Action Abstraction in Timed Process Algebra

The Case for an Untimed Silent Step

Michel A. Reniers and Muck van Weerdenburg

Department of Mathematics and Computer Science
Eindhoven University of Technology (TU/e)
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands
M.A.Reniers@tue.nl M.J.van.Weerdenburg@tue.nl

Abstract. This paper discusses action abstraction in timed process algebras. It is observed that the leading approaches to action abstraction in timed process algebra all maintain the timing of actions, even if these are abstracted from.
This paper presents a novel approach to action abstraction in timed process algebras. Characteristic for this approach is that in abstracting from an action, also its timing is abstracted from. We define an abstraction operator and a timed variant of rooted branching bisimilarity and establish that this notion is an equivalence relation and a congruence.

1 Introduction

One of the main tools in analysing processes in a process algebra setting is abstraction. Abstraction allows for the removal of information that is regarded as unobservable (or irrelevant) for the verification purpose at hand. Abstraction is introduced in the form of an action abstraction operator, called hiding, or in the form of data abstraction through abstract interpretations. In action hiding, certain action names are made anonymous and/or unobservable by replacing them by a predefined *silent step* (also called internal action) denoted by τ.

In the field of untimed process algebra, there is reasonable consensus about the properties of the silent step. In ACP-style process algebras [BK84] the notion of (rooted) branching bisimilarity, as put forward by Van Glabbeek and Weijland in [GW89,GW96], is mostly adopted. The few timed versions of rooted branching bisimilarity found in the literature (see [Klu93,BB95,Zwa01]) all agree on maintaining the timing of actions, even if these actions are abstracted from. In all of these approaches the passing of time by itself (i.e., without subsequent action execution or termination) can be observed. As a consequence, not as many identifications between processes can be made as is desirable for verification purposes.

Therefore, we study an action abstraction mechanism that not only abstracts from an action, but also from its timing. We introduce an *untimed silent step* into a timed process algebra. We define a timed version of rooted branching bisimilarity based on this untimed silent step, show that it is an equivalence and a congruence, and present a remarkably straightforward axiomatisation for this notion of equivalence. We give a short account of the identifications between processes that can be obtained using this equivalence. This is done by showing simplifications of the PAR protocol using the notions of equivalence from the literature and the notion introduced in this paper.

It should be mentioned that when studying timed process algebras (or timed automata for that matter), one encounters a number of different interpretations of the interaction between actions and time. There are the so-called two-phase models, where the progress of time is modeled separately from action execution, and there is the time-stamped setting, where time progress and action execution are modeled together. Two-phase models are used in [BM02], and time-stamped models are found in timed μCRL [RGvdZvW02], for example. In this paper, we study timed rooted branching bisimilarity in the context of an absolute time, time-stamped model.

Structure First, we introduce a simple timed process algebra with absolute timing and a time-stamped model (Section 2). This process algebra serves as a vehicle for our discussions on abstraction and equality of processes. It contains primitives that are fundamental to virtually every timed process algebra. In Section 3, we discuss the notions of timed rooted branching bisimilarity as they are encountered in the literature. In Section 4, we adapt the timed process algebra to incorporate our ideas for abstraction and equality for timed processes interpreted in a time-stamped model. In Section 5, we illustrate the consequences of our definitions on the PAR protocol. In Section 6, we present axioms for timed strong bisimilarity and timed rooted branching bisimilarity. In Section 7, we discuss some standard extensions of our rather minimal setting in some limited depth. In Section 8, we discuss the possibilities and impossibilities of adapting our notions to other settings in timed process algebra from the literature. Section 9 wraps up the paper.

2 The Universe of Discourse

In this section, we introduce a simple time-stamped process algebra without abstraction that serves well for (1) a more formal exposition of our discomfort with the existing ways of dealing with abstraction in timed process algebra, (2) a discussion of the possible solutions, and (3) the treatment of the chosen solution.

The timed process algebra presented in this section, $\mathrm{BSP}^{@}_{\mathrm{abs}}$ (for *Basic Sequential Processes* with *absolute time* and *time-stamping*), is an extension of the process theory BSP from [BBR07] with absolute-timing and time-stamping (both syntactically and semantically) inspired by the process algebra *timed* $\mu\mathrm{CRL}$ [RGvdZvW02][1].

We first present the starting point of our deliberations. We assume a set Time that is totally ordered by $\leq$ with smallest element 0 that represents the time domain[2]. We also assume a set Act of actions, *not* containing τ.

The signature of the process algebra $\mathrm{BSP}^{@}_{\mathrm{abs}}$ consists of the following constants and operators:

- for each $t \in \mathrm{Time}$, a timed deadlock constant $0^{@t}$. The process $0^{@t}$ idles upto time t and then deadlocks.
- for each $t \in \mathrm{Time}$, a timed termination constant $1^{@t}$. The process $1^{@t}$ idles upto time t and then terminates successfully.
- for each $a \in \mathrm{Act}$ and $t \in \mathrm{Time}$, an action prefix operator $a^{@t}.\_$. The process $a^{@t}.p$ represents the process that idles upto time t, executes action a at that time and after that behaves as process p insofar time allows.
- the alternative-composition operator $\_ + \_$. The process $p + q$ represents the nondeterministic choice between the processes p and q. The choice is resolved by the execution of an action or an occurrence of a termination.
- for each $t \in \mathrm{Time}$, a time-initialisation operator $t \gg \_$. The process $t \gg p$ is p limited to those alternatives that execute their first action not before time t.

Terms can be constructed using variables and the elements from the signature. Closed terms are terms in which no variables occur. We decide to allow the execution of more than one action at the same moment of time (in some order). There are no fundamental reasons for this choice: we could equally well have adopted the choice to disallow such *urgent* actions.

Next, we provide a structured operational semantics for the closed terms from this process algebra. We define the following transition relations and predicates:

[1] Note that in the original semantics of timed $\mu\mathrm{CRL}$ [Gro97], a two-phase model is used with states consisting of a closed process term and a moment in time, and separate action transitions $\xrightarrow{a}$ and a time transition $\xrightarrow{\iota}$.

[2] It does not matter for the treatment whether this time domain is discrete or dense.

– a time-stamped action-transition relation $\_ \xrightarrow{a}_t \_$ (with $a \in \mathrm{Act}$ and $t \in \mathrm{Time}$), representing the execution of an action a at time t.
– a time-stamped termination predicate $\_ \downarrow_t$ (with $t \in \mathrm{Time}$), representing successful termination at time t.
– a time-parameterised delay predicate $\_ \rightsquigarrow_t$ (with $t \in \mathrm{Time}$), representing that a process can delay until at least time t.

The reason for including the delay predicate is to discriminate between differently timed deadlocks: we have $0^{@3} \rightsquigarrow_3$, whereas $0^{@2} \not\rightsquigarrow_3$. These transition relations and predicate are defined by means of a so-called term deduction system [AFV01]. The deduction rules are presented in Table 1. In this table and others in the rest of this paper, x, x', y, and y' are variables representing arbitrary process terms, $a \in \mathrm{Act}$ is an action name, $I \subseteq \mathrm{Act}$, and $t, u \in \mathrm{Time}$.

$$\frac{}{0^{@t} \rightsquigarrow_u}\,[u \le t] \qquad \frac{}{1^{@t} \downarrow_t} \qquad \frac{}{1^{@t} \rightsquigarrow_u}\,[u \le t] \qquad \frac{}{a^{@t}.x \xrightarrow{a}_t t \gg x}$$

$$\frac{}{a^{@t}.x \rightsquigarrow_u}\,[u \le t] \qquad \frac{x \xrightarrow{a}_t x'}{\begin{array}{c}x + y \xrightarrow{a}_t x' \\ y + x \xrightarrow{a}_t x'\end{array}} \qquad \frac{x \downarrow_t}{\begin{array}{c}x + y \downarrow_t \\ y + x \downarrow_t\end{array}} \qquad \frac{x \rightsquigarrow_t}{\begin{array}{c}x + y \rightsquigarrow_t \\ y + x \rightsquigarrow_t\end{array}}$$

$$\frac{x \xrightarrow{a}_u x'}{t \gg x \xrightarrow{a}_u x'}\,[t \le u] \qquad \frac{x \downarrow_u}{t \gg x \downarrow_u}\,[t \le u] \qquad \frac{}{t \gg x \rightsquigarrow_u}\,[u \le t] \qquad \frac{x \rightsquigarrow_u}{t \gg x \rightsquigarrow_u}$$

Table 1. Structured Operational Semantics of $\mathrm{BSP}^{@}_{\mathrm{abs}}$.

Strong timed bisimilarity (as defined in [RGvdZvW02], for example) is a congruence for all operators from this process algebra. One can quite easily obtain a sound and complete axiomatisation of strong bisimilarity. The details are omitted as they are of no importance to the goal of this paper.

3 Abstraction and the Timed Silent Step

In order to facilitate abstraction of actions, usually a special atomic action $\tau \notin \mathrm{Act}$ is assumed that represents an *internal action* or *silent step*. Also, an abstraction operator τ_I (for $I \subseteq \mathrm{Act}$) is used for specifying which actions need to be considered internal. This leads to the following extensions to the signature of the process algebra:

– for each $t \in \mathrm{Time}$, a silent step prefix operator $\tau^{@t}.\_$. The process $\tau^{@t}.p$ represents the process that idles upto time t, executes silent step τ at that time and after that behaves as process p insofar time allows.
– for each $I \subseteq \mathrm{Act}$, an abstraction operator τ_I. The process $\tau_I(p)$ represents process p in which all actions from the set I are made invisible (i.e., replaced by silent step τ).

To express execution of a silent step at a certain time t the predicate $\_ \xrightarrow{\tau}_t \_$ is used. The silent step prefix operator has precisely the same deduction rules as the action prefix operator (with a replaced by τ). The deduction rules for the abstraction operator are given below.

$$\frac{x \xrightarrow{a}_t x'}{\tau_I(x) \xrightarrow{a}_t \tau_I(x')}\,[a \notin I] \qquad \frac{x \xrightarrow{a}_t x'}{\tau_I(x) \xrightarrow{\tau}_t \tau_I(x')}\,[a \in I]$$

$$\frac{x \xrightarrow{\tau}_t x'}{\tau_I(x) \xrightarrow{\tau}_t \tau_I(x')} \qquad \frac{x \downarrow_t}{\tau_I(x) \downarrow_t} \qquad \frac{x \rightsquigarrow_t}{\tau_I(x) \rightsquigarrow_t}$$

Again, congruence of timed strong bisimilarity is obvious and obtaining a sound and complete axiomatisation of timed strong bisimilarity is not difficult either.

Timed Rooted Branching Bisimilarity In the rest of this section, we discuss several timed versions of the well-known notion of rooted branching bisimilarity [GW89,GW96] that appeared in the literature. We refer to the relevant literature for definitions of these notions. We only present some characteristic equalities and inequalities between processes to illustrate the notions.

In [Klu93, Chapter 6], Klusener defines notions of rooted timed branching bisimilarity for a timed process algebra in a setting that does not allow for consecutive actions at the same moment in time, i.e., non-urgent actions. Two semantics and equivalences are defined, both in a setting with time-stamped action transitions. The first semantics, the so-called *idle* semantics employs idle transitions to model time passing. The second, called the *term* semantics, uses an ultimate delay predicate instead. Characteristic for the equivalences is that an action transition a at time t in one process may be mimicked in another process by a well-timed sequence (i.e., a sequence in which the timing of the subsequent actions does not decrease) of silent steps that is ultimately followed by an a-transition at time t. The intermediate states need to be related with the original state (at the right moment in time). Klusener shows that in his setting these two semantics and equivalences coincide. In almost the same setting[3], using the term semantics, Fokkink proves a completeness result for the algebra of regular processes [Fok94,Fok97]. By means of the following examples we will discuss the equivalences of Klusener.

Example 1 (No-Choice Silent Step). The three processes $\tau_{\{b\}}(a^{@1}.b^{@2}.c^{@4}.0^{@5})$, $\tau_{\{b\}}(a^{@1}.b^{@3}.c^{@4}.0^{@5})$ and $a^{@1}.c^{@4}.0^{@5}$ are obviously considered equal. Thus, the timing of the action that is hidden is of no importance insofar it does not disallow other actions from occurring (due to ill-timedness).

Example 2 (Time-Observed Silent Step). The processes $\tau_{\{b\}}(a^{@1}.(b^{@2}.(c^{@3}.0^{@4} + d^{@3}.0^{@4}) + d^{@3}.0^{@4}))$ and $a^{@1}.(c^{@3}.0^{@4} + d^{@3}.0^{@4})$ are distinguished by the notion of rooted timed branching bisimilarity from [Klu93, Chapter 6]. The reason is that in the first process at time 2 it may be determined that the d will be executed at time 3, while in the latter process term the choice between the c and the d at 3 can not be done earlier than at time 3.

Example 3 (Swapping). The processes $\tau_{\{b\}}(a^{@1}.(b^{@2}.c^{@3}.0^{@4} + d^{@3}.0^{@4}))$ and $\tau_{\{b\}}(a^{@1}.(c^{@3}.0^{@4} + b^{@2}.d^{@3}.0^{@4}))$ are considered equal with respect to Klusener's notion of equality, since in both processes it is decided at time 2 whether the c or the d will be executed at time 3.

It is interesting to note that, if one considers Klusener's definition of timed rooted idle branching bisimilarity in a setting in which urgent actions are allowed, the swapping of silent steps as portrayed in this example does not hold anymore. With timed rooted branching bisimilarity as defined for the term semantics though, it remains valid.

Example 4 (Time-Choice). According to [Klu93] the processes $\tau_{\{b\}}(a^{@1}.(b^{@3}.0^{@4} + c^{@2}.0^{@4}))$ and $a^{@1}.(0^{@4} + c^{@2}.0^{@4})$ are equal, since the passage of time already decides at time point 2 whether or not the alternative $c^{@2}.0^{@4}$ occurs or not.

Baeten and Bergstra introduce the silent step to relative time, absolute time and parametric time (i.e., a mixture of both relative and absolute time) versions of ACP with discrete time in [BB95]. A difference

[3] Fokkink uses a relative-time syntax and semantics and defines the ultimate delay predicate slightly different.

with the work of Klusener is that time steps are explicit in the syntax in [BB95]. In [BBR00], a complete axiomatisation for timed rooted branching bisimilarity is provided, for a variant of this theory. With respect to the four examples presented before, the only difference between Klusener's notion and Baeten and Bergstra's notion is that the latter does *not* consider the processes from Example 3 (Swapping) equal.

In [Zwa01], Van der Zwaag defines a notion of timed branching bisimilarity for a process algebra that has almost the same syntax and semantics as ours. In the setting studied by Van der Zwaag there is no successful termination. In [FPW05], Fokkink et al show that the notion of timed branching bisimilarity as put forward by Van der Zwaag is not an equivalence for dense time domains and therefore present a stronger notion of timed branching bisimilarity that is an equivalence indeed. Also, the definitions are extended to include successful termination. These notions of timed rooted branching bisimilarity are similar to that of Baeten and Bergstra for the examples presented before.

The way in which abstraction of actions leads to very precisely timed silent steps can be considered problematic (from a practical point of view). This was also recognised by Baeten, Middelburg and Reniers in [BMR02] in the context of a relative-time discrete-time process algebra with two-phase time specifications. The equivalences as described above are not coarse enough in practical cases such as the PAR protocol. An attempt is made to establish a coarser equivalence (called abstract branching bisimilarity) that "treats an internal action always as redundant if it is followed by a process that is only capable of idling till the next time slice." This leads to an axiom (named DRTB4) of the form $\tau_{\{a\}}(a^{@t}.x) = \tau_{\{a\}}(t \gg x)$ (in a different syntax).

Although we support the observation of the authors from [BMR02] that a coarser notion of equivalence is needed, we have several problems with the treatment of this issue in [BMR02]. The first is that the authors have sincere problems in defining the equivalence on the structured operational semantics. This problem is ultimately solved by using the (standard) definition of rooted branching (tail) bisimilarity from [BBR00] in combination with a structured operational semantics that is a silent-step-saturated version of the original semantics. Second, the axioms for standard operators such as parallel composition need to be adapted in a non-trivial way.

4 Untimed Silent Step

In this section, we present a novel abstraction mechanism in timed process algebra that is inspired by the opinion that *the timing of a silent step as such is not observable*. Therefore, one might consider defining an abstraction operator that abstracts from an action *and* from its timing. One should be careful though, that abstraction from the timing of action a may not result in an abstraction of the consequences of this timing of a for the rest of the process!

In the next section, we formally present our novel approach to action abstraction in timed process algebras. First we give the consequences of our intuition about the equality (called timed rooted branching bisimilarity, denoted by $\leftrightarrow_{\mathrm{rb}}$, see Section 4.2 for a definition) of the example processes from the previous section.

The timing of the action that is hidden is of no importance insofar it does not disallow other actions from occurring (due to ill-timedness). Therefore, the processes from Example 1 (No-Choice) should be considered equal:

$$\tau_{\{b\}}(a^{@1}.b^{@2}.c^{@4}.0^{@5}) \quad \leftrightarrow_{\mathrm{rb}} \quad \tau_{\{b\}}(a^{@1}.b^{@3}.c^{@4}.0^{@5}) \quad \leftrightarrow_{\mathrm{rb}} \quad a^{@1}.c^{@4}.0^{@5}$$

The processes from Example 2 (Time-Observed Silent Step) are equal in our setting since we do not wish to consider the timing of the internal step relevant:

$$\tau_{\{b\}}(a^{@1}.(b^{@2}.(c^{@3}.0^{@4} + d^{@3}.0^{@4}) + d^{@3}.0^{@4})) \quad \leftrightarrow_{\mathrm{rb}} \quad a^{@1}.(c^{@3}.0^{@4} + d^{@3}.0^{@4})$$

The processes from Example 3 (Swapping) are different processes, since by executing the silent step, an option that was there before has disappeared:

$$\tau_{\{b\}}(a^{@1}.(b^{@2}.c^{@3}.0^{@4} + d^{@3}.0^{@4})) \quad \not\leftrightarrow_{\text{rb}} \quad \tau_{\{b\}}(a^{@1}.(c^{@3}.0^{@4} + b^{@2}.d^{@3}.0^{@4}))$$

Since we do not allow to take the timing of the abstracted action into account, we cannot have the equality of the processes from Example 4 (Time-Choice):

$$\tau_{\{b\}}(a^{@1}.(b^{@3}.0^{@4} + c^{@2}.0^{@4})) \quad \not\leftrightarrow_{\text{rb}} \quad a^{@1}.(0^{@4} + c^{@2}.0^{@4})$$

In contrast with the other equivalences discussed in this paper, the process $\tau_{\{b\}}(a^{@1}.(b^{@3}.0^{@4} + c^{@2}.0^{@4}))$ can only be 'simplified' to $a^{@1}.(\tau.0^{@4} + c^{@2}.0^{@4})$. Thus the silent step remains.

In our opinion, in [BMR02] too many silent steps can be omitted. Consider for example the process $\tau_{\{a\}}(a^{@1}.0^{@2} + b^{@3}.0^{@4})$. In [BMR02], it is considered to be equal to $b^{@3}.0^{@4}$. In our opinion, the execution of the internal step disables the execution of action b altogether.

4.1 Abstraction using the Untimed Silent Step

We propose to extend the process algebra from Section 2 with the following primitives instead of the timed silent action prefix operators and abstraction operator from Section 3:

- the silent step prefix operator $\tau.\_$. The process $\tau.p$ performs an internal action (not at any specific time) and thereafter behaves as p.
- for each $I \subseteq \text{Act}$, the abstraction operator τ_I. The process $\tau_I(p)$ represents process p where all actions from the set I are made invisible (replaced by the untimed silent step τ). It should be noted that the consequences of the timing of the abstracted action are not abstracted from.

In the structured operational semantics, we add a relation $\_ \xrightarrow{\tau} \_$ that represents the execution of an untimed silent step. For alternative composition and time initialisation we add deduction rules for this new transition relation (the first two deduction rules in Table 2). In the second deduction rule for the abstraction operator

$$\frac{x \xrightarrow{\tau} x'}{x + y \xrightarrow{\tau} x' \quad y + x \xrightarrow{\tau} x'} \qquad \frac{x \xrightarrow{\tau} x'}{t \gg x \xrightarrow{\tau} t \gg x'} \qquad \frac{}{\tau.x \xrightarrow{\tau} x}$$

$$\frac{x \xrightarrow{a}_t x'}{\tau_I(x) \xrightarrow{a}_t \tau_I(x')}\ [a \notin I] \qquad \frac{x \xrightarrow{a}_t x'}{\tau_I(x) \xrightarrow{\tau} \tau_I(x')}\ [a \in I] \qquad \frac{x \xrightarrow{\tau} x'}{\tau_I(x) \xrightarrow{\tau} \tau_I(x')}$$

$$\frac{x \downarrow_t}{\tau_I(x) \downarrow_t} \qquad \frac{x \rightsquigarrow_t}{\tau_I(x) \rightsquigarrow_t}$$

Table 2. Structured Operational Semantics of untimed silent step and abstraction operator.

one can see that a timed action is replaced by an untimed silent step in case the action is to be abstracted from. Also note that the consequences of the timing of the action are imposed on the rest of the process by means of the time-initialisation operator in the deduction rule for action-transitions of the action prefix operator (in Table 1). This means that the process x' incorporates the fact that time t has been reached.

Example 5. Somewhat surprisingly, the process $p = a^{@2}.\tau_{\{b\}}(b^{@1}.0^{@4})$ is not ill-timed. This is a consequence of our decision that the timing of abstracted actions is not observable. Thus the process p is equal to $a^{@2}.0^{@4}$ and of course also to $a^{@2}.\tau_{\{b\}}(b^{@3}.0^{@4})$ (which can hardly be considered ill-timed). It should be noted that in a relative time setting this phenomenon of ill-timedness does not occur.

4.2 Timed Rooted Branching Bisimilarity

In the following definition we use the notation $p \Rightarrow q$ to denote that q can be reached from p by executing an arbitrary number (possibly zero) of τ-transitions. The notation $p \overset{(\tau)}{\to} q$ means $p \overset{\tau}{\to} q$ or $p = q$.

Definition 1 (Timed Rooted Branching Bisimilarity). *Two closed terms p and q are* timed branching bisimilar, *notation $p \leftrightarrow_{\mathrm{b}} q$, if there exists a symmetric binary relation R on closed terms, called a* timed branching bisimulation *relation, relating p and q such that for all closed terms r and s with $(r, s) \in R$:*

1. *if $r \overset{a}{\to}_t r'$ for some $a \in \mathrm{Act}$, $t \in \mathrm{Time}$, and closed term r', then there exist closed terms s^* and s' such that $s \Rightarrow s^* \overset{a}{\to}_t s'$, $(r, s^*) \in R$ and $(r', s') \in R$;*
2. *if $r \overset{\tau}{\to} r'$ for some closed term r', then there exist closed terms s^* and s' such that $s \Rightarrow s^* \overset{(\tau)}{\to} s'$, $(r, s^*) \in R$ and $(r', s') \in R$;*
3. *if $r \downarrow_t$ for some $t \in \mathrm{Time}$, then there exists a closed term s^* such that $s \Rightarrow s^* \downarrow_t$ and $(r, s^*) \in R$;*
4. *if $r \rightsquigarrow_t$ for some $t \in \mathrm{Time}$, then there exists a closed term s^* such that $s \Rightarrow s^* \rightsquigarrow_t$ and $(r, s^*) \in R$.*

If R is a timed branching bisimulation relation, we say that the pair (p, q) satisfies the root condition *with respect to R if*

1. *if $p \overset{a}{\to}_t p'$ for some $a \in \mathrm{Act}$, $t \in \mathrm{Time}$, and closed term p', then there exists a closed term q' such that $q \overset{a}{\to}_t q'$ and $(p', q') \in R$;*
2. *if $p \overset{\tau}{\to} p'$ for some closed term p', then there exists a closed term q' such that $q \overset{\tau}{\to} q'$ and $(p', q') \in R$;*
3. *if $p \downarrow_t$ for some $t \in \mathrm{Time}$, then $q \downarrow_t$.*
4. *if $p \rightsquigarrow_t$ for some $t \in \mathrm{Time}$, then $q \rightsquigarrow_t$.*

Two closed terms p and q are called timed rooted branching bisimilar, *notation $p \leftrightarrow_{\mathrm{rb}} q$, if there is a timed branching bisimulation relation R relating p and q such that the pairs (p, q) and (q, p) satisfy the root condition with respect to R.*

4.3 Properties of Timed Rooted Branching Bisimilarity

In this section, we show that timed rooted branching bisimilarity as defined in the previous section is indeed an equivalence. Moreover we show that it is a congruence for the rather restricted set of operators introduced.

Theorem 1. *Timed rooted branching bisimilarity is an equivalence relation.*

Proof. A proof that timed rooted branching bisimilarity is an equivalence relation is given in Appendix A.

Theorem 2. *Timed bisimilarity and timed rooted branching bisimilarity are congruences for all operators from the signature of the process algebra $\mathrm{BSP}^{@}_{\mathrm{abs}}$.*

Proof. The deduction rules are in the path format, thus congruence of timed bisimilarity follows from the meta-theory in [BV93]. See [AFV01] for an overview of the type of meta-theory used here. A proof of congruence of timed rooted branching bisimilarity is given in Appendix B.

Furthermore, obviously timed rooted branching bisimilarity identifies strictly more process than timed strong bisimilarity does.

Theorem 3. *Timed strongly bisimilar processes are timed rooted branching bisimilar: i.e.,* $\underline{\leftrightarrow} \subseteq \underline{\leftrightarrow}_{\mathrm{rb}}$.

Proof. Trivial.

From the examples presented in the previous sections, we can easily conclude that our notion of equality is incomparable with the notions from Klusener [Klu93], Baeten and Bergstra [BB95], and Van der Zwaag [Zwa01]. We claim that the notion of abstract branching bisimilarity from [BMR02] is coarser than ours.

5 Case Study: PAR Protocol

5.1 Specification of the PAR Protocol

In [BMR02] the Positive Acknowledgement Retransmission protocol is used to illustrate the need for a coarser equivalence. In this paper, we will use the same protocol for illustrating the problem and (later on) the solution. The following informal description of the protocol is almost taken verbatim from [BMR02].

In the PAR protocol, the sender (S) waits for a positive acknowledgement before a new datum is transmitted. If an acknowledgement is not received within a complete protocol cycle, the old datum is retransmitted. In order to avoid duplicates due to retransmission, data are labeled with an alternating bit from $B = \{0, 1\}$. The configuration of the PAR protocol is given in Figure 1 by means of a connection diagram. The protocol entities are a sender S, a receiver R, a forward channel K, and a backward channel L.

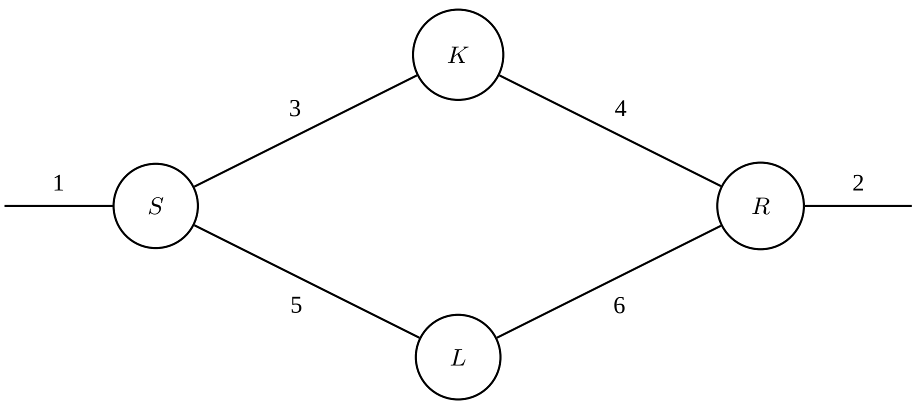

Fig. 1. Connection diagram for the PAR protocol. The numbers on the connection lines indicate the port numbers.

The process S waits until a datum d is offered at an external port (port 1). Then S packs it with an alternating bit b in a frame (d, b) , and then delivers it at the port used for sending (port 3). It is assumed that this a constant amount of time t_S. Then S waits until an acknowledgement ack is offered at port 5. When the acknowledgement does not arrive within a certain time period (t'_S), S repeats the delivery of the frame and again waits for an acknowledgement. When the acknowledgement arrives in time, S starts waiting for a datum again.

The receiver R waits until a frame (d, b) is offered at port 4. It unpacks it, delivers the datum at external port 2 in case the bit b is the right one (this takes t_R time), and offers an acknowledgement at internal port 6 (t'_R time). Then the receiver goes back to waiting for a frame. In case the right bit was received the alternating bit is flipped.

The channels K and L pass frames and acknowledgements, respectively. These channels are supposed to be unreliable, they may produce an error instead of passing on data. The channels require t_K and t_L time for passing the data, respectively.

In this paper, we do not present specifications for the separate protocol entities. We immediately give the expanded version where all occurrences of parallel composition operators are removed. Note that we have used notations such as $\sum_{t'} p$ that describe a potentially infinite alternative composition consisting of one alternative of p for each t'. We refrain from giving operational semantics for this operator, called summation [RGvdZvW02] or alternative quantification [Lut02].

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= c_3(d,b)^{@t}.\left(\begin{array}{l} c_4(d,b)^{@t+t_K}.s_2(d)^{@t+t_K+t_R}.c_6(ack)^{@t+t_K+t_R+t'_R}.Z_{d,b,t+t_K+t_R+t'_R} \\ + \sum_{k \le t_K} error^{@t+k}.Y_{d,b,t+t'_S} \end{array} \right) \\
Z_{d,b,t} &= c_5(ack)^{@t+t_L}.X_{\bar{b},t+t_L} + \sum_{l \le t_L} error^{@t+l}.U_{d,b,t+t'_S-t_K-t_R-t'_R} \\
U_{d,b,t} &= c_3(d,b)^{@t}.\left(\begin{array}{l} c_4(d,b)^{@t+t_K}.c_6(ack)^{@t+t_K+t'_R}.V_{d,b,t+t_K+t'_R} \\ + \sum_{k \le t_K} error^{@t+k}.U_{d,b,t+t'_S} \end{array} \right) \\
V_{d,b,t} &= c_5(ack)^{@t+t_L}.X_{\bar{b},t+t_L} + \sum_{l \le t_L} error^{@t+l}.U_{d,b,t+t'_S-t_K-t'_R}
\end{aligned}
$$

We obtained the above expanded version of the PAR protocol by carefully translating the version of [BMR02] to the process algebra that is used in this paper. We believe that we can also define a similar parallel composition operator and we believe that we can obtain the above result from expansion of parallel-composition operators in our setting too.

5.2 Abstraction of Internal Actions using Timed Silent Steps

The application of the abstraction operator from the previous section with

$$I = \{c_3(d,b), c_4(d,b) \mid d \in D, b \in B\} \cup \{c_5(ack), c_6(ack), error\}$$

to the expanded version of the PAR protocol from Section 5.1 results in the following recursive specification:

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= \tau^{@t}.\left(\begin{array}{l} \tau^{@t+t_K}.s_2(d)^{@t+t_K+t_R}.\tau^{@t+t_K+t_R+t'_R}.Z_{d,b,t+t_K+t_R+t'_R} \\ + \sum_{k \leq t_K} \tau^{@t+k}.Y_{d,b,t+t'_S} \end{array} \right) \\
Z_{d,b,t} &= \tau^{@t+t_L}.X_{\overline{b},t+t_L} + \sum_{l \leq t_L} \tau^{@t+l}.U_{d,b,t+t'_S-t_K-t_R-t'_R} \\
U_{d,b,t} &= \tau^{@t}.\left(\begin{array}{l} \tau^{@t+t_K}.\tau^{@t+t_K+t'_R}.V_{d,b,t+t_K+t'_R} \\ + \sum_{k \leq t_K} \tau^{@t+k}.U_{d,b,t+t'_S} \end{array} \right) \\
V_{d,b,t} &= \tau^{@t+t_L}.X_{\overline{b},t+t_L} + \sum_{l \leq t_L} \tau^{@t+l}.U_{d,b,t+t'_S-t_K-t'_R}
\end{aligned}
$$

With respect to timed rooted branching bisimilarity as defined in the previous section, this description can be simplified to the following:

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= \tau^{@t+t_K}.s_2(d)^{@t+t_K+t_R}.Z_{d,b,t+t_K+t_R+t'_R} + \sum_{k \leq t_K} \tau^{@t+k}.Y_{d,b,t+t'_S} \\
Z_{d,b,t} &= \tau^{@t+t_L}.X_{\overline{b},t+t_L} + \sum_{l \leq t_L} \tau^{@t+l}.U_{d,b,t+t'_S-t_K-t_R-t'_R} \\
U_{d,b,t} &= \tau^{@t+t_K}.V_{d,b,t+t_K+t'_R} + \sum_{k \leq t_K} \tau^{@t+k}.U_{d,b,t+t'_S} \\
V_{d,b,t} &= \tau^{@t+t_L}.X_{\overline{b},t+t_L} + \sum_{l \leq t_L} \tau^{@t+l}.U_{d,b,t+t'_S-t_K-t'_R}
\end{aligned}
$$

Again, this result is just obtained by translating the result from [BMR02] to our setting.

The silent steps that occur cannot be left out with respect to the standard notion of timed rooted branching bisimilarity. Thus the abstraction from the internal communication actions has not lead to a drastic simplification of the process.

5.3 Abstraction of Internal Actions using Untimed Silent Steps

The following process description is obtained after abstraction from the internal communications and actions from I.

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= \tau.(t+t_K) \gg s_2(d)^{@t+t_k+t_R}.Z_{d,b,t+t_K+t_R+t'_R} + \sum_{k \le t_K} \tau.(t+k) \gg Y_{d,b,t+t'_S} \\
Z_{d,b,t} &= \tau.(t+t_L) \gg X_{\bar{b},t+t_L} + \sum_{l \le t_L} \tau.(t+l) \gg U_{d,b,t+t'_S-t_K-t_R-t'_R} \\
U_{d,b,t} &= \tau.(t+t_K) \gg V_{d,b,t+t_K+t'_R} + \sum_{k \le t_K} \tau.(t+k) \gg U_{d,b,t+t'_S} \\
V_{d,b,t} &= \tau.(t+t_L) \gg X_{\bar{b},t+t_L} + \sum_{l \le t_L} \tau.(t+l) \gg U_{d,b,t+t'_S-t_K-t'_R}
\end{aligned}
$$

Observe that none of the processes execute an action before time t and that therefore all time initialisations disappear (using $t'_S > t_K + t_R + t'_R + t_L$). We thus get the following specification.

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= \tau.s_2(d)^{@t+t_k+t_R}.Z_{d,b,t+t_K+t_R+t'_R} + \tau.Y_{d,b,t+t'_S} \\
Z_{d,b,t} &= \tau.X_{\bar{b},t+t_L} + \tau.U_{d,b,t+t'_S-t_K-t_R-t'_R} \\
U_{d,b,t} &= \tau.V_{d,b,t+t_K+t'_R} + \tau.U_{d,b,t+t'_S} \\
V_{d,b,t} &= \tau.X_{\bar{b},t+t_L} + \tau.U_{d,b,t+t'_S-t_K-t'_R}
\end{aligned}
$$

This way many simplifications have already been achieved and allows us to simplify the specification even more. We can, for example, eliminate V by substitution in U. This gives us the following equation for U:

$$
U_{d,b,t} = \tau.(\tau.X_{\bar{b},t+t_K+t'_R+t_L} + \tau.U_{d,b,t+t'_S}) + \tau.U_{d,b,t+t'_S}
$$

Note the pattern of axiom (B). With (B) we get:

$$
U_{d,b,t} = \tau.X_{\bar{b},t+t_K+t'_R+t_L} + \tau.U_{d,b,t+t'_S}
$$

If we also move the $t_K + t_R + t'_R$ in Y to Z, we get the following specification.

$$
\begin{aligned}
X_{b,t} &= \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S} \\
Y_{d,b,t} &= \tau.s_2(d)^{@t+t_k+t_R}.Z'_{d,b,t} + \tau.Y_{d,b,t+t'_S} \\
Z'_{d,b,t} &= \tau.X_{\bar{b},t+t_K+t_R+t'_R+t_L} + \tau.U_{d,b,t+t'_S} \\
U_{d,b,t} &= \tau.X_{\bar{b},t+t_K+t'_R+t_L} + \tau.U_{d,b,t+t'_S}
\end{aligned}
$$

Note the strong similarity of Z' and U. By adding an extra parameter to U we get the final specification.

$$X_{b,t} = \sum_{t'} \sum_{d \in D} r_1(d)^{@t+t'}.Y_{d,b,t+t'+t_S}$$

$$Y_{d,b,t} = \tau.s_2(d)^{@t+t_k+t_R}.U'_{d,b,t,t_R} + \tau.Y_{d,b,t+t'_S}$$

$$U'_{d,b,t,u} = \tau.X_{\bar{b},t+t_K+u+t'_R+t_L} + \tau.U'_{d,b,t+t'_S,0}$$

The silent steps that are left are essential. The silent steps in Y determine whether or not an error occurred in channel K, and those in U' determine the same for channel L. As these errors result in an additional delay before the next action occurs, they are not redundant.

6 Axioms for Timed Rooted Branching Bisimilarity

In Table 3 we present axioms for timed strong bisimilarity. The axioms (A1)-(A3) express some standard properties of alternative composition. Axiom (WT) (for well-timedness) describes that the time that is reached by executing an action is passed on to the subsequent process. The axioms (A6a)-(A6d) describe the properties of timed deadlocks, especially the circumstances under which they can be removed from the process description. An important equality that can be derived for closed terms p is $p + 0^{@0} = p$.

Axioms (I1)-(I7) describe how the time-initialisation operator can be eliminated from terms. Note that the silent step neglects this operator (axiom (I6)). Axioms (H1)-(H6) describe how the abstraction operator can be eliminated. Note that the timing of an action that is abstracted from is passed on to the rest of the process (axiom (H4)). The time-initialisation operator in the right-hand side of axiom (H3) is needed in order to enforce the timing restriction from the action prefix before applying further abstractions.

(A1)	$x + y = y + x$	(A6a)	$0^{@t} + 0^{@u} = 0^{@\max(t,u)}$
(A2)	$(x + y) + z = x + (y + z)$	(A6b)	$u \le t \Rightarrow 1^{@t} + 0^{@u} = 1^{@t}$
(A3)	$x + x = x$	(A6c)	$u \le t \Rightarrow a^{@t}.x + 0^{@u} = a^{@t}.x$
(WT)	$a^{@t}.x = a^{@t}.t \gg x$	(A6d)	$u \le t \Rightarrow \tau.(x + 0^{@t}) + 0^{@u} = \tau.(x + 0^{@t})$
(I1)	$t \gg 0^{@u} = 0^{@\max(t,u)}$	(H1)	$\tau_I(0^{@t}) = 0^{@t}$
(I2)	$u < t \Rightarrow t \gg 1^{@u} = 0^{@t}$	(H2)	$\tau_I(1^{@t}) = 1^{@t}$
(I3)	$u \ge t \Rightarrow t \gg 1^{@u} = 1^{@u}$		
(I4)	$u < t \Rightarrow t \gg a^{@u}.x = 0^{@t}$	(H3)	$a \notin I \Rightarrow \tau_I(a^{@t}.x) = a^{@t}.\tau_I(t \gg x)$
(I5)	$u \ge t \Rightarrow t \gg a^{@u}.x = a^{@u}.x$	(H4)	$a \in I \Rightarrow \tau_I(a^{@t}.x) = \tau.\tau_I(t \gg x)$
(I6)	$t \gg \tau.x = \tau.t \gg x$	(H5)	$\tau_I(\tau.x) = \tau.\tau_I(x)$
(I7)	$t \gg (x + y) = t \gg x + t \gg y$	(H6)	$\tau_I(x + y) = \tau_I(x) + \tau_I(y)$

Table 3. Axioms for timed strong bisimilarity and timed rooted branching bisimilarity.

We claim that the axioms from Table 3 are sound and complete for timed strong bisimilarity on closed terms. These axioms are (of course; see Theorem 3) also valid for timed rooted branching bisimilarity. In Table 4, one additional axiom is presented for timed rooted branching bisimilarity. The reader should notice that this axiom resembles the untimed axiom for rooted branching bisimilarity $a.(\tau.(x + y) + x) = a.(x + y)$ meticulously. Also, it is expected that the axioms from both tables provide a sound and complete axiomatisation of timed rooted branching bisimilarity on closed terms.

Example 6. Consider the process term $p = \tau_{\{b\}}(a^{@1}.(b^{@3}.(c^{@2} + d^{@4}) + c^{@2}))$. Using the axioms for the abstraction operator and for time initialisation we have $p = a^{@1}.(\tau.3 \gg (c^{@2} + d^{@4}) + c^{@2}) =$

(B) $a^{@t}.(\tau.(x+y)+x) = a^{@t}.(x+y)$

Table 4. Axiom for timed rooted branching bisimilarity.

$a^{@1}.(\tau.d^{@4}+c^{@2})$. Observe that although the action b is abstracted from, its timing properties are propagated. This example explains the occurrence of the time-initialisation operator in axiom (H4). Without the occurrence of the time-initialisation operator in axiom (H4) the following would be valid derivations: $p = a^{@1}.(\tau.(c^{@2}+d^{@4})+c^{@2}) = a^{@1}.(c^{@2}+d^{@4})$ and $p = a^{@1}.(\tau.d^{@4}+c^{@2})$. The resulting equality of $a^{@1}.(c^{@2}+d^{@4})$ and $p = a^{@1}.(\tau.d^{@4}+c^{@2})$ is completely counterintuitive.

A similar example can be constructed to show that for the axioms (H3) and (H4) the time initialisation must indeed be placed inside the abstraction operator and not outside.

7 Extensions of the Timed Process Algebra

In order to illustrate that our restriction to the very limited set of operators is not inspired by fundamental limitations, in this section we extend the timed process algebra with some operators that are frequently encountered in timed process algebra in the ACP community.

7.1 Sequential Composition

We propose the following deduction rules for the binary sequential-composition operator. Compared to such deduction rules in a setting with timed silent step we only need to have separate deduction rules for the untimed silent step. Also these deduction rules are comparable to those of sequential composition in μCRL and mCRL2 if one neglects the fact that these languages have a different termination predicate.

$$\frac{x \xrightarrow{a}_t x'}{x \cdot y \xrightarrow{a}_t x' \cdot y} \qquad \frac{x \downarrow_u \quad u \gg y \xrightarrow{a}_t y'}{x \cdot y \xrightarrow{a}_t y'} \qquad \frac{x \xrightarrow{\tau} x'}{x \cdot y \xrightarrow{\tau} x' \cdot y}$$

$$\frac{x \downarrow_u \quad u \gg y \xrightarrow{\tau} y'}{x \cdot y \xrightarrow{\tau} y'} \qquad \frac{x \downarrow_u \quad u \gg y \downarrow_t}{x \cdot y \downarrow_t} \qquad \frac{x \leadsto_t}{x \cdot y \leadsto_t} \qquad \frac{x \downarrow_u \quad u \gg y \leadsto_t}{x \cdot y \leadsto_t}$$

Timed strong bisimilarity and timed rooted branching bisimilarity are congruences for sequential composition.

Theorem 4 (Congruence). *Timed bisimilarity and timed rooted branching bisimilarity are congruences for sequential composition.*

Proof. The deduction rules are in the path format, thus congruence of timed bisimilarity follows from the meta-theory in [BV93]. A proof of congruence of timed rooted branching bisimilarity is given in Appendix B.

In the section on future work in [FPW05], the authors mention that it is an interesting question whether a timed rooted branching bisimilarity is a congruence for a simple timed basic process algebra such as Baeten and Bergstra's BPA$_{\rho\delta}$ [BB91] (note that this theory is called BPA$^{ur}_{\rho\delta}$ in [FPW05]) which features time-stamped urgent actions. Although we did not address precisely this theory, for $\mathrm{BSP}^{@}_{\mathrm{abs}}$ extended with sequential composition we have established congruence of timed rooted branching bisimilarity (with untimed silent steps).

Below we present axioms for sequential composition that express the basic properties of sequential composition such as distributivity over alternative composition (axiom (A4)) and associativity (axiom (A5)) and at the same time allows for the elimination of sequential composition from every closed process term (axioms (A5a), (A5b), (A7), and (A8)).

(A4) $(x+y)\cdot z = x\cdot z + y\cdot z$	(A5a) $a^{@t}.x\cdot y = a^{@t}.(x\cdot y)$	(A7) $0^{@t}\cdot x = 0^{@t}$
(A5) $(x\cdot y)\cdot z = x\cdot(y\cdot z)$	(A5b) $\tau.x\cdot y = \tau.(x\cdot y)$	(A8) $1^{@t}\cdot x = t \gg x$

It can be shown that these axioms are sound for timed strong bisimilarity (and hence also for timed rooted branching bisimilarity), that this extension with sequential composition is a conservative ground-extension (see [BMR05] for a definition), and that sequential composition can be eliminated from closed terms.

In, amongst others, timed μCRL, the equality $\tau_I(x\cdot y) = \tau_I(x)\cdot\tau_I(y)$ is valid for closed terms. As a consequence of the introduction of the untimed silent step and its corresponding abstraction operator this equality does not hold any longer. This can be seen by considering the processes $p = a^{@2}.1^{@2}$ and $q = b^{@1}.1^{@3}$. Let $I = \{b\}$. Now, the process $\tau_I(p\cdot q)$ is timed rooted branching bisimilar to the process $a^{@2}.0^{@2}$ whereas the process $\tau_I(p)\cdot\tau_I(q)$ is timed rooted branching bisimilar to $a^{@2}.1^{@3}$, which are obviously not timed rooted branching bisimilar.

7.2 Parallel Composition

Next we consider a parallel composition operator without communication. There are many different parallel composition operators in the literature, especially with respect to termination and time-synchronisation options. Here, we choose to mimic the termination and time-synchronisation options of the parallel composition operator from [BR]: both termination and time-progress are fully synchronised between the components.

$$\frac{x \xrightarrow{a}_t x' \quad y \rightsquigarrow_t}{\begin{array}{c} x \parallel y \xrightarrow{a}_t x' \parallel t \gg y \\ y \parallel x \xrightarrow{a}_t t \gg y \parallel x' \end{array}} \qquad \frac{x \xrightarrow{\tau} x'}{\begin{array}{c} x \parallel y \xrightarrow{\tau} x' \parallel y \\ y \parallel x \xrightarrow{\tau} y \parallel x' \end{array}} \qquad \frac{x \downarrow_t \quad y \downarrow_t}{x \parallel y \downarrow_t} \qquad \frac{x \rightsquigarrow_t \quad y \rightsquigarrow_t}{x \parallel y \rightsquigarrow_t}$$

Timed strong bisimilarity and timed rooted branching bisimilarity are congruences for sequential composition.

Theorem 5 (Congruence). *Timed bisimilarity and timed rooted branching bisimilarity are congruences for parallel composition.*

Proof. The deduction rules are in the path format, thus congruence of timed bisimilarity follows from the meta-theory in [BV93]. A proof of congruence of timed rooted branching bisimilarity is given in Appendix B.

As can be seen from the above deduction rules, the introduction of the untimed silent step and the corresponding abstraction operator have no influence at all. Therefore, we claim that an axiomatisation can easily be given following [RGvdZvW02,BR].

Note that the equality $\tau_I(x \parallel y) = \tau_I(x) \parallel \tau_I(y)$ is not valid for closed terms. This can be seen as follows. Consider the processes $p = a^{@1}.0^{@3}$ and $q = b^{@2}.0^{@3} + c^{@2}.0^{@3}$ and the set of actions $I = \{b\}$ to be abstracted from. The composition $\tau_I(p \parallel q)$ reduces to the process $a^{@1}.(\tau.0^{@3} + c^{@2}.0^{@3})$. On the other hand $\tau_I(p) \parallel \tau_I(q)$ reduces to $a^{@1}.(\tau.0^{@3} + c^{@2}.0^{@3}) + \tau.a^{@1}.0^{@3}$. It is not hard to see that these are different.

8 Other Timed Process Algebra Settings

The process algebra that we have chosen as our universe of discourse can be classified (both syntactically and semantically) as an absolute-time time-stamped process algebra. As mentioned before, in the literature there are some other versions available, with respect to both the syntax used and the semantics adopted. In this section, we discuss, with respect to the semantics, how the abstraction technique presented here for an absolute-time time-stamped process algebra can be carried over to other types of timed process algebras and what problems are expected to arise in doing so.

In a setting where the time-stamping mechanism uses relative time the treatment becomes even simpler. In such a setting $a^{@t}.p$ means that a is to be executed t time after the execution of the previous action (or after the conception of the process). As a consequence of this relative-timing the problem of ill-timedness is avoided. Therefore, the time-initialisation operator can be left out. Instead, one needs to have a mechanism for updating the relative time-stamp of the initial actions of the subsequent process due to abstraction:

$$\frac{x \xrightarrow{a}_t x' \quad a \in I}{\tau_I(x) \xrightarrow{\tau} t \circledast \tau_I(x')}$$

where $t \circledast p$ means that t time has to be added to the time-stamp of the first visible action from p. For example $3 \circledast a^{@5}.p$ behaves as $a^{@8}.p$. An example of such an operator is the time shift operator (t)_ (also with negative t!) that has been used by Fokkink for defining timed branching bisimilarity in [Fok94].

We have chosen to carry out our deliberations in a time-stamped setting because this setting allows for a very natural definition of the abstraction operator since the timing of the action (before abstraction) and the action itself are tightly coupled in the model. To illustrate the difficulties that arise in defining abstraction in a two-phase model, we look at the following processes (in the syntax of [BMR05,BR]). Note that $\sigma.$_ is a time step prefix operator and $\underline{\underline{a}}.$_ is an immediate action prefix operator.

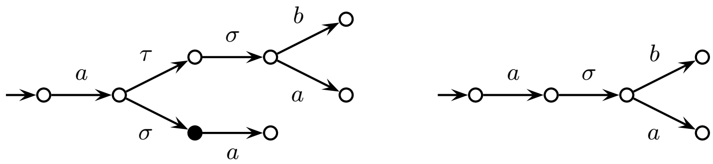

Fig. 2. Processes $\underline{\underline{a}}.(\sigma.\underline{\underline{a}}.\underline{\underline{0}} + \underline{\underline{\tau}}.\sigma.(\underline{\underline{a}}.\underline{\underline{0}} + \underline{\underline{b}}.\underline{\underline{0}}))$ and $\underline{\underline{a}}.\sigma.(\underline{\underline{a}}.\underline{\underline{0}} + \underline{\underline{b}}.\underline{\underline{0}})$

As we have discussed in Section 4, we consider these processes equivalent. However, to express this in an equivalence, we need to be able to relate the states of both processes. In the diagram above one can see that the first process can make a time transition that results in a state (the black one) that has no corresponding state in the second process. The essence of this problem is that one tries to relate states that are reached solely by time steps such as the black one. We thus believe the solution is to not relate such states, even if they exist.

9 Concluding Remarks

In this paper, we have introduced a notion of abstraction that abstracts from the identity of an action and its timing, resulting in an untimed silent step. We have developed an accompanying notion of equality of processes, also called timed rooted branching bisimilarity. We have shown that this notion is an equivalence relation and a congruence for all operators considered in this paper and as such may be a meaningful tool in analysing and verifying systems. A first experiment, on the PAR protocol, indicates that our notions allow for a much clearer and smaller representation of the abstract system than the standard notions do. An axiomatisation of timed rooted branching bisimilarity for closed process terms is given with an axiom for the removal of untimed silent steps that resembles the well-known axiom from untimed process algebra.

In case one does not accept our reasoning for adopting the untimed silent step, one can keep the abstraction operator and timed silent steps (since they are considered relevant) as usual and add an untimed silent step and an abstraction operator that only abstracts from the timing of the silent steps. This way one can control whether or not to use untimed silent step, for example depending on the properties that need to be validated.

In this paper, we have made many claims about the timed process algebra with untimed silent steps. Of course, these claims need to be substantiated further. Also, it is worthwhile to study our notion of abstraction in other timed settings, most notably those with relative timing and where timing is described by separate timing primitives (decoupled from actions) as in [BM02] and most other mainstream timed process algebras.

We have illustrated the differences and similarities between the different definitions of timed rooted branching bisimilarity from literature and our version by means of examples only. A more thorough comparison is needed. Also, a comparison with timed versions of weak bisimilarity (e.g., [MT92,Che93,QdFA93,HSZF93]) should be performed.

The success of an abstraction mechanism and notion of equality not depend only on the theoretical properties (though important) of these notions, but much more so on the practical suitability of these notions. Therefore, we need to perform more case studies to observe whether these notions contribute to a better/easier verification of correctness and/or properties of relevant systems.

We are, in line with our previous work ([BMR05,BR]), very interested in obtaining a collection of theories that are nicely related by means of conservativity results and embeddings. Therefore, it is interesting to extend the rather limited timed process algebra from this paper with untimed action prefix operators $a.\_$ in order to formally study, in one framework, the relationship between rooted branching bisimilarity on untimed processes and our timed version.

A complementary way of specifying a timed system is by means of a timed (modal) logic. It is worthwhile to get a deeper understanding of our notion of action abstraction and timed rooted branching bisimilarity by considering the relationship with modal logics for timed systems as has been done for strong bisimilarity [Par81] and Hennessy-Milner logic [HM85]. We have good hope that the majority of the logics that are used for the specification of properties of timed systems are preserved by our notion of timed rooted branching bisimilarity.

Acknowledgements We acknowledge useful comments from Jos Baeten, Pieter Cuijpers, Wan Fokkink, Jan Friso Groote, Bas Luttik, Bas Ploeger, Yaroslav Usenko, and Tim Willemse.

References

[AFV01] L. Aceto, W.J. Fokkink, and C. Verhoef. Structural operational semantics. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, chapter 3, pages 197–292. Elsevier, 2001.

[BB91] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. *Formal Aspects of Computing*, 3(2):142–188, 1991.

[BB95] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra with abstraction. In H. Reichel, editor, *FCT'95, International Conference on Fundamentals of Computation Theory*, volume 965 of *Lecture Notes in Computer Science*, pages 1–15, Dresden, 1995. Springer-Verlag.

[BBR00] J.C.M. Baeten, J.A. Bergstra, and M.A. Reniers. Discrete time process algebra with silent step. In G.D. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language, and Interaction: Essays in Honour of Robin Milner*, Foundations of Computing Series, chapter 18, pages 535–569. MIT Press, 2000.

[BBR07] J.C.M. Baeten, T. Basten, and M.A. Reniers. Process algebra: Equational theories of communicating processes. 2007.

[BK84] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1/3):109–137, 1984.

[BM02] J.C.M. Baeten and C.A. Middelburg. *Process Algebra with Timing*. EATCS Monographs. Springer-Verlag, 2002.

[BMR02] J.C.M. Baeten, C.A. Middelburg, and M.A. Reniers. A new equivalence for processes with timing: With an application to protocol verification. Technical Report CSR 02-10, Eindhoven University of Technology, Department of Computer Science, 2002.

[BMR05] J.C.M. Baeten, M.R. Mousavi, and M.A. Reniers. Timing the untimed: Terminating successfully while being conservative. In A. Middeldorp, V. van Oostrom, F. van Raamsdonk, and R.C. de Vrijer, editors, *Processes, Terms and Cycles: Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop, on the Occasion of His 60th Birthday*, volume 3838 of *Lecture Notes in Computer Science*, pages 251–279. Springer-Verlag, 2005.

[BR] J.C.M. Baeten and M.A. Reniers. Duplication of constants in process algebra. Accepted for publication in: Journal of Logic and Algebraic Programming.

[BV93] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *CONCUR'93, International Conference on Concurrency Theory*, volume 715 of *Lecture Notes in Computer Science*, 1993.

[Che93] L. Chen. A model for real-time process algebras (extended abstract). In A.M. Borzyszkowski and S. Sokolowski, editors, *Mathematical Foundations of Computer Science 1993, 18th International Symposium, MFCS'93, Gdansk, Poland, August 30 - September 3, 1993, Proceedings*, volume 711 of *Lecture Notes in Computer Science*, pages 372–381. Springer-Verlag, 1993.

[Fok94] W.J. Fokkink. *Clock, Trees and Stars in Process Theory*. PhD thesis, University of Amsterdam, 1994.

[Fok97] W.J. Fokkink. An axiomatization for regular processes in timed branching bisimulation. *Fundamenta Informaticae*, 32:329–340, 1997.

[FPW05] W.J. Fokkink, J. Pang, and A. Wijs. Is timed branching bisimilarity an equivalence indeed?. In *Formal Modeling and Analysis of Timed Systems, Third International Conference, FORMATS 2005, Uppsala, Sweden, September 26-28, 2005, Proceedings*, volume 3829 of *Lecture Notes in Computer Science*, pages 258–272. Springer-Verlag, 2005.

[Gro97] J.F. Groote. The syntax and semantics of timed μCRL. Technical Report SEN-R9709, CWI, Amsterdam, 1997.

[GW89] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G.X. Ritter, editor, *Information Processing 89*, pages 613–618. North-Holland, 1989.

[GW96] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3):555–600, 1996.

[HM85] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, 32:137–161, 1985.

[HSZF93] C. Ho-Stuart, H.S.M. Zedan, and M. Fang. Congruent weak bisimulation with dense real-time. *Information Processing Letters*, 46(2):55–61, 1993.

[Klu93] A.S. Klusener. *Models and Axioms for a Fragment of Real Time Process Algebra*. PhD thesis, Eindhoven University of Technology, 1993.

[Lut02] Bas Luttik. *Choice Quantification in Process Algebra*. PhD thesis, University of Amsterdam, April 2002.

[MT92] F. Moller and C.M.N. Tofts. Behavioural abstraction in TCCS. In W. Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, volume 623 of *Lecture Notes in Computer Science*, pages 559–570. Springer-Verlag, 1992.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5^{th} GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[QdFA93] J. Quemada, D. de Frutos, and A. Azcorra. TIC: A TImed calculus. *Formal Aspects of Computing*, 5(3):224–252, 1993.

[RGvdZvW02] M.A. Reniers, J.F. Groote, M.B. van der Zwaag, and J. van Wamel. Completeness of timed μCRL. *Fundamenta Informaticae*, 50(3-4):361–402, 2002.

[Zwa01] M.B. van der Zwaag. The cones and foci proof technique for timed transition systems. *Information Processing Letters*, 80(1):33–40, 2001.

A Timed Rooted Branching Bisimilarity is an Equivalence

In order to conclude that timed rooted branching bisimilarity as defined in Section 4.2 is an equivalence, it has to be shown that it is reflexive, symmetric, and transitive.

Reflexivity

The reflexivity of timed rooted branching bisimilarity follows directly from the fact that the relation R that relates every closed process term to itself is a timed branching bisimulation relation and that it satisfies the root condition for all its pairs of closed process terms.

Symmetry

The symmetry of timed rooted branching bisimilarity follows immediately from the requirement in the definition of a timed branching bisimulation relation that it has to be symmetric.

Transitivity

Lemma 1. *Let p and q be processes and R a timed branching bisimulation relation such that $(p, q) \in R$. For all p' such that $p \Rightarrow p'$, we have that there is a q' such that $q \Rightarrow q'$ and $(p', q') \in R$.*

Proof. We prove this lemma by induction on the length of the derivation of $\Rightarrow$.

- $p \Rightarrow p'$ because $p = p'$. We also have $q \Rightarrow q$ and $(p, q) \in R$.
- $p \Rightarrow p'$ because there is a p^* such that $p \Rightarrow p^* \xrightarrow{\tau} p'$. By induction we have that there is a q^* such that $q \Rightarrow q^*$ and $(p^*, q^*) \in R$. The latter, with $p^* \xrightarrow{\tau} p'$, gives us that there are q^{**} and q' such that $q^* \Rightarrow q^{**} \xrightarrow{(\tau)} q'$, $(p^*, q^{**}) \in R$ and $(p', q') \in R$. Because $q \Rightarrow q^* \Rightarrow q^{**}$ trivially means $q \Rightarrow q^{**}$, we have $q \Rightarrow q^{**} \xrightarrow{(\tau)} q'$. Thus we have $q \Rightarrow q'$ with $(p', q') \in R$.

Definition 2. *Let R and R' be two relations. We define the* symmetric composition *of R and R', notation $R \bullet R'$, as follows $R \bullet R' = (R \circ R') \cup (R' \circ R)$.*

Obviously, the symmetric composition of two symmetrical relations is again symmetrical.

Lemma 2. *Let R and R' be timed branching bisimulation relations. The relation $R \bullet R'$ is a timed branching bisimulation relation.*

Proof. Let R and R' be timed branching bisimulation relations and let p and r be processes such that $(p, r) \in R \bullet R'$. Also take process q such that $(p, q) \in R$ and $(q, r) \in R'$.

1. If $p \xrightarrow{a}_t p'$, then, because of $(p, q) \in R$, we know that there are q^* and q' such that $q \Rightarrow q^* \xrightarrow{a}_t q'$, $(p, q^*) \in R$ and $(p', q') \in R$. Because of $(q, r) \in R'$ and Lemma 1 we have that there is a r^* such that $r \Rightarrow r^*$ and $(q^*, r^*) \in R'$. From this, and $q^* \xrightarrow{a}_t q'$, it follows that there are r^{**} and r' such that $r^* \Rightarrow r^{**} \xrightarrow{a}_t r'$, $(q^*, r^{**}) \in R'$ and $(q', r') \in R'$. Therefore we have $r \Rightarrow r^{**} \xrightarrow{a}_t r'$ with $(p, r^{**}) \in R \bullet R'$ and $(p', r') \in R \bullet R'$.

2. If $p \xrightarrow{\tau} p'$, then, because of $(p, q) \in R$, we know that there are q^* and q' such that $q \Rightarrow q^* \xrightarrow{(\tau)} q'$, $(p, q^*) \in R$ and $(p', q') \in R$. Because of $(q, r) \in R'$ and Lemma 1 we have that there is a r^* such that $r \Rightarrow r^*$ and $(q^*, r^*) \in R'$. From this, and $q^* \xrightarrow{(\tau)} q'$, it follows that there are r^{**} and r' such that $r^* \Rightarrow r^{**} \xrightarrow{(\tau)} r'$, $(q^*, r^{**}) \in R'$ and $(q', r') \in R'$. Therefore we have $r \Rightarrow r^{**} \xrightarrow{(\tau)} r'$ with $(p, r^{**}) \in R \bullet R'$ and $(p', r') \in R \bullet R'$.
3. If $p \downarrow_t$, then, because of $(p, q) \in R$, we know that there is a q' such that $q \Rightarrow q' \downarrow_t$ and $(p, q') \in R$. Because of $(q, r) \in R'$ and Lemma 1 we have that there is a r^* such that $r \Rightarrow r^*$ and $(q', r^*) \in R'$. From this, and $q' \downarrow_t$, it follows that there is a r' such that $r^* \Rightarrow r' \downarrow_t$ and $(q', r') \in R'$. Therefore we have $r \Rightarrow r' \downarrow_t$ with $(p, r') \in R \bullet R'$.
4. If $p \rightsquigarrow_t$, then, because of $(p, q) \in R$, we know that there is a q' such that $q \Rightarrow q' \rightsquigarrow_t$ and $(p, q') \in R$. Because of $(q, r) \in R'$ and Lemma 1 we have that there is a r^* such that $r \Rightarrow r^*$ and $(q', r^*) \in r'$. From this, and $q' \rightsquigarrow_t$, it follows that there is a r' such that $r^* \Rightarrow r' \rightsquigarrow_t$ and $(q', r') \in R'$. Therefore we have $r \Rightarrow r' \rightsquigarrow_t$ and $(p, r') \in R \bullet R'$.

The proof for the case that $(p, q) \in R'$ and $(q, r) \in R$ is similar.

Theorem 6. *Timed rooted branching bisimilarity $\underline{\leftrightarrow}_{\mathrm{rb}}$ is transitive. That is, if $p \underline{\leftrightarrow}_{\mathrm{rb}} q$ and $q \underline{\leftrightarrow}_{\mathrm{rb}} r$, then also $p \underline{\leftrightarrow}_{\mathrm{rb}} r$ (for all processes p,q and r).*

Proof. Let p, q and r be processes such that $p \underline{\leftrightarrow}_{\mathrm{rb}} q$ and $q \underline{\leftrightarrow}_{\mathrm{rb}} r$. This means that there are timed branching bisimulation relations R and R' such that $(p, q) \in R$ and $(q, r) \in R'$ and the root condition holds for (p, q) (with respect to R) and for (q, r) (with respect to R'). By Definition 2 we have that $(p, q) \in R \bullet R'$ and Lemma 2 says that $R \bullet R'$ is a timed branching bisimulation relation. Thus, we only need that the root condition holds for (p, r) with respect to $R \bullet R'$, which follows straightforwardly from the fact that it holds for (p, q) with respect to R and for (q, r) with respect to R'.

B Timed Rooted Branching Bisimilarity is a Congruence

In this appendix, proofs are given for congruence of timed rooted branching bisimilarity with respect to all operators introduced in this paper.

B.1 Action Prefix

Assume that $p \underline{\leftrightarrow}_{\mathrm{rb}} q$. Furthermore, assume that R is the witness for this assumption. Define

$$R' = \{(a^{@t}.p, a^{@t}.q), (a^{@t}.q, a^{@t}.p)\} \cup R_{\gg},$$

where $R_{\gg}$ is the relation that is used to prove congruence with respect to the time-initialisation operator.

For the proof that the pairs from $R_{\gg}$ satisfy the transfer conditions please refer to the proof of congruence with respect to the time-initialisation operator. Thus it remains to verify this for the pairs $(a^{@t}.p, a^{@t}.q)$ and $(a^{@t}.q, a^{@t}.p)$. Due to symmetry considerations it suffices to consider the pair $(a^{@t}.p, a^{@t}.q)$ only.

Since the process term $a^{@t}.p$ does not have silent step transitions and termination predicates, these cases are trivially satisfied.

- Suppose that $a^{@t}.p \xrightarrow{a}_t p'$ for some closed term p'. Then, by inspection of the deduction rules it follows that $p' = t \gg p$. Using the deduction rules we also obtain $a^{@t}.q \xrightarrow{a}_t t \gg q$. We also have that $(t \gg p, t \gg q) \in R'$. From this it follows, take $q^* = a^{@t}.q$ and $q' = t \gg q$, that there exist q^* and q' such that $a^{@t}.q \Rightarrow q^* \xrightarrow{(a)}_t q'$ and $(a^{@t}.p, q^*) \in R'$ and $(p', q') \in R'$.

– Suppose that $a^{@t}.p \rightsquigarrow_u$ for some $u \in \mathrm{Time}$. Then, by inspection of the deduction rules it follows that $u \leq t$. Then we also have $a^{@t}.q \rightsquigarrow_u$. From this it follows, take $q^* = a^{@t}.q$, that there exists a q^* such that $a^{@t}.q \Rightarrow q^* \rightsquigarrow_u$ and $(a^{@t}.p, q^*) \in R'$.

As a part of the above proof of the transfer conditions we have already shown that the pair $(a^{@t}.p, a^{@t}.q)$ satisfies the root condition.

B.2 Alternative Composition

Assume that $p_1 \leftrightarrow_{\mathrm{rb}} q_1$ and that $p_2 \leftrightarrow_{\mathrm{rb}} q_2$. Furthermore, assume that R_1 and R_2 are the witnesses for these assumptions. Define

$$R = \{(p_1 + p_2, q_1 + q_2), (q_1 + q_2, p_1 + p_2)\} \cup R_1 \cup R_2.$$

It is trivial that the pairs from R that are also in R_1 or R_2 satisfy the transfer conditions of timed branching bisimilarity. Thus it remains to verify this for the pairs $(p_1 + p_2, q_1 + q_2)$ and $(q_1 + q_2, p_1 + p_2)$. Due to symmetry considerations it suffices to consider the pair $(p_1 + p_2, q_1 + q_2)$ only.

– Suppose that $p_1 + p_2 \xrightarrow{a}_t p$ for some $a \in \mathrm{Act}$, $t \in \mathrm{Time}$, and closed term p. Then, by inspection of the deduction rules it follows that $p_1 \xrightarrow{a}_t p$ or $p_2 \xrightarrow{a}_t p$. The two cases are symmetrical, thus we only consider the case that $p_1 \xrightarrow{a}_t p$. Since $(p_1, q_1) \in R_1$ and R_1 is a branching bisimulation relation that satisfies the root condition for (p_1, q_1) it follows that there exists a closed process term q such that $q_1 \xrightarrow{a}_t q$ and $(p, q) \in R_1$. Then we also have $q_1 + q_2 \xrightarrow{a}_t q$ and $(p, q) \in R_1$. From this it follows, take $q^* = q_1 + q_2$, that there exist q^* and q such that $q_1 + q_2 \Rightarrow q^* \xrightarrow{a}_t q$ and $(p_1 + p_2, q^*) \in R$ and $(p, q) \in R$.
– Suppose that $p_1 + p_2 \xrightarrow{\tau} p$ for some closed term p. Then, by inspection of the deduction rules it follows that $p_1 \xrightarrow{\tau} p$ or $p_2 \xrightarrow{\tau} p$. The two cases are symmetrical, thus we only consider the case that $p_1 \xrightarrow{\tau} p$. Since $(p_1, q_1) \in R_1$ and R_1 is a branching bisimulation relation that satisfies the root condition for (p_1, q_1) it follows that there exists a closed process term q such that $q_1 \xrightarrow{\tau} q$ and $(p, q) \in R_1$. Then we also have $q_1 + q_2 \xrightarrow{\tau} q$ and $(p, q) \in R_1$. From this it follows, take $q^* = q_1 + q_2$, that there exist q^* and q such that $q_1 + q_2 \Rightarrow q^* \xrightarrow{(\tau)} q$ and $(p_1 + p_2, q^*) \in R$ and $(p, q) \in R$.
– Suppose that $p_1 + p_2 \downarrow_t$ for some $t \in \mathrm{Time}$. Then $p_1 \downarrow_t$ or $p_2 \downarrow_t$. These cases are symmetrical, thus we only consider the first case. Since $(p_1, q_1) \in R_1$ and R_1 is a branching bisimulation relation that satisfies the root condition for (p_1, q_1) it follows that $q_1 \downarrow_t$. Then we also have $q_1 + q_2 \downarrow_t$. From this it follows, take $q^* = q_1 + q_2$, that there exists a q^* such that $q_1 + q_2 \Rightarrow q^* \downarrow_t$ and $(p_1 + p_2, q^*) \in R$.
– Suppose that $p_1 + p_2 \rightsquigarrow_t$ for some $t \in \mathrm{Time}$. Then $p_1 \rightsquigarrow_t$ or $p_2 \rightsquigarrow_t$. These cases are symmetrical, thus we only consider the first case. Since $(p_1, q_1) \in R_1$ and R_1 is a branching bisimulation relation that satisfies the root condition for (p_1, q_1) it follows that $q_1 \rightsquigarrow_t$. Then we also have $q_1 + q_2 \rightsquigarrow_t$. From this it follows, take $q^* = q_1 + q_2$, that there exists a q^* such that $q_1 + q_2 \Rightarrow q^* \rightsquigarrow_t$ and $(p_1 + p_2, q_1, q_2) \in R$.

As a part of the above proof of the transfer conditions we have already shown that the pair $(p_1 + p_2, q_1 + q_2)$ satisfies the root condition.

B.3 Time initialisation

Assume that $p \leftrightarrow_{\mathrm{rb}} q$. Furthermore, assume that R is the witness for this assumption. Define

$$R' = \{(t \gg p', t \gg q') \mid (p', q') \in R\} \cup R.$$

It is trivial that the pairs from R' that are also in R satisfy the transfer conditions of timed branching bisimilarity. Thus it remains to verify this for the pairs $(t \gg p', t \gg q')$ with $(p', q') \in R$. Thereto, consider arbitrary p' and q' such that $(p', q') \in R$.

- Suppose that $t \gg p' \xrightarrow{a}_u p''$ for some $a \in \mathrm{Act}$, $u \in \mathrm{Time}$, and closed process term p''. By inspection of the deduction rules his must be due to $p' \xrightarrow{a}_u p''$ and $t \leq u$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exist q^* and q'' such that $q' \Rightarrow q^* \xrightarrow{a}_u q''$ and $(p', q^*) \in R$ and $(p'', q'') \in R$. But then also $t \gg q' \Rightarrow t \gg q^* \xrightarrow{a}_u q''$ and $(t \gg p', t \gg q^*) \in R'$ and $(p'', q'') \in R'$.
- Suppose that $t \gg p' \xrightarrow{\tau} p''$ for some closed process term p''. By inspection of the deduction rules his must be due to $p' \xrightarrow{\tau} p'''$ for some p''' such that $p'' = t \gg p'''$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exist q^* and q''' such that $q' \Rightarrow q^* \overset{(\tau)}{\rightarrow} q'''$ and $(p', q^*) \in R$ and $(p''', q''') \in R$. But then also $t \gg q' \Rightarrow t \gg q^* \overset{(\tau)}{\rightarrow} t \gg q'''$ and $(t \gg p', t \gg q^*) \in R'$ and $(t \gg p''', t \gg q''') \in R'$.
- Suppose that $t \gg p' \downarrow_u$ for some $u \in \mathrm{Time}$. This must be due to $p' \downarrow_u$ and $t \leq u$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exists a q^* such that $q' \Rightarrow q^* \downarrow_u$ and $(p', q^*) \in R$. But then also $t \gg q' \Rightarrow t \gg q^* \downarrow_u$ and $(t \gg p', t \gg q^*) \in R'$.
- Suppose that $t \gg p' \rightsquigarrow_u$ for some $u \in \mathrm{Time}$. This must be due to $p' \rightsquigarrow_u$ or $u \leq t$. In the first case, since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exists a q^* such that $q' \Rightarrow q^* \rightsquigarrow_u$. and $(p', q^*) \in R$. But then also $t \gg q' \Rightarrow t \gg q^* \rightsquigarrow_u$. and $(t \gg p', t \gg q^*) \in R'$. In the second case, it immediately follows that $t \gg q' \rightsquigarrow_u$. From this it follows, take $q^* = t \gg q'$, that there exists a q^* such that $t \gg q' \Rightarrow q^* \rightsquigarrow_u$ and $(t \gg p', q^*) \in R'$.

The proof that the pair $(t \gg p, t \gg q)$ satisfies the root condition follows the same lines as the above proofs and is therefore omitted.

B.4 Silent Step Prefix

Assume that $p \leftrightarrow_{\mathrm{rb}} q$. Furthermore, assume that R is the witness for this assumption. Define

$$R' = \{(\tau.p, \tau.q), (\tau.q, \tau.p)\} \cup R.$$

It is trivial that the pairs from R' that are also in R satisfy the transfer conditions of timed branching bisimilarity. Thus it remains to verify this for the pairs $(\tau.p, \tau.q)$ and $(\tau.q, \tau.p)$. Due to symmetry considerations it suffices to consider the pair $(\tau.p, \tau.q)$ only.

Since the process term $\tau.p$ does not have action transitions, termination predicates and delay predicates, these cases are trivially satisfied.

- Suppose that $\tau.p \xrightarrow{\tau} p'$ for some closed term p'. Then, by inspection of the deduction rules it follows that $p' = p$. Using the deduction rules we also obtain $\tau.q \xrightarrow{\tau} q$. We also have that $(p, q) \in R$. From this it follows, take $q^* = \tau.q$ and $q' = q$, that there exist q^* and q' such that $\tau.q \Rightarrow q^* \overset{(\tau)}{\rightarrow} q$ and $(p_1, q^*) \in R'$ and $(p, q) \in R'$.

As a part of the above proof of the transfer conditions we have already shown that the pair $(\tau.p, \tau.q)$ satisfies the root condition.

B.5 Abstraction

Assume that $p \underline{\leftrightarrow}_{\mathrm{rb}}\, q$. Furthermore, assume that R is the witness for this assumption. Define

$$R' = \{(\tau_I(p'), \tau_I(q')) \mid (p', q') \in R\} \cup R.$$

It is trivial that the pairs from R' that are also in R satisfy the transfer conditions of timed branching bisimilarity. Thus it remains to verify this for the pairs $(\tau_I(p'), \tau_I(q'))$ with $(p', q') \in R$. Thereto, consider arbitrary p' and q' such that $(p', q') \in R$.

- Suppose that $\tau_I(p') \xrightarrow{a}_u p''$ for some $a \in \mathrm{Act}$, $u \in \mathrm{Time}$, and closed process term p''. By inspection of the deduction rules his must be due to $a \notin I$ and $p' \xrightarrow{a}_u p'''$ for some closed term p''' such that $p'' = \tau_I(p''')$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exist q^* and q''' such that $q' \Rightarrow q^* \xrightarrow{a}_u q'''$ and $(p', q^*) \in R$ and $(p''', q''') \in R$. But then also $\tau_I(q') \Rightarrow \tau_I(q^*) \xrightarrow{a}_u \tau_I(q''')$ and $(\tau_I(p'), \tau_I(q^*)) \in R'$ and $(\tau_I(p'''), \tau_I(q''')) \in R'$.
- Suppose that $\tau_I(p') \xrightarrow{\tau} p''$ for some closed process term p''. By inspection of the deduction rules his must be due to (1) $a \in I$ and $p' \xrightarrow{a}_t p'''$ for some p''' and $t \in \mathrm{Time}$ such that $p'' = t \gg p'''$, or due to (2) $p' \xrightarrow{\tau} p'''$ for some closed term p''' such that $p'' = \tau_I(p''')$.
  In the first case, since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exist q^* and q''' such that $q' \Rightarrow q^* \xrightarrow{a}_t q'''$ and $(p', q^*) \in R$ and $(p''', q''') \in R$. But then also $\tau_I(q') \Rightarrow \tau_I(q^*) \xrightarrow{a}_t \tau_I(q''')$ and $(\tau_I(p'), \tau_I(q^*)) \in R'$ and $(\tau_I(p'''), \tau_I(q''')) \in R'$.
  In the second case, since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exist q^* and q''' such that $q' \Rightarrow q^* \xrightarrow{(\tau)} q'''$ and $(p', q^*) \in R$ and $(p''', q''') \in R$. But then also $\tau_I(q') \Rightarrow \tau_I(q^*) \xrightarrow{(\tau)} \tau_I(q''')$ and $(\tau_I(p'), \tau_I(q^*)) \in R'$ and $(\tau_I(p'''), \tau_I(q''')) \in R'$.
- Suppose that $\tau_I(p') \downarrow_t$ for some $t \in \mathrm{Time}$. This must be due to $p' \downarrow_t$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exists a q^* such that $q' \Rightarrow q^* \downarrow_t$ and $(p', q^*) \in R$. But then also $\tau_I(q') \Rightarrow \tau_I(q^*) \downarrow_t$ and $(\tau_I(p'), \tau_I(q^*)) \in R'$.
- Suppose that $\tau_I(p') \leadsto_t$ for some $t \in \mathrm{Time}$. This must be due to $p' \leadsto_t$. Since $(p', q') \in R$ and R is a branching bisimulation relation it follows that there exists a q^* such that $q' \Rightarrow q^* \leadsto_t$ and $(p', q^*) \in R$. But then also $\tau_I(q') \Rightarrow \tau_I(q^*) \leadsto_t$ and $(\tau_I(p'), \tau_I(q^*)) \in R'$.

B.6 Sequential Composition

Assume that $p_1 \underline{\leftrightarrow}_{\mathrm{rb}}\, q_1$ and that $p_2 \underline{\leftrightarrow}_{\mathrm{rb}}\, q_2$. Furthermore, assume that R_1 and R_2 are the witnesses for these assumptions. Define

$$R = \{(p_1' \cdot p_2, q_1' \cdot q_2), (q_1' \cdot q_2, p_1' \cdot p_2) \mid (p_1', q_1') \in R_1\} \cup R_{\gg_2},$$

where $R_{\gg_2}$ is the relation that is used to prove congruence with respect to the time-initialisation operator, taking R_2 for R.

For the proof that the pairs from $R_{\gg_2}$ also satisfy the transfer conditions please refer to the proof of congruence with respect to the time-initialisation operator. Thus it remains to verify this for the pairs $(p_1' \cdot p_2, q_1' \cdot q_2)$ and $(q_1' \cdot q_2, p_1' \cdot p_2)$ with $(p_1', q_1') \in R_1$. Due to symmetry considerations it suffices to consider the pairs $(p_1' \cdot p_2, q_1' \cdot q_2)$. Thereto, consider arbitrary p_1' and q_1' such that $(p_1', q_1') \in R_1$.

- Suppose that $p_1' \cdot p_2 \xrightarrow{a}_t p$ for some $a \in \mathrm{Act}$, $t \in \mathrm{Time}$, and closed term p. Then, by inspection of the deduction rules it follows that $p_1' \xrightarrow{a}_t p'$ for some p' such that $p = p' \cdot p_2$, or $p_1' \downarrow_u$ and $u \gg p_2 \xrightarrow{a}_t p$ for some $u \in \mathrm{Time}$. In the first case, since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exist q^* and q such that $q_1' \Rightarrow q^* \xrightarrow{a}_t q$ and $(p_1', q^*) \in R_1$ and $(p', q) \in R_1$. But then also $q_1' \cdot q_2 \Rightarrow q^* \cdot q_2 \xrightarrow{a}_t q \cdot q_2$ and $(p_1' \cdot p_2, q^* \cdot q_2) \in R$ and $(p' \cdot p_2, q \cdot q_2) \in R$.

In the second case, by inspection of the deduction rules it follows that $p_2 \xrightarrow{a}_t p$ and $u \leq t$. Since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exists a q such that $q_1' \Rightarrow q \downarrow_u$ and $(p_1', q) \in R_1$. And since $(p_2, q_2) \in R_2$ and R_2 is a branching bisimulation relation that satisfies the root condition for (p_2, q_2) it follows that there exists a q' such that $q_2 \xrightarrow{a}_t q'$ and $(p, q') \in R_2$. Due to $u \leq t$ we also have $u \gg q_2 \xrightarrow{a}_t q'$. But then also $q_1' \cdot q_2 \Rightarrow q \cdot q_2 \xrightarrow{a}_t q'$ and $(p_1' \cdot p_2, q \cdot q_2) \in R$ and $(p, q') \in R$.

– Suppose that $p_1' \cdot p_2 \xrightarrow{\tau} p$ for some closed term p. Then, by inspection of the deduction rules it follows that $p_1' \xrightarrow{\tau} p'$ for some p' such that$p = p' \cdot p_2$, or $p_1' \downarrow_u$ and $u \gg p_2 \xrightarrow{\tau} p$ for some $u \in \mathrm{Time}$. In the first case, since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exist q^* and q such that $q_1' \Rightarrow q^* \xrightarrow{(\tau)} q$ and $(p_1', q^*) \in R_1$ and $(p', q) \in R_1$. But then also $q_1' \cdot q_2 \Rightarrow q^* \cdot q_2 \xrightarrow{(\tau)} q \cdot q_2$ and $(p_1' \cdot p_2, q^* \cdot q_2) \in R$ and $(p' \cdot p_2, q \cdot q_2) \in R$.

In the second case, by inspection of the deduction rules it follows that there exists a p' such that $p_2 \xrightarrow{\tau} p'$ and $p = u \gg p'$. Since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exists a q such that $q_1' \Rightarrow q \downarrow_u$ and $(p_1', q) \in R_1$. And since $(p_2, q_2) \in R_2$ and R_2 is a branching bisimulation relation that satisfies the root condition for (p_2, q_2) it follows that there exists a q' such that $q_2 \xrightarrow{\tau} q'$ and $(p', q') \in R_2$. Then we also have $u \gg q_2 \xrightarrow{\tau} u \gg q'$ and $(u \gg p', u \gg q') \in R_{\gg}$. But then also $q_1' \cdot q_2 \Rightarrow q \cdot q_2 \xrightarrow{\tau} u \gg q'$ and $(p_1' \cdot p_2, q \cdot q_2) \in R$ and $(p, u \gg q') \in R$.

– Suppose that $p_1' \cdot p_2 \downarrow_t$ for some $t \in \mathrm{Time}$. This must be due to $p_1' \downarrow_u$ and $u \gg p_2 \downarrow_t$ for some $u \in \mathrm{Time}$. Furthermore, from inspection of the deduction rules it follows that $p_2 \downarrow_t$ and $u \leq t$. Since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exists a q^* such that $q_1' \Rightarrow q^* \downarrow_u$ and $(p_1', q^*) \in R_1$. And since $(p_2, q_2) \in R_2$ and R_2 is a branching bisimulation relation that satisfies the root condition for (p_2, q_2) it follows that $q_2 \downarrow_t$. Due to $u \leq t$ we also have $u \gg q_2 \downarrow_t$. But then also $q_1' \cdot q_2 \Rightarrow q^* \cdot q_2 \downarrow_t$ and $(p_1' \cdot p_2, q^* \cdot q_2) \in R$.

– Suppose that $p_1' \cdot p_2 \rightsquigarrow_t$ for some $t \in \mathrm{Time}$. This must be due to $p_1' \rightsquigarrow_t$ or $p_1' \downarrow_u$ and $u \gg p_2 \rightsquigarrow_t$ for some $u \in \mathrm{Time}$. In the first case, since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exists a q^* such that $q_1' \Rightarrow q^* \rightsquigarrow_t$ and $(p_1', q^*) \in R_1$. But then also $q_1' \cdot q_2 \Rightarrow q^* \cdot q_2 \rightsquigarrow_t$ and $(p_1' \cdot p_2, q^* \cdot q_2) \in R$.

In the second case, by inspection of the deduction rules it follows that $p_2 \rightsquigarrow_t$ and $u \leq t$. Since $(p_1', q_1') \in R_1$ and R_1 is a branching bisimulation relation it follows that there exists a q^* such that $q_1' \Rightarrow q^* \downarrow_u$ and $(p_1', q^*) \in R_1$. And since $(p_2, q_2) \in R_2$ and R_2 is a branching bisimulation relation that satisfies the root condition for (p_2, q_2) it follows that $q_2 \rightsquigarrow_t$. Due to $u \leq t$ we also have $u \gg q_2 \rightsquigarrow_t$. But then also $q_1' \cdot q_2 \Rightarrow q^* \cdot q_2 \rightsquigarrow_t$ and $(p_1' \cdot p_2, q^* \cdot q_2) \in R$.

The proof that the pair $(p_1' \cdot p_2, q_1' \cdot q_2)$ satisfies the root condition follows the same lines as the above proofs and is therefore omitted.

B.7 Parallel Composition

Assume that $p_1 \leftrightarrow_{\mathrm{rb}} q_1$ and that $p_2 \leftrightarrow_{\mathrm{rb}} q_2$. Furthermore, assume that R_1 and R_2 are the witnesses for these assumptions. Define $R_{\gg_1}$ to be the smallest relation such that $R_1 \subseteq R_{\gg_1}$ and if $(p, q) \in R_{\gg_1}$ then also $(t \gg p, t \gg q) \in R_{\gg_1}$. Define $R_{\gg_2}$ to be the smallest relation such that $R_2 \subseteq R_{\gg_2}$ and if $(p, q) \in R_{\gg_2}$ then also $(t \gg p, t \gg q) \in R_{\gg_2}$. Define

$$R = \{(p_1' \parallel p_2', q_1' \parallel q_2') \mid (p_1', q_1') \in R_{\gg_1} \wedge (p_2', q_2') \in R_{\gg_2}\}.$$

We first prove that $R_{\gg_1}$ and $R_{\gg_2}$ are branching bisimulation relations. The proofs for $R_{\gg_1}$ and $R_{\gg_2}$ are essentially the same, thus we will only give the proof for $R_{\gg_1}$. We do this by induction on the construction of relation $R_{\gg_1}$.

– Suppose that a pair is in $R_{\gg_1}$ since it is in R_1. The transfer conditions hold trivially for such a pair since R_1 is assumed to be a rooted branching bisimulation relation.
– Suppose that a pair is in $R_{\gg_1}$ because it is of the form $(t \gg p, t \gg q)$ for some p and q such that $(p, q) \in R_{\gg_1}$. By induction we have that the pair (p, q) satisfies the transfer conditions. Then, by the proof of congruence with respect to the time-initialisation operator, we have that the transfer conditions hold for $(t \gg p, t \gg q)$.

It remains to verify that the transfer conditions hold for the pairs $(p_1' \parallel p_2', q_1' \parallel q_2')$ with $(p_1', q_1') \in R_{\gg_1}$ and $(p_2', q_2') \in R_{\gg_2}$. Thereto, consider arbitrary p_1', q_1', p_2' and q_2' such that $(p_1', q_1') \in R_{\gg_1}$ and $(p_2', q_2') \in R_{\gg_2}$.

– Suppose that $p_1' \parallel p_2' \xrightarrow{a}_t p$ for some $a \in \mathrm{Act}$, $t \in \mathrm{Time}$, and closed term p. Then, by inspection of the deduction rules it follows that $p_1' \xrightarrow{a}_t p'$ and $p_2' \rightsquigarrow_t$ for some p' such that $p = p' \parallel t \gg p_2'$, or $p_2' \xrightarrow{a}_t p'$ and $p_1' \rightsquigarrow_t$ for some p' such that $p = t \gg p_1' \parallel p'$. Assume the first case. Since $(p_1', q_1') \in R_{\gg_1}$ and $R_{\gg_1}$ is a branching bisimulation relation it follows that there exist q_1^* and q such that $q_1' \Rightarrow q_1^* \xrightarrow{a}_t q$ and $(p_1', q_1^*) \in R_{\gg_1}$ and $(p', q) \in R_{\gg_1}$. Furthermore, since $(p_2', q_2') \in R_{\gg_2}$ and $R_{\gg_2}$ is a branching bisimulation it follows that there exists a q_2^* such that $q_2' \Rightarrow q_2^* \rightsquigarrow_t$ and $(p_2', q_2^*) \in R_{\gg_2}$. But then also $q_1' \parallel q_2' \Rightarrow q_1^* \parallel q_2' \Rightarrow q_1^* \parallel q_2^* \xrightarrow{a}_t q \parallel t \gg q_2^*$ and $(p_1' \parallel p_2', q_1^* \parallel q_2^*) \in R$ and $(p' \parallel t \gg p_2', q \parallel t \gg q_2^*) \in R$. The alternative case is symmetric to this one.
– Suppose that $p_1' \parallel p_2' \xrightarrow{\tau} p$ for some closed term p. Then, by inspection of the deduction rules it follows that $p_1' \xrightarrow{\tau} p'$ for some p' such that $p = p' \parallel p_2'$, or $p_2' \xrightarrow{\tau} p'$ for some p' such that $p = p_1' \parallel p'$. Assume the first case. Since $(p_1', q_1') \in R_{\gg_1}$ and $R_{\gg_1}$ is a branching bisimulation relation it follows that there exist q_1^* and q such that $q_1' \Rightarrow q_1^* \xrightarrow{(\tau)} q$ and $(p_1', q_1^*) \in R_{\gg_1}$ and $(p', q) \in R_{\gg_1}$. But then also $q_1' \parallel q_2' \Rightarrow q_1^* \parallel q_2' \xrightarrow{(\tau)} q \parallel q_2'$ and $(p_1' \parallel p_2', q_1^* \parallel q_2') \in R$ and $(p' \parallel p_2', q \parallel q_2') \in R$. The alternative case is symmetric to this one.
– Suppose that $p_1' \parallel p_2' \downarrow_t$ for some $t \in \mathrm{Time}$. Then, by inspection of the deduction rules it follows that $p_1' \downarrow_t$ and $p_2' \downarrow_t$. Since $(p_1', q_1') \in R_{\gg_1}$ and $R_{\gg_1}$ is a branching bisimulation relation it follows that there exists a q_1^* such that $q_1' \Rightarrow q_1^* \downarrow_t$ and $(p_1', q_1^*) \in R_{\gg_1}$. Furthermore, since $(p_2', q_2') \in R_{\gg_2}$ and $R_{\gg_2}$ is a branching bisimulation relation it follows that there exists a q_2^* such that $q_2' \Rightarrow q_2^* \downarrow_t$ and $(p_2', q_2^*) \in R_{\gg_2}$. But then also $q_1' \parallel q_2' \Rightarrow q_1^* \parallel q_2' \Rightarrow q_1^* \parallel q_2^* \downarrow_t$ and $(p_1' \parallel p_2', q_1^* \parallel q_2^*) \in R$.
– Suppose that $p_1' \parallel p_2' \rightsquigarrow_t$ for some $t \in \mathrm{Time}$. Then, by inspection of the deduction rules it follows that $p_1' \rightsquigarrow_t$ and $p_2' \rightsquigarrow_t$. Since $(p_1', q_1') \in R_{\gg_1}$ and $R_{\gg_1}$ is a branching bisimulation relation it follows that there exists a q_1^* such that $q_1' \Rightarrow q_1^* \rightsquigarrow_t$ and $(p_1', q_1^*) \in R_{\gg_1}$. Furthermore, since $(p_2', q_2') \in R_{\gg_2}$ and $R_{\gg_2}$ is a branching bisimulation relation it follows that there exists a q_2^* such that $q_2' \Rightarrow q_2^* \rightsquigarrow_t$ and $(p_2', q_2^*) \in R_{\gg_2}$. But then also $q_1' \parallel q_2' \Rightarrow q_1^* \parallel q_2' \Rightarrow q_1^* \parallel q_2^* \rightsquigarrow_t$ and $(p_1' \parallel p_2', q_1^* \parallel q_2^*) \in R$.

The proof that the pair $(p_1 \parallel p_2, q_1 \parallel q_2)$ satisfies the root condition follows the same lines as the above proofs and is therefore omitted.