On Applying the AGM Theory to DLs and OWL

Giorgos Flouris, Dimitris Plexousakis, and 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. It is generally acknowledged that any Knowledge Base (KB) should be able to adapt itself to new information received. This problem has been extensively studied in the field of belief change, the dominating approach being the AGM theory. This theory set the standard for determining the rationality of a given belief change mechanism but was placed in a certain context which makes it inapplicable to logics used in the Semantic Web, such as Description Logics (DLs) and OWL. We believe the Semantic Web community would benefit from the application of the AGM theory to such logics. This paper is a preliminary study towards the feasibility of this application. Our approach raises interesting theoretical challenges and has an important practical impact too, given the central role that DLs and OWL play in the Semantic Web.

1 Introduction

One of the crucial tasks towards the realization of the vision of the Semantic Web is the encoding of human knowledge in special structures (ontologies), using certain formal encodings (representation languages), such as DLs [3] and OWL [5]. Simply encoding the knowledge is not enough though; knowledge needs to be updated as well. There are several reasons for that: a piece of knowledge that was previously unknown, classified or otherwise unavailable may have become known; or a mistake may have occurred in the conceptualization of the domain or during the input; or the domain itself may have changed. In all these cases the ontology needs to be updated to accommodate the change. Even the development of an ontology is a highly iterative revision process, in which the ontology passes through several revising steps before reaching its "final" version.

For all the above reasons, developing an automatic, consistent and rational updating method for ontologies is a task of great interest to the Semantic Web community. Despite this fact, the problem of ontology updating has been generally disregarded in the relevant literature [13]. In the current paper, we view this problem as a special case of the general problem of *belief change* (also known as *belief revision*) [8], which deals with the updating of a KB in the face of new information.

The problem of belief change has been extensively studied in the literature, resulting in several interesting results, the most important approach being the work by Alchourron, Gärdenfors and Makinson (AGM for short) in [1], known as the *AGM theory*. In that paper, the authors did not attempt to introduce a new algorithm for belief change; instead, they proposed certain rationality constraints (known as the *AGM postulates*) which should be satisfied by any rational belief change algorithm, thus setting

the foundations for future research on the subject. The importance of the AGM theory lies in the fact that these postulates were accepted by most researchers as an appropriate condition to determine the rationality of a certain belief change operator.

Unfortunately, the AGM theory is based on assumptions [1] that generally fail for DLs and OWL [7]; thus, the AGM theory cannot be directly applied to the Semantic Web. Moreover, to the authors' knowledge, there has been no attempt towards a general standard of rationality for belief change operators, in the AGM pattern, for several logics outside the AGM framework (such as the logics used in the Semantic Web). One possible way to address this issue would be to introduce several different, language-specific postulates that take into account the peculiarities of each language.

In this work, we opt for a more general approach. We believe that the concept of rationality is largely independent of the underlying knowledge representation scheme, despite the different properties of each language. Thus, it may be possible to use any condition determining the rationality of a belief change operator in several different contexts; this avoids the problem of "reinventing the wheel" for each different logic.

Given the appeal of the AGM model in the belief change literature, we believe that dropping the AGM assumptions and using the theory in a more general context is a reasonable initial choice. Some people may disagree on whether the AGM theory is the best choice for the Semantic Web; only future research can uncover the strengths and weaknesses of this method, as well as of its alternatives. This paper focuses on the AGM theory of *contraction* [1], by determining whether this theory can be successfully generalized to apply to DLs and OWL. This is the first step towards evaluating the feasibility of applying the AGM theory in the context of the Semantic Web.

The idea of using the intuitions behind the AGM theory to develop a more general version was initially pursued by the authors in [6], [7], where this generalization was defined and its properties were studied; the use of belief change techniques to address the problem of ontology updating has also been independently considered in [12], [13], [16]. Such techniques could be useful in automating the third phase of ontology evolution (as defined in [17]), under which the change(s) to be made in the ontology in response to a certain need are determined (Semantics of Change phase).

In the current paper, we extend the work presented in [7]; we study the feasibility of applying the generalized AGM theory of contraction to DLs and OWL, develop conditions under which (the generalized version of) the AGM theory can (or cannot) be applied to DLs and show that the approach fails for OWL. Our focus lies on the theoretical aspects of our approach; practical issues, like implementation, or applications of our method to specific languages are given less weight. Throughout this paper, for uniformity purposes, we will use the term KB to refer to ontologies as well.

2   Preliminaries

2.1   Description Logics (DLs) and the Web Ontology Language (OWL)

The term *Description Logics* [3] refers to a family of knowledge representation languages, heavily used in the Semantic Web [4]. In DLs, *classes* are used to represent basic concepts, *roles* to represent basic binary relations between objects and *individuals* to represent objects. Those primitive notions can be combined using certain

operators (such as ¬, ⊓, ∃ etc) to produce more complex *terms*. Finally, *connectives* are used to represent relationships between terms, such as inclusion (⊑), disjointness (disj) and others. Each such relationship is called an *axiom*. Axioms dealing with classes and roles form the *Tbox*, while axioms dealing with individuals form the *Abox*. The operators and connectives that a certain DL admits determine the type and complexity of the available axioms, which, in turn, determine the expressive power and the reasoning complexity of the DL. Reasoning in DLs is based on standard model-theoretic semantics. For more details on DLs and their semantics, see [3]. In the following, the term *DL Knowledge Base* (*DL KB*) will refer to a set of general Tbox and/or Abox axioms representing knowledge regarding a domain of interest.

The Web Ontology Language [5], known as OWL, is a knowledge representation language that is expected to play an important role in the future of the Semantic Web, as it has become a W3C Recommendation. OWL comes in three flavors (or species), namely OWL Full, OWL DL and OWL Lite, with varying degree of expressive power and reasoning complexity. In OWL, knowledge is represented using an RDF-like syntax. OWL contains several features allowing the representation of complex relationships between classes, roles and objects in a pattern very similar to the one used in DLs; this close relationship was verified in [10], where OWL DL and OWL Lite (with their secondary annotation features removed) were shown equivalent to the DLs SHOIN$^+$(D) and SHIF$^+$(D) respectively. On the other hand, OWL Full provides a more complete integration with RDF, containing features not normally allowed in DLs; furthermore, its inference problem is undecidable [10]. For more details on OWL and the differences between OWL Full, OWL DL and OWL Lite, refer to [5].

2.2 The AGM Theory and Its Generalization

The problem of belief change deals with the updating of a KB in the face of new, possibly contradictory, information. Undoubtedly, the most influential work in the area of belief change is the work by AGM [1]. In that paper, three fundamental operations of belief change were defined, namely *expansion*, *revision* and *contraction*, as well as a set of rationality postulates that should apply to each of the above operations.

In the current paper, we restrict our attention to the operation of contraction (denoted by '−') which refers to the consistent removal of a piece of information from a KB when this information is no longer believed. Contraction was chosen for our initial approach because, according to AGM, it is the most fundamental among the three belief change operators [1], [8]. Indeed, the theoretical importance of contraction has been accepted by most researchers, even though revision (which refers to consistent addition of information) is more often used in practical applications.

AGM used several assumptions when formulating their theory. Under these assumptions, a logic is a pair <L,Cn>, where L is a set containing all the expressions of the logic and Cn is a *consequence operator* that satisfies the Tarskian axioms (iteration, inclusion, monotony). Using this consequence operator, we can define the implication relation as: K⊨X ⇔ Cn(X)⊆Cn(K). This is the only assumption that was kept during the generalization of the AGM theory in [7]; AGM additionally assumed that the logic is closed under the standard operators (¬, ∧, etc); they also assumed that the

consequence operator includes classical tautological implication, it is compact and it satisfies the "rule of introduction of disjunctions in the premises".

It is easy to see that many AGM assumptions fail for DLs and OWL. For example, a DL is not necessarily closed under the usual operators (¬, ∧, etc); DL axioms are of equational form (e.g., A⊓B⊑C), so the negation of an axiom cannot be defined in general. Furthermore, many DLs are not compact. The same holds for OWL, as well as for many other families of knowledge representation languages [7]. On the other hand, our more general framework engulfs DLs, since, for any given DL, we can take L to be the set of all axioms that can be formed in this DL and Cn(X) the set of all implications of a set of axioms X⊆L under the standard model-theoretic semantics of DLs [3]. Similar facts hold for OWL.

Regarding the operation of contraction, AGM assumed that a KB is a set of propositions of the underlying logic (say K⊆L) which is closed under logical consequence (i.e., K=Cn(K)), also called a *theory*. Any single expression x∈L of the logic 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). In [7], the definition of the contraction operator was slightly extended to include cases where both operands are sets of expressions of the underlying logic (i.e., K′=K−X, for K,X⊆L).

The above assumptions allow any binary operator to be a "contraction" operator, which, of course, should not be the case; for this reason, AGM introduced several restrictions on the result of a contraction operation. First of all, the result 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 contracted expression is no longer believed or implied. 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* [8]. The above intuitions were formalized in a set of six postulates, the *basic AGM postulates for contraction*; these are omitted due to lack of space, but can be found in [1].

As shown by the above analysis, the intuitions that led to the development of the AGM postulates are independent of the underlying knowledge representation language. On the other hand, the formulation of the AGM postulates themselves depends on the AGM assumptions (see [1]). For this reason, in [7], each AGM postulate was reformulated in such a way as to be applicable to all logics in our more general framework, while preserving the original intuition that led to its definition. The resulting postulates can be found in the following list, where the naming and numbering of each postulate corresponds to the original AGM naming and numbering [7]:

(K−1) Closure:	Cn(K−X)=K−X
(K−2) Inclusion:	K−X⊆Cn(K)
(K−3) Vacuity:	If X⊄Cn(K), then K−X=Cn(K)
(K−4) Success:	If X⊄Cn(∅), then X⊄Cn(K−X)
(K−5) Preservation:	If Cn(X)=Cn(Y), then K−X=K−Y
(K−6) Recovery:	K⊆Cn((K−X)∪X)

Unfortunately, it soon became clear that not all logics in our wide framework can admit a contraction operator that satisfies the (generalized) AGM postulates. Following this observation, we defined a logic to be *AGM-compliant* iff a contraction operator that satisfies the generalized AGM postulates can be defined in the given logic. This class of logics was characterized using the following proposition [7]:

Proposition 1. A logic <L,Cn> is AGM-compliant iff for all sets K, X⊆L such that Cn(∅)⊂Cn(X)⊂Cn(K) there is a Z⊆L such that Cn(Z)⊂Cn(K) and Cn(X∪Z)=Cn(K).

With the above postulates, we have succeeded in developing a generalized version of the AGM theory for contraction. The generalized AGM postulates can be used in all logics in our framework; however, for a non-AGM-compliant logic, such an option does not make much sense, as no contraction operator satisfying the postulates (K−1)-(K−6) can be defined. Proposition 1 is the tool that allows us to determine whether this is the case or not.

3   Conditions for AGM-Compliance

3.1   General Intuition and Main Results

In the following, we will refer to a DL as a pair <L,Cn>, where L is the set that contains all the axioms that can be defined in this DL and Cn is the consequence operator under the standard model-theoretic semantics of DLs [3]. Initially, we will consider DLs that allow for the top concept ⊤ and the connective ⊑ (applicable to concept terms, at least), plus an arbitrary number of other connectives and/or operators.

Our approach is based on the following observation: take two sets of DL axioms of the form K={A⊒⊤}, X={B⊒⊤} such that Cn(∅)⊂Cn(X)⊂Cn(K). Set Z={A⊒B}; Z is a good candidate for the set Z required by proposition 1, since Cn(Z)⊆Cn(K) and Cn(X∪Z)=Cn(K). There is a catch though: proposition 1 requires that Cn(Z)⊂Cn(K); in the above approach sometimes it so happens that Cn(Z)=Cn(K). For example, if K={A⊒⊤} and X={¬A⊔∃R.A⊔∀R.⊥⊒⊤} for some role R, then it holds that Cn(∅)⊂Cn(X)⊂Cn(K). If we take Z as above, we get Z={A⊒¬A⊔∃R.A⊔∀R.⊥}, which is equivalent to K={A⊒⊤}, so Cn(Z)=Cn(K). Thus, the constructed Z does not satisfy the conditions set by proposition 1 (example provided by Thomas Studer, personal communication). To deal with this problem, the idea must be somehow refined in order to guarantee that Cn(Z)⊂Cn(K) will hold in all cases. This refinement is described and proved in a more general setting in the following lemma:

Lemma 1. Consider the sets of axioms K={A_j⊒⊤ | j∈J} and X={B⊒⊤}. If Cn(∅)⊂Cn(X)⊂Cn(K) and there is an interpretation I such that B^I=∅, then there is a set Z such that Cn(Z)⊂Cn(K) and Cn(X∪Z)=Cn(K).

Proof. Set Z={A_j⊒B | j∈J} and assume that Z⊭K. Then Z⊭X. By the hypothesis, there is an interpretation I such that B^I=∅; for this interpretation, Z is obviously satisfied, while X is not. This is a contradiction, so Z⊨K. On the other hand, K⊭Z; thus, Cn(Z)⊂Cn(K). The relation Cn(X∪Z)=Cn(K) is obvious by the transitivity of ⊑.   □

Lemma 1 guarantees the existence of the set Z required by proposition 1, but only for sets K, X of a special form. This might cause one to believe that it is of limited use; on the contrary, lemma 1 forms the backbone of our theory. Before showing that, we will show that the prerequisites of proposition 1 need to be checked for only a subset of all the possible (K,X) pairs:

Lemma 2. Consider a logic $<L,Cn>$ and two sets K, $X \subseteq L$, such that $Cn(\emptyset) \subset Cn(X) \subset Cn(K)$. If there are sets K′, $X′ \subseteq L$ such that $Cn(K′)=Cn(K)$, $Cn(\emptyset) \subset Cn(X′) \subseteq Cn(X)$ and a $Z \subseteq L$ such that $Cn(Z) \subset Cn(K′)$, $Cn(X′ \cup Z)=Cn(K′)$, then $Cn(Z) \subset Cn(K)$ and $Cn(X \cup Z)=Cn(K)$.

Proof. Obviously $Cn(Z) \subset Cn(K)$. Since $Cn(X′) \subseteq Cn(X) \subset Cn(K)$ we can conclude that $Cn(X \cup Z) \supseteq Cn(X′ \cup Z)=Cn(K′)$, so $Cn(X \cup Z)=Cn(K)$.   □

Now consider any two sets of axioms K, $X \subseteq L$ of the underlying DL, such that $Cn(\emptyset) \subset Cn(X) \subset Cn(K)$. If K and X are of the form required by lemma 1, then we are done; lemma 1 allows us to find a set Z that satisfies the requirements of proposition 1 for an AGM-compliant logic. If, on the other hand, K or X are not of the desired form, lemma 2 shows the way; all we need is to find two sets K′, X′ of the desired form such that $Cn(K′)=Cn(K)$ and $Cn(\emptyset) \subset Cn(X′) \subseteq Cn(X)$. Then, lemma 1 can be applied for K′, X′ and the resulting set Z can be propagated to K, X using lemma 2. These ideas lead to the main result of this section:

Theorem 1. Consider a DL $<L,Cn>$, such that:

- For all $K \subseteq L$ there is a $K′ \subseteq L$ such that $K′=\{A_j \sqsupseteq \top \mid j \in J\}$ and $Cn(K)=Cn(K′)$
- For all $X \subseteq L$ there is a $X′ \subseteq L$ such that $X′=\{B \sqsupseteq \top\}$, there is an interpretation I such that $B^I=\emptyset$ and $Cn(\emptyset) \subset Cn(X′) \subseteq Cn(X)$

Then this DL is AGM-compliant.

The important question is, in which DLs do the sets K′, X′ required by theorem 1 exist? With the aid of table 1, it can be shown that several very expressive DLs allow transformations resulting in these K′, X′. Table 1 shows how each of the axiom types commonly used in DLs can be equivalently rewritten in the form $A \sqsupseteq \top$. Using this table, we can generate K′, X′ as required by theorem 1 as follows: for K′, replace each axiom of K with its equivalent in the second column; for X′, select one non-tautological axiom of X, replace it with its equivalent from table 1 (say $B \sqsupseteq \top$) and set $X′=\{\forall \top_R.B \sqsupseteq \top\}$.

All the transformations in table 1 can be shown using model-theoretic arguments. Moreover, K′ as defined above obviously fulfills the requirements of theorem 1. For X′, notice that the axiom selected from X is non-tautological, so there is an interpretation for which $B^I \neq \top^I$; for this interpretation, it holds that $(\forall \top_R.B)^I=\emptyset$. Furthermore, $X \vDash \{B \sqsupseteq \top\} \vDash \{\forall \top_R.B \sqsupseteq \top\}$ and $Cn(X′) \neq Cn(\emptyset)$, so X′ is of the desired form as well.

Table 1 shows that the necessary transformations are possible for axioms involving concepts, roles and even individuals. Thus, our results apply also to DL KBs that contain a non-empty Abox. In table 1, A, B refer to concept terms, R, S refer to role terms and a, b refer to individuals. All operators subscripted by $\cdot_R$ (in the third column)

Table 1. Transforming axioms into the form A⊒⊤

Axiom	Equivalent axiom of the proper form	Required operators
A⊑B	¬A⊔B⊒⊤	¬,⊔
R⊑S	∀(R⊓¬S).⊥⊒⊤	⊥,∀,¬$_R$,⊓$_R$
A⋢B	∃⊤$_R$.(A⊓¬B)⊒⊤	¬,⊓,∃,⊤$_R$
R⋢S	∃⊤$_R$.∃(R⊓¬S).⊤⊒⊤	∃,⊤$_R$,¬$_R$,⊓$_R$
A≅B	(¬A⊔B) ⊓ (A⊔¬B)⊒⊤	¬,⊔,⊓
R≅S	∀(R⊓¬S).⊥ ⊓ ∀(S⊓¬R).⊥⊒⊤	⊥,⊓,∀,¬$_R$,⊓$_R$
A≇B	∃⊤$_R$.[(A⊓¬B) ⊔ (B⊓¬A)]⊒⊤	¬,⊓,⊔,∃,⊤$_R$
R≇S	∃⊤$_R$.∃[(¬R⊓S) ⊔ (¬S⊓R)].⊤⊒⊤	∃,⊤$_R$,¬$_R$,⊓$_R$,⊔$_R$
A⊏B	(¬A⊔B) ⊓ ∃⊤$_R$.(B⊓¬A)⊒⊤	¬,⊓,⊔,∃,⊤$_R$
R⊏S	∀(R⊓¬S).⊥ ⊓ ∃⊤$_R$.∃(S⊓¬R).⊤⊒⊤	⊥,⊓,∃,∀,⊤$_R$,¬$_R$,⊓$_R$
A\⊏B	∀⊤$_R$.∃⊤$_R$.(A⊓¬B) ⊔ ∀⊤$_R$.(¬B⊔A)⊒⊤	¬,⊓,⊔,∃,∀,⊤$_R$
R\⊏S	∀⊤$_R$.∃⊤$_R$.∃(R⊓¬S).⊤ ⊔ ∀⊤$_R$.∀(S⊓¬R).⊥⊒⊤	⊥,⊔,∃,∀,⊤$_R$,¬$_R$,⊓$_R$
disj(A,B)	¬A⊔¬B⊒⊤	¬,⊔
disj(R,S)	∀(R⊓S).⊥⊒⊤	⊥,∀,⊓$_R$
A(a)	¬{a}⊔A⊒⊤	¬,⊔,{…}
R(a,b)	∃R.{b}⊔¬{a}⊒⊤	¬,⊔,∃,{…}
a=b	¬{a}⊔{b}⊒⊤	¬,⊔,{…}
a≠b	¬{a}⊔¬{b}⊒⊤	¬,⊔,{…}

apply to role terms; the other operators apply to concept terms or individuals, depending on the context. Likewise, connectives apply to concepts, roles or individuals, depending on the context. The symbol ⊤$_R$ refers to the top role, i.e., the role connecting every individual to every individual and the connective \⊏ stands for non-proper-inclusion. The symbols ¬ and ∃ refer to full (rather than atomic) negation and full (rather than limited) existential quantification respectively.

The above analysis shows that, if the DL under question contains the operators necessary for the transformations of table 1, then it is AGM-compliant. The required operators are the constant ⊤ and the connective ⊑ for the basic case (lemma 1), the operators of table 1 for the transformation of K plus the operator ∀ and the constant ⊤$_R$ for the transformation of X. Notice that there is a certain amount of redundancy in table 1; by eliminating this redundancy the following corollary can be shown:

Corollary 1. A DL containing the constants ⊤, ⊤$_R$, the operators ¬, ⊓, ∀, ¬$_R$, ⊓$_R$, {…}, the concept connective ⊑ plus any connectives from table 1 is AGM-compliant.

3.2 Discussion

Notice that corollary 1 provides one possible application of theorem 1; the family of DLs described by corollary 1 is not the only AGM-compliant one. There are several reasons for that: first of all, the transformations we propose are probably not the only possible ones. Other transformations for K and X would possibly generate a different set of operators required for AGM-compliance.

For example, an alternative for the transformation of X is to take $X'=\{\forall R.B \sqsupseteq \top\}$ for a "fresh" role name R instead of $X'=\{\forall \top_R.B \sqsupseteq \top\}$. The validity of this alternative can be easily shown using model-theoretic arguments. Notice that this requires a fresh role name, so the logic must admit an infinite number of role names. Additionally, this alternative transformation introduces roles which are completely irrelevant to the original KB and the contracted expression; the introduction of new, irrelevant roles during each contraction operation may appear irrational for some applications, despite the fact that it results to an AGM-compliant operation.

Additionally, corollary 1 gives a *minimal* set of operators that are needed to guarantee AGM-compliance. Any additional operators do not bar AGM-compliance (notice however that any additional connectives might). Thus, all logics that contain more operators than the DL described in corollary 1 are AGM-compliant too.

Furthermore, some of the operators could be replaced by others; for example the combination $\{\neg, \forall\}$ is equivalent to the combination $\{\neg, \exists\}$. Similar facts hold for other operators as well. Moreover, the constants $\top$ and $\top_R$ could be removed from the minimal required set of operators, because they can be replaced by $A \sqcup \neg A$ and $R \sqcup \neg R$ respectively. Of course, this requires that there is at least one concept (A) and at least one role (R) in the namespace of the logic, but this is hardly an assumption.

As it is clear by theorem 1, the operators we need to guarantee AGM-compliance are just those that are required to produce the sets K′, X′; for example, if we are interested in DL KBs without an Abox, then the operator {…} is not necessary, i.e., it could be removed from the minimal set of operators required for AGM-compliance. Similarly, certain logics disallow certain connectives or certain uses of ones. Such restrictions might affect (i.e., reduce) the required minimal operator set (by allowing simpler transformations). Furthermore, in some DLs it might be the case that for all concept terms B there is an interpretation I such that $B^I=\emptyset$; if this is the case, then the last transformation for X ($\forall \top_R.B \sqsupseteq \top$) is not necessary and we could set $X'=\{B \sqsupseteq \top\}$.

In theorem 1 we state that the DL under question must allow for concept hierarchies (connective $\sqsubseteq$). This is a reasonable assumption, since most interesting DLs do satisfy it. However, it turns out that it is also an unnecessary one. To show that, we will use the concept of *equivalence of logics with respect to AGM-compliance* that appeared in [6], where it was shown that equivalent logics have the same status as far as AGM-compliance is concerned. Now, using model-theoretic arguments, we can show the following equivalences: $\{A \sqsupseteq \top\} \Leftrightarrow \{A \cong \top\} \Leftrightarrow \{\neg \forall \top_R.A \sqsubseteq \top\} \Leftrightarrow \{\neg \forall \top_R.A \not\cong \top\} \Leftrightarrow \{\neg \forall \top_R.A \not\sqsupseteq \top\} \Leftrightarrow \{A \setminus \sqsubseteq \top\} \Leftrightarrow \{disj(\top, \neg A)\}$. These equivalences are all definable using the minimal set of operators of corollary 1. Using

these transformations and proposition 5 in [6], we can show that a DL that contains the operators required by corollary 1 plus any of the usual concept connectives ($\cong$, $\sqsubset$, $\not\cong$, $\not\sqsubseteq$, $\backslash\sqsubset$, disj(.,.)), but not $\sqsubseteq$, is equivalent to a similar DL that contains the same operators and connectives as well as the connective $\sqsubseteq$. The latter logic (which includes $\sqsubseteq$) is AGM-compliant by corollary 1; thus the original logic (which does not include $\sqsubseteq$) is AGM-compliant too (since the two logics are equivalent). This argumentation shows that the existence of concept hierarchies in the DL under question is not mandatory for corollary 1 to be applicable; any of the usual concept connectives would do.

The AGM-compliance of a certain family of DLs is the primary result of this section; however, the constructive proof employed in theorem 1 has the secondary effect of suggesting one possible contraction operator that satisfies the generalized AGM postulates. Indeed, if $Cn(\emptyset) \subset Cn(X) \subset Cn(K)$, the principal case in an AGM-compliant contraction operation, then by setting $K-X=Cn(Z)$, where Z is the set constructed in the proof of lemma 1, we get an AGM-compliant result for the contraction. This set Z can be constructed in linear time on the number of axioms in K, X (this is obvious; see the proof of lemma 1). We can complete the definition of the contraction operator for the non-principal cases as follows: if $Cn(X) \not\subseteq Cn(K)$, then $(K-3)$ leaves us little choice: $K-X=Cn(K)$; if $Cn(X)=Cn(\emptyset)$ then $(K-6)$ implies $K-X=Cn(K)$; finally, if $Cn(K)=Cn(\emptyset)$ or $Cn(K)=Cn(X)$, then $K-X=Cn(\emptyset)$ is a valid choice. These results can be computed in constant time. Thus, given an oracle that solves the reasoning problem of the underlying DL in constant time, the result of this contraction operator can be computed in linear time on the number of axioms in K, X. So, the computational bottleneck of the above contraction operator is the inference problem of the underlying DL. However, the semantic properties of this operator have not been studied; this is reserved for future work.

Many of the required operators of corollary 1 are standard in most interesting DLs. One exception is the operator {...}, which is common in many DLs, but could not be classified as "standard". Fortunately, this operator is not necessary for AGM-compliance if we assume an empty Abox in the DL under question. A more important problem is posed by the role operators ($\neg_R$, $\sqcap_R$, $\top_R$), which do not appear in most DLs. These operators are required when role connectives are admitted and for the transformation of X, unless we use the alternative transformation with the fresh role name. Thus, role operators are not necessary if axioms involving roles are not allowed in the DL under question and an alternative transformation for X is available.

One last (but certainly not least) observation that can be made is that theorem 1 and its various corollaries do not provide a complete characterization of AGM-compliant DLs. However, it looks like this characterization is close to being complete: all the AGM-compliant DLs that we have considered fall into one of the theorem's innumerable variations and corollaries; those who don't, eventually turn out to be non-AGM-compliant (see the next sections for some examples). It is part of our future work to determine whether this pattern is simply coincidental or not.

4   Conditions for Non-AGM-Compliance

Unfortunately, many DLs are not AGM-compliant; to show that, we will initially show the following simple lemma, which is applicable in any logic:

Lemma 3. Consider a logic <L,Cn> and a set K⊆L. Set X={x∈L | Cn({x})⊂Cn(K)}. If Cn(∅)⊂Cn(X)⊂Cn(K) then <L,Cn> is not AGM-compliant.

Proof. Take any set Z⊆L such that Cn(Z)⊂Cn(K). Then, obviously Z⊂X, so Cn(X∪Z)=Cn(X)⊂Cn(K). Thus, we can find no Z⊆L as required by proposition 1 for the sets K,X⊆L, which proves that <L,Cn> is not AGM-compliant.   □

Lemma 3 states that, if a logic contains a belief which cannot be deduced by all its proper consequences combined, then this logic is not AGM-compliant. Unfortunately, this is the case for many DLs that admit axioms between role terms but forbid the use of operators $\neg_R$, $\sqcap_R$. Indeed, the axiom R⊑S implies ∃R.A⊑∃S.A, $(\leq_2 R)\sqsupseteq(\leq_2 S)$, etc, but sometimes all such implications combined do not imply R⊑S, as shown below:

Theorem 2. Consider a DL with the following properties:

- The DL admits at least two role names (say R, S) and one concept name (say A)
- The DL admits at least one of the operators ∀, ∃, $(\geq_n)$, $(\leq_n)$, for at least some n
- The DL admits any (or none) of the operators ¬, ⊓, ⊔, ⁻, ⊤, ⊥, {…}
- The DL admits only the connective ⊑ applicable to both concepts and roles

Then this DL is not AGM-compliant.

Sketch of Proof. Set K={R⊑S}, X={x∈L | Cn({x})⊂Cn(K)}, as in lemma 3. We define two interpretations I, I′, as follows:

$\Delta^I=\Delta^{I'}=\{a_1,a_2,b_1,b_2,c\}$
$B^I=B^{I'}=\emptyset$ for all concepts B
$y^I=y^{I'}=c$ for all individuals y
$R_0{}^I=R_0{}^{I'}=\emptyset$ for all roles R_0, other than R, S
$R^I=R^{I'}=\{(a_1,b_1), (b_1,a_1), (a_2,b_2), (b_2,a_2)\}$
$S^I=\{(a_1,b_1), (b_1,a_1), (a_2,b_2), (b_2,a_2)\}$
$S^{I'}=\{(a_1,b_2), (b_2,a_1), (a_2,b_1), (b_1,a_2)\}$

Notice that the two interpretations differ only in the interpretation of the role S. An easy induction on the number of operators of a concept term C shows that $C^I=C^{I'}$ for all C in all DLs considered by the hypothesis. Thus, any axiom involving concept terms is satisfied by I iff it is satisfied by I′. Using induction, we can also show that all axioms in X that involve role terms are actually tautological.

Thus, I satisfies K (obviously), so it satisfies X (because K⊨X); since I satisfies X, I′ satisfies X (by the results above), but K is not satisfied by I′ (obviously). Thus Cn(X)⊂Cn(K). To complete the proof, we need to show that Cn(X)≠Cn(∅); this follows from the fact that at least one of the operators ∀, ∃, $(\geq_n)$, $(\leq_n)$ (for some n) exists in the DL. The above, combined with lemma 3, conclude the proof.   □

The above negative result persists if the DL under question admits $\cong$ (applicable to both concepts and roles) instead of $\sqsubseteq$, or if it admits both connectives; the proof is identicsal. The same result can be shown (using the same proof) if we add transitive roles (the axiom Trans(.)), axioms with individuals, functional-only roles and/or qualified number restrictions. This analysis uncovers a rule of thumb regarding DLs: if theorem 1 cannot be applied, then there is good chance that lemma 3 will be applicable for a set of the form $\{R \sqsubseteq S\}$ or $\{R \cong S\}$ (for any two roles R, S). This provides a simple test to determine whether a DL is AGM-compliant, applicable to many DLs.

5   A Case Study: OWL

5.1   OWL DL and OWL Lite Without Annotation Features

One of the corollaries of theorem 2 is that OWL DL and OWL Lite without annotation features are not AGM-compliant. To show this, we will use the result of [10] that identifies OWL DL and OWL Lite (without annotations) as equivalent to $SHOIN^+(D)$ and $SHIF^+(D)$ respectively. $SHOIN^+(D)$ is a very expressive DL allowing for the following operators: $\top, \bot, \sqcap, \sqcup, \neg, \{\ldots\}, ^-, \exists, \forall, (\geq_n), (\leq_n)$. In addition, it allows a *datatype theory* (D), which is a mapping from a set of datatypes to a set of values plus a mapping from data values to their denotation (see [10] for details). To make datatypes useful, the logic also allows *datatype* (or *concrete*) *roles*, which are binary relationships between individuals and data values, as well as the operators $\exists, \forall, (\geq_n), (\leq_n), \{\ldots\}$ for datatype roles and data values. The axioms allowed in this logic are concept, role and datatype role hierarchies; individual inclusion, equality and inequality; role transitivity (for object roles only); and a new *concept existence* axiom (see [10]). $SHIF^+(D)$ is just $SHOIN^+(D)$ without the $\{\ldots\}$ constructor and with the at-least and at-most constructors limited to 0 and 1. $SHOIN^+(D)$ and $SHIF^+(D)$ can be shown non-AGM-compliant, so OWL DL and OWL Lite are not AGM-compliant either:

Corollary 2. $SHOIN^+(D)$ and $SHIF^+(D)$ are not AGM-compliant.

Proof. The only difference from the proof of theorem 2 is the existence of datatypes; to remedy this problem, augment the interpretations with a datatype domain $(\Delta_D^I, \Delta_D^{I'})$ and map all datatype roles to the empty set. The rest of the proof is identical. $\square$

5.2   OWL with Annotation Features

But what if annotation features are included? Does this make the situation any better? Unfortunately not: the annotation features are meant to be read by humans, so they carry no special meaning for the system (they imply nothing) and the same negative results apply here. There is one exception though: the owl:imports annotation feature carries some meaning for the parser, making it substantially different from the other annotation constructs. More specifically, owl:imports is a meta-logical annotation property forcing the parser to include another KB (ontology) in the current KB. In effect, the axiom owl:imports(O) has exactly the same implications as O itself. One may

believe that owl:imports does not add anything new to the language in terms of expressiveness, because owl:imports(O) can be replaced by the axioms of O themselves.

Unfortunately, there is one problem with this approach: owl:imports must be replaced dynamically at the time when the consequences of a certain KB are calculated (at "run-time"). However, the naïve approach above replaces owl:imports statically, thus losing the connection between our ontology and O. This would work nicely until O is changed; if O is somehow revised, then the correct, dynamic approach should replace owl:imports with the axioms of the new O, while the static approach would leave our KB with the axioms of the old O. Notice that this would not be an issue if we could somehow guarantee that O would remain static and unchanged; however, given the dynamic nature of the Web, such an assumption would be highly unrealistic.

Therefore, the axiom owl:imports(O) is not equivalent to O; rephrasing this fact in the terminology of this paper, we conclude that, for K={owl:imports(O)}, it holds that $Cn(K)=Cn(O) \cup K$ and $Cn(O) \subset Cn(K)$, thus making lemma 3 applicable for K. Since owl:imports is allowed in all three flavors of OWL, we conclude that OWL Full, OWL DL and OWL Lite (with annotation features) are non-AGM-compliant. Furthermore, this analysis shows that any fragment of OWL that contains the owl:imports construct and at least one other non-tautological expression is non-AGM-compliant.

6   Discussion and Directions for Future Work

6.1   Application to DLs in the Literature

Our study was kept at a fairly abstract level; we did not focus on any specific DL but dealt with the DL family as a whole, including DLs that have not yet been considered in the literature. This approach allows our results to be of use to researchers who develop new DLs; if the focus is on developing a DL that can be rationally updated, then AGM-compliance should be a desirable feature of the new DL, along with high expressive power, low reasoning complexity etc.

However, theorems 1 and 2 can be applied to several DLs that have already been considered in the literature as well. We provide an indicative (but not necessarily complete) list of DLs for which a definite answer regarding AGM-compliance can be given. For a definition of the logics below, refer to [2], [3], [5], [10], [11], [14].

The following DLs can be shown to be non-AGM-compliant: SH, SHI, SHIN, SHOIN, SHOIN(D), SHOIN$^+$, SHOIN$^+$(D), SHIQ, SHIF, SHIF(D), SHIF$^+$, SHIF$^+$(D); all these logics admit role hierarchies, so these results are actually corollaries of theorem 2. For similar reasons, adding role hierarchies to the AL family leads to non-AGM-compliance; that is, FL_0 and FL^- with role axioms and all DLs between ALH and ALHCIOQ are non-AGM-compliant. This family includes several logics, such as ALHE, ALHNC, etc. None of the three flavors of OWL is AGM-compliant if the owl:imports axiom is included; OWL DL and OWL Lite without their annotation features are non-AGM-compliant either. These facts were proven in section 5.

The addition of role operators to the AL family results in some AGM-compliant DLs, such as ALCO$^{\neg,\sqcap,\sqcup}$, ALC$^{\neg,\sqcap,\sqcup}$ with empty Abox, ALCO$^{(\neg),\sqcup}$ and ALCO$^{(\neg),\sqcap,\sqcup}$

with no axioms involving role terms and $ALC^{(\neg),\sqcup}$ and $ALC^{(\neg),\sqcap,\sqcup}$ with empty Abox and no axioms involving role terms. AGM-compliance persists if we add more operators (but not new axiom types) to any of the above logics; for example, all DLs with more operators than $ALC^{\neg,\sqcap,\sqcup}$ and no Abox (such as ALB) are AGM-compliant.

If we have an infinite pool of role names, we can use the alternative transformation introduced in section 3.2 to produce X′; this makes ALC (and all languages with more operators than ALC) AGM-compliant, provided that no axioms involving role terms are included and that the Abox is empty. Similarly, all languages with more operators than ALCO are AGM-compliant if they do not allow axioms involving roles.

As shown by the above results, it is the absence of role operators (role intersection, union and complement) that bars AGM-compliance in most cases. For this reason, we highly encourage research on DLs that admit these operators due to their nice behavior with respect to updates. Unfortunately, very few logics with role operators have been studied in the literature (notable exceptions being [11], [14]), so the computational overhead caused by such operators is largely unknown.

6.2   Role Operators, Negation, the Levi Identity and Ontology Revision

An additional advantage of the use of role operators (especially $\top_R$) is the fact that they allow the definition of an *axiom's negation*. The negation of $A \sqsubseteq B$ is $A \not\sqsubseteq B$, but most logics do not allow axioms with the connective $\not\sqsubseteq$. However, $A \not\sqsubseteq B$ is equivalent to $\exists \top_R.(A \sqcap \neg B) \sqsupseteq \top$ (see table 1), so the negation of $A \sqsubseteq B$ can be defined indirectly using $\top_R$; similar facts hold for other axiom types as well. This concept can be extended to finite sets of axioms by noticing that the set $X=\{A_j \sqsupseteq B_j \mid j \in J\}$ is equivalent to $\{\sqcap_{j \in J}(A_j \sqcup \neg B_j) \sqsupseteq \top\}$, which is a singular set, so it has a negation, as above.

It must be emphasized at this point that not all AGM-compliant DLs are closed with respect to axiom negation. The negation of a set of axioms in an AGM-compliant logic, when available, is a very important concept, because it allows us to use the Levi identity: $K+X=Cn((K-\neg X) \cup X)$ to produce a revision operator from a given contraction operator [8]. This identity says that, in order to revise a KB with some set of axioms X, we can first contract $\neg X$ and then add X. The contraction operation is needed to guarantee that no inconsistency will arise when X is added to the new KB.

Therefore, for these logics, the problem of ontology revision can be solved indirectly through the problem of contraction, which is studied in this paper; this way, the definition of a rational contraction operator is of dual significance. As future work, we are planning to study the problem of revision more thoroughly. Due to the above facts, a related issue is the refinement of the proposed contraction operator for AGM-compliant DLs, to produce an operator that will be based on semantic rather than syntactic considerations, in addition to being AGM-compliant.

6.3   Evaluation of AGM-Compliance

The purpose of this paper is to evaluate the usefulness of applying the AGM theory to DLs and OWL. As the above analysis indicates, OWL does not support the AGM

postulates, so the approach is not useful for OWL ontologies. Regarding DLs, the situation is much better: there are certain DLs in which an AGM-compliant contraction operator can be defined, as well as several non-AGM-compliant DLs. Our results do not currently provide a complete characterization of AGM-compliance for DLs; this is an important goal for our future work.

The AGM theory has always been the most influential approach to the problem of belief change, because it properly captures common intuition regarding the concept of rationality and it has several interesting theoretical properties [8]. However, our results showed that there are certain problems regarding its application to certain DLs, for the operation of contraction. On the other hand, as we showed in [6], all logics <L,Cn> admit a contraction operator that satisfies (K−1)-(K−5), i.e., all AGM postulates except the recovery postulate. Coincidentally, the only seriously debated AGM postulate is the postulate of recovery, as some works (e.g., [9]) state that (K−6) is counter-intuitive; for a thorough examination on the theoretical implications of using (K−6) see [15]. It is generally acceptable however that the recovery postulate cannot be dropped unless replaced by some other constraint that would somehow express the Principle of Minimal Change. Given the negative results appearing in this paper and the above facts, we believe it is useful to work on a "replacement" of the recovery postulate, or on some approximation of it, that would properly capture the Principle of Minimal Change in addition to being applicable to non-AGM-compliant DLs.

As far as AGM-compliant DLs are concerned, we believe that research on ontology change should use the feature of AGM-compliance, thus taking advantage of the numerous results that appeared in the literature on belief change and the AGM theory during the past 20 years. For this reason, we plan to continue our research on the application of the AGM theory to the DLs that support it. In this respect, notice that AGM-compliance simply guarantees the existence of a contraction operator that satisfies the basic AGM postulates for contraction; one of our future goals is to determine the relation of AGM-compliance to other results related to the AGM theory, such as the various representation theorems [8], the supplementary AGM postulates [1] etc.

7 Conclusion

The AGM theory is a mature and widely accepted model for belief change with several applications; a further application of this theory in DLs will hopefully indicate rational methods for updating such logics. This paper partly evaluated the applicability and usefulness of this approach by determining whether contracting a DL KB using the AGM model is possible for certain DLs and by providing a roadmap allowing one to check AGM-compliance for DLs not covered by this work. We also described one possible AGM-compliant contraction operator for the DLs that were found to allow one and showed that OWL is incompatible with the AGM theory.

We are hoping that our work will help in uncovering the limitations of the AGM theory with respect to DLs, by verifying the applicability of the method in certain

DLs and forcing us to consider alternative approaches in others. DLs and OWL have an important role to play in the design of the Semantic Web [4], so our research has the potential to find applications in ontology evolution and merging and,consequently, in the automation of the task of ontology maintenance on the Semantic Web.

Acknowledgments

References

1. Alchourron, C., Gärdenfors, P., Makinson, D.: On the Logic of Theory Change: Partial Meet Contraction and Revision Functions. Journal of Symbolic Logic 50 (1985) 510-530
2. Baader, F., Sattler, U.: An Overview of Tableau Algorithms for Description Logics. Studia Logica 69 (2001) 5-40
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press (2002)
4. Baader, F., Horrocks, I., Sattler, U.: Description Logics as Ontology Languages for the Semantic Web. In Hutter, D., Stephan, W. (eds.): Festschrift in honor of Jörg Siekmann, Lecture Notes in Artificial Intelligence, Springer-Verlag (2003)
5. Dean, D., Schreiber, G., Bechhofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., McGuiness, D., Patel-Schneider, P., Stein, L.A.: OWL Web Ontology Language Reference. W3C Recommendation (2004). Available at URL: http://www.w3.org/TR/owl-ref/
6. Flouris, G., Plexousakis, D., Antoniou, G.: AGM Postulates in Arbitrary Logics: Initial Results and Applications. Technical Report FORTH-ICS/TR-336, April 2004
7. Flouris, G., Plexousakis, D., Antoniou, G.: Generalizing the AGM Postulates: Preliminary Results and Applications. In Proceedings of the 10th International Workshop on Non-Monotonic Reasoning (2004) 171-179
8. Gärdenfors, P.: Belief Revision: An Introduction. In Gärdenfors, P., (ed.) Belief Revision, Cambridge University Press (1992) 1-20
9. Hansson, S. O.: Knowledge-level Analysis of Belief Base Operations. Artificial Intelligence 82 (1996) 215-235
10. Horrocks, I., Patel-Schneider, P.: Reducing OWL Entailment to Description Logic Satisfiability. Journal of Web Semantics, 1(4) (2004) 345-357
11. Hustadt, U., Schmidt, R.A.: Issues of Decidability for Description Logics in the Framework of Resolution. In Automated Deduction in Classical and Non-Classical Logics, vol. 1761, LNAI, Springer, (2000) 191-205
12. Kang, S.H., Lau, S.K.: Ontology Revision Using the Concept of Belief Revision. In Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (2004) 8-15
13. Lee, K., Meyer, T.: A Classification of Ontology Modification. In Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (2004) 248-258, Cairns, Australia
14. Lutz, C., Sattler, U.: Mary Likes All Cats. In Proceedings of the 2000 International Workshop in Description Logics (2000) 213-226

15. Makinson, D.: On the Status of the Postulate of Recovery in the Logic of Theory Change. Journal of Philosophical Logic 16 (1987) 383-394
16. Meyer, T., Lee, K., Booth, R.: Knowledge Integration for Description Logics. In Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning (2005)
17. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-driven Ontology Evolution Management. In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, vol. 2473, LNCS, Springer-Verlag, (2002) 285-300