Modelling Communicating Agents in Timed Reasoning Logics

Natasha Alechina, Brian Logan, and Mark Whitsey

School of Computer Science and IT, University of Nottingham, UK.
(nza,mtw,bsl)@cs.nott.ac.uk

Abstract. Practical reasoners are resource-bounded—in particular they require time to derive consequences of their knowledge. Building on the Timed Reasoning Logics (TRL) framework introduced in [1], we show how to represent the time required by an agent to reach a given conclusion. TRL allows us to model the kinds of rule application and conflict resolution strategies commonly found in rule-based agents, and we show how the choice of strategy can influence the information an agent can take into account when making decisions at a particular point in time. We prove general completeness and decidability results for TRL, and analyse the impact of communication in an example system consisting of two agents which use different conflict resolution strategies.

1 Introduction

Most research in logics for belief, knowledge and action (see, for example, [2–11]) 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 derives all the consequences of its rules instantaneously.

While this is a reasonable assumption in some situations, there are many cases where the time taken to do deliberation is of critical importance. Practical agents take time to derive the consequences of their beliefs, and, in a dynamic environment, the time required by an agent to derive the consequences of its observations will determine whether such derivations can play an effective role in action selection. Another example involves more standard analytical reasoning and a classical domain for the application of epistemic logics: verifying cryptographic protocols. An agent intercepting a coded message usually has all the necessary "inference rules" to break the code. The only problem is that if the encoding is decent, it would take the intercepting agent millennia to actually derive the answer. On the other hand, if the encryption scheme is badly designed or the key length is short, the answer can be derived in an undesirably short period of time. 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 show how to model the execution of communicating rule-based agents using Timed Reasoning Logics (TRL). TRL is a context-logic style formalism for describing rule-based resource bounded reasoners who take time to derive the consequences of their knowledge. This paper builds on the work in [1], where we introduced

TRL. In that paper, we described how our logic can model different rule application and conflict resolution strategies, and proved soundness and completeness of the logic TRL(STEP) which captures the *all rules at each cycle* rule application strategy used by step logic [12] (for another example of a TRL(STEP) logic, see [13]). We also showed how to model a *single rule at each cycle* strategy similar to that employed by the CLIPS [14] rule-based system architecture, and sketched a logic TRL(CLIPS). In this paper, we prove a general soundness and completeness result for TRL, from which soundness and completeness of TRL(CLIPS) follows. We study TRL(CLIPS) in more detail and give a detailed example involving two communicating agents using different CLIPS conflict resolution strategies.

2 Model of an agent

In this section we outline a simple model of the kind of rule-based agent whose execution cycle we wish to formalise.

A rule-based agent consists of a working memory and one or more sets of condition-action rules. The working memory constitutes the agent's state, and the rules form the agent's program. We assume that agents repeatedly execute a fixed *sense-think-act* cycle. At each tick of the clock, an agent senses its environment and information obtained by sensing is added to the previously derived facts and any a priori knowledge in the agent's working memory. The agent then evaluates the condition-action rules forming its program. The conditions of each rule are matched against the contents of the agent's working memory and a subset of the rules are fired. This typically adds or deletes one or more facts from working memory and/or results in some external actions being performed in the agent's environment. For the purposes of this paper the only external action we assume is a 'communication' action which allows agents to communicate facts currently held in working memory to other agents.

Our interest here is with the rule application and conflict resolution strategy adopted by the agent. 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 rule instance in the ordering at each cycle, or they can fire all rule instances in the conflict set on each cycle once, or they can repeatedly compute the conflict set and fire all the rule instances it contains set 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 quiesence* respectively.

3 Timed Reasoning Logics (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 [2] 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 [15] 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 [4] 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* [16, 17, 12]. 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 [19], 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).

The approach we describe in this paper, Timed Reasoning Logics (TRL), avoids the problem of logical omniscience and is at the same time decidable. TRL is a context-logic style formalism for describing rule-based resource bounded reasoners who take time to derive the consequences of their knowledge. 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 [19], 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.

This paper builds on the work in [1], where we introduced TRL. In this section, we give a slightly more general formulation of TRL than that given in [1], and prove its soundness and completeness.

3.1 TRL Syntax

Our choice of syntax is influenced both by step logics and context logics and by Gabbay's Labelled Deductive Systems [20]. 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 $A = \{a, b, c, i, j, i_1, \ldots, i_n \ldots\}$.

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 $\mathcal{L}_t^i$.

Well formed formulas in the agent's languages $\mathcal{L}_t^i$ are defined in the usual way. For example, if $\mathcal{L}_0^a$ (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 $\mathcal{L}_0^a$ 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 $\mathcal{L}_i^t$, then $(i,t) : \phi$ is a well-formed labelled formula of TRL.

The general form of an inference rule in TRL is:

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

with a possible side condition of the form: provided that $(i_1, t) : \phi_1, \ldots, (i_n, t) : \phi_n$ and the set Δ_t of all formulas derived at the previous stage in the derivation (see Definition 1 below) satisfy some property. For example, a side condition for a defeasible rule may be that some formula is *not* in Δ_t.

A significant restriction on the format of possible TRL rules is that *only finitely many formulas labelled* t should be derivable starting with a finite set of labelled formulas Γ, for any t. For example, supposing we had an operator B_a for "agent a believes that", then the following negative introspection rule:

$$(a, t+1) : \neg B_a\phi \quad \text{given that} \quad (a,t) : \phi \notin \Delta_t$$

cannot be introduced in unrestricted form since it would generate infinitely many formulas at step $t+1$.

A simple example of a TRL rule is an inference rule corresponding to a rule in the agent's program. If agent a's program contains the rule

$$A(x), B(x) \rightarrow C(x)$$

then the corresponding inference rule in TRL would be

$$\frac{(a,t) : A(x), \ (a,t) : B(x)}{(a, t+1) : C(x)}$$

Depending on the agent's rule application strategy, the TRL inference rule may have a side condition stating, for example, that it may only be applied if no other rule is applicable.

Another kind of rule which we will see later is used to model communication between agents. For example,

$$\frac{(a,t) : \phi}{(b, t+1) : B_a\phi}$$

expresses the fact that whenever a believes ϕ, at the next step b believes $B_a\phi$. In this paper, we do not explicitly model message passing. Instead we assume that whenever an agent derives a fact of a certain form it communicates this fact to other agents. The message arrives at the next tick of the clock, and is ‘observed’ immediately. In the example above, whenever a derives ϕ, it sends a message containing ϕ to b, which arrives at $t + 1$. This model corresponds to perfect broadcast communication with a fixed one tick delay.

The derivability relation in a TRL logic may be non-monotonic due to the agent’s rule application strategy (e.g. only one of the rules is applied at each cycle) or to the presence of defeasible rules. Before we give a formal definition of derivability, we need a couple of auxiliary definitions. Let R be a set of TRL rules and $\varDelta$ a finite set of labelled formulas. Then by $R(\varDelta)$ we denote the set of all labelled formulas derivable from $\varDelta$ by one application of a rule in R. Formally, $R(\varDelta)$ is the set of all labelled formulas $(i, t+1) : \phi$ such that there is a rule in R of the form

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

and $(i_1, t) : \phi_1, \ldots, (i_n, t) : \phi_n \in \varDelta$ and any side condition of the rule, holds for $(i_1, t) : \phi_1, \ldots, (i_n, t) : \phi_n$ and $\varDelta$. Finally, given a set of labelled formulas $\varGamma$, we write $\varGamma_k$ for the subset of $\varGamma$ labelled by time point k (formulas in $\varGamma$ of the form $(j, k) : \psi$ for any agent j).

Definition 1. *Given a set of TRL rules R, a labelled formula $(i, t) : \phi$ is derivable using R from a set of labelled formulas $\varGamma$:*

$$\varGamma \vdash_R (i, t) : \phi$$

if there exists a sequence of finite sets *of labelled formulas*

$$\varDelta_0, \varDelta_1, \ldots, \varDelta_t,$$

such that $(i, t) : \phi \in \varDelta_t$ and

1. *$\varDelta_0$ is the union of $\varGamma_0$ and all axioms in R labelled by time 0 (i.e., $(j, 0)$ for some agent j).*
2. *$\varDelta_k$ is the union of $\varGamma_k$ and $R(\varDelta_{k-1})$.*

3.2 TRL Semantics

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, i.e. $\mathcal{L}^i_t$. 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 TRL states. Each local state in a TRL model is 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. 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 the additional notions of observation and inference, which constrain how the next state of an agent will look.

Each agent has a program—a set of rules which it uses to derive its next state given its current state and any new beliefs it obtains by observing the world. We therefore equip each model with an obs function and a set of inf_i functions (one for each agent i). Intuitively, obs models observations, which we take to include inter-agent communication, and 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 (we thus model observations as being believed instantaneously). Each inf_i models agent i's computation of a new state by mapping a finite set of formulas in the language $\mathcal{L}_t^i$ to another finite set of formulas in the language $\mathcal{L}_{t+1}^i$. Intuitively, inf_i takes the tokens in agent i's state at time t, applies the rules in i's program to them to obtain a new set of tokens, which, together with i's observations at time $t+1$, constitute its state at time $t+1$.

Definition 2 (TRL Model). *Let A be a set of agents and $\{\mathcal{L}_t^i : i \in A, t \in \mathbb{N}\}$ a set of agent languages. A TRL model M is a tuple $\langle obs, inf_i, \{m_t^i : i \in A, t \in \mathbb{N}\}\rangle$ where obs is a function which maps a pair (i,t) to a finite set of formulas in $\mathcal{L}_t^i$, inf_i is a function from finite sets of formulas in $\mathcal{L}_t^i$ to finite sets of formulas in $\mathcal{L}_{t+1}^i$, and each m_t^i is a finite set of formulas in $\mathcal{L}_t^i$ such that $m_{t+1}^i = inf_i(m_t^i) \cup obs(i, t+1)$.*

Definition 3 (Satisfaction and Logical Entailment). *A labelled formula $(i,t) : \phi$ is true in a model, written $M \models (i,t) : \phi$, iff $\phi \in m_t^i$ (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.*

3.3 Soundness and completeness of TRL

In this section we prove a general soundness and completeness result for TRL systems. We are going to show that given a set of TRL rules R (the only condition on R is that starting from a finite set of premises, it only produces a finite set of consequences labelled t, for any t) and a set of TRL models S, describing possible runs of a multi-agent system, R is sound and complete with respect to S if, and only if, S is the set of models which *conform* to R in the sense defined below.

Definition 4. *A TRL model M* conforms *to a set of TRL rules R if for every rule in R of the form*

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

possibly with some side condition on Δ_t, M satisfies the property that if for all premises of the rule, $\phi_k \in m_t^k$, and the side condition of the rule holds for $\bigcup_{j \in A} m_t^j$ substituted for Δ_t, then $\psi \in m_{t+1}^i$.

Before proving the main theorem, we need one more notion, similar to the notion of a knowledge-supported model in [19]:

Definition 5. *[Minimal Model] A TRL model M conforming to a set of TRL 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 of the form*

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

   for all premises of the rule, $\phi_k \in m^k_{t-1}$ and the side condition of the rule holds for $\bigcup_{j\in A} m^j_{t-1}$ (in other words, ϕ is in m^i_t because the model conforms to R)
2. *or $(i,t):\phi \in \Gamma$ in which case $\phi \in obs(m^i_t)$.*

A minimal model for Γ only satisfies the formulas in Γ and their logical consequences.

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

Proof. The proof goes by induction on t. If $t=0$, then the only way $\phi \in m^i_0$ 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 $s \leq t$, $\phi \in m^j_s$ iff $\Gamma \vdash_R (j,s):\phi$. Let $\phi \in m^i_{t+1}$. Then either $(i,t+1):\phi \in \Gamma$ hence $\Gamma \vdash_R (i,t+1):\phi$, or there is a rule in R of the form

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

such that $\psi=\phi$ and $\phi_1,\ldots,\phi_n \in m^i_t$ (and the side condition of the rule holds for the set of formulas in the union of all states at time t). By the inductive hypothesis, $\Gamma \vdash_R (i_k,t):\phi_k$. Hence by this same rule, $\Gamma \vdash_R (i,t+1):\phi$.

Theorem 1. *Given a set of TRL 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 all 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, $\Gamma \vdash_R \phi$.

Theorem 2. *Given a set of TRL 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 all models conforming to R.*

Proof. From Theorem 1 above, the 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 $\phi \in m^i_t$ in M_Γ. On the other hand, from Lemma 1, if $\phi \in m^i_t$ then $\Gamma \vdash_R (i,t):\phi$. Hence $\phi \in m^i_t$ 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^i_t to see if ϕ is there.

4 TRL(CLIPS)

As an example of a logical model of an agent based on TRL, we show how to model a simple system consisting of two communicating agents. The agents use a CLIPS-style [14] *single rule at each cycle* rule application strategy. However each agent uses a different CLIPS conflict resolution strategy. We show that the adoption of different conflict resolution strategies by each agent can result in a reduction in the time required to derive information for action selection.

CLIPS has been used to build a number of agent-based systems (see, e.g., [21]). In CLIPS each rule has a *salience* reflecting its importance in problem solving. 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. CLIPS supports a variety of conflict resolution strategies including *depth*, *breadth*, *simplicity*, *complexity*, *lex*, *mea*, and *random*. The default strategy, called *depth*, gives preference to new rule instances; *breadth* places older rule instances higher. Once the conflict set has been computed, CLIPS fires the highest ranking rule instance in the conflict set at each cycle.

Consider an agent with the following set of rules using the *depth* conflict resolution strategy:

```
R1: tiger(x) -> large-carnivore(x)
R2: large-carnivore(x) -> dangerous(x)
```

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

```
0:tiger(c)
```

then at the next cycle the agent would derive

```
1:large-carnivore(c)
```

Assume that at this cycle the agent observes a second tiger, and a corresponding fact is asserted into working memory:

```
1:tiger(d)
```

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

```
2:large-carnivore(d)
```

Both "`large-carnivore(c)`" and "`large-carnivore(d)`" match R2, but "`large-carnivore(d)`" will be preferred since it it is a more recent instance of R2 than "`large-carnivore(c)`". On the following cycle the agent will derive

```
3:dangerous(d)
```

Finally the agent derives:

```
4:dangerous(c)
```

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 tigers instead of one at cycle 1).

It is easy to see that the TRL logic corresponding to the *single rule at each cycle* strategy is non-monotonic. For instance, in the example above, $\{$`0:tiger(c)`$\} \vdash$ `2:dangerous(c)`, but $\{$`0:tiger(c)`, `1:tiger(d)`$\} \nvdash$ `2:dangerous(c)`.

To reflect salience of rules, we assume that there is a partial order $\leq_{j,r}$ on the set of rules $\mathcal{R}_j = \{R_1, \dots, R_n\}$ which correspond to the rules of agent j's program. Note that the logic will contain more rules describing agent j in addition to $\mathcal{R}_j$; e.g. rules which model observation, or the fact that formulas persist in the state. To determine which rule instance will be fired at a given step in a TRL(CLIPS) derivation, we need to compute a 'conflict set' of sets of premises matching rules in $\mathcal{R}_j$, order it by a total order, and fire the rule with the premises which come top in that order. The total order on the conflict set is determined by the agent's conflict resolution strategy.

To be more formal, let Δ_t be the set of all formulas derived at step t. Let $C_{j,t}$ be the the conflict set for j at t, namely $C_{j,t} = \{\langle (j,t) : \phi_1,\ \dots,\ (j,t) : \phi_n, R_i \rangle :$ $(j,t) : \phi_{1\dots n} \in \Delta_t$, $R_i \in \mathcal{R}_j$, and $(j,t) : \phi_1, \dots, (j,t) : \phi_n$ match $R_i\}$.

Define the order $<_{depth}$ (depth order on $C_{j,t}$, to be read as 'lower in the depth order') as follows:

$$\langle (j,t) : \phi_1, \dots, (j,t) : \phi_n, R_i \rangle <_{depth} \langle (j,t) : \psi_1, \dots, (j,t) : \psi_n, R_m \rangle$$

iff

1. $R_i \leq_{r,j} R_m$ (R_i has lower salience); or
2. $R_i =_{r,j} R_m$, but $\langle (j,t) : \phi_1, \dots, (j,t) : \phi_n, R_i \rangle$ is an earlier rule instance, that is, for some Δ_s with $s < t$, $\langle (j,s) : \phi_1, \dots, (j,s) : \phi_n, R_i \rangle \in C_{j,s}$ and $\langle (j,s) : \psi_1, \dots, (j,s) : \psi_n, R_m \rangle \notin C_{j,s}$; or
3. $(j,t) : \phi_1, \dots, (j,t) : \phi_n$ and $(j,t) : \psi_1, \dots, (j,t) : \psi_n$ match rules of the same salience and were added to the conflict set at the same time, but $(j,t) : \phi_1, \dots, (j,t) : \phi_n$ is lower in some arbitrary, e.g., lexicographic, order.

For the breadth order $<_{breadth}$, we reverse the second clause of the definition; now the premises which belong to a conflict set $C_{j,s}$ for the earliest time s are higher in the order.

We introduce meta-logical abbreviation $top_{j,depth}(\phi_1, \dots, \phi_m, \Delta_t)$ and $top_{j,breadth}(\phi_1, \dots, \phi_m, \Delta_t)$ to indicate that the set of premises $\phi_1, \dots, \phi_m$ is the highest in the $<_{depth}$ ($<_{breadth}$) order among the conflict set $C_{j,t}$ of formulas from Δ_t.

Finally, we need to account for the refractoriness of the CLIPS rule application strategy: any rule instance is only used once in the TRL(CLIPS) derivation. To be precise, for any rule R_j and a set of premises $(i,t) : \phi_1, \ldots, (i,t) : \phi_n$ matching this rule, if at some step $s < t$ the rule R_j was fired with a set of premises which were the same but for step label (e.g. $(i,s) : \phi_1, \ldots, (i,s) : \phi_n$), then $(i,t) : \phi_1, \ldots, (i,t) : \phi_n$ are excluded from the conflict set $C_{i,t}$.

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

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

provided $top_{i,depth}((i,t) : \phi_1, \ldots, (i,t) : \phi_n, \Delta_t)$, namely the premises of the rule are maximal in the $<_{depth}$ order in the conflict set for i at t. In what follows, we refer to such a proviso as 'standard proviso for depth order'. For example, the agent a from the example above has a rule:

$$\frac{(a,t) : \mathit{Tiger}(x), \Delta_t}{(a,t+1) : \textit{Large-Carnivore(x)}}$$

provided $top_{a,depth}((a,t) : \mathit{Tiger}(x), \Delta_t)$

For monotonic agents (who keep all the facts they derived earlier) we have an additional monotonicity rule which does not have a side condition, is always applicable, and is excluded from the ordering of the internal agent rules proper:

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

To give an example of an observation rule, suppose that the agent a gets some of its information about the world from agent b. In particular, if b decides that something is nearby, then at the next step a also decides that it is nearby:

$$\frac{(b,t) : \mathit{Near}(x)}{(a,t+1) : \mathit{Near}(x)}$$

This rule also does not have any side conditions.

The notion of derivation in TRL(CLIPS) is a special case of TRL derivation as given in Definition 1.

4.1 Example

In this section we give a worked example of a derivation in TRL(CLIPS). Our example involves two agents, a and b. They have the same set of rules with the same salience order and start with the same set of observations, but a uses the *depth* strategy, while b uses the *breadth* conflict resolution strategy. We show that they both can reach the same conclusion, (classify a tiger as a dangerous object), however if they communicate, they can reach this conclusion faster.

The rules corresponding to the program rules of agent a are (with the standard proviso for depth order):

$$\frac{(a,t) : \mathit{Large}(x), (a,t) : \mathit{Carnivore}(x), (a,t) : \mathit{Near}(x), (a,t) : \mathit{Free}(x), \Delta_t}{(a,t+1) : \mathit{Dangerous}(x)} R1$$

$$\frac{(a,t) : \textit{Bengal-Tiger}(x), \mathit{\Delta}_t}{(a,t+1) : \mathit{Tiger}(x)} R2$$

$$\frac{(a,t) : \mathit{Tiger}(x), \mathit{\Delta}_t}{(a,t+1) : \mathit{Large}(x)} R3$$

$$\frac{(a,t) : \mathit{Tiger}(x), \mathit{\Delta}_t}{(a,t+1) : \mathit{Carnivore}(x)} R4$$

$$\frac{(a,t) : \mathit{Distance} < 5m(x), \mathit{\Delta}_t}{(a,t+1) : \mathit{Near}(x)} R5$$

$$\frac{(a,t) : \neg \mathit{Caged}(x), \mathit{\Delta}_t}{(a,t+1) : \mathit{Free}(x)} R6$$

The rules for agent b are the same, with $top_{a,depth}$ replaced with $top_{b,breadth}$. The salience order on rules is $R1 \geq_r R2 \geq_r \{R3, R4, R5, R6\}$.

In addition, both agents have the monotonicity rule and the following communication rules:

$$\frac{(a,t) : \mathit{Large}(x)}{(b,t+1) : \mathit{Large}(x)}$$

$$\frac{(a,t) : \mathit{Carnivore}(x)}{(b,t+1) : \mathit{Carnivore}(x)}$$

$$\frac{(b,t) : \mathit{Near}(x)}{(a,t+1) : \mathit{Near}(x)}$$

$$\frac{(b,t) : \mathit{Free}(x)}{(a,t+1) : \mathit{Free}(x)}$$

Suppose both agents start with the same set of observations, corresponding to a sighting of a Bengal tiger at a distance less than 5 meters, and apparently uncaged: $(a,0) : \textit{Bengal-Tiger}(c), (a,0) : \mathit{Distance} < 5m(c), (a,0) : \neg\mathit{Caged}(c)$, $(b,0) : \textit{Bengal-Tiger}(c), (b,0) : \mathit{Distance} < 5m(c), (b,0) : \neg\mathit{Caged}(c)$. At this step, both agent's conflict sets are the same: all formulas match one of the rules, but the highest salience rule is R2, in the case of a matched by $(a,0) : \textit{Bengal-Tiger}(c)$. The other two rule instances in $C_{a,0}$ are $(a,0) : \mathit{Distance} < 5m(c)$ matching R5 and $(a,0) : \neg\mathit{Caged}(c)$ matching R6 (similarly for $C_{b,0}$). So at the next step, $\mathit{\Delta}_1$ contains $(a,1) : \textit{Bengal-Tiger}(c), (a,1) : \mathit{Distance} < 5m(c), (a,1) : \neg\mathit{Caged}(c)$, by the monotonicity rule, and $(a,1) : \mathit{Tiger}(c)$ by R2, and corresponding formulas for b. From step 1, the conflict sets of the two agents diverge: agent a places a new rules instance, $(a,1) : \mathit{Tiger}(c)$ which matches R3, at the top of the conflict set, while agent b favours one of the old rule instances, let's say R5. The new formulas in $\mathit{\Delta}_2$ are $(a,2) : \mathit{Large}(c)$, and $(b,2) : \mathit{Near}(c)$.

At this stage, the top rule instance for a is $(a,2) : \mathit{Tiger}(c)$ matching R4, while the top rule instance for b is $(b,2) : \neg\mathit{Caged}(c)$ matching R6. In addition, both agents have now derived formulas of the kind they communicate to each other; so at the next step, a will discover that c is nearby and b will discover that c is large. The new formulas in $\mathit{\Delta}_3$ are $(a,3) : \mathit{Carnivore}(c)$, $(a,3) : \mathit{Near}(c)$, $(b,3) : \mathit{Free}(c)$, $(b,3) : \mathit{Large}(c)$.

At the next step, both agents will acquire the facts $(a, 4) : Large(c)$, $(a, 4) : Carnivore(c)$, $(a, 3) : Near(c)$, $(a, 4) : Free(c)$, and will match the rule with the top salience, R1, to derive $(a, 5) : Dangerous(c)$ (similarly for b). The reader will easily verify that it would have taken the agents longer to derive $Dangerous(c)$ without communication.

5 Conclusion

In this paper we showed how to model the execution of communicating rule-based agents using Timed Reasoning Logics (TRL). Our framework allow us to model agents at a fine-grained level, so that we can prove, for example, that the agent will use so many computation cycles to arrive at a given conclusion.

In previous work [1], we showed how to model a *single rule at each cycle* strategy similar to that employed by the CLIPS [14] rule-based system architecture, and sketched a logic TRL(CLIPS). In this paper, we prove a general soundness and completeness result for TRL, from which soundness and completeness of TRL(CLIPS) follows. We study TRL(CLIPS) in more detail and give a detailed example involving two communicating agents using CLIPS rule application strategy. The example is quite simple, but it demonstrates that we can compare different agent designs and prove properties of various conflict resolution strategies in the presence of communication between agents.

In the future, we plan to add a more fine-grained analysis of action and communication to the TRL framework. It would also be interesting to investigate more systematically the impact of communication on the time required by agents to reach a given conclusion.

References

1. Alechina, N., Logan, B., Whitsey, M.: A complete and decidable logic for resource-bounded agents. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004), ACM Press (2004)
2. Levesque, H.: A logic of implicit and explicit belief. In: Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI '84). (1984) 198–202
3. Fagin, R., Halpern, J.: Belief, awareness and limited reasoning. In: Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85), Los Angeles, CA (1985) 491–501
4. Konolige, K.: A Deduction Model of Belief. Morgan Kaufman (1986)
5. Lakemeyer, G.: Steps towards a first-order logic of explicit and implict belief. In Halpern, J.Y., ed.: Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference, San Francisco, Morgan Kaufmann (1986) 325–340
6. Parikh, R.: Knowledge and the problem of logical omniscience. In: Methodologies for Intelligent Systems, Proceedings of the Second International Symposium, North-Holland (1987) 432–439
7. Rao, A.S., Georgeff, M.P.: Modeling rational agents within a BDI-architecture. In: Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91). (1991) 473–484

8. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning about Knowledge. MIP Press, Cambridge, Mass. (1995)
9. Moore, R.C.: Logic and Representations. Number 39 in CSLI Lecture Notes. CSLI Publications (1995)
10. van der Hoek, W., van Linder, B., Meyer, J.J.C.: An integrated modal approach to rational agents. In Wooldridge, M., Rao, A., eds.: Foundations of Rational Agency. Kluwer, Dordrecht (1999) 133–168
11. Singh, M.P.: Know-How. In Wooldridge, M., Rao, A., eds.: Foundations of Rational Agency. Kluwer, Dordrecht (1999) 81–104
12. Elgot-Drapkin, J., Miller, M., Perlis, D.: Memory, reason and time: the Step-Logic approach. In Cummins, R., Pollock, J., eds.: Philosophy and AI: Essays at the Interface. MIT Press, Cambridge, Mass. (1991) 79–103
13. Whitsey, M.: Timed reasoning logics: An example. In: Proceedings of the Logic and Communication in Multi-Agent Systems Workshop (LCMAS 2004), Loria (2004)
14. Software Technology Branch, Lyndon B. Johnson Space Center Houston: CLIPS Reference Manual: Version 6.21. (2003)
15. Fagin, R., Halpern, J.: Belief, awareness and limited reasoning. Artificial Intelligence **34** (1988) 39–76
16. Drapkin, J., Perlis, D.: A preliminary excursion into Step-Logics. Proceedings of the SIGART International Symposium on Methodologies for Intelligent Systems (1986) 262–269
17. Elgot-Drapkin, J., Perlis, D.: Reasoning situated in time I: Basic concepts. Journal of Experimental and Theoretical Artificial Intelligence **2** (1990) 75–98
18. Nirkhe, M., Kraus, S., Perlis, D.: 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. Grant, J., Kraus, S., Perlis, D.: A logic for characterizing multiple bounded agents. Autonomous Agents and Multi-Agent Systems **3** (2000) 351–387
20. Gabbay, D.M.: Labeled Deductive Systems: Volume I - Foundations. Oxford University Press (1996)
21. NASA: Proceedings of the Third Conference on CLIPS (CLIPS'94), Lyndon B. Johnson Space Center, NASA (1994)