A Complete and Decidable Logic for Resource-Bounded Agents

Natasha Alechina Brian Logan Mark Whitsey
School of Computer Science
University of Nottingham
Nottingham, UK
{nza,bsl,mtw}@cs.nott.ac.uk

Abstract

We propose a context-logic style formalism, Timed Reasoning Logics (TRL), to describe resource-bounded reasoners who take time to derive consequences of their knowledge. The semantics of TRL is grounded in the agent's computation, allowing an unambiguous ascription of the set of formulas which the agent actually knows at time t. We show that TRL can capture various rule application and conflict resolution strategies that a rule-based agent may employ, and analyse two examples in detail: TRL(STEP) which models an all rules at each cycle *strategy similar to that assumed in step logic [5], and TRL(CLIPS) which models a* single rule at each cycle *strategy similar to that employed by the CLIPS [22] rule based system architecture. We prove a general completeness and decidability results for TRL(STEP).*

1. Introduction

The main problem we address in this paper is that of modelling time bounded reasoners, namely agents which are able to produce plans or derive consequences of their beliefs but take time to deliberate. Most research in logics for belief, knowledge and action (see, for example, [15, 7, 12, 14, 19, 20, 8, 17, 23, 21]) makes the strong assumption that whatever reasoning abilities an agent may have, the results of applying those abilities to a given problem are available immediately. For example, if an agent is capable of reasoning from its observations and some restricted set of logical rules, it will derive all the consequences which are derivable using its rules instantaneously.

In some situations this is a reasonable assumption. For example, the agent may do a very simple kind of deliberation in a non time-critical environment, and we may safely ignore a small delay involved in deliberation. However, there are many cases where the time taken to do deliberation is of critical importance. One obvious example is that of planning in a dynamic environment: an agent may be able to produce a perfectly good plan to reach its goal, but if planning takes too long the result may be irrelevant, e.g., if the problem that the agent is trying to solve has changed. Similarly, if we have a theorem-proving agent whose sole purpose is to check whether something is a tautology, assuming that it already knows all tautologies defeats the purpose of modelling. The kind of logical results we want to be able to prove are therefore of the form *agent i is capable of reaching conclusion ϕ within time bound t.*

In this paper we present Timed Reasoning Logics (TRL), a context-logic style formalism for describing resource bounded reasoners who take time to derive the consequences of their knowledge. This meta-logic is similar to the logic described in [11], but is, in addition, decidable. The semantics of TRL uses syntactic notions but is grounded in the agent's computation [24] (e.g., the values of the agent's internal variables or the set of facts in the agent's working memory), allowing an unambiguous ascription of the set of formulas which the agent actually knows at time t. Our logic is parameterised by the agent's rule application strategy, and we show how to model two strategies: an *all rules at each cycle* strategy similar to that assumed in step logic [5], and a *single rule at each cycle* strategy similar to that employed by the CLIPS [22] rule based system architecture.

The remainder of this paper is organised as follows. In the next section we survey some related work and motivate the development of TRL. In section 3 we introduce the models and language of TRL. The rest of the paper contains formal development of TRL(STEP) (completeness, decidability and embedding in step logic) and TRL(CLIPS).

2. Motivation

In this section, we briefly review some related work and motivate the development of TRL.

The literature contains many attempts at providing a logic of limited or restricted reasoning. However most of these do not explicitly take account of time. For example, Levesque's [16] logic of implicit and explicit belief restricts an agent's explicit beliefs (the classical possible worlds notion) by allowing non-classical (either incomplete or impossible) worlds to enter an agent's epistemic accessibility relation. Although agents need not then believe all classical tautologies, they remain perfect reasoners in relevance logic. In [6] Fagin & Halpern propose an alternative approach to restricting possible worlds semantics which involves a syntactic *awareness* filter, such that an agent only believes a formula if it (or its subterms) are in his awareness set. Agents are modelled as perfect reasoners whose beliefs are restricted to some syntactic class compatible with the awareness filter. Konolige [12] represents beliefs as sentences belonging to an agent's belief set, which is closed under the agent's deduction rules. A deduction model assigns a set of rules to each agent, allowing representation of agents with differing reasoning capacities within a single system. However the deduction model tells us what a set of agents will believe after an indefinitely long period of deliberation.

The only logical research we are aware of which represents reasoning as a process that explicitly requires time is *step logic* [3, 5, 4]. However, until recently, step logic lacked adequate semantics. In [18] Nirkhe, Kraus & Perlis propose a possible-worlds type semantics for step logic. However this re-introduces logical omniscience: once an agent learns that ϕ, it simultaneously knows all logically equivalent statements. In more recent work [11], Grant, Kraus & Perlis propose a semantics for step logic which does not result in logical omniscience, and prove soundness and completeness results for families of theories describing timed reasoning. However, their logic for reasoning about time-limited reasoners is first-order and hence undecidable (even if the agents described are very simple).

We therefore propose a new approach, Timed Reasoning Logics, which avoids the problem of logical omniscience and is at the same time decidable. Not surprisingly, in order to avoid logical omniscience, a logic for reasoning about beliefs has to introduce syntactic objects representing formulas in its semantics. In [11], domains of models of the meta-logic for reasoning about agents contain objects corresponding to formulas of the agent's logic. We have chosen a different approach, where models correspond to sets of agent's states together with a transition relation (similar to [8]). States are identified with finite sets of formulas and the transition relation is computed using the agent rules (for a detailed discussion, see [1]). An advantage of our approach is that the semantics of the logic is *grounded* (see [24]) in the state of the agent (e.g., the values of the agent's internal variables or the set of facts in the agent's working memory), allowing an unambiguous ascription of the set of formulas which the agent actually knows. Similarly, the rules comprising the agent's program give rise to a set of inference rules specifying how one set of formulas can be transformed into another. For the purposes of this paper, we assume that agents are deterministic, so the transition relation is actually a function.

An often-made objection to syntactic or sentential approaches to representing beliefs is that an agent may believe, e.g., $p \wedge q$ but not believe $q \wedge p$. From our point of view, this is not paradoxical: an agent equipped with a conjunction commutativity rule will derive $q \wedge p$ from $p \wedge q$ at the next tick of the clock but would require some non-trivial computational effort to establish that two formulas, each thousands of symbols long, are permutation instances of each other. It is exactly this correlation between the difficulty of the task and the number of steps required by a given agent to solve it that we are interested in investigating.

In what follows, we assume that an agent repeatedly executes a fixed *sense-think-act* cycle. Information obtained by observation (and any *a priori* knowledge) is stored in the agent's working memory. At each tick of the clock, the agent executes its program which consists of a set of condition-action rules. The rules are matched against the contents of the agent's working memory and a subset of the rules are fired. This may update working memory and/or trigger some external action in the agent's environment. In general, the conditions of a rule can be consistently matched against the items in working memory in more than one way, giving rise to a number of distinct *rule instances*. Following standard rule based system terminology we call the set of rule instances the *conflict set* and the process of deciding which subset of rule instances are to be fired at any given cycle *conflict resolution*.

Agents can adopt a wide range of rule application and conflict resolution strategies. For example, they can order the conflict set and fire only the first instance in the ordering at each cycle, or they can fire all rule instances in the conflict set on each cycle once (as step logic does), or they can repeatedly compute the conflict set and fire all the rule instances it contains until no new facts can be derived at the current cycle. We call these three strategies *single rule at each cycle*, *all rules at each cycle*, and *all rules to quiescence* respectively. TRL allows us to distinguish between these different rule application and conflict resolution strategies and reason about the implications of adopting different strategies, for example, the point at which a particular fact will be derived or whether it will ever be derived at all. In section 4 we consider the simple case in which the agent applies all its rules once to all the premises that match (as in step logic). In section 6, we consider a CLIPS-style agent that fires a single rule at each cycle, and show that this requires a non-monotonic logic.

3. Timed Reasoning Logics (TRL)

We define a family of logics called TRL parametrised by a set of agents A and a rule system (set of inference rules and associated rule application strategy) for each agent.

3.1. TRL models

To be able to reason about steps in deliberation and the time deliberation takes, we need a set of steps, or logical time points, which we will assume to be the set of natural numbers. To be able to reason about several agents, we also have a non-empty set of agents or reasoners $A = \{1, \ldots, i, \ldots, \}$. Each state in the model is going to be indexed by an element of the index set $I = A \times \mathbb{N}$, which is the set of pairs (i, t), where i is an agent and t is the step number.

Each agent $i \in A$ has a local state which we assume can be described by a finite set of formulas in some logical language (propositional, predicate, modal, etc.). Different agents may use different languages. To be able to model changes in the agent's language, such as acquiring new names for things etc., we also index the language by time points: at time t, agent i speaks the language $\mathcal{L}^i_t$. We identify the local state of agent i at time t, m^i_t, with a finite set $\{\phi_1, \ldots, \phi_n\}$ of formulas of the agent's language at time $t, \mathcal{L}^i_t$. At this point we don't require anything else in addition to finiteness, in particular this set may be empty or inconsistent.

A TRL model is a set of local states indexed by pairs (i, t). In addition, a TRL model should satisfy constraints which make it a valid representation of a run of a multi-agent system. To formulate those constraints, we need additional notions of observation and inference which constrain how the next state of an agent is going to look.

Each agent has some rules to produce a new state given its current state and any new beliefs obtained by observation. To model observation, we equip each model with a function obs which takes a step t and an agent i as arguments and returns a finite set of formulas in the agent's language at that step. This set is added to the agent's state at the same step (observations are instantaneous). To model the agent's computation of a new state, we have a set of functions inf_i, one for each agent i, which maps a finite set of formulas in the language $\mathcal{L}^i_t$ to another finite set of formulas in the language $\mathcal{L}^i_{t+1}$.

Definition 1 *Let A be a set of agents and $\{\mathcal{L}^i_t : i \in A, t \in \mathbb{N}\}$ a set of agent languages. A TRL model M is a tuple $\langle obs, inf_i, \{m^i_t : i \in A, t \in \mathbb{N}\}\rangle$ where obs is a function which maps a pair (i, t) to a finite set of formulas in $\mathcal{L}^i_t$, inf_i is a function from finite sets of formulas in $\mathcal{L}^i_t$ to finite sets of formulas in $\mathcal{L}^i_{t+1}$, and each m^i_t is a finite set of formulas in $\mathcal{L}^i_t$ such that $m^i_{t+1} = inf_i(m^i_t) \cup obs(i, t+1)$.*

3.2. TRL syntax

Our choice of syntax is influenced by context logics as defined for example in [10] and Gabbay's Labelled Deductive Systems [9].

Well formed formulas in the agent's languages $\mathcal{L}^i_t$ are defined in the usual way. For example, if $\mathcal{L}^a_0$ (the agent a's language at time 0) is a simple propositional logic with propositional variables $p_0, p_1, \ldots, p_n$, then a well formed formula ϕ of $\mathcal{L}^a_0$ is defined as

$$\phi = p_i | \neg\phi | \phi \rightarrow \phi | \phi \wedge \phi | \phi \vee \phi$$

As in context logic, we use labelled formulas to distinguish between beliefs of different agents at different times. If i is an agent, t is a moment of time, and ϕ a well-formed formula of the language $\mathcal{L}^t_i$, then $(i, t) : \phi$ is a well-formed labelled formula of TRL.

A labelled formula $(i, t) : \phi$ is true in a model, $M \models (i, t) : \phi$, iff $\phi \in m^i_t$ (the state indexed by (i, t) in M contains ϕ). A labelled formula $(i, t) : \phi$ is valid, $\models (i, t) : \phi$, iff for all models $M, M \models (i, t) : \phi$. Let Γ be a set of labelled formulas. Γ logically entails $(i, t) : \phi, \Gamma \models (i, t) : \phi$, if in all models where Γ is true, $(i, t) : \phi$ is true.

4. TRL(STEP)

In this section we model an agent which uses a step logic-style *all rules at each cycle* rule application strategy. At each step the agent applies all of its rules to all the formulas which match them. This results in a simple and natural meta-logic which describes the agent's reasoning in time.

The syntax of TRL(STEP) rules is as follows:

$$\frac{(i_1, t) : \phi_1, \ldots, (i_n, t) : \phi_n}{(i, t+1) : \psi}$$

Here, t is a universally quantified variable over time points, and $i_1, \ldots, i_n, i$ are fixed labels corresponding to names of agents.

Let R be a set of TRL(STEP) inference rules. A labelled formula $(i, t) : \phi$ is *derivable* from a set of labelled formulas Γ using R ($\Gamma \vdash_R (i, t) : \phi$) if there is a sequence of labelled formulas $(i_1, t_1) : \phi_1, \ldots, (i_n, t_n) : \phi_n$ such that:

1. each formula in the sequence is either a member of Γ, or is obtained from Γ by one of the inference rules in L; and
2. the last labelled formula in the sequence is $(i, t) : \phi$, namely $(i_n, t_n) : \phi_n = (i, t) : \phi$.

It is convenient to distinguish two kinds of TRL(STEP) rules. The first kind of rule involves just one agent and corresponds to this agent's internal inference rules (inf_i function). We call these rules *internal rules*.

For example, an agent i who can use modus ponens will have a rule MP:

$$\frac{(i,t) : \phi \quad (i,t) : \phi \rightarrow \psi}{(i,t+1) : \psi}$$

This means, for any moment of time t, and formulas ϕ, ψ, if i believes ϕ and $\phi \rightarrow \psi$ at t, then at $t+1$ it will believe ψ. If we want to express that the reasoner i is monotonic, we add a rule:

$$\frac{(i,t) : \phi}{(i,t+1) : \phi}$$

Given a set of rules R, we will refer to the subset corresponding to internal rules as R_{inf}.

The second kind of rule involves several agents and corresponds to exchange of information between agents, which we model using the obs function. We call these rules *communication rules*.

In principle, interaction between agents should be modelled by describing how one agent's decisions result in actions which cause certain changes in the environment, which may then be noticed by other agents. In this paper, we want to concentrate solely on modelling resource-bounded reasoning, avoiding reasoning about actions. For simplicity, we assume that parts of agents' states are directly observable by other agents; when a formula is placed there, it is observed at the next time point by the other agents. This model corresponds to perfect broadcast communication with a fixed one tick delay.

Communication rules have the form:

$$\frac{(i,t) : \phi}{(j,t+1) : \psi}$$

For example, if the whole of agent i's state is observable by agent j, we can have a rule which says that whenever i believes ϕ at t, at $t+1$ j will believe that i believes ϕ:

$$\frac{(i,t) : \phi}{(j,t+1) : B_i\phi}$$

Here, the language of the agent j contains a belief operator B_i, and $B_i\phi$ the stands for 'i believes that ϕ'. Communication rules correspond to constraints on the obs function. We will refer to them as R_{obs}.

4.1. Completeness and decidability of TRL(STEP)

We say that a model M *conforms* to a set of TRL(STEP) rules R if

1. For every rule in R_{inf} of the form

$$\frac{(i,t) : \phi_1, \ldots, (i,t) : \phi_n}{(i,t+1) : \psi}$$

inf_i in M satisfies the property

$$\phi_1, \ldots, \phi_n \in m_t^i \Longrightarrow \psi \in inf_i(m_t^i)$$

in other words, inf_i is computed using all, and only, the rules in R_{inf}.

2. for each rule in R_{obs} of the form

$$\frac{(i,t) : \phi}{(j,t+1) : \psi}$$

obs in M satisfies the property

$$\phi \in m_t^i \Longrightarrow \psi \in obs(j,t+1)$$

We are going to prove a general completeness result, that a TRL(STEP) system characterised by a certain set of rules R is complete with respect to the set of models conforming to R. Before doing this, we need one more notion, similar to the notion of a knowledge-supported model in [11]:

Definition 2 *A TRL model M conforming to set of TRL(STEP) rules R is a minimal model for a set of labelled formulas Γ if for every i, t and ϕ, $\phi \in m_t^i$ iff one of the following holds:*

1. *there is a rule in R_{inf} of the form*

$$\frac{(i,t) : \phi_1, \ldots, (i,t) : \phi_n}{(i,t+1) : \phi}$$

and $\phi_1, \ldots, \phi_n \in m_{t-1}^i$ (in other words, ϕ is forced by the inf function)

2. *or $\phi \in obs(i,t)$ in which case $(i,t) : \phi \in \Gamma$ or there is a rule in R_{obs} of the form*

$$\frac{(j,t) : \psi}{(i,t+1) : \phi}$$

and $\psi \in m_{t-1}^j$.

Lemma 1 *Let M be a minimal model for Γ conforming to R. Then for every formula ϕ, $\phi \in m_t^i$ iff $\Gamma \vdash_R (i,t) : \phi$.*

Proof. The proof goes by induction on t. If $t = 0$, then the only way $\phi \in m_0^i$ is because $\phi \in obs(i,0)$ hence $(i,0) : \phi \in \Gamma$ so $\Gamma \vdash_R (i,0) : \phi$. Inductive hypothesis: suppose that for all agents j and all $s \leq t$, $\phi \in m_s^j$ iff $\Gamma \vdash_R (j,s) : \phi$. Let $\phi \in m_{t+1}^i$. Then either $\phi \in inf_i(m_t^i)$ or $\phi \in obs(i,t+1)$. In the former case, there is a rule in R of the form

$$\frac{(i,t) : \phi_1, \ldots, (i,t) : \phi_n}{(i,t+1) : \psi}$$

such that $\psi = \phi$ and $\phi_1, \ldots, \phi_n \in m_t^i$. By the inductive hypothesis, $\Gamma \vdash_R (i,t) : \phi_i$. Hence by this same rule, $\Gamma \vdash_R (i,t+1) : \phi$. In the latter case, either $(i,t+1) : \phi \in \Gamma$

hence $\Gamma \vdash_R (i, t+1) : \phi$, or there is a rule in R_{obs} of the form

$$\frac{(j,t) : \psi}{(i, t+1) : \phi}$$

and $\psi \in m_t^j$. In this case, by the inductive hypothesis, $\Gamma \vdash_R (j,t) : \psi$ so by the rule above, $\Gamma \vdash_R (i, t+1) : \phi$. $\dashv$

Theorem 1 *Given a set of TRL(STEP) rules R, for any finite set of labelled formulas Γ and a labelled formula ϕ, $\Gamma \vdash_R \phi$ iff $\Gamma \models_{\mathcal{R}} \phi$ where $\mathcal{R}$ is the set of models conforming to R.*

Proof. Soundness ($\Gamma \models_{\mathcal{R}} \phi \Rightarrow \Gamma \vdash_R \phi$) is standard: clearly, in a model conforming to R the rules in R preserve validity.

Completeness: suppose $\Gamma \models_{\mathcal{R}} \phi$. Consider a minimal model for Γ, M_Γ, conforming to R. Since $\Gamma \models_{\mathcal{R}} \phi$ and our particular model M_Γ conforms to R and satisfies Γ, $M_\Gamma \models \phi$. From Lemma 1, $\Gamma \vdash_R \phi$. $\dashv$

Theorem 2 *Given a set of TRL(STEP) rules R, for any finite set of labelled formulas Γ and a labelled formula ϕ, it is decidable whether $\Gamma \vdash_R \phi$ or $\Gamma \models_{\mathcal{R}} \phi$ where $\mathcal{R}$ is the set of models conforming to R.*

Proof. From the Theorem 1 above, the two questions whether $\Gamma \vdash_R (i,t) : \phi$ and whether $\Gamma \models_{\mathcal{R}} (i,t) : \phi$ where $\mathcal{R}$ is the set of models conforming to R, are equivalent. Consider a minimal model M_Γ for Γ. If $\Gamma \models_{\mathcal{R}} (i,t) : \phi$, then $\phi \in m_t^i$ in M_Γ. On the other hand, from Lemma 1, if $\phi \in m_t^i$ then $\Gamma \vdash_R (i,t) : \phi$. Hence $\phi \in m_t^i$ iff $\Gamma \vdash_R (i,t) : \phi$ iff $\Gamma \models_{\mathcal{R}} (i,t) : \phi$.

It is easy to see that given that Γ is finite and rules in R only produce a finite number of new formulas at each step, the initial segment of M (up to step t) can be constructed in time bounded by a tower of exponentials in $|\Gamma|$ of height t (but nevertheless bounded). Then we can inspect m_t^i to see if ϕ is there. $\dashv$

5. Embedding into step logic

In this section we embed TRL(STEP) in the logic introduced by Grant, Kraus and Perlis in [11]. Grant, Kraus and Perlis consider a hierarchy of first order languages. The agents reason in the language L_{ag}, and the meta-logic is formulated in the language L_{me}. L_{me} has a name $\ulcorner\phi\urcorner$ for every $\phi \in L_{ag}$, a 3-ary predicate symbol K where $\mathsf{K}(i, t, \ulcorner\phi\urcorner)$ means that the agent i knows ϕ at time t (and other intentional predicates which we omit here), and functions on terms corresponding to names of formulas: neg, $conj$ and imp. It is assumed that the resulting terms are interpreted as expected, that is $[[neg(\ulcorner\phi\urcorner)]] = \neg\phi$, $[[conj(\ulcorner\phi\urcorner, \ulcorner\psi\urcorner)]] = \phi \wedge \psi$ and $[[imp(\ulcorner\phi\urcorner, \ulcorner\psi\urcorner)]] = \phi \rightarrow \psi$ in all structures.

A typical axiom of the meta-logic in the language of L_{me} is of the form

$$\forall i, t, x, y, z \,.\, \psi(i, t, x, y) \rightarrow \mathsf{K}(i, t{+}1, z)$$

For instance, the fact that all agents can use modus ponens would be represented as the following axiom:

MP $\quad \forall i, t, x_1, x_2 \,.\, \mathsf{K}(i, t, x_1) \wedge \mathsf{K}(i, t, imp(x_1, x_2) \rightarrow \mathsf{K}(i, t{+}1, x_2)$

Most of the axioms given in [11] are Horn clauses (they contain at most one positive K atom). Note that every Horn clause axiom of the form

$$\forall i, t, x_1, \ldots, x_n, x \bigwedge_j \mathsf{K}(i, t, x_j) \rightarrow \mathsf{K}(i, t{+}1, x)$$

corresponds to a TRL(STEP) inference rule of the form

$$\frac{(i,t) : \phi_1, \ldots, (i,t) : \phi_n}{(i, t+1) : \phi}$$

However L_{me} is a much more expressive language than TRL(STEP). For example, axioms which have negations of K atoms in the premises of the rules cannot be expressed as TRL(STEP) rules. Grant, Kraus and Perlis prove that every theory $\mathcal{T}$ which consists of their axioms and a set of observation axioms (ground atoms corresponding to observations), has a minimal Herbrand model $\mathcal{H}$. They also define a special kind of model called *knowledge supported model*. We modify their definition for the case when K is the only predicate in the language of L_{me}:

Definition 3 *A model M of $\mathcal{T}$ is knowledge supported if $M \models \mathsf{K}(i, t, \ulcorner\phi\urcorner)$ only if $\mathsf{K}(i, t, \ulcorner\phi\urcorner)$ is an observation axiom of $\mathcal{T}$, or there exists an axiom of the form $\forall \bar{t}(\phi(\bar{t_1}, \bar{t_2}) \rightarrow \mathsf{K}(\bar{t_3}))$ and a substitution θ such that $\mathsf{K}(\bar{t_3})\theta = \mathsf{K}(i, t, \ulcorner\phi\urcorner)$ and $M \models \phi(\bar{t_1}, \bar{t_2})\theta$.*

They prove that the minimal Herbrand model of $\mathcal{T}$ is knowledge supported, and prove soundness and completeness of their meta-logic using this fact.

The notion of a knowledge supported model is very similar to the minimal model for a set of TRL(STEP) formulas Γ introduced in the previous section. We can make this precise as follows. Fix a set of Horn clause axioms $\mathcal{A}$ in the language of L_{me}. Let $R_{\mathcal{A}}$ be the set of labelled inference rules in the language of TRL(STEP) which corresponds to $\mathcal{A}$. Let Γ_{me} be a set of ground atoms of the form $\mathsf{K}(i, t, \ulcorner\phi\urcorner)$, and $\Gamma_{TRL} = \{(i,t) : \phi : \mathsf{K}(i, t, \ulcorner\phi\urcorner) \in \Gamma_{me}\}$.

Theorem 3 *Let M_{me} be a knowledge supported model of $\mathcal{A} \cup \Gamma_{me}$ and M_{TRL} be a minimal model of Γ_{TRL} conforming to $R_{\mathcal{A}}$. For any formula ϕ of L_{ag},*

$$M_{me} \models \mathsf{K}(i, t, \ulcorner\phi\urcorner) \text{ iff } M_{TRL} \models (i,t) : \phi$$

Proof. Similar to the proof of Lemma 1. $\dashv$

This essentially defines an embedding of TRL(STEP) into a meta-logic of Grant, Kraus and Perlis. As is to be expected, this shows that TRL(STEP) is a less expressive logic and explains why it is decidable while the logic defined in [11] is not. However, we believe that TRL(STEP) has additional advantages over the approach of Grant et al. In TRL, we essentially reason about state transition systems, and while the states are somewhat unorthodox (being collections of formulas) the 'temporal' part of the logic is perfectly standard and computationally feasible [1]. On the other hand, the first order approach (where names of formulas, agents, and moments of time are all objects of the individual domain and the meta-logic includes operations of arithmetic to reason about time points) leads to a complex system even for simple agents. We also believe that a (non-monotonic) logic for an agent using CLIPS-style rule application strategy (as described in the next section), is much easier to handle within TRL than within an axiomatic system.

6. TRL(CLIPS)

Grant, Kraus & Perlis consider agents who are either guaranteed to apply a rule within n steps, or are guaranteed to apply it 'eventually'. However this is not the only or even the most natural rule application strategy which a rule-based agent may use. It is also interesting to investigate rule application strategies motivated by existing rule based system architectures, e.g., CLIPS [22] and SOAR [13].

As an example, we show how to model one of the conflict resolution strategies of the CLIPS rule based system [22]. In CLIPS each rule has a *salience*, (reflecting its importance in problem solving) and each fact in working memory has a *time stamp* which records the cycle at which the fact was added to working memory. CLIPS uses a *single rule at each cycle* rule application strategy. At each cycle, all rules are matched against the facts in working memory and any new rule instances are added to the conflict set. Rule matching is refractory, i.e., rules don't match against the same set of premises more than once. New rule instances are placed above all rule instances of lower salience and below all rules of higher salience. If rule instances have equal salience, ties are broken by the conflict resolution strategy. The default strategy, called *depth*, gives preference to rule instances which matched against more recent facts. Once the conflict set has been computed, CLIPS fires the first rule instance in the conflict set at each cycle.

As an example, consider an agent with the following set of rules:

```
R1: dalmatian(x) -> dog(x)
```

1 TRL(STEP) can be embedded in a linear time temporal logic with 'next' operator and a non-standard epistemic modality, similar to the one developed in [2].

```
R2: dog(x) -> dangerous(x)
```

R1 has greater salience than R2. The agent's working memory contains the following fact:

```
0: dalmatian(snoopy)
```

which is tagged with the time (0) at which it was asserted into working memory. Then at the next cycle an agent with a CLIPS depth-style conflict resolution strategy would derive

```
1: dog(snoopy)
```

Assume that at this cycle the agent makes a new observation and a corresponding fact is asserted into working memory:

```
1: dalmatian(spot)
```

Instances of R1 have greater salience than instances of R2, so on the following cycle the agent will derive

```
2: dog(spot)
```

Both "`dog(snoopy)`" and "`dog(spot)`" match R2, but "`dog(spot)`" will be preferred since it has a higher (more recent) time stamp than "`dog(snoopy)`". On the following cycle the agent will derive

```
3: dangerous(spot)
```

Finally the agent derives:

```
4: dangerous(snoopy)
```

This is trivial example. However, in general, the time at which a fact is derived can be significant. For example, in developing an agent we may wish to ensure that it responds to dangers as soon as they are perceived rather than after classifying objects in the environment. In our short example, the delay in identifying danger is just one step, but it is easy to modify the example to make the delay arbitrarily long (by introducing n new dalmatians instead of one at cycle 1).

In section 4 TRL(STEP) logic was formulated to model agents which used the *all rules at each cycle* rule application strategy. It is easy to see that in the case of the *single rule at each cycle* strategy, the corresponding TRL logic becomes non-monotonic. For instance, in the example above the agent would have derived "`dangerous(snoopy)`" at step 2 if the fact "`dalmatian(spot)`" had not been asserted. Defining a logic corresponding to the *single rule at each cycle* strategy is an interesting challenge, to which we devote the rest of this section. We call the logic TRL(CLIPS) and for simplicity we formulate it here for a single agent, since the only difference from TRL(STEP) is in the internal rules of the agent.

To reflect salience of rules, we assume that there is a partial order $\leq_r$ on the set of rules $R_1, \ldots, R_n$, and introduce a meta-logical abbreviation $top(R_i, \Delta)$ to mean that for a given a set of labelled formulas Δ, R_i is maximal in the order $\leq_r$ among the rules which match any premises from Δ.

When there are several rule instances for rules R_i satisfying $top(R_i, \Delta)$, we need to order them, this time by a total order, and choose the maximal element in the order as the rule instance to apply. This total order corresponds to the agent's conflict resolution strategy. We have chosen to model the depth strategy, but other strategies can be treated in a similar way.

To reflect the depth rule application strategy, we need to incorporate time stamps explicitly in the language. They seem to correspond quite closely to the step labels, but in fact the information they carry is different: $(i, t) : \phi$ means that ϕ is in the agent i's state at time t, and the time stamp corresponds to the time when the information ϕ was first acquired. Let us assume that the agent's language is first order and all predicates are augmented with an extra argument for a time stamp, for example $Dalmatian(snoopy, 0)$. Define the order $<_d$ (depth order on sets of premises) as follows: $\phi_1, \dots, \phi_n <_d \psi_1, \dots, \psi_n$ if

1. $t_1, \dots, t_n$ are time stamps of $\phi_1, \dots, \phi_n$ (ordered in decreasing order of timestamps), $s_1, \dots, s_n$ are time stamps of $\psi_1, \dots, \psi_n$ (also ordered), and $t_1, \dots, t_n < s_1, \dots, s_n$ (pointwise), or
2. $t_1, \dots, t_n = s_1, \dots, s_n$ but $\phi_1, \dots, \phi_n <_a \psi_1, \dots, \psi_n$ in some arbitrary (for example lexicographic) total order $<_a$.

We introduce another meta-logical abbreviation $top_d(\phi_1, \dots, \phi_m, \Delta)$ to indicate that the set of premises $\phi_1, \dots, \phi_m$ match a rule R_i which has the property $top(R_i, \Delta)$ and are, furthermore, maximal in the $<_d$ order among such sets of premises.

The rules of a *single rule at each cycle* agent i using the depth strategy then become (for $\phi_1, \dots, \phi_n$ imply ψ):

$$\frac{\begin{array}{c}(i,t) : \Delta,\ (i,t) : \phi_1,\ \dots,\ (i,t) : \phi_n \\ \text{and } top_d((i,t) : \phi_1, \dots, (i,t) : \phi_n, (i,t) : \Delta)\end{array}}{(i, t+1) : \Delta,\ (i, t+1) : \psi}$$

Here, $(i, t) : \Delta$ is a set of formulas labelled (i, t) which contains all formulas available at step (i, t) in the derivation. Note that $\phi_1, \dots, \phi_n \in \Delta$, so $(i, t+1) : \Delta$ includes $(i, t+1) : \phi_1, \dots, (i, t+1) : \phi_n$ (we do not throw premises away).

For example, the agent from the example above has rules R1:

$$\frac{\begin{array}{c}(i,t) : \Delta,\ (i,t) : Dalmatian(x, s) \\ \text{and } top_d((i,t) : Dalmatian(x, s), (i,t) : \Delta)\end{array}}{(i, t+1) : \Delta,\ (i, t+1) : Dog(x, t+1)}$$

and R2:

$$\frac{\begin{array}{c}(i,t) : \Delta,\ (i,t) : Dog(x, s) \\ \text{and } top_d((i,t) : Dog(x, s),\ (i,t) : \Delta)\end{array}}{(i, t+1) : \Delta,\ (i, t+1) : Dangerous(x, t+1)}$$

The notion of the derivation is similar to the one introduced in section 4, although applicability of the rules of agent i at step n depends on $(i, n) : \Delta$. To account for refractoriness we stipulate that repeated application of the same rule to the same (modulo step labels) premises is not allowed. An example derivation is given in the appendix.

Analogously to TRL(STEP), we can prove that it is decidable whether a labelled formula $(i, t) : \phi$ follows from a set of labelled formulas Γ with respect to a set of TRL(CLIPS) rules R (and orderings on rules and formulas).

7. Conclusions

We introduce a family of logics, TRL, to reason about time-bounded reasoners. The semantics uses syntactic notions but is grounded in the agent's computation. We prove a general completeness and decidability result for TRL(STEP) logic which models an *all rules at each cycle* rule application strategy. We show that TRL(STEP) can be embedded in the logic introduced in [11] and although TRL is less expressive it has an advantage of being decidable. To illustrate the flexibility of the TRL framework, we also show how to model agents which fire only one rule at each cycle. The resulting logic TRL(CLIPS) is parametrised by the ordering on the set of matching rule instances (we consider the ordering corresponding to depth strategy of CLIPS in detail) and is non-monotonic. Our future work will involve a more systematic investigation of logics corresponding to various rule application and conflict resolution strategies.

References

[1] N. Alechina and B. Logan. Ascribing beliefs to resource bounded agents. In *Proc. First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, pages 881–888. ACM Press, July 2002.

[2] N. Alechina and B. Logan. Ascribing beliefs to resource bounded agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, volume 2, pages 881–888, Bologna, July 2002. ACM Press.

[3] J. Drapkin and D. Perlis. A preliminary excursion into Step-Logics. *Proceedings of the SIGART International Symposium on Methodologies for Intelligent Systems*, pages 262–269, 1986.

[4] J. Elgot-Drapkin, M. Miller, and D. Perlis. Memory, reason and time: the Step-Logic approach. In R. Cummins and J. Pollock, editors, *Philosophy and AI: Essays at the Interface*, pages 79–103. MIT Press, Cambridge, Mass., 1991.

[5] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

[6] R. Fagin and J. Halpern. Belief, awareness and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[7] R. Fagin and J. Y. Halpern. Belief, awareness, and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490, 1985.

[8] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, Mass., 1995.

[9] D. M. Gabbay. *Labelled Deductive Systems: Volume I - Foundations*. Oxford UNiversity Press, 1996.

[10] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatability. *Artificial Intelligence*, 127(2):221–259, 2001.

[11] J. Grant, S. Kraus, and D. Perlis. A logic for characterizing multiple bounded agents. *Autonomous Agents and Multi-Agent Systems*, 3(4):351–387, 2000.

[12] K. Konolige. *A Deduction Model of Belief*. Morgan Kaufmann, San Francisco, Calif., 1986.

[13] J. E. Laird, A. Newell, and P. S. Rosenbloom. SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33:1–64, 1987.

[14] G. Lakemeyer. Steps towards a first-order logic of explicit and implict belief. In J. Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference*, pages 325–340, San Francisco, Calif., 1986. Morgan Kaufmann.

[15] H. J. Levesque. A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence, AAAI-84*, pages 198–202. AAAI, 1984.

[16] H. J. Levesque. A logic of implicit and explicit belief. In *Proc. National Conference on Artificial Intelligence (AAAI '84)*, pages 198–202, 1984.

[17] R. C. Moore. *Logic and Representation*. Number 39 in CSLI Lecture Notes. CSLI Publications, 1995.

[18] M. Nirkhe, S. Kraus, and D. Perlis. Thinking takes time: a modal active-logic for reasoning in time. Technical Report CS-TR-3249, University of Maryland, Department of Computer Science, 1994.

[19] R. Parikh. Knowledge and the problem of logical omniscience. In *Methodologies for Intelligent Systems, Proceedings of the Second International Symposium*, pages 432–439. North-Holland, 1987.

[20] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, 1991.

[21] M. P. Singh. Know-how. In M. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, pages 81–104. Kluwer Academic, Dordrecht, 1999.

[22] Software Technology Branch, Lyndon B. Johnson Space Center, Houston. *CLIPS Reference Manual: Version 6.21*, June 2003.

[23] W. van der Hoek, B. van Linder, and J.-J. C. Meyer. An integrated modal approach to rational agents. In M. Wooldridge and A. Rao, editors, *Foundations of Rational Agency*, pages 133–168. Kluwer Academic, Dordrecht, 1999.

[24] M. Wooldridge. Computationally grounded theories of agency. In E. Durfee, editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 13–20. IEEE Press, 2000.

A. Example derivation

Here we give a derivation in the logic describing an agent using *single rule at each cycle* rule application strategy with depth conflict resolution of the example from Section 6.

First, assume that $Dalmatian(snoopy, 0)$ is the only information available at step 0.

1. $(i, 0) : Dalmatian(snoopy, 0)$. $(i, 0) : \Delta = \{(i, 0) : Dalmatian(snoopy, 0)\}$. The only rule that matches is R1 and the only matching rule instance is $(i, 0) : Dalmatian(snoopy, 0)$.
2. $(i, 1) : Dog(snoopy, 1)$ from 1 by R1. Now $(i, 1) : \Delta = \{(i, 1) : Dalmatian(snoopy, 0), (i, 1) : Dog(snoopy, 1)\}$. $Dalmatian(snoopy, 0)$ was used with R1 before so $(i, 1) : Dalmatian(snoopy, 0)$ is not a matching rule instance. The only rule that matches is R2 and the only match for R2 is $(i, 1) : Dog(snoopy, 1)$.
3. $(i, 2) : Dangerous(snoopy, 2)$ from 2 by R2.

Now assume that we have $Dalmatian(snoopy, 0)$, $Dog(snoopy, 1)$ and $Dalmatian(spot, 1)$ at step 1:

1. $(i, 1) : Dalmatian(snoopy, 0)$
2. $(i, 1) : Dog(snoopy, 1)$
3. $(i, 1) : Dalmatian(spot, 1)$. Now $(i, 1) : \Delta = \{(i, 1) : Dalmatian(snoopy, 0), (i, 1) : Dog(snoopy, 1), (i, 1) : Dalmatian(spot, 1)\}$. Both R1 and R2 match but $top(R1, \Delta)$ and only one premise matches R1 so $top_d(Dalmatian(spot, 1), (i, 1) : \Delta)$.
4. $(i, 2) : Dog(spot, 2)$ from 3 by R1. Now $(i, 2) : \Delta = \{(i, 2) : Dalmatian(snoopy, 0), (i, 2) : Dog(snoopy, 1), (i, 2) : Dalmatian(spot, 1), (i, 2) : Dog(spot, 2)\}$, $top(R2, \Delta)$ and there are two matches for R2: $(i, 2) : Dog(snoopy, 1)$ and $(i, 2) : Dog(spot, 2)$. Since $(i, 2) : Dog(spot, 2)$ is more recent, $top_d((i, 2) : Dog(spot, 2), (i, 2) : \Delta)$.
5. $(i, 3) : Dangerous(spot, 3)$ from 4 by R2.
6. $(i, 4) : Dangerous(snoopy, 4)$ from 2 by R2 (this is the only rule instance we have left).