Revision and Update based on Stratified Forward Chaining

El-Hassan BEZZAZI
LIFL U.A. 369 du CNRS, Université de Lille I
59655 Villeneuve d'Ascq, FRANCE
E-mail: bezzazi@lifl.fr

September 27, 1998

Abstract

We propose an inference algorithm based on forward chaining for handling inconsistencies that may occur in a program containing general and exceptional rules. We called this inference process stratified forward chaining because it is based on an ordered stratification of the set of literals appearing in the knowledge base. Given such a stratification, elementary forward chaining steps are achieved along its strata starting from the first one which is taken to be the input set of literals (the basic facts). The literals generated this way are combined in such a way that the specific information is priviliged over less specific one when they conflict each other. Actually this kind of inference generalizes the skeptical inference in unambigious inheritance systems for which we prove it is consistent. We consider in this context some of the properties relative to preferential logic and change operators where the consequence operation is a nonmonotonic operation as it is the case with Stratified Forward Chaining.

1 Introduction

In this work we investigate some of the properties of nonmonotonic logic which were established using classical logic as being the object language in a less rich framework but yet enough expressive to consider some interesting situations. Pieces of knowledge are represented as literals and rules. The inference engine we propose to represent the underlying logic of the agent is based on an ordered stratification of the set of literals appearing in the knowledge base which consists of a finite set of rules. Given such a stratification, elementary forward chaining steps are achieved along its strata starting from the first one which is taken to be the input set of literals (the basic facts). The literals generated this way are combined in such a way that the specific information is privileged over less specific one when they conflict each other. Forward chaining allows for efficient computations as it is the case with inheritance systems which are known to be efficient nonmonotonic systems. An eminent property of this stratified forward chaining is Consistency Preservation which makes the consistency of the consequences depend only on the consistency of the set of literals which the inference operation starts from.

This work was motivated by previous work on change operators and forward chaining [3] which in its turn was motivated by the concern to extend the set of conclusions obtained from a knowledge base and an exceptional information by extending the so called ranked revision. Ranked revision consists in computing the consequences of a set of literals by forward chaining using a subset of the set of rules for which it is consistent. This subset is computed in a precise way using forward chaining. However the conclusions it yields miss many others one would intuitively expect by using some kind of transitivity for instance [2]. It is precisely this kind of say cautious transitivity that we are attempting to catch by Stratified Forward Chaining.

The paper is organized as follows: In section 2 we give basic definitions and notations. In particular we recall the conditions usually required to be satisfied by a consequence operation [8]. These conditions are relativized to our framework and the Consistency Preservation condition is then introduced. In section 3 we define an operation called cautious extension which roughly speaking extends a consistent set of literals by those literals of another set that preserve consistency. This operation is used to define in section 4 an inference operation called Stratified Forward Chaining. We describe in this section how to construct the stratification upon which the inference is based. In section 5 we emphasize the nonmonotonic properties of Stratified Forward Chaining by relativizing to our framework the rules of preferential logic [7]. In section 6 we show that given any syntactical consequence operation satisfying our relativized version of Inclusion, Idempotence, Cautious Monotony and Consistency Preservation a natural revision operator is defined which satisfy our relativized version of AGM postulates for revision operators. This revision operator makes a fundamental use of cautious extension. It is also shown that iterated revision is rather well supported. In section 6 we do the same as in section 7 by considering update operators instead of revision operators with respect to the Katsuno and Mendelzon postulates [6]. In section 8 we show the consistency and the completeness of stratified forward chaining for the well known inference in inheritance systems based on preemption when these systems are unambiguous [5]. Finally in our conclusion we outline some of the issues that have drawn our attention while writing this paper.

2 Preliminaries

Let At be the set of atoms. A literal is an atom or a negation of an atom. The set of literals is $Lit = At \cup \neg At$. However we may use a succession of $\neg$ before an atom which is to be reduced in the usual way by erasing the occurrences of two successive $\neg$.

A rule is a sentence of the shape $l_1, l_2, \ldots, l_n \rightarrow l_{n+1}$ where l_i is a literal for $i = 1, \ldots, n+1$.

A knowledge base is a finite set rules.

A set of literals is said to be consistent if it does not contain two opposite literals, *i.e.* an atom and its negation.

The consistent part of a set of literals L is the set $\tilde{L}$ which is obtained by removing from L all opposite literals.

Let R and L be a finite set of rules and a finite set of literals respectively.

A rule $r \in R$ is L-fireable iff its body is in L.

We define the set of consequences of L by one step forward chaining with respect to R as:

$fc(R, L) = L \cup \{l \in Lit : l_1, l_2, \ldots, l_n \rightarrow l$ is an L-fireable rule of R $\}$.

Let $\{L_i\}_{i \in \omega}$ be the sequence defined by:

(i) $L_0 = L$

(ii) $L_{i+1} = fc(R, L_i)$

The closure of L by forward chaining w.r.t. R is defined by:

$$C_{fc}(R, L) = \cup_{i \in \omega} L_i$$

Notice that it is easily established that if $L' \subseteq C_{fc}(R, L)$ then $C_{fc}(R, L \cup L') = C_{fc}(R, L)$

In the AGM classical framework, the beliefs of the agent are represented in a propositional language closed under the usual connectives $\neg$, $\wedge$, $\vee$ and $\rightarrow$ and any consequence operation used to represent the underlying logic of the agent is required to satisfy some conditions called Inclusion, Idempotence, Monotony, Supraclassicality, Deduction and Compactness [8]. However when we take instead of the propositional language a language which is closed only under conjunction starting from literals and when we consider a syntactical consequence operation $\mathcal{C}$, Supraclassicality turns out to be equivalent to Inclusion and Deduction doesn't make sense anymore because of the absence of implication as a connective in our language. Therefore in our framework the conditions above reduce to the four conditions:

Inclusion:	$L \subseteq \mathcal{C}(L)$
Idempotence:	$\mathcal{C}(\mathcal{C}(L)) = \mathcal{C}(L)$
monotony:	$\mathcal{C}(L) \subseteq \mathcal{C}(L')$ whenever $L \subseteq L'$
Compactness:	If $l \in \mathcal{C}(L)$ then $l \in \mathcal{C}(L')$ for some finite $L' \subseteq L$.

As a matter of fact forward chaining satisfy the four of these conditions[1] and was investigated as a consequence operation underlying the definition of syntactical revision operators in [3]. The main idea in this paper is to investigate consequence operations which satisfy these conditions except eventually the monotony condition (so it is a nonmonotonic consequence operation). Instead we will require Cautious Monotony [8] and an additional condition called Consistency Preservation:

Cautious Monotony: If $L \subseteq \mathcal{C}(L')$ then $\mathcal{C}(L') \subseteq \mathcal{C}(L \cup L')$

Consistency Preservation: If L is consistent then $\mathcal{C}(L)$ is consistent.

Notice that when $\mathcal{C}$ satisfies Inclusion then Consistency Preservation is equivalent to saying that L is consistent iff $\mathcal{C}(L)$ is consistent. In the sequel we will refer to a consequence operation that satisfies Inclusion, Idempotence, Cautious Monotony and Consistency preservation as an **admissible** consequence operation. It is easily established that if $\mathcal{C}$ satisfies Inclusion, Idempotence and Cautious Monotony then it also satisfies the Cut property and hence the property of cumulativity:

Cut: If $L \subseteq \mathcal{C}(L')$ then $\mathcal{C}(L \cup L') \subseteq \mathcal{C}(L')$

Cumulativity: If $L \subseteq \mathcal{C}(L')$ then $\mathcal{C}(L') = \mathcal{C}(L \cup L')$

[1] Compactness is true for any syntactic consequence when the set of rules is finite as long as propositional calculus is concerned which will be always the case in this paper.

3 Cautious extension

We will use in the sequel of this paper an operation on sets of literals called cautious extension. This is a simple and yet a fundamental operation for the stratified forward chaining as well as for the revision operation we shall define. Cautious extension denoted by $\oplus$ is the non commutative binary operation on sets of literals defined by:

$$L \oplus L' = L \cup (L' - \{l \in L' : \neg l \in L\})$$

So $L \oplus L'$ is the extension of L by literals of L' that are not opposite to literals in L. It can also be written as:

$$L \oplus L' = L \cup (\widetilde{L \cup L'})$$

Some of the properties of this operation are listed in the following lemma. Their proof is quite straightforward.

Lemma 1 (Properties of $\oplus$) *The following properties hold for any sets of literals L, L', L'':*
(i) $L \subseteq L \oplus L'$
(ii) $L \oplus L' \subseteq L \cup L'$
(iii) $\oplus$ *is associative:* $(L \oplus L') \oplus L'' = L \oplus (L' \oplus L'')$
(iv) If $L' \subseteq L \oplus L''$ *then* $L' \subseteq L \oplus (L' \oplus L'')$
(v) If $L' \cup L''$ *is consistent then* $(L \oplus L') \cup L'' = (L \cup L'') \oplus L'$
(vi) If $(L \oplus L') \cup L''$ *is consistent then* $(L \oplus L') \cup L'' = (L \cup L'') \oplus L'$

Proof.

(i) and (ii) are immediate by definition of $L \oplus L'$

(iii) We show only one inclusion namely $(L \oplus L') \oplus L'' \subset L \oplus (L' \oplus L'')$, the other inclusion has a similar proof.

let $l \in (L \oplus L') \oplus L''$.

If $l \in L$ then $l \in L \oplus (L' \oplus L'')$ else necessarily $\neg l \notin L$ and if $l \in L'$ then $l \in L' \oplus L''$ and so $l \in L \oplus (L' \oplus L'')$. If $l \in L''$ and $l \notin L \cup L'$ then $\neg l \notin L'$ otherwise $\neg l \in L \oplus L'$ and then $l \notin (L \oplus L') \oplus L''$. But then $l \in L' \oplus L''$ and since $\neg l \notin L$, $l \in L \oplus (L' \oplus L'')$

(iv) let $L' \subset L \oplus L''$ and $l \in L'$. If $l \in L$ then $l \in L \oplus (L' \oplus L'')$. If $l \notin L$ then necessarily $l \in L''$ and $\neg l \notin L$. Now since $l \in L' \oplus L''$ and $\neg l \notin L$, $l \in L \oplus (L' \oplus L'')$. So $L' \subset L \oplus (L' \oplus L'')$. Conversely, let L' be a set of literals such that $L' \subset L \oplus (L' \oplus L'')$ and $l \in L'$. If $l \in L$ then $l \in L \oplus L''$ else necessarily $l \in L' \oplus L''$

(v) This is easily seen.

(vi) $(L \oplus L') \cup L'' = L \cup L' - \{l \in L' : \neg l \in L\} \cup L'' = (L \cup L'') \cup L' - \{l \in L' : \neg l \in L\}$. Since $(L \oplus L') \cup L''$ is consistent by hypothesis the set $\{l \in L : \neg l \in L''\}$ is empty and we can write $(L \oplus L') \cup L'' = (L \cup L'') \cup L' - \{l \in L' : \neg l \in L\} - \{l \in L' : \neg l \in L''\}$ which can be written again $(L \oplus L') \cup L'' = (L \cup L'') \cup L' - \{l \in L' : \neg l \in (L \cup L'')\} = (L \cup L'') \oplus L'$ ■

4 Stratified forward chaining

Definition 2 (stratification) *Let R be a non-empty set of rules. Let $Lit(R)$ be the set of literals that appear in R. We say that the sequence $\langle S_1, \ldots, S_n \rangle$ is a stratification for R iff it is a partition of $Lit(R)$ such that if $l_1, l_2, \ldots, l_p \rightarrow l$ is a rule of R then: $\forall k \in [1..p] : stratum(l_k) < stratum(l)$ where $stratum(l) = i$ iff $l \in S_i$. R is said to be stratified iff its set of stratifications is not empty. A stratification is said to be based on a set of literals L iff it is a stratification for R/L which is the subset of R that consists of all $C_{fc}(R, L)$-fireable rules of R[2].*

Definition 3 (ordering) *We now define the decreasing lexicographical ordering $\prec$ on the set of stratifications augmented by the empty sequence ε of a stratified set of rules.*
(i) $\varepsilon \prec \Sigma$ for any stratification Σ,
(ii) if $\Sigma = \langle S_1, \ldots, S_n \rangle$ and $\Sigma' = \langle S'_1, \ldots, S'_p \rangle$ are two non empty stratifications then $\Sigma \prec \Sigma'$ iff $S'_1 \subseteq S_1$ and $S_1 \neq S'_1$ or $S_1 = S'_1$ and $\langle S_2, \ldots, S_n \rangle \prec \langle S'_2, \ldots, S'_p \rangle$

Proposition 4 *Let R be a stratified set of rules. The ordering $\prec$ over its stratifications has a least element.*

Proof. We are going to give an algorithm that computes a stratification which we shall prove is the least stratification w.r.t. $\prec$. ■

Definition 5 (best stratification) *The best stratification of a stratified set of rules R is the least stratification of R according to $\prec$.*

We will denote the best stratification of a stratified set of rules R by $\Sigma(R) = \langle S_1(R), \ldots, S_n(R) \rangle$

[2]It is a simple fact that if R is stratified then so is R/L

4.1 Computing the best stratification of a stratified set of rules

We give in this subsection an algorithm that assigns to the literals appearing in R the rank of their stratum. The definition of the function *stratum* is extended to sets of literals as $stratum(L) = \{stratum(l) : l \in L\}$. $head(\Delta)$ is the set of literals that appear in the head of a rule of Δ.

Algorithm:
Input: A non empty set of rules R={$L_i \rightarrow l_i$:1≤i≤n}
Output: the stratification of R if it is stratified

```
let Δ=R
for all l∈Lit(Δ) let stratum(l)=0
let h=1
repeat
for all l∈Lit(Δ) if l∉head(Δ) then set stratum(l)=h
Let Δ_h={L→l∈ Δ:  max(stratum(L))=h and min(stratum(L))≠0}
set Δ = Δ − Δ_h and set h=h+1
until Δ_h is empty
if Δ is empty then
%make the last stratum%
for all l∈Lit(R) if stratum(l)=0 then set stratum(l)=h
%define the stratification%
for all j∈[1..h] set S_j={l∈Lit(R):stratum(l)=j}
else R is not stratified.
```

Proposition 6 *The algorithm above computes the least stratification of R w.r.t. $\prec$.*

Proof. (sketch of proof) The repeat loop terminates because at each step of the loop Δ_h is set to a subset of Δ which cardinal is strictly decreased if Δ_h is not empty by setting Δ to $\Delta - \Delta_h$. If after the repeat loop Δ is not empty then it contains precisely those rules that prevent the stratification by forming cycles. If it is empty then it is easy to see that the sequence $\langle S_1, \ldots, S_h\rangle$ computed by the algorithm is a partition of $Lit(R)$ since each literal of $Lit(R)$ has been assigned a unique stratum by the algorithm. It is also easy to see that it is a stratification since given a rule $l_1, l_2, \ldots, l_p \rightarrow l$ of R all of the literals of its body are assigned a stratum in the algorithm before l. Indeed, for the current set of rules Δ, this assignment is done for the literals that do not appear as a head of a rule. Now we show that it is the best stratification. Let $\langle S'_1, \ldots, S'_n\rangle$ be some stratification of R. It is easy to see that $S'_1 \subset S_1$. If $S_1 \neq S'_1$ we are done, otherwise let $k > 1$ be the least integer s.t. $S_k \neq S'_k$

and let l be a literal in S'_k. We denote by $head^{-1}(l)$ the set of rules r of R s.t. $head(r) = l$. Since $\langle S'_1, \ldots, S'_n \rangle$ is a stratification $body(head^{-1}(l)) \subset \cup_{1 \leq i \leq j} S'_i$ for some $j < k$. So $body(head^{-1}(l)) \subset \cup_{1 \leq i \leq j} S_i$ by hypothesis. Now the algorithm runs precisely s.t. $j = k-1$ since $l \notin \cup_{1 \leq i \leq j} S_i$. So $l \in S_{j+1} = S_k$. Thus $S'_k \subset S_k$ and $\langle S_1, \ldots, S_n \rangle \prec \langle S'_1, \ldots, S'_p \rangle$ ■

Remark 1 *In the sequel we shall abbreviate for convenience* $fc(R, L), C_{fc}(R, L)$, *and* $S_i(R/L)$, *to respectively* $fc(L), C_{fc}(L)$, *and* $S_i(L)$ *as long as no ambiguity is encountered. We will also use the following notations:* $\mathbf{S}_{<i+1}(L) = \mathbf{S}_{\leq i}(L) = \cup_{1 \leq j \leq i} S_j(L)$, *and* $\mathbf{S}_{>i}(L) = \mathbf{S}_{\geq i+1}(L) = \cup_{i < j \leq n} S_j(L)$.

Definition 7 (Stratified forward chaining) *Let* R *and* L *be respectively a set of rules and a set of literals. We define the stratified forward chaining of* L *with* R, *denoted* $C_s(L)$ *in two steps.*
First we define the sequence $\bar{\Sigma}(L) = \langle \bar{S}_1(L), \ldots, \bar{S}_n(L) \rangle$ *by:*
$\bar{S}_1(L) = L$
$\bar{S}_{i+1}(L) = \widetilde{fc}(\bar{\mathbf{S}}_{\leq i}(L) \cap \mathbf{S}_{\leq i}(L))$ *for* $i = 1, ..., n-1$ *where* $\bar{\mathbf{S}}_{\leq i}(L) = \oplus_{1 \leq j \leq i} \bar{S}_j(L)$
Then we put:

$$C_s(L) = \bar{\mathbf{S}}_{\leq n}(L) = \oplus_{1 \leq i \leq n} \bar{S}_i(L)$$

Remark 2 *It is easy to see that given a stratified set of rules* R *and a set of literals* L, *by definition Stratified Forward Chaining satisfies the Inclusion and the Consistency Preservation condition and therefore the only case where* $C_s(L)$ *is inconsistent is when* L *is inconsistent. Notice also that if* $l \in C_s(L)$ *and* $stratum(l) = p$ *then* $l \in \oplus_{1 \leq i \leq p} \bar{S}_i(L)$.

Lemma 8 *If* $L' \subseteq Lit(R/L)$ *then* $R/(L \cup L') = R/L$

Proof. $R/(L \cup L')$ consists of all $C_{fc}(L \cup L')$-fireable rules of R. Since $L' \subseteq Lit(R/L)$ and $Lit(R/L) = C_{fc}(R, L)$ we have $L' \subseteq C_{fc}(L)$. So $C_{fc}(L \cup L') = C_{fc}(L)$ and then $R/L \cup L' = R/L$ ■

Remark 3 *Restraining the stratification to* R/L *when computing* $C_{fc}(L)$ *instead of the stratification of* R *is a technical point. If we just use the stratification of* R, *this computation may be complicated by literals which are irrelevant for the information in* L *and which alter the stratification based on* L. *Consider for example* $R = \{a \rightarrow b, a' \rightarrow b, a \rightarrow \neg b, c \rightarrow c', c' \rightarrow \neg b\}$ *and* $L = \{a, a'\}$. *The stratification for* R *yields* $S_1 = \{a, a', c\}$, $S_2 = \{b, c'\}$ *and* $S_3 = \{\neg b\}$ *while the stratification based on* L *yields two strata,* $S_1 = \{a, a'\}$ *and* $S_2 = \{b, \neg b\}$.

Lemma 9 *If* $l \in C_s(L)$ *then* $\bar{S}_i(L \cup l) = \bar{S}_i(L)$ *for* $i = 2, ..., n$

Proof. Suppose $stratum(l) = k$. $\bar{S}_i(L \cup l) = \widetilde{fc}(\bar{\mathbf{S}}_{<i}(L \cup l) \cap \mathbf{S}_{<i}(L \cup l))$ for $i = 2, ..., n$. Since by the previous lemma $R/(L \cup l) = R/L$ we have: $\bar{S}_i(L \cup l) = \widetilde{fc}(\bar{\mathbf{S}}_{<i}(L \cup l) \cap \mathbf{S}_{<i}(L))$ for $i = 2, ..., n$

case $i = 2$: By definition we have $\bar{S}_1(L \cup l) = L \cup l$, so $\bar{S}_2(L \cup l) = \widetilde{fc}(\bar{\mathbf{S}}_{\leq 1}(L \cup l) \cap \mathbf{S}_{\leq 1}(L)) = \widetilde{fc}(L \cup l \cap S_1(L))$. If $k \geq 2$ then $\widetilde{fc}(L \cup l \cap S_1(L)) = \widetilde{fc}(\bar{S}_1(L) \cap S_1(L)) = \bar{S}_2(L)$.

Suppose the equality $\bar{S}_i(L \cup l) = \bar{S}_i(L)$ true for $2 \leq i \leq p$ with $p < n$ and let us prove it is also true for $p+1$.We have $\bar{S}_{p+1}(L \cup l) = \widetilde{fc}(\bar{\mathbf{S}}_{\leq p}(L \cup l) \cap \mathbf{S}_{\leq p}(L \cup l))$ and by induction hypothesis:
$\bar{S}_{p+1}(L \cup l) = \widetilde{fc}((\oplus_{1 \leq j \leq p} \bar{S}_j(L) \cup l) \cap \mathbf{S}_{\leq p}(L))$. If $k > p$ then $\bar{S}_{p+1}(L \cup l) = \widetilde{fc}(\oplus_{1 \leq j \leq p} \bar{S}_j(L) \cap \mathbf{S}_{\leq p}(L))$. If $k \leq p$ then $l \in \oplus_{1 \leq j \leq p} \bar{S}_j(L)$ since $l \in \oplus_{1 \leq i \leq k} \bar{S}_i(L)$ by the remark above and $\oplus_{1 \leq i \leq k} \bar{S}_i(L) \subset \oplus_{1 \leq j \leq p} \bar{S}_j(L)$. So in both cases we have $\bar{S}_{p+1}(L \cup l) = \bar{S}_{p+1}(L)$ ■

Proposition 10 *If $L' \subseteq C_s(L)$ then $\bar{S}_i(L \cup L') = \bar{S}_i(L)$ for $i = 2, ..., n$*

Proof. The property is trivially true when $L' = \emptyset$.

Suppose the property true for any set $L' \subseteq C_s(L)$ of cardinality k and let us prove that it is also true for $L' \cup l$ when $l \in C_s(L) = C_s(L \cup L')$. We have $\bar{S}_i(L \cup L' \cup l) = \bar{S}_i(L \cup L')$ for $i = 2, ..., n$ by the previous lemma and then $\bar{S}_i(L \cup L' \cup l) = \bar{S}_i(L)$ for $i = 2, ..., n$ by hypothesis ■

5 Connection with nonmonotonic logics

In this section we check some of the properties of nonmonotonic logic as stated by preferential logics [7]. We are interested in literals one can infer from the set of rules R according to the inference relation defined by $L \mid\!\sim_s L'$ iff $L' \subseteq C_s(L)$.

Definition 11 *The properties called Reflexivity, And, Cautious Monotony and Cut for an inference relation $\mid\!\sim$ are given respectively by the following rules:*

$$\textbf{REF} \quad \frac{}{L \mid\!\sim L}$$

$$\textbf{AND} \quad \frac{L \mid\!\sim L' \quad L \mid\!\sim L''}{L \mid\!\sim L' \cup L''}$$

$$\textbf{CM} \quad \frac{L \mid\!\sim L' \quad L \mid\!\sim L''}{L \cup L'' \mid\!\sim L'}$$

$$\textbf{CUT} \quad \frac{L \cup L' \mid\!\sim L'' \quad L \mid\!\sim L'}{L \mid\!\sim L''}$$

Proposition 12 *The inference relation* $\mid\!\sim_s$*satisfies* **REF**, **AND**, **CM** *and* **CUT**.

Proof.

REF and AND are easily established.

CM: Suppose $L'' \subseteq C_s(L)$ and, without loss of generality, that $L'' \cap L = \emptyset$. By definition, $C_s(L \cup L'') = \oplus_{1 \leq i \leq n} \bar{S}_i(L \cup L'')$. Then by proposition 10 $C_s(L \cup L'') = (L \cup L'') \oplus (\oplus_{2 \leq i \leq n} \bar{S}_i(L))$. Now by 1 *(v)* we have $C_s(L \cup L'') = (L \oplus (\oplus_{2 \leq i \leq n} \bar{S}_i(L))) \cup L''$ since $L'' \subseteq \oplus_{2 \leq i \leq n} \bar{S}_i(L)$ given that $\oplus_{2 \leq i \leq n} \bar{S}_i(L)$ is consistent by construction. So $C_s(L \cup L'') = C_s(L) \cup L'' = C_s(L)$. Thus if $L' \subseteq C_s(L)$ and $L'' \subseteq C_s(L)$ then $L' \subseteq C_s(L \cup L'') = C_s(L)$.

CUT: Suppose that $L' \subseteq C_s(L \cup L'')$ and $L'' \subseteq C_s(L)$. We have just seen in CM above that in this case $C_s(L \cup L'') = C_s(L)$. So $L' \subseteq C_s(L)$ ■

The following corollary shows that the stratified forward chaining satisfies the Idempotence property and therefore it is an admissible consequence operation.

Corollary 13 $C_s(L) = C_s(C_s(L))$

Proof. In our proof of **CM** we have stated that if $L'' \subseteq C_s(L)$ then $C_s(L \cup L'') = C_s(L)$. So $C_s(L \cup C_s(L)) = C_s(L)$ because $C_s(L) \subseteq C_s(L)$ but we have also $C_s(L \cup C_s(L)) = C_s(C_s(L))$ because $L \subseteq C_s(L)$ ■

6 Connection with revision theory

Instead of taking the propositional logic for representing the beliefs of the agent and instead of considering the classical consequence to be the logic of the agent as in AGM framework, we investigate an approach to belief revision where beliefs are expressed as literals and rules (pure Horn clauses) and the underlying logic is some admissible consequence operation $\mathcal{C}$ (*i.e.* satisfying the four properties of the introduction Inclusion, Idempotence, Cautious Monotony and Consistency preservation) *e.g.* stratified forward chaining.

We define the absolute revision of a set of literals L by another set of literals L' as follows

$$L \circ L' = L' \oplus \tilde{L}$$

Actually the revision operation should apply to a set of literals closed by the underlying consequence relation. In our setting we should focus on the definition of the operation $\mathcal{C}(L) * L'$. In the definition we are to give for this operator the set L in $\mathcal{C}(L)$ is distinguished from the rest of the literals of $\mathcal{C}(L)$ as being its generator, a similar differenciation is done in [9]. The operator $*$ is defined by

$$\mathcal{C}(L) * L' = \mathcal{C}(L \circ L')$$

Similarly, we define the expansion operation of the knowledge base $\mathcal{C}(L)$ by the set of literals L as:

$$\mathcal{C}(L) + L' = \mathcal{C}(L \cup L')$$

The postulates we are about to give now ensure in particular that the revision of $\mathcal{C}(L)$ by L' coincides with its expansion by the same set when they are consistent to each other. These postulates constitute a relativized version to our framework of the classical revision postulates. We call them syntactical revision postulates. $\bullet$ is an operation that takes as argument a couple of sets of literals. the first set is closed under the consequence operation of interest and represents the belief set to be revised by the second set. The result of this operation is a new belief set.

(SR1) $L' \subseteq \mathcal{C}(L) \bullet L'$

(SR2) If $\mathcal{C}(L) \cup L'$ is consistent, then $\mathcal{C}(L) \bullet L' = \mathcal{C}(L) + L'$

(SR3) If L' is consistent then $\mathcal{C}(L) \bullet L'$ is also consistent

(SR4) If $(\mathcal{C}(L) \bullet L') \cup L''$ is consistent then $\mathcal{C}(L) \bullet (L' \cup L'') = (\mathcal{C}(L) \bullet L') + L''$

Proposition 14 *The operation $*$ satisfies the syntactical revision postulates.*

Proof. SR1: $L' \subseteq L' \oplus \tilde{L} \subseteq \mathcal{C}(L' \oplus \tilde{L}) = \mathcal{C}(L) * L'$ by the Inclusion property.

SR2: If $\mathcal{C}(L) \cup L'$ is consistent, then $L \circ L' = L' \cup L$ and $\mathcal{C}(L) + L' = \mathcal{C}(L \cup L')$. So $\mathcal{C}(L) + L' = \mathcal{C}(L \circ L') = \mathcal{C}(L) * L'$.

SR3: $\mathcal{C}(L) * L' = \mathcal{C}(L' \oplus \tilde{L})$ is consistent iff $L' \oplus \tilde{L}$ is consistent by the Consistency Preservation property and this is true iff L' is consistent by definition of cautious extension.

SR4: Suppose $(\mathcal{C}(L) * L') \cup L'' = \mathcal{C}(L \circ L') \cup L''$ is consistent. Since $(L' \oplus \tilde{L}) \cup L''$ is consistent by the Consistency Preservation property we have $(L \circ L') \cup L'' = (L' \oplus \tilde{L}) \cup L'' = (L' \cup L'') \oplus \tilde{L} = L \circ (L' \cup L'')$ by virtue of lemma 1 *(vi)*. So $\mathcal{C}(L) * (L' \cup L'') = \mathcal{C}(L \circ (L' \cup L'')) = \mathcal{C}((L \circ L') \cup L'')$ and then $\mathcal{C}(L) * (L' \cup L'') = \mathcal{C}(L \circ L') + L'' = (\mathcal{C}(L) * L') + L''$ ■

6.1 Iterated revision

The four relativized postulates proposed by Darwiche and Pearl [4] to account for the iterated belief change are given below for a revision operation $\bullet$ defined as above.

(SR5) If $L' \subseteq L''$ then $(\mathcal{C}(L) \bullet L') \bullet L'' = \mathcal{C}(L) \bullet L''$
(SR6) If $L' \cup L''$ is inconsistent, then $(\mathcal{C}(L) \bullet L') \bullet L'' = \mathcal{C}(L) \bullet L''$
(SR7) If $L' \subseteq \mathcal{C}(L) \bullet L''$ then $L' \subseteq (\mathcal{C}(L) \bullet L') \bullet L''$
(SR8) If $\neg l \notin \mathcal{C}(L) \bullet L''$ and $l \in L'$ then $\neg l \notin (\mathcal{C}(L) \bullet L') \bullet L''$

Proposition 15 *Supposing that all of L, L' and L'' are consistent the revision operation $*$ satisfies three of the four relativized postulates above and fails to satisfy **(SR6)**.*

Proof. SR5: Indeed $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L'' \oplus \widetilde{(L' \oplus \tilde{L})}) = \mathcal{C}(L'' \oplus (L' \oplus L)) = \mathcal{C}((L'' \oplus L') \oplus L)$ and so $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L'' \oplus L) = \mathcal{C}(L) * L''$.

SR6: This rule is not satisfied. Take L, L' and L'' to be respectively $\{t\}, \{b, s\}, \{\neg s\}$ for example and consider $R = \emptyset$. Clearly $L' \cup L''$ is inconsistent and $(\mathcal{C}(L) * L') * L'' = \{t, b, \neg s\}$ and $\mathcal{C}(L) * L'' = \{t, \neg s\}$. This rule is actually counterintuitive for this example if we think of t as tweety, b as bird and s as singing in that there is no reason to reject b.

SR7: Suppose $L' \subseteq \mathcal{C}(L) * L''$, then $L' \cup L''$ is consistent by Consistency Preservation. $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L \circ (L' \circ L'')) = \mathcal{C}((L'' \oplus L') \oplus L) = \mathcal{C}((L'' \cup L') \oplus L)$. So by the lemma 1 *(v)* $(\mathcal{C}(L) * L') * L'' = \mathcal{C}((L'' \oplus L) \cup L')$ and then $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L'' \oplus L)$ by Cumulativity since $L' \subseteq \mathcal{C}(L) * L'' = \mathcal{C}(L'' \oplus L)$ by hypothesis.

SR8: Suppose $\neg l \in (\mathcal{C}(L) * L') * L'' = \mathcal{C}(L'' \oplus (L' \oplus L))$ and $l \in L'$. Then by Consistency Preservation $\neg l \in L''$. So $\neg l \in L'' \oplus L$ and then $\neg l \in \mathcal{C}(L'' \oplus \tilde{L}) = \mathcal{C}(L) * L''$.

Remark 4 *A property satisfied by $*$ and which has not been proposed as a postulate for iterated revision which generally deals with different operators is relative associativity:*

(SR9) $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L) * (L' \circ L'')$

Indeed, $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L'' \oplus (L' \oplus L)) = \mathcal{C}((L'' \oplus L') \oplus L)$ *and so* $(\mathcal{C}(L) * L') * L'' = \mathcal{C}(L) * (L'' \oplus L') = \mathcal{C}(L) * (L' \circ L'')$ ■

7 Connection with update theory

We consider in this section our relativized version of the postulates proposed by Katsuno and Mendelzon [6] to characterize the operation of updating a knowledge base. Here again as with the revision operation and for the same reasons we have dropped the postulate concerning the syntax independence. We have also dropped from the original setting two postulates (the two last ones) which deal with disjunctive formulas since in our setting formulas are only conjunction of literals. Consider an operation $\diamond$ that takes as argument a couple of sets of literals. The first set represents the data base to be updated by the second data set. The result of this operation is a new data base.$\diamond$ is said to be a syntactical update operator iff it satisfies the following postulates:

(SU1) $L' \subset \mathcal{C}(L \diamond L')$

(SU2) If $L' \subset \mathcal{C}(L)$ then $\mathcal{C}(L \diamond L') = \mathcal{C}(L)$

(SU3) If both L and L' are consistent then $L \diamond L'$ is also consistent.

(SU4) If $(L \diamond L') \cup L''$ is consistent then $\mathcal{C}(L \diamond (L' \cup L'')) \subset \mathcal{C}((L \diamond L') \cup L'')$

(SU4') If $(L \diamond L') \cup L''$ is inconsistent then so is $\mathcal{C}((L \diamond L') \cup L'')$

(SU5) Suppose both of $L \diamond L'$ and $L \diamond L''$ are consistent then if $L'' \subset \mathcal{C}(L \diamond L')$ and $L' \subset \mathcal{C}(L \diamond L'')$ then $\mathcal{C}(L \diamond L') = \mathcal{C}(L \diamond L'')$

We now check that the absolute revision operator $\circ$ defined above satisfies these postulates when $\mathcal{C}$ is a syntactical consequence operation that satisfies the four properties above.

Proposition 16 *The operator $\circ$ defined by $L \circ L' = L' \oplus \tilde{L}$ is a syntactical update operator*

Proof.

SU1: $L' \subset \mathcal{C}(L \circ L') = \mathcal{C}(L) * L'$ is an immediate consequence of (R1) above.

SU2: Suppose $L' \subset \mathcal{C}(L)$. It follows that $L \circ L' = L \cup L'$ and then $\mathcal{C}(L \circ L') = \mathcal{C}(L \cup L') = \mathcal{C}(L)$ since $\mathcal{C}$ is cumulative.

SU3: This is trivial since $L \circ L' = L' \oplus \tilde{L}$ is consistent iff L' is consistent.

SU4: This is an immediate consequence of (SR4).

SU4': This is immediate from the property of consistency preservation enjoyed by $\mathcal{C}$.

SU5: If $L'' \subset \mathcal{C}(L \circ L')$ and $L' \subset \mathcal{C}(L \circ L'')$ then $\mathcal{C}(L \circ L') = \mathcal{C}((L \circ L') \cup L'')$ and $\mathcal{C}(L \circ L'') = \mathcal{C}((L \circ L'') \cup L')$ by the property of cumulativity enjoyed by $\mathcal{C}$. Now if $L \circ L'$ and $L \circ L''$ are supposed consistent then all of $(L \circ L') \cup L''$, $(L \circ L'') \cup L'$ and $L' \cup L''$ are consistent and hence $(L \circ L') \cup L'' = (L \circ L') \circ L'' = L \circ (L' \circ L'') = L \circ (L' \cup L'')$ and similarly $(L \circ L'') \cup L' = (L \circ L'') \circ L' = L \circ (L'' \circ L') = L \circ (L'' \cup L')$ and so $\mathcal{C}(L \circ L') = \mathcal{C}(L \circ (L' \cup L'')) = \mathcal{C}(L \circ L'')$ ■

8 Connection with inference in inheritance systems

Definition 17 *A set of rules Γ is an inheritance system (or inheritance network) iff all of its rules (or links) are of the shape $l \rightarrow l'$ where l is a positive literal and for any positive literal $l \in lit(\Gamma), \neg l \notin \widetilde{fc}(l)$. A path over Γ is a finite sequence $\{l_i \rightarrow l_{i+1}\}_{1 \leq i \leq n}$ henceforth denoted as $l_1 \rightarrow l_2 \rightarrow \ldots \rightarrow l_{n+1} (n \geq 1)$ where all of the literals are positive except possibly the last. Two rules are said to be contradictory iff they have the same body and opposite heads. An inheritance network Γ is said to be non contradictory iff it does not contain contradictory rules.*

Definition 18 **1.** *If $l \rightarrow l' \in \Gamma$ then $\Gamma \models l \rightarrow l'$*

2. *Let $\sigma = l \rightarrow \sigma_1 \rightarrow l_1 \rightarrow l'$ be a path in Γ, then $\Gamma \models \sigma$ iff*

- **(i)** $\Gamma \models l \rightarrow \sigma_1 \rightarrow l_1$
- **(ii)** $l_1 \rightarrow l' \in \Gamma$
- **(iii)** $l \rightarrow \neg l' \notin \Gamma$
- **(iv)** *for any node $l'' \in lit(\Gamma)$ such that $\Gamma \models l \rightarrow \tau \rightarrow l''$ and $l'' \rightarrow \neg l' \in \Gamma$ there is a node l_0 such that $\Gamma \models l \rightarrow \tau_1 \rightarrow l_0 \rightarrow \tau_2 \rightarrow l''$ and $l_0 \rightarrow l'$.*

Let $\mid\!\sim_\Gamma$ be the inference relation defined on $lit(\Gamma)$ by $l \mid\!\sim_\Gamma l'$ iff $\Gamma \models l \rightarrow \sigma \rightarrow l'$ for some (possibly empty) path σ. Non contradictory inheritance networks enjoy the soundness property [5]:

Proposition 19 *If Γ is a non contradictory inheritance net then:*
$l \mid\!\sim_\Gamma l' \Rightarrow l \not\mid\!\sim_\Gamma \neg l'$

Definition 20 *The inheritance network* Γ *is said to be unambiguous iff it is non contradictory and:*
$(l \mid\!\sim_\Gamma l_0$ *and* $l_0 \to l'' \in \Gamma) \Rightarrow (\ l \not\mid\!\sim_\Gamma \neg l'' \Leftrightarrow l \mid\!\sim_\Gamma l''\ )$.

The following proposition states that stratified forward chaining is coherent with the inference relation defined by Horty et al. [5] when the underlying net is unambiguous.

Proposition 21 *Let* Γ *be a non ambiguous inheritance network and let* $\mid\!\sim_\Gamma$ *be the inference relation defined on* $lit(\Gamma)$ *by* $l \mid\!\sim_\Gamma l'$ *iff* $\Gamma \vDash l \to \sigma \to l'$ *for some (possibly empty) path* σ, *then* $l \mid\!\sim_\Gamma l'$ *iff* $l' \in C_s(l)$ *and* $l \neq l'$

Proof. l being fixed the proof is done by induction on $stratum(l')$. If $l \mid\!\sim_\Gamma l'$ or $l' \in C_{sfc}(l)$ then $stratum(l') \geq 2$. (base case) Suppose $stratum(l') = 2$ then necessarily $l \to l' \in \Gamma$ and so $l \mid\!\sim_\Gamma l'$ by definition of $\mid\!\sim_\Gamma$. On the other hand $l' \in \widetilde{fc}(l)$ since Γ is non contradictory. Now $C_{sfc}(l) = \cup_{1 \leq i \leq n} \bar{S}_i(l)$ with $\bar{S}_1(l) = \{l\}$ and $\bar{S}_2(l) = \widetilde{fc}(\bar{S}_1(l) \cap S_1(l)) = \widetilde{fc}(l)$ because $\bar{S}_1(l) = S_1(l) = \{l\}$. So $l' \in \bar{S}_2(l)$ and therefore $l' \in \bar{S}_1(l) \cup \bar{S}_2(l) \subset C_{sfc}(l)$.
(induction step) Suppose that the equivalence $l \mid\!\sim_\Gamma l'$ iff $l' \in C_{sfc}(l)$ and $l \neq l'$ holds for all l' such that $stratum(l') \leq p$ with $p \geq 2$. ($\Rightarrow$) Consider an l'' such that $l \mid\!\sim_\Gamma l''$ and $stratum(l'') = p+1$. We have $\Gamma \vDash l \to \sigma \to l''$ for some non empty path $\sigma = \sigma_1 \to l'$. Let then σ be such a path with l' s.t. $stratum(l')$ is minimal. This implies that $\Gamma \vDash l \to \sigma$ and $l' \to l'' \in \Gamma$. Since $stratum(l') \leq p$ and $l \mid\!\sim_\Gamma l'$ we have $l' \in C_{sfc}(l)$. Let k be the least integer such that $l' \in \bar{S}_k(l)$ (we have then $k \leq stratum(l')$). Then $l'' \in \widetilde{fc}(\bar{\mathbf{S}}_{\leq k}(l) \cap \mathbf{S}_{\leq k}(l))$ unless $l_0 \to \neg l'' \in \Gamma$ for some $l_0 \in \bar{\mathbf{S}}_{\leq k}(l) \cap \mathbf{S}_{\leq k}(l)$. If this were the case then $l_0 \in C_{sfc}(l)$ and $stratum(l_0) \leq k \leq p$. Then by induction hypothesis $l \mid\!\sim_\Gamma l_0$. Now by the definition above for inference in inheritance nets there exists a literal l_1 such that $\Gamma \vDash l \to \tau_1 \to l_1 \to \tau_2 \to l_0$ and $l_1 \to l''$ for some paths τ_1 and τ_2. Notice then that $l \mid\!\sim_\Gamma l_1$ and $l_1 \to l''$ with $stratum(l_1) < k \leq stratum(l')$ which contradicts the fact that $stratum(l')$ is minimal. Thus $l'' \in \widetilde{fc}(\bar{\mathbf{S}}_{\leq k}(l) \cap \mathbf{S}_{\leq k}(l)) = \bar{S}_{k+1}(l) \subset C_{sfc}(l)$.

($\Leftarrow$) Conversely suppose $l'' \in C_{sfc}(l)$ and $l \neq l''$ then there is a rule $l_0 \to l'' \in \Gamma$ with $l_0 \in \bar{\mathbf{S}}_{\leq k}(l) \cap \mathbf{S}_{\leq k}(l)$ and $stratum(l_0) < p+1$ minimal. So $l_0 \in C_{sfc}(l)$ and by induction hypothesis $l \mid\!\sim_\Gamma l_0$. Now since Γ is unambiguous $l \mid\!\sim_\Gamma l''$ unless $l \mid\!\sim_\Gamma \neg l''$. If this were the case then there is a node l_1 such that $\Gamma \vDash l \to \tau_1 \to l_1 \to \tau_2 \to l_0$ and $l_1 \to \neg l''$. We have then by induction hypothesis $l \mid\!\sim_\Gamma l_1$ since $stratum(l_1) < stratum(l_0) < p+1$ and $l_1 \in \bar{S}_{k_1}(l)$ with $k_1 \leq stratum(l_1)$. Now $\neg l'' \in \widetilde{fc}(\bar{\mathbf{S}}_{\leq k_1}(l) \cap \mathbf{S}_{\leq k_1}(l))$ since $k_1 < stratum(l_0)$ and then $\neg l'' \in \widetilde{fc}(\bar{\mathbf{S}}_{\leq k_1}(l) \cap \mathbf{S}_{\leq k_1}(l)) \subset C_{sfc}(l)$ since $k_1 \leq stratum(l_1) < stratum(\neg l'')$ but we have supposed $l'' \in C_{sfc}(l)$ so necessarily $l \mid\!\sim_\Gamma l''$ ∎

9 Conclusion

In this paper, we begun the investigation of a special kind of consequence operations by strengthening the classical definition of such an operation. These consequence operations enjoy the Consistency Preservation condition which reduces the requirements on the theory inferred to requirements on the generator of the theory. In particular we presented one such consequence operation named stratified forward chaining which is close to reasoning over inheritance nets. Nevertheless still much work has to be done. In this respect we list some of the issues for future work:

- Compare different stratified forward chaining with respect to the different stratifications they are based on.
- Investigate the combining of ranked revision and revision with stratified forward chaining.
- Define and investigate revision operators starting as we did in this paper from absolute revision but in the more general framework of propositional logic.
- Investigate the extension to the predicate case.
- Investigate the addition of priorities to the rules.

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, D. Makinson, and R. Pino Pérez. Beyond rational monotony: some strong non-horn rules for nonmonotonic inference relations. *Journal of Logic and Computation*, **6**, 605-631, 1997

[3] H. Bezzazi, S. Janot, S. Konieczny and R. Pino Pérez. Analysing rational properties of change operators based on forward chaining. To appear in *Transactions and change in logic databases.* LNCS, Springer, 1998.

[4] A. Darwiche and J. Pearl. On the logic of iterated belief revision. In Ronald Fagin, editor. *Proceedings of the fifth Conference on Theoretical Aspects of Reasoning about Knowledge.* 5-23. Morgan Kauffman, Pacific Grove, CA. March 1994.

[5] J. Horty, R. Thomason and D. Touretzky.A skeptical theory of inheritance in nonmonotonic networks.*Artificial Intelligence*, **42**,311–348, 1990.

[6] H. Katsuno and A.O. Mendelzon. On the difference between updating a knowledge database and revising it. In P. Gärdenfors, editor, *Belief Revision.* Cambridge, MA, 1992. Cambrigde tracts in theoretical computer science, # 29.

[7] S. Kraus, D. Lehmann and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence*, **44**,167–207, 1990.

[8] D. Makinson. Genearal patterns in nonmonotonic reasoning. *In Handbook of Logic in Artificial Intelligence and Logic Programming, Vol 3: Non Monotonic Reasoning and Uncertain Reasoning, D. Gabbay, C. G. Hogger and J. A. Robinson, eds. pp. 35-110. Clarendon Press, Oxford, 1994.*

[9] M-A Williams. Transmutations of Knowledge Systems. In J. Doyle, E. Sandewall, and P. Torasso (eds), *Principles of Knowledge Representation and Reasoning.* Cambridge, MA, 1992. Morgan Kaufmann Publishers, San Mateo, CA, 619-629, 1994.