Generalizing the AGM Postulates: Preliminary Results and Applications

Giorgos Flouris, Dimitris Plexousakis, Grigoris Antoniou

Institute of Computer Science, FO.R.T.H.
P.O. Box 1385, GR 71110, Heraklion, Greece
{fgeo, dp, antoniou}@ics.forth.gr

Abstract

One of the crucial actions any reasoning system must undertake is the updating of its Knowledge Base (KB). This problem is usually referred to as the problem of belief change. The AGM approach, introduced in (Alchourron, Gärdenfors, and Makinson 1985), is the dominating paradigm in the area but it makes some non-elementary assumptions about the logic at hand which disallow its direct application in some classes of logics. In this paper, we drop all such assumptions and determine the necessary and sufficient conditions for a logic to support AGM-compliant operators. Our approach is directly applicable to a much broader class of logics. We apply our results to establish connections between the problem of updating in Description Logics (DLs) and the AGM postulates. Finally, we investigate why belief base operators cannot satisfy the AGM postulates in standard logics.

Introduction

The problem of *belief change* is concerned with the updating of an agent's beliefs in the face of new information that is possibly contradictory with the agent's current beliefs. Being able to dynamically change the stored data is very important in any Knowledge Representation (KR) system. Mistakes may have occurred during the input; or some new information may have become available; or the world represented by the KB may have changed. In all such cases, the agent's beliefs should change to reflect this fact.

The most influential work on belief change is (Alchourron, Gärdenfors, and Makinson 1985). Instead of trying to find a specific method for dealing with the problem of belief change, the authors of (Alchourron, Gärdenfors, and Makinson 1985) chose to investigate the properties that such a method should have in order to be intuitively appealing. The result was a set of postulates (named AGM postulates after the initials of the authors) that every belief change operator should satisfy. That paper had a major influence in most subsequent works on belief change, being the dominating paradigm in the area ever since. The publication of the AGM postulates was followed by a series of works, by several authors, studying the postulates' effects or providing equivalent formulations. A list of relevant works, which is far from being exhaustive, includes (Alchourron and Makinson 1985), (Gärdenfors 1992), (Gärdenfors and Makinson 1988), (Grove 1988), (Katsuno and Mendelzon 1990), (Katsuno and Mendelzon 1992).

Given the (almost) universal acceptance of the AGM postulates in the belief change community as the defining paradigm for belief change operations, it would be desirable to be able to directly apply them in any type of logic. Unfortunately, this is not possible due to the assumptions made by the authors in the original formulation of the postulates.

One important example of application of belief change that cannot be directly accommodated by the AGM model is the problem of updating in DLs. DL is one of the leading formalisms for storing and manipulating knowledge in the Semantic Web. DLs allow the representation of sophisticated relations between concepts and roles; the sophistication of these relations varies, depending on the DL at hand and determines the expressive power as well as the (algorithmic) complexity of reasoning in this DL.

DL bases consist of 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*. Roughly, the Tbox corresponds to the "schema" of the KB and the Abox to the "data" of the KB. For a detailed description of DLs see (Baader et al. 2002).

One problem that has been generally disregarded in the DL literature is the updating of the Tbox. As outlined before, updating a KB is important for several reasons; updating the Tbox of a DL is even more desirable because it allows the simultaneous building of an ontology by different work teams, followed by the merging of the resulting Tboxes. We would like to apply the AGM theory directly to this problem. Unfortunately we cannot do that, due to the assumptions made in the postulates' original formulation, which overrule logics such as DLs.

In the present paper, we drop most assumptions of the AGM theory and study a much broader class of logics. We determine the necessary and sufficient conditions required for an operator that complies with the AGM postulates to exist in a given logic and apply the results to the problem of updating a DL Tbox. The results of our research can be further applied to shed light on our inability (Hansson 1996) to develop base contraction operators that satisfy the AGM postulates. Only informal sketches of proofs will be presented; detailed proofs can be found in the full version of this paper (Flouris, Plexousakis, and Antoniou 2004).

Setting and Terminology

When dealing with logics we often assume the existence of operators such as negation, conjunction, disjunction etc, as well as the existence of an implication operator that allows us to conclude facts from other facts. The implication operator is usually assumed to include classical tautological implication. Such an approach overrules some important types of logics, such as equational logic (see (Burris 1998) for details on equational logic).

In this paper, to preserve generality, we will make fewer assumptions. A logic is just a set (denoted by L), equipped with a consequence (implication) operator (denoted by Cn). The set L contains all the available propositions of the logic. In Propositional Calculus (PC) for example a, $a \wedge b$, $a \vee (\neg b) \in L$. The consequence operator is a function mapping sets of propositions to sets of propositions. We assume that the Cn operator satisfies the *Tarskian axioms*, namely *iteration* (Cn(Cn(A))=Cn(A)), *inclusion* ($A \subseteq Cn(A)$) and *monotony* ($A \subseteq B$ implies $Cn(A) \subseteq Cn(B)$). For A, $B \subseteq L$, we will say that A *implies* B (denoted by $A \models B$) iff $B \subseteq Cn(A)$. In the rest of this paper, the term *logic* will refer to a pair <L, Cn> that satisfies the Tarskian axioms.

Notice that, unlike the assumptions made in the AGM theory, we do not assume the existence of any operators; this means that we can only "connect" propositions by grouping them in a set. Moreover, the consequence operator is not required to include classical tautological implication, is not necessarily compact and could violate the "rule of introduction of disjunctions in the premises". The above assumptions are general enough to include most interesting classes of logics.

Previous Work – AGM Postulates

In their attempt to formalize the field of belief change, Alchourron, Gärdenfors and Makinson in (Alchourron, Gärdenfors, and Makinson 1985) defined three different types of belief change, namely *expansion*, *revision* and *contraction*. 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. The authors introduced a set of postulates for revision and contraction that formally describe the properties that such an operator should satisfy. Our work deals with the operation of contraction only. Dealing with revision, as well as with other non-trivial belief change operators such as update and erasure (Katsuno and Mendelzon 1992) is an interesting topic of future work.

Under the AGM approach, a KB is a set of propositions K closed under logical consequence (K=Cn(K)), also called a *theory*. Any expression $x \in L$ can be contracted from the KB. The operation of contraction can be formalized as a function mapping the pair (K, x) to a new KB K′ (denoted by K′=K−x). Of course, not all functions can be used for contraction. First of all, the new KB (K′) should be a theory itself. As already stated, contraction is an operation that is used to *remove* knowledge from a KB; thus the result should not contain any new, previously unknown, information. Moreover, contraction is supposed to return a new KB such that the expression x is no longer believed; hence x should not be among the consequences of K−x. Finally, the result should be syntax-independent and should remove as little information from the KB as possible, in accordance with the *Principle of Minimal Change*.

The above intuitions were formalized in a set of six postulates, the *basic AGM postulates for contraction*. Here, we will present a slightly generalized version of the postulates, in which the contracted belief can be any set of propositions $A \subseteq L$ (instead of any proposition $x \in L$):

(K−1) Closure: Cn(K−A)=K−A
(K−2) Inclusion: $K-A \subseteq K$
(K−3) Vacuity: If $A \not\subseteq Cn(K)$, then K−A=K
(K−4) Success: If $A \not\subseteq Cn(\varnothing)$, then $A \not\subseteq Cn(K-A)$
(K−5) Preservation: If Cn(A)=Cn(B), then K−A=K−B
(K−6) Recovery: $K \subseteq Cn((K-A) \cup A)$

In general, these postulates express common intuition regarding the operation of contraction and were accepted by most researchers. The only postulate that has been seriously debated is the postulate of recovery (K−6). Some works (Fuhrmann 1991), (Hansson 1996) state that (K−6) is counter-intuitive. Others (Hansson 1999) 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 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 will deal with this problem at a later point. In any case, it is not the aim of this work to settle this debate; our attempt is to find which logics support the AGM postulates and use these results in real-world applications.

Logics and AGM Postulates

The version of the postulates presented in the previous section can be directly applied in our more general framework. While, as shown in (Alchourron, Gärdenfors, and Makinson 1985), in the more restrictive AGM framework there always exist several functions that satisfy the AGM postulates, it turns out that in some logics in the class we consider there is no such contraction function. As the following theorem shows, it is the recovery postulate that causes this problem:

Theorem 1 In every logic there exists a contraction operator satisfying (K−1)-(K−5).

If the recovery postulate is added to our list of desirable postulates, then the above proposition fails. Take for example $L=\{a,b\}$, $Cn(\emptyset)=\emptyset$, $Cn(\{a\})=Cn(\{a,b\})=\{a,b\}$, $Cn(\{b\})=\{b\}$. It can be easily proven that <L, Cn> is indeed a logic. Notice that all theories (except $Cn(\emptyset)$) contain b. So, if we attempt to contract b (or better: {b}) from {a,b} the result can be no other than $\emptyset$, or else the postulate of success would be violated. But then, the recovery postulate is not satisfied, as can be easily verified.

It would be interesting to search for the distinctive property that does not allow us to define a contraction operator that satisfies the AGM postulates in the above logic. The answer is quite easy once we examine the situation a bit closer. Take the two sets $A=\{a,b\}$ and $B=\{b\}$. It holds that $Cn(B)\subset Cn(A)$, which implies that B carries "less information" than A. Suppose that we contract B from A and get a set C ($C=A-B$). By the postulates of inclusion and success, we conclude that: $Cn(C)\subset Cn(A)$. Furthermore, by the postulate of recovery we get: $Cn(A)\subseteq Cn(C\cup B)$, which implies that we must select a set C that "fills 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 strictly 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. In the example presented, the problem lies in the fact that there is no such expression (or set of expressions) in the given L for the pair of sets A, B; thus no operator can satisfy all the basic AGM postulates for contraction. Once we deal with some technicalities and limit cases, it turns out that the situation presented is typical in all logics that do not support the AGM postulates for contraction:

Definition 1 Consider a logic <L, Cn> and a set $A\subseteq L$:

- The logic <L, Cn> is called *AGM-compliant with respect to the basic postulates of contraction* (or simply *AGM-compliant*) iff there exists an operator that satisfies the basic AGM postulates for contraction (K−1)-(K−6).
- The set A is called *decomposable* iff for all $B\subseteq L$ such that $Cn(\emptyset)\subset Cn(B)\subset Cn(A)$ there exists a $C\subseteq L$ such that $Cn(C)\subset Cn(A)$ and $Cn(A)=Cn(B\cup C)$.
- The logic <L, Cn> is called *decomposable* iff all $A\subseteq L$ are decomposable. Equivalently, the logic <L, Cn> is called decomposable iff for all $A, B\subseteq L$ such that $Cn(\emptyset)\subset Cn(B)\subset Cn(A)$ there exists a $C\subseteq L$ such that $Cn(C)\subset Cn(A)$ and $Cn(A)=Cn(B\cup C)$.

Theorem 2 A logic <L, Cn> is AGM-compliant iff it is decomposable.

In the general case, for every set $A\subseteq L$ there exist several sets $B\subseteq L$ with the property that $Cn(\emptyset)\subset Cn(B)\subset Cn(A)$. In order to check L for decomposability we have to check, for each $A\subseteq L$, whether all these sets (B) have a "relative complement" (C) in the sense described above ($Cn(C)\subset Cn(A)$ and $Cn(A)=Cn(B\cup C)$). But there is an alternative option: imagine a family of subsets of Cn(A) with the property that every other subset of Cn(A) either implies or is implied by one of them. Such a set actually "cuts" the beliefs implied by A in two. It also has a very special place in our theory, so we give it a special name:

Definition 2 Consider a logic <L, Cn>, a set $A\subseteq L$ and a family S of sets 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.

Assume now a set $A\subseteq L$, a cut S of A and a set B that is implied by all the sets in a cut. Take $C=A-B$. Since S is a cut and $Cn(C)\subset Cn(A)$ by (K−2) and (K−4), C will either imply or be implied by a set in S. If it implies a set in S, then it also implies B, so it does not satisfy success. If it is implied by a set in S (say $X\in S$), then $Cn(B)\subseteq Cn(X)$ and $Cn(C)\subseteq Cn(X)$, so it is necessarily the case that $Cn(B\cup C)\subseteq Cn(X)\subset Cn(A)$, so recovery is not satisfied. Once again, after we deal with some technicalities, it turns out that this is another equivalent characterization of AGM-compliant logics:

Theorem 3 A logic <L, Cn> is AGM-compliant iff for every $A\subseteq L$ and every cut S of A it holds that $Cn(\cap_{X\in S}Cn(X))=Cn(\emptyset)$.

Consider a family of (proper) subsets of Cn(A) with the property that it contains all the maximal (proper) subsets of Cn(A). This family is obviously a cut, called a max-cut:

Definition 3 Consider a logic <L, Cn>, a set $A \subseteq L$ and a family S of sets such that:

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

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

Notice that every set A has at most one max-cut. Moreover, if there exists a cut S with the property that $Cn(\cap_{X \in S} Cn(X)) \neq Cn(\emptyset)$, then the same property will hold for the max-cut as well (if it exists). So, max-cuts give us the option to check only one cut S of a set A for whether $Cn(\cap_{X \in S} Cn(X)) \neq Cn(\emptyset)$, instead of checking all possible cuts of A. This fact might allow the development of an algorithm for checking decomposability of a logic; evaluating this possibility is part of our future work.

The downside with max-cuts is that they do not always exist; a max-cut of a set A always exists if there is a finite number of equivalence classes implied by A, but in infinite logics there could be $A \subseteq L$ for which no max-cut exists. In any case the following theorem holds:

Theorem 4 Consider a logic <L, Cn> and a set $A \subseteq L$ which has a max-cut S. Then A is decomposable iff for the max-cut S it holds that $Cn(\cap_{X \in S} Cn(X)) = Cn(\emptyset)$.

A Definition of Description Logics

The theorems of the previous section constitute our arsenal for checking whether any logic is AGM-compliant or not. We will try to apply these results in a real case study, namely the problem of updating a Tbox in a DL. As already noted, a Tbox consists of a set of facts (axioms) about concepts and roles of a DL. The general intuition behind the contraction of a DL Tbox is the same as in the AGM case. When contracting a Tbox T with an expression (axiom) x -or with a set of axioms A- one should check whether x (or A) is a consequence of T and, if so, remove some of the axioms of T so as to prevent x (or A) from being a consequence of the new Tbox $T' = T - x$ (or $T' = T - A$). Moreover, this removal must be done minimally, in the sense of the Principle of Minimal Change, which is expressed by the vacuity and recovery postulates. It turns out that, for several DLs, this is not possible.

To address the problem formally, we must first decide on a formal definition of a DL. To our knowledge, there is no such definition. We will assume that a DL consists of a *namespace*, i.e. a set containing concept names and role names (such as "Mother", "has_child", "A", "B", "C" etc), a set of *constants* (such as $\top$, $\bot$), a set of *operators* (such as $\sqcup$, $\sqcap$, $\neg$ etc) and a set of *connectives* (such as $\sqsubseteq$, $\sqsupseteq$, $\cong$ etc), often called *relations*.

The constants and the elements of the namespace combine with the operators in the usual way to form *terms* (such as $(A \sqcap B) \sqcup (\neg C) \sqcup \bot$). An *axiom* of a DL is an expression of the form XRY where X, Y are terms and R is a connective (such as $A \cong B \sqcup C$, where X="A", Y="$B \sqcup C$" and R="$\cong$"). A *Tbox* is a set of axioms. Each DL allows different constants, operators and connectives and has its own conventions and rules for forming axioms; all these limitations define a set L of available axioms in the given DL. The semantics of the given DL determine which axioms are implied by any given set of axioms; in effect, they determine the consequence operator (Cn) of the logic. It is trivial to see that all DLs satisfy the Tarskian axioms. Thus the pair <L, Cn> defined as above identifies the given DL and is a logic in our sense.

Notice that an expression of the form A(x), denoting that x is an instance of the concept A, is also an expression of a DL; yet it is part of the Abox (not the Tbox). Since in this paper we are only interested in updating the Tbox, we will ignore such expressions.

Before dealing with the general case of retracting an axiom from a Tbox, we must deal with a special type of contraction that could appear in a Tbox: contracting all that is known about a given concept or role. 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 T". In the basic framework described above, such a contraction cannot be expressed. For this reason, we will add to our DL definition the *existence assertion operator*, a modal operator denoted by %. This operator can only be applied to elements of the namespace (e.g. %A) and denotes the fact that the concept (or role) A exists in the KB, without giving any further information about A. With this operator, the previously impossible contraction can be expressed using %A ($T - \{\%A\}$). So, the set of our allowable axioms (L) is enriched with all expressions of the form %A for A element of the namespace.

The introduction of this operator affects the consequence operator as well: every axiom "uses" at least some elements of the namespace, so it should imply their existence. The formal definition of the elements "used" by an axiom (or a set of axioms) is presented in (Flouris, Plexousakis, and Antoniou 2004). Informally, an element is used by an axiom iff it appears in the axiom. The set formed by the elements used by an axiom x (or by a set of axioms K) is denoted by U(x) (or U(K)). For example, the axiom x="$A \cong B \sqcap C$" uses the namespace elements A, B, C, so U(x)={A, B, C}. Thus, x implies (among other things) %A, %B, %C. Formally, for an element A of the namespace and a KB K, if $A \in U(K)$ then $\%A \in Cn(K)$.

Thus, in order to express the fact that there is a concept/role A in a KB, we must either form an axiom that contains it (implicit existence) or denote this by %A (explicit existence). Without the operator %, such a fact can only be denoted implicitly.

The existence assertion operator leads us to the introduction of the concept of *Closed Namespace Assumption* (CNA for short). It is usually desirable to assume that concepts or roles that are not used by a DL KB (i.e. they appear nowhere in the KB) do not exist, as far as the KB is concerned. This assumption is usually made implicitly in the literature. It can be formally expressed using operator %: for an element A of the namespace and a KB K, if $A \notin U(K)$ then $\%A \notin Cn(K)$. Notice that the converse of this implication is forced by the % definition. Thus, the CNA implies that $\%A \in Cn(K)$ iff $A \in U(K)$.

The consequences of making this general assumption (CNA) 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. Take x="$A \cong A$" for example. Obviously, A is used by x ($A \in U(x)$), so, by the definition of %, it follows that $\{x\} \models \{\%A\}$. If we allow x to be a consequence of the Tbox K, then $x \in Cn(K)$ thus $\%A \in Cn(K)$. But $A \notin U(K)$ by hypothesis, so the CNA implies that $\%A \notin Cn(K)$, a contradiction.

This argument additionally shows that allowing the existence assertion operator without the CNA does not make much sense. Take any A in the namespace, x="$A \cong A$" and any Tbox K. If CNA is not applied then $x \in Cn(K)$, thus $\%A \in Cn(K)$ for any Tbox K and any A in the namespace. Another useful remark is that the expressions $A \cong A$ and %A are equivalent; thus one could view %A as a shortcut for $A \cong A$ (in DLs where $\cong$ is allowed, of course). Finally, notice that the logic <L, Cn> corresponding to a DL satisfies the Tarskian axioms with or without the operator % and the CNA.

It is obvious that the AGM theory cannot be used to study this problem. Firstly, there are no operators on axioms. Axioms in DLs are of equational nature (e.g. $A \cong B \sqcup C$); thus, if x is an axiom then the expression $\neg x$ is usually undefined; the same goes for expressions of the form $x \wedge y$, $x \vee y$ etc, for x, y axioms of a DL. Secondly, many DLs are not compact. Therefore, the semantics of such a logic are too far from the logics considered in (Alchourron, Gärdenfors, and Makinson 1985), making their framework inapplicable in this context. In the next section, we will investigate the relation of DLs as defined here with the AGM postulates using our more general framework.

Description Logics and AGM Postulates

The application (or not) of % and the CNA greatly changes the semantics of a DL; so we have to split our analysis in DLs with these features and DLs without them. It is obvious that these features enhance the expressiveness of a DL; unfortunately, they can be proven incompatible with the AGM postulates.

To verify this, notice 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 $\{A \cong B\}$ (for A, B concepts of the namespace) is not implied by {%A,%B}. If we attempt to contract {%A} from $Cn(\{A \cong B\})$, 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 $B \cong B$, $B \sqsubseteq B$ etc, which are all implied by %B. Thus, the result should be Cn({%B}). But then the postulate of recovery is violated because $Cn(\{\%A,\%B\}) \subset Cn(\{A \cong B\})$. The situation presented is typical in all DLs that contain non-trivial expressions (i.e. expressions not implied by the mere existence of their namespace elements). It can be generalized as follows:

Theorem 5 Consider a DL <L, Cn> with % and the CNA, with the property that there is at least one set $X \subseteq L$ such that $Cn(X) \supset Cn(\cup_{A \in U(X)}\{\%A\})$, $U(X) \neq \emptyset$ and U(X) is finite. Then <L, Cn> is not AGM-compliant.

This result shows that any non-trivial DL with % and the CNA is not AGM-compliant. However, acknowledging that the introduction of the existence assertion operator and the CNA was mainly due to a technicality, we will not give up on our study of DLs; instead 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 (Baader et al. 2002) for details) to be non-AGM-compliant, as this theorem shows:

Theorem 6 Assume a DL <L, Cn> 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 constants $\{\top, \bot\}$
- Any (or all) of the operators $\{\neg$ (full or atomic), $\sqcap$, $\sqcup\}$ are allowed, as well as at least one of the operators $\{\exists$ (full or limited), $\forall$, $\geq n$, $\leq n\}$
- Only the connective $\{\cong\}$ is allowed

Then <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 $R \cong S$ can only be implied by itself and its symmetric ($S \cong R$). Indeed, it can be easily verified that $\{R \cong S\}$ implies all propositions of the form: $\forall R.X \cong \forall S.X$, $\exists R.X \cong \exists S.X$, $(\geq nR) \cong (\geq nS)$, $(\leq nR) \cong (\leq nS)$, for any concept term X and any number $n \geq 0$. If we group all the above implications in a set, say A, then it can be proved that $Cn(\{R \cong S\}) = \{R \cong S\} \cup \{S \cong R\} \cup Cn(A)$ and $Cn(A) \subset Cn(\{R \cong S\})$. These facts combined imply that the family $S_{cut} = \{A\}$ is a cut of $\{R \cong S\}$, whose intersection is, obviously, equal to $Cn(A) \neq Cn(\emptyset)$.

Theorem 6 implies that FL_0, FL^-, AL and all DLs of the family AL[U][E][N][C] (see (Baader et al. 2002) 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. Notice that the result presented does not imply anything as far as more expressive DLs are concerned; it is possible that a more expressive DL is AGM-compliant. If this is the case, it would be interesting to find the connective(s) and/or operator(s) that would guarantee AGM-compliance.

Belief Base Contraction

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 (Hansson 1996), (Hansson 1999). Furthermore, some authors (Fuhrmann 1991), (Nebel 1989) 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 are due to the fact that the AGM model performs belief change operations on the whole *set* of beliefs; this model does not distinguish between *explicit facts* (acquired directly from observations) and *implicit facts* (implied by the observations). Alternatively, one could perform belief change operations on a small subset of his beliefs, a *base*, which contains only the explicit 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.

Regarding contraction, the main difference between belief base contraction and belief set contraction 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. In this 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 new, modified postulates are:

(B−1) Base closure: $K-A \subseteq L$

(B−2) Base inclusion: $K-A \subseteq K$

(B−3) Base vacuity: If $A \notin Cn(K)$, then $K-A=K$

(B−4) Base success: If $A \notin Cn(\emptyset)$, then $A \notin Cn(K-A)$

(B−5) Base preservation: If $Cn(A)=Cn(B)$, then $K-A=K-B$

(B−6) Base recovery: $K \subseteq Cn((K-A) \cup A)$

Notice that K does not necessarily refer to a theory now; 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)). These postulates can also be found in (Fuhrmann 1991). 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) it should be the case that $\{a \wedge b\} - \{a\} = \emptyset$; 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 (Fuhrmann 1991). Others dropped the AGM postulates altogether, and developed a new set of postulates from scratch (Hansson 1996); 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 (Alchourron, Gärdenfors, and Makinson 1985). Some authors made some brief informal analysis on the reasons behind this failure. In (Fuhrmann 1991) for example, it was claimed that, since a base A does not (in general) contain all the propositions that it implies (Cn(A)), it does not satisfy the prerequisites of the AGM theory (disjunctive syllogism, tautological implication etc). Thus, only *superredundant* bases (i.e. theories) can have enough logical power to satisfy base recovery and the other postulates. 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 AGM-compliant operators for belief bases to exist. His syllogism can be repeated as follows: assume 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. More formally, we get:

Definition 4 Consider a logic <L, Cn> and a set $A\subseteq L$:

- The logic <L, Cn> is called *AGM-compliant with respect to the basic postulates of belief base contraction* (or simply *base-AGM-compliant*) iff there exists an operator that satisfies the basic AGM-postulates for belief base contraction (B−1)-(B−6).
- The set A is called *base-decomposable* iff for all $B\subseteq L$ such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$ there exists a $C\subseteq L$ such that $C\subseteq A$, $Cn(C)\subset Cn(A)$ and $Cn(A)=Cn(B\cup C)$.
- The logic <L, Cn> is called *base-decomposable* iff all $A\subseteq L$ are base-decomposable. Equivalently, the logic <L, Cn> is called base-decomposable iff for all A, $B\subseteq L$ such that $Cn(\varnothing)\subset Cn(B)\subset Cn(A)$ there exists a $C\subseteq L$ such that $C\subseteq A$, $Cn(C)\subset Cn(A)$ and $Cn(A)=Cn(B\cup C)$.

Theorem 7 A logic <L, Cn> is base-AGM-compliant iff it is base-decomposable.

Notice that the only difference between decomposability and base-decomposability has to do with the selection of the set C: in decomposability, C must be implied by A ($Cn(C)\subset Cn(A)$); in base-decomposability, C must *additionally* be a subset of A ($C\subseteq A$). It is obvious that base-decomposability is a stronger condition than decomposability. This should be expected, as (B−2) is stronger than (K−2). 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).

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 5 Consider a logic <L, Cn>, a set $A\subseteq L$ and a family S of sets such that:

- For all $X\in S$, $Cn(X)\subset Cn(A)$
- For all $Y\subseteq L$ such that $Cn(Y)\subset Cn(A)$ 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.

Theorem 8 A logic <L, Cn> is base-AGM-compliant iff for every $A\subseteq L$ and every base-cut S of A it holds that $Cn(\cap_{X\in S}Cn(X))=Cn(\varnothing)$.

The above analysis resolves the question of why the logics of the AGM framework do not support base AGM contraction operators in the general case: such logics are not base-decomposable. The compactness property of such logics and the semantics of the conjunction operator allows the replacement of any set A with an equivalent expression x, such that $Cn(\{x\})=Cn(A)$. By (B−2), the only acceptable results of the contraction of {x} by any belief would be {x} and $\varnothing$, which are (generally) both overruled by the other postulates. Thus, there are several sets of the form {x} which are not base-decomposable. The contraction example $\{a\wedge b\}-\{a\}$ presented earlier in this section was an application of this general example.

On the other hand, theorems 7 and 8 imply that there exist non-superredundant bases satisfying the condition of base-decomposability. One example proving this is the class of logics $<L_I, Cn_I>$ where $L_I=\{x_i \mid i\in I\}$ and $Cn_I(\varnothing)=\varnothing$, $Cn_I(\{x_i\})=\{x_i\}$ for all $i\in I$ and $Cn_I(\{x_i, x_j\})=L_I$ for $i\neq j$, $i,j\in I$. It is easy to show that all logics in this class are base-decomposable, thus every set $A\subseteq L_I$ is base-decomposable, even if it is not superredundant. So, it should not be assumed beforehand that an AGM-compliant operator for non-superredundant belief bases is impossible.

Unfortunately, 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. Moreover, most logics contain base-decomposable sets, some of which may not be superredundant. For example, if the underlying logic is PC with only two atoms a, b, then the set $A=\{a\vee b, a\vee\neg b\}$ is base-decomposable, even though the logic itself is not base-AGM-compliant and the set A is not a theory ($a\in Cn(A)$, but $a\notin A$).

Such logics have some interest: take a logic <L, Cn> and a base-decomposable set $A\subseteq L$. It can be proved that for any $B\subseteq L$ 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 bases, 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.

Discussion

The description of the connection of belief base contraction with the AGM theory completes the main results of our research, which are the following:

- The original AGM theory restricted the study of belief change operators in a special class of logics. AGM studied the properties necessary for a rational belief change operator in such logics, proposed the AGM postulates and showed that they make sense because there are always several belief change operators satisfying them in each logic of the class they considered. We took a different route. We studied a much broader class of logics, with minimal restrictions as to their properties, and formulated the necessary and sufficient conditions for a given logic in this wider class to support a rational belief change operator. In this context, the word "rational" means "satisfying the AGM postulates". We showed that, in our class, there are logics which do not support rational belief change operations. On the other hand, there are AGM-compliant logics that do not fall into the class originally considered by AGM.
- Our framework allows the study of several logics with respect to the AGM postulates, even if these logics do not support the original prerequisites of the AGM theory. An immediate application of this fact is our study regarding DLs.
- Previous work has completely disregarded the base recovery postulate as far as belief base contraction operators are concerned. We investigated the reasons why the base recovery postulate is so problematic when applied in belief base contraction operators. We showed that this problem is related not only to the base recovery postulate but to the logic under question and the selection of the base as well. There are logics that are base-AGM-compliant. Furthermore, in some non-base-AGM-compliant logics, there exist base-decomposable sets. Unfortunately, there is no guarantee that such sets will always be finite; so some of the problems with the postulate still remain.

These results can be further exploited. Take for example any set A in an AGM-compliant logic. Decomposability implies that, if there exists a set B implied by A (but not equivalent to A), then A can be broken down in at least two other sets, say B and C. Subsequently, B may be further broken down in two smaller sets and so on. This procedure may continue indefinitely, or it may ultimately stop at a set which has no further implications (except $Cn(\varnothing)$ 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 are called *roots* of the logic.

The roots of a logic represent the smallest pieces of information the logic can express (the most vague information). In some logics, 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 several interesting properties, including decomposability. In such logics, any set can be uniquely defined in terms of the roots it implies and any set that contains its roots is base-decomposable. We omit the formal phrasing and the proof of the above results due to lack of space.

Another interesting result deals with *equivalent* logics; a rather strong form of equivalence between logics is defined in (Flouris, Plexousakis, and Antoniou 2004). One important property of this relation is the fact that it preserves AGM-compliance. Thus, once a logic has been proven to be AGM-compliant (or non-AGM-compliant), we can propagate this result to all its equivalent logics.

The second important property of equivalence is related to the connection of our theory with lattice theory (see (Grätzer 1971) for details on lattice theory). Any given logic in our framework can be mapped to a complete lattice, by mapping each belief (set of expressions) of the logic to an element of the lattice and using $\models$ (or its symmetric) as the partial order of the lattice. In fact, it can be proved that the class of logics we consider (modulo the above equivalence relation) is isomorphic to the class of complete lattices.

This result allows any logic to be represented as a complete lattice and vice-versa. Therefore, all results related to our theory can be formulated in terms of lattice theory. Lattice theory provides a nice and clear visualization of the concepts and results presented. Moreover, 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. Describing this alternative representation in more detail was omitted due to lack of space.

Conclusion and Future Work

In this paper, we studied the class of logics that satisfy the Tarskian axioms and developed results regarding the connection of the AGM theory with arbitrary logics from this class. It was shown that some, but not all, logics in this wide class admit AGM-compliant contraction operators. Using our approach we showed that some DLs are not AGM-compliant. It would be interesting to find a DL that supports an AGM-compliant operator. We are currently studying this problem; it is possible that some of the very expressive DLs is AGM-compliant.

A further application of our results dealt with the problem of applying the AGM postulates in belief base operations. We established the exact properties required in order for a logic to support an AGM-compliant operator for belief bases and explained why the logics originally considered in the AGM framework are not base-AGM-compliant. Furthermore, we showed that, in some cases, a careful selection of the base can guarantee the existence of an AGM-compliant operator for belief bases. This is trivially true for base-AGM-compliant logics, but it also applies for some of the other logics. Despite this encouraging result, several problems regarding the application of the AGM postulates (especially the recovery postulate) in belief base operations still remain (such as finiteness of base-decomposable bases).

We consider the above results important because they provide 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. They also allow the reconsideration, on new grounds, of several approaches regarding belief base contraction operators.

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, update and erasure (Katsuno and Mendelzon 1992). Moreover, it would be interesting to study the *supplementary* AGM postulates, a set of additional postulates for contraction proposed by AGM in (Alchourron, Gärdenfors, and Makinson 1985).

Since the original publication of the AGM theory, several equivalent formulations were published such as partial meet functions (Alchourron, Gärdenfors, and Makinson 1985), safe contraction operations (Alchourron and Makinson 1985), systems of spheres (Grove 1988), epistemic entrenchment orderings (Gärdenfors and Makinson 1988) and persistent assignments and interpretation orderings (Katsuno and Mendelzon 1990). 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

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

Alchourron, C.; and Makinson, D. 1985. On the Logic of Theory Change: Safe Contraction. *Studia Logica* 44: 405-422.

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

Burris, S. N. ed. 1998. *Logic for Mathematics and Computer Science*. New Jersey: Prentice Hall.

Grätzer G. ed. 1971. *Lattice Theory: First Concepts and Distributive Lattices*. San Francisco: W. H. Freeman & co.

Flouris, G.; Plexousakis, D.; and Antoniou, G. 2004. AGM Postulates in Arbitrary Logics: Initial Results and Applications, Technical Report, TR-336, ICS-FORTH.

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

Gärdenfors, P. 1992. Belief Revision: An introduction. In Gärdenfors, P. ed. *Belief Revision*, 1-20. Cambridge University Press.

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

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

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

Hansson, S. O. ed. 1999. *A Textbook of Belief Dynamics*. Kluwer Academic Publishers.

Katsuno H.; and Mendelzon A. 1990. Propositional Knowledge Base Revision and Minimal Change, Technical Report, KRR-TR-90-3, Technical Reports on Knowledge Representation and Reasoning, University of Toronto.

Katsuno, H.; and Mendelzon, A. 1992. On the Difference Between Updating a Logical Database and Revising it. In Gärdenfors, P. ed., *Belief Revision*, 183-203. Cambridge University Press.

Nebel, B. 1989. A Knowledge Level Analysis of Belief Revision. In Brachman, R. J.; Levesque H. J.; and Reiter R. eds. In Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning (KR'89), 301-311. Toronto, ON.