From Causality Semantics to Duration Timed Models

Walid Belkhir
Laboratoire d’Informatique Fondamentale de Marseille
Université de Provence, Marseille, France.
belkhir@cmi.univ-mrs.fr

ABSTRACT
The interleaving semantics is not compatible with both action refinement and durational actions. Since many true concurrency semantics are congruent w.r.t. action refinement, notably the causality and the maximality ones [Cos93, Gla90], this has challenged us to study the dense time behavior - where the actions are of arbitrary fixed duration - within the causality semantics of Da Costa [Cos93].

We extend the causal transition systems with the clocks and the timed constraints, and thus we obtain an over class of timed automata where the actions need not to be atomic. We define a real time extension of the formal description technique CSP, called duration-CSP, by attributing the duration to actions. We give the operational timed causal semantics of duration-CSP as well as its denotational semantics over the class of timed causal transition systems. Afterwards, we prove that the two semantics are equivalent. Finally we extend the duration-CSP language with a refinement operator $\rho$ - that allows to replace an action with a process - and prove that it preserves the timed causal bisimulation.

Keywords
Causality semantics, true concurrency, durational process algebra, timed automata, action refinement.

1. INTRODUCTION
Many complex systems such as communication protocols, networks and embedded systems require a top down design where processes are modeled at different levels of abstraction. To carry on, at every level of abstraction, each action might be replaced by a more complicated process. This is known as the concept of action refinement [CS93, SpC94, FMCW02, KK09]. It turns out that the actions are no longer atomic: they are divisible into small parts. On the other hand, many industrial systems exhibit quantitative behavior, including timing and minimal performance. As a consequence, many real time extensions have been suggested for process algebra [MT90, LL97, BCAM00, Yi91]. However, the common point of all these extensions is that they are based on the action atomicity hypothesis. It was pointed out [Gla90, Cos93, Sai96] that the non atomicity of actions as well as the action refinement require a truth concurrency semantics instead of the interleaving semantics.

In this paper we suggest an approach that integrates both the timed constraints and durational actions without replacing the action with the two atoms events: its starting and finishing ones, which leads to a huge combinatorial explosion. Our approach consists in using a truth concurrency semantics called the timed causal semantics which extends the causality semantics of [Cos93, Gla90]. We extend the formal description technique CSP with both durational actions and timed constraints. Afterwards we describe its semantics by means of the timed causal semantics. To convince the reader that the interleaving semantics can not be used to deal with the durational actions, let us consider the two processes $P = a; b; \text{stop} + b; a; \text{stop}$ and $Q = a; \text{stop} || b; \text{stop}$. The process $P$ expresses a choice between $a$ followed by $b$ and $b$ followed by $a$. The process $Q$ expresses a parallel execution of $a$ and $b$. Note that, if we consider that $duration(a) = 0$ and $duration(b) = 0$, then the two processes describe, in some sense, the same behavior. However, if we consider that $duration(a) > 0$ and $duration(b) > 0$ then the execution of $P$ requires at least an amount of times equals to $duration(a) + duration(b)$, and the execution of $Q$ may be done in $max\{duration(a), duration(b)\}$.

In a next step we extend the causal transition systems of [Cos93] with clocks and timed constraints in the same spirit of the timed automata [AD94]. We shall call this model the timed causal transition system. We recall that the causal transition system formalism enriches the usual transition system one with the notion of causality. As a consequence the timed causal transition system formalism allows to express the timed constraints over the actions of arbitrary duration without the need of replacing each action by its starting and finishing event. As an application we show how to generate a timed causal transition system out of a duration-CSP process, and prove the correctness of this generation.

The paper is organized as follows. Section 2 recalls the rudiments of the causality semantics as given in [Cos93]. In section 3 the definition of the causal transition system formalism and its timed extension are given. In section 4 we extend...
the kernel of CSP with action duration and timed constraints and
we give its timed causal operational semantics. In section 3 we give
the denotational semantics of duration-CSP in terms of the
timed causal transition system model. This section is concluded by
a proof that the two semantics are equivalent, Theorem 1. In section 4
we enrich the language duration-CSP with the refinement operator $\rho$
that allows to replace an action with a more complicated process. The
new language is called duration – CSP$_\rho$, afterwards, we give the
timed causal semantics of this language, notably, the semantics
of the refinement operator. Finally we prove that the refinement operator
preserves the timed causal bisimulation, Theorem 2.

In section 5 some current and future works are given. The
proofs are given in the Appendix.

2. CAUSALITY SEMANTICS

In this section we recall, through simple examples, the principles
of the causality semantics as defined in Cos93. The aim of the cause-
ality semantics is to distinguish between the sequential and the parallel
execution. To be more precise, a parallel execution of two actions can
not be substituted by their interleaved execution. To this goal, a transi-
tion from state $s_1$ to $s_2$ has the form $s_1 \xrightarrow{a,b} s_2$; it is equipped
with an extra data: (i) the event $x$ which identifies the beginning of
the execution of the action $a$, and (ii) the (finite) set $E$ of events
which corresponds to the set of causes of the action $a$, i.e. the action
$a$ is possible if all the causes belonging to $E$ terminate. For example
let us consider the two processes $P$ and $Q$ defined by: $P = a; b; stop + b; a; stop$ and
$Q = a; stop || b; stop$. We recall that "$;" is the prefixing operator, "$||" is the parallel composition, and "$ + " is the
choice operator. At the beginning, the execution of both $P$ and $Q$
does not depend on any event, therefore the initial configuration associated
with $P$ (resp. $Q$) is of the form $\emptyset$ (resp. $\emptyset$[$Q$]). By applying the causality semantics to the
configuration $\emptyset$[$P$] the following derivations are possible:

$$\emptyset[P] \xrightarrow{a} \{x\}[b; stop] \xrightarrow{b}[y][stop]$$

The event $x$ (resp. $y$) corresponds to the beginning of the execution
of the action $a$ (resp. $b$). According to the semantics of the prefix
operator "$;" and the execution of the action $b$ depends on the
termination of the execution of the action $a$. Again, by applying
the causality semantics to the configuration $\emptyset$[$Q$], the
following derivations are possible:

$$\emptyset[Q] \xrightarrow{a} \{x\}[stop] \xrightarrow{b}[y][stop]$$

As before, the event $x$ (resp. $y$) corresponds to the beginning
of the execution of the action $a$ (resp. $b$). The main
difference is that both the actions $a$ and $b$ does not depend
on each other.

The Figure 2.1 shows all the possible derivations which can be
obtained by applying the causality semantics to $P$ and $Q$. This gives rise to the notion of causal transition systems
which will be formalized in the next section.

3. TIMED CAUSAL TRANSITION SYSTEMS

In this section we formalize the notion of causal transition
systems. Afterwards, we enrich them with clocks and
timed constraints in order to specify the timed behaviour.

Throughout this paper we let $\mathcal{E}$ be a countable set of events,
 ranged by $x,y,z\ldots$. Let $\mathcal{L}$ be a countable set of actions,
ranged by $a,b,c\ldots$. If $a \in \mathcal{L}$ then we denote by $d(a)$ the
duration of the action $a$, where $d(a) \in \mathbb{R}^+$. 

Definition 1. A causal transition system, or a CTS
for short, over $\mathcal{E}$ is a tuple $(S,s_0,T,l,\psi,\zeta,\eta)$ where:

- $(S,s_0,T,l)$ is a labeled transition system over $\mathcal{L}$, that
  is, $S$ is a finite set of states, $s_0 \in S$ is the initial state,
  $T \subseteq S \times S$ is the set of transitions, and $l : T \rightarrow \mathcal{L}$ is
  the labeling function of transitions,
- $\psi : S \rightarrow 2^\mathcal{E}$ is the function that associates to each
  state a finite set of events, the latter being potentially
  in progress at this state,
- $\zeta : T \rightarrow 2^\mathcal{E}$ is the function that associates to each
  transition $t \in T$ a finite set of events, these events
denote the direct causes of $t$,
- $\eta : T \rightarrow \mathcal{E}$ is the function that associates to each
  transition $t \in T$ the event attached to the occurrence
  of the action $l(t)$,

such that the following conditions hold: for each transition
$(s,s') \in T$ we have that

i. $\eta(s,s') \in \psi(s'),$
ii. $\zeta(s,s') \cap (\psi(s') - \eta(s,s')) = \emptyset,$
iii. $\zeta(s,s') \subseteq \psi(s)$ and $\psi(s') - \zeta(s,s') \subseteq \psi(s).$

In the next a transition $t$ will be denoted by $s_1 \xrightarrow{E} s_2$, i.e.
$l(t) = a$, $\zeta(t) = E$, and $\eta(t) = x$. 

Figure 2.1: Causal transition systems of the processes $P$ and $Q$. 

$P = a; b; stop + b; a; stop$ $Q = a; stop || b; stop$
The key idea. Now, we add to the CTS the notions of clocks and timed constraints in order to be able to specify the quantitative behaviour over durational actions. The key idea consists in considering the events themselves as a sort of local clocks. As a consequence, the values of the clocks give sufficient information about the progress of the actions, notably about their termination. For instance consider the timed process $R$ defined by $R = a\{4\} \cdot 0^{100}b; \text{stop}$.

Fig 3.1: The timed-CTS of $R$

We use functions called clock assignments, a mapping from $\text{Clk}$ to $\mathbb{R}^+$. Let $\nu$ denote such function, and $\Omega$ denote the clock assignment that maps all $c_e \in \text{Clk}$ to 0. For $d \in \mathbb{R}^+$, let $\nu + d$ denote the clock assignment that maps all $c_e \in \text{Clk}$ to $v(c_e) + d$. For $\lambda \subseteq \text{Clk}$, let $[\lambda \mapsto 0] \nu$ denote the clock assignment that maps all clocks in $\lambda$ to 0 and coincide with $\nu$ for the clocks in $\text{Clk} \setminus \lambda$.

The semantics of a timed-CTS is a transition system whose configurations are pairs $(s, \nu)$, the starting configuration is $(s_0, \Omega)$, and the transitions are given by the rules:

- $(s, \nu) \xrightarrow{a} (s', \nu + d)$, for $d \in \mathbb{R}^+$;
- $(s, \nu) \xrightarrow{E} (s', \nu')$ if $s \xrightarrow{E} s'$ and moreover: (i) $\nu$ satisfies the constraint $\varphi$, (ii) $\nu' = [\lambda \mapsto 0] \nu$, and (iii) all the actions related to the events $E$ have terminated.

4. DURATION-CSP AND ITS OPERATIONAL TIMED CAUSAL SEMANTICS

Now we introduce the action duration to the formal description technique CSP [Hoa85]. Due to the lack of space the prefixing operator " $\triangleright$ " is denoted by " $;$ " . Moreover, we do not distinguish between the internal and the external choice. The syntax of duration-CSP is given by the following grammar:

$$ P ::= \text{stop} \mid \text{skip} \{d\} \mid \Theta^d P \mid a\{d\}; P \mid P + Q \mid P[|L|]Q \mid P \setminus L \mid P \bigtriangleup Q $$

where $d \in \mathbb{R}^+$ and $L \subseteq \mathcal{L}$. The primitive process $\text{stop}$ represents the process that communicates nothing, and $\text{skip}$ represents successful termination i.e. the process $\text{skip}\{d\}$ performs the successful termination action $\delta$ in the time interval $[0,d]$ and transforms into $\text{stop}$. Let $a \in \mathcal{L}$ be an action and $d \in \mathbb{R}^+$. The process $a\{d\}; P$ expresses that the execution of $a$ must be in the time interval $[0,d]$, and after the termination of $a$ this process behaves like $P$. The process $\Theta^d P$ means that the starting of $P$ is possible only after a passage of $d$ units of time. "$\bigtriangleup"$ is the choice operator. The parallel composition $P[|L|]Q$ allows computation in $P$ and $Q$ to proceed simultaneously and independently apart on the actions in $L$ on which both processes must be synchronized. We shall write $||$ for $[|\emptyset|]$. The hiding operator $P \setminus L$ makes the actions in $L$ unobservable. The interruption operator $P \bigtriangleup Q$ allows the computation to begin in $P$ and to be interrupted by $Q$.

Operational semantics of Duration-CSP.

Now we describe the behaviour of duration-CSP processes step by step by means of the operational semantics over the timed causal configurations. Before this, we first define the timed causal configurations and introduce some standard operations on them. The untimed configurations and the related operations have been defined in [Cos93].

Definition 3. The set $\mathcal{C}_s$ of timed causal configurations is defined as follows:
For instance, the configuration \( \{ x=a:t_x \} \) means that the execution of the process \( P \) depends on the termination of the action \( a \) which is identified by the event \( x \), moreover, \( t_x \) counts the time elapsed from the beginning of \( a \). We say that a timed causal configuration is in the canonical form if it can not be simplified by distributing the set of events over the algebraic operators. For instance, the configuration \( E_r[a; \text{stop} + b; \text{stop}] \) is not in the canonical form because it can be reduced to the configuration \( E_r[a; \text{stop}] + E_r[b; \text{stop}] \), the latter being in the canonical form.

**Lemma 1.** Every canonical timed causal configuration in \( C_r \) has one of the following forms:

\[
E_r[\text{stop}] \quad E_r[\text{skip}(d)] \quad \Theta^d P_r \quad E_r[a\{d\}; P]
\]

\[
P_r + Q_r \quad P_r \Delta Q_r \quad \forall \delta \in E \times L \times \mathcal{E} \quad \forall \delta \in E \times L \times \mathcal{E}
\]

where \( P_r \) and \( Q_r \) are in the canonical form.

Next we assume that all the configurations are in the canonical form.

**Definition 4.** The function \( \psi : C_r \rightarrow 2^E \times \mathcal{L} \times \mathbb{R}^+ \), that determines the events of a given configuration is defined by:

\[
\psi(E_r[\text{stop}]) = \psi(E_r[\text{skip}(d)]) = \psi(E_r[a\{d\}; P]) = E_r
\]

\[
\psi(\Theta^d P_r) = \psi(P_r \backslash L) = \psi(P_r)
\]

\[
\psi(P_r + Q_r) = \psi(P_r \mid |L|) \psi(Q_r) = \psi(P_r \Delta Q_r) = \psi(P_r) \cup \psi(Q_r)
\]

**Definition 5.** Let \( \mathcal{R}_r \in C_r \) and \( x, y \in \mathcal{E} \), the substitution of \( x \) by \( y \) in \( \mathcal{R}_r \), denoted by \( \mathcal{R}_r[y/x] \), is defined by induction on \( \mathcal{R}_r \) as follows:

\[
(E_r[\text{stop}])[y/x] = E_r[y/x][\text{stop}]
\]

\[
(E_r[\text{skip}(d)])[y/x] = E_r[y/x][\text{skip}(d)]
\]

\[
(\Theta^d P_r)[y/x] = \Theta^d P_r[y/x]
\]

\[
(E_r[a\{d\}; P])[y/x] = E_r[y/x][a\{d\}; P]
\]

\[
(P_r + Q_r)[y/x] = P_r[y/x] + Q_r[y/x]
\]

\[
(P_r \mid |L|) \backslash L = P_r[y/x] \backslash L
\]

\[
(P_r \Delta Q_r)[y/x] = P_r[y/x] \Delta Q_r[y/x]
\]

where \( E_r[y/x] \) is again the obvious substitution over the set of events.

Let \( E_r \in 2^E \times \mathcal{L} \times \mathbb{R}^+ \), we say that all the actions in \( E_r \) have finished and write \( \text{Finish}(E_r) \), if for all \( x : a : t_x \in E_r \) we have that \( t_x > d(a) \). Let \( \text{get} : 2^E \rightarrow \mathcal{E} \) be a function satisfying \( \text{get}(E) \in E, \forall E \in 2^E \setminus \{ \emptyset \} \).

The timed transition over the timed causal configurations, denoted by \( \sim \subseteq C_r \times Act_r \times C_r \) where \( Act_r = (2^E \times \mathcal{L} \times \mathbb{R}^+) \cup \mathbb{R}^+ \), is defined as follows:

\[ \text{0. Stop process:} \]

\[
\neg \text{Finish}(E_r) \quad E_r[\text{stop}] \quad \sim \quad E_r[d; \text{stop}]
\]

\[ \text{I. Skip process:} \]

\[
(\text{I.a}) \quad \text{Finish}(E_r) \quad E_r[\text{skip}(d)] \quad \sim \quad E_r[\text{skip}(d)]
\]

(\text{I.b}) \quad \text{Finish}(E_r) \quad E_r[\text{skip}(d)] \quad \sim \quad E_r[\text{skip}(d)]

\[ \text{II. Prefix operator:} \]

\[
(\text{II.a}) \quad \text{Finish}(E_r) \quad E_r[a\{d\}; P] \quad \sim \quad E_r[a\{d\}; P]
\]

(\text{II.b}) \quad \text{Finish}(E_r) \quad E_r[a\{d\}; P] \quad \sim \quad E_r[a\{d\}; P]

\[ \text{III. Choice operator:} \]

\[
(\text{III.a}) \quad P_r \quad \sim \quad P_r' \quad Q_r \quad \sim \quad Q_r'
\]

\[
P_r + P_r' \quad \sim \quad P_r + P_r'
\]

(\text{III.b}) \quad P_r + P_r' \quad \sim \quad P_r + P_r'

\[ \text{IV. Parallel composition operator:} \]

\[
(\text{IV.a}) \quad P_r \quad \sim \quad P_r' \quad Q_r \quad \sim \quad Q_r'
\]

\[
P_r \mid |L| \quad \sim \quad P_r' \mid |L| \quad \sim \quad P_r' \mid |L| \quad Q_r
\]

\[
(\text{IV.b}) \quad P_r \quad \sim \quad P_r' \quad a \notin L \cup \{ \delta \}
\]

\[
Q_r \mid |L| \quad \sim \quad Q_r' \mid |L| \quad \sim \quad Q_r' \mid |L| \quad Q_r
\]

\[
(\text{IV.c}) \quad P_r \quad \sim \quad P_r' \quad Q_r \quad \sim \quad Q_r'
\]

\[
E_r[z/x] \quad \sim \quad E_r[z/x]
\]

\[
\text{where in the last two rules we have}\ y = \text{get}(\mathcal{E} - \{(\text{ψ}(P_r') - \{x\}) \cup \text{ψ}(P_r)\}). \text{ To avoid any confusion with the definition of \text{ψ} given in Definition 4 here we consider that \text{ψ} : C_r \rightarrow 2^E \text{ but we still use the same symbol, the type of \text{ψ} is clarified by the context.} \]
V. Hide operator:

\[(V.a) \quad \begin{array}{l}
\frac{E \in \Delta \tau \quad \rho' \cap E \neq \emptyset}{\rho \setminus \rho'} \\
\frac{E \notin \Delta \tau \quad \rho' \cap E = \emptyset}{\rho \setminus \rho'}
\end{array} \quad (V.b) \quad \begin{array}{l}
\frac{E \in \Delta \tau \quad \rho' \cap E \neq \emptyset}{\rho \setminus \rho'} \\
\frac{E \notin \Delta \tau \quad \rho' \cap E = \emptyset}{\rho \setminus \rho'}
\end{array} \]

(V.r) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\rho \setminus \rho'}
\end{array}

VI. Interruption operator:

\[(VI.a) \quad \begin{array}{l}
\frac{E \in \Delta \tau \quad \rho' \cap E \neq \emptyset}{\rho \setminus \rho'} \\
\frac{E \notin \Delta \tau \quad \rho' \cap E = \emptyset}{\rho \setminus \rho'}
\end{array} \quad (VI.b) \quad \begin{array}{l}
\frac{E \in \Delta \tau \quad \rho' \cap E \neq \emptyset}{\rho \setminus \rho'} \\
\frac{E \notin \Delta \tau \quad \rho' \cap E = \emptyset}{\rho \setminus \rho'}
\end{array} \quad (VI.c) \quad \begin{array}{l}
\frac{E \in \Delta \tau \quad \rho' \cap E \neq \emptyset}{\rho \setminus \rho'} \\
\frac{E \notin \Delta \tau \quad \rho' \cap E = \emptyset}{\rho \setminus \rho'}
\end{array} \]

(VI.r) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\rho \setminus \rho'}
\end{array}

VII. Delay operator:

\[(VII.a) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\Theta E \rho \rightharpoonup \Theta E \rho'}
\end{array} \quad \begin{array}{l}
(VII.r') \quad \frac{\rho \rightharpoonup \rho'}{\Theta^d E \rho \rightharpoonup \Theta^d E \rho'}
\end{array} \quad (VII.r) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\Theta E \rho \rightharpoonup \Theta E \rho'}
\end{array}

\[(VII.a) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\Theta E \rho \rightharpoonup \Theta E \rho'}
\end{array} \quad \begin{array}{l}
(VII.r') \quad \frac{\rho \rightharpoonup \rho'}{\Theta^d E \rho \rightharpoonup \Theta^d E \rho'}
\end{array} \quad (VII.r) \quad \begin{array}{l}
\frac{\rho \rightharpoonup \rho'}{\Theta E \rho \rightharpoonup \Theta E \rho'}
\end{array}
\]

VIII. Passage of time:

\[\neg \text{Finish}(E_x) \quad \forall \varepsilon \geq 0 \exists \epsilon \leq d \quad \neg \text{Finish}(E_x + \epsilon)
\]

\[E_x \[P\] \rightharpoonup E_{x+d}[P]
\]

Definition 6. Let \(R_x \subseteq C_x\), the passage of \(d\) units of time over \(R_x\), denoted by \(R_x + d\), is defined by induction on \(R_x\) as follows:

\[E_x \[P\] + d = E_{x+d}[P]
\]

\[(P \setminus Q_x) + d = (P + d) \cup (Q_x + d)
\]

\[(P \setminus L) + d = (P + d) \setminus L
\]

\[(P \setminus L \cup Q_x) + d = (P + d) \setminus (Q_x + d)
\]

\[(P \setminus \Delta Q_x) + d = (P + d) \setminus (Q_x + d)
\]

where

\[\begin{array}{l}
\emptyset + d = \emptyset \\
(\{x : a : t_x\}) + d = \{x : a : t_x + d\}
\end{array}
\]

\[\{E_x \cup \{x : a : t_x\}\} + d = (E_x + d) \cup \{x : a : t_x + d\}
\]

Definition 7. Given a duration-CSP process \(P\), the operational semantics of \(P\) over the class of the timed causal configurations \(C_x\), denoted by \(P_{op}\), consists in associating to \(P\) the set of timed causal configurations generated by the relation \(\rightharpoonup \subseteq C_x \times \text{Act}_x \times C_x\), starting from the configuration \(\emptyset[P]\).

5. A DENOTATIONAL SEMANTICS

In this section we describe how to generate a timed-CTS (see Definition 2) from a duration-CSP specification. To this goal, we define the timed causal transition relation \(\rightharpoonup \subseteq C \times \text{tr}s \times C\), where \(C\) is defined exactly as the set of the timed configurations \(C_x\) given in Definition 3, apart that \(E_x \in 2^{E \times \mathbb{L}}\) instead of \(E_x \in 2^{E \times \mathbb{L} \times \mathbb{R}^+}\) and hence \(E_x\) will be denoted by \(E\); and the timed transition \(\text{tr}s \in (2^{E \times \mathbb{L} \times \mathbb{L} \times \mathbb{E}) \times 2^\varphi \times 2^\varphi\). We recall that \(2^\varphi\) is the set of timed constraints.

1. Skip process:

\[(1.a) \quad \emptyset[\text{skip}\{u\}][x = \text{get}(\mathcal{E})\}
\]

2. Prefix operator:

\[(2.a) \quad \emptyset[a\{u\}; P][x = \text{get}(\mathcal{E})\]

3. Choice operator:

\[(3.a) \quad \rho'[P] \quad \rho'[Q]
\]

\[(3.b) \quad \rho'[P] \quad \rho'[Q]
\]

4. Parallel composition operator:

\[(4.a) \quad \rho'[P] \quad \rho'[Q]
\]

\[(4.b) \quad \rho'[P] \quad \rho'[Q]
\]

5. Hide operator:

\[(5.a) \quad \rho'[P] \quad \rho'[Q]
\]

\[(5.b) \quad \rho'[P] \quad \rho'[Q]
\]
6. Interruption operator:

\[
\begin{align}
(6.a) \quad & \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q}} \quad \frac{\mathcal{P}^{(Ea_y,\varphi,\lambda)}}{\mathcal{P}^{(Ea_y,\varphi,\lambda)(y/x) \triangle \mathcal{Q}}
\end{align}
\]

\[
y = \text{get}(E - ((\psi(P')) - \{x\}) \cup \psi(Q))
\]

\[
(6.b) \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}} \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}}
\]

\[
(6.c) \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}} \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}}
\]

7. Delay operator:

\[
\Theta^d \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P}^{(Ea_y,\varphi,\lambda)}} \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}}
\]

The substitutions \(\varphi[c_x/c_1]\) and \(\lambda[c_y/c_1]\) as well as the union \(\lambda_1 \cup \lambda_2\) are defined in the most obvious way. Now we define the function \(\mathcal{F}^{\leq u}(E)\). Intuitively, the timed constraint \(\mathcal{F}^{\leq u}(E)\) of a given transition \(t\) expresses that all the actions in \(E\) must terminate and the transition \(t\) can happen in the time interval \([0,u]\) starting from the termination moment of the last finished action(s) of \(E\), i.e.:

\[
\mathcal{F}^{\leq u}(E) = \bigwedge_{x:a \in E} (d(a) \leq c_x) \land \bigvee_{x:a \in E} (c_x \leq d(a) + u)
\]

(1)

Definition 8. The delay function \(\varphi + d\) is defined by induction on \(\varphi\) as follows:

\[
\begin{align}
(\varphi_1 \land \varphi_2) + d &= (\varphi_1 + d) \land (\varphi_2 + d) \\
(\varphi_1 \lor \varphi_2) + d &= (\varphi_1 + d) \lor (\varphi_2 + d) \\
(\alpha \leq \varphi) + d &= \alpha + d \leq \varphi \\
(c_x \leq \varphi) + d &= c_x \leq \varphi + d \\
(c_x \leq \beta) + d &= c_x \leq \beta + d
\end{align}
\]

Remark 1. By construction (i.e., by the construction of the timed constraints in the rules (1.a), (1.b), (2.a), (2.b), (4.c), and 7), the timed constraints have the following form:

\[
\varphi = \phi_1 \land \cdots \land \phi_n
\]

\[
\phi_i = \bigwedge_{x:a \in E} (\alpha \leq c_x) \land \bigvee_{x:a \in E} (c_x \leq \beta)
\]

where 

\[
\alpha, \beta \in \mathbb{R}^+ \quad \text{and} \quad \alpha \leq \beta.
\]

We state one of the most properties of the function \(\mathcal{F}^{\leq u}(\cdot) + d\):

Lemma 2. Let \(s_1 \xrightarrow[\mathcal{F}^{\leq u}(E)+d,c_x]} {\mathcal{P} \triangle \mathcal{Q}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}}
\]

\[
\text{Let } s_1 \xrightarrow{(Ea_x,\varphi,\lambda)} \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}}
\]

\[
\text{Let } s_2 \xrightarrow{(Ea_y,\varphi,\lambda)} \quad \frac{\mathcal{P}^{(Ea_x,\varphi,\lambda)}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P}^{(Ea_y,\varphi,\lambda)}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}} \quad \frac{\mathcal{P} \triangle \mathcal{Q}}{\mathcal{P} \triangle \mathcal{Q} \mathcal{P} \triangle \mathcal{Q}}
\]

Definition 9. Given a duration-CSP process \(P\), the denotational semantics of \(P\) over the class of timed-CTS, denoted by \([P]\), consists in associating to \(P\) the timed-CTS which is generated by the transition relation \(\rightarrow \in \mathcal{C} \times \text{Act} \times \mathcal{C}\) given in Section 5 starting from the configuration \(\emptyset [P]\).

Equivalence of the operational and denotational semantics.

We arrive at the final point of this section: we prove that the two semantics are equivalent. The notion of equivalence is formalized through the notion of \(\tau\)-bisimulation.

Let \(f : A \rightarrow B\) and let \(A' \subseteq A\) and \(B' \subseteq B\). The parametrized restrictions of \(f\) w.r.t. its domain and codomain are defined respectively as follows:

\[
f_{\tau}(A') := \{ (a,b) \mid a \in A' \} \quad \text{and} \quad f_{\tau}(B') := \{ (a,b) \mid b \in B' \}
\]

Definition 10. A \(\tau\)-bisimulation linking the states of a timed-CTS and the timed causal configurations of \(\mathcal{C}_{\tau}\) is a binary relation \(\mathcal{R}\) that comes with an events’ bijection \(f : \mathcal{E} \rightarrow \mathcal{E}\), and satisfying the following conditions:

1.1. if \(\langle s,\nu \rangle \xrightarrow{E} \langle s',\nu' \rangle\) then there exists \(P_{\tau} \xrightarrow{\mathcal{F}^{\leq u_{\tau}}} \mathcal{P}_{\tau}'\) such that

\[
i : z : b \in E\text{ if and only if }f(z) : b : t \in F_{\tau}, \text{ for some } t \in \mathbb{R}^+\text{, and}
\]

\[
i : (\langle s',\nu' \rangle, \mathcal{P}_{\tau}') \in \mathcal{R}\text{ where }f' := (f_{\varepsilon}(\psi(\nu') - x)) \pi_{\varepsilon}(\psi(\nu') - y) \cup \{ (x,y) \}.
\]

1.2. if \(\langle s,\nu \rangle \xrightarrow{d} \langle s',\nu' \rangle\) then \(P_{\tau} \xrightarrow{d} \mathcal{P}_{\tau}'\) and \(\langle s,\nu \rangle, \mathcal{P}_{\tau}' \in \mathcal{R}\).

2.1. if \(P_{\tau} \xrightarrow{d} \mathcal{P}_{\tau}'\) then there exists \(\langle s,\nu \rangle \xrightarrow{\mathcal{F}^{\leq u_{\tau}}} \langle s',\nu' \rangle\) such that

\[
i : z : b \in E\text{ if and only if }f(z) : b : t \in F_{\tau}, \text{ for some } t \in \mathbb{R}^+\text{, and}
\]

\[
i : (\langle s',\nu' \rangle, \mathcal{P}_{\tau}') \in \mathcal{R}\text{ where }f' := (f_{\varepsilon}(\psi(\nu') - x)) \pi_{\varepsilon}(\psi(\nu') - y) \cup \{ (x,y) \}.
\]

2.2. if \(P_{\tau} \xrightarrow{d} \mathcal{P}_{\tau}'\) then \(\langle s,\nu \rangle \xrightarrow{d} \langle s',\nu' \rangle\) and \(\langle s,\nu \rangle, \mathcal{P}_{\tau}' \in \mathcal{R}\).

A timed-CTS and a set of timed causal configuration are \(\tau\)-bisimilar if and only if there exists a \(\tau\)-bisimulation containing their initial configurations.

Theorem 1. The operational and the denotational semantics \((\cdot)^{op}\) and \([\cdot]\) are equivalent, i.e. for each duration-CSP process \(P\) there exists a \(\tau\)-bisimulation \(\mathcal{R}\) such that \([\{P]\] = \mathcal{P}_{\tau}^{\text{top}} \in \mathcal{R}\).

6. SIMPLE CASE STUDY

As a simple application we illustrate the use of duration-CSP through a simplified version of the Tick-Tock protocol [LLDS94], the latter has been used for the assessment of timing formal description techniques.
The tick-Tock case contains three entities called sender, receiver and service, see Figure 6.1. Moreover, service interacts with sender and receiver through their SAPs Ss-SAP and Sr-SAP, respectively. In the sequel we restrict ourselves to the specification of the service. The description of the service is as follows. Service transmits data from sender to receiver. The exchanges are performed thought the corresponding SAPs in an atomic way and carried out a data called the cell. Service must satisfies the following requirements:

- Immediate acceptance. This requirement is specified as follows:
  
  process ImmAccept[Del,Sr-SAP]:= Del;  
  ( Sr-SAP(); ImmAccept[Del,Sr-SAP] ) + ImmAccept[Del,Sr-SAP]  
  endproc

- Service. The three above processes have to synchronize on the internal action Del. Since Del is an internal action, it must be hidden. The behaviour of the process Service is as follows:

  process Service[Ss-Sap]:=
  {Frequency[Ss-SAP] |[Ss-SAP]|  
  ( Medium[Ss-SAP,Del] |[Del]| ImmAccept[Del,Sr-SAP])  
  ) \{Del\}  
  endproc

We note that all the actions are atomic apart the action TRANS we denotes the transmission delay. Therefore the duration of TRANS should belong to the interval $[\tau_{\min}, \tau_{\max}]$. As a matter of fact it is not hard to change the semantics of language by considering the actions to be of a variable duration instead of a fixed one. Finally we point out that one of the interesting features of duration-CSP - with its timed causal semantics- is that it allows the refinement of a given action, notably the action TRANS in this example, into a more complicated process which allows an incremental design of the system. The refinement operator as well as its semantics and properties are discussed in the following section.

### 7. ACTION REFINEMENT IN DURATION-CSP

One of the interesting steps during the hierarchical design of complex systems is the refinement of an action $a$ into a process. As a matter of fact, one can associate to each specifica- tion a level of abstraction basing on the details of the actions with compose the specification. For instance, given a specification $E$ of abstraction level $N$, the refinement $\rho(a, P, E)$ of an action $a$ by a process $P$ in the specification $E$ means that when passing from the abstraction level $N$ to $N+1$ the refinement operator will exhibits the internal structure of the action $a$, that is, a would be replaced by the process $P$ at the level $N+1$. There have been many earlier works to curry on action refinement in process algebra, let us mention [CS93, SpC94, FMCW02, KK09].

In this section we enrich the language duration-CSP with the refinement operator $\rho$. The new language is called duration-CSP($\rho$) and in this example, into a more complicated process which allows an incremental design of the system. The refinement operator as well as its semantics and properties are discussed in the following section.

#### The syntax of duration-CSP($\rho$)

The syntax of duration-CSP($\rho$) is given as follows:

- if $P$ is a duration-CSP process then $P$ is again a duration-CSP($\rho$) process,
- if $a$ is an action, $P$ is a duration-CSP process and $Q$ is a duration-CSP($\rho$) process, then $\rho(a, P, Q)$ is a duration-CSP($\rho$) process.

### 6.1 Specification of service with duration-CSP

The specification of service is given in such a way each timed requirement is given as a duration-CSP process.

It is composed of three processes: Frequency, Medium and ImmAccept.

- **Frequency.** The frequency behaviour of service is:
  
  process Frequency[Ss-SAP]:=
  Ss-SAP(); $\Theta$ Frequency[Ss-SAP] + $\Theta$ Frequency[Ss-SAP]  
  endproc

- **Medium.** The Medium must satisfy both the transmission delay and spacing between deliveries requirements:
  
  process Medium[Ss-SAP,Del] :=
  (Ss-SAP; TRANS; Del; Stop ||| Medium[Ss-SAP,Del] ) ||| Del; $\Theta$ Medium [Ss-SAP,Del]  
  endproc

**Figure 6.1: The protocol.**

Frequency. A cell form sender is only accepted from service at precise, punctual instants within a period of $\pi$ units of time.

Transmission delay. Service provides a cell to receiver between $\tau_{\min}$ and $\tau_{\max}$ units of time after its emission.

Spacing between deliveries. There is a delay of at least $\delta$ units of times between two consecutive offers of cells at Sr-SAP.

Immediate acceptance. A cell offered by service to receiver must be immediately accepted by receiver, otherwise the service loses the cell immediately.

Loss free transmission. No cell is lost during its transmission through service.
In order to define the timed causal semantics of the refinement operator \( \rho \), we introduce a new kind of operator on the timed causal configurations \( C_r \), called partial sequencing operator and denoted by \( \gg^x \). Intuitively, the semantics of \( P \gg^x Q \) means that all the actions of \( Q \) which do not depend on the termination of the event \( x \) are in concurrence with the actions of \( P \), however the execution of the remaining actions of \( Q \) must wait for the successful termination of \( P \). Besides the distributivity of the event names over the refinement operator, i.e. for every \( E \in 2^{\mathbb{L} \times \mathbb{R}^+} \) and every process \( \rho(a, P, Q) \),

\[
E_r[\rho(a, P, Q)] = \rho(a, P, E_r[Q])
\]

Again we can extend Lemma 1 to obtain:

**Lemma 3.** Every canonical timed causal configuration has one of the following forms:

\[
E_r[\text{stop}] E_r[\text{skip} \{d\}] \Theta^d P_r E_r[a \{d\}; P] \ P_r + Q_r P_r [[L] \parallel] P_r \ P_r \triangle Q_r P_r \gg^x Q_r
\]

where \( P_r \) and \( Q_r \) are in the canonical form.

The function \( \psi : C_r \to 2^{\mathbb{L} \times \mathbb{R}^+} \) that determines the set of events of a given timed configuration of duration-CSP \( P \) is the same as that of Definition 2 extended with the following rules:

\[
\psi(P \gg^x Q) = \psi(P) \cup (\psi(Q) - \{x\})
\]

\[
\psi(\rho(a, P, Q)) = \psi(Q)
\]

### 7.1 Operational semantics of duration-CSP \( \rho \)

This subsection introduce the operational semantics of duration-CSP \( \rho \) in the same way as we have done with duration-CSP.

**Definition 11.** The timed transition over the timed causal configurations of duration-CSP \( \rho \), denoted again by \( \tau \), is the relation that satisfies the rules 0, \ldots, VIII extended with the following rules:

\[
\begin{align*}
R.1 & \quad \frac{\rho(a, P, Q) \ E_a^x \ P'}{P \gg^x Q \ E_a^x \ P'[z/y] \gg^x Q} \\
& \quad z = \text{get}(E - (\psi(P) - \{y\}) \cup (\psi(Q) - \{x\}))
\end{align*}
\]

\[
\begin{align*}
R.2 & \quad \frac{\rho(a, P, Q) \ E_a^x \ P'}{P \gg^x Q \ E_a^x \ Q'[z/x]} \\
& \quad z = \text{get}(E - ((\psi(P) \cup (\psi(Q') - \{y\}) \cup \{x\}))
\end{align*}
\]

\[
\begin{align*}
R.3 & \quad \frac{\rho(a, P, Q) \ E_a^x \ P'}{P \gg^x Q \ E_a^x \ P \gg^x Q'[z/y]} \\
& \quad z = \text{get}(E - (\psi(P) \cup (\psi(Q) - \{y\}) \cup \{x\}))
\end{align*}
\]

\[
\begin{align*}
R.4 & \quad \frac{\rho(a, P, Q) \ E_a^x \ P'}{\rho(a, P, Q) \ E_a^x \ P' \ x \neq a} \\
& \quad \rho(a, P, Q) \ E_a^x \ P' \ x \neq a
\end{align*}
\]

The rules R.1, R.2, R.3, R.τ.1 and R.τ.2 define the semantics of the partial sequencing operator \( \gg^x \). That is, the rule R.1 expresses the fact that the occurrence of any action in the configuration \( P \) remains possible in the configuration \( P \gg^x Q \); however the remaining of the event \( y \) is necessary because \( y \) may be the event of some action which is already running in the configuration \( Q \). The rule R.2 expresses the case of the successful termination of \( P \). Note that the event \( x \) is renamed with \( z \) which identifies the successful termination of \( P \). The rule R.3 expresses that the occurrence of all the actions of the configuration \( Q \) which do not depend on the termination of the event \( x \) – i.e. on the successful termination of the configuration \( P \) – can be executed in the configuration \( P \gg^x Q \). The rule R.τ.1 shows that the time is allowed only to elapse in the left part of the configuration \( P \gg^x Q \) whenever \( Q \) is waiting for the termination of the event \( x \). However the rule R.τ.2 allows the elapse of time in both parts of the configuration \( P \gg^x Q \) if \( Q \) is not waiting for the termination of \( x \).

The rules R.4, R.5 and R.τ.3 give the semantics of the refinement operator \( \rho \). The rule R.4 shows the case when the configuration \( Q \) provides an action \( b \) which is not subject to the refinement; in this case the action \( b \) remains possible in the configuration \( \rho(a, P, Q) \). The rule R.5 expresses the case when the configuration \( Q \) provides the action \( a \) which has to be refined into the process \( P \). Hence the execution of the action \( a \) must be replaced by the execution of the process \( P \). Since the execution of \( a \) depends on the termination of all the events of \( E_r \), then every action of \( P \) depends also on the termination of the same set of events. Moreover, it is clear that all the actions of \( Q' \) which depend on the termination of \( a \) must also depend on the successful termination of \( E_r[\rho(\cdot)] \), however the remaining actions are executed in parallel with \( E_r[\rho(\cdot)] \). This shows the usefulness of the partial sequencing operator \( \gg^x \) in expressing the semantics of the refinement operator.

The following Theorem shows the main property of the refinement operator \( \rho \); it expresses that the refinement operator preserves the timed causal bisimulation.

**Theorem 2.** For every timed configuration \( P, Q \) of duration-CSP \( \rho \), for every action \( a \) and for every duration-CSP process \( E, \rho(a, E, P) \gg^x \rho(a, E, Q) \).

\[\text{Definition 2.} \quad \text{Indeed we mean the timed causal bisimulation that links the timed configurations and which is defined in a routine way; see the appendix Definition 12.}\]
8. CURRENT AND FUTURE WORKS

At the moment we are looking for a probabilistic extension of the timed causal transition systems in the following way: rather than considering that the actions have a fixed duration, it is more realistic to attribute to them a probabilistic duration that follows a certain distribution, notably a normal (Gaussian) distribution. Within this model, many problems suggest themselves such as the model checking one. This is an orthogonal formalism w.r.t. the probabilistic timed automata [JLS07] where the probabilities are attributed to the transitions rather than the actions.

An other work consists in considering the model checking of the duration logics [CHR91, Lev04] over the timed causal transition systems.

Finally we emphasize that it is not useful to encode the timed-CTS model into the timed automata one since this implies the loss of the notion of true concurrency and gives rise to a combinatorial explosion due to the fact of splitting each action into two events: the starting and the finishing one. The implementation of an environment that integrates the timed-CTS model, the duration-CSP language and the refinement operator $\rho$ should not provide any technical difficulties.

9. REFERENCES

[AD94] R. Alur and D. L. Dill. A theory of timed automata. Theoretical Computer Science, 126:183–235, 1994.
[BCAM00] J.C.M. Baeten and Eindhoven The Netherlands C. A. Middelburg. Process algebra with timing: Real time and discrete time. In Handbook of Process Algebra, pages 627–684. Elsevier, 2000.

[CHR91] Z. Chaochen, C. A. R. Hoare, and Anders P. R. A calculus of durations. Inf. Process. Lett., 40(5):269–276, 1991.
[Cos93] R. J. Coelho Da Costa. Systèmes de transitions étiquetés causaux: une nouvelle approche pour la description du comportement événementiel de systèmes concurrents. PhD thesis, Université de Toulouse 3, 1993.
[CS93] Jean-Pierre Courtiat and Djamel-Eddine Saïdouni. Action refinement in lotos. In PSTV, pages 341–354, 1993.

[FMCW02] Harald Fecher, Mila E. Majster-Cederbaum, and Jinzhao Wu. Refinement of actions in a real-time process algebra with a true concurrency model. Electr. Notes Theor. Comput. Sci., 70(3), 2002.

[Gla90] R.J. Glabbeek. Comparative Concurrency Semantics and Refinement of Actions. PhD thesis, University of Amsterdam, 1990.

[Hoa85] C. A. R. Hoare. Communicating sequential processes. Communications of the ACM, 21:666–677, 1985.

[JLS07] Marcin Jurdzinski, François Laroussinie, and Jeremy Sproston. Model checking probabilistic timed automata with one or two clocks. In TACAS, pages 170–184, 2007.

[KK09] M. Kapus-Kolar. An action refinement operator for e-lotos with true concurrency. Comput. Stand. Interfaces, 31(1):77–87, 2009.

[Lev04] V. I. Levin. Interval logic and some of its applications. In Logical investigations. No. 11 (Russian), pages 172–186. “Nauka”, Moscow, 2004.

[LL97] Luc Léonard and Guy Leduc. An introduction to et-lotos for the description of time-sensitive systems. Comput. Netw. ISDN Syst., 29:271–292, 1997.

[LLD94] Luc Léonard, Guy Leduc, and André Danthine. The tick-tock case study for the assessment of timed firts, 1994.

[MT90] Faron Moller and Chris M. N. Tofts. A temporal calculus of communicating systems. In CONCUR’90, pages 401–415, 1990.

[Saï96] D. Saïdouni. Sémantique de maximalité: application au raffinement d’actions dans LOTOS. PhD thesis, Université de Toulouse 3, 1996.

[SpC94] Djamel-Eddine Saïdouni and Jean-pierre Courtiat. Syntactic action refinement in presence of multiway synchronization. In In proc. of the workshop on Semantics of Specification Languages (SoSL’93, pages 289–303. Springer Verlag, 1994.

[Yi91] Wang Yi. Ccs + time = an interleaving model for real time systems. In ICALP, pages 217–228, 1991.
Appendix: proofs of the statements

**Lemma 1.** Every canonical timed causal configuration in $C_{\tau}$ has one of the following forms:

- $e_\tau[\text{stop}]

- \Theta^d P_r

- \Theta^d \tau

- \Theta^d P_r = \Theta^d P_r \cup \Theta^d \tau

- \Theta^d \tau \cup \Theta^d \tau$

where $P_r$ and $Q_r$ are in the canonical form.

**Proof.** We prove by induction that every timed causal configuration which is not under one of these forms can be reduced by distributing the set of events over the algebraic operators. The proof of the same lemma but upon the untimed configurations was given in Cosm, however we adapt it to the timed configurations.

If a given timed configuration $\mathcal{R}'_r$ can be obtained from $\mathcal{R}_r$ by distributing the set of events over the algebraic operators then we write $\mathcal{R}_r \hookrightarrow \mathcal{R}'_r$. We only consider the cases where the timed causal configuration is of the form $e_\tau[R]$:

- $R = \Theta^d P_r.

- R = P \cup Q.

- R = P \cup Q.

- R = P \cup Q.

- R = P \cup Q.

This ends the proof of Lemma 1.

**Lemma 2.** Let $s_1$ be a timed transition of a given timed-CTS. The action $b$ is enabled in the timed interval $[\tau + d, \tau + d + u]$ where $d \in \mathbb{R}^+$ is the time stamp of the termination of the last finished action in $E$.

**Proof.** Recall the definition of $\mathcal{F}^{\leq u}$ (see Equation 1) at page 4:

\[
\mathcal{F}^{\leq u}(E) = \bigwedge_{x \in E} (d(a) \leq c_x) \land \bigvee_{x \in E} (c_x \leq d(a) + u)
\]

(2)

therefore by the definition of + (see Definition 8), we get

\[
(F^{\leq u} + d)(E) = \bigwedge_{x \in E} (d(a) + d \leq c_x) \land \bigvee_{x \in E} (c_x \leq d(a) + d + u)
\]

On the one hand, the constraint $\Phi_1$ ensures that the action $b$ is enabled in the interval $[\tau + d, \infty]$, where $\tau$ is the time stamp of the termination of the last finished action in $E$.

On the other hand, the condition $\Phi_2$ states that the action $b$ is enabled in the interval $[0, \tau_{\text{max}}]$ where

\[
\tau_{\text{max}} = \max_{x \in E} (\tau_x + d + u) = \max_{x \in E} (\tau_x) + d + u
\]

where $\tau_x$ is the time stamp of the termination of the action $a \in E$. Hence,

\[
\tau_{\text{max}} = \tau + d + u
\]

Therefore, the constraint $\Phi_2$ states that the action $b$ is enabled in the interval $[\tau + d, \tau + d + u]$. We conclude that the constraint $\Phi_1 \land \Phi_2$ states that the action $b$ is enabled in the interval $[\tau + d, \tau + d + u]$.

**Theorem 1.** The operational and the denotational semantics ($\cdot)^\sigma$ and [] are equivalent, i.e. for each duration-CSP process $P$ there exists a $\tau$-bisimulation $\mathcal{R}$ such that $(\cdot)^\sigma \in \mathcal{R}$.

**Proof.** We construct a binary relation $\mathcal{R}$ linking the elements of $[P]$ and $P^\tau$. Afterward we prove that it is a $\tau$-bisimulation. First of all we came assume that $\mathcal{R}$ comes with the identity function $Id : \text{appendix.tex, v0.1,v202009/10/1719 : 03 : 50belkhirExp}$ over the set of events, i.e. we do not need to rename the events.

We let

\[
\mathcal{R} = (\mathcal{R}_0 \cup \mathcal{R}_0) \cup \cdots \cup (\mathcal{R}_n \cup \mathcal{R}_n) \cup \cdots
\]

where

\[
\mathcal{R}_0 = \{ (\rho[P], \circ [0[P]]) \cup \{ (P, Q_r) \text{ s.t. } \exists d \in \mathbb{R} (\rho[P], \circ d \rightarrow P) \text{ and } \circ [0[P] \sim Q_r) \}
\]

\[
\mathcal{R}_{m+1} = \{ (P^{m+1}, Q_r^{m+1}) \text{ s.t. } \exists (P^m, Q_r^m) \in \mathcal{R}_m \text{ s.t. } P^m \circ d \rightarrow P^m \text{ and } Q_r^m \circ d \rightarrow Q_r^m \text{ by some action } a \} \cup \{ (P', Q') \text{ s.t. } \exists d \in \mathbb{R} \text{ s.t. } P^m \circ d \rightarrow P' \text{ and } Q_r^m \circ d \rightarrow Q' \}
\]

During the construction of $\mathcal{R}_i$, $i = 0, \ldots, n$, we require that the invariants $\text{(SYNCH1)}$ and $\text{(SYNCH2)}$ hold.

The invariant $\text{(SYNCH1)}$ is defined as follows: for each pair $((\mathcal{R}, \nu), \psi) \in \mathcal{R}_n$, the pair $(\nu(\mathcal{R}), \nu(\mathcal{R}))$ is synchronized in the following sense:

\[
z : b : t_z \in \psi(\mathcal{R}_x) \iff z : b \in \psi(\mathcal{R}) \text{ and } t_z = \nu(c_z)
\]

To give the definition of the invariant $\text{(SYNCH2)}$ we need some notations. Let us define the function $\mathcal{F}(\cdot)$ that takes a timed configuration (in $\mathcal{C}$, or in $\mathcal{I}$) and returns only the duration-CSP process by deleting recursively the set of
events:
\[ \mathcal{F}(\mathcal{E}, [\text{stop}]) = \text{stop} \]
\[ \mathcal{F}(\mathcal{E}, [\text{skip} \{ d \}]) = \text{skip} \{ d \} \]
\[ \mathcal{F}(\mathcal{E}, [a \{ d \}; P]) = a \{ d \}; P \]
\[ \mathcal{F}(\Theta^d \mathcal{P}_r) = \Theta^d \mathcal{F}(\mathcal{P}_r) \]
\[ \mathcal{F}(\mathcal{P}_r \setminus L) = \mathcal{F}(\mathcal{P}_r) \setminus L \]
\[ \mathcal{F}(\mathcal{P}_r + \mathcal{Q}_r) = \mathcal{F}(\mathcal{P}_r) + \mathcal{F}(\mathcal{Q}_r) \]
\[ \mathcal{F}(\mathcal{P}_r \{L\} \mathcal{Q}_r) = \mathcal{F}(\mathcal{P}_r) \{L\} \mathcal{F}(\mathcal{Q}_r) \]
\[ \mathcal{F}(\mathcal{P}_r \Delta \mathcal{Q}_r) = \mathcal{F}(\mathcal{P}_r) \Delta \mathcal{F}(\mathcal{Q}_r) \]

We let also, for \( i \in \mathbb{N} \), \( \mathcal{R}_i \) to be:
\[ \mathcal{R}_i = \{ \{ (o[P], O), \sigma[P] \} \}
\]
\[ \{ (R^i, R^+) \in \mathcal{R}_i \text{ s.t. } \exists (R^{i-1}, R^+, R^+) \in \mathcal{R}_{i-1} \text{ s.t. } \]
\[ R^{i-1} \xrightarrow{\sigma} R^i \text{ and } R_{\sigma}^{i-1} \xrightarrow{a} R^i \text{ for some } a \}
\]
\[ \text{if } i \geq 1 \]

The invariant \( (\text{SYNCH2}) \) is given by:
\[ \forall i \in \mathbb{N}, \forall (R^i, R^+, \mathcal{O}) \in \mathcal{R}_i, \text{ we have that} \]
\[ (i) \mathcal{F}(R^i) = \mathcal{F}(R^+) \text{ and (ii) the pair} (R^i, R^+) \text{ is synchronized} \]
\[ (\text{SYNCH2}) \]

Now we shall prove that \( \mathcal{R} \) is a \( \tau \)-bisimulation. For this aim, it is enough to prove that, for each \( n \in \mathbb{N} \), \( \mathcal{R}_n \cup \mathcal{R}_{n+1} \) is a \( \tau \)-bisimulation i.e. \( \mathcal{R}_n = \emptyset \). The proof is by induction on \( n \).

**Initial step** \( n = 0 \). i.e. we consider \( \mathcal{R}_0 \) defined by:
\[ \mathcal{R}_0 = \{ \{ (o[P], O), \sigma[P] \} \cup \}
\[ \{ (P, Q_r) \text{ s.t. } \exists d \in \mathbb{R} \{ (o[P], O) \xrightarrow{d} P \}
\]
and \( o[P] \xrightarrow{d} \}

In this step we shall prove that (i) \( \mathcal{R}_0 = \emptyset \), (ii) \( \mathcal{R}_0 \) satisfies the invariants \( (\text{SYNCH1}) \) and \( (\text{SYNCH2}) \), and (iii) \( \mathcal{R}_1 \) satisfies the invariant \( (\text{SYNCH2}) \). The proof now is by structural induction on \( P \).

**Case (i).** The case \( P = \text{stop} \) is obvious.

**Case (ii).** \( P = \text{skip} \{ u \} \). The rule \( (1.a) \) of the denotational semantics ensures that \( \forall d \leq u \) there is a derivation
\[ o[\text{skip} \{ u \}], O \xrightarrow{d} o[\text{skip} \{ u \}], O + d \]

In the same way, the rule \( (1.\tau) \) of the operational semantics allows, for each \( d \in [0, u] \), the derivation
\[ o[\text{skip} \{ u \}] \xrightarrow{d} o[\text{skip} \{ u - d \}] \]

This shows that \( \mathcal{R}_0 = \emptyset \), therefore \( \mathcal{R}_0 \cup \mathcal{R}_1 \) is a \( \tau \)-bisimulation.

Now we show that \( \mathcal{R}_0 \) satisfies the invariants \( (\text{SYNCH1}) \) and \( \mathcal{R}_1 \) satisfies the invariant \( (\text{SYNCH2}) \). Note that \( \mathcal{R}_0 \) satisfies trivially the invariant \( (\text{SYNCH1}) \) because \( \psi(o[\text{skip} \{ u \}], O) = \psi(o[\text{skip} \{ u - d \}], O) = \emptyset \). Also, \( \mathcal{R}_0 \) satisfies the invariant \( (\text{SYNCH2}) \) because \( \mathcal{R}_0 = \{ (o[\text{skip} \{ u \}], O), o[\text{skip} \{ u \}] \} \). To show that \( \mathcal{R}_1 \) satisfies the invariant \( (\text{SYNCH2}) \), we consider \( \mathcal{R}_1 \). The latter is obtained first by applying the rule \( (1.a) \) of the denotational semantics to \( o[\text{skip} \{ u \}], O + d \) giving rise to the derivation:
\[ o[\text{skip} \{ u \}], O + d) \xrightarrow{d} (\text{[\text{stop}], O + d}[, O \rightarrow 0]) \]

And by applying the rule \( (1.a) \) of the operational semantics to the configuration \( o[\text{skip} \{ u - d \}] \) giving rise to the derivation:
\[ o[\text{skip} \{ u - d \}] \xrightarrow{d} (\text{[\text{stop}], O \rightarrow 0}) \]

Therefore
\[ \mathcal{R}_1 = \{ ((\text{[\text{stop}]}, O + d)[O \rightarrow 0]), (\text{[\text{stop}]}) \} \]

Note that \( \mathcal{R} \) satisfies the invariant \( (\text{SYNCH2}) \) because
\[ (i) \mathcal{F}(\text{[\text{stop}]}) = \mathcal{F}(\text{[\text{stop}]}) = \text{stop} \]

and
\[ (ii) \text{ clearly the pair} ((\text{[\text{stop}]}, O + d)[O \rightarrow 0]), (\text{[\text{stop}]}) \]

is synchronized since the clock \( c_0 \) is reset to zero.

**Case (iii).** The case \( P = a \{ u \}; Q \) is similar to the previous one apart that we deal here with the action \( a \) instead of \( \delta \), and with the process \( Q \) instead of the process \( \text{stop} \).

**Case (iv).** The case \( P = Q + R \) is straightforward by applying the induction hypothesis to \( Q \) and \( R \).

**Case (v).** \( P = P_1 || L || P_2 \). First we show that \( \mathcal{R}_0 = \emptyset \). The rule \( IV.\tau \) of the operational semantics implies that that if
\[ o[P_1 || L || P_2] \xrightarrow{d} o[P' || L || P_2] \]

then
\[ o[P_1] \xrightarrow{d} o[P'] \]

i = 1, 2.

By applying the induction hypothesis to both \( P_1 \) and \( P_2 \) we get the possible derivations:
\[ o[P_1 || L || P_2], O \xrightarrow{d} o[P_1 || L || P_2], O + d \]

Hence
\[ (o[P_1 || L || P_2], O) \xrightarrow{d} (o[P_1 || L || P_2], O + d) \]

This shows that \( \mathcal{R}_0 = \emptyset \). Note that \( \mathcal{R}_0 \) satisfies the invariants \( (\text{SYNCH1}) \) and \( (\text{SYNCH2}) \) (the same arguments used in \( \text{Case (ii)} \) hold). Let us show that \( \mathcal{R}_1 \) satisfies the invariant \( (\text{SYNCH2}) \). To this goal let \( a \) be an action, we consider the case when \( a \notin L \cup \{ \delta \} \) and \( i = 1 \). The case when \( a \notin L \cup \{ \delta \} \) and \( i = 2 \) and the case when \( a \in L \cup \delta \) are handled similarly. Let \( i = 1 \) and assume the derivation
\[ o[P'_1] \xrightarrow{a \in [0, u]} [Q'_1] \]

The induction hypothesis shows that the following derivation is possible:
\[ o[P_1], O + d \xrightarrow{a \in [0, u]} (\{ x.a \} [Q'_1] || [Q_1], O + d[O \rightarrow 0]) \]

and ensures that \( \mathcal{F}(\{ x.a \} [Q'_1] || [Q_1]) = \mathcal{F}(\{ x.a \} [Q_1]) = Q_1 \). Therefore by applying the rule \( IV.a \) of the operational semantics and considering the derivation \( \mathcal{R}_1 \) above we get the derivation:
\[ o[P'_1] || [L] || o[P_2] \xrightarrow{a \in [0, u]} [Q'_1] || [L] || o[P_2] \]
Also by applying the rule (4.a) of the denotational semantics and considering the rule (3) above we get the derivation:

\[ \langle g, (P_0, \emptyset), O + d \rangle \xrightarrow{\sigma_{x \rightarrow y}} \langle (x \rightarrow y) \circ Q_1, |L| \circ Q_2, O + d([x] \rightarrow 0) \rangle \]

Thus

\* \( R_1 = \{((x \rightarrow y) \circ Q_1, |L| \circ Q_2, O + d([x] \rightarrow 0)), (x \rightarrow y) \circ Q_2, O \}

and it is easy to check that \( R_1 \) satisfies the invariant (SYNCH₂).

**Case (vi).** The cases of the hide operator (rules (V.a) and (V.b)) and of the interruption operator (rules (VI.a), (VI.b) and (VI.c)) are handled by the induction machinery.

**Case (vii).** If \( P = \emptyset^Q \), then it suffices to prove the following Claim:

**Claim 1.** Let \( d \in R^+ \), \( tr = s \xrightarrow{(e, x \rightarrow y, \lambda)} s' \) be a transition of a given time-CTS and \( tr_+ \) be the same transition apart that we replace \( e \) with \( e + d \), i.e. \( tr_+ = s_+ \xrightarrow{(e + d, x \rightarrow y, \lambda)} s_+ \). Then, the transition \( tr \) allows the action \( a \) at the time stamp \( \tau \) if and only \( tr_+ \) allows \( a \) at the time stamp \( \tau + d \).

**Proof.** [of the Claim] Straightforward from the definition of the delay function \( + \) (see Definition 3) since \( e + d \) lifts every (atomic) constraint \( \alpha \leq e \) to \( \alpha + d \leq e + d \), and \( e + d \leq \beta + d \). This ends the proof of the Claim. \( \Box \)

**Induction step:** \( n > 0 \).

That is, we consider \( R_{n+1} \) defined above by:

\[ R_{n+1} = \{(P, Q) \mid \exists (P^{n-1}, Q^{n-1}) \in R_n \text{ s.t. } P^{n-1} \xrightarrow{E}[a] P \text{ and } Q_{n-1} \xrightarrow{E}[a] Q \text{ for some action } a \} \cup \]

\{ (P', Q') \mid \exists d \in R \text{ s.t. } P \xrightarrow{d} P' \text{ and } Q_{n+1} \xrightarrow{d} Q' \}

We recall that the induction hypothesis implies that \( R_{n-1} \) satisfies the invariants (SYNCH₁) and (SYNCH₂), and that \( R_n \) satisfies the invariant (SYNCH₂). As we have done in the initial step, in this step we shall prove that (i) \( R_n = \emptyset \), (ii) \( R_n \) satisfies the invariants (SYNCH₁) and (SYNCH₂) and (iii) \( R_{n+1} \) satisfies the invariant (SYNCH₂). As a consequence of the induction hypothesis \( R_n \) may be written as:

\[ R_n = \{((x \rightarrow y) \circ Q_1, |L| \circ Q_2) \mid \exists (P^{n-1}, Q^{n-1}) \in R_{n-1} \text{ s.t. } P^{n-1} \xrightarrow{E}[a] E[P] \text{ and } Q_{n-1} \xrightarrow{E}[a] E[R_{n+1} \circ Q_2] \text{ for some action } a \} \cup \]

\{ ((P', Q'), Q) \mid \exists d \in R \text{ s.t. } E[P], \nu \xrightarrow{d} P' \text{ and } E[R_{n+1}] \xrightarrow{d} Q' \}

where the pair \( (E, E_r) \) is synchronized. Again, the proof is by structural induction on \( P \) and similar to the one given in the initial step.

**Case (i).** The case \( P = \emptyset \) is obvious because the pair \( (E, E_r) \) is synchronized.

**Case (ii).** \( P = \text{skip}(u) \). The rule (1.b) of the denotational semantics ensures that \( \forall 0 \leq d \leq u \) and counting form the moment when all the actions of \( E \) have finished (see the definition of \( E \leq E_r \)) there is a derivation

\[ \langle E, \text{skip}(u), \nu \rangle \xrightarrow{d} \langle E, \text{skip}(u), \nu + d \rangle \]

In the same way, the rule (1.\( \tau \)) of the operational semantics allows, for each \( e \in [0, u] \), such that all the actions of \( E_r \) have finished, the derivation

\[ E_r, \text{skip}(u) \xrightarrow{d} E_r, \text{skip}(u - d) \]

Since the pair \( (E, E_r) \) is synchronized thus \( R_n = \emptyset \), therefore \( R_n \cup R_n \) is a \( \tau \)-bimorphism. Using the same arguments of the **Case (ii)** of the initial step one can see easily that \( R_n \) satisfies the invariants (SYNCH₁), (SYNCH₂) and \( R_{n+1} \) satisfies the invariant (SYNCH₂).

**Case (iii).** The case \( P = \alpha \{ u \}; Q \) is similar to the previous one apart that we deal here with the action \( a \) instead of \( \delta \), and with the process \( Q \) instead of the process \( stop \).

**Case (iv).** The case \( P = Q + R \) is straightforward by applying the induction hypothesis to \( Q \) and \( R \).

**Case (v).** \( P = \{ P_1, |L| \}, \emptyset \). First we show that \( R_n = \emptyset \). The rule (IV.\( \tau \)) of the operational semantics implies that if

\[ E_r \circ P_1, |L| \xrightarrow{d} E_r \circ P_1'| \]

then

\[ E_r \circ P_1 = E_r \circ P_1' \quad i = 1, 2. \]

By applying the induction hypothesis to both \( P_1 \) and \( P_2 \) we get the possible derivations:

\[ (E_r \circ P_1, |L|) \xrightarrow{d} (E_r \circ P_1', |L|, P_2' \quad i = 1, 2. \]

Hence

\[ (E_r \circ P_1, |L|, P_2) \xrightarrow{d} (E_r \circ P_1', |L|, P_2', P_2' \quad i = 1, 2. \]

Since the pair \( (E, E_r) \) is synchronized, then \( R_n = \emptyset \). Note that for the same reason, \( R_n \) satisfies trivially the invariants (SYNCH₁) and (SYNCH₂). Let us show that \( R_1 \) satisfies the invariant (SYNCH₂). To this goal let \( a \) be an action, we consider the case when \( a \notin L \cup \{ \delta \} \) and \( i = 1 \). The case when \( a \in L \cup \{ \delta \} \) and \( i = 2 \) and the case when \( a \in L \cup \{ \delta \} \) are handled similarly. Let \( i = 1 \) and assume the derivation:

\[ E_r \circ P_1' \xrightarrow{E}[a] (x \rightarrow a) \circ Q_1' \]

The induction hypothesis shows that the following derivation is possible:

\[ (E_r \circ P_1', |L|) \xrightarrow{d} (E_r \circ P_1', |L|, P_2', P_2' \quad i = 1, 2. \]

and ensures that \( \mathcal{F}(E, (x \rightarrow a) \circ Q_1') = \mathcal{F}(E, (x \rightarrow a) \circ Q_1) = Q_1 \). Therefore by applying the rule (I.\( \nu \)) of the operational semantics and considering the derivation (4) above we get the derivation:

\[ E_r \circ P_1' \xrightarrow{E}[a] (x \rightarrow a) \circ Q_1 |L| \circ P_2' \]

\[ E_r \circ P_1 \]
Also by applying the rule (4.a) of the denotational semantics and considering the rule (9) above we get the derivation:

\[ (\langle x.a \rangle [Q_1] \sqcup [L] \sqcup [P_2], O + d) \overset{a_0}{\rightarrow} (\langle x.a \rangle [Q_1] \sqcup [L] \sqcup [P_2], O + d[c_x \rightarrow 0]) \]

Thus

\[ \mathcal{R}_1 = \{(x.a)[Q_1] \sqcup [L] \sqcup [P_2], O + d[c_x \rightarrow 0],
\langle x.a \rangle [Q_1] \sqcup [L] \sqcup [P_2]\} \]

and it is easy to check that \( \mathcal{R}_1 \) satisfies the invariant SYNCH2.

**Case (vi).** The cases of the hide operator (rules (V.a) and (V.b)), of the interruption operator (rules (VI.a), (VI.b) and (VI.c)), and the delay operator are handled by the induction machinery.

This ends the proof of Theorem 1.

**Theorem 2.** For every timed configuration \( P, Q \) of duration-CSP, for every action \( a \) and for every duration-CSP process \( E \), if \( P \rightarrow_{\tau} Q \) then \( \rho(a, E, P) \rightarrow_{\tau} \rho(a, E, Q) \).

**Proof.** First we construct a binary relation linking the elements of \( \rho(a, E, P) \) and \( \rho(a, E, Q) \), and second we prove that it is a timed causal bisimulation.

We let \( \mathcal{R} = \mathcal{R}_1 \sqcup \mathcal{R}_2 \) where

\[ \mathcal{R}_1 = \{(\rho(a, E, P),\rho(a, E, Q))_f \text{ s.t. } (P, Q)_f \in \mathcal{R}' \} \]

such that \( \mathcal{R}' \) is a timed causal bisimulation, such bisimulation does exist by the hypothesis of the Theorem.

\[ \mathcal{R}_2 = \{(P \gg \tau P\gg \tau P) \gg \tau Q^+ \}_f \text{ s.t. } (P\gg \tau P\gg \tau P)[x/y], Q^+\gg \tau y \}_f \in \mathcal{R} \}

where

\[ v \notin \psi(P^+) - \{x\} \cup f_1^{-1}(\psi(Q^+) - \{y\}) \]

\[ w \notin f_1[\psi(P^+) - \{x\} \cup (\psi(Q^+) - \{y\})] \]

\[ f' = f_1[\psi(P^+) - \{x\}] \cup f_1^{-1}(\psi(Q^+) - \{y\}), \text{ and} \]

\[ f = f' \cup I_{d}(\psi(P^+)) \]

Now we show that \( \mathcal{R} \) is a timed causal bisimulation.

**Initial step**

That is, we verify that \( \mathcal{R}_1 \) is a timed causal bisimulation:

1. If \( \rho(a, E, P) \overset{E_{a_0}}{\rightarrow} H \) then we distinguish two cases according to \( H \):

   - \( H \equiv \rho(a, E, P') \), therefore \( P \overset{E_{a_0}}{\rightarrow} P' \) and \( a \neq b \).

   According to the hypothesis there exists a derivation \( \rho(a, E, Q) \overset{E_{a_0}}{\rightarrow} \rho(a, E, Q') \) such that

   (a) the definition of \( f \) ensures that for each \( u \in \psi(\rho(a, E, P)) \), if \( u \notin E \) and \( f(u) \in \psi(\rho(a, E, Q)) \) then \( f(u) \notin F_{\tau} \),

   (b) since there exist \( v, w \in E \) such that

   \( (P\gg \tau P[y/x], Q^\prime[w/y])_f \in \mathcal{R}' \)

   then

   \[ f'' = f_1[\psi(P^+) - \{x\}] \cup f_1^{-1}(\psi(Q^+) - \{y\}) \]

   \[ \cup \{v, w\} \]

   by using the definition ?? it follows that

   \[ (\rho(a, E, P\gg \tau P[y/x]),\rho(a, E, Q^\prime[w/y]))_f \in \mathcal{R}_1 \]

   where

   \[ f = f_1[\psi(P\gg \tau P[y/x])] \cup f_1^{-1}(\psi(Q^+) - \{x\}) \]

   \[ \cup \{v, w\} \]

   - \( H \equiv R \gg \tau \rho(a, E, P') \), then \( P \overset{E_{a_0}}{\rightarrow} P' \) and \( \rho[E] \overset{a_0}{\rightarrow} R \). According to the hypothesis we have that \( Q \overset{E_{a_0}}{\rightarrow} Q' \), and by taking

   \[ x \notin \psi(P - \{x\}) \cup \psi(Q^+) - \{s\}, \]

   it follows that

   \[ \rho(a, E, Q) \overset{E_{a_0}}{\rightarrow} \rho(a, E, Q') \]

2. similar to 1.

3. If \( \rho(a, E, P) \overset{d}{\rightarrow} \rho(a, E, P') \), then \( P \overset{d}{\rightarrow} P' \). According to the hypothesis there exists a derivation \( P \gg \tau P' \) such that \( (P', Q')_f \in \mathcal{R}_1 \) for some \( f \), therefore it follows that \( (\rho(a, E, P'), \rho(a, E, Q'))_f \in \mathcal{R}_1 \).

**Induction step.** In this step we consider the elements of \( \mathcal{R}_2 \), these elements are of the form \( (P \gg \tau P\gg \tau P \gg \tau Q^+) \), where \( P \gg \tau P[y/x], Q^\prime[w/y])_f \in \mathcal{R} \) with

\[ f' = f_1[\psi(P \gg \tau P[y/x])] \cup f_1^{-1}(\psi(Q^+) - \{y\}) \cup \{v, w\} \]

1.1. \( P \gg \tau P^+ \overset{E_{a_0}}{\rightarrow} H \), we distinguish three cases according to \( H \):

   - \( H \equiv \rho(a, E, P') \), then \( P \overset{E_{a_0}}{\rightarrow} P' \). By assuming that \( z \notin \psi(P' \gg \tau P') \cup f_1^{-1}(\psi(Q^+) \cup \{y\}), \)

   and applying the rule R.1 we obtain the derivation \( P \gg \tau Q^+ \overset{E_{a_0}}{\rightarrow} P' \gg \tau Q^+ \), and we have done.

   - \( H \equiv \rho'[z/x] \), then \( P \overset{E_{a_0}}{\rightarrow} P' \) and \( a = i \).

   By assuming that \( z \notin \psi(P^+) \cup f_1^{-1}(\psi(Q^+) - \{y\}), \)

   and applying the rule R.2 we obtain the derivation \( P \gg \tau Q^+ \overset{E_{a_0}}{\rightarrow} Q^+ \gg \tau y \), and we have done.

   - \( H \equiv P \gg \tau P', \) then the rule R.3 implies that \( P^+ \overset{E_{a_0}}{\rightarrow} P^+ \); by applying the induction hypothesis there exists a derivation \( Q^\prime \gg \tau P^+ \). By assuming \( s \notin \psi(P') \) we get \( Q^{+} \gg \tau P^+ \overset{E_{a_0}}{\rightarrow} Q^+ \gg \tau Q^+ \).

1.2. \( P \gg \tau P^+ \overset{d}{\rightarrow} H \), we distinguish two cases according to \( H \):

   - \( H \equiv \rho'[z/x] \gg \tau P^+ \), the rule R.1 implies that \( P \overset{d}{\rightarrow} P' \) with \( z \notin \psi(P^+) \). The induction hypothesis ensures that \( (P^+, Q^+) \in \mathcal{R} \), hence, \( y \in \psi(Q^+) \).

   By applying the rule R.1 we get the derivation \( P \gg \tau Q^+ \overset{d}{\rightarrow} P' \gg \tau Q^+ \).

   - \( H \equiv \rho' \gg \tau P^+ \), the rule R.2 implies that \( P \overset{d}{\rightarrow} P^+ \) and \( P^+ \overset{d}{\rightarrow} P^+ \) with \( x \notin \psi(P^+) \). Hence \( y \notin \psi(Q^+) \).

   By applying the rule R.2 we get the derivation: \( P \gg \tau Q^+ \overset{d}{\rightarrow} P' \gg \tau Q^+ \).

\[ \square \]
6. ON THE TIMED CAUSAL BISIMULATION OVER THE TIMED CONFIGURATIONS

Definition 12. A $\tau$-bisimulation linking the timed causal configurations of $C_\tau$ is a binary relation $R$ that comes with an events' bijection $f : E \to E$, and satisfying the following conditions:

1.1. if $Q_\tau \xrightarrow{E_{\sigma}} Q'_\tau$ then there exists $P_\tau \xrightarrow{F_{\sigma}} P'_\tau$ such that
   i. $z : b : t \in E_\tau$ if and only if $f(z) : b : t \in F_\tau$, for some $t \in \mathbb{R}^+$, and
   ii. $(Q'_\tau, P'_\tau) f' \in R$ where
       $f' := (f \pi_1(\psi(Q'_\tau)) - x) \sqcup (\psi(P'_\tau) - y) \cup \{(x, y)\}.$

1.2. if $Q_\tau \sim_d Q'_\tau$ then $P_\tau \sim_d P'_\tau$ and $(Q'_\tau, P'_\tau)_f \in R$.

2.1. if $P_\tau \xrightarrow{E_{\sigma}} P'_\tau$ then there exists $Q_\tau \xrightarrow{F_{\sigma}} Q'_\tau$ such that
   i. $z : b : t \in E_\tau$ if and only if $f(z) : b : t \in F_\tau$, for some $t \in \mathbb{R}^+$, and
   ii. $(Q'_\tau, P'_\tau) f' \in R$ where
       $f' := (f \pi_1(\psi(Q'_\tau)) - x) \sqcup (\psi(P'_\tau) - y) \cup \{(x, y)\}.$

2.2. if $P_\tau \sim_d P'_\tau$ then $Q_\tau \sim_d Q'_\tau$ and $(Q'_\tau, P'_\tau)_f \in R$. 