Formalizing a Deductively Open Belief Space

CSE Technical Report 2000-02

Frances L. Johnson and Stuart C. Shapiro

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

January 24, 2000

Abstract

A knowledge representation and reasoning system must be able to deal with contradictions and revise beliefs. There has been much research in belief revision in the last decade, but this research tends to be either in the Coherence camp (AGM) or the Foundations (TMS) camp with little crossover. Most theoretical postulates on belief revision and belief contraction assume a deductively closed belief space - something that is computationally hard (or impossible) to produce in an implementation. This makes it difficult to analyze implemented belief revision systems using the theoretical postulates. This paper offers a formalism that describes a deductively open belief space (DOBS). It then uses this formalism to alter the AGM integrity constraints for a DOBS. A DOBS uses a base set of hypotheses, but only deduces beliefs from that base as the result of specific queries. Thus, it can grow over time even if the base remains static, and can never be referred to as consistent - only either inconsistent or "not known to be inconsistent." This work and future alterations to the traditional postulate formalisms will better enable system/postulate comparisons.

Introduction

A knowledge representation and reasoning system must be able to deal with contradictions and revise beliefs. There has been much research in belief revision (Martins 1991; Gärdenfors 1992; Martins 1992a; Martins 1992b; Gärdenfors and Rott 1995; Friedman and Halpern 1996), but this research tends to be either in the Coherence camp or the Foundations (TMS) camp with little crossover. 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.

Most formalized postulates for belief revision and belief contraction come from theorists (as opposed to implementers), who assume a deductively closed belief space (DCBS). This is something that is computationally hard (or impossible) to produce in an implementation, which makes it difficult to compare the operations of implemented belief revision systems with the theoretical postulates.

Our research began with the goal of altering the AGM postulates – and others (Hansson 1993) – for a deductively ***open*** belief space (DOBS), a belief space that builds up its explicit beliefs gradually. We quickly realized that the first step was to formalize the DOBS followed by altering the AGM integrity constraints upon which the postulates were formed. This paper offers our DOBS formalism and a DOBS version of the integrity constraints. For this paper, we will assume that the belief revision system is complete and uses Classical Propositional Logic. The next section contains a brief overview of AGM terminology and Integrity Constraints for belief revision (Alchourron, Gärdenfors, and Makinson 1985). The sections following define a DOBS and its belief change operations, and propose a DOBS version of the Integrity Constraints. In the final section, we present a discussion of our findings and issues that we plan to explore in the future. Most important is how these DOBS integrity constraints will help us to formulate postulates for DOBS belief change operations.

Terminology and DCBS Integrity Constraints

As mentioned above, the system discussed is assumed to be complete and using Classical Propositional Logic. Propositions may also be referred to as sentences. 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 is unasserted if and only if it is *not* accepted – this is not the same as believing its negation. For the purposes of this paper, *an inconsistency* refers to a pair of contradictory propositions, P and $\sim P$, as opposed to their conjunction, $P \wedge \sim P$. A belief space is a set of believed propositions.

Gärdenfors and Rott (Gärdenfors and Rott 1995) list four "integrity constraints" or "rationality postulates" for belief revision that are the basic guidelines for developing postulates for belief change:

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 (deductive closure);
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.

Constraints 3 and 4 can conflict with each other, so properly weighting and combining them is an open question for both theorists and implementers. For example: How do you choose between retraction of many weak beliefs vs. one strong belief? The deductive closure of a DCBS gives it a decided advantage over the DOBS, which does not have access to its implicit beliefs. All a DOBS can do is minimize the loss of what information it does have.

This paper focuses on constraints 1 and 2. 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, so we need to define some DOBS terms that can be used to rewrite constraint 2 for a DOBS.

$+$	**Expansion:** Addition of a proposition, p, to a belief space, **K**, such that $\mathbf{K} + p = \mathrm{Cn}(\mathbf{K} \cup \{p\})$
$\dot{-}$	**Contraction:** Retraction of a proposition, p, from a belief space, **K**, such that $\mathbf{K} \dot{-} p \nvdash p$
$*$	**Revision:** Consistent addition of a proposition, p, to a belief space, **K**, such that $\mathbf{K}*p$ is consistent, (equivalent to $\mathrm{Cn}((\mathbf{K} \dot{-} \sim p) \cup \{p\})$.

Figure 1: Table representing the AGM belief change operations.

When discussing deductive closure, we use the AGM definition of a consequence operation, Cn, where Cn(**K**) denotes the deductive closure of a belief space, **K**, and **K** is a deductively closed belief space (DCBS) under Cn if $\mathbf{K} = \mathrm{Cn}(\mathbf{K})$ (Alchourron, Gärdenfors, and Makinson 1985). The AGM belief change operations are shown in Figure 1.

Why a Deductively Open Belief Space (DOBS)

The integrity constraints and belief revision operations mentioned above assume a deductively closed belief space (DCBS) within some language $\boldsymbol{L}$. For classical propositional logic, this is an infinite belief space. Even if only one of every set of logically equivalent propositions is included, to make the belief space finite, its size is on the order of 2^{2^n} sentences, where n = the number of atomic propositions in the language $\boldsymbol{L}$ (e.g. over 4 trillion sentences if n = 5). This makes implementation computationally hard if not impossible.

Forming postulates about how a deductively closed belief space would be altered by the various belief change operations is helpful in establishing theoretical guidelines for belief revision. To compare how well these postulates are satisfied by an *implemented* belief revision system, however, they must be altered to fit the broad constraints of an implemented system: it must function within a finite (and reasonable) amount of time and use a finite memory and reasoning space. Since even a finite belief space that is deductively closed can become unmanageable, implementations must consider using a Deductively Open Belief Space (DOBS).

Defining the DOBS

A DOBS, by definition, is a belief space that is not guaranteed to contain ***all*** the possible inferences from the beliefs it holds (only some subset of them) or to know ***all*** the possible ways that its beliefs can be derived (only those derivations that it has already performed). A DOBS begins as an empty set to which hypotheses can be added. The beliefs derived from those hypotheses, however, are added gradually over time, as it considers them and discovers them derivable – not all at once.

The entire Belief State is represented by a knowledge base, KB. The DOBS is the Belief Space of the knowledge base, BS(KB). Given a propositional language $\boldsymbol{L}$, consisting of all the well-formed propositions formed from some set of proposition letters, a belief state is defined as:

$\mathbf{KB} =_{def} \langle \mathbf{HYPS, DERS, B, A, J} \rangle$ where

$\mathbf{HYPS} \subseteq \boldsymbol{L}$, $\mathbf{DERS} \subseteq \mathrm{Cn}(\mathbf{HYPS})$,$\mathbf{B} \subseteq \mathbf{HYPS}$,

$\mathbf{A} \subseteq \{\langle p,os \rangle\}$, and $\mathbf{J} \subseteq \{\langle q,js \rangle\}$ where

$p \in \mathbf{HYPS} \cup \mathbf{DERS} \wedge$

$q \in \mathbf{DERS} \wedge$

$os \subseteq$ **HYPS** $\wedge$

$js \subseteq$ **HYPS**$\cup$**DERS** $\wedge$

$js \neq \{q\} \wedge$

$os \vdash p \wedge$

$js \vdash_1 q$,

where $js \vdash_1 q$ means that the set *js* derives the proposition *q* by using ***only one*** rule of inference and ***in one step***. Unless otherwise noted, assume that all future examples and definitions are using **KB** = <**HYPS, DERS, B, A, J**> as their starting belief state.

HYPS represents all the hypotheses ever introduced into **KB**. **DERS** represents all the propositions ever derived from **HYPS**, and **A** and **J** are the record of ***just those*** derivations – in the styles of an ATMS (**A**) and a JTMS (**J**). **B** represents the set of currently believed hypotheses. All propositions in the knowledge base are represented in **A** by at least one pair. All propositions in **DERS** are represented in **J** by at least one pair.

Since a DOBS can have propositions that are derivable but not, yet, derived, we introduce the concept of a proposition, *p*, being ***known to be derivable*** from a set of propositions, α. This is denoted as $\alpha \vdash_{\mathbf{KB}} p$ and is defined by the rules below.

1. A hypothesis is known to derive itself:

   $p \in$ **HYPS** $\Rightarrow \{p\} \vdash_{\mathbf{KB}} p$.
2. A justification set, *js*, for a proposition, *p*, is known to derive *p*: $\langle p,js \rangle \in$ **J** $\Rightarrow js \vdash_{\mathbf{KB}} p$.
3. An origin set for a proposition, *p*, is known to derive *p*: $\langle p,os \rangle \in$ **A** $\Rightarrow os \vdash_{\mathbf{KB}} p$.
4. $\vdash_{\mathbf{KB}}$ is transitive:

   $\forall q[(q \in \beta \wedge \alpha \vdash_{\mathbf{KB}} q)] \wedge \beta \vdash_{\mathbf{KB}} p \Rightarrow \alpha \vdash_{\mathbf{KB}} p$
5. A superset of a set that derives a proposition also derives that proposition:

   $\alpha \vdash_{\mathbf{KB}} p \wedge \alpha \subseteq \beta \Rightarrow \beta \vdash_{\mathbf{KB}} p$.

A proposition *p* can be an element of both **HYPS** and **DERS** if it is ***both*** asserted as a hypothesis ***and*** known to be derivable from some $os \subseteq$ **HYPS** where $p \notin os$.

D is the set of derived propositions that are currently believed, and **BS**(**KB**) is the set of all currently believed propositions (the DOBS):

$$\mathbf{D}(\mathbf{KB}) =_{def} \{\alpha \mid \alpha \in \mathbf{DERS} \wedge \mathbf{B} \vdash_{\mathbf{KB}} \alpha\}$$

$$\mathbf{BS}(\mathbf{KB}) =_{def} \mathbf{B} \cup \mathbf{D}$$

In other words, **KB** represents all the propositions that exist in the system along with a record of how they were derived, and **BS**(**KB**) represents only those propositions that are currently believed. Although a DCBS "forgets" what it no longer believes, its "omniscient" deductive closure allows it to instantly remember anything that is re-believed. The DOBS must keep track of the disbelieved propositions and derivations to avoid having to repeat earlier derivations when disbelieved propositions are returned to the belief space. The diagram in Figure 2, below, shows most of these concepts.

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

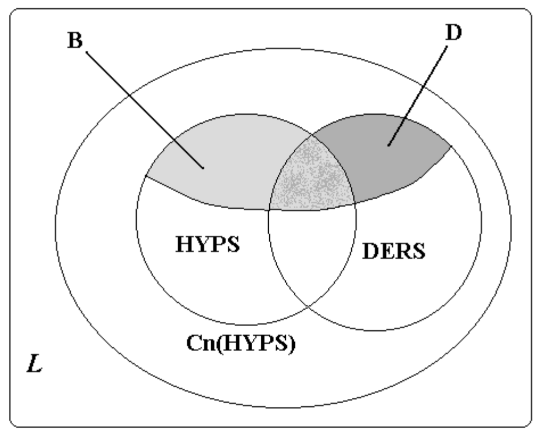

Figure 2. The propositions of a **KB** are the area within the two circles marked **HYPS** and **DERS**. **B** is the lighter shaded area, including the double-shaded intersection. **D** is the darker shaded area, including the double-shaded intersection. The double-shaded intersection represents believed propositions, which have been *both* asserted as hypotheses *and* derived from asserted hypotheses. The belief space represented here, **BS**(**KB**), is the entire shaded region.

KB-Closure and K-consistency

Because we are removing the omniscience of a DCBS and its consequence operation, we must "remember" as much as possible about our DOBS, including propositions that we no longer believe. Once a base set of hypotheses, **B**, is chosen, the closure of **B** is limited by **KB** (i.e. by its derivation records in **A** and **J**). We call this closure $Cn_{\mathbf{KB}}$, and it is defined below:

$$Cn_{\mathbf{KB}}(\mathbf{B}) =_{def} \{\alpha \mid \mathbf{B} \vdash_{\mathbf{KB}} \alpha\} = \mathbf{BS}(\mathbf{KB})$$

A DOBS is inconsistent if a contradiction has been *or can be* derived from its beliefs. A DOBS, **BS**(**KB**) or $Cn_{\mathbf{KB}}(\mathbf{B})$, is ***k-inconsistent*** IFF $\exists p[\, p \in Cn_{\mathbf{KB}}(\mathbf{B}) \wedge \sim p \in Cn_{\mathbf{KB}}(\mathbf{B})]$. If a DOBS is *not* k-inconsistent, then we will call it ***k-consistent*** – there are no inconsistencies in $Cn_{\mathbf{KB}}(\mathbf{B})$, it is *not known to be inconsistent*. This means a DOBS can be both inconsistent and k-consistent at the same time: For example, **B**={A, P, P⇒~A}, but ~A has not, yet, been derived.

It has been suggested that this is unacceptable, causing a lack of confidence in the information provided by the DOBS system. There are no certainties in a real world implementation, and you can never know for sure if you have correct information… only a system's best guess. Our goal is to try to make that best guess as good as possible.

It is always the case that, for a given **KB** = <**HYPS, DERS, B, A, J**>, $A \in \mathbf{D}$ IFF $\mathbf{B} \vdash_{\mathbf{KB}} A$, therefore $\mathbf{BS} \vdash A$ IFF $\mathbf{B} \vdash A$, and likewise for $\vdash_{\mathbf{KB}}$. If $\exists$ **B,B′** s.t. $\mathbf{B} \subseteq \mathbf{B'}$, then 1) if **B** is k-inconsistent, then **B′** is k-inconsistent, and 2) if **B′** is k-consistent, then **B** is k-consistent.

We distinguish the propositions in **HYPS** from those in **DERS** to allow a foundations approach to belief revision. A coherence approach can be implemented by storing all propositions as hypotheses in **HYPS** while retaining their derivation history for later revision needs.

Operations on a DOBS

The operations on a DOBS Belief State, **KB** = <**HYPS, DERS, B, A, J**>, are KB-Closure (described above), expansion, contraction, revision and query. The belief change operations are functions that take a Knowledge Base and a proposition and return an altered Knowledge Base. They are:

1. **+ Expansion**
   KB+*A* is the addition of a belief *A* to **KB** by adding *A* as a hypothesis to **B**, and therefore to **HYPS**, and can be specifically referred to as ***hypothesis-addition***. The result is a new **KB′ = <HYPS′, DERS, B′, A′, J>** s.t. $\mathbf{HYPS'} = \mathbf{HYPS} \cup \{A\}$, $\mathbf{B'} = \mathbf{B} \cup \{A\}$, and $\mathbf{A'} = \mathbf{A} \cup \{<A,\{A\}>\}$. It is possible for *A* to already be an element of **HYPS** and even **B**.
2. **∸ Contraction**
   KB∸*A* is the retraction of a belief *A* from **KB** by retracting some of the elements of **B** that are known to derive *A* in order to form a new **KB′ = <HYPS, DERS, B′, A, J>** s.t. ***not***$(\mathbf{B'} \vdash_{\mathbf{KB'}} A)$ -- i.e. $A \notin \mathbf{BS'}$. It is possible for *A* to not be an element of **BS**.
3. *** Revision**
   KB**A* is the ***consistent*** addition of *A* to **B**. Insisting on consistency requires that ~*A* be retracted from **BS**. The result is a new **KB′ = <HYPS′, DERS, B′, A′, J>** s.t. $\mathbf{HYPS'} = \mathbf{HYPS} \cup \{A\}$, $\mathbf{A'} = \mathbf{A} \cup \{<A,\{A\}>\}$, $A \in \mathbf{B'}$, $\mathbf{B'} \subseteq \mathbf{B} \cup \{A\}$, and **B′** is k-consistent.

Expanding the DOBS using Query

As explained before, the system builds its explicit beliefs as new propositions are considered and derived from the current belief space, $Cn_{\mathbf{KB}}(\mathbf{B})$. This happens through the query process – when the system is queried about whether a sentence *A* can be derived from the Belief State, **KB** (i.e. $\mathbf{B} \vdash_{\mathbf{KB}}^{?} A$).

The query **Q**(*A*, **KB**) is a way to ask the knowledge base **KB** whether a proposition *A* is derivable from its existing beliefs. The query process proceeds in two stages:

1. If *A* is already present in the belief space, there is no change to the knowledge state.
2. If *A* is recognized as an axiom (using axiom templates that are part of the system) it is added to **HYPS** and *B* and the pair < *A*, $\varnothing$ > is added to **J**. This is axiom addition; and, with an empty justification set, *A* will remain in the belief space from now on.
3. If *A* is not present as a belief, the query process identifies some α where $\alpha \vdash_{\mathbf{KB}} A$ using one rule of inference and only one step. This is done by finding an implication with A as a consequent. The antecedents are used to form α. The system then recursively queries for the elements in α. This is called back-chaining.
4. Steps 1, 2, and 3 are repeated for all queries, until either *A* is derived or queries for all possible sets of α have failed.

Although there might be more than one way to derive *A* from **KB**, finding any single derivation of *A* that is supported by the current beliefs – even a pre-existing one – should end the query successfully. The query fails if it cannot derive *A*. In a complete system, if $\mathbf{B} \vdash A$, then the query process is guaranteed to derive *A*. In an incomplete system, even though $\mathbf{B} \vdash A$, the query process might not derive *A*. An implemented system might be incomplete if it has to impose restrictions on the query process (e.g. to eliminate long or computationally expensive derivation attempts, satisfy time restrictions, etc.).

Query-addition ($+_{\mathbf{Q}}$)

If a one-step query, **Q**(p,**KB**), is successful, a new proposition, p, is added by query-addition ($+_{\mathbf{Q}}$) to **DERS**. The actual query-addition refers to the method of adding pairs to **A** and **J** , thus altering them to form an **A′** and **J′**.

For the query to be successful (in one step), it had to find some js s.t. $js \vdash_1 p \ \land\ js \subseteq \mathbf{BS}$. The tuple $\langle p, js\rangle$ is then query-added to **J**: $\mathbf{J} +_{\mathbf{Q}} \langle p, js\rangle =_{\text{def}} (\mathbf{J} \cup \{\langle p, js\rangle\})$

Warning: It is possible to store this derivation information in **A**, as a $\langle p,os\rangle$ pair, by forming an origin set, os, for p from the union of the origin sets for each of the elements in js. For example: $js = \{q, q \Rightarrow p\}$, where the origin sets for q and $q \Rightarrow p$ are $\{q \land p\}$ and $\{q \Rightarrow p\}$, respectively. The os for p would then be $\{q \land p, q \Rightarrow p\}$. But there are three things to beware of:

1. If an element of js has more than one origin set, then there will be multiple origin sets for p. (e.g., if q also has the origin set $\{q \land s\}$, then p would have a second os: $\{q \land s, q \Rightarrow p\}$.
2. To avoid duplication and foster minimality of the origin sets, use the following definition:
   $\mathbf{A} +_{\mathbf{Q}} \langle p, os\rangle =_{\text{def}}$
   $(\mathbf{A} \cup \{\langle p, os\rangle\}) - \{\langle p, os_j\rangle\}$, for all os_j s.t. $os \subset os_j$
   This is to guarantee minimality and a lack of duplication of the origin set, os, for any given proposition in **DERS**: i.e.
   $\forall \langle p, \alpha\rangle\ [\langle p, \alpha\rangle \in \mathbf{A} \Rightarrow \nexists \beta[\langle p, \beta\rangle \in \mathbf{A} \land \beta \subset \alpha\ ]\ ]$.
   Continuing the above example: if p is later derived directly from $\{q \land p\}$, its new os would be $\{q \land p\}$, which would replace the os $\{q \land p, q \Rightarrow p\}$.
3. If you use **A** exclusively, you need to consider your algorithms carefully or your KB-closure might be a subset of the KB-closure when js was also used. Continuing our example: After retracting $q \land p$ and $q \land s$, $p,q \notin \mathbf{BS}$ anymore. The addition of q as a hypothesis would restore p to **BS** ***if*** the js $\{q, q \Rightarrow p\}$ was referenced. But, if only **A** was available for derivation records, $p \notin \mathbf{BS}$ until a new query is made.

When a query makes recursive queries, the above process is iterated, continually adjusting **KB** along the way through query-addition. After each query-addition, **BS** is regenerated (using the KB-Closure operation). It is possible, therefore, to query-add derivable propositions to **KB** (and, therefore, **BS**) without ever deriving the initial proposition that was queried for, A.

Query Postulates

Whether it succeeds or fails, the query, **Q**(A,**KB**), is a function **Q** that takes a proposition A and a knowledge base **KB** = <**HYPS,DERS,B,A,J**> as arguments. It returns an altered knowledge base **KB′** = <**HYPS, DERS′, B, A′, J′**> that has the following properties (where **BS** and **BS′** are used to represent **BS**(**KB**) and **BS**(**KB′**) respectively):

- Q0) If **KB** is a knowledge base, then **Q**(A,**KB**) is a knowledge base
- Q1) If $A \in \mathbf{BS}$, then $\mathbf{KB'} = \mathbf{KB}$
- Q2†) If $\mathbf{B} \vdash A$, then $A \in \mathbf{BS'}$
- Q3) If $\mathbf{B} \nvdash A$, then $A \notin \mathbf{BS'}$
- Q4‡) If **B** is k-inconsistent and $A \notin \mathbf{BS}$, then $\mathbf{DERS'} - \mathbf{DERS} = \mathbf{BS'} - \mathbf{BS} = \{A\}$ (since anything follows from a contradiction)
- Q5) If $\mathbf{KB'} \neq \mathbf{KB}$ and $A \in \mathbf{KB'}$, then $\exists p, \beta$ s.t. $\langle p,\beta\rangle \in \mathbf{J'}$, $\beta \subseteq \mathbf{BS}$, and either $p = A$, or p was derived in a successful attempt to derive A
- Q6) If $\mathbf{KB'} \neq \mathbf{KB}$ and $A \in \mathbf{KB'}$, then $\exists p, \beta$ s.t. $\langle p,\beta\rangle \in \mathbf{J'}$, $\beta \subseteq \mathbf{BS}$, and p was derived in an unsuccessful attempt to derive A

† Q2 – deriving A is not guaranteed if the system is incomplete.

‡ Q4 does not apply if the logic is paraconsistent.

The initial **KB**, $\mathbf{KB_0}$, is the tuple $\langle\emptyset,\emptyset,\emptyset,\emptyset,\emptyset\rangle$. Note that **HYPS**, **DERS**, **A**, and **J** increase monotonically with the exception of the recommended minimality constraint of query-addition to **A**.

Integrity Constraints for a DOBS

Now that we have formalized a DOBS, we can assess the key changes necessary to adjust the list of integrity constraints (Gärdenfors and Rott 1995) so that they can be used as belief revision guidelines for a DOBS. Alterations are in ***bold italics***. Additions or clarifications are in *plain italics*:

1. A knowledge base should be kept ***k-consistent*** whenever possible.
2. If a proposition can be derived from the beliefs in the knowledge base *using the derivations currently known*, then it should be included in that knowledge base (***kb-closure***). *Likewise, if a proposition is not in the knowledge base, but can be derived from the beliefs in the knowledge base, it should be produced if queried for.*

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.

Constraint 1 suggests that a system should activate belief revision as soon as an inconsistency is detected. Constraint 2 recommends that we avoid re-deriving a proposition from a set of propositions from which it has already been derived. It furthermore suggests that we strive for a system that is as complete as possible, so that, from a set of hypotheses, the user can expect to derive any proposition that would be included in the deductive closure of that set.

Constraint 3 reminds us that we are forced to analyze the system based on the knowledge we "know," a subset of the total knowledge that a DCBS has. Fortunately, this means there is less information to analyze during a retraction and, since we queried to get this information, it is more likely to be of interest to us. A might bother us with belief revision decisions about obscure information than our query-generated DOBS. However, more information does imply better choices about how to minimize information loss, so we return to the issue of constraint 2 and the need to build our belief space as quickly as possible.

Lastly, constraint 4 seems to need no adjustment for compliance with the needs and restrictions of a DOBS. A DCBS (with its deductive closure) might make a connection between some seemingly unimportant proposition and some important information, whereas the limited knowledge of a DOBS might not have made (or derived) that connection. In this sense, a DOBS is still less "reliable" than the theoretical DCBS. All the more reason to focus on constraint 2 – attempting to improve the quality of completeness – in hopes that we can query for those important pieces of information and still make well-informed choices.

Discussion and Future Work

We have analyzed the concepts of a DOBS and presented a formalism that is flexible enough to be useful to both the coherence theory researchers as well as those working in the foundations camp. The term k-consistent enables a more direct reference to the DOBS state of "not knowing whether a DOBS is consistent." The detailed formalism offers a guideline for retaining derivation information for a DOBS, using both the ATMS and/or JTMS style; and the DOBS integrity constraints offer implementers basic concepts for optimizing their belief revision systems.

The next step in this research is to formulate belief revision postulates specific to a DOBS, using the altered integrity constraints. We hope to offer a DOBS version of not only the AGM postulates, but also Hansson's base contraction postulates (Hansson, 1993) and postulates proposed for ranked beliefs. We also hope to continue to provide brief comments regarding postulate adherence for paraconsistent logics and/or incomplete systems. Implementers will, then, be able to analyze how well their systems meet the standards of the postulates.This is a "pressing issue" (Hansson 1999, p. 367) for those doing belief revision research in Computer Science.

References

Alchourron, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic,* Vol 20, Num. 2, 510-530.

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

Gärdenfors, P. 1992. *Belief Revision*. Cambridge Computer Tracks. Cambridge: Cambridge University Press

Gärdenfors, P.; and Rott, H. 1995. Belief Revision. In: Gabbay, D.; Hogger, C. J.; and Robinson, J.A. eds. *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol 4, Epistemic and Temporal Reasoning. Oxford: Clarendon Press. 35-131.

Hansson, S. O. 1999. *A Textbook of Belief Dynamics: Theory Change and Database Updating*, Vol 11 , 'Applied Logic Series' of Kluwer Academic Publishers, Dordrecht.

Hansson, S.O. 1993. Reversing the Levi Identity. *Journal of Philosophical Logic* 22:637-669

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

Martins, J. P. 1992a. Belief Revision. In Shapiro, S. C. ed Encyclopedia of Artificial Intelligence, 110-116. New York: John Wiley & Sons, second edition

Martins, J. P. 1992b. Truth Maintenance Systems. In Shapiro, S. C. ed Encyclopedia of Artificial Intelligence, 1613-1622. New York: John Wiley & Sons, second edition