Recovery of Nonmonotonic Theories

Cees Witteveen[*] Wiebe van der Hoek[†]
witt@cs.tudelft.nl wiebe@cs.ruu.nl

Abstract

We present a framework for recovery of nonmonotonic theories, i.e. of theories that are interpreted using a nonmonotonic semantics. Recovery of a nonmonotonic theory is needed if it does not have a model under the given nonmonotonic semantics, i.e. if the theory is non-monotonically inconsistent. With classical theories, inconsistency can only be removed by contracting the current theory; for nonmonotonic databases, however, it is in general unclear how to restore the consistency of a theory: indeed, several options for recovery that use (mixtures of) contractions and expansions have been proposed in the literature. In this paper, we propose a more fundamental approach to study the recovery problem by stating some rationality postulates for recovery. In these postulates we assume that, when recovering a theory T with respect to some intended semantics, one can fall back on a weaker, so called back-up semantics for T. Based on these rationality postulates, our general conclusion is that, in most cases, contraction is not adequate to handle nontrivial recovery problems in nonmonotonic theories.

1 Introduction

The idea about nonmonotonic theories is that they allow one to use common sense reasoning patterns to infer facts that *normally* can be expected, given a state of affairs represented by the theory. Usually, the use of such non-classical reasoning schemes enables one to draw more and stronger conclusions than a classical reasoning scheme would do. If, however, the world is not as normal as one expects it to be, such reasoning patterns significantly diverge from classical ones. For example, suppose a nonmonotonic reasoner Alice knows that (i) $Y(z)$ will be the case if $X(z)$ is not the case and (ii) normally $X(z)$ is not the case. Then Alice will expect both $not\, X(z)$ and $Y(z)$ to

[*]Delft University of Technology, Department of Mathematics and Computer Science, P.O.Box 356, 2600 AJ Delft, The Netherlands.

[†]Utrecht University, Department of Computer Science, Padualaan 14, 3584 CH Utrecht, The Netherlands,

hold. If, however, Alice observes for some a that $X(a)$ actually is true, she will not be surprised if $Y(a)$ is not the case, since she no longer had a reason to expect $Y(a)$ to hold. One might say that in this case the nonmonotonic machinery is able to deal with the unexpected in the sense that the new information $X(a)$ makes a shift in the reasoner's theory: from a theory T in which $not X(a)$ and $Y(a)$ hold, into a theory T' in which $X(a)$ holds and $Y(a)$ is undetermined. Here, the nonmonotonic mode of our reasoner is perfectly demonstrated— adding the fact $X(a)$ to the theory T makes the reasoner retract $not X(a)$ and, induced by that retraction, also every reason for $Y(a)$ is lost.

However, unexpected behaviour of the world may also cause the nonmonotonic reasoner to face a situation from which he cannot recover. This can happen, for example, if a conclusion based on nonmonotonic reasoning from expectations clashes with an observation that contradicts it without giving any direct clue for the reason why the rule used to infer this expectation could not be used. For example, suppose that Bob (i) uses a reasoning mechanism that $Y(z)$ will be the case if he has no reason to assume that $X(z)$ is the case and (ii) he has no direct reason to assume that $X(a)$ is true, but (iii) Bob actually observes that $Y(a)$ is not the case. In this case, using his nonmonotonic inference machine Bob might derive an inconsistency, since using (i) and (ii), he will derive that $Y(a)$ holds, contradicting his observation $Y(a)$.

Confronted with such difficulties, in principle we could choose between to possible ways to overcome them:

1. *change the underlying reasoning mechanism*
   The original nonmonotonic semantics is considered as defective and has to be replaced by a more sophisticated semantics that is also able to handle such (slight) abnormalities.

2. *change the theory*
   Instead of considering these properties as defects of a nonmonotonic semantics, one could also reason that such consequences have to be expected if the world apparently is not as normal as expected and therefore, instead of changing our semantics, we have to change our ideas about what actually is the case. Adapting to such slight abnormalities should be sufficient to make our nonmonotonic reasoning applicable again.

Semantically, using a nonmonotonic logic to reason about a theory T boils down to selecting a subset of intended models from some given collection of models for T, which we call the backup semantics for T. Those intended models are useful for making predictions about the world if it is as normal as we suppose it is. In slightly abnormal circumstances then, we should not be surprised to find that some conceivable, but non-intended models will better describe what we might expect. Should we blame our semantics for failing to provide the right models? We don't think so, since this semantics was intended to be used if the world was behaving normal. This explains why we adhere to the principle *change the theory and not the logic*.

If we are in a situation in which the intended set of models for T is empty, whereas the set of models from the backup semantics is not, we consider the latter set as a first approximation of the set of intended models— it assumes of *less* facts that they normally hold. Since the backup semantics is weaker, therefore allows for more abnormalities to occur and also is successful (i.e, *does* yield models), it may give us clues about which abnormalities we have to be prepared for. This information then, can be used to change the original theory T to a *recovered* theory T'. Since the abnormalities have now been accounted for in $'$, it seems natural to apply our original intended semantics to T' to derive our standard expectations *modulo* the abnormalities we discovered. Returning to Bob once again, he might fall back to a classical reasoner, interpreting (i) as (i'): $\neg X(z) \rightarrow Y(z)$ which, together with his observation yields the fact $X(a)$, providing Bob with a new theory T' consisting of (i), (ii), (iii), $X(a)$ and $\neg Y(a)$.

Let us now present a more formal example from the area of logic programming. The example not only provides a reason why changing the theory has advantages over changing the reasoning mechanism, it also demonstrates that the backup semantics need not be classical: it may be nonmonotonic as well. Suppose we have the following program:

$$
\begin{array}{rrcl}
P: & \neg a & \leftarrow & \text{not } b \\
 & c & \leftarrow & \text{not } d \\
 & a & \leftarrow &
\end{array}
$$

Let the stable model semantics be our intended semantics. For this program P, the stable model semantics is contradictory and does not succeed in resolving the conflict between a and $\neg a$. Using the weaker minimal model semantics as a backup semantics, we have two minimal models $I_1 = \{a, b, c\}$ and $I_2 = \{a, b, d\}$. Hence, using the backup semantics with a skeptical reasoning mode, we conclude b. But clearly, revising P by adding $b \leftarrow$ to it, results in a program P' having as its unique stable model the interpretation I_1.

This example shows how a stable reasoner may fall back on a weaker minimal model semantics to recover from a theory P that has no stable models. It also demonstrates the advantages of changing the theory P rather than the underlying reasoning mode. For suppose that our reasoner would decide to stick to the skeptical minimal model semantics, once having discovered that the intended semantics fails for the current theory P. Given the two minimal models I_1 and I_2 then, he would not be able to predict c anymore, whereas, for a stable reasoner, c is a desirable conclusion, since the conflict between a and $\neg a$ seems to be independent from the reasons to expect c to be true: the fact that the world is behaving abnormal in *some* respect is, for normal people, not a reason to assume that it is abnormal in *every* respect.

Summarizing, the backup semantics only serves to indicate the abnormalities that we have to account for. This information is used to translate the original theory T into a theory T' such that T' has intended models. The models of the backup semantics serve as a first approximation, or upperbound, for the recovered set of intended models of the theory T and will be used to select the intended models from. In this way, exceptions to the expected are really treated as exceptional, rather than forcing them to become predictable.

Position and overview of the paper

Comparing this recovery process with the recovery of classical theories, there are some resemblances but also some differences. A classical theory has to be recovered if the theory is not satisfiable. In the dominant AGM [3] framework for recovery of classical theories, also a transformational approach is applied. A recovery of such an inconsistent theory T always comes down to contracting T by deleting some statements from it in such a way that the contraction T' is consistent: the intended models of T are obtained from the classical models of T'. In recovery of non-monotonic theories, it is only the set of intended models of the theory T that is empty, while, viewed in a classical way, T still might have classical models. With respect to the recovery process this means that it is not clear at all which kind of recovered theory T' of T would be a most suitable: in some cases it might be that an expansion of T has intended models, in other cases it might be that a contraction of T has suitable intended models, while in still other cases a recovered theory which is obtained by adding some statements to T and deleting others, is a most suitable candidate for the recovered theory.

It comes as no surprise that in the literature different proposals for recovery of nonmonotonic theories have been offered. Many of these approaches, however, seem to work well for only a particular formalism, and for some ad hoc reasons. What the field lacks is a formulation of the ideas underlying the recovery process in a clear and unifying way. It is the purpose of this paper first of all to state some clear and very general *rationality postulates* such a recovery process has to satisfy. Essentially, these postulates describe which properties a suitable nonmonotonic recovery operation R should have independently from the specific properties of the nonmonotonic logic used. We introduce these postulates for recovery in Section 2, after we have identified some crucial properties of nonmonotonic consequence operators. Then, in Section 3, using those properties, we state some results pertaining to the *type* of the recovery operation that should be applied in order to satisfy the rationality postulates. Our general conclusion is that for mainstream nonmonotonic semantics, contrary to what one might expect on the basis of recovery of classical theories, theory-contraction is *not suitable* for recovery of nonmonotonic theories. Finally, in Section 4, we show how the framework can be applied successfully in the recovery of logic programs.

2 The framework

2.1 Preliminaries

Given a language L, a *theory* T is any subset of L. We assume to have a way to assign to each T some (possibly empty) set of models $Mod(T)$ in some specified class. For any class of theories $\mathcal{T}$ a *semantics* Sem then is a way to associate consequences φ to some $T \in \mathcal{T}$, based on $Mod(T)$. Such a semantics is called *well-behaved* w.r.t. T, if $Sem(T)$ is defined and is not equal to L.[1] We often identify a semantics Sem with a *consequence operation* $C^{Sem} : 2^L \to 2^L$, where, in this paper, we stipulate that $C^{Sem}(T) = L$ in case $Sem(T)$ is not defined. Generalizing the above, we say that a consequence operator C is well-behaved w.r.t T if $C(T) \neq L$. We focus theories that have more than one semantics, i.e. a backup semantics with associated consequence operator C_b and a intended semantics that corresponds to C_i. Slightly abusing terminology, if $\mathcal{T}$ is a set of theories, we say that a *twin semantics (for* $\mathcal{T}$*)* is a tuple $\mathcal{S} = (\mathcal{T}, C_b, C_i)$ with the following property of *supra-inferentiality*:

For all $A \subseteq L$, $C_b(A) \subseteq C_i(A)$ (*Supra*).

A *recovery operator* is a computable function $R : \mathcal{T} \to \mathcal{T}$. Given a twin semantics $\mathcal{S}$ for $\mathcal{T}$ and a recovery operator R on $\mathcal{T}$, we call the tuple $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ a *recovery framework*.

Properties of consequence operators
In this paper, we want to state some general results about the properties a suitable recovery operator should have. These properties partly depend on some abstract properties of the consequence operators C_b and C_i. Therefore we recall (see e.g. [9]) some general properties along which one can classify consequence operators:

$A \subseteq C(A)$ (*Inclusion*)
$C(A) = C(C(A))$ (*Idempotency*)
If $A \subseteq B$ then $C(A) \subseteq C(B)$ (*Monotony*)
If $A \subseteq B \subseteq C(A)$ then $C(B) \subseteq C(A)$ (*Cut*)
If $A \subseteq B \subseteq C(A)$ then $C(A) \subseteq C(B)$ (*Cautious Monotony*)

A *classical* inference operation C will also be denoted by Cn. An inference operation C is called *Tarskian*[2] if it satisfies *Inclusion*, *Idempotency* and *Monotony*, it satisfies *Cumulativity* if both *Cut* and *Cautious Monotony* hold for C. Finally, C is called a *cumulative* inference operation, if it satisfies *Inclusion* and *Cumulativity*.

[1] Note that under this definition, if $Mod(T) = \emptyset$, both a credulous and a skeptical Sem determines $Sem(T)$ to be not well-behaved.

[2] In particular, Cn is a Tarskian consequence operator

The following weaker forms of Cut and Cautious Monotony are also useful:

If $A \subseteq B \subseteq C(A)$ and $C(A) \neq L$ then $C(B) \neq L$ (*Weak Cut*)

If $A \subseteq B \subseteq C(A)$ and $C(B) \neq L$ then $C(A) \neq L$ (*Weak Monotony*)

To see that *Cut* implies *Weak Cut*, assume that $A \subseteq B \subseteq C(A)$ and $C(A) \neq L$. With *Cut* we infer that $C(B) \subseteq C(A)$ and, since $C(A) \neq L$, we immediately have $C(B) \neq L$, so *Weak Cut* holds. The same holds for the relation between *Cautious Monotony* and *Weak Monotony*. We say that C satisfies *Weak Cumulativity* if C satisfies both *Weak Cut* and *Weak Monotony*. Furthermore C is called *weakly cumulative* if it satisfies *Inclusion* and *Weak Cumulativity*.

The role of the weak principles in nonmonotonic logics

Our main motivation to introduce the weak variants of Cut and Monotony, is that they help us in distinguishing the mainstream nonmonotonic semantics from other (non-classical) semantics. Let us, following [9] and others, make a distinction between a *skeptical* and a *choice mode*[3] of using a consequence operator. It is well-known that mainstream nonmonotonic logics as Reiter's default logic, auto-epistemic logic and the stable model semantics of logic programming do not satisfy *Cautious Monotony* neither in the skeptical, nor in the choice mode. With respect to *Cut*, however, a distinction has to be made between these modes: while the skeptical modes of nonmonotonic consequence operations in general do satisfy *Cut*, their choice modes do not satisfy it (see [9]). This means these principles fail to distinguish these logics uniformly, i.e. independently from the mode in which they are used.

As we will show now, our weak principles are capable to characterize these mainstream nonmonotonic logics uniformly. We show, using default logic as an example, that irrespective of the mode (skeptical or choice) in which the nonmonotonic inference operator is used, these nonmonotonic operators all satisfy *Weak Cut*, but fail to satisfy *Weak Monotony*. By the correspondences between default logic and other nonmonotonic logics, this result also holds for auto-epistemic logic and the stable model semantics of logic programming.

Proposition 2.1 Let D be an arbitrary set of default rules and let C_D denote the inference operator using D and based on Reiter's default logic. Then C_D does not satisfy *Weak Monotony* but does satisfy *Weak Cut*, irrespective from the mode (skeptical or choice) in which it is used.

PROOF We first show that in none of the inference modes *Weak Monotony* is satisfied. It suffices to present one counter example. Take the following set of defaults $D = \{\frac{\emptyset;\neg c}{c}\}$. Consider $C_D(\emptyset) = L$. Since $\{c\} \subseteq L$, we have $\emptyset \subseteq \{c\} \subseteq L$, but *Weak Monotony* fails, since $C_D(\{c\}) = Cn(\{c\}) \neq L$, both in the skeptical and in the choice mode.

[3]We do not consider the credulous mode for consequence operators since it behaves rather irregularly. For example, $C_i(A)$ may contain both ϕ and $\neg\phi$ without also having $\phi \wedge \neg\phi$.

Next we show that Default Logic does satisfy *Weak Cut* in both modes. Since it satisfies *Cut* in the skeptical mode, we only have to show that it satisfies *Weak Cut* in the choice mode. From [10], we know that Default Logic satisfies the *Confirmation of Evidence* (CE) principle, stating that for every default theory $\Delta = (W, D)$, the theory $\Delta' = (W \cup W', D)$ has at least one extension E, whenever (W, D) has a consistent extension E and $W' \subseteq E$. Now let A, B be sets of sentences such that $A \subseteq B \subseteq C_D(A)$ and suppose that $C_D(A) \neq L$. Hence, both A and B are consistent sets of sentences and there is some consistent extension E_A of the default theory (A, D) such that $C_D(A) = E_A$. Since $B \subseteq E_A$, $B - A \subseteq E_A$, and hence, by the Confirmation of Evidence principle, there is at least one consistent extension E_B for $(A \cup (B - A), D) = (B, D)$. Select such an extension E_B. Since E_B is consistent, $C_D(B) \neq L$ and *Weak Cut* is satisfied.

Hence, Default Logic also satisfies *Weak Cut* if the choice mode of inference is used. ■

With respect to the dominant semantics of other formalisms like Auto-epistemic Logic and Logic Programming, we can easily show the same results. We conclude that irrespective of the mode in which consequence operators based on mainstream nonmonotonic semantics are used, they all satisfy *Weak Cut* and none of them satisfies *Weak Monotony*.

Some first results about recovery frameworks

Using the abstract principles we can derive some properties that will turn out to be useful when dealing with recovery.

Observation 2.2 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be an arbitrary recovery framework, where C_b satisfies Inclusion and C_i satisfies Weak Cut. Then for every $T \in \mathcal{T}$, $C_i(R(T)) \neq L$ implies $C_i(C_b(R(T))) \neq L$.

PROOF Let $T \in \mathcal{T}$. Since the underlying twin semantics $\mathcal{S} = (\mathcal{T}, C_b, C_i)$ satisfies $Supra$ and C_b satisfies $Inclusion$, we have $R(T) \subseteq C_b(R(T)) \subseteq C_i(R(T))$. By *Weak Cut*, we have $C_i(R(T)) \neq L$ implies $C_i(C_b(R(T))) \neq L$. ■

Observation 2.3 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be an arbitrary recovery framework, where C_b satisfies Inclusion and C_i satisfies Weak Monotony.
Then for every $T \in \mathcal{T}$, $C_i(C_b(R(T))) \neq L$ implies $C_i(R(T)) \neq L$.

PROOF Let $T \in \mathcal{T}$. Since the underlying twin semantics $\mathcal{S} = (\mathcal{T}, C_b, C_i)$ satisfies *Supra* and C_b satisfies *Inclusion*, we have $R(T) \subseteq C_b(R(T)) \subseteq C_i(R(T))$. By *Weak Monotony* it follows that $C_i(C_b(R(T)) \neq L$ implies that $C_i(R(T)) \neq L$. ■

Combining these observations, we have the following useful corollary:

Corollary 2.4 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be an arbitrary recovery framework, where C_b satisfies Inclusion and C_i satisfies Weak Cumulativity. Then for every $T \in \mathcal{T}$, $C_i(C_b(R(T))) \neq L$ iff $C_i(R(T)) \neq L$.

The conclusion of this corollary is a weaker variant of the well-known stronger absorption principle $C_iC_b = C_i = C_bC_i$ that holds when C_i is cumulative and C_i is supra-inferential with respect to an operator C_b satisfying *Inclusion* (cf. [9]).

2.2 The postulates

In this section we introduce our postulates for recovery and use them to define a successful recovery framework, of which we also give an example. We then derive some simple but useful results about recovery frameworks and elaborate on the weak principles.

Postulates for recovery
Given a recovery framework $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ we formulate some postulates to characterize a recovery approach for C_i-theories, using a backup semantics C_b.

R1. Success:
$C_i(R(T))$ is well-behaved whenever $C_b(T)$ is well-behaved.
This means that the recovery should be *successful*: if in the back-up semantics, one can attach a meaning to T, this postulate guarantees that $R(T)$ is well-behaved with respect to the intended semantics.

R2. Conservativity:
$R(T) = T$ whenever $C_i(T)$ is well-behaved.
This postulate guarantees that recovery is done in a *conservative* way: a recovery only leads to a change of T if it is necessary to do so, i.e, if one is unable to assign T a meaning under the intended semantics.

R3. Back-up preservation:
$C_b(T) \subseteq C_b(R(T))$.
Since the original theory is meaningful under the backup semantics, we do not want to lose information obtainable from the original theory when using the transformed theory $R(T)$.

R4. Back-up inclusion:
$C_b(R(T)) \subseteq C_b(T)$
This postulate constrains the recovery R by requiring that all backup consequences of $R(T)$ should be derivable from $C_b(T)$. That is, we are not allowed to add some new information when using the transformed theory.

The intention of a recovery framework is to characterize recovery operations that are both intuitively acceptable and *successful*:

Definition 2.5 We say that a recovery framework $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ is successful if for every $T \in \mathcal{T}$ such that $C_b(T)$ is well-behaved, $R(T)$ satisfies the postulates R1 to R4.

This does not exclude recovery frameworks that are successful in a *trivial* way, for example if $C_b(T)$ is not well-behaved for any $T \in \mathcal{T}$ or $C_i(T)$ is well-behaved for every $T \in \mathcal{T}$. Therefore, we define a *nontrivially* successful recovery framework as follows:

Definition 2.6 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a successful recovery framework. We say that $\mathcal{R}$ is *non-trivially* successful if there exists at least one $T \in \mathcal{T}$ such that $C_b(T)$ is well-behaved and $C_i(T)$ is not well-behaved.

Before we demonstrate that recovery frameworks exist that are non-trivially successful, we first make the following observation about successful frameworks, saying that the intended consequences of the recovered theory are bounded below by the backup consequences of the original theory.

Observation 2.7 If $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ is a successful recovery framework then $C_b(T) \subseteq C_i(R(T))$.

PROOF By Postulate R3, $C_b(T) \subseteq C_b(R(T))$. Since the underlying twin semantics $\mathcal{S} = (\mathcal{T}, C_b, C_i)$ satisfies *Supra*, $C_b(R(T)) \subseteq C_i(R(T))$. Hence, $C_b(T) \subseteq C_i(R(T))$. ∎

Example 2.8 Let us relate the postulates to an example. Consider the following program

$$\begin{array}{rrcl} P: & \neg a & \leftarrow & \text{not } b \\ & a & \leftarrow & \end{array}$$

Let us take for C_b the minimal model semantics, and for C_i the stable semantics. In this case, the conclusion $\neg a$ of the first rule is attacked by an observation a, without giving direct evidence for b to be true. Using the stable semantics, we have to conclude that the program is contradictory, since we expect both $\neg a$ and a to be true. Hence, there is no stable model for this program and we conclude that the associated (intended) inference operator C_i applied to the program P is not well-behaved. Still, we could reason as follows: if the program P represents all we know, then, from the apparent contradiction that both a and $\neg a$ seem to hold we would derive that it is impossible to assume that b is not true. Hence, we are forced to assume that b is true. But combining this information with the program P, we conclude that this recovered program has an intuitively acceptable stable model in which we expect both a and b to hold.

It is not difficult to see that such a line of reasoning is sanctioned by using the weaker minimal model semantics C_b: there is exactly one minimal model of P and in this model both a and b are true. We use such information in the recovery $R(P)$ of P, since it gives us a clue about what abnormalities should be taken into (explicit) account. Summarizing, when taking $R(P) = P \cup \{b \leftarrow\}$, we infer that for this particular P,this approach gives rise to a successful recovery.

3 Successful and unsuccessful frameworks

The recovery postulates R1-R4 restrict the class of possible recovery operations to the ones that are considered to be *acceptable*, i.e., well-behaved. They do not, however, tell us whether, given some class of theories, the class of acceptable recovery functions will be non-empty. That is, we do not know whether a recovery framework will be (nontrivially) successful or not.

In this section we will study the interaction between the postulates R1-R4 and some abstract properties of inference operations in order to to find out

1. in which cases the recovery framework cannot be applied, i.e. when the recovery of a theory as we have proposed is not possible in a successful way;
2. which recovery functions can be excluded if the recovery framework can be applied;
3. in which cases the recovery framework is guaranteed to be (nontrivially) successful.

3.1 General failure for Weak Cumulative semantics

We show that there is no recovery R satisfying the postulates if $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ is a recovery framework based on a twin semantics, where the intended semantics is weakly cumulative and the back-up semantics satisfies Inclusion.

In fact, we prove a slightly stronger result showing that every recovery framework satisfying the first three postulates R1-R3 cannot be non-trivially successful.

Theorem 3.1 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a recovery framework, where C_b satisfies Inclusion and C_i is weakly cumulative. Then $\mathcal{R}$ cannot be nontrivially successful with respect to the recovery postulates R1-R3.

PROOF Assume, on the contrary, that there exists a recovery framework $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ that is nontrivially successful and where C_i satisfies *Weak Cumulativity*. By Definition 2.6, there exists a theory $T \in \mathcal{T}$ such that $C_b(T)$ and $C_i(R(T))$ are well-behaved, but $C_i(T)$ is not.

Since C_i satisfies *Weak Cumulativity* and C_b satisfies *Inclusion*, by Observation 2.4, we have that $C_i(R(T)) \neq L$ implies $C_i(C_b(R(T)) \neq L$. Since, by assumption, $C_i(R(T)) \neq L$, it follows that $C_i(C_b(R(T)) \neq L$.

Since C_b satisfies *Inclusion* and R satisfies Postulate R3, we have

$$T \subseteq C_b(T) \subseteq C_b(R(T)) \tag{1}$$

By *Inclusion* and *Supra*, it follows that

$$C_b(R(T)) \subseteq C_b(C_b(R(T))) \subseteq C_i(C_b(R(T))) \tag{2}$$

Hence, combining these inclusions, we have

$$T \subseteq T \subseteq C_i(C_b(R(T)))$$

Since C_i satisfies *Weak Monotony* and $C_i(C_b(R(T)) \neq L$, by Observation 2.5 it follows that $C_i(T) \neq L$, contradicting the assumption that $C_i(T) = L$. So $\mathcal{R}$ cannot be nontrivially successful; a contradiction. ∎

The following corollary is immediate:

Corollary 3.2 *Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a recovery framework, where C_b satisfies Inclusion and C_i satisfies Weak Cumulativity. Then $\mathcal{R}$ cannot be nontrivially successful with respect to the recovery postulates R1-R4.*

This result shows that our framework cannot be used if the intended nonmonotonic semantics is (weakly) cumulative. This is the case with such systems as the Closed World Assumption (CWA), system C ([7]) and some reconstructions of Default Logic as Brewka's Cumulative Default Logic[4].

Note that our theorem also excludes such approaches if only the postulates R1-R3 are used. It is easy to show that, for example Pereira's Contradiction Removal Semantics ([12]) using the classical consequence operator Cn as the backup semantics, satisfies the postulates R1-R3 and hence cannot be successful if the intended semantics is cumulative.

Although there are some weakly cumulative nonmonotonic logics, as we remarked before, the mainstream semantics for nonmonotonic logics as Default Logic, Autoepistemic logic and nonmonotonic logic programming do not satisfy *Weak Cumulativity*, but satisfy weaker principles. So let us now consider the cases where the intended semantics is weaker and try to find out which types of recovery functions can or cannot be used that satisfy the postulates.

3.2 Failure for specific recovery functions

To exclude specific types of recovery functions in successful recovery frameworks, in this section we concentrate on two major types of recovery functions:

Definition 3.3 A recovery function R is called an *expansion* if for all $T \in \mathcal{T}$ we have $T \subseteq R(T)$ and R is called a *retraction* if for all $T \in T$ we have $R(T) \subseteq T$.

[4]and of course assuming a backup semantics that satisfies *Inclusion*.

It turns out that the two principles making up *Weak Cumulativity* nicely discriminate between expansions and contractions:

Theorem 3.4 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a recovery framework, where C_b satisfies Inclusion, C_i satisfies Weak Monotony and R is an expansion. Then $\mathcal{R}$ cannot be nontrivially successful w.r.t. the postulates R1-R4.

PROOF Suppose that $\mathcal{R}$ is nontrivially successful, then there is a theory $T \in \mathcal{T}$ such that $C_b(T)$ and $C_i(R(T))$ are well-behaved but $C_i(T)$ is not, i.e. $C_i(T) = L$. This implies that $R(T) \neq T$. Hence, since R is an expansion, $T \subseteq R(T)$. Therefore, since C_b satisfies Inclusion, we have $T \subseteq R(T) \subseteq C_b(R(T))$. From this, by R4 and Supra, it follows that $T \subseteq R(T) \subseteq C_b(T) \subseteq C_i(T)$. Using Weak Monotony, $C_i(R(T)) \neq L$ implies that $C_i(T) \neq L$. By assumption, $C_i(R(T)) \neq L$, hence $C_i(T) \neq L$; a contradiction. ■

On the other hand, if we require C_i to satisfy *Weak Cut* instead of *Weak Monotony*, *contraction* is no longer applicable:

Theorem 3.5 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a recovery framework, where C_b satisfies Inclusion, C_i satisfies Weak Cut and R is a contraction. Then $\mathcal{R}$ cannot be nontrivially successful w.r.t. the postulates R1-R3.

PROOF Suppose $\mathcal{R}$ is nontrivially successful, then there is a theory $T \in \mathcal{T}$ such that $C_b(T)$ and $C_i(R(T))$ are well-behaved but $C_i(T)$ is not. This implies that $T \neq R(T)$. Hence, since R is a contraction, $R(T) \subseteq T$. Therefore, since C_b satisfies *Inclusion*, we have $R(T) \subseteq T \subseteq C_b(T)$. By R3 and Supra, we have $R(T) \subseteq T \subseteq C_b(T) \subseteq C_b(R(T)) \subseteq C_i(R(T))$. Using *Weak Cut*, $C_i(R(T)) \neq L$ implies that $C_i(T) \neq L$. ■

This shows that *Weak Monotony* and *Weak Cut* nicely discriminate between two types of recoveries. Since we have seen that the dominant nonmonotonic logics satisfy *Weak Cut*, but not *Weak Monotony*, we conclude that

in general, retractions are not useful in non-monotonic theory recovery.

Example 3.6 We all know that, normally, adults do not read research reports; on the other hand, normally, researchers do. But normally, researchers are not prototypical adults. So suppose you meet a researcher R. Of course, you will expect R to read reports and you also expect him not to behave like a normal adult. But suppose that you learn that R is not reading research reports. Wouldn't you guess that he is an abnormal researcher? Let us model this example with Reiter's default theory and consider the following set of defaults:

$$D = \{\frac{researcher; \neg abResearcher}{abAdult}, \frac{\emptyset; \neg abResearcher}{readReports}, \frac{\emptyset; \neg abAdult}{\neg readReports}\}$$

If $W_1 = \{researcher\}$, i.e., we know that someone is a researcher, the default theory $\Delta_1 = (W_1, D)$ has exactly one (intuitively correct) extension

$$E = Cn(\{researcher, abAdult, readReports\}).$$

If we learn that this person does not read research reports, i.e. $W_2 = \{researcher, \neg readReports\}$, $\Delta_2 = (W_2, D)$ does not have a Reiter-extension, although it seems intuitively right to expect that $abResearcher$ will hold.

Let us look at a weaker default semantics, such as the minimal extension semantics[5], for which Δ_2 has an extension. Under this semantics, the unique minimal extension is

$$E_{min} = Cn(\{researcher, \neg readReports, abResearcher\}).$$

Taking this semantics as our backup semantics, we have $C_b(W_2) = E_{min}$.

Since we know that the consequence operator C_i based on the Reiter semantics satisfies *Weak Cut* and the minimal extension semantics is cumulative, according to Theorem 3.5 we cannot retract information from W_2 to obtain a Reiter extension for Δ_2.

We decide to apply an expansion: $R(W_2) = W_2 \cup \{abResearcher\}$ and, indeed, we observe that $C_b(W) = C_b(R(W))$ while $C_i(R(W))$ has a unique extension equal to E_{min}, so for this theory, the recovery operator satisfies the postulates.

3.3 Intermediate Conclusions

The results just obtained show that, first of all, our recovery framework for nonmonotonic theories contrasts with recovery frameworks for classical theories. To recover a classical theory from inconsistency, one is almost forced to apply a contraction operator to the inconsistent theory in order to rescue its intended meaning. Such a retraction, however, is not suitable for recovery of theories using a nonmonotonic semantics that satisfies both the weak principles we have introduced.

These results, however does not apply to this framework, but also to less restrictive frameworks as e.g. the Contradiction Removal framework of Pereira et al. (see [1]). Using the language of logic programming, the main idea behind this approach is that logic programs that do not have acceptable models can be revised adequately by removing assumptions. These assumptions are literals of the form not_l. Removal of such an assumption not_l can be accomplished by adding a rule $l \leftarrow$ to the program and taking the acceptable models of this expanded program as the intended models of the original program. It turns out that, taking a classical semantics as the backup semantics, the Contradiction Removal Semantics is a special recovery framework in which the postulates R1-R3 are satisfied. This means that (i) it cannot be applied if the intended semantics satisfies *Weak Cumulativity* and (ii) since the Contradiction

[5]A minimal extension of a default theory $\Delta = (W, D)$ is a minimal set E containing W and closed under classical consequences and application of default rules.

Removal Semantics aims at adding a minimal set of revisions, the expansion approach can be justified by pointing out that retraction never can be an option, as we will show in the next section.

At second look, one observes that the results we have obtained are *negative*: they do not tell us which conditions had to be satisfied in order to *guarantee* that a revision framework would be successful. Since we want to concentrate on recovery methods for mainstream nonmonotonic logics, in the next sections we will first of all show that, whenever the recovery framework can be applied but retraction is not possible, we can always rely on *expansion* as a successful recovery method.

Next, we will investigate which conditions have to be satisfied in order to apply an expansion successfully.

3.4 Applying Expansions

In this subsection we will concentrate on the use of expansions as useful recovery functions.

First we show that expansions can be used to represent more general recovery functions whenever these are successfully applicable. That is, expansions can be used as indicators (whenever contractions are not applicable) to check whether or not there exist successful recovery functions. It turns out that we can construct such an expansion in a uniform way. Next we show, that whenever one wants to recover a theory T by changing it in a minimal way, one has to use expansions. So in case of *minimal-change* recovery, expansions are the only successful recovery functions.

Our first result shows that expansion frameworks are able to represent all successful recovery frameworks, whenever the backup semantics is cumulative and the intended semantics is a nonmonotonic one, satisfying *Weak Cut*.

Theorem 3.7 Let $(\mathcal{T}, C_b, C_i)$ be a twin semantics where C_b is cumulative and C_i satisfies Weak Cut. Then there exists a successful recovery framework $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ satisfying the postulates R1-R4 iff there exists a successful recovery framework $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$, where R' is an expansion.

PROOF The if-direction is obvious: take $R = R'$. To prove the only-if direction, suppose that $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ is a successful recovery framework. Define the function R' as $R'(T) = R(T) \cup T$. Note that R' is an expansion.

We show that the framework $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$ is a successful recovery framework. So let $T \in \mathcal{T}$ and assume that $C_b(T) \neq L$.

- To show that postulate R1 is satisfied, it satisfies to show that $C_i(R'(T)) \neq L$.

  Since C_b is cumulative, it satisfies *Inclusion*, hence $T \subseteq C_b(T)$ and $R(T) \subseteq C_b(R(T))$. Since $R'(T) = R(T) \cup T$, it follows that $R'(T) \subseteq C_b(R(T)) \cup C_b(T)$. Moreover, since R satisfies R3 and R4, $C_b(R(T)) = C_b(T)$. Hence, we have $R'(T) \subseteq C_b(R(T))$. By *Supra* and the definition of $R'(T)$, we have $R(T) \subseteq R'(T) \subseteq C_b(R(T)) \subseteq C_i(R(T))$. Remember that R satisfies the recovery postulates, so $C_i(R(T)) \neq L$. Now applying *Weak Cut* immediately implies that $C_i(R'(T)) \neq L$. So R' satisfies R1.

- Postulate P2 is satisfied, since $C_i(T) \neq L$ implies that $R(T) = T$, since R satisfies Postulate R2. Hence $R'(T) = T$ and R' satisfies R2.
- To show that R3 and R4 are satisfied by R', we show that $C_b(T) = C_b(R'(T))$. We have $T \subseteq R(T) \cup T$. By *Inclusion* and the fact that R satisfies R3 and R4 it follows that $T \subseteq R'(T) = R(T) \cup T \subseteq C_b(R(T)) \cup C_b(T) = C_b(T) \cup C_b(T) = C_b(T)$. C_b is cumulative, so we have $C_b(R'(T)) = C_b(C_b(T)) = C_b(T)$, since cumulativity of C_b implies *Idempotency*.

■

Theorem 3.7 shows that using a mainstream nonmonotonic logic and a cumulative back-up semantics expansions are able to characterize successful recovery frameworks.

In some cases, however, we are able to prove a much stronger result. Let us define a recovery framework a *minimal change* recovery framework if the recovery operator R minimizes the difference between T and $R(T)$:

Definition 3.8 Let $(\mathcal{T}, C_b, C_i)$ be a twin semantics. We call $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ a successful minimal change recovery framework based on $\mathcal{S}$ if for every successful recovery framework $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$ based on $\mathcal{S}$ and every $T \in \mathcal{T}$ it holds that $R(T) \ominus T \subseteq R'(T) \ominus T$. Here $X \ominus Y = (X - Y) \cup (Y - X)$, the symmetric difference between X and Y.

It is not difficult to see that the only recovery operators that can be used in a successful minimal change recovery framework are expansions if we use a cumulative backup semantics and an intended semantics satisfying *Weak Cut*.

Theorem 3.9 Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a nontrivial successful minimal change recovery framework where C_b is cumulative and C_i satisfies *Weak Cut*. Then R has to be an expansion.

PROOF Let $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ be a nontrivial successful recovery framework where C_b is cumulative and C_i satisfies *Weak Cut*. Assume that R is not an expansion. By Theorem 3.5 R cannot be a contraction. Hence, there is a theory $T \in \mathcal{T}$ such that $R(T) = T' \cup N$ where $T' \subseteq T$, $T' \neq T$, $N \neq \emptyset$ and $N \cap T = \emptyset$. By (the proof of) Theorem 3.7, if we define a recovery operator R' by $R'(T) = R(T) \cup T$, the framework $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$ is also a successful recovery framework. Since $R'(T) = R(T) \cup T = (T' \cup N) \cup T = T \cup N$, we have

$$R'(T) \ominus T = N \subset (T - T') \cup N = R(T) \ominus T$$

But that implies that $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ cannot be a minimal change recovery framework; contradiction. Therefore R has to be an expansion. ∎

It might be difficult to find a successful expansion framework. If, however, we can assume that the backup semantics is cumulative, there is an easy way to tell whether or not there is a successful recovery framework:

Theorem 3.10 Let $(\mathcal{T}, C_b, C_i)$ be a twin semantics where C_b is cumulative and C_i satisfies Weak Cut. Then there exists a successful expansion framework $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ satisfying the postulates R1-R4 iff the recovery framework $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$ is successful, where R' is the full expansion $R'(T) = C_b(T)$ iff $C_i(T) = L$ and $R'(T) = T$ else.

PROOF
(⇒). Suppose that $\mathcal{R} = (\mathcal{T}, C_b, C_i, R)$ is a successful expansion framework. We show that $\mathcal{R}' = (\mathcal{T}, C_b, C_i, R')$, is successful, too. So assume $C_b(T) \neq L$. Since R is an expansion, by Postulate R4, $T \subseteq R(T) \subseteq C_b(T)$, By cumulativity of C_b, this implies $C_b(R(T)) = C_b(T) = C_b(C_b(T)) = C_b(R'(T))$. Hence, it is immediate that R' satisfies R3 and R4. By definition, R' satisfies R2. Finally, we have to show that R' satisfies R1, i.e. that $C_i(R'(T)) \neq L$. But that is easy, since $T \subseteq C_b(T) = C_b(R(T)) \subseteq C_i(R(T))$. Since $C_i(R(T)) \neq L$, by *Weak Cut* it follows that $C_i(C_b(T)) = C_i(R'(T)) \neq L$.
(⇐). Trivial, since $R'(T)$ by definition is an expansion. ∎

As a consequence, we can easily show now that the framework cannot be applied to the CWA with classical logic as the backup semantics: Take for example $T = \{p \vee q\}$. Now $Cn(T) \neq L$, but $CWA(T) = L$. Since $CWA(T) = CWA(Cn(T)) = L$, there exists no successful recovery framework satisfying the postulates R1-R4 for the closed world assumption semantics.

4 Recovery of Logic programs

The main goal of this section is to show that our recovery framework can be applied successfully to logic programming, especially to the stable model semantics of extended logic programs.

We assume to reader to be acquainted with the basic concepts and notations used in logic programming (cf. [5, 8]). We consider the class of finite, propositional normal logic programs with explicit negation and we will call such programs simply *logic programs*. Such a program consists of a finite set of rules of the form $l_0 \leftarrow l_1, \ldots l_m, not l_{m+1}, \ldots, not l_{m+n}$, $m, n \geq 0$, where each l_i is a literal Given some fixed set of propositional symbols, $\mathcal{P}_{elp}$ denotes the set of all normal programs with explicit negation.

As usual, an interpretation of a program P is denoted by the set of literals true in that interpretation. An interpretation M is called a model of P if M satisfies every rule of P. Given a model M of P, $G(P, M)$ denotes the Gelfond-Lifschitz reduction of P with M. $Rules(B_P)$ denotes the set of all possible rules that can be formed by using atoms occurring in P.

We use $Mod(P)$ to denote the set of classical models of P; $MinMod(P)$ denotes the set of minimal models and $Stable(P)$ the set of stable models of P. These sets are related by $Stable(P) \subseteq MinMod(P) \subseteq Mod(P)$.

Given such a semantics $Sem \in \{Mod, MinMod, Stable\}$ and a program P, we define the associated inference operation C^{Sem} as

$$C^{Sem}(P) = \{\phi \in Rules(B_P) \mid Sem(P) \models \phi\}$$

It is not difficult to show that for every P and every such a semantics Sem, C^{Sem} satisfies *Weak Cut*.

We will now prove a very general result for recovery of logic programs, showing that if the stable model semantics is used as the intended model semantics, we can use as our backup semantics every cumulative semantics Sem such that $Sem(P) \neq \emptyset$ and $Stable(P) \subseteq Sem(P) \subseteq MinMod(P)$, i.e. every semantics weaker than the stable semantics and consisting of minimal models [6]. We will call such a semantics a *potential back-up* semantics (w.r.t. the stable semantics):

Definition 4.1 Let Sem be a semantics for $\mathcal{P}_{elp}$. Sem is called a potential back-up semantics if for every P, $Stable(P) \subseteq Sem(P) \subseteq MinMod(P)$.

The following proposition is very helpful in proving properties of a potential back-up semantics Sem:

Proposition 4.2 Let Sem be a potential back-up semantics for $\mathcal{P}_{elp}$ and P a program such that $Sem(P) \neq \emptyset$. Then $Sem(P) \models \phi_{Sem,P}$ where $\phi_{Sem,P} = \bigvee_{M \in Sem(P)} \bigwedge M$.

PROOF Note that $\phi_{Sem,P}$ is just the disjunction of all (finite) models M in $Sem(P)$ expressed as (finite) conjunctions of literals true in M. ■

We will need the following lemma pertaining to properties of stable models:

Lemma 4.3 (Marek & Truszczyński [10]) Let M be a model of a program P and let $M_{G(P,M)}$ be the least model $G(P, M)$. Then $M_{G(P,M)} \subseteq M$.

The following lemma shows that a potential back-up semantics can be used as a back-up semantics in a successful recovery framework with the stable semantics as the intended semantics.

[6] An example of such a semantics is the positivist semantics (see [4]).

Lemma 4.4 Let Sem be a potential back-up semantics for $\mathcal{P}_{elp}$. Then, for every P such that $Sem(P) \neq \emptyset$, there is a program P' containing P, such that $C^{Sem}(P) = C^{Sem}(P')$ and $Stable(P') \neq \emptyset$.

PROOF Since $Sem(P) \neq \emptyset$, we also have $MinMod(P) \neq \emptyset$. Hence, according to Proposition 4.2, $Sem(P) \models \phi_{Sem,P}$ where $\phi_{Sem,P} = \bigvee_{M \in Sem(P)} \bigwedge M$. Let $CNF(\phi_{Sem,P})$ be the conjunctive normal form of $\phi_{Sem,P}$. For every disjunction $\delta = x_1 \vee x_2 \vee \ldots x_m$ occurring in $CNF(\phi_{Sem,P})$, let P_δ^{Sem} be the program

$$P_\delta^{Sem} = \{x_i \leftarrow \text{not } x_1, \ldots \text{not } x_{i-1}, \text{not } x_{i+1}, \ldots, \text{not } x_m \mid i = 1, \ldots, m\}$$

and, finally, let P^{Sem} be the union of all such programs P_δ^{Sem}, δ being a disjunction occurring in $CNF(\phi_{Sem,P})$. We show that $P' = P \cup P^{Sem}$ satisfies the conditions.

First of all, we prove the following claims:

- Claim 1. $P \subseteq P' \subseteq C^{Sem}(P)$.
  The first inclusion is by definition of P'. Since C^{Sem} is cumulative it satisfies *Inclusion*, hence $P \subseteq C^{Sem}(P)$. Furthermore, since $Sem(P) \models \phi_P$ it follows that $Sem(P) \models P^{Sem}$. Hence, $P^{Sem} \subseteq C^{Sem}(P)$. Therefore, $P' = P \cup P^{Sem} \subseteq C^{Sem}(P)$.

- Claim 2. *For every $M \in Sem(P)$ and every $l \in M$, P^{Sem} contains at least one rule $l \leftarrow \alpha$ such that $M \models l$ and $M \models \alpha$.*
  Let $M \in Sem(P)$ such that $l \in M$. By definition, $M \models l$. Since $Sem(P) \subseteq MinMod(P)$, M is a minimal model of $CNF(\phi_{Sem,P})$. Consider the set of disjunctions δ occurring in $CNF(\phi_P)$ such that l occurs in δ. Since M minimally satisfies $CNF(\phi_{Sem,P})$, there is at least one disjunction $\delta_l = l \vee z_1 \vee \ldots \vee z_k$ containing l, such that M minimally satisfies δ_l, i.e. for $i = 1, \ldots, k$, $M \not\models z_i$. Now $P_{\delta_l}^{Sem}$ contains the rule $l \leftarrow \text{not } z_1, \ldots, \text{not } z_k$ and $M \models \text{not } z_1 \wedge \ldots \wedge \text{not } z_k$. Hence, there exists at least one rule $l \leftarrow \alpha \in P^{Sem}$ such that $M \models l \wedge \alpha$.

From Claim 1 and the fact that C^{Sem} satisfies cumulativity, we immediately derive that $C^X(P) = C^X(P')$, i.e. the semantics X is invariant under the transformation from P to P'.

Using Claim 2, it is easy to show that every model $M \in X(P)$ is also stable model of P': we only have to prove that M is the minimal model of the reduction $G(P', M)$ of P' w.r.t. M. So let $M \in X(P)$. From Claim 2 above, it follows that, for every literal l occurring in M, there is at least one rule $l \leftarrow \alpha_l$ in $\bar{P}$ such that $M \models l \wedge \alpha_l$. Hence, since α contains only default-negated literals, by definition of $G(P', M)$, for every $l \in M$, the rule $l \leftarrow$ occurs in $G(P', M)$. Therefore, l occurs in the least model $M_{G(P',M)}$ of $G(P', M)$. This implies that $M_{G(P',M)} \supseteq M$. Hence, by Lemma 4.3, $M = M_{G(P',M)}$ and therefore, $M \in Stable(P')$. ∎

Note that, since Since C^{Stable} satisfies *Cut*, it also satisfies *Weak Cut*. However, $C^{Stable}(P)$ does not satisfy *Weak Monotony* as can be seen from the program $P = \{\neg a \leftarrow \text{not } b\ ;\ a \leftarrow\}$: Although $C^{Stable}(P) = L$, we have $C^{Stable}(P \cup \{b \leftarrow\}) \neq L$.

Hence, using the results obtained in the previous sections and the previous lemma, we can state the following main results:

Theorem 4.5 Let $\mathcal{R} = (\mathcal{P}, C^X, C^{Stable}, R)$ be a recovery framework for elp-programs, where X is a potential back-up semantics and for every $P \in \mathcal{P}$, $R(P) = P$ if $Stable(P) \neq \emptyset$ and $R(P) = P \cup P^X$ else. Then $\mathcal{R}$ is a successful recovery framework.

PROOF First of all, we note that $C^X(P) \subseteq C^{Stable}(P)$, by definition of the consequence operator for logic programs and the fact that $Stable(P) \subseteq X(P)$. We show that R1-R4 are satisfied whenever $C^X(P)$ is well-behaved:

- R1 If $C^X(P)$ is well-behaved, $X(P) \neq \emptyset$. By Lemma 4.4 and the definition of R it follows immediately that $Stable(R(P)) = Stable(P \cup P^X) \neq \emptyset$ and therefore $C^{Stable}(R(P))$ is well-behaved.
- R2 By definition of R
- R3+R4 Again by Lemma 4.4.

■

Theorem 4.6 Let $\mathcal{R} = (\mathcal{P}, C_b, C^{Stable}, R)$ be a non-trivially successful minimal-change recovery framework for elp-programs, where C_b is cumulative. Then R has to be an expansion.

PROOF Immediately from Theorem 4.5 and Theorem 3.9 ■

Remark. Since we do not require the intended semantics to be two-valued, it is also possible to revise logic programs with explicit negation using the Well-Founded (WF) semantics (see [14]) as the intended semantics and, for example, the standard three-valued Kleene semantics as the backup semantics. A program like:

$$\begin{array}{rcl} P\ :\ a & \leftarrow & \\ c & \leftarrow & a, \text{not } b \\ \neg c & \leftarrow & \end{array}$$

does not have an acceptable WF-model: $WF(P) = \{a, c, \neg c, \neg b\}$ is contradictory. Its least three-valued model (under the knowledge-ordering of truth-values), however, is $M = \{a, \neg c\}$. So the following program

$$\begin{array}{rcl} P'\ :\ a & \leftarrow & \\ c & \leftarrow & a, \text{not } b \\ \neg c & \leftarrow & \\ b & \leftarrow & \text{not } b \end{array}$$

has an acceptable well-founded model $WF(P') = \{a, \neg c\}$ identical to M. ■

5 Conclusions

We have presented a framework and some postulates for recovery of nonmonotonic theories. We have shown that in case the intended semantics is a mainstream nonmonotonic semantics, under very general conditions set for the back-up semantics, recovery cannot be accomplished by retraction operations. This distinguishes nonmonotonic recovery from the AGM framework for recovery of classical theories.

This leaves only room for recovery operations in which either a part of the theory is retracted and at the same time information is added to the resulting theory as well (pure) expansion operators in which a theory is recovered by adding information to it. As a special case, the Contradiction Removal framework developed by Pereira and Alferes (see [1]), satisfies our first three rationality postulates and makes use of expansions as recovery operators.

Our results show that, whenever R is a mixed recovery that satisfies the postulates R1-R4, it can always be replaced by a successful expansion that does not produce more changes. In particular, we have shown that whenever the backup semantics is cumulative, syntactically minimal recovery operators for nonmonotonic theories have to be expansions in order to be successful.

This result can be related to the approach to theory recovery of Inoue and Sakama (see [6]), where they propose to revise a theory T by means of a minimal set of additions I and removals O such that $R(T) = T + I - O$ has an acceptable model. Their proposal thus comes down to advocating a *mixed recovery* approach. Our results show that, whenever R is a mixed recovery that satisfies the postulates R1-R4, it can always be replaced by a successful expansion that does not produce more changes.

Finally, in a case study of recovery in nonmonotonic logic programming, we have shown that a stable model for a classical consistent program always can be approximated using a weaker cumulative (backup) semantics. The evidential semantics presented by Seipel ([13]) can be seen as a special case of our framework, taking (partial) minimal model semantics as the backup semantics and (partial) stable model semantics as the intended semantics.

References

[1] J.J. Alferes, L. M. Pereira, *Reasoning with Logic Programming*, Springer Verlag, Heidelberg, 1996.

[2] J. Dix, *Nichtmonotones Schliessen und dessen Anwendung auf Semantiken logischer Programme*, PhD. Thesis, Karlsruhe, 1992.

[3] P. Gärdenfors, *Belief Revision*, Cambridge University Press, 1992.

[4] N. Bidoit, Negation in rule-based database languages: a survey, *Theoretical Computer Science*, **78**, (1991), 3–83.

[5] J. Dix and G. Brewka, Knowledge Representation with Logic Programs, Research Report 15/96, Fachberichte Informatik, Universität Koblenz-Landau, Koblenz, 1996.

[6] K. Inoue, C. Sakama, Abductive Framework for Nonmonotonic Theory Change. *Proceedings IJCAI'95*, 1995.

[7] S. Kraus, D. Lehmann, M. Magidor, NonMonotonic Reasoning , Preferential Models and Cumulative Logics, *Artificial Intelligence*, 44 (1), 1990, pp. 167–207.

[8] J. Lloyd, Foundations of Logic Programming, Springer Verlag, 1988.

[9] D. Makinson, General Patterns in Nonmonotonic Reasoning, in: D.M. Gabbay, C.J. Hogger, J.A. Robinson (eds) *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 3, Nonmonotonic Reasoning, Oxford Science Publications, Oxford, 1994.

[10] V. Marek and M. Truszczyński, *Nonmonotonic Logic*, Springer Verlag, Heidelberg, 1993.

[11] Morris, P., Stable Closures, Defeasible Logic and Contradiction Tolerant Reasoning, *Proceedings of the 7th National Conference on Artificial Intelligence*, 1988.

[12] L. M. Pereira, J. J. Alferes and J. N. Aparicio. Contradiction Removal within well-founded semantics. In: A. Nerode, W. Marek and V. S. Subrahmanian, (eds.), *First International Workshop on Logic Programming and Non-monotonic Reasoning*, MIT Press, 1991.

[13] D. Seipel. Partial Evidential Stable Models For Disjunctive Deductive Databases. In: T. Przymusinski (ed.) Proc. Workshop on Logic Programming and Knowledge Representation (LPKR'97, 1997.

[14] A. van Gelder, K.A. Ross, J.S. Schlipf. Unfounded Sets and Well-Founded Semantics for General Logic Programs, Proceedings Seventh ACM Symposium on Principles of Database Systems, (PODS'88), 1988, pp. 221-230.

[15] C. Witteveen and W. van der Hoek, Revision by Communication. In: V. Marek, A. Nerode and M. Truszczyński, editors, *Logic Programming and Non-Monotonic Reasoning, LNAI 928,* pp. 189–202, Springer Verlag, 1995.

[16] C. Witteveen and W. van der Hoek, Some Postulates for Nonmonotonic Theory Revision Applied to Logic Programming. In: R. Dyckhoff, H. Herre and P. Schroeder-Heister, editors, *Extensions of Logic Programming, LNAI 1050*, pp. 303–318, Springer Verlag, 1995.