A Finite Base Belief Revision System

Simon Dixon
Knowledge Systems Group
Basser Department of Computer Science
University of Sydney, NSW 2006
Australia

e-mail: simon@cs.su.oz.au

ABSTRACT

Theories of dynamic reasoning invariably are more elegant on paper than they are in an implementation. A major criticism of the AGM belief revision logic [Gärdenfors 1988] is that it relies on calculations using infinite belief sets to satisfy its postulates for rational belief revision. This has stopped it from being used in any practical systems for reasoning. [Williams 1993] proposes a theory of belief revision which uses finite theory bases to represent the belief sets, and an ordered partition of the base, called the ensconcement ordering, to represent the epistemic entrenchment ordering of the AGM model. Williams defines operators for expansion, contraction and revision which correspond to the AGM operators, and shows that these achieve a "best approximation" to the AGM logic, for a finite reasoning agent. We describe the implementation of Williams' finite base belief revision theory, and discuss the practical issues involved in implementing a system which is guaranteed to terminate in reasonable time. We show some simple test cases, and give a brief analysis of the system's performance on these examples.

1. Introduction

Traditionally, most automated reasoning systems have been based on some variant of propositional or first-order logic. But logic-based reasoning is limited by the undecidability of first order logic, so that it is not possible to build a system which is both correct and efficient, without restricting the nature of the logic. Completeness must be sacrificed if we want to guarantee termination of the algorithm. We have two mutually exclusive goals: the theoretical, to propose an elegant reasoning system (which may have no elegant implementation), and the pragmatic, to create a useable product (which may require us to simplify the theoretical solutions). Our aim, with many others, is to attempt to bridge the gap between these two ideals, and present a workable product which has strong ties to an accepted theoretical model of reasoning.

1.1 Belief Revision

The AGM logic of belief revision has attracted considerable attention as a candidate for a general theory of reasoning. It tries to model how an agent updates its state of belief (epistemic state) as a result of receiving new information. An epistemic state is represented by a belief set (a set of sentences from a given language, in this case first order logic, which is closed under logical consequences), and an epistemic entrenchment ordering (a linear order on all sentences in the language). The dynamic behaviour is achieved by three operators which map epistemic states to belief sets, namely expansion, contraction and revision. These operators, which model rational change of belief, conform to a set of rationality postulates [Gärdenfors 1988], but also have constructive definitions based on epistemic entrenchment.

Recently, a number of translations between the AGM logic and other forms of nonmonotonic reasoning have appeared in the literature. [Gärdenfors 1990] claims that the AGM postulates effectively satisfy every known postulate in the literature on nonmonotonic inference operators, and conversely, all of the AGM postulates are satisfied by some kinds of nonmonotonic inference operators in the literature. [Peppas 1992] defines a mapping between the AGM logic and systems of spheres [Grove 1988], and also between these systems and nice pre-orders on models. Also, [Gärdenfors 1991] describes the relationship between the default reasoning of [Poole 1988] and the AGM logic. These translations to and from different nonmonotonic reasoning systems justify our interest in implementing a system based on the AGM logic.

1.2 Finite Bases

A major problem with such an implementation is that it is not possible in practice to perform the full theory closure which the AGM paradigm requires, since logical closure generally results in infinite theories. [Williams 1993] argues that it may not even be desirable to do so, for modelling a limited reasoning agent. The best solution is to represent the theories by finite bases, which are subsets of the theory which, when closed, yield the full theory. All of the theory change operations will be performed on the theory base, with the aim that the resulting belief set should be a base of the theory which would have been obtained had it been possible to calculate it.

1.3 Overview

In section 2, we give a brief description of the AGM paradigm, and discuss some complexity considerations which are faced in using this logic. The third section describes a solution to the complexity limitations of the AGM logic, using finite theory bases to represent the infinite theories, and a finite ordering on the elements of the base which represents the epistemic entrenchment ordering on the theory. In the following section, the implementation of this finite base belief revision system is presented, including an outline of the data structures, algorithms, and techniques used in designing the system. Section 5 analyses the performance of the system on some simple examples. The paper concludes with a discussion of the significance of these results, and proposes several areas of further research.

2. The AGM Belief Revision Paradigm

The AGM logic of belief revision [Gärdenfors 1988] describes the change in epistemic state when a belief that may or may not be consistent with current beliefs is added to or retracted from the belief set. In the AGM logic, the current belief set is represented by a set of first order sentences, denoted K, which is closed with respect to the classical consequence operator, Cn. That is, $K = Cn(K)$. There is also an ordering on the sentences of the language, called the epistemic entrenchment, which represents the relative values of each sentence to the current epistemic state.

Three basic operators are used to modify the belief set, these are expansion, contraction and revision. The expansion of the belief set by a sentence α, denoted K^+_α, is the acceptance of α into K, regardless of whether or not it is inconsistent with the current belief set. Contraction, denoted K^-_α, removes the sentence α from the belief set, together with any other sentences which must be removed to maintain the consistency and closure of the belief set. Finally, revision involves the acceptance of a belief α into K such that the resulting belief state, denoted K^*_α, is consistent. This is achieved by removing any beliefs in K which contradict α, and then adding α into the belief set. This is stated formally in the Levi identity: $K^*_\alpha = (K^-_{\neg\alpha})^+_\alpha$. Its converse, $K^-_\alpha = K \cap K^*_{\neg\alpha}$, is called the Harper identity.

2.1 Rationality Postulates

A non-constructive formulation of the AGM belief change operators is via a set of rationality postulates. These postulates ensure that the operations are successful, syntax independent, and consistent whenever it is possible, and result in a logically closed belief set. The operators are also required to make only minimal changes to the overall belief set.

The postulates define an expansion to be the set of consequences of the original belief set with the new belief added to it. That is $K^+_\alpha = Cn(K \cup \{\alpha\})$. Contraction and revision, however, are not uniquely defined by their postulates, for there are in general many contraction and revision functions on a belief set which satisfy the AGM postulates.

2.2 A Constructive Formulation via Epistemic Entrenchment

For this reason, an ordering, called epistemic entrenchment, is placed on the sentences in the language, to determine which beliefs are preferred to remain in the belief set during a contraction or revision operation. The axioms which entrenchment must obey are found in [Gärdenfors 1988], and it is shown that given an entrenchment on a belief set, a contraction function may be defined uniquely by the (C-) Condition:

$\beta \in K^-_\alpha$ iff $\beta \in K$ and either $\vdash \alpha$ or $Ent(\alpha) < Ent(\alpha \vee \beta)$

The revision function can then be uniquely defined via the Levi identity.

A major weakness of the AGM logic is that the belief revision operators map a belief set with an entrenchment ordering to a belief set. That is, in the resulting epistemic state, the new entrenchment relation is not specified, so it is not possible to perform two successive contraction or revision operations. Neither is it possible to form the composition of two operations, unless the second operation is expansion, which is independent of the entrenchment relation. The only attempt to solve this problem is the Ordinal Conditional Functions of [Spohn 1988], which work in a much more general framework than the AGM logic. In Spohn's work, the epistemic state is represented by a function which gives an ordering on the models of the language, and all epistemic changes - expansion, contraction, revision and change of entrenchment without changing the theory - are performed by the same operator, with differing parameters.

2.3 Complexity Considerations

In implementing a belief revision system based on the AGM logic, the first problem to be faced is choosing a finite representation for the generally infinite belief set. If the theory is finitely axiomatisable, then it is possible to represent the theory by a finite base, from which all of the other beliefs can be derived via the consequence operator. The first complexity issue to be faced is the difficulty of calculating all of the beliefs which logically follow from the theory base. Theory membership is calculated by adding the negation of the sentence to be tested into the belief set, and then using resolution to determine whether the resulting set of sentences is satisfiable. If it is not satisfiable, then the sentence whose negation we inserted into the belief set is derivable from the finite base. Resolution is sound and complete, so we are guaranteed the correctness of any results obtained by this method, but satisfiability is NP-complete in the propositional case, and undecidable in the first order case. That is, we cannot guarantee termination of our algorithm, without sacrificing completeness. In the implementation, we introduce a way of forcing termination by limiting the number of times a clause can be used in a proof. For different applications, this number can be varied, but in general, it means that the system cannot create or use arbitrarily long terms.

The second representation problem in the AGM logic is that of the epistemic entrenchment. The entrenchment is an ordering on all sentences in the language. It is noted in [Gärdenfors 1988] that the entrenchment relation can be completely specified by a linear order on the dual atoms of the language (disjunctions containing every atom in the language as either a positive or negative atom). This reduces the number of relations to 2^n, where n is the number of atoms in the language, which is still too many to represent efficiently. We use a finite ordering from [Williams 1993], called ensconcement, which is an ordered partition of the finite base of the theory. It has been shown that this representation is equivalent to the entrenchment relation, as long as the entrenchment relation is finitely representable.

The next consideration is the complexity involved in performing each of the AGM operations. Expansion is very simple: a belief and its consequences are added to the belief set. As discussed above, we are not able to calculate all of the consequences of a belief, so we just add the belief itself to the theory base.

Contraction and revision are more involved. To perform a contraction, using the (C-) Condition given in section 2.2, requires that we test, for every element of a possibly infinite belief set, how its entrenchment compares with that of its disjunction with the sentence being removed. This is the best constructive definition of contraction, and yet it does not provide a computable algorithm. We present an alternative contraction operator on finite bases which satisfies most of the AGM postulates, but, in particular, fails to satisfy the recovery postulate (K-5):

$$K \subset (K^-_\alpha)^+_\alpha$$

The revision operator has similar complexity to the contraction operator, as it is defined by the Levi identity as the composition of a contraction and an expansion operation.

3. A Theory of Finite Base Belief Revision

[Williams 1993] presents a theory of finite base belief revision which satisfies "as many as possible" of the AGM postulates. The basic idea behind this theory is to represent the belief set by a finite set of sentences, from which the rest of the theory can be derived by the normal first order consequence relation, and to represent the entrenchment ordering by an ordered partition of the sentences in the base. The ordering on the belief base is called an ensconcement ordering, and the resulting level of entrenchment of a sentence, which can be calculated from the ensconcement, is called its rank.

In order to be consistent with the AGM entrenchment relation, the ensconcement must obey the following axiom:

$$\text{For all } \beta\in\Gamma,\ \{\alpha\in\Gamma : \beta<\alpha\} \nvdash \beta$$

The rank of a sentence in the base is equal to its ensconcement, and the rank of a sentence outside the base is equal to the lowest ensconcement of all the sentences in the subset of the base from which the sentence is derived. If there is more than one such derivation, then the highest ranking derivation is taken. If there is no such derivation, then the sentence is outside the belief set, and thus it has the minimum entrenchment (rank = 0), in agreement with the AGM model.

3.1 Formulations of Finite Base Operators

The revision and contraction operators obey a set of rationality postulates which correspond very closely to the AGM postulates. In particular, the expansion of a belief base is defined to be the addition of the sentence to the base, but the contraction and revision, are not constrained to unique functions by the postulates. As is necessary in the AGM paradigm, the extra structure from the ordering on sentences is required to provide unique contraction and revision operators. Given the ensconcement ordering, there are explicit constructions for the contraction and revision of a theory base, similar to the (C-) Condition and Levi identity. These are the ($C^{\ominus}$) Condition:

$$\beta\in\Gamma^{\ominus}_{\alpha} \quad \text{iff} \quad \beta\in\Gamma \quad \text{and} \quad \text{either} \quad \bigcup_{i=rank(\alpha)+1}^{n} \Gamma_i \cup \{\neg\alpha\} \vdash \beta, \text{ or} \vdash\alpha$$

and the finite base Levi identity:

$$Cn(\Gamma^{\otimes}_{\alpha}) = (\Gamma^{\ominus}_{\neg\alpha})^{+}_{\alpha}$$

3.2 Closure and Recovery

As described earlier, in implementing a finite base belief revision system, it is not possible to calculate the full closure of the belief set. The result of this is that beliefs which are not in the base but are derivable from it may get "lost" when the base is contracted and they are no longer derivable. In other words, the contraction operator does not guarantee minimal change in the same way that the AGM contraction operator does.

This means that the recovery postulate (AGM postulate K-5) is not satisfied. The postulate states that a contraction followed by an expansion by the same sentence recovers all of the sentences from the original theory. This guarantees the conservativity of contractions; nothing may be removed unnecessarily from the belief set during a contraction. The first postulate for finite base contraction, which requires the contraction of a base to be a subset of the base, is inconsistent with the recovery postulate, and in order to regain recovery, it has been shown [Nebel 1990] that other sentences from outside the base must be added to the theory base.

3.3 Comparison with AGM Paradigm

We have already seen that the finite base system does not satisfy the AGM closure postulate (K-1), nor the recovery postulate (K-5). Another difference between the finite base model and the AGM paradigm is that the finite base contraction and revision operators depend on the syntax of the base, whereas the corresponding AGM operators are syntax-independent. We claim that this is a necessary product of using a finite base and an ensconcement ordering. That is, the syntax of the base affects the entrenchment ordering on the theory, so that two different bases for the same theory will generally represent different orderings, and hence generate different contraction and revision functions.

The finite base system satisfies the rest of the AGM postulates, which ensures a high correspondence between the AGM operators and their respective finite base model counterparts. [Williams 1993] presents two theorems which describe the exact relationship between the two systems of belief revision. The first relates the AGM and finite base contraction operations:

$$\Gamma^{\ominus}_{\alpha} = (Cn(\Gamma))^{-}_{\alpha} \cap \Gamma$$

The finite base contraction operator is therefore the nearest approximation to a base for the theory obtained by the AGM contraction operation. Because of the limited expressiveness of finite bases, it may not be a base for the contracted theory, but only a subset of the desired base. This is a direct result of postulate 1 for finite base contraction, which is:

$$\Gamma^{\ominus}_{\alpha} \subset \Gamma$$

On the other hand, the finite base revision operator corresponds exactly to the AGM equivalent:

$$Cn(\Gamma^{\otimes}_{\alpha}) = (Cn(\Gamma))^{*}_{\alpha}$$

A similar result is (trivially) true of the two expansion operators.

4. The Implementation

4.1 Motivation

The goal of the implementation was to create a program which would be guaranteed to terminate in reasonable time, and produce a correct answer, in all but the pathological cases. That is, we want a belief revision system which is practical to use in real-world domains, as well as in research.

Another aim was to discover how the ensconcement relationship could be used to achieve various styles of reasoning, and whether the restriction of the epistemic entrenchment relation to finitely representable relations makes a significant difference to the expressive power of the system. Very little has been written on the semantics of the entrenchment relation, because of the difficulty of calculating the effects of making changes to the entrenchment relation. It is hoped that by automating the system, it will be easier to understand intuitively how an ordering can be chosen which accurately represents our current epistemic state.

4.2 Syntax and Representation

The front end of the system contains a parser for the set of all quantifier-free first order expressions. The syntax is similar to that of Prolog; variables are those symbols which begin with an upper case letter, and constants those which begin with a lower case letter. The semantics are also similar, in that all variables are treated as implicitly universally quantified, and existential quantification is achieved via Skolem functions. Explicit quantifiers will be added to the system at a later date.

For the resolution theorem prover, it was necessary to convert all of the formulae into conjunctive normal form (CNF). This was done by the parser as the formula was read. The original syntax of the formula was also stored as a string, so that the output of the program could refer to the elements of the base in the same form as they were entered. This avoids the confusion that would be caused if the user failed to recognise the conversion of the formula from the base into CNF.

Note that it is not possible to separate a clause from the CNF expression in which it appears, since that would alter the ensconcement ordering, which is dependent on the syntax of the base. So although the resolution algorithm works at the clause level, the clauses always remain grouped in their conjuncts for the purpose of preserving the structure of the finite base.

4.3 Ordered Linear Resolution and Unification

It was important for the mechanics of the program to run efficiently, and to guarantee termination, or else the purpose of constructing the system would be defeated. We employed a resolution strategy called OL-deduction [Chang 1973], which is a generalisation of the Prolog-style SLD-resolution for Horn clauses, and we also used the unification algorithm from [Chang 1973].

OL-deduction is sound and complete, which guarantees correctness of all results, but does not guarantee that the program will terminate. Any method of ensuring termination also destroys the completeness of the resolution procedure. The resolution procedure tracks its own progress, so that it can detect any "suspicious" looking repetition in the use of clauses. If a proof repeats the use of a clause with the same set of variable substitutions, then it has reached an infinite loop, and the proof is terminated at that point. Alternatively, if a rule continues to be used with different variable substitutions each time, then it is possible that it has become stuck in an infinite loop, and the proof is terminated after a set number of repetitions. The termination of a proof does not mean that no other proofs are sought; the system backtracks in an attempt to find a proof from a different set of clauses. In general, since we are interested in the rank of a formula, we need to find the proof of highest overall rank. (The rank of a proof is determined by the lowest ranking clause used in the proof.) This can be done in two ways: either generate all proofs, and return the value of the maximum ranked proof, or else attempt to prove the clause using only the highest ranked formulae, and as long as the proof continues to fail, add successively lower ranked clauses, until a proof succeeds. The rank of the proof is then equal to the rank of the last set of formulae added to our current proof set.

4.4 Algorithms for Maintaining Consistency

The OL-refutation procedure assumes the consistency of the original set of clauses, or else a trivial proof of every clause exists. We consider it expedient to disallow inconsistencies by some form of error message, since the whole point of a belief revision operator is to map the current belief state to the nearest consistent belief state containing the sentence by which we are revising the theory. So as each belief is added to the base, it is checked for consistency with the base and rejected if it is not consistent. Note that revision and contraction can never introduce inconsistency, unless we try to revise the theory by the negation of a tautology, or contract a tautology from the system. It is desirable to trap these types of operations before they happen, because a decision to disbelieve something known to be true, or to believe something known to be untrue, cannot be called a rational change of epistemic state, which is what we are trying to model.

Another level of consistency exists, required by the ensconcement relation. As sentences are added to the base, it is important to check that this relation remains consistent with its definition, that:

$$\text{For all } \beta \in \Gamma,\ \{\alpha \in \Gamma : \beta < \alpha\} \nvdash \beta$$

When a revision operation is performed, the new sentence must be given a ranking in the ensconcement ordering. This may result in other sentences being moved into a higher ensconcement layer because they can be proved from sets of clauses containing the new sentence. The revision and expansion algorithms check the consistency of the ensconcement ordering as each operation is performed. In this way, consistency is maintained at both levels at all times.

Note that this resulting ordering is not required by the AGM logic. In fact, the AGM logic does not put any constraint at all on the entrenchment relation after an AGM operation. The AGM logic uses a principle of minimal change for the belief sets, and we extend this principle to apply to the entrenchment ordering, so that no beliefs change their rank unless required to do so by the definition of rank or ensconcement.

5. Examples

The program accepts input in free format, using the following notation:

~	negation
|	disjunction
&	conjunction
->	implication
f	expansion by formula *f*
n:*f*	expansion by formula *f*, setting rank(*f*) = *n*
!*f*	contraction by formula *f*
**f*	revision by formula *f*
?*f*	query formula *f*
??	print entire base

If the optional rank of a formula is not given, it defaults to the last rank value given. When the knowledge base is expanded by a formula, the formula is tested for logical consistency with the current base, and rejected if inconsistent. Then the rank of other formulae in the base are adjusted to keep them consistent with the definition of ensconcement. A warning message is printed to inform the user of any changes.

5.1 Example 1 : Revision

```
Input
    20: p(X) -> q(X)
    1: p(a)
    * ˜q(a)
    ??
Output
    Complete database:
     20 : p(X) -> q(X)
      1 : ˜q(a)
```

In this example, we revise the knowledge base by $q(a)$, which forces the removal of the lowest ranked belief from which we can derive $q(a)$, which is in this case $p(a)$.

5.2 Example 2 : Contraction

```
Input
    20: p(X) -> q(X)
    10: p(X)
    !q(c)
    ??
Output
    Complete database:
     20 : p(X) -> q(X)
```

This demonstrates what is perhaps the greatest criticism of the finite base contraction operator, that it does not necessarily perform a minimal change theory contraction. Since the contraction of a base is always a subset of that base, the contraction operator removes from the base the lowest ranked formulae from which $q(c)$ can be derived. It may be argued that it is only $p(c)$ which contributes to the proof of $q(c)$, and that the minimal change principle is violated by removing unrelated facts such as $p(a)$, $p(b)$, from the belief set.

6. Conclusions and Further Research

In this paper, we have presented a description of a finite base belief revision system which is intended to approximate the behaviour of the AGM logic. The system runs efficiently, although it does not guarantee all solutions. Much work is still to be done in analysing the system's performance on larger knowledge bases, and comparing its results to other nonmonotonic reasoning frameworks.

6.1 Revision of Orderings

One largely unexplored field of research in the AGM logic is that of entrenchment revision. The definition of the entrenchment relation is non-constructive, so that we are given no indication of how a consistent entrenchment relation may be obtained, nor how we choose a particular entrenchment relation to satisfy certain properties. [Dixon 1992a] specifies an entrenchment which allows the AGM logic to simulate the behaviour of the ATMS [de Kleer 1986], and this work is extended in [Dixon 1992b] to allow coherence-based reasoning as well. These results are only relevant for propositional Horn clauses, and there are still many open questions regarding the nature of the entrenchment relation itself.

Also, there is not yet any algorithm by which we may change the entrenchment of the beliefs in our theory, without necessarily altering the theory itself. This operation is non-trivial, since the entrenchment axioms place a number of constraints on the relations. As mentioned earlier, the entrenchment relation following an AGM operation is completely unspecified, and if we wish to model sequential belief changes, then we must develop an algorithm to revise the entrenchment as well as the theory.

A generalisation of the AGM logic which does this is Spohn's ordinal conditional functions (OCF) [Spohn 1988], which satisfy the AGM postulates, but also determine a new ordering on the models after the operation has been performed. The disadvantage of this approach is that it is all calculated on the model level, so it still remains to be mapped to the belief level. Like Spohn, we require that a sentence being added to the theory be accompanied by its level of entrenchment, but we apply a minimal change consistency maintenance algorithm to revise the ordering, whereas Spohn's OCF performs a complete recalculation of the ordering.

In future work we plan to extend the system to a full AGM implementation, and compare its performance with the finite base system. We will also be using established relationships between the AGM logic and various nonmonotonic reasoning frameworks to perform nonmonotonic reasoning with the AGM system.

7. References

[Chang 1973] Chang, C.L. and Lee, R.C.T. (1973) *Symbolic Logic and Mechanical Theorem Proving*, Academic Press.

[de Kleer 1986] de Kleer, J. (1986) *An Assumption-Based TMS*, Artificial Intelligence 28, pp. 127-162.

[Dixon 1992a] Dixon, S. and Foo N. (1992) *Encoding the ATMS in AGM Logic (Revised)*, CS Technical Report 441, University of Sydney.

[Dixon 1992b] Dixon, S. and Foo N. (1992) *Beyond Foundational Reasoning*, Proc. 5th Australian Joint Conf. on Artificial Intelligence, World Scientific.

[Gärdenfors 1988] Gärdenfors, P. (1988) *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, Bradford Books, MIT Press, Cambridge, MA.

[Gärdenfors 1990] Gärdenfors, P. (1990) *Belief Revision and Nonmonotonic Logic: Two Sides of the Same Coin?*, Proc. 9th European Conf. on Artificial Intelligence, ed. L.C. Aiello, Pitman.

[Gärdenfors 1991] Gärdenfors, P. and Makinson, D. (1991) *Nonmonotonic Inference Based on Expectations*, Proc. 2nd Int. Conf. on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann.

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

[Nebel 1990] Nebel, B. (1990) *Representation and Reasoning in Hybrid Representation Systems*, Springer-Verlag, Berlin.

[Peppas 1992] Peppas, P. and Williams, M. (1992) *A Unified View of Constructive Modellings for Revision*, Annals of Mathematics and Artificial Intelligence, to appear.

[Poole 1988] Poole, D. (1988) *A Logical Framework for Default Reasoning*, Artificial Intelligence 36, 27-47.

[Spohn 1988] Spohn, W. (1988) *Ordinal Conditional Functions: A Dynamic Theory of Epistemic States*, in Harper W.L. and Skyrms, B. (eds.), Causation in Decision, Belief Change, and Statistics, II, 105-134, Kluwer.

[Williams 1993] Williams, M. (1993) *On The Logic of Theory Base Change*, CS Technical Report 459, University of Sydney.