TIME SATISFACTION IN COORDINATED MULTI-AGENT SYSTEMS
WITH RELATIVE TIME RATES

Said Layadi
MISC Laboratory
Constantine 2 University
25000, Constantine, Algeria
E-mail: layadi@misc-umc.org

Jean-Michel Ilié
Universities of Paris UPMC
and Paris Descartes
4 place Jussieu, 75005, Paris, France
E-mail: jean-michel.ili@upmc.fr

Djamel-Eddine Saidouni
MISC Laboratory
Constantine 2 University
25000, Constantine, Algeria
E-mail: saidouni@misc-umc.org

KEYWORDS
Relative time rates, Region graph, Multi-agent systems, Ambient systems, Timed transport fleets.

ABSTRACT
In this paper, we aim at checking time satisfaction for multi agent systems, the agents of which must coordinate. As in real systems, agents can behave in true concurrency and may have relative time rates. In this context we show that the problem can be reduced to the searching over a specific time region graph.

INTRODUCTION
The coordination problem is a recurrent problem in multi-agent systems. Mainly, this consists in maintaining the synchronization of the agent plans with respect to some objective (Guivarch et al. 2012; Pi et al. 2013).

In this paper, we consider real time application wherein the agent plans refers to timed actions whose durations are known. Agents are assumed to be able to communicate via reliable protocols, in order to achieve some plan called coordinated plan. We aim at providing a technique for the agents to check the satisfaction of temporal properties over such a plan.

As far as time is concerned, the first difficulty to cope with is the heterogeneity of the agent performances. Hybrid Timed automata are often used to represent timed behavioral problems with different speeds, since the represented clocks can be associated with distinguished speeds. In our approach, we propose to model plan in a more attractive way using the standard algebraic language based on LOTOS, seeing plans as the execution of concurrent processes (Chauoche et al. 2014). In fact, LOTOS specifications supplies modular concepts useful here to part some plan over several agents.

Moreover, LOTOS-basic specification are known to be translated in a specific class of timed automata called Duration Actions Timed Automata (DATA for short), such that DATA is demonstrated to be the most compacted way to represent the true concurrency of processes (Belala and Saidouni 2005).

DATA compactly represents the possibly infinite behaviors of the coordinated plans. In this paper, we show how to build a (finite) region graph structure from a DATA specification, in order to check the timing constraints defined over the coordinated plans, taking into account relative time rates that distinguishes the speeds of the agents in coordination.

The schedule of the paper is the following: In Section II, the specification of a coordinated multi-agent system is presented, based on LOTOS concepts, followed by the semantical DATA representation model. In Section III, we show how to extend the operational rules to build a DATA, to take into account both coordination and heterogeneity of the agents. In Section IV, a specific algorithm is defined to build a region graph, while taking the relative time rates into account. The next section is our conclusions, including some of our immediate perspectives.

COORDINATED MAS SPECIFICATION
In this paper, a multi agent system is a tuple $MAS=(Ag, Plan, Act, \tau, \gamma)$ where $Ag$ is set of agents; $Plan$ is a set of agent plans, called coordinated plans since some of them are realized by several agents, $Act$ is the set of actions mentioned in these plans. $\tau=(\tau_o)_{o\in Ag}$ is a rate mapping associating a relative time rate with each agent characterizing the speed of the agent to execute their actions, $\gamma$ is a duration mapping assigning a global duration value to each action of $Act$, evaluated in a number of execution cycles (called time units).

In the following, we describe the specification of coordinated plans as an extension of the language Basic LOTOS, precising which parts of the plans are dedicated to each agent.

The Relative Time Model
In the MAS systems, agents are assumed to have a notion of clocks to achieve their actions. Since agents can have different speeds, we assume that the speeds are relative according to a global time-scale (denoted absolute time), thus the duration performance of some action can be more or less important, depending on the agent considered to execute the action.
In Figure 1, the action \( a \) of global duration 2, is achieved two times faster by the agent \( q \) than by agent \( p \).

![Figure 1: Relative Time over Two Agents](image)

In order to coordinate, each truck is equipped with a software agent able to discuss and synchronize with the other agents in the system. Both agents \( A \) and \( B \) refer to the following coordinated plan:

\[
P ::= E_A \parallel [\text{meet}] E_B \\
E_A ::= \text{move}_A(l_i); \text{meet}; \text{move}_A(l_j); \text{exit} \\
E_B ::= \text{move}_B(l_j); \text{get}_B; \text{move}_B(l_j); \text{meet}; \text{exit}
\]

Moreover, the respective learnt experiences of \( A \) and \( B \) about transporting yields a duration for each of these actions. For the simplicity of the example, all duration of actions are assumed equal to 1 time unit.

**FROM BASIC LOTOS WITH ACTION DURATIONS TO DATA WITH RELATIVE TIME RATES**

The DATA model of a LOTOS specification is a semantics structure designed as a transition system. It is used to express the causality and temporal conditions that must hold throughout the executions of the actions in plans. A DATA with Relative time rates (DATA-R for short), inherits from the DATA structure. The common preliminaries are the followings:

Let \( \mathcal{M} \), ranged over \( x,y,\ldots \), be a set of clocks with non negative real values (in \( \mathbb{R}_{\geq 0} \)) and consider \( H \subset \mathcal{M} \). The set \( \Phi_H \) of temporal constraints \( G \) over \( H \) is defined by the syntax \( G := \text{true} \mid \neg G \mid G \land G \mid x < t \), where \( x \) is a clock in \( H \), \( \leq \in \{=,\leq,\leq_2\} \) and \( t \in \mathbb{R}_{\geq 0} \). Further, \( F_t \) specifies a temporal constraint of the form \( x - t \). A valuation \( \nu \) for \( H \) is a mapping which associates a value in \( \mathbb{R}_{\geq 0} \) with each clock \( x \in H \). The set of all valuations for \( H \) is denoted \( \Xi(H) \). The satisfaction relation \( \models \) for temporal constraints over \( \Xi(H) \) is \( v \models \varphi \iff \nu(x) \models \varphi \) such that \( v \in \Xi(H) \). For \( \xi \in H \), \( [\xi \mapsto 0] \nu \) indicates the valuation for \( H \) which assigns the value \( 0 \) to each \( x \in \xi \), and agrees with the valuation \( \nu \) for the other clocks of \( H \).

**Definition (DATA).** W.r.t. a countable set of clocks \( \mathcal{M} \), a **Durational Action Timed Automaton** \( \mathcal{A} \) is a tuple \((S, I_s, S_0, H, \Sigma)\) where:

- \( S \) is a finite set of states,
- \( H \subset \mathcal{M} \) is a finite set of clocks,
- \( I_s : S \rightarrow 2^\Phi_H \) is a mapping to associate a set of temporal constraints with each state \( s \). Each one expresses by its clock, the fact that some action is potentially in execution in \( s \), under a duration condition.
- \( s_0 \in S \) is the initial state and \( I_s(s_0) = \emptyset \)
- \( T : S \times 2^{\Phi_H} \times \text{Act} \times H \times S \) is the set of transitions. A transition \((s, G, a, x, s')\) represents the change from the state \( s \) to \( s' \), launching the performance of the action \( a \), represented by the clock \( x \) once it is reset and submitted to the temporal constraint \( G \) to be satisfied.

**Definition (DATA with relative time rates).** A **DATA with relative time rates** (DATA-R) over the set of agents \( A \) is a structure \((\mathcal{A}, \pi)\) where \( \mathcal{A} = (S, I_s, S_0, H, \Sigma) \) is a

**Example of Transport**

The frame example of this paper simply concerns two trucks \( A \) and \( B \) such that \( A \) initially placed in the location \( l_1 \) must pick up the load in the location \( l_2 \) and delivers it to the location \( l_4 \). As the load is initially placed in \( l_2, B \) which is initially placed in the location \( l_5 \), proposes to get the load from \( l_3 \) in such a way that \( A \) can meet \( B \) in \( l_2 \) and take the load. The problem for \( A \) and \( B \) is to meet them in minimum time, in case they start at the same time.
Observe that several clocks can be assigned to each agent, said to belong to that agent. The valuations of the clocks of an agent behave synchronously. In a DATA with relative time rates, the clocks that belong to different agents evolve synchronously but under the relative time rates assigned to these agents. Each such rate assigned to the agent $p$ represents the speed of $p$ and depends on some global time reference, from which the local time of $p$ is derived. Formally, a mapping $\tau$ is defined from every global time value $t$ of $\mathbb{R}_{\geq 0}$, assigning the tuple $(\tau_p(i))_{p \in \text{Ag}}$ to $\tau(t)$, s.t. $\tau_p(i) \in \mathbb{R}_{\geq 0}$ is the local time for $p$, taken at time $t$. Initially, $\tau_p(0) = 0$.

The building of a DATA from a LOTOS expression with durations requires to define semantical operational rules, figuring out the derivation of every possible subexpression according to LOTOS operators.

The DATA-R structure has in fact the same structure as a DATA one, representing state by state, the causality and temporal constraints to respect. Prior to formally define the DATA-R rules, we now focus on our frame example to bring out its associate DATA-R structure. The reader can refer e.g. to (Courtiat and Saidouni 1995; Saidouni 1996) for details.

**Building of a DATA-R Structure (Unformal)**

Each state of a DATA refers to a LOTOS (sub-)expression to evaluate, the analysis of which yields the next actions offered to the execute, hence the definition of the next state. Motivated by a dynamic management of clocks the set $\mathcal{M}$ is used as a collection of freed clocks, so that every offered action is assigned to a clock of $\mathcal{M}$ to specify its execution corresponding to the time the execution of the action is surely achieved. Any freed clock can be reused to the execution of another offered action, once being reset to 0.

Figure 2 highlights the DATA(R)-structure of the LOTOS frame example. From the subexpressions $E_A$ and $E_B$, it appears that both actions $move_A(l_2)$ and $move_B(l_3)$ can be concurrently offered. So, each are starting from the initial state is labeled by the offered action, e.g. $move_A(l_2)$, the clock of $\mathcal{M}$ that is allocated to the action. Initially, there is not any action in execution, therefore the set of temporal constraints attached to the initial state is empty, and the guard attached to the transitions are true (hence not represented).

From state to state, several offered actions are launched, considering their clocks are distinct and can behave synchronously. When several actions are offered concurrently from some state, their shuffle is represented in whatever order, by successive transitions and states. This can be seen at the state 2, where the concurrent actions $move_A(l_2)$ and $move_B(l_3)$ are successively launched, in whatever order.

The concurrency of executions is abstractly represented at the state level, since the clocks associated with the launched actions are assumed to progress synchronously, each one submitted to a duration condition corresponding to the duration of the associated action. In the state 2, the set $\{x \geq 1, y \geq 1\}$ expresses that the clocks $x$ and $y$, respectively attached to $move_A(l_2)$ and $move_B(l_3)$, can progress synchronously but none of them can overpass a duration of 1 time unit.

![Figure 2: Example of DATA-R Structure](image)

In the DATA model, the termination of the execution of an action is not explicitly acted in the states, and even can occur several (successive) states after the action is launched. Such a termination is only specified through the causality of actions, when launching a new action. For instance from the state 2, the action $get\_load_B$ is launched, but only under the guard $x \geq 1$, which surely corresponds to the termination of the action $move_B(l_3)$, just previously started by the agent $B$.

As another consequence, a clock mentioned in a guard of a transition becomes freed as the transition is considered, hence can be immedianately reused. That is why the launching of $get\_load_B$ is associated with the clock $y$ although it is previously associated with the previous action $move_B(l_3)$, of the agent $B$.

Lastly, when a parallel composition occurs, like from the state 4 where the action $meet$ can be offered, only one transition is designed even if both agents $A$ and $B$ are concerned.

**Semantical Rules to Build a DATA-R**

The semantical rules producing the DATA-R model are detailed in Table 1, wherein each cell concerns one of
LOTOS operators. As for a DATA model, there may be several subcases. Unlike DATA, the LOTOS expression to consider are labeled symbolically by an agent name, moreover, the function π is used to specify the agent time rate which is used when making the clock progresses.

All the derivation rules are built in the same way. The state to be derived is denoted μ[Ε] and is a pair composed of a (behavior) expression E written in LOTOS, and a set F of duration conditions (F ∈ 2^θ[τ]) which constrains the derivation of E. When performing a derivation, the offered action is generically named a, moreover, the function get defined from M, is used to obtain a free clock for a.

Consider the frame example again to have an insight of the possible derivations, in respect to the DATA-R represented in Figure 2. From the initial state 0, the action move(l₁) in the left expression s.t. E₀ = move(l₁); E₀ is directly offered to execution, since there is no prior duration conditions to satisfy (Φ). A clock x, given by the function get and set to 0, is attached to the action. In the state 1, the left expression E₀ is now constrained by the termination of move(l₂), hence the condition duration (x ≥ 1), so:

\[ \psi(\delta[Ε]) \text{ extracts from the expression } E \text{ the set of clocks that are used in } E. \]
\[ \psi(κ[Ε]) \text{ yields the set } F \text{ of duration conditions. } \]
\[ \psi(κ[F]) \text{ extracts the set of clocks from a set of duration conditions } F. \]
\[ (A[Ε]K) \text{ removes from } F \text{ the set of durations that use a clock of the set } K. \]
\[ (A[Ε])σ \text{ substitutes a subset of clocks of } F \text{ by as many clocks, according to the specification in } σ. \]

**Definition.** The transition relation of the DATA-R → ∈ C × 2^θ[τ] × Act x H x C is defined as the smallest relation satisfying the rules in Table 1.

**REGION AUTOMATON**

In reference to the DATA-R, a reachability graph of (timed) configurations can be computed by attaching to each state of the DATA-R, a valuation of the defined clocks. Inspired from (Alur and Dill, 1994), such configurations can be aggregated according to an equivalence relation preserving the runs. This relation is deduced from an equivalence relation defined on clocks. An equivalence class of clock valuations is named a clock region.

We further assume that the clock constraints of the DATA-R only involve integer constants. As precise in (Alur and Dill 1994), it is not a theoretical limitation but offers easier presentation.

Within some region, the progression of the clock valuations w.r.t. different agents refers to the notion of slopes.

**Equivalence Classes of Clock Valuations**

Let \( A = ((S, L, S_0, H, T), π) \) be a DATA-R over a set Ag.

**Definition (slope) .** Let x (resp. y) be a clock that belongs to the agent p (resp. q) and evolves according to the rate function \( τ_x \) (resp. \( τ_y \)). We define slope_{xy} as the ratio of local-time rate functions \( τ_p \) and \( τ_q \), denoted \( slope_{xy} = τ_p/τ_q \) (see Figure 3).

As the number of clock constraints used in the DATA-R is necessarily limited, we can determine for each used clock x, an integer value \( c_x ∈ N \) which corresponds to the greatest value x is compared in some clock constraint (guard) in the DATA-R. Moreover, whatever the pair of clocks x and y, the parameter slope_{xy} is assumed an integer constant.

We now define the equivalence relation that is used to build the regions. The notation \( \lfloor v(x) \rfloor \) represents the integer value of the valuation v(x), and fract its decimal part.
### Table 1: Derivation Rules Producing the DATA-R Model

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{exit} \xrightarrow{F, a_x} \text{stop}$</td>
<td>$x = \text{get}(\mathcal{M})$ and $\pi(x) = p$</td>
</tr>
<tr>
<td>$E_p \xrightarrow{F, a_x} E_p'$</td>
<td>$\alpha \neq \delta$</td>
</tr>
<tr>
<td>$\mathcal{F}_{q} \xrightarrow{F, a_x} E_p'$</td>
<td>$E_p \Rightarrow \mathcal{F}_{q}$</td>
</tr>
<tr>
<td>$E_p \xrightarrow{F, a_x} E_p'$</td>
<td>$\mathcal{F}_{q} \Rightarrow E_p'$</td>
</tr>
<tr>
<td>$\mathcal{F}_{q} \xrightarrow{F, a_x} E_p'$</td>
<td>$E_p \Rightarrow \mathcal{F}_{q}$</td>
</tr>
</tbody>
</table>

(a) $a \in L \cup \{\delta\}$

(b) $y = \text{get} \left( \mathcal{M} - \left( \psi(E_p) \cup \psi(F_q) \right) \right)$ and $\pi(y) = p$

$z = \text{get} \left( \mathcal{M} - \left( \psi(E_p) \cup \psi(F_q) \right) \right)$ and $\pi(z) = \{p \text{ if } T_p \leq T_q, \text{ otherwise}\}$

#### Figure 3: Two Clocks Evolution with Different Speeds

**Definition (equivalence relation ~).** We define the equivalence relation ~ over the set of all clock valuations, $v$~$v$' iff all the following conditions hold:

1. For all $x \in H$, either $[v(x)]$ and $[v'(x)]$ are the same, or both $v(x)$ and $v'(x)$ are greater than $c_x$.
2. For all $x, y \in H$ with $v(y) \leq c_x$, $v(y') \leq c_y$ and $x$ (resp. $y$) evolves according to $\tau_p$ (resp. $\tau_q$):

   - $c \cdot \frac{1}{\text{slope}_{xy}} \leq v'(x) \leq (c + 1) \cdot \frac{1}{\text{slope}_{xy}}$ if $c \leq \frac{1}{\text{slope}_{xy}}$
   - $\text{slope}_{xy} \cdot \text{fract}(v(x)) \leq \text{slope}_{xy} \cdot \text{fract}(v(y))$ if $\text{fract}(v(x)) \leq \text{fract}(v(y))$

An equivalence class of clock valuations induced by ~ is a clock region of $A$.

#### The Representation of Clock Regions

Each region can be specified by the finite set of clock constraints it satisfies. The notation $[\psi]$ represents the clock region to which $v$ belongs. In Figure 4, the region $[0 \leq x < 2]$ contains all the clock valuations that satisfy $0 \leq y < 2$.

**Example 1.** Consider two clocks $x$ and $y$ which evolve at different rates such that $\text{slope}_{xy} = 2$, $c_x = 2$ and $c_y = 2$. The clock regions obtained by the relation ~ are depicted in Figure 4, containing 15 corner points, 38 open line segments and 23 open regions.

**Definition (slope$_{max}$).** For each clock $x \in H$, we define $\text{slope}_{max}(x)$ as the largest value of $\text{slope}_{xy}$ for all $y \in H$.

In the example 1:

- $\text{slope}_{x(y)} = \max(\text{slope}_{x(y)}, \text{slope}_{x(y)}) = \max(2, 1/2) = 2$.
- $\text{slope}_{x(y)} = \max(\text{slope}_{x(y)}, \text{slope}_{x(y)}) = \max(1/2, 1) = 1$.

Note that if $x$ is the fastest clock in $H$ then $\text{slope}_{x(y)} = \text{slope}_{x(y)} = 1$.

The smallest amount of time in which $x$ cannot stay in the same region is $1/\text{slope}_{x(y)}$. In the example 1, the clock $x$ changes the region every half unit of time corresponding to $1/\text{slope}_{x(y)} = 1/2$, whereas $y$ does this change every one unit of time (except for regions represented by points).
The representation of a clock region accords with the two following points:

1. For each clock \( x \) which evolves according to \( \tau_p \), there is one clock constraint taken from the set:
\[
\{ x = c | c = 0, \frac{1}{\text{slope}_{xy}}, \frac{2}{\text{slope}_{xy}}, ..., 1 + \frac{1}{\text{slope}_{xy}}, 1 + \frac{1}{\text{slope}_{xy}}, ..., c_y \ \text{for all} \ y \in H \}
\]
\[
\bigcup \left\{ \begin{array}{l}
\left[ \frac{c - \frac{1}{\text{slope}_{xy}}}{\text{slope}_{xy}}, \frac{1}{\text{slope}_{xy}}, \frac{2}{\text{slope}_{xy}}, ..., 1 + \frac{1}{\text{slope}_{xy}}, 1 + \frac{1}{\text{slope}_{xy}}, ..., c_x \right] \\
\end{array} \right. 
\end{array}
\]
\( \bigcup \{ x > c_x \} \).

2. For each pair of clocks \( x \) and \( y \) which evolve respectively according to \( \tau_p \) and \( \tau_q \) such that \( c < c < c + (1/\text{slope}_{\max}(y)) \) and \( d < y < d + (1/\text{slope}_{\max}(y)) \) appear in (1) for some \( c \) and \( d \), whether \( \text{slope}_{xy}(x-c) \) is less than, equal to or greater than \( (y-d) \).

The time-successors of clock regions

In the following, we introduce the time-successor relation over clock regions. When time advances from any clock valuation \( v \) in the region \( \alpha \), we will reach all its time-successors \( \alpha' \). Formally, we say that \( \alpha' \) is a time-successor of the region \( \alpha \) if there are \( v \) in \( \alpha \), \( v' \) in \( \alpha' \), \( t \in \mathbb{R}_{0} \) such that \( v' = v \oplus \tau(t) \), with \( v \oplus \tau(t) = (v(x) + \tau_p(t))_{x}^{-1}(v(y)) \).

For example, in Figure 4, the five time-successors of the region \( \alpha' = [(x=2), (y=2), (x=2), (y=2), (x=2), (y=2)] \) are itself, \( [(x=2), (y=2), (x=2), (y=2)] \), \( [(x=2), (y=2)] \) and \( [(x=2), (y=2)] \). These regions are those covered by a line drawn from any point in \( \alpha \) parallel to the line \( y = \text{slope}_{xy} \cdot 2x \) (in the upwards direction).

To compute each time-successor of a region \( \alpha \), we must give (i) for every clock \( x \), a constraint of the form \( x = c \) or \( c < c < c + (1/\text{slope}_{\max}(y)) \) or \( c < c_y \) and (ii) for every pair \( x \) and \( y \) such that \( c < c < c + (1/\text{slope}_{\max}(y)) \) and \( d < y < d + (1/\text{slope}_{\max}(y)) \) appear in (i), the ordering relationship between \( \text{slope}_{xy}(x-c) \) and \( y-d \).

The possible time-successors are now presented according to 3 cases:

**First**, each clock \( x \) in the region \( \alpha \) satisfies the constraint \( (x > c_x) \); in this case there is only one time-successor to \( \alpha \) which is \( \alpha' \) itself. This is the case of region \( [(x=2), (y=2)] \) in Figure 4.

**Second**, this case is considered when there is at least, in the region \( \alpha \), one clock \( x \) which satisfies the constraint \( x = c \) for some \( c \leq c_x \). The set \( H_0 \) contains all clocks appearing in similar constraint form as \( x \). The clock region \( \alpha \) will be changed immediately when the time progresses, because the fractional part of each clock in \( H_0 \) becomes different from zero. The clock regions \( \alpha \) and \( \beta \) have the same time-successors, s.t. \( \beta \) is specified by:
1. A set of clock constraints specified as follows:
   a. For each clock \( x \in H_0 \):
      i. If \( \alpha \) satisfies \( (x = c) \) then \( \beta \) satisfies \( (x = c) \);
      ii. If \( \alpha \) satisfies \( (x = c+y) \) then \( \beta \) satisfies \( (x = c+y) \).
   b. For each clock \( x \in H_0 \), the clock constraint in \( \alpha \) remains the same in \( \beta \).
2. The ordering relation between \( \text{slope}_{xy}(x-c) \) and \( (y-d) \) of each pair of clocks \( x, y \) in \( \alpha \) is the same as that in \( \beta \), such that \( x < c_x \) and \( y < c_y \) hold in the region \( \alpha \).

For example, in Figure 4, the time-successors of the region \( [(x=0), (0 < y < 1)] \) are the same as the time-successors of the region \( [0 < x < 1] \).

**Finally**, if the first and the second case do not apply, then let \( H_0 \) be the set of clocks \( x \) for which the region \( \alpha \) satisfies two constraints \( c < c < c + (1/\text{slope}_{\max}(y)) \) and \( c < c < c + (1/\text{slope}_{\max}(y)) \) for all clocks \( y \) for which the region \( \alpha \) satisfies \( d < y < d + (1/\text{slope}_{\max}(y)) \). Thus, when the time progresses, the clocks in \( H_0 \) take the values \( c + (1/\text{slope}_{\max}(y)) \). Therefore, the time-successors of the region \( \alpha \) are the 2 regions \( \alpha \) and \( \beta \), together with all the time-successors of \( \beta \), such that \( \beta \) is specified by:
1. A set of clock constraints, described as follows:
   a. For each clock \( x \in H_0 \), if \( \alpha \) satisfies \( c < c < c + (1/\text{slope}_{\max}(y)) \) then \( \beta \) satisfies \( (x = c + (1/\text{slope}_{\max}(y))) \);
   b. For each clock \( x \in H_0 \), the clock constraint in \( \alpha \) remains the same in \( \beta \).
2. For each pair of clocks \( x \) and \( y \) such that both intervals \( (c < c < c + (1/\text{slope}_{\max}(y))) \) and \( (d < y < d + (1/\text{slope}_{\max}(y))) \) appear in (1.b), the ordering relationship between \( \text{slope}_{xy}(x-c) \) and \( (y-d) \) in \( \alpha \) remains the same in \( \beta \).

For example, the time-successors of the region \( [0 < x < 1, (0 < y < 1)] \) in Figure 4, include the region itself, the region \( [0 < x < 0.5, (y = 1)] \) and all the time-successors of \( [0 < x < 0.5, (y = 1)] \).
**Algorithm (region automaton).**

Let $A = (S, I, \alpha, s_0, H, T, \pi)$ be a DATA-R over the set of agents $\mathcal{A}$. The region automaton $R(A)$ is an automaton over the alphabet $Act$ such that

- The configurations of $R(A)$ are of the form $<s, \alpha>$ where $s$ is a state of $A$ and $\alpha$ is a clock region. The initial configuration of $R(A)$ is $<s_0, \emptyset>$.
- A transition of $R(A)$, from the configuration $<s, \alpha>$ to $<s', \alpha'>$ is labeled by $a \in Act$ if there is a transition $(s, G, a, x, s')$ in $T$ and a clock region $\alpha''$ which satisfies:
  - $\alpha'' = \alpha$, a time-successor of the region $\alpha$.
  - $\alpha'' = G$.
  - $\alpha'' = \{ x \mid x \rightarrow 0 \}$ or $\alpha''$.

Figure 5 highlights a part of the region graph of the frame example.

![Region Graph](image)

**CONCLUSIONS AND PERSPECTIVES**

The proposed algebraic language extends the standard LOTOS specification, with a capacity of describing timed plans with action durations, that can be shared in between some coordinated agents.

Although agents can be heterogeneous, they can reasonably be (re)synchronized to start the execution of any coordinated plan and that they can behave under relative time rates.

Taking benefit from the semantics of such a specification, we demonstrated how to build a finite (time) region graph, which is bisimilar in respect to the infinite set of behaviors implied by the specification. Compared to existing approaches, our technique benefits from true concurrency properties, without increasing the size complexity of the region graph.

To exemplify our technique, we applied it to a simple but realistic ambient use case, wherein the coordination of trucks is required. Agents are currently reduced to having one clock, however, the extension to several is immediate.

Nevertheless, interesting concepts have been already developed through other models, that can be compatible with the computation of regions, e.g. [hybrid automata][priced automata]. This motivates us to investigate the possibility of integrating similar concepts within our approach.

**REFERENCES**


