Connections Between the ATMS and AGM Belief Revision

Simon Dixon and Norman Foo
Knowledge Systems Group
Basser Department of Computer Science
University of Sydney
Sydney, NSW 2006
AUSTRALIA

Abstract

The Assumption-based Truth Maintenance System (ATMS) [de Kleer, 1986] is the most well known implementation of any dynamic reasoning system. Some connections have been established between the ATMS and various nonmonotonic logics (e.g. autoepistemic logic [Reinfrank *et al.*, 1989]). We describe the relationship between the ATMS and the AGM logic of belief [Gärdenfors, 1988], and show that it is possible to simulate the behaviour of the ATMS using the AGM logic by encoding the justificational information as an epistemic entrenchment ordering. The ATMS context switching is performed by AGM expansion and contraction operations. We present an algorithm for calculating this entrenchment ordering, and prove its correctness relative to a functional specification of the ATMS. This result demonstrates that the AGM logic, which is based on the coherence theory of justification, is able to achieve both coherence and foundational style behaviour via the choice of epistemic entrenchments.

1 Introduction

Few would dispute the necessity of having a method of representing justificational information in a nonmonotonic reasoning system, but there is less agreement on what form these justifications should take. There are two distinct philosophical approaches to formalising the requirements for a belief to be justified by another belief or set of beliefs; these are called the *foundational* and *coherence* theories of justification [Pappas and Swain, 1978; Gärdenfors, 1989].

1.1 The Foundational Theory

Foundational reasoning is based on the concept that each fact or belief is accepted on the grounds of other beliefs which justify it. These beliefs are in turn justified by others, forming a chain of supports for each belief. Infinite chains of supports are disallowed, as are cycles in the chains, so that each chain of inferences starts from a set of premises or assumptions. These are called the foundational beliefs, since they give support to the whole of the belief set, and yet are not supported themselves by any other beliefs.

A number of operational systems based on the foundational model have been developed, the most well-known of which is the Assumption-based Truth Maintenance System (ATMS) [de Kleer, 1986]. The ATMS is used in diagnosis and qualitative physics, so functionality and efficiency are real concerns in its design. Hence we have in the ATMS an operational and reasonably efficient implementation of the foundational style of reasoning.

1.2 The Coherence Theory

The coherence theory of justification takes a different view of what constitutes a valid justification or reason to believe a proposition. A belief is considered valid on the basis of its coherence with all of the other beliefs, rather than having an explicit justification, as required by the foundational theory. In other words, a belief or fact need not have any explicit support for it to be included in the set of beliefs. The main criterion for the acceptance of a belief is that it is coherent with all or as many as possible of the other accepted beliefs. In this way, beliefs are able to justify each other in a circular fashion, so there is no concept of a foundational belief.

The coherence theory is also accompanied by a principle of minimal change: when accepting a belief which is inconsistent with the belief set, the aim is to modify the belief set as little as possible whilst incorporating the new belief and maintaining a coherent set of beliefs.

The most well-developed system of this sort is the AGM logic of belief [Gärdenfors, 1988]. The AGM logic is defined by a set of rationality postulates, which are intended to capture the notion of rational change of belief. In addition to these postulates, there is an ordering on the beliefs, called the epistemic entrenchment [Gärdenfors and Makinson, 1988], which ensures that the system can evaluate a unique solution within the constraints imposed by the postulates. One major drawback with the AGM logic is that the operations all produce closed theories, which are usually infinite. Various solutions to this problem have been proposed, using finite theory bases to represent belief sets. See [Nebel, 1989; 1991; Williams, 1993], where the AGM postulates are weakened to provide more efficient revision algorithms. An implementation of Williams' approach is described in [Dixon, 1993].

1.3 Motivation and Overview of the Paper

One desirable property of any reasoning system is a means of expressing explicit justificational information, and reasoning with it foundationally. We show that although the

AGM logic is based on the coherence theory of justification, it is possible to use epistemic entrenchment to achieve foundational behaviour. We describe an algorithm for translating the justificational information stored by the ATMS into an entrenchment ordering on the beliefs in the AGM model. This algorithm uses the AGM expansion and contraction operations to perform ATMS context switches. We prove that this algorithm creates a system which is behaviourally equivalent to the ATMS, relative to the specification of the ATMS given in section 2.1.

One motivation for developing this algorithm was to show that by using epistemic entrenchment it is possible for the AGM logic to exhibit foundational behaviour, without the limitations of the ATMS. In the ATMS, the problem solver and the truth maintenance systems are separate, so that the justifications are not part of the logic. Attempts have been made to formalise the logic of the ATMS [Reiter and de Kleer, 1987; Selman and Levesque, 1990]. We do not attempt to formalise the logic of the ATMS in this paper, but we have formalised the functional behaviour of the ATMS in order to prove the correctness of the simulation algorithm. There is no corresponding problem with formalising the AGM logic, since the justifications are part of the logic, so it is possible to reason about the system directly. Also, the ATMS's separation of justifications from the data means that the justifications cannot be altered without restarting the system, whereas the AGM logic allows the justifications to be varied dynamically.

By establishing a constructive connection between the AGM logic and the ATMS, we can use existing relationships between the ATMS and other logics to calculate the epistemic entrenchment relations which correspond to these logics. To date there have been very few constructive definitions of entrenchment relations which achieve the same behaviour as other systems.

In section 2, we describe the ATMS and give a formal specification of its behaviour. The AGM logic is outlined in section 3, followed by the algorithm for simulating the ATMS in the AGM logic in section 4, with the proof of the correctness of the algorithm in section 5. Section 6 summarises the paper, and outlines areas of future research.

2 The ATMS

The ATMS is an implementation of the foundational approach to modelling states of belief. Each belief must be justified by some set of beliefs, and any proposition that does not have such a justification is not accepted as a belief. The exception to this rule is the set of foundational beliefs, which require no justification at all. In the ATMS, these beliefs are called *assumptions*. For a justification to be valid, the beliefs which justify any proposition must be justified themselves, so that chains of justifications are formed.

The foundational theory places two restrictions on these chains: firstly they must be acyclic, so that no proposition can form part of its own justification, and secondly the chains must be finite, regressing to the beliefs which require no further justification. The ATMS removes the restriction that justifications be acyclic, since the "ATMS mechanism will never mistakenly use it as a basis for support" [de Kleer, 1986, p.155]. Naturally, the ATMS has no way of representing infinite chains of justifications. We place one additional restriction on the ATMS: we do not allow the assumptions to be justified, in keeping with the concept of foundational beliefs. [de Kleer, 1986, p.147] provides a technique which "avoids ever having to justify assumptions", so this restriction does not reduce the power of the ATMS at all.

In the ATMS, each proposition is represented by a node, and is treated as atomic. Any logical relationship between nodes must be provided by the problem solver, in the form of justifications which are passed to the TMS. The justifications represent propositional Horn clauses, with the normal provability relationships, except that inconsistency is avoided by the use of *nogoods*, which have a separate data structure. Nogoods are sets of assumptions which cannot be held simultaneously, and they are used to reduce the size of the search space, improving the efficiency of the ATMS.

The current set of assumptions is called the *environment*, and the set of facts derivable from this environment is called the *context*. An atomic proposition p is held to be true in an environment E if and only if it is a member of the current context. In this work, ATMS provability is denoted $E \models_{ATMS} p$.

For a node representing a proposition to be held true in a particular context, it must have a well-founded supporting justification, that is, a chain of justifications starting from a set of assumptions, where each of these assumptions is part of the current environment. The only other conditions under which a node is considered true are when it is an assumption, and therefore requires no justification, and when the current environment is inconsistent, in which case all nodes are accepted as true.

The implementation of the ATMS achieves a high level of efficiency by creating a static data structure in place of a theorem prover to calculate revisions of belief sets. The justifications and nogoods are "pre-compiled" into this data structure initially, and after this they are not consulted again. For each proposition p, the ATMS creates a node representing it, and associates a *label* with it, which stores the justificational information. The label represents the set of environments in which the proposition p is true. The label must be consistent, sound, complete and minimal. A label is consistent when each of its members is a consistent environment. That is, a label must not contain an environment which is a superset of any nogood set. The label for p is sound when p is derivable from each environment of the label. If every environment from which p can be derived is a superset of some environment of its label, then the label is complete. A label is minimal if no environment of the label is a superset of any other.

2.1 ATMS Functional Specification

Let E^* represent the set of all assumptions. As described above, no member of E^* may appear as the consequent of any justification. Let a and c represent atoms, and A a set of atoms. The input to the ATMS consists of a set of justifications, $J = \{(A,c) \mid (\bigwedge_{x \in A} x) \to c\}$, a set of nogood environments, $N = \{A \subseteq E^* \mid \neg \bigwedge_{x \in A} x\}$, and the current environment, $E = \{a \in E^* \mid a\}$.

$E \models_{ATMS} p$ if and only if

1. $\exists\, C \in N$ such that $C \subseteq E$, (E is inconsistent), or
2. $p \in E$, (p is an assumption), or
3. $\exists\, (A,p) \in J$ such that $\forall\, a \in A$, $E \models_{ATMS} a$, (p has a well-founded justification).

(3) may be reformulated in terms of the ATMS label:

3*. $\exists\, L \in Label(p)$ such that $L \subseteq E$.

3 The AGM Logic

The AGM belief revision logic has been described at length in [Gärdenfors, 1988]. It is based on a logically closed set of beliefs, K, with the closure operator denoted *Conseq*, so that $K = Conseq(K)$. Three basic operations are central to the system: expansion, contraction and revision, denoted K_a^+, K_a^- and K_a^* respectively, where a is the sentence by which the belief set is being expanded, contracted or revised. The inconsistent belief set is denoted $K_\perp$. Expansion involves adding a new belief to the belief set, with all of its consequences. A contraction is the removal of a belief from the belief set, accompanied by the removal of sufficient other beliefs so that the belief set remains consistent and closed under logical consequence. That is, the belief being removed must not be derivable from the remaining belief set. The third operation, revision, comprises the addition of a new belief to the belief set such that any conflicting beliefs are removed from the belief set, maintaining the consistency of the system. Revision can be defined as a contraction of the negation of the belief, followed by an expansion operation; this is called the Levi identity.

The postulates for contraction and revision do not define a unique function for either operation, so it is necessary to add some further constraints, in the form of an epistemic entrenchment relation, which is an ordering on the members of the belief set. Let $Ent(p)$ denote the entrenchment of proposition p, where the entrenchments are ordered by the normal relational operators: <, ≤ and =. Entrenchment orderings obey the following five axioms [Gärdenfors, 1988]:

(EE1) If $Ent(A) \le Ent(B)$ and $Ent(B) \le Ent(C)$ then $Ent(A) \le Ent(C)$

(EE2) If $A \vdash B$ then $Ent(A) \le Ent(B)$

(EE3) For any A and B, $Ent(A) \le Ent(A \wedge B)$ or $Ent(B) \le Ent(A \wedge B)$

(EE4) When $K \ne K_\perp$, $A \notin K$ iff $Ent(A) \le Ent(B)$ for all B

(EE5) If $Ent(B) \le Ent(A)$ for all B then $\vdash A$

Then the AGM operators may be defined as follows:

Expansion: $K_\alpha^+ = Conseq(K \cup \alpha)$

Contraction: $K_\alpha^- = \{\beta \in K \mid \vdash \alpha \text{ or } Ent(\alpha) < Ent(\alpha \vee \beta)\}$

Revision: $K_\alpha^* = (K_{\neg\alpha}^-)_\alpha^+$

4 Translation Algorithm

4.1 Constructing an Entrenchment Relation

To specify an entrenchment ordering completely requires an ordering on the dual atoms. The number of dual atoms is exponential in the number of atoms in the language, and this gives us reason to look for a partial ordering which contains sufficient information to define the contractions which interest us, and which is consistent with the entrenchment axioms. In doing this, we are not defining an alternative to the AGM entrenchment axioms, but instead we are allowing many of the entrenchment values to be unknown, since we will never need to consult these values to implement the AGM contraction operator. Thus the partial entrenchment relation which we define may be extended to any full entrenchment relation (as long as it is consistent with the AGM entrenchment axioms), without affecting our results.

Since we are modelling ATMS environment changes, the only beliefs to be added or contracted are ATMS assumptions, which are atomic. In fact, our algorithm contracts one atom at a time, and for the results we require, the only beliefs we need to check for membership in K are also atomic (ATMS nodes). From the definition of contraction, we only need to know entrenchment relations between each assumption and the disjunction of the same assumption with each other atom. For any atoms a and b, (EE2) constrains this relation to $Ent(a) \le Ent(a \vee b)$, which allows two possibilities: $Ent(a) < Ent(a \vee b)$ or $Ent(a) = Ent(a \vee b)$. For this algorithm, it is sufficient to use only 5 distinct entrenchment values, denoted E_1, E_2, E_3, E_4 and E_5, where $E_1 < E_2 < E_3 < E_4 < E_5$. We denote the entrenchment of a proposition p, when the AGM system represents the ATMS in environment E, by $Ent_E(p)$. Then by (EE4), $Ent(p) = E_1$ iff $p \notin K$, for consistent K. By (EE5), $Ent(p) = E_5$ only if p is a tautology. Since ATMS justifications cannot be altered once they are provided by the problem solver, they are entrenched at the next highest level of entrenchment, E_4. The atoms in the current environment have entrenchment E_2, and all disjunctions of an assumption and any other atom are initially given the entrenchment E_3.

4.2 Definitions

Let the set of all atoms be denoted Σ.

From the functional specification of the ATMS, it is possible to work backwards to a definition of the label of a node. Each environment in the label of a node p is a support set for p, so the label is a set of support sets, which we may define as follows:

$A \in Label(p)$ if and only if

1. $A \subseteq E^*$
2. $A \models_{ATMS} p$
3. $(\forall\, A' \subset A)\,[A' \not\models_{ATMS} p\,]$
4. $(\neg\, \exists\, C \in N)\,[C \subseteq A\,]$

The first condition requires that the label contain assumptions only, that is, the foundational beliefs of the node. The second condition ensures soundness, the third minimality, and the fourth consistency. Completeness is achieved when the label contains all sets fulfilling the four conditions.

The above definition uses the ATMS provability relation to extract the internal data structure from the ATMS. However, an explicit definition is necessary to implement the following algorithm. The label update algorithm is performed as follows:

Update_Label (p)
For each justification $(B,p) \in J$
  If the label of any member of B is empty
    Continue with next justification
  For all choices of one environment from each member of B
    Let L be the union of these environments
    If L subsumes any nogood environment
      Continue with next choice
    Else if L is subsumed by any environment in $Label(p)$
      Continue with next choice
    Else if L subsumes any environments in $Label(p)$
      Remove those environments from $Label(p)$
    Add L to $Label(p)$
If $Label(p)$ has changed
  For each justification $(B,q) \in J$ such that $p \in B$
    Update_Label (q)
End {*Update_Label*}

The foundational beliefs of an atom p in the environment E, denoted $FB(p,E)$, is the set of support sets of p which are held to be true in E. If no such set exists, then the atom has no well-founded justification. Formally:

$$FB(p,E) = \{ A \in Label(p) \mid A \subseteq E \}$$

The essential support set, $ES(p,E)$, is defined:

$$ES(p,E) = \bigcap_{X \in FB(p,E)} X$$

From the functional specification of the ATMS, an atom is considered true if and only if it is an assumption, or the environment is inconsistent, or it has a well-founded justification. In terms of the above definitions:

$$\begin{aligned} E \vdash_{ATMS} p \leftrightarrow\ & (p \in E) \vee \\ & (\exists\, C \in N)\ [C \subseteq E] \vee \\ & (FB(p,E) \neq \emptyset) \end{aligned}$$

Using these definitions, we now describe the algorithm which calculates the entrenchments for the AGM logic to simulate the ATMS behaviour.

4.3 Algorithm ATMS_to_AGM

This algorithm calculates the epistemic entrenchments necessary to simulate ATMS environment change in the AGM logic via the AGM belief change operators described in section 3. The belief set K contains all of the current beliefs, and corresponds to the ATMS context. Obviously, since K is logically closed, it is a much larger set than the corresponding ATMS context, but we show that for all $p \in \Sigma$, $E \vdash_{ATMS} p \leftrightarrow p \in K$.

The algorithm encodes the current support relationships in the entrenchment relation. That is, if p is an essential support for q, then the removal of p must force the removal of q, and this, according to the definition of contraction, requires that the entrenchment relation contain $Ent(p) = Ent(p \vee q)$. Since $Ent(p) = E_2$, this is achieved by setting $Ent(p \vee q) = E_2$. Alternatively, if q is not dependent on p, then we have $Ent(p) < Ent(p \vee q)$, by setting $Ent(p \vee q) = E_3$.

For each environment change, the translation algorithm computes the corresponding changes in the entrenchment relation from the changes in the essential support sets. For each new assumption A in the essential support of the proposition B, the entrenchment of $A \vee B$ is set to E_2, so that $Ent(A) = Ent(A \vee B)$. Conversely, for any A which was a member of the essential support set of B but is no longer in the essential support set, we let $Ent(A \vee B) = E_3$, so that $Ent(A) < Ent(A \vee B)$.

Initially, K contains the justifications and nogoods in their logical (Horn clause) form. That is, for each $(A,p) \in J$, $((\bigwedge_{x \in A} x) \rightarrow p) \in K$, and for each $C \in N$, $(\neg \bigwedge_{x \in C} x) \in K$. Also, for each of these formulas f, $Ent(f) = E_4$. The initial value of Old_Environment is $\emptyset$.

Algorithm ATMS_to_AGM
For each *New_Environment*
  $\forall\, x \in (Old\_Environment - New\_Environment)$
    { Remove all assumptions which are no longer true }
    $K := K_x^-$ { AGM contraction }
    { Note that AGM logic implies that $Ent(x) := E_1$ }
    $\forall\, c$ such that $x \in \bigcup_{X \in FB(c, Old\_Environment)} X$
      $\forall\, y \in (ES(c, New\_Environment) - ES(c, Old\_Environment))$
        { Entrench new essential supports }
        $Ent(y \vee c) := E_2$ (i.e. $Ent(y) = Ent(y \vee c)$)
      $\forall\, y \in (ES(c, Old\_Environment) - ES(c, New\_Environment))$
        { Change entrenchments of old essential supports }
        $Ent(y \vee c) := E_3$ (i.e. $Ent(y) < Ent(y \vee c)$)
  $\forall\, x \in (New\_Environment - Old\_Environment)$
    { Add all assumptions which have become true }
    $K := K_x^+$ { AGM expansion }
    $Ent(x) := E_2$
    $\forall\, y \in \Sigma,\ Ent(x \vee y) := E3$ { Default entrenchment }
    $\forall\, c$ such that $x \in \bigcup_{X \in Label(c)} X$
      $\forall\, y \in (ES(c, New\_Environment) - ES(c, Old\_Environment))$
        { Entrench new essential supports }
        $Ent(y \vee c) := E_2$ (i.e. $Ent(y) = Ent(y \vee c)$)
      $\forall\, y \in (ES(c, Old\_Environment) - ES(c, New\_Environment))$
        { Change entrenchments of old essential supports }
        $Ent(y \vee c) := E_3$ (i.e. $Ent(y) < Ent(y \vee c)$)
  $Old\_Environment := New\_Environment$
End { ATMS_to_AGM }

Note that the default entrenchment, $Ent(x) < Ent(x \vee y)$, for atomic x and y, encodes the fact that x and y are unrelated, since if x is removed from K, y is unaffected. It is the job of the problem solver, not the TMS, to point out any relationships between the various atoms, or if it is considered expedient, to refrain from doing so.

The amount of work performed by the algorithm suggests that simulating the ATMS in AGM logic is not trivial. The reason for this is that foundational reasoning is independent of history - for example the ATMS context depends only on the current environment, and is not affected by any previous environment, whereas the AGM logic relies on a principle of minimal change when moving from one theory to the next. Unfortunately, this minimal change principle cannot also apply to the entrenchment relation, thus extensive revisions of the entrenchment relation are often necessary to keep the system "effectively independent" of its previous state.

4.4 Example

Consider a simple system with just two rules:

$A \to D,\ \ B \wedge C \to D$

Now suppose we want to change environment from $Old\_Environment = \{A,B\}$ to $New\_Environment = \{B,C\}$. The entrenchment relation in *Old_Environment* will contain:

E_1: C

E_2: $A,\ B,\ D,\ A{\vee}D$

E_3: $A{\vee}B,\ B{\vee}D$

The essential support for D in *Old_Environment* is $\{A\}$, and in *New_Environment* it is $\{B,\ C\}$. Applying the algorithm above, we first remove A from the environment. Since A features in one of the foundational beliefs of D, the essential supports of D are recalculated, giving $Ent(A{\vee}D) = E_3$ and $Ent(B{\vee}D) = Ent(C{\vee}D) = E_2$. In the next step, C is added to the environment with entrenchment E_2, and the formulae $C{\vee}A$, $C{\vee}B$, and $C{\vee}D$ are entrenched at level E_3. Finally, since C appears in the label of D, the essential supports of D are calculated again, giving a relation containing:

E_1: A

E_2: $B,\ C,\ D,\ B{\vee}D,\ C{\vee}D$

E_3: $A{\vee}B,\ A{\vee}C,\ A{\vee}D,\ B{\vee}C$

5 Theorem and Proof of Correctness

The ATMS and AGM systems are equivalent if and only if:

Theorem 1: $(\forall p \in \Sigma)\ \ [\, E \models_{ATMS} p \ \leftrightarrow\ p \in K\,]$

We will prove this theorem by induction on the number of expansion and contraction operations. Firstly, it is necessary to prove two important lemmas which establish the relationship between entrenchment and the notion of essential support, and that between essential support and ATMS provability.

5.1 Lemma 1

For all $a{\in}E$, and for all $b{\in}\Sigma$,

$Ent_E(a) \ = \ Ent_E(a \vee b) \ \ \leftrightarrow\ \ a \in ES(b,E)$

5.2 Lemma 2

If E is a consistent environment and $a \neq b$ then

$a \in ES(b,E) \ \ \leftrightarrow\ \ (E \models_{ATMS} b) \wedge ((E - \{a\}) \not\models_{ATMS} b)$

Lemmas 1 and 2 are proved in [Dixon and Foo, 1992a].

5.3 Proof of Theorem 1

Initial case

Before any operations are performed, $E = \emptyset$. In this case, $E \models_{ATMS} a \ \leftrightarrow\ \exists\,(B,a) \in J$ such that $(\forall\, b \in B)\ [E \models_{ATMS} b\,]$. Since no assumptions are true, the chains of justifications must end in formulae of the form $(\emptyset,x)$, which are called *premises*.

Part (1): Suppose $E \models_{ATMS} a$. We show by induction on the length (n) of the longest chain of justifications that $a \in K$.

When $n = 1$, we have $(\emptyset,a) \in J$, which in Horn clause form is the formula a, and hence $a \in K$.

Inductive hypothesis: assume that if $E \models_{ATMS} a$ then $a \in K$ for all $n \leq k$.

Consider a proof of a with maximum length $k{+}1$ justifications. Then $\exists\,(B,a) \in J$ such that $\forall\, b \in B,\ E \models_{ATMS} b$ in at most k steps. Then by the inductive hypothesis, each b is in K. Since K is logically closed, we also have $(\bigwedge_{x \in B} x) \in K$. Combining this with $(\bigwedge_{x \in B} x) \to a$, which is also in K, since $(B,a) \in J$, we can conclude by modus ponens that $a \in K$.

We conclude by induction that for all n, $a \in K$.

Part (2): Conversely, suppose $a \in K$. Then there exists a sequence of resolutions of clauses in K from $J \cup N$ such that the final step produces the clause a. We know that no clauses from N are used in the resolution, because when one of the resolving clauses has no positive literals, the resulting clause will have no positive literals. Therefore no such clause can be involved in a sequence of resolutions which produces a positive literal. We now show by induction on the number (n) of resolution steps that $E \models_{ATMS} a$.

Initial case: when n=0, then $(\emptyset,a) \in J$. Hence $E \models_{ATMS} a$.

Inductive step: assume for all $n \leq k$ that $E \models_{ATMS} a$.

Consider a sequence of $k+1$ resolution steps ending with the clause a. Then there exists in this sequence a clause containing the positive literal a, corresponding to some justification $(B,a) \in J$. Let this clause be $\neg b_1 \vee \cdots \vee \neg b_m \vee a$. (i.e. $B = \{b_1, \ldots, b_m\}$). Then for $1 \leq i \leq m$, $b_i \in K$, and the resolution proof for each b_i takes $\leq k$ steps, so by the inductive hypothesis $E \models_{ATMS} b_i$. But $(\{b_1, \ldots, b_m\},a) \in J$, so by the definition of the ATMS, $E \models_{ATMS} a$.

By induction, we have shown that for all n, if $a \in K$ then $E \models_{ATMS} a$. This completes the initial case of the proof of the theorem.

Inductive step

Assume that after m expansions and contractions that $p \in K \leftrightarrow E \models_{ATMS} p$, for all $p \in \Sigma$. Consider the $(m{+}1)$st operation.

Case (1): Expansion by a

Let $E' = E \cup \{a\}$.

Suppose $E' \models_{ATMS} b$. Then one of the following 4 cases must hold:

(a) $E \models_{ATMS} b$

(b) $E \not\models_{ATMS} b$ and $\exists\ C \in N$ such that $C \subseteq E'$

(c) $E \not\models_{ATMS} b$ and $a = b$

(d) $E \not\models_{ATMS} b$ and $\exists\,(A,b) \in J$ such that $\forall\, x \in A,\ E' \models_{ATMS} x$

Case (a): By the inductive hypothesis, $b \in K$. Using AGM axiom (K^+3), $K \subseteq K^+_a$, we have $b \in K^+_a$.

Case (b): $E \subseteq K$, from the inductive hypothesis, since $\forall\, x \in E,\ E \models_{ATMS} x$. Also $a \in K^+_a$, by AGM axiom (K^+1). Since $K \subseteq K^+_a$ (AGM axiom (K^+3)) and $C \subseteq E'$, we have $C \subseteq K^+_a$. But since $C \in N$, $(\neg \bigwedge_{x \in C} x) \in K \subseteq K^+_a$, and by logical closure $K^+_a = K_\perp$. Thus $b \in K^+_a$.

Case (c): By AGM axiom (K^+1), $b \in K^+_b$.

Case (d): We show that $b \in K^+_a$ by induction on the number (n) of justifications used in proving $E' \models_{ATMS} b$.

Initial case: when $n = 0$, by Cases (a)-(c), we have $b \in K^+_a$.

Inductive step: assume that for all $n \leq k$ we have $b \in K^+_a$. Consider a proof using $k+1$ justifications, including a justification for b, $(A,b) \in J$. Then for all $x \in A,\ E' \models_{ATMS} x$. Now if Case (a), (b) or (c) applies, we have shown $x \in K^+_a$. Alternatively, if Case (d) applies, the proof of x contains $\leq k$ justifications, and hence by the inductive hypothesis $x \in K_a^+$. Also, since $(A,b) \in J$, $[(\bigwedge_{x \in A} x) \to b] \in K \subseteq K_a^+$, so by closure we have $b \in K_a^+$.

Hence, by induction $b \in K_a^+$ for all n.

Conversely, assume that $b \in K_a^+$. That is, $b \in Conseq(K \cup \{a\})$.

We show by induction on the number (n) of resolutions involved in proving b from $K \cup \{a\}$ that $E' \models_{ATMS} b$.

Initial case: when $n = 0$, either $b \in K$ or $b = a$. If $b \in K$, then by the inductive hypothesis of the main proof, $E \models_{ATMS} b$, and hence since $E \subseteq E'$, we have $E' \models_{ATMS} b$. Otherwise, if $b = a$, then $b \in E'$ and thus $E' \models_{ATMS} b$.

Inductive step: assume for all $n \le k$ that $E' \models_{ATMS} b$. Now consider a resolution proof of b requiring $k+1$ steps. This procedure uses a formula from J of the form $\neg r_1 \vee \cdots \vee \neg r_m \vee b$, where $r_i \in K_a^+$, $1 \le i \le m$. For this to resolve to b, there are resolution proofs for all r_i, $1 \le i \le m$, with each of these procedures requiring $\le k$ steps. Hence by the inductive hypothesis $E' \models_{ATMS} r_i$, $1 \le i \le m$. Also $(\{r_1, \ldots, r_m\}, b) \in J$, so by the definition of the ATMS, $E' \models_{ATMS} b$.

Hence for all n, $E' \models_{ATMS} b$, and thus we have shown that $E' \models_{ATMS} b \leftrightarrow b \in K_a^+$.

Case (2): Contraction by a

Let $E' = E - \{a\}$.

Suppose $E' \models_{ATMS} b$. Then $E \models_{ATMS} b$, since $E' \subseteq E$, and hence by the inductive hypothesis, $b \in K$. Then by Lemma 2, $a \notin ES(b,E)$. From (EE2) we have $Ent_E(a) \le Ent_E(a \vee b)$, and using Lemma 1, this result strengthens to $Ent_E(a) < Ent_E(a \vee b)$. Finally, from the definition of contraction, we have $b \in K_a^-$.

Conversely, suppose that $b \in K_a^-$. Then by the definition of contraction, $Ent_E(a) < Ent_E(a \vee b)$, and by Lemma 1, $a \notin ES(b,E)$. We also know that $b \in K$, since $K_a^- \subseteq K$. So $E \models_{ATMS} b$, by the inductive hypothesis, and then by Lemma 2 we conclude that $E' \models_{ATMS} b$.

Thus we have shown that $\forall b \in \Sigma$, $E' \models_{ATMS} b \leftrightarrow b \in K_a^-$, after the $(m+1)$st operation.

Therefore, we have shown by induction that, $p \in K \leftrightarrow E \models_{ATMS} p, \forall p \in \Sigma$, and the algorithm correctly chooses an entrenchment so that the behaviour of the AGM system is equivalent to that of the ATMS.

6 Conclusions and Further Research

We have shown that the ATMS can be simulated in the AGM logic by using a suitable epistemic entrenchment relation to encode the foundational justifications of the beliefs, and using contraction and expansion operations to perform context changes. The ATMS encoding does not specify a complete entrenchment relation, but calculates the class of entrenchments for which the AGM's behaviour with respect to atoms is equivalent to the ATMS. Thus the coherence-based AGM logic is able to mimic the behaviour of a foundational system, the ATMS. This illustrates an advantage of the AGM logic with entrenchment over the many purely foundational systems: it is possible to express all types of justificational information using epistemic entrenchment. Current research involves using epistemic entrenchment to model non-foundational justifications [Dixon and Foo, 1992b], such as a coherence-based truth maintenance system. Another advantage is that the AGM logic is a formal system, so it is possible to reason directly about the system, which is not true of the ATMS.

Other extensions of this work are rule revision, which is not allowed by the ATMS, but could be easily implemented in the AGM logic by decreasing the entrenchment of the justifications. One application of rule revision would be to model default rules and exceptions, by creating a stratified rule base, with various sets of rules being placed at different levels of entrenchment. Current research also includes extending the implementation in [Dixon, 1993] to a belief revision system which satisfies all of the AGM postulates.

References

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

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

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

[Dixon, 1993] Dixon, S. (1993) *A Finite Base Belief Revision System*, Proc. 16th Australian Computer Science Conference, Queensland University of Technology.

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

[Gärdenfors and Makinson, 1988] Gärdenfors, P. and Makinson, D. (1988) *Revisions of Knowledge Systems Using Epistemic Entrenchment*, Proc. of Second Conf. on Theoretical Aspects of Reasoning About Knowledge, pp. 83-95, ed. M.Y. Vardi, Morgan Kaufmann.

[Gärdenfors, 1989] Gärdenfors, P. (1989) *The Dynamics of Belief Systems: Foundations vs. Coherence Theories*, Department of Philosophy, Lund University.

[Nebel, 1989] Nebel, B. (1989) *A Knowledge Level Analysis of Belief Revision*, KR-89, Morgan Kaufmann.

[Nebel, 1991] Nebel, B. (1991) *Belief Revision and Default Reasoning: Syntax-Based Approaches*, KR-91, Morgan Kaufmann.

[Pappas and Swain, 1978] Pappas, G.S. and Swain M. (1978) *Essays on Knowledge and Justification*, pp. 30-35 and 184-308, Cornell University Press.

[Reinfrank *et al.*, 1989] Reinfrank, M., Dressler, O. and Brewka, G. (1989) *On the Relation Between Truth Maintenance and Autoepistemic Logic*, IJCAI-89, Morgan Kaufmann.

[Reiter and de Kleer, 1987] Reiter, R. and de Kleer, J. (1987) *Foundations of Assumption-Based Truth Maintenance Systems: Preliminary Report*, AAAI-87, Morgan Kaufmann.

[Selman and Levesque, 1990] Selman, B. and Levesque H.J. (1990) *Abductive and Default Reasoning: A Computational Core*, AAAI-90, MIT Press.

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