Accommodative Belief Revision*

Satu Eloranta[1], Raul Hakli[2], Olli Niinivaara[1], and Matti Nykänen[3]

[1] Department of Computer Science
P.O. Box 68 (Gustaf Hällströmin katu 2b)
FIN-00014 University of Helsinki, Finland
[2] Department of Philosophy
P.O. Box 9 (Siltavuorenpenger 20 A)
FIN-00014 University of Helsinki, Finland
`firstname.lastname@cs.helsinki.fi`
[3] University of Kuopio
Department of Computer Science
P.O. Box 1627
FIN-70211 Kuopio, Finland
`matti.nykanen@cs.uku.fi`

Abstract. Accommodative revision is a novel method of non-prioritized belief revision. The epistemic state of an agent contains both knowledge that is immune to revision and beliefs that are allowed to change. Incoming information is first revised by the knowledge of the agent, and then the epistemic state of the agent is revised using this modified input. The properties of the method are studied and examples of its use are given.

Key words: belief change, belief revision, non-prioritized belief revision, integrity constraints, knowledge

1 Introduction

In belief revision, an agent obtains new information about a static world. On one hand, the input may be considered as the most recent and as such the most reliable piece of information. In that case, if the new information contradicts the beliefs of the agent, it needs to give up some of the old beliefs in order to maintain consistency of beliefs [1]. However, this framework, called *prioritized belief revision*, allows even self-contradictory input to be accepted into the beliefs of the agent.

On the other hand, in *non-prioritized belief revision* (see [16] for a survey) the input is not necessarily accepted. The agent may have some information that it will refuse to give up at any situation. In computer science such information might be called *integrity constraints* [22], in philosophy *knowledge* [17]. In belief revision literature the term *core beliefs* has also been used [16].

Instead of rejecting the input that the agent knows to be impossible, we aim to find a charitable interpretation that retains as much as possible of the input. For instance, suppose that we hear that "Jaakko Kuusisto, a winner of the Sibelius violin contest, gives a concert at the forthcoming open air music festival". However, we know for a fact that although Jaakko has participated in the contest as well, it is actually his brother Pekka who has managed to win it. Our natural reaction would be to think that "the speaker must have got either the first name or the bit about the contest victory wrong. But there will be a concert by either of the two brothers, that much I can believe now".

The amount of information obtained from an "unbelievable input" may vary. Let us consider a modification of an example by Hansson [15]. Amy tells the agent that she saw a three-toed woodpecker with a red forehead and a red rump just outside her window. When it comes to birds, the agent has more knowledge than Amy: The agent knows that a three-toed woodpecker neither has a red forehead nor a red rump. The agent has various possibilities when making sense of the impossible statement. With some benevolence, it can come to one of the following conclusions: (1) Amy saw a bird with a red forehead and a red rump, but it was not a three-toed woodpecker, (2) Amy saw a bird with a red forehead and a red rump or Amy saw a three-toed woodpecker (but not one with a red forehead or a red rump), or (3) at least Amy saw some kind of a bird outside her window.

In this paper, we introduce *accommodative revision*, a method for non-prioritized belief revision. The basic idea is to use knowledge as a filter that the incoming information has to pass through before the epistemic state can be revised. The agent will modify the input to accord with its knowledge.

Our proposal has the following properties: (1) input inconsistent with knowledge will not be accepted, but the input will be modified to produce an acceptable formula prior to revising the epistemic state, (2) only knowledge is used to modify the input, and (3) the modification of the input and the revision of the epistemic state are performed as two separate phases. We will also describe an implementation that is publicly available to allow small-scale experimentation of our proposal[4].

In our two-phase revision, at first the input is revised by the knowledge of the agent, giving a new input formula. Then the epistemic state of the agent is revised by the new formula. Thus our proposal is closely related to selective revision [12] and might be considered as a generalization of the revision method presented in [2] (see Sect. 3 for comparison). Other, more recent non-prioritized belief revision proposals based on the modification of input sentence do also exist, e.g. [5], [20] and [3], but they are based on syntactic manipulation whereas our method uses previously defined (semantically-oriented) belief-revision operators already known to satisfy certain principles.

The outline of the paper is as follows. Section 2 recalls the basic ideas of belief revision. In Sect. 3 we give the definition of our method, and in Sect. 4 we study its properties. In Sect. 5 we introduce an implementation of accommodative revision with various sample operators. In Sect. 6 we analyze some examples. Section 7 is devoted for conclusions.

---

[4] `http://www.cs.helsinki.fi/group/protean/abr/`

2 Preliminaries

In belief revision, an agent evolves its epistemic state due to incoming information called epistemic input. At first the input is classified, and the way the epistemic state will be changed depends on the result of the classification. On the meta level, the change is guarded by rationality criteria. Alchourrón, Gärdenfors and Makinson [1] have proposed a set of principles for belief revision known as the *AGM-postulates*. Darwiche and Pearl [7,8] have proposed an additional set of postulates for belief revision in order to rule out change operators that give unintuitive results in iterated revisions. The main principles in these sets of postulates are maintaining consistency of the beliefs and minimality of change.

We assume that the set of propositional formulas believed in any epistemic state is deductive closed. For each epistemic state T, let T_B denote the *belief set* of the state, that is, the deductive closure of formulas believed in the state. Let $\circ$ denote a revision operator, that is, a function from epistemic states and propositional input formulas into epistemic states. The AGM-postulates and the DP-postulates are rephrased here as follows:

(R1): $A \in (T \circ A)_B$.
(R2): If $\neg A \notin T_B$, then $(T \circ A)_B = T_B + A$.
(R3): If $A \not\models \perp$, then $\perp \notin (T \circ A)_B$.
(R4): If $A \equiv B$, then $(T \circ A)_B = (T \circ B)_B$.
(R5): $(T \circ (A \wedge B))_B \subseteq (T \circ A)_B + B$.
(R6): If $\neg B \notin (T \circ A)_B$, then $(T \circ A)_B + B \subseteq (T \circ (A \wedge B))_B$.
(DP1): If $A \models B$, then $((T \circ B) \circ A)_B = (T \circ A)_B$.
(DP2): If $A \models \neg B$, then $((T \circ B) \circ A)_B = (T \circ A)_B$.
(DP3): If $A \in (T \circ B)_B$, then $A \in ((T \circ A) \circ B)_B$.
(DP4): If $\neg A \notin (T \circ B)_B$, then $\neg A \notin ((T \circ A) \circ B)_B$.

Here A and B denote propositional (input) formulas, and $T_B + A$ denotes the deductive closure of the set $T_B \cup \{A\}$. In the AGM-postulates (R1)–(R6), the epistemic input is always prioritized over the old beliefs due to postulate (R1). Postulates (R1)–(R4) are considered basic: every (prioritized) belief-revision operator should satisfy them. Postulates (R5)–(R6) are supplementary. Note that only belief sets of epistemic states are used in the formulation: Because we do not make other assumptions about the representation of epistemic states, we do not use equality nor equivalence of epistemic states in the formulation.

If epistemic states were functionally dependent on the belief sets in the states, then the joint set of postulates would result in triviality of logic, that is, no three satisfiable but pairwise inconsistent formulas could exist [13,10]. Ruling out inconsistent epistemic states and self-contradictory epistemic input does not solve the problem [10,11]. Thus for the joint set of postulates, more elaborate epistemic states are needed.

As known [14], belief revision involves ordering among possible worlds. Those possible worlds that are minimal in the ordering are the most plausible worlds (the doxastic alternatives) in the state. The possible worlds modelling the new formula that are minimal in the ordering will be the most plausible worlds in the revised state. Spohn [24] has argued that this ordering should be part of the epistemic state, because it is altered in the process of revision.

To represent epistemic states, Spohn [24] introduced *ranking functions* (alias Ordinal Conditional Functions, OCFs), which are functions from the set of possible worlds into ordinals. The ordinal of a world is its rank. The smaller the rank, the less disbelieved is the world. The most plausible worlds (the doxastic alternatives) are the worlds with rank 0. The rank of a proposition (a set of possible worlds) is the minimum of the ranks of the worlds within it. Spohn rules out both inconsistent epistemic states and self-contradictory epistemic input. To update the ranking, Spohn [24] also introduced a method, in which the rankings are shifted. Darwiche and Pearl [7] introduced a belief-revision operator based on Spohn's framework to accommodate the principles for iterated belief revision. We, too, shall adopt and elaborate on Spohn's framework in Sect. 5.1.

We will treat knowledge in belief change as integrity constraints have been considered in theory change. *Integrity constraints* are used to express those properties that should always hold. When defining the effect that the integrity constraints have on belief change, Katsuno and Mendelzon [18] have required that the result entails the integrity constraints. Then, with any belief-revision operator satisfying the AGM-postulates, the integrity constraints actively take part in belief revision.

Let us consider the definition by Katsuno and Mendelzon [18] in more detail. Using T to denote an epistemic state, A to denote an epistemic input, $\circ$ to denote a belief revision operator, and IC to denote a propositional formula expressing integrity constraints, they defined the effect of integrity constraints on belief revision as

$$T \circ^{IC} A =_{def} T \circ (A \wedge IC). \qquad (1)$$

We will adopt this definition and develop it further.

3 The Principle of Accommodative Revision

Before introducing accommodative revision, let us specify our framework. For each epistemic state T, let T_B denote the belief set of the state as in Sect. 2, and let T_K denote the set of all the propositional formulas constituting the knowledge in the state. Both sets are deductively closed, thus all the tautologies will be included in them. We shall restrict our accommodative revision only to those epistemic states in which the following static constraints are satisfied:

(S1): $T_K \subseteq T_B$.
(S2): $\perp \notin T_K$.

Condition (S1) says that the agent believes what it knows, and condition (S2) says that the knowledge set does not include contradictions.

However, we do not assume that the sets T_B and T_K constitute the epistemic state. In fact, Sect. 5.1 uses Spohn's [24] ranking functions to represent epistemic states, yet the general method is independent of the representation.

We shall make a strong assumption that the set T_K can be represented by a propositional formula. This assumption can be justified when assuming that the knowledge in the epistemic state is obtained in a finite sequence of monotonous knowledge expansions starting from a state in which nothing (except tautologies) was known. Our way of meeting this assumption will be given in Sect. 5.1.

We extend Definition (1) as follows. Let T denote an epistemic state, let K denote a propositional formula representing the set T_K, and let A denote a propositional input formula. We define the *accommodative revision* of the state T by the formula A as

$$T \otimes A =_{def} T \circ (A * K) \tag{2}$$

in which $\otimes$ denotes an accommodative revision operator, $\circ$ denotes belief revision of epistemic states, and $*$ denotes revision of propositional formulas. Note that we do not propose a single operator but a scheme in which various operators can be applied.

The philosophical justification of our method is the following: We consider the modified input as an estimate of the formula that the source of the input would have given, had it had all the knowledge that our agent has.

Let us compare our proposal with some related work. As a special case of accommodative revision, we will get screened revision [19] by defining that $A * K \equiv K$ whenever $\neg A \in T_K$. Our proposal resembles the proposal by Bellot et al. [2], which also has the same two phases, but uses the same fixed distance-based revision operator to update both the input by the knowledge and the epistemic state. Our proposal lets the agent choose the two components separately, without imposing limitations on the representation of epistemic states. Thus our proposal is a generalization of theirs.

In selective revision [12], some function is to be used to replace the input by a new formula that is typically entailed by the original input. In our proposal, not only is the complement of the knowledge contracted from the input, but the knowledge is incorporated into the input. Without the latter, our proposal could be considered as an instantiation of selective revision.

4 Features of Accommodative Revision

Let us analyze some features of our accommodative revision. We want to prove that the operators in the family of accommodative revision accomplish non-prioritized belief revision satisfying the AGM postulates (R2)–(R4) and a modification of (R1). We first make some assumptions of the operators $*$ and $\circ$ used as components in accommodative revision.

We assume that the modification function $*$ is a function from pairs of propositional formulas into propositional formulas and it satisfies at least the basic AGM-postulates rephrased here for this framework as follows: (MR1): $A * K \models K$, (MR2): If $K \not\models \neg A$, then $A * K \equiv A \wedge K$, (MR3): If $K \not\models \perp$, then $A * K \not\models \perp$, (MR4): If $A \equiv A'$ and $K \equiv K'$, then $A * K \equiv A' * K'$. Here A, A', K and K' denote propositional formulas.

We assume that $\circ$ is a function from epistemic states and propositional formulas into epistemic states. We assume that the operator $\circ$ satisfies at least the basic AGM-postulates (R1)–(R4). We shall also use the following extra condition to ensure that $\circ$ does not change the knowledge in the state:

(R0): $\quad (T \circ A)_K = T_K.$

We shall restrict our accommodative revision only to those epistemic states in which the static constraints (S1) and (S2) from Sect. 3 are satisfied.

Now, using these assumptions, we can prove that an accommodative-revision operator $\otimes$ preserves these static constraints and has the following features:

(AR0): $\quad (T \otimes A)_K = T_K.$
(AR1): $\quad$ If $\neg A \notin T_K$, then $A \in (T \otimes A)_B$.
(AR2): $\quad$ If $\neg A \notin T_B$, then $(T \otimes A)_B = T_B + A$.
(AR3): $\quad \perp \notin (T \otimes A)_B$.
(AR4): $\quad$ If $A \equiv A'$, then $(T \otimes A)_B = (T \otimes A')_B$.

(AR0), (AR2), and (AR4) are equivalent to postulates (R0), (R2), and (R4) correspondingly. (AR3) is stronger than (R3): It says that accommodative revision always results in epistemic states with non-contradictory belief sets. According to (AR1), the new formula is accepted, if it is not contradictory to the knowledge in the state. Note that all these are derived properties, not principles set for accommodative revision.

We can see that accommodative revision fails to satisfy the basic AGM-postulates only when the input is contradictory to the knowledge in the state: In those cases the success postulate (R1) fails. Thus accommodative revision is non-prioritized belief revision. Accommodative revision preserves the static constraints, and it is able to guarantee non-contradictory belief sets at all occasions.

Theorem 1. *If the operators $*$ and $\circ$ used as components in an accommodative revision operator $\otimes$ satisfy postulates (MR1)–(MR4) and (R0)–(R4) respectively, then given any epistemic state T satisfying constraints (S1) and (S2), then the state $T \otimes A$ satisfies (S1), (S2), and (AR0)–(AR4).*

Proof. Let T denote an epistemic state, A denote a propositional formula, and let K denote a propositional formula equivalent to T_K. We assume that $T_K \subseteq T_B$ and $\perp \notin T_K$. By definition, $(T \otimes A) = (T \circ (A * K))$.

By (R0), $(T \otimes A)_K = (T \circ (A * K))_K = T_K$, thus (AR0) holds. Then because $\perp \notin T_K$, $\perp \notin (T \otimes A)_K$ and (S2) holds. Because by (R1), $A * K \in (T \circ (A * K))_B = (T \otimes A)_B$, (MR1) gives us $A * K \models K$, thus $K \in (T \otimes A)_B$. Then by (AR0), (S1) holds.

To prove (AR1), let us assume $\neg A \notin T_K$. Then by (MR2), $A * K \equiv A \wedge K$, and (R4) gives us $(T \circ (A * K))_B = (T \circ (A \wedge K))_B$. Thus by (R1), $A \wedge K \in (T \circ (A \wedge K))_B = (T \circ (A * K))_B = (T \otimes A)_B$ and then $A \in (T \otimes A)_B$.

To prove (AR2), assume $\neg A \notin T_B$. Then by (S1), $\neg A \notin T_K$, and (MR2) gives us $A * K \equiv A \wedge K$. Then by (R4), $(T \circ (A * K))_B = (T \circ (A \wedge K))_B$. For contradiction, assume $\neg(A \wedge K) \in T_B$. Because $\neg(A \wedge K) \equiv \neg A \vee \neg K$ and

$K \in T_B$, we get $\neg A \in T_B$, a contradiction. Thus $\neg(A \wedge K) \notin T_B$. Then (R2) gives us $(T \circ (A \wedge K))_B = T_B + A \wedge K$. Because $K \in T_B$, $T_B + A \wedge K = T_B + A$. Thus $(T \otimes A)_B = (T \circ (A * K))_B = T_B + A$.

Let us next prove that (AR3) holds. Because $K \not\models \perp$, (MR3) gives us $A * K \not\models \perp$. Then by (R3), $\perp \notin (T \circ (A * K))_B = (T \otimes A)_B$.

To prove (AR4), assume $A \equiv A'$. Then by (MR4), $A * K \equiv A' * K$, thus (R4) gives us $(T \otimes A)_B = (T \circ (A * K))_B = (T \circ (A' * K))_B = (T \otimes A')_B$. □

Whenever input does not contradict knowledge, accommodative revision retains the properties of the revision operator $\circ$ used in definition.

Theorem 2. *If the operators $*$ and $\circ$ satisfy postulates (MR1)–(MR4) and (R0)–(R4) respectively, then for each postulate (R5), (R6), (DP1)–(DP4), if the operator $\circ$ satisfies the postulate in question, accommodative revision also has the corresponding property below:*

(AR5): If $\neg(A \wedge B) \notin T_K$, then $(T \otimes (A \wedge B))_B \subseteq (T \otimes A)_B + B$.
(AR6): If $\neg(A \wedge B) \notin T_K$ and $\neg B \notin (T \otimes A)_B$, then $(T \otimes A)_B + B \subseteq (T \otimes (A \wedge B))_B$.
(AR7): If $A \models B$ and $\neg A \notin T_K$, then $((T \otimes B) \otimes A)_B = (T \otimes A)_B$.
(AR8): If $A \models \neg B$, $\neg A \notin T_K$ and $\neg B \notin T_K$, then $((T \otimes B) \otimes A)_B = (T \otimes A)_B$.
(AR9): If $A \in (T \otimes B)_B$, then $A \in ((T \otimes A) \otimes B)_B$.
(AR10): If $\neg A \notin (T \otimes B)_B$, then $\neg A \notin ((T \otimes A) \otimes B)_B$.

Proof. Assume $\neg(A \wedge B) \notin T_K$. By (S2), $\perp \notin T_K$, thus $\neg A \notin T_K$ and $\neg B \notin T_K$. By (MR2), $(A \wedge B) * K \equiv (A \wedge B \wedge K)$ and $A * K \equiv A \wedge K$. By (R4), $(T \otimes A \wedge B)_B = (T \circ ((A \wedge B) * K))_B = (T \circ (A \wedge B \wedge K))_B$. If $\circ$ satisfies (R5), then by (R5), (R4) and (MR2), $(T \circ (A \wedge B \wedge K))_B \subseteq (T \circ (A \wedge K))_B + B = (T \circ (A * K))_B + B = (T \otimes A)_B$, (AR5) holds. To prove (AR6), assume also that $\neg B \notin (T \otimes A)_B$. Then by definition, $\neg B \notin (T \circ (A \wedge K))_B$. If $\circ$ satisfies (R6), then by (R6), (R4) and (MR2), $(T \otimes A)_B + B = (T \circ (A * K))_B + B = (T \circ (A \wedge K))_B + B \subseteq (T \circ (A \wedge B \wedge K))_B = (T \circ ((A \wedge B) * K))_B = (T \otimes (A \wedge B))_B$, (AR6) holds.

To prove (AR7), assume $\neg A \notin T_K$ and $A \models B$. Then $\neg B \notin T_K$ and $(A \wedge K) \models (B \wedge K)$. If $\circ$ satisfies (DP1), then by (MR2), (R4), and (DP1), $((T \otimes B) \otimes A)_B = ((T \circ (B * K)) \circ (A * K))_B = ((T \circ (B \wedge K)) \circ (A \wedge K))_B = (T \circ (A \wedge K))_B = (T \circ (A * K))_B = (T \otimes A)_B$.

To prove (AR8), assume $\neg A, \neg B \notin T_K$ and $A \models \neg B$. Thus $A \wedge K \models \neg B \wedge K$. If $\circ$ satisfies (DP2), then by (MR2), (R4), and (DP2), $((T \otimes B) \otimes A)_B = ((T \circ (B * K)) \circ (A * K))_B = ((T \circ (B \wedge K)) \circ (A \wedge K))_B = (T \circ (A \wedge K))_B = (T \circ (A * K))_B = (T \otimes A)_B$.

To prove (AR9), let us assume $A \in (T \otimes B)_B$. Then by (AR3), (S1), and (AR0), $\neg A \notin T_K$ and $A \wedge K \in (T \otimes B)_B = (T \circ (B * K))_B$. By (MR2) and (R4) $((T \otimes A) \otimes B)_B = ((T \circ (A * K)) \circ (B * K))_B = ((T \circ (A \wedge K)) \circ (B * K))_B$. If $\circ$ satisfies (DP3), then by (DP3), $A \wedge K \in ((T \circ (A \wedge K)) \circ (B * K))_B = ((T \circ (A * K)) \circ (B * K))_B = ((T \otimes A) \otimes B)_B$.

To prove (AR10), assume $\neg A \notin (T \otimes B)_B$. Then by (S1) and (AR0), $\neg A \notin (T \otimes B)_K = T_K$, and thus by (MR2), $(A * K) \equiv (A \wedge K)$. For contradiction, assume

$\neg(A \wedge K) \in (T \otimes B)_B$. Because $\neg(A \wedge K) \equiv \neg A \vee \neg K$ and $K \in (T \otimes B)_B$, we get $\neg A \in (T \otimes B)_B$, a contradiction. Thus $\neg(A \wedge K) \notin (T \otimes B)_B = (T \circ (B * K))_B$. If $\circ$ satisfies (DP4), then by (DP4) and (R4), $\neg(A \wedge K) \notin ((T \circ (A \wedge K)) \circ (B * K))_B = ((T \circ (A * K)) \circ (B * K))_B = ((T \otimes A) \otimes B)_B$. $\square$

5 An Implementation

5.1 The Underlying Framework

As stated in Sect. 3, we have adopted Spohn's ranking functions (OCFs) κ [24, Definition 4] as our representation for the epistemic state T. However, we restrict their range to the natural numbers augmented with infinity ∞. Intuitively, the *rank* $\kappa(w)$ is the agent's degree of disbelief towards this world w being the actual one. This ranking extends to formulas A as the minimum rank of their models: $\kappa(A) = \min\{\kappa(w) \colon w \in \mathrm{Mod}(A)\}$. Their connection to beliefs is that $T_B = \{A \colon \kappa(\neg A) > 0\}$ and to knowledge that $T_K = \{A \colon \kappa(\neg A) = \infty\}$. The OCF definition has two more requirements: (i) $\kappa^{-1}(0) \neq \emptyset$ so that the lowest rank is normalized to be 0. (ii) The agent must be indifferent towards new vocabulary: if $\kappa(w) \neq \kappa(w')$ then w and w' must disagree on some atomic formula p which it has already encountered in some formula.

At least one of $\kappa(A)$ or $\kappa(\neg A)$ is 0 [24, Theorem 2 (a)]. If they both are, then the agent believes neither A nor $\neg A$. For instance, in the *initial* ranking function κ_0, which is everywhere 0, only the tautologies are known and nothing else is believed.

Spohn's *A, α-conditionalization* [24, Definition 6] constructs from a given OCF κ another OCF

$$\kappa_{A,\alpha}(w) =_{def} \begin{cases} \kappa(w) - \kappa(A) & \text{if } \kappa(w) < \infty \text{ and } w \in \mathrm{Mod}(A) \\ \alpha + (\kappa(w) - \kappa(\neg A)) & \text{if } \kappa(w) < \infty \text{ and } w \notin \mathrm{Mod}(A) \\ \kappa(w) & \text{if } \kappa(w) = \infty \end{cases} \tag{3}$$

where $\mathrm{Mod}(A)$ is ranked to 0 while $\mathrm{Mod}(\neg A)$ is ranked to the given constant $\alpha > 0$ without altering the distances within these two moving parts. However, we must ensure $\kappa(A) < \infty$ to meet requirement (i). Following Darwiche and Pearl [7], we use $\kappa_{A,1}$ as our belief revision operation when A is not believed and κ otherwise.

5.2 The Program

We have implemented the approach taken in Sect. 5.1 as a library of functions in the functional programming language Haskell [21]. This library can be loaded into a Haskell interpreter such as GHCi (see `http://haskell.org/ghc/`) which then provides a text-based environment where the user can experiment with different instantiations of our operator scheme. It offers the following three functions:

$$\texttt{initial} = \kappa_0. \tag{4}$$

$$\texttt{know}\ \kappa\ A = \begin{cases} \kappa_{A,\infty} & \text{if } \kappa(A) < \infty \\ \kappa & \text{otherwise.} \end{cases} \tag{5}$$

$$\texttt{hear}_*\ \kappa\ A = \begin{cases} \kappa & \text{if } \kappa(A) = 0 \\ \kappa_{A,1} & \text{if } 0 < \kappa(A) < \infty \\ \texttt{hear}_*\ \kappa\ (A * K) & \text{otherwise.} \end{cases} \tag{6}$$

Constant (4) is the initial OCF. Requirement (ii) shows how the logical vocabulary can be extended dynamically as needed, so we do not have to give it explicitly. Function (5) expands the knowledge at κ with A, but discards A if $\neg A \in T_K$ already. Function (6) is our revision scheme from Definition (2). Here the higher-order parameter $*$ is the syntactic propositional formula revision operator used. This representation of T_K as a single formula K assumed in Sect. 3 is now justified since the current T_K has developed from the `initial` OCF through a finite sequence of `know` steps.

The main goal in designing the implementation was to allow experimenting with different $*$. It is namely not clear at the outset which one corresponds best to our intuition about "the closest alternative to A which I can believe", as witnessed by the different choices in the woodpecker example in the introduction. Currently the implementation offers the four operators described in Sect. 5.3. Defining additional ones is also straightforward using Haskell.

Compared with the COBA 2.0 system [9], ours is decidedly much narrower in scope: it has neither a graphical user interface (GUI) nor a satisfiability (SAT) solver to support processing any but the smallest knowledge bases. On the other hand, experimentation dictates that our implementation must in turn be able to retain and view several versions for the same knowledge base concurrently in memory, namely the revisions of the same base using different operators $*$. This is conveniently supplied by the underlying Haskell interpreter.

5.3 Modification Operators

We shall next give the definitions of the four semantically-oriented belief-revision operators we implemented as modification operators.

For defining these operators, we define the *difference* $w \triangle w'$ between two worlds $w, w' \in \mathcal{W}$ as the set of the atomic formulas having a different truth value in w and w', that is, $w \triangle w' = (w \setminus w') \cup (w' \setminus w)$. These sets are compared either by using the subset relation or the cardinalities of the sets:

$$\begin{aligned} \mathit{diff}(T, A) &= \min(\{w \triangle w' : w \in \mathrm{Mod}(T), w' \in \mathrm{Mod}(A)\}, \subseteq), \\ \mathit{dist}(T, A) &= \min(\{|w \triangle w'| : w \in \mathrm{Mod}(T), w' \in \mathrm{Mod}(A)\}, \leq), \\ \mathit{p\_diff}(w, A) &= \min(\{w \triangle w' : w' \in \mathrm{Mod}(A)\}, \subseteq). \end{aligned}$$

When determining the minimal difference, *diff* and *p_ diff* use the subset relation in comparison, while *dist* compares the cardinalities of the sets. The first two of the functions search for the minimal differences between two model sets, while the last function compares one model to a set of models pointwise.

We use four semantically-oriented belief-revision operators as modification operators: Dalal's [6] operator $*_D$, Satoh's [23] operator $*_S$, Weber's [25] operator $*_W$, and Borgida's [4] operator $*_B$. For all the operators, we define $\mathrm{Mod}(A*K) = \mathrm{Mod}(K)$ whenever $\mathrm{Mod}(A) = \emptyset$, otherwise the operators are defined as follows [18]:

$$\begin{aligned}
\mathrm{Mod}(A *_D K) &= \{w \in \mathrm{Mod}(K) : \exists w' \in \mathrm{Mod}(A), |w \bigtriangleup w'| = \mathit{dist}(A,K)\},\\
\mathrm{Mod}(A *_S K) &= \{w \in \mathrm{Mod}(K) : \exists w' \in \mathrm{Mod}(A), w \bigtriangleup w' \in \mathit{diff}(A,K)\},\\
\mathrm{Mod}(A *_W K) &= \{w \in \mathrm{Mod}(K) : \exists w' \in \mathrm{Mod}(T),\ w \bigtriangleup w' \subseteq \textstyle\bigcup \mathit{diff}(A,K)\},\\
\mathrm{Mod}(A *_B K) &= \mathrm{Mod}(A \wedge K), \text{ if } A \wedge K \text{ is satisfiable, otherwise}\\
\mathrm{Mod}(A *_B K) &= \textstyle\bigcup_{w \in \mathrm{Mod}(A)} \{w' \in \mathrm{Mod}(K) : w \bigtriangleup w' \in \mathit{p_\,diff}(w,K)\}.
\end{aligned}$$

Only Dalal's operator satisfies all (R1)–(R6), Satoh's and Borgida's operators satisfy (R1)–(R5), Weber's operator satisfies (R1)–(R4).

The operators define rules to produce the new set of models, but they do not define the outcome of the addition as a formula. A formula A' may be the result of the revision $A * K$, if $\mathrm{Mod}(A') = \mathrm{Mod}(A * K)$.

6 Experiments on Accommodative Revision

We shall next experiment. Let us start by considering the classic, simple example of a dinosaur and a vase given by Fermé and Hansson [12].

Example 1 ("Dinosaur broke grandma's vase!").

On your return home, your son tells you that a dinosaur has broken grandmother's vase in the living room. Assuming that you know that dinosaurs do not exist this claim cannot be entirely correct, but you may still want to accept as true that the vase has been broken.[5] We can formalize the example as follows:

a grandma's vase is intact,
b grandma's vase is in the living room,
c a dinosaur broke grandma's vase,
d dinosaurs exist.

What you know is $\neg d \wedge (\neg d \to \neg c)$. Assume that you have come to believe (independently) that a and that b. This means that we have four epistemically possible worlds with both c and d false in each. The most plausible one is the world in which a and b are true. Thus prior to the revision the epistemic state is

[5] In fact, the world changes in this example so it might be more appropriate to do a belief update rather than a revision, but we will ignore this now since the example has often been used to illustrate non-prioritized revision.

world	a	b	c	d	rank
w_{12}	1	1	0	0	0
w_4	0	1	0	0	1
w_8	1	0	0	0	1
w_0	0	0	0	0	2

Your son then tells you that $b \wedge c \wedge \neg a$. Revising this with a formula K representing your knowledge, $\neg c \wedge \neg d$, gives as result $A * K \equiv \neg a \wedge b \wedge \neg c \wedge \neg d$.

Revision of beliefs with this formula makes the only world satisfying this formula to become the most plausible one. As a result, you believe that grandma's vase is broken in the living room but not that it was a dinosaur that broke it.

The execution of the example with our Haskell implementation confirms the result. The example is so simple that all the modification operators we implemented result in the same revised epistemic state:

world	a	b	c	d	rank
w_4	0	1	0	0	0
w_{12}	1	1	0	0	1
w_8	1	0	0	0	2
w_0	0	0	0	0	3

As an example of a case where different modification functions give us different results, let us consider again the three-toed woodpecker example from the introduction.

Example 2 (The three-toed woodpecker).

We formalize the situation as follows:

a Amy saw a bird,
b Amy saw a three-toed woodpecker,
c Amy saw a bird with a red forehead,
d Amy saw a bird with a red rump.

Initially the agent knows that a three-toed woodpecker neither has a red forehead nor a red rump, and that if Amy saw a three-toed woodpecker or she saw a bird with a red forehead or she saw a bird with a red rump outside her window, then she saw a bird outside her window. Thus the agent has the knowledge $K \equiv (b \rightarrow (\neg c \wedge \neg d)) \wedge ((b \vee c \vee d) \rightarrow a)$.

Amy then tells that $a \wedge b \wedge c \wedge d$. Now, the agent may have different results of the modification of the input depending on which modification function is used:

$$(a \wedge b \wedge c \wedge d) *_D K \equiv (a \wedge \neg b \wedge c \wedge d),$$
$$(a \wedge b \wedge c \wedge d) *_S K \equiv (a \wedge \neg b \wedge c \wedge d) \vee (a \wedge b \wedge \neg c \wedge \neg d),$$
$$(a \wedge b \wedge c \wedge d) *_B K \equiv (a \wedge \neg b \wedge c \wedge d) \vee (a \wedge b \wedge \neg c \wedge \neg d),$$
$$(a \wedge b \wedge c \wedge d) *_W K \equiv (a \wedge \neg b) \vee (a \wedge b \wedge \neg c \wedge \neg d).$$

Here Dalal's operator gave the result (1) mentioned in the example in Sect. 1, Satoh's and Borgida's operators gave the result (2), and Weber's operator gave the result (3).

However, had Amy told that $b \wedge c \wedge d$, the results would have been different:

$(b \wedge c \wedge d) *_D K \equiv (a \wedge \neg b \wedge c \wedge d)$,
$(b \wedge c \wedge d) *_S K \equiv (a \wedge \neg b \wedge c \wedge d) \vee (a \wedge b \wedge \neg c \wedge \neg d)$,
$(b \wedge c \wedge d) *_B K \equiv (a \wedge \neg b \wedge c \wedge d) \vee (a \wedge b \wedge \neg c \wedge \neg d) \vee (\neg a \wedge \neg b \wedge \neg c \wedge \neg d)$
$(b \wedge c \wedge d) *_W K \equiv K$.

Here Dalal's and Satoh's operators still give charitable results, but Borgida's and Weber's operators do not.

In general, the more permissive is the revision operator, the less information is left to be obtained from the modified input (but the less likely it is that the agent has ruled out the actual state of affairs). Also the combination of pointwise revision and incomplete input is likely to give uncharitable results.

7 Conclusion

To illustrate the effect of knowledge on belief revision, we have presented a non-prioritized belief revision method that we call accommodative revision. In this method, the input is first revised with the knowledge of the agent. Beliefs are then revised with the resulting modified input. The properties of the method have been studied, a prototype implementation has been described, and some experiments have been analyzed.

The two components of accommodative revision can be chosen separately. The method does not call for any particular representation of epistemic states nor any principles on the components of accommodative revision other than the basic AGM-postulates. Accommodative revision is not meant to be the only way for an agent to update its epistemic state. Rather, it is to provide the agent with a set of tools to build convenient new belief-change operators to go with the old ones.

References

1. 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, 1985.
2. D. Bellot, C. Godefroid, P. Han, J.P. Prost, K. Schlechta, and E. Wurbel. A semantical approach to the concept of screened revision. *Theoria*, 63:24–33, 1997.
3. Salem Benferhat, Souhila Kaci, Daniel Le Berre, and Mary-Anne Williams. Weakening conflicting information for iterated revision and knowledge integration. *Artificial Intelligence*, 153:339–371, 2004.
4. Alexander Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems*, 10(4):565–603, 1985.
5. Maria R. Cravo, João P. Cachopo, Ana C. Cachopo, and João P. Martins. Permissive belief revision. In *Proceedings of the 10th Portuguese Conference on Artificial Intelligence on Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving, LNCS 2258*, pages 335–348. Springer, 2001.

6. Mukesh Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the AAAI-88 Seventh National Conference on Artificial Intelligence*, pages 475–479. Morgan Kaufmann, 1988.
7. Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. In Ronald Fagin, editor, *Theoretical Aspects of Reasoning about Knowledge. Proceedings of the Fifth Conference (TARK 1994)*, pages 5–23, 1994.
8. Adnan Darwiche and Judea Pearl. On the logic of iterated belief revision. *Artificial Intelligence*, 89(1-2):1–29, 1997.
9. James P. Delgrande, Daphne H. Liu, Torsten Schaub, and Sven Thiele. COBA 2.0: A consistency-based belief change system. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 4724 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 78–90. Springer-Verlag, 2007.
10. Satu Eloranta. Updating the knowledge base (in Finnish). Technical Report C-1995-5, Department of Computer Science, University of Helsinki, 1995.
11. Satu Eloranta. Dynamic features of the knowledge base. Technical Report C-2004-90, Department of Computer Science, University of Helsinki, 2004.
12. Eduardo L. Fermé and Sven Ove Hansson. Selective revision. *Studia Logica*, 63(3):331–342, 1999.
13. Michael Freund and Daniel Lehmann. Belief revision and rational inference. Technical Report TR 94-16, Hebrew University, 1994.
14. Adam Grove. Two modellings for theory change. *Journal of Philosophical Logic*, 17:157–170, 1988.
15. Sven Ove Hansson. Belief contraction without recovery. *Studia Logica*, 50(2):251–260, 1991.
16. Sven Ove Hansson. A survey of non-prioritized belief revision. *Erkenntnis*, 50(2-3):413–427, 1999.
17. Jaakko Hintikka. *Knowledge and Belief.* Cornell University Press, 1962.
18. Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledgebase revision and minimal change. *Artificial Intelligence*, 52(3):263–294, 1992.
19. David Makinson. Screened revision. *Theoria*, 63:14–23, 1997.
20. Juliano Maranhão. Refinement. a tool to deal with inconsistencies. In *Proceedings of the Eighth International Conference on Artificial Intelligence and Law, ICAIL 2001*, pages 52–59, 2001.
21. Simon Peyton Jones, editor. *Haskell 98 Language and Libraries: the Revised Report.* Cambridge University Press, 2003. Also available as *Journal of Functional Programming*, whole number 13(1), 2003.
22. Raymond Reiter. On integrity constraints. In *2nd Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 97–111. Morgan Kaufmann, 1988.
23. Ken Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, pages 455–462, 1988.
24. Wolfgang Spohn. Ordinal conditional functions: a dynamic theory of epistemic state. In William L. Harper and Brian Skyrms, editors, *Causation in Decision, Belief Change, and Statistics*, volume II, pages 105–134. Kluwer Academic Publishers, 1988.
25. Andreas Weber. Updating propositional formulas. In *Proceedings of the First International Conference on Expert Database Systems*, 1986.