Forward chaining and change operators

Hassan Bezzazi, Stéphane Janot, Sébastien Konieczny and Ramón Pino Pérez

LIFL U.A. 369 du CNRS, Université de Lille I
59655 Villeneuve d'Ascq, FRANCE
E-mail: {bezzazi,janot,konieczn,pino}@lifl.fr

Abstract. We investigate three change operators which are defined using forward chaining. We prove that one of them is an update operator in the Katsuno-Mendelzon sense. Concerning the other two operators one of them is a revision operator in the Alchourrón-Gärdenfors-Makinson sense; the other one satisfies the basic postulates for revision.

Introduction

Revision is the process of according an old knowledge base with a new evidence. It has been formally studied [1, 8, 10] and several operators have already been proposed [4, 7]. The problem is that, in general, revision is a complex process [6] and is not efficiently computable.

In this paper, we investigate three change operators based on forward chaining. The use of forward chaining provides us with an efficient way for computing the revision of a knowledge base.

Furthermore these operators are readily suitable for expert systems based on the same kind of inference namely forward chaining. Thus, we have an easy way to include non-monotonic reasoning in such systems. Such operators may have numerous other applications, in diagnosis systems for example.

Our approach is close to REVISE [5] and Revision Programming [13] but is less complex, since we use only forward chaining on propositional formulae.

We propose three knowledge change operators. The first one, called *factual revision*, updates a set of facts with another set of facts coding a new evidence, according to a set of rules which can be seen as integrity constraints of the system. The second one, called *ranked revision*, is based on a hierarchy over the rules which denotes how the rules are *exceptional*, and when a new evidence arrives, it finds the least exceptional rules consistent with this new information. The last one, called *hull revision*, uses the two approaches above and computes maximal sets of rules that are consistent with the new information.

We also study properties of these operators. We prove that the factual revision can be seen as an update operator, satisfying Katsuno and Mendelzon postulates and that it has good iteration properties. In the same way, ranked revision can be seen as a revision operator, according to Alchourrón-Gärdenfors-Makinson (AGM) postulates. Concerning hull revision we prove only the basic postulates of change.

The paper is organized as follows: section 1 contains the basic definitions; in section 2 we define factual revision and we give an algorithm to compute it;

section 3 is devoted to definitions of ranked revision and hull revision; in section 4 we study the properties of these knowledge change operators. And, finally, we conclude with some remarks and some perspectives for future work.

1 Preliminaries

Our framework is propositional logic.

A literal (or fact) is an atom or a negation of an atom. The set of literals will be denoted Lit. A rule is a formula of the shape $l_1 \wedge l_2 \wedge \cdots \wedge l_n \rightarrow l_{n+1}$ where l_i is a literal for $i = 1, \ldots, n+1$. A rule as before will be denoted $l_1, l_2, \ldots, l_n \rightarrow l_{n+1}$. We admit rules of the form $\rightarrow l$ which actually code facts.

Let R and F be a finite set of rules and a finite set of literals respectively. A program P is the set of the form $R \cup F$ and we will say that the elements of R are the rules of P and the elements of F are the facts of P.

Let $P = R \cup F$ be a program. We define the set of consequences by forward chaining of P, denoted $C_{fc}(P)$, as the smallest set of literals F' such that:
(i) $F \subseteq F'$.
(ii) If $l_1, l_2, \ldots, l_n \rightarrow l$ is in R and $l_i \in F'$ for $i = 1, \ldots, n$ then $l \in F'$.
(iii) If F' contains two opposite literals then $F' = Lit$

A program P is said to be consistent iff $C_{fc}(P)$ does not contain two opposite literals, *i.e.* an atom and its negation.

Let R, F and F' be a finite set of rules and two finite sets of literals respectively. F is said to be R-consistent iff $R \cup F$ is consistent (with respect to forward chaining). F is said to be $R \cup F'$-consistent iff $R \cup (F \cup F')$ is consistent.

Let L and P be a set of literals and a program as previously defined we say that L is a fc-consequence of P iff $L \subseteq C_{fc}(P)$.

1.1 Revision and update postulates

Let φ be a formula representing a knowledge base and let μ be a formula representing a new piece of information. $\varphi \circ \mu$ will denote a formula representing the changes that μ produces in φ. The operator $\circ$ is a revision operator [1, 10] if it satisfies the following postulates:

(R1) μ is a consequence of $\varphi \circ \mu$.
(R2) If $\varphi \wedge \mu$ is consistent then $\varphi \circ \mu$ is equivalent to $\varphi \wedge \mu$.
(R3) If μ is consistent then $\varphi \circ \mu$ is consistent.
(R4) If $\varphi_1 \equiv \varphi_2$ and $\mu_1 \equiv \mu_2$ then $\varphi_1 \circ \mu_1 \equiv \varphi_2 \circ \mu_2$.
(R5) $\varphi \circ (\mu \wedge \phi)$ is a consequence of $(\varphi \circ \mu) \wedge \phi$.
(R6) If $(\varphi \circ \mu) \wedge \phi$ is consistent then $(\varphi \circ \mu) \wedge \phi$ is a consequence of $\varphi \circ (\mu \wedge \phi)$.

An operator satisfying all postulates above except **R4** is said to be a syntactical revision operator.

The operator $\circ$ is an update operator [11] if it satisfies the following postulates:

(U1) μ is a consequence of $\varphi \circ \mu$.
(U2) If μ is a consequence of φ then $\varphi \circ \mu$ is equivalent to φ.

(U3) If both ϕ and μ are consistent then $\varphi \circ \mu$ is also consistent.
(U4) If $\varphi_1 \equiv \varphi_2$ and $\mu_1 \equiv \mu_2$ then $\varphi_1 \circ \mu_1 \equiv \varphi_2 \circ \mu_2$.
(U5) $\varphi \circ (\mu \wedge \phi)$ is a consequence of $(\varphi \circ \mu) \wedge \phi$.
(U6) If μ_2 is a consequence of $(\varphi \circ \mu_1)$ and μ_1 is a consequence of $(\varphi \circ \mu_2)$ then $\varphi \circ \mu_1$ is equivalent to $\varphi \circ \mu_2$.
(U7) If φ is complete then $\varphi \circ (\mu_1 \vee \mu_2)$ is a consequence of $(\varphi \circ \mu_1) \wedge (\varphi \circ \mu_2)$.
(U8) $(\varphi_1 \vee \varphi_2) \circ \mu$ is equivalent to $(\varphi_1 \circ \mu) \vee (\varphi_2 \circ \mu)$.

An operator satisfying the postulates **U1** to **U8** except **U4** is said to be a syntactical update operator.

Note that in those postulates the notions of consequence, consistency and equivalence are the classical ones. But we can instantiate these postulates to different logics. We will see some examples in section 4.

2 Factual revision

Let R be a fixed set of rules which in this context can be seen as our background theory or our integrity constraints. Let F be a set of facts which can be seen as our beliefs about the world. We would like to define the change produced by a set of facts F' coding a new information about the world. This is the object of the following definition:

$$F \diamond_R F' = \begin{cases} F' & \text{if } F' \text{ is not } R\text{-consistent} \\ \langle F_1 \cup F', \ldots, F_n \cup F' \rangle & \text{where } \{F_1, \ldots, F_n\} \text{ is the set of subsets of } F \\ & \quad \text{which are maximal and } R \cup F'\text{-consistent} \end{cases}$$

So more generally than a set of facts F we are considering n-tuples of sets of facts $\langle F_1, \ldots, F_n \rangle$ called *flocks* in the literature [7].

We put $\langle F_1, \ldots, F_n \rangle \cdot \langle F'_1, \ldots, F'_m \rangle \stackrel{\text{def}}{=} \langle F_1, \ldots, F_n, F'_1, \ldots, F'_m \rangle$ and

$$\langle F_1, \ldots, F_n \rangle \diamond_R F' \stackrel{\text{def}}{=} (F_1 \diamond_R F') \cdot (F_2 \diamond_R F') \cdots (F_n \diamond_R F')$$

In order to investigate the relation between $\diamond_R$ and the postulates of change we need to define the intensional content (the consequences) of a flock $\mathcal{F} = \langle F_1, \ldots, F_n \rangle$. So we define the consequences by forward chaining (with respect to R) of such a flock, denoted $C^R_{fc}(\mathcal{F})$, by the following:

$$C^R_{fc}(\mathcal{F}) = \bigcap_{i=1}^{n} C_{fc}(F_i \cup R)$$

In section 4 we will show that $\diamond_R$ is an update operator

2.1 Update algorithm

For the sake of completeness in this subsection we give an algorithm to compute $F \diamond_R F'$. This is computed in two steps: first, we compute sets of facts that lead to inconsistency, called *contradictory sets*. Then, given the set SC of contradictory sets, we compute the minimal hitting sets of SC. The maximal subsets of F such that $F \cup F'$ is R-consistent are the sets $F \setminus H$, where H is a minimal hitting set of SC.

First step. A contradictory set C is a subset of F corresponding to a way of proving a pair of opposite literals from $R \cup F \cup F'$: C is a contradictory set if C is a subset of F and there exists a minimal subset R' of R such that $R' \cup C \cup F'$ is not consistent and, for each l in C, l appears in the body of a rule of R'.

We assume that, for every atom a that appears in the knowledge base, we have an implicit rule $a, \neg a \rightarrow \perp$.

To compute the contradictory sets when updating F with F', we build a contradiction tree $T_{F,F'}$, starting from $\perp$: a node is a pair$(L; C)$, where L is a list of literals to prove to obtain contradiction and C is a partial contradictory set. We start with the node $(\perp; \{\}\ )$. Let $N = (l_1, l_2, \ldots, l_n; C)$ be a node of $T_{F,F'}$. The successors of N are computed as follows:

- if $l_1 \in F'$ or if l_1 is already in C, then $(l_2, \ldots, l_n; C)$ is the only successor of N
- else for each rule $g_1, g_2, \ldots, g_p \rightarrow l_1$, $(g_1, g_2, \ldots, g_p, l_2, \ldots, l_n\ ;\ C \cup \{l_1\})$ is a successor of N and if $l_1 \in F$, then $(l_2, \ldots, l_n; C \cup \{l_1\})$ is a successor of N.

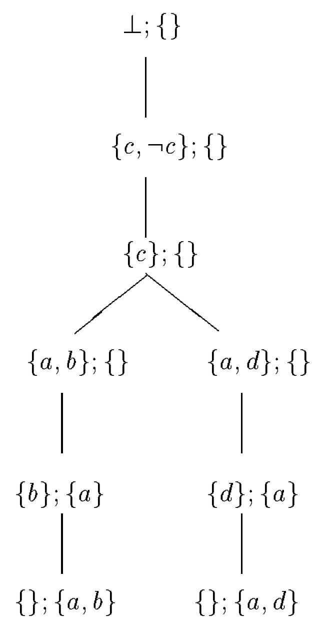
A branch terminates with an empty list of literals or with a node that cannot be developed. If a branch ends with $(\emptyset; C)$, then C is a contradictory set of facts. Note that if we suppose that F' is consistent with R, we can't obtain $(\emptyset; \emptyset)$. $T_{F,F'}$ doesn't give only the minimal contradictory sets, but all the ways to entail a pair of opposite literals from R and $F \cup F'$.

Example. We consider the program $R \cup F$, with $R = \{a, b \rightarrow c\ ;\ a, d \rightarrow c\}$ and $F = \{a, b, d\}$. When updating F with $\{\neg c\}$, we obtain two contradictory sets $\{a, d\}$ and $\{a, b\}$. Fig 1 shows the contradiction tree (to simplify, we consider only the rule $\neg c, c \rightarrow \perp$ at the first step, since the only pair of contradictory literals that actually appears in this case is $c, \neg c$).

Second step. The contradiction tree produces a set of contradictory sets $SC = \{C_1, ..., C_n\}$. To update F with F' we compute all the maximal subsets S of F such that $S \cup F'$ is R-consistent. The subsets S of F are obtained by removing from F at least one element of each contradictory set: if H is a set of facts such that, for each element C_i of SC, $S \cap C_i \neq \emptyset$, then $(F \setminus H) \cup F'$ is consistent. H is usually called a hitting set of SC. To find the maximal consistent subsets of F, we need all the minimal hitting sets (by set inclusion) of SC.

The figure 2 illustrates the algorithm we implemented in Prolog to compute the minimal hitting sets. This algorithm is very close to the one given by Reiter in [14]. Let $SC = \{C_1, C_2, ..., C_n\}$. We try to construct a hitting set of SC by

Fig. 1. Contradiction tree



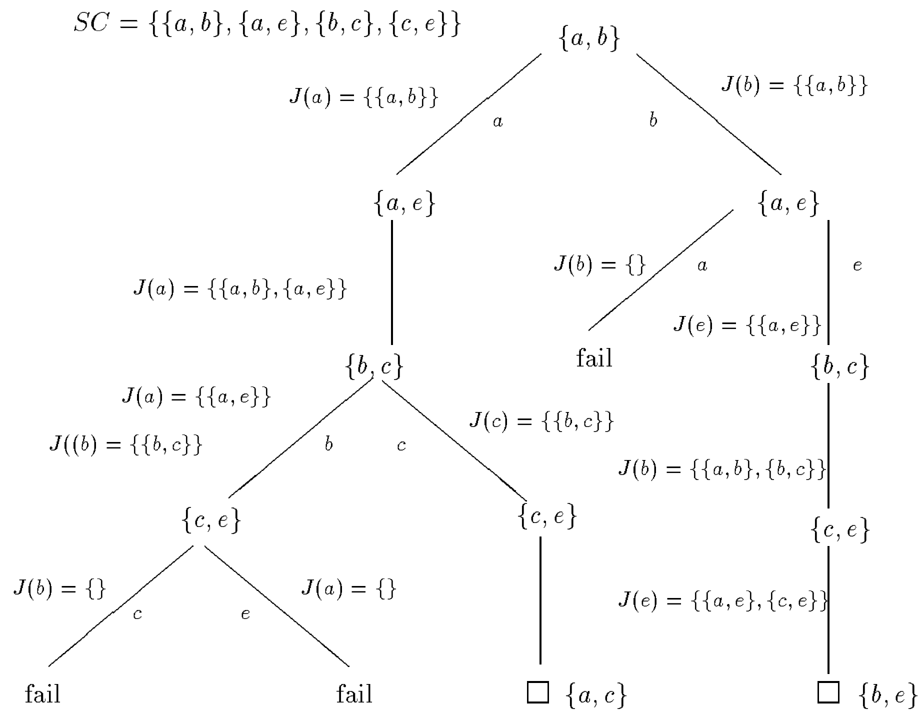
examining the elements of SC one by one: if the current set C_i is not already hit by the partial hitting set HS, we add one of the literals of C_i in HS. To know if a hitting set is minimal, we maintain a set of justifications $J(l)$ for each literal l of the hitting set: $J(l)$ contains all the sets of literals that are hit only by l. When a new literal l must be added to the hitting set, the justification sets are updated by removing all the sets containing l. If one of the justification sets becomes empty, the current hitting set is not minimal anymore and so the construction fails.

Concerning the relationships between our algorithm and Reiter's one notice that we construct the same kind of HS-tree, where nodes are labeled with elements of SC and edges are labeled with elements of the hitting sets. The main difference is the use of justification sets instead of tree pruning. Tree pruning is used in Reiter's algorithm in order to compute not all the hitting sets but only the minimal ones. In our algorithm, this is done with justification sets and each minimal hitting set is computed in a unique branch.

3 Ranked revision and hull revision

In the case of factual revision the set of rules is fixed. When it is not the case a natural question that one may ask is how to change a set of rules when a new piece of information arrives. The aim of this section is to give an answer to this question when the new piece of information is a set of facts.

Fig. 2. Minimal hitting sets



Minimal Hitting sets: $\{a, c\}$ and $\{b, e\}$

The change operators introduced in this section are inspired from the duality existing between revision and rational inference relations [9]. So the first operator can be seen as the 'relativization' of the rational closure [12] to the forward chaining logic. The second operator is an extension of the first one and it is aimed to satisfy a little bit more of transitivity (see [2, 3]).

Let R be a set of rules.

Definition 1 (Exceptional sets of literals and rules). A set of literals L is said to be exceptional with respect to R iff L is not R-consistent and a rule $L \longrightarrow l$ of R is said to be exceptional in R iff L is exceptional in R.

A similar definition of exceptionality for a formula can be found in [12].

Let $(R_i)_{i\in\omega}$ be the decreasing sequence defined by: R_0 is R and R_{i+1} is the set of all exceptional rules of R_i. Since R is finite there is a smallest integer n_0 such that for all $m \geq n_0$ we have $R_m = R_{n_0}$.

Definition 2 (Base). Let n_0 be as before. If $R_{n_0} \neq \emptyset$ we say that $\langle R_0, \ldots, R_{n_0}, \emptyset\rangle$ is the base of R. If $R_{n_0} = \emptyset$ we say that $\langle R_0, \ldots, R_{n_0}\rangle$ is the base of R.

Definition 3 (Completely exceptional). Let $\langle R_0, \ldots, R_n\rangle$ be the base of R. L is said completely exceptional if it is exceptional in all R_i for $0 \leq i < n$ (thus an inconsistent set of literals is completely exceptional).

Definition 4 (Rank function). Let $\langle R_0, \ldots, R_n\rangle$ be the base of R. Let $\rho : \mathcal{P}(Lit) \longrightarrow [1..n]$ be the rank function defined as follows: $\rho(L) = \min\{i \in [1..n] : L \text{ is } R_i\text{-consistent}\}$ if L is consistent, otherwise $\rho(L) = n$. It is a fact that if $L \subseteq L'$ then $\rho(L) \leq \rho(L')$

Definition 5 (Ranked revision). Let R and L be a set of rules and a set of literals respectively. We define the ranked revision of R by L, denoted $R \circ_{rk} L$, as follows:

$$R \circ_{rk} L = R_{\rho(L)} \cup L$$

In other words we take in the base of rules the first set of rules that agrees with the new information.

Let $I(L)$ be the set of maximal subsets of R which are consistent with L and which contain $R_{\rho(L)}$ when L is not completely exceptional and $\emptyset$ otherwise. Define $h : \mathcal{P}_{fin}(Lit) \longrightarrow \mathcal{P}(R)$ by $h(L) = \bigcap I(L)$

Definition 6 (Hull revision). The hull revision of a set of rules R by a set of literals L denoted $R \circ_h L$ is defined as follows

$$R \circ_h L = h(L) \cup L$$

Remark. By the definitions it is easy to see that

$$C_{fc}(R \circ_{rk} L) \subseteq C_{fc}(R \circ_h L)$$

Thus one can say that $\circ_h$ is a conservative extension of $\circ_{rk}$.

3.1 Examples

In this subsection we give examples that illustrate the behaviour of ranked revision and hull operators and at the same time the differences between them.

In the following examples we are interested in facts one can infer from the rule base R according to the inference relation defined by $L \vdash l$ iff $l \in C_{fc}(R \circ L)$ when $\circ$ is $\circ_{rk}$ or $\circ_h$.

Consider $R = \{b \to f, b \to w, o \to b, o \to \neg f\}$ where b, o, f, w stand respectively for birds, ostriches, fly and have wings. It is easy to see that the base is $< R_0, R_1, R_2 >$ where $R_0 = R$ and $R_1 = \{o \to b, o \to \neg f\}$ and $R_2 = \emptyset$. Notice that $\rho(b) = 0$ so $I(b) = R_0 = R$ and therefore $R \circ_{rk} \{b\} = R \circ_h \{b\} = R \cup \{b\}$ and $C_{fc}(R \circ_h \{b\}) = \{b, f, w\}$.

For the same R, an easy computation shows that $\rho(o) = 1$. Since the set $\{o \to b, o \to \neg f, b \to w\}$ is the unique extension of R_1 consistent with $\{o\}$ we have $h(o) = \{o \to b, o \to \neg f, b \to w\}$ so $R \circ_h \{o\} = \{o \to b, o \to \neg f, b \to w\} \cup \{o\}$. Since $\rho(o) = 1$ we have $R \circ_{rk} \{o\} = \{o \to b, o \to \neg f\} \cup \{o\}$. Therefore $C_{fc}(R \circ_h \{o\}) = \{b, o, \neg f, w\}$ but $C_{fc}(R \circ_{rk} \{o\}) = \{b, o, \neg f\}$.

Another classic taxinomic example (the calculations are left to the reader) is given by $R = \{m \to s, c \to m, c \to \neg s, n \to c, n \to s\}$ where m, s, c, n stand respectively for mollusc, shell, cephalopod and nautili. The base is $< R_0, R_1, R_2, R_3 >$ where $R_0 = R$, $R_1 = \{c \to m, c \to \neg s, n \to c, n \to s\}$ and $R_2 = \{n \to c, n \to s\}$ and $R_3 = \emptyset$.

We have $C_{fc}(R \circ_h \{n\}) = \{n, c, s, m\}$ and $C_{fc}(R \circ_{rk} \{n\}) = \{n, c, s\}$; this shows that the hull revision allows more inferences. In some other cases the revisions coincide, for instance $C_{fc}(R \circ_h \{c, \neg n\}) = \{c, \neg n, m, \neg s\} = C_{fc}(R \circ_{rk} \{c, \neg n\})$.

3.2 Computing hull revision

In this subsection we show how, via a simple coding, we can compute the hull revision by using the factual revision defined in section 2.

The base $< R_0, \ldots, R_n >$ of R is easily computed.

To compute the class of maximal subsets of R which are consistent with L and which contain R_k we use our update algorithm in the following way.

Let $\ell' : R \longrightarrow \{r_1, \ldots, r_m\}$ be a bijection where the r_i are new atoms. Define $\ell : R \cup L \longrightarrow \{r_1, \ldots, r_m\} \cup L$ by $\ell(r) = \ell'(r)$ if $r \in R$ and $\ell(a) = a$ if $a \in L$. Let $M(R)$ be the modification of R in the following way: each rule $L \to l$ of R is replaced by the rule $r, L \to l$ where $r = \ell(L \to l)$. Note that the maximal subsets of R which are consistent with L and which contain $R_{\rho(L)}$ are then those corresponding to the maximal subsets of the base of facts $\ell(R)$ computed as being its possible updatings with respect to $M(R)$ by $L \cup \ell(R_{\rho(L)})$. More precisely we have:

$$R \circ_h L = \ell^{-1}\{ \bigcap [\ell(R) \diamond_{M(R)} (L \cup \ell(R_{\rho(L)}))\ ]\ \}$$

In order to illustrate this method take the following example: $R = \{b \to f, b \to w, o \to b, o \to \neg f\}$. Let $L = \{o\}$. Define $\ell : R \cup L \longrightarrow \{1, 2, 3, 4\} \cup L$ such that $M(R) = \{1, b \to f\ ,\ 2, b \to w\ ,\ 3, o \to b\ ,\ 4, o \to \neg f\}$. We have seen above that $R_{\rho(L)} = \{o \to b, o \to \neg f\}$ so $\ell(R_{\rho(L)}) = \{3, 4\}$. Therefore

$$\ell(R) \diamond_{M(R)} (L \cup \ell(R_{\rho(L)})) = \{1, 2, 3, 4\} \diamond_{M(R)} \{o\} \cup \{3, 4\} = \langle \{2, 3, 4, o\} \rangle$$

and so $R \circ_h L = \ell^{-1}(\{2, 3, 4, o\}) = \{o \to b, o \to \neg f, b \to w\} \cup \{o\}$.

4 Change properties for $\diamond_R$, $\circ_{rk}$ and $\circ_h$

Let R, L and $\mathcal{F}$ be a set of rules, a set of literals and a flock respectively. We say that L is a R-consequence of $\mathcal{F}$ iff $L \subseteq C^R_{fc}(\mathcal{F})$.

Let $\mathcal{F} = \langle F_1, \ldots, F_n \rangle$ be a flock. We put $\mathcal{F} \cup L = \langle F_1 \cup L, \ldots, F_n \cup L \rangle$

Theorem 7. *The operator $\diamond_R$ is a syntactical update operator. More precisely, it satisfies the relativized versions of* **U1**, **U3**, **U5**, **U6** *and* **U8** *to the notions of R-consequence and R-consistency. Moreover $\diamond_R$ satisfies a weak version of* **U2** *in which we suppose that the old knowledge L is R-consistent.*

Proof: U1: We want to show that L' is a R-consequence of $L \diamond_R L'$, *i.e.* that L' is a subset of $C^R_{fc}(L \diamond_R L')$. Assume that $L \diamond_R L' = \langle F_1, \ldots, F_n \rangle$. By definition of $\diamond_R$, we have $L' \subseteq F_i$ for $i = 1, \ldots, n$. Therefore $L' \subseteq \cap_{i=1}^n C_{fc}(R \cup F_i) = L \diamond_R L'$.
U2 weak: Assume that L is R-consistent. Suppose that L' is a R-consequence of L. We want to show that the R-consequences of L and $L \diamond_R L'$ are equal, *i.e.* $C_{fc}(R \cup L) = C^R_{fc}(L \diamond_R L')$. By the assumption, it is clear that $L \diamond_R L' = \langle L \cup L' \rangle$. But $C^R_{fc}(\langle L \cup L' \rangle) = C_{fc}(R \cup L \cup L') = C_{fc}(R \cup L)$, the last equality is true by the supposition. We conclude by transitivity.
U3: If L and L' are R-consistent by definition it is clear that $L \diamond_R L'$ is R-consistent. In fact if L' is R-consistent then $L \diamond_R L'$ is R-consistent.
U5: We want to show that the R-consequences of $L \diamond_R (L' \cup L'')$ are a subset of the R-consequences of $(L \diamond_R L') \cup L''$. If $L' \cup L''$ is not R-consistent there is nothing to prove. Thus suppose that $L' \cup L''$ is R-consistent. By definition we have

$$\begin{aligned} L \diamond_R (L' \cup L'') &= \langle L_1, \ldots, L_n \rangle \cup L' \cup L'' \\ (L \diamond_R L') \cup L'' &= \langle K_1, \ldots, K_p \rangle \cup L' \cup L'' \end{aligned}$$

where L_i is a maximal subset of L such that $L_i \cup L' \cup L''$ is R-consistent, for $i = 1, \ldots, n$, and K_j is a maximal subset of L such that $K_j \cup L'$ is R-consistent, for $j = 1, \ldots, p$. Thus, for each K_j we have either $K_j \cup L' \cup L''$ is not R-consistent and $C_{fc}(R \cup K_j \cup L' \cup L'') = Lit$ or $K_j = L_i$ for one i. Therefore if $l \in \bigcap_{i=1}^n C_{fc}(R \cup L_i \cup L' \cup L'')$ then $l \in \bigcap_{j=1}^p C_{fc}(R \cup K_j \cup L' \cup L'')$.
U6: Suppose that $L_2 \subseteq C^R_{fc}(L \diamond_R L_1)$ and $L_1 \subseteq C^R_{fc}(L \diamond_R L_2)$. We want to see that $C^R_{fc}(L \diamond_R L_1) = C^R_{fc}(L \diamond_R L_2)$. Put

$$\begin{aligned} L \diamond_R L_1 &= \langle L^1_1, \ldots, L^1_{n_1} \rangle \cup L_1 \\ L \diamond_R L_2 &= \langle L^2_1, \ldots, L^2_{n_2} \rangle \cup L_2 \end{aligned}$$

where L^1_i is a maximal subset of L such that $L^1_i \cup L_1$ is R-consistent for $i = 1, \ldots, n_1$ and L^2_j is a maximal subset of L such that $L^2_j \cup L_2$ is R-consistent for $j = 1, \ldots, n_2$. From the hypothesis it is easy to see that:
(a) $L_1 \cup L^2_j$ is R-consistent for $j = 1, \ldots n_2$ and
(b) $L_2 \cup L^1_i$ is R-consistent for $i = 1, \ldots n_1$.
From (b) we have $\forall\ i \in \{1, \ldots, n_1\}\ \exists\ j \in \{1, \ldots, n_2\}$ such that $L^1_i \subseteq L^2_j$ and from (a) we have $\forall\ j \in \{1, \ldots, n_2\}\ \exists\ i \in \{1, \ldots, n_1\}$ such that $L^2_j \subseteq L^1_i$. But this implies, by maximality of sets L^1_i and L^2_j, that $n_1 = n_2$ and there is a permutation σ of $\{1, \ldots, n_1\}$ such that $L^1_i = L^2_{\sigma(i)}$. Without loss of generality we can suppose that σ is the identity. Finally note that

$$\begin{aligned} C^R_{fc}(\langle L^1_1, \ldots, L^1_{n_1} \rangle \cup L_1) &= C^R_{fc}(\langle L^1_1, \ldots, L^1_{n_1} \rangle \cup L_1 \cup L_2) \\ &= C^R_{fc}(\langle L^2_1, \ldots, L^2_{n_1} \rangle \cup L_2 \cup L_1) \\ &= C^R_{fc}(\langle L^2_1, \ldots, L^2_{n_1} \rangle \cup L_2) \end{aligned}$$

U8: It is trivially verified by definition. □

Note that the postulate **U7** does not make sense in our setting because we have no disjunction of facts.

Remark also that the operator $\diamond_R$ satisfies **R3** (*if the new information is consistent, the result is consistent*) and this is in fact incompatible with the whole version of **U2**. This observation can be seen as an advantage of the operator $\diamond_R$.

Theorem 8. *The operator $\circ_{rk}$ is a syntactical revision operator. More precisely, it satisfies the relativized versions of* **R1**, **R2**, **R3**, **R5** *and* **R6** *to the notions of consequence and consistency by forward chaining.*

The operator $\circ_h$ satisfies the relativized versions of **R1**, **R2** *and* **R3**.

Proof: We do the verifications for $\circ_h$ concerning the postulates **R1**, **R2**, **R3** (the postulates for $\circ_{rk}$ are verified in an analogous way). Then we verify **R5** and **R6** for $\circ_{rk}$

R1: We want to show that L is a fc-consequence of $R \circ_h L$. This is true because either $R \circ_h L = h(L) \cup L$ or $R \circ_h L = L$ and $C_{fc}(R \circ_h L)$ contains L in both cases.

R2: Suppose that L is R-consistent. We want to show that $R \circ_h L = R \cup L$. This is straightforward by definition.

R3: Suppose L consistent. We want to show that $R \circ_h L$ is also consistent. This is true because either $R \circ_h L = h(L) \cup L$ and $C_{fc}(h(L) \cup L)$ is consistent by definition or $R \circ_h L = L$ and L is consistent by hypothesis.

R5 and **R6** for $\circ_{rk}$: We suppose that $(R \circ_{rk} L) \cup L'$ is fc-consistent (otherwise **R5** is trivial). It is enough to show that $R \circ_{rk} (L \cup L') = (R \circ_{rk} L) \cup L'$. By hypothesis $(R_{\rho(L)} \cup L) \cup L'$ is fc-consistent. Thus $\rho(L \cup L') \leq \rho(L)$ and then $\rho(L \cup L') = \rho(L)$. From this we conclude easily. □

Observation 9 *The hull operator does not satisfy neither* **R5** *nor* **R6**.

Proof: In order to show that **R5** does not hold it is enough to consider the following example: the set of rules R is defined by $R = \{b \to w, w \to w', w' \to f, o \to b, o \to \neg f\}$. The base is in this case $\langle R_0, R_1, R_2 \rangle$ with $R_0 = R$, $R_1 = \{o \to b, o \to \neg f\}$ and $R_2 = \emptyset$. Put $L = \{o\}$ and $L' = \{w'\}$. It is not hard to establish that $h(L) = R_1$ and $h(L \cup L') = R_1 \cup \{b \to w, w \to w'\}$. Thus $C_{fc}((R \circ_h L) \cup L') = \{o, w', b, \neg f\}$ and $C_{fc}(R \circ_h (L \cup L')) = \{o, w', b, \neg f, w\}$. Therefore $C_{fc}(R \circ_h (L \cup L')) \not\subseteq C_{fc}((R \circ_h L) \cup L')$, that is **R5** does not hold.

To prove that **R6** does not hold we consider the following example: Put $R = \{r_0, r_1, r_2\}$ where $r_0 = a \to c$, $r_1 = e \to \neg c$, $r_2 = b \to \neg c$. Put $L = \{a, e\}$ and $L' = \{b\}$. The base for R is $\langle R, \emptyset \rangle$. Then it is easy to see that $R_{\rho(L)} = R_{\rho(L \cup L')} = \emptyset$ and

$$\begin{aligned} I(L) &= \{\{r_0, r_2\}, \{r_1, r_2\}\} \quad \text{and} \\ I(L \cup L') &= \{\{r_0\}, \{r_1, r_2\}\} \end{aligned}$$

Thus $h(L) = \{r_2\}$ and $h(L \cup L') = \emptyset$. Therefore $\neg c \in C_{fc}((R \circ_h L) \cup L')$ and $\neg c \notin C_{fc}(R \circ_h (L \cup L'))$, so **R6** fails. □

Conclusion

We have proposed in this paper three knowledge change operators based on forward chaining. The aim of this work was to study change operators with nice logical properties and a simple (operational) semantics. We have shown that the three operators defined have desirable properties. In particular we have proved that factual update can be seen as an update operator, in the Katsuno and Mendelzon sense, and that ranked revision is, up to syntax independence axiom, a revision operator, in the Alchourrón-Mendelzon-Gärdenfors sense. These operators, based on forward chaining, are easily computable, in particular the operator of ranked revision is polynomial. The hull revision is a bit more complicated but it can be computed with the help of the two other operators.

It would be interesting to extend the factual revision definition to allow rule revision when no factual revision is possible.

References

1. C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change. *Journal of Symbolic Logic*, 50:510–530, 1985.
2. H. Bezzazi and R. Pino Pérez. Rational transitivity and its models. In *Proc. of the Twenty-Sixth International Symposium on Multiple-Valued Logic*, Santiago de Compostela, Spain, May, 1996, pp. 160–165, IEEE Computer Society Press.
3. H. Bezzazi, D. Makinson, and R. Pino Pérez. Beyond rational monotony: some strong non-horn rules for nonmonotonic inference relations. *Journal of Logic and Computation*, 199?. To appear.
4. M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proc. of the Seventh National Conference on Artificial Intelligence (AAAI'88)*, pp. 475–479, 1988.
5. C. Damasio, W. Nedjdl, and L. M. Pereira. Revise: An extended logic programming system for revising knowledge bases. In *Proc. of the 4th International Conference on Principles of Knowledge Representation and Reasoning*, pp. 607-618, Morgan Kaufmann, 1994.
6. T. Eiter and G. Gottlob. On the complexity of propositional knowledge base revision, updates and counterfactuals. In *Proc. of 11th Symposium on Principles of Database Systems*, pp. 261–273, ACM Press, 1992.
7. R. Fagin, G. Kuper, J.D. Ullman, and M.Y. Vardi. Updating logical databases. *Advances in Computing Research*, 3:1–18, 1986.
8. P. Gärdenfors. *Knowledge in Flux: modeling the dynamics of epistemic states.* MIT press, Cambridge, MA, 1988.
9. P. Gärdenfors and D. Makinson. Relations between the logic of theory change and nonmonotonic logic. In *The Logic of Theory Change, Workshop, Konstanz, FRG, Octuber 1989*, pages 185–205. Springer-Verlag, 1989. Lecture Notes in Artificial Intelligence 465.
10. H. Katsuno and A.O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.
11. H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge database and revising it. In P. Gärdenfors, editor, *Belief Revision*. Cambridge Uni-

versity Press, Cambridge, 1992. Cambrigde tracts in theoretical computer science, # 29.
12. D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence*, 55:1–60, 1992.
13. V. Marek and M.Truszczynski. Revision programming, Database Updates and Integrity Constraints. In *Proc. of 5th International Conference of Database Theory*, Prague, Czech Republic, January 11-13, 1995. Lecture Notes in Computer Science, Vol. 893, Springer, 1995, pages 368–382.
14. R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32:57–95, 1987.