

# Timing the Untimed: Terminating Successfully while Being Conservative

J. C. M. Baeten, M.R. Mousavi, and M. A. Reniers

Department of Computer Science,  
Eindhoven University of Technology (TU/e),  
NL-5600 MB Eindhoven, The Netherlands

**Abstract.** There have been several timed extensions of ACP-style process algebras with successful termination. None of them, to our knowledge, are equationally conservative (ground-)extensions of ACP with successful termination. Here, we point out some design decisions which were the possible causes of this misfortune and by taking different decisions, we propose a spectrum of timed process algebras ordered by equational conservativity ordering.

## 1 The Untimed Past

The term “process algebra” was coined by Jan Bergstra and Jan Willem Klop in [8] to denote an algebraic approach to concurrency theory. Their process algebra had uniform atomic actions  $a_i$  for  $i \in I$  (with  $I$  some index set), sequential composition  $;$ , choice (alternative composition)  $+$  and left merge  $\ll$  as the basic composition operators.<sup>1</sup>

Much of the core theory of [8] remained intact in the course of more than 20 years of developments in the ACP-school (for Algebra of Communicating Processes) of process algebra. Their theory has however been subject to a number of, rather important, extensions and improvements. Next, we list some of the developments that are most relevant to the subject matter of this paper.

1. A major improvement over the process algebra of [8] was combining the concepts of *communication and concurrency* in the *Algebra of Communicating Processes (ACP)* which was proposed by Bergstra and Klop in [9,10]. In the process algebra of [8], parallel composition  $x \parallel y$  was a shorthand as defined below.

$$x \parallel y \doteq (x \ll y) + (y \ll x)$$

There was no possibility for the parallel components to communicate or synchronize. The situation was improved in [9,10] by introducing a (total) communication function, defining a communication merge operator  $|$  and raising the parallel composition operator  $\parallel$  to a basic composition operator in the algebra, rather than a defined term.

---

<sup>1</sup> Sequential composition was called “concatenation” and choice was called “union” in [8].

- Another major improvement has been the addition of *identity elements*. Bergstra and Klop in [8] did study the addition of a constant 0 which is an identity element for both nondeterministic choice and sequential composition but then they ruled out this option by observing that the addition of 0 leads to the following counter-intuitive equality:

$$x \cdot y = (x + 0) \cdot y = (x \cdot y) + (0 \cdot y) = (x \cdot y) + y$$

The above equality states that the sequential composition may forget about its first argument which is indeed pathological. A couple of years had to pass to reveal that, as in ordinary rings, two process constants  $\epsilon$  and  $\delta$  can be used to give  $\_ \cdot \_$  and  $\_ + \_$  their identity elements, respectively [16,22]. (Note that unlike in rings, left-distributivity of choice over sequential composition is still prohibited in the extended process algebra.) Hence, the process algebra  $PA_{\delta}^{\epsilon}$  of [22] had two extra constants  $\epsilon$  and  $\delta$ . A different proposal for the interplay of  $\epsilon$  and parallel composition was formulated in [4,7]. There, a new unary function symbol  $\surd(\_)$  is added to the signature in order to capture the possibility of termination for complex terms.  $ACP$  of [9,10] had  $\delta$  as an identity element for choice but lacked  $\epsilon$ . In both [16,22],  $\epsilon$  is added to  $ACP$  resulting in  $ACP^{\epsilon}$ . The constant  $\epsilon$  denotes termination, whereas the action constant encompasses both the action execution and the termination afterwards.

- The third improvement concerning the subject matter of this paper was the addition of *quantitative time*. Baeten and Bergstra, in [2], proposed a real-time-stamped extension of  $ACP$ . In [3], they extend  $ACP$  with discrete time using prefix operators  $\sigma_{rel}\_.$  and  $\sigma_{abs}\_.$  for relative and absolute timing, respectively.

Vereijken tried to extend the result of the first and second improvements with the third aspect in Chapter 6 of his Ph.D. thesis [20]. There, he introduced  $ACP_{drt,\epsilon} - ID$  as a discrete time extension of  $ACP^{\epsilon}$  (here,  $-ID$  denotes the absence of an immediate deadlock constant). However, as it turns out, the above three extensions do not match perfectly: while the extensions in each direction can be interpreted as a conservative one, there is no conservativity result for the extension of  $ACP^{\epsilon}$  with timing. In the next section, we review design decisions on the way to timing untimed process algebras. Among the design decisions, we try to find possible cause(s) for this misfortune and will try to improve the situation by redesigning the extensions. This way, we may deviate from the commonly accepted principles of  $ACP$ , as we see appropriate. The result will be a lattice of process theories ordered by equational conservativity ordering.

## 2 Timing the Untimed

The following design decisions have to be taken in order to extend an untimed process algebra with timing information:

1. Delayable vs. urgent actions: When extending an untimed process algebra with timing, a natural question is how to deal with the timing behavior of untimed basic actions. One choice is to regard them as urgent actions without any timing behavior. Another choice is to allow for an arbitrary timing behavior and introduce new urgent actions. The same decision has to be taken for deadlock and successful termination constants. We believe that taking the untimed actions (deadlock, termination) to be delayable is the more natural choice. The fact that no timing information is given should allow for an arbitrary timing of the implementation, rather than only allowing for the case of the urgent process. Further elaboration of the arguments can be found in [1].
2. Time stamped actions vs. separation of actions and time: Timing can be added to actions in terms of time stamps or alternatively, in terms of a separate delay operator. We choose the second option, as does most of the literature [17,19,15].
3. Time (non)determinism: Time determinism means that passage of time cannot make a choice. Usually, in timed extensions of *ACP* a weaker version of time determinism is used by forcing that time cannot make a choice unless one of the options prevents time to pass. In the latter case, a time transition shall resolve the choice in favor of options that allow for it. This is called weak time-determinism, and allows for a simple description of a time-out mechanism.
4. Time domain: Several decisions can be taken for the appropriate time domain. Existence or absence of a least element, discreteness or denseness of the time domain and having a partial (branching) or full (linear) ordering on the elements of the time domain often lead to different timed theories. Here, we choose for a discrete time domain with a total linear ordering. This choice leads to the simplest theory, and is also the choice taken most often in the literature [17,19,15] (see [20] for an overview).
5. Relative vs. absolute time: The timing information may be taken relative to the successful termination of causally preceding actions or alternatively, to a fixed starting point. Here, we take the relative timing approach which is simpler to deal with. Our discussions carry over to the setting with absolute timing. We refer to [5] for more information on relative and absolute timing.

Let us take the above design decisions, and let us consider the timed extension of a simple process algebra,  $BPA_\delta^\epsilon$ , the theory *BPA* of [8] extended with constants  $\delta$  and  $\epsilon$  (see [20,1] for an overview of the above design decisions). Operational rules for this theory are given in Table 1.

We fix the notation used for transition system specifications (TSS) in this paper, as follows. The table containing the TSS is labelled with  $TSS(Name)$  where *Name* is the name of the process theory for which we are defining a TSS. For example in Table 1, we define a TSS for  $BPA_\delta^\epsilon$  and hence the table is labelled by it. Then, we name the TSS which we include in the definition, i.e., the TSS being extended. In an extension, we include the signature, transition relations, predicates and deduction rules from the original TSS and hence do not re-state

$TSS(BPA_\delta^\epsilon(A))$			
constant: $\delta, \epsilon, (a)_{a \in A}$ ;		binary: $-\cdot-, -+ -;$	
$-\downarrow;$		$(-\xrightarrow{a}-)_{a \in A};$	
$x, x', y, y';$			
$\epsilon \downarrow$	$a \xrightarrow{a} \epsilon$	$\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y}$	$\frac{x \downarrow \quad y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'}$
	$\frac{x \downarrow \quad y \downarrow}{x + y \downarrow}$	$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'}$	$\frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$

**Table 1.** Transition system specification for  $BPA_\delta^\epsilon(A)$ .

them in the extended TSS. Table 1 does not extend any previous theory and hence the extension line is empty. Subsequently, we give the signature of the theory in terms of function symbols and their arities. In Table 1, the signature consists of constants  $\delta, \epsilon$  and actions  $a \in A$ , as well as binary function symbols  $-\cdot-$  and  $-+ -$  for sequential composition and choice, respectively. The transition relations and predicates being defined by the TSS follow afterwards. In the TSS of Table 1,  $\rightarrow$  is the transition relation labelled by  $a \in A$  and  $\downarrow$  is the termination predicate. Finally, the set of deduction rules is presented. Most of the deduction rules given in Table 1 are quite standard and self-explanatory.

Before we continue with the extension of the basic process algebra to the timed setting, we fix the semantics of TSS's in terms of their induced transition relations and predicates. We write  $TSS(A) \models p \xrightarrow{l} q$  and  $TSS(A) \models P(p)$ , where  $p$  and  $q$  are closed terms from the signature of  $TSS(A)$ , and  $\xrightarrow{l}$  is a transition relation and  $P$  is a predicate, and by that we mean *formulae*  $p \xrightarrow{l} q$  and  $P(p)$  are *provable* in  $TSS(A)$ . Due to the presence of negative premises (in the TSS's that are yet to be presented in this paper), it is not clear what is a proof for negative premises and several different interpretations exist in the literature (see [13] for an overview). However, for the purpose of TSS's presented in this paper all the existing interpretations coincide (since they are all *strictly stratified*, see [14,11] for the definition and details) and hence, it does not matter which interpretation we choose. Next, we define the notion of *stable model* as an intuitive semantics for TSSs with negative premises.

**Definition 1 (Stable Model).** We say a positive closed formula  $\phi$  is *provable* from a set of positive formulae  $T$  and a TSS  $tss$ , denoted by  $(T, tss) \vdash \phi$  when there is a well-founded upwardly branching tree with nodes labelled by closed formulae such that:

- the root node is labelled by  $\phi$ , and

- if the label of a node  $q$ , denoted by  $\psi$ , is a positive formula and  $\{\psi_i \mid i \in I\}$  is the set of labels of the nodes directly above  $q$ , then there is a deduction rule  $\frac{\{\chi_i \mid i \in I\}}{\chi}$  in  $tss$  (N.B.  $\chi_i$  can be a positive or a negative formula) and a substitution  $\sigma$  such that  $\sigma(\chi) = \psi$  and for all  $i \in I$ ,  $\sigma(\chi_i) = \psi_i$ ;
- if the label of a node  $q$ , denoted by  $p \xrightarrow{l}$ , is a negative formula then there exists no  $p'$  such that  $p \xrightarrow{l} p' \in T$  (or similarly, if it is of the form  $\neg p \downarrow$  then  $p \notin \downarrow$ ).

A *stable model* defined by  $tss$  is a set of formulae  $T$  such that for all closed positive formulae  $\phi$ ,  $\phi \in T$  if and only if  $(T, tss) \vdash \phi$ .

Using the notion of stable model, we can associate a transition system to each closed term in the signature. To define an equality on transition systems and turn them into model for process algebras, we need a notion of behavioral equivalence. Strong bisimilarity is one such notion of behavioral equivalence which can be efficiently checked in practice and usually leads to elegant theories.

**Definition 2 (Bisimilarity).** A symmetric relation  $R$  on closed terms is called a (strong) bisimulation relation with respect to a transition relations  $\rightarrow$  and a predicates  $P$ , when for all  $(p, q) \in R$ , and for all labels  $l$  and closed terms  $p'$ ,

- if  $p \xrightarrow{l} p'$  then there exists a  $q'$  such that  $q \xrightarrow{l} q'$  and  $(p', q') \in R$ , and
- if  $p \downarrow$  then  $q \downarrow$ .

We write  $TSS(A) \models p \Leftrightarrow p'$  for closed terms  $p$  and  $p'$  when they are (strongly) bisimilar with respect to the stable model of  $TSS(A)$ .

To extend  $BPA_\delta^\epsilon$  with timing, we add the unit time transitions  $\mapsto$  (which can be considered as an acronym for  $\xrightarrow{1}$  where 1 is a fresh label dedicated to time transitions). At the same time, we add additional constants  $\underline{\epsilon}$  (termination in the current time slice),  $\underline{\delta}$  (deadlock in the current time slice),  $\underline{a}$  (action execution in the current time slice, for  $a \in A$ ) and  $\sigma$  (unit delay). The resulting extension is called  $BPA_{drt, \delta}^\epsilon$ ; see Table 2.

The first three deduction rules of Table 2 specify the delayable nature of  $\delta$ ,  $\epsilon$  and  $a$  whereas the next three rules specify the undelayable nature of  $\underline{\delta}$ ,  $\underline{\epsilon}$  and  $\underline{a}$ . Let's next focus on the last three deduction rules: they specify the time-deterministic behavior of choice, i.e., time transitions cannot decide about a choice unless one of the two arguments of choice prohibits time from progressing. The remaining deduction rules specify the behavior of sequential composition. These are rather involved since they have to maintain time-determinism. If a delay can take place in two forms, i.e., by delaying the first argument and by delaying the second argument after the termination of the first, then both options are kept open (the first rule). Otherwise, if exactly one of these two forms is possible then there remains no choice and only possible delay takes place (the second and the third rule).

$TSS(BPA_{drt,\delta}^\epsilon(A))$					
$TSS(BPA_\delta^\epsilon(A))$					
constant: $\underline{\delta}, \underline{\epsilon}, (\underline{a})_{a \in A}, \sigma$ ;					
$- \mapsto -;$					
$x, x', y, y'$ ;					
$\delta \mapsto \delta$	$\epsilon \mapsto \epsilon$	$a \mapsto a$	$\underline{\epsilon} \downarrow$	$\underline{a} \xrightarrow{a} \underline{\epsilon}$	$\sigma \mapsto \underline{\epsilon}$
$\frac{x \mapsto x' \quad x \downarrow \quad y \mapsto y'}{x \cdot y \mapsto x' \cdot y + y'}$		$\frac{x \mapsto x' \quad x \not\downarrow}{x \cdot y \mapsto x' \cdot y}$			
$\frac{x \mapsto x' \quad y \not\mapsto}{x \cdot y \mapsto x' \cdot y}$		$\frac{x \not\mapsto \quad x \downarrow \quad y \mapsto y'}{x \cdot y \mapsto y'}$			
$\frac{x \mapsto x' \quad y \mapsto y'}{x + y \mapsto x' + y'}$		$\frac{x \mapsto x' \quad y \not\mapsto}{x + y \mapsto x'}$		$\frac{x \not\mapsto \quad y \mapsto y'}{x + y \mapsto y'}$	

**Table 2.** Transition system specification for  $BPA_{drt,\delta}^\epsilon(A)$ .

While this theory can be worked out in full, and indeed has an elegant axiomatization, it does lead to complications. These complications have to do with the fact that the action constants involve both action execution and termination. In the timed extension, the immediate and delayable options for action execution and termination lead to four different combinations:

- Action execution after an arbitrary delay, followed by termination after an arbitrary delay ( $a$ );
- Immediate action execution followed by immediate termination ( $\underline{a}$ );
- Action execution after an arbitrary delay, followed by immediate termination ( $\epsilon \cdot \underline{a}$ );
- Immediate action execution, followed by termination after an arbitrary delay ( $\underline{a} \cdot \epsilon$ ).

In the extension of  $BPA_\delta^\epsilon$  with timing, we are forced to take the first option. The problem is that this does not match with timed extensions of process algebra without the  $\epsilon$  constant. In  $BPA_\delta$  [7,9], the operational rule for action constants is  $a \xrightarrow{a} \surd$ , where  $\surd$  is not a process expression but a special symbol denoting termination. Extending with time, time transitions can be added before action execution, but not afterwards, so we are forced to take the third option.

In [20], the author follows this, so he interprets untimed actions following the third choice. When extending with  $\epsilon$  he gets  $a = \epsilon \cdot \underline{a}$ . As a result, the extension is not conservative, as the ground equation  $a \cdot \epsilon = a$  does not hold any longer in the timed extension.

To combat this mismatch, in [1] it is proposed to separate action execution and termination: by replacing action constants by action prefixing, termination becomes explicit. As we will show further on, separating action execution and termination by abandoning the idea of basic actions as constructors in the signature resolves many of the difficulties.

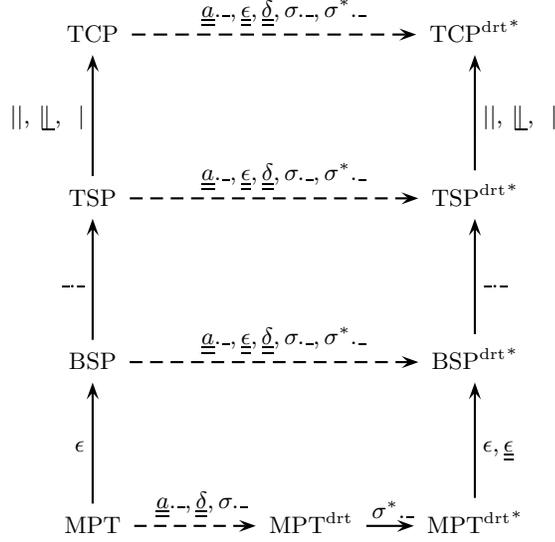
This involves a deviation of a basic design decision in the untimed process algebra: *action prefixing* is taken as the basic composition operator instead of *sequential composition*, and sequential composition is added later on. However, both in [1] and in [6] conservativity is not maintained for the operators  $\parallel$  and  $|$  that are used to define parallel composition. This inadequacy is solved in the present paper.

Thus, we choose a departure point for the extension of process algebra with termination and timing that has action prefixing instead of action constants. Following [1,6], we choose a basic theory called MPT (for Minimal Process Theory) with deadlock, action prefixing and choice. Then, we extend this theory with successful termination which results in the theory BSP. Sequential and parallel composition are subsequently added to BSP, resulting in the theories TSP and TCP, respectively. Observe that *ACP* cannot be considered a conservative (ground-)extension of TCP since the signature of TCP is not contained in the signature of *ACP* and vice versa TCP is not a conservative (ground-)extension of *ACP* since the signature of *ACP* is not contained in the signature of TCP. However, it is possible to embed *ACP* into TCP by mapping the action constants  $a \in A$  on the TCP-terms  $a.\epsilon$ .

As our goal is to have equationally conservative ground-extensions of process algebras, we extend our theory with timing at each level and establish the conservativity of the extension. In order to make the transition to timed settings smoother, in addition to the time delay operator  $\sigma._.$ , we add an arbitrary time delay operator  $\sigma^*._.$  which is very helpful in the axiomatization of complex theories such as TSP and TCP. The result of the extension of theory  $X$  with undelayable action prefix, undelayable termination and deadlock, (discrete) time delay and arbitrary delay operators is denoted by  $X^{drt^*}$ . The lattice of process theories that we present in this paper (ordered by equational (ground-)conservativity relation) is depicted in Figure 1. Each arrow is labelled by the function symbols introduced in the target of the extension.

To give a formal meaning to the arrows presented in Figure 1, we define a few concepts regarding conservativity. The first definition concerns the traditional notion of operational conservativity.

**Definition 3 (Operational Conservativity [12]).** Consider TSS's  $TSS(A)$  and  $TSS(B)$  defined on signatures  $\Sigma_A$  and  $\Sigma_B$  such that  $TSS(B)$  includes  $TSS(A)$  in its definition. Also, let  $C(\Sigma_x)$  denote the set of closed terms built upon  $\Sigma_x$ . The TSS  $TSS(B)$  is an *operationally conservative extension* of  $TSS(A)$  when  $\forall p \in C(\Sigma_A) \quad TSS(A) \vDash p \downarrow \Leftrightarrow TSS(B) \vDash p \downarrow$ , and  $\forall p \in C(\Sigma_A) \quad \forall l \in L_B \quad \forall p' \in C(\Sigma_B) \quad TSS(A) \vDash p \xrightarrow{l} p' \Leftrightarrow TSS(B) \vDash p \xrightarrow{l} p'$ .



**Fig. 1.** The Lattice of Process Theories and Their Timed Extensions

In the above definition, implicitly it is not allowed to have old relations from old terms ( $C(\Sigma_P)$ ) to a new term ( $C(\Sigma_B)$ ). Note that the transition relations and predicates in the above definition are taken from the extended TSS, i.e.,  $TSS(B)$  and hence an operationally conservative extension denies any new transition or predicate from the terms from the old syntax, i.e., from  $C(\Sigma_A)$ . This turns out to be too restrictive for time extensions since we decided to interpret untimed basic actions as delayable and hence we have to add timing behavior to them. This has been noted in [18] where the following alternative and more relaxed notion of orthogonality was proposed.

**Definition 4 (Orthogonality [18]).** Consider TSS's  $TSS(A)$  and  $TSS(B)$  defined on signatures  $\Sigma_A$  and  $\Sigma_B$  such that  $TSS(B)$  includes  $TSS(A)$  in its definition. The TSS  $TSS(B)$  is an *orthogonal* extension of  $TSS(A)$  when

1.  $\forall_{p \in C(\Sigma_A)} TSS(A) \models p \downarrow \Leftrightarrow TSS(B) \models p \downarrow$  and  
 $\forall_{p, p' \in C(\Sigma_A)} \forall_{l \in L_A} TSS(A) \models p \xrightarrow{l} p' \Leftrightarrow TSS(B) \models p \xrightarrow{l} p'$  and
2.  $\forall_{p, p' \in C(\Sigma_A)} TSS(A) \models p \Leftrightarrow p' \Leftrightarrow TSS(B) \models p \Leftrightarrow p'$ .

Note that it follows from Definitions 3 and 4 that an operationally conservative extension is orthogonal [18]. Both operational conservativity and orthogonality are useful means to obtain equational conservativity as defined below.

**Definition 5 (Equational Conservativity [18]).** An equational theory  $B$  on signature  $\Sigma_B$  is an *equationally conservative ground-extension* of equational the-

ory  $A$  on  $\Sigma_A$  if and only if  $\Sigma_A \subseteq \Sigma_B$  and for all  $p, p' \in C(\Sigma_A)$ ,  $A \vdash p = p' \Leftrightarrow B \vdash p = p'$ .

If the axioms of equational theory  $A$  are (syntactically) included among the axioms of equational theory  $B$  and  $B$  is an equationally conservative ground-extension of  $A$ , then  $B$  is an *equationally conservative extension* of  $A$ .

In the above definition  $X \vdash p = p'$  means that  $p = p'$  is *derivable* from the equations in  $X$ . In our settings, this means that  $=$  is the congruence relation induced by the equations in  $X$ . We drop the prefixes equational and equationally and simply write conservativity and conservative (ground)-extension in the remainder.

In the outline presented in Figure 1, normal arrows denote equationally conservative extensions and dashed arrows denote equationally conservative ground-extensions. Operational conservativity and orthogonality are mostly used as a means to prove conservative extensions and conservative ground-extensions, respectively. In the present paper, we focus on the process theories and on their interrelationships using the previously mentioned notions of conservativity. We present soundness and ground-completeness theorems for all theories, but omit their proofs altogether.

### 3 Minimal Process Theory

#### 3.1 MPT

The equational theory Minimal Process Theory (MPT) is specified in Table 3.

_ MPT(A) _____		
constant: $\delta$ ;    unary: $(a._)_{a \in A}$ ;    binary: $_ + _$ ;		
$x, y, z$ ;		
$x + y = y + x$	$A1$	
$(x + y) + z = x + (y + z)$	$A2$	
$x + x = x$	$A3$	
$x + \delta = x$	$A6$	
_____		

**Table 3.** MPT(A)

The transition system specification associated to the terms of MPT is given in Table 4.

It is straightforward to check that the equational theory of MPT is a sound and ground-complete axiomatization for its transition system semantics modulo strong bisimulation.

— $TSS(MPT(A))$ —		
constant: $\delta$ ;	unary: $(a.\_ )_{a \in A}$ ;	binary: $- + \_;$
$(\_ \xrightarrow{a} \_ )_{a \in A}$		
$x, x', y, y'$ ;		
$a.x \xrightarrow{a} x$	$\frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'}$	$\frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$

**Table 4.** Transition system specification for  $MPT(A)$ .

**Theorem 6 (Soundness).** *Let  $p$  and  $q$  be two closed  $MPT(A)$ -terms. If  $MPT(A) \vdash p = q$ , then  $TSS(MPT(A)) \vDash p \Leftrightarrow q$ .*

**Theorem 7 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed  $MPT(A)$ -terms. If  $TSS(MPT(A)) \vDash p \Leftrightarrow q$ , then  $MPT(A) \vdash p = q$ .*

### 3.2 $MPT^{\text{drt}}$

The equational theory of  $MPT$  with discrete relative timing ( $MPT^{\text{drt}}$ ) [1] is specified in Table 5. It adds undelayable action prefixing  $\underline{a}.\_$ , undelayable deadlock  $\underline{\delta}$  plus a time delay operator  $\sigma.\_$  to the signature of  $MPT$  and uses no auxiliary operators for the axiomatization. Prefixing binds stronger than the other operators.

— $MPT^{\text{drt}}(A)$ —			
constant: $\delta, \underline{\delta}$ ;	unary: $(a.\_ )_{a \in A}, (\underline{a}.\_ )_{a \in A}, \sigma.\_;$	binary: $- + \_;$	
$x, y, z$ ;			
$x + y = y + x$	A1	$\delta = \sigma.\delta$	DD1
$x + (y + z) = (x + y) + z$	A2	$a.x = \underline{a}.x + \sigma.a.x$	DA1
$x + x = x$	A3	$a.x + \delta = a.x$	A6DD
$x + \underline{\delta} = x$	A6DRT	$\sigma.x + \sigma.y = \sigma.(x + y)$	DRTF

**Table 5.**  $MPT^{\text{drt}}(A)$

This equational theory is not a conservative extension of  $MPT$  since the axiom  $x + \delta = x$  of  $MPT$  does not hold anymore. The role of deadlock  $\delta$  (now called delayable deadlock) as a unit element for alternative composition is taken over by undelayable deadlock  $\underline{\delta}$ , see Axiom A6DRT. The behavior of delayable

deadlock and action prefix is defined recursively by means of the axioms DD1 (Delayable Deadlock) and DA1 (Delayable Action). Axiom A6DD expresses that delayable deadlock is still a unit element for alternative composition of delayable processes. Finally, axiom DRTF (Discrete Relative Time Factorization) expresses that passage of time by itself cannot determine a choice. Hence, this axiom implements the time-determinism discussed in the previous section.

The transition system specification of  $\text{MPT}^{\text{drt}}$  is given in Table 6. It consists of the deduction rules of MPT (Table 4) as well as new deduction rules defining the time transitions  $\mapsto$  for MPT terms as well as action and time transitions for the newly introduced terms.

$\frac{\text{--- } TSS(\text{MPT}^{\text{drt}}(A)) \text{---}}{TSS(\text{MPT}(A))}$		
constant: $\underline{\delta}$ ;    unary: $(\underline{a.})_{a \in A}, \sigma.;$		
$\frac{- \mapsto -;}{x, x', y, y';}$		
$\delta \mapsto \delta$	$a.x \mapsto a.x$	$\frac{x \mapsto x' \quad y \mapsto y'}{x + y \mapsto x' + y'}$
$\frac{x \mapsto x' \quad y \not\mapsto}{x + y \mapsto x'}$		
$\frac{x \not\mapsto \quad y \mapsto y'}{x + y \mapsto y'}$		
$\underline{a.x} \xrightarrow{a} x \quad \sigma.x \mapsto x$		

**Table 6.** Transition system specification for  $\text{MPT}^{\text{drt}}(A)$ .

**Theorem 8 (Soundness).** *Let  $p$  and  $q$  be two closed  $\text{MPT}^{\text{drt}}(A)$ -terms. If  $\text{MPT}^{\text{drt}}(A) \vdash p = q$ , then  $TSS(\text{MPT}^{\text{drt}}(A)) \models p \Leftrightarrow q$ .*

**Theorem 9 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed  $\text{MPT}^{\text{drt}}(A)$ -terms. If  $TSS(\text{MPT}^{\text{drt}}(A)) \models p \Leftrightarrow q$ , then  $\text{MPT}^{\text{drt}}(A) \vdash p = q$ .*

As explained before,  $\text{MPT}^{\text{drt}}$  cannot be a conservative extension of MPT due to the omission of Axiom A6 ( $x + \delta = x$ ) from  $\text{MPT}^{\text{drt}}$ . It is a conservative ground-extension, nevertheless, since for closed MPT-terms  $x$ , this axiom is derivable from the axioms of  $\text{MPT}^{\text{drt}}$ .

**Theorem 10 (Conservative ground-extension).**  *$\text{MPT}^{\text{drt}}$  is a conservative ground-extension of MPT.*

*Proof.* We apply the meta-theorems from [18]. The transition system specification that consists of the deduction rules in the first row of Table 6 is source-preserving and strictly stratified using the number of symbols of terms as a measure, and the sources of the conclusions cover the syntax of MPT (see [18] for definitions of notions used here). Furthermore, the deduction rules in the second row of Table 6 have source-dependent negative time transitions as a premise. Hence  $TSS(\text{MPT}(A))$  with deduction rules of the first and the second row of Table 6 is a granting extension of  $TSS(\text{MPT}(A))$  and hence an orthogonal extension. The extension of the resulting transition system specification with deduction rules of the third row is conservative, hence also orthogonal. Since orthogonality is a preorder,  $TSS(\text{MPT}^{\text{drt}}(A))$  is an orthogonal extension of  $TSS(\text{MPT}(A))$ . Combined with the facts that both MPT and  $\text{MPT}^{\text{drt}}$  are sound and complete, we have that  $\text{MPT}^{\text{drt}}$  is an equationally conservative ground-extension of MPT.

### 3.3 $\text{MPT}^{\text{drt}*}$

A further discrete relative time extension of MPT, called  $\text{MPT}^{\text{drt}*}$  makes use of an auxiliary operator  $\sigma^*_-$  to axiomatize this extension:  $\sigma^*p$  denotes that the execution of  $p$  can be started in any time slice (present or future). Note that the intuitions of  $\delta$  and  $a._-$  are in line with the “any-time-slice” interpretation of the untimed constants and action prefix operators. This time iteration operator comes very handy in the axiomatization of delayable actions, particularly in the more involved theories that we encounter in the rest of this paper. The axiomatization of  $\text{MPT}^{\text{drt}*}$  is given in Table 7.

$\text{MPT}^{\text{drt}*}(A)$			
constant: $\delta, \underline{\delta}$ ;    unary: $(a._-)_{a \in A}, (\underline{a._-})_{a \in A}, \sigma._-, \sigma^*_-$ ;    binary: $_+ \_-$			
$x, y, z;$			
$x + y = y + x$	A1	$\delta = \sigma.\delta$	DD1
$x + (y + z) = (x + y) + z$	A2	$a.x = \underline{a}.x + \sigma.a.x$	DA1
$x + x = x$	A3	$\delta = \sigma^*\underline{\delta}$	DD2
		$a.x = \sigma^*\underline{a}.x$	DA2
$x + \underline{\delta} = x$	A6DRT	$\sigma.x + \sigma.y = \sigma.(x + y)$	DRTF
$\sigma^*x = x + \sigma.\sigma^*x$	ATS	$\sigma^*x + \sigma^*y = \sigma^*(x + y)$	DRTIF
$\sigma^*\sigma.x = \sigma.\sigma^*x$	DRTA	$\sigma^*\sigma^*x = \sigma^*x$	TITI

**Table 7.**  $\text{MPT}^{\text{drt}*}(A)$

In this equational theory the any time slice constant and action prefix operators are defined in terms of their current time slice counterparts and time iteration by axioms DD2 and DA2. Axiom ATS (Any Time Slice) recursively

defines time iteration. Axiom DRTIF (Discrete Relative Time Iteration Factorization) expresses that time factorization also applies to time iteration. Axiom DRTA (Discrete Relative Time Axiom) explains that consecutive occurrences of time iteration and time delay can be switched. It is in line with the observation that both  $\sigma.\sigma^*p$  and  $\sigma^*\sigma.p$  are solutions to the recursive specification  $X = \sigma.p + \sigma.X$ . Axiom TITI (Time Iteration Time Iteration) says that two consecutive time iterations are equivalent to only one time iteration. As a consequence, any number of consecutive time iterations is considered to be equivalent to a single one. Note that axioms DD1 and DA1 are derivable from the other axioms (e.g.,  $\delta = \sigma^*\underline{\delta} = \underline{\delta} + \sigma\sigma^*\underline{\delta} = \sigma\delta$ ).

It is not the case that the newly introduced operators can all be eliminated. Nevertheless, the newly introduced syntax has some redundancy in the sense that either the time iteration operator or the delayable deadlock and delayable action prefixes can be eliminated from closed terms.

The transition system specification associated to the terms of  $\text{MPT}^{\text{drt}^*}$  is given in Table 8. It adds deduction rules defining the behavior of time iteration.

$$\begin{array}{c}
 \frac{}{TSS(\text{MPT}^{\text{drt}^*}(A))} \text{-----} \\
 TSS(\text{MPT}^{\text{drt}}(A)) \\
 \hline
 \text{unary: } \sigma^* \text{ -;} \\
 \hline
 \\
 \hline
 x, x'; \\
 \\
 \frac{x \xrightarrow{a} x'}{\sigma^*x \xrightarrow{a} x'} \quad \frac{x \mapsto x'}{\sigma^*x \mapsto \sigma^*x + x'} \quad \frac{x \not\mapsto}{\sigma^*x \mapsto \sigma^*x} \\
 \hline
 \end{array}$$

**Table 8.** Transition system specification for  $\text{MPT}^{\text{drt}^*}(A)$ .

**Theorem 11 (Soundness).** *Let  $p$  and  $q$  be two closed  $\text{MPT}^{\text{drt}^*}(A)$ -terms. If  $\text{MPT}^{\text{drt}^*}(A) \vdash p = q$ , then  $TSS(\text{MPT}^{\text{drt}^*}(A)) \vDash p \Leftrightarrow q$ .*

**Theorem 12 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed  $\text{MPT}^{\text{drt}^*}(A)$ -terms. If  $TSS(\text{MPT}^{\text{drt}^*}(A)) \vDash p \Leftrightarrow q$ , then  $\text{MPT}^{\text{drt}^*}(A) \vdash p = q$ .*

**Theorem 13 (Conservative extension).**  *$\text{MPT}^{\text{drt}^*}$  is a conservative extension of  $\text{MPT}^{\text{drt}}$ .*

*Proof.* The sources of the conclusions of all deduction rules from Table 8 mention a new operator. Therefore, the extension of  $TSS(\text{MPT}^{\text{drt}}(A))$  with these deduction rules is conservative. Hence,  $TSS(\text{MPT}^{\text{drt}^*}(A))$  is an orthogonal extension of  $TSS(\text{MPT}^{\text{drt}}(A))$ . Since both theories are sound and complete and the axioms of  $\text{MPT}^{\text{drt}}(A)$  are included in  $\text{MPT}^{\text{drt}^*}(A)$ , it follows from the meta-results of [18] that  $\text{MPT}^{\text{drt}^*}$  is an equationally conservative extension of  $\text{MPT}^{\text{drt}}$ .

## 4 Successful termination: Basic Sequential Processes

In this section, we discuss the extension of the theories MPT and  $\text{MPT}^{\text{drt}^*}$  from the previous section, with a constant denoting successful termination.

### 4.1 BSP

The process theory MPT is a *minimal* theory; not much can be expressed in it. One aspect that cannot be addressed is successful termination. The distinction between successful and unsuccessful termination turns out to be essential when sequential composition is introduced. In order to express successful termination, the new constant  $\epsilon$ , referred to as the empty process or the termination constant, is introduced. The extension of the process theory MPT with the empty process  $\epsilon$  results in process theory BSP, the theory of Basic Sequential Processes. This section gives the equational theory as well as its term model.

Table 9 defines process theory BSP. The only difference between the signature of MPT and the signature of BSP is the constant  $\epsilon$ . The axioms of BSP, see Table 9, are exactly the axioms of MPT, given in Table 3.

$\frac{\text{--- BSP}(A) \text{---}}{\text{MPT}(A)}$ $\frac{\text{---}}{\text{constant: } \epsilon;}$ $\text{---}$
---

**Table 9.** BSP( $A$ )

The transition system specification associated to the terms of BSP is given in Table 10. It adds deduction rules defining the termination behavior of the new constant  $\epsilon$  and of the syntax of MPT.

**Theorem 14 (Soundness).** *Let  $p$  and  $q$  be two closed BSP( $A$ )-terms. If  $\text{BSP}(A) \vdash p = q$ , then  $TSS(\text{BSP}(A)) \vDash p \Leftrightarrow q$ .*

**Theorem 15 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed BSP( $A$ )-terms. If  $TSS(\text{BSP}(A)) \vDash p \Leftrightarrow q$ , then  $\text{BSP}(A) \vdash p = q$ .*

**Theorem 16 (Conservative extension).** *BSP is a conservative extension of MPT.*

*Proof.* Both MPT and BSP are sound and ground-complete equational theories for  $TSS(\text{MPT}(A))$  and  $TSS(\text{BSP}(A))$ . Also, following [12],  $TSS(\text{BSP}(A))$  is an operationally conservative (orthogonal) extension of  $TSS(\text{MPT}(A))$ . Furthermore, axioms of  $\text{MPT}(A)$  are all included in  $\text{BSP}(A)$ . Thus, we conclude that  $\text{BSP}(A)$  is an equationally conservative extension of  $\text{MPT}(A)$ .

$$\begin{array}{c}
\frac{}{TSS(BSP(A))} \text{-----} \\
TSS(MPT(A)) \\
\text{constant: } \epsilon; \\
\hline
-\downarrow; \\
\hline
x, y; \\
\hline
\epsilon \downarrow \quad \frac{x \downarrow}{x + y \downarrow} \quad \frac{y \downarrow}{x + y \downarrow} \\
\hline
\end{array}$$

**Table 10.** Transition system specification for  $BSP(A)$ .

## 4.2 $BSP^{\text{drt}^*}$

In this section, the timed process algebra  $MPT^{\text{drt}^*}$  from the previous section is extended with the constants *any time slice termination*  $\epsilon$  and *current time slice termination*  $\underline{\epsilon}$ . The axioms of the process theory  $BSP^{\text{drt}^*}$  are given in Table 11. The axiom DT (Delayable Termination) defines the any time slice constant in terms of its current time slice counterpart and time iteration. Note that  $\epsilon = \underline{\epsilon} + \sigma^* \epsilon$  is derivable from the axioms of Table 11.

$$\begin{array}{c}
\frac{}{BSP^{\text{drt}^*}(A)} \text{-----} \\
MPT^{\text{drt}^*}(A) \\
\text{constant: } \epsilon, \underline{\epsilon}; \\
\hline
\epsilon = \sigma^* \underline{\epsilon} \text{ DT} \\
\hline
\end{array}$$

**Table 11.**  $BSP^{\text{drt}^*}(A)$

The transition system specification associated to the terms of  $BSP^{\text{drt}^*}$  is given in Table 12. It adds deduction rules defining the termination behavior of the new constant  $\underline{\epsilon}$  and the time iteration  $\sigma^*$  and the time behavior of  $\epsilon$ .

**Theorem 17 (Soundness).** *Let  $p$  and  $q$  be two closed  $BSP^{\text{drt}^*}(A)$ -terms. If  $BSP^{\text{drt}^*}(A) \vdash p = q$ , then  $TSS(BSP^{\text{drt}^*}(A)) \models p \Leftrightarrow q$ .*

**Theorem 18 (Ground-completeness).** *Let  $p$  and  $q$  be two arbitrary closed  $BSP^{\text{drt}^*}(A)$ -terms. If  $TSS(BSP^{\text{drt}^*}(A)) \models p \Leftrightarrow q$ , then  $BSP^{\text{drt}^*}(A) \vdash p = q$ .*

**Theorem 19 (Conservative ground-extension).**  *$BSP^{\text{drt}^*}$  is a conservative ground-extension of  $BSP$ .*

$\frac{\text{--- } TSS(\text{BSP}^{\text{drt}*}(A)) \text{---}}{TSS(\text{BSP}(A)), TSS(\text{MPT}^{\text{drt}*}(A))}$		
constant: $\underline{\epsilon}$ ;		
$x$ ;		
$\underline{\epsilon} \downarrow$	$\epsilon \mapsto \epsilon$	$\frac{x \downarrow}{\sigma^* x \downarrow}$

**Table 12.** Transition system specification for  $\text{BSP}^{\text{drt}*}(A)$ .

*Proof.* This proof is similar to the proof that  $\text{MPT}^{\text{drt}*}$  is a conservative ground-extension of  $\text{MPT}$ . Add the second deduction rule of 12 to the granting part and the first and third deduction rules of Table 12 to the conservative part. Note that the deduction rules for  $\sigma^*$  that are added to  $TSS(\text{MPT}^{\text{drt}}(A))$  to obtain  $TSS(\text{MPT}^{\text{drt}*}(A))$  should be added to the conservative part as well.

**Theorem 20 (Conservative extension).**  $\text{BSP}^{\text{drt}*}$  is a conservative extension of  $\text{MPT}^{\text{drt}*}$ .

*Proof.* Both  $\text{MPT}^{\text{drt}*}$  and  $\text{BSP}^{\text{drt}*}$  are sound and ground-complete equational theories for  $TSS(\text{MPT}^{\text{drt}*}(A))$  and  $TSS(\text{BSP}^{\text{drt}*}(A))$ . Also, using the meta-theory from [12],  $TSS(\text{BSP}^{\text{drt}*}(A))$  is an operationally conservative (orthogonal) extension of  $TSS(\text{MPT}^{\text{drt}*}(A))$ . Thus, we conclude that  $\text{BSP}^{\text{drt}*}(A)$  is an equationally conservative extension of  $\text{MPT}^{\text{drt}*}(A)$ .

## 5 Sequential composition

### 5.1 TSP

This section treats the extension with a *sequential composition* operator. Given two process terms  $p$  and  $q$ , the term  $p \cdot q$  denotes the sequential composition of  $p$  and  $q$ . The intuition of this operation is that upon the *successful* termination of process  $p$ , process  $q$  is started. If process  $p$  ends in a deadlock, also the sequential composition  $p \cdot q$  deadlocks. Thus, a pre-requisite for a meaningful introduction of a sequential composition operator is that successful and unsuccessful termination can be distinguished. As already explained in Section 4, this is not possible in the theory  $\text{MPT}$  as all processes end in deadlock. Thus, as a starting point the theory  $\text{BSP}$  of the previous section is used. This theory is extended with sequential composition to obtain the *Theory of Sequential Processes* TSP. It turns out that the empty process is a neutral element for sequential composition:  $x \cdot \epsilon = \epsilon \cdot x = x$ .

To obtain the axioms of the process theory TSP, the axioms from Table 13 are added to the axioms of the process theory  $\text{BSP}$  from Table 9. Axiom A5 states

TSP(A)		
BSP(A)		
binary: $\cdot$ $\cdot$ $\cdot$ ;		
$x, y, z : P$ ;		
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4	$\delta \cdot x = \delta$ A7
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5	$x \cdot \epsilon = x$ A8
$a.x \cdot y = a.(x \cdot y)$	A10	$\epsilon \cdot x = x$ A9

**Table 13.** The process theory TSP(A).

that sequential composition is associative. As mentioned before, and now formally captured in the axioms A8 and A9, the empty process is a neutral element with respect to sequential composition. Axiom A7 states that after a deadlock has been reached no continuation is possible. Axiom A4 describes the distribution of sequential composition over alternative composition from the right. Recall that the other distributivity property is not desired as it does not respect the moment of choice. Finally, Axiom A10 describes the relation between sequential composition and action prefixes.

**Theorem 21 (Elimination).** *For any closed TSP(A)-term  $p$ , there exists a closed BSP(A)-term  $q$  such that  $\text{TSP}(A) \vdash p = q$ .*

*Proof.* The property is proven by providing a term rewriting system with the same signature as TSP(A) such that

1. each rewrite step transforms a process term into a process term that is derivably equal,
2. the term rewriting system is strongly normalizing, and
3. no closed normal form of the term rewriting system contains a sequential composition operator.

Consider the term rewriting system consisting of the following rewrite rules; for any  $a \in A$ , and TSP(A)-terms  $x, y, z$ :

$$\begin{array}{ll}
 (x + y) \cdot z \rightarrow x \cdot z + y \cdot z & \delta \cdot x \rightarrow \delta \\
 (x \cdot y) \cdot z \rightarrow x \cdot (y \cdot z) & \epsilon \cdot x \rightarrow x \\
 a.x \cdot y \rightarrow a.(x \cdot y) &
 \end{array}$$

Each of the rewrite rules is obtained directly from an axiom of TSP by replacing  $=$  by  $\rightarrow$ . As a consequence, each rewrite step transforms a process term in a process term that is derivably equal.

The second step of the proof, the strong normalization of the given term rewriting system, is standard.

The last part of the proof is to show that no closed normal form of the above term rewriting system contains a sequential composition operator. Thereto, let

$u$  be a normal form of the above term rewriting system. Suppose that  $u$  contains at least one sequential composition operator. Then,  $u$  must contain a subterm of the form  $v \cdot w$  for some closed TSP( $A$ )-terms  $u$  and  $v$ . This subterm can always be chosen in such a way that  $v$  is a closed BSP( $A$ )-term. It follows immediately from the structure of closed BSP( $A$ )-terms that one of the above rewrite rules can be applied to  $v \cdot w$ . As a consequence,  $u$  is not a normal form. This contradiction implies that  $u$  must be a closed BSP( $A$ )-term.

The transition system specification associated to the terms of TSP is given in Table 14. It adds deduction rules defining action transitions and termination behavior of sequential composition.

$$\begin{array}{c}
\frac{}{TSS(\text{TSP}(A))} \text{-----} \\
TSS(\text{BSP}(A)) \\
\text{binary: } \cdot \ ; \\
\text{-----} \\
x, y, x', y'; \\
\frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y} \quad \frac{x \downarrow \quad y \xrightarrow{a} y'}{x \cdot y \xrightarrow{a} y'} \quad \frac{x \downarrow \quad y \downarrow}{x \cdot y \downarrow} \\
\text{-----}
\end{array}$$

**Table 14.** Transition system specification for TSP( $A$ ).

**Theorem 22 (Soundness).** *Let  $p$  and  $q$  be two closed TSP( $A$ )-terms. If  $\text{TSP}(A) \vdash p = q$ , then  $TSS(\text{TSP}(A)) \vDash p \Leftrightarrow q$ .*

**Theorem 23 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed TSP( $A$ )-terms. If  $TSS(\text{TSP}(A)) \vDash p \Leftrightarrow q$ , then  $\text{TSP}(A) \vdash p = q$ .*

**Theorem 24 (Conservative extension).** *TSP is a conservative extension of BSP.*

*Proof.* Both BSP and TSP are sound and ground-complete equational theories for  $TSS(\text{BSP}(A))$  and  $TSS(\text{TSP}(A))$ . Also, following [12],  $TSS(\text{TSP}(A))$  is an operationally conservative (orthogonal) extension of  $TSS(\text{BSP}(A))$ . Furthermore, all axioms of BSP are among the axioms of TSP. Thus, we conclude that TSP is an equationally conservative extension of BSP.

Now that the process theories have been extended with sequential composition, the relationship with the process theory  $BPA_5^c$  can be considered. Syntactically, there is still a mismatch between  $BPA_5^c$  and TSP since the former has constants  $a \in A$  and the latter does not have those. There are two (equivalent)

ways to overcome this difference. First, we can extend TSP with such constants, using axiom

$$a = a.\epsilon,$$

or second, we can use the notion of embedding to find that the process theory  $BPA_\delta^\epsilon$  can be embedded into TSP taking  $a$  as  $a.\epsilon$ .

## 5.2 TSP<sup>drt\*</sup>

In this section, the process theory  $BSP^{\text{drt}^*}$  is extended to the process theory  $TSP^{\text{drt}^*}$ . This extension is obtained by extending the signature of  $BSP^{\text{drt}^*}$  with the *sequential composition* operator  $\cdot$ . The axioms of the process theory are given in Table 15.

$\frac{}{BSP^{\text{drt}^*}(A)}$	
binary: $\cdot$ ;	
$x, y, z : P;$	
$(x + y) \cdot z = x \cdot z + y \cdot z$	A4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	A5
$\underline{a}.x \cdot y = \underline{a}.(x \cdot y)$	A10DRa
$(\sigma.x) \cdot y = \sigma.(x \cdot y)$	A10DRb
$\sigma^*.x \cdot y = \sigma^*(x \cdot y)$	A10DRc
$\underline{\delta} \cdot x = \underline{\delta}$	A7DR
$x \cdot \underline{\epsilon} = x$	A8DR
$\underline{\epsilon} \cdot x = x$	A9DR

**Table 15.** The process theory  $TSP^{\text{drt}^*}(A)$ .

Sequential composition is as before, but here the role of unit that was played by  $\epsilon$  in the untimed theory, is taken over by the current time slice termination constant  $\underline{\epsilon}$  (see Axioms A8DR and A9DR). In this setting it can be derived that  $\epsilon \cdot x = \sigma^*.x$  instead. The sequential composition operator has two left-zero elements: both undelayable deadlock (see Axiom A7DR) and delayable deadlock act as such. The axioms  $\delta \cdot x = \delta$  and  $a.x \cdot y = a.(x \cdot y)$  from the untimed theory have disappeared since they are derivable from the remaining axioms<sup>2</sup>. Axioms A10DRb and A10DRc express that the passage of time is measured relative to the previous action and thus has no consequences for the future actions: the timing of  $y$  is relative to the last action of  $x$ , regardless the time prefix or time iteration operator.

**Theorem 25 (Elimination).** *For any closed  $TSP^{\text{drt}^*}(A)$ -term  $p$ , there exists a closed  $BSP^{\text{drt}^*}(A)$ -term  $q$  such that  $TSP^{\text{drt}^*}(A) \vdash p = q$ .*

<sup>2</sup> If these axiom were the only axioms of TSP that disappear, then one could consider keeping them, since it would allow for conservativity instead of ground-conservativity.

The transition system specification associated to the terms of  $\text{TSP}^{\text{drt}^*}$  is given in Table 16. It adds deduction rules defining time transitions of sequential composition.

$$\begin{array}{c}
\frac{}{TSS(\text{TSP}^{\text{drt}^*}(A))} \frac{}{TSS(\text{TSP}(A)), TSS(\text{BSP}^{\text{drt}^*}(A))} \\
\hline
\hline
x, x', y, y'; \\
\frac{x \mapsto x' \quad x \downarrow \quad y \mapsto y'}{x \cdot y \mapsto x' \cdot y + y'} \quad \frac{x \mapsto x' \quad x \not\downarrow}{x \cdot y \mapsto x' \cdot y} \\
\frac{x \mapsto x' \quad y \not\mapsto}{x \cdot y \mapsto x' \cdot y} \quad \frac{x \not\mapsto \quad x \downarrow \quad y \mapsto y'}{x \cdot y \mapsto y'} \\
\hline
\hline
\end{array}$$

**Table 16.** Transition system specification for  $\text{TSP}^{\text{drt}^*}(A)$ .

**Theorem 26 (Soundness).** *Let  $p$  and  $q$  be two closed  $\text{TSP}^{\text{drt}^*}(A)$ -terms. If  $\text{TSP}^{\text{drt}^*}(A) \vdash p = q$ , then  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \Leftrightarrow q$ .*

**Theorem 27 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed  $\text{TSP}^{\text{drt}^*}(A)$ -terms. If  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \Leftrightarrow q$ , then  $\text{TSP}^{\text{drt}^*}(A) \vdash p = q$ .*

The process theory  $\text{TSP}^{\text{drt}^*}$  is not a conservative extension of TSP as was the case for all extensions from untimed to timed process algebra. Besides the untimed identity  $x + \delta = x$  that does not hold in the timed setting, also the untimed identity  $x \cdot \epsilon = x$  does not hold in the timed extension:  $\underline{\epsilon} \cdot \epsilon \neq \underline{\epsilon}$ .

**Theorem 28 (Conservative ground-extension).**  *$\text{TSP}^{\text{drt}^*}$  is a conservative ground-extension of TSP.*

*Proof.* We show that for all  $p$  and  $p'$  in the syntax of TSP,

1.  $TSS(\text{TSP}(A)) \models p \xrightarrow{a} p' \Leftrightarrow TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \xrightarrow{a} p'$ ,
2.  $TSS(\text{TSP}(A)) \models p \downarrow \Leftrightarrow TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \downarrow$ ,
3. there exists a closed  $\text{TSP}(A)$ -term  $q$  such that  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \mapsto q$ ,  
and
4. for all closed  $\text{TSP}^{\text{drt}^*}(A)$ -terms  $q$  such that  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \mapsto q$  then  
 $TSS(\text{TSP}(A)) \models p \Leftrightarrow q$  and  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \Leftrightarrow q$ .

If we show the above list of items to be true, it follows that  $TSS(\text{TSP}^{\text{drt}^*}(A))$  is an orthogonal extension of  $TSS(\text{TSP}(A))$ : The first condition of orthogonality is item 1 in the above list. So, it only remains to show that  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p \Leftrightarrow p' \Leftrightarrow TSS(\text{TSP}(A)) \models p \Leftrightarrow p'$ , which follows immediately from the above items.

To check the first and the second item, one should note that first, a proof in  $\text{TSP}^{\text{drt}^*}$  for an  $a$ -transition  $\xrightarrow{a}$  with a source term from TSP or a termination predicate  $\downarrow$  on a TSP-term only involves deduction rules from TSP. Such deduction rules do not have negative premises and are source-dependent [18]. Hence, all such transition and predicate formulae are included in the stable model of TSP. Second, for the inclusion in the other direction, all proofs in TSP remain valid in  $\text{TSP}^{\text{drt}^*}$  and since deduction rules of TSP do not have negative premises, all the proven transitions and predicates of TSP are included in the stable model of  $\text{TSP}^{\text{drt}^*}$ .

We prove the last two items in one go. To that end, we use a structural induction on closed TSP( $A$ )-term  $p$ .

If  $p$  is a constant, i.e.,  $\delta$  or  $\epsilon$ , then it can make a self time transition using one of the following deduction rules (from Table 6 and Table 12 respectively) and these are the only matching rules for such constants to make a time transition.

$$\delta \mapsto \delta \quad \epsilon \mapsto \epsilon$$

Bisimilarity (w.r.t. both TSS's) is reflexive and hence self-transitions satisfy the criteria of item 4.

If  $p$  is of the form  $a.p'$  (for some closed TSP( $A$ )-term  $p'$ ) then it can make a self time transition due to the following deduction rule (from Table 6) and this is the only matching deduction rule for  $p$  to make a time transition.

$$a.x \mapsto a.x$$

If  $p$  is of the form  $p' + q'$ , then by the induction hypothesis,  $p'$  and  $q'$  can make time transitions and all their time transitions are to bisimilar terms (w.r.t. both TSS's). Then  $p' + q'$  can make time transitions using the following deduction rule (from Table 6).

$$\frac{x \mapsto x' \quad y \mapsto y'}{x + y \mapsto x' + y'}$$

and since bisimilarity is a congruence for both  $TSS(\text{TSP}(A))$  and  $TSS(\text{TSP}^{\text{drt}^*}(A))$  (both TSS's are in the PANTH format of [21]), time transitions of  $p' + q'$  that are due to this rule are to bisimilar terms w.r.t. both TSS's. Furthermore,  $p' + q'$  cannot make a time transition using the other two deduction rules for choice in Table 6 since both of them have negative premises denying a time transition from  $p'$  or  $q'$ . Thus, all transitions of  $p' + q'$  are to bisimilar terms w.r.t. both TSS's.

If  $p$  is of the form  $p'.q'$ , then either  $TSS(\text{TSP}(A)) \models p' \downarrow$  (hence, following item 2,  $TSS(\text{TSP}^{\text{drt}^*}(A)) \models p' \downarrow$ ) or  $\neg TSS(\text{TSP}(A)) \models p' \downarrow$  (hence,  $TSS(\text{TSP}^{\text{drt}^*}(A)) \not\models$

$p' \downarrow$ ). Then, by induction hypothesis,  $p' \cdot q'$  can make a time transition due to the first or second deduction rule of Table 16 given below, respectively,

$$\frac{x \mapsto x' \quad x \downarrow \quad y \mapsto y'}{x \cdot y \mapsto x' \cdot y + y'} \quad \frac{x \mapsto x' \quad x \not\downarrow}{x \cdot y \mapsto x' \cdot y}$$

and these are the only possibilities for  $p$  to make time transitions as the other two deduction rules of Table 16 deny time transitions of  $p'$  or  $q'$ . Suppose that  $p' \mapsto p''$ ,  $q' \mapsto q''$  and  $p' \rightleftharpoons p''$  and  $q' \rightleftharpoons q''$  w.r.t. both TSS's. If  $p' \downarrow$  and the time transition of  $p$  is due the left-hand-side deduction rule, then it is easy to check that  $p' \rightleftharpoons p' + \epsilon$  w.r.t. both TSS's. Hence,  $p' \cdot q' \rightleftharpoons (p' + \epsilon) \cdot q' \rightleftharpoons p' \cdot q' + \epsilon \cdot q' \rightleftharpoons p' \cdot q' + q' \rightleftharpoons p'' \cdot q'' + q''$  (following the axioms of both  $\text{TSP}^{\text{drt}^*}$  and TSP, Theorems 22 and 26 and congruence of bisimilarity for both TSS's). Hence, in this case  $p$  makes a time transition to bisimilar terms w.r.t. both TSS's. If  $p' \not\downarrow$  and the transition is due to the right-hand-side rule, then  $p' \cdot q' \rightleftharpoons p'' \cdot q'$  and again the transition of  $p$  is to a bisimilar term w.r.t. both TSS's.

**Theorem 29 (Conservative extension).**  $\text{TSP}^{\text{drt}^*}$  is a conservative extension of  $\text{BSP}^{\text{drt}^*}$ .

*Proof.* Both  $\text{BSP}^{\text{drt}^*}$  and  $\text{TSP}^{\text{drt}^*}$  are sound and ground-complete equational theories for both  $\text{TSS}(\text{BSP}^{\text{drt}^*}(A))$  and  $\text{TSS}(\text{TSP}^{\text{drt}^*}(A))$ . Also, following [12],  $\text{TSS}(\text{TSP}^{\text{drt}^*}(A))$  is an operationally conservative (orthogonal) extension of  $\text{TSS}(\text{BSP}^{\text{drt}^*}(A))$ . Thus, we conclude that  $\text{TSP}^{\text{drt}^*}$  is an equationally conservative extension of  $\text{BSP}^{\text{drt}^*}$ .

## 6 Parallel composition

### 6.1 TCP

The formal definition of process theory TCP, the Theory of Communicating Processes, is given in Table 17. The theory includes the encapsulation operator, as this operator is essential to enforce communication between processes. As before, it has as a parameter the set of actions  $A$ . Besides this, it has as a second parameter a commutative and associative partial communication function  $\gamma : A \times A \rightarrow A$ . The signature of process theory TCP extends the signature of the process theory TSP with the merge operator  $\parallel$ , the left merge operator  $\ll$ , the synchronization merge operator  $|$  and the encapsulation operator  $\partial_H$ . The four new operators bind stronger than choice but weaker than action prefix.

Following the, by now, standard practice of [9], parallel composition is broken up into three alternatives: the part where the first step comes from  $x$ , the part where the first step comes from  $y$  and the part where  $x$  and  $y$  execute together.

$$x \parallel y = x \ll y + y \ll x + x | y$$

To tackle the axiomatization of the left merge operator, the following axioms are used [8].

$$a.x \ll y = a.(x \parallel y) \quad (x + y) \ll z = x \ll z + y \ll z.$$

Finally, what remains is the behavior of parallel composition with respect to the termination constants  $\delta$  and  $\epsilon$ . As the termination behavior of parallel composition is coded into the communication merge operator, this is of no concern here, and the following laws can be put forward.

$$\delta \ll x = \delta \quad \epsilon \ll x = \delta$$

Next, the standard axioms of communication merge operator are introduced [9].

$$\begin{array}{ll} (x + y) \mid z = x \mid z + y \mid z & x \mid (y + z) = x \mid y + x \mid z \\ \delta \mid x = \delta & x \mid \delta = \delta \\ a.x \mid b.y = c.(x \parallel y) & \text{if } \gamma(a, b) = c \\ a.x \mid b.y = \delta & \text{if } \gamma(a, b) \text{ not defined.} \end{array}$$

What remains are the cases where  $\epsilon$  appears as an argument of the communication merge operator.

$$\begin{array}{ll} \epsilon \mid \epsilon = \epsilon & \\ a.x \mid \epsilon = \delta & \epsilon \mid a.x = \delta. \end{array}$$

The axiom system presented in Table 17 contains an axiom stipulating the commutativity of the communication merge. This allows to save on the number of axioms required for the communication merge operator.

**Theorem 30 (Elimination).** *For any closed TCP( $A, \gamma$ )-term  $p$ , there exists a closed TSP( $A$ )-term  $q$  such that  $\text{TCP}(A, \gamma) \vdash p = q$ .*

The transition system specification associated to the terms of TCP is given in Table 18. It adds deduction rules defining action transitions and termination behavior of encapsulation and the parallel composition operators.

**Theorem 31 (Soundness).** *Let  $p$  and  $q$  be two closed TCP( $A, \gamma$ )-terms. If  $\text{TCP}(A, \gamma) \vdash p = q$ , then  $TSS(\text{TCP}(A, \gamma)) \vDash p \Leftrightarrow q$ .*

**Theorem 32 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed TCP( $A, \gamma$ )-terms. If  $TSS(\text{TCP}(A, \gamma)) \vDash p \Leftrightarrow q$ , then  $\text{TCP}(A, \gamma) \vdash p = q$ .*

**Theorem 33 (Conservative extension).** *TCP is a conservative extension of TSP.*

*Proof.* Both TSP and TCP are sound and ground-complete equational theories for  $TSS(\text{TSP}(A))$  and  $TSS(\text{TCP}(A, \gamma))$ . Also, following [12],  $TSS(\text{TCP}(A, \gamma))$  is an operationally conservative (orthogonal) extension of  $TSS(\text{TSP}(A))$ . Thus, we conclude that TCP is an equationally conservative extension of TSP.

The relationship between the theories  $ACP^\epsilon$  and TCP is similar to the relationship between  $BPA_\delta^\epsilon$  and TSP. One can either extend TCP with the constants  $a \in A$  using axioms  $a = a.\epsilon$  or one can embed  $ACP^\epsilon$  into TCP by mapping  $a$  onto  $a.\epsilon$ .

TCP( $A, \gamma$ )			
TSP( $A$ );			
unary: $(\partial_H)_{H \subseteq A}$ ;    binary: $-  -, -\ll-, - -, -;$			
$x, y, z : P$ ;			
$\partial_H(\epsilon) = \epsilon$	D1		
$\partial_H(\delta) = \delta$	D2		
$\partial_H(a.x) = \delta$ if $a \in H$	D3		
$\partial_H(a.x) = a.\partial_H(x)$ otherwise	D4		
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D5		
$x  y = x\ll y + y\ll x + x y$	M	$x y = y x$	SC1
$\delta\ll x = \delta$	LM1		
$\epsilon\ll x = \delta$	LM2	$x  \epsilon = x$	SC2
$a.x\ll y = a.(x  y)$	LM3	$\epsilon x + \epsilon = \epsilon$	SC3
$(x + y)\ll z = x\ll z + y\ll z$	LM4		
$\delta x = \delta$	CM1	$(x  y)  z = x  (y  z)$	SC4
$(x + y) z = x z + y z$	CM2	$(x y) z = x (y z)$	SC5
$\epsilon \epsilon = \epsilon$	CM3	$(x\ll y)\ll z = x\ll(y  z)$	SC6
$a.x \epsilon = \delta$	CM4	$(x y)\ll z = x (y\ll z)$	SC7
$a.x b.y = c.(x  y)$ if $\gamma(a, b) = c$	CM5		
$a.x b.y = \delta$ if $\gamma(a, b)$ not defined	CM6	$x\ll\delta = x \cdot \delta$	SC8

**Table 17.** The process theory TCP( $A, \gamma$ ).

$TSS(\text{TCP}(A, \gamma))$			
$TSS(\text{TSP}(A))$ ;			
unary: $(\partial_H)_{H \subseteq A}$ ;    binary: $-  -, -\ll-, - -, -;$			
$x, y, x', y'$ ;			
$\frac{x \downarrow \quad y \downarrow}{x  y \downarrow}$	$\frac{x \downarrow \quad y \downarrow}{x y \downarrow}$	$\frac{x \downarrow}{\partial_H(x) \downarrow}$	$\frac{x \xrightarrow{a} x' \quad a \notin H}{\partial_H(x) \xrightarrow{a} \partial_H(x')}$
$\frac{x \xrightarrow{a} x'}{x  y \xrightarrow{a} x'    y}$	$\frac{y \xrightarrow{a} y'}{x  y \xrightarrow{a} x    y'}$	$\frac{x \xrightarrow{a} x'}{x\ll y \xrightarrow{a} x'    y}$	
$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y' \quad \gamma(a, b) = c}{x  y \xrightarrow{c} x'    y'}$	$\frac{x \xrightarrow{a} x' \quad y \xrightarrow{b} y' \quad \gamma(a, b) = c}{x y \xrightarrow{c} x'   y'}$		

**Table 18.** Transition system specification for TCP( $A, \gamma$ ).

## 6.2 TCP<sup>drt\*</sup>

In this section, the process theory TSP<sup>drt\*</sup> is extended to the process theory TCP<sup>drt\*</sup>. This extension is obtained by extending the signature of TSP<sup>drt\*</sup> with the current time slice timeout operator  $\nu$ , the encapsulation operators  $\partial_H$  (for  $H \subseteq A$ ), and the parallel composition operators  $-||-$ ,  $-||_-$ , and  $-|-$ .

The axioms of the process theory are given in Table 19.

The current time slice time out operator disallows all initial passage of time. It extracts the part of the behavior that executes an action or performs termination in the current time slice. The encapsulation operator is as defined before: encapsulation disallows the actions that occur in the set  $H$  and allows all other behavior including passage of time.

The axioms for parallel composition and the auxiliary operators are such that parallel processes have to synchronize the passage of time until one of the processes can terminate, and within each time slice interleave their actions or communicate. To stay as closely as possible to the interpretation of the axioms in the untimed setting, it is necessary for both left merge and communication merge to synchronize the passage of time as well (axioms LM6DR and CM10DR).

**Theorem 34 (Elimination).** *For any closed TCP<sup>drt\*</sup>(A,  $\gamma$ )-term  $p$ , there exists a closed TSP<sup>drt\*</sup>(A)-term  $q$  such that  $\text{TCP}^{\text{drt}^*}(A, \gamma) \vdash p = q$ .*

The transition system specification associated to the terms of TCP<sup>drt\*</sup> is given in Table 20. It adds deduction rules defining the behavior of the “now” operator and it adds deduction rules defining time transitions of encapsulation and the parallel composition operators.

**Theorem 35 (Soundness).** *Let  $p$  and  $q$  be two closed TCP<sup>drt\*</sup>(A,  $\gamma$ )-terms. If  $\text{TCP}^{\text{drt}^*}(A, \gamma) \vdash p = q$ , then  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \Leftrightarrow q$ .*

**Theorem 36 (Ground-completeness).** *Let  $p$  and  $q$  be arbitrary closed TCP<sup>drt\*</sup>(A,  $\gamma$ )-terms. If  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \Leftrightarrow q$ , then  $\text{TCP}^{\text{drt}^*}(A, \gamma) \vdash p = q$ .*

**Theorem 37 (Conservative ground-extension).** *TCP<sup>drt\*</sup> is a conservative ground-extension of TCP.*

*Proof.* Similar to the proof of Theorem 28. We show that for all  $p$  and  $p'$  in the syntax of TCP,

1.  $TSS(\text{TCP}(A, \gamma)) \models p \xrightarrow{a} p' \Leftrightarrow TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \xrightarrow{a} p'$ ,
2.  $TSS(\text{TCP}(A, \gamma)) \models p \downarrow \Leftrightarrow TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \downarrow$ ,
3. there exists a closed TCP(A)-term  $q$  such that  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \mapsto q$ ,  
and
4. for all closed TCP<sup>drt\*</sup>(A,  $\gamma$ )-terms  $q$  such that  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \mapsto q$   
then  $TSS(\text{TCP}(A, \gamma)) \models p \Leftrightarrow q$  and  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma)) \models p \Leftrightarrow q$ .

TCP <sup>drt*</sup> (A, $\gamma$ )			
TSP <sup>drt*</sup> (A);			
unary: $\nu, (\partial_H)_{H \subseteq A}$ ;    binary: $-  -, -\underline{\underline{}}-, -  -;$			
$x, y, z : P;$			
$\partial_H(\underline{\underline{\epsilon}}) = \underline{\underline{\epsilon}}$	D1DR	$\nu(\underline{\underline{\epsilon}}) = \underline{\underline{\epsilon}}$	RTO1
$\partial_H(\underline{\underline{\delta}}) = \underline{\underline{\delta}}$	D2DR	$\nu(\underline{\underline{\delta}}) = \underline{\underline{\delta}}$	RTO2
$\partial_H(\underline{\underline{a}}.x) = \underline{\underline{\delta}}$ if $a \in H$	D3DR	$\nu(\underline{\underline{a}}.x) = \underline{\underline{a}}.x$	RTO3
$\partial_H(\underline{\underline{a}}.x) = \underline{\underline{a}}.\partial_H(x)$ otherwise	D4DR		
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	D5	$\nu(x + y) = \nu(x) + \nu(y)$	RTO4
$\partial_H(\sigma.x) = \sigma.\partial_H(x)$	D6DR	$\nu(\sigma.x) = \underline{\underline{\delta}}$	RTO5
$\partial_H(\sigma^*x) = \sigma^*\partial_H(x)$	D7DR	$\nu(\sigma^*x) = \nu(x)$	RTO6
$x   y = y   x$	SC1	$(x   y)   z = x   (y   z)$	SC5
$x    \underline{\underline{\epsilon}} = x$	SC2DR	$(x    y)    z = x    (y    z)$	SC6
$\underline{\underline{\epsilon}}   x + \underline{\underline{\epsilon}} = \underline{\underline{\epsilon}}$	SC3DR	$(x   y)    z = x   (y    z)$	SC7
$(x    y)    z = x    (y    z)$	SC4	$x    \underline{\underline{\delta}} = x.\underline{\underline{\delta}}$	SC8DR
$x    y = x    y + y    x + x   y$	M		
$\underline{\underline{\delta}}    x = \underline{\underline{\delta}}$	LM1DR	$\underline{\underline{\delta}}   x = \underline{\underline{\delta}}$	CM1DR
$\underline{\underline{\epsilon}}    \underline{\underline{\delta}} = \underline{\underline{\delta}}$	LM2DR	$(x + y)   z = x   z + y   z$	CM2
$\underline{\underline{a}}.x    y = \underline{\underline{a}}.(x    y)$	LM3DR	$\underline{\underline{\epsilon}}   \underline{\underline{\epsilon}} = \underline{\underline{\epsilon}}$	CM3DR
$(x + y)    z = x    z + y    z$	LM4	$\underline{\underline{a}}.x   \underline{\underline{\epsilon}} = \underline{\underline{\delta}}$	CM4DR
$\sigma.x    \underline{\underline{\delta}} = \underline{\underline{\delta}}$	LM5DR	$\underline{\underline{a}}.x   \underline{\underline{b}}.y = \underline{\underline{c}}.(x    y)$ if $\gamma(a, b) = c$	CM5DR
$\sigma.x    \underline{\underline{\epsilon}} = \sigma.x$	LM6DR	$\underline{\underline{a}}.x   \underline{\underline{b}}.y = \underline{\underline{\delta}}$ if $\gamma(a, b)$ not defined	CM6DR
$\sigma.x    (\underline{\underline{a}}.y + z) = \sigma.x    z$	LM7DR	$\sigma.x   \nu(y) = \underline{\underline{\delta}}$	CM7DR
$\sigma.x    (\nu(y) + \sigma.z) = \sigma.(x    z)$	LM8DR	$\sigma.x   \sigma.y = \sigma.(x   y)$	CM8DR
$\sigma^*x    \sigma^*\nu(y) = \sigma^*(x    \sigma^*\nu(y))$	LM9DR	$\sigma^*x   \sigma^*y = \sigma^*(x   \sigma^*y + \sigma^*x   y)$	CM9DR

**Table 19.** The process theory TCP<sup>drt\*</sup>(A,  $\gamma$ ).

$\frac{\text{--- } TSS(\text{TCP}^{\text{drt}^*}(A, \gamma))}{TSS(\text{TCP}(A, \gamma)), TSS(\text{TSP}^{\text{drt}^*}(A))};$		
unary: $\nu(\_)$ ;		
$x, x', y, y'$ ;		
$\frac{x \downarrow}{\nu(x) \downarrow}$	$\frac{x \xrightarrow{a} x'}{\nu(x) \xrightarrow{a} x'}$	$\frac{x \mapsto x'}{\partial_H(x) \mapsto \partial_H(x')}$
$\frac{x \mapsto x' \quad y \mapsto y'}{x \parallel y \mapsto x' \parallel y'}$	$\frac{x \mapsto x' \quad y \mapsto y'}{x \parallel\!\!\! \parallel y \mapsto x' \parallel\!\!\! \parallel y'}$	$\frac{x \mapsto x' \quad y \mapsto y'}{x \mid y \mapsto x' \mid y'}$
$\frac{x \mapsto x' \quad y \not\mapsto y' \quad y \downarrow}{x \parallel y \mapsto x'}$	$\frac{x \not\mapsto y \mapsto y' \quad x \downarrow}{x \parallel y \mapsto y'}$	$\frac{x \mapsto x' \quad y \not\mapsto y' \quad y \downarrow}{x \parallel\!\!\! \parallel y \mapsto x'}$

**Table 20.** Transition system specification for  $\text{TCP}^{\text{drt}^*}(A, \gamma)$ .

From these, it follows that  $TSS(\text{TCP}^{\text{drt}^*}(A, \gamma))$  is an orthogonal extension of  $TSS(\text{TCP}(A, \gamma))$ .

The first and second items in the above list hold trivially since a proof in  $\text{TCP}^{\text{drt}^*}$  for an  $a$ -transition  $\xrightarrow{a}$  with a source term from TCP or a termination predicate  $\downarrow$  on a TCP-term only involves deduction rules from TCP. Such deduction rules do not have negative premises and are source-dependent [18]. Hence, all such transition and predicate formulae are included in the stable model of TCP. For the inclusion in the other direction, all proofs in TCP remain valid in  $\text{TCP}^{\text{drt}^*}$  and since deduction rules of TCP do not have negative premises, all the proven transitions and predicates of TCP are included in the stable model of  $\text{TCP}^{\text{drt}^*}$ .

For the last two items, we use a structural induction on closed TCP-term  $p$ . For function symbols in the signature of TSP, the arguments given in Theorem 28 remain valid. Hence, it only remains to check that for terms of the form  $\partial_H(p')$ ,  $p' \parallel q'$ ,  $p' \parallel\!\!\! \parallel q'$  and  $p' \mid q'$  the last two items hold, assuming that these items hold for  $p'$  and  $q'$  (if applicable).

If  $p$  is of the form  $\partial_H(p')$ , then it can make a time transition due to the following rule:

$$\frac{x \mapsto x'}{\partial_H(x) \mapsto \partial_H(x')}$$

Using the above deduction rule, one can only prove self time transitions and bisimilarity (w.r.t. both TSS's) is reflexive. Furthermore, the above rule is the only rule using which  $\partial_H(p')$ , hence  $p$ , can make a time transition.

If  $p$  is of the form  $p' \parallel q'$ , it can make a time transition due to the following rule:

$$\frac{x \mapsto x' \quad y \mapsto y'}{x \parallel y \mapsto x' \parallel y'}$$

and given that the time transitions of  $p'$  and  $q'$  are only to bisimilar terms (w.r.t. both TSS's) and bisimilarity is a congruence (since both  $TSS(\text{TCP}(A, \gamma))$  and  $TSS(\text{TCP}^{\text{drt}*}(A, \gamma))$  are in the PANTH format of [21]), the transition of  $p' \parallel q'$  or  $p$  is also to a bisimilar term (w.r.t. both TSS's). Furthermore,  $p$  cannot make a time transition due to any of the other two rules for time transitions of merge since both of them have negative premises denying time transitions from  $p'$  or  $q'$ .

If  $p$  is of the form  $p' \ll q'$ , it can make a time transition due to the following rule:

$$\frac{x \mapsto x' \quad y \mapsto y'}{x \ll y \mapsto x' \ll y'}$$

and given that the time transitions of  $p'$  and  $q'$  are only to bisimilar terms (w.r.t. both TSS's) and bisimilarity is a congruence, the transition of  $p' \ll q'$  or  $p$  is also to a bisimilar term (w.r.t. both TSS's). Furthermore,  $p$  cannot make a time transition due to the other rule for time transitions of left merge since it has a negative premise denying time transitions from  $q'$ .

If  $p$  is of the form  $p' \mid q'$ , it can make a time transition due to the following rule:

$$\frac{x \mapsto x' \quad y \mapsto y'}{x \mid y \mapsto x' \mid y'}$$

and given that the time transitions of  $p'$  and  $q'$  are only to bisimilar terms (w.r.t. both TSS's) and bisimilarity is a congruence, the transition of  $p' \mid q'$  or  $p$  is also to a bisimilar term (w.r.t. both TSS's). There is no other deduction rule using which  $p$  can make a time transition and this concludes the proof of the last two items.

**Theorem 38 (Conservative extension).**  $\text{TCP}^{\text{drt}*}$  is a conservative extension of  $\text{TSP}^{\text{drt}*}$ .

*Proof.* Both  $\text{TSP}^{\text{drt}*}$  and  $\text{TCP}^{\text{drt}*}$  are sound and ground-complete equational theories for  $TSS(\text{TSP}^{\text{drt}*}(A))$  and  $TSS(\text{TCP}^{\text{drt}*}(A, \gamma))$ . Also, following [12],  $TSS(\text{TCP}^{\text{drt}*}(A, \gamma))$  is an operationally conservative (orthogonal) extension of  $TSS(\text{TSP}^{\text{drt}*}(A))$ . Thus, we conclude that  $\text{TCP}^{\text{drt}*}$  is an equationally conservative extension of  $\text{TSP}^{\text{drt}*}$ .

## 7 Concluding remarks

When we extend untimed process algebra with timing, this extension is usually not a conservative extension, as some axioms of untimed process algebra hold

for untimed processes only. However, what can be achieved is a conservative ground-extension, a notion that is introduced here. In this paper, we present timed extensions of an incremental presentation of process algebras, involving termination constants, alternative composition, sequential composition and parallel composition with communication, where in each case it is shown we have a conservative ground-extension. In previous papers, conservativity was always violated in some way.

For the realization of these timed extensions, it was necessary to change a basic design principle of ACP-style process algebra: action constants, involving both action execution and termination, are replaced by action prefixing.

## References

1. J. C. M. Baeten. Embedding untimed into timed process algebra: the case for explicit termination. *Mathematical Structures in Computer Science (MSCS)*, 13(4):589–618, 2003.
2. J. C. M. Baeten and J. A. Bergstra. Real Time Process Algebra. *Formal Aspects of Computing*, 3:142–188, 1991.
3. J. C. M. Baeten and J. A. Bergstra. Discrete time process algebra. *Formal Aspects of Computing*, 8(2):188–208, 1996.
4. J. C. M. Baeten and R. J. van Glabbeek. Merge and termination in process algebra. In K. V. Nori, editor, *Proceeding of the Seventh Conference on Foundations of Software Technology and Theoretical Computer Science (FST&TCS'87)*, volume 287 of *Lecture Notes in Computer Science*, pages 153–172. Springer-Verlag, Berlin, Germany, 1987.
5. J. C. M. Baeten and C. A. Middelburg. *Process Algebra with Timing*. EATCS Monographs. Springer-Verlag, Berlin, Germany, 2002.
6. J. C. M. Baeten and M. A. Reniers. Timed process algebra (with a focus on explicit termination and relative-timing). In M. A. Bernardo and F. Corradini, editors, *Proceedings of the International School on Formal Methods for the Design of Real-Time Systems (SFM-RT'04)*, volume 3185 of *Lecture Notes in Computer Science*, pages 59–97. Springer-Verlag, Berlin, Germany, 2004.
7. J. C. M. Baeten and W. P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1990.
8. J. A. Bergstra and J. W. Klop. Fixed point semantics in process algebra. Technical Report IW 206/82, Mathematical Center, Amsterdam, The Netherlands, 1982.
9. J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1-3):109–137, 1984.
10. J. A. Bergstra and J. W. Klop. Algebra of communicating processes. In J. W. de Bakker, M. Hazewinkel, and J. K. Lenstra, editors, *Proceedings of the CWI Symposium Mathematics and Computer Science*, pages 89–138. North-Holland, Amsterdam, The Netherlands, 1986.
11. R. Bol and J. F. Groote. The meaning of negative premises in transition system specifications. *Journal of the ACM (JACM)*, 43(5):863–914, Sept. 1996.
12. W. J. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation (I&C)*, 146(1):24–54, 1998.
13. R. J. van Glabbeek. The meaning of negative premises in transition system specifications II. *Journal of Logic and Algebraic Programming (JLAP)*, 60-61:229–258, 2004.

14. J. F. Groote. Transition system specifications with negative premises. *Theoretical Computer Science (TCS)*, 118(2):263–299, 1993.
15. M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117(2):221–239, 1995.
16. C. P. J. Koymans and J. L. M. Vrancken. Extending process algebra with the empty process. Technical Report 1, Logic Group Preprint Series, Department of Philosophy, Utrecht University, Utrecht, The Netherlands, 1985. Extended and enhanced version appeared as [22].
17. F. Moller and C. M. N. Tofts. A temporal calculus of communicating systems. volume 458 of *Lecture Notes in Computer Science*, pages 401–415.
18. M. Mousavi and M. A. Reniers. Orthogonal extensions in structural operational semantics. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1214–1225. Springer-Verlag, Berlin, Germany, 2005.
19. X. Nicollin and J. Sifakis. The algebra of timed processes ATP: theory and application. *Information and Computation (I&C)*, 114(1):131–178, Oct. 1994.
20. J. J. Vereijken. *Discrete Time Process Algebra*. PhD thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands, 1997.
21. C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. *Nordic Journal of Computing*, 2(2):274–302, 1995.
22. J. L. M. Vrancken. The algebra of communicating processes with empty process. *Theoretical Computer Science*, 177(2):287–328, 1997.