NASA/CR-2001-210651
ICASE Report No. 2001-2

A Faster-than Relation for Asynchronous Processes

Gerald Lüttgen
The University of Sheffield, Sheffield, United Kingdom

Walter Vogler
Universität Augsburg, Augsburg, Germany

ICASE
NASA Langley Research Center
Hampton, Virginia

Operated by Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

January 2001

Form SF298 Citation Data

Report Date *("DD MON YYYY")* 00JAN2001	**Report Type** N/A	**Dates Covered (from... to)** *("DD MON YYYY")*

Title and Subtitle A Faster-than Relation for Asynchronous Processes	**Contract or Grant Number**
	Program Element Number
Authors Gerald Lüttgen, Walter Vogler	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) ICASE NASA Langley Research Center Hampton, Virginia	**Performing Organization Number(s)** ICASE Report No. 2001-2
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
NASA/CR-2001-210651

Abstract
Abstract. This paper introduces a novel (bi)simulation{based faster{than preorder which relates asyn-chronous processes with respect to their worst{case timing behavior. The studies are conducted for a conservative extension of the process algebra CCS, called TACS, which permits the speci cation of maximal time bounds of actions. TACS complements work in plain process algebras which compares asynchronous processes with respect to their functional reactive behavior only, and in timed process algebras which focus on analyzing synchronous processes. The most unusual contribution of this paper is in showing that the pro- posed faster{than preorder coincides with two other and at least equally appealing preorders, one of which considers the absolute times at which actions occur in system runs. The paper also develops the semantic theory of TACS: itcharacterizes the largest precongruence contained in the faster{than preorder, presents an axiomatization in a fragment of the algebra, and investigates a corresponding weak faster{than preorder. A small example relating two implementations of a simple storage system testi es to the practical utility of the new theory.

Subject Terms

Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 31	

A FASTER–THAN RELATION FOR ASYNCHRONOUS PROCESSES*

GERALD LÜTTGEN† AND WALTER VOGLER‡

Abstract. This paper introduces a novel (bi)simulation–based faster–than preorder which relates asynchronous processes with respect to their worst–case timing behavior. The studies are conducted for a conservative extension of the process algebra CCS, called TACS, which permits the specification of maximal time bounds of actions. TACS complements work in plain process algebras which compares asynchronous processes with respect to their functional reactive behavior only, and in timed process algebras which focus on analyzing synchronous processes. The most unusual contribution of this paper is in showing that the proposed faster–than preorder coincides with two other and at least equally appealing preorders, one of which considers the absolute times at which actions occur in system runs. The paper also develops the semantic theory of TACS: it characterizes the largest precongruence contained in the faster–than preorder, presents an axiomatization in a fragment of the algebra, and investigates a corresponding weak faster–than preorder. A small example relating two implementations of a simple storage system testifies to the practical utility of the new theory.

Key words. asynchronous systems, bisimulation, faster–than preorder, process algebra, timing behavior

Subject classification. Computer Science

1. Introduction. *Process algebras* [7, 8, 18, 21, 26] provide a widely studied framework for reasoning about the behavior of concurrent systems. Early approaches, including Milner's *Calculus of Communicating Systems* (CCS) [26], focused on semantic issues of asynchronous processes, where the relative speeds between processes running in parallel are not bounded, i.e., one process may be arbitrarily slower or faster than another. This leads to a simple and mathematically elegant semantic theory analyzing the functional behavior of systems regarding their causal interactions with their environments. To include time as an aspect of system behavior, *timed process algebras* [5, 19, 28, 32, 34, 38] were introduced. They usually model synchronous systems where processes running in parallel are under the regime of a common global clock and have a fixed speed. A well–known representative of discrete timed process algebras is Hennessy and Regan's Timed Process Language (TPL) [19] which extends CCS by a timeout operator and a clock prefix demanding that exactly one time unit *must* pass before activating the argument process. Research papers on timed process algebras usually do not relate processes with respect to speed; the most notable exception is work by Moller and Tofts [29] which considers a faster–than preorder within a CCS–based setting, where processes are essentially attached with lower time bounds [28]. In practice, however, often upper time bounds are known to a system designer, determining how long a process may delay its execution. These can be used to compare the *worst–case timing behavior* of processes. The assumption of upper time bounds for *asynchronous* processes already is exploited in distributed algorithms [24] and was investigated by the second author in the

*This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1–97046 while the authors were in residence at ICASE, NASA Langley Research Center, Hampton, Virginia 23681–2199, USA.

†Department of Computer Science, The University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP, U.K., e–mail: g.luettgen@dcs.shef.ac.uk.

‡Institut für Informatik, Universität Augsburg, D–86135 Augsburg, Germany, e–mail: vogler@informatik.uni-augsburg.de.

setting of Petri nets [9, 22, 35, 36]. The latter work adapted DeNicola and Hennessy's notion of testing [16], where the derived must–preorder is interpreted as faster–than relation. Recently, these results have been transferred to a process–algebraic setting [23, 37] whose semantics, however, is still based on testing.

In this paper we develop a novel (bi)simulation–based approach to compare asynchronous systems with respect to their worst–case timing behavior. To do so, we extend CCS by a rather specific notion of clock prefixing "$\sigma.$", where σ stands for one time unit or a single clock tick. In contrast to TPL, we interpret $\sigma.P$ as a process which *may* delay *at most* one time unit before executing P. Similar to TPL, however, we view the occurrence of actions as instantaneous. This results in a new process algebra extending CCS, to which we refer as *Timed Asynchronous Communicating Systems* (TACS). To make our intuition of upper–bound delays more precise, consider the processes $\sigma.a.\mathbf{0}$ and $a.\mathbf{0}$, where a denotes an action or port as in CCS. While the former process may delay an enabled communication on port a by one time unit, the latter process must engage in the communication. In this sense, action a is *non–urgent* in $\sigma.a.\mathbf{0}$ but *urgent* in $a.\mathbf{0}$. However, if a communication on port a is not enabled, then process $a.\mathbf{0}$ may wait until some communication partner is ready. Technically, we allow $a.P$ to wait in any case; to enforce a communication resulting in the internal action τ, a time step in TACS is preempted by an urgent τ. This is similar to timed process algebras employing the *maximal progress assumption* [19, 38]; however, in these algebras and in contrast to TACS, any internal computation is considered to be urgent. For TACS we introduce a (bi)simulation–based *faster–than preorder* which exploits the knowledge of upper time bounds: a process is faster than another if both are linked by a relation which is a strong bisimulation for actions and a simulation for time steps.

The main contribution of this paper is the formal underpinning of our preorder which justifies why it is a good candidate for a faster–than relation on processes. There are at least two very appealing alternative definitions for such a preorder. First, one could allow the slower process to perform extra time steps when simulating an action or time step of the faster process. Second and probably even more important is the question of how exactly the faster process can match a time step and the subsequent behavior of the slower one. For illustrating this issue, consider the runs $a\sigma\sigma b$ and $\sigma a\sigma b$ which might be exhibited by some processes. One can argue that the first run is faster than the second one since action a occurs earlier in the run and since action b occurs at absolute time 2 in both runs, measured from the start of each run. With this observation in mind, we define a second variant of our faster–than preorder, where a time step of the slower process is either simulated immediately by the faster one or might be performed later on. As a main result, we prove that both variants coincide with our faster–than preorder that has a more elegant and concise definition. This justifies our faster–than preorder as a reference preorder for relating asynchronous processes with respect to their worst–case timing behavior. In addition, this paper develops the semantic theory of the faster–than preorder: we characterize the coarsest precongruence contained in our preorder, demonstrate that TACS with this precongruence is a conservative extension of CCS with bisimulation, and axiomatize our precongruence for finite sequential processes. We also study the corresponding weak faster–than preorder, which abstracts from internal computation, and its semantic theory. To testify to the utility of our novel framework, we apply it to a small example dealing with two implementations of a simple storage system.

The remainder of this paper is organized as follows. The next section presents the process algebra TACS, while Sec. 3 introduces three variants of a faster–than preorder and shows all of them to coincide. Sec. 4 develops the semantic theory of our preorder and its "weak" correspondence, which is then applied to an example in Sec. 5. Finally, Secs. 6 and 7 discuss related work and present our conclusions, respectively.

2. Timed Asynchronous Communicating Systems. This section defines the syntax and semantics of our novel process algebra *Timed Asynchronous Communicating Systems* (TACS) which conservatively extends CCS [26] by a concept of global, discrete time. This concept is introduced by a non–standard interpretation of clock prefixing "$\sigma.$" as mentioned in the introduction. Intuitively, a process $\sigma.P$ *can at most* (but must not) delay one time unit before having to execute process P, provided that P can engage in a communication with the environment or in some internal computation. The semantics of TACS is based on a notion of transition system that involves two kinds of transitions, *action transitions* and *clock transitions*. Action transitions, like in CCS, are local handshake communications in which two processes may synchronize to take a joint state change together. A clock represents the progress of time, which manifests itself in a recurrent global synchronization event, the clock transition. As indicated above, action and clock transitions are not orthogonal concepts, since a clock transition can only occur if the process under consideration cannot engage in an urgent internal computation.

Syntax of TACS. Let Λ be a countable set of actions, or ports, not including the distinguished unobservable, *internal* action τ. With every $a \in \Lambda$ we associate a *complementary action* $\overline{a}$. We define $\overline{\Lambda} =_{\rm df} \{\overline{a} \,|\, a \in \Lambda\}$ and take $\mathcal{A}$ to denote the set $\Lambda \cup \overline{\Lambda} \cup \{\tau\}$ of all actions. Complementation is lifted to $\Lambda \cup \overline{\Lambda}$ by defining $\overline{\overline{a}} =_{\rm df} a$. As in CCS [26], an action a communicates with its complement $\overline{a}$ to produce the internal action τ. We let $a, b, \ldots$ range over $\Lambda \cup \overline{\Lambda}$ and $\alpha, \beta, \ldots$ over $\mathcal{A}$ and, moreover, we represent (potential) clock ticks by the symbol σ. The syntax of our language is then defined as follows:

$$P \quad ::= \quad \mathbf{0} \;\mid\; x \;\mid\; \alpha.P \;\mid\; \sigma.P \;\mid\; P + P \;\mid\; P|P \;\mid\; P \setminus L \;\mid\; P[f] \;\mid\; \mu x.P$$

where x is a *variable* taken from a countably infinite set $\mathcal{V}$ of variables, $L \subseteq \mathcal{A} \setminus \{\tau\}$ is a *restriction set*, and $f : \mathcal{A} \to \mathcal{A}$ is a *finite relabeling*. A finite relabeling satisfies the properties $f(\tau) = \tau$, $f(\overline{a}) = \overline{f(a)}$, and $|\{\alpha \,|\, f(\alpha) \neq \alpha\}| < \infty$. The set of all terms is abbreviated by $\widehat{\mathcal{P}}$ and, for convenience, we define $\overline{L} =_{\rm df} \{\overline{a} \,|\, a \in L\}$. Moreover, we use the standard definitions for the semantic *sort* $\mathsf{sort}(P) \subseteq \Lambda \cup \overline{\Lambda}$ of some term P, *free* and *bound* variables (where μx binds x), *open* and *closed* terms, and *contexts* (terms with a "hole"). A variable is called *guarded* in a term if each occurrence of the variable is in the scope of an action prefix. Moreover, we require for terms of the form $\mu x.P$ that x is guarded in P. We refer to closed and guarded terms as *processes*, with the set of all processes written as $\mathcal{P}$, and denote syntactic equality by $\equiv$.

Semantics of TACS. The *operational semantics* of a TACS term $P \in \widehat{\mathcal{P}}$ is given by a labeled transition system $\langle \widehat{\mathcal{P}}, \mathcal{A} \cup \{\sigma\}, \longrightarrow, P \rangle$ where $\widehat{\mathcal{P}}$ is the set of states, $\mathcal{A} \cup \{\sigma\}$ the alphabet, $\longrightarrow \subseteq \widehat{\mathcal{P}} \times \mathcal{A} \cup \{\sigma\} \times \widehat{\mathcal{P}}$ the transition relation, and P the start state. Before we proceed, it is convenient to introduce sets $\mathcal{U}(P)$, for all terms $P \in \widehat{\mathcal{P}}$, which include the *urgent actions*, as discussed in the introduction, in which P can initially engage. These sets are inductively defined along the structure of P, as shown in Table 2.1. Strictly speaking, $\mathcal{U}(P)$ does not necessarily contain *all* urgent actions. For example, for $P = \tau.0 + \sigma.a.\mathbf{0}$ we have $\mathcal{U}(P) = \{\tau\}$, although action a is also urgent, because the clock transition of P is preempted according to our notion of maximal progress. However, in the sequel we need the urgent action set of P only for determining whether P can initially perform an urgent τ. For this purpose, our syntactic definition of urgent action sets is just fine since $\tau \in \mathcal{U}(P)$ if and only if τ is urgent in P.

Now, the operational semantics for action transitions and clock transitions can be defined via *structural operational rules* which are displayed in Tables 2.2 and 2.3, respectively. For action transitions, the rules are exactly the same as for CCS, with the exception of our new clock–prefix operator. For clock transitions,

TABLE 2.1

Urgent action sets

$\mathcal{U}(\mathbf{0})$	$=_{\mathrm{df}} \emptyset$	$\mathcal{U}(x)$	$=_{\mathrm{df}} \emptyset$	$\mathcal{U}(P \setminus L)$	$=_{\mathrm{df}} \mathcal{U}(P) \setminus (L \cup \overline{L})$
$\mathcal{U}(\alpha.P)$	$=_{\mathrm{df}} \{\alpha\}$	$\mathcal{U}(P+Q)$	$=_{\mathrm{df}} \mathcal{U}(P) \cup \mathcal{U}(Q)$	$\mathcal{U}(P[f])$	$=_{\mathrm{df}} \{f(\alpha) \mid \alpha \in \mathcal{U}(P)\}$
$\mathcal{U}(\sigma.P)$	$=_{\mathrm{df}} \emptyset$	$\mathcal{U}(P\|Q)$	$=_{\mathrm{df}} \mathcal{U}(P) \cup \mathcal{U}(Q) \cup \{\tau \mid \mathcal{U}(P) \cap \overline{\mathcal{U}(Q)} \neq \emptyset\}$	$\mathcal{U}(\mu x.P)$	$=_{\mathrm{df}} \mathcal{U}(P)$

TABLE 2.2

Operational semantics for TACS *(action transitions)*

$$\text{Act}\quad \frac{}{\alpha.P \xrightarrow{\alpha} P} \qquad \text{Pre}\quad \frac{P \xrightarrow{\alpha} P'}{\sigma.P \xrightarrow{\alpha} P'}$$

$$\text{Sum1}\quad \frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'} \qquad \text{Sum2}\quad \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$$

$$\text{Com1}\quad \frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \text{Com2}\quad \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \qquad \text{Com3}\quad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

$$\text{Rel}\quad \frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]} \qquad \text{Res}\quad \frac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L}\ \alpha \notin L \cup \overline{L} \qquad \text{Rec}\quad \frac{P \xrightarrow{\alpha} P'}{\mu x.P \xrightarrow{\alpha} P'[\mu x.P/x]}$$

our semantics is set up such that, if $\tau \in \mathcal{U}(P)$, then a clock tick σ of P is inhibited, in accordance with our adapted variant of maximal progress. For the sake of simplicity, let us write $P \xrightarrow{\gamma} P'$ instead of $\langle P, \gamma, P' \rangle \in \longrightarrow$, for $\gamma \in \mathcal{A} \cup \{\sigma\}$, and say that P *may engage in* γ *and thereafter behave like* P'. Sometimes it is also convenient to write $P \xrightarrow{\gamma}$ for $\exists P'.\, P \xrightarrow{\gamma} P'$.

According to our operational rules, the *action–prefix* term $\alpha.P$ may engage in action α and then behave like P. If $\alpha \neq \tau$, then it may also *idle*, i.e., engage in a clock transition to itself, as process $\mathbf{0}$ does. The *clock–prefix* term $\sigma.P$ can engage in a clock transition to P and, additionally, it can perform any action transition that P can since σ represents a delay of *at most* one time unit. The *summation operator* $+$ denotes nondeterministic choice such that $P+Q$ may behave like P or Q. Time has to proceed equally on both sides of summation, whence $P+Q$ can engage in a clock transition and delay the nondeterministic choice if and only if both P and Q can. As a consequence, e.g., process $\sigma.a.\mathbf{0} + \tau.\mathbf{0}$ cannot engage in a clock transition; in particular, a is not urgent, but nevertheless it has to occur without delay if it occurs at all. The *restriction operator* $\setminus L$ prohibits the execution of actions in $L \cup \overline{L}$ and, thus, permits the scoping of actions. $P[f]$ behaves exactly as P where actions are renamed by the *relabeling* f. The term $P|Q$ stands for the *parallel composition* of P and Q according to an interleaving semantics with synchronized communication on complementary actions resulting in the internal action τ. Again, time has to proceed equally on both sides of the operator. The side condition ensures that $P|Q$ can only progress on σ, if it cannot engage in any urgent internal computation, in accordance with our notion of maximal progress. Finally, $\mu x.\, P$ denotes *recursion*, i.e., $\mu x.\, P$ behaves as a distinguished solution of the equation $x = P$.

The operational semantics for TACS possesses several important properties, in analogy to many temporal process algebras [19, 38]. First, it is *time–deterministic*, i.e., processes react deterministically to clock ticks, reflecting the intuition that progress of time does not resolve choices. Formally, $P \xrightarrow{\sigma} P'$ and $P \xrightarrow{\sigma} P''$ implies $P' \equiv P''$, for all $P, P', P'' \in \widehat{\mathcal{P}}$. Second, according to our variant of *maximal progress*, a term P can

TABLE 2.3
Operational semantics for TACS *(clock transitions)*

tNil	$\dfrac{-}{\mathbf{0} \xrightarrow{\sigma} \mathbf{0}}$	tSum	$\dfrac{P \xrightarrow{\sigma} P' \quad Q \xrightarrow{\sigma} Q'}{P + Q \xrightarrow{\sigma} P' + Q'}$	tRes	$\dfrac{P \xrightarrow{\sigma} P'}{P \setminus L \xrightarrow{\sigma} P' \setminus L}$
tAct	$\dfrac{-}{a.P \xrightarrow{\sigma} a.P}$	tCom	$\dfrac{P \xrightarrow{\sigma} P' \quad Q \xrightarrow{\sigma} Q'}{P \mid Q \xrightarrow{\sigma} P' \mid Q'}\ \tau \notin \mathcal{U}(P \mid Q)$	tRel	$\dfrac{P \xrightarrow{\sigma} P'}{P[f] \xrightarrow{\sigma} P'[f]}$
tPre	$\dfrac{-}{\sigma.P \xrightarrow{\sigma} P}$	tRec	$\dfrac{P \xrightarrow{\sigma} P'}{\mu x.P \xrightarrow{\sigma} P'[\mu x.P/x]}$		

engage in a clock transition exactly if it cannot engage in an urgent internal transition. Formally, $P \xrightarrow{\sigma}$ if and only if $\tau \notin \mathcal{U}(P)$, for all $P \in \widehat{\mathcal{P}}$.

We conclude this section by two simple lemmas which will be used in the next sections. The first one highlights the implications of guardedness in our calculus. As with the abovementioned properties of time determinism and maximal progress, it can be proved via induction on the structure of P.

LEMMA 2.1. *Let $P, P', Q \in \widehat{\mathcal{P}}$, let $x \in \mathcal{V}$ be guarded in P, and let $\gamma \in \mathcal{A} \cup \{\sigma\}$.*

1. *$P \xrightarrow{\gamma} P'$ implies $P[\mu x.Q/x] \xrightarrow{\gamma} P'[\mu x.Q/x]$.*
2. *$P[\mu x.Q/x] \xrightarrow{\gamma} P'[\mu x.Q/x]$ implies $\exists P'' \in \widehat{\mathcal{P}}.\ P \xrightarrow{\gamma} P''$ and $P'[\mu x.Q/x] \equiv P''[\mu x.Q/x]$.*

The second lemma concerns the *sort* of a term P, which is the set of labels of all transitions reachable in the transition system with start state P, i.e., $\mathsf{sort}(P) =_{\mathrm{df}} \{\alpha \in \mathcal{A} \mid \exists P'.\ P \longrightarrow^* P' \xrightarrow{\alpha}\}$, where $\longrightarrow^*$ denotes the reflexive and transitive closure of $\longrightarrow$ (when abstracting from transition labels).

LEMMA 2.2. *The set $\mathsf{sort}(P)$ of any term $P \in \widehat{\mathcal{P}}$ is finite.*

This statement follows from the facts that terms have finite length and that relabelings f satisfy the condition $|\{\alpha \mid f(\alpha) \neq \alpha\}| < \infty$. The above lemma establishes the well–definedness of some terms constructed below, which include a generalization of the summation operator indexed over actions contained in sorts. Note that TACS just provides a binary summation operator, i.e., only finite summations can be expressed.

3. Design Choices for (Bi)Simulation–based Faster–than Relations. In the following we define a reference faster–than relation, called *naive faster–than preorder*, which is inspired by Milner's notions of *simulation* and *bisimulation* [26]. Our main objective is to convince the reader that this simple faster–than preorder with its concise definition is not chosen arbitrarily. This is done by showing that it coincides with two other preorders which formalize a notion of faster–than as well and which are possibly more intuitive. The semantic theory of our faster–than relation will then be developed in the next section.

DEFINITION 3.1 (Naive faster–than preorder). *A relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a* naive faster–than relation *if the following conditions hold for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$.*

1. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\ Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
2. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\ P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
3. *$P \xrightarrow{\sigma} P'$ implies $\exists Q'.\ Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*

We write $P \sqsupseteq_n Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some naive faster–than relation $\mathcal{R}$.

Note that the behavioral relation $\sqsupseteq_{\mathrm{n}}$, as well as all other behavioral relations on processes defined in the sequel, can be extended to open terms by the usual means of closed substitution [26]. It is fairly easy to see that $\sqsupseteq_{\mathrm{n}}$ is a preorder, i.e., it is transitive and reflexive; moreover, $\sqsupseteq_{\mathrm{n}}$ is the largest naive faster–than relation. Technically speaking, the naive faster–than preorder refines bisimulation on action transitions by requiring simple simulation on clock transitions. Intuitively, $P \sqsupseteq_{\mathrm{n}} Q$ holds if P is faster than (or at least as fast as) Q, and if both processes are functionally equivalent (cf. Clauses (1) and (2)). Here, "*P is faster than Q*" means the following: if P may let time pass and the environment of P has to wait, then this should also be the case if one considers the slower (or equally fast) process Q instead (cf. Clause (3)). However, if Q lets time pass, then P is not required to match this behavior. Intuitively, we use bounded delays and are, accordingly, interested in worst–case behavior. Hence, clock transitions of the fast process must be matched, but not those of the slow process; behavior after an unmatched clock transition can just as well occur quickly without the time step, whence it is catered for in Clause (2). We come back to this issue shortly.

As the naive faster–than preorder is the basis of our approach, it is very important that its definition is intuitively convincing. There are two immediate questions which arise from our definition and are dealt with separately in the following two sections.

3.1. Question I. The first question emerges from the observation that Clauses (1) and (3) of Def. 3.1 require that an action or a time step of P must be matched with just this action or time step by Q. What if we are less strict? Maybe we should allow the slower process Q to perform some additional time steps when matching the behavior of P. This idea is formalized in the following definition of a variant of our faster–than preorder, which we refer to as *delayed faster–than preorder*. Here, $\xrightarrow{\sigma}{}^{+}$ and $\xrightarrow{\sigma}{}^{*}$ stand for the transitive and the transitive reflexive closure of the clock transition relation $\xrightarrow{\sigma}$, respectively.

Definition 3.2 (Delayed faster–than preorder). *A relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a* delayed faster–than relation *if the following conditions hold for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$.*

1. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\sigma}{}^{*} \xrightarrow{\alpha} \xrightarrow{\sigma}{}^{*} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
2. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
3. *$P \xrightarrow{\sigma} P'$ implies $\exists Q'.\, Q \xrightarrow{\sigma}{}^{+} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*

We write $P \sqsupseteq_{d} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some delayed faster–than relation $\mathcal{R}$.

As usual, one can derive that $\sqsupseteq_{\mathrm{d}}$ is a preorder and that it is the largest delayed faster–than relation. In the following we will show that both preorders $\sqsupseteq_{\mathrm{n}}$ and $\sqsupseteq_{\mathrm{d}}$ coincide. The proof of this first coincidence result is based on a syntactic relation $\succ$ on terms, which is defined next and which is similar to the progress preorder used in [23]. The objective for its definition is to provide a useful technical handle on the relation between clock transitions and speed, analogue to the "up to"–techniques employed for reasoning about bisimulation [33]. Thus, the relation $\succ$ is constructed such that we have property $(*)$: $P \xrightarrow{\sigma} P'$ implies $P' \succ P$, for any $P, P' \in \widehat{\mathcal{P}}$ (cf. Prop. 3.7(1)).

Definition 3.3. *The relation $\succ \subseteq \widehat{\mathcal{P}} \times \widehat{\mathcal{P}}$ is defined as the smallest relation satisfying the following properties, for all $P, P', Q, Q' \in \widehat{\mathcal{P}}$.*

Always:	(1) $P \succ P$	(2) $P \succ \sigma.P$
$P' \succ P$ and $Q' \succ Q$ implies:	(3) $P'\|Q' \succ P\|Q$	(4) $P' + Q' \succ P + Q$
	(5) $P' \setminus L \succ P \setminus L$	(6) $P'[f] \succ P[f]$
$P' \succ P$ and x is guarded in P implies:	(7) $P'[\mu x.\, P/x] \succ \mu x.\, P$	

Note that relation $\succ$ is not transitive and that it is not only defined for processes but for arbitrary, especially for open terms. The crucial clauses of the above definition are Clauses (2) and (7). Since we want $P \xrightarrow{\sigma} P'$ to imply $P' \succ P$, we clearly must include Clause (2). Additionally, Clause (7) covers the unwinding of recursion; for its motivation consider, e.g., the transition $\mu x.\,\sigma.a.\sigma.b.x \xrightarrow{\sigma} a.\sigma.b.\mu x.\,\sigma.a.\sigma.b.x$.

To establish the desired property $(*)$ of $\succ$, we need to state and prove some technical lemmas. The first two lemmas are concerned with the preservation of $\succ$ under substitution and with the preservation of substitution by $\succ$, respectively.

LEMMA 3.4. *Let $P, P', Q \in \widehat{\mathcal{P}}$ such that $P' \succ P$, and let $y \in \mathcal{V}$. Then:*

1. *y is guarded in P if and only if y is guarded in P'.*
2. *$P'[Q/y] \succ P[Q/y]$.*

Proof. Both statements can be proved by induction on the inference length of $P' \succ P$. The only interesting case concerns Case (7) of Def. 3.3, where, for both parts, we can assume $y \neq x$, since x is neither free in $P'[\mu x.P/x]$ nor in $\mu x.P$. Now assume $P'[\mu x.P/x] \succ \mu x.P$ due to $P' \succ P$.

1. If there exists an unguarded occurrence of y in $\mu x.P$, then there is also one in P and, by induction, in P'. The latter occurrence is also present after substituting $\mu x.P$ for x. Otherwise, y is guarded in $\mu x.P$, in P, and, by induction, in P'. Hence, every free occurrence of y in $P'[\mu x.P/x]$ either stems from P' and is guarded in P', or it is in a subterm of $\mu x.P$, where it is guarded.
2. By Barendregt's Assumption, we may assume that there is no free occurrence of x in Q and, by induction, $P'[Q/y] \succ P[Q/y]$. Hence, $(P'[\mu x.P/x])[Q/y] \equiv (P'[Q/y])[\mu x.(P[Q/y])/x] \succ \mu x.(P[Q/y]) \equiv (\mu x.P)[Q/y]$.

The other cases are straightforward and, thus, are omitted here. □

LEMMA 3.5. *Let $P, Q, Q', R \in \widehat{\mathcal{P}}$ and $x \in \mathcal{V}$ guarded in Q' such that $P \succ Q \equiv Q'[\mu x.R/x]$. Then there exists some $P' \in \widehat{\mathcal{P}}$ satisfying $P \equiv P'[\mu x.R/x]$ and $P' \succ Q'$.*

Proof. The proof is by induction on the size of Q', including a case analysis on the structure of Q'. The only interesting case is $Q' \equiv \mu y.S$ for some $y \in \mathcal{V}$ and $S \in \widehat{\mathcal{P}}$, where we can assume $P \not\equiv Q$ as well as $y \neq x$, and that y is not free in R. Now, $Q \equiv \mu y.(S[\mu x.R/x])$ and $P \equiv S'[\mu y.S[\mu x.R/x]/y]$ with $S' \succ S[\mu x.R/x]$. By induction hypothesis we can write S' as $S''[\mu x.R/x]$ for some S'' satisfying $S'' \succ S$. We can further write P as $S''[\mu y.S/y][\mu x.R/x]$ since y is not free in R. Finally, we may conclude this case by setting $P' \equiv S''[\mu y.S/y]$. □

This second lemma will become especially important in the next section (cf. Lemma 3.15). The following lemma relates $\succ$ to our notion of urgent action sets.

LEMMA 3.6. *Let $P, Q \in \widehat{\mathcal{P}}$.*

1. *If x is guarded in P, then $\mathcal{U}(P[Q/x]) = \mathcal{U}(P)$.*
2. *If $Q \succ P$, then $\mathcal{U}(Q) \supseteq \mathcal{U}(P)$.*

Proof. The proof of Part (1) is an easy induction on the structure of P. Part (2) follows by induction on the inference length of $Q \succ P$. Here, one needs to use Part (1) for Case (7) of Def. 3.3; observe that x is guarded in P' by Lemma 3.4(1). □

Now we have established the machinery which we need to prove the above property $(*)$ and, equally important, to prove that $\succ$ is a naive faster–than relation.

PROPOSITION 3.7.

1. *$P \xrightarrow{\sigma} P'$ implies $P' \succ P$, for any terms $P, P' \in \widehat{\mathcal{P}}$.*
2. *The relation $\succ$ satisfies the defining clauses of a naive faster–than relation, also on open terms; hence, $\succ_{|\mathcal{P}\times\mathcal{P}} \subseteq \sqsupseteq_n$.*

Proof. The proof of Part (1) is a straightforward induction on the length of inference of $P \xrightarrow{\sigma} P'$. For proving Part (2) we show that, for $P' \succ P$, the three clauses in the definition of $\sqsupseteq_{\mathrm{n}}$ are satisfied. This is done by induction on the inference length of $P' \succ P$. We only consider the interesting parts for some of the cases of Def. 3.3.

(2) $\underline{P \succ \sigma.P}$: Our semantics states that $P \xrightarrow{\alpha} P'$ if and only if $\sigma.P \xrightarrow{\alpha} P'$, for some P', thereby implying the first two clauses in Def. 3.1. If $P \xrightarrow{\sigma} P'$, then $\sigma.P \xrightarrow{\sigma} P$ and $P' \succ P$ by Part (1).

(3) $\underline{P'|Q' \succ P|Q}$: If $P'|Q' \xrightarrow{\alpha} P'_1|Q'$, for some P'_1, due to $P' \xrightarrow{\alpha} P'_1$ (cf. Rule (Com1)), then $P \xrightarrow{\alpha} P_1$ with $P'_1 \succ P_1$ and $Q' \succ Q$ by induction hypothesis. Hence, $P|Q \xrightarrow{\alpha} P_1|Q$ and $P'_1|Q' \succ P_1|Q$. The other cases involving Rules (Com2) and (Com3) are similar.
If $P'|Q' \xrightarrow{\sigma} P'_1|Q'_1$, for some P'_1 and Q'_1, due to $P' \xrightarrow{\sigma} P'_1$ and $Q' \xrightarrow{\sigma} Q'_1$ (cf. Rule (tCom)), then $P \xrightarrow{\sigma} P_1$ and $Q \xrightarrow{\sigma} Q_1$ with $P'_1 \succ P_1$ and $Q'_1 \succ Q_1$ by induction hypothesis. Using Lemma 3.6(2) we conclude from $P'|Q' \xrightarrow{\sigma}$ that $P|Q \xrightarrow{\sigma} P_1|Q_1$ and $P'_1|Q'_1 \succ P_1|Q_1$.

(7) $\underline{P'[\mu x.P/x] \succ \mu x.P}$: By Rule (Rec) any α–transition of $\mu x.P$ is of the form $\mu x.P \xrightarrow{\alpha} P_1[\mu x.P/x]$, for some P_1 with $P \xrightarrow{\alpha} P_1$. Then, by induction hypothesis, $P' \xrightarrow{\alpha} P'_1$ for some P'_1 satisfying $P'_1 \succ P_1$. Hence, $P'[\mu x.P/x] \xrightarrow{\alpha} P'_1[\mu x.P/x]$ by Lemma 2.1(1) since x is guarded in P' by Lemma 3.4(1), and we obtain $P'_1[\mu x.P/x] \succ P_1[\mu x.P/x]$ by Lemma 3.4(2).
On the other hand, any α–transition of $P'[\mu x.P/x]$ is of the form $P'[\mu x.P/x] \xrightarrow{\alpha} P'_1[\mu x.P/x]$ for some P'_1, where $P' \xrightarrow{\alpha} P''_1$ for some $P''_1 \in \widehat{\mathcal{P}}$ such that $P'_1[\mu x.P/x] \equiv P''_1[\mu x.P/x]$ by Lemma 2.1(2), since x is guarded in P' by Lemma 3.4(1). Thus, by induction hypothesis, $P \xrightarrow{\alpha} P_1$ with $P''_1 \succ P_1$, as well as $\mu x.P \xrightarrow{\alpha} P_1[\mu x.P/x]$ and $P'_1[\mu x.P/x] \equiv P''_1[\mu x.P/x] \succ P_1[\mu x.P/x]$ by Lemma 3.4(2).
The treatment of clock transitions is analogous.

The other parts are easier to prove and, therefore, are omitted. □

We are now able to state and prove our first main result.

THEOREM 3.8 (Coincidence I). *The preorders $\sqsupseteq_n$ and $\sqsupseteq_d$ coincide.*

Proof. Clearly, any naive faster–than relation, including $\succ_{|\mathcal{P}\times\mathcal{P}}$ according to Prop. 3.7(2), is a delayed one. Thus, it suffices to show that the largest delayed faster–than relation $\mathcal{R}$ is a naive faster–than relation. Hence, consider some arbitrary terms P and Q such that $P\,\mathcal{R}\,Q$.

If $P \xrightarrow{\sigma} P'$ for some process P', then we have $Q \equiv Q_0 \xrightarrow{\sigma} Q_1 \xrightarrow{\sigma} \cdots \xrightarrow{\sigma} Q_n$ and $P'\,\mathcal{R}\,Q_n$, for some $n \geq 1$ and some processes $Q_0, Q_1, \ldots, Q_n$. By Prop. 3.7(1) we get $Q_n \succ \cdots \succ Q_1 \succ Q$. Since $\succ_{|\mathcal{P}\times\mathcal{P}} \subseteq \mathcal{R}$ (see above) and since $\mathcal{R}$ is transitive, we conclude $P'\,\mathcal{R}\,Q$.

If $P \xrightarrow{\alpha} P'$ for some process P' and some action α, then we have $Q \equiv Q_0 \xrightarrow{\sigma} Q_1 \xrightarrow{\sigma} \cdots \xrightarrow{\sigma} Q_{n-1} \xrightarrow{\alpha} Q'_{n-1} \xrightarrow{\sigma}{}^{*} Q'$ and $P'\,\mathcal{R}\,Q'$, for some $n \geq 1$ and some processes $Q_0, Q_1, \ldots, Q_{n-1}, Q'_{n-1}, Q'$. Hence, we may conclude $P'\,\mathcal{R}\,Q'_{n-1}$ in analogy to the previous case. Since $Q_{n-1} \succ \cdots \succ Q_0$ by Prop. 3.7(1), we infer by repeated application of Prop. 3.7(2) that $Q_i \xrightarrow{\alpha} Q'_i$, for $0 \leq i \leq n-1$, such that $Q'_{n-1} \succ \cdots \succ Q'_0 \equiv Q''$. As above, this implies $P'\,\mathcal{R}\,Q''$ and $Q \xrightarrow{\alpha} Q''$.

The case $Q \xrightarrow{\alpha} Q'$, for some process P' and some action α, is obvious. □

This coincidence result justifies our preference of the simple and technically more elegant naive faster–than preorder $\sqsupseteq_{\mathrm{n}}$ over the probably more intuitive delayed faster–than preorder $\sqsupseteq_{\mathrm{d}}$. Nevertheless, $\sqsupseteq_{\mathrm{d}}$ could in practice be more useful since there exist delayed faster–than relations which are not naive faster–than relations, such as the relation $\{\langle \alpha.\mathbf{0}, \sigma^i.\alpha.\sigma^j.\mathbf{0}\rangle, \langle \alpha.\mathbf{0}, \alpha.\sigma^j.\mathbf{0}\rangle, \langle \mathbf{0}, \mathbf{0}\rangle\}$, for $i, j \in \mathbb{N}$ with $i > 0$. Note that this refers to the relations which define the preorders, and not to the preorders themselves.

3.2. Question II. We now turn to a second question which might be raised regarding the definition of the naive faster–than preorder $\sqsupseteq_{\mathrm{n}}$. Should one add a fourth clause to the definition of $\sqsupseteq_{\mathrm{n}}$ that permits, but not requires, the faster process P to match a clock transition of the slower process Q? More precisely, P might be able to do whatever Q can do after a time step, or P might itself have to perform a time step in order to match Q. Hence, a candidate for a fourth clause is

$$(4) \quad Q \xrightarrow{\sigma} Q' \text{ implies } \langle P, Q'\rangle \in \mathcal{R} \text{ or } \exists P'.\, P \xrightarrow{\sigma} P' \text{ and } \langle P', Q'\rangle \in \mathcal{R}\,.$$

Unfortunately, this requirement is not as sensible as it might appear at first sight. Consider the processes $P =_{\mathrm{df}} \sigma^n.a.\mathbf{0} \,|\, a.\mathbf{0} \,|\, \overline{a}.\mathbf{0}$ and $Q =_{\mathrm{df}} \sigma^n.a.\mathbf{0} \,|\, \sigma^n.a.\mathbf{0} \,|\, \overline{a}.\mathbf{0}$, for $n \geq 1$. Obviously, we expect P to be faster than Q. However, Q can engage in a clock transition to $Q' =_{\mathrm{df}} \sigma^{n-1}.a.\mathbf{0} \,|\, \sigma^{n-1}.a.\mathbf{0} \,|\, \overline{a}.\mathbf{0}$. According to Clause (4) and since $P \not\xrightarrow{\sigma}$, we would require P to be faster than Q'. This conclusion, however, should obviously be deemed wrong according to our intuition of "faster than."

The point of this example is that process P, which is in some components faster than Q, cannot mimic a clock transition of Q with a matching clock transition. However, since P is equally fast in the other components, it cannot simply leave out the time step. The solution to this situation is to remember within the relation $\mathcal{R}$ how many clock transitions P missed out and, in addition, to allow P to perform these clock transitions later. Thus, the computation $Q \xrightarrow{\sigma}{}^{n} a.\mathbf{0} \,|\, a.\mathbf{0} \,|\, \overline{a}.\mathbf{0} \xrightarrow{a} \mathbf{0} \,|\, a.\mathbf{0} \,|\, \overline{a}.\mathbf{0} \xrightarrow{a} \mathbf{0} \,|\, \mathbf{0} \,|\, \overline{a}.\mathbf{0}$ of Q, where we have no clock transitions between the two action transitions labeled by a, can be matched by P with the computation $P \xrightarrow{a} \sigma^n.a.\mathbf{0} \,|\, \mathbf{0} \,|\, \overline{a}.\mathbf{0} \xrightarrow{\sigma}{}^{n} a.\mathbf{0} \,|\, \mathbf{0} \,|\, \overline{a}.\mathbf{0} \xrightarrow{a} \mathbf{0} \,|\, \mathbf{0} \,|\, \overline{a}.\mathbf{0}$. This matching is intuitively correct, since the first a occurs faster in the considered trace of P than in the trace of Q, while the second a occurs at the same absolute time measured from the system start; only the time relative to the first a is greater for P. Observe that this example also testifies to the need to remember arbitrary large numbers of time steps, as $n \geq 1$ is finite but arbitrary. We formalize the above ideas in the following definition.

DEFINITION 3.9 (Family of faster–than preorders). *A family $(\mathcal{R}_i)_{i\in\mathbb{N}}$ of relations in $\mathcal{P} \times \mathcal{P}$, indexed by natural numbers (including 0), is a* family of indexed–faster–than relations *if the following conditions hold for all $i \in \mathbb{N}$, $\langle P, Q\rangle \in \mathcal{R}_i$, and $\alpha \in \mathcal{A}$.*

1. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$.*
2. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$.*
3. *$P \xrightarrow{\sigma} P'$ implies (a) $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$, or (b) $i > 0$ and $\langle P', Q\rangle \in \mathcal{R}_{i-1}$.*
4. *$Q \xrightarrow{\sigma} Q'$ implies (a) $\exists P'.\, P \xrightarrow{\sigma} P'$ and $\langle P', Q'\rangle \in \mathcal{R}_i$, or (b) $\langle P, Q'\rangle \in \mathcal{R}_{i+1}$.*

We write $P \sqsupseteq_i Q$ if $\langle P, Q\rangle \in \mathcal{R}_i$ for some family of indexed–faster–than relations $(\mathcal{R}_i)_{i\in\mathbb{N}}$.

Intuitively, $P \sqsupseteq_i Q$ means that process P is faster than process Q provided that P may delay up to i additional clock ticks which Q does not need to match. Observe that there exists a family of largest indexed–faster–than relations, but it is not clear that these relations are transitive. We establish, however, a stronger result by showing that our naive faster–than preorder $\sqsupseteq_{\mathrm{n}}$ coincides with $\sqsupseteq_0$. The proof of this result uses a family of purely syntactic relations $\succ_i$, for $i \in \mathbb{N}$, similar to relation $\succ$ in Def. 3.3.

DEFINITION 3.10. *The relations $\succ_i \subseteq \widehat{\mathcal{P}} \times \widehat{\mathcal{P}}$, for $i \in \mathbb{N}$, are defined as the smallest relations satisfying the following properties, for all $P, P', Q, Q', P_1, \ldots, P_n \in \widehat{\mathcal{P}}$ and $i, j \in \mathbb{N}$.*

Always:	(1) $P \succ_i P$		
$P_1 \succ P_2 \succ \cdots \succ P_n$ *implies:*	(2a) $P_1 \succ_i \sigma^j.P_n$		
$P' \succ_i P$ *and* $Q' \succ_i Q$ *implies:*	(2b) $\sigma.P' \succ_{i+1} P$		
	(3) $P'\|Q' \succ_i P\|Q$	(4) $P' + Q' \succ_i P + Q$	
	(5) $P' \setminus L \succ_i P \setminus L$	(6) $P'[f] \succ_i P[f]$	
$P' \succ_i P$ *and* x *is guarded in* P *implies:*	(7a) $P'[\mu x.\, P/x] \succ_i \mu x.\, P$		
$P' \succ_i P$ *and* x *is guarded in* P' *implies:*	(7b) $\mu x.\, P' \succ_i P[\mu x.\, P'/x]$		

Observe that Clauses (7a) and (7b) deal with an unwinding of recursion on both sides of $\succ_i$. This is related to our aim to match clock transitions from both sides of $\mathcal{R}_i$. Similarly, we allow the addition of σ on both sides of $\succ_i$ in Clauses (2a) and (2b) and also in more general situations than in Def. 3.3. The next lemma compares the relations $\succ_i$, for all $i \in \mathbb{N}$, to the relation $\succ$; it also compares the relations $\succ_i$ among themselves.

LEMMA 3.11.

1. *$\succ_i \subseteq \succ_{i+1}$, for all $i \in \mathbb{N}$.*
2. *$\succ \subseteq \succ_0$; in particular, $P \xrightarrow{\sigma} P'$ implies $P' \succ_0 P$, for any $P, P' \in \widehat{\mathcal{P}}$.*
3. *$P' \succ P$ (whence, $P \xrightarrow{\sigma} P'$) implies $P \succ_i P'$, for all $i > 0$, and for any $P, P' \in \widehat{\mathcal{P}}$.*

Proof. For Part (1) consider $P \succ_i Q$ and show $P \succ_{i+1} Q$ by induction on the inference of $P \succ_i Q$. The proof of Part (2) is analogous; for case $P \succ \sigma.P$ recall that $P \succ P$ and, hence, $P \succ_0 \sigma.P$. Also the proof of Part (3) is analogous; for case $P \succ \sigma.P$ use $P \succ_{i-1} P$ which implies $\sigma.P \succ_i P$. For the latter, the premise $i > 0$ is needed. Finally, observe that Clause 3.3(7) is matched by Clause 3.10(7b). □

This lemma states some useful facts about our syntactic relations. In particular, Part (3) compares $\succ^{-1}$ with $\succ_i$, for $i > 0$. We need, however, five more technical lemmas before we can prove our second coincidence theorem. The first one of these is the analogue of Lemma 3.4.

LEMMA 3.12. *Let $P, P', Q \in \widehat{\mathcal{P}}$ such that $P' \succ_i P$, and let $y \in \mathcal{V}$.*

1. *y is guarded in P if and only if y is guarded in P'.*
2. *$P'[Q/y] \succ_i P[Q/y]$.*

Proof. The proof is similar to the one of Lemma 3.4. In case $P_1 \succ_i \sigma^j.P_n$ (cf. Rule 3.10(2a)), use Lemma 3.4(2) to obtain $P_1[Q/y] \succ \cdots \succ P_n[Q/y]$. □

The second lemma states that $\succ_0$ is reflexive and that the relations $\succ_i$ only relate functionally equivalent terms, in the sense of strong bisimulation.

LEMMA 3.13. *Let $P, Q, R \in \widehat{\mathcal{P}}$ such that $P \succ_i Q$, and let $\alpha \in \mathcal{A}$. Then:*

1. *$R \succ_0 R$.*
2. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $P' \succ_i Q'$.*
3. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $P' \succ_i Q'$.*

Proof. While the proof of Part (1) is obvious, the ones for Parts (2) and (3) are similar to the "functional" part of Prop. 3.7(2). In Case (2a) we use that $\sigma^j.P_n \xrightarrow{\alpha} P_n'$ if and only if $P_n \xrightarrow{\alpha} P_n'$ if and only if $P_1 \xrightarrow{\alpha} P_1'$

with $P'_1 \succ \cdots \succ P'_n$, where the latter is inferred by Prop. 3.7(2). In Case (2b) we exploit the property $\succ_i \subseteq \succ_{i+1}$ of Lemma 3.11(1). Moreover, the proof for Case (7) is analogous to the one of Prop. 3.7(2) when using Lemma 3.12 instead of Lemma 3.4. □

The third lemma builds a bridge between relation $\succ_0$ and urgent action sets.

LEMMA 3.14. *$Q \succ_0 P$ implies $\mathcal{U}(Q) \supseteq \mathcal{U}(P)$, for any $P, Q \in \widehat{\mathcal{P}}$.*

Proof. The proof is by induction on the inference length of $Q \succ_0 P$. For Clause (2a) use Lemma 3.6(2) if $j = 0$. Observe that Clause (2b) does not apply. For Clause (7), employ Lemmas 3.6(1) and 3.12(1). □

The fourth lemma just serves as a prerequisite for proving the fifth lemma.

LEMMA 3.15. *If $P_1, P_2, \ldots, P_n \in \widehat{\mathcal{P}}$ for some $n \in \mathbb{N}$ such that $P_1 \succ P_2 \succ \cdots \succ P_n$, and if $P_n \xrightarrow{\sigma} P'$ for some $P' \in \widehat{\mathcal{P}}$, then $P_1 \succ_i P'$, for all $i > 0$.*

Proof. The proof is by induction on the structure of P_n. We may assume that all P_i are different and, by Lemma 3.11(3), that $n > 1$. First observe that P_n cannot be of the form x or $\tau.P$. If P_n is $\mathbf{0}$ or of the form $a.P$, we have $P' \equiv P_n$ and are done by Clause 3.10(2a) with $j = 0$. If P_n is $\sigma.P$, then $P_1 \succ \cdots \succ P_{n-1} \equiv P \equiv P'$, and we are done by Clauses 3.10(2a) or (1). The other cases are quite straightforward, except for $P_n \equiv \mu x.Q$. Here, $P_{n-1} \equiv Q'_{n-1}[\mu x.Q/x]$ with $Q'_{n-1} \succ Q$; by Lemma 3.4(1), x is guarded in Q'_{n-1} since it is guarded in Q. By repeated application of Lemmas 3.5 and 3.4(1), we conclude that each P_i, for $1 \leq i \leq n-1$, is of the form $Q'_i[\mu x.Q/x]$ and such that $Q'_1 \succ \cdots \succ Q'_{n-1}$. Furthermore, we have $P' \equiv Q'_n[\mu x.Q/x]$ with $Q \xrightarrow{\sigma} Q'_n$. Now we may apply the induction hypothesis to the Q_i's to obtain $Q'_1 \succ_i Q'_n$, which implies $P_1 \equiv Q'_1[\mu x.Q/x] \succ_i Q'_n[\mu x.Q/x] \equiv P'$ by Lemma 3.12(2). □

Finally, the fifth lemma establishes properties similar to those stated in Clauses (3) and (4) of Def. 3.9.

LEMMA 3.16. *Let $P \succ_i Q$ for some $P, Q \in \widehat{\mathcal{P}}$.*

1. *$P \xrightarrow{\sigma} P'$ implies*
   - *either: $i = 0$ and $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $P' \succ_i Q'$,*
   - *or: $i > 0$ and $P' \succ_{i-1} Q$.*
2. *$Q \xrightarrow{\sigma} Q'$ implies $P \succ_{i+1} Q'$.*

Proof. Both parts are proved by induction on the inference length of $P \succ_i Q$. We only consider the more interesting cases here.

- Part 1:
  (1) For $i > 0$, the time step $P \xrightarrow{\sigma} P'$ implies $P' \succ_j Q \equiv P$, for all j, by Lemmas 3.11(1) and (2).
  (2a) For $i > 0$, the time step $P_1 \xrightarrow{\sigma} P_0$ implies $P_0 \succ P_1 \succ \cdots \succ P_n$; hence, $P_0 \succ_{i-1} \sigma^j.P_n$. For $i = 0$ and $j > 0$, the same argument shows $P_0 \succ_i \sigma^{j-1}.P_n$, where $\sigma^j.P_n \xrightarrow{\sigma} \sigma^{j-1}.P_n$. For $i = j = 0$, by repeated application of Prop. 3.7, $P_1 \xrightarrow{\sigma} P'_1$ implies $P_n \xrightarrow{\sigma} P'_n$ for some P'_n satisfying $P'_1 \succ \cdots \succ P'_n$.
  (2b) Observe that $\sigma.P' \xrightarrow{\sigma} P'$ and $P' \succ_i P$ by the assumption of Def. 3.10(2b) and that $i + 1 > 0$. The remaining cases are straightforward for $i > 0$. In case of Clause (7) we only have to consider transitions of the form $P'[\mu x.P/x] \xrightarrow{\sigma} P''[\mu x.P/x]$ (by Lemma 2.1) or $\mu x.P' \xrightarrow{\sigma} P''[\mu x.P'/x]$, where $P'' \succ_{i-1} P$ by induction hypothesis. Then, we are done by employing Lemma 3.12(2) for Clause (7b). Finally, let us consider the case $i = 0$. This is largely analogous using Lemma 3.12(2) when dealing with Clauses (7a) and (7b). For Clause (3), apply Lemma 3.14 to deduce that the right–hand side can engage in a time step.

- Part 2:
  (1) $P \xrightarrow{\sigma} P'$ implies $P \succ_{i+1} P'$ by Lemma 3.11(3).
  (2a) Use Lemma 3.15 in case $j = 0$.
  (7) In case of Rule (7a), employ similar arguments as above using Lemma 3.12(2).

This completes the proof of Lemma 3.16. □

Using the above lemmas we can now proof the main result of this section.

THEOREM 3.17 (Coincidence II). *The preorders $\sqsupseteq_{\approx_n}$ and $\sqsupseteq_{\approx_0}$ coincide.*

Proof. Let $\mathcal{R}_{i\in\mathbb{N}}$ be a family of faster–than relations. Then, according to Def. 3.9, $\mathcal{R}_0$ is a naive faster–than relation, whence $\sqsupseteq_{\approx_0} \subseteq \sqsupseteq_{\approx_{\mathrm{n}}}$. For the reverse inclusion consider the largest naive faster–than relation $\mathcal{R}$ and define a family of $\mathcal{R}_i$, for $i \in \mathbb{N}$, by

$$R\,\mathcal{R}_i\,Q \;\text{ if }\; \exists P.\, R\,\mathcal{R}\,P \succ_i Q.$$

We check that these $\mathcal{R}_i$ satisfy Def. 3.9. Consider $R\,\mathcal{R}\,P \succ_i Q$.

1. If $R \xrightarrow{\alpha} R'$, then $P \xrightarrow{\alpha} P'$ with $R'\,\mathcal{R}\,P'$ by the definition of $\mathcal{R}$, as well as $Q \xrightarrow{\alpha} Q'$ with $P' \succ_i Q'$ by Lemma 3.13(2).
2. The case $Q \xrightarrow{\alpha} Q'$ is analogous and uses Lemma 3.13(3).
3. If $R \xrightarrow{\sigma} R'$, then $P \xrightarrow{\sigma} P'$ with $R'\,\mathcal{R}\,P'$. Now, Lemma 3.16(1) shows $Q \xrightarrow{\sigma} Q'$ with $R'\,\mathcal{R}_0\,Q'$, for $i = 0$, and $R'\,\mathcal{R}_{i-1}\,Q$, otherwise.
4. If $Q \xrightarrow{\sigma} Q'$, then $P \succ_{i+1} Q'$ by Lemma 3.16(2). Thus, $R\,\mathcal{R}_{i+1}Q'$.

This finishes the proof, since Lemma 3.13(1) implies $\mathcal{R} \subseteq \mathcal{R}_0$ □

Summarizing, we hope to have convinced the reader that our naive faster–than preorder is a sensible candidate for a faster–than preorder, as it coincides with two other candidates which seem to be at least equally appealing but are technically not as simple.

4. Semantic Theory of our Faster–than Relation. This section focuses (i) on developing a fully–abstract precongruence based on our naive faster–than preorder, (ii) on establishing its semantic theory, and (iii) on introducing a corresponding "weak" variant which abstracts from internal, unobservable actions.

4.1. A Fully–abstract Faster–than Relation. A shortcoming of the naive faster–than preorder $\sqsupseteq_{\approx_{\mathrm{n}}}$, as introduced above, is that it is not compositional. As an example, consider the processes $P =_{\mathrm{df}} \sigma.a.\mathbf{0}$ and $Q =_{\mathrm{df}} a.\mathbf{0}$, for which $P \sqsupseteq_{\approx_{\mathrm{n}}} Q$ holds according to Def. 3.1. Intuitively, however, this should not be the case, as we expect $P \equiv \sigma.Q$ to be strictly slower than Q. Technically, if we compose P and Q in parallel with process $R =_{\mathrm{df}} \overline{a}.\mathbf{0}$, then $P|R \xrightarrow{\sigma} a.\mathbf{0}|\overline{a}.\mathbf{0}$, but $Q|R \not\xrightarrow{\sigma}$, since any clock transition of $Q|R$ is preempted due to $\tau \in \mathcal{U}(Q|R)$. Hence, $P|R \not\sqsupseteq_{\approx_{\mathrm{n}}} Q|R$, i.e., $\sqsupseteq_{\approx_{\mathrm{n}}}$ is *not* a precongruence.

The reason for P and Q being equally fast according to $\sqsupseteq_{\approx_{\mathrm{n}}}$ lies in our SOS–rules: we allow Q to delay arbitrarily, since this might be necessary in a context where no communication on a is possible; thus, an additional potential delay as in P makes no difference; in fact, P and Q have exactly the same transitions. As R shows, we have to take a refined view once we fix a context, and the example indicates that, in order to find the largest precongruence contained in $\sqsupseteq_{\approx_{\mathrm{n}}}$, we have to take the urgent action sets of processes into account. The preorder $\sqsupseteq_{\approx}$, which repairs the precongruence defect of $\sqsupseteq_{\approx_{\mathrm{n}}}$, is defined as follows. According to $\sqsupseteq_{\approx}$ we generally have that P is strictly faster than $\sigma.P$, which is to be expected intuitively.

DEFINITION 4.1 (Strong faster–than precongruence). *A relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a* strong faster–than relation *if the following conditions hold for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$.*

1. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
2. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \xrightarrow{\alpha} P'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*
3. *$P \xrightarrow{\sigma} P'$ implies $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$ and $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*

We write $P \sqsupseteq\!\!\!\approx Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some strong faster–than relation $\mathcal{R}$.

Again, it is easy to see that $\sqsupseteq\!\!\!\approx$ is a preorder, that it is contained in $\sqsupseteq\!\!\!\approx_{\mathrm{n}}$, and that $\sqsupseteq\!\!\!\approx$ is the largest strong faster–than relation. Note that $\succ$, when restricted to processes, is not only a naive, but also a strong faster–than relation according to Lemma 3.6(2) and Prop. 3.7(2). As desired, we obtain the following full–abstraction result.

THEOREM 4.2 (Full abstraction). *The preorder $\sqsupseteq\!\!\!\approx$ is the* largest precongruence *contained in $\sqsupseteq\!\!\!\approx_n$.*

Proof. We first need to establish that $\sqsupseteq\!\!\!\approx$ is a precongruence. This can be done in the usual fashion [26]. Indeed, when comparing our technical framework to the bisimulation approach for the timed process algebra CSA developed in [11], which in turn extends CCS, then most cases of the compositionality proof can be easily adapted. One exception is our clock–prefix operator in TACS, for which we need to show that $P \sqsupseteq\!\!\!\approx Q$ implies $\sigma.P \sqsupseteq\!\!\!\approx \sigma.Q$. This is obvious, however, since the initial clock transition of $\sigma.P$ can be matched by the initial clock transition of $\sigma.Q$ and since all action transitions of $\sigma.P$ and $\sigma.Q$ are those of P and Q according to Rule (Pre). In addition, we present the compositionality proof for parallel composition, as it involves the rather unusual side condition regarding urgent action sets. By the definition of $\sqsupseteq\!\!\!\approx$, it suffices to prove that $\mathcal{R} =_{\mathrm{df}} \{\langle P|R, Q|R\rangle \mid P \sqsupseteq\!\!\!\approx Q,\ R \in \mathcal{P}\}$ is a strong faster–than relation. Therefore, let $\langle P|R, Q|R\rangle \in \mathcal{R}$.

- *Action transitions:* The cases $P|R \xrightarrow{\alpha} S$ and $Q|R \xrightarrow{\alpha} S$, for some $\alpha \in \mathcal{A}$ and $S \in \mathcal{P}$, follows along the lines of the corresponding cases in CCS [26] and, therefore, are omitted here.
- *Clock transitions:* Let $P|R \xrightarrow{\sigma} S$ for some $S \in \mathcal{P}$. According to the only applicable Rule (tCom) we know that (i) $P \xrightarrow{\sigma} P'$ for some $P' \in \mathcal{P}$, (ii) $R \xrightarrow{\sigma} R'$ for some $R' \in \mathcal{P}$, (iii) $\mathcal{U}(P) \cap \overline{\mathcal{U}(R)} = \emptyset$ as well as $\tau \notin \mathcal{U}(P)$ and $\tau \notin \mathcal{U}(R)$, and (iv) $S \equiv P'|R'$. Since $P \sqsupseteq\!\!\!\approx Q$, there exists a process Q' such that $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$, $Q \xrightarrow{\sigma} Q'$, and $P' \sqsupseteq\!\!\!\approx Q'$. Therefore, we may conclude $Q|R \xrightarrow{\sigma} Q'|R'$ by Rule (tCom) since $\mathcal{U}(Q) \cap \overline{\mathcal{U}(R)} = \emptyset$, and $\mathcal{U}(Q|R) = \mathcal{U}(Q) \cup \mathcal{U}(R) \subseteq \mathcal{U}(P) \cup \mathcal{U}(R) = \mathcal{U}(P|R)$, by the definition of urgent action sets and the fact that $\tau \notin \mathcal{U}(P)$, $\tau \notin \mathcal{U}(Q)$, and $\tau \notin \mathcal{U}(R)$. Moreover, $\langle P'|R', Q'|R'\rangle \in \mathcal{R}$ holds by the definition of $\mathcal{R}$, which finishes the proof.

The proof of the compositionality of recursion requires one to introduce a notion of *strong faster–than up to*. This definition and the compositionality proof itself is very similar to the one in CCS with respect to strong bisimulation [26].

We are left with establishing that $\sqsupseteq\!\!\!\approx$ is the *largest* precongruence contained in $\sqsupseteq\!\!\!\approx_{\mathrm{n}}$. The proof is a slight adaptation of one for CSA in [11]. As it is non–standard, it is worth presenting it in full here. From universal algebra, it is known that the largest precongruence $\sqsupseteq\!\!\!\approx_{\mathrm{n}}^{+}$ contained in the preorder $\sqsupseteq\!\!\!\approx_{\mathrm{n}}$ exists, and that $P \sqsupseteq\!\!\!\approx_{\mathrm{n}}^{+} Q$ if and only if $\forall$ TACS contexts $C[x].\, C[P] \sqsupseteq\!\!\!\approx_{\mathrm{n}} C[Q]$, where a TACS context $C[x]$ is a TACS term with one free occurrence of the variable x and no free occurrences of other variables. Recall that, for any context $C[x]$, term $C[P]$ is obtained by substituting P for x in $C[x]$ without any α–conversion, i.e., free variables in P might be captured. As $\sqsupseteq\!\!\!\approx$ is a precongruence contained in $\sqsupseteq\!\!\!\approx_{\mathrm{n}}$, we have $\sqsupseteq\!\!\!\approx \subseteq \sqsupseteq\!\!\!\approx_{\mathrm{n}}^{+}$, and it remains to show that $P \sqsupseteq\!\!\!\approx Q$, for some processes $P, Q \in \mathcal{P}$, whenever $C[P] \sqsupseteq\!\!\!\approx_{\mathrm{n}} C[Q]$, for all TACS contexts $C[x]$. For this it

suffices to consider the preorder $\precsim_a =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \precsim_{\mathrm{n}} C_{PQ}[Q]\}$. Here, $C_{PQ}[x] =_{\mathrm{df}} x \mid H_{PQ}$ and

$$H_{PQ} \;=_{\mathrm{df}}\; \mu x.(e.\mathbf{0} + \sum\{\tau.(D_L + d_L.x) \mid L \subseteq \overline{\mathsf{sort}(P)} \cup \overline{\mathsf{sort}(Q)}\})\,,$$

where D_L is defined as $\sum_{d\in L} d.\mathbf{0}$. Note that H_{PQ} is well–defined according to Lemma 2.2. The actions e and d_L and their complements are supposed to be "fresh" actions. In this section we do not exploit the presence of the distinguished action e, but we do so when re–using the above context in the proof of Thm. 4.15. To finish off our proof of Thm. 4.2, it is sufficient to establish the inclusion $\precsim_a \subseteq \precsim$, since the inclusion $\precsim_{\mathrm{n}}^+ \subseteq \precsim_a$ obviously holds.

We show that $\precsim_a$ is a strong faster–than relation according to Def. 4.1. Let P, $Q \in \mathcal{P}$ such that $P \precsim_a Q$, i.e., we have $C_{PQ}[P] \precsim_{\mathrm{n}} C_{PQ}[Q]$ by the definition of $\precsim_a$. In the following we consider two cases distinguishing whether process P performs an action transition or a clock transition. In each case the transition of P leads to a transition of $C_{PQ}[P]$. According to the definition of $\precsim_{\mathrm{n}}$ matching transitions must exist which mimic each step. From the existence of these transitions we may conclude additional conditions which are sufficient to establish $\precsim_a$ as a strong faster–than relation.

- *Situation 1:* Let $P \xrightarrow{\alpha} P'$ for some process P' and some action α. According to our operational semantics we have $C_{PQ}[P] \equiv P|H_{PQ} \xrightarrow{\alpha} P'|H_{PQ} \equiv C_{PQ}[P']$. This transition can only be matched by a corresponding transition of Q, say $Q \xrightarrow{\alpha} Q'$ for some Q'. This is even true in case $\alpha \equiv \tau$, because the τ–successors of H_{PQ} have the distinguished actions d_L enabled. Therefore, we have $C_{PQ}[Q] \equiv Q|H_{PQ} \xrightarrow{\alpha} Q'|H_{PQ} \equiv C_{PQ}[Q']$ and $C_{PQ}[P'] \precsim_{\mathrm{n}} C_{PQ}[Q']$. Because $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, one can check that also $C_{P'Q'}[P'] \precsim_{\mathrm{n}} C_{P'Q'}[Q']$ holds by construction of our contexts $C[x]$ (cf. a similar situation discussed in [30]). Thus, $P' \precsim_a Q'$. A transition $Q \xrightarrow{\alpha} Q'$ can be matched analogously.
- *Situation 2:* Let $P \xrightarrow{\sigma} P'$ for some term P'. As illustrated in Fig. 4.1 we let $C_{PQ}[P]$ perform a τ–transition to $P|H_L$, where $H_L =_{\mathrm{df}} D_L + d_L.H_{PQ}$ and $L =_{\mathrm{df}} \{\overline{c} \mid c \in (\mathsf{sort}(P) \cup \mathsf{sort}(Q)) \setminus \mathcal{U}(P)\}$. Then, $P|H_L$ can perform a clock transition to $P'|H_L$ according to Rule (tCom). Finally, we let $P'|H_L$ engage in the d_L–transition to $P'|H_{PQ}$.
  $C_{PQ}[Q]$ has to match the first step by a τ–transition to $Q|H_L$, since only this term has the distinguished action d_L enabled.
  Now we take a closer look at the second step. We have to match a clock transition. Therefore, Q has to perform a clock transition to some Q', and H_L has to idle, i.e., $Q|H_L \xrightarrow{\sigma} Q'|H_L$. According to Rule ($\mathsf{tCom}$), the condition $\mathcal{U}(Q) \cap \overline{\mathcal{U}(H_L)} = \emptyset$ has to be satisfied. Because of the choice of L, this implies $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$.
  Finally, the last step can only be matched by the transition $Q'|H_L \xrightarrow{d_L} Q'|H_{PQ}$. Thus, $C_{PQ}[P'] \equiv P'|H_{PQ} \precsim_{\mathrm{n}} Q'|H_{PQ} \equiv C_{PQ}[Q']$.
  Since $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ as well as $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, it follows in analogy to Situation (1) that $C_{P'Q'}[P'] \precsim_{\mathrm{n}} C_{P'Q'}[Q']$, i.e., $P' \precsim_a Q'$.

Thus, $\precsim_a$ is a strong faster–than relation, i.e., $\precsim_a \subseteq \precsim$ according to Def. 4.1. Hence, $\precsim_{\mathrm{n}}^+ \subseteq \precsim_a \subseteq \precsim$ which, together with the inclusion $\precsim \subseteq \precsim_{\mathrm{n}}^+$ obtained earlier yields $\precsim = \precsim_{\mathrm{n}}^+$, as desired. □

We conclude this section by showing that TACS is a conservative extension of CCS [26]. As noted earlier, we can interpret any process not containing a σ–prefix as CCS process, since then all relevant semantic rules for action transitions are the same as the ones for CCS. Moreover, for all TACS terms, we can adopt the equivalence *strong bisimulation* [26], in signs $\sim$, which is defined just as $\precsim$ when omitting the third clause of

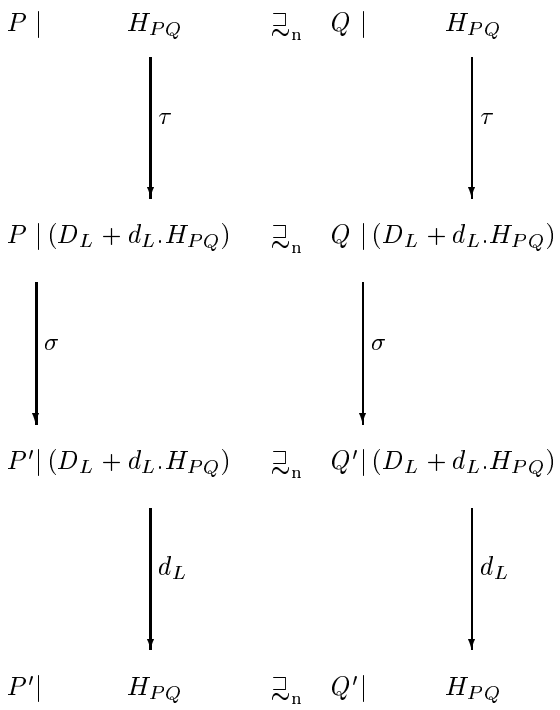

FIG. 4.1. *Largest precongruence proof: Illustration of Situation 2*

Def. 4.1. Furthermore, we denote the term obtained from some term $P \in \widehat{\mathcal{P}}$ when deleting all σ's by $strip(P)$. We may now state the following conservativity results.

THEOREM 4.3 (Conservativity). *Let P, $Q \in \mathcal{P}$.*

1. *Always $P \sqsupseteq\!\!\!\approx Q$ implies $P \sim Q$.*
2. *If P and Q do not contain any σ–prefixes, then $P \sqsupseteq\!\!\!\approx Q$ if and only if $Q \sqsupseteq\!\!\!\approx P$ if and only if $P \sim Q$.*
3. *Always $P \sim strip(P)$; furthermore, $P \stackrel{\sigma}{\longrightarrow} P'$ implies $P \sim P'$.*

Proof. The first part is an immediate consequence of the definitions of $\sim$ and $\sqsupseteq\!\!\!\approx$. The second part follows by the fact that terms without σ–prefixes (i) can only make a clock transition to themselves, namely if and only if no internal transition is enabled, and (ii) possess the same urgent actions whenever they are related by $\sqsupseteq\!\!\!\approx$ or $\sim$, since any action they can perform is urgent. For the first claim of the third part, one shows by structural induction on terms $P \in \widehat{\mathcal{P}}$ that the action transitions of $strip(P)$ are exactly all transitions $strip(P) \stackrel{\alpha}{\longrightarrow} strip(P')$ where $P \stackrel{\alpha}{\longrightarrow} P'$. For the second claim of the third part, one first proves that $P \stackrel{\sigma}{\longrightarrow} P'$ implies that $strip(P)$ and $strip(P')$ are identical up to unfolding of recursion. Then, one applies the first claim to finish the proof. □

This result shows that our strong faster–than preorder refines the well–established notion of strong bisimulation. Moreover, if no bounded delays occur in some processes, then these processes run in zero–time, and our strong faster–than preorder coincides with strong bisimulation. In other words, the strong faster–than preorder is thus restricted to consider the "functional" behavior of such processes only, irrespective of their relative speeds. That the bounded delays in TACS processes do not influence any "functional" behavior, is demonstrated in the third part of the above result.

TABLE 4.1
Axiomatization for finite sequential processes

(A1)	$t+u$	$=$ $u+t$	(D1)	$\mathbf{0}[f]$	$=$ $\mathbf{0}$	
(A2)	$t+(u+v)$	$=$ $(t+u)+v$	(D2)	$(\alpha.t)[f]$	$=$ $f(\alpha).(t[f])$	
(A3)	$t+t$	$=$ t	(D3)	$(\sigma.t)[f]$	$=$ $\sigma.(t[f])$	
(A4)	$t+\mathbf{0}$	$=$ t	(D4)	$(t+u)[f]$	$=$ $t[f]+u[f]$	
(P1)	$\sigma.t+\tau.u$	$=$ $t+\tau.u$	(C1)	$\mathbf{0}\setminus L$	$=$ $\mathbf{0}$	
(P2)	$a.t+\sigma.a.u$	$=$ $a.t+a.u$	(C2)	$(\alpha.t)\setminus L$	$=$ $\mathbf{0}$	$\alpha\in L\cup\overline{L}$
(P3)	$t+\sigma.t$	$=$ t	(C3)	$(\alpha.t)\setminus L$	$=$ $\alpha.(t\setminus L)$	$\alpha\notin L\cup\overline{L}$
(P4)	$\sigma.(t+u)$	$=$ $\sigma.t+\sigma.u$	(C4)	$(\sigma.t)\setminus L$	$=$ $\sigma.(t\setminus L)$	
(P5)	t	$\sqsupseteq$ $\sigma.t$	(C5)	$(t+u)\setminus L$	$=$ $(t\setminus L)+(u\setminus L)$	

The above embedding of CCS gives the technical conservation result in Thm. 4.3(2), but this might intuitively not be very pleasing: one might expect that the parallel execution of actions is faster than their arbitrary sequential execution, but the result shows that processes $a.\mathbf{0}|b.\mathbf{0}$ and $a.b.\mathbf{0}+b.a.\mathbf{0}$ are equally fast with respect to $\precsim$. Intuitively, for things happening with no time between them, it is difficult to see whether they happened one after the other or together. Of course, the zero–time between a and b is just a mathematical abstraction, but a useful one; it stands for a very short, negligible time. As an alternative, one could follow the approach of [23] and assume that actions might take some time, and for a uniform embedding of CCS one can give each action a bounded delay of one. Technically, this means to embed ordinary CCS–terms into TACS by inserting a σ–prefix before each action. Thm. 4.3(2) shows that this translation does not change any "functional" behavior. With this embedding, however, the classical expansion law "$a.\mathbf{0}\,|\,b.\mathbf{0} \;=\; a.b.\mathbf{0}+b.a.\mathbf{0}$" is not preserved due to timing: $\sigma.a.\mathbf{0}\,|\,\sigma.b.\mathbf{0}$ is strictly faster than $\sigma.a.\sigma.b.\mathbf{0}+\sigma.b.\sigma.a.\mathbf{0}$; consider the matching of a clock transition.

4.2. Axiomatization. In this section we provide a sound and complete axiomatization of our strong faster–than precongruence $\precsim$ for the class of finite sequential processes. According to standard terminology, a process is called *finite sequential* if it does neither contain any recursion operator nor any parallel operator. Although this class seems to be rather restrictive at first sight, it is simple and rich enough to demonstrate, by studying axioms, how exactly our semantic theory for $\precsim$ in TACS differs from the one for strong bisimulation in CCS [26]. We refer the reader to the end of this section for a discussion on the implications when considering to axiomatize larger classes of processes. As a notational convention we write $\mathcal{P}^{\mathsf{fin}}_{\mathsf{seq}}$ for the set of all finite sequential processes, ranged over by s, t, and u.

Now, we turn to the axioms for strong faster–than precongruence which are displayed in Table 4.1, where any axiom of the form $t=u$ should be read as two axioms $t\sqsupseteq u$ and $u\sqsupseteq t$. We write $\vdash t\sqsupseteq u$ if $t\sqsupseteq u$ can be derived from the axioms. Axioms (A1)–(A4), (D1)–(D4), and (C1)–(C5) are exactly the ones for strong bisimulation in CCS [26]. Hence, the semantic theory of our calculus is distinguished from the one for strong bisimulation by the additional Axioms (P1)–(P5). Intuitively, Axiom (P1) reflects our notion of maximal progress or urgency, namely that a process, which can engage in an internal urgent action, cannot delay. Axiom (P2) states that, if an action occurs "urgent" and "non–urgent" in a term, then it is indeed

urgent, i.e., the non–urgent occurrence of the action may be transformed into an urgent one. Axiom (P3) is similar in spirit, but cannot be derived from Axiom (P2) and the other axioms. Axiom (P4) is a standard axiom in timed process algebras and testifies to the fact that time is a deterministic concept which does not resolve choices. Finally, Axiom (P5) encodes our elementary intuition of σ–prefixes and speed within TACS, namely that any process t is faster than process $\sigma.t$ which might delay the execution of t by one clock tick.

The correctness of our axioms with respect to $\precsim$ can be established as usual [26]. However, it is worth noting that all axioms are sound for arbitrary TACS processes, not only for finite sequential ones. To prove the completeness of our axiomatization for finite sequential processes, we introduce a notion of *normal form* which is based on the following definition. A finite sequential process t is called *in summation form* if it is of the shape

$$t \;\equiv\; \sum_{i \in I} \alpha_i.t_i \;\; [\, + \, \sigma.t_\sigma \,]$$

where (i) I denotes a finite index set, (ii) all the t_i are in summation form, (iii) the subterm in brackets is optional and, if it exists, t_σ is in summation form, and (iv) $\alpha_i \in \mathcal{A}$, for all $i \in I$. Moreover, $\sum$ is the indexed version of $+$; we adopt the convention that the sum over the empty index set is identified with process $\mathbf{0}$. As expected, we obtain the following result.

PROPOSITION 4.4. *For any $t \in \mathcal{P}^{fin}_{seq}$ there exists some $u \in \mathcal{P}^{fin}_{seq}$ in summation form such that $\vdash t = u$.*

Proof. The proof proceeds by induction on the size of process t, i.e., the number of operators contained in t. Please observe, for the induction base, that process $\mathbf{0}$ is trivially in summation form. For the induction step, using Axioms (C1)–(C5) and Axioms (D1)–(D4), one can eliminate restrictions and relabelings as usual [26]. Consequently, t is transformed into a process which is just a sum of prefixed terms. In case of several σ–prefixed terms, these can be merged into one by (repeatedly) applying Axiom (P4) and possibly Axioms (A1) and (A2). Then, the processes trailing the prefixes can be brought into summation form according to the induction hypothesis. The proof details are quite straightforward and, thus, are omitted in this report. □

In the remainder, the following definition of the set of initial actions, in which some process t in summation form can engage in, will prove useful: $\mathcal{I}(t) =_{\mathrm{df}} \mathcal{U}(t)\; [\cup\, \mathcal{I}(t_\sigma)\,]$. It is easy to establish that $\mathcal{I}(t)$ is compatible with our operational semantics, i.e., the equality $\mathcal{I}(t) = \{\alpha \in \mathcal{A} \mid t \xrightarrow{\alpha}\}$ holds.

DEFINITION 4.5 (Normal form). *The process $\sum_{i \in I} \alpha_i.t_i \;\; [\, + \, \sigma.t_\sigma \,]$ in summation form is in* normal form *if all terms t_i, for $i \in I$, are in normal form and, in case the optional term in brackets is present, the following conditions are satisfied: (i) $t_\sigma \not\equiv \mathbf{0}$; (ii) $\forall i \in I.\ \alpha_i \not\equiv \tau$; (iii) $\forall i \in I.\ \alpha_i \notin \mathcal{I}(t_\sigma)$; and (iv) term t_σ is in normal form.*

Before we state the key proposition that every finite sequential process can be transformed into normal form, we note that Conds. (ii) and (iii) exactly correspond to our abovementioned intuitions regarding Axioms (P1) and (P2), respectively.

PROPOSITION 4.6. *For any $t \in \mathcal{P}^{fin}_{seq}$, there exists some $u \in \mathcal{P}^{fin}_{seq}$ in normal form such that $\vdash t = u$ and $\mathcal{U}(t) \subseteq \mathcal{U}(u)$.*

Note that, as one can check in the following proof, the set of urgent actions might increase when transforming a process into normal form due to the application of Axiom (P1), whereas the set of initial actions cannot change. This former inclusion is exploited in the completeness proof of our axiomatization.

Proof. According to Prop. 4.4 we may assume t to be in summation form. Now, the proof is by induction on the size of process $t \equiv \sum_{i\in I} \alpha_i.t_i \; [+ \; \sigma.t_\sigma \;]$. In the following, we only comment on the more interesting proof steps and do not explicitly mention applications of Axioms (A1) and (A2). Especially, the statement of the proposition is trivially true for the induction base $t \equiv \mathbf{0}$. Moreover, if the optional summand $\sigma.t_\sigma$ does not exist, then one just needs to apply the induction hypothesis to normalize all t_i, for $i \in I$, and the proof is done. Hence, we may assume that the summand $\sigma.t_\sigma$ is present. If Cond. (ii) is violated, i.e., if $\alpha_i = \tau$ for some $i \in I$, then $\vdash t = t' =_{\mathrm{df}} \sum_{i\in I} \alpha_i.t_i + t_\sigma$ by Axiom (P1). Observe that t' is in summation form, has smaller size than t, and satisfies $\mathcal{U}(t) \subseteq \mathcal{U}(t')$. One can now finish off this case by applying the induction hypothesis. Thus, we may assume that Cond. (ii) holds and turn our attention to establishing Cond. (iii). We first (repeatedly) use Axioms (A3) and (P2) and then Axiom (P4) to infer $\vdash \sum_{i\in I} \alpha_i.t_i + \sigma.t_\sigma = \sum_{i\in I} \alpha_i.t_i + \sigma.(\sum_{i\in I} \alpha_i.t_i) + \sigma.t_\sigma = \sum_{i\in I} \alpha_i.t_i + \sigma.(\sum_{i\in I} \alpha_i.t_i + t_\sigma)$. We can now apply the induction hypothesis to process $\sum_{i\in I} \alpha_i.t_i + t_\sigma$ and obtain a term t'' in normal form satisfying $\vdash \sum_{i\in I} \alpha_i.t_i + t_\sigma = t''$ and $\mathcal{U}(\sum_{i\in I} \alpha_i.t_i + t_\sigma) \subseteq \mathcal{U}(t'')$. From this inclusion, it is easy to see that term t'' can be written as $\sum_{k\in K} \gamma_k.t''_k + \sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma \;]$, for some index sets K and J, such that $\{\alpha_i \mid i \in I\} = \{\gamma_k \mid k \in K\}$ and $\{\gamma_k \mid k \in K\} \cap \{\beta_j \mid j \in J\} = \emptyset$. This implies $(*)$ $\alpha_i \notin \mathcal{I}(\sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma])$. By applying the above transformation backwards, i.e., by employing Axioms (P2) and (P4), we infer $\vdash t = \sum_{i\in I} \alpha_i.t_i + \sum_{k\in K} \gamma_k.t''_k + \sigma.(\sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma \;])$. The latter term satisfies Cond. (iii) due to property $(*)$ and still satisfies Cond. (ii), too. By induction we can normalize the processes t_i, for $i \in I$, while $\sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma \;]$ and the t''_k are in normal form since t'' is. Finally, in case $\sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma \;] \equiv \mathbf{0}$, we can eliminate the subterm $\sigma.(\sum_{j\in J} \beta_j.t''_j \; [+ \; \sigma.t''_\sigma \;])$ since $\vdash \mathbf{0} = \mathbf{0} + \sigma.\mathbf{0} = \sigma.\mathbf{0}$ by Axioms (P3) and (A4). This establishes Cond. (i), and we are done. □

Before we can proceed to our completeness theorem, we need to state a technical lemma.

LEMMA 4.7. *Let $t \equiv \sum_{i\in I} \alpha_i.t_i \; [+ \; \sigma.t_\sigma \;]$ and $u \equiv \sum_{j\in J} \beta_j.u_j \; [+ \; \sigma.u_\sigma \;]$ be processes in normal form such that $t \mathrel{\underline{\approx}} u$. Moreover, let $B \subseteq \{\beta_j \mid j \in J\}$.*

1. *$\{\beta_j \mid j \in J\} \subseteq \{\alpha_i \mid i \in I\}$.*
2. *$\sum_{\{i\in I \mid \alpha_i\in B\}} \alpha_i.t_i \mathrel{\underline{\approx}} \sum_{\{j\in J \mid \beta_j\in B\}} \beta_j.u_j$.*
3. *Always $\sum_{\{i\in I \mid \alpha_i\notin B\}} \alpha_i.t_i \; [+ \; \sigma.t_\sigma \;] \mathrel{\underline{\approx}} \sum_{\{j\in J \mid \beta_j\notin B\}} \beta_j.u_j \; [+ \; \sigma.u_\sigma \;]$.*

Proof.

- Part (1): If $\alpha_i \equiv \tau$ for some $i \in I$, then the summand $\sigma.t_\sigma$ does not exist and the claim follows from Def. 4.1(2). Otherwise, t can engage in a σ–transition, whence the claim coincides with $\mathcal{U}(u) \subseteq \mathcal{U}(t)$ which follows from Def. 4.1(3).

We are proving the other two statements separately and proceed along the case distinction explicit in the definition of $\mathrel{\underline{\approx}}$.

- Part (2): If the right–hand side can engage in an action transition, say $\sum_{\{j\in J \mid \beta_j\in B\}} \beta_j.u_j \xrightarrow{\beta_{j'}} u_{j'}$, then $u \xrightarrow{\beta_{j'}} u_{j'}$ and $t \xrightarrow{\beta_{j'}}$, by the definition of $\mathrel{\underline{\approx}}$. Since $B \subseteq \{\alpha_i \mid i \in I\}$ by (1), we have $\beta_{j'} \equiv \alpha_{i'}$, for some $i' \in I$, such that $t_\sigma \not\xrightarrow{\alpha_{j'}}$ by Cond. (iii) of normal forms. Hence, $\sum_{\{i\in I \mid \alpha_i\in B\}} \alpha_i.t_i \xrightarrow{\alpha_{i'}} t_{i'}$ and $t_{i'} \mathrel{\underline{\approx}} u_{j'}$. The case where the left–hand side engages in an action transition is analogous. Moreover, it is easy to see that both sides have the same sets of urgent actions and, if τ is not among these actions, then both terms can idle on σ.
- Part (3): The proof of this part is by induction on the size of process u. Since the induction base, i.e., $u \equiv \mathbf{0}$, is trivial, we only focus on the induction step.

If the left–hand side $\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ]$ can engage in an $\alpha_{i'}$–transition to $t_{i'}$, for some $\alpha_{i'}\notin B$, then so can t. Since $\alpha_{i'}\notin B$, the matching $\alpha_{i'}$–transition of u, according to the definition of $\sqsupseteq\!\!\!\approx$, also exists for the right–hand side $\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_j.u_j\ [+\ \sigma.u_\sigma\ ]$. A $\beta_{j'}$–transition of the right–hand side, for $j'\in\{j\in J\,|\,\beta_j\notin B\}$, can be treated analogously.

If the left–hand side can engage in an α–transition to some term t'_σ due to $\sigma.t_\sigma\stackrel{\alpha}{\longrightarrow}t'_\sigma$ for some $\alpha\in\mathcal{A}$, then $t\stackrel{\alpha}{\longrightarrow}t'_\sigma$ and $\alpha\notin B$ by (1) and Cond. (iii) of normal forms. Hence, the right–hand side can match this transition in the same way as u does according to the definition of $\sqsupseteq\!\!\!\approx$. A β–transition of the right–hand side, due to $\sigma.u_\sigma\stackrel{\beta}{\longrightarrow}u'_\sigma$ for some action β and some term u'_σ, can be dealt with in an analogous fashion.

It remains to consider the case $\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ]\stackrel{\sigma}{\longrightarrow}\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ t_\sigma\ ]$. If $\tau\in B$, then none of the optional summands exists, and $\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i$ and $\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_i.u_j$ can idle just as t and u can. If $\tau\notin B$, then $t\stackrel{\sigma}{\longrightarrow}\sum_{i\in I}\alpha_i.t_i\ [+\ t_\sigma\ ]$ and, according to the definition of $\sqsupseteq\!\!\!\approx$ and our operational rules: (a) $u\stackrel{\sigma}{\longrightarrow}\sum_{j\in J}\beta_j.u_j\ [+\ u_\sigma\ ]$, whence $\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_j.u_j\ [+\ \sigma.u_\sigma\ ]\stackrel{\sigma}{\longrightarrow}\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_j.u_j\ [+\ u_\sigma\ ]$; (b) $\mathcal{U}(u)\subseteq\mathcal{U}(t)$ which implies $\mathcal{U}(\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_j.u_j\ [+\ \sigma.u_\sigma\ ])=\mathcal{U}(u)\setminus B\subseteq\mathcal{U}(t)\setminus B=\mathcal{U}(\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ])$; (c) $\sum_{i\in I}\alpha_i.t_i\ [+\ t_\sigma\ ]\sqsupseteq\!\!\!\approx\sum_{j\in J}\beta_j.u_j\ [+\ u_\sigma\ ]$. Since the processes in (c) are again in normal form, one can apply the induction hypothesis to obtain $\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ t_\sigma\ ]\sqsupseteq\!\!\!\approx\sum_{\{j\in J\,|\,\beta_j\notin B\}}\beta_j.u_j\ [+\ u_\sigma\ ]$, as desired. Note that the urgent actions of t_σ and u_σ cannot be in B.

This completes the proof of Lemma 4.7. □

The next lemma essentially states the desired completeness result for specific finite sequential processes, namely those whose corresponding normal forms do not contain the optional σ–summand.

LEMMA 4.8. *Let $t\equiv\sum_{i\in I}\alpha_i.t_i$ and $u\equiv\sum_{j\in J}\beta_j.u_j$ be in normal form such that $t\sqsupseteq\!\!\!\approx u$. Then, $\vdash t\sqsupseteq u$.*

Proof. The proof is done by induction on the sum of the process sizes of t and u. For the induction base we have $t\equiv u\equiv\mathbf{0}$; hence, $\vdash\mathbf{0}\sqsupseteq\mathbf{0}$ trivially holds. In the induction step we reason as follows. According to the definition of $\sqsupseteq\!\!\!\approx$, there exists for each $i'\in I$ some $j'\in J$ such that $\alpha_{i'}=\beta_{j'}$ and $t_{i'}\sqsupseteq\!\!\!\approx u_{j'}$. By induction hypothesis we may conclude $\vdash t_{i'}\sqsupseteq u_{j'}$, whence $\vdash\alpha_{i'}.t_{i'}+\sum_{j\in J}\beta_j.u_j\sqsupseteq\beta_{j'}.u_{j'}+\sum_{j\in J}\beta_j.u_j=\sum_{j\in J}\beta_j.u_j$ by Axiom (A3) and possibly Axioms (A1) and (A2). By repeating this reasoning for each $i\in I$, we obtain $\vdash\sum_{i\in I}\alpha_i.t_i+\sum_{j\in J}\beta_j.u_j=t+u\sqsupseteq u=\sum_{j\in J}\beta_j.u_j$. Analogously, we can infer $\vdash t\sqsupseteq t+u$. Hence, $\vdash t\sqsupseteq u$ by transitivity. □

Finally, we are able to state and prove the main result of this section.

THEOREM 4.9 (Correctness & completeness). *For finite sequential processes t and u we have: $\vdash t\sqsupseteq u$ if and only if $t\sqsupseteq\!\!\!\approx u$.*

Proof. The correctness "$\Longrightarrow$" of our axiom system follows by induction on the length of the inference $\vdash t\sqsupseteq u$, as usual; we leave it as an exercise to the reader to show that indeed $\sqsupseteq$ may be safely replaced by $\sqsupseteq\!\!\!\approx$ in each axiom. Thus, we are left with proving completeness "$\Longleftarrow$". By Prop. 4.6 we may assume that the processes t and u are in normal form. If neither t nor u possesses an optional σ–summand, we are done by Lemma 4.8. Otherwise, we proceed by induction on the sum of the process sizes of t and u as follows.

We first apply Lemma 4.7(2) to $t\equiv\sum_{i\in I}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ]$, $u\equiv\sum_{j\in J}\beta_j.u_j\ [+\ \sigma.u_\sigma\ ]$, and $B=\{\beta_j\,|\,j\in J\}$, which yields $\sum_{\{i\in I\,|\,\alpha_i\in B\}}\alpha_i.t_i\sqsupseteq\!\!\!\approx\sum_{\{j\in J\,|\,\beta_j\in B\}}\beta_j.u_j$. As at least one of t_σ and u_σ is missing, we may apply the induction hypothesis to conclude $\vdash\sum_{\{i\in I\,|\,\alpha_i\in B\}}\alpha_i.t_i\ \sqsupseteq\ \sum_{\{j\in J\,|\,\beta_j\in B\}}\beta_j.u_j$.

Furthermore, by Lemma 4.7(3), $\sum_{\{i\in I\,|\,\alpha_i\notin B\}}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ] \sqsupseteq\!\!\!\approx \mathbf{0}\ [+\ \sigma.u_\sigma\ ]$. If $B\neq\emptyset$, one can apply the induction hypothesis to conclude that this relation is also derivable in our axiom system, and we are done. Otherwise, both t and u possess a σ–transition, which yields $\sum_{i\in I}\alpha_i.t_i\ [+\ t_\sigma\ ] \sqsupseteq\!\!\!\approx u_\sigma$ by the definition of $\sqsupseteq\!\!\!\approx$, with $u_\sigma\equiv\mathbf{0}$ if the summand $\sigma.u_\sigma$ is absent. According to the induction hypothesis (observe that at least one σ is missing when compared to t and u) we obtain $\vdash \sum_{i\in I}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ] \sqsupseteq u_\sigma$. Hence, we may conclude $\vdash \sum_{i\in I}\alpha_i.t_i\ [+\ \sigma.t_\sigma\ ] \sqsupseteq \sigma.(\sum_{i\in I}\alpha_i.t_i)\ [+\ \sigma.t_\sigma\ ] \sqsupseteq \sigma.(\sum_{i\in I}\alpha_i.t_i\ [+\ t_\sigma\ ]) \sqsupseteq \sigma.u_\sigma \sqsupseteq \mathbf{0}\ [+\ \sigma.u_\sigma\ ]$ by Axioms (P5), (P4), and (A4), by the above, and by the fact $\vdash \mathbf{0}+\sigma.\mathbf{0}=\mathbf{0}$. □

It is very desirable to extend our axiomatization to cover parallel composition, too, but this is non–trivial and still an open problem. As already mentioned, $\sigma.a.\mathbf{0}\,|\,\sigma.b.\mathbf{0}$ is strictly faster than $\sigma.a.\sigma.b.\mathbf{0}+\sigma.b.\sigma.a.\mathbf{0}$; but since σ is synchronized, a more sensible expansion law would try to equate $\sigma.a.\mathbf{0}\,|\,\sigma.b.\mathbf{0}$ with $\sigma.(a.\mathbf{0}\,|\,b.\mathbf{0})$. Unfortunately, this law does not hold, since the latter process can engage in an a–transition to $\mathbf{0}\,|\,b.\mathbf{0}$ and is therefore strictly faster. Thus, our situation is the same as in Moller and Tofts' paper [29] which also considers a bisimulation–type faster–than relation for asynchronous processes, but which deals with best–case rather than worst–case timing behavior. It turns out that the axioms for the sequential sub–calculus given in [29] are all true in our setting; however, we have the additional Axioms (P1) and (P2) which both are valid since σ is just a potential delay that can occur in certain contexts. Also Moller and Tofts do not treat parallel composition completely, just some expansion–like inequalities are listed. Once we know how parallel composition can be dealt with, extending our axiomatization to regular sequential processes, i.e., the class of finite–state sequential processes that do not contain restriction and relabeling operators inside recursion, can be done by adapting Milner's technique for uniquely characterizing recursive processes by systems of equations in normal form [25].

4.3. Abstracting from Internal Computation. The strong faster–than precongruence introduced in Sec. 4.1 is too discriminating for verifying systems in practice. It requires that two systems have to match each others action transitions exactly, even those labeled with the internal action τ. Consequently, one would like to abstract from τ's and develop a faster–than precongruence from the point of view of an external observer. As our algebra is a derivative of CCS, our approach closely follows the lines of [26].

We start off with the definition of a *naive weak faster–than preorder* which requires us to introduce the following auxiliary notations. For any action α, we define $\hat{\alpha}=_{\mathrm{df}}\epsilon$, if $\alpha=\tau$, and $\hat{\alpha}=_{\mathrm{df}}\alpha$, otherwise. Further, we let $\stackrel{\epsilon}{\Longrightarrow}=_{\mathrm{df}}\stackrel{\tau}{\longrightarrow}^{*}$ and write $P\stackrel{\alpha}{\Longrightarrow}Q$ if there exist R and S such that $P\stackrel{\epsilon}{\Longrightarrow}R\stackrel{\alpha}{\longrightarrow}S\stackrel{\epsilon}{\Longrightarrow}Q$.

Definition 4.10 (Naive weak faster–than preorder). *A relation $\mathcal{R}\subseteq\mathcal{P}\times\mathcal{P}$ is a* naive weak faster–than relation *if the following conditions hold for all $\langle P,Q\rangle\in\mathcal{R}$ and $\alpha\in\mathcal{A}$.*

1. *$P\stackrel{\alpha}{\longrightarrow}P'$ implies $\exists Q'.\,Q\stackrel{\hat{\alpha}}{\Longrightarrow}Q'$ and $\langle P',Q'\rangle\in\mathcal{R}$.*
2. *$Q\stackrel{\alpha}{\longrightarrow}Q'$ implies $\exists P'.\,P\stackrel{\hat{\alpha}}{\Longrightarrow}P'$ and $\langle P',Q'\rangle\in\mathcal{R}$.*
3. *$P\stackrel{\sigma}{\longrightarrow}P'$ implies $\exists Q',Q'',Q'''.\,Q\stackrel{\epsilon}{\Longrightarrow}Q''\stackrel{\sigma}{\longrightarrow}Q'''\stackrel{\epsilon}{\Longrightarrow}Q'$ and $\langle P',Q'\rangle\in\mathcal{R}$.*

We write $P\sqsupseteq\!\!\!\approx_n Q$ if $\langle P,Q\rangle\in\mathcal{R}$ for some naive weak faster–than relation $\mathcal{R}$.

Since no urgent action sets are considered, it is easy to see that $\sqsupseteq\!\!\!\approx_{\mathrm{n}}$ is not a precongruence. To get closer to our goal to define an observational faster–than precongruence, we re–define the third clause of the above definition; please note the analogy to the third clause of Def. 4.1.

Definition 4.11 (Weak faster–than preorder). *A relation $\mathcal{R}\subseteq\mathcal{P}\times\mathcal{P}$ is a* weak faster–than relation *if the following conditions hold for all $\langle P,Q\rangle\in\mathcal{R}$ and $\alpha\in\mathcal{A}$.*

1. $P \xrightarrow{\alpha} P'$ *implies* $\exists Q'.\, Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$.
2. $Q \xrightarrow{\alpha} Q'$ *implies* $\exists P'.\, P \stackrel{\hat{\alpha}}{\Longrightarrow} P'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$.
3. $P \xrightarrow{\sigma} P'$ *implies* $\exists Q', Q'', Q'''.\, Q \stackrel{\epsilon}{\Longrightarrow} Q'' \xrightarrow{\sigma} Q''' \stackrel{\epsilon}{\Longrightarrow} Q'$, $\mathcal{U}(Q'') \subseteq \mathcal{U}(P)$, *and* $\langle P', Q' \rangle \in \mathcal{R}$.

We write $P \sqsupseteq\!\!\approx Q$ *if* $\langle P, Q \rangle \in \mathcal{R}$ *for some weak faster–than relation* $\mathcal{R}$.

From this definition we may conclude that $\sqsupseteq\!\!\approx$ is the largest weak faster–than relation and that $\sqsupseteq\!\!\approx$ is a preorder. In addition, the following proposition holds.

PROPOSITION 4.12. *The relation* $\sqsupseteq\!\!\approx$ *is a precongruence for all operators except summation. Moreover,* $\sqsupseteq\!\!\approx$ *is characterized as the largest such precongruence contained in* $\sqsupseteq\!\!\approx_n$.

Proof. In the following we prove the precongruence property, i.e., we show that $\sqsupseteq\!\!\approx$ is compositional with respect to action prefixing, clock prefixing, parallel composition, restriction, relabeling, and recursion. Most cases are standard and can be checked along the lines of [26]. The case of clock prefixing is also easy and quite similar to the "strong" case. Therefore, we restrict ourselves to the case of parallel composition. For this proof, the following property turns out to be useful. Let P, P', $Q \in \mathcal{P}$ such that $P \stackrel{\epsilon}{\Longrightarrow} P'$. Then

$$P|Q \stackrel{\epsilon}{\Longrightarrow} P'|Q \qquad \text{and} \qquad Q|P \stackrel{\epsilon}{\Longrightarrow} Q|P' \tag{4.1}$$

This property can be proved by induction on the "length" of the weak transition $P \stackrel{\epsilon}{\Longrightarrow} P'$. For the compositionality proof regarding parallel composition, it is by Def. 4.11 sufficient to establish that

$$\mathcal{R} =_{\mathrm{df}} \{\langle P|R, Q|R \rangle \,|\, P \sqsupseteq\!\!\approx Q,\, R \in \mathcal{P}\}$$

is a weak faster–than relation. Let $\langle P|R, Q|R \rangle$ be an arbitrary pair in $\mathcal{R}$.

- *Action transitions:* The cases where $P|R \xrightarrow{\alpha} S$ and $Q|R \xrightarrow{\alpha} S$, for some $S \in \mathcal{P}$ and $\alpha \in \mathcal{A}$ are standard.
- *Clock transitions:* Let $P|R \xrightarrow{\sigma} S$ for some $S \in \mathcal{P}$. By the only applicable Rule (tCom) we know that (i) $P \xrightarrow{\sigma} P'$ for some $P' \in \mathcal{P}$, (ii) $R \xrightarrow{\sigma} R'$ for some $R' \in \mathcal{P}$, (iii) $\mathcal{U}(P) \cap \overline{\mathcal{U}(R)} = \emptyset$ as well as $\tau \notin \mathcal{U}(P)$ and $\tau \notin \mathcal{U}(R)$, and (iv) $S \equiv P'|R'$. Since $P \sqsupseteq\!\!\approx Q$, there exist terms $Q', Q'', Q''' \in \mathcal{P}$ such that $Q \stackrel{\epsilon}{\Longrightarrow} Q'' \xrightarrow{\sigma} Q''' \stackrel{\epsilon}{\Longrightarrow} Q'$, $\mathcal{U}(Q'') \subseteq \mathcal{U}(P)$, and $P' \sqsupseteq\!\!\approx Q'$. First, observe that $\mathcal{U}(Q'') \cap \overline{\mathcal{U}(R)} \subseteq \mathcal{U}(P) \cap \overline{\mathcal{U}(R)} = \emptyset$ and that $\tau \notin \mathcal{U}(Q'')$. Applying Property (4.1) and Rule (tCom) again, we conclude $Q|R \stackrel{\epsilon}{\Longrightarrow} Q''|R \xrightarrow{\sigma} Q'''|R' \stackrel{\epsilon}{\Longrightarrow} Q'|R'$. Moreover, $\mathcal{U}(Q''|R) = \mathcal{U}(Q'') \cup \mathcal{U}(R) \subseteq \mathcal{U}(P) \cup \mathcal{U}(R) = \mathcal{U}(P|R)$, since $\tau \notin \mathcal{U}(Q'')$, $\tau \notin \mathcal{U}(P)$, and $\tau \notin \mathcal{U}(R)$. Finally, $\langle P'|R', Q'|R' \rangle \in \mathcal{R}$ holds due to the definition of $\mathcal{R}$, which completes this proof part.

To conclude this part of the proof, we want to remark that, in order to show $\sqsupseteq\!\!\approx$ to be compositional with respect to recursion, we need to define a notion of *weak faster–than preorder up to* $\sqsupseteq\!\!\approx$ (cf. [33]), which can be done in the obvious fashion. Then, the proof is similar to the corresponding one in [26].

We are left with establishing the "largest" claim. From universal algebra we know that the *largest* precongruence $\sqsupseteq\!\!\approx_{\mathrm{n}}^{-}$ —for all operators except summation— contained in $\sqsupseteq\!\!\approx_{\mathrm{n}}$ exists. Since $\sqsupseteq\!\!\approx$ is such a precongruence, the inclusion $\sqsupseteq\!\!\approx \subseteq \sqsupseteq\!\!\approx_{\mathrm{n}}^{-}$ holds. Thus, it remains to show $\sqsupseteq\!\!\approx_{\mathrm{n}}^{-} \subseteq \sqsupseteq\!\!\approx$. Consider the relation $\sqsupseteq\!\!\approx_a =_{\mathrm{df}} \{\langle P, Q \rangle \,|\, C_{PQ}[P] \sqsupseteq\!\!\approx_{\mathrm{n}} C_{PQ}[Q]\}$, where the terms $C_{PQ}[x]$ are defined as in the proof of Thm. 4.2. Since x is simply put in parallel with process H_{PQ} in $C_{PQ}[x]$, we have that $P \sqsupseteq\!\!\approx_{\mathrm{n}}^{-} Q$ implies $C_{PQ}[P] \sqsupseteq\!\!\approx_{\mathrm{n}}^{-} C_{PQ}[Q]$ and $C_{PQ}[P] \sqsupseteq\!\!\approx_{\mathrm{n}} C_{PQ}[Q]$; we conclude that $\sqsupseteq\!\!\approx_{\mathrm{n}}^{-} \subseteq \sqsupseteq\!\!\approx_a$. The other necessary inclusion, $\sqsupseteq\!\!\approx_a \subseteq \sqsupseteq\!\!\approx$, is established by proving that $\sqsupseteq\!\!\approx_a$ is a weak faster–than relation. Let $P, Q \in \mathcal{P}$ such that $P \sqsupseteq\!\!\approx_a Q$, and consider the following two situations.

- *Situation 1:* Let $P \xrightarrow{\alpha} P'$ for some $P' \in \mathcal{P}$ and some $\alpha \in \mathcal{A}$. According to our operational semantics we may derive $C_{PQ}[P] \equiv P|H_{PQ} \xrightarrow{\alpha} P'|H_{PQ} \equiv C_{PQ}[P']$. This transition can only be matched by a corresponding weak transition of Q, say $Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$, for some $Q' \in \mathcal{P}$, since only process H_{PQ} has the distinguished action e enabled. Therefore, we have $C_{PQ}[Q] \equiv Q|H_{PQ} \stackrel{\hat{\alpha}}{\Longrightarrow} Q'|H_{PQ} \equiv C_{PQ}[Q']$ and $C_{PQ}[P'] \sqsupseteq_{\mathrm{n}} C_{PQ}[Q']$. Because $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$, also $C_{P'Q'}[P'] \sqsupseteq_{\mathrm{n}} C_{P'Q'}[Q']$ holds. Thus, $P' \sqsupseteq_a Q'$. The case where $Q \xrightarrow{\alpha} Q'$, for some $Q' \in \mathcal{P}$ and some $\alpha \in \mathcal{A}$, is analogue.
- *Situation 2:* Let $P \xrightarrow{\sigma} P'$ for some $P' \in \mathcal{P}$. As illustrated in Fig. 4.2, $C_{PQ}[P]$ can perform a τ–transition to $P|H_L$, where $H_L =_{\mathrm{df}} D_L + d_L.H_{PQ}$ and $L =_{\mathrm{df}} \{\overline{c} \,|\, c \in (\mathsf{sort}(P) \cup \mathsf{sort}(Q)) \setminus \mathcal{U}(P)\}$. Then, $P|H_L$ can engage in a σ–transition to $P'|H_L$ according to Rule (tCom). Finally, we consider the step $P'|H_L \xrightarrow{d_L} P'|H_{PQ}$.

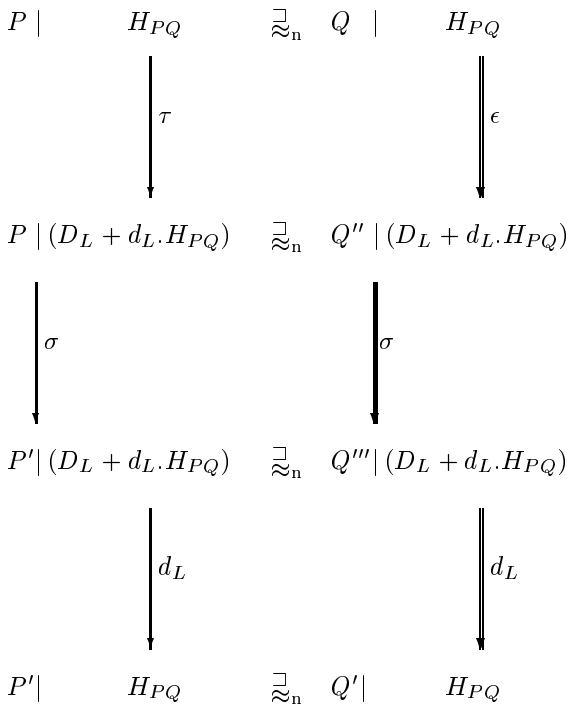

FIG. 4.2. *Largest precongruence proof: Illustration of Situation (2)*

Let us have a look at the first step. Since $C_{PQ}[P] \sqsupseteq_{\mathrm{n}} C_{PQ}[Q]$, we have $C_{PQ}[Q] \stackrel{\epsilon}{\Longrightarrow} W''$, for some $W'' \in \mathcal{P}$. We know that H_{PQ} has to perform a τ–transition to H_L but cannot take part in a communication, since e and d_L are distinguished actions. However, Q may be able to perform some τ–transitions to some process $Q'' \in \mathcal{P}$, i.e., $Q \stackrel{\epsilon}{\Longrightarrow} Q''$ and $P|H_L \sqsupseteq_{\mathrm{n}} Q''|H_L$.
Now we consider the more interesting second step. Since $P|H_L \sqsupseteq_{\mathrm{n}} Q''|H_L$, we know of the existence of some $W''' \in \mathcal{P}$ such that $Q''|H_L \stackrel{\sigma}{\Longrightarrow} W'''$ and $P'|H_L \sqsupseteq_{\mathrm{n}} W'''$. According to our operational semantics, Q'' and H_L have to perform a naive temporal weak σ–transition. Since H_L cannot take part in a communication (see above), it can only engage in an idling σ–transition $H_L \xrightarrow{\sigma} H_L$, and we conclude $W''' \equiv Q'''|H_L$ for some process $Q''' \in \mathcal{P}$ such that $Q'' \stackrel{\sigma}{\Longrightarrow} Q'''$, i.e., $Q'' \stackrel{\epsilon}{\Longrightarrow} Q_1''' \xrightarrow{\sigma} Q_2''' \stackrel{\epsilon}{\Longrightarrow} Q'''$ for some $Q_1''', Q_2''' \in \mathcal{P}$. Then, $Q''|H_L \stackrel{\epsilon}{\Longrightarrow} Q_1'''|H_L \xrightarrow{\sigma} Q_2'''|H_L \stackrel{\epsilon}{\Longrightarrow} Q'''|H_L$ must hold. According to Rule (tCom) the condition $\mathcal{U}(Q_1''') \cap \overline{\mathcal{U}(H_L)} = \emptyset$ has to be satisfied in order that the clock tick may occur. By the choice of L, this condition implies $\mathcal{U}(Q_1''') \subseteq \mathcal{U}(P)$, as desired.

Finally, let $P'|H_L \xrightarrow{d_L} P'|H_{PQ} \equiv C_{PQ}[P']$. Since $P'|H_L \mathrel{\supseteq\!\!\!\approx_{\rm n}} Q'''|H_L$, we have $Q'''|H_L \stackrel{d_L}{\Longrightarrow} W'$, for some $W' \in \mathcal{P}$. We know that H_L performs its d_L–transition to H_{PQ} since e is a distinguished action. However, Q''' may engage in some τ–transitions to some $Q' \in \mathcal{P}$, i.e., $Q''' \stackrel{\epsilon}{\Longrightarrow} Q'$, and $C_{PQ}[P'] \equiv P'|H_{PQ} \mathrel{\supseteq\!\!\!\approx_{\rm n}} Q'|H_{PQ} \equiv C_{PQ}[Q']$.
We have established the existence of processes $Q', Q'''_1, Q'''_2 \in \mathcal{P}$ such that $Q \stackrel{\epsilon}{\Longrightarrow} Q'''_1 \xrightarrow{\sigma} Q'''_2 \stackrel{\epsilon}{\Longrightarrow} Q'$ and $\mathcal{U}(Q'''_1) \subseteq \mathcal{U}(P)$. Also $C_{P'Q'}[P'] \mathrel{\supseteq\!\!\!\approx_{\rm n}} C_{P'Q'}[Q']$ holds, i.e., $P' \mathrel{\supseteq\!\!\!\approx_a} Q'$ since $C_{PQ}[P'] \mathrel{\supseteq\!\!\!\approx_{\rm n}} C_{PQ}[Q']$, $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$, and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$.

Thus, $\mathrel{\supseteq\!\!\!\approx_a}$ is indeed a weak faster–than relation, and we are done. □

The reason for the non–compositionality of the summation operator is similar to that with respect to observational equivalence in CCS [26]. Fortunately, the summation fix used for other bisimulation–based timed process algebras, such as CSA [11], proves effective for TACS, too.

Definition 4.13 (Weak faster–than precongruence). *A relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a* weak faster–than precongruence relation *if the following conditions hold for all $\langle P, Q\rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$.*

1. *$P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \mathrel{\supseteq\!\!\!\approx} Q'$.*
2. *$Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \stackrel{\alpha}{\Longrightarrow} P'$ and $P' \mathrel{\supseteq\!\!\!\approx} Q'$.*
3. *$P \xrightarrow{\sigma} P'$ implies $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$, and $\exists Q'.\, Q \xrightarrow{\sigma} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.*

We write $P \mathrel{\supseteq\!\!\!\simeq} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some weak faster–than precongruence relation $\mathcal{R}$.

We first show that $\mathrel{\supseteq\!\!\!\simeq}$ is indeed a precongruence and also present a simple full–abstraction result.

Proposition 4.14. *The relation $\mathrel{\supseteq\!\!\!\simeq}$ is the* largest *precongruence contained in $\mathrel{\supseteq\!\!\!\approx}$.*

Proof. The compositionality of $\mathrel{\supseteq\!\!\!\simeq}$ is easy to show for the cases of action and clock prefixing, restriction, and relabeling. In the following we deal with the remaining, more interesting cases. Let $P, Q, R, S \in \mathcal{P}$ be such that $P \mathrel{\supseteq\!\!\!\simeq} Q$ and $R \mathrel{\supseteq\!\!\!\simeq} S$. Then (1) $P|R \mathrel{\supseteq\!\!\!\simeq} Q|R$ and (2) $P + R \mathrel{\supseteq\!\!\!\simeq} Q + R$, which is established as follows.

1. According to Def. 4.13, it is sufficient to prove that the relation

$$\mathcal{R} =_{\rm df} \{\langle P|R, Q|R\rangle \mid P \mathrel{\supseteq\!\!\!\simeq} Q\,;\, R \in \mathcal{P}\}$$

   is a weak faster–than precongruence relation. Let $\langle P|R, Q|R\rangle \in \mathcal{R}$ be arbitrary.
   - *Action transitions:* The cases where $P|R \xrightarrow{\alpha} S$ or $Q|R \xrightarrow{\alpha} S$, for some $S \in \mathcal{P}$ and $\alpha \in \mathcal{A}$, are standard.
   - *Clock transitions:* Let $P|R \xrightarrow{\sigma} S$, for some $S \in \mathcal{P}$. This case can easily be treated along the lines of the corresponding case in the proof of the precongruence property of $\mathrel{\supseteq\!\!\!\simeq}$.
2. By Def. 4.13 it is sufficient to establish that the relation

$$\mathcal{R} =_{\rm df} \{\langle P + R, Q + R\rangle \mid P \mathrel{\supseteq\!\!\!\simeq} Q\,;\, R \in \mathcal{P}\}$$

   is a weak faster–than precongruence relation. Let $\langle P + R, Q + R\rangle \in \mathcal{R}$ be arbitrary.
   - *Action transitions:* Let $P + R \xrightarrow{\alpha} V$, for some $\alpha \in \mathcal{A}$ and $V \in \mathcal{P}$. Since the operational rules for summation with respect to actions are identical to the ones in CCS, and the definition of weak faster–than precongruence coincides with the one of observational congruence in CCS in this particular case, the proof follows along the lines of the corresponding proof in CCS.
   - *Clock transitions:* Let $P + R \xrightarrow{\sigma} V$, for some $V \in \mathcal{P}$, i.e., $P \xrightarrow{\sigma} P'$ and $R \xrightarrow{\sigma} R'$ for some $P', R' \in \mathcal{P}$, and $V \equiv P' + R'$ by Rule (tSum). Since $P \mathrel{\supseteq\!\!\!\simeq} Q$ we know of the existence of

some $Q' \in \mathcal{P}$ such that $Q \xrightarrow{\sigma} Q'$, $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$, and $P' \supseteq\!\!\!\!\sim Q'$. Therefore, we may conclude $Q + R \xrightarrow{\sigma} Q' + R'$ by Rule (tSum), as well as $\langle P' + R', Q' + R' \rangle \in \mathcal{R}$ by the definition of $\mathcal{R}$. Moreover, we have $\mathcal{U}(Q + R) = \mathcal{U}(Q) \cup \mathcal{U}(R) \subseteq \mathcal{U}(P) \cup \mathcal{U}(R) = \mathcal{U}(P + R)$ by the definition of urgent action sets, which finishes this part of the proof.

To show that $\supseteq\!\!\!\!\sim$ is compositional with respect to recursion, we have to adapt a notion of "up to" again.

A relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a *weak faster–than precongruence relation up to* $\supseteq\!\!\!\!\sim$ if the following conditions hold for every $\langle P, Q \rangle \in \mathcal{R}$ and $\alpha \in \mathcal{A}$.

1. $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \stackrel{\alpha}{\Longrightarrow} Q'$ and $P' \; \mathcal{R} \supseteq\!\!\!\!\sim Q'$, and
2. $Q \xrightarrow{\alpha} Q'$ implies $\exists P'.\, P \stackrel{\alpha}{\Longrightarrow} P'$ and $P' \supseteq\!\!\!\!\sim \mathcal{R} \; Q'$, and
3. $P \xrightarrow{\sigma} P'$ implies $\exists Q'.\, Q \xrightarrow{\sigma} Q'$, $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$, and $P' \; \mathcal{R} \supseteq\!\!\!\!\sim Q'$.

The proof follows pretty much the standard lines (cf. [26]) and, therefore, is omitted here.

We are left with establishing the "largest" claim. From universal algebra we know that the *largest* precongruence $\supseteq\!\!\!\!\approx^{+}$ in $\supseteq\!\!\!\!\approx$ exists and also that $\supseteq\!\!\!\!\approx^{+} = \{\langle P, Q \rangle \mid \forall C[x].\, C[P] \supseteq\!\!\!\!\approx C[Q]\}$. Since $\supseteq\!\!\!\!\sim$ is a precongruence which is contained in $\supseteq\!\!\!\!\approx$, the inclusion $\supseteq\!\!\!\!\sim \subseteq \supseteq\!\!\!\!\approx^{+}$ holds. Thus, it remains to show $\supseteq\!\!\!\!\approx^{+} \subseteq \supseteq\!\!\!\!\sim$. Consider the relation $\supseteq\!\!\!\!\sim_a =_{\mathrm{df}} \{\langle P, Q \rangle \mid P + c.\mathbf{0} \supseteq\!\!\!\!\approx Q + c.\mathbf{0}\,$, where $c \notin \mathsf{sort}(P) \cup \mathsf{sort}(Q)\}$. By definition of $\supseteq\!\!\!\!\sim_a$ we have $\supseteq\!\!\!\!\approx^{+} \subseteq \supseteq\!\!\!\!\sim_a$. We establish the other necessary inclusion $\supseteq\!\!\!\!\sim_a \subseteq \supseteq\!\!\!\!\sim$ by proving that $\supseteq\!\!\!\!\sim_a$ is a weak faster–than precongruence relation. Let $P \supseteq\!\!\!\!\sim_a Q$, i.e., $P + c.\mathbf{0} \supseteq\!\!\!\!\approx Q + c.\mathbf{0}$, and distinguish the following cases.

- *Action transitions:* Let $P \xrightarrow{\alpha} P'$, i.e., $\alpha \not\equiv c$ and $P + c.\mathbf{0} \xrightarrow{\alpha} P'$ by Rule (Sum1). Since $P \supseteq\!\!\!\!\sim_a Q$ we conclude the existence of some $V \in \mathcal{P}$ satisfying $Q + c.\mathbf{0} \stackrel{\hat{\alpha}}{\Longrightarrow} V$ and $P' \supseteq\!\!\!\!\approx V$. Because c is a distinguished action we have $V \not\equiv Q$ and, thus, $V \equiv Q'$ and $Q \stackrel{\alpha}{\Longrightarrow} Q'$, for some $Q' \in \mathcal{P}$.
- *Clock transitions:* Let $P \xrightarrow{\sigma} P'$. By Rules (tAct) and (tSum), $P + c.\mathbf{0} \xrightarrow{\sigma} P' + c.\mathbf{0}$ holds. Since $P \supseteq\!\!\!\!\sim_a Q$ we know of the existence of some $V, V', V'' \in \mathcal{P}$ such that $Q + c.\mathbf{0} \stackrel{\epsilon}{\Longrightarrow} V' \xrightarrow{\sigma} V'' \stackrel{\epsilon}{\Longrightarrow} V$, $\mathcal{U}(V') \subseteq \mathcal{U}(P)$, and $P' + c.\mathbf{0} \supseteq\!\!\!\!\approx V$. Because c is a distinguished action not in the sorts of P and Q, we conclude $V' \equiv Q + c.\mathbf{0}$, $V'' \equiv Q' + c.\mathbf{0}$ for some $Q' \in \mathcal{P}$, $V \equiv V''$, $Q \xrightarrow{\sigma} Q'$, and $\mathcal{U}(Q) \subseteq \mathcal{U}(P)$. Moreover, $P' \supseteq\!\!\!\!\sim_a Q'$ by the definition of $\supseteq\!\!\!\!\sim_a$ and the fact that $\mathsf{sort}(P') \subseteq \mathsf{sort}(P)$ and $\mathsf{sort}(Q') \subseteq \mathsf{sort}(Q)$.

This shows that $\supseteq\!\!\!\!\sim_a$ is a weak faster–than precongruence relation. Hence, $\supseteq\!\!\!\!\sim_a \subseteq \supseteq\!\!\!\!\sim$, as desired. □

Now we are able to state the main theorem of this section.

THEOREM 4.15 (Full–abstraction). *The relation* $\supseteq\!\!\!\!\sim$ *is the* largest precongruence *contained in* $\supseteq\!\!\!\!\approx_n$.

Proof. The claim follows from a general result established in universal algebra since (1) $\supseteq\!\!\!\!\approx$ is the largest precongruence —for all operators except summation— contained in $\supseteq\!\!\!\!\approx_{\mathrm{n}}$ (cf. Prop. 4.12) and since (2) $\supseteq\!\!\!\!\sim$ is the largest precongruence —for all operators including summation— contained in $\supseteq\!\!\!\!\approx$ (cf. Prop. 4.14). □

5. Example. We demonstrate the utility of our semantic theory for TACS by means of a small example dealing with two implementations of a 2–place storage in terms of an array and a buffer, respectively. Both can be defined using some definition of a 1–place buffer, e.g., $B_e =_{\mathrm{df}} \mu x.\sigma.in.\overline{out}.x$, which can alternately engage in communications with the environment on channels *in* and *out* [26]. Observe that we assume a communication on channel *out* to be urgent, while process B_e may autonomously delay a communication on channel *in* by one clock tick (cf. the single clock–prefix in front of action *in*). Finally, subscript e of process B_e should indicate that the 1–place buffer is initially empty. On the basis of B_e, one may now define

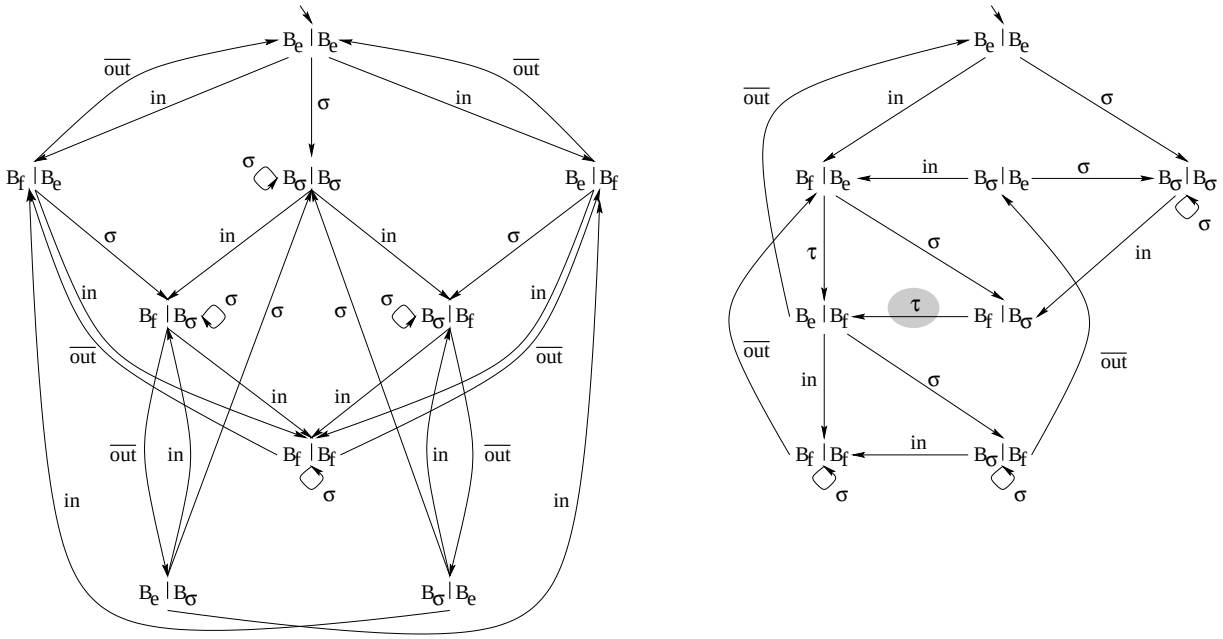

FIG. 5.1. *Semantics of the array variant (left) and the buffer variant (right).*

a 2–place array 2ARR and a 2–place buffer 2BUF as follows:

$$\text{2ARR} \;=_{\text{df}}\; B_e \,|\, B_e \qquad \text{and} \qquad \text{2BUF} \;=_{\text{df}}\; (B_e[c/out] \,|\, B_e[c/in]) \setminus \{c\}\,.$$

While 2ARR is simply the (independent) parallel composition of two 1–place buffers, 2BUF is constructed by sequencing two 1–place buffers, i.e., by taking the output of the first 1–place buffer to be the input of the second one (cf. the auxiliary internal channel c). Intuitively, we expect the array to behave functionally identical to the buffer, i.e., both should alternate between *in* and *out* actions. However, 2ARR should be faster than 2BUF since it can always output some of its contents immediately. In contrast, 2BUF needs to pass any item from the first to the second buffer cell, before it can output the item.

The semantics of the 2–place array 2ARR and our 2–place buffer 2BUF are depicted in Fig. 5.1 on the left and right, respectively. For notational convenience, we let B_σ stand for the process $\text{in}.\overline{\text{out}}.B_e$ and B_f for $\overline{\text{out}}.B_e$. Moreover, we leave out the restriction operator $\setminus\{c\}$ in the terms depicted for the buffer variant. The highlighted τ–transition indicates an urgent internal step of the buffer. Hence, process $(\text{B}_\text{f}|\text{B}_\sigma) \setminus \{c\}$ cannot engage in a clock transition. The other τ–transition depicted in Fig. 5.1 is non–urgent. As desired, our semantic theory for TACS relates 2ARR and 2BUF. Formally, this may be witnessed by the weak faster–than relation given in Table 5.1. It is easy to check, by employing Def. 4.11, that this relation is indeed a weak faster–than preorder, whence $\text{2ARR} \gtrapprox \text{2BUF}$. Moreover, since both 2ARR and 2BUF does not possess any initial internal transitions, they can also easily be proved to be weak faster–than precongruent, according to Def. 4.13. Thus, $\text{2ARR} \gtrapprox \text{2BUF}$, i.e., the 2–place array is faster than the 2–place buffer in all contexts, although functionally equivalent, which matches our abovementioned intuition.

6. Discussion and Related Work. This section highlights the unique features of our approach when compared to related work. There exists a large number of papers on both continuous and discrete timed process algebras; we refer the reader to [6] for a survey. Usually, these algebras focus on modeling *synchronous* systems, where components are under the regime of a global clock, and do not present faster–than relations.

TABLE 5.1
Pairs in the considered weak faster–than relation

$\langle (\mathrm{B_e}\|\mathrm{B_e}\ ,\ (\mathrm{B_e}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B_f}\|\mathrm{B_e}\ ,\ (\mathrm{B_f}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B_e}\|\mathrm{B_f}\ ,\ (\mathrm{B_f}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B}_\sigma\ ,\ (\mathrm{B}_\sigma\|\mathrm{B}_\sigma) \setminus \{c\}\rangle$
$\langle (\mathrm{B_f}\|\mathrm{B_e}\ ,\ (\mathrm{B_e}\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B_f}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B_f}\|\mathrm{B}_\sigma) \setminus \{c\}\rangle$	$\langle \mathrm{B_f}\|\mathrm{B_f}\ ,\ (\mathrm{B_f}\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B_f}\ ,\ (\mathrm{B_f}\|\mathrm{B}_\sigma) \setminus \{c\}\rangle$
$\langle (\mathrm{B_f}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B}_\sigma\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B_e}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B_e}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B_e}\ ,\ (\mathrm{B_e}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B_f}\ ,\ (\mathrm{B}_\sigma\|\mathrm{B_f}) \setminus \{c\}\rangle$
$\langle (\mathrm{B}_\sigma\|\mathrm{B_f}\ ,\ (\mathrm{B_e}\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B_f}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B_e}\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B_e}\|\mathrm{B_f}\ ,\ (\mathrm{B_e}\|\mathrm{B_f}) \setminus \{c\}\rangle$	$\langle \mathrm{B_e}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B}_\sigma\|\mathrm{B_e}) \setminus \{c\}\rangle$
$\langle (\mathrm{B_f}\|\mathrm{B}_\sigma\ ,\ (\mathrm{B_f}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B_f}\ ,\ (\mathrm{B_f}\|\mathrm{B_e}) \setminus \{c\}\rangle$	$\langle \mathrm{B}_\sigma\|\mathrm{B_e}\ ,\ (\mathrm{B}_\sigma\|\mathrm{B_e}) \setminus \{e\}\rangle$	

The latter is not surprising because, as argued in [29], it seems unlikely that, for synchronous systems, a faster–than preorder satisfying a few reasonable properties and being a precongruence for parallel composition exists. Traditionally, timed process algebras aiming at reasoning about synchronous systems have two common features: a delay operator specifying the exact time a process has to wait before it can proceed, and a timeout operator stating which enabled actions are withdrawn and which ones are additionally offered at a particular instant of time. In contrast, our work deals with *asynchronous* systems where actions are not enabled or disabled as time passes. Indeed, we added discrete time to CCS simply to evaluate the performance of asynchronous processes and not to increase the functional expressiveness of CCS. We did this by introducing a clock prefix operator specifying a single time bound which we interpreted as upper bound for delays. Some other timed process algebras annotate actions or processes with upper as well as with lower time bounds in the form of timing intervals [5, 15]; however, no faster–than relations have been defined in these settings.

The idea to investigate a (bi)simulation–based approach to compare the worst–case timing behavior of asynchronous systems was born out of the second author's research on faster–than preorders developed around DeNicola and Hennessy's testing theory [16]. This research was first conducted within the setting of Petri nets [9, 22, 35, 36] and then for a TCSP–style [34] process algebra, called PAFAS [23, 37]. The justification for adopting a testing approach is reflected in a fundamental result stating that the faster–than testing preorder based on continuous–time semantics coincides with the analogue testing preorder based on discrete–time semantics [23]. This result, however, depends very much on the testing setting and is different from the sort of discretization obtained for timed automata [2]. Nevertheless, PAFAS has certain disadvantages when compared to TACS. First of all, note that TACS allows one to specify arbitrary upper time bounds by nesting σ–prefixes. In PAFAS, every action has the same integrated upper time bound, namely 1, i.e., an a–prefix in PAFAS corresponds to a $\sigma.a.$–prefix in TACS. Our algebra TACS is also more expressive than PAFAS from a different point of view. Consider a process of the form $\sigma.(P|Q)$, for which the best counterpart in PAFAS is $\tau.(P|Q)$. Here, the τ–step incorporates a potential delay, but it can also decide choices which a σ–step cannot. Moreover, the equational laws established for the faster–than testing preorder of PAFAS, which provided an axiomatization for the class of finite sequential processes just as we did in this paper, are quite complicated. In contrast, the simple axioms presented here provide a clear, comprehensive insight into our semantics.

Some researchers consider *testing* [16] to be the more intuitive approach to semantics than *bisimulation* [26]. However, we feel that both are related within our setting. Essentially, the faster–than testing preorder presented for PAFAS in [23] is characterized as inclusion of traces annotated by refusal sets which underly the TACS approach, too. In our faster–than precongruences we require that, when a time step is matched, the urgent action set of the faster process contains the urgent action set of the slower one. One

may also say that non–urgent actions can be refused at this moment. If we call a set of non–urgent actions a refusal set, we could replace any clock transition by multiple transitions, one for each refusal set. Then, each refusal–set–transition of the faster process is matched by an equally labeled transition of the slower one.

Regarding other research concerning faster–than relations, our approach is most closely related to work by Moller and Tofts [29] who developed a bisimulation–based faster–than preorder within the discrete–time process algebra $\ell TCCS$. In their approach, asynchronous processes are modeled without any progress assumption. Instead, processes may idle arbitrarily long and, in addition, fixed delays may be specified. Hence, their setting considers best–case behavior, as the worst–case would be that for an arbitrary long time nothing happens. Moller and Tofts present an axiomatization of their faster–than preorder for finite sequential processes and discuss the problem of axiomatizing parallel composition, for which only valid laws for special cases are provided (cf. Sec. 4.2). It has to be mentioned here that the axioms and the behavioral preorder of Moller and Tofts do not completely correspond. In fact, writing σ for what is actually written (1) in [29], $a.\sigma.b.\mathbf{0} + a.b.\mathbf{0}$ is equally fast as $a.b.\mathbf{0}$, which does not seem to be derivable from the axioms. Also, the intuition behind relating these processes is not so clear, since $a.a.\sigma.b.\mathbf{0} + a.a.b.\mathbf{0}$ is not necessarily faster than or equally fast as $a.a.b.\mathbf{0}$. Since the publication in 1991, also Moller and Tofts noticed this shortcoming of their preorder [27]. The problem seems to lie in the way in which a transition $P \xrightarrow{a} P'$ of the faster process is matched: For intuitive reasons, the slower process must be allowed to perform time steps before engaging in a. Now the slower process is ahead in time, whence P' should be allowed some additional time steps. What might be wrong is that P' must perform these additional time steps immediately. We assume that a version of our indexed faster–than relation, which relaxes the latter requirement, would be more satisfactory. It would also be interesting to study the resulting preorder and compare it in detail to our faster–than precongruence; this should give a better understanding what worst–case and best–case timing behavior means for asynchronous systems in (bi)simulation–based settings.

A different idea for relating processes with respect to speed was investigated by Corradini et al. [14] within the so–called ill–timed–but–well–caused approach [1, 17]. The key of this approach is that components attach local time stamps to actions; however, actions occur as in an untimed algebra. Hence, in a sequence of actions exhibited by different processes running in parallel, local time stamps might decrease. This way, the timed algebra technically stays very close to untimed ones, but the "ill–timed" runs make the faster–than preorder of Corradini et al. difficult to relate to our approach.

Other research compares the efficiency of untimed CCS–like terms by counting internal actions either within a testing framework [13, 31] or a bisimulation–based setting [3, 4]. In all these approaches, except in [13] which does not consider parallel composition, runs of parallel processes are seen to be the interleaved runs of their component processes. Consequently, e.g., process $(\tau.a.\mathbf{0} \mid \tau.\overline{a}.b.\mathbf{0}) \setminus \{a\}$ is as efficient as process $\tau.\tau.\tau.b.\mathbf{0}$, whereas in our setting $(\sigma.a.\mathbf{0} \mid \sigma.\overline{a}.b.\mathbf{0}) \setminus \{a\}$ is strictly faster than $\sigma.\sigma.\tau.b.\mathbf{0}$.

7. Conclusions and Future Work. To consider the worst–case efficiency of asynchronous processes, i.e., those processes whose functional behavior is not influenced by timing issues, we defined the process algebra TACS. This algebra conservatively extends CCS by a clock prefix, which represents a delay of at most one time unit, and it takes time to be discrete. For TACS processes we then introduced a simple (bi)simulation–based faster–than preorder and showed this to coincide with two other variants of the preorder, both of which might be intuitively more convincing but which are certainly more complicated. We also developed a semantic theory for our faster–than preorder, including a coarsest precongruence result and an axiomatization for finite sequential processes, and investigated a corresponding "weak" preorder.

Future work should proceed along two orthogonal directions involving both theoretical and practical aspects. From a theory point of view, we intend to extend our axiomatization to larger classes of processes and also to our weak faster–than preorder. Recent papers provide an outline how the latter can be done for recursive processes in the presence of preemption [10, 20]; as a first step, one could also restrict attention to processes where parallel composition only occurs as top–level operator. Moreover, it remains an open question whether the faster–than precongruence, when defined for continuous time, coincides with the one presented here for discrete time, as is the case in the testing scenario presented in [35]. For putting the novel theory into practice, we plan to implement our process algebra and a decision procedure for our faster–than precongruence in the Concurrency Workbench [12], a formal verification tool.

REFERENCES

[1] L. Aceto and D. Murphy, *Timing and causality in process algebra*, Acta Informatica, 33 (1996), pp. 317–350.

[2] R. Alur and D. Dill, *A theory of timed automata*, Theoretical Computer Science, 126 (1994), pp. 183–235.

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

[4] S. Arun-Kumar and V. Natarajan, *Conformance: A precongruence close to bisimilarity*, in International Workshop on Structures in Concurrency Theory (STRICT '95), J. Desel, ed., Workshops in Computing, Springer-Verlag, May 1995, pp. 55–68.

[5] J. Baeten and J. Bergstra, *Real time process algebra*, Formal Aspects of Computing, 3 (1991), pp. 142–188.

[6] J. Baeten and C. Middelburg, *Process Algebra with Timing: Real Time and Discrete Time*, in Bergstra et al. [8], 2000, ch. 10.

[7] J. Baeten and W. Weijland, *Process Algebra*, Vol. 18 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, Cambridge, UK, 1990.

[8] J. Bergstra, A. Ponse, and S. Smolka, eds., *Handbook of Process Algebra*, Elsevier Science, 2000.

[9] E. Bihler and W. Vogler, *Efficiency of token-passing MUTEX-solutions – Some experiments*, in 19th International Conference on the Application and Theory of Petri Nets (ICATPN '98), J. Desel and M. Silva, eds., Vol. 1420 of Lecture Notes in Computer Science, Lisbon, Portugal, June 1998, Springer-Verlag, pp. 185–204.

[10] M. Bravetti and R. Gorrieri, *A complete axiomatization for observational congruence of prioritized finite-state behaviors*, in 27th International Colloquium on Automata, Languages and Programming (ICALP 2000), U. Montanari, J. Rolim, and E. Welzl, eds., Vol. 1853 of Lecture Notes in Computer Science, Geneva, Switzerland, July 2000, Springer-Verlag, pp. 744–755.

[11] R. Cleaveland, G. Lüttgen, and M. Mendler, *An algebraic theory of multiple clocks*, in 8th International Conference on Concurrency Theory (CONCUR '97), A. Mazurkiewicz and J. Winkowski, eds., Vol. 1243 of Lecture Notes in Computer Science, Warsaw, Poland, July 1997, Springer-Verlag, pp. 166–180.

[12] R. Cleaveland and S. Sims, *The NCSU Concurrency Workbench*, in Computer Aided Verification (CAV '96), R. Alur and T. Henzinger, eds., Vol. 1102 of Lecture Notes in Computer Science, New Brunswick, NJ, USA, July 1996, Springer-Verlag, pp. 394–397.

[13] R. Cleaveland and A. Zwarico, *A theory of testing for real time*, in 6th Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 1991, IEEE Computer Society Press, pp. 110–119.

[14] F. Corradini, R. Gorrieri, and M. Roccetti, *Performance preorder and competitive equivalence*, Acta Informatica, 34 (1997), pp. 805–835.

[15] F. Corradini and M. Pistore, *Closed interval process algebra versus open interval process algebra*, Acta Informatica, 37 (2000).

[16] R. DeNicola and M. Hennessy, *Testing equivalences for processes*, Theoretical Computer Science, 34 (1983), pp. 83–133.

[17] R. Gorrieri, M. Roccetti, and E. Stancampiano, *A theory of processes with durational actions*, Theoretical Computer Science, 140 (1995), pp. 73–94.

[18] M. Hennessy, *Algebraic Theory of Processes*, MIT Press, Boston, MA, USA, 1988.

[19] M. Hennessy and T. Regan, *A process algebra for timed systems*, Information and Computation, 117 (1995), pp. 221–239.

[20] H. Hermanns and M. Lohrey, *Priority and maximal progress are completely axiomatisable*, in 9th International Conference on Concurrency Theory (CONCUR '98), D. Sangiorgi and R. de Simone, eds., Vol. 1466 of Lecture Notes in Computer Science, Nice, France, September 1998, Springer-Verlag, pp. 237–252.

[21] C. Hoare, *Communicating Sequential Processes*, Prentice Hall, London, UK, 1985.

[22] L. Jenner and W. Vogler, *Fast asynchronous systems in dense time*, in 23rd International Colloquium on Automata, Languages and Programming (ICALP '96), F. Meyer auf der Heide and B. Monien, eds., Vol. 1099 of Lecture Notes in Computer Science, Paderborn, Germany, July 1996, Springer-Verlag, pp. 75–86.

[23] ——, *Comparing the efficiency of asynchronous systems*, in 5th International AMAST Workshop on Formal Methods for Real-time and Probabilistic Systems (ARTS '99), J.-P. Katoen, ed., Vol. 1601 of Lecture Notes in Computer Science, Bamberg, Germany, May 1999, Springer-Verlag, pp. 172–191.

[24] N. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1996.

[25] R. Milner, *A complete inference system for a class of regular behaviours*, Journal of Computer and System Sciences, 28 (1984), pp. 439–466.

[26] ——, *Communication and Concurrency*, Prentice Hall, London, UK, 1989.

[27] F. Moller, Private communication, 2000.

[28] F. Moller and C. Tofts, *A temporal calculus of communicating systems*, in 1st International Conference on Concurrency Theory (CONCUR '90), J. Baeten and J. Klop, eds., Vol. 458 of Lecture Notes in Computer Science, Amsterdam, The Netherlands, August 1990, Springer-Verlag, pp. 401–415.

[29] ——, *Relating processes with respect to speed*, in 2nd International Conference on Concurrency Theory (CONCUR '91), J. Baeten and J. Groote, eds., Vol. 527 of Lecture Notes in Computer Science, Amsterdam, The Netherlands, August 1991, Springer-Verlag, pp. 424–438.

[30] V. Natarajan, *Degrees of Delay: Semantic Theories for Priority, Efficiency, Fairness, and Predictability in Process Algebras*, Ph.D. thesis, North Carolina State University, Raleigh, NC, USA, August 1996.

[31] V. Natarajan and R. Cleaveland, *An algebraic theory of process efficiency*, in 11th Annual Symposium on Logic in Computer Science (LICS '96), New Brunswick, NJ, USA, July 1996, IEEE Computer Society Press, pp. 63–72.

[32] G. Reed and A. Roscoe, *The timed failures-stability model for CSP*, Theoretical Computer Science, 211 (1999), pp. 85–127.

[33] D. Sangiorgi and R. Milner, *The problem of 'weak bisimulation up to'*, in 3rd International Conference on Concurrency Theory (CONCUR '92), R. Cleaveland, ed., Vol. 630 of Lecture Notes in Computer Science, Stony Brook, NY, USA, August 1992, Springer-Verlag, pp. 32–46.

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

[35] W. Vogler, *Faster asynchronous systems*, in 6th International Conference on Concurrency Theory (CONCUR '95), I. Lee and S. Smolka, eds., Vol. 962 of Lecture Notes in Computer Science, Philadelphia, PA, USA, August 1995, Springer-Verlag, pp. 299–312.

[36] ——, *Efficiency of asynchronous systems and read arcs in Petri nets*, in 24th International Colloquium on Automata, Languages and Programming (ICALP '97), P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, eds., Vol. 1256 of Lecture Notes in Computer Science, Bologna, Italy, July 1997, Springer-Verlag, pp. 538–548.

[37] W. Vogler and L. Jenner, *Axiomatizing a fragment of PAFAS*, Electronic Notes in Computer Science, 39 (2000).

[38] W. Yi, *CCS + time = An interleaving model for real time systems*, in 18th International Colloquium on Automata, Languages and Programming (ICALP '91), J. Leach Albert, B. Monien, and M. Rodríguez Artalejo, eds., Vol. 510 of Lecture Notes in Computer Science, Madrid, Spain, July 1991, Springer-Verlag, pp. 217–228.