Classical and General Frameworks for Recovery

Wiebe van der Hoek[1] and **Cees Witteveen**[2]

Abstract. Theory recovery is the process of restoring consistency of a theory with respect to some intended semantics. Recently, a general framework for theory recovery has been developed using a twin semantics. In this paper we establish some relationships between this recovery framework and the well-known AGM postulates for theory revision. We show that the AGM revision postulates can be easily adapted to theory recovery and AGM-style recovery can be embedded in the general recovery framework. We also generalize the AGM postulates to deal with recovery of not necessarily closed theories and we show that the general recovery framework specialised to a classical intended semantics satisfies the AGM postulates. It turns out that the backup semantics used is a special cumulative, non-inclusive paraconsistent semantics. Finally, we generalize the AGM postulates for recovery to cases where the intended semantics does not need to be classical. We show that in such cases the postulates allow for recovery by expansions.

1 INTRODUCTION AND MOTIVATION

Theory recovery refers to the process of restoring consistency of a theory that, according to some *intended semantics*, is inconsistent. In [7], a framework called *twin-semantics* for theory recovery was introduced where a weaker *backup semantics* is used to guide this recovery process. The recovery process was captured by using a recovery function R that, given a theory T, transforms it into another theory T' in order to restore consistency. The general conception there was that the intended semantics allows one to draw more conclusions from T than the backup semantics, capturing the idea that, when everything (described by T) is *as expected*, reasoning from T is done firmly. However, as soon as some *abnormalities* are obtained (in T), the intended semantics may easily fail to assign a meaning to T, and the reasoner can fall back to a more modest, but weaker semantics (the backup semantics), for which T still makes sense. The challenge then is to give an account for the abnormalities in some theory $T' = R(T)$ which, (i) from the perspective of the backup semantics, is equal to T and (ii) can be assigned a non-trivial meaning in the intended semantics, so that one can carry on reasoning using this semantics, together with T'. The theory T is said to be *recovered* using R, yielding $T' = R(T)$.

The emphasis in [7] was on giving rationality postulates for R, and, given these postulates, to establish a relation between the *type* of recovery function (in [7], *contractions, expansions* and *mixed recovery* functions were distinguished) on the one hand, and *abstract properties* of the intended and backup semantics, on the other hand.

[1] Utrecht University, Department of Computer Science, Padualaan 14, 3584 CH Utrecht, The Netherlands, email: wiebe@cs.uu.nl

[2] Delft University of Technology, Department of Information Technology and Systems, P.O.Box 356, 2600 AJ Delft, The Netherlands, email: witt@cs.tudelft.nl

In particular, in [7] it was demonstrated that some nonmonotonic backup semantics would require R to be an expansion instead of a contraction whenever the intended semantics would render a theory T as inconsistent.

In principle, one can iterate the process of recovering: in [8] an example was given of a default theory T and three types of semantics. The theory only had a meaning in the weakest semantics, which could be used to transform T into some recovered theory $R_1(T)$. This new theory still was inconsistent for the third semantics, but now the second semantics was able to yield a new theory $R_2(R_1(T))$ which was meaningful for the third semantics. (Note that in this case, the second semantics plays both the role of an intended, and as a backup semantics).

However, in all the examples considered thus far, the weakest backup semantics used was *classical*, i.e., it was always assumed that the theories to be recovered were classically consistent —only from the perspective of a stronger, supra-classical, semantics the theories could be inconsistent. In this paper, we want to relax this assumption, and focus on the case where the *intended semantics* is that of classical logic. Thus, we want to put the twin-semantics to work to recover classically inconsistent theories.

Dealing with classically inconsistent theories, a natural challenge is to relate the twin-semantics approach to the prominent AGM framework of theory revision ([1]). To do so, we firstly turn the AGM revision postulates into recovery postulates and show that every AGM recovery framework based on these postulates can be embedded in the general recovery framework. Then we adapt the AGM postulates for recovery to base theories and we show that every general recovery framework where the intended semantics is classical, satisfies the AGM postulates for recovery. With respect to the abstract properties of the backup semantics involved, we show that every backup semantics necessarily has to be cumulative. Finally, we generalize the AGM-inspired recovery frameworks to intended semantics that do not need to be classical. We show that in such cases AGM recovery frameworks also allow for recovery by expansion. We conclude that in AGM-inspired recovery, the recovery method is dependent on the abstract properties of the intended semantics used.

1.1 Preliminaries

Given a language L, a *theory* T is any, not necessarily closed, subset of L. Given a set of theories, a *consequence operation* $C : 2^L \to 2^L$ is said to be *well-behaved w.r.t.* T if $C(T) \neq L$. Usually, such a C is induced by a semantics for L. We focus on theories that have more than one semantics, i.e. a *backup semantics* with associated consequence operator C_{bck} and an *intended semantics* that corresponds to C_{int}. Slightly abusing terminology, if $\mathcal{T}$ is a set of theories, we say that a *twin semantics (for* $\mathcal{T}$) is a tuple $\mathcal{S} = (\mathcal{T}, C_{bck}, C_{int})$ with the following property of *supra-inferentiality*:

For all $A \subseteq L$, $C_{bck}(A) \subseteq C_{int}(A)$ (*supra*)

To measure the difference between two theories w.r.t. the backup semantics, a *distance function* $d_{bck} : 2^L \times 2^L \to R^+$ is used. This function is minimally specified by (i) $d_{bck}(A, B) = 0$ whenever $C_{bck}(A) = C_{bck}(B)$ and (ii) $d_{bck}(A, B) = d_{bck}(B, A)$ for all $A, B \subseteq L$. A *recovery operator* is a computable function $R : \mathcal{T} \to \mathcal{T}$. A recovery operator 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 $R(T) \subseteq T$. Given a twin semantics $\mathcal{S}$ for $\mathcal{T}$, a recovery operator R on $\mathcal{T}$ and a distance function d_{bck}, we call the tuple $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ a *recovery framework* (*using* d_{bck}).

Given a recovery framework $\mathcal{R}$, it turns out that the properties a suitable recovery operator R should have partly depend on some abstract properties of the consequence operators C_{bck} and C_{int}. Therefore we recall (see e.g. [5]) 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*[3] 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. 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*)

We say that C satisfies *weak cumulativity* if C satisfies both weak cut and weak monotony and C is called *weakly cumulative* if it satisfies inclusion and weak cumulativity.

2 RECOVERY USING TWIN SEMANTICS

We recall ([7]) the following postulates for recovery:

R1. Success : $C_{int}(R(T)) \neq L$ whenever $C_{bck}(T) \neq L$.
R2. Conservativity : $R(T) = T$ whenever $C_{int}(T) \neq L$.
R3. Sufficiency : $R(T) \subseteq C_{bck}(T)$.
R4. Minimization : $d_{bck}(T, R(T)) \leq d_{bck}(T, R'(T))$ for every R' satisfying R1-R3.

The intention of these postulates is to characterize recovery operations that are both intuitively acceptable and *successful*: A recovery framework $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ is *successful* if, for every $T \in \mathcal{T}$, $R(T)$ satisfies the postulates R1 to R4, and, moreover, there exists a $T \in \mathcal{T}$ such that $C_{bck}(T)$ is well-behaved while $C_{int}(T)$ is not well-behaved. If, moreover, it also holds that $d_{bck}(T, R(T)) = 0$, we say that postulate R4 is *strongly* satisfied and $\mathcal{R}$ is *strongly successful*. $\mathcal{R}$ is said to be *everywhere* successful if for every T, $C_{int}(C_{bck}(T)) \neq L$. Finally, a recovery framework $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ is said to be *saturated* if $R = C_{bck}$.

We briefly restate some results about $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$, as obtained in [7]. First of all, if C_{bck} satisfies inclusion and C_{int} is weakly cumulative, $\mathcal{R}$ cannot be strongly successful. A similar negative result applies to expansions: if R is an expansion, C_{bck} satisfies inclusion, and C_{int} weak monotony then again, $\mathcal{R}$ cannot be strongly successful. Surprisingly enough, there are also conditions under which one cannot successfully recover by performing a contraction, i.e., in cases that C_{bck} satisfies inclusion, and C_{int} satisfies weak cut.

On the basis of these results one may conclude that the distinction Makinson ([5]) has made between two clusters of nonmonotonic consequence operators, one cluster satisfying inclusion and cut (the grounded cluster) and the other cluster satisfying cumulativity (the minimal model cluster), has some major consequences for the type of recovery operation to apply, viz. that *expansions* cannot be applied in the *minimal model cluster* characterized by preferential entailment and Poole's default logic; and *contractions* are not useful in the *grounded cluster* containing a.o. DL, AEL and NMLP.

Fortunately, one can also identify conditions under which recovery frameworks *are* successful. First of all, it appears that recovery by expansion is as successful as general recovery frameworks whenever the backup semantics is cumulative and the intended semantics is a nonmonotonic one, satisfying weak cut. This 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, a much stronger result holds. Let us define a *minimal change* recovery framework as a recovery framework where R minimizes the difference between T and $R(T)$, i.e., where, for every successful recovery framework $\mathcal{R}'$ it holds that $R(T) \ominus T \subseteq R'(T) \ominus T$, where $X \ominus Y = (X - Y) \cup (Y - X)$, the symmetric difference between X and Y. Using this machinery, it is not difficult to prove 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*.

3 AGM REVISION AND THEORY RECOVERY

The recovery framework as discussed above differs from the standard AGM approach[1, 3] to theory revision in three respects:

1. As exemplified in the AGM approach to revision, only one semantics, the intended semantics, is used. Moreover, this semantics is a *classical* one.
2. The AGM framework deals with *revision* of a theory if new information has to be incorporated, instead of recovery. So in order to compare the two frameworks more adequately, we should either translate the recovery framework into a revision framework or vice-versa.
3. The AGM approach deals with (classically) closed theories, while the general recovery framework does not require theories to be closed.

To deal with these differences, first we will simply translate the AGM revision postulates into postulates for recovery. Using these postulates, we define AGM recovery framework for closed theories that characterizes the class of recovery functions satisfying these postulates. We show that AGM recovery can be embedded in the general recovery framework.

[3] In particular, the classical consequence operator Cn is a tarskian consequence operator.

Next, these postulates will be adapted in a simple way to deal with theories that are not closed under classical consequence. The result is a set of four AGM-style theory recovery postulates.

We show that whenever the intended semantics is restricted to a classical semantics, the general recovery framework can be embedded into an AGM-recovery framework for base theories. Using this correspondence, we derive some properties of the backup semantics when AGM recovery is seen as a special case of the recovery framework. It turns out that in this case the backup semantics is a special paraconsistent semantics, satisfying cumulativity.

Finally, we address some issues when AGM recovery is generalized, allowing not only for a classical semantics as the intended semantics but also for stronger (supra-classical) semantics to occur.

3.1 AGM revision and AGM recovery

To start with, let us take the well-known AGM postulates for revision.

K*1. $Cn(K * A) = K * A$
K*2. $A \in K * A$
K*3. $K * A \subseteq K + A$
K*4. If $\neg A \notin K$, then $K + A \subseteq K * A$.
K*5. $K * A \neq L$ iff $\nvdash \neg A$
K*6. $A \leftrightarrow B$ implies $K * A = K * B$
K*7 $K * (A \wedge B) \subseteq (K * A) + B$
K*8 $\neg B \notin K * A$ implies $((K * A) + B) \subseteq K * (A \wedge B)$

To embed the AGM framework for revision into the general recovery framework, first of all we have to find a natural way for modeling recovery as a particular form of revision. A very simple and natural solution is to use an AGM recovery operator R_{AGM} defined as the revision of a theory T with $\top$: $R_{AGM}(T) = T * \top$. It is easily seen that according to the postulates this operator restores consistency whenever it is necessary (K*5) and leaves consistent (classically) closed theories untouched (K*3 + K*4).

Taking $A = \top$, the standard AGM postulates for revision reduce to the following AGM-recovery postulates AGM1 - AGM4:

From K*1 we derive

AGM1* $Cn(R_{AGM}(T)) = R_{AGM}(T)$

Since K*2 translates to $\top \in R_{AGM}(T)$, it is a consequence of AGM1 (and the properties of Cn). So our second postulate is derived from K*3,taking into account that $T + \top = Cn(T) + \top = T$:

AGM2* $R_{AGM}(T) \subseteq T$

The next postulate is a straightforward translation from K*4:

AGM3* If $T \neq L$, then $T \subseteq R_{AGM}(T)$

Clearly, K*5 translates to

AGM4* $R_{AGM}(T) \neq L$

The remaining postulates now all are covered by these four postulates: Note that K*6 is trivially satisfied since we only allow $A = B = \top$; K*7 reduces to $R_{AGM}(T) \subseteq R_{AGM}(T) + \top$ and is implied by AGM1; finally, K*8 reduces to: $\perp \notin R_{AGM}(T)$ implies $(R_{AGM}(T) + \top) \subseteq R_{AGM}(T)$. By AGM4, this reduces to $R_{AGM}(T) + \top \subseteq R_{AGM}(T)$ and again, this inclusion can be easily derived from AGM1 and the properties of Cn. Note that, by AGM2, contraction is a hard-wired property of AGM recovery operators R_{AGM}, of course due to the fact that all theories are closed.

Let $\mathcal{T}$ be the class of classically closed theories. Note that by AGM2*, for every $T \in \mathcal{T}$, $R_{AGM}(T) \subseteq Cn(T)$, hence Cn is supra-inferential w.r.t. C_{bck}, so $(\mathcal{T}, R_{AGM}, Cn)$ is a twin semantics. Therefore, we can define an *AGM-recovery framework* as a special (saturated) recovery framework $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, Cn, R_{AGM})$ where $R_{AGM} = C_{bck}$ and R_{AGM} satisfies the postulates AGM1*-AGM4*. The following result shows that an AGM-recovery framework can be embedded in the general recovery framework:

Theorem 1 *Let $\mathcal{T}$ be a class of (classically) closed theories and $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, Cn, R_{AGM})$ an AGM-recovery framework with $R_{AGM} = C_{bck}$. Then $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, Cn, R_{AGM})$ is an everywhere strongly successful recovery framework satisfying the recovery postulates R1-R4.*

PROOF By AGM1* and AGM4*, we have $Cn(R_{AGM}(T)) = R_{AGM}(T) \neq L$. Hence, postulate R1 is satisfied. By AGM2* + AGM3*, $R_{AGM}(T) = T$, whenever $Cn(T) \neq L$, so postulate R2 is satisfied. Since $R_{AGM}(T) = C_{bck}(T)$, postulate R3 is trivially satisfied. Finally, by AGM4*, $R_{AGM}(T) \neq L$, hence, by AGM2*+AGM3*, it follows that $R_{AGM}(T) = R_{AGM}(R_{AGM}(T))$. Since $R_{AGM} = C_{bck}$, $C_{bck}(R_{AGM}(T) = C_{bck}(T)$ and postulate R4 is (strongly) satisfied. ∎

What can we say about the revision operator R if the class $\mathcal{T}$ of theories we consider only contains classically closed theories? First of all, note that there is only one theory T that is inconsistent under the intended semantics Cn, viz. $T = L$, and that this is the only case where $R(T) \neq T$. Hence, R is completely determined if $R(L)$ has been specified. Classical AGM approaches to contraction acknowledge the idea that a contraction of T with a sentence φ has to deal with maximal subsets $T' \subseteq T$ for which $T' \nvdash \varphi$. Let us denote the set of all such subsets T' of T by $T \angle \varphi$. Then, a contraction of T with φ typically is comprised of making some selection from the set $T \angle \varphi$. To fix some more notation, for any set $\mathcal{X}$ of sets X, let $Sel(\mathcal{X})$ denote any collection $\{X_1, X_2, \ldots\}$ from $\mathcal{X}$ and let Sel_1 be a selection function such that, $Sel_1(\mathcal{X})$ is always a singleton. Specializing to the case where $T = L$ and $\varphi = \perp$, we must restrict such selections to $L \angle \perp$, the set of (classically) maximal consistent sets Σ.

1 *Maxichoice*. Here, $R(T) = Cn(Sel_1(L \angle \perp)) = Sel_1(L \angle \perp)$. Thus, recovery from classical inconsistency here is simply falling back upon one designated maximal consistent set. Such a recovery supports the wish to hold on to a maximal number of consequences of T ($R(T)$ does not allow any additional conclusion), but suffers from the arbitrariness of Sel_1.
2 *Full meet*. This kind of recovery takes all maximal sets into account, and is defined by $R(T) = Cn(\bigcap(L \angle \perp)) = Cn(\emptyset)$. This seems to be the most careful way of recovering: as soon as an inconsistent theory T is encountered, one removes all contingent facts from T.
3 *Partial meet*. Tries to combine the best of two worlds by defining $R(T) = Cn(\bigcap Sel(L \angle \perp)) = \bigcap Sel(L \angle \perp)$. Here, upon hitting an inconsistency, one falls back upon some determined consistent theory $R(T)$, which does not have to be a maximal consistent set.

It follows immediately that all three recovery functions R described above lead to successful recovery frameworks $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ and, conversely, every such framework can be conceived as the result of applying a partial meet contraction. To see the latter, observe that for a successful framework, R has to satisfy $R(L) = Cn(A)$ for some consistent set of sentences A. Let

M be the set of classical models m for which $m \models A$. Note that the theory of every m, $Th(m)$, is a maximal consistent set. Hence, $R(T) = Cn(\bigcap Sel(L\angle\bot)) = Cn(\bigcap\{Th(m)|m \models A\}$.

Summarizing, when restricting ourselves to classically closed theories, the contraction T' of every inconsistent theory T is the same and rather arbitrary. Hence, we will look into theories that are not necessarily classically closed and we will adapt the AGM postulates accordingly.

3.2 Adapting the AGM postulates to non-closed theories

Since we want to deal with AGM recovery of not necessarily closed theories, we will generalise the postulates AGM1*- AGM*4. The simplest generalisation is to replace in these postulates every occurrence of a theory T by its closed form $Cn(T)$: whenever we have to deal with closed theories, these new postulates will reduce to the original AGM postulates for recovery:

AGM1 $Cn(Cn(R_{AGM}(T))) = Cn(R_{AGM}(T))$;
AGM2 $Cn(R_{AGM}(T)) \subseteq Cn(T)$;
AGM3 If $Cn(T) \neq L$, then $Cn(T) \subseteq Cn(R_{AGM}(T))$;
AGM4 $Cn(R_{AGM}(T)) \neq L$.

Note that in the context of classical recovery, postulate AGM1 is superfluous as it is implied by idempotency of Cn. However, since we want to generalize from classical recovery later on, we will keep it here. We now easily derive the following properties of R_{AGM}:

Proposition 1 *Let $R_{AGM}(T) = T * \top$ satisfy the AGM postulates AGM1-AGM4. Then (i) $Cn(R_{AGM})$ is an idempotent operator. (ii) $Cn(R_{AGM}(T)) = Cn(T)$ whenever T is (classically) consistent.*

Let us now again consider a general recovery framework where the intended semantics is classical, i.e. $C_{int} = Cn$. Surprisingly, it turns out that if the framework is everywhere strongly successful, it also satisfies the AGM postulates for recovery, whatever the backup semantics may be:

Theorem 2 *Let $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ be an everywhere strongly successful recovery framework, where $C_{int} = Cn$. Then R satisfies the AGM postulates AGM1-AGM4 for recovery.*

PROOF AGM1 is trivially satisfied by the idempotency of Cn; By R3, $R(T) \subseteq C_{bck}(T)$. Since $C_{int} = Cn$ is supra-inferential, we have $R(T) \subseteq C_{bck}(T) \subseteq Cn(T)$; by monotony and idempotency of Cn it follows that $Cn(R(T)) \subseteq Cn(T)$; so AGM2 follows. By R2, we have $R(T) = T$, whenever $Cn(T) \neq L$, immediately implying AGM3. By R3, $R(T) \subseteq C_{bck}(T)$, hence by monotony of Cn, $Cn(R(T)) \subseteq Cn(C_{bck}(T))$. Since $\mathcal{R}$ is everywhere successful, $Cn(bup(T)) \neq L$. Hence, $Cn(R(T)) \neq L$ and AGM4 is satisfied. ∎

Note that this embedding result heavily depends on the monotonic properties of Cn. Also note that this embedding is not restricted to saturated recovery frameworks and that the embedding does not depend on the choice of the particular backup semantics involved: it only requires Cn to be supra-inferential w.r.t. C_{bck}.

Remark Note that the converse relation does not hold: there are recovery functions R satisfying the AGM postulates AGM1-AGM4 that cannot be used in any recovery framework satisfying the recovery postulates R1-R4. To give an example: take a (full meet) recovery R such that $R(T) = Cn(T)$ whenever T is consistent and $R(T) = Cn(\emptyset)$ else. This recovery function satisfies the AGM postulates AGM1-AGM4 but violates recovery postulate R2 whenever T is not closed. It is possible, however to show that the recovery postulates can be satisfied by weakening R2 to the following postulate wR2:

wR2 $C_{int}(R(T)) = C_{int}(T)$ whenever $C_{int}(T) \neq L$

and changing the distance function d_{bck} to $d_{bck}(A, B) = 0$, whenever $C_{int}(C_{bck}(A)) = C_{int}((C_{bck}(B))$. After these modifications, it can be shown that AGM recovery can be strictly embedded in (a weaker) general recovery framework.

3.3 Paraconsistent semantics and AGM-recovery

Although Theorem 2 showed that the embedding result is independent from the specification of the backup semantics, it turns out that we can easily derive some special properties that have to hold for any backup semantics used in a classical recovery framework.

In general, a semantics represented by its consequence relation C, is called *paraconsistent* if there exists a theory T such that $C(T) \neq L$, while $Cn(T) = L$.

If $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, C_{int}, R)$ is an everywhere strongly successful recovery framework, where $Cn(C_{bck}) = C_{bck}$, i.e. the backup semantics satisfies left absorption, C_{bck} is an example of such a para-consistent consequence operator [4] We would like to characterize the abstract properties a paraconsistent backup semantics should satisfy if (i) $R_{AGM} = C_{bck}$ and (ii) $C_{bck} = Cn(C_{bck})$ - i.e. the backup semantics is classically closed - in an (everywhere) successful AGM recovery framework:

Proposition 2 *Let $\mathcal{R} = (\mathcal{T}, C_{bck}, d_{bck}, Cn, R_{AGM})$ be an everywhere strongly successful AGM recovery framework, where $Cn \circ C_{bck} = C_{bck} = R_{AGM}$. Then C_{bck} is a paraconsistent consequence operator satisfying non-inclusiveness, idempotency and cumulativity, but not necessarily monotony.*

PROOF Paraconsistency follows immediately. If $Cn(T) = L$, by AGM4 we have $Cn(C_{bck}(T)) = Cn(R_{AGM}(T)) \neq L$, hence $Cn \circ C_{bck}$ is not inclusive. Idempotency follows from Proposition 1.

Cumulativity also follows easily: First of all, if $T \subseteq T' \subseteq C_{bck}(T)$, then T must be consistent. For else, by inclusion and monotony of Cn we would have $T \subseteq Cn(T') \subseteq Cn(C_{bck}(T))$. Now, $Cn(T') = L$ and, by AGM4, $Cn(C_{bck}(T)) \neq L$; contradiction. So, we can assume that T is consistent. Now by supra-inferentiality, monotony and idempotency of Cn we have $T \subseteq T' \subseteq Cn(C_{bck}(T)) \subseteq Cn(Cn(T) = Cn(T)$. Then, by cumulativity of Cn we derive $Cn(T') = Cn(T)$. Since, T is consistent, by AGM2+AGM3, $Cn(C_{bck}(T)) = Cn(T) = Cn(T') = Cn(C_{bck}(T'))$. Therefore, $C_{bck} = Cn \circ C_{bck}$ satisfies cumulativity. Monotony does not need to be satisfied: for example take a maxi-choice recovery function R_{AGM} and two theories T and T' such that $Cn(T') = L$ and T is consistent. Then $T \subseteq T'$, while $C_{bck}(T) = Cn(T) \supseteq C_{bck}(T')$. ∎

3.4 Generalized AGM recovery frameworks

AGM revision and recovery is restricted to the use of a classical semantics as the intended semantics. To generalize it to recovery

[4] This corresponds to a *neutral paraconsistent logic*, cf. [2].

in other intended semantics, we use an operator C_{int} to denote an arbitrary intended semantics and a saturated recovery operator $C_{bck} = R_{AGM}$ to denote both the recovery function and the backup semantics.

AGMg1 $C_{int}(C_{int}(R_{AGM}(T))) = C_{int}(R_{AGM}(T))$;
AGMg2 $C_{int}(R_{AGM}(T)) \subseteq C_{int}(T))$;
AGMg3 If $C_{int}(T) \neq L$, then $C_{int}(T) \subseteq C_{int}(R_{AGM}(T))$;
AGMg4 $C_{int}(R_{AGM}(T)) \neq L$.

Clearly these postulates simply reduce to the standard AGM postulates if $C_{int} = Cn$ and all theories T are classically closed. We will show now that by a suitable choice of intended and backup semantics, a recovery function R_{AGM} expanding a theory T instead of contracting it, also satisfies the **AGMg** postulates.

To motivate this generalization, we present an example where, using a non-classical semantics as an intended semantics, an intuitive acceptable recovery of an inconsistent theory is obtained by *expanding* the theory instead of contracting it. We then show that in general such expansions can be shown to satisfy the generalized AGM postulates.

Example 1 Suppose that you are in an elevator and you push the button for the 13th floor (a). You know that if you can assume the system not[5] to be faulty (not x), then the cage moves up to the 13th floor (b). This rule can be represented by the (logic programming) rule $b \leftarrow a$, not x. Secondly, if the system is faulty (x) and you can assume there is no other person to notice this (not c), then you call the mechanic (d): this rule will be represented by the rule $d \leftarrow x$, not c. Using the stable semantics[6] as your intended semantics indeed you will expect to move up to the 13th floor i.e. you expect b to hold. But suppose that it happens that the cage doesn't move up to the 13th floor, i.e. $\neg b$ holds. In that case, we will expect the system to be faulty and, since we have no evidence that another person already noticed this fact, we will expect that d will hold, too. Consider the theory $T = \{a \leftarrow;\ \neg b \leftarrow;\ b \leftarrow a, \text{not } x;\ \ d \leftarrow x, \text{not } c\ \}$ representing this scenario. Unfortunately it is inconsistent: that is $C_{stable}(T) = L$ i.e. it does not have a stable model. But every strict subtheory of T as well as the supertheory $T' = T \cup \{x \leftarrow\}$ does have a stable model. Clearly, only the latter corresponds to our expectations: in the unique stable model of T', both x and d are true. So here, taking the *expansion* $T' = R(T) = T \cup \{x \leftarrow\}$ would recover the theory from inconsistency in an intuitively acceptable way.

We will now analyse this example in a more general setting. Let us take as the class of theories $\mathcal{T}$ the class of all (classical consistent) logic programs with explicit negation over some (finite) Herbrand Base B. That is, every such program T is a consistent subset of the set $L = Rules(B)$ consisting of all normal rules r that can be formed by taking atoms from B (cf. [7]). As the intended semantics we use the *skeptical stable semantics* represented by the consequence operator $C_{int} = C^{stable}$. The skeptical stable consequences $C^{stable}(T)$ of T consist of all rules $r \in Rules(B)$ that are true in every stable model of T. The "classical consequence" operator Cn applied to a program T is defined as: $Cn(T) = \{r \in Rules(B) \mid T \models r\}$.

The following proposition is based on some well-known properties of the stable semantics (cf. [6]) for propositional normal logic programs:

[5] The negation operator we use here is the negation by default operator.
[6] suitably extended to logic programs with both default and explicit negation

Proposition 3 *For every normal logic program* T,

(i) $C^{stable}(C^{stable}(T)) = C^{stable}(T)$,
(ii) $T \subseteq T' \subseteq Cn(T)$ *implies* $C^{stable}(T') \subseteq C^{stable}(T)$,
(iii) $C^{stable}(Cn(T)) \neq L$.

Now, as a recovery function, let us use the following function R: $R(T) = T$, whenever $C^{stable}(T) \neq L$; otherwise $R(T) = Cn(T)$[7]. Using the proposition stated above, it is easy to see that the generalised AGM postulates AGMg1 - AGMg4 are satisfied. Hence, since $R(T)$ *expands* the current theory T if T needs to be recovered, this shows that recovery by expansion can also satisfy the (generalised) AGM-postulates.

It is not difficult to come up with more complex examples where a nonmonotonic intended semantics is used for theories that even might be classically inconsistent. In such cases, we might prefer to use an iterative approach as mentioned in the introduction, where a classical semantics is used as both an intended semantics (to restore classical consistency) and as a backup semantics in the (second) recovery process w.r.t. the nonmonotonic semantics. In such cases, we can show that a mixed recovery (cf. [4]), where some part of the theory has to be removed before expansion is applied, is a most suitable recovery method.

4 CONCLUSIONS

We have shown that the AGM framework for revision can be easily transformed into a recovery framework. This framework can be easily embedded into the general framework for recovery. Adapting the AGM postulates to recovery of base theories shows that the specialization of the general framework for recovery to classical semantics as the intended semantics can be shown to satisfy these AGM postulates. Finally, we have shown that the AGM postulates can be easily generalized to recovery of non-classical theories and we have shown that in some cases also recovery by expansion satisfies the AGM postulates.

ACKNOWLEDGEMENTS

We would like to thank the anonymous referees for their useful comments.

REFERENCES

[1] Alchourrón, C. E., P. Gärdenfors and D. Makinson, On the Logic of Theory Change: Partial Meet Contraction and Revision Functions, *Journal of Symbolic Logic*, **50**, 510–530.
[2] J. Beziau, The future of paraconsistent logic, in: A. Karpenko (ed.), *Logical Studies*, 2, Moscow, 1998.
[3] P. Gärdenfors, *Belief Revision*, Cambridge University Press, 1992.
[4] K. Inoue, C. Sakama, Abductive Framework for Nonmonotonic Theory Change. *Proceedings IJCAI'95*, 1995.
[5] 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.
[6] V. Marek and M. Truszczyński, *Nonmonotonic Logic*, Springer Verlag, Heidelberg, 1993.
[7] C. Witteveen and W. van der Hoek, Recovery of (Non)monotonic Theories, *Artificial Intelligence*, **106**, 1, 139-159, 1998.
[8] C. Witteveen and W. van der Hoek, 'How to recover from (non)monotonic inconsistencies. To appear in a CSLI volume, special issue on ITALLC'98, 2000.

[7] This corresponds to taking Cn as the backup semantics.