Termination in Timed Process Algebra

J.C.M. Baeten M.A. Reniers

Department of Mathematics and Computing Science,
Eindhoven University of Technology,
{J.C.M.Baeten,M.A.Reniers}@tue.nl , http://www.win.tue.nl/{~josb,~michelr}

Abstract. We investigate different forms of termination in timed process algebras. The integrated framework of discrete and dense time, relative and absolute time process algebras is extended with forms of successful and unsuccessful termination. The different algebras are interrelated by embeddings and conservative extensions.
Note: Full version, including proofs, see [BR00].

1 Introduction

Process algebras that incorporate some form of timing, enabling quantitive analysis of time performance, have been studied extensively by now (see e.g. [BB91,HR95,MT90,NS94,QdFA93]), and successfully applied in a number of case studies (see e.g. [BR97,SDJ+91]).

Different versions of timing are needed, depending on the application. We have discrete vs. dense timing, and relative vs. absolute timing. All these versions are treated in a common framework, and interrelated by embeddings and conservative extension relations, in [BM99]. There, considerable attention was given to the inaction constant δ, standing for unsuccessful termination or deadlock. In the untimed theory, this constant has several roles. On the one hand, it is the neutral element for alternative composition (very useful in algebraical calculations and extensions with generalized sum over a data type), on the other hand it stands for a blocked (or encapsulated) atomic action. These roles diverge when timing is added. As the interpretation of an untimed atomic action a in a timed setting is an action that happens at some unspecified moment in time, the blocked action δ can let time pass, but can execute no action. But in a timed setting, some processes cannot let time pass indefinitely, as they must execute some action by some specified time. Thus, δ cannot be the neutral element of alternative composition, and this role is taken over by a new constant $\dot{\delta}$. The constant $\dot{\delta}$ (usually called *immediate deadlock*) denotes a state where inconsistent timing is encountered, a state that cannot be reached by just idling. In this paper, we call this a *virtual state*: no activity is allowed in such a state, not even of another process put in parallel. To complete the picture, apart from the untimed δ and the virtual $\dot{\delta}$ Baeten and Middelburg also have the relative discrete time $\underline{\underline{\delta}}$, the relative dense time $\tilde{\tilde{\delta}}$, the absolute discrete time $\underline{\delta}$ and the absolute dense time $\tilde{\delta}$.

The empty process ϵ denoting successful termination or *skip* has not been studied nearly as well. The untimed theory was investigated in [KV85,BG87,Vra97], with timing we only have discrete relative timing in [Ver97,BV97]. Here, we remedy this situation. In the untimed theory, also ϵ has more roles. On the one hand, it serves as the neutral element for sequential composition (and parallel composition), on the other hand, it stands for the process that executes no actions but terminates at some unspecified time. With timing, if ϵ stands for the latter process, that allows passage of time, we introduce a new constant $\dot{\epsilon}$ that is the neutral element for sequential and parallel composition. The process $\dot{\epsilon}$ also denotes a virtual state, where no parallel activity can preceed the termination. With this constant, we finally have a complete interpretation of the constant process a in a timed setting: upon executing the action, what remains is $\dot{\epsilon}$, and we have an interpretation of the 'mysterious' $\surd$ appearing in SOS rules as a process term. We complete the picture for relative dense time with the addition of the constant $\tilde{\tilde{\epsilon}}$.

Thus, we have extended the integrated framework of process algebras with timing with a dense time process algebra with relative timing and empty process; the various subtheories are still related by means of embeddings and conservative extensions, as we show in this paper. This extension with the empty process, in itself needed for a clear understanding of termination in timed process algebra, is also needed in order to give semantics for programming languages and specification languages that involve skip: examples are SDL, see [IT94] and MSC, see [Ren99].

2 The basic process algebra

2.1 The basic process algebra

We start with the presentation of Basic Process Algebra with dense relative timing and neutral elements for alternative and sequential composition, called $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$. This process algebra is based on the algebra BPA^{srt} of [BM99] and extends this by the constant $\dot{\epsilon}$. Using $\dot{\epsilon}$ the delay operators of [BM99] can be replaced by delay constants. Further, we have only minor modifications.

We have the following syntax:

- *alternative composition* or sum, written $+$. The choice is resolved by the execution of an action, not by the mere passage of time (see axiom SRT3, Table 1).
- *sequential composition* or product, written $\cdot$. Relative timing means that time is measured from the execution of the previous action.
- *atomic actions* $\tilde{\tilde{a}}$, where $a \in A$, a given set of atoms. The process $\tilde{\tilde{a}}$ denotes immediate and urgent execution of the action a, at the current moment of time, followed by immediate and successful termination.
- *time stop* $\tilde{\tilde{\delta}}$. Time cannot progress any more beyond the current moment of time, and no termination can take place. However, the current moment of time is a consistent state.

- *immediate deadlock* $\dot{\delta}$. The process $\dot{\delta}$ denotes a virtual state, the current moment of time cannot be reached by just idling. This is the neutral element of alternative composition (see axiom A6ID in Table 1).
- *urgent termination* $\tilde{\tilde{\epsilon}}$, denoting successful termination at the current moment of time. The current moment of time is a consistent state.
- *immediate empty process* $\dot{\epsilon}$. The immediate empty process denotes a virtual state. It is also the neutral element of sequential composition (see axioms A8IE, A9IE) and the remainder of $\tilde{\tilde{a}}$ after action execution.
- the *relative delay constant* $\tilde{\tilde{\sigma}}^p$. This constant describes a delay of an amount of time p where p is a non-negative real number. When $p = 0$, nothing is left (see axiom IE).

In the sequel we will write A_δ for $A \cup \{\delta\}$.

$x + y = y + x$	A1	$x + \dot{\delta} = x$	A6ID
$(x + y) + z = x + (y + z)$	A2	$\dot{\delta} \cdot x = \dot{\delta}$	A7ID
$x + x = x$	A3	$\dot{\epsilon} \cdot x = x$	A8IE
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$x \cdot \dot{\epsilon} = x$	A9IE
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5		
$\dot{\epsilon} + \tilde{\tilde{\delta}} = \tilde{\tilde{\epsilon}}$	UT	$\tilde{\tilde{\sigma}}^0 = \dot{\epsilon}$	IE
$\tilde{\tilde{a}} + \tilde{\tilde{\delta}} = \tilde{\tilde{a}}$	A6SRa	$\tilde{\tilde{\sigma}}^p \cdot \tilde{\tilde{\sigma}}^q = \tilde{\tilde{\sigma}}^{p+q}$	SRT2
$\tilde{\tilde{\delta}} \cdot x = \tilde{\tilde{\delta}}$	A7SR	$\tilde{\tilde{\sigma}}^p \cdot (x + y) = \tilde{\tilde{\sigma}}^p \cdot x + \tilde{\tilde{\sigma}}^p \cdot y$	SRT3
		$\tilde{\tilde{\sigma}}^r + \tilde{\tilde{\delta}} = \tilde{\tilde{\sigma}}^r$	A6SRb

Table 1. Axioms of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ ($a \in A_\delta$, $p, q \geq 0$, $r > 0$).

If we compare the axioms of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ with the axioms of BPA^{srt} [BM99] we find that there are not many differences. The axioms A1-A5, A6ID-A7ID, A6SRa, and A7SR are identical. The axioms SRT2-SRT3 and A6SRb are only different in the sense that we now have the constant $\tilde{\tilde{\sigma}}^p$ instead of the operator σ^p_{rel}. Axiom SRT1 of BPA^{srt} ($\sigma^0_{\mathrm{rel}}(x) = x$) is in the setting of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ derivable from axioms IE and A8IE. Axiom SRT4 of BPA^{srt} ($\sigma_{\mathrm{rel}}(x \cdot y) = \sigma_{\mathrm{rel}}(x) \cdot y$) is not necessary anymore as it is part of axiom A5, again due to the fact that we have a delay constant instead of an operator.

The process algebra $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ adds to the axioms of BPA^{srt} the axioms explaining that the immediate empty process is a neutral element for sequential composition (A8IE, A9IE). Axiom UT expresses that urgent termination can be expressed in terms of urgent deadlock and immediate termination. Axiom IE expresses that immediate termination is the same as $\tilde{\tilde{\sigma}}^0$.

As we have just explained, the constants for immediate termination and urgent termination are both expressible in terms of other constants. It is impossible

to express the delay constant in terms of the operator without using something like the immediate termination constant.

2.2 Operational semantics

The definition of an operational semantics by means of SOS deduction rules is also standard. By means of these rules, we define binary relations $_ \xrightarrow{a} _$ and $_ \overset{t}{\mapsto} _$ and unary relations $_\downarrow$ and $_\uparrow$ on closed terms P (for $a \in A$ and $t > 0$). Intuitively, they have the following meaning: $x \xrightarrow{a} x'$ means that x evolves into x' by executing atomic action a; $x \overset{t}{\mapsto} x'$ means that x evolves into x' by waiting for time t; $x\downarrow$ means that x has an option to terminate successfully, it holds for all process with $\dot{\epsilon}$ as an alternative; and $x\uparrow$ means that x has reached a virtual state, it only holds for processes $\dot{\delta}$ and $\dot{\epsilon}$.

- $x \xrightarrow{a} x'$ means that x evolves into x' by executing atomic action a
- $x \overset{t}{\mapsto} x'$ means that x evolves into x' by waiting for time t
- $x\downarrow$ means that x has an option to terminate successfully
- $x\uparrow$ means that x has reached a virtual state

For BPA^{srt} similar predicates and relations are used. There are two differences. First, we replace the immediate deadlock predicate ID by the predicate $\uparrow$. Their meaning is the same however. Second, the introduction of the empty process also has a positive side for the description of the operational semantics of the process algebra as it allows the separation of the action step relation and the termination predicate. This means that the relation $\xrightarrow{a}$ and the predicate $\xrightarrow{a}\surd$ can be replaced by the relation $\xrightarrow{a}$ and the predicate $\downarrow$. Another advantage is that all states of the transition system correspond to process terms.

In BPA^{srt} time-determinism is enforced in the term deduction system explicitly by adopting the following deduction rules for $+$:

$$\frac{x \overset{r}{\mapsto} x' \quad y \overset{r}{\mapsto} y'}{x + y \overset{r}{\mapsto} x' + y'}, \qquad \frac{x \overset{r}{\mapsto} x' \quad y \overset{r}{\not\mapsto}}{x + y \overset{r}{\mapsto} x'}, \qquad \frac{x \overset{r}{\not\mapsto} \quad y \overset{r}{\mapsto} y'}{x + y \overset{r}{\mapsto} y'}.$$

For BPA$^{srt}_{\dot{\epsilon}}$ similar deduction rules for $+$ can be defined, but due to the existence of a neutral element for sequential composition we get an intricate interplay between time transitions and termination of sequentially composed processes. For example, consider the process $(\tilde{\tilde{\sigma}}^1 + \tilde{\tilde{\epsilon}}) \cdot \tilde{\tilde{\sigma}}^2 \cdot \tilde{\tilde{b}}$. This process can delay up to 2 time units for both its alternatives. The principle of time-determinism requires that we combine these time transitions. This however is in general impossible. For, in order to determine the process term q a term of the form $x \cdot y$ evolves into after waiting for time t ($x \cdot y \overset{t}{\mapsto} q$), we have to consider all pairs of positive reals r and s such that $r + s = t$, and see if there are terms x' and y' such that $x \overset{r}{\mapsto} x'$, $x'\downarrow$, and $y \overset{s}{\mapsto} y'$. All possible y' must be summands of the term q. In discrete time with empty process, the number of possibilities is finite and case distinction works [BV97], here, this is no longer the case and we have to look for a different solution.

Therefore, we will not attempt to explicitly incorporate time-determinism in the deduction system. Rather, we formulate the time-determinism principle indirectly in the definition of equivalence (bisimilarity). This cannot be achieved by adopting the normal definition of strong bisimilarity. This can be seen as follows. Consider the processes $\tilde{\tilde{\sigma}}^1 \cdot (\tilde{\tilde{a}} + \tilde{\tilde{b}})$ and $\tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{a}} + \tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{b}}$. The first has only one possibility for delaying for 1 time unit: $\tilde{\tilde{\sigma}}^1 \cdot (\tilde{\tilde{a}} + \tilde{\tilde{b}}) \overset{1}{\mapsto} \tilde{\tilde{\sigma}}^0 \cdot (\tilde{\tilde{a}} + \tilde{\tilde{b}})$. The second has two such posibilities: $\tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{a}} + \tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{b}} \overset{1}{\mapsto} \tilde{\tilde{\sigma}}^0 \cdot \tilde{\tilde{a}}$ and $\tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{a}} + \tilde{\tilde{\sigma}}^1 \cdot \tilde{\tilde{b}} \overset{1}{\mapsto} \tilde{\tilde{\sigma}}^0 \cdot \tilde{\tilde{b}}$. Neither one of these resulting processes is strongly bisimilar with $\tilde{\tilde{\sigma}}^0 \cdot (\tilde{\tilde{a}} + \tilde{\tilde{b}})$. Only the addition of a deduction rule combining time transitions as is used in BPA^{srt} is not a solution as the transitions mentioned above are still there.

Hence, we look for another definition of equivalence. For this purpose we can have a look at the notion of σ-bisimilarity as it was used in the Algebra of Timed Frames [BFM96]. There, time transitions are defined separately and time-determinism is enforced in the definition of σ-bisimilarity by requiring that if a process s can do a time transition, then the set of all such time transitions from s must be related to the set of all such time transitions from the process t. We present the deduction rules in Table 2 and we define σ-bisimilarity for $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$.

$$[1]\ \tilde{\tilde{a}} \overset{a}{\to} \dot{\epsilon} \qquad [2]\ \tilde{\tilde{\epsilon}} \downarrow \qquad [3]\ \dot{\epsilon} \downarrow \qquad [4]\ \tilde{\tilde{\sigma}}^0 \downarrow$$

$$[5]\ \dot{\epsilon} \uparrow \qquad [6]\ \dot{\delta} \uparrow \qquad [7]\ \tilde{\tilde{\sigma}}^0 \uparrow \qquad [8]\ \tilde{\tilde{\sigma}}^{p+r} \overset{r}{\mapsto} \tilde{\tilde{\sigma}}^p$$

$$[9]\ \frac{x \overset{a}{\to} x'}{x + y \overset{a}{\to} x'} \qquad [10]\ \frac{y \overset{a}{\to} y'}{x + y \overset{a}{\to} y'} \qquad [11]\ \frac{x \downarrow}{x + y \downarrow} \qquad [12]\ \frac{y \downarrow}{x + y \downarrow}$$

$$[13]\ \frac{x \uparrow, y \uparrow}{x + y \uparrow} \qquad [14]\ \frac{x \overset{r}{\mapsto} x'}{x + y \overset{r}{\mapsto} x'} \qquad [15]\ \frac{y \overset{r}{\mapsto} y'}{x + y \overset{r}{\mapsto} y'}$$

$$[16]\ \frac{x \overset{a}{\to} x'}{x \cdot y \overset{a}{\to} x' \cdot y} \qquad [17]\ \frac{x \downarrow, y \overset{a}{\to} y'}{x \cdot y \overset{a}{\to} y'} \qquad [18]\ \frac{x \uparrow, x \not\downarrow}{x \cdot y \uparrow} \qquad [19]\ \frac{x \uparrow, x \downarrow, y \uparrow}{x \cdot y \uparrow}$$

$$[20]\ \frac{x \downarrow, y \downarrow}{x \cdot y \downarrow} \qquad [21]\ \frac{x \overset{r}{\mapsto} x'}{x \cdot y \overset{r}{\mapsto} x' \cdot y} \qquad [22]\ \frac{x \downarrow, y \overset{r}{\mapsto} y'}{x \cdot y \overset{r}{\mapsto} y'} \qquad [23]\ \frac{x \overset{r}{\mapsto} x', x' \downarrow, y \overset{s}{\mapsto} y'}{x \cdot y \overset{r+s}{\mapsto} y'}$$

Table 2. Deduction rules for BPA with relative real time ($a \in A, r > 0, p \geq 0$).

Definition 1 (σ-bisimilarity). *A symmetric, binary relation $R \subseteq \mathcal{P}(P) \times \mathcal{P}(P)$ is called a σ-bisimulation relation if for all $S, T \subseteq P$ with SRT we have*

- *for all $s \in S$: if $s \downarrow$ then there exists $t \in T$ such that $t \downarrow$;*
- *if $s \uparrow$ for all $s \in S$, then $t \uparrow$ for all $t \in T$;*

- *for all* $s \in S$ *and* $a \in A$*: if* $s \xrightarrow{a} s'$ *for some* $s' \in P$*, then there exists* $t \in T$ *such that* $t \xrightarrow{a} t'$ *for some* $t' \in P$ *and* $\{s'\}R\{t'\}$*;*
- *for all* $s \in S$ *and* $r > 0$*: if* $s \overset{r}{\mapsto} s'$ *for some* $s' \in P$*, then there exists* $t \in T$ *such that* $t \overset{r}{\mapsto} t'$ *for some* $t' \in P$ *and* $\{s' \mid \exists_{s \in S} s \overset{r}{\mapsto} s'\}R\{t' \mid \exists_{t \in T} t \overset{r}{\mapsto} t'\}$*;*

Two processes p *and* q *are* σ*-bisimilar, notation* $p \leftrightarrow_\sigma q$*, if there exists a* σ*-bisimulation relation* R *such that* $\{p\}R\{q\}$*.*

Although the above definition of σ-bisimilarity serves our purposes in identifying "equivalent" pocesses, we are not satisfied with it for several reasons. First, in σ-bisimulations not only processes have to be related, but also sets of processes. This is counterintuitive. Second, the definition of σ-bisimilarity does not allow to use the meta-theory for obtaining congruence and completeness results presented in [BV95]. With this formulation of equivalence of processes the proof that it is a congruence for the operators of the process algebra and the proof that the axiomatization of σ-bisimilarity is complete are long and tedious. In the process algebra BPA^{srt} there was no need for σ-bisimilarity; there strong bisimilarity (in the terminology of [BV95]) could be used.

Because of these drawbacks we looked for a different solution. Instead of bundling all outgoing time transitions into one time transition to the sum of the terms, we can also get bisimilar systems by *saturating* the transition systems, by putting in extra time transitions. The idea of saturating transition systems dates back to Bergstra and Klop [BK88] and Van Glabbeek [Gla87] where saturation with silent steps was used to turn weakly bisimilar transition systems into saturated strongly bisimilar systems. Here, we adapt this idea to turn σ-bisimilar systems into saturated strongly bisimilar systems. Stated more precisely, we just define time transitions without regard to the existence of other time transitions. Next, we saturate in two ways: whenever we have from one point two time transitions to two different terms, we add a time transition to the sum of the terms (deduction rule [35]), and whenever we have a time transition to a term that has different summands, we add a time transition to each of its summands (deduction rule [36]). On the resulting, saturated transition systems we can use strong bisimulation.

We extend the operational semantics with a relation $\prec$ which expresses that a process term is a syntactical summand of another process term (see Table 3). We claim that this relation is a partial order.

Definition 2 (Strong bisimilarity). *A symmetric, binary relation* $R \subseteq P \times P$ *is called a bisimulation relation if for all* $s, t \in P$ *such that* sRt *we have*

- *if* $s \downarrow$ *then* $t \downarrow$*;*
- *if* $s \uparrow$ *then* $t \uparrow$*;*
- *for all* $a \in A$ *and* $s' \in P$*: if* $s \xrightarrow{a} s'$*, then there exists* $t' \in P$ *such that* $t \xrightarrow{a} t'$ *and* $s'Rt'$*;*
- *for all* $r \in R_{\geq 0}$ *and* $s' \in P$*: if* $s \overset{r}{\mapsto} s'$*, then there exists* $t' \in P$ *such that* $t \overset{r}{\mapsto} t'$ *and* $s'Rt'$*;*
- *for all* $s' \in P$*: if* $s' \prec s$*, then there exists* $t' \in P$ *such that* $t' \prec t$ *and* $s'Rt'$*.*

$$[24]\ x \prec x \qquad [25]\ \dot{\delta} \prec x \qquad [26]\ \frac{x \not\downarrow}{\tilde{\tilde{\delta}} \prec x} \qquad [27]\ \frac{x \downarrow}{\dot{\epsilon} \prec x} \qquad [28]\ \frac{x \downarrow \quad x \not\downarrow}{\tilde{\tilde{\epsilon}} \prec x}$$

$$[29]\ \frac{x' \prec x}{x' \prec x + y} \qquad [30]\ \frac{y' \prec y}{y' \prec x + y} \qquad [31]\ \frac{x' \prec x \quad y' \prec y}{x' + y' \prec x + y}$$

$$[32]\ \frac{x \prec y}{x \cdot z \prec y \cdot z} \qquad [33]\ \frac{x \downarrow \quad y \prec z}{y \prec x \cdot z} \qquad [34]\ \frac{x \prec y}{\tilde{\tilde{\sigma}}^p \cdot x \prec \tilde{\tilde{\sigma}}^p \cdot y}$$

$$[35]\ \frac{x \overset{r}{\mapsto} x' \quad x \overset{r}{\mapsto} x''}{x \overset{r}{\mapsto} x' + x''} \qquad [36]\ \frac{x \overset{r}{\mapsto} x' \quad x'' \prec x'}{x \overset{r}{\mapsto} x''}$$

Table 3. Deduction rules for BPA with relative real time ($a \in A, r > 0, p \geq 0$).

Two processes p and q are bisimilar if there exists a bisimulation relation R such that pRq.

By saturating the transition systems, we have obtained that the time factorisation axiom SRT3 is valid on our transition systems. Thus, we can use the strong bisimulation relation induced by the relations and predicates of our term deduction systems (termination, virtual state, action execution, and time passage), and all the standard results on term deduction systems (such as the congruence property, and conservative extension results) become easily available. We feel this gives a definite advantage over the cumbersome to use notion of σ-bisimilarity.

Theorem 1. *For all closed $BPA_{\dot{\epsilon}}^{srt}$-terms p and q we have $p \underline{\leftrightarrow} q$ implies $p \underline{\leftrightarrow}_\sigma q$.*

Proof. See Appendix A.

Conjecture 1. For all closed $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$-terms p and q we have $p \underline{\leftrightarrow}_\sigma q$ implies $p \underline{\leftrightarrow} q$.

Theorem 2 (Equivalence). *Strong bisimilarity is an equivalence relation.*

Proof. We have to show that strong bisimilarity is reflexive, symmetric, and transitive. The first and second are trivial and therefore omitted. We prove that strong bisimilarity is transitive. Suppose that $R_1 : x \underline{\leftrightarrow} y$ and $R_2 : y \underline{\leftrightarrow} z$. Let $R = R_1 \circ R_2$. We prove that R is a strong bisimulation. Consider an arbitrary pair $(p, r) \in R$. As pRr we have the existence of q such that pR_1q and qR_2r.

- Suppose $p \downarrow$. As R_1 is a strong bisumlation and pR_1q we have $q \downarrow$. As R_2 is a strong bisimulation and qR_2r we have $r \downarrow$.
- Suppose $p \uparrow$. Similar to the previous case.
- Suppose $p \overset{a}{\rightarrow} p'$. As pR_1q we have the existence of q' such that $q \overset{a}{\rightarrow} q'$ and $p'R_1q'$. As qR_2r we have the existence of r' such that $r \overset{a}{\rightarrow} r'$ and $q'R_2r'$. Hence we also have the existence of r' such that $r \overset{a}{\rightarrow} r'$ and $p'Rr'$.
- Suppose $p \overset{t}{\mapsto} p'$. Similar to the previous case.

– Suppose $p' \prec p$. Similar to the previous case.

Definition 3. *We define the norm $|_|$ on closed terms as follows: for closed terms x and y, $a \in A_\delta$, and $p \in R_{\geq 0}$*

$$|\dot{\epsilon}| = \left|\dot{\delta}\right| = \left|\tilde{\tilde{\epsilon}}\right| = \left|\tilde{\tilde{\delta}}\right| = \left|\tilde{\tilde{a}}\right| = 1, \qquad |x \cdot y| = |x| \times |y| + 1,$$
$$\left|\tilde{\tilde{\sigma}}^p\right| = p + 1, \qquad |x + y| = |x| + |y|.$$

Theorem 3 (Congruence). *Strong bisimilarity is a congruence for the operators of the process algebra $BPA_{\dot{\epsilon}}^{srt}$.*

Proof. In order to prove that strong bisimulation is a congruence we conjecture that the term deduction system presented is isomorphic with the term deduction system obtained by omitting the delay constants and adding unary delay operators σ_{rel}^p with the following deduction rules:

$$\frac{x \downarrow}{\sigma_{\text{rel}}^0(x) \downarrow} \qquad \frac{x \uparrow}{\sigma_{\text{rel}}^0(x) \uparrow} \qquad \frac{x \xrightarrow{a} x'}{\sigma_{\text{rel}}^0(x) \xrightarrow{a} x'} \qquad \sigma_{\text{rel}}^{p+r}(x) \overset{r}{\mapsto} \sigma_{\text{rel}}^p(x)$$

$$\frac{x \overset{r'}{\mapsto} x'}{\sigma_{\text{rel}}^{r+r'}(x) \overset{r+r'}{\mapsto} (x')} \qquad \frac{x \prec y}{\sigma_{\text{rel}}^p(x) \prec \sigma_{\text{rel}}^p(y)}$$

Actually these deduction rules are obtained from the deduction rules of the original term deduction system by considering $\sigma_{\text{rel}}^p(x)$ and $\tilde{\tilde{\sigma}}^p \cdot x$ equivalent. The term deduction system constructed in this way is stratifiable and in panth format. Then, as a consequence we find that strong bisimularity is a congruence [Ver94]. The stratification is defined as follows:

$$\begin{aligned} S(x \downarrow) &= |x|, \\ S(x \uparrow) &= |x|, \\ S(x \xrightarrow{a} x') &= |x|, \\ S(x \overset{r}{\mapsto} x') &= 2\omega + |x|, \\ S(x \prec y) &= \omega + |y|. \end{aligned}$$

As an example, for the second deduction rule from Table 3 we have to prove that $S(x \uparrow) < S(\tilde{\tilde{\delta}} \prec x)$ for arbitrary closed term x. As $S(x \uparrow) = |x|$ and $S(\tilde{\tilde{\delta}} \prec x) = \omega + |x|$ this is the case.

As another example, for the last deduction rule from Table 3 we have to prove that $S(x \overset{r}{\mapsto} x') \leq S(x \overset{r}{\mapsto} x'')$ and $S(x'' \prec x') \leq S(x \overset{r}{\mapsto} x'')$. This is obviously the case.

2.3 Theoretical results

SOUNDNESS

Next, we define a notion of *basic terms*. These are useful in proofs.

Definition 4. *The set of all basic terms is the smallest set $\mathcal{B}$ that satisfies*

1. $\dot{\delta} \in \mathcal{B}$;
2. $\dot{\epsilon} \in \mathcal{B}$;
3. *for* $a \in A_\delta$ *and* $x \in \mathcal{B}$: $\tilde{\tilde{a}} \cdot x \in \mathcal{B}$;
4. *for* $p > 0$ *and* $x \in \mathcal{B}$: $\tilde{\tilde{\sigma}}^p \cdot x \in \mathcal{B}$;
5. *for* $x, y \in \mathcal{B}$: $x + y \in \mathcal{B}$.

Every closed term is derivably equal to a basic term. Using this, we can prove that $\mathrm{BPA}_\epsilon^{srt}$ is a complete axiomatization of strong bisimilarity on closed terms by induction on the structure of basic terms.

Theorem 4 (Elimination). *For every closed term s there exists a basic term t such that $s = t$.*

Proof. Easy; by induction on the structure of closed term s. We used the following lemma: for every two basic terms t_1 and t_2 there exists a basic term t_3 such that $t_1 \cdot t_2 = t_3$. This lemma is easily proven with induction on basic term t_1.

As a consequence of the above elimination theorem we also have that every closed term x can be written as

$$\sum_{i \in I} \tilde{\tilde{a}}_i \cdot x_i + \tilde{\tilde{\sigma}}^u \cdot x' + \sum_{j \in J} \dot{\epsilon},$$

where I and J are finite index sets, $a_i \in A_\delta$ for all $i \in I$, and x_i and x' are basic terms. This can easily be proven.

Theorem 5 (Completeness). *The process algebra BPA_ϵ^{srt} is a complete axiomatization of strong bisimilarity on closed BPA_ϵ^{srt} terms.*

Proof. For a proof of this theorem we refer to Appendix B.

3 Parallelism and communication

In this section, we define some auxiliary operators and extend the theory with parallel composition. The auxiliary operators help in the axiomatizations to come, but are also useful in their own right.

3.1 Encapsulating actions and time

First of all, we have the encapsulation operator ∂_H, well-known from standard process algebra. The operator ∂_H will block all actions from the set of actions H ($H \subseteq A$). Time can go forward, and actions outside H can occur. The operator is used to enforce communication between parallel components. The "now" operator ν_{rel} will block initial passage of time, forcing a process to start with an action at the current moment of time. The υ_{rel} is a time-out operator: in ${\upsilon_{\mathrm{rel}}}^p(x)$, the first action of x must occur within p time units from now.

$$[37]\ \frac{x \xrightarrow{a} x' \quad a \notin H}{\partial_H(x) \xrightarrow{a} \partial_H(x')} \qquad [38]\ \frac{x \downarrow}{\partial_H(x) \downarrow} \qquad [39]\ \frac{x \uparrow}{\partial_H(x) \uparrow} \qquad [40]\ \frac{x \overset{r}{\mapsto} x'}{\partial_H(x) \overset{r}{\mapsto} \partial_H(x')}$$

$$[41]\ \frac{x \prec y}{\partial_H(x) \prec \partial_H(y)} \qquad [42]\ \frac{x \xrightarrow{a} x'}{{\upsilon_{\mathrm{rel}}}^r(x) \xrightarrow{a} x'} \qquad [43]\ \frac{x \downarrow}{{\upsilon_{\mathrm{rel}}}^r(x) \downarrow} \qquad [44]\ {\upsilon_{\mathrm{rel}}}^0(x) \uparrow$$

$$[45]\ \frac{x \uparrow}{{\upsilon_{\mathrm{rel}}}^r(x) \uparrow} \qquad [46]\ \frac{x \overset{r}{\mapsto} x'}{{\upsilon_{\mathrm{rel}}}^{p+r}(x) \overset{r}{\mapsto} {\upsilon_{\mathrm{rel}}}^p(x')} \qquad [47]\ \frac{x \xrightarrow{a} x'}{\nu_{\mathrm{rel}}(x) \xrightarrow{a} x'} \qquad [48]\ \frac{x \downarrow}{\nu_{\mathrm{rel}}(x) \downarrow}$$

$$[49]\ \frac{x \uparrow}{\nu_{\mathrm{rel}}(x) \uparrow} \qquad [50]\ \frac{x \prec y}{{\upsilon_{\mathrm{rel}}}^r(x) \prec {\upsilon_{\mathrm{rel}}}^r(y)} \qquad [51]\ \frac{x \prec y}{\nu_{\mathrm{rel}}(x) \prec \nu_{\mathrm{rel}}(y)}$$

Table 4. Deduction rules for the auxiliary operators ($a \in A, r > 0, p \geq 0$).

$\partial_H(\dot{\delta}) = \dot{\delta}$	
$\partial_H(\dot{\epsilon}) = \dot{\epsilon}$	
$\partial_H(\tilde{\tilde{a}} \cdot x) = \tilde{\tilde{a}} \cdot \partial_H(x)$	if $a \notin H$
$\partial_H(\tilde{\tilde{a}} \cdot x) = \tilde{\tilde{\delta}}$	if $a \in H$
$\partial_H(\tilde{\tilde{\sigma}}^r \cdot x) = \tilde{\tilde{\sigma}}^r \cdot \partial_H(x)$	
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	
${\upsilon_{\mathrm{rel}}}^p(\dot{\delta}) = \dot{\delta}$	$\nu_{\mathrm{rel}}(\dot{\delta}) = \dot{\delta}$
${\upsilon_{\mathrm{rel}}}^r(\dot{\epsilon}) = \dot{\epsilon}$	$\nu_{\mathrm{rel}}(\dot{\epsilon}) = \dot{\epsilon}$
${\upsilon_{\mathrm{rel}}}^0(x) = \dot{\delta}$	
${\upsilon_{\mathrm{rel}}}^r(\tilde{\tilde{a}} \cdot x) = \tilde{\tilde{a}} \cdot x$	$\nu_{\mathrm{rel}}(\tilde{\tilde{a}} \cdot x) = \tilde{\tilde{a}} \cdot x$
${\upsilon_{\mathrm{rel}}}^{p+q}(\tilde{\tilde{\sigma}}^p \cdot x) = \tilde{\tilde{\sigma}}^p \cdot {\upsilon_{\mathrm{rel}}}^q(x)$	$\nu_{\mathrm{rel}}(\tilde{\tilde{\sigma}}^r \cdot x) = \tilde{\tilde{\delta}}$
${\upsilon_{\mathrm{rel}}}^p(x + y) = {\upsilon_{\mathrm{rel}}}^p(x) + {\upsilon_{\mathrm{rel}}}^p(y)$	$\nu_{\mathrm{rel}}(x + y) = \nu_{\mathrm{rel}}(x) + \nu_{\mathrm{rel}}(y)$

Table 5. Axioms of the auxiliary operators ($a \in A_\delta, p, q \geq 0, r > 0$).

Definition 5. *We extend the definition of $|_|$ on closed terms as follows: for closed terms x, $H \subseteq A$, and $p \in R_{\geq 0}$*

$$|\partial_H(x)| = |x|, \qquad |\upsilon_{rel}{}^p(x)| = |x|, \qquad |\nu_{rel}(x)| = |x|.$$

Theorem 6 (Congruence). *Strong bisimilarity is a congruence for the operators of the process algebra BPA_ϵ^{srt}.*

Proof. The term deduction is still stratifiable and in panth format. Therefore congruence follows immediately.

The axioms given in Table 5 are sound, the operators can be eliminated from every closed term,

Theorem 7 (Elimination). *For every closed term s there exists a basic term t such that $s = t$.*

Proof. It is easy to show that for every basic term t' there exists a basic term t such that $f(t') = t$, where f is one of the encapsulation operators ∂_H, $\upsilon_{\mathrm{rel}}{}^p$, or ν_{rel}. The proof is by induction on the structure of basic term t'. Then, these encapsulation operators can be eliminated from every closed term by applying this result repeatedly to an innermost occurrence of one of the encapsulation operators until no such operator remains. It is important to realize that any subterm without encapsulation operators is a closed $\mathrm{BPA}_\epsilon^{srt}$-term and hence is equal to a basic term using Theorem 4. This also applies to the term resulting after the last occurrence of an encapsulation operator has been replaced.

Theorem 8 (Completeness). *The process algebra presented in this section is a complete axiomatization of strong bisimilarity on closed terms.*

Proof. This theorem is proven using the meta-theory of [BV95]. Thereto, we first prove that the process algebra is an operationally conservative extension of the process algebra $\mathrm{BPA}_\epsilon^{srt}$. Using this, and some facts already proven, we obtain that it is also an equationally conservative extension. Completeness is then obtained by also using the elimination property.

First, we prove that the process algebra is an operationally conservative extension of the process algebra $\mathrm{BPA}_\epsilon^{srt}$. Both the term deduction systems of $\mathrm{BPA}_\epsilon^{srt}$ and the process algebra are pure, well-founded, and in panth format. This is sufficient to establish that the process algebra is an operationally conservative extension of $\mathrm{BPA}_\epsilon^{srt}$.

Then, suppose that $s = t$ for some closed $\mathrm{BPA}_\epsilon^{srt}$-terms s and t. Since the axioms are sound we have $s \leftrightarrows t$. By the fact that the process algebra is an operationally conservative extension we have $\mathrm{BPA}_\epsilon^{srt} \vdash s \leftrightarrows t$. Completeness of $\mathrm{BPA}_\epsilon^{srt}$ then gives $\mathrm{BPA}_\epsilon^{srt} \vdash s = t$. As the axioms of $\mathrm{BPA}_\epsilon^{srt}$ are contained in the axioms of the process algebra we have $s = t$. Hence we have proven that the process algebra is an equationally conservative extension of $\mathrm{BPA}_\epsilon^{srt}$.

Finally, to prove the completeness theorem, suppose that $s \leftrightarrows t$ for arbitrary closed terms s and t. By the elimination property we have the existence of

basic terms s' and t' such that $s = s'$ and $t = t'$. By the soundness of the axioms we have $s \leftrightarrows s'$ and $t \leftrightarrows t'$. Hence, since $\leftrightarrows$ is an equivalence we also have $s' \leftrightarrows t'$. Note that s' and t' are closed $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$-terms. Since $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ is a complete axiomatization of $\leftrightarrows$ on closed $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$-terms we obtain $\mathrm{BPA}_{\dot{\epsilon}}^{srt} \vdash s' = t'$. Since the process algebra is an equationally conservative extension of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$, we also have $s' = t'$. Combined with $s = s'$ and $t = t'$ this gives $s = t$.

There is an interesting relationship between the now operator and the time-out operators.

Theorem 9. *For $p \geq 0$, $r > 0$ and closed terms x we have*

1. $\upsilon_{rel}{}^{p}(x) + x = x$;
2. $\nu_{rel}(x) + \upsilon_{rel}{}^{r}(x) = \upsilon_{rel}{}^{r}(x)$;
3. $\nu_{rel}(x) + \upsilon_{rel}{}^{0}(x) = \nu_{rel}(x)$;
4. $x + \nu_{rel}(x) = x$.

Proof.

1. For $p = 0$ this is trivial: $\upsilon_{\mathrm{rel}}{}^{p}(x) + x = \upsilon_{\mathrm{rel}}{}^{0}(x) + x = \dot{\delta} + x = x$. For $p > 0$ the theorem is proven by induction on the structure of basic term x.
2. Easy, by induction on the structure of basic term x.
   - $x \equiv \dot{\epsilon}$. Trivial as $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \nu_{\mathrm{rel}}(\dot{\epsilon}) + \upsilon_{\mathrm{rel}}{}^{r}(\dot{\epsilon}) = \dot{\epsilon} + \dot{\epsilon} = \dot{\epsilon} = \upsilon_{\mathrm{rel}}{}^{r}(\dot{\epsilon}) = \upsilon_{\mathrm{rel}}{}^{r}(x)$.
   - $x \equiv \dot{\delta}$. Trivial as $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \nu_{\mathrm{rel}}(\dot{\delta}) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \dot{\delta} + \upsilon_{\mathrm{rel}}{}^{r}(x) = \upsilon_{\mathrm{rel}}{}^{r}(x)$.
   - $x \equiv \tilde{\tilde{a}} \cdot x'$ for some $a \in A_\delta$ and basic term x'. Then $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \nu_{\mathrm{rel}}(\tilde{\tilde{a}} \cdot x') + \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{a}} \cdot x') = \tilde{\tilde{a}} \cdot x' + \tilde{\tilde{a}} \cdot x' = \tilde{\tilde{a}} \cdot x' = \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{a}} \cdot x') = \upsilon_{\mathrm{rel}}{}^{r}(x)$.
   - $x \equiv \tilde{\tilde{\sigma}}^{q} \cdot x'$ for some $q > 0$ and basic term x'. If $r > q$ then $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \nu_{\mathrm{rel}}(\tilde{\tilde{\sigma}}^{q} \cdot x') + \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{q} \cdot x') = \tilde{\tilde{\delta}} + \tilde{\tilde{\sigma}}^{q} \cdot \upsilon_{\mathrm{rel}}{}^{r-q}(x') = \tilde{\tilde{\delta}} \cdot \upsilon_{\mathrm{rel}}{}^{r-q}(x') + \tilde{\tilde{\sigma}}^{q} \cdot \upsilon_{\mathrm{rel}}{}^{r-q}(x') = (\tilde{\tilde{\delta}} + \tilde{\tilde{\sigma}}^{q}) \cdot \upsilon_{\mathrm{rel}}{}^{r-q}(x') = \tilde{\tilde{\sigma}}^{q} \cdot \upsilon_{\mathrm{rel}}{}^{r-q}(x') = \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{q} \cdot x') = \upsilon_{\mathrm{rel}}{}^{r}(x)$. If $r \leq q$ then $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{q}(x) = \nu_{\mathrm{rel}}(\tilde{\tilde{\sigma}}^{q} \cdot x') + \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{q} \cdot x') = \tilde{\tilde{\delta}} + \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{r} \cdot \tilde{\tilde{\sigma}}^{q-r} \cdot x') = \tilde{\tilde{\delta}} + \tilde{\tilde{\sigma}}^{r} \cdot \upsilon_{\mathrm{rel}}{}^{0}(\tilde{\tilde{\sigma}}^{q-r} \cdot x') = \tilde{\tilde{\delta}} \cdot \upsilon_{\mathrm{rel}}{}^{0}(\tilde{\tilde{\sigma}}^{q-r} \cdot x') + \tilde{\tilde{\sigma}}^{r} \cdot \upsilon_{\mathrm{rel}}{}^{0}(\tilde{\tilde{\sigma}}^{q-r} \cdot x') = (\tilde{\tilde{\delta}} + \tilde{\tilde{\sigma}}^{r}) \cdot \upsilon_{\mathrm{rel}}{}^{0}(\tilde{\tilde{\sigma}}^{q-r} \cdot x') = \tilde{\tilde{\sigma}}^{r} \cdot \upsilon_{\mathrm{rel}}{}^{0}(\tilde{\tilde{\sigma}}^{q-r} \cdot x') = \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{r} \cdot \tilde{\tilde{\sigma}}^{q-r} \cdot x') = \upsilon_{\mathrm{rel}}{}^{r}(\tilde{\tilde{\sigma}}^{q} \cdot x') = \upsilon_{\mathrm{rel}}{}^{r}(x)$.
   - $x \equiv x_1 + x_2$ for some basic terms x_1, x_2. By induction we have $\nu_{\mathrm{rel}}(x_1) + \upsilon_{\mathrm{rel}}{}^{r}(x_1) = \upsilon_{\mathrm{rel}}{}^{r}(x_1)$ and $\nu_{\mathrm{rel}}(x_2) + \upsilon_{\mathrm{rel}}{}^{r}(x_2) = \upsilon_{\mathrm{rel}}{}^{r}(x_2)$. Hence $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{r}(x) = \nu_{\mathrm{rel}}(x_1 + x_2) + \upsilon_{\mathrm{rel}}{}^{r}(x_1 + x_2) = \nu_{\mathrm{rel}}(x_1) + \upsilon_{\mathrm{rel}}{}^{r}(x_1) + \nu_{\mathrm{rel}}(x_2) + \upsilon_{\mathrm{rel}}{}^{r}(x_2) = \upsilon_{\mathrm{rel}}{}^{r}(x_1) + \upsilon_{\mathrm{rel}}{}^{r}(x_2) = \upsilon_{\mathrm{rel}}{}^{r}(x_1 + x_2) = \upsilon_{\mathrm{rel}}{}^{r}(x)$.
3. $\nu_{\mathrm{rel}}(x) + \upsilon_{\mathrm{rel}}{}^{0}(x) = \nu_{\mathrm{rel}}(x) + \dot{\delta} = \nu_{\mathrm{rel}}(x)$.
4. $x = x + \upsilon_{\mathrm{rel}}{}^{r}(x) = x + \upsilon_{\mathrm{rel}}{}^{r}(x) + \nu_{\mathrm{rel}}(x) = x + \nu_{\mathrm{rel}}(x)$.

3.2 Parallel composition operator

Now we have all machinery in place in order to define parallel composition. The parallel composition of two processes x and y, notation $x \parallel y$, describes the interleaving of their behaviour. An action can be executed by $x \parallel y$ iff one of the

operands can execute this action or this action is the result of the simultaneous execution of an action from x and an action from y. This last possibility is usually called communication. The possible communications are specified by the *communication* function $\gamma : A \times A \to A$. This function is a parameter of the process algebra and can be chosen dependent on the application. The function γ is partial, commutative and associative. We extend γ to a total function γ^* as follows: for $a, b \in A$

$$\begin{array}{lll} \gamma^*(a,\delta) & = \delta, & \\ \gamma^*(\delta,a) & = \delta, & \\ \gamma^*(\delta,\delta) & = \delta, & \\ \gamma^*(a,b) & = \gamma(a,b) & \text{if } \gamma(a,b) \text{ is defined,} \\ \gamma^*(a,b) & = \delta & \text{if } \gamma(a,b) \text{ is not defined.} \end{array}$$

In the sequel we will write γ^* as γ.

The process $x \parallel y$ can let time pass if both operands can let time pass or if only one of them can let time pass and the other process has no other option than immediate termination.

Besides the parallel composition operator we also present the left-merge operator $\lfloor\!\lfloor$ and communication merge operator $|$. We present the operational rules in Tables 6 and 7, and present additional rules for the saturation of time steps. These auxiliary operators are needed in order to axiomatise parallel composition. The process $x \lfloor\!\lfloor y$ behaves as the process $x \parallel y$ except that the first action to be executed must be from process x. The process $x \mid y$ also behaves as the process $x \parallel y$ except that this time the first action must be a communication. This intuition of the operators explains the main axiom of parallel composition: $x \parallel y = (x \lfloor\!\lfloor y + y \lfloor\!\lfloor x) + x \mid y$.

$$[52]\ \frac{x \xrightarrow{a} x' \quad y \not\uparrow}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad [53]\ \frac{x \not\uparrow \quad y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} \qquad [54]\ \frac{x \xrightarrow{a} x' \quad y \uparrow \quad y \downarrow}{x \parallel y \xrightarrow{a} x'}$$

$$[55]\ \frac{x \uparrow \quad x \downarrow \quad y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} y'} \qquad [56]\ \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y' \quad \gamma(a,b) = c}{x \parallel y \xrightarrow{c} x' \parallel y'}$$

$$[57]\ \frac{x \overset{r}{\mapsto} x' \quad y \overset{r}{\mapsto} y'}{x \parallel y \overset{r}{\mapsto} x' \parallel y'} \qquad [58]\ \frac{x \overset{r}{\mapsto} x' \quad y \downarrow \quad y \uparrow}{x \parallel y \overset{r}{\mapsto} x'} \qquad [59]\ \frac{x \downarrow \quad x \uparrow \quad y \overset{r}{\mapsto} y'}{x \parallel y \overset{r}{\mapsto} y'}$$

$$[60]\ \frac{x \downarrow \quad y \downarrow}{x \parallel y \downarrow} \qquad [61]\ \frac{x \uparrow \quad x \not\downarrow}{x \parallel y \uparrow} \qquad [62]\ \frac{y \uparrow \quad y \not\downarrow}{x \parallel y \uparrow} \qquad [63]\ \frac{x \uparrow \quad y \uparrow}{x \parallel y \uparrow}$$

$$[64]\ \frac{x \prec y \lfloor\!\lfloor z}{x \prec y \parallel z} \qquad [65]\ \frac{x \prec y \lfloor\!\lfloor z}{x \prec z \parallel y} \qquad [66]\ \frac{x \prec y \mid z}{x \prec y \parallel z}$$

Table 6. Deduction rules for parallel composition ($a, b, c \in A, r > 0$).

$$[67]\ \frac{x \xrightarrow{a} x' \quad y \not\uparrow}{x \lfloor\!\lfloor y \xrightarrow{a} x' \parallel y} \qquad [68]\ \frac{x \xrightarrow{a} x' \quad y \uparrow \quad y \downarrow}{x \lfloor\!\lfloor y \xrightarrow{a} x'} \qquad [69]\ \frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y' \quad \gamma(a,b) = c}{x \mid y \xrightarrow{c} x' \parallel y'}$$

$$[70]\ \frac{x \overset{r}{\mapsto} x' \quad y \overset{r}{\mapsto} y'}{x \lfloor\!\lfloor y \overset{r}{\mapsto} x' \parallel y'} \qquad [71]\ \frac{x \overset{r}{\mapsto} x' \quad y \downarrow \quad y \uparrow}{x \lfloor\!\lfloor y \overset{r}{\mapsto} x'} \qquad [72]\ \frac{x \overset{r}{\mapsto} x' \quad y \overset{r}{\mapsto} y'}{x \mid y \overset{r}{\mapsto} x' \parallel y'}$$

$$[73]\ \frac{x \downarrow \quad y \downarrow}{x \lfloor\!\lfloor y \downarrow} \qquad [74]\ \frac{x \uparrow \quad x \not\downarrow}{x \lfloor\!\lfloor y \uparrow} \qquad [75]\ \frac{y \uparrow \quad y \not\downarrow}{x \lfloor\!\lfloor y \uparrow} \qquad [76]\ \frac{x \uparrow \quad y \uparrow}{x \lfloor\!\lfloor y \uparrow} \qquad [77]\ \frac{x \uparrow}{x \mid y \uparrow}$$

$$[78]\ \frac{y \uparrow}{x \mid y \uparrow} \qquad [79]\ \frac{x \prec y}{x \lfloor\!\lfloor z \prec y \lfloor\!\lfloor z} \qquad [80]\ \frac{x \prec y}{x \mid z \prec y \mid z} \qquad [81]\ \frac{x \prec y}{z \mid x \prec z \mid y}$$

Table 7. Deduction rules for auxiliary operators for parallelism ($a, b, c \in A, r > 0$).

Strong bisimilarity (Definition 2) is a congruence for the operators of the process algebra $\mathrm{ACP}^{srt}_{\dot{\epsilon}}$.

Definition 6. *We extend the definition of $|_|$ on closed terms as follows: for closed terms x and y*

$$|x \parallel y| = |x \lfloor\!\lfloor y| = |x \mid y| = |x| + |y|.$$

Theorem 10 (Congruence). *Strong bisimilarity is a congruence for the operators of the process algebra $ACP^{srt}_{\dot{\epsilon}}$.*

Proof. The term deduction is still stratifiable and in panth format. Therefore congruence follows immediately.

In Table 8 we present axioms for the operators introduced in this section. As explained before parallel composition is expressed in terms of the auxiliary operators. Next, we will explain these.

The axioms for the communication merge operator are based on the structure of basic terms. Note that the two processes involved in the communication merge have to let time pass simultaneously (Axiom ...).

For the left-merge operator the situation is more complex. For the left operand we distinguish cases based on the structure of basic terms. For the case $\tilde{\tilde{a}} \cdot x$ we additionaly make assumptions about the structure of the right operand. The following cases are distinguished: $\dot{\epsilon}$, $\dot{\delta}$, and $y + \tilde{\tilde{\delta}}$ for arbitrary y. In the first case the right operand has already terminated and hence the right operand is on its own. In the second case obviously an immediate deadlock results. In the third case the right argument is not a virtual state and hence the action a can be executed. The following lemma indicates that every basic term can be written in one of the forms discussed above.

Lemma 1 (Representation). *For every basic term x we have $x = \dot{\epsilon}$ or $x = \dot{\delta}$ or $x = x + \tilde{\tilde{\delta}}$.*

Proof. Easy by induction on the structure of basic term x.

If the left operand is of the form $\tilde{\tilde{\sigma}}^r \cdot x$, we must determine whether the right operand can also let time progress. Four cases can be distinguished: $\dot{\epsilon}$, $\dot{\delta}$, $\nu_{\mathrm{rel}}(y) + \tilde{\tilde{\delta}}$ and $\upsilon_{\mathrm{rel}}{}^r(y) + \tilde{\tilde{\sigma}}^r \cdot z$. The first two cases are treated as before. The third case expresses that the right operand cannot let time pass and hence a deadlock remains. The last case expresses that the right operand can at least let r time pass.

BLA BLA BLA BLA

We claim that these axioms are sound.

$x \parallel y = (x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x) + x \mid y$		
$\dot{\delta} \mathbin{\lfloor\!\lfloor} x = \dot{\delta}$	$\dot{\delta} \mid x = \dot{\delta}$	
$x \mathbin{\lfloor\!\lfloor} \dot{\delta} = \dot{\delta}$	$x \mid \dot{\delta} = \dot{\delta}$	
$\dot{\epsilon} \mathbin{\lfloor\!\lfloor} (x + y) = \dot{\epsilon} \mathbin{\lfloor\!\lfloor} x + \dot{\epsilon} \mathbin{\lfloor\!\lfloor} y$	$\dot{\epsilon} \mid x = \dot{\delta}$	
$\dot{\epsilon} \mathbin{\lfloor\!\lfloor} \tilde{\tilde{a}} \cdot x = \dot{\delta}$	$x \mid \dot{\epsilon} = \dot{\delta}$	
$\dot{\epsilon} \mathbin{\lfloor\!\lfloor} \tilde{\tilde{\sigma}}^r \cdot x = \dot{\delta}$	$\tilde{\tilde{a}} \mid \tilde{\tilde{b}} = \tilde{\tilde{c}}$	if $\gamma(a, b) = c$
$x \mathbin{\lfloor\!\lfloor} \dot{\epsilon} = x$	$\tilde{\tilde{a}} \mid \tilde{\tilde{b}} = \tilde{\tilde{\delta}}$	if $\gamma(a, b)$ is undefined
$(x + y) \mathbin{\lfloor\!\lfloor} z = x \mathbin{\lfloor\!\lfloor} z + y \mathbin{\lfloor\!\lfloor} z$	$x \mid (y + z) = x \mid y + x \mid z$	
	$(x + y) \mid z = x \mid z + y \mid z$	
$\tilde{\tilde{a}} \cdot x \mathbin{\lfloor\!\lfloor} (y + \tilde{\tilde{\delta}}) = \tilde{\tilde{a}} \cdot (x \parallel (y + \tilde{\tilde{\delta}}))$	$\tilde{\tilde{a}} \cdot x \mid \tilde{\tilde{b}} \cdot y = (\tilde{\tilde{a}} \mid \tilde{\tilde{b}}) \cdot (x \parallel y)$	
$\tilde{\tilde{\sigma}}^r \cdot x \mathbin{\lfloor\!\lfloor} (\nu_{\mathrm{rel}}(y) + \tilde{\tilde{\delta}}) = \tilde{\tilde{\delta}}$	$\tilde{\tilde{\sigma}}^r \cdot x \mid \tilde{\tilde{b}} \cdot y = \tilde{\tilde{\delta}}$	
$\tilde{\tilde{\sigma}}^p \cdot x \mathbin{\lfloor\!\lfloor} (\upsilon_{\mathrm{rel}}{}^p(y) + \tilde{\tilde{\sigma}}^p \cdot z) = \tilde{\tilde{\sigma}}^p \cdot (x \mathbin{\lfloor\!\lfloor} z)$	$\tilde{\tilde{a}} \cdot x \mid \tilde{\tilde{\sigma}}^r \cdot y = \tilde{\tilde{\delta}}$	
	$\tilde{\tilde{\sigma}}^p \cdot x \mid \tilde{\tilde{\sigma}}^p \cdot y = \tilde{\tilde{\sigma}}^p \cdot (x \mid y)$	

Table 8. Axioms of $\mathrm{ACP}^{srt}_{\dot{\epsilon}}$ $(a, b, c \in A_\delta,\ p, q \geq 0,\ r > 0)$.

For understanding the axiomatisation of $\mathbin{\lfloor\!\lfloor}$ it is necessary to realize that every closed term x can be written as $\dot{\epsilon}$, as $\nu_{\mathrm{rel}}(x) + \tilde{\tilde{\delta}}$ or as $\upsilon_{\mathrm{rel}}{}^p(x) + \tilde{\tilde{\sigma}}^p \cdot x'$ for some $p > 0$ and closed term x'.

Next, we will take some steps towards a result that states that the parallel composition operator (and all auxiliary operators) can be eliminated from closed terms. The specific form of the axioms for left merge require that each basic term can be written in a certain format. Below we present this result.

Theorem 11 (Representation). *For basic term x we have $x = \nu_{rel}(x)$ or there exists $r > 0$ such that there exists basic term x' such that $x = \upsilon_{rel}{}^r(x) + \tilde{\tilde{\sigma}}^r \cdot x'$ and $|x'| < |x|$.*

Proof. For a proof of this theorem we refer to Appendix C.

Theorem 12 (Elimination). *For every closed term x there exists a basic term y such that $x = y$.*

Proof. See Appendix D.

It is not hard to prove that the process algebra $\mathrm{ACP}_{\dot{\epsilon}}^{srt}$ is a conservative extension of the process algebra $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$. Combining this result with the elimination theorem and the completeness of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ gives that $\mathrm{ACP}_{\dot{\epsilon}}^{srt}$ is complete.

Theorem 13 (Completeness). *The process algebra $ACP_{\dot{\epsilon}}^{srt}$ is a complete exiomatization of strong bisimilarity on closed $ACP_{\dot{\epsilon}}^{srt}$-terms.*

Proof. The proof of completeness of $\mathrm{ACP}_{\dot{\epsilon}}^{srt}$ is similar to the proof of completeness of the process algebra ... (Theorem 8).

As a last part of this section we present some basic results about parallel composition that can be derived with the theory at hand.

Theorem 14. *For closed terms x and y we have*

1. $\dot{\epsilon} \parallel x = x \parallel \dot{\epsilon} = x$
2. $\dot{\delta} \parallel x = x \parallel \dot{\delta} = \dot{\delta}$
3. $x \parallel y = y \parallel x$
4. $(x \parallel y) \parallel z = x \parallel (y \parallel z)$

Proof. The first item is proven easily with induction on the structure of basic term x. The second item follows directly from the axioms as follows: $\dot{\delta} \parallel x = \dot{\delta} \mathbin{\lfloor\!\lfloor} x + x \mathbin{\lfloor\!\lfloor} \dot{\delta} + \dot{\delta} \mid x = \dot{\delta} + \dot{\delta} + \dot{\delta} = \dot{\delta}$ and $x \parallel \dot{\delta} = x \mathbin{\lfloor\!\lfloor} \dot{\delta} + \dot{\delta} \mathbin{\lfloor\!\lfloor} x + x \mid \dot{\delta} = \dot{\delta} + \dot{\delta} + \dot{\delta} = \dot{\delta}$. For the third item we prove the propositions $x \mid y = y \mid x$ and $x \parallel y = y \parallel x$ simultaneously by induction on $|x| + |y|$. For the first proposition we additionaly use case distinction on the structure of the basic terms x and y. The second proposition then follows easily: $x \parallel y = x \mathbin{\lfloor\!\lfloor} y + y \mathbin{\lfloor\!\lfloor} x + x \mid y = y \mathbin{\lfloor\!\lfloor} x + x \mathbin{\lfloor\!\lfloor} y + y \mid x = y \parallel x$. The proof of associativity of parallel composition is long and tedious, see Appendix E.

4 Relation with other basic process algebras

In this section, we consider the relation of the process algebra $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ with other process algebras. As means to compare different process algebras we consider the notions of conservative extensions, embeddings, and SRM-specifications.

4.1 Comparing process algebras

A process algebra $T' = (\Sigma', E')$ is a *conservative extension* of the process algebra $T = (\Sigma, E)$ if (1) $\Sigma \subseteq \Sigma'$ and (2) for all closed terms s and t over the signature Σ: $(\Sigma, E) \vdash s = t$ if and only if $(\Sigma', E') \vdash s = t$. This is an interesting notion of comparing process algebras as it says that the smaller process algebra is contained precisely in the larger process algebra. No identities are lost and none are added.

An *explicit definition* of an operator or constant f in the process algebra (Σ, E) is an equation $f(x_1, \cdots, x_n) = t$ where t is a term over the signature

Σ that does not contain other free variables than $x_1, \cdots, x_n$. An extension of (Σ, E) with only explicitly defined operators and constants is called a *definitional extension* of (Σ, E). The following theorem states that a definitional extension is a special type of conservative extension.

Theorem 15. *If the process algebra T' is a definitional extension of the process algebra T, then T' is a conservative extension of T.*

An *embedding* of a process algebra $T = (\Sigma, E)$ in a process algebra $T' = (\Sigma', E')$ is a term structure preserving injective mapping $\hbar$ from the terms of T to the terms of T' such that for all closed terms s, t of T, $T \vdash s = t$ implies $T' \vdash \hbar(s) = \hbar(t)$.

Theorem 16 (Baeten and Middelburg). *Let $T = (\Sigma, E)$ and $T' = (\Sigma', E')$ be process algebras such that all operators from the signature Σ that are not in the signature Σ' can be defined in T' by explicit definitions. Let $T'' = (\Sigma'', E'')$ be the resulting definitional extension of T'. If the axioms of T are derivable for closed terms in T'', then T can be embedded in T'.*

4.2 Without empty processes

First we compare the process algebra $\mathrm{BPA}_{\epsilon}^{srt}$ with the fragment of the process algebra BPA^{srt} from [BM99] where we omit the relative time-out and relative initialization operators. The signature of this fragment consists of the urgent actions, urgent deadlock, immediate deadlock, alternative and sequential composition, and the unary relative delay operator σ^p_{rel}. The axioms of this fragment of BPA^{srt} are the axioms A1-A5, A6ID-A7ID, A6SRa, and A7SR from Table 1 and additionally the axioms SRT1-SRT4 and A6SRb given below.

$\sigma^0_{\mathrm{rel}}(x) = x$	SRT1
$\sigma^p_{\mathrm{rel}}(\sigma^q_{\mathrm{rel}}(x)) = \sigma^{p+q}_{\mathrm{rel}}(x)$	SRT2
$\sigma^p_{\mathrm{rel}}(x) + \sigma^p_{\mathrm{rel}}(y) = \sigma^p_{\mathrm{rel}}(x+y)$	SRT3
$\sigma^p_{\mathrm{rel}}(x) \cdot y = \sigma^p_{\mathrm{rel}}(x \cdot y)$	SRT4
$\sigma^r_{\mathrm{rel}}(x) + \tilde{\tilde{\delta}} = \sigma^r_{\mathrm{rel}}(x)$	A6SRb

The relative delay operator σ^p_{rel} of BPA^{srt} can be defined explicitly on $\mathrm{BPA}_{\epsilon}^{srt}$ by the equation $\sigma^p_{\mathrm{rel}}(x) = \tilde{\tilde{\sigma}}^p \cdot x$. We call the resulting definitional extension $\mathrm{BPA}_{\epsilon}^{srt}+\sigma^p_{\mathrm{rel}}$. For closed terms over the signature of $\mathrm{BPA}_{\epsilon}^{srt}+\sigma^p_{\mathrm{rel}}$, the axioms of BPA^{srt} are derivable from the axioms of $\mathrm{BPA}_{\epsilon}^{srt}+\sigma^p_{\mathrm{rel}}$.

Theorem 17. *For closed terms over the signature of $BPA_{\epsilon}^{srt}+\sigma^p_{rel}$, the axioms of BPA^{srt} are derivable from the axioms of $BPA_{\epsilon}^{srt}+\sigma^p_{rel}$.*

Proof. We have to consider each of the axioms of BPA^{srt}. The axioms A1-A5, A6ID-A7ID, A6SRa, and A7SR are also axioms of $\mathrm{BPA}_{\epsilon}^{srt}$. For the other axioms

we have the following derivations:

$$\begin{array}{l}
\sigma^0_{\rm rel}(x) = \tilde{\tilde{\sigma}}^0 \cdot x = \dot{\epsilon} \cdot x = x \\
\sigma^p_{\rm rel}(\sigma^q_{\rm rel}(x)) = \tilde{\tilde{\sigma}}^p \cdot (\tilde{\tilde{\sigma}}^q \cdot x) = (\tilde{\tilde{\sigma}}^p \cdot \tilde{\tilde{\sigma}}^q) \cdot x = \tilde{\tilde{\sigma}}^{p+q} \cdot x = \sigma^{p+q}_{\rm rel}(x) \\
\sigma^p_{\rm rel}(x) + \sigma^p_{\rm rel}(y) = \tilde{\tilde{\sigma}}^p \cdot x + \tilde{\tilde{\sigma}}^p \cdot y = \tilde{\tilde{\sigma}}^p \cdot (x+y) = \sigma^p_{\rm rel}(x+y) \\
\sigma^p_{\rm rel}(x) \cdot y = (\tilde{\tilde{\sigma}}^p \cdot x) \cdot y = \tilde{\tilde{\sigma}}^p \cdot (x \cdot y) = \sigma^p_{\rm rel}(x \cdot y) \\
\sigma^r_{\rm rel}(x) + \tilde{\tilde{\delta}} = \tilde{\tilde{\sigma}}^r \cdot x + \tilde{\tilde{\delta}} = \tilde{\tilde{\sigma}}^r \cdot x + \tilde{\tilde{\delta}} \cdot x = (\tilde{\tilde{\sigma}}^r + \tilde{\tilde{\delta}}) \cdot x = \tilde{\tilde{\sigma}}^r \cdot x = \sigma^r_{\rm rel}(x)
\end{array}$$

Then, as a consequence of Theorem 16, we have that BPA^{srt} can be embedded in $\mathrm{BPA}^{srt}_{\dot{\epsilon}}$ using $\hbar$ defined by $\hbar(\dot{\delta}) = \dot{\delta}$, $\hbar(\tilde{\tilde{a}}) = \tilde{\tilde{a}}$, $\hbar(x_1 + x_2) = \hbar(x_1) + \hbar(x_2)$, $\hbar(x_1 \cdot x_2) = \hbar(x_1) \cdot \hbar(x_2)$, $\hbar(\sigma^p_{\rm rel}(x)) = \tilde{\tilde{\sigma}}^p \cdot \hbar(x)$.

4.3 Initial delay constants

In this section, we will introduce the initial delay constants $\tilde{\tilde{\sigma}}^{[0,p)}$ for all $p \in R_{\geq 0}$. The constant $\tilde{\tilde{\sigma}}^{[0,p)}$ describes an arbitrary delay in the time interval $[0,p)$. If $p = \infty$ we obtain the so-called time iteration constant also denoted as ϵ.

For the purpose of embedding the time-free theory $\mathrm{BPA}_{\delta\epsilon}$ in the standard real time process algebra we need the time iteration constant. However, only introducing the time iteration constant does not suffice. In that case the timeout operator cannot be eliminated from closed terms anymore.

[82] $\tilde{\tilde{\sigma}}^{[0,r)} \downarrow$	[83] $\tilde{\tilde{\sigma}}^{[0,0)} \uparrow$	[84] $\tilde{\tilde{\sigma}}^{[0,r+r')} \overset{r}{\mapsto} \tilde{\tilde{\sigma}}^{[0,r')}$
[85] $\tilde{\tilde{\sigma}}^{p} \prec \tilde{\tilde{\sigma}}^{[0,p+r)}$	[86] $\tilde{\tilde{\sigma}}^{[0,p)} \prec \tilde{\tilde{\sigma}}^{[0,p+r)}$	

Table 9. Deduction rules for initial delay constants ($r, r' > 0, p \geq 0$).

$\tilde{\tilde{\sigma}}^{[0,0)} = \dot{\delta}$	$\tilde{\tilde{\sigma}}^{[0,p)} \cdot \tilde{\tilde{\delta}} + \tilde{\tilde{\sigma}}^p \cdot \dot{\delta} = \tilde{\tilde{\sigma}}^p \cdot \dot{\delta}$
$\partial_H(\tilde{\tilde{\sigma}}^{[0,p)} \cdot x) = \tilde{\tilde{\sigma}}^{[0,p)} \cdot \partial_H(x)$	
$\tilde{\tilde{\sigma}}^{[0,p+q)} + \tilde{\tilde{\sigma}}^{[0,p)} = \tilde{\tilde{\sigma}}^{[0,p+q)}$	$\tilde{\tilde{\sigma}}^{[0,p+r)} + \tilde{\tilde{\sigma}}^p = \tilde{\tilde{\sigma}}^{[0,p+r)}$
$\tilde{\tilde{\sigma}}^{[0,p)} \cdot x + \tilde{\tilde{\sigma}}^{[0,p)} \cdot y = \tilde{\tilde{\sigma}}^{[0,p)} \cdot (x+y)$	$\upsilon_{\rm rel}{}^{p+r}(\tilde{\tilde{\sigma}}^{[0,r)} \cdot \nu_{\rm rel}(x)) = \tilde{\tilde{\sigma}}^{[0,r)} \cdot \nu_{\rm rel}(x)$
$\tilde{\tilde{\sigma}}^{[0,p)} \cdot \tilde{\tilde{\sigma}}^{[0,q)} = \tilde{\tilde{\sigma}}^{[0,p+q)}$	$\nu_{\rm rel}(\tilde{\tilde{\sigma}}^{[0,r)} \cdot x) = \nu_{\rm rel}(x)$

Table 10. Axioms of generalised delay constants ($p, q \geq 0, r > 0$).

4.4 Time-free

UITLEG $\mathrm{BPA}_{\delta\epsilon}$ + AXIOMAS

A6	$x + \delta = x$
A7	$\delta \cdot x = x$
A8	$x \cdot \epsilon = x$
A9	$\epsilon \cdot x = x$

There are several embeddings of the time-free theory $\mathrm{BPA}_{\delta\epsilon}$ into $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$. First, we have the embedding where $a = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot \tilde{\tilde{a}}$, $\delta = \dot{\delta}$, and $\epsilon = \dot{\epsilon}$. This embedding however does not correspond to our intuition that the deadlock constant from untimed process algebra is delayable.

The more interesting embedding is the following: $a = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot \tilde{\tilde{a}} \cdot \tilde{\tilde{\sigma}}^{[0,\infty)}$, $\epsilon = \tilde{\tilde{\sigma}}^{[0,\infty)}$ and $\delta = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot \dot{\delta}$. Alternatively, the embedding can be described as:

$$\begin{aligned} \epsilon &= \tilde{\tilde{\sigma}}^{[0,\infty)} \\ \delta &= \epsilon \cdot \dot{\delta} \\ a &= \epsilon \cdot \tilde{\tilde{a}} \cdot \epsilon \end{aligned}$$

We can easily derive the axioms of $\mathrm{BPA}_{\delta\epsilon}$ for closed terms as follows. The axioms A1-A5 are also part of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$ and need therefore not be considered. Axiom A7 is easily derived as follows:

$$\delta \cdot x = (\epsilon \cdot \dot{\delta}) \cdot x = \epsilon \cdot (\dot{\delta} \cdot x) = \epsilon \cdot \dot{\delta} = \delta$$

For the axioms A8 and A9 we use induction on the structure of closed $\mathrm{BPA}_{\delta\epsilon}$-terms. We present the two proofs simultaneously:

- $x \equiv \epsilon$. Then $x \cdot \epsilon = \epsilon \cdot \epsilon = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot \tilde{\tilde{\sigma}}^{[0,\infty)} = \tilde{\tilde{\sigma}}^{[0,\infty)} = \epsilon = x$ and $\epsilon \cdot x = \epsilon \cdot \epsilon = \epsilon = x$.
- $x \equiv \delta$. Then $x \cdot \epsilon = \delta \cdot \epsilon = \epsilon \cdot \dot{\delta} \cdot \epsilon = \epsilon \cdot \dot{\delta} = \delta$ and $\epsilon \cdot x = \epsilon \cdot \delta = \epsilon \cdot \epsilon \cdot \dot{\delta} = \epsilon \cdot \dot{\delta} = \delta = x$.
- $x \equiv x_1 \cdot x_2$. Then $x \cdot \epsilon = x_1 \cdot x_2 \cdot \epsilon = x_1 \cdot x_2 = x$ and $\epsilon \cdot x = \epsilon \cdot x_1 \cdot x_2 = x_1 \cdot x_2 = x$.
- $x \equiv x_1 + x_2$. Then $x \cdot \epsilon = (x_1 + x_2) \cdot \epsilon = x_1 \cdot \epsilon + x_2 \cdot \epsilon = x_1 + x_2 = x$ and $\epsilon \cdot x = \epsilon \cdot (x_1 + x_2) = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot (x_1 + x_2) = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot x_1 + \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot x_2 = \epsilon \cdot x_1 + \epsilon \cdot x_2 = x_1 + x_2 = x$.

Using the above, we easily obtain derivability of axiom A6 for closed $\mathrm{BPA}_{\delta\epsilon}$-terms:

$$x + \delta = \epsilon \cdot x + \epsilon \cdot \dot{\delta} = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot x + \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot \dot{\delta} = \tilde{\tilde{\sigma}}^{[0,\infty)} \cdot (x + \dot{\delta}) = \epsilon \cdot x = x.$$

5 Conclusion

Using our theory, we can replace the delay operators of timed ACP by delay constants, thus simplifying axioms and calculations. We have a clear separation of action execution and termination in our operational rules. All states in transition systems correspond to process terms. In order to ensure that the axiom of time factorisation holds in our operational model, we use the technique of *saturation* of transition systems. This technique was introduced for silent steps in [BK88,Gla87]; here, we use saturation of time steps. This avoids negative premises in our rules, avoids the use of bisimulations relating sets of states, and allows to use the results that are based on the format of the SOS rules. This is an improvement in the treatment of time steps.

We can embed the untimed theory in the timed theory. We considered two ways of doing this. Preferably, we would have liked an embedding where $a = \epsilon \cdot \tilde{\tilde{a}}$. This is incompatible with the law $x \cdot \epsilon = x$. We believe that this embedding can be achieved if we redesign the basic process algebras BPA_δ and $\mathrm{BPA}_{\delta\epsilon}$ [Bae00].

In this paper we have not discussed the extension of discrete time process algebras with the empty process. However, this can be easily achieved on the basis of the material presented. The combination of the empty process and discrete, relative time process algebra has been studied in [Ver97,BV97]. Among the conclusions in this work are the following two: the empty process can not be straightforwardly be combined with the immediate deadlock process of [BB96,BB97,BBR99], and the behaviour of the empty process is not always into accordance with one's first intuition. In this paper we present a combination of the empty process with standard real-time process algebra in relative timing with immediate deadlock in which the behaviour of the empty process is clear in all cases.

References

[Bae00] J.C.M. Baeten. Process algebra with explicit termination. Draft, 2000.

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

[BB96] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. *Formal Aspects of Computing*, 8(2):188–208, 1996.

[BB97] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra: absolute time, relative time and parametric time. *Fundamenta Informatica*, 29:51–76, 1997.

[BBR99] J.C.M. Baeten, J.A. Bergstra, and M.A. Reniers. The silent step in time. In G.D. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1999. To appear.

[BFM96] J.A. Bergstra, W.J. Fokkink, and C.A. Middelburg. Algebra of timed frames. *International Journal Computer Mathematics*, 61:227–255, 1996.

[BG87] J.C.M. Baeten and R.J. van Glabbeek. Merge and termination in process algebra. In K.V. Nori, editor, *Foundations of Software Technology and Theoretical Computer Science VII*, volume 287 of *Lecture Notes in Computer Science*, pages 153–172, Pune, 1987. Springer-Verlag.

[BK88] J.A. Bergstra and J.W. Klop. A complete inference system for regular processes with silent moves. In F.R. Drake and J.K. Truss, editors, *Proceedings Logic Colloquium 1986*, Hull, 1988.

[BM99] J.C.M. Baeten and C.A. Middelburg. Process algebra with timing: real time and discrete time. Technical Report CSR 99-11, Eindhoven University of Technology, Department of Computing Science, 1999. To appear in: Handbook of Process Algebra, editors, J.A. Bergstra, A. Ponse, S.A. Smolka, Elsevier 2000.

[BR97] S.H.J. Bos and M.A. Reniers. The I^2C-bus in discrete-time process algebra. *Science of Computer Programming*, 29(1-2):235–258, 1997.

[BR00] J.C.M. Baeten and M.A. Reniers. Termination in timed process algebra. Technical Report CSR 2000-??, Eindhoven University of Technology, Department of Computing Science, 2000. To appear. `http://www.win.tue.nl/~michelr/website.ps.Z`.

[BV95] J.C.M. Baeten and C. Verhoef. Concrete process algebra. In S. Abramsky, Dov M. Gabbay, and T.S.E. Maibaum, editors, *Semantic Modelling*, volume 4 of *Handbook of Logic in Computer Science*, pages 149–268. Oxford University Press, 1995.

[BV97] J.C.M. Baeten and J.J. Vereijken. Discrete-time process algebra with empty process. In M. Bruné, A. van Deursen, and J. Heering, editors, *Dat is dus heel interessant*, pages 5–24. CWI, 1997. Liber Amicorum dedicated to Paul Klint.

[Gla87] R.J. van Glabbeek. Bounded nondeterminism and the aproximation induction principle in process algebra. In F.J. Brandenburg, G. Vidal-Nauquet, and M. Wirsing, editors, *Proceedings of STACS'87*, volume 247 of *Lecture Notes in Computer Science*, pages 336–347. Springer-Verlag, 1987.

[HR95] M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 177(2):221–239, 1995.

[IT94] ITU-T. *Recommendation Z.100: Specification and Description Language (SDL)*. ITU-T, Geneva, June 1994.

[KV85] C.P.J. Koymans and J.L.M. Vrancken. Extending process algebra with the empty process ε. Technical Report Logic Group Preprint Series 1, University Utrecht, Department of Philosophy, 1985.

[MT90] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *CONCUR'90 - Theories of Concurrency: Unification and Extension*, volume 458 of *Lecture Notes in Computer Science*, pages 401–415, Amsterdam, 1990. Springer-Verlag.

[NS94] X. Nicollin and J. Sifakis. The algebra of timed processes, ATP: Theory and application. *Information and Computation*, 114(1):131–178, 1994.

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

[Ren99] M.A. Reniers. *Message Sequence Chart: Syntax and Semantics*. PhD thesis, Eindhoven University of Technology, June 1999.

[SDJ+91] S. Schneider, J. Davies, D.M. Jackson, G.M. Reed, J.N. Reed, and A.W. Roscoe. Timed CSP: Theory and practice. In J.W. de Bakker, C. Huizing, W.P. de Roever, and G. Rozenberg, editors, *Real Time: Theory and Practice*, volume 600 of *Lecture Notes in Computer Science*, pages 640–675. Springer-Verlag, 1991.

[Ver94] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. In B. Jonsson and J. Parrow, editors,

CONCUR'94: Concurrency Theory, volume 836 of *Lecture Notes in Computer Science*, pages 433–448, Uppsala, 1994. Springer-Verlag.

[Ver97] J.J. Vereijken. *Discrete-time process algebra.* PhD thesis, Eindhoven University of Technology, 1997.

[Vra97] J.L.M. Vrancken. The algebra of communicating processes with empty process. *Theoretical Computer Science*, 177(2):287–328, 1997.

A Proof of strong bisimilarity implies σ-bisimilarity

Assume that $R : p \leftrightarrows q$. Define R_σ as follows:

$$R_\sigma = \{(S,T) \mid \forall_{s\in S}\ \exists_{t\in T} (s,t) \in R \wedge \forall_{t\in T}\ \exists_{s\in S}\ (s,t) \in R\}$$

Next, we prove that R_σ is a σ-bisimulation relation. Consider arbitrary $SR_\sigma T$.

Suppose $s \downarrow$ for some $s \in S$. Then, obviously there exists $t \in T$ such that sRt. Hence, $t \downarrow$.

Suppose $s \uparrow$ for all $s \in S$. Then, obviously $t \uparrow$ for all $t \in T$.

Suppose $s \overset{r}{\mapsto} s'$ for some $s \in S$. Then there exists $t \in T$ such that sRt. Hence $t \overset{r}{\mapsto} t'$ for some t'. Furthermore we must prove that $\{s' \mid \exists_{s\in S}\ s \overset{r}{\mapsto} s'\} R_\sigma \{t' \mid \exists_{t\in T}\ t \overset{r}{\mapsto} t'\}$. Obviously the case.

B Proof of completeness of $\mathrm{BPA}_{\dot\epsilon}^{srt}$

In this appendix we prove that $\mathrm{BPA}_{\dot\epsilon}^{srt}$ is a complete axiomatization of strong bisimilarity on closed $\mathrm{BPA}_{\dot\epsilon}^{srt}$- terms. First, we give lemma that relates the operational semantics to the equational derivability.

Lemma 2.

1. *if* $x \downarrow$, *then* $x = \dot\epsilon + x$;
2. *if* $x \not\downarrow$, *then* $x = x + \tilde{\tilde{\delta}}$;
3. *if* $x \overset{a}{\to} x'$, *then* $x = \tilde{\tilde{a}} \cdot x' + x$;
4. *if* $x \prec y$, *then* $x + y = y$;
5. *if* $x \overset{r}{\mapsto} x'$, *then* $x = \tilde{\tilde{\sigma}}^r \cdot x' + x$.

Proof.

1. By induction on the structure of closed term x.
2. By induction on the structure of closed term x, using the first item.
3. By induction on the structure of closed term x, using the first item.
4. By induction on the depth of the proof of $x \prec y$, using the first two items. The last step in the proof is an application of deduction rule:
   [24] Obviously we have $x + x = x$.
   [25] Then $x \equiv \dot\delta$. We obviously have $x + y = \dot\delta + y = y$.
   [26] Then $x \equiv \tilde{\tilde{\delta}}$. From $y \not\downarrow$ and the second item we have $y = y + \tilde{\tilde{\delta}}$. Then $x + y = \tilde{\tilde{\delta}} + y = y$.

[27] Then $x \equiv \dot{\epsilon}$. By the first item and $y \downarrow$ we have $y = \dot{\epsilon} + y$. Hence $x + y = \dot{\epsilon} + y = y$.

[28] Then $x \equiv \tilde{\tilde{\epsilon}}$. By the first two items we have $y = \dot{\epsilon} + y$ and $y = y + \tilde{\tilde{\delta}}$. Then $x + y = \tilde{\tilde{\epsilon}} + y = \dot{\epsilon} + \tilde{\tilde{\delta}} + y = \dot{\epsilon} + y = y$.

[29] Then $y = x' + y'$ and $x \prec x'$. By induction we then have $x + x' = x'$. Then $x + y = x + x' + y' = x' + y' = y$.

[30] Similar to the previous case.

[31] Then $x \equiv x' + x''$ and $y \equiv y' + y''$ for some x', x'', y', y'' such that $x' \prec y'$ and $x'' \prec y''$. By induction we then have $x' + y' = y'$ and $x'' + y'' = y''$. Then $x + y = x' + x'' + y' + y'' = x' + y' + x'' + y'' = y' + y'' = y$.

[32] Then $x \equiv x' \cdot z$ and $y \equiv y' \cdot z$ for some x', y', z such that $x' \prec y'$. By induction we then have $x' + y' = y'$. Thus, $x + y = x' \cdot z + y' \cdot z = (x' + y') \cdot z = y' \cdot z = y$.

[33] Then $y \equiv y' \cdot z$ for some y', z such that $y' \downarrow$ and $x \prec z$. By the first item we have $y' + \dot{\epsilon} = y'$. By induction we have $x + z = z$. Thus, $x + y = x + y' \cdot z = x + (\dot{\epsilon} + y') \cdot z = x + \dot{\epsilon} \cdot z + y' \cdot z = x + z + y' \cdot z = z + y' \cdot z = (\dot{\epsilon} + y') \cdot z = y' \cdot z = y$.

[34] Then $x \equiv \tilde{\tilde{\sigma}}^p \cdot x'$ and $y \equiv \tilde{\tilde{\sigma}}^p \cdot y'$ for some x', y' such that $x' \prec y'$. By induction we have $x' + y' = y'$. Thus, $x + y = \tilde{\tilde{\sigma}}^p \cdot x' + \tilde{\tilde{\sigma}}^p \cdot y' = \tilde{\tilde{\sigma}}^p \cdot (x' + y') = \tilde{\tilde{\sigma}}^p \cdot y' = y$.

5. By induction on the depth of the proof of $x \overset{r}{\mapsto} x'$ using the first and fourth item. The last step in the proof is an application of deduction rule:

[8] Then we must have $x \equiv \tilde{\tilde{\sigma}}^{p+r}$ and $x' \equiv \tilde{\tilde{\sigma}}^p$ for some $p, r \in R_{\geq 0}$ and $r > 0$. Then

$$\tilde{\tilde{\sigma}}^r \cdot x' + x = \tilde{\tilde{\sigma}}^r \cdot \tilde{\tilde{\sigma}}^{p-r} + \tilde{\tilde{\sigma}}^p = \tilde{\tilde{\sigma}}^p + \tilde{\tilde{\sigma}}^p = \tilde{\tilde{\sigma}}^p = x.$$

[14] Then $x \equiv x_1 + x_2$ for some x_1, x_2 such that $x_1 \overset{r}{\mapsto} x'$. By induction we then have $\tilde{\tilde{\sigma}}^r \cdot x' + x_1 = x_1$. Then

$$\tilde{\tilde{\sigma}}^r \cdot x' + x = \tilde{\tilde{\sigma}}^r \cdot x' + x_1 + x_2 = x_1 + x_2 = x.$$

[15] Similar to the previous case.

[21] Then $x \equiv x_1 \cdot x_2$ and $x_1 \overset{r}{\mapsto} x_1'$ for some x_1, x_2, x_1' such that $x' \equiv x_1' \cdot x_2$. By induction we have $\tilde{\tilde{\sigma}}^r \cdot x_1' + x_1 = x_1$. Then

$$\tilde{\tilde{\sigma}}^r \cdot x' + x = \tilde{\tilde{\sigma}}^r \cdot x_1' \cdot x_2 + x_1 \cdot x_2 = (\tilde{\tilde{\sigma}}^r \cdot x_1' + x_1) \cdot x_2 = x_1 \cdot x_2 = x.$$

[22] Then $x \equiv x_1 \cdot x_2$ for some x_1, x_2 such that $x_1 \downarrow$ and $x_2 \overset{r}{\mapsto} x'$. By the first item and $x_1 \downarrow$ we have $x_1 = \dot{\epsilon} + x_1$. By induction we have $\tilde{\tilde{\sigma}}^r \cdot x' + x_2 = x_2$.

Then

$$\begin{aligned}
\tilde{\tilde{\sigma}}^r \cdot x' + x &= \tilde{\tilde{\sigma}}^r \cdot x' + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + (\dot{\epsilon} + x_1) \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + \dot{\epsilon} \cdot x_2 + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + x_2 + x_1 \cdot x_2 \\
&= x_2 + x_1 \cdot x_2 \\
&= \dot{\epsilon} \cdot x_2 + x_1 \cdot x_2 \\
&= (\dot{\epsilon} + x_1) \cdot x_2 \\
&= x_1 \cdot x_2 \\
&= x.
\end{aligned}$$

[23] Then $x \equiv x_1 \cdot x_2$, $x_1 \overset{r_1}{\mapsto} x_1'$, $x_1' \downarrow$, and $x_2 \overset{r_2}{\mapsto} x'$ for some x_1, x_2, x_1' and r_1, r_2 such that $r = r_1 + r_2$. By the first item we have $x_1' = \dot{\epsilon} + x_1'$. By induction we have $\tilde{\tilde{\sigma}}^{r_1} \cdot x_1' + x_1 = x_1$ and $\tilde{\tilde{\sigma}}^{r_2} \cdot x' + x_2 = x_2$. Then

$$\begin{aligned}
\tilde{\tilde{\sigma}}^r \cdot x' + x &= \tilde{\tilde{\sigma}}^r \cdot x' + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + (\tilde{\tilde{\sigma}}^{r_1} \cdot x_1' + x_1) \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + \tilde{\tilde{\sigma}}^{r_1} \cdot x_1' \cdot x_2 + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + \tilde{\tilde{\sigma}}^{r_1} \cdot (\dot{\epsilon} + x_1') \cdot x_2 + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^r \cdot x' + \tilde{\tilde{\sigma}}^{r_1} \cdot (x_2 + x_1' \cdot x_2) + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^{r_1} \cdot (\tilde{\tilde{\sigma}}^{r_2} \cdot x' + x_2 + x_1' \cdot x_2) + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^{r_1} \cdot (x_2 + x_1 \cdot x_2) + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^{r_1} \cdot (\dot{\epsilon} + x_1') \cdot x_2 + x_1 \cdot x_2 \\
&= \tilde{\tilde{\sigma}}^{r_1} \cdot x_1' \cdot x_2 + x_1 \cdot x_2 \\
&= (\tilde{\tilde{\sigma}}^{r_1} \cdot x_1' + x_1) \cdot x_2 \\
&= x_1 \cdot x_2 \\
&= x.
\end{aligned}$$

[35] Then $x \overset{r}{\mapsto} x_1$ and $x \overset{r}{\mapsto} x_2$ for some x_1, x_2 such that $x' \equiv x_1 + x_2$. By induction we have $\tilde{\tilde{\sigma}}^r \cdot x_1 + x = x$ and $\tilde{\tilde{\sigma}}^r \cdot x_2 + x = x$. Then

$$\begin{aligned}
\tilde{\tilde{\sigma}}^r \cdot x' + x &= \tilde{\tilde{\sigma}}^r \cdot (x_1 + x_2) + x \\
&= \tilde{\tilde{\sigma}}^r \cdot x_1 + \tilde{\tilde{\sigma}}^r \cdot x_2 + x \\
&= \tilde{\tilde{\sigma}}^r \cdot x_1 + x \\
&= x.
\end{aligned}$$

[36] Then $x \overset{r}{\mapsto} x''$ for some x'' such that $x' \prec x''$. By the fourth item and $x' \prec x''$ we have $x'' + x' = x''$. By induction we have $\tilde{\tilde{\sigma}}^r \cdot x'' + x = x$.

Then

$$\begin{aligned}\tilde{\tilde{\sigma}}^r \cdot x' + x &= \tilde{\tilde{\sigma}}^r \cdot x' + \tilde{\tilde{\sigma}}^r \cdot x'' + x \\ &= \tilde{\tilde{\sigma}}^r \cdot (x' + x'') + x \\ &= \tilde{\tilde{\sigma}}^r \cdot x'' + x \\ &= x.\end{aligned}$$

Now we turn to the completeness of $\mathrm{BPA}_{\dot{\epsilon}}^{srt}$. We only have to prove that $x + y \underline{\leftrightarrow} y$ implies $x + y = y$.

Suppose $x + y \underline{\leftrightarrow} y$. By induction on the structure of basic term x.

- $x \equiv \dot{\delta}$. Obviously $x + y = \dot{\delta} + y = y$.
- $x \equiv \dot{\epsilon}$. Hence $x + y \downarrow$. Thus we obtain $y \downarrow$. By Lemma 2.1 this gives $y = \dot{\epsilon} + y$. Clearly $x + y = \dot{\epsilon} + y = y$.
- $x \equiv \tilde{\tilde{a}} \cdot x'$.
  - $a \equiv \delta$. Then $x \not\downarrow$. Therefore $x + y \not\downarrow$. Thus $y \not\downarrow$. By Lemma 2.2 we thus have $y = y + \tilde{\tilde{\delta}}$. We find $x + y = \tilde{\tilde{\delta}} \cdot x' + y = \tilde{\tilde{\delta}} + y = y$.
  - $a \in A$. Then $x \xrightarrow{a} \dot{\epsilon} \cdot x'$. Hence $x + y \xrightarrow{a} \dot{\epsilon} \cdot x'$. Therefore $y \xrightarrow{a} y'$ for some y' such that $\dot{\epsilon} \cdot x' \underline{\leftrightarrow} y'$. Obviously $x' \underline{\leftrightarrow} y'$. By induction we have $x' = y'$. By Lemma 2.3 we have $y = \tilde{\tilde{a}} \cdot y' + y$. Then $x + y = \tilde{\tilde{a}} \cdot x' + y = \tilde{\tilde{a}} \cdot y' + y = y$.
- $x \equiv \tilde{\tilde{\sigma}}^r \cdot x'$. Then $x \overset{r}{\mapsto} \tilde{\tilde{\sigma}}^0 \cdot x'$. Hence $x + y \overset{r}{\mapsto} \tilde{\tilde{\sigma}}^0 \cdot x'$ and therefore $y \overset{r}{\mapsto} y'$ for some y' such that $\tilde{\tilde{\sigma}}^0 \cdot x' \underline{\leftrightarrow} y'$. Then also $x' \underline{\leftrightarrow} y'$. By induction we then have $x' = y'$. By Lemma 2.5 we have $y = \tilde{\tilde{\sigma}}^r \cdot y' + y$. Then $x + y = \tilde{\tilde{\sigma}}^r \cdot x' + y = \tilde{\tilde{\sigma}}^r \cdot y' + y = y$.
- $x \equiv x' + x''$.

C Proof of representation theorem

The proof of this theorem requires that we strengthen the proposition to be proven. Hence, we consider the following proposition: For basic term x we have $x = \nu_{\mathrm{rel}}(x)$ or there exists $p > 0$ such that for all q: $0 < q \leq p$ there exists basic term x' such that $x = \upsilon_{\mathrm{rel}}{}^q(x) + \tilde{\tilde{\sigma}}^q \cdot x'$ and $|x'| < |x|$.

This proposition is proven by induction on the structure of basic term x. The first three cases ($x = \dot{\epsilon}$, $x = \dot{\delta}$, and $x = \tilde{\tilde{a}} \cdot x'$) are trivial as in these cases $x = \nu_{\mathrm{rel}}(x)$.

The case $x = \tilde{\tilde{\sigma}}^r \cdot x'$ ($r > 0$) is also not difficult. Let $0 < r' \leq r$. Then $x = x + \upsilon_{\mathrm{rel}}{}^{r'}(x) = \tilde{\tilde{\sigma}}^r \cdot x' + \upsilon_{\mathrm{rel}}{}^{r'}(x) = \tilde{\tilde{\sigma}}^{r'} \cdot (\tilde{\tilde{\sigma}}^{r-r'} \cdot x') + \upsilon_{\mathrm{rel}}{}^{r'}(x)$. Note that $\left|\tilde{\tilde{\sigma}}^{r-r'} \cdot x'\right| = (r - r' + 1) \times |x'| + 1 < (r + 1) \times |x'| + 1 = \left|\tilde{\tilde{\sigma}}^r \cdot x'\right| = |x|$.

Finally, we treat the case $x = x_1 + x_2$ in more detail. Four cases can be distinguished:

1. $x_1 = \nu_{\mathrm{rel}}(x_1)$ and $x_2 = \nu_{\mathrm{rel}}(x_2)$. Trivial.
2. $x_1 = \nu_{\mathrm{rel}}(x_1)$ and there exists $r_2 > 0$ such that for all all $0 < r_2' \leq r_2$ there exists x_2' with $x_2 = \upsilon_{\mathrm{rel}}{}^{r_2'}(x_2) + \tilde{\tilde{\sigma}}^{r_2'} \cdot x_2'$ and $|x_2'| < |x_2|$. Note that

$x_1 = x_1 + \upsilon_{\mathrm{rel}}{}^{r_2'}(x_1) = \nu_{\mathrm{rel}}(x_1) + \upsilon_{\mathrm{rel}}{}^{r_2'}(x_1) = \upsilon_{\mathrm{rel}}{}^{r_2'}(x_1)$. Then trivially: $x = x_1 + x_2 = \upsilon_{\mathrm{rel}}{}^{r_2'}(x_1) + \upsilon_{\mathrm{rel}}{}^{r_2'}(x_2) + \tilde{\tilde{\sigma}}^{r_2'} \cdot x_2' = \upsilon_{\mathrm{rel}}{}^{r_2'}(x_1 + x_2) + \tilde{\tilde{\sigma}}^{r_2'} \cdot x_2' = \upsilon_{\mathrm{rel}}{}^{r_2'}(x) + \tilde{\tilde{\sigma}}^{r_2'} \cdot x_2'$. Note that $|x_2'| < |x_2| < |x_1 + x_2|$.

3. there exists $r_1 > 0$ such that for all $0 < r_1' \leq r_1$ there exists x_1' with $x_1 = \upsilon_{\mathrm{rel}}{}^{r_1'}(x_1) + \tilde{\tilde{\sigma}}^{r_1'} \cdot x_1'$ and $|x_1'| < |x_1|$ and $x_2 = \nu_{\mathrm{rel}}(x_2)$. Similar to the previous case.
4. there exist $r_1 > 0$ and $r_2 > 0$ such that for all $0 < r_1' \leq r_1$ and $0 < r_2' \leq r_2$ there exist x_1' and x_2' with $x_1 = \upsilon_{\mathrm{rel}}{}^{r_1'}(x_1) + \tilde{\tilde{\sigma}}^{r_1'} \cdot x_1'$ and $x_2 = \upsilon_{\mathrm{rel}}{}^{r_2'}(x_2) + \tilde{\tilde{\sigma}}^{r_2'} \cdot x_2'$. Without loss of generality we may assume that $r_1 \leq r_2$. By the induction hypothesis we find that there exists x_2' such that $x = \upsilon_{\mathrm{rel}}{}^{r_1'}(x_2) + \tilde{\tilde{\sigma}}^{r_1'} \cdot x_2'$ and $|x_2'| < |x_2|$. Then $x_1 + x_2 = \upsilon_{\mathrm{rel}}{}^{r_1'}(x_1) + \tilde{\tilde{\sigma}}^{r_1'} \cdot x_1' + \upsilon_{\mathrm{rel}}{}^{r_1'}(x_2) + \tilde{\tilde{\sigma}}^{r_1'} \cdot x_2' = \upsilon_{\mathrm{rel}}{}^{r_1'}(x_1 + x_2) + \tilde{\tilde{\sigma}}^{r_1'} \cdot (x_1' + x_2')$. From $|x_1'| < |x_1|$ and $|x_2'| < |x_2|$ we find $|x_1' + x_2'| < |x_1 + x_2|$.

D Proof of elimination theorem for $\mathrm{ACP}_{\dot{\epsilon}}^{srt}$

We prove the following propositions simultaneously for all basic terms x and y with induction on $|x| + |y|$:

1. there exists a basic term z such that $x \mathbin{\lfloor\!\lfloor} y = z$;
2. there exists a basic term z such that $x \mid y = z$;
3. there exists a basic term z such that $x \parallel y = z$.

The first proposition is the most complex of the three. We start by distinguishing cases for x according to the definition of basic terms. For the case $x = \dot{\epsilon}$ we proceed by distinguishing cases for y according to the definition of basic terms as well. For the case $x = \dot{\delta}$ the elimination of $\mathbin{\lfloor\!\lfloor}$ is trivial: $\dot{\delta} \mathbin{\lfloor\!\lfloor} y = \dot{\delta}$. For the case $x = \tilde{\tilde{a}} \cdot x'$ we distinguish the cases $y = \dot{\epsilon}$, $y = \dot{\delta}$ and $y = y + \tilde{\tilde{\delta}}$. In case $y = \dot{\epsilon}$ or $y = \dot{\delta}$ we easily find that a basic term results. For $y = y + \tilde{\tilde{\delta}}$ we find $x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{a}} \cdot x' \mathbin{\lfloor\!\lfloor} (y + \tilde{\tilde{\delta}}) = \tilde{\tilde{a}} \cdot (x' \parallel (y + \tilde{\tilde{\delta}})) = \tilde{\tilde{a}} \cdot (x' \parallel y)$. By induction we have the existence of a basic term z such that $x \parallel y = z$. Then $\tilde{\tilde{a}} \cdot (x \parallel y) = \tilde{\tilde{a}} \cdot z$ which is also a basic term. For the case $x = \tilde{\tilde{\sigma}}^{r} \cdot x'$ we distinguish four cases for y according to the representation theorem and representation lemma above.

1. $y = \dot{\epsilon}$. Then $x \mathbin{\lfloor\!\lfloor} y = x \mathbin{\lfloor\!\lfloor} \dot{\epsilon} = x$.
2. $y = \dot{\delta}$. Then $x \mathbin{\lfloor\!\lfloor} y = x \mathbin{\lfloor\!\lfloor} \dot{\delta} = \dot{\delta}$.
3. $y = \nu_{\mathrm{rel}}(y) + \tilde{\tilde{\delta}}$. Then $x \parallel y = \tilde{\tilde{\sigma}}^{r} \cdot x' \mathbin{\lfloor\!\lfloor} (\nu_{\mathrm{rel}}(y) + \tilde{\tilde{\delta}}) = \tilde{\tilde{\delta}} = \tilde{\tilde{\delta}} \cdot \dot{\epsilon}$.
4. $y = \upsilon_{\mathrm{rel}}{}^{r'}(y) + \tilde{\tilde{\sigma}}^{r'} \cdot y'$ for some $r' > 0$ and basic term y' such that $|y'| < |y|$. If $r' \geq r$ then $y = \upsilon_{\mathrm{rel}}{}^{r}(y) + \tilde{\tilde{\sigma}}^{r} \cdot y''$ and $|y''| < |y|$. Then $x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{\sigma}}^{r} \cdot x' \mathbin{\lfloor\!\lfloor} (\upsilon_{\mathrm{rel}}{}^{r}(y) + \tilde{\tilde{\sigma}}^{r} \cdot y'') = \tilde{\tilde{\sigma}}^{r} \cdot (x' \mathbin{\lfloor\!\lfloor} y'')$. By induction we have the existence of basic term z such that $x' \mathbin{\lfloor\!\lfloor} y'' = z$. Hence $x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{\sigma}}^{r} \cdot z$, which is a basic term. If $r' < r$, then $x \mathbin{\lfloor\!\lfloor} y = \tilde{\tilde{\sigma}}^{r} \cdot x' \mathbin{\lfloor\!\lfloor} (\upsilon_{\mathrm{rel}}{}^{r'}(y) + \tilde{\tilde{\sigma}}^{r'} \cdot y') = \tilde{\tilde{\sigma}}^{r'} \cdot \tilde{\tilde{\sigma}}^{r-r'} \cdot x' \mathbin{\lfloor\!\lfloor} (\upsilon_{\mathrm{rel}}{}^{r'}(y) + \tilde{\tilde{\sigma}}^{r'} \cdot y') = \tilde{\tilde{\sigma}}^{r'} \cdot (\tilde{\tilde{\sigma}}^{r-r'} \cdot x' \mathbin{\lfloor\!\lfloor} y')$. As $\left|\tilde{\tilde{\sigma}}^{r-r'} \cdot x' \mathbin{\lfloor\!\lfloor} y'\right| = \left|\tilde{\tilde{\sigma}}^{r-r'} \cdot x'\right| + |y'| = (r - r' + 1) \times$

$|x'|+1+|y'| < (r+1)\times|x'|+1+|y| = \left|\tilde{\tilde{\sigma}}^{r}\cdot x'\right|+|y| = \left|\tilde{\tilde{\sigma}}^{r}\cdot x' \lfloor\!\lfloor y\right| = |x \lfloor\!\lfloor y|$, by induction we have the existence of basic term z such that $\tilde{\tilde{\sigma}}^{r-r'}\cdot x' \lfloor\!\lfloor y' = z$. Hence $x \lfloor\!\lfloor y = \tilde{\tilde{\sigma}}^{r'}\cdot z$, which is a basic term.

For the case $x = x_1+x_2$ we apply the induction hypothesis and the distributivity law of left merge: $x \lfloor\!\lfloor y = (x_1+x_2) \lfloor\!\lfloor y = x_1 \lfloor\!\lfloor y + x_2 \lfloor\!\lfloor y = z_1 + z_2$.

For the second proposition we distinguish cases for both x and y according to the definition of basic terms. The proof is easy. The third proposition follows immediately from the first and second proposition and the fact that the alternative composition of basic terms gives a basic term again.

E Associativity of parallel composition

In this appendix we prove that parallel composition is associative. Thereto, we prove the following equations simultaneously with induction on $|x|+|y|+|z|$:

$$\begin{aligned}
&(x \lfloor\!\lfloor y) \lfloor\!\lfloor z = x \lfloor\!\lfloor (y \parallel z),\\
&(x \mid y) \mid z = x \mid (y \mid z),\\
&(x \mid y) \lfloor\!\lfloor z = x \mid (y \lfloor\!\lfloor z),\\
&(x \parallel y) \parallel z = x \parallel (y \parallel z).
\end{aligned}$$

These axioms are in ACP-literature usually referred to as the *Axioms of Standard Concurrency*. Allthough we prove the equations simultaneously we present their proofs separatly to enhance readability.

Assume that

$$x = \sum_{i\in I} \tilde{\tilde{a}}_i \cdot x_i + \tilde{\tilde{\sigma}}^{u}\cdot x' + \sum_{j\in J} \dot{\epsilon}$$

and

$$y = \sum_{k\in K} \tilde{\tilde{b}}_k \cdot y_k + \tilde{\tilde{\sigma}}^{v}\cdot y' + \sum_{l\in L} \dot{\epsilon}$$

and

$$z = \sum_{m\in M} \tilde{\tilde{c}}_m \cdot z_m + \tilde{\tilde{\sigma}}^{w}\cdot z' + \sum_{n\in N} \dot{\epsilon}$$

for some $a_i, b_k, c_m \in A_\delta$ and basic terms $x_i, x', y_k, y', z_m, z'$ such that $u = 0$ iff $x' = \dot{\delta}$, $v = 0$ iff $y' = \dot{\delta}$, $w = 0$ iff $z' = \dot{\delta}$.

- $(x \lfloor\!\lfloor y) \lfloor\!\lfloor z = x \lfloor\!\lfloor (y \parallel z)$. If $y = \dot{\delta}$ or $z = \dot{\delta}$ or $y = \dot{\epsilon}$ or $z = \dot{\epsilon}$ the proposition is trivially proven. Hence from now on we may assume that $y = y + \tilde{\tilde{\delta}}$ and $z = z + \tilde{\tilde{\delta}}$.
  Now, we will distinguish for x the basic term cases. The cases $x = \dot{\epsilon}$, $x = \dot{\delta}$, $x = \tilde{\tilde{a}}\cdot x'$, and $x = x' + x''$ are trivial and therefore omited. What remains is the case $x = \tilde{\tilde{\sigma}}^{u}\cdot x'$ (with $u > 0$). Let $uvw = \min(u,v,w)$. If $uvw = 0$, then necessarily $v = 0$ or $w = 0$. Therefore $y' = \dot{\delta}$ or $z' = \dot{\delta}$. We obtain $(\tilde{\tilde{\sigma}}^{u}\cdot x' \lfloor\!\lfloor y) \lfloor\!\lfloor z = \tilde{\tilde{\delta}}$ and $\tilde{\tilde{\sigma}}^{u}\cdot x' \lfloor\!\lfloor (y \parallel z) = \tilde{\tilde{\delta}}$.

Now suppose that $uvw > 0$. Observe that

$$y \parallel z = \sum_{k\in K} \tilde{\tilde{b}}_k \cdot (y_k \parallel z) + \sum_{m\in M} \tilde{\tilde{c}}_m \cdot (z_m \parallel y)$$

$$+ \sum_{k\in K} \sum_{m\in M} \gamma(\tilde{\tilde{b}}_k, c_m) \cdot (y_k \parallel z_m)$$

$$+ \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \parallel \tilde{\tilde{\sigma}}^{w-uvw} \cdot z') + \sum_{n\in N} y + \sum_{l\in L} z$$

Then

$$\tilde{\tilde{\sigma}}^{u} \cdot x' \lfloor\!\lfloor (y \parallel z) = \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \lfloor\!\lfloor (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \parallel \tilde{\tilde{\sigma}}^{w-uvw} \cdot z'))$$

and

$$(\tilde{\tilde{\sigma}}^{u} \cdot x' \lfloor\!\lfloor y) \lfloor\!\lfloor z = \tilde{\tilde{\sigma}}^{uvw} \cdot ((\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{v-uvw} \cdot y') \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')$$

By induction we have $\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \lfloor\!\lfloor (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \parallel \tilde{\tilde{\sigma}}^{w-uvw} \cdot z') = (\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{v-uvw} \cdot y') \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-uvw} \cdot z'$ since $\left|\tilde{\tilde{\sigma}}^{u-uvw} \cdot x'\right| < |x|$, $\left|\tilde{\tilde{\sigma}}^{v-uvw} \cdot y'\right| < |y|$, and $\left|\tilde{\tilde{\sigma}}^{w-uvw} \cdot z'\right| < |z|$.

- $(x \mid y) \mid z = x \mid (y \mid z)$. Let $uvw = \min(u, v, w)$. If $uvw > 0$

$$(x \mid y) \mid z = \sum_{i\in I} \sum_{k\in K} \sum_{m\in M} \gamma(\gamma(a_i, \tilde{\tilde{b}}_k), c_m) \cdot ((x_i \parallel y_k) \parallel z_m)$$

$$+ \tilde{\tilde{\sigma}}^{uvw} \cdot ((\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid \tilde{\tilde{\sigma}}^{v-uvw} \cdot y') \mid \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')$$

and

$$x \mid (y \mid z) = \sum_{i\in I} \sum_{k\in K} \sum_{m\in M} \gamma(a_i, \gamma(\tilde{\tilde{b}}_k, c_m)) \cdot (x_i \parallel (y_k \parallel z_m))$$

$$+ \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \mid \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')).$$

By the induction hypthesis we have $(x' \parallel y') \parallel z' = x' \parallel (y' \parallel z')$. By the associativity of the communication function we have $\gamma(\gamma(a_i, b_k), c_m) = \gamma(a_i, \gamma(b_k, c_m))$. By the induction hypthesis we have $(\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid \tilde{\tilde{\sigma}}^{v-uvw} \cdot y') \mid \tilde{\tilde{\sigma}}^{w-uvw} \cdot z' = \tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \mid \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')$.
If $uvw = 0$ then we have similar results without the alternatives headed by the delay constants.

- $(x \mid y) \lfloor\!\lfloor z = x \mid (y \lfloor\!\lfloor z)$. Let $uvw = \min(u, v, w)$, $uv = \min(u, v)$, and $vw = \min(v, w)$.

$$(x \mid y) \lfloor\!\lfloor z = (\sum_{i\in I} \sum_{k\in K} \gamma(\tilde{\tilde{a}}_i, b_k) \cdot (x_i \parallel y_k) + \tilde{\tilde{\sigma}}^{uv} \cdot (\tilde{\tilde{\sigma}}^{u-uv} \cdot x' \mid \tilde{\tilde{\sigma}}^{v-uv} \cdot y')) \lfloor\!\lfloor z$$

$$= \sum_{i\in I} \sum_{k\in K} \gamma(\tilde{\tilde{a}}_i, b_k) \cdot ((x_i \parallel y_k) \parallel z)$$

$$+ \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{uv-uvw} \cdot (\tilde{\tilde{\sigma}}^{u-uv} \cdot x' \mid \tilde{\tilde{\sigma}}^{v-uv} \cdot y') \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')$$

$$= \sum_{i\in I} \sum_{k\in K} \gamma(\tilde{\tilde{a}}_i, b_k) \cdot ((x_i \parallel y_k) \parallel z)$$

$$+ \tilde{\tilde{\sigma}}^{uvw} \cdot ((\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid \tilde{\tilde{\sigma}}^{v-uvw} \cdot y') \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-uvw} \cdot z')$$

$$
\begin{aligned}
x \mid (y \lfloor\!\lfloor z) &= x \mid (\textstyle\sum_{k \in K} \tilde{\tilde{b}}_k \cdot (y_k \parallel z)) \\
&+ x \mid (\tilde{\tilde{\sigma}}^{vw} \cdot (\tilde{\tilde{\sigma}}^{v-vw} \cdot y' \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-vw} \cdot z')) \\
&+ x \mid (\textstyle\sum_{l \in L} \sum_{n \in N} \dot{\epsilon}) \\
&= \textstyle\sum_{i \in I} \sum_{k \in K} \gamma(\tilde{\tilde{a_i}}, b_k) \cdot (x_i \parallel (y_k \parallel z)) \\
&+ \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid (\tilde{\tilde{\sigma}}^{vw-uvw} \cdot (\tilde{\tilde{\sigma}}^{v-vw} \cdot y' \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-vw} \cdot z'))) \\
&= \textstyle\sum_{i \in I} \sum_{k \in K} \gamma(\tilde{\tilde{a_i}}, b_k) \cdot (x_i \parallel (y_k \parallel z)) \\
&+ \tilde{\tilde{\sigma}}^{uvw} \cdot (\tilde{\tilde{\sigma}}^{u-uvw} \cdot x' \mid (\tilde{\tilde{\sigma}}^{v-uvw} \cdot y' \lfloor\!\lfloor \tilde{\tilde{\sigma}}^{w-uvw} \cdot z'))
\end{aligned}
$$

Applying the induction hypotheses gives the identity $(x \mid y) \lfloor\!\lfloor z = x \mid (y \lfloor\!\lfloor z)$.

– $(x \parallel y) \parallel z = x \parallel (y \parallel z)$. Trivial using the previous three propositions.