AGM Postulates in Arbitrary Logics: Initial Results and Applications

Giorgos Flouris, Dimitris Plexousakis, Grigoris Antoniou
Institute of Computer Science
Foundation for Research and Technology - Hellas
Science and Technology Park of Crete P.O. Box 1385
GR 711 10 Heraklion, Crete, Greece
Tel: +30 2810 391600
Fax: +30 2810 391601
{fgeo, dp, antoniou}@ics.forth.gr

TR-336, April 2004,
ICS-FORTH

Abstract. The problem of belief change refers to the updating of a Knowledge Base (KB) in the face of new, possibly contradictory information. The AGM theory, introduced in [1], is the dominating paradigm in the area of belief change. Unfortunately, the assumptions made by the authors of [1] in the formulation of their theory restrict its use to a specific class of logics. In this report, we investigate the possibility of extending their framework in a wider class of formalisms by examining the applicability of the AGM postulates in logics originally excluded from the AGM model. We conclude that in the wider class that we consider there are logics which do not admit AGM-compliant operators. As a case study, we investigate the applicability of the AGM theory to Description Logics (DLs). Furthermore, we use our results to shed light on our inability to develop AGM-compliant operators for belief bases.

Introduction

The problem of *belief change* is concerned with the updating of an agent's beliefs when new information becomes known. There are several reasons why an agent should change his beliefs: mistakes may have occurred during the input, or some new information may have become available, or the world represented by the agent's KB may have changed. In all such cases, the agent's KB should be properly modified to accommodate the new fact.

The need to dynamically change the contents of a KB makes belief change a very important problem in Knowledge Representation (KR), attracting much attention in the literature during the last few decades. The most influential approach was the one presented by Alchourron, Gärdenfors and Makinson (AGM for short) in [1]. In that work, the authors did not try to introduce a specific method for dealing with the problem; instead, they introduced a set of properties (named *AGM postulates*) that any rational method of belief change should satisfy.

The AGM approach was almost universally accepted as the dominating paradigm in the area of belief change. Unfortunately though, the AGM theory is based on certain non-elementary assumptions regarding the underlying KR system, which disallow its application to certain interesting classes of logics. To our knowledge, there has been no attempt to extend the theory beyond its original scope.

This work attempts to fill this gap by relaxing the original AGM assumptions and studying the feasibility of applying the AGM theory to a generic class of arbitrary "logics" with only minimal assumptions as to their properties. We will show that the class of logics compatible with the AGM postulates lies somewhere between the class AGM considered and the wider class that we consider. This will be proven by identifying the properties of a logic compatible with the AGM paradigm (termed *AGM-compliant logic*). We further exploit the initial results of our method by presenting several equivalent characterizations of AGM-compliant logics as well as other related properties of such logics. As a case study we consider the applicability of the AGM theory to DLs, an important class of logics originally excluded from the AGM model. Finally, we attempt to further expand our model to deal with operations on *belief bases*, a related family of belief change operations (see [15]). This report will be kept at a fairly technical level, including all the necessary proofs and background to make it self-contained.

Previous Work

The Problem of Belief Change

As already mentioned, the problem of belief change deals with the updating of a KB in the face of new information. The problem becomes important (and difficult) when the new information is contradictory with the currently held beliefs. In such cases, typically some of the old beliefs are dropped in order to accommodate the new information without introducing a contradiction. This policy is usually referred to as the *Principle of Primacy of New Information* ([6]). Furthermore, the dropped beliefs should be as "few" as possible, under some kind of metric. This policy is referred to as the *Principle of Minimal Change* ([9]), also known as the *Principle of Persistence of Prior Knowledge* ([6], [7]).

An important property of the problem of belief change is that it comes in several flavors. The belief change operation considered depends on three very important parameters: the underlying knowledge representation formalism, the type of belief change operator considered and the method used to perform the belief change, as there are usually several intuitively correct ways to perform a given belief change operation.

The effects of the underlying KR-scheme in the considered belief change operation are obvious. One cannot start considering the properties of a belief change operation unless at least some of the properties of the underlying KR system are fixed.

Under any given KR formalism, several types of operations can be defined. The most important operations that have been proposed in the literature are *expansion*, *revision*, *contraction* ([1]), *update* and *erasure* ([19]). Expansion is the addition of a sentence to a KB, without taking any special provisions for maintaining consistency; revision is similar, with the important difference that the result should be a consistent set of beliefs; contraction is required when one wishes to consistently *remove* a sentence from their beliefs instead of adding one. Update is similar to revision, but it refers to changes made because of dynamic changes in the world, while revision refers to changes made in the KB under a static world. In other words, revision is performed when we have learned something new regarding a static world, while update is performed when we need to accommodate a dynamic change of the world in our KB. Similarly, erasure refers to the removal of a belief from our KB when the world changes dynamically.

Any given belief change operation typically involves some kind of choice which can be made in several ways. For example, imagine the propositional KB {a, a→b} and the revision "¬b", which, obviously, contradicts with the KB. There are several equally rational ways to accommodate this revision. One could choose to remove the proposition "a" from the KB. Another choice would be to remove the proposition "a→b" from the KB. Alternatively, one could ignore the revision and keep the KB as is. Finally, several other options can be imagined, such as replacing a proposition of the KB with another, or dropping both propositions etc. All the above approaches succeed in accommodating the revision in the KB, thus they could all be considered "rational" approaches to the problem. To decide on the best method to perform this revision, some preliminary extra-logical assumptions should be taken into account, by considering some propositions more "important" than others ([3], [11]) or by using a kind of metric that measures the "knowledge loss" incurred by each possible change and using the one that minimizes it ([13], [20]).

AGM Setting

Acknowledging the above facts, Alchourron, Gärdenfors and Makinson in [1] decided to take an indirect approach to the problem. First of all, they considered a class of logical formalisms that satisfy certain desirable properties, such as compactness, disjunctive syllogism and others. This class contains many important logics used for KR, such as First-Order Logic (FOL) and Predicate Calculus (PC). This partially resolves the first problem, because it allows the generalization of the results to one important class of logics. Secondly, they dealt with three belief change operations, namely expansion, revision and contraction. Each of them was considered separately, due to their different semantics. Thirdly, instead of proposing one single solution to the problem, they chose to present a set of rationality postulates (named *AGM postulates*) that each belief change operator should satisfy. They presented one set of postulates per operation and argued that only the operations that satisfy these postulates can be classified as "rational" belief change operations. Let us see their options one by one.

In the AGM setting, it is assumed that the underlying KR system uses some kind of logical formalism (a logic L) equipped with a *consequence operator* Cn. The Cn operator is a function mapping sets of propositions of L to sets of propositions of L and is assumed to satisfy the *Tarskian axioms*, which are:

1. $Cn(Cn(A))=Cn(A)$ for all $A\subseteq L$ (iteration)
2. $A\subseteq Cn(A)$ for all $A\subseteq L$ (inclusion)
3. For all $A, B\subseteq L$, $A\subseteq B$ implies $Cn(A)\subseteq Cn(B)$ (monotony)

Furthermore, it was assumed that the logic contains the usual connectives (such as $\wedge$, $\vee$, $\neg$, $\rightarrow$ etc). The Cn operator was assumed to include classical tautological implication and to satisfy the *Rule of Introduction of Disjunction in the Premises*, which states that $Cn((x\vee y)\wedge z)=Cn((x\wedge z)\vee(y\wedge z))$ for all $x, y, z\in L$. Finally, Cn was assumed to be compact. According to the AGM setting, a KB is any set of propositions K that is closed under logical consequence (i.e. $K=Cn(K)$), also called a *theory*.

AGM considered three types of belief change operations, namely expansion, revision and contraction. In this work we will only deal with contraction which is considered the most basic operation ([1]); dealing with other operations is part of our future work. For this reason, we will only describe contraction. Contraction is a very useful operation used when some information is no longer reliable. For example, when we realize that a measuring instrument is malfunctioning we must contract all knowledge acquired by this instrument, because this information is no longer reliable. When we contract a KB with a sentence x, we inform the KB that we no longer believe that x is true; this does *not* necessarily imply that x is false (or that ¬x is true). Under the AGM setting, a belief change operator is a function mapping a KB K and a proposition x to a new KB K′. Thus, a contraction operator is a function mapping a pair (K, x) to a new KB K′, denoted by K′=K−x.

As already noted, AGM did not introduce a specific belief change operator but a set of properties that such an operator should satisfy to be considered "rational". These properties were formalized in a set of six postulates, termed the *basic AGM postulates for contraction*. These postulates are:

(K−1) K−x=Cn(K−x)	(closure)
(K−2) K−x⊆K	(inclusion)
(K−3) If x∉K then K−x=K	(vacuity)
(K−4) If x∉Cn(∅) then x∉K−x	(success)
(K−5) If Cn({x})=Cn({y}) then K−x=K−y	(preservation)
(K−6) K⊆Cn((K−x)∪{x})	(recovery)

The closure postulate guarantees that the result of the contraction operation will be a theory, because under the AGM setting only theories can be KBs. The inclusion postulate guarantees that the operation of contraction will not add any knowledge previously unknown to the KB; this would be irrational, as the contraction operation is used to remove knowledge from a KB. The postulate of vacuity covers the special case where the contracted expression is not part of our theory, so it is not known to be true; in this case, there is no need to remove anything from our knowledge, so the new knowledge (after the contraction) should be the same as the old one. The success postulate guarantees that the contraction will be "successful", i.e. the contracted expression will no longer be a part of our theory as required by the intuition behind the contraction operation; notice that if the contracted expression is a tautology then it cannot (and need not) be removed from the theory. The preservation postulate guarantees that a contraction operation is syntax-independent, i.e. the result of a contraction operation does not depend on the syntactic formulation of a proposition x, but only on its logical properties. Finally, the recovery postulate constrains a contraction operation to only remove knowledge that is relevant to the contracted expression x; thus if a previously contracted expression is later added to our theory, all previous knowledge should be recovered.

Notice that the postulates of vacuity and recovery are both partial formulations of the Principle of Minimal Change. This principle states that knowledge is valuable; thus when a belief change operation is performed upon a KB, the changes performed upon the KB to accommodate this change should be as few as possible, under some proper metric.

Motivation and Goals

The AGM view that only the operations that satisfy their postulates constitute rational belief change operators was accepted by many researchers, who agreed that the postulates express common intuition regarding the operation of contraction. This is true because the AGM approach is based on simple arguments that reflect common intuition on what the effects of a contraction should be. This intuitive appeal was reflected in most subsequent works on belief change. Many researchers studied the belief change operations they proposed with respect to AGM-compliance ([6], [23]) or studied the connection of existing operations with the AGM theory ([20]). Some equivalent formalizations of the AGM postulates were developed ([3], [11], [13], [20]) while other researchers studied the postulates effects and properties ([2], [9], [10], [21], [22]) or criticized the AGM model ([8], [15], [16], [23]), sometimes providing alternative formalizations ([8], [15], [17], [18]).

The main target of criticism against the AGM postulates was the postulate of recovery. Some works ([8], [17]) state that (K−6) is counter-intuitive. Others ([15]) state that it forces a contraction operator to remove too little information from the KB. However, it is generally acceptable that the recovery postulate cannot be dropped unless replaced by some other constraint, such as filtering ([8]), that would somehow express the Principle of Minimal Change. Another common criticism has to do with the problematic connection of belief base contraction with recovery; we deal with this problem in a later section.

Despite the criticisms, the influence of the AGM postulates in the relevant literature is indisputable. Recognizing this fact, we would like to extend the AGM theory in applications beyond its original scope. Unfortunately, the extension of the AGM theory to arbitrary logics voids all the results originally produced by AGM; not all of them hold in the more general class we consider. More specifically, there are logics in the class we consider in which no AGM-compliant operators exist. We will develop results allowing us to check whether the AGM theory makes sense for certain classes of logics; if it does, then it would be rational to search for an AGM-compliant operator to perform belief change in such logics; if it does not, then we should find alternative rationality postulates. Examples of important classes of logics originally excluded from the AGM model, is DLs (see [4]) and equational logic (see [5]).

We will attempt to keep this study at a very abstract level. There will be only minimal assumptions regarding the logic under question. Moreover, we will not assume any structure such as operators and complex expressions in a logic. As a result, our theory will be applicable in a very wide class of logics.

Terminology and Setting

In this work, a *logic* is a pair <L, Cn> consisting of a set (L) and a function (Cn). The set L, called the *set of expressions of the logic* or the *language of the logic*, determines the allowable expressions of the logic. The use of the set L abstracts away from the use of any operators, such as ¬, ∨, ∧, → etc. For example, in standard PC, it holds that: a, b, ¬a, a∧b, a∨¬b, a→b∈L. In fact, L (in PC) contains all the well-formed formulas of PC. The function Cn, called the *consequence operator*, maps sets of expressions to sets of expressions. So, the Cn function is formally defined as: Cn:P(L)→P(L), where P(L) is the powerset of L (2^L). We will assume that the Cn operator satisfies the Tarskian axioms for a reasonable consequence operator, presented in the previous section, namely iteration, inclusion and monotony. For a set A, Cn(A) is assumed to contain all the expressions implied by A. For example, in PC, the consequence operator is determined by the PC semantics; so, the following hold (among other things): a∨b, a∨¬b, a∈Cn({a}), but a∧b∉Cn({a}).

The above assumptions allow any set to be a "logic", provided that it has been equipped with a proper consequence operator; thus they allow a great variety of "logics" to be considered. Notice that we do *not* assume the existence of any operators in the logic. If they exist, they are assumed to be an internal part of the logic; for example the connection between the expressions "a", "b" and "a∨b" in PC will be only implicit and reflected in the Cn operator. If the usual semantics hold, we would expect (among other things) that $Cn(\{a\vee b\})\subseteq Cn(\{a\})$, $a\vee b\in Cn(\{b\})$ etc.

Since operators are not taken into account in our setting, the only way to "connect" expressions of the logic is by grouping them into sets of expressions. This type of "connective" will be heavily used, as in most cases we will develop results that deal with sets of expressions, instead of single expressions. Obviously, this is a more general setting, because any expression $x\in L$ can be equivalently viewed as the singular set $\{x\}\subseteq L$.

For two sets A, $B\subseteq L$, we will say that A *implies* B iff $B\subseteq Cn(A)$. We denote this fact by $A\models B$. It can be easily proven that $A\models B$ iff $Cn(B)\subseteq Cn(A)$. It is easy to see that $\models$ is a relation; when considered as such, $\models$ is a partial order upon P(L). It can be proven (see [8]) that if the Cn operator satisfies the Tarskian axioms then the relation $\models$ defined as above satisfies the following for any A, B, $C\subseteq L$:

1. If $A\subseteq B$ then $B\models A$ (reflexivity)
2. If $B\models A$ and $B\cup A\models C$ then $B\models C$ (transitivity)
3. If $B\models A$ then $B\cup C\models A$ (weakening)

In fact, the following can be proven as well (see [8]): given a partial order $\models$ that satisfies reflexivity, transitivity and weakening, the consequence operator defined as: $Cn(A)=\{x\in L \mid A\models\{x\}\}$ for any given $A\subseteq L$ satisfies iteration, inclusion and monotony (i.e. the Tarskian axioms).

Using the deduction ordering, we can define an equivalence relation, $\cong$, as follows: $A\cong B$ iff $A\models B$ and $B\models A$. Equivalently, $A\cong B$ iff $Cn(A)=Cn(B)$. It can be easily verified that $\cong$ indeed has the properties required for an equivalence relation (reflexive, symmetric and transitive). We will denote by [A] the equivalence class of a set A. Any set of expressions $A\subseteq L$ will be called a *belief*. If A is a singular set of the form $A=\{x\}$ for some $x\in L$, then A will be called an *atomic belief*. If a belief A is closed under logical consequence (i.e. $A=Cn(A)$), then A will be called a *theory*.

In the AGM setting, only atomic beliefs can be contracted from a theory; we relax this assumption, by allowing any belief to be contracted from a theory. In our setting, a contraction operator is a function mapping a pair (K, A), where K is a theory and A is a belief, to a theory K′ denoted by $K'=K-A$. It is easy to reformulate the AGM postulates to fit our more general terminology:

(K−1) $K-A=Cn(K-A)$ (closure)
(K−2) $K-A\subseteq K$ (inclusion)
(K−3) If $A\nsubseteq K$ then $K-A=K$ (vacuity)
(K−4) If $A\nsubseteq Cn(\emptyset)$ then $A\nsubseteq K-A$ (success)
(K−5) If $Cn(A)=Cn(B)$ then $K-A=K-B$ (preservation)
(K−6) $K\subseteq Cn((K-A)\cup A)$ (recovery)

Intuition for Decomposition

General Thoughts

The introduction of the AGM postulates in [1] was accompanied by a proof that every logic can accommodate several operators that satisfy them. Of course, this is true in the class of logics AGM considered, in which several nice properties hold.

Unfortunately, once these properties are dropped, the above result no longer holds. In effect, in the more generic framework that we consider, there are logics that do not admit any AGM-compliant operator. In this work, our main goal is to identify the conditions under which an AGM-compliant operator can be defined in a given logic. As already mentioned, we will restrict ourselves in the study of the contraction postulates. So, let us define the logics that contain an operator that complies with the AGM postulates for contraction:

Definition 1 A logic <L, Cn> is called *AGM-compliant with respect to the basic postulates for contraction* (or simply *AGM-compliant*) iff there exists a contraction function '−' that satisfies the basic AGM postulates for contraction (K−1)-(K−6).

As the following lemma shows, there always exists a contraction operator that satisfies postulates (K−1)-(K−5):

Lemma 1 In any logic <L, Cn> there exists a function that satisfies (K−1)-(K−5).

Proof

Assume the trivial function defined as:

$A-B=A$ iff $B\not\subseteq A$

$A-B=Cn(\varnothing)$ iff $B\subseteq A$

It is easy to show that this function satisfies (K−1)-(K−5) in any logic.

Operators that satisfy (K−1)-(K−5) are called *withdrawal operations*. Thus lemma 1 implies that a withdrawal operation can be defined in any logic. However, notice that the function used in the proof of the lemma is not really a "rational" contraction operator, because it causes a complete loss of previous knowledge when $B\subseteq A$. This property is not accordant with the Principle of Minimal Change; not surprisingly, this operator does not satisfy the postulate of recovery.

Once the recovery postulate is added to our list of desirable postulates the result of lemma 1 is no longer true. Take for example the logic <L, Cn>, where $L=\{a,b\}$, $Cn(\varnothing)=\varnothing$, $Cn(\{b\})=\{b\}$, $Cn(\{a\})=Cn(L)=L$. It is trivial to show that this logic satisfies the Tarskian axioms. Take $A=L$, $B=\{b\}$ and set $C=A-B$, for some operator '−'. Suppose that '−' satisfies the AGM postulates (K−1)-(K−6). Then $C=Cn(C)$ by (K−1). If either $a\in C$ or $b\in C$ then $B\subseteq Cn(C)=C$, an absurdity by (K−4). Thus $C=\varnothing$. But then the postulate of recovery is violated because $Cn(B\cup C)=Cn(B)\subset A$. So, this logic is not AGM-compliant.

Decomposability

Let us examine the situation a bit closer. Assume any logic <L, Cn> and two sets A, $B\subseteq L$. Set $C=A-B$ and suppose that '−' satisfies the basic AGM postulates for contraction. If $Cn(A)=Cn(\varnothing)$ or $Cn(B)=Cn(\varnothing)$ or $Cn(B)\not\subseteq Cn(A)$ then the postulates leave us little choice for the selection of the set C: it must be the case that $Cn(C)=Cn(A)$. This is because of the closure and inclusion postulates in the first case, because of the recovery postulate in the second and because of the vacuity postulate in the third case.

In the more interesting case where $Cn(\varnothing)\subset Cn(B)\subseteq Cn(A)$, the postulates imply three main restrictions: first, the result C should be a subset of A (inclusion postulate); second, the result C should not contain B (success postulate); third, the result C should be such that, after adding back B to it, we should get A again (recovery postulate). More formally, the postulate of inclusion implies that

$Cn(C) \subseteq Cn(A)$, the postulate of success implies that $B \not\subseteq Cn(C)$, while the postulate of recovery implies that $Cn(A) \subseteq Cn(B \cup C)$. If $Cn(C)=Cn(A)$, then by our hypothesis that $Cn(B) \subseteq Cn(A)$, the condition $B \not\subseteq Cn(C)$ is violated. So, the condition $Cn(C) \subseteq Cn(A)$ can be rewritten as $Cn(C) \subset Cn(A)$. With this modification, the requirement $B \not\subseteq Cn(C)$ is redundant in the presence of the conditions $Cn(C) \subset Cn(A)$ and $Cn(A) \subseteq Cn(B \cup C)$, because assuming that $B \subseteq Cn(C)$ we get $Cn(B \cup C)=Cn(C) \subset Cn(A)$, a contradiction. Furthermore, since $Cn(B) \subseteq Cn(A)$ and $Cn(C) \subseteq Cn(A)$ we conclude that $Cn(B \cup C) \subseteq Cn(A)$, so the third requirement can be equivalently rewritten as $Cn(A)=Cn(B \cup C)$. Concluding, we can reduce our requirements to the following two:

- $Cn(C) \subset Cn(A)$
- $Cn(A)=Cn(B \cup C)$

The above two restrictions show that the selected result C must "fill the gap" between A and B. In other words, it must be the case that A can be "decomposed", with respect to B, in two sets B and C, such that both sets contain less knowledge than A when taken separately, but they have the same "informational strength" as A when combined. So, the result $C=A-B$ could be viewed as a kind of "complement" of B with respect to A. The set of complement beliefs of a set with respect to another can be formally defined as follows:

Definition 2 Assume any logic $<L, Cn>$ and two beliefs $A, B \subseteq L$. We define the set of *complement beliefs* of B with respect to A, denoted by $B^{-}(A)$ as follows:

- If $Cn(\emptyset) \subset Cn(B) \subseteq Cn(A)$, then $B^{-}(A)=\{C \subseteq L \mid Cn(C) \subset Cn(A)$ and $Cn(B \cup C)=Cn(A)\}$
- In any other case, $B^{-}(A)=\{C \subseteq L \mid Cn(C)=Cn(A)\}$

Notice that the second bullet has been included for completeness and takes care of all the special cases. The distinctive property that does not allow the logic of the previous example to admit an AGM-compliant operator is the existence of a pair of sets with an empty set of complement beliefs. In the example presented there is no complement belief of $B=\{b\}$ with respect to $A=\{a,b\}$ in the given L ($B^{-}(A)=\emptyset$), thus no operator can satisfy all the basic AGM postulates for contraction. The situation presented is typical in all logics that do not support the AGM postulates for contraction. To prove this fact formally we need the following definition:

Definition 3 Assume any logic $<L, Cn>$ and any set $A \subseteq L$.

- The set $A \subseteq L$ is *decomposable* iff $B^{-}(A) \neq \emptyset$ for every $B \subseteq L$
- The logic $<L, Cn>$ is *decomposable* iff for every $A, B \subseteq L$ $B^{-}(A) \neq \emptyset$ (or equivalently iff all $A \subseteq L$ are decomposable)

Notice that, $B^{-}(A)=\emptyset$ implies that $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$. Indeed, if $Cn(\emptyset) \subset Cn(B)=Cn(A)$ then $Cn(\emptyset) \in B^{-}(A)$ and in any other case $A \in B^{-}(A)$. So we only need to check $B^{-}(A)$ for pairs of sets A, B for which $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$. This is also the principal case in the contraction $A-B$; the case in which we have problems with the recovery postulate. The following theorem proves that the two concepts of definition 1 and definition 3 are equivalent:

Theorem 1 A logic $<L, Cn>$ is AGM-compliant iff it is decomposable.

Proof

(⇒) Assume that the logic is AGM-compliant, so there exists a contraction function '−' that satisfies the basic AGM postulates for contraction.

It suffices to show that for any A, B⊆L for which Cn(∅)⊂Cn(B)⊂Cn(A) it is the case that $B^{-}(A)$≠∅.

Suppose any A, B⊆L such that Cn(∅)⊂Cn(B)⊂Cn(A) and let C=A−B.
By the postulate of closure: C=Cn(C).
By the postulate of inclusion: C⊆Cn(A).

Suppose that C=Cn(A) then Cn(B)⊂Cn(A)=C which is a contradiction by the postulate of success since Cn(∅)⊂Cn(B).

Thus Cn(C)=C⊂Cn(A).
By the postulate of recovery Cn(A)⊆Cn((A−B)∪B)=Cn(C∪B).
Moreover, Cn(B)⊆Cn(A) and Cn(C)⊆Cn(A) thus Cn(B∪C)⊆Cn(A).
Combining the last two relations we get: Cn(A)=Cn(B∪C).

Thus, there exists a C such that Cn(A)=Cn(B∪C) and Cn(C)⊂Cn(A), so C∈$B^{-}(A)$, therefore $B^{-}(A)$≠∅.

(⇐) Now assume that the logic is decomposable. We define the function '−' such that A−B=Cn(C) for some C∈$B^{-}(A)$. This function is well-defined because $B^{-}(A)$≠∅. We assume that this function depends only on A and Cn(B). In other words, suppose two B_1, B_2 such that Cn(B_1)=Cn(B_2). It is obvious that $B_1^{-}(A)=B_2^{-}(A)$, yet the function '−' may select different (even non-equivalent) beliefs in the two cases. We exclude this case by requiring that, in the above example, A−B_1=A−B_2.
Now, we can prove that the function '−' as defined satisfies the basic AGM postulates for contraction.
For the postulate of closure (K−1) this is obvious.
For (K−2), notice that all X∈$B^{-}(A)$ have the property that Cn(X)⊆Cn(A) by definition, so the postulate of inclusion holds.
For the postulate of vacuity (K−3), the result follows from the definition of $B^{-}(A)$.
For the postulate of success (K−4), we take the different cases:

- If Cn(∅)⊂Cn(B)⊆Cn(A), then assume that B⊆A−B. Then Cn(A)=Cn((A−B)∪B)=Cn(A−B)⊂Cn(A), a contradiction.
- In any other case, if B≠Cn(∅) then A−B=Cn(A)⊉B.

The postulate of preservation (K−5) follows from the definition of '−' function; for any equivalent B_1, B_2 it holds that A−B_1=A−B_2.
For the postulate of recovery (K−6), we will again take the different cases:

- If Cn(∅)⊂Cn(B)⊆Cn(A), then A⊆Cn(A)=Cn((A−B)∪B) by the definition of $B^{-}(A)$.
- In any other case Cn((A−B)∪B)=Cn(Cn(A)∪B)⊇Cn(Cn(A))⊇A.

We can also get the following corollary:
Corollary 1 A logic <L, Cn> is AGM-compliant iff for all A, B⊆L with Cn(∅)⊂Cn(B)⊂Cn(A) there exists a C⊆L such that Cn(B∪C)=Cn(A) and Cn(C)⊂Cn(A). Equivalently, the logic <L, Cn> is AGM-compliant iff for all A, B⊆L with Cn(∅)⊂Cn(B)⊂Cn(A) it holds that $B^{-}(A)$≠∅.

Notice that if we close any $C \in B^{-}(A)$ under consequence, then the set Cn(C) is a candidate result for the operator A−B. Furthermore, these are the only possible results for A−B, if '−' satisfies the AGM postulates. If this set is empty, then no AGM-compliant contraction function can be defined. In the standard PC case for example, $B^{-}(A)$ is the set being formed by the results of all partial meet contraction functions that can be defined for the operation A−B.

Furthermore, the above result shows that in order for an AGM-compliant operator to exist in a given logic <L, Cn>, all subsets of any set $A \subseteq L$ should be "paired" with at least one other subset of A which "complements" them in A (i.e. their union is equivalent to A). From the point of view of the base, the above results show that any belief A in an AGM-compliant logic can be broken down in "smaller" beliefs with respect to any of its subsets; i.e. for each subset of A there must exist at least another one that "pairs" it in forming a decomposition of A.

Cuts

We can formulate an equivalent characterization of decomposable (AGM-compliant) logics, using "cuts". A *cut* of a belief A is a family S of beliefs that are properly implied by A with the property that any belief implied by A either implies or is implied by a member of the family S. More formally:

Definition 4 Assume any logic <L, Cn>, a belief $A \subseteq L$ and a family S of beliefs ($S \subseteq P(L)$) such that:

- For all $X \in S$, $Cn(X) \subset Cn(A)$
- For all $Y \subseteq L$ such that $Cn(Y) \subset Cn(A)$ there is a $X \in S$ such that $Cn(Y) \subseteq Cn(X)$ or $Cn(X) \subseteq Cn(Y)$

Then S is called a *cut* of A.

Cuts are important structures. Any set B implied by A ($B \subseteq Cn(A)$) either implies ($X \subseteq Cn(B)$) or is implied ($B \subseteq Cn(X)$) by a set X in a cut S. Thus, a cut S in a sense "divides" the beliefs implied by A in two. The importance of cuts in our theory stems from their close connection with AGM-compliance. Take any set $A \subseteq L$, a cut S of A and a set B that is implied by all the sets in the cut S. If we assume that the operation C=A−B satisfies the AGM postulates, then by the inclusion postulate we conclude that $Cn(C) \subseteq Cn(A)$. But then C will either imply or be implied by a set $X \in S$. If C implies X ($X \subseteq Cn(C)$), then $B \subseteq Cn(X) \subseteq Cn(C)$, so the result does not satisfy success. If C is implied by X ($C \subseteq Cn(X)$), then $B \subseteq Cn(X)$, $C \subseteq Cn(X)$ so $Cn(B \cup C) \subseteq Cn(X) \subset Cn(A)$, so recovery is not satisfied. Thus B has no complements with respect to A. Once we deal with some technicalities and limit cases, it turns out that this is another equivalent characterization of AGM-compliant logics:

Theorem 2 Suppose a logic <L, Cn> and a set $A \subseteq L$. Then A is decomposable iff for all cuts S of A it is the case that $Cn(\cap_{X \in S} Cn(X)) = Cn(\emptyset)$.

Proof

($\Rightarrow$) Suppose that A is decomposable.

If $Cn(A)=Cn(\emptyset)$ then there is no cut of A (by the definition of a cut), so the theorem holds trivially.

Assume that $Cn(A) \neq Cn(\emptyset)$ and that for some cut S of A it holds that $Cn(\cap_{X \in S} Cn(X)) = B \neq Cn(\emptyset)$. Obviously it holds that $B=Cn(B) \subseteq Cn(X)$ for all $X \in S$ thus $B=Cn(B) \subset Cn(A)$, so $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$.

Take any C such that $Cn(C)\subset Cn(A)$. By the definition of the cut, there exists a $X\in S$ such that $Cn(C)\subseteq Cn(X)$ or $Cn(X)\subseteq Cn(C)$.
If $Cn(C)\subseteq Cn(X)$ then:

$Cn(B\cup C)\subseteq Cn(Cn(X)\cup Cn(X))=Cn(X)\subset Cn(A)$ by the definition of a cut, so $C\notin B^{-}(A)$.
If $Cn(X)\subseteq Cn(C)$ then:

$Cn(B)\subseteq Cn(X)\subseteq Cn(C)\Rightarrow Cn(B\cup C)=Cn(Cn(B)\cup Cn(C))=Cn(C)\subset Cn(A)$, so $C\notin B^{-}(A)$.
Thus $B^{-}(A)=\varnothing$, so A is not decomposable, which is a contradiction.

($\Leftarrow$) Now suppose that A is not decomposable. Then by definition there exists a set $B\subseteq L$ such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$ and $B^{-}(A)=\varnothing$. We will construct a cut S such that $Cn(\cap_{X\in S}Cn(X))\neq Cn(\varnothing)$.

Take any $C\subseteq L$ such that $Cn(C)\subset Cn(A)$. Then $Cn(B\cup C)\subseteq Cn(Cn(A)\cup Cn(A))=Cn(A)$.
Suppose that $Cn(B\cup C)=Cn(A)$. Then, $C\in B^{-}(A)\Rightarrow B^{-}(A)\neq\varnothing$, a contradiction.

So for any $C\subseteq L$ with $Cn(C)\subset Cn(A)$ it holds that $Cn(B\cup C)\subset Cn(A)$.

Take the family of beliefs: $S=\{B\cup Y \mid Cn(Y)\subset Cn(A)\}$.

For all $X\in S$ $Cn(X)\subset Cn(A)$, by the previous result.

Furthermore, for any $Y\subseteq L$, $Cn(Y)\subset Cn(A)$ implies $B\cup Y\in S$ and $Cn(Y)\subseteq Cn(B\cup Y)$.
Thus S is a cut. Furthermore, for all $X\in S$ it holds that $Cn(X)\supseteq Cn(B)$, therefore:

$Cn(\cap_{X\in S}Cn(X))\supseteq Cn(B)\supset Cn(\varnothing)$, which is a contradiction by our original hypothesis.
Thus the set A is decomposable.

The following corollary is immediate:
Corollary 2 A logic <L, Cn> is AGM-compliant iff for all $A\subseteq L$ and all cuts S of A it is the case that $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$.

Max-cuts

Cuts gives us an alternative method to check logics for decomposability. One important property of cuts is the following: suppose that we find a cut S with the property that $Cn(\cap_{X\in S}Cn(X))\neq Cn(\varnothing)$. If we take another cut S′ which contains "larger" sets than S, then S′ will also have the property that $Cn(\cap_{X\in S'}Cn(X))\neq Cn(\varnothing)$. So, cuts with "larger" sets (beliefs) are more likely to have a non-empty intersection. This fact motivates us to search for the "largest" cut S_{max} and check whether this cut has a non-empty intersection. The result of this intuition is the notion of max-cut:
Definition 5 Assume a logic <L, Cn>, a belief $A\subseteq L$ and a family S of beliefs ($S\subseteq P(L)$) such that:

- For all $X\in S$, $Cn(X)\subset Cn(A)$
- For all $Y\subseteq L$ such that $Cn(Y)\subset Cn(A)$ there is a $X\in S$ such that $Cn(Y)\subseteq Cn(X)$
- For all $X\in S$, $Cn(X)=X$
- For all X, $Y\in S$ $Cn(X)\subseteq Cn(Y)$ implies that X=Y

Then S is called a *max-cut* of A.

The definition implies that a max-cut of a set A contains exactly the maximal proper subsets of A. In infinite logics there is no guarantee that such maximal proper subsets of a given set A exist, so max-cuts do not always exist. However, when a max-cut of a set A exists, then it is unique, as shown in the following proposition:

Proposition 1 Assume a logic <L, Cn> and a belief $A \subseteq L$. If there exists a max-cut of A, then this max-cut is unique.

Proof

Suppose S_1, S_2 two max-cuts of A and a $X \in S_1$.

Since $X \in S_1$, we conclude that $Cn(X) \subset Cn(A)$.

Then, since S_2 is a max-cut, there exists a $Y \in S_2$ such that $Cn(X) \subseteq Cn(Y)$.

Since $Y \in S_2$, we conclude that $Cn(Y) \subset Cn(A)$.

Since S_1 is a max-cut, there exists a $X' \in S_1$ such that $Cn(Y) \subseteq Cn(X')$.

We conclude that $X, X' \in S_1$ and $Cn(X) \subseteq Cn(Y) \subseteq Cn(X')$, thus $X=X'$.

So we have that $Cn(X)=Cn(X')=Cn(Y)$.

Since $X \in S_1$ and S_1 is a max-cut it follows that $X=Cn(X)$.

Since $Y \in S_2$ and S_2 is a max-cut it follows that $Y=Cn(Y)$.

Thus $X=Cn(X)=Cn(Y)=Y$, which implies that for all $X \in S_1$ we have that $X=Y \in S_2$.

Therefore $S_1 \subseteq S_2$.

Using similar arguments we get $S_2 \subseteq S_1$, thus we conclude that $S_1=S_2$, so the max-cut is unique.

The following proposition gives us a partial answer regarding the existence of a max-cut:

Proposition 2 Assume any logic <L, Cn> and any $A \subseteq L$. If $P(Cn(A))/\cong$ is finite, then there exists a max-cut of A.

Proof

If $Cn(A)=Cn(\emptyset)$ then $S=\emptyset$ is a max-cut, so there exists a max-cut.

If $Cn(A) \neq Cn(\emptyset)$, then there is at least one $B \subseteq L$ such that $Cn(B) \subset Cn(A)$ (for example set $B=\emptyset$). Take any B with this property.

We find all sequences of the form $X_1, X_2, \ldots, X_n$ with the following properties:

- For all $i=1,2,\ldots,n$, $X_i=Cn(X_i)$
- For all $i=2,3,\ldots,n$, $X_{i-1} \subset X_i$
- It holds that $X_1=Cn(B)$, $X_n=Cn(A)$

In effect, we have found sequences of theories whose first member is equal to Cn(B), its last member is equal to Cn(A) and every belief in every sequence is strictly larger than the previous one in the sequence.

There always exists such a sequence, for example (Cn(B),Cn(A)) is a valid sequence.

Furthermore, all sequences have at least two members, namely Cn(B) and Cn(A).

The fact that $P(Cn(A))/\cong$ is finite, guarantees that all such sequences will be finite and that there is a finite number of them.

We select the sequence with the maximum number of elements (say m). If there is more than one with m elements we select one arbitrarily. Once again, the finiteness of $P(Cn(A))/\cong$ guarantees the existence of such a sequence.

We denote this sequence by $H_B=(X_1,X_2,\ldots,X_m)$ and set $B_+=X_{m-1}$.

We can immediately conclude that $B_+=Cn(B_+) \subset Cn(A)$.

Furthermore $Cn(B) \subseteq B_+$ for all $B \subseteq L$ such that $Cn(B) \subset Cn(A)$.

Suppose that there exists a $Y \subseteq L$ such that $Cn(B_+) \subset Cn(Y) \subset Cn(A)$.

Then the sequence $H'=(X_1,X_2,\ldots,X_{m-1},Cn(Y),X_m)$ would be a valid sequence with more elements than H_B, a contradiction.

Thus there is no $Y \subseteq L$ such that $Cn(B_+) \subset Cn(Y) \subset Cn(A)$.

We conclude that if there exists a Y such that $Cn(B_+)\subseteq Cn(Y)\subset Cn(A)$ then $Cn(Y)=Cn(B_+)$.

We set $S=\{B_+ \mid B\subseteq L, Cn(B)\subset Cn(A)\}$.
We claim that S is a max-cut.
Indeed, the following have been proven to be true:

- For all $X\in S$, $Cn(X)\subset Cn(A)$ (since $Cn(B_+)\subset Cn(A)$ for any $B\subseteq L$, $Cn(B)\subset Cn(A)$)
- For all $Y\subseteq L$ such that $Cn(Y)\subset Cn(A)$, there is a $X=Y_+\in S$ such that $Cn(Y)\subseteq Cn(Y_+)=Cn(X)$
- For all $X\in S$, $Cn(X)=X$ (since $Cn(B_+)=B_+$ for all $B\subseteq L$, $Cn(B)\subset Cn(A)$)
- Take any X, $Y\in S$ and suppose that $Cn(X)\subseteq Cn(Y)$. Suppose also that $X=B_+$, $Y=C_+$ for some B, $C\subseteq L$ with the property that $Cn(B)\subset Cn(A)$, $Cn(C)\subset Cn(A)$. We get that $X=Cn(X)\subset Cn(A)$, $Y=Cn(Y)\subset Cn(A)$. Since $Cn(B_+)=Cn(X)\subseteq Cn(Y)\subset Cn(A)$ we get (as proven before) that $Cn(B_+)=Cn(Y)\Rightarrow X=Y$.

Thus S is a max-cut and the proof is complete.

<u>*Corollary 3*</u> The following hold:

- Assume any logic <L, Cn> with the property that $P(L)/\cong$ is finite and any belief $A\subseteq L$. Then there exists a max-cut of A.
- Assume any logic <L, Cn> with the property that $L/\cong$ is finite and any belief $A\subseteq L$. Then there exists a max-cut of A.
- Assume any logic <L, Cn> and any belief $A\subseteq L$. If $Cn(A)/\cong$ is finite then there exists a max-cut of A.

The following theorem outlines the importance of max-cuts in the decomposability theory:

<u>*Theorem 3*</u> Assume any logic <L, Cn> and a belief $A\subseteq L$. If there exists a max-cut S of A, then A is decomposable iff $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$.

<u>*Proof*</u>

($\Rightarrow$) It is obvious that a max-cut is also a cut. So, if A is decomposable then $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$ (by theorem 2).

($\Leftarrow$) Now suppose that $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$.
Take any cut S′ of A.

For every $X\in S$ it holds that $Cn(X)\subset Cn(A)$, so there exists a $Y\in S'$ such that $Cn(X)\supseteq Cn(Y)$ or $Cn(Y)\supseteq Cn(X)$.

Suppose that $Cn(Y)\supset Cn(X)$. Then, $Cn(Y)\subset Cn(A)$, so there exists a $X'\in S$ such that $Cn(X')\supseteq Cn(Y)\supset Cn(X)\Rightarrow Cn(X')\supseteq Cn(X)$. By the definition of the max-cut, we conclude that: $X'=X\Rightarrow Cn(X)=Cn(X')$. Thus $Cn(Y)=Cn(X)$, a contradiction.
Thus, for every $X\in S$ there exists a $Y\in S'$ such that $Cn(X)\supseteq Cn(Y)$.
Similarly, for every $Y\in S'$ there exists a $X\in S$ such that $Cn(X)\supseteq Cn(Y)$, because S is a max-cut.
Thus:
$Cn(\varnothing)=Cn(\cap_{X\in S}Cn(X))\supseteq Cn(\cap_{Y\in S'}Cn(Y))$, so:
$Cn(\cap_{Y\in S'}Cn(Y))=Cn(\varnothing)$ for all cuts S of A, thus A is decomposable.

The following corollary is immediate:

Corollary 4 The following are true:

- Assume any logic <L, Cn> and a belief $A \subseteq L$ with the property that $P(Cn(A))/\cong$ is finite. Then A is decomposable iff for the max-cut S of A it holds that: $Cn(\cap_{X \in S} Cn(X)) = Cn(\varnothing)$.
- Assume any logic <L, Cn> such that L is finite. The logic <L, Cn> is AGM-compliant iff for the max-cut S of every $A \subseteq L$ it holds that $Cn(\cap_{X \in S} Cn(X)) = Cn(\varnothing)$.
- Assume any logic <L, Cn> with the property that $P(L)/\cong$ is finite. The logic <L, Cn> is AGM-compliant iff for the max-cut S of every $A \subseteq L$ it holds that $Cn(\cap_{X \in S} Cn(X)) = Cn(\varnothing)$.
- Assume any logic <L, Cn> with the property that $L/\cong$ is finite. The logic <L, Cn> is AGM-compliant iff for the max-cut S of every $A \subseteq L$ it holds that $Cn(\cap_{X \in S} Cn(X)) = Cn(\varnothing)$.

Theorem 3, combined with the other propositions and corollaries proved above, allows us to determine whether a set $A \subseteq L$ is decomposable or not using only one test, namely calculating $Cn(\cap_{X \in S} Cn(X))$ for the max-cut S of A, provided that such a max-cut exists. This property may allow the development of an algorithm to check for decomposability; evaluating this possibility is part of our future work.

Equivalent Logics

Definition of equivalence

Our general framework allows a great number of logics to be defined. Some of them are effectively “the same”, even though not “equal”. For example, if we rename all elements of a logic and change the consequence function accordingly, we get a logic that has the same properties as the original one. For a less trivial example, a logic based on PC with a countable number of atoms and allowing operators $\neg$, $\wedge$ only is effectively “the same” as one that has a countable number of atoms and operators $\neg$, $\vee$ only. To describe these intuitions formally, we will define equivalent logics in a way that fits both our intuition on “equivalence” and our needs in the following sections of this report.

Obviously, the equivalence relation should depend both on the set itself and on the consequence operator. Two equivalent logics must have sets of beliefs (as expressed by the set L) that have the same “structure” (as expressed by the consequence operator Cn). To verify equivalence between two logics, we must find a mapping between the beliefs of these two logics with properties that guarantee that the structure is preserved during the mapping. This requirement can be expressed using the following definition:

Definition 6 Let $<L_1, Cn_1>$, $<L_2, Cn_2>$ two logics and $\models_1$, $\models_2$ the partial orderings implied by Cn_1, Cn_2 respectively. The two logics will be called *equivalent*, denoted by $<L_1, Cn_1> \sim <L_2, Cn_2>$ iff there exists a mapping $f: P(L_1) \rightarrow P(L_2)$ with the following properties:

- For all $A, B \subseteq L_1$, $A \models_1 B \Leftrightarrow f(A) \models_2 f(B)$
- For all $D \subseteq L_2$, there exists a $C \subseteq L_1$ such that $Cn_2(f(C)) = Cn_2(D)$

If the mapping is relevant, we will write $<L_1, Cn_1> \sim_f <L_2, Cn_2>$, to denote that the two logics are equivalent via the mapping f.

In effect, for two logics $<L_1, Cn_1>$, $<L_2, Cn_2>$ to be equivalent, there must exist a mapping between the beliefs of these logics $f:P(L_1)\rightarrow P(L_2)$ with the following properties:

1. The ordering implied by the consequence operators (or, equivalently, the deduction ordering) of each logic should be preserved during the mapping.
2. Each belief in $P(L_1)$ should be mapped to a belief in $P(L_2)$.
3. Each belief in $P(L_2)$ should have a counterpart in $P(L_1)$, i.e. a belief in $P(L_1)$ which is mapped to it (or to one of its equivalent beliefs).

It is trivial to see that the requirements of the definition are enough to preserve the whole structure of a logic during the mapping. The following lemma is rather trivial and summarizes the relevant results:

Lemma 2 If $<L_1, Cn_1>\sim_f<L_2, Cn_2>$, then for all A, $B\subseteq L_1$ it holds that:

1. $A\vDash_1 B\Leftrightarrow f(A)\vDash_2 f(B)$
2. $A\nvDash_1 B\Leftrightarrow f(A)\nvDash_2 f(B)$
3. $A\cong_1 B\Leftrightarrow f(A)\cong_2 f(B)$
4. $A\ncong_1 B\Leftrightarrow f(A)\ncong_2 f(B)$
5. $Cn_1(A)\subseteq Cn_1(B)\Leftrightarrow Cn_2(f(A))\subseteq Cn_2(f(B))$
6. $Cn_1(A)\nsubseteq Cn_1(B)\Leftrightarrow Cn_2(f(A))\nsubseteq Cn_2(f(B))$
7. $Cn_1(A)=Cn_1(B)\Leftrightarrow Cn_2(f(A))=Cn_2(f(B))$
8. $Cn_1(A)\neq Cn_1(B)\Leftrightarrow Cn_2(f(A))\neq Cn_2(f(B))$
9. $Cn_1(A)\subset Cn_1(B)\Leftrightarrow Cn_2(f(A))\subset Cn_2(f(B))$
10. $Cn_1(A)\not\subset Cn_1(B)\Leftrightarrow Cn_2(f(A))\not\subset Cn_2(f(B))$
11. $Cn_2(f(\varnothing))=Cn_2(\varnothing)$
12. $Cn_2(f(L_1))=Cn_2(L_2)=L_2$

Proof

The proof is obvious by the definition and is omitted.

Properties of equivalence

In most logics, there are several ways to express the same belief; in other words, there are several beliefs which are equivalent. Stripping a logic from such redundant beliefs and keeping one belief per equivalence class, should result to an equivalent logic, because no expressive power has been removed. The following proposition shows that, in addition to this fact, the only way for two logics to be equivalent is by having the same cardinality of equivalence classes which, in addition, have the same structure:

Proposition 3 Let $<L_1, Cn_1>$, $<L_2, Cn_2>$ two logics. Then $<L_1, Cn_1>\sim<L_2, Cn_2>$ iff there exists a $g:P(L_1)/\cong_1\rightarrow P(L_2)/\cong_2$ such that:

- g is 1-1 and onto
- For all A, $B\subseteq L_1$ it holds that $[A]\vDash_1[B]\Leftrightarrow g([A])\vDash_2 g([B])$

Proof

($\Rightarrow$) Suppose that $<L_1, Cn_1>\sim_f<L_2, Cn_2>$.

Then, let $g:P(L_1)/\cong_1\rightarrow P(L_2)/\cong_2$ such that $g([A])=[f(A)]$ for all $[A]\in P(L_1)/\cong_1$.

Then for A, $B\subseteq L_1$ we have that:

$g([A])=g([B]) \Leftrightarrow [f(A)]=[f(B)] \Leftrightarrow f(A)\cong_2 f(B) \Leftrightarrow A\cong_1 B \Leftrightarrow [A]=[B]$.

Thus g is 1-1.

Let $D\subseteq L_2$. Then, there exists a $C\subseteq L_1$ such that:

$Cn_2(D)=Cn_2(f(C)) \Leftrightarrow D \cong_2 f(C) \Leftrightarrow [D]=[f(C)] \Leftrightarrow [D]=g([C])$.
Thus, for any $[D] \in P(L_2)/\cong_2$ there is a $[C] \in P(L_1)/\cong_1$ such that $g([C])=[D]$, so g is onto.
Finally, let A, $B \subseteq L_1$. Then:

$[A] \vDash_1 [B] \Leftrightarrow A \vDash_1 B \Leftrightarrow f(A) \vDash_2 f(B) \Leftrightarrow [f(A)] \vDash_2 [f(B)] \Leftrightarrow g([A]) \vDash_2 g([B])$.
($\Leftarrow$) We define a function $f:P(L_1) \rightarrow P(L_2)$ with the property that for all $A \subseteq L_1$, $f(A) \cong_2 g([A])$.
Then for $A,B \subseteq L_1$ we get:

$A \vDash_1 B \Leftrightarrow [A] \vDash_1 [B] \Leftrightarrow g([A]) \vDash_2 g([B]) \Leftrightarrow f(A) \vDash_2 f(B)$.
Moreover, let $D \subseteq L_2$. Since g is onto, there is a $[C] \in P(L_1)/\cong_1$ such that:
$g([C])=[D] \Leftrightarrow f(C) \cong_2 D \Leftrightarrow Cn_2(f(C))=Cn_2(D)$.
Thus $<L_1, Cn_1> \sim_f <L_2, Cn_2>$.

Proposition 3 implies that the relation ~ as defined here is a very strong equivalence relation. Two logics are equivalent if (and only if) the sets of their equivalence classes are isomorphic and these classes have the same "structure", as this is determined by the partial ordering implied by the consequence operator. This makes it quite difficult for two logics to be equivalent; despite that, several interesting equivalences do exist. To justify its name, the relation ~ must be reflexive, symmetric and transitive. Indeed:
Proposition 4 The relation ~ is an equivalence relation, i.e. it is reflexive, symmetric and transitive.
Proof
For all logics $<L, Cn>$ it holds that $<L, Cn> \sim_f <L, Cn>$ where f is the identity function: $f(A)=A$ for all $A \subseteq L$, thus ~ is reflexive.
Suppose that for two logics $<L_1, Cn_1>$, $<L_2, Cn_2>$ it holds that $<L_1, Cn_1> \sim <L_2, Cn_2>$. Then, by proposition 3 there is a function $g:P(L_1)/\cong_1 \rightarrow P(L_2)/\cong_2$ that is 1-1, onto and for all A, $B \subseteq L_1$ it holds that $[A] \vDash_1 [B] \Leftrightarrow g([A]) \vDash_2 g([B])$.
For the function $f=g^{-1}$, it holds that $f:P(L_2)/\cong_2 \rightarrow P(L_1)/\cong_1$, f is 1-1, onto and for all A, $B \subseteq L_2$ it holds that $[A] \vDash_2 [B] \Leftrightarrow g(g^{-1}([A])) \vDash_2 g(g^{-1}([B])) \Leftrightarrow g^{-1}([A]) \vDash_2 g^{-1}([B]) \Leftrightarrow f([A]) \vDash_2 f([B])$.
Thus, by proposition 3 again $<L_2, Cn_2> \sim <L_1, Cn_1>$, so ~ is symmetric.
Finally, if for logics $<L_1, Cn_1>$, $<L_2, Cn_2>$, $<L_3, Cn_3>$ it holds that:
$<L_1, Cn_1> \sim_f <L_2, Cn_2>$, $<L_2, Cn_2> \sim_g <L_3, Cn_3>$, then set $h:P(L_1) \rightarrow P(L_3)$, such that $h(A)=g(f(A))$ for all $A \subseteq L_1$. Then for all A, $B \subseteq L_1$ we get:

$A \vDash_1 B \Leftrightarrow f(A) \vDash_2 f(B) \Leftrightarrow g(f(A)) \vDash_3 g(f(B)) \Leftrightarrow h(A) \vDash_3 h(B)$.
Furthermore, take any $D \subseteq L_3$. Then there is a $D' \subseteq L_2$ such that $g(D') \cong_3 D$.
For $D' \subseteq L_2$, there is a $C \subseteq L_1$ such that $f(C) \cong_2 D'$.
But: $f(C) \cong_2 D' \Leftrightarrow g(f(C)) \cong_3 g(D') \Leftrightarrow h(C) \cong_3 D$.
So: $<L_1, Cn_1> \sim_h <L_3, Cn_3>$, thus ~ is transitive.

One of the most important properties of the equivalence relation is that two equivalent logics are so much the same in structure, that even operations like $\cup$ or $\cap$ are preserved during the transition from one to the other. More specifically:
Lemma 3 Let $<L_1, Cn_1> \sim_f <L_2, Cn_2>$ and $A,B \subseteq L_1$. Then:
- $f(A \cup B) \cong_2 f(A) \cup f(B)$
- $f(Cn_1(A) \cap Cn_1(B)) \cong_2 Cn_2(f(A)) \cap Cn_2(f(B))$

Proof

(For $\cup$) It holds that:

$A\cup B\models_1 A$ and $A\cup B\models_1 B$, thus $f(A\cup B)\models_2 f(A)$ and $f(A\cup B)\models_2 f(B)$.

Let $D\subseteq L_2$ such that $D\models_2 f(A)$, $D\models_2 f(B)$.
Then there is a $C\subseteq L_1$ such that $f(C)\cong_2 D$.
Given that $f(C)\cong_2 D$ it also holds that:

$f(C)\models_2 f(A)$ and $f(C)\models_2 f(B)$ thus $C\models_1 A$ and $C\models_1 B$, which implies that:

$C\models_1 A\cup B \Leftrightarrow f(C)\models_2 f(A\cup B) \Leftrightarrow D\models_2 f(A\cup B)$.

Thus, $f(A\cup B)\models_2 f(A)$, $f(A\cup B)\models_2 f(B)$ and for all $D\subseteq L_2$ such that $D\models_2 f(A)$, $D\models_2 f(B)$ it holds that $D\models_2 f(A\cup B)$.
Therefore: $f(A\cup B)\cong_2 f(A)\cup f(B)$.
(For $\cap$) It holds that:

$A\models_1 Cn_1(A)\cap Cn_1(B)$ and $B\models_1 Cn_1(A)\cap Cn_1(B)$ thus:

$f(A)\models_2 f(Cn_1(A)\cap Cn_1(B))$ and $f(B)\models_2 f(Cn_1(A)\cap Cn_1(B))$

Let $D\subseteq L_2$ such that $f(A)\models_2 D$, $f(B)\models_2 D$.
Then there is a $C\subseteq L_1$ such that $f(C)\cong_2 D$.
Given that $f(C)\cong_2 D$ it also holds that:

$f(A)\models_2 f(C)$ and $f(B)\models_2 f(C)$ thus $A\models_1 C$ and $B\models_1 C$, which implies that:

$Cn_1(A)\cap Cn_1(B)\models_1 C \Leftrightarrow f(Cn_1(A)\cap Cn_1(B))\models_2 f(C) \Leftrightarrow f(Cn_1(A)\cap Cn_1(B))\models_2 D$.

Thus, $f(A)\models_2 f(Cn_1(A)\cap Cn_1(B))$, $f(B)\models_2 f(Cn_1(A)\cap Cn_1(B))$ and for all $D\subseteq L_2$ such that $f(A)\models_2 D$, $f(B)\models_2 D$ it holds that $f(Cn_1(A)\cap Cn_1(B))\models_2 D$.
Therefore: $f(Cn_1(A)\cap Cn_1(B))\cong_2 Cn_2(f(A))\cap Cn_2(f(B))$.

The following proposition provides a simple test to prove equivalence between two similar logics:
Proposition 5 Assume two logics $<L_1, Cn_1>$, $<L_2, Cn_2>$ such that:

- $L_1\subseteq L_2$
- For all $A\subseteq L_1\subseteq L_2$, it holds that $Cn_1(A)=Cn_2(A)\cap L_1$
- There is a function $f:L_2\backslash L_1\rightarrow P(L_1)$ such that for all $x\in L_2\backslash L_1$ it holds that $Cn_2(\{x\})=Cn_2(f(x))$

Then, $<L_1, Cn_1>\sim<L_2, Cn_2>$.
Proof
Set $f_0:L_2\rightarrow P(L_1)$ such that:

- $f_0(x)=\{x\}$, $x\in L_1$
- $f_0(x)=f(x)$, $x\in L_2\backslash L_1$

By the definition of f_0 and the hypothesis regarding function f, we conclude that for any $x\in L_2$ it holds that $Cn_2(\{x\})=Cn_2(f_0(x))$.
Set $f_*:P(L_2)\rightarrow P(L_1)$ such that: $f_*(A)=\cup_{x\in A}f_0(x)$.
Then for any $A\subseteq L_2$, it holds that:
$Cn_2(A)=Cn_2(\cup_{x\in A}\{x\})=Cn_2(\cup_{x\in A}Cn_2(\{x\}))=Cn_2(\cup_{x\in A}Cn_2(f_0(x)))=$
$=Cn_2(\cup_{x\in A}f_0(x))=Cn_2(f_*(A))$.

Furthermore, it can be proven that for any A, $B\subseteq L_1$ it holds that $A\models_1 B\Leftrightarrow A\models_2 B$.

Indeed, suppose that $A\models_1 B$. Then $Cn_1(A)\subseteq Cn_1(B)$. Take any $x\in Cn_2(A)$:

- If $x\in L_1$, then $x\in Cn_2(A)\cap L_1=Cn_1(A)\subseteq Cn_1(B)=Cn_2(B)\cap L_1\subseteq Cn_2(B)$. Thus: $x\in Cn_2(B)$.

- If $x \in L_2 \backslash L_1$, then $f_0(x) \subseteq L_1$ and $Cn_2(\{x\})=Cn_2(f_0(x))$. Since $x \in Cn_2(A)$, we conclude that $f_0(x) \subseteq Cn_2(A)$. For any $y \in f_0(x)$, it holds that $y \in Cn_2(A)$ and $y \in L_1$, so by the previous case we conclude that $y \in Cn_2(B)$, thus $f_0(x) \subseteq Cn_2(B)$. This fact, along with the equality $Cn_2(\{x\})=Cn_2(f_0(x))$ implies that $x \in Cn_2(B)$.

The above two cases imply that $Cn_2(A) \subseteq Cn_2(B)$, thus $A \vDash_2 B$.

Now assume that $A \vDash_2 B$. Then:

$Cn_2(A) \subseteq Cn_2(B) \Rightarrow Cn_2(A) \cap L_1 \subseteq Cn_2(B) \cap L_1 \Rightarrow Cn_1(A) \subseteq Cn_1(B) \Rightarrow A \vDash_1 B$.

Thus $A \vDash_1 B \Leftrightarrow A \vDash_2 B$.
Set $g: P(L_1) \rightarrow P(L_2)$ such that $g(A)=A$ for all $A \subseteq L_1$.

Then: $A \vDash_1 B \Leftrightarrow A \vDash_2 B \Leftrightarrow g(A) \vDash_2 g(B)$.
Take a $C \subseteq L_2$. Then $f_*(C) \subseteq L_1$ and $Cn_2(g(f_*(C))=Cn_2(f_*(C))=Cn_2(C)$, so for any $C \subseteq L_2$ there is a $D=f_*(C) \subseteq L_1$ such that $Cn_2(g(D))=Cn_2(C)$.
By the existence of g we conclude that $<L_1, Cn_1> \sim <L_2, Cn_2>$.

Equivalence and decomposability

The above lemma is a nice example of a useful property that the equivalence relation has. However, our main goal is decomposability; for the equivalence relation to be useful for our purposes, it must preserve decomposability. This is true, as the following result shows:

Proposition 6 Let $<L_1, Cn_1>$, $<L_2, Cn_2>$, such that $<L_1, Cn_1> \sim_f <L_2, Cn_2>$. If $A \subseteq L_1$ is decomposable, then $f(A) \subseteq L_2$ is decomposable.

Proof

If $Cn(A)=Cn(\varnothing)$ then A is decomposable and $A \cong_1 \varnothing \Leftrightarrow f(A) \cong_2 f(\varnothing) \Leftrightarrow Cn_2(f(A))=Cn_2(\varnothing)$, thus f(A) is decomposable.
If $Cn(A) \neq Cn(\varnothing)$ then set $A'=f(A)$ and take any $B' \subseteq L_2$ such that:

$Cn_2(\varnothing) \subset Cn_2(B') \subset Cn_2(A')$.
There is a $B \subseteq L_1$ such that $f(B) \cong_2 B'$.

Then: $Cn_1(\varnothing) \subset Cn_1(B) \subset Cn_1(A)$.

A is decomposable, so there is a $C \subseteq L_1$ such that $Cn_1(C) \subset Cn_1(A)$ and $Cn_1(B \cup C)=Cn_1(A)$.
Set $C'=f(C) \subseteq L_2$.

Then $Cn_1(C) \subset Cn_1(A) \Leftrightarrow Cn_2(C') \subset Cn_2(A')$ and
$Cn_1(B \cup C)=Cn_1(A) \Leftrightarrow B \cup C \cong_1 A \Leftrightarrow$
$\Leftrightarrow A'=f(A) \cong_2 f(B \cup C) \cong_2 f(B) \cup f(C)=B' \cup C' \Leftrightarrow$
$\Leftrightarrow Cn_2(B' \cup C')=Cn_2(A')$.
So $C' \in B'^{-}(A')$, so $B'^{-}(A') \neq \varnothing$.
We conclude that f(A) is decomposable.

The above proposition has the following important corollary:

Corollary 5 Assume $<L_1, Cn_1>$, $<L_2, Cn_2>$, such that: $<L_1, Cn_1> \sim_f <L_2, Cn_2>$. Then:

- For all $A \subseteq L_1$ it holds that A: decomposable iff f(A): decomposable
- L_1 is decomposable iff L_2 is decomposable

Proof

For the first property notice that $<L_2, Cn_2> \sim_g <L_1, Cn_1>$ for some function g, so if f(A) is decomposable then $g(f(A)) \cong_1 A$ is decomposable.
The second property follows from the first.

The above results (especially corollary 5) show that the equivalence relation preserves decomposability (thus AGM-compliance). The mapping required in definition 6 is so strong that either all equivalent logics are AGM-compliant or all are not AGM-compliant. This will be very helpful in our subsequent analysis on examples of AGM-compliant and non-AGM-compliant logics, because once a logic has been proven to be AGM-compliant (or not AGM-compliant) we can propagate this result to all its equivalent logics.

Connection with Lattice Theory

Another important property of the equivalence relation is that it makes possible to map every logic to a complete lattice. A lattice is a special type of *poset*. A poset is a set equipped with a *partial order* ≤, i.e. a relation satisfying *reflexivity* ($x \le x$), *antisymmetry* ($x \le y$ and $y \le x$ implies $x=y$) and *transitivity* ($x \le y$ and $y \le z$ implies $x \le z$). A *lattice* is a poset P with the additional property that any finite subset H of P ($H \subseteq P$) has a greatest lower bound (or *infimum*, denoted by infH) and a least upper bound (or *supremum*, denoted by supH). It can be proven that if infH and supH exist, then they are unique. A *complete lattice* is a poset P for which infH and supH exist for any set $H \subseteq P$ (not necessarily finite). We will denote lattices by the pair <P, ≤>. Two lattices $<P_1, \le_1>$, $<P_2, \le_2>$ are called *equivalent* (denoted by $<P_1, \le_1> \sim <P_2, \le_2>$) iff there exists a 1-1 and onto mapping $f:P_1 \to P_2$ such that for any $x,y \in P_1$, $x \le_1 y$ iff $f(x) \le_2 f(y)$. For more detailed information on lattice theory see [12].

The fact that any logic is equipped with a partial ordering relation (⊧) implies a possible connection between logics and lattice theory. This connection can be established by mapping each belief of the logic to an element in the lattice and using ⊧ (or its symmetric) as the partial order ≤ of the lattice. In this section we will formally establish this connection and prove that the concepts of lattice theory can be used in our study of AGM-compliance.

Take any logic <L, Cn> and the equivalence relation ≅ upon beliefs that is implied by Cn. Set $P=P(L)/\cong$. We define the relation ≤ such that for any $[A], [B] \in P$: $[A] \le [B]$ iff $[A] \models [B]$, or equivalently: $[A] \le [B]$ iff $Cn(A) \supseteq B$. It can be proven that <P, ≤> is a complete lattice. Now let <P, ≤> be any complete lattice. We set L=P and for any A, $B \subseteq L$ we set $A \models B$ iff $infA \le infB$. The Cn operator implied by ⊧ can be equivalently defined as follows: for any $A \subseteq L$, $Cn(A)=\{x \in L \mid x \ge infA\}$. It can be shown that the above logic <L, Cn> satisfies the Tarskian axioms.

The following lemmas use the two mappings above to prove the connection between lattices and logics. We define:

- LAT_0={<P, ≤> | <P, ≤> is a complete lattice}, the set of complete lattices.
- $LAT=LAT_0/\sim$, the quotient space of LAT_0 with respect to the equivalence relation on lattices (~).
- LOG_0={<L, Cn> | <L, Cn> is a logic}, the set of all logics that satisfy the Tarskian axioms.
- $LOG=LOG_0/\sim$, the quotient space of LOG_0 with respect to the equivalence relation on logics (~).

Using the above terminology we can prove the following lemmas:

<u>*Lemma 4*</u> There is a 1-1 mapping $f:LOG \to LAT$.

<u>*Proof*</u>

We define: $f:LOG \to LAT$ such that for any class of logics $[<L,Cn>] \in LOG$:
$f([<L,Cn>])=[<P,\leq>]$, where $P=P(L)/\cong$ and for any $[A]$, $[B] \in P$, $[A] \leq [B]$ iff $Cn(A) \supseteq Cn(B)$ (equivalently $[A] \leq [B]$ iff $A \vDash B$).
Initially, we must prove that $f([<L,Cn>])=[<P,\leq>] \in LAT$ for any logic $<L,Cn> \in LOG$.
The relation $\leq$ is a partial order because:

- For any $[A] \in P$, $A \vDash A$, thus $[A] \leq [A]$, so reflexivity holds.
- For any $[A]$, $[B] \in P$, if $[A] \leq [B]$ and $[B] \leq [A]$ then $A \vDash B$ and $B \vDash A$, thus $A \cong B \Leftrightarrow [A]=[B]$, so antisymmetry holds.
- For any $[A]$, $[B]$, $[C] \in P$, if $[A] \leq [B]$ and $[B] \leq [C]$ then $A \vDash B$ and $B \vDash C$ thus $A \vDash C \Leftrightarrow A \leq C$, so transitivity holds.

Therefore the pair $<P,\leq>$ is a poset.
Assume any set $H \subseteq P$. We set $[C]=[\cup_{[X] \in H} Cn(X)]$.
By set theory, for any $[X] \in H$: $C \supseteq Cn(X) \Rightarrow Cn(C) \supseteq Cn(X) \Rightarrow C \vDash X \Rightarrow [C] \leq [X]$.
So, $[C]$ is a lower bound of H. Let any other lower bound $[D]$.
Then, for any $[X] \in H$ it holds that $[D] \leq [X] \Rightarrow Cn(D) \supseteq Cn(X)$.
Thus $Cn(D) \supseteq \cup_{[X] \in H} Cn(X)=C \Rightarrow D \vDash C \Rightarrow [D] \leq [C]$.
We conclude that $[C]$ is a greatest lower bound of H, thus $[C]=\inf H$.
Using a lemma in [12] we conclude that $<P,\leq>$ is a complete lattice.
Thus $<P,\leq> \in LAT_0 \Rightarrow [<P,\leq>] \in LAT$, so f is well defined.
Suppose now that for $[<L_1,Cn_1>]$, $[<L_2,Cn_2>] \in LOG$ it holds that:
$f([<L_1,Cn_1>])=f([<L_2,Cn_2>])$.
Set $[<P_1,\leq_1>]=f([<L_1,Cn_1>])$, $[<P_2,\leq_2>]=f([<L_2,Cn_2>])$.
By: $f([<L_1,Cn_1>])=f([<L_2,Cn_2>])$ we conclude that $<P_1,\leq_1> \sim <P_2,\leq_2>$.
So there is a mapping $g:P_1 \to P_2$ that is 1-1 and onto and has the property that for all $[A]$, $[B] \in P_1=P(L_1)/\cong_1$: $[A] \leq_1 [B]$ iff $g([A]) \leq_2 g([B])$.
Set $h:P(L_1)/\cong_1 \to P(L_2)/\cong_2$ such that $h([A])=g([A])$ for all $[A] \in P(L_1)/\cong_1$.
Then h is 1-1, onto and for all $[A]$, $[B] \in P(L_1)/\cong_1$ it holds that:
$[A] \vDash_1 [B] \Leftrightarrow [A] \leq_1 [B] \Leftrightarrow g([A]) \leq_2 g([B]) \Leftrightarrow g([A]) \vDash_2 g([B]) \Leftrightarrow h([A]) \vDash_2 h([B])$.
Therefore, by proposition 3, $<L_1,Cn_1> \sim <L_2,Cn_2> \Rightarrow [<L_1,Cn_1>]=[<L_2,Cn_2>]$.
We conclude that $f:LOG \to LAT$ is 1-1.

Lemma 5 There is a 1-1 mapping $f:LAT \to LOG$.
Proof
We define a function $f:LAT \to LOG$ such that for any class of complete lattices $[<P,\leq>] \in LAT$:
$f([<P,\leq>])=[<L,Cn>]$, where $L=P$ and for any $A \subseteq L=P$, $Cn(A)=\{x \in L=P \mid x \geq \inf A\}$. $<P, \leq>$ is a complete lattice, so $\inf A$ exists for all $A \subseteq P$.
Initially, we must prove that $f([<P,\leq>])=[<L,Cn>] \in LOG$ for any complete lattice $<P,\leq> \in LAT_0$.
To do that, we must prove that iteration, inclusion and monotony hold for Cn.
Indeed:

- Take any $A \subseteq L=P$. It holds that $\inf A \leq \inf A$ thus $\inf A \in Cn(A)$. Furthermore for all $x \in Cn(A)$, $x \geq \inf A$. The above two facts imply that: $\inf(Cn(A))=\inf A$, thus: $Cn(Cn(A))=\{x \in L=P \mid x \geq \inf(Cn(A))\}=\{x \in L=P \mid x \geq \inf A\}=Cn(A)$, so iteration holds.

- Take any $A \subseteq L=P$ and any $x \in A$. By the definition of infimum, it follows that $x \geq \inf A$, thus $x \in Cn(A) \Rightarrow A \subseteq Cn(A)$, so inclusion holds.
- Take any $A \subseteq B \subseteq L=P$. $A \subseteq B$ obviously implies $\inf A \geq \inf B$. Thus: $Cn(A)=\{x \in L=P \mid x \geq \inf A\} \subseteq \{x \in L=P \mid x \geq \inf B\}=Cn(B)$, so monotony holds.

We conclude that $<L,Cn> \in LOG_0 \Rightarrow [<L,Cn>] \in LOG$, thus f is well defined.
We also need to prove that f is 1-1.
Indeed, suppose that for $[<P_1,\leq_1>], [<P_2,\leq_2>] \in LAT$ it holds that:
$f([<P_1,\leq_1>])=f([<P_2,\leq_2>])$.
Set $[<L_1,Cn_1>]=f([<P_1,\leq_1>])$, $[<L_2,Cn_2>]=f([<P_2,\leq_2>])$.
By $f([<P_1,\leq_1>])=f([<P_2,\leq_2>])$ we conclude that $<L_1,Cn_1>\sim<L_2,Cn_2>$.
So there is a mapping $g:P(L_1)\rightarrow P(L_2)$ with the properties that:

- For all $A, B \subseteq L_1$: $A \vDash_1 B$ iff $g(A) \vDash_2 g(B)$
- For all $D \subseteq L_2$ there exists a $C \subseteq L_1$ such that $g(C) \cong_2 D$

Initially we notice that $P_1=L_1$, $P_2=L_2$.

Secondly, it is trivial to prove that for $A,B \subseteq L_1$, $A \vDash_1 B \Leftrightarrow \inf_1 A \leq_1 \inf_1 B$.
Similarly, for all $A,B \subseteq L_1$, $A \cong_1 B \Leftrightarrow \inf_1 A=\inf_1 B$.

Finally, for all $A,B \subseteq L_2$, $A \vDash_2 B \Leftrightarrow \inf_2 A \leq_2 \inf_2 B$ and $A \cong_2 B \Leftrightarrow \inf_2 A=\inf_2 B$.
Set $h:P_1\rightarrow P_2$ such that $h(x)=\inf_2 g(\{x\})$ for all $x \in P_1=L_1$.
Then let any $x,y \in P_1$. If $h(x)=h(y) \Leftrightarrow \inf_2 g(\{x\})=\inf_2 g(\{y\}) \Leftrightarrow g(\{x\}) \cong_2 g(\{y\}) \Leftrightarrow$
$\Leftrightarrow \{x\} \cong_1 \{y\} \Leftrightarrow \inf_1\{x\}=\inf_1\{y\} \Leftrightarrow x=y$, so h is 1-1.
Let any $y \in P_2$. Set $D=\{y\} \subseteq L_2$. Then there exists a $C \subseteq L_1$ such that $g(C) \cong_2 D$. Set $\inf_1 C=x \in P_1=L_1$. Then $\inf_1 C=\inf_1\{x\} \Leftrightarrow C \cong_1 \{x\} \Leftrightarrow g(C) \cong_2 g(\{x\})$.
Combining the above relations, we conclude that:
$g(\{x\}) \cong_2 D \Leftrightarrow \inf_2 g(\{x\})=\inf_2 D=\inf_2\{y\}=y \Leftrightarrow h(x)=y$, thus h is onto.
Finally take any $x,y \in P_1=L_1$. Then:

$x \leq_1 y \Leftrightarrow \inf_1\{x\} \leq_1 \inf_1\{y\} \Leftrightarrow \{x\} \vDash_1 \{y\} \Leftrightarrow g(\{x\}) \vDash_2 g(\{y\}) \Leftrightarrow$
$\Leftrightarrow \inf_2 g(\{x\}) \leq_2 \inf_2 g(\{y\}) \Leftrightarrow h(x) \leq_2 h(y)$.
Therefore $<P_1,\leq_1>\sim<P_2,\leq_2> \Rightarrow [<P_1,\leq_1>]=[<P_2,\leq_2>]$.
We conclude that $f:LAT\rightarrow LOG$ is 1-1.

The above lemmas guarantee that there is a 1-1 mapping from LAT to LOG and vice-versa. This has the following corollary:
Corollary 6 The spaces LOG and LAT are isomorphic. In other words, there is a 1-1 and onto mapping from LOG to LAT.
Proof
By lemma 4, there is a 1-1 mapping from LOG to LAT.
By lemma 5, there is a 1-1 mapping from LAT to LOG.
Combining the above facts with the Schroder-Bernstein theorem, which can be found in any basic book on set theory (see [14] for example), we conclude that there is a 1-1 and onto mapping from LOG to LAT.

The above corollary, combined with corollary 5, implies that, as far as decomposability is concerned, we can view every logic as a lattice and vice-versa. Indeed, take any logic that is, say, decomposable. Then all its equivalent logics are decomposable too and they are all mapped to the same class of equivalent lattices; in effect, they are all mapped to the "same" lattice. The same holds if the logic is not decomposable. So, instead of studying the properties of logics with respect to decomposability, we could equivalently study the properties of lattices.

This alternative representation is very useful because it allows the use of the richness of lattice theory to develop results on logics. The concepts and results of lattice theory that have been developed over the years can be used directly in our framework; this ability may allow the development of deeper results regarding AGM-compliant logics. Moreover, lattices provide a nice and clear visualization to explain the intuition behind the theory of AGM-compliance. Using this mapping, we can reformulate all results presented in this report in terms of lattice theory; this reformulation is easy and is omitted.

Description Logics

Description Logics and AGM Postulates

Description Logics is one of the leading formalisms for representing knowledge in the Semantic Web. To satisfy the different needs of the various applications, a great variety of DLs has been defined. Each DL allows the definition of concepts, roles between concepts and relations between concepts and roles. It also allows the definition of instances of concepts and roles. The allowable relations and operations between such elements determine the expressive power and the algorithmic complexity of reasoning in each DL. For a detailed description of DLs, see [4].

One problem that has been generally disregarded in the DL literature is the updating of the Tbox of a DL KB. The Tbox roughly corresponds to the schema of a DL KB. Updating a Tbox is a very important problem; the ability to update it in a rational way allows the simultaneous building of an ontology by different work teams, followed by the merging of the resulting Tboxes. As before, we initially restrict ourselves to the contraction operator, which in this context refers to the removal of a fact (axiom) from the DL Tbox.

The problem of contracting a Tbox follows the same basic intuitions as the general contraction problem. A Tbox consists of a set of facts (axioms) regarding a domain of discourse. When contracting a Tbox K with an expression (axiom) x -or with a set of axioms A- one should check whether x (or A) is a consequence of K and, if so, remove some of the axioms of K so as to prevent x (or A) from being a consequence of the new Tbox K′=K−x (or K′=K−A).

Since the AGM theory is the leading formalism in the belief change area, we would like to apply it to this problem. Unfortunately, we cannot do that directly, due to the assumptions made in the postulates' original formulation. Using our results on decomposability, we can study the possibility of applying the AGM postulates in the DL Tbox updating problem.

Defining a DL

Before applying our theory to this problem, we need to define a DL as a pair of the form <L, Cn>. To do that, we must use some formal definition of what a DL is; unfortunately though, there has not been (up to our knowledge) any such definition in the literature. To provide a formal definition, we must uncover the properties that are shared by all DLs. We notice that DL bases consist of relations that are formed between concepts, roles and their instances. The part of the KB that deals with the concept/role instances is called the *Abox*, while the part that deals with the concepts/roles themselves is called the *Tbox*. The Tbox contains assertions regarding the terminological axioms of the DL KB; in other words, the relations between the concepts and the roles of a KB are stored in a Tbox. The Abox contains assertions regarding the instances of the concepts and roles of the KB; in this case, relations between such instances are stored. Roughly, the Tbox corresponds to the "schema" of the KB and the Abox to the "data" of the KB.

Any DL contains a *namespace*, i.e. a set containing concept names, role names and names for their instances (such as Mother, Father, A, B, has_Mother, has_Parent, Mary, Jim, a, b, has_Mother1, has_Parent1), a set of *operators* (such as ⊓, ⊔, ¬, ∀, ∃, ⊥, ⊤ etc) and a set of *connectives* (such as ⊑, ⊒, ≅, ⊏, ⋣ etc), often called *relations*. The elements of the namespace combine with the operators (and possibly parentheses) in the usual way to form *terms* (such as (A⊓B)⊔(¬C)⊔⊥). Similarly, terms combine with the connectives to form *axioms* (such as D⊓E≅(A⊓B)⊔(¬C)⊔⊥, A⊑B⊔C etc). A DL KB is a set of axioms.

The symbols used for the namespace, the operators and the connectives are assumed disjoint to avoid confusion. The names in the namespace for the various object types are similarly assumed disjoint, allowing us to determine whether any given name corresponds to a concept, role or a concept/role instance. Operators are usually unary or binary, but we allow arbitrary n-ary operators for generality. Operators with no operands (0-ary) are usually referred to as *constants*. In any given DL L we denote its namespace by N_L, its set of operators by O_L and its set of relations by R_L.

Each operator is actually a function; despite that, we will use the standard notation in the literature for forming terms, instead of the more formal prefix one. For example, we will write A⊔B instead of ⊔(A,B) and ∀R.A instead of ∀(R,A). We assume that each operator is accompanied by its own rules for forming terms. For example, the ∀ operator is followed by a role and a concept (∀R.A, for R role and A concept); so if A, B are concepts then the expression ∀A.B is not a term, but the expression A⊓B is. In some cases, it is useful to distinguish between terms referring to concepts (*concept terms*), roles (*role terms*) or instances (*instance terms*).

Similarly, a connective is actually a relation and each has its own rules for forming axioms. For example, for A, B concept terms and R role term the expression A≅R is not an axiom, but the expression A≅B is. Again, we use the standard in the literature infix notation instead of the more formal mathematical one (A≅B instead of (A,B)∈≅). Relations are usually binary, but we allow relations of arbitrary arity to preserve generality.

In this work, we are interested in the problem of Tbox updating, so we will assume that the namespace does not contain any instance names; furthermore, all the operators and connectives that we will consider deal with concepts and/or roles only. This way, a DL KB consists of a Tbox only and no Abox.

As already mentioned, each DL allows a different namespace, a different set of operators and connectives and has its own rules and conventions for forming terms and axioms; all these limitations define a set L of available axioms in the given DL. Furthermore, each DL is accompanied by its semantics, usually given in model-theoretic terms. Such semantics determines which axioms are implied by any given set of axioms; in effect, it determines the consequence operator (Cn) of the logic. It is trivial to see that all DLs satisfy the Tarskian axioms. Thus, the pair <L, Cn> as defined above identifies the given DL and is a logic in our sense, so, in the following, the term DL will refer to a pair <L, Cn>.

Closed Namespace Assumption

The framework described above will be referred to as the *basic DL framework*. However, when it comes to updating a DL Tbox, some more features specific to the problem should be taken into account. In the DL context, it is usually desirable to make contractions of the form: "remove all references of the concept/role A from the Tbox K". Such contractions are used to remove all information regarding a specific concept or role from a KB. Unfortunately, this special type of contraction cannot be accommodated by the basic framework described above.

This feature is closely related to the assumption that the *only* namespace elements known in a DL KB are the ones that are explicitly mentioned in the KB. In other words, a concept (or role) that does not appear in any axiom in the Tbox does not exist, as far as the KB is concerned. This feature is usually desirable when it comes to DL KBs because a DL namespace usually contains several concepts/roles that can be defined, but only a small subset of them will be relevant to any given KB. We will refer to this assumption as the *Closed Namespace Assumption* (or CNA for short). This assumption cannot be accommodated in the basic framework presented above. For example, the axiom $A \sqsubseteq B$ implies that $A \sqsubseteq B \sqcup C$ for any concept C, so $A \sqsubseteq B$ implies the existence of any concept C, contrary to the CNA.

In order to accommodate the above features we will add to our basic DL definition a special symbol termed the *existence assertion operator* and denoted by %. The symbol % is a modal operator that can only be applied to elements of the namespace (e.g. %A, for $A \in N_L$); it denotes the fact that the concept (or role) A exists in the KB, without giving any further information about A. Using this operator we can express the fact that there is a concept/role A in a KB either implicitly (by forming an axiom that contains it) or explicitly (by adding %A to our Tbox). Without the operator %, such a fact can only be denoted implicitly.

The addition of this operator resolves the problems mentioned above with the basic framework. Firstly, the previously impossible contraction can be expressed using %A by the operation $K - \{\%A\}$. Secondly, we can formally express what it means for a concept/role A to "exist" in a KB K: a concept/role A "exists" in a KB K iff $K \models \{\%A\}$, i.e. if K implies its existence.

To address these issues more formally, we need to consider the effects of the addition of the existence assertion operator to a DL <L, Cn>. First of all, we need to enrich the set of allowable axioms (L) with all expressions of the form %A for $A \in N_L$. Secondly, we need to specify the axioms that imply and/or are implied by an expression of the form %A; equivalently, we need to define the semantics of such an operator.

An initial thought regards the objects "used" in an expression. Any axiom "uses" some elements of the namespace, so it should imply their existence. For example, the expression $A \cong B \sqcap C$ "uses" the objects (concepts/roles) A, B, C so it should imply their existence: $\{A \cong B \sqcap C\} \models \{\%A, \%B, \%C\}$. On the other hand, the CNA constrains an axiom (or a Tbox in general) to imply the existence of the objects it "uses" *only*; it should not imply the existence of objects it does not "use". So, for example: $\{A \cong B \sqcap C\} \not\models \{\%D\}$.

Regarding the implications of an expression of the form %A, we notice that such an expression merely implies the existence of an object A. It should not imply any other facts, except from those that are equivalent to the existence of a namespace element A (such as A≅A for example). To formally describe the above facts we need the following definition:

Definition 7 Let any expression X of a DL <L, Cn>. We define $U(X) \subseteq N_L$ to be the set of namespace elements *used* by X. This set is defined inductively as follows:

- If $X \in N_L$ then $U(X)=\{X\}$
- If X is a term of the form $o(X_1,X_2,\ldots,X_n)$ for some n-ary operator $o \in O_L$ and terms $X_1,X_2,\ldots,X_n$, then $U(X)=U(X_1) \cup U(X_2) \cup \ldots \cup U(X_n)$
- If X is an axiom of the form $r(X_1,X_2,\ldots,X_n)$ for some n-ary connective $r \in R_L$ and terms $X_1,X_2,\ldots,X_n$, then $U(X)=U(X_1) \cup U(X_2) \cup \ldots \cup U(X_n)$

Expanding the definition to sets of axioms, for $P=\{x_i \mid i \in I\}$ a set of axioms we set $U(P)=\cup_{i \in I} U(x_i)$.

By convention, if the operator $o \in O$ in the second bullet is 0-ary, we set $U(X)=\varnothing$. Similarly for the empty set of axioms we set $U(\varnothing)=\varnothing$. Using the above definition, we can formally describe how the semantics of a DL are affected by the addition of the % operator; in any DL <L, Cn>, the following hold regarding the $\models$ relation (or equivalently the Cn operator):

(%1) If P is a set of axioms then $P \models \{\%X\}$ iff $X \in U(P)$

(%2) If $N' \subseteq N_L$ and $\cup_{X \in N'}\{\%X\} \models P$ then $P \cong \cup_{X \in U(P)}\{\%X\}$

The above axioms capture the intuition regarding the % operator. The first axiom (%1) enforces any set to imply the existence of the namespace elements it uses only. The second axiom (%2) implies that the only facts that can be derived by a set of existence assertions are the ones that are equivalent to the existence of the namespace elements they use. Such sets of axioms (which are implied by the existence of the namespace elements they use) will usually be referred to as *trivial facts*, because they actually carry no important information; they only imply the existence of their namespace elements. The addition of the existence assertion operator and the CNA to the basic framework, along with their semantics as expressed by (%1), (%2), constitute the *extended DL framework*, which is more expressive that the basic one.

Discussion on the CNA

The introduction of the existence assertion operator and the CNA are based on the same intuition. Whether one uses the basic or the extended framework depends on the application at hand. Under the basic framework, it is assumed that the KB contains information regarding all the namespace elements, even though some elements may not be mentioned at all. For such elements, the KB is assumed to carry zero information. Under the extended framework, the only objects that exist in the KB are the ones that are mentioned somehow in the KB (implicitly or explicitly). The other elements simply do not exist. For this reason, it seems reasonable that the existence assertion operator and the CNA should be kept or dropped together.

The consequences of the different semantics of the two frameworks should not be underestimated; CNA rules out several possible implications of a given Tbox. For example, if A is not used in a Tbox K at all (or, more formally, if $A \notin U(K)$), then expressions like $A \cong A$, $A \sqsubseteq A$ and $A \sqsubseteq A \sqcup B$ are *not* consequences of the Tbox. This might look like an absurdity, but technically it is not. To prove that, take x="A≅A" for example. By (%1), since $A \in U(x)$ we get that $\{x\} \models \{\%A\}$; thus, if we suppose that

K⊧{x}, we get: K⊧{x}⊧{%A}, thus A∈U(K) by (%1), which contradicts our hypothesis.

The above argument gives an additional, formal reason for the CNA and the existence assertion operator to be kept or dropped together. On the one hand, the intuition regarding the CNA cannot be formally defined without the % operator. On the other hand, if we use the existence assertion operator without the CNA, then for any A in the namespace we get that ∅⊧{A≅A}⊧{%A}, which shows that %A is a consequence of any Tbox, for any $A \in N_L$. So, the addition of the % operator without the CNA does not make much sense as %A∈Cn(∅) for any $A \in N_L$. Finally, the following lemma may prove helpful in the following:

Lemma 6 In any DL <L, Cn> under the extended DL framework the following hold:

- For any P⊆L it holds that $Cn(\cup_{X \in U(P)}\{\%X\}) \subseteq Cn(P)$
- For any P, Q⊆L, P⊧Q implies U(P)⊇U(Q)
- For any P, Q⊆L, P≅Q implies U(P)=U(Q)
- For any P⊆L it holds that U(P)=U(Cn(P))

Proof

For the first fact note that by (%1) we get that for any X∈U(P) it holds that P⊧{%X}, thus $P \models \cup_{X \in U(P)}\{\%X\}$. The last relation implies $Cn(\cup_{X \in U(P)}\{\%X\}) \subseteq Cn(P)$.

For the second fact, let any X∈U(Q). Then P⊧Q⊧{%X}, thus, by (%1) we get that X∈U(P). Therefore, U(Q)⊆U(P).

The third fact is implied by the second trivially.

The fourth is implied by the relation P≅Cn(P) and the third fact.

On Updating a DL Schema (Tbox)

Initial Thoughts

As already mentioned, the intuition behind the general belief change problem and the problem of Tbox updating in a DL KB is the same. However, the AGM theory cannot be directly used to study the problem. Firstly, there are no operators on axioms. Axioms in a DL Tbox are of equational nature (e.g. A≅B⊔C); thus, if x is an axiom then the expression ¬x is usually undefined; the same goes for expressions of the form x∧y, x∨y etc, for x, y axioms of a DL Tbox. Secondly, many DLs are not compact. Therefore, the semantics of such a logic are too far from the logics considered in [1], making the AGM framework inapplicable in the DL context.

Since the prerequisites of the AGM theory do not hold, we cannot determine whether there exists a contraction operator that satisfies the AGM postulates in advance. It would be interesting to study whether the DLs considered in the previous section admit an AGM-compliant operator, despite not being part of the original AGM model. This can be done using our theory on AGM-compliance, because a logic <L, Cn> corresponding to a DL satisfies the Tarskian axioms both under the basic and under the extended framework.

DLs with CNA

Since the basic and the extended framework have very different semantics, we will have to split our analysis in DLs with CNA and DLs without CNA. We will first deal with the extended framework (which supports the CNA). It is obvious that this framework enhances the expressiveness of a DL; unfortunately, it can be proven incompatible with the AGM postulates.

To verify this, recall that the mere existence of some namespace elements is insufficient to imply any non-trivial expression (or set of expressions) that contains them. For example, the set {A≅B} (for A, B concepts of the namespace) is not implied by {%A,%B}. If we attempt to contract {%A} from Cn({A≅B}), the result must use no element of the namespace other than B, or else the postulates of success and/or inclusion would be violated. The only expressions that can be formed using B alone are trivial expressions such as B≅B, B⊑B etc, which are all implied by %B. Thus, the result should be Cn({%B}). But then the postulate of recovery is violated because Cn({%A,%B})⊂Cn({A≅B}). The situation presented is typical in all DLs that contain non-trivial expressions. To make this point more formal, we will prove the following theorem:

Theorem 4 Consider a DL <L, Cn> under the extended framework, with the property that there is at least one set P⊆L such that $Cn(P)\supset Cn(\cup_{A\in U(P)}\{\%A\})$, U(P)≠∅ and U(P) is finite. Then <L, Cn> is not AGM-compliant.

Proof

Let $D=\{Q\subseteq L \mid Cn(Q)\supset Cn(\cup_{A\in U(Q)}\{\%A\}), U(Q)\neq\emptyset, U(Q) \text{ is finite}\}$.

D≠∅ by hypothesis.

We select a set P∈D: |U(P)|≤|U(Q)| for all Q∈D.

The fact that U(Q) is finite for every Q∈D, guarantees the existence of such a P.

If there are more than one with this property, we select one arbitrarily.

Set m=|U(P)|>0, $P'=\cup_{A\in U(P)}\{\%A\}\subseteq L$. We will prove that $P'^{-}(P)=\emptyset$.

Initially notice that U(P)≠∅, thus there is at least one $A\in N_L$ such that A∈U(P). We conclude that %A∈Cn(P′), thus Cn(P′)≠Cn(∅).

Furthermore, by the selection of P we conclude that: Cn(P′)⊂Cn(P).

Thus: Cn(∅)⊂Cn(P′)⊂Cn(P).

Furthermore U(P′)=U(P).

Let Q⊆L such that Cn(Q)⊂Cn(P) and Cn(Q∪P′)=Cn(P).

Since Cn(Q)⊂Cn(P), by lemma 6, we get that U(Q)⊆U(P)=U(P′), so:

|U(Q)|≤|U(P)|=m.

There are two different cases:

- If |U(Q)|<m then by the selection of the set P it holds that:
  $Cn(Q)\not\subset Cn(\cup_{A\in U(Q)}\{\%A\})$. By lemma 6 we get: $Cn(Q)\subseteq Cn(\cup_{A\in U(Q)}\{\%A\})$.
  Combining the above relations: $Cn(Q)=Cn(\cup_{A\in U(Q)}\{\%A\})$. Since U(Q)⊆U(P), we get: $Cn(Q)\subseteq Cn(\cup_{A\in U(P)}\{\%A\})=Cn(P')$, thus: Cn(Q∪P′)=Cn(P′)⊂Cn(P), a contradiction.
- If |U(Q)|=m, then we conclude that U(Q)=U(P′)=U(P).
  But then, $Cn(P')=Cn(\cup_{A\in U(P)}\{\%A\})=Cn(\cup_{A\in U(Q)}\{\%A\})\subseteq Cn(Q)$, thus:
  Cn(Q∪P′)=Cn(Q)⊂Cn(P), a contradiction.

We conclude that there is no set Q⊆L such that Cn(Q)⊂Cn(P) and Cn(Q∪P′)=Cn(P).

Thus <L, Cn> is not AGM-compliant.

The above theorem shows that any non-trivial DL under the extended framework is not AGM-compliant. Thus, there can be no interesting DL supporting the CNA that admits an AGM-compliant operator. If we wish to use these features, we need to introduce a new set of rationality postulates to handle contraction.

DLs without CNA

Despite this negative result, we will not give up on our study of DLs. Acknowledging the fact that the introduction of the existence assertion operator and the CNA was mainly due to a technicality, we will drop the % operator and the CNA from the DL framework and study whether such DLs are AGM-compliant. Of course, this limits our expressiveness, as we can no longer express contractions of the form “remove all references of the concept/role A from the Tbox”. Unfortunately, even without the CNA, our study (up to now) has not revealed any important type of DL that is AGM-compliant. On the contrary, we proved some logics of the AL family (see [4] for details) to be non-AGM-compliant.

In the rest of this section we will consider DLs <L, Cn> under the basic DL framework with the following properties:

- There are at least two role names and at least one concept name in the namespace
- The DL contains any (or all) of the 0-ary operators (constants): $\{\bot, \top\}$
- Any (or all) of the operators $\{\neg$ (full or atomic), $\sqcap$, $\sqcup\}$ are allowed (applying to concepts only)
- At least one of the operators $\{\exists$ (full or limited), $\forall$, $\geq n$, $\leq n\}$ is allowed
- Only equality axioms are allowed

Full negation can be applied to any concept term, while atomic negation can be applied to namespace concept elements only. Full existential quantification is of the form $\exists R.A$, where R is a role and A is a concept. In limited existential quantification, the concept A is restricted to be equal to $\top$, so it is of the form $\exists R.\top$ for some role R. To avoid possible confusion, we will use the symbols $\neg_f$ for full negation, $\neg_a$ for atomic negation, $\exists_f$ for full existential quantification and $\exists_l$ for limited existential quantification and drop the usual symbols $\neg$, $\exists$. Moreover, without loss of generality, we will assume that R, S are two role names and A is a concept name contained in N_L. Thus $N_L \supseteq \{R, S, A\}$. Using these assumptions, we can formally define the properties of the above family of DLs:

- $N_L \supseteq \{R, S, A\}$, where R, S are role names and A is a concept name
- $O_L \subseteq \{\bot, \top, \neg_f, \neg_a, \sqcap, \sqcup, \exists_f, \exists_l, \forall, \geq n, \leq n\}$ and $O_L \cap \{\exists_f, \exists_l, \forall, \geq n, \leq n\} \neq \emptyset$
- $R_L = \{\cong\}$

Each of these operators has its own rules for forming terms. For example, $\neg_f$, $\sqcap$ and $\sqcup$ are restricted to apply to concept terms only. Thus, for any expression of the form $X \sqcap Y$ for example, both X and Y are assumed to be concept terms. For further details on the properties of the operators see [4]. These rules exactly define the set L of the allowable expressions of the logic.

The semantics of the operators and the connective are usually given in model-theoretic terms. We define an *interpretation* of a DL as a pair (D^I, I) consisting of a set D^I and a function I, such that $D^I \neq \emptyset$. The function I maps any concept A in the namespace to a set $A^I \subseteq D^I$ and any role R to a set $R^I \subseteq D^I \times D^I$. To expand the function I to more complex concepts and roles (i.e. terms) we assume that I has the following properties:

- $\bot^I = \emptyset$ (*Bottom Concept*)
- $\top^I = D^I$ (*Top Concept*)

- $(\neg_f A)^I = D^I \setminus A^I$, for any concept term A (*Full Negation*)
- $(\neg_a A)^I = D^I \setminus A^I$, for any concept $A \in N_L$ (*Atomic Negation*)
- $(A \sqcap B)^I = A^I \cap B^I$, for any concept terms A, B (*Intersection*)
- $(A \sqcup B)^I = A^I \cup B^I$, for any concept terms A, B (*Union*)
- $(\exists_f R.A)^I = \{x \in D^I: \exists y \in A^I\ (x,y) \in R^I\}$, for any role term R and any concept term A (*Full Existential Quantification*)
- $(\exists_l R.\top)^I = \{x \in D^I: \exists y \in D^I\ (x,y) \in R^I\}$, for any role term R (*Limited Existential Quantification*)
- $(\forall R.A)^I = \{x \in D^I: \forall y \in D^I\ (x,y) \in R^I \text{ implies } y \in A^I\}$, for any role term R and any concept term A (*Universal Quantification*)
- $(\geq nR)^I = \{x \in D^I: |\{y \in D^I: (x,y) \in R^I\}| \geq n\}$, for any role term R and any $n \in \mathbb{N}$ (*At-least Number Restriction*)
- $(\leq nR)^I = \{x \in D^I: |\{y \in D^I: (x,y) \in R^I\}| \leq n\}$, for any role term R and any $n \in \mathbb{N}$ (*At-most Number Restriction*)

For any two terms X, Y, an axiom of the form $X \cong Y$ is *satisfied by an interpretation* (D^I, I) iff $X^I = Y^I$. An axiom is *valid* iff it is satisfied by all interpretations. A set of axioms P is *satisfied by an interpretation* (D^I, I) iff every $x \in P$ is satisfied by the interpretation (D^I, I). A set of axioms is *valid* iff it is satisfied by all interpretations. An axiom x (or a set Q) *is implied by* a set of axioms P iff x (or Q) is satisfied by all interpretations that satisfy P. We denote this fact using the symbol $\models$ (i.e. $P \models \{x\}$, $P \models Q$). We define the consequence operator Cn of the logic as: $Cn(P) = \{x \in L: P \models \{x\}\}$, for any $P \subseteq L$. Obviously, $P \models Q$ iff $Cn(Q) \subseteq Cn(P)$, as usual.

Using the above terminology, we are now ready to prove that the above family of DLs does not contain any decomposable DL. Initially, we will prove the following lemmas:

<u>*Lemma 7*</u> Assume a DL <L, Cn> such that:

- $N_L \supseteq \{R, S, A\}$, where R, S are role names and A is a concept name
- $O_L \subseteq \{\bot, \top, \neg_f, \neg_a, \sqcap, \sqcup, \exists_f, \exists_l, \forall, \geq n, \leq n\}$ and $O_L \cap \{\exists_f, \exists_l, \forall, \geq n, \leq n\} \neq \emptyset$
- $R_L = \{\cong\}$

Set $P = \{R \cong S\}$, $Q = \{x \in L \mid Cn(\{x\}) \subset Cn(P)\}$ and take any $x \in Q$. If x is of the form: x="$T_1 \cong T_2$", where T_1, T_2 are role terms, then $T_1 = T_2$.

<u>*Proof*</u>

Obviously, for any $x \in Q$, $x \in Cn(P)$ thus $Q \subseteq Cn(P)$ or, equivalently, $P \models Q$.
We notice that there are no operators in the logic that result in role terms. Thus T_1, $T_2 \in N_L$, i.e. they are elements of the namespace.
Take the interpretation (D^I, I) such that:
$D^I = \{x_i \mid i \in N_L\}$
For any concept A in the namespace set: $A^I = \{x_A\}$
Set: $R^I = S^I = \{(x_R, x_R), (x_S, x_S)\}$
For any role R′ in the namespace, such that $R' \neq R$ and $R' \neq S$ set: $R'^I = \{(x_{R'}, x_{R'})\}$
Obviously $R^I = S^I$, so $R \cong S$ is satisfied by (D^I, I), so P is satisfied, thus Q is satisfied.
Therefore x="$T_1 \cong T_2$" is satisfied, so $T_1^I = T_2^I$.
It is easy to see that for any $T_1, T_2 \in N_L$ $T_1^I = T_2^I$ implies one of the following:

- $T_1 = R$, $T_2 = S$. In this case x="$R \cong S$", so $Cn(\{x\}) = Cn(P)$, thus $x \notin Q$, a contradiction

- T_1=S, T_2=R. In this case x="S≅R", so Cn({x})=Cn(P), thus x∉Q, a contradiction
- $T_1=T_2$

So, if $T_1 \neq T_2$ then there exists an interpretation (D^I,I) that satisfies P but not x, which is a contradiction because P⊧x. Thus $T_1=T_2$.

<u>Lemma 8</u> Assume a DL <L, Cn> such that:

- $N_L \supseteq \{R, S, A\}$, where R, S are role names and A is a concept name
- $O_L \subseteq \{\bot, \top, \neg_f, \neg_a, \sqcap, \sqcup, \exists_f, \exists_l, \forall, \geq n, \leq n\}$ and $O_L \cap \{\exists_f, \exists_l, \forall, \geq n, \leq n\} \neq \emptyset$
- $R_L=\{\cong\}$

Set P={R≅S}, Q={x∈L | Cn({x})⊂Cn(P)}. Then Cn(Q)⊂Cn(P).

<u>Proof</u>

It is obvious that for any x∈Q, x∈Cn(P), thus Cn(Q)⊆Cn(P). So, it suffices to find an interpretation that satisfies Q but not P.

We define two interpretations as follows:

(D^I,I): $D^I=\{a_1,a_2,b\}$, $A^I=\{a_1,a_2\}$, $R^I=\{(b,a_1)\}$, $S^I=\{(b,a_1)\}$, $X^I=\emptyset$ for $X \in N_L \setminus \{R,S,A\}$

$(D^{I'},I')$: $D^{I'}=\{a_1,a_2,b\}$, $A^{I'}=\{a_1,a_2\}$, $R^{I'}=\{(b,a_1)\}$, $S^{I'}=\{(b,a_2)\}$, $X^{I'}=\emptyset$ for $X \in N_L \setminus \{R,S,A\}$

Notice that the only difference between (D^I,I) and $(D^{I'},I')$ is in the interpretation of the role S.

We can immediately deduce the following facts:

- (D^I,I) satisfies P (because $R^I=S^I$)
- (D^I,I) satisfies Q (because it satisfies P and P⊧Q)
- $(D^{I'},I')$ does not satisfy P (because $R^{I'} \neq S^{I'}$)

We will also prove that $(D^{I'},I')$ satisfies Q. To prove that, we will initially prove that for any concept term X it holds that $a_1 \in X^I$ iff $a_2 \in X^I$.

Indeed, if X contains no operators then the result follows by the definition of (D^I,I).

If X contains n operators, then we will use induction on n.

Suppose that the last (outermost) operator is:

- Bottom Concept (⊥): obviously $a_1,a_2 \notin \bot^I=\emptyset$, so the result holds.
- Universal Concept (⊤): obviously $a_1,a_2 \in \top^I=D^I$, so the result holds.
- Full Negation ($\neg_f$): then $X=\neg_f Y$ for some concept term Y with n−1 operators. Thus $a_1 \in X^I \Leftrightarrow a_1 \notin Y^I \Leftrightarrow a_2 \notin Y^I \Leftrightarrow a_2 \in X^I$.
- Atomic Negation ($\neg_a$): then $X=\neg_a Y$ for some namespace concept Y (which has no operators). Thus $a_1 \in X^I \Leftrightarrow a_1 \notin Y^I \Leftrightarrow a_2 \notin Y^I \Leftrightarrow a_2 \in X^I$.
- Intersection (⊓): then $X=Y_1 \sqcap Y_2$ for some concept terms Y_1, Y_2 with at most n−1 operators each. Thus $a_1 \in X^I \Leftrightarrow a_1 \in Y_1^I$ and $a_1 \in Y_2^I \Leftrightarrow a_2 \in Y_1^I$ and $a_2 \in Y_2^I \Leftrightarrow a_2 \in X^I$.
- Union (⊔): use similar argumentation as in the case with the intersection.
- Full Existential Quantifier ($\exists_f$): then $X=\exists_f R'.Y$ for some role R′ (with no operators) and some concept term Y (with n−1 operators). If R′≠R and R′≠S, then $R'^I=\emptyset$, thus $X^I=\emptyset$ and the result holds. If R′=R and $a_1 \in Y^I$ then $X^I=\{b\}$ and the result holds. Similarly, if R′=R and $a_1 \notin Y^I$ then $X^I=\emptyset$ and the result holds again. The same argumentation is followed if R′=S.
- Limited Existential Quantifier ($\exists_l$): then $X=\exists_l R'.\top$ for some role R′ (with no operators). If R′≠R and R′≠S, then $R'^I=\emptyset$, thus $X^I=\emptyset$ and the result holds. If R′=R or R′=S then $X^I=\{b\}$ and the result holds again.

- Value Restriction ($\forall$): then $X=\forall R'.Y$ for some role R' (with no operators) and some concept term Y (with n−1 operators). If $R' \neq R$ and $R' \neq S$, then $R'^I=\emptyset$, thus $X^I=D^I$ and the result holds. If $R'=R$ and $a_1 \in Y^I$ then $X^I=\{b\}$ and the result holds. Similarly, if $R'=R$ and $a_1 \notin Y^I$ then $X^I=\emptyset$ and the result holds again. The same argumentation is followed if $R'=S$.
- At-least Number Restriction ($\geq$): then $X=(\geq nR')$ for some role R' (with no operators). If n=0 then $X^I=D^I$ and the result holds. If n=1 and $R'=R$ or $R'=S$ then $X^I=\{b\}$ and the result holds again. If n=1 and $R' \neq R$, $R' \neq S$ then $X^I=\emptyset$ and the result holds. If n>1 then $X^I=\emptyset$, regardless of R' and the result holds again.
- At-most Number Restriction ($\leq$): then $X=(\leq nR')$ for some role R' (with no operators). If n=0 and $R' \neq R$, $R' \neq S$ then $X^I=D^I$ and the result holds. If n=0 and $R'=R$ or $R'=S$ then $X^I=\{a_1,a_2\}$ and the result holds again. If n>0 then $X^I=D^I$, regardless of R' and the result holds.

We conclude that $a_1 \in X^I$ iff $a_2 \in X^I$ for any concept term X.
Using similar argumentation we can conclude that the same holds for the interpretation $(D^{I'},I')$, i.e. $a_1 \in X^{I'}$ iff $a_2 \in X^{I'}$ for any concept term X.
Furthermore, we can show that for any concept term X it holds that $X^I=X^{I'}$.
To prove this notice that for any namespace element Y other than S it holds that $Y^I=Y^{I'}$. So if X does not contain S, the result holds trivially.
Suppose that X contains S and has n operators. We will use induction on n.
If n=0, then X=S, but S is not a concept term, so the result holds trivially.
In the general case, suppose that the last (outermost) operator is:

- Bottom Concept ($\bot$): obviously $\bot^I=\bot^{I'}=\emptyset$, so the result holds.
- Universal Concept ($\top$): obviously $\top^I=D^I=D^{I'}=\top^{I'}$, so the result holds.
- Full Negation ($\neg_f$): then $X=\neg_f Y$ for some concept term Y with n−1 operators. Then $X^I=D^I \backslash Y^I=D^{I'} \backslash Y^{I'}=X^{I'}$ by the induction hypothesis.
- Atomic Negation ($\neg_a$): the result is proved using the same argumentation as in full negation.
- Intersection ($\sqcap$): then $X=Y_1 \sqcap Y_2$ for some concept terms Y_1, Y_2 with at most n−1 operators each. Thus $X^I=Y_1^I \cap Y_2^I=Y_1^{I'} \cap Y_2^{I'}=X^{I'}$ by the induction hypothesis.
- Union ($\sqcup$): use similar argumentation as in the case with the intersection.
- Full Existential Quantifier ($\exists_f$): then $X=\exists_f R'.Y$ for some role R' (with no operators) and some concept term Y (with n−1 operators). It holds that $Y^I=Y^{I'}$ by the inductive hypothesis. If $R' \neq S$ then $R'^I=R'^{I'}$ and the result follows trivially. If $R'=S$ then, if $a_1 \in Y^I$ it follows that $a_2 \in Y^I$ thus $Y^I=Y^{I'} \supseteq \{a_1,a_2\}$. Thus $X^I=\{b\}=X^{I'}$. If $a_1 \notin Y^I$ then $a_2 \notin Y^I$ so $Y^I \cap \{a_1,a_2\}=Y^{I'} \cap \{a_1,a_2\}=\emptyset$. Thus $X^I=\emptyset=X^{I'}$. So the result holds.
- Limited Existential Quantifier ($\exists_l$): then $X=\exists_l R'.\top$ for some role R' (with no operators). The result is shown using the same argumentation as with the full existential quantifier.
- Value Restriction ($\forall$): then $X=\forall R'.Y$ for some role R' (with no operators) and some concept term Y (with n−1 operators). It holds that $Y^I=Y^{I'}$ by the inductive hypothesis. If $R' \neq S$, then $R'^I=R'^{I'}$ and the result follows trivially. If $R'=S$ then, if $a_1 \in Y^I$ it follows that $a_2 \in Y^I$, thus $Y^I=Y^{I'} \supseteq \{a_1,a_2\}$. Thus $X^I=\{b\}=X^{I'}$. If $a_1 \notin Y^I$ then $a_2 \notin Y^I$ so $Y^I \cap \{a_1,a_2\}=Y^{I'} \cap \{a_1,a_2\}=\emptyset$. Thus $X^I=\emptyset=X^{I'}$. So the result holds.

- At-least Number Restriction (≥): then X=(≥nR′) for some role R′ (with no operators). For R′=S, it is easy to see that, for n≤1, $X^I=\{b\}=X^{I'}$ and for n>1, $X^I=\varnothing=X^{I'}$. So the result holds.
- At-most Number Restriction (≤): then X=(≤nR′) for some role R′ (with no operators). For R′=S, it is easy to see that, for n=0, $X^I=\varnothing=X^{I'}$ and for n>0, $X^I=D^I=D^{I'}=X^{I'}$. So the result holds.

Now, take any x∈Q. By the definition of the DL, x="$T_1 \cong T_2$" for two terms T_1,T_2.

- If T_1, T_2 are role terms, then by lemma 7, $T_1=T_2$, so x is obviously satisfied by $(D^{I'},I')$.
- If T_1, T_2 are concept terms, then we conclude: $T_1^I=T_1^{I'}$, $T_2^I=T_2^{I'}$ (by the above result) and $T_1^I=T_2^I$ (since x∈Cn(Q)⊆Cn(P) and P is satisfied by (D^I,I)). Thus $T_1^{I'}=T_2^{I'}$, which implies that x is satisfied by $(D^{I'},I')$.

Therefore, x is satisfied by $(D^{I'},I')$, for any x∈Q, thus Q is satisfied by $(D^{I'},I')$.

This means that there is an interpretation that satisfies Q but not P, so Cn(Q)⊂Cn(P).

Notice that in the above lemmas we do *not* assume that O_L contains all the operators listed. Their proof can be repeated for each possible combination of operators in the DL. Using these lemmas we can prove the following:

<u>*Theorem 5*</u> Assume a DL <L, Cn> such that:

- $N_L \supseteq \{R, S, A\}$, where R, S are role names and A is a concept name
- $O_L \subseteq \{\bot, \top, \neg_f, \neg_a, \sqcap, \sqcup, \exists_f, \exists_l, \forall, \geq n, \leq n\}$ and $O_L \cap \{\exists_f, \exists_l, \forall, \geq n, \leq n\} \neq \varnothing$
- $R_L=\{\cong\}$

Then <L, Cn> is not AGM-compliant.

<u>*Proof*</u>

Set P={R≅S}, Q={x∈L | Cn({x})⊂Cn(P)}.

Since $O_L \cap \{\exists_f, \exists_l, \forall, \geq n, \leq n\} \neq \varnothing$, Cn(Q)≠Cn(∅).

Take the family of sets S_{cut}={Q}.

Then, Cn(Q)⊂Cn(P) by lemma 8.

Furthermore, take any set Q′⊆L such that Cn(Q′)⊂Cn(P). For any x∈Q′ it holds that Cn({x})⊆Cn(Q′)⊂Cn(P), thus x∈Q. We conclude that Q′⊆Q⇒Cn(Q′)⊆Cn(Q).

Thus S_{cut} is a cut of P. The intersection of all sets in S_{cut} is, obviously, equal to Cn(Q)≠Cn(∅). Thus, by corollary 2, <L, Cn> is not AGM-compliant.

The problem here lies in the lack of operators for connecting roles with each other. Because of this absence, for two roles R, S, the expression R≅S is equivalent only with itself and its symmetric (S≅R). We exploit this fact by defining Q to contain all the proper implications of R≅S and showing that Q alone forms a cut.

Theorem 5 implies that FL_0, FL^-, AL and all DLs of the family AL[U][E][N][C] (see [4] for the definition of these DLs) are not AGM-compliant, provided that they only allow for equality axioms. It would be an interesting topic of future work to study the effect of allowing inclusion axioms in any of the above DLs. The result presented does not imply anything as far as more (or less) expressive DLs are concerned; it is possible that a more (or less) expressive DL is AGM-compliant. If this is the case, it would be interesting to find the connective(s) and/or operator(s) that guarantee/bar AGM-compliance.

Belief Base Contraction

Belief Base Operations

One of the criticisms that the AGM model had to face was the fact that theories are (in general) infinite structures, thus no reasonable algorithm based entirely on the AGM model could be developed ([15], [16], [17]). Furthermore, some authors ([8], [23]) state that our beliefs regarding a domain stem from a small, finite number of observations, facts, rules, measurements, laws etc regarding the given domain; the rest of our beliefs are simply derived from such facts and should be removed once their logical support is lost.

The above problems motivated the development of *belief base operations*, a class of belief change operations that do not require that the KB is expressed by a theory; in effect, any set can be a KB, called a *belief base*. The consequences of a belief base are still considered part of our knowledge; however, this model distinguishes between *explicit facts* (acquired directly from observations) and *implicit facts* (implied by the observations), because explicit facts are stored in the base while implicit facts are not. In the AGM model, all facts (both explicit and implicit) are part of the KB, so there is no distinction between implicit and explicit facts.

Under this viewpoint, belief change operations affect the stored facts only; the consequences of the belief base are only implicitly affected, due to changes in the explicit facts. More specifically, when contracting a belief base, only elements (beliefs) from the base itself can be removed; once this removal is complete we can recalculate the consequences of the new base (the new implicit facts). Belief change operations on belief bases appeared as a reasonable alternative to the AGM model due to their nice computational properties and intuitive appeal.

Connection with the AGM Theory

In the belief base context, the AGM requirement that contraction should be performed upon a theory and result in a theory is dropped. This seemingly small difference has some severe effects on the operators considered. An initial effect is the fact that the AGM postulates should be slightly modified to deal with belief base contraction. This is relatively easy to do; the main difference in the two cases is the fact that in belief set contraction the result should be a subset of the theory, while in belief base contraction the result should be a subset of the base. The new, modified postulates are:

(B−1) $K-A\subseteq L$	(base closure)
(B−2) $K-A\subseteq K$	(base inclusion)
(B−3) If $A\notin Cn(K)$, then $K-A=K$	(base vacuity)
(B−4) If $A\notin Cn(\varnothing)$, then $A\notin Cn(K-A)$	(base success)
(B−5) If $Cn(A)=Cn(B)$, then $K-A=K-B$	(base preservation)
(B−6) $K\subseteq Cn((K-A)\cup A)$	(base recovery)

The above postulates can also be found in [8]. Notice that under the new postulates, K does not necessarily refer to a theory; any set would do in this context. Similarly, the result K−A could be any set. The other postulates are the same as in belief set contraction. Despite this fact, (B−2) is stronger than (K−2) because it forces the contraction function to remove elements from the base K only (instead of the theory of K, Cn(K)).

Unfortunately, for most logics and belief bases, these postulates do not make much sense due to the base recovery postulate. Take for example the operation $\{a\wedge b\}-\{a\}$ in PC. Due to the postulates (B−2) and (B−4) we have to remove $\{a\wedge b\}$ from our belief base, or else the belief $\{a\}$ will emerge as a consequence of the new base. So it should be the case that $\{a\wedge b\}-\{a\}=\varnothing$; but this violates the base recovery postulate, as can be easily verified. Thus, there can be no AGM-compliant base contraction operator that can handle this case.

The effects of this observation were immediate in the literature and led to the rejection of the base recovery postulate. As already explained in previous sections, the base recovery postulate cannot be dropped unless replaced by other postulates, which would capture the intuition behind the Principle of Minimal Change. Some authors did that, by replacing the base recovery postulate with other constraints, such as filtering ([8]). Others dropped the AGM postulates altogether, and developed a new set of postulates from scratch ([15], [17]); this approach is reasonable, as the AGM postulates were developed with belief set contraction operators in mind. In either case, the AGM postulates were characterized as unsuitable to handle belief base contraction operators, even by their developers ([1], [22]).

Base Decomposability

To understand why belief base AGM postulates are inapplicable in some common logics, such as PC, we need to study the effects of the base inclusion postulate. This postulate forces us to select a subset of the base as the result of the contraction. When contracting a belief set K, our options for the result are among the subsets of Cn(K); when contracting a belief base K, our options for the result are limited to the subsets of K. So, dealing with belief base contraction is actually the same as assuming that the logic at hand is less expressive than in the belief set case.

Fuhrmann in [8] used this observation to claim that the only bases that can satisfy the base AGM postulates are the *superredundant* bases, i.e. those that are closed under logical consequence (theories). This claim was based on the fact that the powerset of a base does not necessarily contain all the beliefs that it implies; thus, when viewed as a logic, the base does not have enough logical power to satisfy the prerequisites of the AGM theory (disjunctive syllogism, tautological implication etc), except in the special case that the base is a theory (i.e. superredundant). With this result at hand, it seemed reasonable to neglect the AGM postulates when dealing with belief bases. But there was a problem with Fuhrmann's approach: he assumed that the prerequisites originally set by AGM were necessary for AGM-compliant operators to exist, which is not the case, as our analysis showed.

Despite this problem, Fuhrmann's analysis paved the way to find the conditions necessary for the existence of AGM-compliant operators for belief bases. His syllogism can be repeated as follows: suppose a condition P that is necessary and sufficient for an AGM-compliant operator to exist in a logic (in the standard case where belief sets are considered). Assume also two sets A, B and the operation $C=A-B$. When dealing with belief sets, we require that there exists a set C satisfying condition P. Because of the inclusion postulate (K–2), this set can be formed using expressions from the set Cn(A), thus $C\subseteq Cn(A)$.

In the belief base case, we again require that there exists a set C that satisfies condition P. In this case however, the base inclusion postulate (B−2) restricts this set to be formed using elements from A only (instead of Cn(A)), thus $C\subseteq A$. As shown in previous sections, the condition P of our analysis is decomposability. To formally establish these results we need the following definitions:

Definition 8 A logic <L, Cn> is called *base-AGM-compliant with respect to the basic postulates for contraction* (or simply *base-AGM-compliant*) iff there exists a contraction function '−' that satisfies the basic base AGM postulates for contraction (B−1)-(B−6).

Definition 9 Assume any logic <L, Cn> and a set $A \subseteq L$.

- The set $A \subseteq L$ is *base decomposable* iff $B^{-}(A) \cap P(A) \neq \emptyset$ for every $B \subseteq L$
- The logic <L, Cn> is *base decomposable* iff for every A, $B \subseteq L$ $B^{-}(A) \cap P(A) \neq \emptyset$ (or equivalently iff all $A \subseteq L$ are base decomposable)

Using the above definitions, the following theorem can be proven, analogously to theorem 1:

Theorem 6 A logic <L, Cn> is base-AGM-compliant iff it is base decomposable.

Proof

(⇒) Suppose that the logic is base-AGM-compliant. Then there exists a base contraction function '−' that satisfies the basic base AGM postulates for contraction.

Take any A, $B \subseteq L$. It is obvious that $\emptyset$, $A \in P(A)$, so if it is not the case that $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$, then $B^{-}(A) \cap P(A) \neq \emptyset$.

In the principal case where $Cn(\emptyset) \subset Cn(B) \subset Cn(A)$, set $C=A-B$. By the base inclusion postulate we get $C \subseteq A$, which implies that $Cn(C) \subseteq Cn(A)$.

Suppose that $Cn(C)=Cn(A)$. Then $Cn(B) \subseteq Cn(A)=Cn(C)$, so the postulate of success is not satisfied, a contradiction. Thus $Cn(C) \subset Cn(A)$.

Suppose now that $C=A$. Then $Cn(C)=Cn(A)$, a contradiction as above.

Combining the above facts we conclude that $C \subset A$ and $Cn(C) \subset Cn(A)$.

This guarantees that $C \in P(A)$ and that $Cn(C) \subset Cn(A)$.

By the base recovery postulate it follows that $A \subseteq Cn(C \cup B)$, thus:

$Cn(A) \subseteq Cn(Cn(C \cup B))=Cn(B \cup C) \subseteq Cn(Cn(A) \cup Cn(A))=Cn(A)$, which implies that: $Cn(B \cup C)=Cn(A)$.

We conclude that $C \in B^{-}(A)$, thus $C \in B^{-}(A) \cap P(A) \Rightarrow B^{-}(A) \cap P(A) \neq \emptyset$.

We have proved that for any A, $B \subseteq L$ it holds that $B^{-}(A) \cap P(A) \neq \emptyset$, so <L, Cn> is base decomposable.

(⇐) Now assume that the logic is base decomposable. We define the function '−' such that $A-B=C$ for some $C \in B^{-}(A) \cap P(A)$. This function is well-defined because $B^{-}(A) \cap P(A) \neq \emptyset$. We assume that this function depends only on A and Cn(B). In other words, suppose two B_1, B_2 such that $Cn(B_1)=Cn(B_2)$. It is obvious that $B_1^{-}(A)=B_2^{-}(A)$ thus $B_1^{-}(A) \cap P(A)=B_2^{-}(A) \cap P(A)$, yet the function '−' may select different (even non-equivalent) beliefs in the two cases. We exclude this case by requiring that, in the above example, $A-B_1=A-B_2$.

Now, we can prove that the function '−' as defined satisfies the basic base AGM postulates for contraction.

For the postulate of base closure this is obvious.

For the postulate of base inclusion, notice that $C \in P(A)$, thus $C \subseteq A$ by definition, so the postulate of base inclusion holds.

For the postulate of base vacuity, the result follows from the definition of $B^{-}(A)$ and the fact that $A \in P(A)$.

For the postulate of base success, we take the different cases:

- If $Cn(\varnothing)\subset Cn(B)\subseteq Cn(A)$, then assume that $B\subseteq Cn(A-B)=Cn(C)$. Then $Cn(C\cup B)=Cn(C)$. But $C\in B^{-}(A)$, therefore $Cn(A)=Cn(C\cup B)=Cn(C)\subset Cn(A)$, a contradiction.
- In any other case, if $B\neq Cn(\varnothing)$ then $C=A-B=Cn(A)\not\supseteq B$.

The postulate of base preservation follows from the definition of '−' function; for any equivalent B_1, B_2 it holds that $A-B_1=A-B_2$.
For the postulate of base recovery, we will again take the different cases:

- If $Cn(\varnothing)\subset Cn(B)\subseteq Cn(A)$, then $A\subseteq Cn(A)=Cn(C\cup B)$ by the definition of $B^{-}(A)$.
- In any other case $Cn((A-B)\cup B)=Cn(Cn(A)\cup B)\supseteq Cn(Cn(A))\supseteq A$.

We can also get the following corollary:
Corollary 7 A logic <L, Cn> is base-AGM-compliant iff for all A, $B\subseteq L$ with $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$ there exists a $C\subseteq L$ such that $Cn(B\cup C)=Cn(A)$, $Cn(C)\subset Cn(A)$ and $C\subset A$. Equivalently, the logic <L, Cn> is base-AGM-compliant iff for all A, $B\subseteq L$ with $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$ it holds that $B^{-}(A)\cap P(A)\neq\varnothing$.

Notice that the only difference between decomposability and base decomposability has to do with the range of the selection of the set C: in decomposability, C must be properly implied by A ($Cn(C)\subset Cn(A)$); in base decomposability, C must *additionally* be a subset of A ($C\subseteq A$). In other words, a set A is base decomposable if each of its proper implications has a "complement" relative to A that can be expressed using propositions in A (and not Cn(A) as was the case with simple decomposability). Obviously, base decomposability is a stronger condition than decomposability. This is not surprising, as (B−2) is stronger than (K−2):
Corollary 8 Assume a logic <L, Cn>. The following hold:

- If the logic <L, Cn> is base decomposable, then it is decomposable.
- If the logic <L, Cn> is base-AGM-compliant, then it is AGM-compliant.
- If a set $A\subseteq L$ is base decomposable, then it is decomposable.

Base Cuts

The close connection between decomposability and base decomposability could imply a similar connection between cuts and a similar structure for bases, the *base cuts*. Indeed, such a connection exists:
Definition 10 Assume a logic <L, Cn>, a belief $A\subseteq L$ and a set of beliefs $S\subseteq P(L)$ such that:

- For all $X\in S$, $Cn(X)\subset Cn(A)$
- For all $Y\subseteq L$ with $Cn(Y)\subset Cn(A)$ and $Y\subset A$, there is a $X\in S$ such that $Cn(Y)\subseteq Cn(X)$ or $Cn(X)\subseteq Cn(Y)$

Then S is called a *base cut* of A.

Base cuts divide only the subsets of A instead of the subsets of Cn(A). Thus, a cut is a base cut, but not vice-versa. The counterpart of theorem 2 for belief base operations is:
Theorem 7 Suppose a logic <L, Cn> and a set $A\subseteq L$. Then A is base decomposable iff for all base cuts S of A it is the case that $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$.
Proof

($\Rightarrow$) Suppose that A is base decomposable.
If Cn(A)=Cn($\varnothing$) then there is no base cut of A (by the definition of a base cut), so the theorem holds trivially.
Assume that Cn(A)$\neq$Cn($\varnothing$) and that for some base cut S of A it holds that Cn($\cap_{X\in S}$Cn(X))=B$\neq$Cn($\varnothing$). Obviously it holds that B=Cn(B)$\subseteq$Cn(X) for all X$\in$S and B=Cn(B)$\subset$Cn(A).
We will prove that B^-(A)$\cap$P(A)=$\varnothing$.
Take any C$\in B^-$(A)$\cap$P(A). Then C$\subseteq$A.
If C=A then Cn(C)=Cn(A) thus C$\notin B^-$(A). So C$\subset$A.
Since C$\in B^-$(A), it holds that Cn(C)$\subset$Cn(A). So, by the definition of the base cut, there exists a X$\in$S such that Cn(C)$\subseteq$Cn(X) or Cn(X)$\subseteq$Cn(C).
Suppose initially that Cn(C)$\subseteq$Cn(X). Then:
Cn(B$\cup$C)$\subseteq$Cn(Cn(X)$\cup$Cn(X))=Cn(X)$\subset$Cn(A) by the definition of the base cut.
Now suppose that Cn(X)$\subseteq$Cn(C). Then:
Cn(B)$\subseteq$Cn(X)$\subseteq$Cn(C)$\Rightarrow$Cn(B$\cup$C)=Cn(Cn(B)$\cup$Cn(C))=Cn(Cn(C))=Cn(C)$\subset$Cn(A).
We conclude that, for all C such that C$\subset$A and Cn(C)$\subset$Cn(A) it holds that Cn(B$\cup$C)$\subset$Cn(A), thus C$\notin B^-$(A).
Thus B^-(A)$\cap$P(A)=$\varnothing$, so A is not base decomposable, a contradiction.
($\Leftarrow$) Now suppose that A is not base decomposable. Then by definition there exists a set B$\subseteq$L such that Cn($\varnothing$)$\subset$Cn(B)$\subset$Cn(A) and B^-(A)$\cap$P(A)=$\varnothing$.
Take any C$\subseteq$L such that C$\in$P(A) and Cn(C)$\subset$Cn(A). There exists at least one such set, for example C=$\varnothing$.
It holds that Cn(B$\cup$C)$\subseteq$Cn(Cn(A)$\cup$Cn(A))=Cn(A).
Suppose that Cn(B$\cup$C)=Cn(A). Then by definition C$\in B^-$(A), thus C$\in B^-$(A)$\cap$P(A), which is a contradiction because B^-(A)$\cap$P(A)=$\varnothing$.
So Cn(B$\cup$C)$\subset$Cn(A) for all C$\in$P(A) with Cn(C)$\subset$Cn(A).
We take the following family of beliefs: S={B$\cup$Y | Cn(Y)$\subset$Cn(A) and Y$\subset$A}.
It follows that for all X$\in$S Cn(X)$\subset$Cn(A).
Furthermore, for any Y$\subseteq$L such that Y$\subset$A and Cn(Y)$\subset$Cn(A) it holds that B$\cup$Y$\in$S and Cn(Y)$\subseteq$Cn(B$\cup$Y).
Thus S is a cut. Furthermore, for all X$\in$S it holds that Cn(X)$\supseteq$Cn(B), therefore:
Cn($\cap_{X\in S}$Cn(X))$\supseteq$Cn(B)$\supset$Cn($\varnothing$), which is a contradiction by our original hypothesis.
Thus the set A is base decomposable.

The following corollary is immediate:
Corollary 9 A logic <L, Cn> is base-AGM-compliant iff for all A$\subseteq$L and all base cuts S of A it is the case that Cn($\cap_{X\in S}$Cn(X))=Cn($\varnothing$).

Discussion on Belief Base Operations

The above analysis opens up a different viewpoint on the connection between belief base operations and the AGM theory. Up to now, it was considered that the AGM postulates were inconsistent with respect to belief base operations because there were no belief base operations that satisfy them. Our analysis implies that the AGM postulates are inconsistent with respect to belief base operations when applied to non-base-AGM-compliant logics, such as the logics considered in the AGM framework.

To show that the logics of the standard AGM framework are not base-AGM-compliant, take any set A in a compact logic <L, Cn> that supports the conjunction operator with the usual semantics. Due to compactness, this set can be equivalently expressed by a finite set: $A_0=\{x_1,\ldots,x_n\}$, such that $Cn(A_0)=Cn(A)$. Furthermore, due to the semantics of $\wedge$, we get that $Cn(A_0)=Cn(\{x_1\wedge\ldots\wedge x_n\})$. Set $x_0=x_1\wedge\ldots\wedge x_n$, $A'=\{x_0\}$. Then $Cn(A)=Cn(A')$. Notice that $P(A')=\{\varnothing, A'\}$. If we take any set $B\subseteq L$ such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A')$, then it is easy to see that $\varnothing\notin B^-(A')$ and $A'\notin B^-(A')$, thus $B^-(A')\cap P(A')=\varnothing$, so the logic is not base decomposable. This simple argument shows that under very generic conditions, which are part of the AGM prerequisites (compactness, conjunction operator with the usual semantics and the existence of two sets A, $B\subseteq L$ such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$), a logic is not base-AGM-compliant. The contraction example $\{a\wedge b\}-\{a\}$ presented earlier in this section was an application of this general example.

On the other hand, there are logics, not in the class AGM considered, that are base-AGM-compliant. Take for example any set I and the logic $<L_I, Cn_I>$ such that $L_I=I$, $Cn_I(\varnothing)=\varnothing$, $Cn_I(\{x\})=\{x\}$ for all $x\in I$, $Cn_I(A)=L_I$ for all A that contain more than one element. It is easy to show that $<L_I, Cn_I>$ is a logic, for any I; it can also be shown that such logics are base decomposable:

Lemma 9 For any I, the structure $<L_I, Cn_I>$ defined as above is a logic, and it is base decomposable.

Proof

Obviously, for all $A,B\subseteq L$, it holds that $A\subseteq Cn_I(A)$, $Cn_I(Cn_I(A))=Cn_I(A)$ and $A\subseteq B$ implies $Cn_I(A)\subseteq Cn_I(B)$. So, $<L_I, Cn_I>$ is a logic.

Take any $A,B\subseteq L$ such that $Cn_I(\varnothing)\subset Cn_I(B)\subset Cn_I(A)$.

If $I=\varnothing$ or I contains only one element, then there are no such sets, so the lemma holds trivially.

If I contains more than one element, then by the definition of Cn_I, we can easily deduce that $B=\{x\}$ for some $x\in L_I$ and $A\supseteq\{y,z\}$ for some $y,z\in L_I$, $y\neq z$.

We will find a $C\subseteq L$ such that $C\in B^-(A)\cap P(A)$.

If x=y then take $C=\{z\}$. It is easy to see that $Cn(C)\subset Cn(A)$, $Cn(B\cup C)=Cn(A)$ and $C\subseteq A$, thus $C\in B^-(A)\cap P(A)$.

If $x\neq y$ then take $C=\{y\}$. Again, it is easy to see that $Cn(C)\subset Cn(A)$, $Cn(B\cup C)=Cn(A)$ and $C\subseteq A$ regardless of the value of x, thus $C\in B^-(A)\cap P(A)$.

So $B^-(A)\cap P(A)\neq\varnothing$ and the logic is base-AGM-compliant.

Lemma 9 implies that there are several (finite and infinite) logics that are base-AGM-compliant. In a base decomposable logic all sets, even non-superredundant, are base decomposable. Lemma 9 provides a proof for this fact as well, because every set in the family of logics considered is base decomposable regardless of it being a theory or not. So, it should not be assumed beforehand that an AGM-compliant operator for non-superredundant belief bases is impossible.

Unfortunately though, most interesting logics are not base-AGM-compliant. We can be partly compensated for this result by the fact that the theories of an AGM-compliant logic are base decomposable, which is actually the claim made by Fuhrmann in [8]:

Lemma 10 Assume a decomposable logic <L, Cn> and any A⊆L. If A=Cn(A), then A is base decomposable.

Proof

Take any B⊆L. Then $B^{-}(A)\neq\varnothing$.

For any $C\in B^{-}(A)$ it holds that $C\subseteq Cn(C)\subset Cn(A)=A\Rightarrow C\in P(A)$.

Thus $B^{-}(A)\subseteq P(A)$, so $B^{-}(A)\cap P(A)=B^{-}(A)\neq\varnothing$, so A is base decomposable.

The following corollary provides an alternative method to check base decomposability, applicable in certain special cases:

Corollary 10 Assume a decomposable logic <L, Cn>. If A=Cn(A) for all A⊆L, then <L, Cn> is base decomposable.

The fact that a given logic is not base decomposable does not preclude the possibility that this logic contains base decomposable sets, some of which may not even be superredundant. This is another compensating factor for the fact that most logics are not base-AGM-compliant. For example, it is easy to show that in PC with only two atoms a, b, the set A={a∨b, a∨¬b} is base decomposable, even though the logic itself is not base-AGM-compliant and A is not a theory (a∈Cn(A), but a∉A).

Such logics have some interest: take a logic <L, Cn> and a base decomposable set A⊆L. The proof of theorem 6 (the (⇐) route) implies that for any B⊆L there exists a C such that by setting C=A−B, the function '−' satisfies the AGM postulates for base contraction; thus we can define a "local" base-AGM-compliant operator, applicable for A only. In some logics we may even be able to find an operator that always results in another base decomposable set, thus "jumping" from one base decomposable set to another base decomposable set. So, when we carefully select the initial base and the result of each contraction operation, we could get base-AGM-compliant contraction operators even for some of the logics that are not base-AGM-compliant. Unfortunately, for logics with an infinite number of equivalence classes there is no guarantee that base decomposable sets will always be finite.

Roots

To shed more light on the intuition behind decomposability, we will study a relevant concept, the concept of *roots* of a logic. Take any decomposable logic <L, Cn> and any set A⊆L. As mentioned before, decomposability implies that, if there is a set A_1 such that $Cn(\varnothing)\subset Cn(A_1)\subset Cn(A)$ then there is also another set A_2 such that $Cn(A_2)\subset Cn(A)$ and $Cn(A_1\cup A_2)=Cn(A)$. In a sense, A is "decomposed" in two "smaller" sets. Subsequently, both A_1 and A_2 can be "decomposed" in A_{11}, A_{12}, A_{21}, A_{22} and so on in a recursive manner. This procedure may continue indefinitely or it may ultimately stop at a set which has no further implications (except Cn(∅) and itself, of course). For finite logics, all such decompositions will ultimately reach some point where no further decomposition is possible, because finiteness guarantees that we will eventually run out of sets; for infinite logics, some of the decompositions may stop and some may continue indefinitely. In either case, sets which cannot be further decomposed play a major role in our theory and are called *roots* of the logic:

Definition 11 Assume a logic <L, Cn>. A set A⊆L is called a *root* of the logic iff there is no B⊆L such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$.

Notice that roots are defined in any logic, regardless of it being decomposable or not. We can immediately deduce a useful property of roots:

Lemma 11 Any root in any logic is base decomposable (thus decomposable).

Proof

Obvious by the definition of roots.

The notion of roots is closely connected to the concept of a *fully independent root set*:

Definition 12 A logic <L, Cn> *contains a fully independent root set* iff there exists a family of beliefs $B_i \subseteq L$, $i \in I$ such that:

- For all $A \subseteq L$ there exists a set $I(A) \subseteq I$ such that $Cn(\cup_{i \in I(A)} B_i) = Cn(A)$
- For all $j \in I$, $B_j \not\subseteq Cn(\cup_{i \in I \setminus \{j\}} B_i)$

The set $R = \{B_i \mid i \in I\}$ is called a *fully independent root set* of <L, Cn>.

The following lemma shows that any belief in logics which contain a fully independent root set is completely characterized by the members of the fully independent root set that it implies:

Lemma 12 Assume a logic <L, Cn> that contains a fully independent root set $R = \{B_i \mid i \in I\}$. For any $A \subseteq L$ there exists a unique set $I(A) \subseteq I$ such that $Cn(\cup_{i \in I(A)} B_i) = Cn(A)$. Furthermore, $I(A) = \{i \in I \mid A \models B_i, B_i \in R\}$.

Proof

The existence of such a set I(A) follows from the definition.

For uniqueness, suppose that there exists a second set $I(A)' \subseteq I$ such that:

$Cn(A) = Cn(\cup_{i \in I(A)} B_i) = Cn(\cup_{i \in I(A)'} B_i)$.

Take any $j \in I(A) \setminus I(A)'$. Then:

$B_j \subseteq Cn(A) = Cn(\cup_{i \in I(A)'} B_i) \subseteq Cn(\cup_{i \in I \setminus \{j\}} B_i)$, a contradiction by the definition.

Thus $I(A) \setminus I(A)' = \emptyset$.

Similarly, we can show that $I(A)' \setminus I(A) = \emptyset$, which implies that $I(A) = I(A)'$.

For the second part of the lemma, take any $j \in I(A)$.

Then $Cn(A) = Cn(\cup_{i \in I(A)} B_i) \supseteq Cn(B_j) \Rightarrow A \models B_j$ thus $I(A) \subseteq \{i \in I \mid A \models B_i, B_i \in R\}$.

Now take any $j \in \{i \in I \mid A \models B_i, B_i \in R\}$. Suppose that $j \notin I(A)$. Then:

$B_j \subseteq Cn(A) = Cn(\cup_{i \in I(A)} B_i) \subseteq Cn(\cup_{i \in I \setminus \{j\}} B_i)$, a contradiction by the definition. So $j \in I(A)$, thus $\{i \in I \mid A \models B_i, B_i \in R\} \subseteq I(A)$. We conclude that $I(A) = \{i \in I \mid A \models B_i, B_i \in R\}$.

In a logic that contains a fully independent root set, even the consequence operator can be characterized using this set:

Lemma 13 Assume a logic <L, Cn> that contains a fully independent root set $R = \{B_i \mid i \in I\}$ and two sets $A, B \subseteq L$. Then $Cn(A) \subseteq Cn(B)$ iff $I(A) \subseteq I(B)$.

Proof

Suppose that $Cn(A) \subseteq Cn(B)$. If $I(A) = \emptyset$, then obviously $I(A) \subseteq I(B)$.

If $I(A) \neq \emptyset$, then take any $j \in I(A)$. Suppose that $j \notin I(B)$.

Then $B_j \subseteq Cn(\cup_{i \in I(A)} B_i) = Cn(A) \subseteq Cn(B) = Cn(\cup_{i \in I(B)} B_i) \subseteq Cn(\cup_{i \in I \setminus \{j\}} B_i)$, a contradiction.

Thus $j \in I(B) \Rightarrow I(A) \subseteq I(B)$.

Now, if $I(A) \subseteq I(B)$ then: $Cn(A) = Cn(\cup_{i \in I(A)} Cn(B_i)) \subseteq Cn(\cup_{i \in I(B)} Cn(B_i)) = Cn(B)$.

The following lemma uncovers some more properties of a logic that contains a fully independent root set:

Lemma 14 Assume a logic <L, Cn> that contains a fully independent root set $R=\{B_i \mid i\in I\}$. Then the following hold:

- For all $j\in I$, $I(B_j)=\{j\}$
- For all $j,k\in I$, $j\neq k$, $Cn(B_j)\cap Cn(B_k)=Cn(\varnothing)$
- If $Cn(A)\subset Cn(B_j)$ for some $j\in I$, then $Cn(A)=Cn(\varnothing)$
- For all $i\in I$, there exists a $x\in L$ such that $Cn(\{x\})=Cn(B_i)$

Proof

The first result is obvious by lemma 12.

For the second, set $C=Cn(B_j)\cap Cn(B_k)$.

Then there is a set $I(C)\subseteq I$ such that $Cn(C)=Cn(\cup_{i\in I(C)}B_i)=Cn(B_j)\cap Cn(B_k)$.

But $Cn(C)\subseteq Cn(B_j)\Rightarrow I(C)\subseteq I(B_j)=\{j\}$ and $Cn(C)\subseteq Cn(B_k)\Rightarrow I(C)\subseteq I(B_k)=\{k\}$.

So, $I(C)\subseteq\{j\}\cap\{k\}=\varnothing$, thus $C=Cn(C)=Cn(\varnothing)$.

For the third result, suppose that $Cn(A)\subset Cn(B_j)$ for some $j\in I$.

Then $I(A)\subset I(B_j)=\{j\}$, thus $I(A)=\varnothing\Rightarrow Cn(A)=Cn(\varnothing)$.

For the fourth result, take an $i\in I$ and set $B=Cn(B_i)$.

If $B=\varnothing$ then $B_i=\varnothing\subseteq Cn(\cup_{j\in I\setminus\{i\}}B_j)$, a contradiction, even if $I=\varnothing$.

So $B\neq\varnothing$.

If for all $x\in B$ it holds that $Cn(\{x\})=Cn(\varnothing)$ then $Cn(B)=Cn(\varnothing)$, so for any $x\in B$ it holds that $Cn(\{x\})=Cn(B)$.

If there is a $x\in B$ such that $Cn(\{x\})\supset Cn(\varnothing)$ then obviously $Cn(\{x\})\subseteq Cn(B)$.

If $Cn(\{x\})\subset Cn(B)=Cn(B_i)$ then by the third result of this lemma we get $Cn(\{x\})=Cn(\varnothing)$, a contradiction by our hypothesis. So $Cn(\{x\})=Cn(B)$.

As already mentioned, there is a close connection between roots and fully independent root sets. The following lemma shows this connection; a fully independent root set (if it exists) is comprised by roots only; moreover, all roots have an equivalent belief in the fully independent root set:

Lemma 15 Assume a logic <L, Cn> that contains a fully independent root set $R=\{B_i \mid i\in I\}$ and the set $R_0=\{X\subseteq L \mid X \text{ is a root of the logic}\}$. Then the following hold:

- $R\subseteq R_0$
- For any $X\in R_0$ there exists a $j\in I$ such that $Cn(X)=Cn(B_j)$

Proof

If there is a $B\subseteq L$ such that $Cn(B)\subset Cn(B_i)$ for some $i\in I$ then $Cn(B)=Cn(\varnothing)$ by lemma 14, so B_i is a root for any $i\in I$. Thus $R\subseteq R_0$.

For the second result, take any $X\in R_0$.

There is a set $I(X)\subseteq I$ such that $Cn(\cup_{i\in I(X)}B_i)=Cn(X)$.

Initially $I(X)\neq\varnothing$ because $Cn(X)\neq Cn(\varnothing)$.

Take any $j, k\in I(X)$ and suppose that $j\neq k$. Then:

$I(B_j)=\{j\}\subset\{j,k\}\subseteq I(X)$, so $\varnothing\subset I(B_j)\subset I(X)$, thus $Cn(\varnothing)\subset Cn(B_j)\subset Cn(X)$, a contradiction because X is a root.

So $j=k$, thus $I(X)=\{j\}$ for some $j\in I$. So there exists a $j\in I$ such that $Cn(X)=Cn(B_j)$.

The above results show that the roots of a logic represent the smallest pieces of information the logic can express (the most vague information). Some logics are completely characterized by their roots, in the sense that every belief of the logic is uniquely characterized by the roots it implies. These logics contain a fully independent root set; in a logic that contains a fully independent root set, any set can be broken down in its roots (in the sense that it is equivalent to the union of the roots it implies) and no root is implied by any combination of the other roots. Such logics possess interesting properties, such as AGM-compliance:

Proposition 7 Assume a logic <L, Cn> that contains a fully independent root set $R=\{B_i \mid i\in I\}$. Then <L, Cn> is AGM-compliant.

Proof

Take any $A,B\subseteq L$ such that $Cn(\emptyset)\subset Cn(B)\subset Cn(A)$. By lemma 13, we conclude that $\emptyset\subset I(B)\subset I(A)$. Set $I(C)=I(A)\setminus I(B)$ and $C=Cn(\cup_{i\in I(C)}B_i)$.

Then: $I(C)\subset I(A)\Rightarrow Cn(C)\subset Cn(A)$ and:

$Cn(B\cup C)=Cn(Cn(B)\cup Cn(C))=Cn(Cn(\cup_{i\in I(B)}B_i)\cup Cn(\cup_{i\in I(C)}B_i))=$

$=Cn((\cup_{i\in I(B)}B_i)\cup(\cup_{i\in I(C)}B_i))=Cn(\cup_{i\in I(A)}B_i)=Cn(A)$.

So $C\in B^-(A)\Rightarrow B^-(A)\neq\emptyset$, so the logic is AGM-compliant.

Finally, there is a close connection between logics that contain a fully independent root set and base decomposability. The following proposition shows that any set that contains its roots is base decomposable:

Proposition 8 Assume a logic <L, Cn> that contains a fully independent root set $R=\{B_i \mid i\in I\}$ and any $A\subseteq L$. If $A\supseteq\cup_{i\in I(A)}B_i$ then A is base decomposable.

Proof

Since <L, Cn> contains a fully independent root set, <L, Cn> is AGM-compliant by proposition 7, so A is decomposable.

Take any $B\subseteq L$ such that $Cn(\emptyset)\subset Cn(B)\subset Cn(A)$. Then $B^-(A)\neq\emptyset$.

Let $D\in B^-(A)$. Then $Cn(D)\subset Cn(A)$, so $I(D)\subset I(A)$.

Set $C=\cup_{i\in I(D)}B_i$. Then $Cn(C)=Cn(\cup_{i\in I(D)}B_i)=Cn(D)$.

Moreover: $C=\cup_{i\in I(D)}B_i\subseteq\cup_{i\in I(A)}B_i\subseteq A$, so $C\in P(A)$.

$Cn(C)=Cn(D)\subset Cn(A)$ since $D\in B^-(A)$.

$Cn(B\cup C)=Cn(B\cup Cn(C))=Cn(B\cup Cn(D))=Cn(B\cup D)=Cn(A)$ since $D\in B^-(A)$.

Thus $C\in B^-(A)$. We conclude that $B^-(A)\cap P(A)\neq\emptyset$, so A is base decomposable.

The concept of roots is quite instructive on the intuition behind decomposition. It uncovers the recursive character of decomposition by providing a base for the recursion. Notice that a base for this recursion does not always exist, even in decomposable logics, because, in infinite logics, such recursion may continue indefinitely. We believe that roots and fully independent root sets may provide more alternative characterizations of decomposable and base decomposable logics, more sophisticated than the ones appearing in proposition 7 and proposition 8.

Conclusion and Future Work

Main Results

The AGM theory is the leading paradigm in the area of belief change but can only be applied in a certain class of logics; we investigated the possibility of applying the results of this theory in a more general setting, i.e. in a wider class of logics than the one originally considered by AGM in their paper ([1]). We showed that the AGM postulates make sense in many logics outside the class considered in [1]; on the other hand, not all logics of our wide class are combatible with the AGM postulates.

We were able to characterize the logics that are compatible with the AGM postulates in several ways, using decomposability (theorem 1), cuts (theorem 2) and max-cuts (theorem 3). Our results provide methods to check whether any given logic is compatible with the AGM postulates. If a logic is AGM-compliant, then it makes sense to use the AGM postulates as a rationality test for any proposed belief change operator. If not, then we have the option to use one of the alternative set of postulates that have been proposed in the literature (for example in [15]), or provide a new, application-specific set of postulates.

One practical application of this work regards DLs, a very popular KR formalism for the Semantic Web. We showed several DLs to be incompatible with the AGM postulates. It is an undergoing research effort to find one that is AGM-compliant. This case study shows that our method can be successfully used to develop relevant results for practical problems, providing a definite answer on whether the search for an AGM-compliant operator is futile or not in any given logic.

On the theoretical side, we investigated the relation between belief change operators on belief bases and AGM-like belief change operators and found very close connections. Our generalization uncovered a new viewpoint on the problematic application of the AGM postulates in belief base operations and showed that the unconditional rejection of the AGM postulates with regard to belief base operations, a common practice in the literature, is not the proper way to handle the problem. There are logics where belief base operations and AGM postulates are compatible, but such logics are not in the class of logics of the original AGM framework.

Despite these encouraging results, it looks like AGM postulates and belief base operations cannot be combined in most interesting logics, except in special cases. Furthermore, in several cases, the use of an AGM-compliant belief base operator causes most of the computational problems of the standard AGM-compliant operators to persist, which is a rather undesirable property.

Two notions may provide the intuition necessary to encourage the development of deeper results on the subject. Firstly, the close connection of the logics that satisfy the Tarskian axioms with lattice theory allows the use of results developed in lattice theory directly to our problem. This connection was shown in corollary 6 and is related to the concept of equivalence between logics, a rather strong form of equivalence that preserves decomposability. Secondly, the concept of roots provides an alternative intuition behind decomposability, uncovering some previously unknown properties of decomposable logics. Unfortunately, this connection is not applicable in all logics, so this is not an equivalent characterization of AGM-compliant logics. We are currently working on exploiting these results.

We consider this work important because it provides a theoretical framework allowing us to study the feasibility of applying the AGM model in logics originally excluded from the AGM theory, such as DLs. It also allows the reconsideration, on new grounds, of several approaches regarding belief base contraction operators.

Future Work

Our study opens up several interesting questions. Only the contraction operator was considered; we believe that our approach could give similar results regarding other operators, such as revision ([1], [9]), update and erasure ([19]). Moreover, it would be interesting to study the *supplementary* AGM postulates, a set of additional postulates for contraction proposed by AGM in [1].

Since the original publication of the AGM theory, several equivalent formulations were introduced such as partial meet functions ([1]), safe contraction operations ([3]), systems of spheres ([13]), epistemic entrenchment orderings ([11])

and persistent assignments and interpretation orderings ([20]). Such approaches could be viewed under the prism of our more general framework; it would be worthwhile to study whether they remain equivalent to the AGM postulates when the original AGM assumptions are lifted.

References

[1] C. Alchourron, P. Gärdenfors, D. Makinson. On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. Journal of Symbolic Logic, 50: 510-530, 1985.

[2] C. Alchourron, D. Makinson. Maps Between Some Different Kinds of Contraction Functions: the Finite Case. Studia Logica, 45: 187-198, 1986.

[3] C. Alchourron, D. Makinson. On the Logic of Theory Change: Safe Contraction. Studia Logica, 44: 405-422, 1985.

[4] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider eds. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, ISBN: 0521781760, 2002.

[5] S. N. Burris. Logic for Mathematics and Computer Science. Prentice Hall, New Jersey, ISBN: 0132859742, 1998.

[6] M. Dalal. Investigations Into a Theory of Knowledge Base Revision: Preliminary Report. In Proceedings of the Seventh National Conference on Artificial Intelligence, 475-479, 1988.

[7] M. Dalal. Updates in Propositional Databases. Technical Report, DCS-TR-222, Department of Computer Science, Rutgers University, 1988.

[8] A. Fuhrmann. Theory Contraction through Base Contraction. Journal of Philosophical Logic, 20: 175-203, 1991.

[9] P. Gärdenfors. Belief Revision: An introduction. In P. Gärdenfors ed., Belief Revision, 1-20, Cambridge University Press, ISBN: 0521412609, 1992.

[10] P. Gärdenfors. The Dynamics of Knowledge and Belief: Foundational vs. Coherence Theories. Revue Internationale de Philosophie 44, 24-46. Reprinted in Knowledge, Belief and Strategic Interaction, ed. by C. Bicchieri and M. L. dalla Chiara, Cambridge University Press, Cambridge, 1992, 377-396, 1990.

[11] P. Gärdenfors, D. Makinson. Revisions of Knowledge Systems using Epistemic Entrenchment. In Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge, 83-95, 1988.

[12] G. Gratzer. Lattice Theory: First Concepts and Distributive Lattices. W. H. Freeman and co. San Francisco, ISBN: 0716704420, 1971.

[13] A. Grove. Two Modellings for Theory Change. Journal of Philosophical Logic, 17: 157-170, 1988.

[14] P. Halmos. Naïve Set Theory. Springer-Verlag, New York, ISBN: 0387900926, 1974.

[15] S. O. Hansson. A Textbook of Belief Dynamics. Kluwer Academic Publishers, ISBN: 0792353293, 1999.

[16] S. O. Hansson. In Defense of Base Contraction. Synthese, 91: 239-245, 1992.

[17] S. O. Hansson. Knowledge-level Analysis of Belief Base Operations. Artificial Intelligence, 82: 215-235, 1996.

[18] S. O. Hansson. Theory Contraction and Base Contraction Unified. The Journal of Symbolic Logic, 58(2): 602-625, 1993.

[19] H. Katsuno, A. Mendelzon. On the Difference Between Updating a Logical Database and Revising it. In P. Gärdenfors ed., Belief Revision, 183-203, Cambridge University Press, ISBN: 0521412609, 1992.

[20] H. Katsuno, A. Mendelzon. Propositional Knowledge Base Revision and Minimal Change. Technical Report KRR-TR-90-3, Technical Reports on Knowledge Representation and Reasoning, University of Toronto, 1990.
[21] D. Makinson. How to Give it up: A Survey of Some Formal Aspects of the Logic of Theory Change. Synthese, 62: 347-363, 1985.
[22] D. Makinson. On the Status of the Postulate of Recovery in the Logic of Theory Change. Journal of Philosophical Logic, 16: 383-394, 1987.
[23] B. Nebel. A Knowledge Level Analysis of Belief Revision. In R. J. Brachman, H. J. Levesque, R. Reiter eds., In Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89), Toronto, ON, 301-311, 1989.