UNIVERSITÄT AUGSBURG

Comparing the Efficiency of Asynchronous Systems

Lars Jenner   Walter Vogler

Report 1998-3   Dezember 1998

INSTITUT FÜR INFORMATIK
D-86135 AUGSBURG

Comparing the Efficiency of Asynchronous Systems

Lars Jenner*, Walter Vogler
Institut für Informatik, Universität Augsburg, D-86135 Augsburg, Germany
{jenner, vogler}@informatik.uni-augsburg.de

Abstract

A timed process algebra is developed for evaluating the temporal worst-case efficiency of asynchronous concurrent systems. For the sake of simplicity, we use a classical CCS-like algebra where actions may occur arbitrarily within a continuous time interval, yielding arbitrary relative speeds of the components. Via the timed testing approach, asynchronous systems are then related w.r.t. their worst-case efficiency, yielding an efficiency preorder. We show that this preorder can just as well be based on much simpler discrete time and that it can be characterized with some kind of refusal traces. Finally, precongruence results are provided for all operators of the algebra, where prefix, choice and recursion require special attention.

1 Motivation and Introduction

Classical process algebras like CCS model asynchronous systems, where the components have arbitrary relative speeds. To consider the temporal behaviour, several timed process algebras have been proposed, where usually systems are regarded as synchronous, i.e. have components with fixed speeds. The easiest of these is SCCS [Mil89], since terms are essentially the same as for CCS; the natural choice to fix the speeds of components is to assume that each action takes one unit of time; so SCCS-semantics differs from CCS-semantics essentially by excluding runs where one component performs many actions while another performs just one.

Our aim is to evaluate the temporal worst-case efficiency of asynchronous concurrent systems modeled with a process algebra, and – as in the case of SCCS – we want to keep things simple by using just classical CCS-like process terms. Furthermore, we will use a variant of (must-) testing [DNH84], where the testing preorder can be interpreted as comparing efficiency.

A usual treatment of asynchronous systems with a timed process algebra is to allow arbitrary idling before each action [Mil89, MT91]; this achieves arbitrary relative speeds, but is not suitable for defining worst-case runs since each action already can take arbitrarily long. Here, we assume each action to be performed within a given time – and to keep things simple as in SCCS, we take 1 as a common upper time bound for all actions. This enforces some progress, but different from SCCS, actions may also be performed faster than necessary; hence, components have arbitrary relative speeds and we take into account all runs of an asynchronous system. E.g. [Lyn96] uses upper time bounds for asynchronous systems in the area of distributed algorithms.

*This work was supported by the DFG-project 'Halbordnungstesten'.

We compare processes via the testing approach developed by [DNH84] and extended to timed testing in a Petri net framework in [Vog95, JV95], where a timed test is an environment together with a time bound. A process is embedded into the environment essentially via parallel composition and satisfies a timed test, if success is reached before the time bound in *every* run of the composed system, i.e. even in the worst case. If some process P satisfies each timed test satisfied by a process Q, then P may be successful in more environments than specified by Q, but it may also be successful in the same environments within a shorter time; therefore, we call it a *faster implementation* of Q, and the testing preorder is naturally an efficiency preorder.

To define this testing formally, we have to define runs of asynchronous systems. In Section 2, we develop a suitable semantics with upper time bound on actions where time is continuous; we try to formalize our intuitive ideas as directly as possible without anticipating any specific treatment that might be necessary to obtain a precongruence in the end. As regards the definition of testing, the classical embedding in the test environment leads to a testing preorder which – surprisingly – is not a precongruence for prefixing; instead of refining the preorder to the coarsest such precongruence (cf. [Jen96]), we get this precongruence directly by using a slightly different, but also intuitive embedding.

Using continuous time is certainly not as simple as intended; e.g. initially each process can make uncountably many different time steps. Our first main result in Section 3 shows that realism and simplicity can be reconciled: we define an analogous efficiency preorder based on discrete time behaviour and show its coincidence with the first one. In Section 4, as usual in a testing approach, we characterize the efficiency preorder – here with some kind of refusal traces. The important point with this second main result is that test environments are asynchronous systems, hence 'temporally weak', but nevertheless reveal the temporal behaviour of tested processes quite in detail; correspondingly, the construction of revealing tests is a little involved.

We also provide precongruence results for parallel composition, hiding, relabeling and prefixing. Finally, in Section 5 we refine the efficiency preorder to a precongruence also for choice: as usual, we additionally have to take into account the (initial) stability of processes. Quite surprisingly, although we consider a preorder, the additional condition on stability is not only an implication but an equivalence. The refined efficiency preorder is then shown to be the coarsest precongruence for all operators of our process algebra that respects inclusion of discrete behaviour. We also provide a precongruence result for recursion. Here, we avoid the introduction of least elements (Ω-terms) and application of cpo-techniques (cf. [Hen88]) and thereby gain some degree of self-containment, but our technique exploits the restriction to guarded recursion.

We have translated the results for Petri nets from [JV95] to a process algebra setting for two reasons: on the one hand, it is shown that the underlying ideas are not model-dependent; on the other hand, the developments here are quite different, in particular since process algebras are much more powerful than finite safe Petri nets; see e.g. the progress preorder in Section 3. For an interesting application of our approach see [JV95], where different implementations of a bounded buffer are distinguished w.r.t. their efficiency; we intend to carry over this example into our process algebra setting, expecting the same results.

2 Continuously timed Processes and Tests

We will use a *CCS*-like process algebra with *TCSP*-like $\|_A$ parallel composition, where A is the set of actions components have to synchronize on. Processes will perform (atomic) actions instantaneously within time 1; time passes continuously (in this section) in between their occurrences. For example, process $a.P$ will idle and then perform action a at some time point in the real interval $[0;1]$, evolving to P. To model this, we introduce continuously timed actions $\langle a,r\rangle$, which carry a 'timer' r whose initial value can be chosen from the interval $[0;1]$ of real numbers. Whenever time passes globally by a certain amount, the timer of a locally activated action will be decreased accordingly. Timer value 0 denotes that the idle time of the respective action has elapsed, hence it must either occur or be deactivated before time may pass further – unless it has to wait for synchronization with another component (i.e. our processes are *patient*). E.g., process $a.P$ corresponds to $\langle a,1\rangle.P$ and can idle, process $\langle a,0\rangle.Q$ can neither idle nor wait, but component $\langle a,0\rangle.Q$ in $(\langle a,0\rangle.Q)\|_{\{a\}}(\langle a,1\rangle.P)$ has to wait for synchronization on a while component $\langle a,1\rangle.P$ still may idle.

We also use two distinguished actions: τ represents internal activity that is unobservable for other components; ω is reserved for observers (test processes) only, which use this action in order to signal success of a test.

Definition 2.1 *timed actions*

Let $\mathbb{A}$ be a possibly infinite set of actions, let ω be a special action – the *success action* –, and let τ be the *internal action.* We define $\mathbb{A}_\omega = \mathbb{A}\cup\{\omega\}$ and $\mathbb{A}_{\omega\tau} = \mathbb{A}_\omega\cup\{\tau\}$. Elements of $\mathbb{A}_{\omega\tau}$ are denoted by $a,b,c,\dots$ (including τ and ω).

Let $\mathbb{T} = [0;1] \subseteq \mathbb{R}_0^+$ be the set of real numbers in the interval $[0;1]$. Elements from $\mathbb{T}$ are denoted by $\rho, r, \dots$

Let $Act = \mathbb{A}_{\omega\tau} \times \mathbb{T} = \{\alpha,\beta,\dots\}$ be the set of *continuously timed actions*, where e.g. $\alpha = \langle a,r\rangle \in Act$. We use a as a shorthand for $\langle a,1\rangle$ and $\underline{a}$ as a shorthand for $\langle a,0\rangle$, which we call an *urgent* action. ■ 2.1

Definition 2.2 *continuously timed and initial process terms and processes*

Let $\Phi : \mathbb{A}_{\omega\tau} \to \mathbb{A}_{\omega\tau}$ be a function such that the set $\{a\in\mathbb{A} \mid \emptyset \neq \Phi^{-1}(a) \neq \{a\}\}$ is finite, $\Phi^{-1}(\omega)\subseteq\{\omega\}$ and $\Phi(\tau)=\tau$; then Φ is a *general relabelling function.*

A *(continuously timed) c-process term* P is generated by the following grammar:

$$P \;::=\; \mathbf{0} \;\big|\; X \;\big|\; \langle a,r\rangle.P \;\big|\; P+P \;\big|\; P\|_A P \;\big|\; P[\Phi] \;\big|\; \mu X.P$$

where $\mathbf{0}$ *(Nil)* is a constant, $X\in\mathcal{X}=\{X,Y,Z,\dots\}$ is a *(process) variable*, $\langle a,r\rangle\in Act$, Φ is a general relabelling function and $A\subseteq\mathbb{A}$ possibly infinite. Additionally, we only allow *guarded* recursion, where *also internal* timed actions $\langle\tau,r\rangle$ may serve as a guard. The set of c-process terms is denoted by $\tilde{\mathbb{P}}_c$.

We distinguish several cases: P is an *initial* process term, if the choice of r is restricted to $r=1$; the set of initial process terms, is denoted by $\tilde{\mathbb{P}}_1$. P is a *(continuously timed) c-process*, if P is closed, i.e. all variables X in P are bound by the according μX-operator; the set of c-processes is denoted by $\mathbb{P}_c$. $\mathbb{P}_1 = \mathbb{P}_c\cap\tilde{\mathbb{P}}_1$ is the set of *initial processes.* ■ 2.2

0 is the Nil-process, which cannot perform any action, but may let pass time without limit. $X \in \mathcal{X}$ is a process variable used for recursion. $\alpha.P$ is (action-) prefixing, known from *CCS*, where $\langle a, r\rangle.P$ is ready to perform action a at some time in $[0, r]$. $P_1 + P_2$ models the choice (sum) of two conflicting processes P_1 and P_2. $P_1\|_A P_2$ is the parallel composition of two processes P_1 and P_2 that run in parallel and have to synchronize on all actions from A; this synchronization discipline is inspired from TCSP.

The general relabelling operation $P[\Phi]$ subsumes the classically distinguished operations relabelling and hiding. These will be understood as special cases of a general relabelling in the following way: if Φ satisfies the condition $\Phi^{-1}(\tau) = \{\tau\}$, then Φ is a *(classical) relabelling function*; if for a set $A \subseteq \mathbb{A}$ Φ satisfies the conditions $\Phi|_A = \tau$ and $\Phi|_{\mathbb{A}_{\omega\tau}\setminus A} = id_{\mathbb{A}_{\omega\tau}}$, then we consider P/A to be a notation equivalent to $P[\Phi]$, where A is called a *hiding set*. The restrictions on general relabelling functions serve several purposes: $\Phi(\tau) = \tau$ ensures that τ cannot be made visible by relabelling, and $\Phi^{-1}(\omega) \subseteq \{\omega\}$ ensures that testable processes will be closed under general relabelling. The finiteness of the set $\{a \in \mathbb{A} \mid \emptyset \neq \Phi^{-1}(a) \neq \{a\}\}$ will ensure later on that the number of different actions ever performable by a given c-process is finite; note that we allow infinite hiding sets, however.

$\mu X.P$ models recursion. Some $X \in \mathcal{X}$ is *guarded* in a c-process term $P \in \tilde{\mathbb{P}}_c$, if each occurrence of X is in a subterm $\alpha.Q$ of P where $\alpha \in Act$ (guard); note that *also internal* timed actions $\langle\tau, r\rangle$ may serve as a guard. In this paper, we only consider c-process terms $\mu X.P$ where X is guarded in P. We say that $P \in \tilde{\mathbb{P}}_c$ is *guarded* if all $X \in \mathcal{X}$ are guarded in P. Note that c-processes are guarded.

Obviously, initial c-process terms coincide essentially with ordinary CCS-like terms, where $\langle a, 1\rangle$ represents simple a.

In order to economize on parentheses, we determine the precedence of the operators in decreasing order as follows: relabelling, prefix, recursion, parallel composition, choice.

Whenever we perform syntactical substitution $P\{Q/X\}$, we assume *free*$(Q) \cap$ *bound*$(P) = \emptyset$ (Barendregt convention), where *free*(P) and *bound*(P) denote the sets of free resp. bound variables in P. If $\mathcal{S}$ is a function $\mathcal{S} : \mathcal{X} \mapsto \tilde{\mathbb{P}}_c$, then $\mathcal{S}$ denotes a *simultaneous substitution* of all variables, and we write $[P]_\mathcal{S}$ for $P\{\mathcal{S}(X)/X, \mathcal{S}(Y)/Y, \ldots\}$.

We intend choice and parallel composition to be commutative and choice to be associative, and we anticipate this by a syntactical congruence:

Definition 2.3 *syntactical congruence*

Syntactical congruence $\equiv\ \subseteq \tilde{\mathbb{P}}_c \times \tilde{\mathbb{P}}_c$ is the least congruence of c-process terms satisfying for all $P_1, P_2, P_3 \in \tilde{\mathbb{P}}_c$ and $A \subseteq \mathbb{A}$:

1. $P_1 + P_2 \equiv P_2 + P_1$
2. $(P_1 + P_2) + P_3 \equiv P_1 + (P_2 + P_3)$
3. $P_1\|_A P_2 \equiv P_2\|_A P_1$

We regard syntactically congruent c-process terms as equal. Therefore, $\sum_{i\in I} P_i$ is used as a shorthand for the sum of all $P_i \in \tilde{\mathbb{P}}_c$, where i is in a finite indexing set I. We define $\sum_{i\in\emptyset} P_i \equiv \mathbf{0}$, and if $|I| = 1$, then $\sum_{j\in\{i\}} P_j \equiv P_i$. ■ 2.3

Now the purely functional behaviour of process terms (i.e. which actions they can perform) is given by the following operational semantics, where syntactical congruence enables us to use only one SOS-rule for choice and two SOS-rules for parallel composition:

Definition 2.4 *Operational semantics of functional behaviour*

Via the following SOS-rules, a ternary relation $\rightarrow \subseteq (\tilde{\mathbb{P}}_c \times \mathbb{A}_{\omega\tau} \times \tilde{\mathbb{P}}_c)$ is defined inductively:

$$\text{Pref}_a \ \frac{}{\langle a, r\rangle.P \xrightarrow{a} P} \qquad \text{Par}_{a1} \ \frac{a \notin A,\ P_1 \xrightarrow{a} P_1'}{P_1\|_A P_2 \xrightarrow{a} P_1'\|_A P_2} \qquad \text{Par}_{a2} \ \frac{a \in A,\ P_1 \xrightarrow{a} P_1',\ P_2 \xrightarrow{a} P_2'}{P_1\|_A P_2 \xrightarrow{a} P_1'\|_A P_2'}$$

$$\text{Sum}_a \ \frac{P_1 \xrightarrow{a} P_1'}{P_1 + P_2 \xrightarrow{a} P_1'} \qquad \text{Rel}_a \ \frac{P \xrightarrow{a} P'}{P[\Phi] \xrightarrow{\Phi(a)} P'[\Phi]} \qquad \text{Rec}_a \ \frac{P \xrightarrow{a} P'}{\mu X.P \xrightarrow{a} P'\{\mu X.P/X\}}$$

For c-process terms $P, P' \in \tilde{\mathbb{P}}_c$ and $a \in \mathbb{A}_{\omega\tau}$, we write $P \xrightarrow{a} P'$ if $(P, a, P') \in \rightarrow$ and $P \xrightarrow{a}$ if there exists a $P'' \in \tilde{\mathbb{P}}_c$ such that $(P, a, P'') \in \rightarrow$.

Finally, we let $\mathcal{A}(P) = \{a \in \mathbb{A}_{\omega\tau} \mid P \xrightarrow{a}\}$ be the set of *activated actions* of P. ■ 2.4

Except for Pref_a and Rec_a, these rules are standard. Pref_a allows an activated action to occur disregarding the value of its timer. Additionally, passage of time will never deactivate actions or activate new ones, and we capture all behaviour that is possible in the standard CCS-like setting without time. Note that rule Rec_a implicitly makes use of guarded recursion [BD91]. It forces us to define an operational semantics not only for c-processes but also for c-process terms (in the premise of Rec_a). On the other hand, it will simplify proofs of operational properties, since it connects induction on inferences with induction on the structure of a c-process.

The set of activated actions of a c-process term P describes its immediate functional behaviour; note that $\mathcal{A}(X)$ is empty for process variables $X \in \mathcal{X}$, reflecting that unbound occurrence of a variable means incomplete implementation and that $\mathcal{A}(P)$ records only actions, not the possibly various timer values associated with the same action in a process.

We have defined the set of activated actions via operational semantics, but $\mathcal{A}(P)$ can equivalently be determined inductively from the syntactical structure of P alone.

The set of activated actions will we preserved both along passage of time and under substitution of guarded variables; furthermore, due to the image-finiteness of general relabelling functions, $\mathcal{A}(P)$ is always finite, and this will be used for the characterization of our testing preorder.

Proposition 2.5

Let $P, Q, R \in \tilde{\mathbb{P}}_c$ be c-process terms, $a \in \mathbb{A}_{\omega\tau}$ and let $X \in \mathcal{X}$ be guarded in P.

1. $\mathcal{A}(P)$ is finite.
2. $P\{Q/X\} \xrightarrow{a} R$ if and only if there exists $P' \in \tilde{\mathbb{P}}_c$ with $P \xrightarrow{a} P'$ and $R \equiv P'\{Q/X\}$; in particular $\mathcal{A}(P) = \mathcal{A}(P\{Q/X\})$.
3. $\mathcal{A}(P)$ can be calculated by induction on the structure of P:

1. Nil: $\mathcal{A}(\mathbf{0}) = \emptyset$
2. Var: $\mathcal{A}(X) = \emptyset$
3. Pref: $\mathcal{A}(\langle a, r\rangle.P) = \{a\}$ for all $a \in \mathbb{A}_{\omega\tau}$
4. Sum: $\mathcal{A}(P_1 + P_2) = \mathcal{A}(P_1) \cup \mathcal{A}(P_2)$
5. Par: $\mathcal{A}(P_1 \|_A P_2) = (\mathcal{A}(P_1) \cap \mathcal{A}(P_2) \cap A) \cup ((\mathcal{A}(P_1) \cup \mathcal{A}(P_2)) \setminus A)$
6. Rel: $\mathcal{A}(P[\Phi]) = \Phi(\mathcal{A}(P))$
7. Rec: $\mathcal{A}(\mu X.P) = \mathcal{A}(P)$

Proof:

1. Induction on the structure of P.

2. Induction on the structure of P:

Nil: $\mathbf{0} \equiv \mathbf{0}\{Q/X\} \xrightarrow{a}$ for no $a \in \mathbb{A}_{\omega\tau}$.

Var: X guarded in $P \equiv Y$ implies $X \not\equiv Y$; this case is analogous to Nil.

Pref: X is guarded in $\langle a, r\rangle.P$ and $(\langle a, r\rangle.P)\{Q/X\} \equiv \langle a, r\rangle.(P\{Q/X\}) \xrightarrow{a} P\{Q/X\}$, which is unique, and $\langle a, r\rangle.P \xrightarrow{a} P$, which is unique.

Sum: X guarded in $P_1 + P_2 \Leftrightarrow X$ guarded in P_1 and P_2.
$(P_1 + P_2)\{Q/X\} \equiv P_1\{Q/X\} + P_2\{Q/X\} \xrightarrow{a} R \Leftrightarrow P_1\{Q/X\} \xrightarrow{a} R \vee P_2\{Q/X\} \xrightarrow{a} R \Leftrightarrow \exists P' : (P_1 \xrightarrow{a} P' \vee P_2 \xrightarrow{a} P') \wedge R \equiv P'\{Q/X\} \Leftrightarrow \exists P' : P_1 + P_2 \xrightarrow{a} P' \wedge R \equiv P'\{Q/X\}$.

Par: X guarded in $P_1 \|_A P_2 \Leftrightarrow X$ guarded in P_1 and P_2.
Let $a \in A$;
then $(P_1 \|_A P_2)\{Q/X\} \equiv P_1\{Q/X\} \|_A P_2\{Q/X\} \xrightarrow{a} R_1 \|_A R_2$
$\Leftrightarrow P_1\{Q/X\} \xrightarrow{a} R_1 \wedge P_2\{Q/X\} \xrightarrow{a} R_2$
$\Leftrightarrow \exists P_1', P_2' : P_1 \xrightarrow{a} P_1' \wedge P_2 \xrightarrow{a} P_2' \wedge R_1 \equiv P_1'\{Q/X\} \wedge R_2 \equiv P_2'\{Q/X\}$
$\Leftrightarrow \exists P_1', P_2' : P_1 \|_A P_2 \xrightarrow{a} P_1' \|_A P_2' \wedge$
$R_1 \|_A R_2 \equiv P_1'\{Q/X\} \|_A P_2'\{Q/X\} \equiv (P_1' \|_A P_2')\{Q/X\}$.
Let $a \notin A$;
then w.l.o.g. $(P_1 \|_A P_2)\{Q/X\} \equiv P_1\{Q/X\} \|_A P_2\{Q/X\} \xrightarrow{a} R_1 \|_A P_2\{Q/X\}$
$\Leftrightarrow P_1\{Q/X\} \xrightarrow{a} R_1 \Leftrightarrow \exists P_1' : P_1 \xrightarrow{a} P_1' \wedge R_1 \equiv P_1'\{Q/X\}$
$\Leftrightarrow \exists P_1' : P_1 \|_A P_2 \xrightarrow{a} P_1' \|_A P_2 \wedge$
$R_1 \|_A P_2\{Q/X\} \equiv P_1'\{Q/X\} \|_A P_2\{Q/X\} \equiv (P_1' \|_A P_2)\{Q/X\}$.

Rel: straightforward induction.

Rec: Let $X \equiv Y$ and $\mu Y.P \xrightarrow{a} P'\{\mu Y.P/Y\}$ due to $P \xrightarrow{a} P'$ by rule Rec_a; then X is bound in $\mu Y.P$ and $P'\{\mu Y.P/Y\}$, hence $(\mu Y.P)\{Q/X\} \equiv \mu Y.P \xrightarrow{a} P'\{\mu Y.P/Y\} \equiv (P'\{\mu Y.P/Y\})\{Q/X\}$.
Let $X \not\equiv Y$; then $(\mu Y.P)\{Q/X\} \equiv \mu Y.(P\{Q/X\})$ since by Barendregt convention Y is not free in Q. By rule Rec_a, $(\mu Y.P)\{Q/X\} \equiv \mu Y.(P\{Q/X\}) \xrightarrow{a} R$ iff $R \equiv R'\{\mu Y.(P\{Q/X\})/Y\}$ and $P\{Q/X\} \xrightarrow{a} R'$ for some R' iff (by ind.) $R \equiv R'\{\mu Y.(P\{Q/X\})/Y\}$ and $R' \equiv P'\{Q/X\}$ with $P \xrightarrow{a} P'$ for some R' and P' iff $R \equiv (P'\{Q/X\})\{\mu Y.(P\{Q/X\})/Y\} \equiv (P'\{\mu Y.P/Y\})\{Q/X\}$ and $P \xrightarrow{a} P'$ for some P' iff $R \equiv P''\{Q/X\}$ and $P'' \equiv P'\{\mu Y.P/Y\}$ and $P \xrightarrow{a} P'$ for some P' and P'' iff $\mu Y.P \xrightarrow{a} P''$ and $R \equiv P''\{Q/X\}$ for some P''.

3. Induction on the inference of $P \xrightarrow{a}$:

Nil: $\mathbf{0} \xrightarrow{a}$ for no $a \in \mathbb{A}_{\omega\tau}$, hence $\mathcal{A}(\mathbf{0}) = \emptyset$.

Var: analogously to Nil.

Pref: $\alpha.P \xrightarrow{a}$ iff $\alpha = \langle a, r\rangle$, hence $\mathcal{A}(\alpha.P) = \{a\}$
Sum: $P_1 + P_2 \xrightarrow{a} \Leftrightarrow P_1 \xrightarrow{a} \vee P_2 \xrightarrow{a} \Leftrightarrow a \in \mathcal{A}(P_1) \vee a \in \mathcal{A}(P_2)$, hence $\mathcal{A}(P_1 + P_2) = \mathcal{A}(P_1) \cup \mathcal{A}(P_2)$.
Par$_1$: $a \notin A \wedge P_1\|_A P_2 \xrightarrow{a} \Leftrightarrow a \notin A \wedge (P_1 \xrightarrow{a} \vee P_2 \xrightarrow{a}) \Leftrightarrow a \notin A \wedge (a \in \mathcal{A}(P_1) \vee a \in \mathcal{A}(P_2))$, hence iff $a \in \mathcal{A}(P_1) \cup \mathcal{A}(P_2)$
Par$_2$: $a \in A \wedge P_1\|_A P_2 \xrightarrow{a} \Leftrightarrow a \in A \wedge (P_1 \xrightarrow{a} \wedge P_2 \xrightarrow{a}) \Leftrightarrow a \in A \wedge (a \in \mathcal{A}(P_1) \wedge a \in \mathcal{A}(P_2))$, hence iff $a \in \mathcal{A}(P_1) \cap \mathcal{A}(P_2)$
Rel: $P[\Phi] \xrightarrow{a} \Leftrightarrow \exists b \in \Phi^{-1}(a) : P \xrightarrow{b} \Leftrightarrow \exists b \in \mathcal{A}(P) : \Phi(b) = a \Leftrightarrow a \in \Phi(\mathcal{A}(P))$
Rec: $\mu X.P \xrightarrow{a} \Leftrightarrow P \xrightarrow{a} \Leftrightarrow a \in \mathcal{A}(P)$. ■ 2.5

As a first step to define timed behaviour, we now give operational rules for the passage of 'wait-time': all components of a system participate in a global time step, and this passage of time is recorded for locally activated actions by decreasing their annotated timer in rule Pref$_c$. Note that time passes disregarding elapsed timers; this might be necessary for a component when waiting for a synchronization partner, and this explains the notion 'wait-time'.

Definition 2.6 *operational semantics for wait-time*

Via the following SOS-rules, a ternary relation $\rightsquigarrow_c \subseteq (\tilde{\mathbb{P}}_c \times \mathbb{T} \times \tilde{\mathbb{P}}_c)$ is defined inductively:

$$\text{Nil}_c \ \frac{}{\mathbf{0} \overset{\rho}{\rightsquigarrow}_c \mathbf{0}} \qquad \text{Pref}_c \ \frac{r' = \max(r - \rho, 0)}{\langle a, r\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, r'\rangle.P} \qquad \text{Sum}_c \ \frac{P_1 \overset{\rho}{\rightsquigarrow}_c P_1',\ P_2 \overset{\rho}{\rightsquigarrow}_c P_2'}{P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c P_1' + P_2'}$$

$$\text{Rel}_c \ \frac{P \overset{\rho}{\rightsquigarrow}_c P'}{P[\Phi] \overset{\rho}{\rightsquigarrow}_c P'[\Phi]} \qquad \text{Par}_c \ \frac{P_1 \overset{\rho}{\rightsquigarrow}_c P_1',\ P_2 \overset{\rho}{\rightsquigarrow}_c P_2'}{P_1\|_A P_2 \overset{\rho}{\rightsquigarrow}_c P_1'\|_A P_2'} \qquad \text{Rec}_c \ \frac{P \overset{\rho}{\rightsquigarrow}_c P'}{\mu X.P \overset{\rho}{\rightsquigarrow}_c P'\{\mu X.P/X\}}$$

For c-process terms $P, P' \in \tilde{\mathbb{P}}_c$ and $\rho \in \mathbb{T}$, we write $P \overset{\rho}{\rightsquigarrow}_c P'$ if $(P, \rho, P') \in \rightsquigarrow_c$. We write $P \overset{\rho}{\rightsquigarrow}_c$, if there exists a $P'' \in \tilde{\mathbb{P}}_c$ such that $(P, \rho, P'') \in \rightsquigarrow_c$. ■ 2.6

Note that a process variable $X \in \mathcal{X}$ has no time semantics, again reflecting the fact that unbound occurrence of a variable means incomplete implementation.

Lemma 2.7 *properties of passage of wait-time*

Let $P, P', P'', Q, R \in \tilde{\mathbb{P}}_c$ be c-process terms, $\rho, \rho', \rho + \rho' \in \mathbb{T}$ and $X \in \mathcal{X}$ be guarded in P.

1. $P \overset{\rho}{\rightsquigarrow}_c$ if and only if P is guarded, and if $P \overset{\rho}{\rightsquigarrow}_c P'$, then P' is guarded.
2. If $P \overset{\rho}{\rightsquigarrow}_c P'$ and $P \overset{\rho}{\rightsquigarrow}_c P''$, then $P' \equiv P''$.
3. $P\{Q/X\} \overset{\rho}{\rightsquigarrow}_c R$ if and only if there exists $P' \in \tilde{\mathbb{P}}_c$ with $P \overset{\rho}{\rightsquigarrow}_c P'$ and $R \equiv P'\{Q/X\}$.
4. If $P \overset{\rho}{\rightsquigarrow}_c P'$, then $\mathcal{A}(P) = \mathcal{A}(P')$.
5. $P \overset{\rho+\rho'}{\rightsquigarrow}_c P''$ if and only if $P \overset{\rho}{\rightsquigarrow}_c P' \overset{\rho'}{\rightsquigarrow}_c P''$ for some P'.

Proof:
1. By induction on the structure of P:
Nil: $\mathbf{0}$ is guarded and $\mathbf{0} \overset{\rho}{\rightsquigarrow}_c \mathbf{0}$.
Var: $P \equiv X$ for $X \in \mathcal{X}$ is not guarded, and for no ρ: $X \overset{\rho}{\rightsquigarrow}_c$.

Pref: $\langle a,r\rangle.P$ is guarded and $\langle a,r\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a,r'\rangle.P$ with $r' = \max(r-\rho,0)$, and $\langle a,r'\rangle.P$ is guarded, too.
Sum: $P \equiv P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c$ iff for $i = 1,2$: $P_i \overset{\rho}{\rightsquigarrow}_c P_i'$ for some P_i', hence by ind. iff both P_1 and P_2 are guarded, in which case both P_1' and P_2' are guarded, too. Thus, $P \equiv P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c$ iff P is guarded and if $P \overset{\rho}{\rightsquigarrow}_c P' \equiv P_1' + P_2'$, then P' is guarded.
Par: analogously to Sum.
Rel: similar to Sum.
Rec: $\mu X.P$ is guarded iff P is guarded iff by ind. $P \overset{\rho}{\rightsquigarrow}_c P'$ for some guarded P' iff $\mu X.P \overset{\rho}{\rightsquigarrow}_c P'\{\mu X.P/X\}$, which is guarded, too.

2. By induction on the inference of $P \overset{\rho}{\rightsquigarrow}_c P'$, $P \overset{\rho}{\rightsquigarrow}_c P''$ resp.

3. For the following reason, it suffices to show the 'if'-direction:

let X be guarded in P and assume that $P \overset{\rho}{\rightsquigarrow}_c P'$ implies $P\{Q/X\} \overset{\rho}{\rightsquigarrow}_c P'\{Q/X\}$; then $P\{Q/X\} \overset{\rho}{\rightsquigarrow}_c R$ implies $P\{Q/X\}$ is guarded by 1., and since X is guarded in P, P itself is guarded; hence $P \overset{\rho}{\rightsquigarrow}_c P'$ for some P' by 1. again, thus $P\{Q/X\} \overset{\rho}{\rightsquigarrow}_c P'\{Q/X\}$ by assumption and $R \equiv P'\{Q/X\}$ by 2.

Induction on the inference of $P \overset{\rho}{\rightsquigarrow}_c P'$, using that P and P' are guarded by 1.:
Nil: $\mathbf{0} \overset{\rho}{\rightsquigarrow}_c \mathbf{0}$ and $\mathbf{0}\{Q/X\} \equiv \mathbf{0}$.
Pref: $\alpha.P \overset{\rho}{\rightsquigarrow}_c \alpha'.P$ and $(\alpha.P)\{Q/X\} \equiv \alpha.(P\{Q/X\}) \overset{\rho}{\rightsquigarrow}_c \alpha'.(P\{Q/X\}) \equiv (\alpha'.P)\{Q/X\}$.
Sum: $P \equiv P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c P'$ iff $\forall_{i=1,2}\ P_i \overset{\rho}{\rightsquigarrow}_c P_i'$ and X guarded in P_i, and $P' \equiv P_1' + P_2'$ by 2. Hence by ind. for $i = 1,2$: $P_i \overset{\rho}{\rightsquigarrow}_c P_i'$ implies $P_i\{Q/X\} \overset{\rho}{\rightsquigarrow}_c P_i'\{Q/X\}$. Thus $P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c P_1' + P_2'$ implies $(P_1 + P_2)\{Q/X\} \equiv P_1\{Q/X\} + P_2\{Q/X\} \overset{\rho}{\rightsquigarrow}_c P_1'\{Q/X\} + P_2'\{Q/X\} \equiv (P_1' + P_2')\{Q/X\}$.
Par: analogously to Sum.
Rel: similar to Sum.
Rec: If $X \equiv Y$, then X not free in $\mu Y.P$, i.e. $(\mu Y.P)\{Q/X\} \equiv \mu Y.P$, and if $\mu Y.P \overset{\rho}{\rightsquigarrow}_c P'$, then $P \overset{\rho}{\rightsquigarrow}_c P''$ for some P'' such that $P' \equiv P''\{\mu Y.P/Y\}$; now X not free in P' either, hence $P'\{Q/X\} \equiv P'$.
Let $X \not\equiv Y$; then $\mu Y.P \overset{\rho}{\rightsquigarrow}_c P'$ implies $P \overset{\rho}{\rightsquigarrow}_c P''$ for some P'' and $P' \equiv P''\{\mu Y.P/X\}$, hence by ind. $P\{Q/X\} \overset{\rho}{\rightsquigarrow}_c P''\{Q/X\}$. Now by Barendregt convention, $(\mu Y.P)$ $\{Q/X\} \equiv \mu Y.(P\{Q/X\})$ and $\mu Y.(P\{Q/X\}) \overset{\rho}{\rightsquigarrow}_c (P''\{Q/X\})\{\mu Y.(P\{Q/X\})/Y\}$ $\equiv (P''\{\mu Y.P/Y\})\{Q/X\} \equiv P'\{Q/X\}$.

4. By induction on the inference of $P \overset{\rho}{\rightsquigarrow}_c P'$. We only consider the Rec-case in the induction, since the other cases are clear:

By rule Rec_c, $\mu X.P \overset{\rho}{\rightsquigarrow}_c R$ implies $R \equiv P'\{\mu X.P/X\}$ and $P \overset{\rho}{\rightsquigarrow}_c P'$ for some P' and by 1., X is guarded in P and P'. Now by rule Rec_a, induction and Proposition 2.5.2: $\mathcal{A}(\mu X.P) = \mathcal{A}(P) = \mathcal{A}(P') = \mathcal{A}(P'\{\mu X.P/X\}) = \mathcal{A}(R)$.

5. Induction:
Nil: $\mathbf{0} \overset{\rho+\rho'}{\rightsquigarrow}_c \mathbf{0}$ and $\mathbf{0} \overset{\rho}{\rightsquigarrow}_c \mathbf{0} \overset{\rho'}{\rightsquigarrow}_c \mathbf{0}$.
Var: For all $X \in \mathcal{X}$ and $\rho, \rho' \in \mathbb{T}$: neither $X \overset{\rho}{\rightsquigarrow}_c$ nor $X \overset{\rho+\rho'}{\rightsquigarrow}_c$.
Pref: $\langle a,r\rangle.P \overset{\rho+\rho'}{\rightsquigarrow}_c \langle a, \max(r-(\rho+\rho'),0)\rangle.P$ and $\langle a,r\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a,\max(r-\rho,0)\rangle.P \overset{\rho'}{\rightsquigarrow}_c \langle a, \max(\max(r-\rho,0)-\rho',0)\rangle.P$ and $\max(\max(r-\rho,0)-\rho',0) = \max(r-(\rho+\rho'),0)$.

Sum: $P_1 + P_2 \overset{\rho+\rho'}{\rightsquigarrow_c} P_1'' + P_2'' \Leftrightarrow \forall_{i=1,2}\, P_i \overset{\rho+\rho'}{\rightsquigarrow_c} P_i'' \Leftrightarrow$ (ind.) $\forall_{i=1,2} \exists P_i' : P_i \overset{\rho}{\rightsquigarrow_c} P_i' \overset{\rho'}{\rightsquigarrow_c} P_i'' \Leftrightarrow$ $\forall_{i=1,2} \exists P_i' : P_1 + P_2 \overset{\rho}{\rightsquigarrow_c} P_1' + P_2' \overset{\rho'}{\rightsquigarrow_c} P_1'' + P_2''$.

Par: analogously to Sum.

Rel: straightforward.

Rec: $\mu X.P \overset{\rho+\rho'}{\rightsquigarrow_c} R$ iff (by rule Rec$_c$) $P \overset{\rho+\rho'}{\rightsquigarrow_c} P''$ for some P'' and $R \equiv P''\{\mu X.P/X\}$ iff (by induction) $\exists P', P'' : P \overset{\rho}{\rightsquigarrow_c} P' \overset{\rho'}{\rightsquigarrow_c} P'' \wedge R \equiv P''\{\mu X.P/X\}$ iff (again by rule Rec$_c$ and by 3. since X is guarded in P and P') $\exists P', P'' : \mu X.P \overset{\rho}{\rightsquigarrow_c} P'\{\mu X.P/X\} \overset{\rho'}{\rightsquigarrow_c} P''\{\mu X.P/X\} \equiv R$. ■ 2.7

The operational semantics of wait-time allows c-processes to wait forever, but our intention was that an urgent action has to occur or be disabled, unless it has to wait for a synchronization partner. We will enforce this using an auxiliary function that calculates for a given action a its residual time $\mathcal{R}(a, P)$ in a c-process term P, i.e. the time until it becomes urgent.

Definition 2.8 *residual time of actions and c-process terms*

The *residual time* of an action $a \in \mathbb{A}_{\omega\tau}$ in a c-process term $P \in \tilde{\mathbb{P}}_c$ is determined by the following inductively defined function $\mathcal{R} : \mathbb{A}_{\omega\tau} \times \tilde{\mathbb{P}}_c \to \mathbb{T}$:

1. Nil: $\mathcal{R}(a, \mathbf{0}) = 1$ for all $a \in \mathbb{A}_{\omega\tau}$
2. Var: $\mathcal{R}(a, X) = 1$ for all $a \in \mathbb{A}_{\omega\tau}$
3. Pref: $\mathcal{R}(a, \alpha.P) = \begin{cases} r & \text{if } \alpha = \langle a, r\rangle \\ 1 & \text{otherwise} \end{cases}$
4. Sum: $\mathcal{R}(a, P_1 + P_2) = \min(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2))$
5. Par: $\mathcal{R}(a, P_1\|_A P_2) = \begin{cases} \max(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2)) & \text{if } a \in A \\ \min(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2)) & \text{if } a \notin A \end{cases}$
6. Rel: $\mathcal{R}(a, P[\Phi]) = \min\{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a)\}$
7. Rec: $\mathcal{R}(a, \mu X.P) = \mathcal{R}(a, P)$

Finally, the residual time of a c-process term $P \in \tilde{\mathbb{P}}_c$ is $\mathcal{R}(P) = \min\{\mathcal{R}(a, P) \mid a \in \mathcal{A}(P)\}$, where $\min \emptyset := 1$. ■ 2.8

We have chosen $\mathcal{R}(a, X) = \mathcal{R}(a, \mathbf{0}) = 1$ mainly for technical reasons (cf. Proposition 2.9.1 below). The Par-case will realize the desired behaviour of waiting in a parallel composition: if P_1 and P_2 have to synchronize on a, then the residual time of a in $P_1\|_A P_2$ is determined by the 'slower' component with larger residual time; if P_1 and P_2 do not have to synchronize on a, the 'faster' component determines the maximal possible delay of a in $P_1\|_A P_2$.

Observe that in the Rel-case (6.) $\Phi^{-1}(a)$ may be empty (where $\min \emptyset = 1$) or infinite; for the latter case, we will show below that for any $P \in \tilde{\mathbb{P}}_c$ there are only finitely many $b \in \mathbb{A}_{\omega\tau}$ with $\mathcal{R}(b, P) \neq 1$ (Proposition 2.9.1 together with Proposition 2.5.1), such that the set $\{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a)\}$ is finite and $\mathcal{R}(a, P[\Phi])$ exists. Similarly, $\mathcal{R}(P)$ exists for each c-process term P, and, hence, the residual time is well-defined in all cases.

In the following Proposition, we ascertain that only activated actions of a c-process term can have a residual time less than 1, and that the residual time of each action in a c-process term is preserved under substitution of guarded variables. Additionally, we show how the residual time of a c-process term can be calculated directly from the residual times of its components, provided there is no parallel composition with synchronisation:

Proposition 2.9

Let $P, P_1, P_2, Q \in \tilde{\mathbb{P}}_c$ be c-process terms, $a \in \mathbb{A}_{\omega\tau}$ and $X \in \mathcal{X}$.

1. $\mathcal{R}(a, P) \in \mathbb{T}$, and $\mathcal{R}(a, P) \neq 1$ implies $a \in \mathcal{A}(P)$.
2. If X is guarded in P, then $\mathcal{R}(a, P) = \mathcal{R}(a, P\{Q/X\})$, thus $\mathcal{R}(P) = \mathcal{R}(P\{Q/X\})$.
3. Except for parallel composition, $\mathcal{R}(P)$ may be calculated directly:

1.	Nil:	$\mathcal{R}(\mathbf{0}) = 1$
2.	Var:	$\mathcal{R}(X) = 1$
3.	Pref:	$\mathcal{R}(\langle a, r\rangle.P) = r$
4.	Sum:	$\mathcal{R}(P_1 + P_2) = \min(\mathcal{R}(P_1), \mathcal{R}(P_2))$
5.	Rel:	$\mathcal{R}(P[\Phi]) = \mathcal{R}(P)$
6.	Rec:	$\mathcal{R}(\mu X.P) = \mathcal{R}(P)$

Proof:

1. Induction on the structure of P; $\mathcal{R}(a, \mathbf{0}) = \mathcal{R}(a, X) = 1$ for all $a \in \mathbb{A}_{\omega\tau}$. Now:

Pref: $\mathcal{R}(a, \langle a, r\rangle.P) = r \in \mathbb{T}$ and $a \in \mathcal{A}(\langle a, r\rangle.P)$. If $a \neq a'$, then $\mathcal{R}(a, \langle a', r\rangle.P) = 1 \in \mathbb{T}$.

Sum: $\mathcal{R}(a, P_1 + P_2) = \min(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2) \in \mathbb{T}$ since by ind. $\mathcal{R}(a, P_1), \mathcal{R}(a, P_2) \in \mathbb{T}$. If $\mathcal{R}(a, P_1 + P_2) \neq 1$, then w.l.o.g. $\mathcal{R}(a, P_1) \neq 1$, hence by ind. $a \in \mathcal{A}(P_1) \subseteq \mathcal{A}(P_1) \cup \mathcal{A}(P_2) = \mathcal{A}(P_1 + P_2)$.

Par: By ind. $\mathcal{R}(a, P_1), \mathcal{R}(a, P_2) \in \mathbb{T}$, hence $\mathcal{R}(a, P_1\|_A P_2) \in \{\min(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2)), \max(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2))\} \subseteq \mathbb{T}$.
If $a \in A$ and $\mathcal{R}(a, P_1\|_A P_2) \neq 1$, then $\mathcal{R}(a, P_1) \neq 1 \neq \mathcal{R}(a, P_2)$, hence by ind. $a \in \mathcal{A}(P_1) \cap \mathcal{A}(P_2)$, thus $a \in \mathcal{A}(P_1\|_A P_2)$.
If $a \notin A$ and $\mathcal{R}(a, P_1\|_A P_2) \neq 1$, then $\mathcal{R}(a, P_1) \neq 1 \vee \mathcal{R}(a, P_2) \neq 1$, hence by ind. $a \in \mathcal{A}(P_1) \cup \mathcal{A}(P_2)$, thus $a \in \mathcal{A}(P_1\|_A P_2)$.

Rel: $\{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a)\} = \{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a) \cap \mathcal{A}(P)\} \cup \{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a) \setminus \mathcal{A}(P)\}$, and by ind., $\{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a) \setminus \mathcal{A}(P)\} = \{1\}$ and, hence $\{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a)\} = \{1\} \cup \{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a) \cap \mathcal{A}(P)\}$, which is finite, thus $\mathcal{R}(a, P[\Phi]) = \min(\{1\} \cup \{\mathcal{R}(b, P) \mid b \in \Phi^{-1}(a) \cap \mathcal{A}(P)\}) \in \mathbb{T}$ by ind. If $\mathcal{R}(a, P) \neq 1$, then $\mathcal{R}(b, P) \neq 1$ for some $b \in \Phi^{-1}(a) \cap \mathcal{A}(P)$, hence by ind. $b \in \mathcal{A}(P)$, and $\Phi(b) = a$ implies $a \in \mathcal{A}(P[\Phi])$.

Rec: $\mathcal{R}(a, \mu X.P) = \mathcal{R}(a, P) \in \mathbb{T}$ by ind., and $\mathcal{R}(a, \mu X.P) = \mathcal{R}(a, P) \neq 1$ implies $a \in \mathcal{A}(P)$ by ind., hence $a \in \mathcal{A}(\mu X.P)$.

2. Induction on the structure of P:

Nil: $\mathcal{R}(a, \mathbf{0}) = 1$ for all $a \in \mathbb{A}_{\omega\tau}$ and $\mathbf{0}\{Q/X\} \equiv \mathbf{0}$.

Var: X is not guarded in $P \equiv X$, and the case $P \equiv Y \not\equiv X$ is analogously to Nil.

Pref: $(\alpha.P)\{Q/X\} \equiv \alpha.(P\{Q/X\})$, hence $\mathcal{R}(a, \alpha.P) = \mathcal{R}(a, \alpha.(P\{Q/X\})) = \mathcal{R}(a, (\alpha.P)\{Q/X\})$.

Sum: X guarded in $P_1 + P_2 \Rightarrow X$ guarded in P_1 and X guarded in P_2. $\mathcal{R}(a, P_1 + P_2) = \min_{i=1,2} \mathcal{R}(a, P_i) = \min_{i=1,2} \mathcal{R}(a, P_i\{Q/X\}) = \mathcal{R}(a, P_1\{Q/X\} + P_2\{Q/X\}) = \mathcal{R}(a, (P_1 + P_2)\{Q/X\})$.

Par, Rel: analogously.

Rec: If $X \equiv Y$, then X is bound in $\mu Y.P$, hence $\mathcal{R}(a, \mu Y.P) = \mathcal{R}(a, (\mu Y.P)\{Q/X\})$.
If $X \not\equiv Y$, then $(\mu Y.P)\{Q/X\} \equiv \mu Y.(P\{Q/X\})$ by BARENDREGT convention, and X guarded in $\mu Y.P$ implies X guarded in P, hence by ind. $\mathcal{R}(a, (\mu Y.P)\{Q/X\}) = \mathcal{R}(a, \mu Y.(P\{Q/X\})) = \mathcal{R}(a, P\{Q/X\}) = \mathcal{R}(a, P) = \mathcal{R}(a, \mu Y.P)$.

3. We exploit the finiteness of $\mathcal{A}(P)$ and the restriction on general relabelling functions Φ in order to swap minima:

1. $\mathcal{R}(\mathbf{0}) = \min_{a \in \mathcal{A}(\mathbf{0})} \mathcal{R}(a, \mathbf{0}) = \min_{a \in \emptyset} \mathcal{R}(a, \mathbf{0}) = 1$.
2. $\mathcal{R}(X) = \min_{a \in \mathcal{A}(X)} \mathcal{R}(a, X) = \min_{a \in \emptyset} \mathcal{R}(a, X) = 1$.
3. $\mathcal{R}(\langle a, r\rangle.P) = \min_{a' \in \mathcal{A}(\langle a,r\rangle.P)} \mathcal{R}(a', \langle a, r\rangle.P) = \mathcal{R}(a, \langle a, r\rangle.P) = r$.
4. $\mathcal{R}(P_1 + P_2) = \min_{a \in \mathcal{A}(P_1+P_2)} \mathcal{R}(a, P_1 + P_2) = \min_{a \in \mathcal{A}(P_1) \cup \mathcal{A}(P_2)} \min(\mathcal{R}(a, P_1), \mathcal{R}(a, P_2))$
   $= \min(\min_{a \in \mathcal{A}(P_1) \cup \mathcal{A}(P_2)} \mathcal{R}(a, P_1), \min_{a \in \mathcal{A}(P_1) \cup \mathcal{A}(P_2)} \mathcal{R}(a, P_2))$
   $\overset{1.}{=} \min(\min_{a \in \mathcal{A}(P_1)} \mathcal{R}(a, P_1), \min_{a \in \mathcal{A}(P_2)} \mathcal{R}(a, P_2)) = \min(\mathcal{R}(P_1), \mathcal{R}(P_2))$.
5. $\mathcal{R}(P[\Phi]) = \min_{a \in \mathcal{A}(P[\Phi])} \mathcal{R}(a, P[\Phi]) = \min_{a \in \Phi(\mathcal{A}(P))} \min_{b \in \Phi^{-1}(a) \cap \mathcal{A}(P)} \mathcal{R}(b, P) \overset{1.}{=} \min_{b \in \mathcal{A}(P)} \mathcal{R}(b, P) = \mathcal{R}(P)$.
6. $\mathcal{R}(\mu X.P) = \min_{a \in \mathcal{A}(\mu X.P)} \mathcal{R}(a, \mu X.P)) = \min_{a \in \mathcal{A}(P)} \mathcal{R}(a, P) = \mathcal{R}(P)$. ■ 2.9

The effect of waiting on the residual time of activated actions is described by the following lemma: if time advances by amount ρ, then the residual time of an activated action is decreased by the same amount, unless it has already been less than ρ, in which case it is zero afterwards. This behaviour is realized locally by rule Pref_c of Definition 2.6.

Lemma 2.10

For c-process terms $P, P' \in \tilde{\mathbb{P}}_c$ and $\rho \in \mathbb{T}$ let $P \overset{\rho}{\rightsquigarrow}_c P'$; then for all $a \in \mathcal{A}(P) = \mathcal{A}(P')$ we have either $\mathcal{R}(a, P) - \mathcal{R}(a, P') = \rho$, or $\mathcal{R}(a, P) < \rho$ and $\mathcal{R}(a, P') = 0$.

Proof:

In this proof, we will deal with minima and maxima in order to calculate residual times. In these calculations, we will often use the following properties without mentioning it:

Let I be a finite set, $\rho \in \mathbb{T}$ and $(x_i)_{i \in I}$, $(y_i)_{i \in I}$ be families of real numbers.
1. $\min_{i \in I} (x_i - y_i) \leq \min_{i \in I} x_i - \min_{i \in I} y_i$.
2. If $x_i - y_i \leq \rho$ for all $i \in I$, then $\min_{i \in I} x_i - \min_{i \in I} y_i \leq \rho$.
3. If $x_i - y_i = \rho$ for all $i \in I$, then $\min_{i \in I} x_i - \min_{i \in I} y_i = \rho$.
4. $\max_{i \in I} (x_i - y_i) \geq \max_{i \in I} x_i - \max_{i \in I} y_i$.
5. If $x_i - y_i \leq \rho$ for all $i \in I$, then $\max_{i \in I} x_i - \max_{i \in I} y_i \leq \rho$.
6. If $x_i - y_i = \rho$ for all $i \in I$, then $\max_{i \in I} x_i - \max_{i \in I} y_i = \rho$.

Proof:
1. Let $\min_{i \in I} x_i = x_j$ with $j \in I$; then $\min_{i \in I} (x_i - y_i) \leq x_j - y_j \leq x_j - \min_{i \in I} y_i = \min_{i \in I} x_i - \min_{i \in I} y_i$.
2. Let $\min_{i \in I} y_i = y_k$ with $k \in I$; then $\min_{i \in I} x_i - \min_{i \in I} y_i = \min_{i \in I} x_i - y_k \leq x_k - y_k \leq \rho$.
3. Follows from 1. and 2.
4. Let $\max_{i \in I} x_i = x_j$ with $j \in I$; then $\max_{i \in I} (x_i - y_i) \geq x_j - y_j \geq x_j - \max_{i \in I} y_i = \max_{i \in I} x_i - \max_{i \in I} y_i$.
5. Follows from 4.

6. $\max_{i\in I} x_i - \max_{i\in I} y_i \leq \rho$ by 4. Let $\max_{i\in I} y_i = y_k$ with $k \in I$; then $\rho = x_k - y_k \leq \max_{i\in I} x_i - y_k = \max_{i\in I} x_i - \max_{i\in I} y_i$.

After verifying $\mathcal{A}(P) = \mathcal{A}(P')$ with Lemma 2.7.4, we now perform induction on the structure of the inference tree of $P \overset{\rho}{\rightsquigarrow}_c P'$, using Lemma 2.7.4 again; the property trivially holds for $\mathbf{0}$ and X, since $\mathcal{A}(\mathbf{0}) = \mathcal{A}(X) = \emptyset$; now:

Pref: $\langle a,r\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, \max(r-\rho,0)\rangle.P \Rightarrow (\mathcal{R}(a,\langle a,r\rangle.P) = r < \rho \wedge \mathcal{R}(a,\langle a,\max(r-\rho,0)\rangle.P) = \mathcal{R}(a,\langle a,0\rangle.P) = 0) \vee (\mathcal{R}(a,\langle a,r\rangle.P) = r \geq \rho \wedge \mathcal{R}(a,\langle a,r\rangle.P) - \mathcal{R}(a,\langle a,\max(r-\rho,0)\rangle.P) = r - (r-\rho) = \rho$.

Sum: $P_1 + P_2 \overset{\rho}{\rightsquigarrow}_c P_1' + P_2' \Rightarrow \forall_{i=1,2}\, P_i \overset{\rho}{\rightsquigarrow}_c P_i' \Rightarrow \forall_{i=1,2}\forall_{a\in\mathcal{A}(P_i)=\mathcal{A}(P_i')}\, (\mathcal{R}(a,P_i) - \mathcal{R}(a,P_i') = \rho \vee (\mathcal{R}(a,P_i) < \rho \wedge \mathcal{R}(a,P_i') = 0))$. For $a \in \mathcal{A}(P_1+P_2) = \mathcal{A}(P_1'+P_2')$ by ind. one of the following cases applies:
  i) $\exists_{i=1,2}\, \mathcal{R}(a,P_i) < \rho \wedge \mathcal{R}(a,P_i') = 0$; then $\mathcal{R}(a,P_1+P_2) = \min_{i=1,2}\mathcal{R}(a,P_i) < \rho$ and $\mathcal{R}(a,P_1'+P_2') = \min_{i=1,2}\mathcal{R}(a,P_i') = 0$.
  ii) $\forall_{i=1,2}\, \mathcal{R}(a,P_i) - \mathcal{R}(a,P_i') = \rho$; then $\mathcal{R}(a,P_1+P_2) - \mathcal{R}(a,P_1'+P_2') = \min_{i=1,2}\mathcal{R}(a,P_i) - \min_{i=1,2}\mathcal{R}(a,P_i') = \rho$.
  iii) $\mathcal{R}(a,P_1) - \mathcal{R}(a,P_1') = \rho \wedge \mathcal{R}(a,P_2) = \mathcal{R}(a,P_2') = 1$ by 2.9.1 (or vice versa); then $\mathcal{R}(a,P_1+P_2) - \mathcal{R}(a,P_1'+P_2') = \min_{i=1,2}\mathcal{R}(a,P_i) - \min_{i=1,2}\mathcal{R}(a,P_i') = \mathcal{R}(a,P_1) - \mathcal{R}(a,P_1') = \rho$.

Par: $P_1\|_A P_2 \overset{\rho}{\rightsquigarrow}_c P_1'\|_A P_2' \Rightarrow \forall_{i=1,2}\, P_i \overset{\rho}{\rightsquigarrow}_c P_i'$.
  For any $a \in (\mathcal{A}(P_1)\cap\mathcal{A}(P_2))\cap A$ by ind. one of the following cases applies:
  i) $\forall_{i=1,2}\, \mathcal{R}(a,P_i) < \rho \wedge \mathcal{R}(a,P_i') = 0$; then $\mathcal{R}(a,P_1\|_A P_2) = \max_{i=1,2}\mathcal{R}(a,P_i) < \rho$ and $\mathcal{R}(a,P_1'\|_A P_2') = \max_{i=1,2}\mathcal{R}(a,P_i') = 0$.
  ii) $\mathcal{R}(a,P_1) - \mathcal{R}(a,P_1') = \rho \wedge \mathcal{R}(a,P_1) \geq \mathcal{R}(a,P_2)$ (or vice versa); then $\mathcal{R}(a,P_1\|_A P_2) - \mathcal{R}(a,P_1'\|_A P_2') = \max_{i=1,2}\mathcal{R}(a,P_i) - \max_{i=1,2}\mathcal{R}(a,P_i') = \mathcal{R}(a,P_1) - \mathcal{R}(a,P_1') = \rho$.
  For any $a \in (\mathcal{A}(P_1)\cup\mathcal{A}(P_2))\setminus A$ by ind. one of the following cases applies:
  i) $\exists_{i=1,2}\mathcal{R}(a,P_i) < \rho \wedge \mathcal{R}(a,P_i') = 0$; then $\mathcal{R}(a,P_1\|_A P_2) = \min_{i=1,2}\mathcal{R}(a,P_i) < \rho$ and $\mathcal{R}(a,P_1'\|_A P_2') = \min_{i=1,2}\mathcal{R}(a,P_i') = 0$.
  ii) $\forall_{i=1,2}\mathcal{R}(a,P_i) - \mathcal{R}(a,P_i') = \rho$; then $\mathcal{R}(a,P_1\|_A P_2) - \mathcal{R}(a,P_1'\|_A P_2') = \min_{i=1,2}\mathcal{R}(a,P_i) - \min_{i=1,2}\mathcal{R}(a,P_i') = \rho$.
  iii) $\mathcal{R}(a,P_1) - \mathcal{R}(a,P_1') = \rho \wedge \mathcal{R}(a,P_2) = \mathcal{R}(a,P_2') = 1$ by 2.9.1 (or vice versa); then $\mathcal{R}(a,P_1\|_A P_2) - \mathcal{R}(a,P_1'\|_A P_2') = \min_{i=1,2}\mathcal{R}(a,P_i) - \min_{i=1,2}\mathcal{R}(a,P_i') = \rho$.

Rel: $P[\Phi] \overset{\rho}{\rightsquigarrow}_c P'[\Phi] \Rightarrow P \overset{\rho}{\rightsquigarrow}_c P'$
  $\Rightarrow \forall_{a\in\mathcal{A}(P)=\mathcal{A}(P')}(\mathcal{R}(a,P) - \mathcal{R}(a,P') = \rho \vee (\mathcal{R}(a,P) < \rho \wedge \mathcal{R}(a,P') = 0))$
  $\Rightarrow \forall_{a\in\mathcal{A}(P[\Phi])=\mathcal{A}(P'[\Phi])}(\min_{b\in\Phi^{-1}(a)}\mathcal{R}(b,P) - \min_{b\in\Phi^{-1}(a)}\mathcal{R}(b,P') = \rho \vee$
  $(\min_{b\in\Phi^{-1}(a)}\mathcal{R}(b,P) < \rho \wedge \min_{b\in\Phi^{-1}(a)}\mathcal{R}(b,P') = 0))$
  $\Rightarrow \forall_{a\in\mathcal{A}(P[\Phi])=\mathcal{A}(P'[\Phi])}(\mathcal{R}(a,P[\Phi]) - \mathcal{R}(a,P'[\Phi]) = \rho \vee$
  $(\mathcal{R}(a,P[\Phi]) < \rho \wedge \mathcal{R}(a,P'[\Phi]) = 0))$.

Rec: $\mu X.P \overset{\rho}{\rightsquigarrow}_c R$ implies $R \equiv P'\{\mu X.P/X\}$ and $P \overset{\rho}{\rightsquigarrow}_c P'$ for some P' by rule Rec$_c$, hence induction yields
  $\forall_{a\in\mathcal{A}(P)=\mathcal{A}(P')}\, \mathcal{R}(a,P) - \mathcal{R}(a,P') = \rho \vee (\mathcal{R}(a,P) < \rho \wedge \mathcal{R}(a,P') = 0)$
  and since P' is guarded by Lemma 2.7.1, with Proposition 2.9.2 follows
  $\forall_{a\in\mathcal{A}(\mu X.P)=\mathcal{A}(P'\{\mu X.P/X\})}\mathcal{R}(a,\mu X.P) - \mathcal{R}(a,P'\{\mu X.P/X\}) = \mathcal{R}(a,P) - \mathcal{R}(a,P') = \rho \vee (\mathcal{R}(a,\mu X.P) = \mathcal{R}(a,P) < \rho \wedge \mathcal{R}(a,P'\{\mu X.P/X\}) = \mathcal{R}(a,P') = 0)$. ■ 2.10

Using the residual time of a c-process term, we are now able to restrict wait-time to the timed behaviour we had in mind originally and which we call 'idle-time'. Alternatively, idle-time could have been defined via SOS-rules intertwined with the rules for wait-time.

Definition 2.11 *passage of idle-time*

For $P, P' \in \tilde{\mathbb{P}}_c$ and $\rho \in \mathbb{T}$ we write $P \xrightarrow{\rho}_c P'$ if $P \overset{\rho}{\rightsquigarrow}_c P'$ and $\rho \leq \mathcal{R}(P)$. ■ 2.11

Most of the properties of wait-time stated in Lemma 2.7 carry over to idle-time analogously, gathered in Proposition 2.12 below. Note that c-processes without activated actions may idle for an arbitrary amount of time by 4., 1. and 3., but if there are activated actions, they may idle at most for time 1 by 5., 1. and 3.

Proposition 2.12 *properties of idle-time*

Let $P, P', P'', Q, R \in \tilde{\mathbb{P}}_c$ be c-process terms and $\rho, \rho', \rho + \rho' \in \mathbb{T}$.

1. $P \xrightarrow{\rho}_c$ iff P is guarded and $\rho \leq \mathcal{R}(P)$, and P' is guarded if $P \xrightarrow{\rho}_c P'$. *(urgency)*
2. If $P \xrightarrow{\rho}_c P'$ and $P \xrightarrow{\rho}_c P''$, then $P' \equiv P''$. *(determinism)*
3. If $P \xrightarrow{\rho}_c P'$, then $\mathcal{A}(P) = \mathcal{A}(P')$. *(persistency)*
4. If $\mathcal{A}(P) = \emptyset$ and $P \xrightarrow{\rho}_c P'$, then $\mathcal{R}(P) = \mathcal{R}(P') = 1$. *(termination)*
5. If $\mathcal{A}(P) \neq \emptyset$ and $P \xrightarrow{\rho}_c P'$, then $\mathcal{R}(P) - \mathcal{R}(P') = \rho$ *(progress)*
6. $P \xrightarrow{\rho+\rho'}_c P''$ if and only if $P \xrightarrow{\rho}_c P' \xrightarrow{\rho'}_c P''$ for some P'. *(continuity)*

Proof:

1. Implication of Definition 2.11 and Lemma 2.7.1.

2. Implication of Definition 2.11 and Lemma 2.7.2.

3. Implication of Definition 2.11 and Lemma 2.7.4.

4. Implication of 3. and Proposition 2.9.1.

5. Implication of Lemma 2.10, Definition 2.11 and Definition 2.8.

6. Follows from Definition 2.11 and Lemma 2.7.5:

$P \xrightarrow{\rho+\rho'}_c P''$

$\Leftrightarrow P \overset{\rho+\rho'}{\rightsquigarrow}_c P'' \wedge \rho + \rho' \leq \mathcal{R}(P)$ (by Def. 2.11)

$\Leftrightarrow \exists P' : P \overset{\rho}{\rightsquigarrow}_c P' \overset{\rho'}{\rightsquigarrow}_c P'' \wedge \rho \leq \mathcal{R}(P) \wedge \rho' \leq \mathcal{R}(P) - \rho$ (by Lemma 2.7.5)

$\Leftrightarrow \exists P' : P \xrightarrow{\rho}_c P' \overset{\rho'}{\rightsquigarrow}_c P'' \wedge \rho' \leq \mathcal{R}(P')$ (by Def. 2.11 and 4. and 5.)

$\Leftrightarrow \exists P' : P \xrightarrow{\rho}_c P' \xrightarrow{\rho'}_c P''$ (by Def. 2.11) ■ 2.12

Both, purely functional and timed behaviour of processes will now be combined in the continuous language of processes. As usual, we will abstract from internal behaviour, but note that internal actions gain some 'visibility' in timed behaviour, since their presence possibly allows to pass more time in between the occurrence of visible actions. For technical reasons, we also need a continuous language that records τ's when we compare processes w.r.t. their temporal progress in the next section.

Definition 2.13 *continuous language of processes*

Let $P, P' \in \tilde{\mathbb{P}}_c$ be *c*-process terms. We write $P \xrightarrow{\varepsilon}_c P'$ if either $\varepsilon \in \mathbb{A}_{\omega\tau}$ and $P \xrightarrow{\varepsilon} P'$, or $\varepsilon \in \mathbb{T}$ and $P \xrightarrow{\varepsilon}_c P'$. We extend this to sequences w and write $P \xrightarrow{w}_c P'$ if $P \equiv P'$ and $w = \lambda$ or there exist $Q \in \mathbb{P}_c$ and $\varepsilon \in (\mathbb{A}_{\omega\tau} \cup \mathbb{T})$ such that $P \xrightarrow{\varepsilon}_c Q \xrightarrow{w'}_c P'$ and $w = \varepsilon w'$.

For a sequence $w \in (\mathbb{A}_{\omega\tau} \cup \mathbb{T})^*$ let w/τ be the sequence w with all τ's removed, let $act(w)$ be the sequence of elements from $\mathbb{A}_{\omega\tau}$ in w, and let $\zeta(w)$ be the sum of time steps in w; note that $\zeta(w/\tau) = \zeta(w)$. We write $P \overset{v}{\Rightarrow}_c P'$, if $P \xrightarrow{w}_c P'$ and $v = w/\tau$.

For a c-process $P \in \mathbb{P}_c$ we define $\mathsf{CL}_\tau(P) = \{w \mid P \xrightarrow{w}_c\}$ to be the *continuous τ-language*, containing the *continuous τ-traces* of P, and $\mathsf{CL}(P) = \{w \mid P \overset{w}{\Rightarrow}_c\}$ to be the *continuous language*, containing the *continuous traces* of P. ■ 2.13

We state in passing that the set of c-processes is closed under occurrence of actions or passage of time, i.e. $P \in \mathbb{P}_c$ and $P \xrightarrow{w}_c P'$ implies $P' \in \mathbb{P}_c$ again.

Based on the continuous language of c-processes, we are now ready to define timed testing and to relate c-processes w.r.t. their efficiency, thereby defining an *efficiency preorder*:

Definition 2.14 *continuously timed tests*

An *initial* process $P \in \mathbb{P}_1$ is *testable* if ω does not occur in P. Any initial process $O \in \mathbb{P}_1$ may serve as a *test process (observer)*.

A *c-timed test* is a pair (O, R), where O is a test process and $R \in \mathbb{R}_0^+$ is the *real time bound*. A testable process P *c-satisfies* a c-timed test (P $must_c$ (O, R)), if each $w \in \mathsf{CL}(\tau.P\|_{\mathbb{A}}O)$ with $\zeta(w) > R$ contains some ω.

For testable processes P and Q, we call P a *continuously faster implementation* of Q, written $P \sqsupseteq_c Q$, if P c-satisfies all c-timed tests that Q c-satisfies. ■ 2.14

Note that in contrast to e.g. [DNH84], execution and not only activation of an ω is necessary for satisfaction of a c-timed test. Note that $\tau.P\|_{\mathbb{A}}O$ is a shorthand for $(\langle\tau, 1\rangle.P)\|_{\mathbb{A}}O$. Usually, one considers the behaviour of $P\|_{\mathbb{A}}O$ when defining a test. This is also done in [Jen96], where it is shown that surprisingly the resulting efficiency preorder is not a precongruence for prefix and therefore has to be refined afterwards. In order to avoid this complication, we have chosen $\tau.P\|_{\mathbb{A}}O$ instead, gaining the same result directly. From an intuitive point of view, the additional τ-prefix represents some internal setup activity before the actual test begins.

Runs with duration less than R may not contain all actions that occur up to time R; hence we only consider runs with a duration greater than the time bound R for test satisfaction. Observe that this definition of c-satisfaction would be of questionable usefulness, if c-processes were able to stop time, i.e. to reach a state from where no time step is possible any more; we will see later on (cf. Corollary 3.4.4) that this doubt is unsubstantiated.

At this point, it is by no means clear how to check $P \sqsupseteq_c Q$ for given testable P and Q. Obviously, it is impossible to apply the definition directly, since there are uncountably many time bounds and, hence, c-timed tests to apply. And even if we could decide $P \sqsupseteq_c Q$ from $\mathsf{CL}(P)$ and $\mathsf{CL}(Q)$ only (which is not the case), $\mathsf{CL}(P)$ and $\mathsf{CL}(Q)$ are still uncountable and hard to handle.

3 Discretization

Intuitively, satisfaction of a c-timed test essentially depends on the 'slowest' sequences in $\mathsf{CL}(\tau.P\|_{\mathbb{A}}O)$; in this section, we will show that these are generated by discrete behaviour only, i.e. those traces with only time steps of duration 1. This will yield a simple theory.

Definition 3.1 *discrete language of processes*

Let $P, P' \in \tilde{\mathbb{P}}_c$ be *c*-process terms. We write $P \xrightarrow{\varepsilon}_d P'$ if either $\varepsilon \in \mathbb{A}_{\omega\tau}$ and $P \xrightarrow{\varepsilon} P'$, or $\varepsilon = 1$ and $P \xrightarrow{\varepsilon}_c P'$; in the latter case we say that P performs a *unit time step*. For sequences $w \in (\mathbb{A}_{\omega\tau} \cup \{1\})^*$, we define $P \xrightarrow{w}_d$ and $P \xRightarrow{w/\tau}_d$ analogously to Definition 2.13.

For a c-process $P \in \mathbb{P}_c$ we define $\mathsf{DL}_\tau(P) = \{w \mid P \xrightarrow{w}_d\}$ to be the *discrete τ-language*, containing the *discrete τ-traces* of P, and $\mathsf{DL}(P) = \{w/\tau \mid w \in \mathsf{DL}_\tau(P)\}$ to be the *discrete language*, containing the *discretes traces* of P. ■ 3.1

Observe that by definition $\mathsf{DL}(P) \subseteq \mathsf{CL}(P)$ and $\mathsf{DL}_\tau(P) \subseteq \mathsf{CL}_\tau(P)$.

We are mainly interested in initial processes (which can be seen as the processes of an ordinary untimed process algebra). Therefore, we will first characterize syntactically those c-processes in $\mathbb{P}_c$ that are reachable from an initial process by only discrete behaviour. These terms represent a discretely timed process algebra and their structure is important e.g. in the proof of Proposition 3.3.

Definition 3.2 *process terms and urgent process terms*

An *urgent process term* U is generated by the following grammar:

$$U \ ::= \ \mathbf{0} \ \Big| \ \underline{a}.I \ \Big| \ U + U \ \Big| \ U\|_A U \ \Big| \ U[\Phi] \qquad \text{(recall that } \underline{a} \text{ is } \langle a, 0\rangle)$$

where $I \in \tilde{\mathbb{P}}_1$ is an *initial* process term, $a \in \mathbb{A}_{\omega\tau}$, $A \subseteq \mathbb{A}$ and Φ a general relabeling function. The set of urgent processes terms is denoted by $\tilde{\mathbb{P}}_0$, and $\mathbb{P}_0 = \tilde{\mathbb{P}}_0 \cap \mathbb{P}_c$ is the set of *urgent processes*;

A *(discretely timed) process term* P is generated by the following grammar:

$$P \ ::= \ I \ \Big| \ U \ \Big| \ P\|_A P \ \Big| \ P[\Phi]$$

where I, A, Φ are as above and $U \in \tilde{\mathbb{P}}_0$ is an *urgent* process term. The set of process terms is denoted by $\tilde{\mathbb{P}}$, the set of (discretely timed) processes is $\mathbb{P} = \tilde{\mathbb{P}} \cap \mathbb{P}_c$. Obviously $\tilde{\mathbb{P}}_1 \cup \tilde{\mathbb{P}}_0 \subseteq \tilde{\mathbb{P}} \subseteq \tilde{\mathbb{P}}_c$. ■ 3.2

Intuitively, an urgent process term is reached whenever a process term performs a unit time step. Hence, an urgent process term usually must not let time pass further; but this is allowed for process terms without activated actions: consider $\mathbf{0} \in \mathbb{P}_0 \cup \mathbb{P}_1$ for this case, which can be seen as both, initial and urgent. Process variables $X \in \mathcal{X}$ are always initial, since they may not let pass time at all, hence cannot be reached by a time step.

Proposition 3.3

Let $P \in \tilde{\mathbb{P}}$ be a process term and $a \in \mathbb{A}_{\omega\tau}$.

1. $\mathcal{R}(a,P) \in \{0,1\}$, thus $\mathcal{R}(P) \in \{0,1\}$; if $P \in \tilde{\mathbb{P}}_1$, then $\mathcal{R}(a,P) = 1$, thus $\mathcal{R}(P) = 1$.
2. If $P \xrightarrow{a} P'$, then $P' \in \tilde{\mathbb{P}}$ and $\forall_{b \in \mathbb{A}_{\omega\tau}} \mathcal{R}(b,P) \leq \mathcal{R}(b,P')$, and $P' \in \tilde{\mathbb{P}}_1$ if $P \in \tilde{\mathbb{P}}_1$.
3. If $P \overset{1}{\leadsto}_c P'$, then $P' \in \tilde{\mathbb{P}}_0$; in particular, $P \xrightarrow{1}_d P'$ implies $P' \in \tilde{\mathbb{P}}_0$.
4. There are $P_1 \in \tilde{\mathbb{P}}_1$ and $w \in \{\tau\}^*$, such that $P_1 \xrightarrow{1w}_d P$, where $w = \lambda$ if $P \in \tilde{\mathbb{P}}_0$.
5. There are $P' \in \tilde{\mathbb{P}}$ and $w \in \{a\}^*$, such that $P \xrightarrow{w}_d P'$ and $\mathcal{R}(a,P') = 1$.

Proof:

1. By induction on the structure of P using Definition 2.2 of initial processes, Definition 3.2 of processes and Definition 2.8.

2. Induction on the structure of P:

Clear for $\mathbf{0}$ and $X \in \mathcal{X}$.

Pref: $\langle a,1\rangle.P \xrightarrow{a} P \in \tilde{\mathbb{P}}$ and $\forall_{b \in \mathbb{A}_{\omega\tau}} \mathcal{R}(b,P) = 1$, since $P \in \tilde{\mathbb{P}}_1$. Analogously for $\langle a,0\rangle.P$.

Sum: If $P_1 + P_2 \xrightarrow{a} P_1'$ then w.l.o.g. $P_1 \xrightarrow{a} P_1'$ and since $P_1 \in \tilde{\mathbb{P}}_0 \cup \tilde{\mathbb{P}}_1 \subseteq \tilde{\mathbb{P}}$ by ind. $P_1' \in \tilde{\mathbb{P}}$, too. Furthermore, by ind., $\forall_{b \in \mathbb{A}_{\omega\tau}} \mathcal{R}(b, P_1 + P_2) = \min(\mathcal{R}(b,P_1), \mathcal{R}(b,P_2)) \leq \mathcal{R}(b,P_1) \leq \mathcal{R}(b,P_1')$. Additionally, $P_1 + P_2 \in \tilde{\mathbb{P}}_1$ iff $P_1, P_2 \in \tilde{\mathbb{P}}_1$, hence by ind. $P_1' \in \tilde{\mathbb{P}}_1$.

Par: Let $a \in A$; then $P_1 \|_A P_2 \xrightarrow{a} P_1' \|_A P_2'$ implies $P_1 \xrightarrow{a} P_1'$ and $P_2 \xrightarrow{a} P_2'$, hence by ind. $P_1', P_2' \in \tilde{\mathbb{P}}$, thus $P_1' \|_A P_2' \in \tilde{\mathbb{P}}$. Furthermore, by ind., for all $b \in A$: $\mathcal{R}(b, P_1 \|_A P_2) = \max(\mathcal{R}(b,P_1), \mathcal{R}(b,P_2)) \leq \max(\mathcal{R}(b,P_1'), \mathcal{R}(b,P_2')) = \mathcal{R}(b, P_1' \|_A P_2')$ and for all $b \in \mathbb{A}_{\omega\tau} \setminus A$: $\mathcal{R}(b, P_1 \|_A P_2) = \min(\mathcal{R}(b,P_1), \mathcal{R}(b,P_2)) \leq \min(\mathcal{R}(b,P_1'), \mathcal{R}(b,P_2')) = \mathcal{R}(b, P_1' \|_A P_2')$.
Now let $a \notin A$; then w.l.o.g. $P_1 \|_A P_2 \xrightarrow{a} P_1' \|_A P_2$ with $P_1 \xrightarrow{a} P_1'$, hence by ind. $P_1' \in \tilde{\mathbb{P}}$, thus $P_1' \|_A P_2 \in \tilde{\mathbb{P}}$. $\mathcal{R}(b, P_1 \|_A P_2) \leq \mathcal{R}(b, P_1' \|_A P_2)$ follows as above.
Furthermore, $P_1 \|_A P_2 \in \tilde{\mathbb{P}}_1$ iff $P_1, P_2 \in \tilde{\mathbb{P}}_1$, hence by ind. in both cases $P_1' \|_A P_2' \in \tilde{\mathbb{P}}_1$, $P_1' \|_A P_2 \in \tilde{\mathbb{P}}_1$ resp.

Rel: If $P[\Phi] \xrightarrow{a} P'[\Phi]$, then $P \xrightarrow{c} P'$ for some $c \in \Phi^{-1}(a)$, hence by ind. $P' \in \tilde{\mathbb{P}}$, thus $P'[\Phi] \in \tilde{\mathbb{P}}$. Furthermore, by ind.: $\mathcal{R}(b, P[\Phi]) = \min\{\mathcal{R}(c,P) \mid c \in \Phi^{-1}(b)\} \leq \min\{\mathcal{R}(c,P') \mid c \in \Phi^{-1}(b)\} = \mathcal{R}(b, P'[\Phi])$. Additionally, by ind., $P'[\Phi] \in \tilde{\mathbb{P}}_1$ if $P \in \tilde{\mathbb{P}}_1$.

Rec: If $\mu X.P \xrightarrow{a} P'$ then $P' \equiv P''\{\mu X.P/X\}$ and $P \xrightarrow{a} P''$ for some P''. Now by ind. $P'' \in \tilde{\mathbb{P}}_1$ since $P \in \tilde{\mathbb{P}}_1$ and $P''\{\mu X.P/X\} \in \tilde{\mathbb{P}}_1$ since $\mu X.P \in \tilde{\mathbb{P}}_1$, hence $P' \in \tilde{\mathbb{P}}_1 \subseteq \tilde{\mathbb{P}}$. Since $\mu X.P, P' \in \tilde{\mathbb{P}}_1$, we have furthermore for all $b \in \mathbb{A}_{\omega\tau}$: $\mathcal{R}(b, \mu X.P) = \mathcal{R}(b, P') = 1$.

3. Induction on the structure of P (the additional property for $\rightarrow_d$ now follows with Definition 2.11):

Clear for $\mathbf{0}$ and $X \in \mathcal{X}$.

Pref: $\langle a,1\rangle.P \overset{1}{\leadsto}_c \langle a,0\rangle.P \in \tilde{\mathbb{P}}_0$ and $\langle a,0\rangle.P \overset{1}{\leadsto}_c \langle a,0\rangle.P \in \tilde{\mathbb{P}}_0$, since $P \in \tilde{\mathbb{P}}_1$.

Sum: If $P_1 + P_2 \overset{1}{\leadsto}_c P_1' + P_2'$ then $P_1 \overset{1}{\leadsto}_c P_1'$ and $P_2 \overset{1}{\leadsto}_c P_2'$ where $P_1, P_2 \in \tilde{\mathbb{P}}_0 \cup \tilde{\mathbb{P}}_1 \subseteq \tilde{\mathbb{P}}$, hence by ind. $P_1', P_2' \in \tilde{\mathbb{P}}_0$, thus $P_1' + P_2' \in \tilde{\mathbb{P}}_0$.

Par, Rel: similarly to Sum.

Rec: If $\mu X.P \overset{1}{\leadsto}_c P'$ then $P' \equiv P''\{\mu X.P/X\}$ and $P \overset{1}{\leadsto}_c P''$ for some P''. Now $P'' \in \tilde{\mathbb{P}}_0$ by ind. and by Lemma 2.7.1 X is guarded in P''; furthermore, $\mu X.P \in \tilde{\mathbb{P}}_1$ and for any subterm $\langle a,r\rangle.Q$ of P'' we have $Q \in \tilde{\mathbb{P}}_1$, hence $Q\{\mu X.P/X\} \in \tilde{\mathbb{P}}_1$, and thus $P' \equiv P''\{\mu X.P/X\} \in \tilde{\mathbb{P}}_0$.

4. Induction on the structure of P:

Nil: $\mathbf{0} \in \tilde{\mathbb{P}}_0$ and $P_1 \equiv \mathbf{0} \xrightarrow{1}_d \mathbf{0}$, thus $w = \lambda$.

Var: If $P \equiv X \in \mathcal{X}$, then $P_1 \equiv \langle\tau, 1\rangle.X \in \tilde{\mathbb{P}}_1$ and $P_1 \xrightarrow{1}_d \langle\tau, 0\rangle.X \xrightarrow{\tau}_d X$.

Pref: For $P \equiv \langle a, 1\rangle.Q \in \tilde{\mathbb{P}}_1$ we may choose $P_1 \equiv \langle\tau, 1\rangle.\langle a, 1\rangle.Q \in \tilde{\mathbb{P}}_1$ with $P_1 \xrightarrow{1}_d \langle\tau, 0\rangle.\langle a, 1\rangle.Q \xrightarrow{\tau}_d P$.
For $P \equiv \langle a, 0\rangle.Q \in \tilde{\mathbb{P}}_0$ we have $Q \in \tilde{\mathbb{P}}_1$, hence we may choose $P_1 \equiv \langle a, 1\rangle.Q \in \tilde{\mathbb{P}}_1$ with $P_1 \xrightarrow{1}_d P$, thus $w = \lambda$.

Sum: If $P \equiv Q+R \in \tilde{\mathbb{P}}_1 \backslash \tilde{\mathbb{P}}_0$, then $P_1 \equiv \langle\tau, 1\rangle.(Q+R) \in \tilde{\mathbb{P}}_1$ with $P_1 \xrightarrow{1}_d \langle\tau, 0\rangle.(Q+R) \xrightarrow{\tau}_d P$.
If $P \equiv Q + R$ and $Q, R \in \tilde{\mathbb{P}}_0$, then there are by ind. and the additional property $Q_1, R_1 \in \tilde{\mathbb{P}}_1$ with $Q_1 \xrightarrow{1}_d Q$ and $R_1 \xrightarrow{1}_d R$; hence $P_1 \equiv Q_1 + R_1 \in \tilde{\mathbb{P}}_1$ and $P_1 \xrightarrow{1}_d P$ by rule Sum$_c$, thus $w = \lambda$.

Par: For $P \equiv Q\|_A R$ with $Q, R \in \tilde{\mathbb{P}}$ there are by ind. $Q_1, R_1 \in \tilde{\mathbb{P}}_1$, and $u, v \in \{\tau\}^*$, such that $Q_1 \xrightarrow{1}_d Q' \xrightarrow{u}_d Q$ and $R_1 \xrightarrow{1}_d R' \xrightarrow{v}_d R$. Since $Q_1, R_1 \in \tilde{\mathbb{P}}_1$, we have also $(Q_1\|_A R_1) \in \tilde{\mathbb{P}}_1$, hence $P_1 \equiv Q_1\|_A R_1 \xrightarrow{1}_d Q'\|_A R'$ by rule Par$_c$. Now by iterated application of rule Par$_{a1}$, $Q'\|_A R' \xrightarrow{u}_d Q\|_A R' \xrightarrow{v}_d Q\|_A R$, hence $P_1 \xrightarrow{1uv}_d P$. If $P \in \tilde{\mathbb{P}}_0$, then also $Q, R \in \tilde{\mathbb{P}}_0$, hence by induction $u = v = \lambda$, thus $w = uv = \lambda$.

Rel: For $P \equiv Q[\Phi]$ there are by ind. $Q_1 \in \tilde{\mathbb{P}}_1$ and $w \in \{\tau\}^*$ such that $Q_1 \xrightarrow{1}_d Q' \xrightarrow{w}_d Q$. Since $Q_1 \in \tilde{\mathbb{P}}_1$ we have $P_1 \equiv Q_1[\Phi] \xrightarrow{1}_d Q'[\Phi]$ with 1. and rule Rel$_c$. Now by iterated application of rule Rel$_a$ exploiting the condition $\Phi(\tau) = \tau$ on general relabelling functions, we get $Q'[\Phi] \xrightarrow{w}_d Q[\Phi]$, hence $P_1 \xrightarrow{1w}_d P$. If $P \in \tilde{\mathbb{P}}_0$, then also $Q \in \tilde{\mathbb{P}}_0$, hence $w = \lambda$ by induction.

Rec: $\mu X.P \in \tilde{\mathbb{P}}_1$, hence $P_1 \equiv \langle\tau, 1\rangle.\mu X.P \in \tilde{\mathbb{P}}_1$ and $P_1 \xrightarrow{1}_d \langle\tau, 0\rangle.\mu X.P \xrightarrow{\tau}_d \mu X.P$.

5. If $\mathcal{R}(a, P) = 1$ we may choose $P' \equiv P$ and $w = \lambda$ and are done. Hence assume (by 1.) in the following $\mathcal{R}(a, P) = 0$, in particular P is not $\mathbf{0}$ or $X \in \mathcal{X}$ or $\mu X.Q$. We perform induction on the structure of P:

Pref: $\langle a, 0\rangle.Q \xrightarrow{a}_d Q \in \tilde{\mathbb{P}}_1 \subseteq \tilde{\mathbb{P}}$, hence we may choose $w = a$.

Sum: For $P \equiv Q + R$ there are by ind. $Q', R' \in \tilde{\mathbb{P}}$ and $u, v \in \{a\}^*$, such that $Q \xrightarrow{u}_d Q'$ and $R \xrightarrow{v}_d R'$ with $\mathcal{R}(a, Q') = \mathcal{R}(a, R') = 1$ and by assumption $uv \neq \lambda$. If w.l.o.g. $u \neq \lambda$, then $P \equiv Q + R \xrightarrow{u}_d Q' \equiv P'$ with $\mathcal{R}(a, P') = 1$ and $w = u$.

Par: For $P \equiv Q\|_A R$ there are by ind. $Q', R' \in \tilde{\mathbb{P}}$ and $u, v \in \{a\}^*$, such that $Q \xrightarrow{u}_d Q'$ and $R \xrightarrow{v}_d R'$ with $\mathcal{R}(a, Q') = \mathcal{R}(a, R') = 1$.
If $a \notin A$, then by iterated application of rule Par$_{a1}$, $P \equiv Q\|_A R \xrightarrow{u}_d Q'\|_A R \xrightarrow{v}_d Q'\|_A R' \equiv P'$ with $\mathcal{R}(a, P') = 1$ and we may choose $w = uv$.
If $a \in A$, let w.l.o.g. $|u| \leq |v|$; then by iterated application of rule Par$_{a2}$, $P \equiv Q\|_A R \xrightarrow{u}_d Q'\|_A R'' \equiv P'$ with $\mathcal{R}(a, P') = 1$ since $\mathcal{R}(a, Q') = 1$, and we may choose $w = u$.

Rel: Let $P \equiv Q_0[\Phi]$ and $B = \Phi^{-1}(a) \cap \mathcal{A}(Q_0)$; $B = \{b_1, \ldots, b_n\}$ is finite since $\mathcal{A}(Q_0)$ is finite. By induction, for $i = 1, \ldots, n$, there are $Q_i \in \tilde{\mathbb{P}}$ and $v_i \in \{b_i\}^*$ such that $Q_{i-1} \xrightarrow{v_i}_d Q_i$ and $\forall_{1 \leq j \leq i}\, \mathcal{R}(b_j, Q_i) = 1$ by 2., hence $\forall_{b \in B}\, \mathcal{R}(b, Q_n) = 1$. Furthermore, 2. and Proposition 2.9.1 imply $\forall_{b \in \Phi^{-1}(a) \backslash B}\, \mathcal{R}(b, Q_0) = 1 = \mathcal{R}(b, Q_n)$. Hence for $w \in \{a\}^*$ with $|w| = \sum_{i=1}^n |v_i|$ we have $P \equiv Q_0[\Phi] \xrightarrow{w}_d Q_n[\Phi] \equiv P'$ with $\mathcal{R}(a, P') = \min\{\mathcal{R}(b, Q_n) \mid b \in \Phi^{-1}(a)\} = 1$. ■ 3.3

Proposition 3.3 states technical properties of (discrete) process terms and discrete behaviour; they are crucial in the proofs of many further developments and are gathered in a more readable manner in Corollary 3.4 below.

First, in 3.3.1, we ascertain that the residual time of (actions in) process terms is always either 0 or 1, reflecting that process terms can perform either no time a step or a unit time step. Properties 2. and 3. ensure that discrete behaviour of a process term yields a process term again, validating the match between the operational Definition 3.1 of discrete behaviour and the syntactical Definition 3.2 of discrete process terms. Additionally, occurrence of actions can only increase the residual time of (actions in) a process term. Properties 4. and 5. are of rather technical nature, but their statements are of intuitive interest, too: any process term is reachable from an initial process term by a $\stackrel{1}{\Rightarrow}_d$ step only, and repetition of a single action will eventually yield a process term, in which this action is not urgent any more. This will be important when characterizing the testing preorder in the next section.

Corollary 3.4

1. The set $\tilde{\mathbb{P}}$ contains exactly those process terms that are reachable from some initial process term $P \in \tilde{\mathbb{P}}_1$ by only discrete behaviour.
2. The set $\tilde{\mathbb{P}}_0$ contains exactly those process terms that are reachable from some initial process term $P \in \tilde{\mathbb{P}}_1$ by performing only a 1-time-step.
3. For each process term $P \in \tilde{\mathbb{P}}$ there is a discretely reachable successor $P' \in \tilde{\mathbb{P}}$ with $\mathcal{R}(P') = 1$.
4. All three results above hold true for respective processes either.
5. Let $P \in \mathbb{P}$ be a process and $P \stackrel{w}{\rightarrow}_d P'$ for some $w \in \mathsf{DL}(P)$. Then for each $R \in \mathbb{R}_0^+$ there is a $w' \in \mathsf{DL}(P')$ (hence $ww' \in \mathsf{DL}(P)$), such that $\zeta(ww') > R$.

Proof:

1. By Proposition 3.3.2, .3 and .4.

2. By Proposition 3.3.3 and .4.

3. By iterated application of Proposition 3.3.5, since $\mathcal{A}(P)$ is finite.

4. $\tilde{\mathbb{P}}$ is closed under $\rightarrow_d$ and $\mathbb{P}_c$ is closed under $\rightarrow_c$, hence under $\rightarrow_d$, thus $\mathbb{P} = \tilde{\mathbb{P}} \cap \mathbb{P}_c$ is closed under $\rightarrow_d$.

5. Let $P \stackrel{w}{\rightarrow}_d P'$ and $R \in \mathbb{R}_0^+$. Then $P' \stackrel{w'}{\rightarrow}_d P''$ for some P'' with $\mathsf{RT}(P'') = 1$ by 3. and 4., hence $P'' \stackrel{1}{\rightarrow}_d P'''$ for some P''' by Proposition 2.12.1, thus $ww'1 \in \mathsf{DL}(P)$ with $\zeta(ww'1) = \zeta(w) + 1$. Now either $\zeta(ww') > R$, or we proceed analogously with P'''. ■ 3.4

From Corollary 3.4.1 follows that the set of processes is closed under discrete behaviour, i.e. $P \in \mathbb{P}$ and $P \stackrel{w}{\rightarrow}_d P'$ for some $w \in \mathsf{DL}_\tau(P)$ implies $P' \in \mathbb{P}$ again. Furthermore, Corollary 3.4.4 states that at least discrete behaviour never yields a time stop. Theorem 3.9 will indicate that this is sufficient also for our definition of c-timed tests to make sense.

So far, we only know that discrete behaviour of an initial process is part of its continuous behaviour, viz $\mathsf{DL}(P) \subseteq \mathsf{CL}(P)$. We now aim to show that discrete behaviour already contains enough information for checking $P \sqsupseteq_c Q$ for testable P and Q. For this purpose, we will map each continuous trace of an initial (c-)process to a discrete trace of the same process. Related traces will exhibit the same behaviour, but at different points in time. We first relate the intermediate c-processes reached when performing such traces.

Definition 3.5 *progress preorder of c-process terms*

The *progress preorder* is the least ternary relation $\succ_\delta \subseteq (\tilde{\mathbb{P}}_c \times \mathbb{T} \times \tilde{\mathbb{P}}_c)$ satisfying:

1. Nil: $\mathbf{0} \succ_\delta \mathbf{0}$ for all $\delta \in \mathbb{T}$
2. Var: $X \succ_\delta X$ for all $\delta \in \mathbb{T}$
3. Pref: $\langle a, r_1\rangle.P \succ_\delta \langle a, r_2\rangle.P$ if $r_2 - r_1 \leq \delta$
4. Sum: $P_1 + P_2 \succ_\delta Q_1 + Q_2$ if $\forall_{i=1,2}\ P_i \succ_\delta Q_i$
5. Par: $P_1 \|_A P_2 \succ_\delta Q_1 \|_A Q_2$ if $\forall_{i=1,2}\ P_i \succ_\delta Q_i$
6. Rel: $P[\Phi] \succ_\delta Q[\Phi]$ if $P \succ_\delta Q$
7. Rec: a) $\mu X.P \succ_\delta \mu X.P$
   b) $P'\{\mu X.P/X\} \succ_\delta \mu X.P$ if $P' \succ_\delta P$
   c) $\mu X.P \succ_\delta P'\{\mu X.P/X\}$ if $P \succ_\delta P'$ ■ 3.5

Intuitively, $P \succ_\delta Q$ means that P and Q are essentially identical up to the values of timers, and if P is ahead of Q, then for at most time δ. However, Q may be ahead of P for an arbitrary amount of time, which is realized locally in the Pref-case, where we allow $r_2 < r_1$.

In cases Rec b) and c), $\mu X.P$ and $P'\{\mu X.P/X\}$ are regarded as structurally identical in two specific situations; this is necessary to make Proposition 3.6.4.a) below true: if $P \equiv \mu X.R \succ_\delta \mu X.R \equiv Q$ and Q makes a time step, then only for Q recursion is unfolded by rule Rec_c. An alternative would have been permitting 0 time steps in the discrete behaviour, which we have reprobated for the sake of compactness, since they only alter recursive terms and blow up discrete traces unnecessarily.

Proposition 3.6

For c-process terms $P, Q, R \in \tilde{\mathbb{P}}_c$ and $\delta, \delta' \in \mathbb{T}$ let $P \succ_\delta Q$. Furthermore, let $a \in \mathbb{A}_{\omega\tau}$, $X \in \mathcal{X}$ and $\rho, \rho_1, \rho_2 \in \mathbb{T}$.

1. $P \succ_0 P$ for all $P \in \tilde{\mathbb{P}}_c$.
2. X is guarded in P iff X is guarded in Q.
3. $\mathcal{A}(P) = \mathcal{A}(Q)$.
4. $\mathcal{R}(a, Q) - \mathcal{R}(a, P) \leq \delta$, in particular $\mathcal{R}(Q) - \mathcal{R}(P) \leq \delta$.
5. If $\delta \leq \delta'$, then $P \succ_{\delta'} Q$.
6. $P\{R/X\} \succ_\delta Q\{R/X\}$.
7. If $P \xrightarrow{a} P'$, then there exists Q' such that $Q \xrightarrow{a} Q'$ and $P' \succ_\delta Q'$, and vice versa.
8. a) If $P \xrightarrow{\rho}_c P'$ and $\delta + \rho \leq 1$, then $P' \succ_{\delta+\rho} Q$.
   b) If $Q \xrightarrow{\rho}_c Q'$ and $0 \leq \delta - \rho$, then $P \succ_{\delta-\rho} Q'$.
   c) If $P \xrightarrow{\rho_1}_c P'$, $Q \xrightarrow{\rho_2}_c Q'$ and $0 \leq \delta + \rho_1 - \rho_2 \leq 1$, then $P' \succ_{\delta+\rho_1-\rho_2} Q'$.
9. In all three cases of 8., $\rightarrow_c$ may be replaced by $\rightsquigarrow_c$.

Proof:

1. Induction on the structure of P.

2. Induction on the inference of $P \succ_\delta Q$; only the Rec-cases (7.) are non-trivial:
a) Clear.
b) X guarded in $\mu Y.P$ iff X guarded in P iff (ind.) X guarded in P' iff X guarded in $P'\{\mu Y.P/Y\}$. (For the last 'iff', observe for '$\Rightarrow$' that Y is guarded in P, hence in P' by ind., so all occurrences of X in $\mu Y.P$ are guarded in $P'\{\mu Y.P/Y\}$. Observe for '$\Leftarrow$' and $X \equiv Y$ that again Y is guarded in P, hence in P' by ind.)
c) Analogously to b).

3. Induction on the inference of $P \succ_\delta Q$; in the Rec-cases observe that X is guarded in P and apply 2. and Proposition 2.5.2.

4. We perform induction on the inference of $P \succ_\delta Q$:

If $P \equiv Q \equiv \mathbf{0}$ or $P \equiv Q \equiv X$, then $\forall_{a \in \mathbb{A}_{\omega\tau}}\ \mathcal{R}(a,Q) - \mathcal{R}(a,P) = 1 - 1 \leq \delta$, since $\mathcal{A}(P) = \mathcal{A}(Q) = \emptyset$. Now:

Pref: $\langle a, r_1\rangle.P \succ_\delta \langle a, r_2\rangle.P \Rightarrow r_2 - r_1 \leq \delta \Rightarrow \forall_{a' \in \mathbb{A}_{\omega\tau}}\ \mathcal{R}(a', \langle a, r_2\rangle.P) - \mathcal{R}(a', \langle a, r_1\rangle.P) = r_2 - r_1 \leq \delta \ \vee\ \mathcal{R}(a', \langle a, r_2\rangle.P) - \mathcal{R}(a', \langle a, r_1\rangle.P) = 1 - 1 \leq \delta$.

Sum: $P_1 + P_2 \succ_\delta Q_1 + Q_2 \Rightarrow \forall_{i=1,2}\ P_i \succ_\delta Q_i \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}} \forall_{i=1,2}\ \mathcal{R}(a, Q_i) - \mathcal{R}(a, P_i) \leq \delta \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}}\ \mathcal{R}(a, Q_1 + Q_2) - \mathcal{R}(a, P_1 + P_2) = \min_{i=1,2} \mathcal{R}(a, Q_i) - \min_{i=1,2} \mathcal{R}(a, P_i) \leq \delta$.

Par: $P_1 \|_A P_2 \succ_\delta Q_1 \|_A Q_2 \Rightarrow \forall_{i=1,2}\ P_i \succ_\delta Q_i \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}} \forall_{i=1,2}\ \mathcal{R}(a, Q_i) - \mathcal{R}(a, P_i) \leq \delta \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}} \max_{i=1,2} \mathcal{R}(a, Q_i) - \max_{i=1,2} \mathcal{R}(a, P_i) \leq \delta \wedge \min_{i=1,2} \mathcal{R}(a, Q_i) - \min_{i=1,2} \mathcal{R}(a, P_i) \leq \delta \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}} \mathcal{R}(a, Q_1 \|_A Q_2) - \mathcal{R}(a, P_1 \|_A P_2) \leq \delta$.

Rel: $P[\Phi] \succ_\delta Q[\Phi] \Rightarrow P \succ_\delta Q \Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}}\ \mathcal{R}(a, Q) - \mathcal{R}(a, P) \leq \delta$
$\Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}}\ \min_{b \in \Phi^{-1}(a)} \mathcal{R}(b, Q) - \min_{b \in \Phi^{-1}(a)} \mathcal{R}(b, P) \leq \delta$
$\Rightarrow \forall_{a \in \mathbb{A}_{\omega\tau}}\ \mathcal{R}(a, Q[\Phi]) - \mathcal{R}(a, P[\Phi]) \leq \delta$.

Rec: a) $\mathcal{R}(\mu X.P) - \mathcal{R}(\mu X.P) = 0 \leq \delta$.
b) Since X is guarded in P, it is guarded in P' by 2., hence $\mathcal{R}(a, P'\{\mu X.P/X\}) = \mathcal{R}(a, P')$ by Proposition 2.9.2. Now $\mathcal{R}(a, \mu X.P) - \mathcal{R}(a, P'\{\mu X.P/X\}) = \mathcal{R}(a, P) - \mathcal{R}(a, P') \leq \delta$ by ind.
c) Analogously to b).

For the additional property we can either choose $a \in \mathcal{A}(P) = \mathcal{A}(Q)$ with $\mathcal{R}(P) = \mathcal{R}(a, P)$ and get $\mathcal{R}(Q) - \mathcal{R}(P) \leq \mathcal{R}(a, Q) - \mathcal{R}(a, P) \leq \delta$, or we have $\mathcal{A}(P) = \mathcal{A}(Q) = \emptyset$ and by Proposition 2.9.1: $\mathcal{R}(Q) - \mathcal{R}(P) = 1 - 1 \leq \delta$.

5. Induction on the inference of $P \succ_\delta Q$; in particular, $r_2 - r_1 \leq \delta \leq \delta'$ in Definition 3.5.3.

6. Induction on the inference of $P \succ_\delta Q$; the case $P \equiv Q$ is covered by 1. and 5., and covers Nil, Var and Rec a).

Pref: $\alpha.P \succ_\delta \beta.P$ depends on α, β and δ only, hence $(\alpha.P)\{R/X\} \equiv \alpha.(P\{R/X\}) \succ_\delta \beta.(P\{R/X\}) \equiv (\beta.P)\{R/X\}$.

Sum, Par, Rel: straightforward induction by 'distributivity' of substitution.

Rec: b) Y is not free in $P'\{\mu Y.P/Y\}$ and $\mu Y.P$, hence assume $X \not\equiv Y$; then by Barendregt convention $P'\{\mu Y.P/Y\}\{R/X\} \equiv P'\{R/X\}\{\mu Y.P\{R/X\}/Y\}$. Now $P' \succ_\delta P$ implies $P'\{R/X\} \succ_\delta P\{R/X\}$ by ind., hence $(P'\{R/X\})\{\mu Y.P\{R/X\}/Y\} \succ_\delta \mu Y.(P\{R/X\}) \equiv (\mu Y.P)\{R/X\}$.
c) Analogously to b).

7. Induction on the inference of $P \xrightarrow{a} P'$:
Nil: $\mathbf{0} \xrightarrow{a}$ for no $a \in \mathbb{A}_{\omega\tau}$.
Var: $X \xrightarrow{a}$ for no $a \in \mathbb{A}_{\omega\tau}$ and no $X \in \mathcal{X}$.
Pref: Let $\langle a, r_1\rangle.P \succ_\delta \langle a, r_2\rangle.P$; then $\langle a, r_1\rangle.P \xrightarrow{a} P$ and $\langle a, r_2\rangle.P \xrightarrow{a} P$ and $P \succ_0 P$ by 1., hence $P \succ_\delta P$ by 5.
Sum, Par, Rel: straightforward induction.
Rec: a) Clear.
b) X is guarded in P and also in P' by 2.

On the one hand, $P'\{\mu X.P/X\} \xrightarrow{a} R$ only if (by Proposition 2.5.2)
$\exists P'' : P' \xrightarrow{a} P'' \wedge R \equiv P''\{\mu X.P/X\}$ only if (by ind.)
$\exists P'', P''' : P \xrightarrow{a} P''' \wedge P'' \succ_\delta P''' \wedge R \equiv P''\{\mu X.P/X\}$
only if (by Rule Rec_a and 6.)
$\exists P'', P''' : \mu X.P \xrightarrow{a} P'''\{\mu X.P/X\} \wedge R \equiv P''\{\mu X.P/X\} \succ_\delta P'''\{\mu X.P/X\}$.

On the other hand, $\mu X.P \xrightarrow{a} R$ only if (by rule Rec_a)
$\exists P''' : P \xrightarrow{a} P''' \wedge R \equiv P'''\{\mu X.P/X\}$ only if (by ind.)
$\exists P'', P''' : P' \xrightarrow{a} P'' \wedge P'' \succ_\delta P''' \wedge R \equiv P'''\{\mu X.P/X\}$
only if (by Proposition 2.5.2 and 6.)
$\exists P'', P''' : P'\{\mu X.P/X\} \xrightarrow{a} P''\{\mu X.P/X\} \succ_\delta R \equiv P'''\{\mu X.P/X\}$.

c) Analogously to b).

8.a) By Definition 2.11, it suffices to show $P \overset{\rho}{\rightsquigarrow}_c P' \wedge \delta + \rho \leq 1 \Rightarrow P' \succ_{\delta+\rho} Q$ by induction on the overall size of P *and* Q (where the size is the number of operators, also counting μX):

Clear for $\mathbf{0}$ and X.
Pref: $\langle a, r_1\rangle.P \succ_\delta \langle a, r_2\rangle.P \Rightarrow r_2 - r_1 \leq \delta$. We distinguish two cases:
i) $\rho \leq r_1$:
then $\langle a, r_1\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, r_1 - \rho\rangle.P$ and
$r_2 - r_1 \leq \delta \Rightarrow r_2 - (r_1 - \rho) = r_2 - r_1 + \rho \leq \delta + \rho \Rightarrow \langle a, r_1 - \rho\rangle.P \succ_{\delta+\rho} \langle a, r_2\rangle.P$.
ii) $\rho > r_1$:
then $\langle a, r_1\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, 0\rangle.P$ and
$r_2 - r_1 \leq \delta \Rightarrow r_2 \leq \delta + r_1 < \delta + \rho \Rightarrow r_2 - 0 \leq \delta + \rho \Rightarrow \langle a, 0\rangle.P \succ_{\delta+\rho} \langle a, r_2\rangle.P$.
Sum, Par, Rel: straightforward induction.
Rec: a) $\mu X.P \overset{\rho}{\rightsquigarrow}_c R$ due to $R \equiv P''\{\mu X.P/X\}$ and $P \overset{\rho}{\rightsquigarrow}_c P''$ for some P'' by rule Rec_c. Hence $P \succ_\delta P$ implies $P'' \succ_{\rho+\delta} P$ by ind. and thus $P''\{\mu X.P/X\} \succ_{\rho+\delta} \mu X.P$ by Definition 3.5.7.b).
b) $P'\{\mu X.P/X\} \overset{\rho}{\rightsquigarrow}_c R$ implies by 2. and Lemma 2.7.3 that $\exists P'' : P' \overset{\rho}{\rightsquigarrow}_c P''$ and $R \equiv P''\{\mu X.P/X\}$. Hence by ind. $P'' \succ_{\delta+\rho} P$, such that $R \succ_{\delta+\rho} \mu X.P$ by Definition 3.5.7.b). Note that P' has at most the size of $P'\{\mu X.P/X\}$, and the sizes might be equal if P' does not contain a free X; but in any case, P has a smaller size than $\mu X.P$ and, thus, induction is applicable.
c) Similar to a) with induction and 6.

8.b) Similar to 8.a). We only consider the Pref-case:
$\langle a, r_1\rangle.P \succ_\delta \langle a, r_2\rangle.P \Rightarrow r_2 - r_1 \leq \delta$. We distinguish two cases:
i) $\rho \leq r_2$:
then $\langle a, r_2\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, r_2 - \rho\rangle.P$ and

$r_2 - r_1 \leq \delta \Rightarrow (r_2 - \rho) - r_1 = r_2 - r_1 - \rho \leq \delta - \rho \Rightarrow \langle a, r_1\rangle.P \succ_{\delta-\rho} \langle a, r_2 - \rho\rangle.P$.

ii) $\rho > r_2$:

then $\langle a, r_2\rangle.P \overset{\rho}{\rightsquigarrow}_c \langle a, 0\rangle.P$ and

$0 - r_1 \leq 0 \leq \delta - \rho \Rightarrow \langle a, r_1\rangle.P \succ_{\delta-\rho} \langle a, 0\rangle.P$.

8.c) If $\delta + \rho_1 \leq 1$, we get $P' \succ_{\delta+\rho_1} Q$ by 8.a) and $P' \succ_{\delta+\rho_1-\rho_2} Q'$ by $0 \leq \delta + \rho_1 - \rho_2$ and 8.b). Otherwise, $P \overset{1-\delta}{\rightsquigarrow}_c P'' \overset{\rho}{\rightsquigarrow}_c P'$ with $\rho = \rho_1 + \delta - 1$ by Lemma 2.7.5. Now $P'' \succ_1 Q$ by 8.a), $P'' \succ_{1-\rho_2} Q'$ by 8.b) and $P' \succ_{\delta+\rho_1-\rho_2} Q'$ by 8.a) again.

9. By the proof of 8. ■ 3.6

Proposition 3.6 provides the elements for emulating each continuous trace of an initial process by a discrete trace that exhibits the same behaviour but consumes more time:

Lemma 3.7

Let $P \in \mathbb{P}_1$ be an initial process; then for each $w \in \mathsf{CL}(P)$ there is a $v \in \mathsf{DL}(P)$, such that $act(v) = act(w)$ and $\zeta(v) \geq \zeta(w)$.

Proof:

We will construct for each $w \in \mathsf{CL}_\tau(P)$ a $v \in \mathsf{DL}_\tau(P)$, such that $act(v) = act(w)$ and $\zeta(v) \geq \zeta(w)$; furthermore, we will show that for P_w and P_v reached after w and v we have $P_v \succ_{\zeta(v)-\zeta(w)} P_w$; by Corollary 3.4.4, this will imply $P_v \in \mathbb{P}$. Then $w/\tau \in \mathsf{CL}(P)$, $v/\tau \in \mathsf{DL}(P)$, $act(v/\tau) = act(w/\tau)$ and $\zeta(v/\tau) = \zeta(v) \geq \zeta(w) = \zeta(w/\tau)$.

The proof is by induction on $|w|$, where for $w = \lambda$ we can choose $v = \lambda$; then $P \succ_0 P$ by Proposition 3.6.1, hence $P_v \succ_{0-0} P_w$.

Hence, assume that for $w \in \mathsf{CL}_\tau(P)$ we have constructed $v \in \mathsf{DL}_\tau(P)$ as desired and consider $w' = w\varepsilon \in \mathsf{CL}_\tau(P)$. We denote the processes reached after w' and the corresponding v' by $P_{w'}$ and $P_{v'}$.

If $\varepsilon = a \in \mathbb{A}_\omega$, then $v' = va$ with $act(v') = act(w')$ and $\zeta(v') = \zeta(v) \geq \zeta(w) = \zeta(w')$. We have $P_w \overset{a}{\rightarrow} P_{w'}$ and by Proposition 3.6.7, there is a $P_{v'}$ such that $P_v \overset{a}{\rightarrow} P_{v'}$ and $P_{v'} \succ_{\zeta(v)-\zeta(w)} P_{w'}$, i.e. $P_{v'} \succ_{\zeta(v')-\zeta(w')} P_{w'}$.

Now let $\varepsilon = \rho \in \mathbb{T}$. If $\rho \leq \zeta(v) - \zeta(w)$ we choose $v' = v$; obviously, $act(v') = act(w')$, $\zeta(v') = \zeta(v) \geq \rho + \zeta(w) = \zeta(w')$. Furthermore, $\zeta(v') - \zeta(w') = \zeta(v) - \zeta(w) - \rho \geq 0$, hence $P_v \succ_{\zeta(v)-\zeta(w)} P_w$ and Proposition 3.6.8.b) yield $P_{v'} \equiv P_v \succ_{\zeta(v')-\zeta(w')} P_{w'}$.

If on the other hand $\rho > \zeta(v) - \zeta(w)$, we choose $v' = v1$. With Proposition 3.6.4, from $P_v \succ_{\zeta(v)-\zeta(w)} P_w$ we conclude $\mathcal{R}(P_v) + \zeta(v) - \zeta(w) \geq \mathcal{R}(P_w)$ and $\mathcal{R}(P_w) \geq \rho > \zeta(v) - \zeta(w)$ by Definition 2.11, i.e. $\mathcal{R}(P_v) > 0$ and $\mathcal{R}(P_v) = 1$ by Proposition 3.3.1. now by Proposition 3.6.2, P_v is guarded iff P_w is guarded, and P_w is guarded by Definition 2.11 and Lemma 2.7.1; hence by Proposition 2.12.1, the time step 1 is allowed after v and $v' = v1 \in \mathsf{DL}_\tau(P)$ with $act(v') = act(w')$. Furthermore, $\zeta(v') = \zeta(v) + 1 \geq \zeta(w) + \rho = \zeta(w')$, and finally, $\rho \leq 1$ and $0 \leq \zeta(v) - \zeta(w)$ give $0 \leq \zeta(v) - \zeta(w) + 1 - \rho$, and $\zeta(v) - \zeta(w) < \rho$ gives $\zeta(v) - \zeta(w) + 1 - \rho \leq 1$; so with Proposition 3.6.8.c) we conclude $P_{v'} \succ_{\zeta(v)-\zeta(w)+1-\rho} P_{w'}$, i.e. $P_{v'} \succ_{\zeta(v')-\zeta(w')} P_{w'}$. ■ 3.7

With this emulation result we can restrict attention to discretely timed testing based on discrete behaviour and discrete time bounds:

Definition 3.8 *discretely timed tests*

For a testable process $P \in \mathbb{P}_1$, an observer $O \in \mathbb{P}_1$ and $D \in \mathbb{N}_0$ define $P\ must_d\ (O, D)$, if each $w \in \mathsf{DL}(\tau.P\|_{\mathbb{A}}O)$, with $\zeta(w) > D$ contains some ω. The relation $\sqsupseteq_d$ is defined accordingly. ■ 3.8

We now give our first main result: although $\sqsupseteq_d$ is based on fewer tests and much more restricted behaviour than $\sqsupseteq_c$, it turns out that both relations define the same efficiency preorder. By this, we have also reached simplicity: we can now work with a CCS-like untimed algebra, extended syntactically by urgent terms (see Definition 3.2) and semantically by 1-time-steps.

Theorem 3.9

The relations $\sqsupseteq_c$ and $\sqsupseteq_d$ coincide. ■ 3.9

Proof:

Let P and Q be testable processes, O an observer and $R \in \mathbb{R}_0^+$. We first show

$$P\ must_c\ (O, R) \ \Leftrightarrow\ P\ must_d\ (O, \lfloor R \rfloor)$$

Assume $P\ \not{m}ust_c\ (O, R)$; then there is a $w \in \mathsf{CL}(\tau.P\|_{\mathbb{A}}O)$ without ω and $\zeta(w) > R$; now by Lemma 3.7, there is a $v \in \mathsf{DL}(\tau.P\|_{\mathbb{A}}O)$ without ω and $\zeta(v) \geq \zeta(w) > \lfloor R \rfloor$, hence $P\ \not{m}ust_d\ (O, \lfloor R \rfloor)$. Now assume $P\ \not{m}ust_d\ (O, \lfloor R \rfloor)$; then there is a $w \in \mathsf{DL}(\tau.P\|_{\mathbb{A}}O)$ without ω and $\zeta(w) > \lfloor R \rfloor$, hence $\zeta(w) \geq \lfloor R \rfloor + 1 > R$; since $\mathsf{DL}(\tau.P\|_{\mathbb{A}}O) \subseteq \mathsf{CL}(\tau.P\|_{\mathbb{A}}O)$, the same w causes $P\ \not{m}ust_c\ (O, R)$.

With this result we conclude $\forall (O, R) : Q\ must_c\ (O, R) \Rightarrow P\ must_c\ (O, R)$ iff $\forall (O, R)$: $Q\ must_d\ (O, \lfloor R \rfloor) \Rightarrow P\ must_d\ (O, \lfloor R \rfloor)$, hence $P \sqsupseteq_c Q$ iff $P \sqsupseteq_d Q$.

Checking $P \sqsupseteq_c Q$ now reduces to checking $P \sqsupseteq_d Q$. But as for testing in general, it is impossible to apply the definition of $\sqsupseteq_d$ directly, since there are still infinitely many discretely timed tests to apply. And as indicated in Section 2, we cannot decide $P \sqsupseteq_d Q$ from $\mathsf{DL}(P)$ and $\mathsf{DL}(Q)$ only, since $\mathsf{DL}(\tau.P\|_{\mathbb{A}}O)$ generally cannot be determined from $\mathsf{DL}(P)$ and $\mathsf{DL}(O)$ alone: e.g. synchronization allows activated actions in one component to wait for a partner in the other one, which is not the case in stand-alone behaviour of a single component, recorded in $\mathsf{DL}(P)$, $\mathsf{DL}(O)$ resp. Technically, DL-inclusion is not a precongruence for parallel composition. Thus, in the next section we will refine the discrete language to a kind of refusal traces, fulfilling the precongruence criterion. Refusal traces of a testable process will allow us to characterize the preorder $\sqsupseteq_d$ denotationally, where we also need the following result, stating that the number of different actions ever performable by a process is finite.

Definition 3.10 *semantic sort of a process*

For a c-process $P \in \mathbb{P}_c$ let $\ell_c(P) = \{a \in \mathbb{A}_{\omega\tau} \mid \exists w \in \mathsf{CL}_\tau(P),\ P' \in \mathbb{P}_c : P \xrightarrow{w}_c P' \xrightarrow{a}_c\}$ be the *continuous semantic sort* of P, and $\ell_d(P) = \{a \in \mathbb{A}_{\omega\tau} \mid \exists w \in \mathsf{DL}_\tau(P),\ P' \in \mathbb{P}_c :$ $P \xrightarrow{w}_d P' \xrightarrow{a}_d\} \subseteq \ell_c(P)$ be the *discrete semantic sort* of P. ■ 3.10

Proposition 3.11

Let $P \in \mathbb{P}_c$.

1. $\mathcal{A}(P) \subseteq \ell_d(P)$
2. For each $w \in \mathsf{CL}_\tau(P)$ there is a $v \in \mathsf{DL}_\tau(P)$ with $act(v) = act(w)$ and no time steps.
3. $\ell_c(P)$ and $\ell_d(P)$ coincide and will both be denoted by $\ell(P)$ (*semantic sort of* P).
4. $\ell(P)$ is finite.

Proof:

1. Clear.

2. We construct an adequate v performing induction on $|w|$ and show additionally that for P_w and P_v reached after w and v resp. we have $P_v \succ_1 P_w$.

By Proposition 3.6.1, $P \succ_0 P$, and by Proposition 3.6.5 $P \succ_1 P$, hence we are done for $w = \lambda$. Thus assume there is an adequate v for a given w and $P_v \succ_1 P_w$.

If $w' = wa$ for $a \in \mathbb{A}_{\omega\tau}$, then by Proposition 3.6.7, $P_w \overset{a}{\rightarrow} P_{w'}$ and $P_v \overset{a}{\rightarrow} P_{v'}$ for some $P_{v'}$ such that $P_{w'} \succ_1 P_{v'}$, hence we have $va \in \mathsf{DL}_\tau(P)$, and there is no time step in va since there is none in v.

If $w' = w\rho$ for $\rho \in \mathbb{T}$, then $P_w \overset{\rho}{\rightarrow}_c P_{w'}$ and by Proposition 3.6.8.b) and $0 \leq \rho \leq 1$, $P_v \succ_{1-\rho} P_{w'}$, hence by Proposition 3.6.5 $P_v \succ_1 P_{w'}$, thus we may choose $v' = v$.

3. $\ell_d \subseteq \ell_c$ by definition, and we are done by 2.

4. For a general relabelling function Φ let $ib(\Phi) = \{a \in \mathbb{A}_{\omega\tau} \,|\, \emptyset \neq \Phi^{-1}(a) \neq \{a\}\}$ (image base of Φ); by definition, $ib(\Phi)$ is finite. Furthermore, let
$\mathcal{L}(P) = \{a \in \mathbb{A}_{\omega\tau} \,|\, a \text{ occurs in } P\} \cup \bigcup_{\Phi \text{ occurs in } P} ib(\Phi)$
be the *syntactic sort* of P, where occurrence means being part of the syntatic structure of P. This definition yields $\mathcal{L}(P\{Q/X\}) \subseteq \mathcal{L}(P) \cup \mathcal{L}(Q)$, which will be used in the Rec-case below. Obviously $\mathcal{L}(P)$ is finite; we show $\ell(P) \subseteq \mathcal{L}(P)$ and are done.

By Proposition 2.5.2, $P \overset{a}{\rightarrow}$ iff $a \in \mathcal{A}(P)$, and by 1. and 2. it suffices to show by induction on the structure of P that $\mathcal{A}(P) \subseteq \mathcal{L}(P)$ and that $P \overset{a}{\rightarrow} P'$ implies $\mathcal{L}(P') \subseteq \mathcal{L}(P)$:

Clear for $\mathbf{0}$ and $X \in \mathcal{X}$.

Pref: $\mathcal{A}(a.P) = \{a\} \subseteq \{a\} \cup \mathcal{L}(P) = \mathcal{L}(a.P)$; furthermore, $a.P \overset{a}{\rightarrow} P$ and $\mathcal{L}(P) \subseteq \{a\} \cup \mathcal{L}(P) = \mathcal{L}(a.P)$.

Sum: By ind. $\mathcal{A}(P_1 + P_2) = \mathcal{A}(P_1) \cup \mathcal{A}(P_2) \subseteq \mathcal{L}(P_1) \cup \mathcal{L}(P_2) = \mathcal{L}(P_1 + P_2)$; furthermore, $P_1 + P_2 \overset{a}{\rightarrow} P'$ implies w.l.o.g. $P_1 \overset{a}{\rightarrow} P'$, hence by ind. $\mathcal{L}(P') \subseteq \mathcal{L}(P_1) \subseteq \mathcal{L}(P_1) \cup \mathcal{L}(P_2) = \mathcal{L}(P_1 + P_2)$.

Par: Analogously to Sum.

Rel: Let $a \in \mathcal{A}(P[\Phi]) = \Phi(\mathcal{A}(P))$; if $a \in ib(\Phi)$, then $a \in \mathcal{L}(P[\Phi])$, otherwise $\Phi^{-1}(a) = \{a\}$ and $a \in \mathcal{A}(P) \subseteq \mathcal{L}(P) \subseteq \mathcal{L}(P[\Phi])$ by induction and definition. Furthermore, $P[\Phi] \overset{a}{\rightarrow} P'[\Phi]$ implies $P \overset{b}{\rightarrow} P'$ for some $b \in \mathbb{A}_{\omega\tau}$ with $\Phi(b) = a$, hence by ind. $\mathcal{L}(P') \subseteq \mathcal{L}(P)$, thus $\mathcal{L}(P'[\Phi]) = \mathcal{L}(P') \cup ib(\Phi) \subseteq \mathcal{L}(P) \cup ib(\Phi) = \mathcal{L}(P[\Phi])$.

Rec: By ind. $\mathcal{A}(\mu X.P) = \mathcal{A}(P) \subseteq \mathcal{L}(P) = \mathcal{L}(\mu X.P)$.
Furthermore, $\mu X.P \overset{a}{\rightarrow} P'\{\mu X.P/X\}$ implies $P \overset{a}{\rightarrow} P'$, hence by ind. $\mathcal{L}(P') \subseteq \mathcal{L}(P)$, thus $\mathcal{L}(P'\{\mu X.P/X\}) \subseteq \mathcal{L}(P') \cup \mathcal{L}(\mu X.P) = \mathcal{L}(P)$. ∎ 3.11

4 Characterization

As a consequence of the last section, from now on we let $\sqsupseteq$ denote the (coinciding) preorders $\sqsupseteq_c$ and $\sqsupseteq_d$. Furthermore, we will merely deal with discrete processes and their discrete behaviour.

We first modify the SOS-rules for wait-time as follows: we only allow unit time steps and record at each time step a so-called *refusal set* Σ of actions which are *not* waiting; i.e. these actions are *not* urgent, they do not have to be performed and can be refused at this moment. Note that additionally and in contrast to passage of wait-time we now prohibit passage of time if there are urgent τ's. This time semantics is also a relaxation of (discrete) idle time: when a unit time step occurs, all actions in $\Sigma \cup \{\tau\}$ are treated correctly w.r.t. passage of idle time.

Definition 4.1 *SOS-rules for refusal of actions*

Via the following SOS-rules, a ternary relation $\rightarrow_r \subseteq (\tilde{\mathbb{P}} \times 2^{\mathbb{A}_\omega} \times \tilde{\mathbb{P}})$ is defined inductively, where $\Sigma, \Sigma_i \subseteq \mathbb{A}_\omega$:

$$\text{Nil}_r \ \frac{}{\mathbf{0} \xrightarrow{\Sigma}_r \mathbf{0}} \qquad \text{Pref}_{r1} \ \frac{}{a.P \xrightarrow{\Sigma}_r \underline{a}.P} \qquad \text{Pref}_{r2} \ \frac{a \notin \Sigma \cup \{\tau\}}{\underline{a}.P \xrightarrow{\Sigma}_r \underline{a}.P}$$

$$\text{Par}_r \ \frac{\forall_{i=1,2}\ P_i \xrightarrow{\Sigma_i}_r P_i',\ \Sigma \subseteq (A \cap \bigcup_{i=1,2} \Sigma_i) \cup ((\bigcap_{i=1,2} \Sigma_i) \setminus A)}{P_1 \|_A P_2 \xrightarrow{\Sigma}_r P_1' \|_A P_2'}$$

$$\text{Sum}_r \ \frac{\forall_{i=1,2}\ P_i \xrightarrow{\Sigma}_r P_i'}{P_1 + P_2 \xrightarrow{\Sigma}_r P_1' + P_2'} \qquad \text{Rel}_r \ \frac{P \xrightarrow{\Phi^{-1}(\Sigma \cup \{\tau\}) \setminus \{\tau\}}_r P'}{P[\Phi] \xrightarrow{\Sigma}_r P'[\Phi]} \qquad \text{Rec}_r \ \frac{P \xrightarrow{\Sigma}_r P'}{\mu X.P \xrightarrow{\Sigma}_r P'\{\mu X.P/X\}}$$

For process terms $P, P' \in \tilde{\mathbb{P}}$, we write $P \xrightarrow{\Sigma}_r P'$ if $(P, \Sigma, P') \in \rightarrow_r$ and call this a *time step*. We write $P \xrightarrow{\Sigma}_r$, if there exists a $P'' \in \tilde{\mathbb{P}}$ such that $(P, \Sigma, P'') \in \rightarrow_r$. ■ 4.1

By Proposition 4.2.1 below, the set of possible refusal sets at a time step is downward closed w.r.t. set inclusion, and by .3, not activated actions can always be refused. Proposition 4.2.4 provides the link between time steps and unit-time-waiting, unit-time-idling resp. Finally, Proposition 4.2.5 is an element needed in the treatment of recursion (Section 5), stating that guarded subterms of a process term are not affected by or involved in time steps or occurrence of actions.

Proposition 4.2

Let $P, Q, R \in \tilde{\mathbb{P}}$ be process terms, let $\Sigma, \Sigma' \subseteq \mathbb{A}_\omega$, let $X \in \mathcal{X}$ and let $\varepsilon \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})$.

1. If $P \xrightarrow{\Sigma}_r Q$ and $\Sigma' \subseteq \Sigma$, then $P \xrightarrow{\Sigma'}_r Q$.
2. If $P \xrightarrow{\Sigma}_r Q$ and $P \xrightarrow{\Sigma'}_r R$, then $Q \equiv R$.
3. If $P \xrightarrow{\Sigma}_r Q$ and $\Sigma' \cap \mathcal{A}(P) = \emptyset$, then $P \xrightarrow{\Sigma \cup \Sigma'}_r Q$.

4. $P \overset{\Sigma}{\rightarrow}_r Q$ if and only if $P \overset{1}{\rightsquigarrow}_c Q$ and $\forall_{a\in\Sigma\cup\{\tau\}}\ \mathcal{R}(a,P)=1$, in particular $P \overset{\mathbb{A}_\omega}{\rightarrow}_r Q$ if and only if $P \overset{1}{\rightarrow}_d Q$.
5. Let X be guarded in P. Then $P\{Q/X\} \overset{\varepsilon}{\rightarrow}_r R$ if and only if there exists $P' \in \tilde{\mathbb{P}}$ with $P \overset{\varepsilon}{\rightarrow}_r P'$ and $R \equiv P'\{Q/X\}$.

Proof:

1. Induction on the inference of $P \overset{\Sigma}{\rightarrow}_r$ using $\Sigma' \subseteq \Sigma$:

Nil: $\mathbf{0} \overset{\Sigma}{\rightarrow}_r \mathbf{0}$ and $\mathbf{0} \overset{\Sigma'}{\rightarrow}_r \mathbf{0}$ for all $\Sigma' \subseteq \Sigma \subseteq \mathbb{A}_\omega$.

Var: Clear.

Pref$_1$: analogously to Nil.

Pref$_2$: $\Sigma' \subseteq \Sigma \wedge \underline{a}.P \overset{\Sigma}{\rightarrow}_r \underline{a}.P \Rightarrow \Sigma' \subseteq \Sigma \wedge a \notin \Sigma\cup\{\tau\} \Rightarrow a \notin \Sigma'\cup\{\tau\} \Rightarrow \underline{a}.P \overset{\Sigma'}{\rightarrow}_r \underline{a}.P$.

Sum: $\Sigma' \subseteq \Sigma \wedge P_1 + P_2 \overset{\Sigma}{\rightarrow}_r Q_1 + Q_2 \Rightarrow \Sigma' \subseteq \Sigma \wedge P_1 \overset{\Sigma}{\rightarrow}_r Q_1 \wedge P_2 \overset{\Sigma}{\rightarrow}_r Q_2 \Rightarrow P_1 \overset{\Sigma'}{\rightarrow}_r Q_1 \wedge P_2 \overset{\Sigma'}{\rightarrow}_r Q_2 \Rightarrow P_1 + P_2 \overset{\Sigma'}{\rightarrow}_r Q_1 + Q_2$.

Par: $\Sigma' \subseteq \Sigma \wedge P_1\|_A P_2 \overset{\Sigma}{\rightarrow}_r Q_1\|_A Q_2 \Rightarrow \Sigma' \subseteq \Sigma \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i)\setminus A) \wedge \forall_{i=1,2}\ P_i \overset{\Sigma_i}{\rightarrow}_r Q_i \Rightarrow P_1\|_A P_2 \overset{\Sigma'}{\rightarrow}_r Q_1\|_A Q_2$.

Rel: $\Sigma' \subseteq \Sigma \wedge P[\Phi] \overset{\Sigma}{\rightarrow}_r Q[\Phi] \Rightarrow \Phi^{-1}(\Sigma'\cup\{\tau\})\setminus\{\tau\} \subseteq \Phi^{-1}(\Sigma\cup\{\tau\})\setminus\{\tau\} \wedge P \overset{\Phi^{-1}(\Sigma\cup\{\tau\})\setminus\{\tau\}}{\rightarrow}_r Q \Rightarrow P \overset{\Phi^{-1}(\Sigma'\cup\{\tau\})\setminus\{\tau\}}{\rightarrow}_r Q \Rightarrow P[\Phi] \overset{\Sigma'}{\rightarrow}_r Q[\Phi]$.

Rec: $\Sigma' \subseteq \Sigma \wedge \mu X.P \overset{\Sigma}{\rightarrow}_r Q \Rightarrow \exists P' : \Sigma' \subseteq \Sigma \wedge P \overset{\Sigma}{\rightarrow}_r P' \wedge Q \equiv P'\{\mu X.P/X\} \Rightarrow \exists P' : P \overset{\Sigma'}{\rightarrow}_r P' \wedge Q \equiv P'\{\mu X.P/X\} \Rightarrow \exists P' : \mu X.P \overset{\Sigma'}{\rightarrow}_r P'\{\mu X.P/X\} \equiv Q$.

2. Induction on the inference of $P \overset{\Sigma}{\rightarrow}_r Q$, $P \overset{\Sigma'}{\rightarrow}_r R$ resp.

3. Induction on the inference of $P \overset{\Sigma}{\rightarrow}_r Q$ using $\Sigma' \cap \mathcal{A}(P) = \emptyset$:

Nil: $\mathbf{0} \overset{\Sigma}{\rightarrow}_r \mathbf{0}$ and $\mathbf{0} \overset{\Sigma\cup\Sigma'}{\rightarrow}_r \mathbf{0}$ for all $\Sigma' \subseteq \mathbb{A}_\omega$.

Var: Clear.

Pref$_1$: analogously to Nil.

Pref$_2$: $\Sigma' \cap \mathcal{A}(\underline{a}.P) = \emptyset \wedge \underline{a}.P \overset{\Sigma}{\rightarrow}_r \underline{a}.P \Rightarrow \Sigma' \cap \{a\} = \emptyset \wedge (\Sigma\cup\{\tau\}) \cap \{a\} = \emptyset \Rightarrow ((\Sigma\cup\Sigma')\cup\{\tau\}) \cap \{a\} = \emptyset \Rightarrow \underline{a}.P \overset{\Sigma\cup\Sigma'}{\rightarrow}_r \underline{a}.P$.

Sum: $\Sigma' \cap \mathcal{A}(P_1 + P_2) = \emptyset \wedge P_1 + P_2 \overset{\Sigma}{\rightarrow}_r Q_1 + Q_2 \Rightarrow \Sigma' \cap \mathcal{A}(P_1) = \emptyset \wedge \Sigma' \cap \mathcal{A}(P_2) = \emptyset \wedge P_1 \overset{\Sigma}{\rightarrow}_r Q_1 \wedge P_2 \overset{\Sigma}{\rightarrow}_r Q_2 \Rightarrow P_1 \overset{\Sigma\cup\Sigma'}{\rightarrow}_r Q_1 \wedge P_2 \overset{\Sigma\cup\Sigma'}{\rightarrow}_r Q_2 \Rightarrow P_1 + P_2 \overset{\Sigma\cup\Sigma'}{\rightarrow}_r Q_1 + Q_2$.

Par: $P_1\|_A P_2 \overset{\Sigma}{\rightarrow}_r Q_1\|_A Q_2 \Rightarrow \Sigma \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i)\setminus A) \wedge \forall_{i=1,2}\ P_i \overset{\Sigma_i}{\rightarrow}_r Q_i$ and from $\Sigma' \cap \mathcal{A}(P_1\|_A P_2) = \emptyset$ we have to conlude $\Sigma' \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i)\setminus A)$. By induction, we may assume the Σ_i to be maximal w.r.t. to the considered property, i.e. we may assume $\forall_{a\in\mathbb{A}_\omega}\ a \notin \mathcal{A}(P_i) \Rightarrow a \in \Sigma_i$; now $\Sigma' \cap \mathcal{A}(P_1\|_A P_2) = \emptyset$ and $a \in \Sigma' \cap A$ implies $a \notin \mathcal{A}(P_1) \cap \mathcal{A}(P_2)$, hence $a \in \bigcup_{i=1,2}\Sigma_i$, and $a \in \Sigma'\setminus A$ implies $a \notin \mathcal{A}(P_1) \cup \mathcal{A}(P_2)$, thus $a \in \bigcap_{i=1,2}\Sigma_i$, yielding $\Sigma' \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i)\setminus A)$.

Rel: $\Sigma' \cap \mathcal{A}(P[\Phi]) = \emptyset \wedge P[\Phi] \overset{\Sigma}{\rightarrow}_r Q[\Phi] \Rightarrow P \overset{\Phi^{-1}(\Sigma\cup\{\tau\})\setminus\{\tau\}}{\rightarrow}_r Q \wedge \Sigma' \cap \Phi(\mathcal{A}(P)) = \emptyset$. Now $\Sigma' \cap \Phi(\mathcal{A}(P)) = \emptyset \Rightarrow \Phi^{-1}(\emptyset) = \emptyset = \Phi^{-1}(\Sigma' \cap \Phi(\mathcal{A}(P))) = \Phi^{-1}(\Sigma') \cap \Phi^{-1}(\Phi(\mathcal{A}(P))) \supseteq \Phi^{-1}(\Sigma') \cap \mathcal{A}(P) \Rightarrow \Phi^{-1}(\Sigma') \cap \mathcal{A}(P) = \emptyset \Rightarrow \Phi^{-1}(\Sigma')\setminus\{\tau\} \cap \mathcal{A}(P) = \emptyset$ and $\Phi^{-1}(\Sigma')\setminus\{\tau\} \cup \Phi^{-1}(\Sigma\cup\{\tau\})\setminus\{\tau\} = \Phi^{-1}((\Sigma\cup\Sigma')\cup\{\tau\})\setminus\{\tau\}$, hence by induction $P \overset{\Phi^{-1}((\Sigma\cup\Sigma')\cup\{\tau\})\setminus\{\tau\}}{\rightarrow}_r Q$, thus $P[\Phi] \overset{\Sigma\cup\Sigma'}{\rightarrow}_r Q[\Phi]$.

Rec: $\Sigma' \cap \mathcal{A}(\mu X.P) = \emptyset \wedge \mu X.P \overset{\Sigma}{\rightarrow}_r Q \Rightarrow \exists P' : \Sigma' \cap \mathcal{A}(P) = \emptyset \wedge P \overset{\Sigma}{\rightarrow}_r P' \wedge$

$Q \equiv P'\{\mu X.P/X\} \Rightarrow \exists P' : P \stackrel{\Sigma\cup\Sigma'}{\rightarrow_r} P' \wedge Q \equiv P'\{\mu X.P/X\} \Rightarrow \mu X.P \stackrel{\Sigma\cup\Sigma'}{\rightarrow_r} P'\{\mu X.P/X\} \equiv Q$.

4. Induction on the structure of P:

Nil: $\mathbf{0} \stackrel{\Sigma}{\rightarrow}_r \mathbf{0}$ and $\mathbf{0} \stackrel{1}{\rightsquigarrow}_c \mathbf{0}$ and $\forall_{a\in\Sigma\cup\{\tau\}}\mathcal{R}(a,\mathbf{0}) = 1$ for all $\Sigma \subseteq \mathbb{A}_\omega$.

Var: For all $X \in \mathcal{X}$ and $\Sigma \subseteq \mathbb{A}_\omega$, neither $X \stackrel{\Sigma}{\rightarrow}_r$ nor $X \stackrel{1}{\rightsquigarrow}_c$.

Pref: $a.P \stackrel{\Sigma}{\rightarrow}_r \underline{a}.P$ and $a.P \stackrel{1}{\rightsquigarrow}_c \underline{a}.P$ and $\forall_{a'\in\Sigma\cup\{\tau\}} \mathcal{R}(a', a.P) = 1$.
$\underline{a}.P \stackrel{\Sigma}{\rightarrow}_r \underline{a}.P$ iff $a \notin \Sigma \cup \{\tau\}$ iff $\underline{a}.P \stackrel{1}{\rightsquigarrow}_c \underline{a}.P \wedge \forall_{a'\in\Sigma\cup\{\tau\}} \mathcal{R}(a', \underline{a}.P) = 1$.

Sum: $P_1 + P_2 \stackrel{\Sigma}{\rightarrow}_r Q_1 + Q_2$ iff $\forall_{i=1,2}\, P_i \stackrel{\Sigma}{\rightarrow}_r Q_i$ iff
$\forall_{i=1,2}\,(P_i \stackrel{1}{\rightsquigarrow}_c Q_i \wedge \forall_{a\in\Sigma\cup\{\tau\}}\mathcal{R}(a,P_i) = 1)$ iff
$\forall_{i=1,2}\, P_i \stackrel{1}{\rightsquigarrow}_c Q_i \wedge \forall_{a\in\Sigma\cup\{\tau\}} \min_{i=1,2} \mathcal{R}(a,P_i) = 1$ iff
$P_1 + P_2 \stackrel{1}{\rightsquigarrow}_c Q_1 + Q_2 \wedge \forall_{a\in\Sigma\cup\{\tau\}}\mathcal{R}(a, P_1 + P_2) = 1$.

Par: $P_1\|_A P_2 \stackrel{\Sigma}{\rightarrow}_r Q_1\|_A Q_2$ iff
$\exists \Sigma_1, \Sigma_2 : \Sigma \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i) \setminus A) \wedge \forall_{i=1,2}\, P_i \stackrel{\Sigma_i}{\rightarrow}_r Q_i$ iff

$\exists \Sigma_1, \Sigma_2 : \forall_{i=1,2}\,(P_i \stackrel{1}{\rightsquigarrow}_c Q_i \wedge \forall_{a\in\Sigma_i\cup\{\tau\}}\mathcal{R}(a,P_i) = 1) \wedge$
$\Sigma \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i) \setminus A)$ iff

$P_1\|_A P_2 \stackrel{1}{\rightsquigarrow}_c Q_1\|_A Q_2 \wedge$
$\exists \Sigma_1 = \{a \in \mathbb{A}_\omega \mid \mathcal{R}(a,P_1) = 1\}, \Sigma_2 = \{a \in \mathbb{A}_\omega \mid \mathcal{R}(a,P_2) = 1\}$:
$\forall_{i=1,2}\forall_{a\in\Sigma_i\cup\{\tau\}}\mathcal{R}(a,P_i) = 1$
(i.e.: $\forall_{a\in\bigcup_{i=1,2}\Sigma_i\cup\{\tau\}} \max_{i=1,2}\mathcal{R}(a,P_i) = 1 \wedge \forall_{a\in\bigcap_{i=1,2}\Sigma_i\cup\{\tau\}} \min_{i=1,2}\mathcal{R}(a,P_i) = 1) \wedge$
$\Sigma \subseteq (A \cap \bigcup_{i=1,2}\Sigma_i) \cup ((\bigcap_{i=1,2}\Sigma_i) \setminus A)$ iff

$P_1\|_A P_2 \stackrel{1}{\rightsquigarrow}_c Q_1\|_A Q_2 \wedge$
$\forall_{a\in(\Sigma\cup\{\tau\})\cap A} \max_{i=1,2}\mathcal{R}(a,P_i) = 1 \wedge \forall_{a\in(\Sigma\cup\{\tau\})\setminus A} \min_{i=1,2}\mathcal{R}(a,P_i) = 1$ iff

$P_1\|_A P_2 \stackrel{1}{\rightsquigarrow}_c Q_1\|_A Q_2 \wedge \forall_{a\in\Sigma\cup\{\tau\}} \mathcal{R}(a, P_1\|_A P_2) = 1$.

Rel: $P[\Phi] \stackrel{\Sigma}{\rightarrow}_r Q[\Phi]$ iff $P \stackrel{\Phi^{-1}(\Sigma\cup\{\tau\})\setminus\{\tau\}}{\rightarrow_r} Q$ iff
$P \stackrel{1}{\rightsquigarrow}_c Q \wedge a \in \Phi^{-1}(\Sigma \cup \{\tau\}) \Rightarrow \mathcal{R}(a,P) = 1$ iff
$P[\Phi] \stackrel{1}{\rightsquigarrow}_c Q[\Phi] \wedge b \in \Sigma \cup \{\tau\} \Rightarrow \min_{a\in\Phi^{-1}(b)} \mathcal{R}(a,P) = 1$ iff
$P[\Phi] \stackrel{1}{\rightsquigarrow}_c Q[\Phi] \wedge \forall_{b\in\Sigma\cup\{\tau\}} \mathcal{R}(b, P[\Phi]) = 1$.

Rec: $\mu X.P \stackrel{\Sigma}{\rightarrow}_r P'\{\mu X.P/X\} \wedge P \stackrel{\Sigma}{\rightarrow}_r P'$ iff $P \stackrel{1}{\rightsquigarrow}_c P' \wedge \forall_{a\in\Sigma\cup\{\tau\}} \mathcal{R}(a,P) = 1$ iff $\mu X.P \stackrel{1}{\rightsquigarrow}_c P'\{\mu X.P/X\} \wedge P \stackrel{1}{\rightsquigarrow}_c P' \wedge \forall_{a\in\Sigma\cup\{\tau\}} \mathcal{R}(a, \mu X.P) = 1$.

The additional property follows with Definition 2.8 of $\mathcal{R}(P)$ and $\mathcal{A}(P) \subseteq \mathbb{A}_{\omega\tau}$ and Definition 2.11.

5. If $\varepsilon = a \in \mathbb{A}_{\omega\tau}$, we are done by Proposition 2.5.2, hence let $\varepsilon = \Sigma \subseteq \mathbb{A}_\omega$. Then $P\{Q/X\} \stackrel{\Sigma}{\rightarrow}_r R$ iff $P\{Q/X\} \stackrel{1}{\rightsquigarrow}_c R$ and $\forall_{a\in\Sigma\cup\{\tau\}}\mathcal{R}(a, P\{Q/X\}) = 1$ by 4. iff $P \stackrel{1}{\rightsquigarrow}_c P'$ for some P' such that $R \equiv P'\{Q/X\}$ and $\forall_{a\in\Sigma\cup\{\tau\}}\mathcal{R}(a,P) = 1$ by Lemma 2.7.3 and Proposition 2.9.2 iff $P \stackrel{\Sigma}{\rightarrow}_r P'$ and $R \equiv P'\{Q/X\}$ by 4. again. ■ 4.2

Combining time steps and occurrence of actions, we now define refusal traces of processes, which refine the discrete language due to Proposition 4.2.4 (part 2), as stated in Theorem 4.4.

Definition 4.3 *refusal traces of processes*

Let $P, P' \in \mathbb{P}$ be processes. We write $P \xrightarrow{\varepsilon}_r P'$, if either $\varepsilon = a \in \mathbb{A}_{\omega\tau}$ and $P \xrightarrow{a} P'$, or $\varepsilon = \Sigma \subseteq \mathbb{A}_\omega$ and $P \xrightarrow{\Sigma}_r P'$. For sequences w, we define $P \xrightarrow{w}_r P'$ and $P \xRightarrow{w}_r P'$ analogously to Definition 2.13.

For a process $P \in \mathbb{P}$, let $\mathsf{RT}_\tau(P) = \{w \mid P \xrightarrow{w}_r\}$ be the *τ-refusal traces* of P, and the set $\mathsf{RT}(P) = \{w \mid P \xRightarrow{w}_r\}$ be the *refusal traces* of P.

act(w) and $\zeta(w)$ are extended to elements from $\mathsf{RT}_\tau(P)$ and $\mathsf{RT}(P)$, i.e. $\zeta(w)$ is the number of time steps (sets) in w. ■ 4.3

Theorem 4.4

Let $P, Q \in \mathbb{P}$ be processes; then $\mathsf{RT}(P) \subseteq \mathsf{RT}(Q)$ implies $\mathsf{DL}(P) \subseteq \mathsf{DL}(Q)$.

Proof:

By Proposition 4.2.4, $P \xrightarrow{1}_d P'$ iff $P \xrightarrow{\mathbb{A}_\omega}_r P'$, hence $\mathsf{DL}(P)$ can be gained from those $w \in \mathsf{RT}(P)$ where $\Sigma = \mathbb{A}_\omega$ for all refusal sets Σ in w, replacing $\mathbb{A}_\omega$ by 1. ■ 4.4

From now on we denote refusal-trace-inclusion and -equivalence of processes by $\leq_r$, $=_r$ resp. and lift this relation to process terms as usual via closed substitutions:

Definition 4.5

Let $P, Q \in \tilde{\mathbb{P}}$ be process terms. We write $P \leq_r Q$ if for all closed substitutions $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}$ where $[P]_\mathcal{S}, [Q]_\mathcal{S} \in \mathbb{P}$ we have $\mathsf{RT}([P]_\mathcal{S}) \subseteq \mathsf{RT}([Q]_\mathcal{S})$. We write $P =_r Q$ if $P \leq_r Q$ and $Q \leq_r P$. ■ 4.5

As for discrete traces, we note that the set of processes is closed under performance of refusal traces, i.e. $P \in \mathbb{P}$ and $P \xrightarrow{w}_r P'$ for some $w \in \mathsf{RT}_\tau(P)$ implies $P' \in \mathbb{P}$ again. The information on temporal and nondeterministic behaviour of a process provided by refusal traces is very similar to the one e.g. contained in the 'barbs' of TPL (see [HR95]). But astonishingly, we will be able to observe this with asynchronous – i.e. weak – test processes.

For technical reasons, in the following we do not only consider the RT-semantics but also the RT_τ-semantics: it will play an important rôle when deriving the precongruence property of RT- and RT_τ-inclusion w.r.t. the recursion operator in Section 5. Note that $\mathsf{RT}_\tau(P)$ does not only treat τ's like visible actions: additionally, by Definition 4.1, all refusal sets Σ in a $w \in \mathsf{RT}_\tau(P)$ implicitly contain τ, i.e. in w after a time step an activated τ must either occur or be disabled before the next time step Σ.

The following developments are concerned with (pre)congruence properties of refusal-trace-equivalence (-inclusion). As indicated in Section 3, DL-inclusion is not a precongruence for parallel composition: it does not record runs of a component in which actions are delayed beyond idle time, which in general is necessary in a parallel composition when waiting for a communication partner. We first show that (τ-) refusal traces serve this purpose:

Definition 4.6 *shuffle of refusal traces w.r.t A*

Let $u, v \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ and $A \subseteq \mathbb{A}$; then $u\|_A v$ is the set of all $w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ such that for some n $u = u_1 \dots u_n$, $v = v_1 \dots v_n$, $w = w_1 \dots w_n$ and for all $k = 1, \dots, n$ one of the following cases applies:

1. $u_k = v_k = w_k = a \in A$
2. $u_k = w_k = a \in \mathbb{A}_\omega \setminus A$ and $v_k = \lambda$
3. $v_k = w_k = a \in \mathbb{A}_\omega \setminus A$ and $u_k = \lambda$
4. $u_k = \Sigma_u \subseteq \mathbb{A}_\omega$, $v_k = \Sigma_v \subseteq \mathbb{A}_\omega$, $w_k = \Sigma \subseteq \mathbb{A}_\omega$ and $\Sigma \subseteq (A \cap (\Sigma_u \cup \Sigma_v)) \cup ((\Sigma_u \cap \Sigma_v) \setminus A)$

For sets $R_1, R_2 \subseteq (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$, we define $R_1 \|_A R_2 = \bigcup \{u \|_A v \mid u \in R_1,\, v \in R_2\}$. ■ 4.6

Observe that if $(v\|_A u) \neq \emptyset$, then by 1. for all $a \in A$ the number of a's is equal in u, v and all w, and by 4. the number of time steps is equal in u, v and all w.

Theorem 4.7

For processes $P_1, P_2 \in \mathbb{P}$, we have $\mathsf{RT}(P_1\|_A P_2) = \mathsf{RT}(P_1)\|_A \mathsf{RT}(P_2)$ and $\mathsf{RT}_\tau(P_1\|_A P_2) = \mathsf{RT}_\tau(P_1)\|_A \mathsf{RT}_\tau(P_2)$. In particular, both RT-inclusion and RT_τ-inclusion are precongruences for parallel composition.

Proof:

It suffices to show the claim for RT-semantics; the same technique then applies for RT_τ-semantics, where τ's are treated like visible actions. Let $P \equiv P_1\|_A P_2$.

'$\subseteq$':

Let $v \in \mathsf{RT}(P)$. Then there is a $w \in \mathsf{RT}_\tau(P)$ such that $v = w/\tau$. We perform induction on the length of w and show that if $P \xrightarrow{w}_r P'$, then there are $w_1 \in \mathsf{RT}_\tau(P_1)$ and $w_2 \in \mathsf{RT}_\tau(P_2)$ such that $w/\tau \in ((w_1/\tau)\|_A(w_2/\tau)) \subseteq (\mathsf{RT}(P_1)\|_A \mathsf{RT}(P_2))$, and – furthermore – if $P_1 \xrightarrow{w_1}_r P_1'$ and $P_2 \xrightarrow{w_2}_r P_2'$, then $P' \equiv P_1'\|_A P_2'$.

For $w = \lambda$ we choose $w_1 = w_2 = \lambda$ such that $w/\tau = \lambda \in ((w_1/\tau)\|_A(w_2/\tau)) = \{\lambda\}$ and $P \equiv P' \equiv P_1\|_A P_2 \equiv P_1'\|_A P_2'$.

So let $w' = w\varepsilon$ and $P \xrightarrow{w}_r P' \xrightarrow{\varepsilon}_r P''$. Then one of the following cases applies:

1. $\varepsilon = a \in A \subseteq \mathbb{A}_\omega$. By ind. $P' \equiv P_1'\|_A P_2'$, and $P_1'\|_A P_2' \xrightarrow{a}_r P''$ iff by rule Par_{a2}: $P'' \equiv P_1''\|_A P_2''$, $P_1' \xrightarrow{a} P_1''$ and $P_2' \xrightarrow{a} P_2''$ for some P_1'', P_2''. Hence $P_1 \xrightarrow{w_1}_r P_1' \xrightarrow{a}_r P_1''$ and $P_2 \xrightarrow{w_2}_r P_2' \xrightarrow{a}_r P_2''$, thus by ind. and Definition 4.6.1 for $w_1' = w_1 a$ and $w_2' = w_2 a$: $w'/\tau = (wa)/\tau = (w/\tau)a \in (((w_1/\tau)a)\|_A((w_2/\tau)a)) = ((w_1 a)/\tau)\|_A((w_2 a)/\tau)) = ((w_1'/\tau)\|_A(w_2'/\tau))$.
2. $\varepsilon = a \in \mathbb{A}_{\omega\tau} \setminus A$. By ind. $P' \equiv P_1'\|_A P_2'$, and $P_1'\|_A P_2' \xrightarrow{a}_r P''$ iff (w.l.o.g.) by rule Par_{a1}: $P'' \equiv P_1''\|_A P_2'$ and $P_1' \xrightarrow{a} P_1''$ for some P_1''. Hence $P_1 \xrightarrow{w_1}_r P_1' \xrightarrow{a}_r P_1''$ and $P_2 \xrightarrow{w_2}_r P_2'$, thus we choose $w_1' = w_1 a$ and $w_2' = w_2$. If $a = \tau$, then $w'/\tau = w/\tau$ and $w_1'/\tau = w_1/\tau$, and by ind. $w/\tau \in ((w_1/\tau)\|_A(w_2/\tau))$. If $a \neq \tau$ we observe $w_2' = w_2\lambda$; then by ind. and Definition 4.6.2 (or .3 resp.): $w'/\tau = (wa)/\tau = (w/\tau)a \in (((w_1/\tau)a)\|_A((w_2/\tau)\lambda)) = (((w_1 a)/\tau)\|_A((w_2\lambda)/\tau)) = ((w_1'/\tau)\|_A(w_2'/\tau))$.
3. $\varepsilon = \Sigma \subseteq \mathbb{A}_\omega$. By ind. $P' \equiv P_1'\|_A P_2'$, and $P_1'\|_A P_2' \xrightarrow{\Sigma}_r P''$ iff by rule Par_r: $P'' \equiv P_1''\|_A P_2''$, $P_1' \xrightarrow{\Sigma_1}_r P_1''$ and $P_2' \xrightarrow{\Sigma}_r P_2''$ for some P_1'', P_2'', such that $\Sigma \subseteq (A \cap (\Sigma_1 \cup \Sigma_2)) \cup ((\Sigma_1 \cap \Sigma_2) \setminus A)$, hence by ind. and Definition 4.6.4: $w'/\tau = (w\Sigma)/\tau = (w/\tau)\Sigma \in (((w_1/\tau)\Sigma_1)\|_A((w_2/\tau)\Sigma_2)) = ((w_1\Sigma_1)/\tau\|_A((w_2\Sigma_2)/\tau)) = ((w_1'/\tau)\|_A(w_2'/\tau))$.

'$\supseteq$':

Let $v \in (\mathsf{RT}(P_1)\|_A \mathsf{RT}(P_2))$. Then there are $w_1 \in \mathsf{RT}(P_1)$ and $w_2 \in \mathsf{RT}(P_2)$ such that $v \in ((w_1/\tau)\|_A(w_2/\tau))$, $P_1 \xrightarrow{w_1}_r P_1'$ and $P_2 \xrightarrow{w_2}_r P_2'$. We show for all these w_1 and w_2 that there is a $w \in \mathsf{RT}_\tau(P)$ with $w/\tau = v \in \mathsf{RT}(P)$, and – furthermore – if $P \overset{w}{\rightarrow}_r P'$ for this w, then $P' \equiv P_1' \|_A P_2'$. We perform induction on $|w_1| + |w_2|$. For $|w_1| + |w_2| = 0$ we choose $w = \lambda$ and get $P \equiv P' \equiv P_1 \|_A P_2 \equiv P_1' \|_A P_2'$. We now distinguish several cases:

1. $w_1 = w_1'\tau$. Then $w_1/\tau = w_1'/\tau$, hence $v \in ((w_1/\tau)\|_A(w_2/\tau)) = ((w_1'/\tau)\|_A(w_2/\tau))$, and by ind. there is a w' with $w'/\tau = v$, $P \overset{w'}{\rightarrow}_r P' \equiv P_1' \|_A P_2'$, $P_1 \overset{w_1'}{\rightarrow}_r P_1'$ and $P_2 \overset{w_2}{\rightarrow}_r P_2'$. Now by rule Par_{a1}: $P' \overset{\tau}{\rightarrow}_r P'' \equiv P_1'' \|_A P_2'$ since $P_1' \overset{\tau}{\rightarrow}_r P_1''$ for some P_2'', and we may choose $w = w'\tau$ since $w/\tau = (w'\tau)/\tau = w'/\tau = v$.
2. $w_2 = w_2'\tau$. Analogously to 1.
3. Neither 1. nor 2. but $w_1 = w_1'a$ with $a \in A \subseteq \mathbb{A}_\omega$. Then $v = v'a$ and $w_2 = w_2'a$ for some v', w_2' by Definition 4.6.1, such that $v' \in ((w_1'/\tau)\|_A(w_2'/\tau))$. Now by ind. there is a w' with $w'/\tau = v'$ and $P \overset{w'}{\rightarrow}_r P' \equiv P_1' \|_A P_2'$, where $P_1 \overset{w_1'}{\rightarrow}_r P_1'$ and $P_2 \overset{w_2'}{\rightarrow}_r P_2'$. By rule Par_{a2} and assumption: $P' \overset{a}{\rightarrow}_r P'' \equiv P_1'' \|_A P_2''$ where $P_1' \overset{a}{\rightarrow} P_1''$, $P_2' \overset{a}{\rightarrow} P_2''$, and we may choose $w = w'a$ since $w/\tau = (w'a)/\tau = (w'/\tau)a = v'a = v$.
4. Neither 1. nor 2. but $w_2 = w_2'a$ with $a \in A \subseteq \mathbb{A}_\omega$. Analogously to 3.
5. Neither 1. nor 2. but $w_1 = w_1'a$ with $a \in \mathbb{A}_\omega \setminus A$. Then by Definition 4.6.2, $v = v'a$ for some v' with $v' \in ((w_1'/\tau)\|_A(w_2/\tau))$, and by ind. there is a w' with $w'/\tau = v'$ and $P \overset{w'}{\rightarrow}_r P' \equiv P_1' \|_A P_2'$, where $P_1 \overset{w_1'}{\rightarrow}_r P_1'$ and $P_2 \overset{w_2}{\rightarrow}_r P_2'$. By rule Par_{a1} and assumption: $P' \overset{a}{\rightarrow}_r P'' \equiv P_1'' \|_A P_2'$ for some P_1'', and we may choose $w = w'a$ since $w/\tau = (w'a)/\tau = (w'/\tau)a = v'a = v$.
6. Neither 1. nor 2. but $w_2 = w_2'a$ with $a \in \mathbb{A}_\omega \setminus A$. Analogously to 5.
7. Neither 1. nor 2. but $w_1 = w_1'\Sigma_1$ with $\Sigma_1 \subseteq \mathbb{A}_\omega$. Then by Definition 4.6.4: $v = v'\Sigma$ and $w_2 = w_2'\Sigma_2$ for some v', w_2' and Σ, Σ_2 such that $\Sigma \subseteq (A \cap (\Sigma_1 \cup \Sigma_2)) \cup ((\Sigma_1 \cap \Sigma_2) \setminus A)$ and $v' \in ((w_1'/\tau)\|_A(w_2'/\tau))$. Now by ind. there is a w' with $w'/\tau = v'$ and $P \overset{w'}{\rightarrow}_r P' \equiv P_1' \|_A P_2'$, where $P_1 \overset{w_1'}{\rightarrow}_r P_1'$ and $P_2 \overset{w_2'}{\rightarrow}_r P_2'$. By rule Par_r and assumption: $P' \overset{\Sigma}{\rightarrow}_r P'' \equiv P_1'' \|_A P_2''$ where $P_1' \overset{\Sigma_1}{\rightarrow}_r P_1''$, $P_2' \overset{\Sigma_2}{\rightarrow}_r P_2''$, and we may choose $w = w'\Sigma$ since $w/\tau = (w'\Sigma)/\tau = (w'/\tau)\Sigma = v'\Sigma = v$.
8. Neither 1. nor 2. but $w_2 = w_2'\Sigma_2$ with $\Sigma_2 \subseteq \mathbb{A}_\omega$. Analogously to 7.

The additional property follows since $\|_A$ is monotonic: consider any $R, R_1, R_2 \subseteq (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ with $R_1 \subseteq R_2$; then by Definition 4.6: $R\|_A R_1 = \bigcup\{u\|_A v \mid u \in R,\, v \in R_1\} \subseteq \bigcup\{u\|_A v \mid u \in R,\, v \in R_1\} \cup \bigcup\{u\|_A v \mid u \in R,\, v \in R_2 \setminus R_1\} = \bigcup\{u\|_A v \mid u \in R,\, v \in R_2\} = R\|_A R_2$. ■ 4.7

We now show that (τ-)refusal-trace-inclusion is also a precongruence for prefix:

Definition 4.8 *prefix of refusal traces*

For $R \subseteq (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ and $a \in \mathbb{A}_\omega$ we define

1. $a.R$ to be the set of all prefixes of
   $\{\Sigma_1 \ldots \Sigma_n a \mid n \in \mathbb{N}_0,\ \Sigma_1 \subseteq \mathbb{A}_\omega,\ \Sigma_2, \ldots, \Sigma_n \subseteq \mathbb{A}_\omega \setminus \{a\}\} \circ R$,
2. $\underline{a}.R$ to be the set of all prefixes of
   $\{\Sigma_1 \ldots \Sigma_n a \mid n \in \mathbb{N}_0,\ \Sigma_1, \ldots, \Sigma_n \subseteq \mathbb{A}_\omega \setminus \{a\}\} \circ R$,
3. $\tau.R = \{\Sigma, \lambda \mid \Sigma \subseteq \mathbb{A}_\omega\} \circ R$,
4. $\underline{\tau}.R = R$. ■ 4.8

Theorem 4.9

Let $P \in \mathbb{P}_1$ be an initial process and $a \in \mathbb{A}_{\omega\tau}$. Then $\mathsf{RT}(a.P) = a.\mathsf{RT}(P)$ and $\mathsf{RT}(\underline{a}.P) = \underline{a}.\mathsf{RT}(P)$. Furthermore, if $a \neq \tau$, then $\mathsf{RT}_\tau(a.P) = a.\mathsf{RT}_\tau(P)$ and $\mathsf{RT}_\tau(\underline{a}.P) = \underline{a}.\mathsf{RT}_\tau(P)$. Finally $\mathsf{RT}_\tau(\tau.P)$ is the set of all prefixes of $\{\tau, \Sigma\tau \mid \Sigma \subseteq \mathbb{A}_\omega\} \circ \mathsf{RT}_\tau(P)$ and $\mathsf{RT}_\tau(\underline{\tau}.P)$ is the set of all prefixes of $\{\tau\} \circ \mathsf{RT}_\tau(P)$. In particular, both RT-inclusion and RT_τ-inclusion are precongruences for prefixing of (initial) processes.

Proof:

Using Definition 4.3, Definition 4.1 and Definition 4.8 for $a \in \mathbb{A}_{\omega\tau}$:

$\mathcal{A}(a.P) = \{a\}$, $a.P \xrightarrow{a}_r P$, $a.P \xrightarrow{\Sigma}_r \underline{a}.P$ for all $\Sigma \subseteq \mathbb{A}_\omega$, $\underline{a}.P \xrightarrow{\Sigma'}_r \underline{a}.P$ if $\tau \neq a \notin \Sigma' \subseteq \mathbb{A}_\omega$ and $\underline{a}.P \xrightarrow{a}_r P$.

For the additional property consider any $R, R_1, R_2 \subseteq (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ with $R_1 \subseteq R_2$; then obviously for any $w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$: $w \in R \circ R_1 \Rightarrow w \in R \circ R_2$, and also for all of their prefixes. ■ 4.9

It is worth noting that due to Definition 4.8.3 and Theorem 4.9 we are not able do find a metric in the domain of sets of refusal traces, for which prefixing is a contractive function; this will rule out application of BANACH's fixpoint theorem when treating recursion in Section 5. We allow τ's as guards for recursion, and they actually gain some visibility in refusal traces due to time steps, but, however, this is not enough for making fixpoints unique modulo RT-equivalence: consider $P \equiv \mu X.\tau.X$ and $Q \equiv \mu X.(\tau.X + a.\mathbf{0})$; we have $\tau.X\{P/X\} =_r P$ and $\tau.X\{Q/X\} =_r Q$, but $P \neq_r Q$.

For the characterization we will also use the precongruence property of (τ-)refusal-trace-inclusion w.r.t. hiding and relabelling:

Definition 4.10 *relabelling of refusal traces*

Let Φ be a general relabelling function, $a \in \mathbb{A}_{\omega\tau}$, $\Sigma \subseteq \mathbb{A}_\omega$ and define $a[\Phi]_\tau^{-1} = \Phi^{-1}(a)$ and $\Sigma[\Phi]_\tau^{-1} = \{\Phi^{-1}(\Sigma \cup \{\tau\}) \setminus \{\tau\}\}$; we extend $[\Phi]_\tau^{-1}$ to sequences $w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ via concatenation $\circ$. We define $[\Phi]^{-1}$ identically, but additionally $\lambda[\Phi]^{-1} = \Phi^{-1}(\tau) \setminus \{\tau\}$. Φ^{-1} is again extended to sequences. ■ 4.10

Theorem 4.11

For a process $P \in \mathbb{P}$ and a general relabelling function Φ we have

1. $\mathsf{RT}(P[\Phi]) = \{w \in (\mathbb{A}_\omega \cup 2^{\mathbb{A}_\omega})^* \mid w[\Phi]^{-1} \cap \mathsf{RT}(P) \neq \emptyset\}$
2. $\mathsf{RT}_\tau(P[\Phi]) = \{w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^* \mid w[\Phi]_\tau^{-1} \cap \mathsf{RT}_\tau(P) \neq \emptyset\}$

Furthermore, both RT-inclusion and RT_τ-inclusion are precongruences for general relabelling $P[\Phi]$ of processes, in particular for relabelling $P[f]$ and hiding P/A.

Proof:

By Definition 4.1, Definition 4.3 and Definition 4.10. For the additional property consider any $R_1, R_2 \subseteq (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ with $R_1 \subseteq R_2$; then $\{w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^* \mid w[\Phi]_\tau^{-1} \cap R_1 \neq \emptyset\} \subseteq \{w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^* \mid w[\Phi]_\tau^{-1} \cap R_2 \neq \emptyset\}$ and the same holds for $[\Phi]^{-1}$. ■ 4.11

Another property needed for the above mentioned test construction is that $\mathbf{0}$ is a zero element for both choice and parallel composition without synchronisation:

Proposition 4.12

Let $P \in \tilde{\mathbb{P}}$ be a process term; then $P \|_\emptyset \mathbf{0} =_r P$ and $P + \mathbf{0} =_r P$.

Proof:

Let $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}$ be a suitable substitution.

First, $[P \|_\emptyset \mathbf{0}]_\mathcal{S} \equiv [P]_\mathcal{S} \|_\emptyset \mathbf{0}$ and $\mathsf{RT}([P]_\mathcal{S} \|_\emptyset \mathbf{0}) = \mathsf{RT}([P]_\mathcal{S}) \|_\emptyset \mathsf{RT}(\mathbf{0})$ by Theorem 4.7. Now since by Definition 4.1 and Definition 4.3, $\mathsf{RT}(\mathbf{0}) = \{\Sigma_1 \dots \Sigma_n \mid n \in \mathbb{N},\, \Sigma_1, \dots, \Sigma_n \subseteq \mathbb{A}_\omega\}$, Proposition 4.2.1 and Definition 4.6 (where only cases 2 and 4 apply, since $A = \emptyset$) yield $\mathsf{RT}([P]_\mathcal{S}) \|_\emptyset \mathsf{RT}(\mathbf{0}) = \mathsf{RT}([P]_\mathcal{S})$.

Second, we show for some $v \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ by induction on $|v|$: $[P + \mathbf{0}]_\mathcal{S} \xrightarrow{v}_r Q$ if and only if $[P]_\mathcal{S} \xrightarrow{v}_r Q$ or $[P]_\mathcal{S} \xrightarrow{v}_r R$ for some R, such that $Q \equiv R + \mathbf{0}$. In the base case $v = \lambda$ we have $[P + \mathbf{0}]_\mathcal{S} \equiv [P]_\mathcal{S} + \mathbf{0} \xrightarrow{\lambda}_r [P]_\mathcal{S} + \mathbf{0} \equiv Q$ and $[P]_\mathcal{S} \xrightarrow{\lambda}_r [P]_\mathcal{S} \equiv R$, thus $Q \equiv R + \mathbf{0}$. Hence assume the claim to hold for some v and consider $v\varepsilon$ with $\varepsilon \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})$. Then $[P + \mathbf{0}]_\mathcal{S} \xrightarrow{v}_r Q \xrightarrow{\varepsilon}_r Q'$ iff by induction either $[P]_\mathcal{S} \xrightarrow{v}_r Q \xrightarrow{\varepsilon}_r Q'$ or $[P]_\mathcal{S} \xrightarrow{v}_r R$ for some R, such that $Q \equiv R + \mathbf{0} \xrightarrow{\varepsilon}_r Q'$. In the first case, we obviously have $[P]_\mathcal{S} \xrightarrow{v\varepsilon}_r Q'$ again. In the second case, first let $\varepsilon = a \in \mathbb{A}_{\omega\tau}$; then $R + \mathbf{0} \xrightarrow{a}_r Q'$ iff $R \xrightarrow{a}_r Q'$ by rule Sum_a, hence $[P]_\mathcal{S} \xrightarrow{v\varepsilon}_r Q'$ again. Now let $\varepsilon = \Sigma \subseteq \mathbb{A}_\omega$; then $R + \mathbf{0} \xrightarrow{\Sigma}_r Q'$ iff by rules Sum_r and Nil_r: $R \xrightarrow{\Sigma}_r R'$ for some R', thus $[P]_\mathcal{S} \xrightarrow{v\varepsilon}_r R'$, such that $Q' \equiv R' + \mathbf{0}$. We conclude $\mathsf{RT}_\tau([P + \mathbf{0}]_\mathcal{S}) = \mathsf{RT}_\tau([P]_\mathcal{S})$, hence $\mathsf{RT}([P + \mathbf{0}]_\mathcal{S}) = \mathsf{RT}([P]_\mathcal{S})$ by Definition 4.3. ∎ 4.12

Finally, we state that refusal traces can always be extended by a time step (after performing all urgent internal activity) and that time steps can be omitted:

Proposition 4.13

Let $P, P', P'' \in \mathbb{P}$ be processes, $w, w' \in (\mathbb{A}_\omega \cup 2^{\mathbb{A}_\omega})^*$ and $\Sigma \subseteq \mathbb{A}_\omega$.

1. $w \in \mathsf{RT}(P)$ if and only if $w\emptyset \in \mathsf{RT}(P)$.
2. $w\Sigma w' \in \mathsf{RT}(P)$ implies $ww' \in \mathsf{RT}(P)$.
3. If $P \xRightarrow{w}_r P'' =_r P' \xRightarrow{w'}_r$, then $ww' \in \mathsf{RT}(P)$.

Proof:

1. 'if': clear by Definition 4.3.
'only-if': assume $P \xRightarrow{w}_r P'$ for some $P' \in \mathbb{P}$; then by Proposition 3.3.5 there is a $t \in \{\tau\}^*$ such that $P' \xrightarrow{t}_r P''$ for some $P'' \in \mathbb{P}$ with $\mathcal{R}(\tau, P'') = 1$; now $P'' \xrightarrow{\emptyset}_r$ by Proposition 4.2.4 and Lemma 2.7.1 since P'' is closed, hence guarded, and finally $P \xRightarrow{w\emptyset}_r$ by Definition 4.3.

2. $w\Sigma w' \in \mathsf{RT}(P)$ implies $w\Sigma w' = (u\Sigma v)/\tau$ for some $u\Sigma v \in \mathsf{RT}_\tau(P)$. Now it suffices to show by induction on $|v|$ that $u\Sigma v \in \mathsf{RT}_\tau(P)$ implies $uv \in \mathsf{RT}_\tau(P)$, where we additionally show that for P_1 reached after uv and P_2 reached after $u\Sigma v$ we have $P_1 \succ_0 P_2$.

The base case is $v = \lambda$, hence $P \xrightarrow{u}_r P_1$ and $P_1 \succ_1 P_1$ by Proposition 3.6.1 and .5. Then by Proposition 4.2.4 and Proposition 3.6.9 and .8b), $P_1 \xrightarrow{\Sigma}_r P_2$ implies $P_1 \succ_0 P_2$. Now assume the property to hold for v.

If $v' = va$ for $a \in \mathbb{A}_{\omega\tau}$, then $P_2 \xrightarrow{a} P_2'$ for some P_2', and since $P_1 \succ_0 P_2$ by assumption, Proposition 3.6.7 implies $P_1 \xrightarrow{a} P_1'$ for some P_1', such that $P_1' \succ_0 P_2'$ again.

If $v' = v\Sigma'$ for $\Sigma' \subseteq \mathbb{A}_\omega$, then $P_2 \xrightarrow{\Sigma'}_r P_2'$ for some P_2' iff $P_2 \overset{1}{\rightsquigarrow}_c P_2'$ and for all $a \in \Sigma' \cup \{\tau\}$ we have $\mathcal{R}(a, P_2) = 1$ by Proposition 4.2.4. Hence, by assumption of $P_1 \succ_0 P_2$ and Proposition 3.6.9 and .8c), $P_1 \overset{1}{\rightsquigarrow}_c P_1'$ for some P_1', such that $P_1' \succ_0 P_2'$ again; furthermore by Proposition 3.6.4, for all $a \in \Sigma' \cup \{\tau\}$ we have $\mathcal{R}(a, P_1) \geq \mathcal{R}(a, P_2) - 0 = \mathcal{R}(a, P_2) = 1$, thus finally $P_1 \xrightarrow{\Sigma'}_r P_1'$ by Proposition 4.2.4.

3. Clear by Definition 4.3. ■ 4.13

We now have gathered all elements for characterising the efficiency preorder via refusal-trace-inclusion, which is our second main result:

Theorem 4.14 *characterization of the testing preorder*

Let P_1, P_2 be testable processes. Then $P_1 \sqsupseteq P_2$ if and only if $P_1 \leq_r P_2$.

Proof:

By Definition 4.5 and Definition 3.8 we may assume $P_1, P_2 \in \mathbb{P}_1$.

'if':

Let (O, D) be a timed test. Then $\mathsf{RT}(P_1) \subseteq \mathsf{RT}(P_2)$ implies $\mathsf{DL}(\tau.P_1 \|_{\mathbb{A}} O) \subseteq \mathsf{DL}(\tau.P_2 \|_{\mathbb{A}} O)$ by Theorem 4.9, Theorem 4.7 and Theorem 4.4. Thus, if P_1 fails the test due to some $w_1 \in \mathsf{DL}(\tau.P_1 \|_{\mathbb{A}} O)$, then so does P_2.

'only if':

We assume $P_1 \sqsupseteq P_2$ and take some $w_1 \in \mathsf{RT}(P_1)$. By Definition 4.1, Proposition 4.2.4 and Definition 3.10, all actions in w_1 are in $\ell(P_1) \cup \ell(P_2)$. Furthermore, by Proposition 4.2.3 and .1, we may assume that for all refusal sets Σ in w_1 we have $\Sigma \subseteq \ell(P_1) \cup \ell(P_2)$, which is finite due to Proposition 3.11.4.

Now let $w = w_1$ if $w_1 = \lambda$ and $w = w_1\emptyset$ otherwise; by Proposition 4.13.1, $w \in \mathsf{RT}(P_1)$, too. Furthermore $\Sigma w \in \mathsf{RT}(\tau.P_1)$ for each $\Sigma \subseteq \ell(P_1) \cup \ell(P_2)$ by Theorem 4.9 and Definition 4.8.3; for technical reasons, we will only consider the case where $\Sigma = \emptyset$.

We will construct a timed test $(O_{\Sigma w}, \zeta(w))$ that is failed by a testable process $P \in \mathbb{P}_1$ if and only if $\Sigma w \in \mathsf{RT}(\tau.P)$. Hence, P_1 fails $(O_{\Sigma w}, \zeta(w))$, thus by assumption P_2 fails $(O_{\Sigma w}, \zeta(w))$, too, and we conclude $\Sigma w \in \mathsf{RT}(\tau.P_2)$. But then $\Sigma w_1 \in \mathsf{RT}(\tau.P_2)$ by Proposition 4.13.1 and $w_1 \in \mathsf{RT}(P_2)$ or $\Sigma w_1 \in \mathsf{RT}(P_2)$ by Theorem 4.9 and Definition 4.8.2, i.e. $w_1 \in \mathsf{RT}(P_2)$ by Proposition 4.13.3, and we are done.

The proof is structured as follows: We first give the construction of $O_{\Sigma w}$ (1), then we show that P fails the test $(O_{\Sigma w}, \zeta(w))$ if $\tau.P$ performs Σw (2), and finally we show that P fails the test $(O_{\Sigma w}, \zeta(w))$ *only if* $\tau.P$ is able to perform Σw (3). All three parts are inductive w.r.t. the structure of w.

(1)

To make induction work, we define $O_{\Sigma w}$ for sequences Σw that end with $\emptyset$ but may start with an arbitrary $\Sigma \subseteq \ell(P_1) \cup \ell(P_2)$. Furthermore, all actions of Σw are in $\ell(P_1) \cup \ell(P_2)$ and all refusal sets are subsets of $\ell(P_1) \cup \ell(P_2)$.

$O_{\Sigma w}$ will consist of several components that communicate via synchronized actions which must not occur in the sort of P_1 or P_2. Hence, let $H = \{b_0, c_0, b_1, c_1, \dots\} \subseteq \mathbb{A}$ be an infinite set such that $H \cap (\ell(P_1) \cup \ell(P_2)) = \emptyset$; H exists since $\ell(P_1)$ and $\ell(P_2)$ are finite and $\mathbb{A}$ is infinite.

The components $Q_{\Sigma w}$, $R_{\Sigma w}$, $S_{\Sigma w}$ and $X_{\Sigma w}$ of $O_{\Sigma w}$ are defined inductively as follows:

The base case is $\Sigma w = \emptyset$:

$$\begin{array}{lcl} Q_\emptyset & \equiv & \omega.\mathbf{0} \\ S_\emptyset & \equiv & \mathbf{0} \\ X_\emptyset & \equiv & \mathbf{0} \end{array}$$

Now let the general case be $\Sigma w = \Sigma a_1 \dots a_n \Sigma' w'$, where $\Sigma' w'$ ends with $\emptyset$. We define:

$$\begin{array}{lcl} Q_{\Sigma w} & \equiv & (b_{\zeta(w)}.Q_{\Sigma' w'}) \parallel_\emptyset (c_{\zeta(w)}.\mathbf{0} + \omega.\mathbf{0}) \\ S_{\Sigma w} & \equiv & b_{\zeta(w)}.a_1 \dots a_n.c_{\zeta(w)}.S_{\Sigma' w'} \\ X_{\Sigma w} & \equiv & (b_{\zeta(w)}.X_{\Sigma' w'}) \parallel_\emptyset (c_{\zeta(w)-1}.\mathbf{0} + \sum_{x \in \Sigma'} x.\mathbf{0}) \end{array}$$

In both cases let

$$R_{\Sigma w} \equiv (X_{\Sigma w} \parallel_\emptyset c_{\zeta(w)}.\mathbf{0})$$

and finally $O_{\Sigma w} \equiv T_{\Sigma w}/H$ where

$$T_{\Sigma w} \equiv Q_{\Sigma w} \parallel_H S_{\Sigma w} \parallel_H R_{\Sigma w}$$

Before detailed formal reasoning, the function and the interplay of the parts are shortly and informally described in the following:

The part $\Sigma a_1 \dots a_n$ of $\Sigma w = \Sigma a_1 \dots a_n \Sigma' w'$ is called the $\zeta(w)$-th round of Σw, started by occurrence of Σ, whereas occurrence of Σ' marks the begin of the $(\zeta(w) - 1)$-th round.

$Q_{\Sigma w}$ is the 'clock'-part of the test, which for each round i of Σw enables an ω that is urgent after the time step starting round i and can only be deactivated by performing the auxiliary action c_i (*completion* of round i) before the next time step.

The 'action-sequence'-part $S_{\Sigma w}$ will ensure that c_i can only occur after performance of the action sequence $a_1 \dots a_n$, which itself must be preceded by the auxiliary action b_i (*begin* of round i). Furthermore, occurence of b_i triggers the activation of the ω for the next round by enabling $Q_{\Sigma' w'}$. This must not happen too early, i.e. b_i and hence c_i will be performed after the time step starting round i and before the next one.

At the beginning of the present round, the 'refusal-set'-part $X_{\Sigma w}$ enables all actions x from the refusal set Σ' of the following round in conflict with the auxiliary action c_{i-1} which has to occur only at completion of the following round. After the time-step of the present round, all x from Σ' have become urgent, but may not occur – i.e. must be refusable by the tested process at the time-step starting the following round.

Finally, $X_{\Sigma w}$ is augmented to $R_{\Sigma w}$ for proof-technical reasons, $T_{\Sigma w}$ puts all three parts via synchronisation together, and $O_{\Sigma w}$ hides the auxiliary actions away. Otherwise, they would have to synchronise with the tested process, which is of course impossible by the definition of H.

(2)
By Definition 3.8, P fails the test $(O_{\Sigma w}, \zeta(w))$ if and only if there is a $u \in \mathsf{DL}(\tau.P \|_{\mathbb{A}} O_{\Sigma w})$ without ω and with $\zeta(u) > \zeta(w)$. By Proposition 4.2.4, this is case if and only if there is a $v \in \mathsf{RT}(\tau.P \|_{\mathbb{A}} O_{\Sigma w})$ without ω and with $\zeta(v) > \zeta(w)$ and all refusal sets in v are $\mathbb{A}_\omega$. By Theorem 4.7, Proposition 4.2.1 and Definition 4.6, such a v exists if and only if $v \in (v_1 \|_{\mathbb{A}} v_2)$ for some $v_1 \in \mathsf{RT}(\tau.P)$ and $v_2 \in \mathsf{RT}(O_{\Sigma w})$ satisfying the following: $\zeta(v_1) = \zeta(v_2) > \zeta(w)$, both v_1 and v_2 are without ω, all refusal sets in both v_1 and v_2 contain ω, and $match(v_1) = v_2$, where $match$ is defined inductively as follows :

1. $match(\lambda) = \lambda$.
2. $match(av') = a\, match(v')$ for $a \in \mathbb{A}_\omega$.
3. $match(\Sigma v') = \overline{\Sigma}\, match(v')$ for $\Sigma \subseteq \mathbb{A}_\omega$, where $\overline{\Sigma}$ denotes $\{\omega\} \cup \mathbb{A} \setminus \Sigma$.

For any testable process P we have $\omega \notin \ell(P)$, hence by Proposition 3.11.1 and Proposition 4.2.3 and .1 we have $\Sigma w \in \mathsf{RT}(\tau.P)$ if and only if $v_1 \in \mathsf{RT}(\tau.P)$, where v_1 is Σw with each refusal set augmented by ω; also, $match(\Sigma w) = match(v_1)$. Hence, in order show that P fails the test $(O_{\Sigma w}, \zeta(w))$ if $\Sigma w \in \mathsf{RT}(\tau.P)$, by the above it suffices to show that $match(\Sigma w) = match(v_1) \in \mathsf{RT}(O_{\Sigma w})$.

In order to apply inductive reasoning, we consider an intermediate state that is reached when $O_{\Sigma w}$ performs $match(\Sigma w)$. Let

$$R^+_{\Sigma w} \equiv X_{\Sigma w} \|_\emptyset (\underline{c}_{\zeta(w)}.\mathbf{0} + \textstyle\sum_{x \in \Sigma} \underline{x}.\mathbf{0})$$

and let $O^+_{\Sigma w} \equiv T^+_{\Sigma w}/H$ where

$$T^+_{\Sigma w} \equiv Q_{\Sigma w} \|_H S_{\Sigma w} \|_H R^+_{\Sigma w}$$

We first observe that the properties

$$\begin{array}{lll} (2.1) & O_\emptyset & \overset{\mathbb{A}_\omega}{\Rightarrow}_r \\ (2.2) & O^+_\emptyset & \overset{\mathbb{A}_\omega}{\Rightarrow}_r \end{array}$$

hold, since (2.1) $O_\emptyset$ is initial and (2.2) $\sum_{x \in \emptyset} \underline{x}.\mathbf{0} \equiv \mathbf{0} \in \mathbb{P}_1$ by Definition 2.3, $Q_\emptyset, S_\emptyset \in \mathbb{P}_1$, and $\underline{c}_0$ is the only urgent action in $R^+_\emptyset$, but has no synchronization partner in $Q_\emptyset$ and $S_\emptyset$, thus $\mathcal{R}(O^+_\emptyset) = 1$, and we are done by Proposition 4.2.4.

Now let $\Sigma w = \Sigma a_1 \ldots a_n \Sigma' w'$. We show the following properties:

$$\begin{array}{llccc} (2.3) & O_{\Sigma w} & \overset{\overline{\Sigma} a_1 \ldots a_n}{\Rightarrow}_r & O'_{\Sigma w} & =_r & O^+_{\Sigma' w'} \\ (2.4) & O^+_{\Sigma w} & \overset{\overline{\Sigma} a_1 \ldots a_n}{\Rightarrow}_r & O^{+'}_{\Sigma w} & =_r & O^+_{\Sigma' w'} \end{array}$$

I.e. from both $O_{\Sigma w}$ and $O^+_{\Sigma w}$ by performing a sequence matching the $\zeta(w)$-th round of w, we reach a process that is RT-equivalent to $O^+_{\Sigma' w'}$. For the proof of (2.3) consider (using Corollary 4.12)

$$\begin{array}{lllcccc} Q_{\Sigma w} & \overset{\mathbb{A}_\omega}{\rightarrow}_r & \underline{b}_{\zeta(w)}.Q_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)}.\mathbf{0} + \underline{\omega}.\mathbf{0}) & \overset{b_{\zeta(w)} c_{\zeta(w)}}{\rightarrow}_r & Q_{\Sigma' w'} \|_\emptyset \mathbf{0} & =_r & Q_{\Sigma' w'} \\ S_{\Sigma w} & \overset{\mathbb{A}_\omega}{\rightarrow}_r & \underline{b}_{\zeta(w)}.a_1 \ldots a_n.c_{\zeta(w)}.S_{\Sigma' w'} & \overset{b_{\zeta(w)} a_1 \ldots a_n c_{\zeta(w)}}{\rightarrow}_r & & & S_{\Sigma' w'} \end{array}$$

and

$$\begin{array}{rlll} R_{\Sigma w} \equiv X_{\Sigma w} & \|_\emptyset \;\; c_{\zeta(w)}.\mathbf{0} & \overset{\mathbb{A}_\omega}{\rightarrow}_r & \\ (\underline{b}_{\zeta(w)}.X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & \|_\emptyset \;\; \underline{c}_{\zeta(w)}.\mathbf{0} & \overset{b_{\zeta(w)}c_{\zeta(w)}}{\rightarrow}_r & \\ (X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & \|_\emptyset \;\; \mathbf{0} & =_r & \\ (X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & & \equiv & R^+_{\Sigma' w'} \end{array}$$

Hence with synchronisation over H by Definition 4.6 and Theorem 4.7 we get

$$T_{\Sigma w} \overset{\mathbb{A}_\omega b_{\zeta(w)} a_1 \ldots a_n c_{\zeta(w)}}{\rightarrow}_r \; \underline{T}'_{\Sigma w} =_r (Q_{\Sigma' w'} \|_H S_{\Sigma' w'}) \|_H R^+_{\Sigma' w'} \equiv T^+_{\Sigma' w'}$$

and hiding H by Definition 4.10 and Theorem 4.11 and finally applying Proposition 4.2.1:

$$O_{\Sigma w} \overset{\overline{\Sigma} a_1 \ldots a_n}{\Rightarrow}_r \; O'_{\Sigma w} =_r O^+_{\Sigma' w'}$$

For (2.4), analogous arguments apply for the components $Q_{\Sigma w}$ and $S_{\Sigma w}$ of $O^+_{\Sigma w}$. For $R^+_{\Sigma w}$ consider (using Corollary 4.12 again):

$$\begin{array}{rlll} R^+_{\Sigma w} \equiv X_{\Sigma w} & \|_\emptyset \;\; (\underline{c}_{\zeta(w)}.\mathbf{0} + \sum_{x\in\Sigma} \underline{x}.\mathbf{0}) & \overset{\overline{\Sigma}\setminus\{c_{\zeta(w)}\}}{\rightarrow}_r & \\ (\underline{b}_{\zeta(w)}.X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & \|_\emptyset \;\; (\underline{c}_{\zeta(w)}.\mathbf{0} + \sum_{x\in\Sigma} \underline{x}.\mathbf{0}) & \overset{b_{\zeta(w)}c_{\zeta(w)}}{\rightarrow}_r & \\ (X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & \|_\emptyset \;\; \mathbf{0} & =_r & \\ (X_{\Sigma' w'} \|_\emptyset (\underline{c}_{\zeta(w)-1}.\mathbf{0} + \sum_{x\in\Sigma'} \underline{x}.\mathbf{0})) & & \equiv & R^+_{\Sigma' w'} \end{array}$$

Hence, analogously to the above:

$$O^+_{\Sigma w} \overset{\overline{\Sigma} a_1 \ldots a_n}{\Rightarrow}_r \; O^{+'}_{\Sigma w} =_r O^+_{\Sigma' w'}$$

Using these properties, we now perform induction on the length of Σw to show that $match(\Sigma w) \in \mathsf{RT}(O^+_{\Sigma w})$:

For $\Sigma w = \emptyset$, by (2.2) we have $\mathbb{A}_\omega \in \mathsf{RT}(O^+_{\Sigma w})$ and $\mathbb{A}_\omega = match(\emptyset)$. For $\Sigma w = \Sigma a_1 \ldots a_n \Sigma' w'$ by (2.4) and Proposition 4.13.4 we have $\overline{\Sigma} a_1 \ldots a_n\, match(\Sigma' w') \in \mathsf{RT}(O^+_{\Sigma w})$ by induction.

It remains to show $match(\Sigma w) \in \mathsf{RT}(O_{\Sigma w})$: for $\Sigma w = \emptyset$ we are done by (2.1); for $\Sigma w = \Sigma a_1 \ldots a_n \Sigma' w'$, we have $O_{\Sigma w} \overset{\overline{\Sigma} a_1 \ldots a_n}{\Rightarrow}_r O'_{\Sigma w} =_r O^+_{\Sigma' w'}$ by (2.3) and $match(\Sigma' w') \in \mathsf{RT}(O^+_{\Sigma' w'})$ by the above, hence we are done by Proposition 4.13.4.

(3)
We now show that P fails the test $(O_{\Sigma w}, \zeta(w))$ only if $\tau.P$ is able to perform Σw.

We say that a refusal trace $v \in \mathsf{RT}(O_{\Sigma w})$ refuses ω if ω does not occur in v but in all refusal sets of v. Now by Theorem 4.7, Definition 4.6 and analogous arguments as in the beginning of part (2), P can fail the test $(O_{\Sigma w}, \zeta(w))$ only if there is a $v \in \mathsf{RT}(O_{\Sigma w})$ that refuses ω with $\zeta(v) > \zeta(w)$ and $match(v) \in \mathsf{RT}(\tau.P)$. We will show that this implies $\Sigma w \in \mathsf{RT}(\tau.P)$ and are done.

By $V(O_{\Sigma w})$ we denote the set of all $v \in \mathsf{RT}(O_{\Sigma w})$ that refuse ω and satisfy $\zeta(v) > \zeta(w)$, and similarly for $Q_{\Sigma w}$ etc. We will determine $V(O_{\Sigma w})$ by induction on the length of Σw, where we first state the following properties:

The base case is $w = \lambda$ and $\Sigma = \emptyset$:

(3.1) $(Q_\emptyset \|_H S_\emptyset) \overset{v}{\Rightarrow}_r$ for a v refusing ω with $\zeta(v) > \zeta(w) = 0$ if and only if $v = \Sigma_v$ for some $\Sigma_v \subseteq \mathbb{A}_\omega$ with $\omega \in \Sigma_v$, hence $\zeta(v) = \zeta(w) + 1 = 1$.

(3.2) There is *no* v refusing ω with $\zeta(v) > 1$ such that $((Q_\emptyset \|_\emptyset (c_1.\mathbf{0} + \omega.\mathbf{0})) \|_H (a'_1 \dots a'_m.c_1.S_\emptyset)) \overset{v}{\Rightarrow}_r$.

Now let $\Sigma w = \Sigma a_1 \dots a_n \Sigma' w'$, where $\Sigma' w'$ ends with $\emptyset$:

(3.3) $(Q_{\Sigma w} \|_H S_{\Sigma w}) \overset{v}{\Rightarrow}_r$ for a v refusing ω with $\zeta(v) > \zeta(w)$ if and only if $v = \Sigma_v b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)} v'$ for some $\Sigma_v \subseteq \mathbb{A}_\omega$ with $\omega \in \Sigma_v$ and v' refuses ω and $\zeta(v) = \zeta(w) + 1$, such that
$(Q_{\Sigma w} \|_H S_{\Sigma w}) \overset{\Sigma_v b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)}}{\Rightarrow}_r \; Q'_{\Sigma' w'} =_r (Q_{\Sigma' w'} \|_H S_{\Sigma' w'}) \overset{v'}{\Rightarrow}_r$.

(3.4) There is *no* v refusing ω with $\zeta(v) > \zeta(w) + 1$ such that $(Q_{\Sigma w} \|_\emptyset (c_{\zeta(w)+1}.\mathbf{0} + \omega.\mathbf{0})) \|_H (a'_1 \dots a'_m.c_{\zeta(w)+1}.S_{\Sigma w}) \overset{v}{\Rightarrow}_r$.

Whereas (3.1) and (3.2) can be checked directly, we show (3.3) and (3.4) by induction using Corollary 4.12, Theorem 4.7 and Proposition 4.13.4:

(3.3) The if-case is clear. $(Q_{\Sigma w} \|_H S_{\Sigma w})$ can perform $b_{\zeta(w)}$, ω or a time step Σ_v. Performance of $b_{\zeta(w)}$ yields $(Q_{\Sigma' w'} \|_\emptyset (c_{\zeta(w)}.\mathbf{0} + \omega.\mathbf{0})) \|_H (a_1 \dots a_n.c_{\zeta(w)}.S_{\Sigma' w'})$, and since $\zeta(w') = \zeta(w) - 1$, by ind. and (3.4) or (3.2), no v refusing ω with $\zeta(v) > \zeta(w) = \zeta(w') + 1$ is possible any more.
Hence, v starts with some $\Sigma_v \subseteq \mathbb{A}_\omega$ with $\omega \in \Sigma_v$; afterwards, only $b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)}$ is possible, since ω is urgent, hence no time step may occur before its deactivation by $c_{\zeta(w)}$; now a process RT-equivalent to $(Q_{\Sigma' w'} \|_H S_{\Sigma' w'})$ is reached, and $v = \Sigma_v b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)} v'$. By ind. or (3.1), $\zeta(v') = \zeta(w') + 1$, hence $\zeta(v) = \zeta(w) + 1$.

(3.4) There are two possibilities for an appropriate v:
i) v starts $a'_1 \dots a'_m c_{\zeta(w)+1}$, reaching a *unique* process, which is RT-equivalent to $(Q_{\Sigma w} \|_H S_{\Sigma w})$; but then (3.3) yields $\zeta(v) = \zeta(w) + 1$ only.
ii) v starts $a'_1 \dots a'_i \Sigma_v a'_{i+1} \dots a'_m c_{\zeta(w)+1}$ with $0 \le i \le m$ and $\Sigma_v \subseteq \mathbb{A}_\omega$ with $\omega \in \Sigma_v$, yielding a *unique* process RT-equivalent to $((\underline{b}_{\zeta(w)}.Q_{\Sigma' w'}) \|_\emptyset (\underline{c}_{\zeta(w)}.\mathbf{0} + \underline{\omega}.\mathbf{0})) \|_H S_{\Sigma w}$; now due to the urgent ω, from here only $b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)}$ is possible, reaching a *unique* process that is RT-equivalent to $(Q_{\Sigma' w'} \|_H S_{\Sigma' w'})$; but then (3.3) or (3.1) yields only $\zeta(v) = 1 + \zeta(w') + 1 = \zeta(w) + 1$ again.

We are now able to determine the set $V(Q_{\Sigma w} \|_H S_{\Sigma w})$: by (3.1), we have $V(Q_\emptyset \|_H S_\emptyset) = \{\Sigma_v \subseteq \mathbb{A}_\omega \mid \omega \in \Sigma_v\}$, and by (3.3) and induction, for $\Sigma w = \Sigma a_1 \dots a_n \Sigma' w'$ we get

$$V(Q_{\Sigma w} \|_H S_{\Sigma w}) = \{\Sigma_v \subseteq \mathbb{A}_\omega \mid \omega \in \Sigma_v\} \circ \{b_{\zeta(w)} a_1 \dots a_n c_{\zeta(w)}\} \circ V(Q_{\Sigma' w'} \|_H S_{\Sigma' w'})$$

For the following let $l = \zeta(w)$ and Σw be of the form

$$\Sigma w = \; \Sigma^l \, a^l_1 \dots a^l_{n_l} \, \Sigma^{l-1} \; \dots \; \Sigma^0$$

Hence $v_1 \in V(Q_{\Sigma w} \|_H S_{\Sigma w})$ is of the form

$$v_1 = \; \Gamma^l \, b_l \, a^l_1 \dots a^l_{n_l} \, c_l \, \Gamma^{l-1} \, b_{l-1} \; \dots \; c_1 \, \Gamma^0,$$

where $\omega \in \Gamma^i \subseteq \mathbb{A}_\omega$ for all $i = 0, \ldots, l$. Now by Theorem 4.7, $v_3 \in V(T_{\Sigma w})$ implies $v_3 \in (v_1 \|_H v_2)$, where $v_1 \in V(Q_{\Sigma w} \|_H S_{\Sigma w})$ and $v_2 \in \mathsf{RT}(R_{\Sigma w})$. By the above and Definition 4.6, $(v_1 \|_H v_2) \neq \emptyset$ only if v_2 is of the form

$$v_2 = \; u_1^l \, \Upsilon^l \, u_2^l \, b_l \, u_3^l \, c_l \, u_1^{l-1} \, \Upsilon^{l-1} \, u_2^{l-1} \, b_{l-1} \; \ldots \; u_3^1 \, c_1 \, u_1^0 \, \Upsilon^0 \, u_2^0 \, u_3^0,$$

where $\omega \in \Upsilon^i \subseteq \mathbb{A}_\omega$ for all $i = 0, \ldots, l$ and $u_j^i \in (\mathbb{A} \setminus H)^*$ for $i = 0, \ldots, l$ and $j = 1, \ldots, 3$. If $u_j^i = au$ for some $i = 0, \ldots, l$ and $j = 1, \ldots, 3$ and $a \in \mathbb{A}$, then a must stem from some sum-part of $X_{\Sigma w}$, hence the respective c_k could not occur any more; observe that the sum-part for c_0 is empty. We conclude $u_j^i = \lambda$ for all $i = 1, \ldots, l$ and all $j = 1, \ldots, 3$.

Furthermore, $R_{\Sigma w} \overset{v_2}{\Rightarrow}_r R'$ if and only if $R_{\Sigma w} \overset{v_2}{\rightarrow}_r R'$, since $\tau \notin \ell(R_{\Sigma w})$, and the derivations of (2) show that $R_{\Sigma w} \overset{v_2}{\rightarrow}_r$ if and only if $\Upsilon^i \subseteq \overline{\Sigma}^i \setminus \{c_i\}$ for all $i = 0, \ldots, l-1$. As said in the very beginning of this proof, we will only consider the case where $\Sigma = \emptyset$ in Σw, hence since $\overline{\emptyset} = \mathbb{A}_\omega$, by Definition 4.6 and the above, we determine $v_3 \in V(T_{\Sigma w})$ to be of the form:

$$v_3 = \; \Gamma^l \, b_l \, a_1^l \ldots a_{n_l}^l \, c_l \, \Gamma^{l-1} \, b_{l-1} \; \ldots \; c_1 \, \Gamma^0,$$

where $\Gamma^i \subseteq \overline{\Sigma}^i$ for all $i = 0, \ldots, l$. Finally, with Theorem 4.11 and Definition 4.10 we calculate for the form of a $v \in V(O_{\Sigma w})$:

$$v = \; \Gamma^l \, a_1^l \ldots a_{n_l}^l \, \Gamma^{l-1} \; \ldots \; \Gamma^0,$$

where $\Gamma^i \subseteq \overline{\Sigma}^l$ for all $i = 0, \ldots, l$, hence

$$match(v) = \; \overline{\Gamma}^l \, a_1^l \ldots a_{n_l}^l \, \overline{\Gamma}^{l-1} \; \ldots \overline{\Gamma}^0,$$

such that $\overline{\Gamma}^i \supseteq \Sigma^i$ for $i = 0, \ldots, l$, thus by Proposition 4.2.1, $match(v) \in \mathsf{RT}(\tau.P)$ implies $\Sigma w \in \mathsf{RT}(\tau.P)$, and we are done. ■ 4.14

5 Full Abstractness

Refusal-trace-inclusion not only characterizes the efficiency preorder, but also makes just the necessary refinements to discrete behaviour of (initial) processes in order to gain a precongruence for parallel composition and prefix:

Corollary 5.1

The RT-semantics is fully abstract w.r.t. DL and parallel composition and prefixing of initial processes, i.e. it gives the coarsest congruence for initial processes and these operators that respects DL-equivalence. For process terms, $\leq_r$ is a precongruence for these operators, and also for hiding and relabelling.

Proof:

Theorem 4.7, Theorem 4.9, Theorem 4.11 and Theorem 4.4 show that RT-equivalence is a congruence and RT-inclusion is a precongruence for parallel composition, prefixing, hiding and relabelling of processes that respects DL-equivalence, -inclusion resp. By Definition 4.5, the result for RT-inclusion carries over to process terms related by $\leq_r$.

If for initial processes P_1, P_2 we have $\mathsf{RT}(P_1) \neq \mathsf{RT}(P_2)$, then the proof of Theorem 4.14 exhibits a test process O such that $\mathsf{DL}(\tau.P_1\|_{\mathbf{A}}O) \neq \mathsf{DL}(\tau.P_2\|_{\mathbf{A}}O)$. (If P_1 or P_2 contains the special action ω, then its rôle in O must be played by some other action $a \notin \ell(P_1)\cup\ell(P_2)$; consider $\mathsf{DL}(\tau.P_i\,\|_{\mathbf{A}_\omega-\{a\}}\,O)$ in this case.). Hence, RT-equivalence is the coarsest congruence that refines DL-equivalence to a congruence for parallel composition and prefixing of initial processes. ■ 5.1

As usual, the testing preorder alone is not a precongruence for choice: e.g. we have $\mathbf{0} \leq_\tau \tau.\mathbf{0}$ and $\tau.\mathbf{0} \leq_\tau \mathbf{0}$, but for $a \neq \tau$, we have neither $\mathbf{0} + a.\mathbf{0} \leq_\tau \tau.\mathbf{0} + a.\mathbf{0}$ (since e.g. $\emptyset\emptyset a \in \mathsf{RT}(\mathbf{0} + a.\mathbf{0})\backslash\mathsf{RT}(\tau.\mathbf{0}+a.\mathbf{0})$), nor $\tau.\mathbf{0}+a.\mathbf{0} \leq_\tau \mathbf{0}+a.\mathbf{0}$ (since e.g. $\{a\}\{a\} \in \mathsf{RT}(\tau.\mathbf{0}+a.\mathbf{0})\backslash\mathsf{RT}(\mathbf{0}+a.\mathbf{0})$). As a consequence, we also have to take into account the (initial) stability of processes, where the example indicates that although we consider a preorder this additional condition is not an implication but an equivalence:

Definition 5.2 *stable processes*

A process $P \in \mathbb{P}$ is *stable*, if no internal action is enabled, i.e. $\tau \notin \mathcal{A}(P)$.

For process terms $P, Q \in \tilde{\mathbb{P}}$ we write $P \leq Q$ if for all closed substitutions $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}$ where $[P]_\mathcal{S}, [Q]_\mathcal{S} \in \mathbb{P}$ we have: $\mathsf{RT}([P]_\mathcal{S}) \subseteq \mathsf{RT}([Q]_\mathcal{S})$ (hence $P \leq_\tau Q$) and additionally $[P]_\mathcal{S}$ stable iff $[Q]_\mathcal{S}$ stable. We write $P = Q$ if $P \leq Q$ and $Q \leq P$.

For all $n \in \mathbb{N}$ we write $P \leq^n_\tau Q$ if for all closed substitutions $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}$ where $[P]_\mathcal{S}, [Q]_\mathcal{S} \in \mathbb{P}$ we have: $v \in \mathsf{RT}_\tau([P]_\mathcal{S})$ and $|v| < n$ implies $v \in \mathsf{RT}_\tau([Q]_\mathcal{S})$. We write $P =^n_\tau Q$ if $P \leq^n_\tau Q$ and $Q \leq^n_\tau P$. We write $P \leq_\tau Q$ ($P =_\tau Q$) if $P \leq^n_\tau Q$ ($P =^n_\tau Q$) for all $n \in \mathbb{N}$. ■ 5.2

The additional definition of a class of RT_τ-inclusions ($\leq^n_\tau$) will support an approximation technique when treating recursion later on. The following results yield that we have defined $\leq$ adequately in order to gain the coarsest precongrence w.r.t. choice that respects RT-inclusion, hence the efficiency preorder:

Theorem 5.3

Let $P, Q \in \tilde{\mathbb{P}}$ be process terms. Then $P \leq_\tau Q$ implies $P \leq Q$, and $P \leq Q$ implies $P \leq_\tau Q$, but none of the reverse implications holds.

Proof:

It suffices to prove the claims for $P, Q \in \mathbb{P}$, where the second claim holds directly by Definition 5.2.

Let $P \leq_\tau Q$, i.e. $\mathsf{RT}_\tau(P) \subseteq \mathsf{RT}_\tau(Q)$, hence $\mathsf{RT}(P) \subseteq \mathsf{RT}(Q)$. If P is not stable, then $\tau \in \mathsf{RT}_\tau(P)$, hence $\tau \in \mathsf{RT}_\tau(Q)$ by assumption and Q is not stable. If P is stable, then $\tau \notin \mathcal{A}(P)$, hence $\mathcal{R}(\tau, P) = 1$ by Proposition 2.9.1, thus $P \xrightarrow{\emptyset}_r P'$ for some P' by Proposition 4.2.4 and Lemma 2.7.1 since P is guarded. Furthermore, $\tau \notin \mathcal{A}(P')$ by Lemma 2.7.4 and $\mathcal{R}(\tau, P') = 1$ by Proposition 2.9.1 again, hence $P' \xrightarrow{\emptyset}_r$ again. Now by assumption also $Q \xrightarrow{\emptyset}_r Q' \xrightarrow{\emptyset}_r$ for some Q', hence $\mathcal{R}(\tau, Q) = \mathcal{R}(\tau, Q') = 1$ by Proposition 4.2.4, thus $\tau \notin \mathcal{A}(Q)$ by Proposition 4.2.4 and Lemma 2.10. We conclude Q stable and are done.

For the reverse implications consider $P \equiv a.\mathbf{0}$, $Q \equiv a.\tau.\mathbf{0}$ and $R \equiv \tau.a.\mathbf{0}$: we have $P \leq Q \leq_\tau R$, but neither $P \leq_\tau Q$, nor $Q \leq R$. ■ 5.3

Theorem 5.4 *refusal traces of a sum*

Let either $P_i \in \mathbb{P}_1$ for all $i \in I$, or $P_i \in \mathbb{P}_0$ for all $i \in I$, hence $\sum_{i\in I} P_i \in \mathbb{P}$.

Let $\mathsf{RT}^n(\sum_{i\in I} P_i) = \{\Sigma_1 \dots \Sigma_n w \in \bigcup_{i\in I} \mathsf{RT}(P_i) \mid \Sigma_1 \dots \Sigma_n \in \bigcap_{i\in I} \mathsf{RT}(P_i),\ \Sigma_i \subseteq \mathbb{A}_\omega,\ w$ does not start with a set$\}$ for each $n \in \mathbb{N}_0$, and let $\mathsf{RT}^n_\tau(\sum_{i\in I} P_i)$ be defined analogously, with RT_τ instead of RT.

Now let $P \equiv \sum_{i\in I} P_i$ and let $I = S \dot\cup \overline{S}$ such that P_i is stable if and only if $i \in S$.

1. $\mathsf{RT}_\tau(P) = \bigcup_{n\in\mathbb{N}_0} \mathsf{RT}^n_\tau(\sum_{i\in I} P_i)$
2. If $S = I$, then $\mathsf{RT}(P) = \bigcup_{n\in\mathbb{N}_0} \mathsf{RT}^n(\sum_{i\in I} P_i)$.
3. If $S \neq I$ and $P \in \mathbb{P}_0$, then $\mathsf{RT}(P) = \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^0(\sum_{i\in I} P_i)$.
4. If $S \neq I$ and $P \in \mathbb{P}_1$, then $\mathsf{RT}(P) = \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^0(\sum_{i\in I} P_i) \cup \mathsf{RT}^1(\sum_{i\in I} P_i)$.

Proof:

1. Let $v \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$; then $v = \Sigma_1 \dots \Sigma_n w$ for some $n \in \mathbb{N}_0$ where $\Sigma_i \subseteq \mathbb{A}_\omega$ for all $i = 1, \dots, n$ and $w \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_\omega})^*$ does not start with a set. Now $P \xrightarrow{\Sigma_1 \dots \Sigma_n}_r P'$ if and only if $P_i \xrightarrow{\Sigma_1 \dots \Sigma_n}_r P_i'$ for all $i \in I$, such that $P' \equiv \sum_{i\in I} P_i'$ by rule Sum$_r$, and $P' \xrightarrow{w}_r$ if and only if $P_j' \xrightarrow{w}_r$ for some $j \in I$ by rule Sum$_a$, hence $v \in \mathsf{RT}_\tau(P)$ if and only if $\Sigma_1 \dots \Sigma_n \in \bigcap_{i\in I} \mathsf{RT}(P_i)$ and $\Sigma_1 \dots \Sigma_n w \in \bigcup_{i\in I} \mathsf{RT}(P_i)$.

2. Similar to 1., where we observe that $v = \Sigma_1 \dots \Sigma_n w$ and w does not start with a set implies that the underlying τ-refusal-trace does not contain a τ up to (if exists) the first $a \in \mathbb{A}_\omega$ in w, since all P_i are stable.

3. For a $w \in (\mathbb{A}_\omega \cup 2^{\mathbb{A}_\omega})^*$ and $P \equiv \sum_{i\in I} P_i \in \mathbb{P}_0$ with S a strict subset of I, we show that $w \in \mathsf{RT}(P)$ if and only if $w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^0(\sum_{i\in I} P_i)$. The case $w = \lambda$ is obvious; hence consider εw with $\varepsilon \in (\mathbb{A}_\omega \cup 2^{\mathbb{A}_\omega})$.

If $\varepsilon = a \in \mathbb{A}_\omega$, then $aw \in \mathsf{RT}(P)$ iff $P_j \xrightarrow{\tau}_r P_j' \xRightarrow{aw}_r$ for some $j \in \overline{S}$ by rule Sum$_a$, or $P_j \xrightarrow{a}_r P_j' \xRightarrow{w}_r$ for some $j \in I$, hence iff $aw \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$, or $aw \in \mathsf{RT}^0(\sum_{i\in I} P_i)$.

If $\varepsilon = \Sigma \subseteq \mathbb{A}_\omega$, then $\Sigma w \in \mathsf{RT}(P)$ iff $P_j \xrightarrow{\tau}_r P_j' \xRightarrow{\Sigma w}_r$ for some $j \in \overline{S}$ by rule Sum$_a$, hence iff $\Sigma w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$. We show that $P \xrightarrow{\Sigma}_r P' \xRightarrow{w}_r$ is impossible: Since $P \in \mathbb{P}_0$, by Proposition 3.3.4 there must be a $Q \in \mathbb{P}_1$, such that $Q \xrightarrow{1}_d P$, hence also $Q1 \overset{1}{\rightsquigarrow}_c P$ by Definition 2.11; now since $\overline{S} \neq \emptyset$, we have $\tau \in \mathcal{A}(Q) = \mathcal{A}(P)$ by Lemma 2.7.4, hence $\mathcal{R}(\tau, P) = 0$ by Lemma 2.10, thus $P \xrightarrow{\Sigma}_r P'$ is impossible by Proposition 4.2.4.

4. For a $w \in (\mathbb{A}_\omega \cup 2^{\mathbb{A}_\omega})^*$ and $P \equiv \sum_{i\in I} P_i \in \mathbb{P}_1$ with S a strict subset of I, we show that $w \in \mathsf{RT}(P)$ if and only if $w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^0(\sum_{i\in I} P_i) \cup \mathsf{RT}^1(\sum_{i\in I} P_i)$. The case $w = \lambda$ is obvious and the case aw for $a \in \mathbb{A}_\omega$ is analogously to the according case in the proof of 3., where we found that $aw \in \mathsf{RT}(P)$ iff $aw \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^0(\sum_{i\in I} P_i)$; hence, we only need to consider Σw with $\Sigma \subseteq \mathbb{A}_\omega$.

First let $\Sigma w \in \mathsf{RT}(P)$. Then either $P_j \xrightarrow{\tau}_r P_j' \xRightarrow{\Sigma w}_r$ for some $j \in \overline{S}$ by rule Sum$_a$ (i.e. $\Sigma w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$), or $P_i \xrightarrow{\Sigma}_r P_i'$ for all $i \in I$ by rule Sum$_r$, such that $P \equiv \sum_{i\in I} P_i \xrightarrow{\Sigma}_r \sum_{i\in I} P_i' \equiv P' \xRightarrow{w}_r$. In the second case, $P \in \mathbb{P}_1$ and $P \xrightarrow{\Sigma}_r P' \xRightarrow{w}_r$ implies $P \overset{1}{\rightsquigarrow}_c P' \xRightarrow{w}_r$ by Proposition 4.2.4 and Proposition 3.3.1, such that $P' \in \mathbb{P}_0$ by Proposition 3.3.3 and P_i'

is stable iff P_i is stable by Lemma 2.7.4 for all $i \in I$. Now we have $P \stackrel{\Sigma}{\to}_r \sum_{i\in I} P_i' \equiv P'$ and $w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i') \cup \mathsf{RT}^0(\sum_{i\in I} P_i')$ by 3., hence $\Sigma w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i) \cup \mathsf{RT}^1(\sum_{i\in I} P_i)$.

Now let $\Sigma w \in \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$. Then for some $j \in \overline{S}$ either $P_j \stackrel{\tau}{\to}_r P_j' \stackrel{\Sigma w}{\Rightarrow}_r$, hence by rule Sum_a also $P \stackrel{\Sigma w}{\Rightarrow}_r$ and $\Sigma w \in \mathsf{RT}(P)$, or $P_j \stackrel{\Sigma}{\to}_r P_j' \stackrel{w}{\Rightarrow}_r$. In the second case, $P \in \mathbb{P}_1$ implies by Proposition 3.3.1 and Proposition 4.2.4 $P_i \stackrel{\Sigma}{\to}_r P_i'$ for all $i \in I$, such that $P \stackrel{\Sigma}{\to}_r \sum_{i\in I} P_i' \equiv P'$. Hence, if $w = \lambda$, we have $\Sigma w \in \mathsf{RT}(P)$ and are done, thus let $w \neq \lambda$. Now $\tau \in \mathcal{A}(P_j')$ by $j \in \overline{S}$ and Lemma 2.7.4, hence $\mathcal{R}(\tau, P_j') = 0$ by Lemma 2.10, thus $P_j' \stackrel{w}{\Rightarrow}_r$ implies $P_j' \stackrel{av}{\Rightarrow}_r$ for some v such that $w = (av)/\tau$, where $a \in \mathbb{A}_{\omega\tau}$, since $P_j' \stackrel{\Sigma'}{\to}_r$ is impossible by $\mathcal{R}(\tau, P_j') = 0$ and Proposition 4.2.4. But then also $P' \stackrel{av}{\Rightarrow}_r$ by rule Sum_a, hence $P' \stackrel{w}{\Rightarrow}_r$, thus $\Sigma w \in \mathsf{RT}(P)$.

Now let $\Sigma w \in \mathsf{RT}^1(\sum_{i\in I} P_i) \setminus \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$. Then $P_j \stackrel{\Sigma w}{\Rightarrow}_r$ for some $j \in S$, hence $P_j \stackrel{\Sigma}{\to}_r P_j' \stackrel{w}{\Rightarrow}_r$ by the stability of P_j. As above, $P \in \mathbb{P}_1$ implies $P_i \stackrel{\Sigma}{\to}_r P_i'$ for all $i \in I$, such that $P \stackrel{\Sigma}{\to}_r \sum_{i\in I} P_i' \equiv P'$, and if $w = \lambda$, we have $\Sigma w \in \mathsf{RT}(P)$ and are done, thus let $w \neq \lambda$. Now $\Sigma w \in \mathsf{RT}^1(\sum_{i\in I} P_i)$ implies $w = aw'$ for some $a \in \mathbb{A}_\omega$, and since P_j' is stable again by Proposition 4.2.4 and Lemma 2.7.4, we have $P_j' \stackrel{aw'}{\Rightarrow}_r$ only if $P_j' \stackrel{a}{\to}_r P_j'' \stackrel{w}{\Rightarrow}_r$, hence also $P' \stackrel{a}{\to}_r P_j'' \stackrel{w}{\Rightarrow}_r$ by rule Sum_a, thus $\Sigma w \in \mathsf{RT}(P)$.

We finally see that $\Sigma w \in \mathsf{RT}^0(\sum_{i\in I} P_i)$ is impossible, and are done. ■ 5.4

Theorem 5.5

Both $\leq_\tau$ and $\leq$ are precongruences for parallel composition, prefixing, hiding and relabelling of process terms, and also for choice.

Proof:

By Theorem 4.7, Theorem 4.9 and Theorem 4.11 and Definition 5.2, $\leq_\tau$ is a precongruence for parallel composition, prefixing, hiding and relabelling of processes, which carries over to process terms, since substitutions distribute over these operations. By the same theorems, for $\leq$ it suffices to show that these operators preserve the condition on stability.

In the following let $P_1, P_1', P_2, P_2' \in \mathbb{P}$ with $P_1 \leq P_1'$ and $P_2 \leq P_2'$, let $A \subseteq \mathbb{A}_\omega$, $a \in \mathbb{A}_{\omega\tau}$ and let Φ be a general relabelling function.

Par: $P_1 \|_A P_2$ stable iff both P_1 stable and P_2 stable iff both P_1' stable and P_2' stable iff $P_1' \|_A P_2'$ stable, hence $P_1 \|_A P_2 \leq P_1' \|_A P_2'$.

Pref: Let $P_1, P_1' \in \mathbb{P}_1$. Then $a.P_1$ stable iff $a \neq \tau$ iff $a.P_1'$ stable, hence $a.P_1 \leq a.P_1'$.

Rel: If P_1 not stable then $P_1[\Phi]$ not stable, since $\Phi(\tau) = \tau$, hence neither P_1' stable nor $P_1'[\Phi]$ stable, thus $P_1[\Phi] \leq P_1'[\Phi]$.

Now assume P_1 stable, hence P_1' stable; we show $\mathcal{A}(P_1) = \mathcal{A}(P_1')$, hence $\mathcal{A}(P_1) \cap \Phi^{-1}(\tau) = \mathcal{A}(P_1') \cap \Phi^{-1}(\tau)$, thus $P_1[\Phi]$ stable iff $P_1'[\Phi]$ stable, yielding $P_1[\Phi] \leq P_1'[\Phi]$:

If $\tau \neq a \in \mathcal{A}(P_1)$, then $a \in \mathsf{RT}(P_1) \subseteq \mathsf{RT}(P_1')$, and since P_1' is stable, $P_1' \stackrel{a}{\Rightarrow}_r$ implies $P_1' \stackrel{a}{\to}_r$, hence $a \in \mathcal{A}(P_1')$.

If $\tau \neq a \notin \mathcal{A}(P_1)$, then $\mathcal{R}(a, P_1) = \mathcal{R}(\tau, P_1) = 1$ by Proposition 2.9.1, hence $P_1 \stackrel{\{a\}}{\to}_r Q$ for some $Q \in \mathbb{P}$ by Proposition 4.2.4; now $\tau, a \notin \mathcal{A}(Q) = \mathcal{A}(P_1)$ by Lemma 2.7.4, hence once again $Q \stackrel{\{a\}}{\to}_r$ by Proposition 4.2.4, thus $\{a\}\{a\} \in \mathsf{RT}(P_1) \subseteq \mathsf{RT}(P_1')$;

since P_1' is stable, $P_1' \overset{\{a\}\{a\}}{\Rightarrow}_r$ implies $P_1' \overset{\{a\}}{\rightarrow}_r R \overset{\{a\}}{\rightarrow}_r$ for some R, hence $\mathcal{R}(a, R) = 1$ by Lemma 4.2.4, thus $a \notin \mathcal{A}(R) = \mathcal{A}(P_1')$ by Lemma 2.10 and Lemma 2.7.4.

We now show that $\leq$ is a precongruence for choice. Let I be an indexing set and let for all $i \in I$ be $P_i, P_i' \in \mathbb{P}_0 \cup \mathbb{P}_1$, such that $P \equiv \sum_{i\in I} P_i$ and $P' \equiv \sum_{i\in I} P_i'$ are processes and $P_i \leq P_i'$ for all $i \in I$. Then P_i stable iff P_i' stable, hence $\sum_{i\in I} P_i$ stable iff all P_i are stable iff all P_i' are stable iff $\sum_{i\in I} P_i'$ stable. Furthermore, $P_i \leq P_i'$ for all $i \in I$ implies $\mathsf{RT}(P_i) \subseteq \mathsf{RT}(P_i')$ for all $i \in I$. Now let $S \subseteq I$, such that $i \in S$ iff P_i (and P_i') stable.

If $S = I$ or $P, P' \in \mathbb{P}_0$ or $P, P' \in \mathbb{P}_1$, then $\mathsf{RT}(P)$ and $\mathsf{RT}(P')$ can be calculated in the same way Theorem 5.4, hence $P \leq P'$; also the case $S \neq I$, $P \in \mathbb{P}_0$ and $P' \in \mathbb{P}_1$ is no problem. Thus we consider the case $S \neq I$, $P \in \mathbb{P}_1$ and $P' \in \mathbb{P}_0$. We are done once we have shown that $\mathsf{RT}^1(\sum_{i\in I} P_i) \subseteq \bigcup_{i\in\overline{S}} \mathsf{RT}(P_i)$.

First take some $\Sigma w \in \mathsf{RT}(P_j)$ with $j \in S$; the as in the Rel-case above, we have $\mathcal{A}(P_j) = \mathcal{A}(P_j')$. Assume $a \in \mathcal{A}(P_j)$, where $a \neq \tau$ since $j \in S$. Now $P_j \overset{a}{\rightarrow}_r$ by Proposition 4.2.4 since $P_j \in \mathbb{P}_1$, but not $P_j' \overset{a}{\rightarrow}_r$, since by $P_j' \in \mathbb{P}_0$ there is a $Q \in \mathbb{P}_1$ with $Q \overset{1}{\rightarrow}_d P_j'$ and $a \in \mathcal{A}(Q) = \mathcal{A}(P_j')$ by Proposition 3.3.4 and Proposition 2.12.3, such that $\mathcal{R}(a, P_j') = 0$ by Lemma 2.10; hence also not $P_j' \overset{a}{\Rightarrow}_r$ since P_j' is stable, and this a contradiction to $P_j \leq P_j'$. We conclude $\mathcal{A}(P_j) = \mathcal{A}(P_j') = \emptyset$ and $w \in (2^{\mathbb{A}_\omega})^*$.

Now consider $\Sigma w \in \mathsf{RT}^1(\sum_{i\in I} P_i)$, which is by definition in some $\mathsf{RT}(P_j)$ for $j \in I$. If $j \in \overline{S}$ we are done, so take $j \in S$. By the above, we have $w = \lambda$ and $\Sigma \in \mathsf{RT}(P_i)$ for each $i \in \overline{S}$ by the definition of RT^1.

We finally see that RT_τ-inclusion is a precongruence for sum by Theorem 5.4.1. ■ 5.5

Theorem 5.6

For initial processes, $\leq$ is fully abstract w.r.t. choice and $\leq_r$.

Proof:

By Theorem 5.3 and Theorem 5.5, we have to show that for any processes $P_1, P_2 \in \mathbb{P}_1$ we have $P_1 \leq P_2$ whenever $\forall P \in \mathbb{P}_1 : P_1 + P \leq_r P_2 + P$.

For given P_1, P_2 assume to the contrary, i.e. $\forall P \in \mathbb{P}_1 : P_1 + P \leq_r P_2 + P$, but $P_1 \not\leq P_2$; choosing $P \equiv \mathbf{0}$, we have $P_1 + \mathbf{0} =_r P_1$ and $P_2 + \mathbf{0} =_r P_2$ by Proposition 4.12, hence $P_1 \leq_r P_2$ by Theorem 5.5, thus the condition on the stability of P_1 and P_2 must be violated, i.e. P_1 stable and P_2 not stable or vice versa. In the following let $P \equiv x.\mathbf{0}$ with $x \in \mathbb{A}_\omega \setminus (\ell(P_1) \cup \ell(P_2))$.

First assume P_1 stable and P_2 not stable; then we have $\emptyset\emptyset x \in \mathsf{RT}(P_1 + P) \setminus \mathsf{RT}(P_2 + P)$, because $\emptyset\emptyset \in \mathsf{RT}(P) \cap \mathsf{RT}(P_1)$ for stable P and P_1 by Theorem 5.4.2 since $\emptyset\emptyset x \in \mathsf{RT}(P)$, but by Theorem 5.4.4, $\emptyset\emptyset x \notin \mathsf{RT}(P_2) + P$, since $\emptyset\emptyset x \notin \mathsf{RT}(P_2)$ by $x \notin \ell(P_2)$ and $\emptyset\emptyset x \notin \mathsf{RT}^0(P_2+P) \cup \mathsf{RT}^1(P_2+P)$ since $|\emptyset\emptyset| > 1$. Now $\mathsf{RT}(P_1+P) \not\subseteq \mathsf{RT}(P_2+P)$ is a contradiction to $P_1 + P \leq_r P_2 + P$.

Now assume P_1 not stable and P_2 stable; then $\emptyset\{x\} \in \mathsf{RT}(P_1 + P) \setminus \mathsf{RT}(P_2 + P)$, because P_1 not stable and $\emptyset\{x\} \in \mathsf{RT}(P_1)$ (observe Proposition 4.2.3), but P and P_2 stable and $\emptyset\{x\} \notin \mathsf{RT}(P)$. ■ 5.6

We finally aim to show that $\leq$ is also a precongruence for (guarded) recursion. Following [Hen88], we consider (initial) process terms as functions in the domain of (τ-)refusal-traces and will exploit their monotonicity w.r.t. $\leq$ and $\leq_\tau$, which essentially results from Theorem 5.5.

Definition 5.7

For closed substitutions $\mathcal{S}, \mathcal{S}' : \mathcal{X} \mapsto \mathbb{P}$ we write $\mathcal{S} \leq \mathcal{S}'$ if $\mathcal{S}(X) \leq \mathcal{S}'(X)$ for all $X \in \mathcal{X}$, and $\mathcal{S} \leq_\tau \mathcal{S}'$ if $\mathcal{S}(X) \leq_\tau \mathcal{S}'(X)$ for all $X \in \mathcal{X}$.

An initial process term $P \in \tilde{\mathbb{P}}_1$ is *monotonic*, if $[P]_\mathcal{S} \leq [P]_{\mathcal{S}'}$ whenever $\mathcal{S} \leq \mathcal{S}'$ for any closed *inital* substitutions $\mathcal{S}, \mathcal{S}' : \mathcal{X} \mapsto \mathbb{P}_1$. τ-monotonicity is defined analogously with $\leq_\tau$ instead of $\leq$.

For each $n \in \mathbb{N}$, $X \in \mathcal{X}$ and initial process term $P \in \tilde{\mathbb{P}}_1$ let P_X^n denote the initial process term defined inductively by $P_X^1 \equiv P$ and $P_X^{n+1} \equiv P\{P_X^n/X\} \equiv P_X^n\{P/X\}$. ∎ 5.7

Now $\mathsf{RT}_\tau(\mu X.P)$ is a fixpoint of the RT_τ-function defined by the initial process term P; furthermore, τ-monotonicity of this function carries over to its iterated applications, where the guardedness of X allows us to ignore up to a certain degree from the relation of the arguments:

Lemma 5.8

Let $P \in \tilde{\mathbb{P}}_1$ be a τ-monotonic initial process term and let $X \in \mathcal{X}$ be guarded in P. Furthermore, let $\mathcal{S}_1, \mathcal{S}_2 : \mathcal{X} \mapsto \mathbb{P}_1$ be closed inital substitutions with $S_1(Y) \leq_\tau S_2(Y)$ for all $Y \not\equiv X$. Then for all $n \in \mathbb{N}$:

1. $\mu X.P =_\tau P_X^n\{\mu X.P/X\}$.
2. $[P_X^n]_{\mathcal{S}_1} \leq_\tau^n [P_X^n]_{\mathcal{S}_2}$.

Proof:

In this proof let $\varepsilon \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_{\omega\tau}})$ and $v \in (\mathbb{A}_{\omega\tau} \cup 2^{\mathbb{A}_{\omega\tau}})^*$. Furthermore, for an initial closed substitution $\mathcal{S}$, variable $X \in \mathcal{X}$ and initial process $Q \in \mathbb{P}$ let $\mathcal{S} \xleftarrow{X} Q$ be the initial closed substitution that coincides with $\mathcal{S}$ in all variables except X, where it denotes Q. Similarly, let $\mathcal{S} - X$ coincide with $\mathcal{S}$ except for X, where it is X.

1. Let $\mathcal{S}$ be a closed substitution such that $[\mu X.P]_\mathcal{S} \in \mathbb{P}$; then $\mathcal{S}(Y) \in \mathbb{P}_1$ if $Y \in$ *free*$(\mu X.P)$ and $\mu X.P\{S(Y)/Y\} \equiv \mu X.P$ if $Y \notin$ *free*$(\mu X.P)$, hence w.l.o.g. we may assume $\mathcal{S}$ to be an initial closed substitution $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}_1$.

We perform induction on $n \in \mathbb{N}$, where in the base case we have by rule Rec_a or rule Rec_τ: $[\mu X.P]_\mathcal{S} \equiv \mu X.[P]_{\mathcal{S}-X} \xrightarrow{\varepsilon}_r R \xrightarrow{v}_r$ iff $[P]_{\mathcal{S}-X} \xrightarrow{\varepsilon}_r Q$ such that $R \equiv Q\{\mu X.[P]_{\mathcal{S}-X}/X\} \xrightarrow{v}_r$ iff $[P\{\mu X.P/X\}]_\mathcal{S} \equiv [P]_{\mathcal{S}-X}\{\mu X.[P]_{\mathcal{S}-X}/X\} \xrightarrow{\varepsilon}_r Q\{\mu X.[P]_{\mathcal{S}-X}/X\} \xrightarrow{v}_r$ by Proposition 4.2.5 since X guarded in P, hence in $[P]_{\mathcal{S}-X}$.

Now let the claim hold for some $n \in \mathbb{N}$ and let $\mathcal{S}_P^n$ denote $\mathcal{S} \xleftarrow{X} [P_X^n\{\mu X.P/X\}]_\mathcal{S}$, and let $\mathcal{S}_P$ denote $\mathcal{S} \xleftarrow{X} [\mu X.P]_\mathcal{S}$. Then $\mathcal{S}_P =_\tau \mathcal{S}_P^n$ by induction, hence $[P_X^{n+1}\{\mu X.P/X\}]_\mathcal{S} \equiv [P]_{\mathcal{S}-X}\{[P_X^n\{\mu X.P/X\}]_\mathcal{S}/X\} \equiv [P]_{\mathcal{S}_P^n} =_\tau [P]_{\mathcal{S}_P}$ by τ-monotonicity of P, and finally $[P]_{\mathcal{S}_P} \equiv [P\{\mu X.P/X\}]_\mathcal{S} =_\tau [\mu X.P]_\mathcal{S}$ is the base case again.

2. We first show for all $n \in \mathbb{N}$, that $|v| < n$ implies for any initial closed substitution $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}_1$ and any initial process terms $Q, R \in \tilde{\mathbb{P}}_1$: $[P^n_X]_{\mathcal{S}-X}\{Q/X\} \xrightarrow{v}_r R$ if and only if $[P^n_X]_{\mathcal{S}-X} \xrightarrow{v}_r P'$ for some $P' \in \tilde{\mathbb{P}}$ with guarded X, such that $R \equiv P'\{Q/X\}$.

In order to improve the readability, we first define $P^n \equiv [P^n_X]_{\mathcal{S}-X}$ for given P, $\mathcal{S}$ and $n \in \mathbb{N}$, and show $P^{n+1} \equiv P^n\{P^1/X\}$ for all $n \in \mathbb{N}$ by induction, where in the base case $n = 1$ we have $P^2 \equiv [P^2_X]_{\mathcal{S}-X} \equiv [P^1_X\{P^1_X/X\}]_{\mathcal{S}-X} \equiv [P^1_X]_{\mathcal{S}-X}\{[P^1_X]_{\mathcal{S}-X}/X\} \equiv P^1\{P^1/X\}$, hence let the claim hold for some $n \in \mathbb{N}$; then $P^{n+1} \equiv [P^{n+1}_X]_{\mathcal{S}-X} \equiv [P^n_X\{P^1_X/X\}]_{\mathcal{S}-X} \equiv [P^n_X]_{\mathcal{S}-X}\{[P^1_X]_{\mathcal{S}-X}/X\} \equiv P^n\{P^1/X\}$ by induction.

Using this, we now show the above property by induction on $n \in \mathbb{N}$, where for $n = 1$ we have $|v| < n$ iff $v = \lambda$ and $P^1\{Q/X\} \xrightarrow{\lambda}_r P^1\{Q/X\} \equiv R$ and $P^1 \xrightarrow{\lambda}_r P^1 \equiv P'$ with X guarded in P' by assumption and $R \equiv P'\{Q/X\}$. Hence let the claim hold for some $n \in \mathbb{N}$ and let $|v| < n$.

First let $P^{n+1}\{Q/X\} \equiv P^n\{(P^1\{Q/X\})/X\} \xrightarrow{v}_r R_1 \xrightarrow{\varepsilon}_r R_2$. Then by induction $P^n \xrightarrow{v}_r P'_1$ such that $R_1 \equiv P'_1\{(P^1\{Q/X\})/X\} \xrightarrow{\varepsilon}_r R_2$ for some P'_1 with guarded X, and from this again by induction also $P^{n+1} \equiv P^n\{P^1/X\} \xrightarrow{v}_r P'_1\{P^1/X\}$. Now Proposition 4.2.5 yields $R_1 \equiv P'_1\{(P^1\{Q/X\})/X\} \xrightarrow{\varepsilon}_r R_2$ only if $P'_1 \xrightarrow{\varepsilon}_r P'_2$ for some P'_2, such that $R_2 \equiv P'_2\{(P^1\{Q/X\})/X\} \equiv (P'_2\{P^1/X\})\{Q/X\}$, and again by Proposition 4.2.5, also $P'_1\{P^1/X\} \xrightarrow{\varepsilon}_r P'_2\{P^1/X\}$. Altogether, $P^{n+1} \xrightarrow{v\varepsilon}_r P'_2\{P^1/X\}$, such that $R_2 \equiv (P'_2\{P^1/X\})\{Q/X\}$, and since X is guarded in P, it is also guarded in $P'_2\{P^1/X\}$.

Now let $P^{n+1} \equiv P^n\{P^1/X\} \xrightarrow{v}_r P'_1 \xrightarrow{\varepsilon}_r P'_2$. Then by ind. $P^n \xrightarrow{v}_r P''_1$ for some P''_1 with guarded X, such that $P'_1 \equiv P''_1\{P^1/X\} \xrightarrow{\varepsilon}_r P'_2$, hence by induction also $P^{n+1}\{Q/X\} \equiv P^n\{(P^1\{Q/X\})/X\} \xrightarrow{v}_r P''_1\{(P^1\{Q/X\})/X\}$. Now by Proposition 4.2.5 we have $P'_1 \equiv P''_1\{P^1/X\} \xrightarrow{\varepsilon}_r P'_2$ only if $P''_1 \xrightarrow{\varepsilon}_r P''_2$ for some P''_2, such that $P'_2 \equiv P''_2\{P^1/X\}$, and then again by Proposition 4.2.5, $P''_1\{(P^1\{Q/X\})/X\} \xrightarrow{\varepsilon}_r P''_2\{(P^1\{Q/X\})/X\}$, too. Thus, $P^{n+1}\{Q/X\} \xrightarrow{v\varepsilon}_r P''_2\{(P^1\{Q/X\})/X\} \equiv (P''_2\{P^1/X\})\{Q/X\} \equiv P'_2\{Q/X\}$, and we are done showing the above property.

Now let $\mathcal{S}$ denote $\mathcal{S}_1 \overset{X}{\leftarrow} \mathcal{S}_2(X)$ and let $|v| < n$. Then by the above property $[P^n_X]_{\mathcal{S}_1} \equiv [P^n_X]_{\mathcal{S}_1-X}\{\mathcal{S}_1(X)/X\} \xrightarrow{v}_r R$ only if $[P^n_X]_{\mathcal{S}_1-X} \xrightarrow{v}_r P'$ such that $R \equiv P'\{\mathcal{S}_1(X)/X\}$ for some P' with guarded X, hence also $[P^n_X]_{\mathcal{S}_1-X}\{\mathcal{S}_2(X)/X\} \equiv [P^n_X]_{\mathcal{S}} \xrightarrow{v}_r P'\{\mathcal{S}_2(X)/X\}$, and it suffices to show $[P^n_X]_{\mathcal{S}} \leq_\tau [P^n_X]_{\mathcal{S}_2}$. We note that $\mathcal{S} \leq_\tau \mathcal{S}_2$ and perform induction on $n \in \mathbb{N}$, where in the base case $n = 1$ we have $[P]_{\mathcal{S}} \leq_\tau [P]_{\mathcal{S}_2}$ by the τ-monotonicity of P, hence let the claim hold for some $n \in \mathbb{N}$.

Let $\mathcal{S}'$ denote $\mathcal{S} \overset{X}{\leftarrow} [P^n_X]_{\mathcal{S}}$ and let $\mathcal{S}'_2$ denote $\mathcal{S}_2 \overset{X}{\leftarrow} [P^n_X]_{\mathcal{S}_2}$. Now by induction $\mathcal{S}' \leq_\tau \mathcal{S}'_2$, thus $[P^{n+1}_X]_{\mathcal{S}} \equiv [P\{[P^n_X]_{\mathcal{S}}/X\}]_{\mathcal{S}} \equiv [P]_{\mathcal{S}'} \leq_\tau [P]_{\mathcal{S}'_2} \equiv [P^{n+1}_X]_{\mathcal{S}_2}$ by the τ-monotonicity of P.
■ 5.8

We now can derive the precongruence property for τ-monotonic and monotonic initial process terms, where we use the fact that for all refusal traces $w \in \mathsf{RT}(\mu X.P)$ there is an underlying τ-refusal trace $v \in \mathsf{RT}_\tau(\mu X.P)$, such that $w = v/\tau$ and $|v| < n$ for some $n \in \mathbb{N}$:

Proposition 5.9

Let $P, Q \in \tilde{\mathbb{P}}_1$ be inital process terms that are both monotonic and τ-monotonic, and let $X \in \mathcal{X}$ be guarded in both P and Q.

1. $P \leq_\tau Q$ implies $\mu X.P \leq_\tau \mu X.Q$.
2. $P \leq Q$ implies $\mu X.P \leq \mu X.Q$.

Proof:

If $\mathcal{S}$ is a closed substitution such that $[\mu X.P]_\mathcal{S}, [\mu X.Q]_\mathcal{S} \in \mathbb{P}$, then – as in the previous proof – we may w.l.o.g. assume $\mathcal{S}$ to be an initial closed substitution $\mathcal{S} : \mathcal{X} \mapsto \mathbb{P}_1$.

1. Let $\mathcal{S}_1$ denote $\mathcal{S} \overset{X}{\leftarrow} [\mu X.P]_\mathcal{S}$ and let $\mathcal{S}_2$ denote $\mathcal{S} \overset{X}{\leftarrow} [\mu X.Q]_\mathcal{S}$. Now Lemma 5.8.1 yields $[\mu X.P]_\mathcal{S} =_\tau [P^n_X\{\mu X.P/X\}]_\mathcal{S} \equiv [P^n_X]_{\mathcal{S}_1}$ and $[Q^n_X]_{\mathcal{S}_2} \equiv [Q^n_X\{\mu X.Q/X\}]_\mathcal{S} =_\tau [\mu X.Q]_\mathcal{S}$ for all $n \in \mathbb{N}$, hence it suffices to show $[P^n_X]_{\mathcal{S}_1} \leq^n_\tau [Q^n_X]_{\mathcal{S}_2}$ for all $n \in \mathbb{N}$. Now $[P^n_X]_{\mathcal{S}_1} \leq^n_\tau [P^n_X]_{\mathcal{S}_2}$ by Lemma 5.8.2, since $\mathcal{S}_1(Y) \leq_\tau \mathcal{S}_2(Y)$ for all $Y \not\equiv X$, hence it suffices to show $[P^n_X]_{\mathcal{S}_2} \leq_\tau [Q^n_X]_{\mathcal{S}_2}$ for all $n \in \mathbb{N}$.

We perform induction on $n \in \mathbb{N}$, where for $n = 1$ we have $[P]_{\mathcal{S}_2} \leq_\tau [Q]_{\mathcal{S}_2}$ by $P \leq_\tau Q$, hence assume the claim to hold for some $n \in \mathbb{N}$. Then $[P^{n+1}_X]_{\mathcal{S}_2} \equiv [P\{[P^n_X]_{\mathcal{S}_2}/X\}]_{\mathcal{S}_2} \leq_\tau [P\{[Q^n_X]_{\mathcal{S}_2}/X\}]_{\mathcal{S}_2}$ by induction and τ-monotonicity of P, and finally $[P\{[Q^n_X]_{\mathcal{S}_2}/X\}]_{\mathcal{S}_2} \leq_\tau [Q\{[Q^n_X]_{\mathcal{S}_2}/X\}]_{\mathcal{S}_2} \equiv [Q^{n+1}_X]_{\mathcal{S}_2}$ by $P \leq_\tau Q$ again.

2. We first show $[P^n_X]_\mathcal{S} \leq [Q^n_X]_\mathcal{S}$ for all $n \in \mathbb{N}$ and any initial closed substitution $\mathcal{S}$. Let $\mathcal{S}^n_P$ denote $\mathcal{S} \overset{X}{\leftarrow} [P^n_X]_\mathcal{S}$ and let $\mathcal{S}^n_Q$ denote $\mathcal{S} \overset{X}{\leftarrow} [Q^n_X]_\mathcal{S}$. Now it suffices to show $\mathcal{S}^n_P \leq \mathcal{S}^n_Q$ for all $n \in \mathbb{N}$, where in turn it is enough to show $\mathcal{S}^n_P(X) \leq \mathcal{S}^n_Q(X)$, since $\mathcal{S}^n_P(Y) \equiv \mathcal{S}(Y) \equiv \mathcal{S}^n_Q(Y)$ for all $Y \not\equiv X$.

We perform induction on $n \in \mathbb{N}$, where in the base case $n = 1$ we have $\mathcal{S}^1_P(X) \equiv [P]_\mathcal{S} \leq [Q]_\mathcal{S} \equiv \mathcal{S}^1_Q(X)$ since $P \leq Q$ by assumption, hence let $\mathcal{S}^n_P \leq \mathcal{S}^n_Q$ for some $n \in \mathbb{N}$.

Then $\mathcal{S}^{n+1}_P(X) \equiv [P^{n+1}_X]_\mathcal{S} \equiv [P\{[P^n_X]_\mathcal{S}/X\}]_\mathcal{S} \equiv [P]_{\mathcal{S}^n_P} \leq [P]_{\mathcal{S}^n_Q}$ by induction and monotonicity of P, and $[P]_{\mathcal{S}^n_Q} \leq [Q]_{\mathcal{S}^n_Q} \equiv \mathcal{S}^{n+1}_Q(X)$ by $P \leq Q$ again. We conclude $[P^n_X]_\mathcal{S} \leq [Q^n_X]_\mathcal{S}$ for all $n \in \mathbb{N}$.

Now take some $w \in \mathsf{RT}([\mu X.P]_\mathcal{S})$. Then $w = v/\tau$ for some $v \in \mathsf{RT}_\tau([\mu X.P]_\mathcal{S}) = \mathsf{RT}_\tau([P^{|v|+1}_X\{\mu X.P/X\}]_\mathcal{S})$ by Lemma 5.8.1, and also $v \in \mathsf{RT}_\tau([P^{|v|+1}_X\{\mu X.Q/X\}]_\mathcal{S})$ by Lemma 5.8.2, hence $w \in \mathsf{RT}([P^{|v|+1}_X\{\mu X.Q/X\}]_\mathcal{S}) \subseteq \mathsf{RT}([Q^{|v|+1}_X\{\mu X.Q/X\}]_\mathcal{S})$, because $[P^{|v|+1}_X\{[\mu X.Q]_\mathcal{S}/X\}]_\mathcal{S} \leq [Q^{|v|+1}_X\{[\mu X.Q]_\mathcal{S}/X\}]_\mathcal{S}$ by the above. Finally, by Lemma 5.8.1 again, we have $w \in \mathsf{RT}([Q^{|v|+1}_X\{\mu X.Q/X\}]_\mathcal{S}) = \mathsf{RT}([\mu X.Q]_\mathcal{S})$, since RT_τ-equivalence implies RT-equivalence. Note that the τ-monotonicity of both P and Q was necessary for the application of Lemma 5.8.

Finally, $\mathcal{A}([\mu X.P]_\mathcal{S}) = \mathcal{A}([P]_\mathcal{S})$ and $\mathcal{A}([\mu X.Q]_\mathcal{S}) = \mathcal{A}([Q]_\mathcal{S})$ by Proposition 2.5.2 since X guarded in P and Q, hence $\tau \in \mathcal{A}([\mu X.P]_\mathcal{S})$ iff $\tau \in \mathcal{A}([P]_\mathcal{S})$ iff $\tau \in \mathcal{A}([Q]_\mathcal{S})$, because $P \leq Q$ iff $\tau \in \mathcal{A}([\mu X.Q]_\mathcal{S})$. Thus $[\mu X.P]_\mathcal{S}$ stable iff $[\mu X.Q]_\mathcal{S}$ stable. ■ 5.9

By showing the τ-monotonicity and monotonicity of all initial process terms by induction on the term structure using Lemma 5.8 and Theorem 5.5, we end up with the desired result:

Theorem 5.10

Both $\leq_\tau$ and $\leq$ are precongruences for recursion.

Proof:

By Proposition 5.9 it suffices to show that all $P \in \tilde{\mathbb{P}}_1$ are both monotonic and τ-monotonic. We perform induction on the structure of P, where all cases except recursion are covered by Theorem 5.5, hence assume $P \in \tilde{\mathbb{P}}_1$ to be monotonic and τ-monotonic by induction, $X \in \mathcal{X}$ be guarded in P and consider $\mu X.P$. Let $\mathcal{S}_1, \mathcal{S}_2$ be inital closed substitutions, let $\mathcal{S}_1'$ denote $\mathcal{S}_1 \overset{X}{\leftarrow} [\mu X.P]_{\mathcal{S}_1}$, and let $\mathcal{S}_2'$ denote $\mathcal{S}_2 \overset{X}{\leftarrow} [\mu X.P]_{\mathcal{S}_2}$.

We first show the τ-monotonicity of $\mu X.P$ and assume $\mathcal{S}_1 \leq_\tau \mathcal{S}_2$ in this case. Then $\mathcal{S}_1'(Y) \leq_\tau \mathcal{S}_2'(Y)$ for all $Y \not\equiv X$, hence $[\mu X.P]_{\mathcal{S}_1} =_\tau [P_X^n]_{\mathcal{S}_1'} \leq_\tau^n [P_X^n]_{\mathcal{S}_2'} =_\tau [\mu X.P]_{\mathcal{S}_2}$ for all $n \in \mathbb{N}$ by Lemma 5.8.1, .2 and .1 again, hence we are done.

We now show the monotonicity of $\mu X.P$ and assume $\mathcal{S}_1 \leq \mathcal{S}_2$ in this case. Furthermore, let $\mathcal{S}_1''$ denote $\mathcal{S}_1' \overset{X}{\leftarrow} \mathcal{S}_2'(X)$; then $\mathcal{S}_1'(Y) \leq_\tau \mathcal{S}_1''(Y)$ for all $Y \not\equiv X$, and $\mathcal{S}_1'' \leq \mathcal{S}_2'$.

Now take some $w \in \mathsf{RT}([\mu X.P]_{\mathcal{S}_1})$. Then $w = v/\tau$ for some $v \in \mathsf{RT}_\tau([\mu X.P]_{\mathcal{S}_1}) = \mathsf{RT}_\tau([P_X^{|v|+1}\{\mu X.P/X\}]_{\mathcal{S}_1}) = \mathsf{RT}_\tau([P_X^{|v|+1}]_{\mathcal{S}_1'})$ by Lemma 5.8.1, and by the above and Lemma 5.8.2 we also have $v \in \mathsf{RT}_\tau([P_X^{|v|+1}]_{\mathcal{S}_1''})$, hence $w \in \mathsf{RT}([P_X^{|v|+1}]_{\mathcal{S}_1''})$. Now we have $\mathsf{RT}([P_X^{|v|+1}]_{\mathcal{S}_2'}) = \mathsf{RT}([P_X^{|v|+1}\{\mu X.P/X\}]_{\mathcal{S}_2}) = \mathsf{RT}([\mu X.P]_{\mathcal{S}_2})$ by Lemma 5.8.1 and since RT_τ-equivalence implies RT-equivalence, hence it remains to show $[P_X^n]_{\mathcal{S}_1''} \leq [P_X^n]_{\mathcal{S}_2'}$ for all $n \in \mathbb{N}$:

We perform induction on $n \in \mathbb{N}$, where in the base case $n = 1$ we have $[P]_{\mathcal{S}_1''} \leq [P]_{\mathcal{S}_2'}$ by $\mathcal{S}_1'' \leq \mathcal{S}_2'$ and the monotonicity of P, hence assume $[P_X^n]_{\mathcal{S}_1''} \leq [P_X^n]_{\mathcal{S}_2'}$ for some $n \in \mathbb{N}$: then $[P_X^{n+1}]_{\mathcal{S}_1''} \equiv [P\{[P_X^n]_{\mathcal{S}_1''}/X\}]_{\mathcal{S}_1''} \leq [P\{[P_X^n]_{\mathcal{S}_2'}/X\}]_{\mathcal{S}_2'} \equiv [P_X^{n+1}]_{\mathcal{S}_2'}$ by induction and monotonicity of P again.

Finally, $\mathcal{A}([\mu X.P]_{\mathcal{S}_1}) = \mathcal{A}([P]_{\mathcal{S}_1})$ and $\mathcal{A}([\mu X.P]_{\mathcal{S}_2}) = \mathcal{A}([P]_{\mathcal{S}_2})$ by Proposition 2.5.2 since X guarded in P, hence $\tau \in \mathcal{A}([\mu X.P]_{\mathcal{S}_1})$ iff $\tau \in \mathcal{A}([P]_{\mathcal{S}_1})$ iff $\tau \in \mathcal{A}([P]_{\mathcal{S}_2})$ since $[P]_{\mathcal{S}_1} \leq [P]_{\mathcal{S}_2}$ iff $\tau \in \mathcal{A}([\mu X.P]_{\mathcal{S}_2})$ Thus $[\mu X.P]_{\mathcal{S}_1}$ stable iff $[\mu X.P]_{\mathcal{S}_2}$ stable. ■ 5.10

6 Related Work

In the literature, several approaches to efficiency preorders have been proposed, from which only representative samples can be considered here.

For untimed CCS-like terms, efficiency preorders based on testing have been investigated in [CZ91] and [NC96a], and bisimulation-based ones in [AKH92] and [AKN95]; in all these approaches, efficiency is measured by counting internal actions, where runs of a parallel composition are seen to be the interleaved runs of the components; consequently, in all cases, $\tau.a\|_{\{a\}}\tau.a$ is as efficient as $\tau.\tau.a$, whereas in our setting $\tau.a\|_{\{a\}}\tau.a$ is strictly faster than $\tau.\tau.a$.

TPL is a CCS-based discretely timed process algebra developed in [HR95], where systems are also related via a must-testing approach. In [NC96b], the resulting preorder is interpreted as to relate systems w.r.t. their temporal and functional 'predictability' rather than efficiency. Systems in TPL can be considered as synchronous, since maximal progress is forced in test application. This gives the test environment more direct control over the temporal behaviour than in our setting; as a consequence, no time bounds are needed for tests. By this, TPL can also be seen as a discrete part of the continuously timed process algebra TimedCSP (cf. [Sch95]), where e.g. the discrete time unit σ is replaced by *WAIT 1* constructs.

In the discretely timed algebra ℓTCCS of [MT91], components may have arbitrary relative speeds, but there is no progress assumption at all and the efficiency preorder is based on a sort of bisimulation; an interpretation in terms of worst-case behaviour is not obvious. [CGR95] gives a different bisimulation based approach, where component speeds are fixed with respect to local clocks (modulo patience for communication in [Cor98]). Here, the operational semantics realizes local passage of time, hence this idea is hard to compare to our approach or any other.

[Bur92] discusses how (the more realistic) continuously timed behaviour can be approximated with discretely timed behaviour; the aim is to ensure that each implementation in the discrete view is indeed an implementation in the continuous view (but not necessarily vice versa). There is no result showing that discrete time gives *complete* information as in our setting.

References

[AKH92] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29:737–760, 1992.

[AKN95] S. Arun-Kumar and V. Natarajan. Conformance: a precongruence close to bisimilarity. In J. Desel, editor, *Structures in Concurrency Theory*, Worksh. in Computing, 55–68. Springer, 1995.

[BD91] E. Badouel and Ph. Darondeau. On guarded recursion. *TCS*, 81:403–408, 1991.

[Bur92] Jerry R. Burch. *Trace Algebra for Automatic Verification of Real-Time Concurrent Systems.* PhD thesis, School of Comp. Sci., Carnegie Mellon Univ., 1992.

[CGR95] F. Corradini, R. Gorrieri, and M. Roccetti. Performance preorder: ordering processes with respect to speed. In J. Wiedermann and P. Hajek, editors, *MFCS '95*, Lect. Notes Comp. Sci. 969, 444–453. Springer, 1995.

[Cor98] Flavio Corradini. On the coarsest congruence within global-clock-bounded equivalence. *TCS*, 1998. (to appear).

[CZ91] R. Cleaveland and A. Zwarico. A theory of testing for real-time. In *Proc. 6th Symp. on Logic in Computer Science*, pages 110–119. IEEE Computer Society Press, 1991.

[DNH84] R. De Nicola and M.C.B. Hennessy. Testing equivalence for processes. *TCS*, 34:83–133, 1984.

[Hen88] M. Hennessy. *An algebraic theory of processes.* the M.I.T. Press, 1988.

[HR95] M. Hennessy and T. Regan. A process algebra for timed systems. *Information and Computation*, 117:221–239, 1995.

[Jen96] L. Jenner. Modular construction of fast asynchronous systems. Technical Report 1996-2, Inst. f. Informatik, Univ. Augsburg, 1996. Available at `http://www1.Informatik.Uni-Augsburg.DE/~jenner/`.

[JV95] L. Jenner and W. Vogler. Fast asynchronous systems in dense time. Technical Report Nr. 344, Inst. f. Mathematik, Univ. Augsburg, 1995. Available at

`http://www1.Informatik.Uni-Augsburg.DE/~jenner/`.
Extended abstract appeared in Proc. ICALP 96.

[Lyn96] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Francisco, 1996.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[MT91] F. Moller and C. Tofts. Relating processes with respect to speed. In J. Baeten and J. Groote, editors, *CONCUR '91*, Lect. Notes Comp. Sci. 527, 424–438. Springer, 1991.

[NC96a] V. Natarajan and R. Cleaveland. An algebraic theory of process efficiency. In *Proc. LICS'96*. 11th Annual IEEE Symposium, 1996.

[NC96b] V. Natarajan and R. Cleaveland. Predictability of real-time systems: A process-algebraic approach. In *Proc. RTSS'96*. 17th IEEE Real-Time Systems Symposium, 1996.

[Sch95] S. Schneider. An operational semantics for timed CSP. *Information and Computation*, 116(2), pp.193-213, 1995.

[Vog95] W. Vogler. Timed testing of concurrent systems. *Information and Computation*, 121:149–171, 1995.