Compositional Belief Update

James Delgrande and **Francis Jeffry Pelletier** and **Matthew Suderman**
School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada, V5A 1S6
{jim,jeffpell,mjsuderm}@cs.sfu.ca

Abstract

We propose an update operator for modifying a knowledge base. The approach differs from other belief change operators in that the definition of the operator is compositional with respect to the sentence to be added. The goal is to provide an update operator that is intuitive, in that its definition is based on a recursive decomposition of the update sentence's structure, and that may be reasonably implemented. We first provide a definition of update phrased in terms of the models of a knowledge base. We subsequently give an algorithm which captures this approach, but where the (syntactic) knowledge base contributes only a linear factor to the complexity of the algorithm. Hence the resulting approach has generally much better complexity characteristics than other operators. However, while the operator satisfies a core group of the benchmark Katsuno-Mendelzon update postulates, not all of the postulates are satisfied. Other Katsuno-Mendelzon postulates can be obtained by suitably restricting the syntactic form of the sentence for update, as we show. In this fashion we also come up with a hierarchy of update operators with, it turns out, Winslett's *standard semantics* as the weakest approach captured.

Introduction

Knowledge bases are not static entities, but rather evolve over time. New information may be added, and old or out-of-date information may be removed. A fundamental issue concerns how such change should be managed. A major body of research addresses this question via the specification of *rationality postulates*, or standards that an adequate change operator should satisfy. These postulates describe belief change at the *knowledge level*, independent of how beliefs are represented and manipulated. In the *AGM approach* (Alchourrón, Gärdenfors, & Makinson 1985; Gärdenfors 1988), standards for revision and contraction functions are given, wherein it is assumed that a knowledge base is receiving information concerning a static domain. Subsequently, Katsuno and Mendelzon (Katsuno & Mendelzon 1992) have explored a distinct notion of belief change, with functions for belief *update* and *erasure*, wherein an agent changes its beliefs in response to changes in the environment. See (Katsuno & Mendelzon 1992) for a comparison between revision and update. Various researchers have proposed specific change operators for belief revision (Borgida 1985; Dalal 1988; Satoh 1988) and update (Forbus 1989; Weber 1986; Winslett 1988). These approaches are formulated in terms of the distance between models of the knowledge base and a sentence for revision or update. In general there has been less work dealing with systems that may be readily implementable (but see for example (Williams 1996; Delgrande & Schaub 2003)).

In this paper we develop a specific update operator, wherein the operator intended to be *compositional* in the sentence μ representing information to be added. Consequently, the operator is defined recursively, in terms of the update sentences. For example, if a knowledge base is to be updated by a disjunction $\mu = a \vee b$, the intuition is that this can be effected by the update by a in combination with the update by b. The goal is to arrive at an operator whose results are intuitive, in that it is based on a recursive decomposition of a formula; hence the (generally abstract) notion of update is here anchored in a more familiar computational setting. Second, the hope is that these operators will be efficiently implementable, at least in some cases, by exploiting restrictions to the syntactic form of the formula. The focus here is on the form of the formula for update; presumably the approach described may be combined with one in which the knowledge base is divided into relevant and irrelevant parts for an update (Parikh 1999).

These goals are generally realised. First, the approach leads to a straightforward algorithm for implementation. This algorithm is efficient, compared to the model-based definition of this and other distance-based operators, in that the size of the knowledge base contributes only a linear factor to the overall complexity. As well, further efficiency is obtained when the input sentence is restricted to disjunctive normal form or when its size is bounded by a constant. As well, the operators have reasonable properties: many of the Katsuno and Mendelzon benchmark properties are satisfied, including those deemed essential by (Herzig & Rifi 1999). Further we show how the remaining Katsuno and Mendelzon properties may be obtained (by restricting the sentence for update), and we relate a restriction of our approach to one by Winslett.

The next section reviews belief update, and describes two specific approaches to update. The section following describes our approach after which, in the next section, we give a discussion and analysis. The last section contains concluding remarks; proofs of theorems are given in the full paper.

Background

Belief Update and Erasure

A formula is said to be *complete* just if it implies the truth or falsity of every other formula. In the "syntactic" approach of (Katsuno & Mendelzon 1992), which we follow in this, a knowledge base is a finite set of formulas (and hence the set of literals is finite and it could be expressed as a single formula), and an *update* function $\diamond$ is a function from $L \times L$ to L satisfying the following postulates.

(U1) $\psi \diamond \mu \vdash \mu$.

(U2) If $\psi \vdash \mu$ then $(\psi \diamond \mu) \equiv \psi$.

(U3) If μ and ψ are satisfiable then so is $\psi \diamond \mu$.

(U4) If $\psi_1 \equiv \psi_2$ and $\mu_1 \equiv \mu_2$ then $(\psi_1 \diamond \mu_1) \equiv (\psi_2 \diamond \mu_2)$.

(U5) $(\psi \diamond \mu) \wedge \phi$ implies $\psi \diamond (\mu \wedge \phi)$.

(U6) If $\psi \diamond \mu_1 \vdash \mu_2$, and $\psi \diamond \mu_2 \vdash \mu_1$ then $(\psi \diamond \mu_1) \equiv (\psi \diamond \mu_2)$.

(U7) If ψ is complete then $(\psi \diamond \mu_1) \wedge (\psi \diamond \mu_2)$ implies $\psi \diamond (\mu_1 \vee \mu_2)$.

(U8) $(\psi_1 \vee \psi_2) \diamond \mu \equiv (\psi_1 \diamond \mu) \vee (\psi_2 \diamond \mu)$

These postulates are not, however, uncontentious. (Herzig & Rifi 1999) discusses the plausibility of the postulates is given, and it is concluded that **U2**, **U5**, and **U6** are undesirable, while **U7** is unimportant. This leaves (according to the authors) **U1**, **U3**, **U4**, and **U8** as being desirable.

There have been various specific update (and revision) operators proposed based on the distance between models. We focus on two, both due to Winslett. The first, the *Possible Models Approach (PMA)* (Winslett 1988) is a well-known example of an update operator satisfying the Katsuno and Mendelzon update postulates. The second, Winslett's *standard semantics* (Winslett 1990) is a "weak approach" to update that is captured in a variant of ours. We denote these operators by $\diamond_{pma}$ and $\diamond_{ss}$ respectively.

For $\psi \diamond_{pma} \mu$, we have that, for each interpretation I of ψ, $\diamond_{pma}$ selects from the interpretations of μ that are "closest" to I. The update is determined by the set of these closest interpretations. The notion of "closeness" between two interpretations I and J is the Hamming distance, given as follows:

Definition 1 *$diff(I, J)$ = The set of all propositional letters on which I and J differ.*

Interpretation J_1 is closer to I than J_2, expressed as a partial ordering $J_1 \leq_I J_2$, just if $diff(I, J_1) \subseteq diff(I, J_2)$. The $\leq_I$-minimal set with respect to μ is designated $Incorporate(Mod(\mu), I)$. ¿From this we can specify the PMA update operator:

$$Mod(\psi \diamond_{pma} \mu) = \bigcup_{I \in Mod(\psi)} Incorporate(Mod(\mu), I).$$

The update $\psi \diamond_{ss} \mu$ is defined so that for each model of ψ, those models of μ that retain the truth values of atoms not in μ are chosen. That is, if $atom(\mu)$ is the set of atoms in μ:

$$Mod(\psi \diamond_{ss} \mu) = \bigcup_{I \in Mod(\psi)} \{J \in Mod(\mu) \mid diff(I, J) \subseteq atom(\mu)\}$$

Example 1 ((Katsuno & Mendelzon 1992)) *Let $L = \{b, m\}$ be the language of discourse. Let $\psi \equiv (b \wedge \neg m) \vee (\neg b \wedge m)$, and $\mu = b$. The interpretations of ψ are $I_1 = (\neg b, m)$, $I_2 = (b, \neg m)$; and the interpretations of μ are: $J_1 = (b, m)$, $J_2 = (b, \neg m)$. Thus $diff(I_1, J_1) = \{b\}$ and $diff(I_1, J_2) = \{b, m\}$, hence $J_1 \leq_{I_1, pma} J_2$, and so $Incorporate(Mod(\mu), I_1) = b$. Similarly, $Incorporate(Mod(\mu), I_2) = \{\}$. Hence, $(\psi \diamond_{pma} \mu) \leftrightarrow b$.[1] The same result obtains for $\diamond_{ss}$.*

For concreteness, take b to mean "the book is on the floor", and m to mean "the magazine is on the floor". So ψ means that either the book or the magazine is on the floor, but not both. A robot is ordered to put the book on the floor. Intuitively, at the end of this action the book will be on the floor, and the location of the magazine will be unknown. Both operators give this result.

Example 2 *Now let $\psi = (\neg b \wedge \neg m)$ and $\mu = (b \vee m)$. Then $(\psi \diamond_{pma} \mu) \leftrightarrow (b \equiv \neg m)$, whereas $(\psi \diamond_{ss} \mu) \leftrightarrow (b \vee m)$.*

Here, neither the book nor the magazine is on the floor. The robot is ordered to put at least one of them on the floor. According to the $\diamond_{pma}$ operator, exactly one will be on the floor after this action, while according to the $\diamond_{ss}$ operator, at least one will be on the floor.

The Approach

Preliminaries

The underlying logic is classical propositional logic. We consider a propositional language L, over a finite set of atoms, or propositional letters, $\mathbf{P} = \{a, b, c, \dots\}$, and truth-functional connectives $\neg$, $\wedge$, $\vee$, $\implies$, and $\equiv$. $Lits$ is the set of literals $\mathbf{P} \cup \{\neg l \mid l \in \mathbf{P}\}$. For a literal l, we use $\bar{l}$ to denote $\neg l$ if $l \in P$ or $l' \in P$ if $\neg l' = l$. Similarly, for a set of literals Γ, we use $\overline{\Gamma}$ to denote the set $\{\bar{l} \mid l \in \Gamma\}$. An *interpretation* of L is a function from $\mathbf{P}$ to $\{T, F\}$. A *model* of a sentence α is an interpretation that makes α true, according to the usual definition of truth. A model can be equated with its defining set of literals. $Mod(\alpha)$ denotes the set of models of sentence α. For simplicity, and in common with Katsuno and Mendelzon, we assume that knowledge bases are finite – that is, knowledge bases are expressible by a finite language. For interpretation ω we write $\omega \models \alpha$ to mean α is true in ω. For interpretation ω and set of literals Γ, we write $\omega \downarrow \Gamma$ to denote the set of literals in ω but containing neither l nor $\bar{l}$ for each $l \in \Gamma$. (That is, the set of literals in ω that do not occur either positively or negatively in Γ. Thus, $\omega \downarrow \Gamma = \omega \setminus (\Gamma \cup \overline{\Gamma})$.) For example, if $L = \{a, b, c\}$ and $\omega = \{a, \neg b, c\}$ then $\omega \downarrow \{b, \neg c\} = \{a\}$.

[1] We use $\equiv$ for *material biconditional* and $\leftrightarrow$ for *logical equivalence*.

We denote the conjunctive normal form of a sentence μ by $cnf(\mu)$ and the disjunctive normal form of μ by $dnf(\mu)$. For the most part we will work with a specific representation of $cnf(\mu)$ (respectively $dnf(\mu)$), called *normal conjunctive normal form (disjunctive normal form)* and denoted $ncnf(\mu)$ ($ndnf(\mu)$). $ncnf(\mu)$ is obtained by converting μ to negation normal form, and then distributing disjunctions over conjunctions insofar as possible. $ndnf(\mu)$ is obtained in the obvious dual manner. The result of this manipulation is that every atom in the relevant language will occur in each of the embedded disjuncts (for $ncnf(\mu)$) and in each of the embedded conjuncts (for $ndnf(\mu)$). $ncnf(\mu)$, $ndnf(\mu)$, $cnf(\mu)$, and $dnf(\mu)$ will each consist of sets of sets of literals. Members of $cnf(\mu)$ ($ncnf(\mu)$) are implicitly conjoined, and the literals in a member of $cnf(\mu)$ ($ncnf(\mu)$) are implicitly disjoined. The analogous, dual, convention holds for $dnf(\mu)$ and $ndnf(\mu)$. We will also use $\bigvee\Gamma$ to denote the disjunction and $\bigwedge\Gamma$ the conjunction of the sentences in Γ.

Later we make extensive use of the notion of the *prime implicants* of a sentence. A consistent set of literals Γ is a *prime implicant* of μ iff: $\Gamma \vdash \mu$ and for $\Gamma' \subset \Gamma$ we have $\Gamma' \not\vdash \mu$.

Intuitions

As we said, our goal is to define update operators in a compositional fashion so that, for updating formula μ, update is defined in terms of the components of μ. For an update $\psi \diamond \mu$, the idea is that each model of ψ is replaced by its closest model(s) in μ (Katsuno & Mendelzon 1992). The notion of "close" for each model of ψ is determined in part by the syntactic structure of μ. That is, μ is recursively decomposed, and the resulting (base case) literals are used to determine models of the update.

Consider how this may be carried out; we are given a knowledge base ψ and a sentence μ, and we wish to determine a new knowledge base where μ is believed. For a base case, $\mu = l$ is a literal; and we wish to add l to the knowledge base ψ. If ψ already implies l then we need do nothing. But if l is not believed, then we wish to arrive at a knowledge base in which l is believed. That is, we want to change the knowledge base only enough so that it entails l. We can do this by replacing each model ω of ψ such that $\omega \not\models l$ by the interpretation $\omega' = (\omega \downarrow \{\bar{l}\}) \cup \{l\}$. Thus, we would have would have that every resulting interpretation would entail l.

Consider next, adding a conjunction of literals $\mu = l_1 \wedge l_2$ to a knowledge base. A knowledge base in which $l_1 \wedge l_2$ is believed will, obviously, be one in which every model of the knowledge base entails both l_1 and l_2. We carry this out by replacing each interpretation $\omega \in Mod(\psi)$ where $\omega \models l_1 \wedge l_2$, with an interpretation $\omega' = (\omega \downarrow \{\overline{l_1}, \overline{l_2}\}) \cup \{l_1, l_2\}$

To add a disjunction of literals $\mu = l_1 \vee l_2$ to a knowledge base we want to modify models so that at least one of l_1 or l_2 is true in the interpretation. We can accomplish this by adding, for each $\omega \in Mod(\psi)$ such that $\omega \not\models l_1 \vee l_2$, interpretations $\omega_1 = (\omega \downarrow \{\overline{l_1}\}) \cup \{l_1\}$ and $\omega_2 = (\omega \downarrow \{\overline{l_2}\}) \cup \{l_2\}$. Finally, in the most general case: to add a disjunction of conjunctions of literals we generalize the last-mentioned method. The general case is where we add $\mu = (l_{1_1} \wedge \cdots \wedge l_{1_i}) \vee (l_{2_1} \wedge \cdots \wedge l_{2_j})$ to a knowledge base. We want to modify models so that at least one of the two conjunctions, $(l_{1_1} \wedge \cdots \wedge l_{1_i})$ or $(l_{2_1} \wedge \cdots \wedge l_{2_j})$, is true in the interpretation. We can accomplish this by replacing the l_1 and l_2 in the simple disjunction of literals by a conjunction of the form used in the immediately preceding case, recursively erasing each disjunct and then taking the union of the results.

A Compositional Update Operator

Based on these intuitions, we define an update operator $\diamond_c$. We begin with some preliminary definitions. In the following, $SetL$ sets the value of select literals of interpretation ω, given a set of formulas Γ. The idea is that ω is a model of the knowledge base and Γ is a set resulting from the partial decomposition of a formula for update. This formalises the procedure given in the discussion above.

Definition 2 *For interpretation ω and $\Gamma \subseteq L$, define $SetL(\omega, \Gamma)$ as follows:*

1. *If $\Gamma \subseteq Lits$ then $SetL(\omega, \Gamma) = \{(\omega \downarrow \Gamma) \cup \Gamma\}$.*
2. *If $\Gamma = \{\alpha \wedge \beta\} \cup \Gamma'$ then $SetL(\omega, \Gamma) = SetL(\omega, \{\alpha, \beta\} \cup \Gamma')$*
3. *If $\Gamma = \{\alpha \vee \beta\} \cup \Gamma'$ then $SetL(\omega, \Gamma) = SetL(\omega, \{\alpha\} \cup \Gamma') \cup SetL(\omega, \{\beta\} \cup \Gamma')$*
4. *If $\Gamma = \{\neg(\alpha \vee \beta)\} \cup \Gamma'$ then $SetL(\omega, \Gamma) = SetL(\omega, \{\neg\alpha, \neg\beta\} \cup \Gamma')$*
5. *If $\Gamma = \{\neg(\alpha \wedge \beta)\} \cup \Gamma'$ then $SetL(\omega, \Gamma) = SetL(\omega, \{\neg\alpha\} \cup \Gamma') \cup SetL(\omega, \{\neg\beta\} \cup \Gamma')$*
6. *If $\Gamma = \{\neg\neg\alpha\} \cup \Gamma'$ then $SetL(\omega, \Gamma) = SetL(\omega, \{\alpha\} \cup \Gamma')$*

Definition 3
$\psi \diamond_c \mu = \{\omega' \mid \omega' \in SetL(\omega, \{\mu\}), \omega \in Mod(\psi)\}$.

The following theorem uses the normal disjunctive normal form of the sentence to update to give an alternative definition for $\diamond_c$. The results are straightforward, but are useful for the penultimate section where we consider properties of the approach.

Theorem 1 *For $\psi, \mu \in L$, we have*

$$\psi \diamond_c \mu = \{(\omega \downarrow \Gamma) \cup \Gamma \mid \omega \in Mod(\psi), \Gamma \in ndnf(\mu)\}$$

It follows from Theorem 1 that

Theorem 2 *For $\psi, \mu \in L$, we have*
$Mod(\psi) \cap Mod(\mu) \subseteq \psi \diamond_c \mu \subseteq Mod(\mu)$.

Properties of the Operators

To start, we consider which of the Katsuno-Mendelzon postulates our operators satisfy.

Theorem 3 *$\diamond_c$ satisfies* **U1**, **U3**, **U5**, **U7**, **U8**; *it fails to satisfy* **U2**, **U4**, *and* **U6**.

We provide rationales for the claims about **U2**, **U4**, and **U6**. Proofs of the others can be found in the full paper. For a counterexample to **U2** consider the second example given illustrating Winslett's approaches, where $\psi = (b \wedge \neg m) \vee (\neg b \wedge m)$ and $\mu = b \vee m$. In our approach, updating by the first disjunct gives interpretations $\{b, \neg m\}$ and $\{b, m\}$ and updating by the second disjunct gives $\{b, m\}$ and $\{b, m\}$. Hence $\psi \diamond_c (b \vee m) \leftrightarrow b \vee m$. **U2** would dictate that the result be ψ. **U2** seems problematic in the context of update. To borrow an example from (Herzig & Rifi 1999; Brewka & Herzberg 1993), suppose an agent believes p (that a certain coin shows heads). Now the world changes because of a toss of this coin (where the agent does not see the result). Letting q be that the coin shows tails, we note that the agent should believe $(p \vee q)$. Yet note that $p \vdash (p \vee q)$; so **U2** would predict that $p \diamond (p \vee q)$ should be p, contrary to what we want. The operator $\diamond_c$, on the other hand, includes an additional model. This appears to make some sense because by updating by $b \vee m$, we are really telling the knowledge base that the world has changed so that one of $b \wedge m$ or $b \wedge \neg m$ or $\neg b \wedge m$ is true. Thus, in this case the update operator behaves like a *Gricean* update operator (Delgrande, Nayak, & Pagnucco 2005), where the goal is to incorporate *all and only* the new information.

A counterexample for **U4** is provided by:

$$a \diamond_c ((\neg a \wedge b) \vee b) = Mod((\neg a \wedge b) \vee (a \wedge b));$$
$$\text{so,} \quad a \diamond_c b = Mod(a \wedge b).$$

The reason that **U4** is not satisfied is that in our compositional approach, parts of a sentence may interact to provide implicit results not explicit in the sentence. Consider $(\neg a \vee b) \wedge (\neg b \vee c)$ for example. Updating by this sentence is effected by updating by the individual components, viz., $(\neg a \vee b)$ and $(\neg b \vee c)$. However, implicit in these parts is the fact that $(\neg a \vee c)$ is also true, and the presence of this sentence would affect the result of the update. We consider this behaviour further below.

A counterexample for **U6** is given by:

$$(a \wedge b) \diamond_c ((a \wedge c) \vee (\neg a \wedge b \wedge c)) = Mod((a \wedge b \wedge c) \vee (\neg a \wedge b \wedge c))$$
$$(a \wedge b) \diamond_c ((b \wedge c) \vee (a \wedge \neg b \wedge c)) = Mod((a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c))$$

U6 (and also **U2**) is not satisfied since $\psi \diamond_c \mu$ includes in some cases interpretations that satisfy μ but which are not in $Mod(\psi)$. We pursue this behaviour further in the next section, where we use this as a point of contrast with Winslett's approach.

Despite failing to satisfy some postulates (which–it should be noted–overlap with the postulates that (Herzig & Rifi 1999) think are undesirable), $\diamond_c$ does exhibit a nice property that operators appearing in the literature and satisfying the Katsuno and Mendelzon postulates fail to satisfy. Let $\mu \wedge \phi$ be a satisfiable propositional sentence. The following version of the disjunction property holds.

Theorem 4 $\psi \diamond_c (\mu \vee \phi) \leftrightarrow (\psi \diamond_c \mu) \vee (\psi \diamond_c \phi)$

Our update operator also satisfies those postulates deemed desirable by (Herzig & Rifi 1999), with the exception of **U4**. As discussed above, **U4** is not satisfied due to the interaction of parts of a sentence. It would seem that if we could "compile out" the implicit information in a sentence then we would obtain substitution of equivalents, as expressed in **U4**. So, one way to satisfy **U4** then would be to redefine $\diamond_c$ so that we first get this information implicit in the interaction of the compositionally distinct parts of the update. We do this by defining operators that consider the set of *prime implicants* of a sentence. We call these modified operator $\diamond_c^{pi}$. Let $PI(\mu)$ be the set of prime implicants of μ.

Definition 4 $\psi \diamond_c^{pi} \mu = \psi \diamond_c PI(\neg\mu)$

Theorem 5 $\diamond_c^{pi}$ *satisfies* **U4**

Although $\diamond_c^{pi}$ satisfies **U4**,

Theorem 6 $\diamond_c^{pi}$ *does not satisfy* **U7**

A counter-example for **U7** is given by $\mu_1 = (a \wedge d) \vee (\neg c \wedge d)$ and $\mu_2 = (\neg a \wedge d) \vee (\neg c \wedge d)$ and $\psi = a \wedge b \wedge c \wedge d$. By definition,

$$\psi \diamond_c^{pi} \mu_1 \ \cap \ \psi \diamond_c^{pi} \mu_2 =$$
$$\psi \diamond_c (\neg a \vee \neg d) \wedge (c \vee \neg d) \ \cap \ \psi \diamond_c (a \vee \neg d) \wedge (c \vee \neg d)$$

which entails $a \wedge b \wedge \neg c \wedge d$. On the other hand, $\psi \diamond_c^{pi} (\mu_1 \vee \mu_2) = \psi \diamond_c \neg d$ which is equal to ψ. Notice that the prime implicants of $\neg\mu_1$ and $\neg\mu_2$ retain the clause $(c \vee \neg d)$ whereas the only prime implicant of $\neg(\mu_1 \vee \mu_2)$ is $\neg d$. As a result, both $\psi \diamond_c^{pi} \mu_1$ and $\psi \diamond_c^{pi} \mu_2$ contain an interpretation ω' just like ω in ψ except that c is negated whereas $\psi \diamond_c^{pi} (\mu_1 \vee \mu_2)$ contains only the interpretations in ψ. This means that $\psi \diamond_c^{pi} \mu_1 \cap \psi \diamond_c^{pi} \mu_2$ does not imply $\psi \diamond_c^{pi} (\mu_1 \vee \mu_2)$.

We can further pursue this direction as follows. For μ let $ModL(\mu)$ be the models of μ, over the language of μ, expressed in disjunctive normal form. For example $ModL((a \vee b) \wedge c)$ would be $(a \wedge b \wedge c) \vee (a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge c)$.

Definition 5 $\psi \diamond_c^{ss} \mu = \psi \diamond_c ModL(\mu)$

We obtain:

Theorem 7 $\psi \diamond_c^{ss} \mu = \psi \diamond_{ss} \mu$.

Thus in this case we capture the basic Winslett approach (Winslett 1990).[2] Hence we obtain a hierarchy of operators, based on the extent to which information in μ is made explicit.

Finally, let's return to **U2**. Consider the following modification of our $\diamond_c$ operator[3]:

Definition 6

$$\psi \diamond_c' \mu = \begin{cases} \psi & \textit{if } \psi \vdash \mu \\ \psi \diamond_c \mu & \textit{otherwise} \end{cases}$$

Now, **U2** states that if we wish to update by μ and $\psi \vdash \mu$, then ψ is unchanged.

Theorem 8 $\diamond_c'$ *satisfies* **U2**

Last, we have already noted that our basic conception of update, given by $\diamond_c$, is distinct from the Winslett approaches. It is also distinct from all other approaches appearing in the literature.

[2]More accurately, we capture a syntax-independent variant of the Winslett approach (Herzig & Rifi 1999).

[3](Borgida 1985) employs a similar definition with respect to a revision operator.

Algorithms and Complexity

In this section we provide algorithms for our operators. We also analyse the complexity of these algorithms under a variety of assumptions. Specifically, we analyse the complexity of the algorithms when applied to any general propositional sentences, any sentences in conjunctive normal form, any sentences in disjunctive normal form, and any sentences whose sizes are bounded by some specified constant.

In the following algorithms, let $\psi, \mu \in L$:

Algorithm $Update(\psi, \mu)$
$\psi' \leftarrow False$
for each clause $\Gamma \in ndnf(\mu)$
  $\psi' \leftarrow \psi' \vee AssignL(\psi, \Gamma)$
return ψ'

Algorithm $AssignL(\psi, \Gamma)$
$\psi'' \leftarrow nnf(\psi)$
for each $l \in \Gamma$
  replace each l in ψ'' by $\bar{l}$
return ψ''

Note that $nnf(\psi)$ is just the negation normal form of ψ. $AssignL$ modifies the composition of the input knowledge base by modifying its negation normal form.

The next theorems say that these algorithms are complete and sound with respect to the operators.

Theorem 9 *For $\psi, \psi' \in L$ and $\Gamma \subseteq Lits$ we have*

1. $AssignL(\psi \wedge \psi', \Gamma) = AssignL(\psi, \Gamma) \wedge AssignL(\psi', \Gamma)$
2. $AssignL(\psi \vee \psi', \Gamma) = AssignL(\psi, \Gamma) \vee AssignL(\psi', \Gamma)$
3. $AssignL(\neg(\psi \vee \psi'), \Gamma) = AssignL(\neg\psi, \Gamma) \wedge AssignL(\neg\psi', \Gamma)$
4. $AssignL(\neg(\psi \wedge \psi'), \Gamma) = AssignL(\neg\psi, \Gamma) \vee AssignL(\neg\psi', \Gamma)$
5. $AssignL(\neg\neg\psi, \Gamma) = AssignL(\psi, \Gamma)$

Corollary 1 *For $\psi \in L$ and $\Gamma \subseteq Lits$ we have* $ndnf(AssignL(\psi, \Gamma)) = AssignL(ndnf(\psi), \Gamma)$.

Theorem 10 *For $\psi, \mu \in L$, $ndnf(Update(\psi, \mu))$ is equivalent to*

$$\bigvee \left\{ \bigwedge (\Gamma' \downarrow \Gamma) \cup \Gamma \mid \Gamma \in ndnf(\mu), \Gamma' \in ndnf(\psi) \right\}.$$

Theorem 11 $\psi \diamond_c \mu = Mod(Update(\psi, \mu))$.

Corollary 2 $\psi \diamond_c \mu \leftrightarrow Update(\psi, \mu)$

Now for the complexity of the algorithms. Let $\psi, \mu \in L$, and for any $\delta \in L$ let $|\delta|$ be the size of δ.

Theorem 12 *The complexity of evaluating $Update(\psi, \mu)$ is:*

1. $O(|\psi| \times 2^{|\mu|})$ *for $\mu \in L$;*
2. $O(|\psi| \times |\mu|)$ *for μ in ndnf; and*
3. $O(|\psi|)$ *for $|\mu| < k$ for some constant k.*

$Update$ is quite efficient compared to $SetL$ in Definition 2, since $SetL$ is defined in terms of the models of the knowledge base whereas $Update$ works with the knowledge base itself (in negation normal form). Further efficiency is obtained when the formula for update is in normal disjunctive normal form. (Eiter & Gottlob 1992) says that the major model-based operators are at least co-NP-complete when the update sentence is in this form. It is reasonable for practical knowledge base systems to put limits on the size or form of the update sentences, so these are positive results.

Conclusion

We have presented belief change operators for updating a knowledge base. The definition of these operators is compositional with respect to the sentence to be added. The intent is to provide operators with transparent definitions, based on the structure of the formula for belief change. As a result we lose some of the standard postulates for update, although we satisfy a core group of the standard postulate set. We achieve full irrelevance of syntax if the sentence for update is replaced by the disjunction of its prime implicants. The approach is interesting because first, it is founded on differing intuitions than other operators, in that it is based on a decomposition of the formula, and second, it allows a straightforward and, under reasonable assumptions, efficient implementation. While distinct from previous update operators that have appeared in the literature, we capture the Winslett's *standard semantics* approach to update in a restriction of our approach. In fact, the update operator, under different syntactic restrictions, may be regarded as constituting a family of update operators of which Winslett's *standard semantics* is the weakest, or base, approach.

An open question concerns combining this approach with one that is designed to exploit the structure of the knowledge base (such as discussed in (Parikh 1999) and characterized in terms of PMA updates in (Peppas, Chopra, & Foo 2004)). A second, technical question that is not fully explored concerns the behaviour of $\diamond_c$ as an erasure operator. For example, let $\psi = (a \vee b) \wedge (\neg a \vee \neg b)$. Then, because $\psi \vdash a \vee b$, we get that $\psi \diamond_c (a \vee b) \leftrightarrow a \vee b$. So, in updating the knowledge base with a formula already implied by the knowledge base, we have actually removed information. This, as mentioned earlier, would not be unreasonable if one considers that an update (in contrast to a revision) by $a \vee b$ asserts that the world has changed so that one of $\{a, b\}$, $\{\neg a, b\}$, $\{a, \neg b\}$ is now true.

References

Alchourrón, C.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic* 50(2):510–530.

Borgida, A. 1985. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems* 10.

Brewka, B., and Herzberg, J. 1993. How to do things with worlds: On formalizing actions and plans. *J. Logic Computation* 3:517–532.

Dalal, M. 1988. Investigations into theory of knowledge base revision. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 449–479.

Delgrande, J., and Schaub, T. 2003. A consistency-based approach for belief change. *Artificial Intelligence* 151(1-2):1–41.

Delgrande, J.; Nayak, A.; and Pagnucco, M. 2005. Gricean belief change. *Studia Logica* 79:97–113.

Eiter, T., and Gottlob, G. 1992. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence* 57(2-3):227–270.

Forbus, K. 1989. Introducing actions into qualitative simulation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1273–1278.

Gärdenfors, P. 1988. *Knowledge in Flux: Modelling the Dynamics of Epistemic States*. Cambridge, MA: The MIT Press.

Herzig, A., and Rifi, O. 1999. Propositional belief update and minimal change. *Artificial Intelligence* 115(1):107–138.

Katsuno, H., and Mendelzon, A. 1992. On the difference between updating a knowledge base and revising it. In Gärdenfors, P., ed., *Belief Revision*, 183–203. Cambridge University Press.

Parikh, R. 1999. Beliefs, belief revision, and splitting languages. In Moss, L.; Ginzburg, J.; and de Rijke, M., eds., *Logic, Language and Computation, Vol 2*. CSLI Publications. 266–278.

Peppas, P.; Chopra, S.; and Foo, N. 2004. Distance semantics for relevance-sensitive belief revision. In *KR2004: Principles of Knowledge Representation and Reasoning*. San Francisco: Morgan Kaufmann.

Satoh, K. 1988. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 455–462.

Weber, A. 1986. Updating propositional formulas. *Proc. First Conference on Expert Database Systems* 487–500.

Williams, M.-A. 1996. Towards a practical approach to belief revision: Reason-based change. In Aiello, L.; Doyle, J.; and Shapiro, S., eds., *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning*, 412–421.

Winslett, M. 1988. Reasoning about action using a possible models approach. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, 89–93.

Winslett, M. 1990. *Updating Logical Databases*. Cambridge: Cambridge University Press.