Revising Non-Monotonic Rule-Based Belief Databases

Ricardo Wehbe

Institut für Informatik und angewandte Mathematik, Universität Bern,
Neubrückstrasse 10, CH-3012 Bern, Switzerland. E-mail: wehbe@iam.unibe.ch.

Abstract. The update of a belief database with disjunctive knowledge which is contradictory with the beliefs of an agent poses some problems when we want to perform belief-revision with minimal changes. We define a representation of beliefs based on rules. Clauses have already been used to represent beliefs. We combine them with defaults, which allows a simple treatment of disjunctive beliefs. The behaviour of the system is analysed in view of the Katsuno-Mendelzon postulates.

1 Introduction

The update of a belief database with disjunctive knowledge which is contradictory with the beliefs of an agent poses some problems when we want to perform belief-revision with minimal changes [6]. We define a representation of beliefs as a set of rules. Clauses have already been used to represent beliefs [2, 11]. We combine them with defaults, which allows a simple treatment of disjunctive beliefs. The representation is fairly simple, both theoretically and algorithmically, and has some nice properties regarding the Katsuno-Mendelzon [7] postulates.

The paper is organised as follows: in section 2 we describe the representation and review briefly some of its most important theoretical aspects. In section 3 we describe the procedure to update a set of rules. In section 4 we analyse the behaviour of the system with respect to the Katsuno-Mendelzon postulates, and in section 5 we present the conclusions and some lines of future research.

2 Representation of Beliefs

Beliefs will be represented by a set of rules, which will be either *clauses* or *defaults*. We assume the existence of a set Π of propositional symbols, whose elements will be denoted by p. An *atom* will be either a symbol of Π or its negation; we will thus have *positive*, *negative*, and *complementary atoms*. The set of all atoms that can be formed with the propositions of Π will be denoted by $\mathsf{AT}(\Pi)$. A set of rules ζ will be said to be *based* on Π if all propositional symbols occurring in ζ belong to Π. All our sets of rules will be based on Π, unless explicitly stated otherwise.

The syntax of the language is given by the following grammar:

$$\begin{aligned}\mathsf{atom} &::= \mathsf{p} \mid \overline{\mathsf{p}}\\ \mathsf{sequent} &::= \varepsilon \mid \mathsf{atom}[,\mathsf{atom}]\\ \mathsf{clause} &::= \mathsf{sequent} \rightarrow \mathsf{atom}\\ \mathsf{default} &::= \mathsf{sequent} \rightsquigarrow \mathsf{atom}\end{aligned}$$

We will omit the arrow in clauses with an empty sequent (also called *facts*).

We will use the notation $\zeta = (\varphi, \delta)$ to denote the set of rules ζ consisting of the set of clauses φ and the set of defaults δ. The intended meaning of a clause $\Gamma \rightarrow \mathsf{p}$ is that if all atoms in Γ are true, then p is also true. The intended meaning of a default $\Gamma \rightsquigarrow \mathsf{p}$ is that if all atoms in Γ are true and there is no proof of $\overline{\mathsf{p}}$, then p is taken to be true. This is a non-monotonic rule, since the addition of new facts may decrease the number of atoms that are true.

We will define the semantics of sets of rules in two steps. First we will define the semantics of sets of clauses and then the semantics of sets of general rules.

A *model* of a set of clauses φ is a maximal consistent set $\mathcal{M} \subseteq \mathsf{AT}(\Pi)$ such that for all clauses $\Gamma \rightarrow \mathsf{p} \in \varphi$, if $\Gamma \subseteq \mathcal{M}$ then $\mathsf{p} \in \mathcal{M}$.

A set of clauses may have one model, several, or none. We will not be interested in the latter (inconsistent) group. An atom p will be said to be *bound* in a set of clauses φ if for all models $\mathcal{M}$ of φ, $\mathsf{p} \in \mathcal{M}$. A proposition $\mathsf{p} \in \Pi$ will be said to be *free* in φ if there are models $\mathcal{M}$, $\mathcal{M}'$ of φ such that $\mathsf{p} \in \mathcal{M}$ and $\overline{\mathsf{p}} \in \mathcal{M}'$. The *invariant* of a set of clauses φ, denoted by $\mathcal{J}(\varphi)$, is the set of all atoms that are bound in φ. By convention, if φ is inconsistent, then $\mathcal{J}(\varphi) = \mathsf{AT}(\Pi)$.

We will use the operator $T_\varphi : 2^{\mathsf{AT}(\Pi)} \mapsto 2^{\mathsf{AT}(\Pi)}$ to construct the invariant of a set of clauses φ. This operator, similar to the one used in classical logic programming [8], is defined as follows:

1. If $\Gamma \rightarrow \mathsf{p} \in \varphi$, and $\Gamma \subseteq S$, then $\mathsf{p} \in T_\varphi(S)$.
2. If $\Gamma, \mathsf{p} \rightarrow \mathsf{q} \in \varphi$, $\Sigma, \overline{\mathsf{p}} \rightarrow \mathsf{q} \in \varphi$, and $\Gamma, \Sigma \subseteq S$, then $\mathsf{q} \in T_\varphi(S)$.

This operator may be iteratively constructed as $T_\varphi \uparrow \omega$, where

$$\begin{aligned}T_\varphi \uparrow 0 &= \emptyset\\ T_\varphi \uparrow (n+1) &= T_\varphi(T_\varphi \uparrow n)\end{aligned}$$

It is possible to show that this set may be computed within a finite number of steps and that all atoms in $T_\varphi \uparrow \omega$ are in the invariant [12].

This operator alone does not suffice to construct the set of invariants (unless we reject the law of the excluded middle, which we do not do here.) For instance, if we have $\varphi = \{\overline{\mathsf{p}} \rightarrow \mathsf{q}, \overline{\mathsf{p}} \rightarrow \overline{\mathsf{q}}\}$, we would have $\mathcal{J}(\varphi) = \{\mathsf{p}\}$, although $T_\varphi \uparrow \omega = \emptyset$.

We have thus to "complete" the invariant with the atoms not captured by the T_φ operator. We define the *reduct* of a set of clauses φ, by a set of atoms $A \subseteq \mathsf{AT}(\Pi)$, denoted by $\mathcal{R}_A(\varphi)$ as the set of clauses constructed from φ as follows:

1. Eliminate all clauses $\Gamma \rightarrow \mathsf{p} \in \varphi$ with $\mathsf{p} \in A$.

2. Eliminate all clauses $\Gamma, \overline{\mathsf{p}}, \Sigma \rightarrow \mathsf{q} \in \varphi$ with $\mathsf{p} \in A$.
3. Eliminate from the remaining clauses all other occurrences of atoms in A.
4. Rewrite all clauses $\Gamma, \overline{\mathsf{p}} \rightarrow \overline{\mathsf{q}} \in \varphi$ with $\mathsf{q} \in A$ as $\Gamma \rightarrow \mathsf{p}$.

We will be interested in the reduct $\mathcal{R}_{T_\varphi \uparrow \omega}(\varphi)$, which we will denote by $\mathcal{R}(\varphi)$. The procedure described next constructs the invariant in a finite number of steps. Details and proofs are in [12]. The construction is based on the following results:

1. If no complementary atoms occur in φ, then $\mathcal{J}(\varphi) = T_\varphi \uparrow \omega$.
2. $\mathcal{J}(\varphi) = T_\varphi \uparrow \omega \cup \mathcal{J}(\mathcal{R}(\varphi))$.
3. If p, $\overline{\mathsf{p}}$ occur in φ, then $\mathcal{J}(\varphi) = \mathcal{J}(\varphi \cup \{\mathsf{p}\}) \cap \mathcal{J}(\varphi \cup \{\overline{\mathsf{p}}\})$.

These results suggest a recursive way to construct the invariant. We construct a binary tree as follows:

1. The root (level 0) is labelled $(T_\varphi \uparrow \omega, \mathcal{R}(\varphi))$.
2. The leaves are the nodels labelled $(S, \mathcal{R})$ where $\mathcal{R}$ has no occurrence of complementary atoms.
3. A node $(S, \mathcal{R})$ (level n) where atoms p, $\overline{\mathsf{p}}$ occur in $\mathcal{R}$ has successor nodes (level $n+1$) $(T_{\mathcal{R}_1} \uparrow \omega, \mathcal{R}(\mathcal{R}_1))$ and $(T_{\mathcal{R}_2} \uparrow \omega, \mathcal{R}(\mathcal{R}_2))$, where $\mathcal{R}_1 = \mathcal{R} \cup \{\mathsf{p}\}$ and $\mathcal{R}_1 = \mathcal{R} \cup \{\overline{\mathsf{p}}\}$.

Then, for each node $\nu = (S, \mathcal{R})$ we compute recursively:

1. If ν is a leaf node, then $\mathcal{J}(\nu) = S$.
2. Otherwise, if ν has successors ν_1 and ν_2, then $\mathcal{J}(\nu) = S \cup (\mathcal{J}(\nu_1) \cap \mathcal{J}(\nu_2))$.

The soundness of this process is shown in [12]. The depth of the tree is bounded by the number of pairs of complementary atoms, since we eliminate a pair each time we go to the higher level. We will assign to each atom p a pair (n, s), where s is the depth of the node in which p occurs and n corresponds to the iteration of the $T_\varphi \uparrow \omega$ operator in which p was obtained. This induces a total order on the pairs (and consequently on the atoms of the invariant.)

$$(n_1, s_1) \sqsubset (n_2, s_2) \text{ if } s_1 < s_2$$
$$(n_1, s) \sqsubset (n_2, s) \text{ if } n_1 < n_2$$

This induces also a partial order on the rules. This completes the first step. Now we will define the semantics of general sets of rules, i.e., with clauses and defaults. Observe that after having obtained the invariant of the set of rules, we may work with $\mathcal{R}_{\mathcal{J}(\varphi)}(\varphi)$ instead of φ.

The semantics of general sets of rules is given by *extensions*. Given a set of rules $\zeta = (\varphi, \delta)$ based on Π, a set $S \subseteq \mathsf{AT}(\Pi)$ is an extension for ζ iff it is a fixpoint of the operator $\Lambda : 2^{\mathsf{AT}(\Pi)} \mapsto 2^{\mathsf{AT}(\Pi)}$, defined as the smallest set such that the following properties hold:

1. $\mathcal{J}(\varphi) \subseteq \Lambda_\zeta(S)$
2. If $p \in \mathcal{J}(\varphi \cup \Lambda_\zeta(S))$ then $p \in \Lambda_\zeta(S)$
3. If $\Gamma \rightsquigarrow p \in \zeta$, and $\Gamma \in \Lambda_\zeta(S)$ and $\overline{p} \notin S$, then $p \in \Lambda_\zeta(S)$

This definition is a special case of the definition given in [4] or [10] adapted to the case of invariants.

Example 1. Let ζ be

$$\begin{array}{ll} \rightsquigarrow \mathsf{p} & \rightsquigarrow \mathsf{q} \\ \mathsf{p} \rightsquigarrow \overline{\mathsf{q}} & \mathsf{q} \rightsquigarrow \overline{\mathsf{p}} \end{array}$$

We have the following extensions for this set of rules: $S_1 = \{\mathsf{p}, \mathsf{q}\}$, $S_2 = \{\mathsf{p}, \overline{\mathsf{q}}\}$, and $S_3 = \{\overline{\mathsf{p}}, \mathsf{q}\}$. This corresponds to a situation in which the agent believes $\mathsf{p} \vee \mathsf{q}$.

We will restrict our sets of rules to *normal forms*, in which defaults will appear only when we want to express disjunctive belief. A *(disjunctive) cluster* for a set of atoms S is a set of defaults $\mathcal{D}(S)$ such that (1) for any atom $\mathsf{p} \in S$ there is a default $\rightsquigarrow \mathsf{p} \in \mathcal{D}(S)$, and (2) for any pair of distinct atoms $\mathsf{p}, \mathsf{q} \in S$ there is a default $\mathsf{p} \rightsquigarrow \overline{\mathsf{q}}$.

Observe that the number of extensions increments exponentially with the number of elements in the disjunction. But this reflects the fact that if we believe $\mathsf{p_1} \vee \cdots \vee \mathsf{p_n}$ there are $2^n - 1$ possible scenarios that we should take into account. The set of defaults of example 1 represents a cluster for the set $\{\mathsf{p}, \mathsf{q}\}$.

We will not interested in arbitrary extensions, but in *admissible extensions*, which are exactly those extensions where at least one default of each cluster has been applied. Note that a set of rules may have no admissible extension, even if it has an extension. For instance, the clusters corresponding to the sets $\{\mathsf{p}, \mathsf{q}\}$, $\{\mathsf{p}, \overline{\mathsf{q}}\}$, $\{\overline{\mathsf{p}}, \mathsf{q}\}$, and $\{\overline{\mathsf{p}}, \overline{\mathsf{q}}\}$ cannot be satisfied simultaneously. A set of rules that has an admissible extension is *admissible*.

A set of rules $\zeta = (\varphi, \delta))$ is in *normal form* if the following conditions hold:

1. The set of rules ζ is admissible.
2. All defaults in δ belong to some cluster $\mathcal{D}(S)$ and $\mathcal{D}(S) \subseteq \delta$.
3. No atom in $\mathcal{J}(\varphi)$ is included in any cluster in δ.
4. There are no clusters $\mathcal{D}(S_1)$ and $\mathcal{D}(S_2)$ with $S_1 = \{\mathsf{p}\} \cup S$ and $S_2 = \{\overline{\mathsf{p}}\} \cup S$.
5. There are no clusters $\mathcal{D}(S_1)$ and $\mathcal{D}(S_2)$ with $S_1 \subseteq S_2$.

The first condition says that we allow no nonsense. The second one states that we have only "complete" disjunctive clusters. The other conditions are meant to avoid redundancies.

To comply with rules 3-5 simple syntactic transformation may be applied. For rule 3, we eliminate all atoms already in the invariants from the clusters. In section 3 we will see how to do this. For rule 4, we eliminate in the same way the complementary atoms from the clusters and for rule 5 we eliminate the weaker cluster ($\mathcal{D}(S_2)$ is *weaker* than $\mathcal{D}(S_1)$ if $S_1 \subseteq S_2$.)

We will not be interested in constructing the whole set of extensions for a given set of rules ζ, but in answering the question, given an atom p, if there is some admissible extension S for ζ such that $\mathsf{p} \in S$. The process to achieve that will be similar to the bottom-up "Prop-Log" interpreter of [9] but with an additional "fairness" constraint to include admissibility: we require for each cluster either that a default be applied during the computation or that a default remains applicable after it. The reason for this will be explained later. Our approach will be bottom-up, starting from the defaults.

Given a set of clusters $\delta = \{\mathcal{D}(S_1), \ldots, \mathcal{D}(S_n)\}$, a *pattern* for δ will be a set of atoms $P_\delta \subset \bigcup_i (S_i)$ such that for each S_i there is at least one atom p such that $\mathsf{p} \in S_i$ and $\mathsf{p} \in P_\delta$. In other words, a pattern is a set of atoms which satisfies the conjuction of the disjunctions of the sets S_i. Note that a set of rules $\zeta = (\varphi, \delta)$ is admissible only if φ is consistent and δ has a pattern. But these conditions alone do not guarantee admissibility.

The procedure to decide whether an atom p belongs to an extension of the set of rules $\zeta = (\varphi, \delta)$ is the following:

1. If $\mathsf{p} \in \mathcal{J}(\varphi)$, finish with success;
2. Otherwise, select a pattern P for the clusters in δ such that $\mathsf{p} \in \mathcal{J}(\mathcal{R}_{\mathcal{J}(\varphi)}(\varphi) \cup P)$; if there is such a pattern, finish with success; otherwise, finish with failure.

If p belongs to some extension, it must either belong to the invariant or to there must be some pattern satisfying the condition stated in the second step of, since otherwise no extension would include p. The following theorem guarantees that this process indeed yields an extension for ζ.

Theorem 1. *Let $\zeta = (\varphi, \delta)$ be a set of rules in normal form and let P be a pattern for the clusters in δ. Then $\mathcal{J}(\varphi \cup P)$ is an extension.*

Proof. First we show that $\Lambda_\zeta(\mathcal{J}(\varphi \cup P)) \subseteq \mathcal{J}(\varphi \cup P)$. We have by monotonicity that $\mathcal{J}(\varphi) \subseteq \mathcal{J}(\varphi \cup P)$ and we have that $\mathcal{J}(\varphi \cup \mathcal{J}(\varphi \cup P)) = \mathcal{J}(\varphi \cup P)$ (see [12]). Assume that there is default $\Gamma \rightsquigarrow \mathsf{p} \in \delta$ with $\Gamma \subseteq \mathcal{J}(\varphi \cup P)$ and $\overline{\mathsf{p}} \notin P$. Then, since P is a pattern, $p \in P$ and thus $p \in \mathcal{J}(\varphi \cup P)$. The result follows from the minimality of $\Lambda_\zeta(\mathcal{J}(\varphi \cup P))$.

Assume now that $\mathcal{J}(\varphi \cup P) \not\subseteq \Lambda_\zeta(\varphi \cup P)$. Assume $P = \{p_1, \ldots, p_n\}$. Construct the sequence $S_0, \ldots, S_n$ where $S_0 = \mathcal{J}(\varphi)$, $S_i = \mathcal{J}(\varphi \cup \{\mathsf{p}_1, \ldots, \mathsf{p}_i\})$, and $S_n = \mathcal{J}(\varphi \cup P)$. By monotonicity $S_i \leq S_j$ for any $0 \leq i < j \leq n$. Since $S_0 \subseteq \Lambda_\zeta(\mathcal{J}(\varphi \cup P))$, there must be some $S_{k+1} \not\subseteq \Lambda_\zeta(\mathcal{J}(\varphi \cup P))$ even though $S_k \subseteq \Lambda_\zeta(\mathcal{J}(\varphi \cup P))$. Thus there is some default $\Gamma \rightsquigarrow \mathsf{p}$ such that either $\Gamma \not\subseteq \Lambda_\zeta(\mathcal{J}(\varphi \cup P))$ or overline$\mathsf{p} \notin \Lambda_\zeta(\mathcal{J}(\varphi \cup P))$. The former case is ruled out by the first part of the proof, and the latter case is ruled out, since otherwise we would have that $\mathcal{J}(\varphi \cup P)$ is inconsistent.

3 Changing Belief in a Set of Rules

The beliefs represented by the set of rules ζ will change "as times goes by." This will be reflected by some actions that will be performed on ζ. The atomic actions are the addition or deletion of an atom, and the addition or deletion of a disjunctive cluster. A complex action will be a combination of atomic actions.

Assume now we have a set of rules ζ. Following the terminology in [1], we will have:

1. Expansions: new beliefs are added to the set of rules, either as clauses or as defaults without clashing with the old ones. Given a formula s, we denote this by $\zeta \oplus s$.
2. Contractions: beliefs are retracted from the set of rules. The contraction of a formula s from ζ is denoted by $\zeta \ominus s$.
3. Belief revision: new beliefs that are inconsistent with the old ones are added to the set of rules. In this case, we have to make some contractions to accomodate the new beliefs. The revision of ζ to accept the formula s is denoted by $\zeta \odot s$.

Retraction is a key feature, since revision could be defined through contraction using the Levi identity [1]: $\zeta \odot s = (\zeta \ominus \overline{s}) \oplus s$. Since expansion is not problematic, contraction is the crucial operation.

3.1 Expansion

Although expansion poses no complex problems, the set of defaults may change as a result of the addition of new atoms or clusters. These changes will be only of syntactic nature and wil be performed to optimise the set of rules.

The rules that may be applied to a set of rules $\zeta = (\varphi, \delta)$ after an expansion are the following:

1. If $\mathsf{p} \in \mathcal{J}(\varphi)$, eliminate $\overline{\mathsf{p}}$ from δ.
2. If $\mathsf{p} \in \mathcal{J}(\varphi)$, eliminate all clusters $\mathcal{D}(S)$ such that $\mathsf{p} \in S$ from δ.
3. If there are two clusters $\mathcal{D}(S_1), \mathcal{D}(S_2) \in \delta$ such that $S_1 = \{\mathsf{p}\} \cup S$ and $S_2 = \{\overline{\mathsf{p}}\} \cup S$, eliminate p from S_1 and $\overline{\mathsf{p}}$ from S_2.
4. If there are two clusters $\mathcal{D}(S_1), \mathcal{D}(S_2) \in \delta$ such that $S_1 \subseteq S_2$, eliminate S_2.

The elimination of a cluster is not problematic, since the set of rules remains in normal form. When we retract an atom from a cluster we have to ensure that what remains is still a complete cluster and not a torso. We delete an atom p from a cluster by deleting all defaults in which the atom appears either positively or negatively. If all that remains from the cluster is a single default $\rightsquigarrow \mathsf{q}$, we transform this in a clause q. This changes nothing, since all extensions would include q anyway. Otherwise, if the number of remaining defaults is bigger than one, these constitute still a cluster.

3.2 Contraction

Let us first consider the case where we want to contract a set of rules $\zeta = (\varphi, \delta)$ by retracting an atom p. There are two possibilities: either $\mathsf{p} \in \mathcal{J}(\varphi)$ or there is some cluster $\mathcal{D}(S) \subseteq \delta$ such that $\mathsf{p} \in S$.

In the first case, recall that we construct the invariant in successive steps. Each new subset of atoms that is added to the invariant is greater than the ones already there according to the ordering $\sqsubset$ define in section 2. An atom may appear as a consequence of one or two clauses. In the first case, we eliminate the clause that produced the atom; in the second case, we analyse the result of eliminating each one of the clauses and retain the one that produces a larger invariant. The elimination of a clause has no effects on the defaults, since no atom in the invariant appears in the defaults. Note that if two clauses contributed to the derivation of p, we eliminate always the bigger one with respect to $\sqsubset$, so as o minimise the effect of the elimination.

Example 2. Let $\zeta = (\varphi, \emptyset)$, where φ is:

$$\begin{aligned} &\mathsf{p} \\ \mathsf{p} &\rightarrow \mathsf{q} \\ \mathsf{p},\ \bar{\mathsf{r}} &\rightarrow \mathsf{s} \\ \mathsf{p},\ \mathsf{q},\ \bar{\mathsf{r}} &\rightarrow \bar{\mathsf{s}} \end{aligned}$$

And suppose that we want to compute $\zeta \ominus \mathsf{r}$. After construction of the invariant with the corresponding pairs (see section 2), we get:
$\mathcal{J}(\varphi) = \{\mathsf{p}_{(1,0)}, \mathsf{q}_{(2,0)}, \mathsf{r}_{(2,1)}\}$
Thus, we choose either the third or the fourth clause.

The retraction of a disjunction $\mathsf{p} \vee \mathsf{q}$ from $\zeta = (\varphi, \delta)$ amounts to the successive retraction of p and q. The retraction of a conjunction $\mathsf{p} \wedge \mathsf{q}$ poses some problems, since we want to preserve as much as possible of our old beliefs. A brute-force procedure retracting p and then q would be excessive, since we would retract more knowledge than necessary.

We have again several possibilities:

- Both p and q are in $\mathcal{J}(\varphi)$.
- Both p and q belong to some disjunctive cluster (possibly the same.)
- $\mathsf{p} \in \mathcal{J}(\varphi)$ and $\mathsf{q} \in \mathcal{D}(S)$.

In the first case, we perform $\zeta \ominus \mathsf{p}$ and $\zeta \ominus \mathsf{q}$ and then we compare the resulting invariants. We choose the case which yields the larger invariant.

If both p and q are obtained by defaults, we consider two cases. If both belong to the same cluster, just choose one of the atoms and eliminate it. If they belong to different clusters, we have that the agent believes $(\mathsf{p} \vee \mathsf{p}_1 \vee \cdots \vee \mathsf{p}_\mathsf{n}) \wedge (\mathsf{q} \vee \mathsf{q}_1 \vee \cdots \vee \mathsf{q}_\mathsf{m})$. To retract $\mathsf{p} \wedge \mathsf{q}$ amounts to rule out the possibility that both p and q simultaneously hold. Thus, we have to add the constraint $\overline{\mathsf{p}} \vee \overline{\mathsf{q}}$, which is the same as adding the cluster $\mathcal{D}(\{\overline{\mathsf{p}}, \overline{\mathsf{q}}\})$.

Example 3. Let $\zeta = (\emptyset, \delta)$ where δ consists of the clusters $\mathcal{D}\{\mathsf{p}, \mathsf{q}\}$ and $\mathcal{D}\{\mathsf{r}, \mathsf{s}\}$ and suppose that we want to retract $\mathsf{p} \wedge \mathsf{r}$. We add then the cluster $\mathcal{D}(\{\overline{\mathsf{p}}, \overline{\mathsf{r}}\})$ and we get:

$$\begin{array}{rrr} \rightsquigarrow \mathsf{p} & \rightsquigarrow \mathsf{r} & \rightsquigarrow \overline{\mathsf{p}} \\ \rightsquigarrow \mathsf{q} & \rightsquigarrow \mathsf{s} & \rightsquigarrow \overline{\mathsf{r}} \\ \mathsf{p} \rightsquigarrow \overline{\mathsf{q}} & \mathsf{r} \rightsquigarrow \overline{\mathsf{s}} & \overline{\mathsf{p}} \rightsquigarrow \mathsf{r} \\ \mathsf{q} \rightsquigarrow \overline{\mathsf{p}} & \mathsf{s} \rightsquigarrow \overline{\mathsf{r}} & \overline{\mathsf{r}} \rightsquigarrow \mathsf{p} \end{array}$$

Observe here that if we would not have imposed the "fairness" constraint, we would have no certainty that the condition $\overline{\mathsf{p}} \vee \overline{\mathsf{r}}$ is fulfilled, since any extension including p and q would make it impossible for any default of the new cluster to be applied.

If we cannot add a cluster $\mathcal{D}(S)$, with $S = \{\mathsf{p}_1, \dots, \mathsf{p}_\mathsf{n}\}$, then it is the case that $\overline{\mathsf{p}_1}, \dots, \overline{\mathsf{p}_\mathsf{n}}$ belong to all models; application of the simplification rules would turn them into clauses and we would be in the case of retracting a conjunction from the set of clauses.

Let us now assume that we have $\zeta = (\varphi, \delta)$ and there is some $\mathsf{p} \in \mathcal{J}(\varphi)$ and some there is some pattern that includes q. We want to retract $\mathsf{p} \wedge \mathsf{q}$ from ζ. In this case we retract the atom in the cluster. This is the same as adding the cluster $\mathcal{D}(\overline{\mathrm{p}}, \overline{\mathrm{q}})$, since the defaults corresponding to $\overline{\mathsf{p}}$ are eliminated by p, and all that remains is the default $\rightsquigarrow \overline{\mathsf{q}}$, which turns into the rule $\overline{\mathsf{q}}$, which in turn eliminates all defaults with q.

4 The Katsuno-Meldenzon Postulates

The Katsuno-Mendelzon postulates [7] have been used as a benchmark for update procedures [6]. We examine to which extent our system complies with these postulates for a set of rules $\zeta = (\varphi, \delta)$.

The first postulate ($\zeta \odot \psi$ implies ψ) is verified for atoms and for disjunctions of atoms, and inductively for any formula.

The second postulate (if ζ implies ψ, then $\zeta \odot \psi$ is equivalent to ψ) holds also for atoms (if $\mathsf{p} \in \mathcal{J}(\varphi)$, then $\zeta \odot \mathsf{p}$ is equivalent to ζ) and for disjunctions of atoms (if we have a cluster $\mathcal{D}(S) \subseteq \delta$, such that $\mathsf{p}_1, \dots, \mathsf{p}_\mathsf{m} \in S$, then $\zeta \odot \mathsf{p}_1 \vee \dots \vee \mathsf{p}_\mathsf{m}$ is equivalent to ζ, since the cluster is already there.)

The third postulate (if ζ and ψ are satisfiable, then also is $\zeta \odot \psi$) is also fulfilled: if ζ and ψ are consistent, then $\zeta \odot \psi$ is consistent.

The fourth postulate (if $\zeta_1 \equiv \zeta_2$ and $\psi_1 \equiv \psi_2$ requires a notion of equivalence for sets of rules. A static notion is not enough, since two sets of rules may have the same invariant but behave differently under updates. Take for instance $\varphi_1 = \{\mathsf{p}, \mathsf{p} \rightarrow \mathsf{q}\}$ and $\varphi_2 = \{\mathsf{p}, \mathsf{q}\}$; although $\mathcal{J}(\varphi_1) = \mathcal{J}(\varphi_2)$, $\varphi_1 \ominus \mathsf{p} = \emptyset$ and $\varphi_2 \ominus \mathsf{p} = \{\mathsf{q}\}$.

We will say that two sets of rules are equivalent iff they are identical. Then it is clear that the fourth postulate is fulfilled for atoms and disjunctions of atoms, and inductively for any complex formula. Note that the fourth postulate is only fulfilled if we adopt this rather strong notion of equivalence. This is because the implicit representation of information with set of rules is, in a dymamic context, dependent of the syntax.

For the fifth postulate (($\zeta \odot \psi) \wedge \xi$ implies $\zeta \odot (\psi \wedge \xi)$ we will consider two cases: either ξ is inconsistent with $\zeta \odot \psi$, in which case the implication is trivially fulfilled, or it is not. In the latter case, we have $\zeta \odot (\psi \wedge \xi) = (\zeta \odot \psi) \odot \xi$.

The sixth postulate (if $\zeta \odot \psi_1$ implies ψ_2 and $\zeta \odot \psi_2$ implies ψ_1 then $\zeta \odot \psi_1 \equiv \zeta \odot \psi_2$) is also fulfilled. The proofs are relatively easy. We do not include them for lack of space.

We will not consider the seventh postulate (if ζ is complete, then the conjunction of $\zeta \odot \psi_1$ and $\zeta \odot \psi_2$ implies $\zeta \odot (\psi_1 \vee \psi_2)$), since the notion of *completeness* does not seem to make much sense in a context with incomplete information.

The eighth postulate ($\zeta_1 \vee \zeta_2 \odot \psi$ implies $\zeta_1 \odot \psi \vee \zeta_2 \odot \psi$) is fulfilled by the definition of update.

5 Conclusions and Future Work

We have presented a frame to represent beliefs by means of sets of rules (clauses and defaults.) The defaults include the syntactic restriction that they may only appear in the form of clusters, which represent disjunctive information. This allows a relatively simple way to handle updates, especially when we have to retract a conjunction as a result of an update and the behaviour of the systems is quite acceptable in terms of the Katsuno-Mendelzon postulates.

Complexity problems may turn out hard if too many clusters occur. Possible solutions include the use of agent with limited reasoning capabilities [2] or interleaving updating and reasoning [11]. We are also currently interested in extending our approach to include introspection and *common knowledge* [3, 5].

References

1. Alchourrón, C.; Gärdenfors, P.; Makinson, D.: *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*, The Journal of Symbolic Logic 50 (2), 1985, pp. 510–530.
2. Alechina, N.; Jago, M.; Logan, B.: *Modal Logics for Communicating Rule-Based Agents*, Proc. of the 17th European Conf. on Artificial Intelligence (ECAI'06), IOS Press, 2006, pp. 322–326.
3. Baltag, A.; Moss, L.; Solecki, S.: *The Logic of Public Announcements, Common Knowledge, and Private Suspicions*,CWI Technical Report SEN-R9922, Amsterdam, 1999.
4. Besnard, Ph.: *An Introduction to Default Logic*, Springer Verlag, 1989.
5. Fagin, R.; Halpern, J.; Moses, Y.; Vardi, M.: *Reasoning About Knowledge*, The MIT Press, 1996.
6. Herzig, A.; Rifi, O.: *Propositional Belief Base Update and Minimal Change*, Artificial Intelligence 115 (1), 1999, pp. 107–138.
7. Katsuno, H.; Mendelzon, A.: *On the Difference between Updating a Knowledge Base and Revising it*, Proc. of the 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR-1991), Morgan-Kaufmann, 1991, pp. 387–394.
8. Lloyd, J.W.: *Foundations of Logic Programming*, Springer Verlag, 1987.
9. Maier, D.; Warren, D.: *Computing with Logic: Logic Programming with Prolog*, The Benjamin / Cummings Publishing Company Ltd., 1988.
10. Marek, W.; Truszczyński, M.: *Nonmonotonic Logic*, Springer-Verlag, 1993.
11. Sadri, F.; Toni, F.: *Interleaving Belief Updating and Reasoning in Abductive Logic Programming*, Proc. of the 17th European Conf. on Artificial Intelligence (ECAI'06), IOS Press, 2006, pp. 442–446.
12. Wehbe, R.: *A Hybrid Representation of Knowledge and Belief*, Workshop on Formal Approaches to Multi-Agent Systems (FAMAS'06), Riva del Garda, 2006.