Programming Resource-Bounded Deliberative Agents

Michael Fisher and Chiara Ghidini*

Department of Computing and Mathematics
Manchester Metropolitan University
Manchester M1 5GD, United Kingdom

EMAIL: {M.Fisher, C.Ghidini}@doc.mmu.ac.uk

Abstract

This paper is concerned with providing a common framework for both the logical specification and execution of agents. While numerous high-level agent theories have been proposed in order to model agents, such as theories of intention, these often have little formal connection to practical agent-based systems. On the other hand, many of the agent-based programming languages used for implementing 'real* agents lack firm logical semantics. Our approach is to define a logical framework in which agents can be specified, and then show how such specifications can be directly executed in order to implement the agent's behaviour.

We here extend this approach to capture an important aspect of practical agents, namely their resource-bounded nature. We present a logic in which resource-boundedness can be specified, and then consider how specifications within this logic can be directly executed. The mechanism we use to capture finite resources is to replace the standard modal logic previously used to represent an agent's beliefs, with a *multi-context* representation of belief, thus providing tight control over the agent's reasoning capabilities where necessary.

This logical framework provides the basis for the specification and execution of agents comprising dynamic (temporal) activity, deliberation concerning goals, and resource-bounded reasoning.

1 Introduction

The METATEM [Barringer *et al,* 1995] and Concurrent METATEM [Fisher, 1995] languages have been used as high-level mechanisms for specifying and executing individual agents and multi-agent systems, respectively. Both are based upon the principle of specifying an agent using temporal logic, and then *directly executing* this specification in order to provide the agent's behaviour. This approach provides a high-level programming notation, while maintaining a close link between the program and its specification.

* Visiting Research Fellow from University of Trento, Italy, supported by the Italian National Research Council (CNR)

While this approach has provided a useful basis for experimentation with both the logical representation and animation of agents, it has become clear that a more refined version of the specification language is required if this framework is to be used for 'real world' agents. In particular, the METATEM family of programming languages originally contained no sophisticated mechanisms for representing deliberation within an agent, i.e. the process that an agent carries out in order to decide which goal/action/plan to attempt.

Thus, inspired by the success of the BDI framework [Rao and Georgeff, 1991] in representing deliberation, the basic METATEM system was extended, in [Fisher, 1997b], with explicit mechanisms for ordering goals. Goals, corresponding to both desires and intentions in the BDI model were, in turn, represented by temporal eventualities. This then allowed deliberation to be represented using user defined functions providing an ordering on the satisfaction of eventualities.

While this provides a simple and concise mechanism for representing and implementing deliberative agents, it does not deal with a further important aspect of 'real' agents, namely their resource-bounded nature [Bratman *et al,* 1988], In particular, the representation of belief was given by extending the temporal basis with a standard modal logic having Kripke semantics [Halpern and Moses, 1992]. As is well known, this does not match the resource-bounded nature of 'real' reasoners [Giunchiglia *et al,* 1993]. Indeed modal logics generally model logically omniscent agents which are forced to believe (and compute) all the logical consequences of their own beliefs.

Thus, in this paper we modify the logic used in [Fisher, 1997b] by replacing the standard KD45 modal logic with a *multi-context* representation of belief [Giunchiglia and Serafini, 1994; Benerecetti *et al,* 1997]. This logic is a modification of KD45 which permits a simple execution mechanism to be employed over belief contexts. Consequently, it allows us to tightly control the use of belief contexts within deliberative agents and so to represent resource-bounded reasoning. In addition, we also investigate the resource-bounded aspects of *temporal* reasoning. Thus, rather than allowing the agent to reason about the full underlying temporal logic, we examine how the agent can be restricted so that it reasons with an *abstraction* of the underlying temporal model.

The paper is structured as follows. In §2, we define the syntax and semantics of the particular logic used to represent

agents. This will combine a simple temporal logic [Emerson, 1990] with a multi-context logic of belief [Benerecetti *et al.,* 1997], In order to provide the basis for the direct execution of this logic, we introduce, in §3, a particular normal form, that extends the Separated Normal Form used in the METATEM [Fisher, 1997a]. An algorithm which can be used to directly execute formulae in this normal form is defined in §4, while an example of such execution is given in §5. The correctness of the execution algorithm is considered in §6. In §7, we consider the effects of restricting both the hypothetical doxastic and temporal reasoning that can be carried out by an agent. Finally, in §8, we provide conclusions from this research and consider future work.

2 Temporal Logic of Bounded Belief

In this section, we give the syntax and semantics of our base logic (called a *Temporal Logic of Bounded Belief* or TLBB for short) which combines propositional linear temporal logic, with a multi-context belief logic. While temporal reasoning is essentially infinite, reasoning about beliefs can be bounded. In this sense, the logic allows the representation of bounded reasoning.

Syntax We are considering a situation with an agent, a, observing and representing beliefs about a set $I = \{1, \ldots, n\}$ of agents. Formulae of TLBB are constructed using the following connectives and proposition symbols.

- A set, $\mathcal{P}$, of propositional symbols.
- Propositional connectives, **true, false,** ¬, ∨, ∧, and =>.
- Temporal connectives, $\bigcirc$, $\Diamond$, $\Box$, $\mathcal{U}$, and $\mathcal{W}$.
- A set, $\{B_1, B_2, \ldots, B_n\}$, of belief operators.

Let I^k be the set the set of (possibly empty) strings of the form $i_1 \ldots i_h$ w $|i_1 \ldots i_h| \leq k$. e call any $\alpha \in I^k$ a *view.* Intuitively, each view in I^k represents a possible nesting of the belief operators. The empty string represents the view of the external agent a. The views[1] that a can build can be organized in a structure such as that presented in Figure 1.

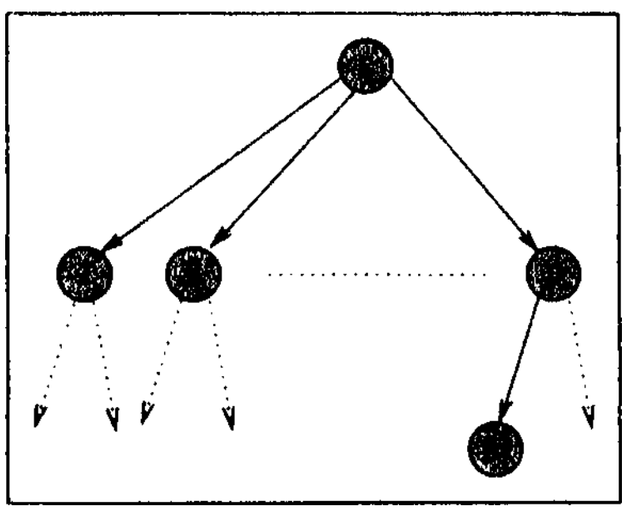

Figure 1: The structure of views

We associate a logical language L_α with each view. The set of well-formed formulae of L_α, denoted by $\mathbf{WFF}_\alpha$, is inductively defined as the smallest set satisfying:

- Any element of $\mathcal{P}$ is in WFF_α
- If ψ and φ are in WFF_α then so are

$$\neg\psi \quad \psi \vee \varphi \quad \psi \wedge \varphi \quad \psi \Rightarrow \varphi \quad \textbf{false} \quad \textbf{true}$$
$$\Diamond\psi \quad \Box\psi \quad \psi \,\mathcal{U}\, \varphi \quad \psi \,\mathcal{W}\, \varphi \quad \bigcirc\psi$$

- $B_i\psi$ [for all $\leq i \leq n$] s a n atomic formula i n $\mathbf{WFF}_\alpha$ if, and only if, ϕ is in $\mathbf{WFF}_{\alpha i}$

The set of well-formed formulae of TLBB comprises the sets of well-formed formulae contained in each L_α.

'A more detailed description of the structure of views we are using can be found in [Benerecetti *et a*/., 1997].

Semantics The semantics of the TLBB language is based on the semantics for contextual reasoning proposed in [Giunchiglia and Ghidini, 1998; Benerecetti *et al,* 1997]. Following this approach, the semantics of every formula is *local* to the view where we consider it. Therefore, we associate to each view α a discrete linear temporal model of time m_α with finite past and infinite future, e.g. $(\mathbb{N}, <)$, in order to interpret the temporal component of each language L_α. Each of these *moments* in time, represented by a temporal index $u \in \mathbb{N}$, provides also a valuation π_α for the propositional part of the language and for (atomic) formulae of the form $B_i\psi$. As usual, we define the semantics of each L_α via the satisfiability relation:

$\langle m_\alpha, u\rangle \models \textbf{true}$	$\langle m_\alpha, u_0\rangle \models \textbf{start}$
$\langle m_\alpha, u\rangle \models p$	iff $\pi_\alpha(u, p) = T$ (where $p \in \mathcal{P}$ or p is of the form $B_i\psi$ with $B_i\psi \in L_\alpha$)
$\langle m_\alpha, u\rangle \models \neg A$	iff $\langle m_\alpha, u\rangle \not\models A$
$\langle m_\alpha, u\rangle \models A \vee B$	iff $\langle m_\alpha, u\rangle \models A$ or $\langle m_\alpha, u\rangle \models B$
$\langle m_\alpha, u\rangle \models \bigcirc A$	iff $\langle m_\alpha, u+1)\rangle \models A$
$\langle m_\alpha, u\rangle \models \Box A$	iff $\forall u' \in \mathbb{N}$. if $(u \leq u')$ then $\langle m_\alpha, u'\rangle \models A$
$\langle m_\alpha, u\rangle \models \Diamond A$	iff $\exists u' \in \mathbb{N}$. if $(u < u')$ then $\langle m_\alpha, u'\rangle \models A$
$\langle m_\alpha, u\rangle \models A\,\mathcal{U}\,B$	iff $\exists u' \in \mathbb{N}$. such that $(u' \geq u)$ and $\langle m_\alpha, u'\rangle \models B$, and $\forall u'' \in \mathbb{N}$, if $(u \leq u'' < u')$ then $\langle m_\alpha, u''\rangle \models A$
$\langle m_\alpha, u\rangle \models A\,\mathcal{W}\,B$	iff $\langle m_\alpha, u\rangle \models A\,\mathcal{U}\,B$ or $\langle m_\alpha, u\rangle \models \Box A$

Satisfiability and validity are defined in the usual way.

The second step in defining the semantics for TLBB is to formalize the relation existing among different views. Indeed only a certain combinations of temporal models are *compatible* and provide a model for TLBB. A *model M* for TLBB is a set $M = \{m_\alpha\}_{\alpha \in I^k}$ such that for all $\alpha \in I^k$:

1. if $m_\alpha \models B_i\phi$ then $m_{\alpha i} \models \phi$;
2. if $m_{\alpha i} \models \phi$ then $m_\alpha \models B_i\phi$;
3. if $m_\alpha \models B_i\phi$ and $B_i\phi \in L_{\alpha i}$ then $m_{\alpha i} \models B_i\phi$;
4. if $m_\alpha \models \neg B_i\phi$ and $B_i\phi \in L_{\alpha i}$ then $m_{\alpha i} \models \neg B_i\phi$;

M satisfies a formula $\phi \in L_\alpha$ if m_α satisfies it.

Let us write $\alpha : \phi$ to mean that ϕ is a formula in L_α. Each model *M* for TLBB satisfies the formulae $\alpha : B_i(\phi \Rightarrow \psi) \Rightarrow (B_i\phi \Rightarrow B_i\psi)$, $\alpha : B_i\phi \Rightarrow B_iB_i\phi$ and $\alpha : \neg B_i\phi \Rightarrow B_i\neg B_i\phi$. They are the TLBB versions of modal axioms K, 4, and 5, respectively, and are obtained from conditions 1-4 in the definition of M. Moreover *M* satisfies the formulae $\alpha : \neg B_i\neg\phi \Rightarrow B_i\phi$ and $\alpha : B_i\phi \Rightarrow \neg B_i\neg\phi$. These are a consequence of the fact that *M* associates a unique temporal model m_α to each

view and ensures that '$\neg B_i \neg$' and 'B_i' are equivalent. While the fact that each view has exactly one B_i-accessible view enforces that each agent must either believe, or disbelieve, every atom, it does simplify the model structure, and hence the execution, required. As we will see in §5,, typical examples of agent reasoning are not affected by this addition.

3 Normal Form for TLBB

Formulae in TLBB can be transformed into a normal form SNF_{BB} (Separated Normal Form for temporal logics of bounded belief). Separated Normal Form (SNF) is used as the basic normal form for linear-time temporal logics [Fisher, 1997a] and has been extended, for example to temporal logics of knowledge in [Dixon *et al,* 1998]. The translation to *SNFBB* uses the renaming technique [Plaisted and Greenbaum, 1986] where complex subformulae are replaced by new propositions and then the truth value of these propositions are linked, in all states, to the formulae they replaced. In linear time temporal logic this is achieved by ensuring that such formulae are in the scope of a "$\Box$' operator, i.e. they hold at all reachable states. For TLBB we introduce the $\Box^*$ operator, which allows arbitrary nesting of B_i and $\Box$ operators to achieve the same effect. As in the case of temporal logics of knowledge, the $\Box^*$ operator can be defined using fixpoints [Dixon *et al.,* 1998].

Formulae in SNF_{BB} are of the general form

$$\Box^* \bigwedge_i T_i$$

where each T_i known as a *rule,* must be in one of the varieties described in Figure 2. Note that k_a, l_b, and / are all literals,

start	$\Rightarrow$	$\bigvee_{b=1}^{r} l_b$	(an *initial* rule)
$\bigwedge_{a=1}^{g} k_a$	$\Rightarrow$	$\bigcirc \left[\bigvee_{b=1}^{r} l_b \right]$	(a *step* rule)
$\bigwedge_{a=1}^{g} k_a$	$\Rightarrow$	$\Diamond l$	(a *sometime* rule)
$\bigwedge_{a=1}^{g} k_a$	$\Rightarrow$	$B_i \left[\bigvee_{b=1}^{r} l_b \right]$	(a *belief* rule)

Figure 2: Rules in SNF_{BB}

and the outer '$\Box^*$' operator that surrounds the conjunction of rules is usually omitted.

Example 3.1 Consider the TLBB formula

$$(\bigcirc B_2\psi) \Rightarrow ((B_1(B_2\varphi)) \wedge \Diamond(B_1(\psi \vee B_2\chi))) . \quad (1)$$

We can go through the following stages to transform this into SNF_{BB}:

a) rename $(\bigcirc B_2\psi)$ by p and add $\neg p \Rightarrow \bigcirc \neg B_2\psi$;

b) rename $\neg B_2\psi$ above b y q and a$q \Rightarrow B_2\neg\psi$;

c) rename.$B_2\varphi$ in (1) by r and add $r \Rightarrow B_2\varphi$;

d) r e n$B_1(\psi \vee B_2\xi)$ n (1) by s and add $s \Rightarrow B_1(\psi \vee B_2\xi)$;

e) rename $(\psi \vee B_2\xi)$ in the above formula by t and add $(t \wedge \neg\psi) \Rightarrow B_2\xi$.

The original formula is now $p \Rightarrow B_1 r \wedge \Diamond s$, which can be split giving the final set of rules in SNFBB as:

1. $p \Rightarrow B_1 r$
2. $p \Rightarrow \Diamond s$
3. $\neg p \Rightarrow \bigcirc q$
4. $q \Rightarrow B_2\neg\psi$
5. $r \Rightarrow B_2\varphi$
6. $s \Rightarrow B_1 t$
7. $(t \wedge \neg\psi) \Rightarrow B_2\chi$.

The key properties of the translation to SNFBB, characterised by the function V are as in the temporal case [Fisher, 1997a].

Theorem 1 *// TLBB formula φ is satisfiable, then$\tau(\varphi)$ is satisfiable.*

Theorem *2IfM is a model for$\tau(\varphi)$ then M is also a model for φ.*

Consequently, as execution is essentially model building in our framework, then we can execute $\tau(\varphi)$ in place of φ.

4 Execution of SNFBB

In this section we give an algorithm for executing a set of SNFBB rules. Since lack of space precludes us from providing the algorithm in full detail, we will first present an outline of the execution mechanism for temporal logic, based on [Barringer *et al,* 1995] and [Fishcr, 1997b], and will follow this with a more detailed account of the execution of the belief element of the logic.

4.1 Executing Temporal Logic

The basic idea underlying METATEM [Barringer *et al,* 1995] is to directly execute a temporal formula, by attempting to build a model for the formula in a simple forward-chaining fashion. This is extended, in [Fisher, 1997b], whereby the choice of which formulae to satisfy is provided by user defined deliberation functions, rather than by a fixed ordering heuristic. An outline of the basic approach, assuming that we are executing the set of rules, i?,, is given below.

1. By examining the *initial* rules, constraints on the possible start states for the temporal model can be generated.
   We choose one of these possible start states, deriving its valuation from the initial rules.
   If all the possible start states have been explored, then we terminate stating that the set of rules, R, is unsatishable.
2. Generate constraints on *next* states, C_n, and constraints *on future* states, C/, by checking applicability in current state of step and sometime rules, respectively.
   C_n represents all the possible choices of valuations for the next state, while C_f provides the set of eventualities that must be satisfied at some time in the future.

3. Make a choice from C_n and check that the chosen valuation is consistent. If there are no unexplored choices, return to a choice point in a previous state.

   The choice mechanism takes into account a combination of *Cf,* the outstanding eventualities, and the deliberation ordering functions [Fisher, 1997b].

4. Generate a new state, s, from the choice made in (3). Note that, by default, if propositions are not constrained we choose to leave them unsatisfied.

   Define *s* as being a successor to the current state and record the eventualities that are still outstanding (i.e. previously generated eventualities that were not satisfied in s); call this set of eventualities *Evs.*

   If any member of *Evs* has been continuously outstanding for more than $2^{5|R|}$ states, then return to a previous choice point and select a different alternative.

5. With current state, s, and the set of outstanding eventualities, *Evs,* go to (2).

The key result here is that, under certain constraints on the choice mechanism within (3), this execution algorithm represents a decision procedure (previously presented in [Barringer *et al.,* 1995]).

Theorem 3 *If a set of SNF rules, R, is executed using the above algorithm, with the proviso that the choice in (3) ensures that the oldest outstanding eventualities are attempted first at each step, then a model for R will be generated if and only if R is satisfiable.*

The above proviso ensures that, if an eventuality is outstanding for an infinite number of steps, then it will be attempted an infinite number of times. Once the choice mechanism is extended to include arbitrary ordering functions, as in [Fisher, 1997b], then a more general version of the above theorem can be given wherein we only require a form of *fairness* on the choice mechanism. While the above proviso effectively means that we *can* potentially explore every possibility, the incorporation in the algorithm of a bound on the number of states that eventualities can remain outstanding, together with the finite model property of the logic, ensures that all of the possible states in the model will be explored if necessary.

4.2 Extending Execution

We now extend the above algorithm to handle the execution of SNFBB- The two main elements of the algorithm affected by this are that the execution process now builds a labelled tree, rather than a sequence, and that once a new belief state is built it must be checked for equivalence with previously generated states. Thus, rather than just generating a set of choices based upon temporal rules, we must now consider both temporal and belief rules. This will (often) lead to the construction of a number of belief contexts and (simulated) temporal sequences in order to derive these choices. For example, in Figure 3 the basic temporal sequence (labelled by 'TL') is being constructed. However, at certain points, belief contexts (e.g. Bl and B2) must be explored in order to decide how to proceed. In addition, within these belief contexts, temporal execution itself can be simulated, e.g. B2(TL). Note

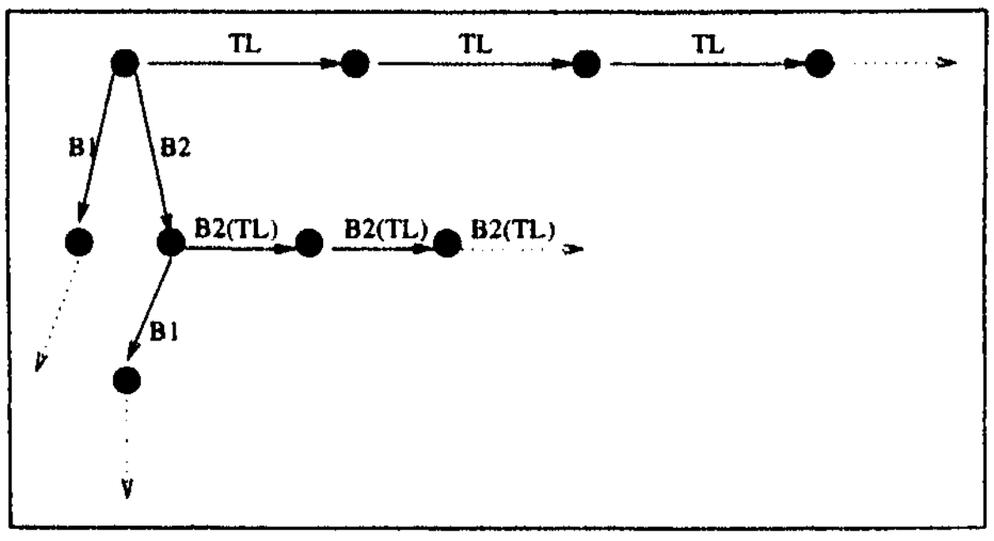

Figure 3: Typical Model Exploration

that exploration of everything within a belief context will be carried out within a finite amount of time.

In order to incorporate the execution of belief operators into the algorithm described in §4.1, we add the following to the end of step (4).

   For all belief operators, B_i., if *expand*$(B_i, R, s, 0)$ returns false, then return to a previous choice point and select a different alternative.

The function *'expand'* explores belief contexts and attempts to build a model for the multi-context beliefs as follows.

expand(B_i R, s, D) : *Boolean*

1. Let $C_B = \bigwedge F$, where $(P \Rightarrow B_i F) \in R$ and $s \models P$ if there are no such rules $P \Rightarrow B_i F$, **return** true.
   Let C'_B be DNF version of C_{Bi}., and remove all inconsistent disjuncts from C'_B..
2. Choose a disjunct, d, from C_B. if no unexplored disjuncts are available then immediately return false.
3. If $D \neq \emptyset$, then conjoin the formulae in *D* with *d* to give *d'* otherwise, let *d'* = *d.* Note that *D* contains the belief constraints from the last state and this operation corresponds to the persistence of beliefs given by axioms 4 and 5, e.g. $\vdash B_i\varphi \Rightarrow B_i B_i\varphi$
4. Generate a new state, *t,* with valuation based on *d!.* If a state equivalent to *t* has occurred previously within this exploration from a basic temporal state, then make a B_i edge from *s* to this previous state and return true.
   If no such equivalent state exists, then make a B_i edge from *s* to t
5. For all belief operators, B_i, if any of *expand*$(B_i, R, t, \{d'\})$ returns false, then return to (2) and select a different alternative.
6. If *tl_expand*$(R, t, \emptyset)$ returns false, then return to choice point (2) and select a different alternative.
7. Return true.

The function *tLexpand(Rides, State, Eventualities)* effectively carries out temporal reasoning in a belief context. However, rather than continuing indefinitely, as the base-level algorithm does, it uses a tableau-like produce to ensure that a finite structure is constructed if the formula is satisfiable.

Rather than giving the detail of *tLexpand,* we note that it is very similar to the base-level temporal execution algorithm, although it does try to recognise repeated temporal states and so construct a finite graph rather than an infinite sequence. We will consider this temporal reasoning aspect further in §7.2.

5 Example

Consider the following example of an agent acting as a travel information provider (note that, for reasons of space, the example is represented in predicate logic where variables range over finite domains).

A) $ask(you, x) \Rightarrow B_{me} \Diamond go(you, x)$
i.e., "if you ask for information about destination x, then I believe that you will holiday in *x* in the future"

B) $B_{me}[\Diamond go(you, x) \Rightarrow \Diamond buy(you, holiday, x)]$
i.e., [4t]"I believe that, if you will holiday in *x* in the future, then you will buy a holiday for *x* in the future"

C) $B_{me} \Diamond buy(you, holiday, x) \Rightarrow \bigcirc send\_info(you, x)$
i.e., "if I believe that you will buy a holiday for *x* in the future, I will send you information about holidays at J":

D) *ask(you,x)*
i.e., "you ask for information on destination :r":

These translate to the following *SNFBB* rules.

$$
\begin{array}{lrcl}
A1. & ask(you, x) & \Rightarrow & B_{me} a \\
A2. & a & \Rightarrow & \Diamond go(you, x) \\
B3. & b & \Rightarrow & B_{me} c \\
B4. & c & \Rightarrow & \Diamond buy(you, holx) \\
B5. & \neg b & \Rightarrow & B_{me} d \\
B6. & \neg b & \Rightarrow & B_{me} \neg go(you, x) \\
B7. & d & \Rightarrow & \bigcirc \neg go(you, x) \\
B8. & d & \Rightarrow & \bigcirc d \\
C9. & e & \Rightarrow & \bigcirc send\_info(you, x) \\
C10. & \neg e & \Rightarrow & B_{me} f \\
C11. & \neg e & \Rightarrow & B_{me} \neg buy(you, holx) \\
C12. & f & \Rightarrow & \bigcirc \neg buy(you, holx) \\
C13. & f & \Rightarrow & \bigcirc f \\
D14. & \textbf{start} & \Rightarrow & ask(you, x)
\end{array}
$$

Execution begins by ensuring that *ask(you,x)* is true in the initial temporal state and then exploring other choices. Thus, if we choose b to be false here, then we get a contradiction in the D_{me}*(TL)* simulation with (effectively) $\Box \neg go(you, x)$ (from rules *B5-B8)* and $\Diamond go(you, x)$ (from rule *A2).*

Similarly, if we choose *e* to be false, we get a contradiction in the B_{me}*(TL)* simulation with (effectively) $\tilde{\Box} \neg buy(you, holx)$ (from rules C'10-C[f]13) and $\Diamond$ *buy (you, holx)* (from rule *B4).*

Thus, we eventually choose to satisfy *ask(you, x), b* and *c* together in the initial state, which in turn leads to the execution of *sendJnfo(you, x)* in the next state.

The model eventually produced is given in Figure 4.

6 Correctness

As in the case with the basic METATEM system, we can show that the extended system can be used as a decision procedure for TLBB if necessary.

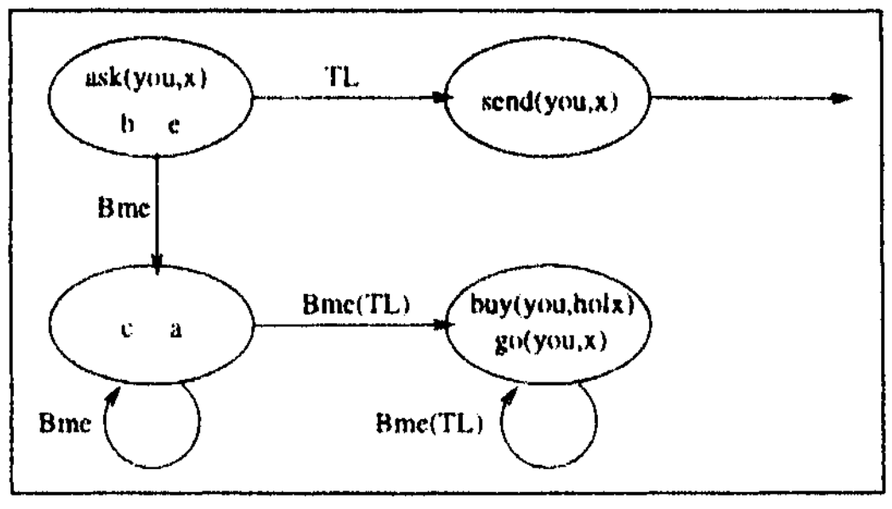

Figure 4: Example Execution

Theorem 4 *If a set of SNFBB rules, R, is executed using the above algorithm, and the choices made by the temporal component ensure that the oldest outstanding eventualities are attempted first at each step, then a model for R will be generated if and only if, R is satisfiable.*

The proof follows by showing that the above execution mechanism will (eventually) explore all the tableau structure for TLBB [Ghidini, 1998; Wooldndge *et aL* 1998]. In particular, the execution within belief contexts explores potential linear sequences of such contexts until either a model is found, or until no unexplored possibilities remain and so the belief formula is unsatisfiable.

In general, the mechanisation of such a logic is relatively easy since there are no axioms defining interactions between the different belief/temporal operators. Thus, the sub-logics within TLBB can effectively be treated separately, and decidability of TLBB is retained [Blackburn and de Rijke, 1997].

7 Resource-Bounded Reasoning

In §4.2, we considered the execution of the temporal logic of bounded belief with (effectively) infinite bounds on belief reasoning capabilities. Now we will examine how the effect that introducing resource bounds of various kinds will have on this execution mechanism.

7.1 Bounded Reasoning

The most obvious thing to do (and indeed this was the main motivation for this work) is to set an explicit bound on the depth of belief reasoning allowed. In order to modify the execution mechanism, we just need to add an extra argument to the *expand* and *tl_expand* functions. This argument will represent the depth of nesting of belief contexts. Thus, in the first call to *expand* (from the basic temporal execution), the depth is 0. In the definition of the *expand* function itself, the depth is incremented when the function is invoked recursively. Finally, we require an additional condition in *expand* stating that backtracking will occur if the depth bound reaches its limit (i.e. the maximum depth of nested beliefs allowed).

Given that we can bound the agent's ability to reason about belief, then can we do the same with temporal reasoning? In principle, yes. However, a bounded multi-context semantics for the type of temporal logic we use is quite complex, and we have not yet considered restricting the logic in this way.

7.2 Abstract Temporal Reasoning

From a practical point of view, carrying out hypothetical reasoning within propositional linear-time temporal logic is quite expensive. While the logic is decidable, the decision problem is PSPACE-complete [Emerson, 1990). Also, although the execution mechanism for METATEM is complete, and thus could be used as a (naive) theorem-prover for the logic, the execution mechanism is not at all efficient in this respect. We propose a modified logical framework which consists of three elements: (full) propositional linear temporal logic for the agent's basic execution; bounded (multi-context) belief; and 'simulated' temporal logic, comprising only ^{4} 0 \ hut no 'O' operator (this logic is only used within belief contexts and is equivalent to KDT4 modal logic). The *tLexpand* function described in §4.2 now executes the KDT4 logic, rather than temporal logic.

While the restriction of the capability of an agent to reason only about 'abstract' elements of time in a belief context may seem too extreme, it does significantly reduce the complexity of the execution mechanism. In addition, it is debatable whether an agent *should* be able to reason about the exact detail of the time structure in which it is situated. Further, if an agent wishes to reason about the temporal behaviour of another agent, then the asynchronous nature of agent execution means that a step in one agent will not necessarily match a step in the other. Thus, perhaps an agent should only be able to reason about another agent reaching a certain state at *some* point in the future. Hence the use, within hypothetical temporal reasoning, of just the '◇' operator.

8 Conclusions and Future Work

We have considered the extension of basic executable temporal logic (of the METATEM style) with bounded reasoning about belief and time. In particular, we have proposed the version of the logic in §7.2 as a practical basis for the high-level logic-based programming of resource-bounded agents.

It is important to note that we are not interested in generating alternative mechanisms for undertaking temporal/modal proof. The main element we are concerned with is the execution of temporal formulae in order to generate an infinite sequence. However, at certain points hypothetical reasoning needs to be carried out and belief contexts must be explored. We could have used standard proof mechanisms for non-interacting temporal and doxastic logics [Wooldridge *et aL,* 1998] at this point (and, indeed, our intention is to provide this possibility within the full implementation). However, (a) we would like to retain the common execution mechanism throughout, (b) by representing belief exploration in terms of execution the whole process should be more amenable to meta-level control, and (c) while belief exploration is neither very common nor very complex, it makes sense to keep the execution mechanism simple.

Finally, the execution mechanism described in this paper provides a way to execute multi-context logics of this form [Giunchiglia and Serafini, 1994] using the Imperative Future [Barringer *et* a/., 1996] style of executable logics.

Future work includes testing a full implementation on larger examples, incorporating belief revision and persistence, and investigating the addition of real-time aspects.

References

[Barringer *etal,* 1995] H. Barringer, M. Fisher, D. Gabbay, G. Gough, and R. Owens. METATI-M: An Introduction. *Formal Aspects of Computing,* 7(5):533-549, 1995.

[Barringer et al, 1996] H. Barringer, M. Fisher, D. Gabbay, R. Owens, and M. Reynolds, editors. *The Imperative Future: Principles of Executable Temporal Logics.* Research Studies Press, 1996.

[Benerecetti *et al,* 1997] M. Benerecetti, F. Giunchiglia, and L. Serafini. Model Checking Multiagent Systems. To appear in Journal of Logic and Computation, 1997.

[Blackburn and de Rijke, 1997] P. Blackburn and M. de Rijke. Why Combine Logics? *Studia Logica,* 59:5-27, 1997.

[Bratman *et al,* 1988] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence,* 4:349-355, 1988.

[Dixon *et al,* 1998] C. Dixon, M. Fisher, and M Wooldridge. Resolution for Temporal Logics of Knowledge. *Journal of Logic and Computation,* 8(3):345-372, 1998.

[Emerson, 1990] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor. *Handbook of Theoretical Computer Science,* pages 996-1072. Elsevier, 1990.

Fagin *et aL,* 1996] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning About Knowledge,* MIT Press, 1996.

[Fisher, 1995] M. Fisher. Representing and Executing Agent-Based Systems. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents.* Springer-Verlag, 1995.

[Fisher, 1997al M. Fisher. A Normal Form for Temporal Logic and its Application in Theorem-Proving and Execution. *Journal of Logic and Computation,* 7(4), 1997.

[Fisher, 1997b) M. Fisher. Implementing BDI-like Systems by Direct Execution. In *Proc. IJCAI-97.* Morgan-Kaufmann, 1997.

[Ghidini, 1998] C. Ghidini. Tableaux for Multi-Context Logics. (Unpublished manuscript.), 1998.

[Giunchiglia and Ghidini, 19981 F. Giunchiglia and C. Ghidini. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. In *Proc. KR-98.* Morgan Kaufmann, 1998.

IGiunchiglia and Serafini, 1994) F Giunchiglia and L. Serafini. Multilanguage hierarchical logics (or: how we can do without modal logics). *Artificial Intelligence,* 65:29-70, 1994.

[Giunchiglia *et aL,* 1993] F. Giunchiglia, L. Serafini, E. Giunchiglia, and M. Frixione. Non-Omniscient Belief as Context-Based Reasoning. In *Proc. IJCAI-93.* Morgan-Kaufmann, 1993.

[Halpern and Moses, 1992] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence,* 54:319-379, 1992.

[Plaisted and Greenbaum, 1986] D. A. Plaisted and S. A. Greenbaum. A Structure-Preserving Clause Form Translation. *Journal of Symbolic Computation,* 2(3):293-304, 1986.

[Rao and Georgeff, 1991] A. S. Rao and M. P. Georgeff. Modeling Agents within a BDI-Architecture. In *Proc. KR-91.* Morgan Kaufmann, 1991.

[Wooldridge *et al,* 1998] M. Wooldridge, C. Dixon, and M. Fisher. A Tableau-Based Proof Method for Temporal Logics of Knowledge and Belief. *Journal of Applied Non-Classical Logics,* 8(3):225~258, 1998.