidability in our resource-limited world. This prevents them from being able to implement most of the belief change theories that have been developed assuming the reasoning agent is ideal (capable of instantaneous deductive closure).

We present a knowledge state formalization that incorporates the facts that

1. its belief space is not guaranteed to be deductively closed
2. it does not guarantee that it can determine if a belief is derivable
3. it cannot guarantee to know all ways that a belief can be derived.

This knowledge state represents a deductively *open* belief space (DOBS) that, at any given time, consists of its explicit beliefs. These explicit beliefs are comprised of the set of base assertions and only those derivations that have been performed *up to that point*. The belief space grows as more beliefs are derived.

Using this formalism, we redefine belief change terminology to apply to a DOBS. This terminology can then be used to alter existing theories thus making them implementable. These altered theories can be used for evaluating and comparing implemented systems.

The next section describes the motivations and assumptions, notation and a brief description of why our DOBS formalism should not be confused with a belief base [Hansson, 1999; Nebel, 1989]. Sections 2 and 3 give background information about belief revision and some traditional belief revision guidelines and postulates. Sections 4 and 5 discuss our DOBS formalism and terminology and the DOBS postulates and belief revision guidelines. Section 6 discusses and compares several KRR systems and how to apply the formalism when considering resource limitations. The last section contains conclusions and future work.

## 1 Preliminaries

### 1.1 Motivations and Assumptions

Most implemented KRR systems run into danger when they promise to implement belief change theories that assume an ideal reasoning agent. One guideline that is difficult to implement with certainty is Hansson's success postulate for contraction [Hansson, 1999]: Unless a formula is a theorem, it should be contracted from a belief space in such a way that it is no longer derivable. This assumes that all ways to derive that belief can be known and eliminated.

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

This is a preliminary version of Frances L. Johnson and Stuart C. Shapiro, Redefining Belief Change Terminology for Implemented Systems, Leopoldo Bertossi and Jan Chomicki, Eds., Working Notes for the IJCAI 2001 Workshop on Inconsistency in Data and Knowledge, IJCAII and AAAI, Seattle, WA, August 6, 2001, 11-21. All quotes should be from, and all citations should be to the published version.

Many KRR systems that implement these ideal agent theories use small domains and/or limited logics (such as propositional logic). These systems can still run into trouble, however, if the user cannot wait for some NP-hard algorithm to complete its run or if the implementation details fall short of the system design. Systems that use large domains or more powerful logics (such as first-order predicate logic with functions, FOPL) have resource-bounded limitations, but each system deals with these limitations in a different way. These differences are hard to compare without a common formalism for describing them.

We assume that KRR systems are being applied to knowledge bases that are growing larger and merging with other knowledge bases. We also assume that the reasoning that needs to be done with these knowledge bases requires a powerful logic like FOPL. Since this logic (and others that are similarly powerful) is undecidable, we claim that it is unreasonable to expect an implemented system to know all ways a given belief might be derived nor to determine in a fixed amount of time whether a set of beliefs is inconsistent. Our DOBS formalism and terminology takes these types of limitations into account and adapts the current belief change guidelines to apply to belief change operations in resource-bounded systems. This new formalism also offers a common ground for comparing implemented systems that use different techniques for dealing with their resource-boundedness.

As a final note, we consider *any* system that is implemented to be resource-limited (or restricted in some way). Even systems using propositional logic must have some size restriction, or their reasoning will exceed available memory or the user's time limitations. Size and time constraints are always a factor to be considered.

### 1.2 Notation and Terminology

Unless otherwise noted, we will simplify this paper by discussing a knowledge base that uses classical, propositional logic. We use a propositional language, $\mathcal{L}$, which is closed under the truth functional operators $\neg, \vee, \wedge, \rightarrow$, and $\leftrightarrow$. Formulas of the language $\mathcal{L}$, including propositional symbols of $\mathcal{L}$, are denoted by lowercase letters $(p, q, r, \ldots)$. Sets of formulas are denoted by uppercase letters $(A, B, C, \ldots)$.

If $p$ can be derived from some set $A$, then we will say that $A$ derives $p$, and will denote it as $A \vdash p$. The classical consequence operator, *Cn*, is defined by $Cn(A) = \{p \mid A \vdash p\}$. The set $A$ is deductively closed if $A = Cn(A)$.

Our formalism and examples can be applied to more powerful logics, such as first-order predicate logic (FOPL) and, possibly, others as well. At this point, we abandon the classical consequence operator, *Cn*, for a more general consequence operator, *C*, to emphasize the potential departure from classical logic.

A belief space $K$ is consistent if and only if for every $p \in Cn(K), \neg p \notin Cn(K)$. For the purpose of this paper, we will refer to an inconsistency as the existence of both some proposition $p$ *and* its negation, $\neg p$, as opposed to their conjunction, $p \wedge \neg p$, or any other formula or set of propositions from which this pair could be derived. This also supports the way that resolution/refutation finds that a set of beliefs is inconsistent — by finding some $p$ and $\neg p$ that reduce to the empty set. We make this choice to reduce the discussion to the base case of a contradiction, though implementations could expand their detection of contradictions to include more complex formulas or groups of formulas.

When we refer to a proposition as a "belief", we will be specifically referring to a proposition that is currently believed by the system. A proposition is "believed" if the system accepts it (asserts that it is true; considers it an *asserted* belief). It can become unasserted if it is retracted; this is not the same as believing its negation. In this paper, we are not considering the term "belief" to refer to a weakened assertion or a non-fact. Also, please note that the term "proposition" can refer to any formula in the system that can be asserted $(p, p \wedge \neg q, p \rightarrow q$, etc.).

### 1.3 A Deductively Open Belief Space (DOBS)

A DOBS is a belief space that is by definition not guaranteed to be deductively closed. It contains some core set of beliefs (a belief base) and the beliefs that *have been* derived from them *so far*. For example, a DOBS might include the beliefs $p$ and $p \rightarrow q$ without the derivable proposition $q$. There is a marked difference between the concept of a DOBS and that of a belief base [Nebel, 1989; Hansson, 1999]. A belief base is a finite set of beliefs that represents the belief space that is its *deductive closure*. The DOBS *is* its belief space of *explicit* beliefs only and can grow through additional deductions *without expanding its base beliefs*.

However, the DOBS for some belief base B *does* form a finite set of beliefs whose deductive closure is the same as that of B — thus, technically, both the DOBS and B are belief bases for the same deductively closed belief space. Our concept for a belief base, however, refers to some core set of beliefs which are asserted with independent standing (as opposed to *any* finite set of beliefs whose closure is a predetermined belief space). See [Hansson, 1999] for a more complete discussion of base beliefs.

## 2 Background

### 2.1 Ideal vs. Resource-Bounded Agents

Belief maintenance research has produced many theories on how to perform belief change operations, but they always have to consider the agent. When working with pure theory, researchers can assume an ideal agent—one that can reason instantly and has infinite space for retaining beliefs. This type of agent can consider the deductive closure of a set of beliefs whenever it has to determine the consistency of that set or remove beliefs to maintain consistency.

In a real-world implementation resources are limited—agents take time to reason, and their memory space is finite. The theoretical deductive closure of a set can be infinite, depending on the logic, so a resource-bounded agent [Wassermann, 1999] cannot consider the deductive closure. Reasoning in an implemented system also takes time, so the system is likely to have implicit beliefs it is not yet aware of.

For this paper, we are not considering an agent's limits of logical ability [Wassermann, 1999]. However, regarding implemented KRR systems, we do recognize that the logic as described by a system designer and the logical abilities of the system as they have been implemented might differ — with the latter falling short of the ideal design. This is a prime example of why even the most simple KRR system cannot guarantee correct information at all times.

### 2.2 Foundations and Coherence Approaches

Systems implemented (and especially those implemented for commonsense reasoning) typically follow a foundations approach [Doyle, 1979; Hansson, 1993] where the belief space is made up of a base set of beliefs, which are self-supporting, and beliefs derived from those base beliefs. Belief change operations alter the base set of beliefs, and derived beliefs that lose their support are no longer believed. We are assuming that ALL the systems we will be discussing are using a foundations approach.

By contrast, a coherence approach [Alchourrón *et al.*, 1985] would allow previously derived beliefs to remain as long as they are not responsible for an inconsistency. . . even if their support was lost. Although our DOBS formalism follows the foundations approach, we offer a coherence version of the DOBS in Section 4.2 which would allow a coherence approach to DOBS reason maintenance. The difference between these two approaches, however, is not of primary interest.

### 2.3 The Need for a DOBS Formalization

Of main interest is that most belief change approaches use the concept of retaining (or returning) consistency to define their belief revision operations. Even a belief base is considered inconsistent if an inconsistency exists in its deductive closure. Whether the inconsistency is found by deductive closure or some procedure, such as resolution refutation, it still requires looking past what is known explicitly into the implicit beliefs to find an inconsistency. This requires time and space. Even a system using classical propositional logic could fail if the process requires more memory/time than the system/user has available.

How do you implement ideal techniques in a non-ideal system? 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, resource-bounded, implemented system. These theories need to define a belief maintenance system that:

1. is not dependent on deductive closure (thus, a DOBS)
2. takes time and computational limitations into account
   - recognizing that these limitations might result in revision choices that are poor in hindsight
3. catches and corrects these poor choices as efficiently as possible.

## 3 Traditional Constraints and Postulates

We discuss below some belief change guidelines and postulates. The belief change operations discussed in this section are expansion, contraction and revision. Expansion is simply the addition of a new belief to the belief space regardless of any inconsistencies that it might cause. Conceptually, contraction is the removal of a belief from a belief space, and revision is adding a belief to the belief space while removing any other beliefs that would contradict it (i.e., *consistent* addition of a belief).

### 3.1 Belief Change Guidelines

Gärdenfors and Rott [Gärdenfors and Rott, 1995] list four guidelines (paraphrased below) for knowledge bases and belief change operations. Belief revision literature refers to these as "integrity constraints", but we will use the term guidelines to avoid confusion with *database* integrity constraints. These guidelines are:

**BCG1** A knowledge base should be kept consistent whenever possible.

**BCG2** If a proposition can be derived from the beliefs in the knowledge base, then it should be included in that knowledge base.

**BCG3** There should be a minimal loss of information during belief revision.

**BCG4** If some beliefs are considered more important or entrenched than others, then belief revision should retract the least important ones.

Constraint 1 is implementable depending on your interpretation of the phrase "whenever possible." We will alter it to clarify what it means in a DOBS system. Constraint 2 as stated precludes the very notion of a DOBS, requiring the creation of some DOBS terms that can be used to rewrite constraint 2 for a DOBS. Constraint 3 will be slightly altered to conform to a DOBS, and constraint 4 will remain unchanged.

### 3.2 AGM Postulates

Contraction and revision are not directly defined, but, instead, are constrained by rationality postulates [Alchourrón *et al.*, 1985], shown below.

**Expansion**
The belief space $K$ expanded by the proposition $p$ is written as $K+p =_{def} Cn(K \cup \{p\})$.

**Contraction**
For contraction of a belief space $K$, by the proposition $p$ ($K– p = K'$), the six basic AGM postulates are:

| | |
|---|---|
| (K–1) $K'$ is a belief space | *Closure* |
| (K–2) $K' \subseteq K$ | *Inclusion* |
| (K–3) If $p \notin K$, then $K' = K$ | *Vacuity* |
| (K–4) If not $\vdash p$, then $p \notin K'$ | *Success* |
| (K–5) $K \subseteq K' + p$ | *Recovery* |
| (K–6) If $\vdash p \leftrightarrow q$, then $K–p = K–q$ | *Extensionality* |

**Revision**
The six basic AGM postulates for revision of a belief space $K$, by the proposition $p$ ($K*p = K'$) are:

| | |
|---|---|
| (K∗1) $K'$ is a belief space | *Closure* |
| (K∗2) $p \in K'$ | *Success* |

(K$*$3) $K' \subseteq K+p$ *Expansion 1*

(K$*$4) If $\neg p \notin K$, then $K+p \subseteq K'$ *Expansion 2*

(K$*$5) $K' = K_\perp$[1] only if $\vdash \neg p$. *Consistency preservation*

(K$*$6) If $\vdash p \leftrightarrow q$, then $K*p = K*q$ *Extensionality*

### 3.3 Hansson's Belief Base Postulates

Hansson [Hansson, 1993; 1999] proposes AGM-style postulates that can apply to base revision: where a belief space $BS$ is revised by performing a belief change operation on some finite belief base $H$, where $Cn(H) = BS$.

Hansson's postulates were written for contraction and revision of a belief base by a *set* of propositions. Below are rewritten versions, altered for contraction (and revision) of a belief base, $H$, by a proposition $p$ (as done in [Gärdenfors and Rott, 1995]), where $Z$ is a set of propositions in $\mathcal{L}$ and $q$ is a proposition in $\mathcal{L}$. Conceptually, contraction of $H$ by $p$ ($H$–$p$) means removing elements of $H$ to form $H'$ such that the belief space for $H'$, ($Cn(H')$), no longer contains $p$. Revision of $H$ by $p$ ($H*p$) means adding $p$ to $H$ (to form $H'$) while maintaining consistency.

**Expansion**

Expansion for a belief base is similar to that for a belief space (above) without the deductive closure. The belief base $H$ expanded by the proposition $p$ would be written as $H+p =_{def} H \cup \{p\} = H'$. The belief space is $Cn(H')$.

**Contraction**

The basic postulates for contraction of a belief base $H$ by a proposition $p$ ($H$–$p = H'$) are:

(H–1) $H' \subseteq H$ *Inclusion*

(H–2) If not $\vdash p$, then $p \notin Cn(H')$ *Success*

(H–3) If $q \in H \setminus H'$, then there is some $H''$ such that $H' \subseteq H'' \subseteq H$ and $p \notin Cn(H'')$, but $p \in Cn(H'' \cup \{q\})$ *Relevance*

(H–4) If it holds for all subsets $H''$ of $H$ that $p \notin Cn(H'')$ if and only if $q \notin Cn(H'')$, then $H$–$p$ = $H$–$q$ *Uniformity*

(H–5) If not $\vdash p$ and each element of $Z$ implies $p$, then $H' = (H \cup Z)$–$p$ *Redundancy*

**Revision**

The basic postulates for revision of a belief base $H$ by the proposition $p$ ($H*p = H'$ whose belief space is $Cn(H')$) are:

(H$*$0) If not $\vdash \neg p$, then $H'$ is consistent. *Consistency*

(H$*$1) $H' \subseteq H \cup \{p\}$ *Inclusion*

(H$*$2) If $q \in H \setminus H'$, then there is some H" such that $H' \subseteq H'' \subseteq H \cup \{p\}$, $H''$ is consistent, and $H'' \cup \{q\}$ is inconsistent *Relevance*

(H$*$3) $p \in Cn(H')$ *Success*

(H$*$4) If for all $H'' \subseteq H$, $H'' + p$ is inconsistent if and only if $H'' + q$ is inconsistent, then $H \cap H*p = H \cap H*q$ *Uniformity*

(H$*$5) If not $\vdash \neg p$ and each formula in $Z$ is logically inconsistent with $p$, then $H' = (H \cup Z)*p$ *Redundancy*

[1] $K_\perp$ is the inconsistent set.

## 4 Formalizing a DOBS

Because a DOBS cannot guarantee deductive closure, it cannot guarantee that it is consistent. This prevents a DOBS from being able to satisfy the guidelines described in Section 3, which center on the concept of guaranteeing consistency.

We have defined and developed a formalization for a DOBS, which also allows us to redefine the terminology surrounding the basic constraints and postulates for belief change. Once altered to conform to DOBS terminology, most of the constraints and postulates above can be adhered to by a DOBS.

### 4.1 Defining a DOBS

A deductively open belief space is formed by a set of assumptions ($HYPS$) and the beliefs that are derived from them ($DERS$). These beliefs are derived gradually over time, so $DERS$ can increase monotonically even as $HYPS$ remains unchanged. These derivations are generated as a result of a query process: either a query from a user about the system (e.g. Does Tweety fly?) or a query generated during inference (e.g. Is Tweety a bird?... if so, then Tweety flies.).

If the system stores derivations (preferable if computing costs are more expensive than memory, so this is the assumption we are making), it only stores derivations that it has actually performed - not all possible derivations. For example: given $HYPS = \{p, p \rightarrow q, r, r \rightarrow q\}$ and $DERS = \{q\}$ (derived from $p$ and $p \rightarrow q$), the system is currently unaware that $r$ and $r \rightarrow q$ also imply $q$.

### 4.2 The Knowledge State That Determines a DOBS

The entire belief state of a DOBS is represented by a knowledge state, $KS$. The DOBS *is* the belief space of the knowledge base, $BS(KS)$. Given the language $\mathcal{L}$ as described in Section 1.2, a belief state is defined as:

$KS =_{def} \langle HYPS, DERS, B, Hist \rangle$,
where $HYPS \subseteq \mathcal{L}, DERS \subseteq C(HYPS), B \subseteq HYPS$, and $Hist$ contains a record of every derivation that has been performed to derive the propositions in $DERS$.

Unless otherwise noted, assume that all future examples and definitions of belief change operations are using $KS =< HYPS, DERS, B, Hist >$ as their starting belief state.

$HYPS$ consists of all the hypotheses (also called assumptions) ever introduced into KS as self-supporting beliefs [Hansson, 1999]. $HYPS$ contains both currently believed hypotheses (the belief base of the DOBS) as well as those that have been retracted from the belief space.

$DERS$ consists of every proposition $p$ ever derived from some set $A \subseteq HYPS \setminus \{p\}$ using one or more inference steps. Whenever a proposition is derived, it becomes a member of $DERS$ and a record of its derivation is stored in $Hist$ (e.g. $h(p, A) \in Hist$ means that $p$ was derived from the set $A$). A proposition can exist in both $HYPS$ and $DERS$.

$B$ consists of all propositions that are currently asserted as being self-supporting beliefs (i.e. the current base beliefs). Therefore, $HYPS \setminus B$ contains propositions that are no longer believed to be self-supporting, although some of these may be in the current belief space as *derived* beliefs *if* they have derivation histories in $Hist$ showing that they can be derived from $B$.

$Hist$ consists of the derivation histories for all elements of $DERS$. There are multiple ways in which this could be implemented (e.g. ATMS, JTMS, derivation tree), but we will formalize it as $h(p, A) \in Hist$ is the implementation independent record of a derivation of $p$ from the set $A$. Every proposition in $DERS$ must have at least one derivation history stored in $Hist$. A proposition can have more than one derivation history. We do not store $h(p, \{p\})$.

**DOBS – Coherence Version**

We distinguish the propositions in $HYPS$ from those in $DERS$ to allow a foundations approach for belief maintenance. A coherence approach can be implemented, however, by inserting each derived belief into $HYPS$ as a self-supporting belief. $B$ should also include all beliefs known to be derivable from $B$ (determined by the derivation histories in $Hist$). In this sense, $B$ *is* the belief space (the DOBS). The derivation histories of any derived beliefs must still be retained in $Hist$ to aid in belief change operations, because they are the only record of how to retract a derivable belief from a DOBS.

The key parts of the KS of a DOBS are illustrated in Figure 1 and described below. $C(B)$ is included for clarity, but it is not an actual part of a KS, and the KS has no information regarding which elements are in $C(B)$ outside of $BS = B \cup D$.

| | |
|---|---|
| **L** | all well-formed formulas (WFFS) in the language |
| **HYPS** | asserted propositions with independent standing<br>—both believed and disbelieved |
| **B** | the currently believed hypotheses |
| **C(B)** | the deductive closure of **B** (belief space of a DCBS)<br>—potentially infinite (depending on the logic used)<br>—included here for clarity (not part of $KS$) |
| **DERS** | propositions known to be derived from **HYPS** |
| **D** | beliefs known to be derived from **B** |
| **B∪D** | Current belief space, **BS** (darkly shaded)<br>—finite subset of **C(B)** |

**B∪D = BS = the DOBS**

**An Example of a $KS$**

Figure 1 and its explanations and examples should help clarify the many parts of a $DOBS$ and its knowledge state, $KS$. Many sections of the $KS$ are described below with examples of the type of propositions that might be in them. For the sake of simplifying this example, assume all propositions inside the circle $HYPS$ were at one time asserted (also in $B$) after which the formulas in $DERS$ were derived (using these base beliefs) as recorded in $Hist$.

The derivations in $Hist$ are stored as pairs containing the belief and the base beliefs underlying its derivation. For example, the pair $\langle c, \{a, a \rightarrow c\}\rangle$ expresses that $c$ was derived from the base beliefs $a$ and $a \rightarrow c$. This is an ATMS style of recording justification for a derived belief.

After these derivations took place, some propositions were retracted from the belief base $B$. These retracted propositions are now in $\mathbf{HYPS} \setminus \mathbf{B}$, and consist of $p, a \rightarrow f, c \rightarrow d, c \rightarrow h, e, f, y, z, z \rightarrow w$, and $z \rightarrow y$.

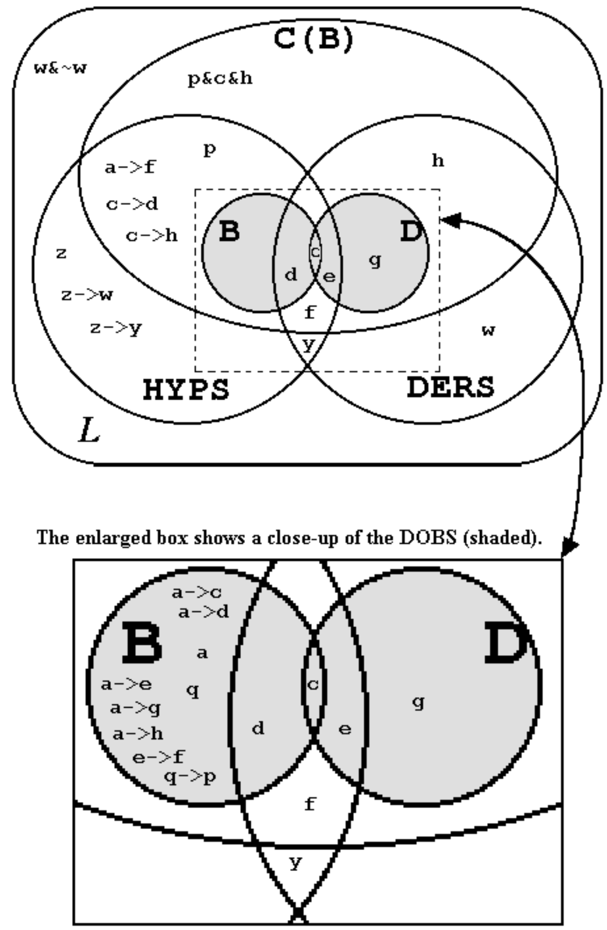


Figure 1: The figure at the top is example of a knowledge state, $KS$. The box at the bottom is an enlargement of the dashed box in the $KS$ diagram. The shaded area represents the DOBS $= BS = B \cup D$. Remember that $C(B)$ is included for comparison purposes. Information regarding its boundaries and contents (with the exception of $BS$) are not a part of the knowledge state $KS$. For this $KS$, $Hist$ contains the following derivations: $\langle c, \{a, a \rightarrow c\}\rangle$, $\langle d, \{c, c \rightarrow d\}\rangle$, $\langle e, \{a, a \rightarrow e\}\rangle$, $\langle f, \{a, a \rightarrow f\}\rangle$, $\langle g, \{a, a \rightarrow g\}\rangle$, $\langle h, \{c, c \rightarrow h\}\rangle$, $\langle w, \{z, z \rightarrow w\}\rangle$, $\langle y, \{z, z \rightarrow y\}\rangle$.

Any propositions requiring one of these for support must be removed from $D$ to $\mathbf{DERS} \setminus \mathbf{D}$ (foundations theory). These propositions that are no longer known to be derivable from $B$ are $f, h, y$, and $w$ (plus $d$ which is asserted in $B$, but not known to be derivable from $B \setminus \{d\}$).

Beliefs in the base $\mathbf{B}$ are asserted as self-supporting and are shown in the enlarged box at the bottom of Figure 1. Two of these propositions, $c$ and $d$, are also in $DERS$, because of the two derivations stored in $Hist$: $\langle c, \{a, a \rightarrow c\}\rangle$ and $\langle d, \{c, c \rightarrow d\}\rangle$, respectively. The first is a currently believed derivation, but the second is disbelieved due to the retraction of the proposition $c \rightarrow d$. This is shown by $c$ being in $\mathbf{B} \cap \mathbf{D}$ whereas $d$ is only in $\mathbf{B} \cap \mathbf{DERS}$.

The three beliefs in $\mathbf{D}$ — $c, e$, and $q$ — are in three different sections All are known to be derived from $B$, but $c$ is also asserted as a hypothesis and $e$ is disbelieved as a hypothesis .

The two propositions not mentioned, yet, are $w \wedge \neg w$ and $p \wedge c \wedge h$. They are located in the areas $\mathcal{L} \setminus (HYPS \cup DERS \cup C(B))$ and $C(B) \setminus (HYPS \cup DERS)$, respectively. These propositions are, actually, not known by KS, but they are included in the diagram as examples of the type of propositions that would be in these areas conceptually.

## 4.3 DOBS Terminology

**KS-derivability**

Since a DOBS can have propositions that are derivable but not, yet, derived, we use the concept of a proposition, $p$, being *known to be derivable* from a set of propositions, $A$. This is denoted as $A \vdash_{KS} p$, read as "$A$ KS-derives $p$", and is defined by the rules below:

1. $\{p\} \vdash_{KS} p$.
2. If there exists some $h(p, A) \in Hist$, then $A \vdash_{KS} p$.
3. $A \vdash_{KS} B$ means that $\forall p[p \in B \rightarrow A \vdash_{KS} p]$.
4. A superset of a set that KS-derives a proposition also KS-derives that proposition:
   $(A \vdash_{KS} p) \wedge (A \subset B) \rightarrow B \vdash_{KS} p$.
5. $(A \vdash_{KS} B) \wedge (B \vdash_{KS} p) \rightarrow A \vdash_{KS} p$. *Transitivity*

**KS-closure**

Because we are removing the omniscience of a DCBS and its consequence operation, we want the DOBS to "remember" as much as possible, including propositions that are no longer believed. Once a base set of hypotheses, $B$, is chosen, the implementable closure of $B$ is limited by $KS$ (i.e. by its derivation histories in $Hist$). We call this *KS-closure*, and its consequence operator is $C_{KS}$. Its definition is:
$C_{KS}(B) =_{def} \{p | B \vdash_{KS} p\}$.

The belief space of a DOBS is defined by its belief state, $KS$, as the KS-closure of its base beliefs ($B$):
$BS(KS) =_{def} C_{KS}(B)$ = *the* DOBS.

$D(KS)$ is the set of *derived* propositions that are currently believed: $D(KS) =_{def} \{p | B \setminus \{p\} \vdash_{KS} p\}$.

In other words, KS represents all the propositions that exist in the system along with a record of how they were derived, and BS(KS) represents only those propositions that are currently believed — therefore, it *is* the DOBS. The KS must keep track of the disbelieved propositions and derivations to avoid having to repeat earlier derivations if disbelieved propositions are returned to the belief space (assuming that memory space is a more plentiful resource than computing speed/time).

For shorthand purposes, $BS(KS)$ and $D(KS)$ can be written as $BS$ and $D$ respectively when their $KS$ is clear from the context. The information that $p \in HYPS \cup DERS$, can be written in a shorthand version as $p \in KS$. This is not to be confused with $p \in BS$, though the latter implies the former.

Observation: $BS = B \cup D$
Proof: By the definitions above, $B \vdash_{KS} A \rightarrow A \subseteq BS$.
a) Prove $B \cup D \subseteq BS$. $\forall p[p \in B \rightarrow B \vdash_{KS} p$. Therefore, $B \subseteq BS$. $\forall p[p \in D \rightarrow B \setminus \{p\} \vdash_{KS} p$. $\forall p[B \setminus \{p\} \subseteq B]$. Therefore, $\forall p[p \in D \rightarrow B \vdash_{KS} p$, thus $D \subseteq BS$.
b) Prove $BS \subseteq B \cup D$. $\forall p[p \in BS \rightarrow (B \vdash_{KS} p) \wedge ((p \in B) \vee (p \notin B))]$. If $p \notin B$, then $B \vdash_{KS} p \Leftrightarrow B \setminus \{p\} \vdash_{KS} p$. Therefore, $\forall p[p \in BS \rightarrow (p \in B) \vee (p \in D)]$. Thus, $\forall p[p \in BS \rightarrow p \in B \cup D]$. ■

**KS-consistency**

Any set is inconsistent if a contradiction has been derived *or can be derived* from its beliefs. Thus, checking for an inconsistency requires examining implicit beliefs. This is time consuming for a DOBS, which can never guarantee a *complete* exploration of its implicit beliefs.

A DOBS is *KS-inconsistent* if and only if $\exists p[p \in C_{KS}(B) \wedge \neg p \in C_{KS}(B)]$. If a DOBS is not KS-inconsistent, then it is called *KS-consistent* — i.e. there are no explicit inconsistencies in $C_{KS}(B)$, so it is *not known to be inconsistent*.

This means a DOBS can be both inconsistent and KS-consistent at the same time: For example, $B = \{q, p, p \rightarrow \neg q\}$, but $\neg q$ has not, yet, been derived. Note that you can also refer any set, $A$, as KS-consistent or KS-inconsistent as long as there is a $KS$ associated with that set from which you can determine the KS-closure of the set, $C_{KS}(A)$.

**KS-consolidation**

Whenever an inconsistency must be resolved, some beliefs must be removed. Which beliefs are removed is often determined by examining the entire belief space and using some guidelines (such as BCG3 and BCG4 from Section 3.1). A DOBS, however, is incomplete, because of the lack of deductive closure. It is possible that newly derived beliefs would add information that might alter the belief contraction choices made earlier.

A belief change operation we have not yet discussed could enable a DOBS to reconsider past revisions. It is consolidation [Hansson, 1999]. Consolidation involves taking a belief base and making it consistent by retracting beliefs.

The DOBS version of this operation is called *KS-consolidation*, and it is constrained by the following postulates (for a belief base, $B$ that is consolidated to produce $B!_k$)[2]:

**KS-C1** $B!_k$ is KS-consistent *KS-consistency*

[2] These guidelines are an altered version of guidelines in [Wassermann, 1999] who generalized from Hansson's 1991 Ph.D. thesis.

**KS-C2** $B!_k \subseteq B$ *inclusion*

**KS-C3** If $p \in B \backslash B!_k$, then there is some $X$ such that $B!_k \subseteq X \subseteq B$, $C_{KS}(X)$ is KS-consistent, and $C_{KS}(X+p)$ is KS-inconsistent *relevance*

A DOBS system should periodically perform KS-consolidation on $HYPS$ (or some subset of $HYPS$) to reconsider past belief change operations which might be affected by more recently added hypotheses or derivations.

## 5 DOBS Constraints and Postulates

The key to understanding DOBS belief change constraints and postulates is to remember that they are applied in terms of the DOBS terminology. When removing *known* inconsistencies, they deal only with *known* derivations (stored in $Hist$) of *known to be derivable* propositions (stored in $DERS$). Deductive closure is not an option. For the purposes of this paper we assume that KS-closure is possible, though, for an extremely large system, time or space might restrict this.

When referring to expanding, contracting or revising a DOBS by some proposition $p$, the operation is actually performed on the base of the DOBS, $B$. The operation can be written using either "DOBS", "$BS$" or "$B$", since the application is clear (i.e. DOBS$*p = B*p$). This is not a problem for the coherence approach, since $D \subseteq B$ for the coherence version of a DOBS, thus its $B$ = $BS$ = the DOBS. For the purpose of comparing to the AGM and Hansson postulates, we will refer to the DOBS as $K$ and $H$ respectively.

### 5.1 Belief Change Guidelines for a DOBS

Now that we have formalized a DOBS, we can assess the key changes necessary to adjust the list of belief change guidelines from Section 3.1 so that they can be used as guidelines for a DOBS. Alterations are in **boldface**. Additions or clarifications are in *italics*. The revised constraints are:

1. a knowledge base should be kept **KS-consistent** whenever possible;
2. if a proposition **is known to be derivable** from the beliefs in the knowledge base *using the derivations currently known*, then it should be included in that knowledge base **(KS-closure)**;
3. there should be a minimal loss of *the known* 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.

### 5.2 DOBS version of the AGM Postulates

To make a DOBS version of the AGM postulates, we substitute DOBS terminology in place of ideal agent terminology. Once altered using the DOBS terminology, many of the postulates for contraction and revision can be adhered to by a DOBS. Consider the DOBS to be the $K$ of the postulates (i.e. $K = BS(KS)$). Remember that any contractions to prevent inconsistencies are limited by the derivation knowledge in $Hist$.

**Expansion**

Expansion: $K + p =_{def} C_{KS}(K \cup \{p\})$.

**Contraction**

(K–1), (K–2), and (K–3) are adhered to unchanged.

(K–4) becomes: If not $\vdash_{KS} p$, then $p \notin$ K–$p$ (i.e. K–$p$ does not *KS-derive* $p$... but $p$ might still be derivable).

(K–5) cannot hold, because it requires that for every two beliefs, $p$ and $q$, in $B$, the system also have the beliefs $p \rightarrow q$ and $q \rightarrow p$. A DOBS cannot guarantee that all these additional beliefs will be present when a belief is removed.

(K–6) can only be adhered to in the following form: If $\vdash_{KS} p \leftrightarrow q$, then K–$p$ = K–$q$.

**Revision**

(K$*$1), (K$*$2), and (K$*$3) are adhered to unchanged.

(K$*$4) cannot be adhered to, because of the lack of deductive closure. It would be possible in a DOBS to have $K = \{p \rightarrow q, \neg q\}$ without $\neg p$ being in $K$, because it had not, yet, been derived. Yet there could be a derivation history $h(q, \{p, p \rightarrow q\})$ in $Hist$ that would require the retraction of either $\neg q$ or $p \rightarrow q$ upon the revision of $K$ by $p$.

(K$*$5) is rewritten as:
$K*p$ is KS-inconsistent only if $\vdash_{KS} \neg p$.

(K$*$6) requires the same alterations as (K–6):
If $\vdash_{KS} p \leftrightarrow q$, then $K*p = K*q$.

### 5.3 DOBS version of Hansson's Belief Base Postulates

In this case, we revert to the base version of a DOBS (not the coherence version). We also use $H$ to refer to $B$ in the postulates. Remember that the revised DOBS = $C_{KS}(H') = BS(H')$ , where $H'$ is the belief base after the belief change operation.

**Expansion**

Expansion remains the same: $H*p = H \cup \{p\}$.

**Contraction**

Since most of the postulates require terminology alteration to be adhered to, we will merely list them. The postulates for DOBS contraction of a belief base $H$ by a proposition $p$ ($H$–$p = H'$) are:

(H–1) $H' \subseteq H$ *Inclusion*

(H–2) If not $\vdash_{KS} p$, then $p \notin C_{KS}(H')$ *Success*

(H–3) If $q \in H \backslash H'$, then there is some $H''$ such that $H' \subseteq H'' \subseteq H$ and $p \notin C_{KS}(H'')$, but $p \in C_{KS}(H'' \cup \{q\})$ *Relevance*

(H–4) If it holds for all subsets $H''$ of $H$ that $p \notin C_{KS}(H'')$ if and only if $q \notin C_{KS}(H'')$, then $H$–$p$ = $H$–$q$ *Uniformity*

(H–5) If not $\vdash_{KS} p$ and each element of $Z$ KS-derives $p$, then $H' = (H \cup Z)$–$p$ *Redundancy*

**Revision**

The postulates for DOBS revision of a belief base $H$ by the proposition $p$ ($H * p = H'$) are:

(H$*$0) If not $\vdash_{KS} \neg p$, then $H'$ is KS-consistent. *Consistency*

(H$*$1) $H' \subseteq H \cup \{p\}$ *Inclusion*

(H$*$2) If $q \in H \setminus H'$, then there is some $H''$ such that $H' \subseteq H'' \subseteq H \cup \{p\}$, $H''$ is KS-consistent, and $H'' \cup \{q\}$ is KS-inconsistent *Relevance*

(H$*$3) $p \in C_{KS}(H')$ *Success*

(H$*$4) If for all $H'' \subseteq H$, $H'' + p$ is KS-inconsistent if and only if $H'' + q$ is KS-inconsistent, then $H \cap H * p = H \cap H * q$ *Uniformity*

(H$*$5) If not $\vdash_{KS} \neg p$ and each formula in $Z$ is KS-inconsistent with $p$, then $H' = (H \cup Z) * p$ *Redundancy*

### 5.4 Discussion

**DOBS Belief Change Guidelines**

Constraint 1 suggests that a system should activate belief revision as soon as an inconsistency is detected. Constraint 2 recommends that a proposition should not need to be re-derived from a set of propositions from which it had previously been derived.

Constraint 3 reminds us that only explicit beliefs can be considered during a belief change operation (e.g. implicit beliefs cannot be considered). Constraint 4 remains unchanged. How to minimize information lost and combine that with constraint 4 is an ongoing debate.

**DOBS Postulates**

Once adjustments for DOBS limitations were made using DOBS terminology, all but a few postulates ((K–5) and (K$*$4)) were able to be adhered to. Although no implemented system can guarantee to adhere to the original postulates, the adjusted postulates provide achievable guidelines for implemented belief change operations.

## 6 Some KRR Systems and Implementation Considerations

### 6.1 Some KRR Systems

In this section, we briefly describe some KRR systems. Our descriptions are not intended to be complete, but will focus of the aspects of interest to this paper. We apologize to all implementers and researchers for the negative slant given to these systems. They were chosen *because* they are excellent examples of *good* systems that are producing interesting results. However, they also illustrate that, no matter how good a system is, it is an implementation in a resource-limited world and open to failure regarding adherence to ideal standards.

**BReLS**

BReLS[Liberatore, 1999] is a system that combines revision, update and merging techniques for belief change. It uses a propositional logic where optional additional information can be included or added regarding the proposition: its credibility, the time at which it is considered true, and the penalty for changing the truth value of the proposition from one time stamp to another.

Interestingly, BReLS considers any knowledge base to be consistent, because it makes models of each knowledge base, and the models are consistent. Each model is a possible world consistent with as much of the of the input information as possible. For example, the input of $p \wedge \neg p$ would produce two models — one where $p$ is true, one where $p$ is false. Queries can be made on different propositions. If a proposition holds in all models, then the proposition is considered valid, else it is not valid. In the example just mentioned, $p$ and $\neg p$ would both be *not valid*. This system can be run over the Internet. The URL is:

http://www.dis.uniroma1.it/ liberato/brels/brels.html.

**SATEN**

SATEN, developed by Mary-Anne Williams, is an “object-oriented, web-based extraction and belief revision engine.”[Williams and Sims, 2000] It uses the AGM approach for belief revision[Alchourrón *et al.*, 1985] and uses user specified rankings for the beliefs. The extraction process recovers a consistent theory base from an inconsistent ranking. SATEN can reason in either propositional or first-order logic. To query whether a belief is in the belief space of the base, you ask what its ranking would be. A degree of 0 means it is not in the belief space, else the appropriate ranking for the belief is given. No reply means it does not know the ranking. This system can be run over the Internet. The URL is:

http://ecommerce.newcastle.edu.au/saten/.

**Cyc**

Doug Lenat and Cycorp have developed Cyc[Cycorp, 2001a]— a large knowledge base and inferencing system that is built upon a core of over a million hand-entered assertions or rules about the world and how it works. This system attempts to perform commonsense reasoning with the help of this large corpus of beliefs (mostly default with some that are monotonic). It divides its knowledge base into smaller contexts called microtheories which contain specialized information regarding specific areas (such as troop movement, physics, movies, etc.). Belief revision is performed within microtheories or within a small group of microtheories that are working together, and the system is only concerned with maintaining consistency within that small group (as opposed to across the entire belief space). For example: in an everyday context, a table is solid, but within a physics context, it is mostly space (between atoms).

A belief can have only one truth value, so no microtheory can contain both $p$ and $\neg p$. For example, $\neg p$ could be expressed as the proposition $p$ with a truth value of *false*. The technique for maintaining consistency is to check for contradictory arguments whenever a proposition is derived or asserted into a microtheory. When contradictions are found, their arguments are analyzed, and a decision is made regarding the truth value of the propositions involved. Rankings of beliefs, however, is not a part of the system — it uses specificity to determine the truth value of a default belief. For example: Opus the penguin does not fly, even though he is a bird, because penguins don’t fly. If there can be no decision based on specificity, the truth value of the default belief is unknown. A default belief loses out to a monotonic one. And, lastly, according to Cyc trainers and other contacts, contradictions that are purely monotonic bring the system to a halt until they are fixed. During Cyc training, Johnson attempted to prove this last statement and failed — revision was performed. The instructors were surprised, but thought the training interface might be the cause. We plan to explore this further, but it is an example of a system behaving differently than it is described.

**SNePS**

SNePS[Shapiro and The SNePS Implementation Group, 1999; Shapiro and Rapaport, 1992] is a KRR system whose belief space represents the belief space of a cognitive agent called Cassie. SNePS reasons using a paraconsistent, relevance-based logic, and is able to reason in multiple contexts, similar to Cyc.

SNePS attempts to derive propositions as they are asked for—either by the user or by the system as it performs backward or forward inference. A contradiction is detected when a specific proposition and its negation are both explicitly present in the belief space. At this point, belief revision is called to resolve the inconsistency and the system considers only the propositions in the current belief space—asserted hypotheses and the propositions currently known to be derived from them—without considering any implicit beliefs. Because justifications for the contradictory beliefs are stored as sets of base beliefs that are known to underlie the contradiction (ATMS-style), SNePS automatically reduces the set of beliefs under consideration to only those involved with the derivation of the contradiction.

The SNePS system is available online for downloading: http://www.cse.buffalo.edu/sneps/ .

**Wassermann's Resource-Bounded System**

Wasserman's formalism for "resource-bounded belief revision" is described *theoretically* in [Wassermann, 1999], but bears mention here, because it specifically deals with the issue of resource-boundedness. Wassermann defines the concept of "embedded local change" — the belief set being altered during some belief change operation using some proposition $p$ should be some subset of the full base of beliefs that is *relevant* to $p$ (i.e. the elements in the set have some relation to $p$). This relation could be syntactic, logical, whatever, but unrelated propositions should not be considered during the belief change operation. She also gives an anytime algorithm for determining this subset of beliefs, and if the set is small enough, then traditional belief change operations (ideal operations) can be used. The algorithm can be stopped whenever the set reaches some predetermined maximum limit (to deal with resource limitations regarding memory or computation time).

## 6.2 Discussion of These Systems

Most of the systems above check consistency either at the time a belief is entered or when consistency is an issue (e.g. during contraction, revision, extraction, etc.). Promising consistency takes time and, depending on the logic, might not be decidable. The discussion below only mentions some of the features that limits these KRR systems. They perform many operations well and are excellent systems, but their implementations or restrictions indicates that ideal postulate implementation is not possible.

**BReLS**

Even a system using the computationally manageable classical propositional logic has problems. According to Liberatore, its developer[Liberatore, 1999]: "[O]ne can produce examples that cannot be correctly dealt by BReLS, ...[and it] can deal only with small problems [currently limited to less than eight literals]." Liberatore feels that more efficient algorithms will help improve performance. *We* feel there will *always* be pitfalls for *any* implemented system.

**SATEN**

Our work often prompts the response that SATEN implements ideal postulates, so we felt obliged to put it to the test. We did find a case where the implementation of SATEN which is available on the Internet was unable to revise a set of beliefs. This case involves the use of a recursive statement which caused SATEN to get into a loop. Williams (through personal conversation) explained that after a set number of iterations, the looping is terminated, but it seems that the entire process that was running on SATEN gets terminated as well. There were no results. The example we used is described below.

The statements used in our example are:

**S1** `Par(bill,chel)`
(Bill is Chel's parent.)

**S2** `*P(*C(Par(P,C)->Anc(P,C)))`
(Parents are ancestors.)

**S3** `-Anc(bill,chel)`
(Bill is not Chel's ancestor.)

**S4** `*A(*P(*C(`
`(Par(P,A)&Anc(A,C))->Anc(P,C))))`
(Parents of ancestors are ancestors.)

SATEN performs as expected when given only the first 3 statements. For example, revising the belief base of S2 and S3 by the belief S1 results in the removal of S3 leaving only S1 and S2 in the belief base (i.e. $\{$S2,S3$\}$*S1 = $\{$S1,S2$\}$).

Adding S4 caused SATEN to fail in its attempt to revise. When the belief base consisted of S2 (at rank 1), S4 (at rank 2) and S3 (at rank 3), then revision by S1 failed. It caused an intolerable delay (hours, when the above run took only a few seconds) and produced no result ... not even the loss of S3 from the base of beliefs.

In this recursive example, if the system continued its search after popping out of the recursive loop, it could return the base of S1, S2, and S4 with a warning that S4 was not fully tested. Alternatively, it could play it safe and remove S4 along with S3. In either case, it would produce results that would comply with the DOBS postulates, with the latter case bringing in the issue of minimal damage.

Note: If the base consists of all four statements ranked in order (e.g. S1 has rank 1, S2 has rank 2, etc.), SATEN will correctly give the degree for

`Anc(bill,chel)`

as 2. It can determine that *before* having to consider S4, so it never gets into a recursive loop.

**Cyc**

As mentioned in Section 6.1, Cyc did not perform as described, and there must be some question as to other possible differences from design theory. Most specifically Cyc literature[Cycorp, 2001b] claims to keep the microtheories *consistent*, for lack of a better word. When asked, contacts agreed that, in spite of a cursory check, it was possible that unknown contradictions might exist that had not, yet, been derived. In this sense, Cyc can only guarantee that its microtheories are *not known to be inconsistent* (or KS-consistent). Ideal terminology, such as *consistent* and *derivable*, is often not appropriate for use with an large or complex implemented system

**SNePS**

SNePS currently recognizes an inconsistency when a belief $p$ and its compliment $\neg p$ are explicitly believed by the system. This means that a system can be inconsistent without being aware of it. This method for detecting and resolving contradictions seems reasonable for a cognitive agent, but not for an expert system where contradictions should be detected and resolved as soon as possible.

Unlike SATEN, SNePS can handle most recursive rules [McKay and Shapiro, 1981], but there are a few which will still produce infinite looping.

**Wasserman**

Wassermann's system description discusses limiting the size of the belief base under consideration during belief change operations to allow the system to use ideal theories. A key issue when attempting to implement this technique is how to determine the size limit for a belief change operation. Certainly, this might be feasible with a system that uses propositional logic. But when using FOPL in SATEN, a four statement belief base (using the statements described above) could not be consolidated. If implemented, Wasserman's system could guarantee that the entire belief space was KS-consistent. For the smaller subset that is operated on, we also feel that KS-consistency is the only realistic goal.

## 6.3 Dealing with Resource-Boundedness

The examples above are merely to illustrate that implemented systems are imperfect. Likewise, even an ideal DOBS will run into resource limitation problems. Each system can alter the DOBS formalism to suit its techniques for handling resource limitations. The following discussion involves implementation concepts, but it addresses them at a general theoretical level without getting into the minute details of implementation. These implementation adjustments to a DOBS must be understood when comparing different implemented systems.

The key to comparing systems is to consider NOT the state at rest, but the state of the DOBS when decisions need to be made.

**Two Extremes**

When implementing a KRR system, one of the key questions is how to balance storage vs. re-derivation — i.e. what gets saved in memory vs. what items should be rederived. A system with a large domain, fast hardware and efficient processes might choose to only save its base beliefs and rederive other beliefs whenever they are needed. In this case, $DERS$ and $HYPS \setminus B$ would remain empty, and $Hist$ would only store information during query or belief change procedures. After the procedure is completed, $Hist$ could be emptied.

Alternately, a system that has lots of memory with fast look-up strategies but has a slow processor or inefficient inference algorithms would favor retaining all derivations and their histories in $DERS$ and $Hist$, respectively. This way, even beliefs that are retracted and then returned to the belief space will have any previously performed relevant derivations immediately available to them.

Obviously, most systems fall somewhere between these two extremes. Cyc follows closely to the first, and SNePS favors the second. In both cases, however, belief change decisions are made only when $Hist$ is full and active. During the decision process, these systems are very similar. SNePS will focus only on the relevant information in $Hist$, while Cyc fills $Hist$ with relevant information.

**Freeing Memory**

The system that stores information in memory might find a need to reduce information stored to free memory space. The information most essential to maintaining the integrity of the knowledge state would be $B$, which must be retained at all costs, though subcontexts that are rarely used could be stored in some secondary storage location.

If KS-consolidation for review of previously discarded hypotheses is rarely used, $HYPS \setminus B$ could be eliminated. Both $DERS$ and $D$ can be rebuilt from $B$ and $Hist$, so these can go as well – $DERS \setminus D$ first. Any removal from $DERS \setminus D$ could also include the removal of the relevant derivation information from $Hist$, since it is no longer essential to deriving $BS$.

The choice to drop $D$ before $Hist$ might be switched if the system is rarely *removing* beliefs from its $BS$. In this case, storing $B$ and $D$ for quick access might be preferable and $Hist$ could be emptied. This raises two concerns.

1. The DOBS terminology is defined by knowing the derivation information in $Hist$. If that information is discarded, then it should be recognized that the system is storing a sub-version of the KS, and that the presence of a belief, $p$, in $D$ is evidence that $B \vdash_{KS} p$.[3]
2. When a belief $p$ is to be retracted (or a contradiction detected), then, derivations relevant to the retraction (or contradiction) should be stored in $Hist$ until the retraction process is completed and the new $B$ and $D$ are established. After this point, $Hist$ could then be emptied.

[3]This use of $B \cup D$ comes into question depending largely on the logic. If using a relevance logic, as in SNePS, the support set underlying a belief determines how that belief can be used in future derivations, making $Hist$ a necessity. For a non-monotonic logic, the assertion of a new belief into $B \cup D$ requires questioning all beliefs in $D$ (or at least those marked as defeasible).

**Summary**

These issues illustrate the need to compare implemented systems by more than their logic, size and theories. Systems should be analyzed according to their shortcomings as well. How do they handle these shortcomings – with ignorance, aggressiveness, optimism or caution? The system must suit the needs of its user. When resource limitations do arise, how does the system handle them?

By using a common formalism (like a DOBS), systems can be compared more easily and benchmarks for analyzing DOBS alterations will allow a standard for comparing and measuring the efficiency and accuracy as well as the shortcomings and potential hazards of implemented systems.

## Conclusion and Future Work

By redefining belief change terminology to suit the limitations of a DOBS, we have altered some ideal constraints and postulates for application in a resource-bounded system. Implemented systems can be compared by how well they adhere to these altered postulates. Altered postulates that cannot be adhered to can be dropped from DOBS guidelines, or perhaps alterations to the DOBS formalization or to a system's implementation might result in their adherence.

By recognizing the pitfalls of implementing a DOBS, the necessity for KS-consolidation was identified. Only by knowing the shortcomings of an implemented system and incorporating them into the theories for that system can we be sure to deal with those shortcomings. Obviously, a choice must be made between size and power of a system vs. the correctness of adapting an ideal formalization. For those systems that are willing to give up the latter, new terminology is the first step toward formalizing their real-world applications.

Future work will include formalizing the query process for a DOBS as well as defining guidelines for evaluating and comparing implemented systems. Additional work should be done to formalize how resource limitations (e.g. space, time, derivation rules) might affect DOBS belief change operations and how to best deal with those limitations.

## Acknowledgments

The authors appreciate the insights and feedback of Bill Rapaport, Haythem Ismail, Ken Regan, Pavan Aduri, and the SNePS Research Group. We are also grateful for the input gained due to our presentation of this material at the Belief Revision Track of the 8th International Workshop on Non-Monotonic Reasoning. Many thanks to the anonymous referees for their comments and to the attendees who took the time to share their thoughts—especially Mary-Anne Williams, Odile Papini, Ronald Yager, Samir Chopra and Renata Wassermann.

## References

[Alchourrón *et al.*, 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *The Journal of Symbolic Logic*, 50(2):510–530, June 1985.

[Cycorp, 2001a] Cycorp. *Cycorp, Creators of the Cyc KNowledge Base*. http://cyc.com/, 2001.

[Cycorp, 2001b] Cycorp. *Features of CycL*. http://cyc.com/cycl.html, 2001.

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

[Gärdenfors and Rott, 1995] Peter Gärdenfors and Hans Rott. Belief revision. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 35–132. Clarendon Press, Oxford, 1995.

[Hansson, 1993] Sven Ove Hansson. Reversing the Levi identity. *Journal of Philosophical Logic*, 22:637–669, 1993.

[Hansson, 1999] Sven Ove Hansson. *A Textbook of Belief Dynamics*, volume 11 of *Applied Logic*. Kluwer, Dordrecht, The Netherlands, 1999.

[Liberatore, 1999] Paolo Liberatore. BReLS: A system for revising, updating, and merging knowledge bases. In *Workshop notes of the Third Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC'99)*, 1999.

[McKay and Shapiro, 1981] Donald P. McKay and Stuart C. Shapiro. Using active connection graphs for reasoning with recursive rules. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 368–374. Morgan Kaufmann, San Mateo, CA, 1981.

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

[Shapiro and Rapaport, 1992] Stuart C. Shapiro and William J. Rapaport. The SNePS family. *Computers & Mathematics with Applications*, 23(2–5):243–275, January–March 1992. Reprinted in [**?**, pp. 243–275].

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

[Wassermann, 1999] Renata Wassermann. *Resource-bounded Belief Revision*. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, 1999.

[Williams and Sims, 2000] Mary-Anne Williams and Aidan Sims. SATEN: An object-oriented web-based revision and extraction engine. In Chitta Baral and Miroslaw Truszyński, editors, *Proceedings of the 8th International Workshop on Non-Monotonic Reasoning NMR'2000*, 2000. CoRR article: cs.AI/0003059.