A Process Algebra with Distributed Priorities ⋆

Rance Cleaveland [1]

Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206, USA

Gerald Lüttgen [2]

Fakultät für Mathematik und Informatik, Universität Passau, 94030 Passau, Germany

V. Natarajan

Networking Hardware Division, IBM Corporation, Research Triangle Park, NC 27709, USA

This paper presents a process algebra for distributed systems in which some actions may take precedence over others. The algebra is distinguished by the design decision that it only allows actions to pre-empt others at the same "location" and therefore captures a notion of *localized precedence*. Using Park's and Milner's notion of strong bisimulation as a basis, we develop a behavioral congruence and axiomatize it for finite processes; we also derive an associated observational congruence and present logical characterizations of our behavioral relations. Simple examples highlight the utility of the theory.

Key words: process algebra, distributed priority, bisimulation, localized pre-emption, axiomatization.

⋆ An abstract of this paper appeared in *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, volume 1119 of *Lecture Notes in Computer Science*, pages 34–49, Pisa, Italy, August 1996, Springer-Verlag.

[1] Research supported by NSF/DARPA grant CCR-9014775, NSF grant CCR-9120995, ONR Young Investigator Award N00014-92-J-1582, NSF Young Investigator Award CCR-9257963, NSF grant CCR-9402807, and AFOSR grant F49620-95-1-0508.

[2] Research support partly provided by the German Academic Exchange Service under grant D/95/09026 (Doktorandenstipendium HSP II/ AUFE).

1 Introduction

Process algebras [18,19,24] constitute a widely studied framework for modeling and verifying concurrent systems [1,10]. Such theories typically consist of a simple calculus with a well-defined operational semantics given in terms of labeled transition systems; a behavioral equivalence is then used to relate implementations and specifications, which are both given as terms in the calculus. In order to facilitate compositional reasoning, in which systems are verified on the basis of the behavior of their components, researchers have devoted great attention to the definition of *behavioral congruences*, which allow the substitution of "equals for equals" inside larger systems. Traditional process algebras focus on modeling the potential nondeterminism that concurrent processes may exhibit; approaches have also been suggested for introducing sensitivity to other aspects of system behavior, including *priority* [2,3,6,7,14,16,20,21,29,35]. The concept of user-defined priorities enables the modeling of systems in which some system transitions (e.g. interrupts) may take precedence over others.

In this paper, we develop an algebraic theory of action priority for *distributed* systems. As in existing work, our aim is to model systems in which some transitions have precedence over others. Our point of departure is that the priority scheme should be localized within individual sites in the system; actions should only be able to pre-empt actions being performed at the "same location." This constraint reflects an essential intuition about distributed systems, namely, that the execution of a process on one processor should not affect the behavior of a process on another processor unless the designer explicitly builds in an interaction (e.g. synchronization) between them. Technically, we begin with a theory of priority that includes a notion of global precedence [7,29] and show how its semantics may be altered to localize capabilities for pre-emption by using locations [26]. We then define a semantics based on the notion of bisimulation [24,32]: we present a strong congruence, axiomatize it for finite processes, and derive an observational congruence along the lines of [24].

Our semantic framework is based on traditional CCS [24]; in particular, we "reduce" concurrency to nondeterminism using *interleaving*. Other semantic theories of concurrency [38] treat parallelism as a primitive notion; such "truly concurrent" frameworks include *causal approaches* — e.g. *partial order semantics* [12,13,37], *event structures* [30], *proved transitions* [4], and *Mazurkiewicz traces* [23] — and *location semantics* [5,15,26,27]. As these theories make concurrency explicit, they may be seen as natural bases for modeling distributed systems. However, in making concurrency generally observable, the frameworks become technically more complex than traditional interleaving-based ones. Our aim in this paper is to examine the extent to which one aspect of distribution — namely, priority localization — may be formalized within an interleaved theory of concurrency. Consequently, we start with an interleaving

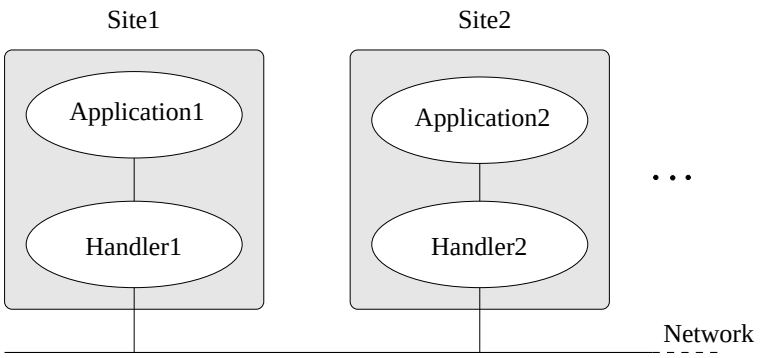

Fig. 1. A distributed system

semantics and introduce only as much sensitivity to "distribution" as our local view of pre-emption requires.

The rest of the paper is organized as follows. In the next section we present a generic example illustrating the need for local pre-emption in modeling systems. The three sections that follow present our calculus and derive the technical results discussed above, while Sect. 6 presents an example showing the application of our theory. The next section discusses some alternatives to our formulation. Sect. 8 focuses on related work, and the last section presents our conclusions and directions for future work. The appendix contains characterizations of our behavioral relations as standard strong bisimulations as well as logical characterizations of these relations. Due to space constraints we omit the more straightforward proofs; these may be found in [8].

2 Motivating Example

The example depicted in Fig. 1 motivates the need for considering a local notion of pre-emption when dealing with priorities in distributed systems. The system consists of two sites (computers), `Site1` and `Site2`, that are connected via the network `Network`. Each site runs an application, `Application1` and `Application2`, respectively, which may send or receive information from the application at the other site via its (interrupt-)handler, `Handler1` or `Handler2`. A handler delivers the message to the network or receives a message for its site from the network and notifies the application by sending an interrupt. Now, we have the following intuitive requirements that the semantics of a design language should satisfy in order to reflect the behavior of the system correctly. First, an interrupt of a handler should pre-empt the normal work of the application at its site, i.e. the application should immediately respond to an interrupt request. Second, both sites should be able to perform internal computations that are local to their site without interference from the other site. In particular, internal activities of `Handler1` should not pre-empt those of `Handler2`, and vice versa. While traditional process-algebraic treat-

ments [7,29] of priority satisfy the first requirement, they typically violate the second, since they allow an action of `Application1` to pre-empt an action of `Application2` if the former has higher priority, even though they are performed on different sites. In general, one would expect priorities at different sites to be incomparable. The semantics given in [7,29], however, do not permit this distinction to be made; the net effect is that some computations that one would expect to find in a distributed system are improperly suppressed. We propose to remedy this shortcoming in this paper by introducing a notion of *local pre-emption*.

3 Syntax and Semantics of $\mathsf{CCS}^{\mathsf{prio}}$

In this section we define the syntax and semantics of our calculus $\mathsf{CCS}^{\mathsf{prio}}$, which is based on CCS [24].

3.1 Syntax of $\mathsf{CCS}^{\mathsf{prio}}$

The syntax of $\mathsf{CCS}^{\mathsf{prio}}$ differs from CCS in that the action set exhibits a priority scheme, i.e. priorities are assigned to actions. It is important to note that an action may have different priorities in different states of the system under consideration. This property of priorities is called *globally dynamic* in [35]. For the sake of simplicity, we restrict ourselves to a two-level priority framework. In Section 7, we discuss how our results presented in this paper can be adapted to multi-level priority schemes. Intuitively, actions represent potential synchronizations that a process may be willing to engage in with its environment. Given a choice between a synchronization on a high priority action and one on a low priority action, a process should choose the former.

Formally, let Λ be a countable set of action labels, not including the *internal* or *silent* action τ. For every *input action* $a \in \Lambda$, there exists a *complementary action* $\overline{a}$, the corresponding *output action*. Let $\overline{\Lambda} =_{\mathrm{df}} \{\overline{a} \mid a \in \Lambda\}$, and let $A = \Lambda \cup \overline{\Lambda} \cup \{\tau\}$, where $\tau \notin \Lambda$, denote the set of all unprioritized actions. Intuitively, an action constitutes a willingness to perform a synchronization on the *port* associated with the action name. Thus a process that wishes to perform action a is willing to "receive" a message on port a, whereas a process that wishes to engage in $\overline{a}$ may send a message via port a. The action τ represents either an internal action of a process or the synchronization of two processes on some port in order to communicate with each other. We use $a, b, \ldots$ to range over Λ and $\alpha, \beta, \ldots$ to range over A.

In order to define *prioritized actions*, let $\underline{\Lambda}$ be a countable set of prioritized

action labels disjoint from Λ. Then $\underline{A} =_{\rm df} \underline{\Lambda} \cup \overline{\underline{\Lambda}} \cup \{\underline{\tau}\}$ is the set of *prioritized actions*, where $\underline{\tau} \notin \underline{\Lambda}$ is the prioritized *internal* or *silent* action. We use $\mathcal{A} =_{\rm df} A \cup \underline{A}$ to denote the set of all actions. Intuitively, prioritized actions represent communication potentials over "important" channels. Therefore, communications involving prioritized actions should be preferred over communications involving unprioritized actions. In the remainder of the paper, we let $\underline{a}, \underline{b}, \ldots$ range over $\underline{\Lambda}$, $\underline{\alpha}, \underline{\beta}, \ldots$ over $\underline{A}$, and γ, δ over $\mathcal{A}$. Additionally, we extend $\bar{}$ by defining $\overline{\overline{\gamma}} = \gamma$, and if $L \subseteq \mathcal{A} \setminus \{\tau, \underline{\tau}\}$ then $\overline{L} =_{\rm df} \{\overline{\gamma} \mid \gamma \in L\}$. A mapping f on $\mathcal{A}$ is a *relabeling* if f preserves priorities (i.e. $f(\Lambda) \subseteq \Lambda$ and $f(\underline{\Lambda}) \subseteq \underline{\Lambda}$), is such that the set $\{\gamma \mid f(\gamma) \neq \gamma\}$ is finite, and satisfies the following: $f(\overline{a}) = \overline{f(a)}$ $f(\overline{\underline{a}}) = \overline{f(\underline{a})}$, $f(\tau) = \tau$, and $f(\underline{\tau}) = \underline{\tau}$.

The syntax of our calculus may now be defined by the BNF

$$P \quad ::= \quad \mathbf{0} \mid \gamma.P \mid P + P \mid P|P \mid P[f] \mid P \setminus L \mid C \stackrel{\rm def}{=} P$$

where f is a relabeling, $L \subseteq \mathcal{A} \setminus \{\tau, \underline{\tau}\}$, and C is a process constant. We use the standard definitions for *sort* of a process, *free* and *bound variables*, *open* and *closed terms*, *guarded recursion*, and *contexts*. We refer to closed and guarded terms as *processes* and denote syntactic equality by $\equiv$. Finally, we let $P, Q, R, \ldots$ range over the set $\mathcal{P}$ of processes.

3.2 Locations

We now introduce the notion of *location*, which will be used in the next section in the operational semantics for $\mathsf{CCS}^{\mathsf{prio}}$ as a basis for deciding when one transition pre-empts another. Intuitively, a location represents the "address(es)" of subterm(s) inside a larger term; when a system performs an action, our semantics will also note the location of the subterm(s) that "generate(s)" this action. Observe that because of the potential for synchronization more than one subterm may be involved in an action. Our account of locations closely follows that of [26].

Formally, let $\mathcal{A}_{\rm addr} =_{\rm df} \{L, R, l, r\}$ be the *address alphabet*, and let $\bullet$ be a special symbol not in $\mathcal{A}_{\rm addr}$. Then $\mathcal{A}ddr =_{\rm df} \{\bullet s \mid s \in \mathcal{A}^*_{\rm addr}\}$ represents the set of (process) *addresses* ranged over by v, w. We abuse notation by omitting $\bullet$ from addresses on occasion. If $\bullet s_1$ and $\bullet s_2$ are addresses then we write $\bullet s_1 \cdot \bullet s_2 = \bullet s_1 s_2$ to represent address concatenation (where $s_1 s_2$ represents the usual concatenation of elements in $\mathcal{A}^*_{\rm addr}$). Further, if $V \subseteq \mathcal{A}ddr$ then we write $V \cdot \zeta$ for $\{v \cdot \zeta \mid v \in V\}$. Intuitively, an element of $\mathcal{A}ddr$ represents the address of a subterm, with $\bullet$ denoting the current term, l (r) representing the left (right) subterm of $+$, and L (R) the left (right) subterm of $|$. For example, in the process $(a.\mathbf{0} \mid b.\mathbf{0}) + c.\mathbf{0}$ the address of $a.\mathbf{0}$ is $\bullet Ll$, of $b.\mathbf{0}$ is $\bullet Rl$, and of $c.\mathbf{0}$ is $\bullet r$.

As mentioned in the introduction, we want to adopt the view that processes on different sides of the parallel operator are logically – not necessarily physically – executed on different processors. Thus, priorities on different sides of the parallel operator are distributed and, therefore, should be incomparable. However, priorities on different sides of the summation operator, which models nondeterministic choice, should be comparable since argument processes of summation are logically scheduled on the same processor. We formalize this intuition in the following *comparability relation* on addresses which is adapted from [17].

Definition 1 *The comparability relation* $\bowtie$ *on addresses is the smallest reflexive and symmetric subset of* $\mathcal{A}ddr \times \mathcal{A}ddr$ *such that for all* $v, w \in \mathcal{A}ddr$ *:*

(i) $\langle v \cdot l, w \cdot r \rangle \in \bowtie$ *, and*
(ii) $\langle v, w \rangle \in \bowtie$ *implies* $\langle v \cdot \zeta, w \cdot \zeta \rangle \in \bowtie$ *for* $\zeta \in \mathcal{A}_{addr}$ *.*

We write $v \bowtie w$ *instead of* $\langle v, w \rangle \in \bowtie$*.*

If $v \in \mathcal{A}ddr$ then we use $[v]$ to denote the set $\{w \in \mathcal{A}ddr \mid v \bowtie w\}$.

Note that the comparability relation is not transitive, e.g. we have $Ll \bowtie r$ and $r \bowtie Rl$ but $Ll \not\bowtie Rl$, since $L \not\bowtie R$. Considering our example $(a.\mathbf{0} \mid b.\mathbf{0}) + c.\mathbf{0}$ above, the addresses of $a.\mathbf{0}$ and $c.\mathbf{0}$, and the addresses of $b.\mathbf{0}$ and $c.\mathbf{0}$, are comparable since they are on different sides of the summation operator. In contrast, the addresses of $a.\mathbf{0}$ and $b.\mathbf{0}$ are incomparable since they are on different sides of the parallel operator.

We may now define the set of (transition) *locations*, $\mathcal{L}oc$, as $\mathcal{L}oc =_{\mathrm{df}} \mathcal{A}ddr \cup (\mathcal{A}ddr \times \mathcal{A}ddr)$. Intuitively, a transition location records the addresses of the components in a term that participate in the execution of a given action. In our algebra, transitions are performed by single processes or pairs of processes (in the case of a synchronization). We define $\langle v, w \rangle \cdot \zeta =_{\mathrm{df}} \langle v \cdot \zeta, w \cdot \zeta \rangle$ and $[\langle v, w \rangle] =_{\mathrm{df}} [v] \cup [w]$ where $v, w \in \mathcal{A}ddr$ and $\zeta \in \mathcal{A}_{\mathrm{addr}}$. We use $m, n, o, \ldots$ to range over $\mathcal{L}oc$ in what follows.

3.3 Semantics of CCS^{prio}

The (*operational*) *semantics* of a CCS$^{\mathrm{prio}}$-process $P \in \mathcal{P}$ is given by a labeled transition system $\langle \mathcal{P}, \mathcal{L}oc \times \mathcal{A}, \longrightarrow, P \rangle$. The transition relation $\longrightarrow \subseteq \mathcal{P} \times (\mathcal{L}oc \times \mathcal{A}) \times \mathcal{P}$ is defined in Tables 2 and 3 using Plotkin-style operational rules. We write $P \xrightarrow{m,\gamma} P'$ instead of $\langle P, \langle m, \gamma \rangle, P' \rangle \in \longrightarrow$ and say that P *may engage in action* γ *offered from location* m *and thereafter behave like process* P'. We also write $P \xrightarrow{\gamma} P'$ if there exists a location $m \in \mathcal{L}oc$ such that $P \xrightarrow{m,\gamma} P'$.

Table 1
Initial action sets

$$\underline{\mathrm{I}}_m(C) =_{\mathrm{df}} \underline{\mathrm{I}}_m(P) \text{ where } C \stackrel{\mathrm{def}}{=} P \qquad \underline{\mathrm{I}}_\bullet(\underline{\alpha}.P) =_{\mathrm{df}} \{\underline{\alpha}\}$$

$$\underline{\mathrm{I}}_{m\cdot l}(P+Q) =_{\mathrm{df}} \underline{\mathrm{I}}_m(P) \qquad \underline{\mathrm{I}}_{n\cdot r}(P+Q) =_{\mathrm{df}} \underline{\mathrm{I}}_n(Q)$$

$$\underline{\mathrm{I}}_m(P[f]) =_{\mathrm{df}} \{f(\underline{\alpha}) \mid \underline{\alpha} \in \underline{\mathrm{I}}_m(P)\} \qquad \underline{\mathrm{I}}_m(P \setminus L) =_{\mathrm{df}} \underline{\mathrm{I}}_m(P) \setminus (L \cup \overline{L})$$

$$\underline{\mathrm{I}}_{m\cdot L}(P|Q) =_{\mathrm{df}} \underline{\mathrm{I}}_m(P) \qquad \underline{\mathrm{I}}_{n\cdot R}(P|Q) =_{\mathrm{df}} \underline{\mathrm{I}}_n(Q)$$

$$\underline{\mathrm{I}}_{\langle m\cdot L, n\cdot R\rangle}(P|Q) =_{\mathrm{df}} \{\underline{\tau} \mid \underline{\mathrm{I}}_m(P) \cap \overline{\underline{\mathrm{I}}}_n(Q) \neq \emptyset\}$$

$$\underline{\mathrm{I}}_M(P) =_{\mathrm{df}} \bigcup\{\underline{\mathrm{I}}_m(P) \mid m \in M\} \qquad \underline{\mathrm{II}}_M(P) =_{\mathrm{df}} \underline{\mathrm{I}}_M(P) \setminus \{\underline{\tau}\}$$

$$\underline{\mathrm{I}}(P) =_{\mathrm{df}} \underline{\mathrm{I}}_{\mathcal{L}oc}(P) \qquad \underline{\mathrm{II}}(P) =_{\mathrm{df}} \underline{\mathrm{I}}(P) \setminus \{\underline{\tau}\}$$

The presentation of the operational rules requires *prioritized initial action sets*, which are defined as the least sets satisfying the rules in Table 1. Intuitively, $\underline{\mathrm{I}}_m(P)$ denotes the set of all prioritized initial actions of P from location m. Note that these sets are either empty or contain exactly one initial transition. $\underline{\mathrm{I}}_m(P) = \emptyset$ means that either m is not a location of P or P is incapable of performing a prioritized action at location m. Additionally, let us denote the set of all prioritized initial actions of process P from locations $M \subseteq \mathcal{L}oc$ by $\underline{\mathrm{I}}_M(P)$ and the set of all prioritized initial actions of process P by $\underline{\mathrm{I}}(P)$. We also define analogous sets restricted to visible actions and denote them by $\underline{\mathrm{II}}_M(P)$, and $\underline{\mathrm{II}}(P)$, respectively.

Note that the initial action sets are defined independently from the transition relation $\longrightarrow$. Therefore, $\longrightarrow$ is well-defined (cf. [36]). The side conditions of the operational semantic rules guarantee that a process does not perform an unprioritized action if it can engage in a prioritized synchronization or internal computation, i.e. a $\underline{\tau}$-transition, from a comparable location. Therefore, $\underline{\tau}$-actions have pre-emptive power over unprioritized actions. The reason that prioritized visible actions do *not* have priority over unprioritized actions is that visible actions only indicate the potential of a synchronization, i.e. the potential of progress, whereas internal actions describe *real* progress in our model.

The semantics of $\mathsf{CCS}^{\mathsf{prio}}$ for prioritized transitions is the same as the usual CCS semantics. The difference arises by the side conditions of the rules for unprioritized transitions. The process $\gamma.P$ may engage in action γ and then behaves like P. The *summation operator* $+$ denotes *nondeterministic choice*. The process $P+Q$ may behave like process P (Q) if Q (P) does not pre-empt unprioritized actions by being able to perform a $\underline{\tau}$-transition. Note that priorities arising from different sides of the summation operator are comparable. The *restriction operator* $\setminus L$ prohibits the execution of actions in $L \cup \overline{L}$. Thus, it permits the scoping of actions. $P[f]$ behaves exactly as the process P with

Table 2
Operational semantics for $\mathsf{CCS}^{\mathsf{prio}}$ (Part I)

$\underline{\mathsf{Act}}$	$\dfrac{}{\underline{\alpha}.P \xrightarrow{\bullet,\underline{\alpha}} P}$	$\mathsf{Act}\quad \dfrac{}{\alpha.P \xrightarrow{\bullet,\alpha} P}$
$\underline{\mathsf{Sum1}}$	$\dfrac{P \xrightarrow{m,\underline{\alpha}} P'}{P+Q \xrightarrow{m\cdot l,\underline{\alpha}} P'}$	$\mathsf{Sum1}\quad \dfrac{P \xrightarrow{m,\alpha} P'}{P+Q \xrightarrow{m\cdot l,\alpha} P'}\ \underline{\tau} \notin \underline{\mathrm{I}}(Q)$
$\underline{\mathsf{Sum2}}$	$\dfrac{Q \xrightarrow{n,\underline{\alpha}} Q'}{P+Q \xrightarrow{n\cdot r,\underline{\alpha}} Q'}$	$\mathsf{Sum2}\quad \dfrac{Q \xrightarrow{n,\alpha} Q'}{P+Q \xrightarrow{n\cdot r,\alpha} Q'}\ \underline{\tau} \notin \underline{\mathrm{I}}(P)$
$\underline{\mathsf{Rel}}$	$\dfrac{P \xrightarrow{m,\underline{\alpha}} P'}{P[f] \xrightarrow{m,f(\underline{\alpha})} P'[f]}$	$\mathsf{Rel}\quad \dfrac{P \xrightarrow{m,\alpha} P'}{P[f] \xrightarrow{m,f(\alpha)} P'[f]}$
$\underline{\mathsf{Res}}$	$\dfrac{P \xrightarrow{m,\underline{\alpha}} P'}{P \setminus L \xrightarrow{m,\underline{\alpha}} P' \setminus L}\ \underline{\alpha} \notin L \cup \overline{L}$	$\mathsf{Res}\quad \dfrac{P \xrightarrow{m,\alpha} P'}{P \setminus L \xrightarrow{m,\alpha} P' \setminus L}\ \alpha \notin L \cup \overline{L}$
$\underline{\mathsf{Con}}$	$\dfrac{P \xrightarrow{m,\underline{\alpha}} P'}{C \xrightarrow{m,\underline{\alpha}} P'}\ C \stackrel{\mathrm{def}}{=} P$	$\mathsf{Con}\quad \dfrac{P \xrightarrow{m,\alpha} P'}{C \xrightarrow{m,\alpha} P'}\ C \stackrel{\mathrm{def}}{=} P$

Table 3
Operational semantics for $\mathsf{CCS}^{\mathsf{prio}}$ (Part II)

$\underline{\mathsf{Com1}}$	$\dfrac{P \xrightarrow{m,\underline{\alpha}} P'}{P\vert Q \xrightarrow{m\cdot L,\underline{\alpha}} P'\vert Q}$	$\mathsf{Com1}\quad \dfrac{P \xrightarrow{m,\alpha} P'}{P\vert Q \xrightarrow{m\cdot L,\alpha} P'\vert Q}\ \underline{\mathbb{I}}_{[m]}(P) \cap \overline{\underline{\mathbb{I}}}(Q) = \emptyset$
$\underline{\mathsf{Com2}}$	$\dfrac{Q \xrightarrow{n,\underline{\alpha}} Q'}{P\vert Q \xrightarrow{n\cdot R,\underline{\alpha}} P\vert Q'}$	$\mathsf{Com2}\quad \dfrac{Q \xrightarrow{n,\alpha} Q'}{P\vert Q \xrightarrow{n\cdot R,\alpha} P\vert Q'}\ \underline{\mathbb{I}}_{[n]}(Q) \cap \overline{\underline{\mathbb{I}}}(P) = \emptyset$
$\underline{\mathsf{Com3}}$	$\dfrac{P \xrightarrow{m,\underline{a}} P' \quad Q \xrightarrow{n,\overline{\underline{a}}} Q'}{P\vert Q \xrightarrow{\langle m\cdot L, n\cdot R\rangle,\underline{\tau}} P'\vert Q'}$	$\mathsf{Com3}\quad \dfrac{P \xrightarrow{m,a} P' \quad Q \xrightarrow{n,\overline{a}} Q'}{P\vert Q \xrightarrow{\langle m\cdot L, n\cdot R\rangle,\tau} P'\vert Q'}\ \underline{\mathbb{I}}_{[m]}(P) \cap \overline{\underline{\mathbb{I}}}(Q) = \emptyset \wedge \underline{\mathbb{I}}_{[n]}(Q) \cap \overline{\underline{\mathbb{I}}}(P) = \emptyset$

actions renamed according to the *relabeling* f. The process $P|Q$ stands for the *interleaved parallel composition* of P and Q with synchronized communication on complementary actions resulting in the internal action τ or $\underline{\tau}$. Since locations on different sides of a parallel operator are incomparable, $\underline{\tau}$'s arising from a location of P (Q) cannot pre-empt the execution of an action, even an unprioritized one, of Q (P). Only if P (Q) engages in a prioritized synchronization with Q (P) can unprioritized actions from a comparable location of P (Q) be pre-empted. For example, the initial a-transition gets pre-empted in the process $(a.\mathbf{0}+\underline{b}.\mathbf{0})\,|\,\overline{\underline{b}}.\mathbf{0}$ but not in the process $(a.\mathbf{0}\,|\,\underline{b}.\mathbf{0})\,|\,\overline{\underline{b}}.\mathbf{0}$. Also observe that actions a and $\overline{\underline{a}}$ cannot synchronize. Thus, one may view the sets of prioritized and unprioritized action labels as being disjoint. Finally, $C \stackrel{\mathrm{def}}{=} P$ denotes a *constant definition*, i.e. C is a recursively defined process that behaves as a distinguished solution of the equation $C = P$.

In what follows we use $\mathcal{S}(P)$ to denote the unprioritized sort of P and $\underline{\mathcal{S}(P)}$ for its prioritized sort. The next property of $\mathsf{CCS}^{\mathsf{prio}}$-processes is important for the proofs of our main theorems.

Lemma 2 (Finite Sorts)
Let $P \in \mathcal{P}$ be a $\mathsf{CCS}^{\mathsf{prio}}$-process. Then $\mathcal{S}(P)$ and $\underline{\mathcal{S}(P)}$ are finite.

The validity of this lemma is an immediate consequence of the fact that relabelings f satisfy the condition $|\{\gamma \mid f(\gamma) \neq \gamma\}| < \infty$.

4 Prioritized Strong Bisimulation

In this section we present an equivalence relation for $\mathsf{CCS}^{\mathsf{prio}}$-processes that is based on bisimulation [32]. Our aim is to characterize the largest congruence contained in the "naive" adaptation of strong bisimulation [24] to our framework obtained by ignoring location information.

Definition 3 (Naive Prioritized Strong Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is called naive prioritized strong bisimulation *if for every $\langle P, Q\rangle \in \mathcal{R}$ and $\gamma \in \mathcal{A}$ the following condition holds.*

$$P \stackrel{\gamma}{\rightarrow} P' \text{ implies } \exists Q'.\, Q \stackrel{\gamma}{\rightarrow} Q' \text{ and } \langle P', Q'\rangle \in \mathcal{R} \ .$$

We write $P \simeq Q$ if there exists a naive prioritized strong bisimulation $\mathcal{R}$ such that $\langle P, Q\rangle \in \mathcal{R}$.

It is straightforward to establish that $\simeq$ is the *largest* naive prioritized strong bisimulation and that $\simeq$ is an equivalence relation. Unfortunately, $\simeq$ is *not* a congruence, which is a necessary requirement for an equivalence to be suitable for compositional reasoning. The lack of compositionality is demonstrated by the following example, which embodies the traditional view that "parallelism = nondeterminism." We have $a.\underline{b}.\mathbf{0} + \underline{b}.a.\mathbf{0} \simeq a.\mathbf{0} \mid \underline{b}.\mathbf{0}$ but $(a.\underline{b}.\mathbf{0} + \underline{b}.a.\mathbf{0}) \mid \overline{\underline{b}}.\mathbf{0} \not\simeq (a.\mathbf{0} \mid \underline{b}.\mathbf{0}) \mid \overline{\underline{b}}.\mathbf{0}$, since the latter can perform an a-transition while the corresponding a-transition of the former process is pre-empted because the right process in the summation can engage in a prioritized communication. The above observation is not surprising since the distribution of processes influences the pre-emption of transitions and, consequently, the bisimulation. We also have the following fact from universal algebra.

Theorem 4 *Let X be an equivalence relation over an algebra $\Re$. Then the largest congruence X^+ in X exists and*

$$X^+ = \{\langle P, Q\rangle \mid \forall\, \Re\text{-}contexts\, C[].\, \langle C[P], C[Q]\rangle \in X\} \ .$$

Consequently, we know that $\simeq$ includes a largest congruence $\simeq^+$ for $\mathsf{CCS}^{\mathsf{prio}}$. In the remainder of this subsection we develop an operational characterization of $\simeq^+$. To do so we need to take the *local* pre-emption of processes into account.

Definition 5 (Prioritized Strong Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a prioritized strong bisimulation *if for every $\langle P, Q\rangle \in \mathcal{R}$, $\alpha \in A$, $\underline{\alpha} \in \underline{A}$, and $m \in \mathcal{L}oc$ the following conditions hold.*

(i) $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ *implies* $\exists Q'. Q \stackrel{\underline{\alpha}}{\Longrightarrow} Q'$ *and* $\langle P', Q'\rangle \in \mathcal{R}$.
(ii) $P \stackrel{m,\alpha}{\longrightarrow} P'$ *implies* $\exists Q', n. Q \stackrel{n,\alpha}{\longrightarrow} Q', \underline{\mathbb{I}}_{[n]}(Q) \subseteq \underline{\mathbb{I}}_{[m]}(P)$, *and* $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \simeq^{\mathrm{l}} Q$ if there exists a prioritized strong bisimulation $\mathcal{R}$ such that $\langle P, Q\rangle \in \mathcal{R}$.

The difference between this definition and that of $\simeq$ is the additional requirement concerning the initial action sets, parameterized with the appropriate locations, in the condition for unprioritized transitions. Intuitively, the prioritized initial action set of a process with respect to some location is a measure of the pre-emptive power of the process relative to that location. Thus, the second condition of Def. 5 states that an unprioritized action α from some location m of the process P must be matched by the same action from some location n of Q and that the pre-emptive power of Q relative to n is at most as strong as the pre-emptive power of P relative to m.

Proposition 6 *The relation $\simeq^{\mathrm{l}}$ is a congruence, i.e. for all $\mathsf{CCS}^{\mathsf{prio}}$-contexts $C[\,]$ we have: $P \simeq^{\mathrm{l}} Q$ implies $C[P] \simeq^{\mathrm{l}} C[Q]$.*

Now, we can state the main theorem of this section.

Theorem 7 *The congruence $\simeq^{\mathrm{l}}$ is the* largest *congruence contained in $\simeq$.*

PROOF. By Theorem 4 the largest congruence $\simeq^+$ in $\simeq$ exists and is characterized by $P \simeq^+ Q$ iff $\forall \mathsf{CCS}^{\mathsf{prio}}$-contexts $C[]. C[P] \simeq C[Q]$.

It is straightforward to show that $\simeq^{\mathrm{l}} \subseteq \simeq$; one need only prove that $\simeq^{\mathrm{l}}$ is a naive prioritized strong bisimulation. The proof is standard and is omitted. Also, since $\simeq^{\mathrm{l}}$ is a congruence we know that $\simeq^{\mathrm{l}} \subseteq \simeq^+$. In order to prove the inclusion $\simeq^+ \subseteq \simeq^{\mathrm{l}}$ it suffices to show that $\mathcal{R} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \simeq C_{PQ}[Q]\}$ is a prioritized strong bisimulation for some $\mathsf{CCS}^{\mathsf{prio}}$-context $C_{PQ}[]$. For our purposes we define $C_{PQ}[X] =_{\mathrm{df}} X \mid H_{PQ}$ for $P, Q \in \mathcal{P}$ where

$$H_{PQ} \stackrel{\mathrm{def}}{=} \sum_{L \subseteq \underline{\mathcal{S}(P)} \cup \underline{\mathcal{S}(Q)}} \underline{\tau}.(\underline{d}_L.H_{PQ} + \underline{D}_L)$$

and $\underline{D}_L =_{\mathrm{df}} \sum_{\underline{c} \in L} \underline{c}.\mathbf{0}$. Note that $\sum$ is the extension of the associative binary operator $+$ to finitely many operands and that H_{PQ} is well-defined by Lemma 2.

We assume that $\underline{d}_L, \overline{\underline{d}}_L \notin \underline{\mathcal{S}(P)} \cup \underline{\mathcal{S}(Q)}$. Such $\underline{d}_L$'s exist because the prioritized sort of a process is finite according to Lemma 2. Note that the context $C_{PQ}[X]$ is adapted from [28].

Now, let $P, Q \in \mathcal{P}$ satisfying $C_{PQ}[P] \simeq C_{PQ}[Q]$ and $P \xrightarrow{m,\alpha} P'$. Therefore, $C_{PQ}[P]$ can engage in the transitions illustrated in the left hand side of Fig. 2 where $L = \{\overline{\underline{c}} \mid \underline{c} \in (\underline{\mathcal{S}(P)} \cup \underline{\mathcal{S}(Q)}) \setminus \underline{\mathbb{I}}_{[m]}(P)\}$. Since $C_{PQ}[P] \simeq C_{PQ}[Q]$, the process $C_{PQ}[Q]$ has to match each step.

$$
\begin{array}{lcl}
C_{PQ}[P] \equiv P \mid H_{PQ} & \simeq & Q \mid H_{PQ} \equiv C_{PQ}[Q] \\
\Big\downarrow \underline{\tau} & & \Big\downarrow \underline{\tau} \\
P \mid \underline{d}_L.H_{PQ} + \underline{D}_L & \simeq & Q \mid \underline{d}_L.H_{PQ} + \underline{D}_L \\
\Big\downarrow m \cdot L, \alpha & & \Big\downarrow n \cdot L, \alpha \\
P' \mid \underline{d}_L.H_{PQ} + \underline{D}_L & \simeq & Q' \mid \underline{d}_L.H_{PQ} + \underline{D}_L \\
\Big\downarrow \underline{d}_L & & \Big\downarrow \underline{d}_L \\
C_{PQ}[P'] \equiv P' \mid H_{PQ} & \simeq & Q' \mid H_{PQ} \equiv C_{PQ}[Q']
\end{array}
$$

Fig. 2. Largest congruence proof - illustration

In order to be able to match the first step, $C_{PQ}[Q]$ has to choose exactly the same branch of H_{PQ} yielding to the process $\underline{d}_L.H_{PQ} + \underline{D}_L$, because only this process is able to execute the distinguished action $\underline{d}_L$. For matching the second step, the process Q must be able to perform an α-transition from some location $n \in \mathcal{L}oc$. According to our semantics for parallel composition, the condition $\underline{\mathbb{I}}_{[n]}(Q) \cap \overline{\underline{\mathbb{I}}}(\underline{d}_L.H_{PQ} + \underline{D}_L) = \emptyset$ has to be satisfied. Because of the choice of L, this implies $\underline{\mathbb{I}}_{[n]}(Q) \subseteq \underline{\mathbb{I}}_{[m]}(P)$. The match of the third step, observing action $\underline{d}_L$, is straightforward. Thus, Fig. 2 shows the existence of some $Q' \in \mathcal{P}$ satisfying $C_{PQ}[P'] \simeq C_{PQ}[Q']$. Since $\underline{\mathcal{S}(P')} \subseteq \underline{\mathcal{S}(P)}$ and $\underline{\mathcal{S}(Q')} \subseteq \underline{\mathcal{S}(Q)}$ it follows that $C_{P'Q'}[P'] \simeq C_{P'Q'}[Q']$, as desired.

The case where P performs a prioritized transition needs no special attention since the condition of Def. 3 and Cond. (i) of Def. 5 are identical in this case. Summarizing, we have shown that all conditions of Def. 5 are satisfied, and we may conclude that $\mathcal{R}$ is a prioritized strong bisimulation. Hence, $P \simeq^{\mathrm{l}} Q$, which completes the proof. $\square$

Axiomatization of $\simeq^{\mathrm{l}}$

In this section we give an axiomatization of $\simeq^{\mathrm{l}}$ for *finite* processes, i.e. processes that do not contain recursion. In order to develop the axiomatization, we add a new binary summation operator $\oplus$ to the process algebra $\mathsf{CCS}^{\mathsf{prio}}$. This operator is called *distributed summation* and needed for giving an expansion axiom (cf. Axiom (E) in Table 4). Its operational semantics is the following.

$$\underline{\mathsf{dSum1}}\ \frac{P \stackrel{m,\underline{\alpha}}{\Longrightarrow} P'}{P \oplus Q \stackrel{m\cdot L,\underline{\alpha}}{\Longrightarrow} P'} \qquad \mathsf{dSum1}\ \frac{P \stackrel{m,\underline{\alpha}}{\longrightarrow} P'}{P \oplus Q \stackrel{m\cdot L,\underline{\alpha}}{\longrightarrow} P'}$$

$$\underline{\mathsf{dSum2}}\ \frac{Q \stackrel{n,\underline{\alpha}}{\Longrightarrow} Q'}{P \oplus Q \stackrel{n\cdot R,\underline{\alpha}}{\Longrightarrow} Q'} \qquad \mathsf{dSum2}\ \frac{Q \stackrel{n,\underline{\alpha}}{\longrightarrow} Q'}{P \oplus Q \stackrel{n\cdot R,\underline{\alpha}}{\longrightarrow} Q'}$$

Now, we turn to the axiom system for prioritized strong bisimulation. We write $\vdash_E P = Q$ if P can be rewritten to Q using the axioms in Tables 4 and 5. Axioms ($\mathsf{lc1}$), ($\mathsf{D1}$), ($\mathsf{S2}$), and ($\mathsf{S3}$) involve side conditions. Regarding Axiom ($\mathsf{lc1}$), we introduce the unary predicate $\natural$ over process terms of the form $\sum_{j\in J}\gamma_j.x_j$ for some nonempty index set J together with the following proof rules: (i) $\natural\underline{\alpha}.x$ and (ii) $\natural x$ and $\natural y$ implies $\natural(x+y)$. Intuitively, $\natural(\sum_{j\in J}\gamma_j.x_j)$ if and only if $\gamma_j \in \underline{A}$ for all $j \in J$. The relation $\sqsubseteq_{\mathrm{i}}$ is the precongruence on finite processes generated from the axioms presented in Table 6 using the laws of inequational reasoning. Its meaning is precised by Lemma 8 below. We write $\vdash_I P \sqsubseteq_{\mathrm{i}} Q$ if P can be related to Q by Axioms ($\mathsf{iC1}$), ($\mathsf{iC2}$), and ($\mathsf{iC3}$), and notate $\vdash_I P =_{\mathrm{i}} Q$ if $\vdash_I P \sqsubseteq_{\mathrm{i}} Q$ and $\vdash_I Q \sqsubseteq_{\mathrm{i}} P$. The axioms in Table 4 are basically those given in [7] and augmented with the corresponding axioms for the distributed summation operator. Moreover, the expansion axiom has been adapted for our algebra (cf. Axiom (E) where $\sum$ is the indexed version of $+$, and $\bigoplus$ is the indexed version of $\oplus$). The axioms in Table 5 are new and show how we may "restructure" locations. They deal with the *distributivity* of the summation operators (Axioms ($\mathsf{D1}$) and ($\mathsf{D2}$)), the *interchangeability* of the summation operators (Axioms ($\mathsf{lc1}$) and ($\mathsf{lc2}$)), and the *saturation* of locations (Axioms ($\mathsf{S1}$), ($\mathsf{S2}$), and ($\mathsf{S3}$)), respectively.

The following lemma presents a semantic interpretation of $\vdash_I P \sqsubseteq_{\mathrm{i}} Q$ that is essential for the soundness and completeness proof of our axiomatization. It uses the notation $\vdash_A P = Q$ meaning that P can be rewritten to Q by using Axioms ($\mathsf{A1}$)–($\mathsf{A4}$) only. Hence, $\vdash_A P = Q$ implies $\vdash_E P = Q$.

Lemma 8 (Semantic Interpretation of $\sqsubseteq_{\mathrm{i}}$)

(i) Let $\vdash_I P \sqsubseteq_{\mathrm{i}} Q$. *Then,* $\underline{\mathrm{I}}(P) \subseteq \underline{\mathrm{I}}(Q)$, *and* $\underline{\tau} \in \underline{\mathrm{I}}(P)$ *if and only if* $\underline{\tau} \in \underline{\mathrm{I}}(Q)$.
(ii) Let $P \equiv \sum_{i=1}^{m}\gamma_i.P_i$ *and* $Q \equiv \sum_{j=1}^{n}\delta_j.Q_j$ *be finite processes such that*

Table 4
Axiomatization of $\simeq^{\mathrm{l}}$ (Axioms E)

(A1)	$x + y = y + x$	(iA1)	$x \oplus y = y \oplus x$
(A2)	$x + (y + z) = (x + y) + z$	(iA2)	$x \oplus (y \oplus z) = (x \oplus y) \oplus z$
(A3)	$x + x = x$	(iA3)	$x \oplus x = x$
(A4)	$x + \mathbf{0} = x$	(iA4)	$x \oplus \mathbf{0} = x$

(P) $\quad \underline{\tau}.x + \alpha.y = \underline{\tau}.x$

(E) $\quad P \equiv \bigoplus_i \sum_j \gamma_{ij}.P_{ij}$ and $Q \equiv \bigoplus_k \sum_l \delta_{kl}.Q_{kl}$ implies $P \mid Q =$

$$\begin{aligned}
&\bigoplus_i \textstyle\sum_j(\gamma_{ij}.(P_{ij} \mid Q) + \sum_k \sum_l \{\tau.(P_{ij} \mid Q_{kl}) \mid \gamma_{ij} = \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in A\} \\
&\qquad + \textstyle\sum_k \sum_l \{\underline{\tau}.(P_{ij} \mid Q_{kl}) \mid \gamma_{ij} = \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in \underline{A}\}) \oplus \\
&\bigoplus_k \textstyle\sum_l(\delta_{kl}.(P \mid Q_{kl}) + \sum_i \sum_j \{\tau.(P_{ij} \mid Q_{kl}) \mid \gamma_{ij} = \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in A\} \\
&\qquad + \textstyle\sum_i \sum_j \{\underline{\tau}.(P_{ij} \mid Q_{kl}) \mid \gamma_{ij} = \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in \underline{A}\})
\end{aligned}$$

(Res1)	$\mathbf{0} \setminus L = \mathbf{0}$	
(Res2)	$(\gamma.x) \setminus L = \mathbf{0}$	$(\gamma \in L \cup \overline{L})$
(Res3)	$(\gamma.x) \setminus L = \gamma.(x \setminus L)$	$(\gamma \notin L \cup \overline{L})$
(Res4)	$(x + y) \setminus L = (x \setminus L) + (y \setminus L)$	
(iRes4)	$(x \oplus y) \setminus L = (x \setminus L) \oplus (y \setminus L)$	

(Rel1)	$\mathbf{0}[f] = \mathbf{0}$	(Rel3)	$(x + y)[f] = x[f] + y[f]$
(Rel2)	$(\gamma.x)[f] = f(\gamma).(x[f])$	(iRel3)	$(x \oplus y)[f] = x[f] \oplus y[f]$

$\underline{\mathrm{I}}(P) \subseteq \underline{\mathrm{I}}(Q)$, and $\underline{\tau} \in \underline{\mathrm{I}}(P)$ if and only if $\underline{\tau} \in \underline{\mathrm{I}}(Q)$. Then there exist processes P' and Q' such that $\vdash_A P' = P$, $\vdash_A Q' = Q$, and $\vdash_I P' \sqsubseteq_{\mathrm{i}} Q'$. The same holds if we replace "$\subseteq$" by "$=$" and "$\sqsubseteq_{\mathrm{i}}$" by "$=_{\mathrm{i}}$."

The proof of this lemma is quite straightforward and, therefore, omitted. The following theorem states the soundness of our axiom system.

Theorem 9 (Soundness)
For $P, Q \in \mathcal{P}$ satisfying $\vdash_E P = Q$ we have $P \simeq^{\mathrm{l}} Q$.

Formally, the soundness of the axioms can be shown by constructing a prioritized strong bisimulation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ for each axiom $P = Q$ such that

Table 5
Axioms E (continued)

(D1) $(x \oplus x') + (y \oplus y') = ((x \oplus x') + y') \oplus ((y \oplus y') + x')$
$(\vdash_I x \sqsubseteq_{\mathrm{i}} x' \,,\, \vdash_I y \sqsubseteq_{\mathrm{i}} y')$

(D2) $(x \oplus y) + \underline{\alpha}.z = (x + \underline{\alpha}.z) \oplus (y + \underline{\alpha}.z)$

(Ic1) $x \oplus \underline{\alpha}.y = x + \underline{\alpha}.y$ $\quad (\natural x)$

(Ic2) $(\underline{\alpha}.x + y) = (\underline{\alpha}.x + y) \oplus \underline{\alpha}.x$

(S1) $(x + \underline{\alpha}.y) \oplus (x' + \underline{\alpha}.y') = (x + \underline{\alpha}.y + \underline{\alpha}.y') \oplus (x' + \underline{\alpha}.y')$

(S2) $(x + \alpha.z) \oplus (y + \alpha.z) = (x + \alpha.z) \oplus y$ $\quad (\vdash_I x \sqsubseteq_{\mathrm{i}} y)$

(S3) $x \oplus y = x + y$ $\quad (\vdash_I x =_{\mathrm{i}} y)$

Table 6
Axiomatization of $\sqsubseteq_{\mathrm{i}}$ (Axioms I)

(iC1) $\underline{\alpha}.x \sqsubseteq_{\mathrm{i}} \underline{\alpha}.y$ $\qquad$ (iC2) $\mathbf{0} \sqsubseteq_{\mathrm{i}} \nu.x \quad \nu \in \mathcal{A} \setminus \{\underline{\tau}\}$ $\qquad$ (iC3) $\alpha.x \sqsubseteq_{\mathrm{i}} \mathbf{0}$

$\langle P, Q \rangle \in \mathcal{R}$. This is obvious for all axioms beside Axioms (D1), (S2), and (S3). Those axioms are only true if the appropriate side condition is satisfied. However, their soundness can easily be established by using the semantic interpretation of the syntactic side conditions given in Lemma 8(i). In order to prove our axiomatization complete, we introduce a notion of *normal form* of processes that is based on the following definition.

Definition 10 (Summation Form)
A process P is in summation form *if it has the form $P \equiv \bigoplus_{i=1}^{m} \sum_{j=1}^{n_i} \gamma_{ij}.P_{ij}$ where $m, n_i \in \mathbb{N}$ and the processes P_{ij} are again in summation form. Per definition, $\mathbf{0}$ is in summation form.*

Intuitively, P is distributed throughout m incomparable locations which themselves consist of n_i comparable locations, $1 \leq i \leq m$. The following proposition states that every finite process can be rewritten into summation form.

Proposition 11 *For every finite process P there exists a process S in summation form such that $\vdash_E S = P$.*

Definition 12 (Normal Form)
Let $P \equiv \bigoplus_{i=1}^{m} \sum_{j=1}^{n_i} \gamma_{ij}.P_{ij}$ be in summation form. We define $\underline{\gamma}_{i*} =_{df} \{\gamma_{ij} \mid 1 \leq j \leq n_i\} \cap \underline{A}$. The process P is said to be in normal form *if the following properties hold.*

(i) $\emptyset \subseteq L \subseteq \underline{\mathrm{I}}(P)$ *implies* $\exists i.\, \underline{\gamma}_{i*} = L$.
(ii) $\underline{\gamma}_{i*} = \underline{\gamma}_{k*}$ *implies* $i = k$.
(iii) $\gamma_{ij} \in A$ *implies* $\forall 1 \leq l \leq n_i.\, \gamma_{il} \not\equiv \underline{\tau}$.
(iv) $\gamma_{ij} \equiv \gamma_{kl} \equiv \underline{\alpha}$ *implies* $\exists j'.\, P_{ij'} \equiv P_{kl}$ *and* $\gamma_{ij'} \equiv \underline{\alpha}$.

Intuitively, Cond. (i) and (ii) state that a term P in normal form contains exactly one incomparable (or "outer") summand for each possible pre-emption potential, i.e. for each subset of the prioritized initial actions of P . Cond. (iii) reflects our notion of pre-emption: an outer summand contains no unprioritized initial actions when it also includes a prioritized internal action. The last condition requires outer summands to be "saturated" in a certain sense with respect to prioritized actions (cf. Axiom (S1)). The following proposition plays a key role in the completeness proof of our axiomatization for finite processes.

Proposition 13 *If P is a finite process, then there exists a normal form N such that $\vdash_E N = P$.*

The proof of this proposition uses induction on the *depth* of a finite process in summation form which is defined to be the maximum number of nested prefixes in that process, as usual. Moreover, it needs the following lemma which can easily be established.

Lemma 14 *Let $P \equiv \bigoplus_{i=1}^{m} \sum_{j=1}^{n_i} \gamma_{ij}.P_{ij}$. Then $\vdash_E P = P \oplus \underline{P}$ and $\vdash_I P \sqsubseteq_{\mathrm{i}} \underline{P}$ where $\underline{P} \equiv \bigoplus_{i=1}^{m} \sum_{j=1}^{n_i} \underline{P}_{ij}$, $\underline{P}_{ij} \equiv \gamma_{ij}.P_{ij}$ if $\gamma_{ij} \in \underline{A}$, and $\underline{P}_{ij} \equiv \boldsymbol{0}$ if $\gamma_{ij} \in A$.*

PROOF of Prop. 13. Let P be a finite process. By Prop. 11 we know of the existence of a process $S \equiv \bigoplus_{i=1}^{m} S_{i*}$, where $S_{i*} \equiv \sum_{j=1}^{n_i} \gamma_{ij}.S_{ij}$, in summation form such that $\vdash_E S = P$. Therefore, it remains to establish that S can be equationally rewritten into a process N in normal form such that $\vdash_E N = S$. This is done by induction on the depth of S . If the depth of S is 0 , then we apply Axiom (iA3) in order to obtain $\vdash_E S = \mathbf{0}$ where $\mathbf{0}$ is obviously in normal form. For the induction step let the depth of S be greater than 0 . Now, we proceed according to the following steps.

(i) We use Lemma 14 to rewrite S to $S \oplus \underline{S}$, where $\underline{S} \equiv \bigoplus_{i=1}^{m} \sum_{j=1}^{n_i} \underline{S}_{ij}$, and $\underline{S}_{ij} \equiv \gamma_{ij}.S_{ij}$, if $\gamma_{ij} \in \underline{A}$, and $\underline{S}_{ij} \equiv \mathbf{0}$, otherwise. We switch the distributed summation operators in $\underline{S}$ into usual summation operators using Axiom (Ic1), and Axioms (A2), (iA2), (A4), and (iA4), if needed. By Axioms (iA1)–(iA3) we duplicate summands of S and regroup them such that for each $\emptyset \neq L \subseteq \underline{\mathrm{I}}(P)$ there exists a distributed summand of the form $\bigoplus_{j \in J} \underline{\alpha}_j.P_j$ satisfying $L = \{\underline{\alpha}_j \mid j \in J\}$. Using Axiom (Ic1) each of the above mentioned distributed summands is rewritten into a summand $\sum_{j \in J} \underline{\alpha}_j.P_j$. Finally, we apply Axiom (iA4) once obtaining the distributed summand $\mathbf{0}$ which fulfills the required condition for $L = \emptyset$. This concludes the establishment of Property (i).

(ii) Whenever $i \neq k$ and $\underline{\gamma_{i*}} = \underline{\gamma_{k*}}$, where $1 \leq i, k \leq m$, merge the terms P_{i*} and P_{k*} into one by applying the following steps. Use Axioms (iA1) and (iA2) to restructure the distributed summands of P such that the terms P_{i*} and P_{k*} are standing side by side. According to Lemma 8(ii) there exist processes P'_{i*} and P'_{k*} such that $\vdash_A P'_{i*} = P_{i*}$, $\vdash_A P'_{k*} = P_{k*}$, and $\vdash_E P'_{i*} =_{\mathrm{i}} P'_{k*}$. Thus, we may rewrite $P_{i*} \oplus P_{k*}$ to $P'_{i*} \oplus P'_{k*}$ and apply Axiom (S3) to substitute $P'_{i*} \oplus P'_{k*}$ by $P'_{i*} + P'_{k*}$. Repeat the above procedure as often as possible. Hence, Property (ii) is established.

(iii) We use Axiom (P) in order to obtain Property (iii).

(iv) As long as Property (iv) is not satisfied, i.e. there exists $\gamma_{ij} \in \underline{\gamma_{k*}}$ for some $1 \leq k \leq m$ and $i \neq k$ but there is no $1 \leq l \leq n_k$ such that $\gamma_{kl}.P_{kl} \equiv \gamma_{ij}.P_{ij}$, then apply Axiom (S1), and possibly Axioms (A1), (A2), (iA1), and (iA2), to add the term $\gamma_{ij}.P_{ij}$ to P_{k*} as an additional summand. Thus, Property (iv) is achieved.

This concludes the proof of the induction step and of the proposition. □

Rewriting a process in its normal form requires restructuring its locations. After this is done, standard techniques used in CCS (cf. [24]) can be applied in order to show our axiomatization complete.

Theorem 15 (Completeness)
For finite processes $P, Q \in \mathcal{P}$ satisfying $P \simeq^{\mathrm{l}} Q$ we have $\vdash_E P = Q$.

PROOF. Let P and Q be finite processes such that $P \simeq^{\mathrm{l}} Q$. By Prop. 13 we may assume w.l.o.g. that P and Q are in normal form, i.e. $P \equiv \bigoplus_{i=1}^{m} P_{i*}$ where $P_{i*} \equiv \sum_{j=1}^{n_i} \gamma_{ij}.P_{ij}$, and $Q \equiv \bigoplus_{k=1}^{r} Q_{k*}$ where $Q_{k*} \equiv \sum_{l=1}^{s_k} \delta_{kl}.Q_{kl}$, and Properties (i)–(iv) of Def. 12 are satisfied. We reason by induction on the maximum of the depths of P and Q.

Induction base: If the maximum depth is 0 then P and Q are of the form $\bigoplus_{i=1}^{m} \mathbf{0}$ and $\bigoplus_{k=1}^{r} \mathbf{0}$, respectively. Both processes are rewritten to $\mathbf{0}$ by applying Axiom (iA3), and since $\vdash_E \mathbf{0} = \mathbf{0}$, we are done.

Induction step: Here, we use the following additional properties.

(I) We assume w.l.o.g. that $P_{ij} \equiv Q_{kl}$ for some $1 \leq i \leq m$, $1 \leq j \leq n_i$, $1 \leq k \leq r$, and $1 \leq l \leq s_k$ whenever $P_{ij} \simeq^{\mathrm{l}} Q_{kl}$. The reason for being able to assume this is that the maximum of the depths of the processes P_{ij} and Q_{kl} is smaller than that of P and Q. Hence the induction hypothesis is applicable, i.e. $\vdash_E P_{ij} = Q_{kl}$ and we may substitute P_{ij} for Q_{kl} since substitution is part of equational reasoning.

(II) Moreover, we may assume w.l.o.g. that P satisfies the following property.

$$\gamma_{ij}.P_{ij} \equiv \gamma_{kl}.P_{kl}\,,\gamma_{ij} \in A\,,\text{ and } i \neq k \text{ imply } \underline{\gamma}_{i*} \not\subseteq \underline{\gamma}_{k*}\ .$$

Otherwise, we apply Lemma 8(ii) that allows us to substitute P_{i*} and P_{k*} by processes P'_{i*} and P'_{k*}, respectively, satisfying $\vdash_A P'_{i*} = P_{i*}$, $\vdash_A P'_{k*} = P_{k*}$, and $\vdash_I P'_{i*} \sqsubseteq_{\mathsf{i}} P'_{k*}$. Now, we apply Axioms (S2), (A1), (A2), (iA1), and (iA2) to achieve the above mentioned property. Note that this transformation does not destroy the normal forms of P and Q, and Property (I). Similarly, we may assume that Q fulfills Property (II).

Proof goal: In the main part of the induction step we show the existence of a bijection $\Phi : \{1,\ldots,m\} \longrightarrow \{1,\ldots,r\}$ such that $\forall 1 \leq i \leq m.\ \vdash_E P_{i*} = Q_{\Phi(i)*}$. Thus, it follows that $\vdash_E P = Q$ by possibly using Axioms (iA1) and (iA2) to reorder and regroup distributed summands.

Main part of the induction step: Since $P \simeq^{\mathrm{l}} Q$ we have $\underline{\mathrm{I}}(P) = \underline{\mathrm{I}}(Q)$. By Properties (i) and (ii) of Def. 12 we may conclude that $m = r$. Moreover, the mapping $\Phi : \{1,\ldots,m\} \longrightarrow \{1,\ldots,r\}$ defined by $\Phi(i) =_{\mathrm{df}} k$ where $\underline{\gamma}_{i*} = \underline{\delta}_{k*}$ is a bijection. It remains to show that $\vdash_E P_{i*} = Q_{\Phi(i)*}$ for some arbitrary $1 \leq i \leq m$. Let $k =_{\mathrm{df}} \Phi(i)$, i.e. $\underline{\gamma}_{i*} = \underline{\delta}_{k*}$. We show that every summand of P_{i*} is syntactically identical to a summand of Q_{k*}. For every $1 \leq j \leq n_i$ we have $P_{i*} \stackrel{\underline{\gamma_{ij}}}{\rightarrow} P_{ij}$ because of Property (iii) of Def. 12.

Case 1: Let $\gamma_{ij} \equiv \underline{\alpha} \in \underline{A}$ for some $j \in \{1,\ldots,n_i\}$. We may derive $P \stackrel{\underline{\alpha}}{\Rightarrow} P_{ij}$ according to our operational rules and the definition of P. Since $P \simeq^{\mathrm{l}} Q$ there exists numbers k' and l', where $1 \leq k' \leq r$ and $1 \leq l' \leq s_{k'}$, such that $Q \stackrel{\underline{\alpha}}{\Rightarrow} Q_{k'l'}$, $\underline{\alpha} \equiv \delta_{k'l'}$, and $P_{ij} \simeq^{\mathrm{l}} Q_{k'l'}$. By Property (I) we have $P_{ij} \equiv Q_{k'l'}$ and, thus, $\gamma_{ij}.P_{ij} \equiv \delta_{k'l'}.Q_{k'l'}$. Moreover, the summand $\delta_{k'l'}.Q_{k'l'}$ syntactically equals with a summand $\delta_{kl}.Q_{kl}$ for some $1 \leq l \leq s_k$ by Property (iv) of Def. 12 since $\delta_{k'l'} \in \underline{\delta}_{k*} = \underline{\gamma}_{i*}$.

Case 2: Let $\gamma_{ij} \equiv \alpha \in A$ for some $j \in \{1,\ldots,n_i\}$. By the definition of P and the operational rules we also have $P \stackrel{\alpha}{\rightarrow} P_{ij}$. Note that we do not include the location of the action $\alpha \in A$ in the label since the corresponding prioritized initial action set is already determined by the index i. Because of $P \simeq^{\mathrm{l}} Q$ we know of the existence of numbers k' and l, where $1 \leq k' \leq r$ and $1 \leq l \leq s_{k'}$, such that $Q \stackrel{\alpha}{\rightarrow} Q_{k'l}$, $\alpha \equiv \delta_{k'l}$, $\underline{\delta}_{k'*} \subseteq \underline{\gamma}_{i*}$, and $P_{ij} \simeq^{\mathrm{l}} Q_{k'l}$. Observe that the inclusion $\underline{\delta}_{k'*} \subseteq \underline{\gamma}_{i*}$ is equivalent to our condition of prioritized initial action set inclusion (cf. Cond. (ii) of Def. 5) by Property (iii) of normal forms. Because of Property (I) we may conclude that $P_{ij} \equiv Q_{k'l}$ and, thus, $\gamma_{ij}.P_{ij} \equiv \delta_{k'l}.Q_{k'l}$. It remains to establish $k' = k$. Since $P \simeq^{\mathrm{l}} Q$ we know of the existence of numbers i' and j', where $1 \leq i' \leq m$ and $1 \leq j' \leq n_{i'}$, such that $P \stackrel{\alpha}{\rightarrow} P_{i'j'}$, $\alpha \equiv \gamma_{i'j'}$, $\underline{\gamma}_{i'*} \subseteq \underline{\delta}_{k'*}$, and $P_{i'j'} \simeq^{\mathrm{l}} Q_{k'l}$. Hence, $P_{i'j'} \equiv Q_{k'l}$ by Property (I) and, thus, $\gamma_{i'j'}.P_{i'j'} \equiv \delta_{k'l}.Q_{k'l}$. Together with $\gamma_{ij}.P_{ij} \equiv \delta_{k'l}.Q_{k'l}$ we conclude $\gamma_{i'j'}.P_{i'j'} \equiv \gamma_{ij}.P_{ij}$. Moreover, we have established $\underline{\gamma}_{i'*} \subseteq \underline{\delta}_{k'*} \subseteq \underline{\gamma}_{i*}$. This

establishes $i = i'$ by Property (II), whence $\underline{\delta}_{k'*} = \underline{\gamma}_{i*} = \underline{\delta}_{k*}$. Now, we may conclude $k = k'$ by Property (ii) of normal forms, as desired.

Similarly, every summand of Q_{k*} is syntactically equal to a summand of P_{i*}. Hence, $\vdash_E P_{i*} = Q_{k*}$ by using Axiom (A3) to eliminate duplicate summands and Axioms (A1) and (A2) to reorder and regroup summands as necessary. □

5 Prioritized Observational Congruence

The behavioral congruence developed in the previous section is too strong for verifying systems in practice, as it requires that two equivalent terms match each other's transitions exactly, even those labeled by internal actions. In this section we remedy this problem by developing a semantic congruence that abstracts away from internal transitions. Our approach follows the lines of [24,29]. We start off with the definition of a naive prioritized weak bisimulation which is an adaptation of observational equivalence [24].

Definition 16 (Naive Weak Transition Relation)
(i) $\hat{\gamma} =_{df} \epsilon$ *if* $\gamma \in \{\underline{\tau}, \tau\}$ *and* $\hat{\gamma} =_{df} \gamma$, *otherwise.*
(ii) $\stackrel{\epsilon}{\Longrightarrow}_\times =_{df} (\stackrel{\tau}{\longrightarrow} \cup \bigcup\{ \stackrel{m,\tau}{\longrightarrow} \mid m \in \mathcal{L}oc\})^*$
(iii) $\stackrel{m,\gamma}{\Longrightarrow}_\times =_{df} \stackrel{\epsilon}{\Longrightarrow}_\times \circ \stackrel{m,\gamma}{\longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}_\times$

In the following we write $P \stackrel{\gamma}{\Longrightarrow}_\times P'$ for $\exists m \in \mathcal{L}oc.\, P \stackrel{m,\gamma}{\Longrightarrow}_\times P'$.

Definition 17 (Naive Prioritized Weak Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is a* naive prioritized weak bisimulation *if for every* $\langle P, Q\rangle \in \mathcal{R}$, *and* $\gamma \in \mathcal{A}$ *the following condition holds.*

$$P \stackrel{\gamma}{\longrightarrow} P' \text{ implies } \exists Q'.\, Q \stackrel{\hat{\gamma}}{\Longrightarrow}_\times Q' \text{ and } \langle P', Q'\rangle \in \mathcal{R} \ .$$

We write $P \approx_\times Q$ *if there exists a naive prioritized weak bisimulation* $\mathcal{R}$ *such that* $\langle P, Q\rangle \in \mathcal{R}$.

It is fairly easy to see that $\approx_\times$ is not a congruence for $\mathsf{CCS}^{\mathsf{prio}}$. On the other hand, it reflects an intuitive approach to abstracting away from internal computation, and consequently we devote the rest of this section to characterizing the largest congruence contained in this relation. To do so, we first redefine the weak transition relation as follows.

Definition 18 (Prioritized Weak Transition Relation)
For $L, M \subseteq \underline{\mathcal{A}} \setminus \{\underline{\tau}\}$ *we define the following notations.*

(i) $\hat{\underline{\tau}} =_{df} \underline{\epsilon}$, $\hat{\underline{a}} =_{df} \underline{a}$, $\hat{\tau} =_{df} \epsilon$, *and* $\hat{a} =_{df} a$.

(ii) $P \xrightarrow[L]{m,\alpha} P'$ *iff* $P \xrightarrow{m,\alpha} P'$ *and* $\underline{\mathbb{I}}_{[m]}(P) \subseteq L$.

(iii) $\stackrel{\epsilon}{\Longrightarrow} =_{df} (\stackrel{\tau}{\Longrightarrow} \cup \bigcup\{\xrightarrow[\emptyset]{m,\tau} \mid m \in \mathcal{L}oc\})^*$

(iv) $\stackrel{\alpha}{\Longrightarrow} =_{df} \stackrel{\epsilon}{\Longrightarrow} \circ \stackrel{\alpha}{\Longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}$

(v) $\stackrel{\epsilon}{\Longrightarrow}_L =_{df} (\stackrel{\tau}{\Longrightarrow} \cup \bigcup\{\xrightarrow[L]{m,\tau} \mid m \in \mathcal{L}oc\})^*$

(vi) $P \stackrel{m,\alpha}{\Longrightarrow}_{L,M} P'$ *iff* $\exists P'', P'''. P \stackrel{\epsilon}{\Longrightarrow}_L P'' \xrightarrow[L]{m,\alpha} P''' \stackrel{\epsilon}{\Longrightarrow} P'$ *and* $\underline{\mathbb{I}}(P'') \subseteq M$.

Intuitively, these definitions are designed to reflect constraints that a process's environment must satisfy in order for the given transition to be enabled. Thus, $P \xrightarrow[L]{m,\alpha} P'$ means that P can engage in action α at location m *provided that* the environment does not offer a (prioritized) communication involving actions in L . If the environment were to offer such a communication, the result would be a $\underline{\tau}$ at a comparable location to m in P, which would pre-empt the α . In a similar vein, $P \stackrel{\epsilon}{\Longrightarrow} P'$ holds if P can evolve to P' via a nonpre-emptable sequence of internal transitions, regardless of the environment's behavior. These internal transitions should therefore involve either $\underline{\tau}$, which can never be pre-empted, or τ, in which case no prioritized actions should be enabled at the same location. Likewise, $P \stackrel{\epsilon}{\Longrightarrow}_L P'$ means that, so long as the environment does not offer to synchronize with P using the (prioritized) actions in L , the process P may engage in a sequence of internal computation steps and become P' . Finally, the M-parameter in $\stackrel{m,\alpha}{\Longrightarrow}_{L,M}$ provides a measure of the pre-emptive impact that a process can have on its environment. From the definition, $P \stackrel{m,\alpha}{\Longrightarrow}_{L,M} P'$ is true if P can engage in some internal computation followed by α, so long as the environment refrains from synchronizations in L , and then some nonpre-emptable internal computation to arrive at P' . In addition, the state at which α is enabled should only offer prioritized communications in M . To understand the role played by M , consider the processes $P \equiv (a.\mathbf{0} + \underline{b}.\mathbf{0}) \mid \underline{c}.\mathbf{0}$ and $Q \equiv \overline{a}.\mathbf{0} + \overline{\underline{c}}.\mathbf{0}$. If one were to define $\stackrel{m,\alpha}{\Longrightarrow}_L$ in the obvious manner, one would conclude that $P \stackrel{lL,a}{\Longrightarrow}_{\{\underline{b}\}} P' \equiv \mathbf{0} \mid \underline{c}.\mathbf{0}$. Since Q offers a communication $\overline{a}$ and no interaction $\overline{\underline{b}}$, one might then be tempted to infer that $P \mid Q \xrightarrow{\langle lL, lR\rangle, \tau} P' \mid Q' \equiv (\mathbf{0} \mid \underline{c}.\mathbf{0}) \mid \mathbf{0}$. However, the operational semantics disallows this; as P and Q can synchronize on $\underline{c}$, Q's $\overline{a}$-transition becomes pre-empted, even though P's a-transition is not (because its location is incomparable with P's $\underline{c}$-transition). On the other hand, $P \stackrel{lL,a}{\Longrightarrow}_{\{\underline{b}\},\{\underline{c}\}} P'$ alerts us to P's pre-emptive capability on $\underline{c}$.

Note that the definition of $P \stackrel{\epsilon}{\Longrightarrow}_L P'$ corresponds with our intuition that internal actions, and therefore their locations, are unobservable. Moreover, an environment of P is not influenced by internal actions performed by P since priorities arising from different sides of the parallel operator are incomparable. Therefore, the parameter M is unnecessary in the definition of the relation $\stackrel{\epsilon}{\Longrightarrow}_L$. Finally, for notational convenience we interpret $\stackrel{m,\epsilon}{\Longrightarrow}_{L,M}$ as $\stackrel{\epsilon}{\Longrightarrow}_L$.

Definition 19 (Prioritized Weak Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a prioritized weak bisimulation *if for every $\langle P, Q\rangle \in \mathcal{R}$, $\alpha \in A$, $\underline{\alpha} \in \underline{A}$, and $m \in \mathcal{L}oc$ the following conditions hold.*

(i) $\exists Q', Q''.\, Q \stackrel{\epsilon}{\Longrightarrow} Q'' \stackrel{\epsilon}{\Longrightarrow} Q',\ \underline{\mathrm{I\!I}}(Q'') \subseteq \underline{\mathrm{I\!I}}(P),$ *and* $\langle P, Q'\rangle \in \mathcal{R}$.
(ii) $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} Q'$ *and* $\langle P', Q'\rangle \in \mathcal{R}$.
(iii) $P \stackrel{m,\alpha}{\longrightarrow} P'$ *implies* $\exists Q', n.\, Q \stackrel{n,\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} Q'$, $L{=}\underline{\mathrm{I\!I}}_{[m]}(P)$, $M{=}\underline{\mathrm{I\!I}}(P)$, *and* $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \underline{\approx} Q$ if there exists a prioritized weak bisimulation $\mathcal{R}$ such that $\langle P, Q\rangle \in \mathcal{R}$.

From this definition we may directly conclude that $\underline{\approx}$ is the *largest* prioritized weak bisimulation and that $\underline{\approx}$ is an equivalence relation. Cond. (i) of Def. 19 guarantees that prioritized weak bisimulation is compositional with respect to the parallel operator. Its necessity is best illustrated by the following example. The processes $P \stackrel{\text{def}}{=} \underline{\tau}.\underline{a}.\mathbf{0}$ and $Q \stackrel{\text{def}}{=} \underline{a}.\mathbf{0}$ would be considered equivalent if Cond. (i) were absent. However, the context $C[X] \stackrel{\text{def}}{=} X \,|(\overline{\underline{a}}.\mathbf{0} + b.\mathbf{0})$ can distinguish them.

Proposition 20 *The equivalence relation $\underline{\approx}$ is a congruence with respect to prefixing, parallel composition, relabeling, and restriction. Moreover, $\underline{\approx}$ is characterized as the* largest congruence *contained in $\approx_\times$, in the subalgebra of CCS^{prio} induced by these operators and recursion.*

In contrast to [29], the summation fix presented in [24] is not sufficient in order to achieve a congruence relation. To see why, let $C \stackrel{\text{def}}{=} \underline{\tau}.D$ and $D \stackrel{\text{def}}{=} \tau.C$. Now define $P \stackrel{\text{def}}{=} \tau.C$ and $Q \stackrel{\text{def}}{=} \underline{\tau}.D$. By Def. 19 we may observe $P \underline{\approx} Q$, but $P + a.\mathbf{0} \not\underline{\approx} Q + a.\mathbf{0}$ since the former can perform an a-action whereas the latter cannot. It turns out that we have to require that observationally congruent processes must have the same prioritized initial actions. This requirement is stronger than Cond. (i) of Def. 19.

Definition 21 (Prioritized Observational Congruence)
We define $P \underline{\approx}^{\mathrm{I}} Q$ if for all $\alpha \in A$, $\underline{\alpha} \in \underline{A}$, and $m \in \mathcal{L}oc$ the following conditions and their symmetric counterparts hold.

(i) $\underline{\mathrm{I}}(P) \supseteq \underline{\mathrm{I}}(Q)$
(ii) $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\underline{\alpha}}{\Longrightarrow} Q'$ *and* $P' \underline{\approx} Q'$.
(iii) $P \stackrel{m,\alpha}{\longrightarrow} P'$ *implies* $\exists Q', n.\, Q \stackrel{n,\alpha}{\underset{L,M}{\Longrightarrow}} Q'$, $L = \underline{\mathrm{I\!I}}_{[m]}(P)$, $M = \underline{\mathrm{I\!I}}(P)$, *and* $P' \underline{\approx} Q'$.

Theorem 22 *The relation $\underline{\approx}^{\mathrm{I}}$ is the largest congruence contained in $\approx_\times$.*

Whereas the proof of the congruence property can be done using standard techniques [24], the remainder of this section is concerned with the more chal-

lenging proof of the "largest" part of the above theorem. The latter is an instance of the following result from universal algebra.

Theorem 23 *Let X and Y be equivalence relations over an algebra $\Re$ such that $X^+ \subseteq Y \subseteq X$. Then $X^+ = Y^+$ holds.*

Here, we choose $X = \approx_\times$ and $Y = \underline{\approx}$. First, we establish $Y^+ = \underline{\approx}^{\mathrm{l}}$.

Proposition 24 $\underline{\approx}^{\mathrm{l}}$ *is the* largest *congruence contained in* $\underline{\approx}$.

PROOF. Since the relation $\underline{\approx}^{\mathrm{l}}$ is a congruence contained in $\underline{\approx}$, it is sufficient to show that for all $\mathsf{CCS}^{\mathsf{prio}}$-contexts $C[]$ and processes $P, Q \in \mathcal{P}$ satisfying $C[P] \underline{\approx} C[Q]$ the relationship $P \underline{\approx}^{\mathrm{l}} Q$ holds. Moreover, we may restrict ourselves to a subset of $\mathsf{CCS}^{\mathsf{prio}}$-contexts. For this proof, we choose the context $C_{PQ}[X] =_{\mathrm{df}} (\underline{c}.\mathbf{0} + d.\mathbf{0}) + X$ where $\underline{c} \notin \underline{\mathcal{S}(P)} \cup \underline{\mathcal{S}(Q)}$ and $d \notin \mathcal{S}(P) \cup \mathcal{S}(Q)$. Note that such actions $\underline{c}$ and d exist by Lemma 2.

Assume $\underline{\tau} \notin \underline{\mathrm{I}}(P)$ and, therefore, $C_{PQ}[P] \xrightarrow{rl,d} \mathbf{0}$. Since $C_{PQ}[P] \underline{\approx} C_{PQ}[Q]$ and $d \notin \mathcal{S}(Q)$ we necessarily have $C_{PQ}[Q] \stackrel{rl,d}{\underset{L,M}{\Longrightarrow}} \mathbf{0}$ and $\mathbf{0} \underline{\approx} \mathbf{0}$ where $L = M = \underline{\mathbb{I}}(P) \cup \{\underline{c}\}$. This requires $\tau \notin \underline{\mathrm{I}}(Q)$ and $\underline{\mathbb{I}}(Q) \subseteq \underline{\mathbb{I}}(P)$. Hence, $\underline{\mathrm{I}}(Q) \subseteq \underline{\mathrm{I}}(P)$ holds.

Let $P \xrightarrow{m,\alpha} P'$ for some $P' \in \mathcal{P}$ and $m \in \mathcal{L}oc$. Then $C_{PQ}[P] \xrightarrow{mr,\alpha} P'$. Since $C_{PQ}[P] \underline{\approx} C_{PQ}[Q]$ there exist $Q' \in \mathcal{P}$ and $o \in \mathcal{L}oc$ satisfying $C_{PQ}[Q] \stackrel{o,\hat{\alpha}}{\underset{L',M'}{\Longrightarrow}} Q'$, $L' = \underline{\mathbb{I}}_{[m]}(P) \cup \{\underline{c}\}$, $M' = \underline{\mathbb{I}}(P) \cup \{\underline{c}\}$, and $P' \underline{\approx} Q'$. We know that $Q' \not\equiv C_{PQ}[Q]$ because $P' \underline{\approx} Q'$ and P' is *not* capable of performing a weak $\underline{c}$-transition. Therefore, the matching step is necessary, even if $\alpha = \tau$. Thus, $Q \stackrel{n,\alpha}{\underset{L,M}{\Longrightarrow}} Q'$ and $P' \underline{\approx} Q'$ for some $n \in \mathcal{L}oc$ where $n \equiv o$ or $o \equiv nr$, $L = \underline{\mathbb{I}}_{[m]}(P)$, and $M = \underline{\mathbb{I}}(P)$.

Finally, let $P \xrightarrow{\alpha} P'$ for some $P' \in \mathcal{P}$. Then $C_{PQ}[P] \xrightarrow{\alpha} P'$. A simpler argumentation than the one above leads to the existence of some $Q' \in \mathcal{P}$ such that $Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$ and $P' \underline{\approx} Q'$.

Since also the symmetric properties hold, all conditions of Def. 21 are satisfied, and we obtain $P \underline{\approx}^{\mathrm{l}} Q$ as desired. $\square$

Further, we have to show that $X^+ \subseteq Y \subseteq X$ i.e. ${\approx_\times}^+ \subseteq \underline{\approx} \subseteq \approx_\times$. The inclusion $\underline{\approx} \subseteq \approx_\times$ follows immediately from the definition of the naive and the prioritized weak transition relation. In order to apply Theorem 23, we have to establish ${\approx_\times}^+ \subseteq \underline{\approx}$. This inclusion turns out to be difficult to show directly. Therefore, we define $\underline{\approx}_{\mathrm{a}} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \approx_\times C_{PQ}[Q]\}$ as auxiliary equivalence relation which lies in between. Here, using the abbreviations $\underline{S} =_{\mathrm{df}} \underline{\mathcal{S}(P)} \cup$

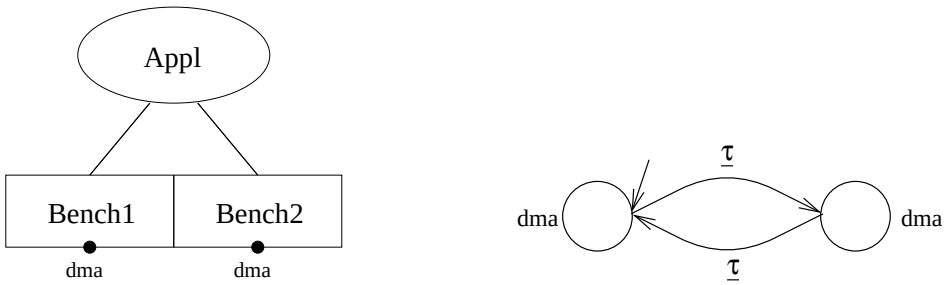

Fig. 3. Example system and its semantics

$\underline{\mathcal{S}(Q)}$ and $S =_{\mathrm{df}} \mathcal{S}(P) \cup \mathcal{S}(Q)$, we define $C_{PQ}[X] =_{\mathrm{df}} X \mid H_{PQ}$ and

$$H_{PQ} \stackrel{\mathrm{def}}{=} \underline{c}.\mathbf{0} + \underline{D_S} + \sum_{\substack{L,\, M \subseteq \overline{S},\\ b \in S}} \underline{\tau}.\Big(\begin{pmatrix} \underline{d}_{L,M,b}.H_{PQ} + \\ \underline{D}_L + e.H_{PQ} + \\ \overline{b}.(\underline{f}.H_{PQ} + \underline{D_S}) \end{pmatrix} \oplus \underline{D}_M \Big).$$

Note that H_{PQ} is well-defined by Lemma 2. Moreover, the processes $\underline{D}_L$ and $\underline{D}_M$ are defined as in the proof of Theorem 7, and the actions $\underline{c}, \underline{d}_{L,M,b}, e, \underline{f}$ are supposed to be 'fresh' actions, i.e. they and their complements are not in the unprioritized or prioritized sort of the processes P and Q under consideration (cf. Lemma 2). By Theorem 4, we may conclude that ${\approx_\times}^+ \subseteq \underline{\approx}_{\mathrm{a}}$. The other necessary inclusion is established by the following proposition. Due to space constraints its proof is omitted here and can be found in [8].

Proposition 25 *The inclusion* $\underline{\approx}_{\mathrm{a}} \subseteq \underline{\approx}$ *holds.*

This proposition completes the establishment of the premises of Theorem 23. Thus, $X^+ = Y^+$, i.e. ${\approx_\times}^+ = \underline{\approx}^+$. Also, we have shown in Prop. 24 that $\underline{\approx}^+ = \underline{\approx}^{\mathrm{l}}$. Hence, ${\approx_\times}^+ = \underline{\approx}^{\mathrm{l}}$, and Theorem 22 is proved.

In App. A it is shown how our prioritized bisimulations can be computed for finite-state processes. In order to apply standard partition refinement algorithms [22,31] the bisimulations are characterized as standard strong bisimulations on enriched transition relations which are defined along the lines of Def. 18 and take local pre-emption potential into account.

6 Example

In this section we demonstrate the utility of $\mathsf{CCS}^{\mathsf{prio}}$ for the verification of distributed systems using an example involving an architecture scheme found in many of today's computers.

Our example system consists of an application that manipulates data from two memory benches (cf. Fig. 3, left-hand side). In order to improve the efficiency in the computer system each bench is connected to a direct-memory-access (DMA) controller. To overcome the low speed of most memory modules, the application Appl works alternately with each memory bench. We model Appl in $\mathsf{CCS}^{\mathsf{prio}}$ by $\mathtt{Appl} \stackrel{\text{def}}{=} \underline{\overline{\mathtt{fetch1}}}.\underline{\overline{\mathtt{fetch2}}}.\mathtt{Appl}$. Each memory bench Bench1 and Bench2 is continuously able to serve the application or to allow the external DMA controller to access the memory via the channel dma. However, if a memory bench has to decide between both activities, then it chooses the former since the progress of the application is considered more important. Consequently, we define $\mathtt{Bench1} \stackrel{\text{def}}{=} \underline{\mathtt{fetch1}}.\mathtt{Bench1} + \mathtt{dma}.\mathtt{Bench1}$ and $\mathtt{Bench2} \stackrel{\text{def}}{=} \underline{\mathtt{fetch2}}.\mathtt{Bench2} + \mathtt{dma}.\mathtt{Bench2}$. The overall system Sys is given by $\mathtt{Sys} \stackrel{\text{def}}{=} (\mathtt{Appl} \mid \mathtt{Bench1} \mid \mathtt{Bench2}) \setminus \{\underline{\mathtt{fetch1}}, \underline{\mathtt{fetch2}}\}$. Since the application uses the memory cells alternately, the DMA is expected to be allowed to access the free memory bench. Therefore, the specification simply is $\mathtt{Spec} \stackrel{\text{def}}{=} \mathtt{dma}.\mathtt{Spec}$. The $\mathsf{CCS}^{\mathsf{prio}}$-semantics of Sys is given in Fig. 3, right hand side, where we abstract away the locations. It is easy to see that the symmetric closure of

$$\{\langle \mathtt{Spec}, \mathtt{Sys}\rangle, \langle \mathtt{Spec}, (\underline{\overline{\mathtt{fetch2}}}.\mathtt{Appl} \mid \mathtt{Bench1} \mid \mathtt{Bench2}) \setminus \{\underline{\mathtt{fetch1}}, \underline{\mathtt{fetch2}}\}\rangle\}$$

is a prioritized weak bisimulation. Note that Cond. (i) of Def. 19 is trivially satisfied since Spec and Sys do not contain any visible prioritized actions. Therefore, we obtain $\mathtt{Spec} \underline{\approx} \mathtt{Sys}$ as expected. However, in the traditional approach to priorities involving global pre-emption [7,29], the dma-loops in the labeled transition system of Sys would be missing, and Sys would not be observationally equivalent to Spec.

7 Extensions of $\mathsf{CCS}^{\mathsf{prio}}$

Up to now we have restricted the number of priority levels in $\mathsf{CCS}^{\mathsf{prio}}$ to two and communication to complementary actions having the same priority. In this section we study the implications for our theory of the removal of these restrictions.

Allowing communication between unprioritized actions and complementary prioritized actions raises the question of whether the resulting internal action should be τ or $\underline{\tau}$. When dealing with local pre-emption this decision has no important consequences for sequential communicating processes, i.e. those in standard concurrent form; however, it is of obvious importance for processes like $(\underline{a}.\mathbf{0} \mid \overline{a}.\mathbf{0}) + b.\mathbf{0}$ in which one has to decide if the b-transition is enabled. One reasonable view is that a communication should be pre-empted whenever one communication partner is pre-empted, i.e. cannot engage in a communication. This implies that the *minimal* priority of the complementary actions ought to

Table 7
Modified operational rules

$$\mathsf{Com1}\ \frac{P \xrightarrow{m,\alpha} P'}{P|Q \xrightarrow{m\cdot L,\alpha} P'|Q}\ \ \underline{\mathbb{I}}_{[m]}(P) \cap (\underline{\overline{\mathbb{I}}}(Q) \cup \overline{\mathbb{I}}(Q)) = \emptyset$$

$$\mathsf{Com3a}\ \frac{P \xrightarrow{m,a} P' \quad Q \xrightarrow{n,\overline{a}} Q'}{P|Q \xrightarrow{\langle m\cdot L, n\cdot R\rangle,\tau} P'|Q'}\ \ \begin{array}{l}\underline{\mathbb{I}}_{[m]}(P) \cap (\underline{\overline{\mathbb{I}}}(Q) \cup \overline{\mathbb{I}}(Q)) = \emptyset \wedge \\ \underline{\mathbb{I}}_{[n]}(Q) \cap (\underline{\overline{\mathbb{I}}}(P) \cup \overline{\mathbb{I}}(P)) = \emptyset\end{array}$$

$$\mathsf{Com3b}\ \frac{P \xrightarrow{m,\underline{a}} P' \quad Q \xrightarrow{n,\overline{a}} Q'}{P|Q \xrightarrow{\langle m\cdot L, n\cdot R\rangle,\tau} P'|Q'}\ \ \underline{\mathbb{I}}_{[n]}(Q) \cap (\underline{\overline{\mathbb{I}}}(P) \cup \overline{\mathbb{I}}(P)) = \emptyset$$

be assigned to the internal action. To reflect this in our operational semantics, we could replace Rules (Com1), (Com2), and (Com3) for parallel composition by the ones presented in Table 7 plus their symmetric versions. The side conditions involve sets $\mathbb{I}(P)$ that include all unprioritized visible initial actions that P can engage in.

It turns out that the largest congruence results concerning our behavioral relations can be carried over to the new calculus; however, the new semantics has algebraic shortcomings, since parallel composition is *not associative*, as illustrated by the following example. Consider the process $(b.\mathbf{0} + \underline{a}.\mathbf{0}) \,|\, (\overline{a}.\mathbf{0} + \underline{c}.\mathbf{0}) \,|\, \underline{\overline{c}}.\mathbf{0}$. When computing the semantics in a left-associative manner, the initial b-transition is pre-empted according to Rule (Com1) since $\underline{a}$ may potentially communicate with $\overline{a}$. However, when first composing the second and third parallel components, the $\overline{a}$-transition is pre-empted, and consequently the b-transition is enabled by Rule (Com1) since $a \notin \overline{\mathbb{I}}((\overline{a}.\mathbf{0} + \underline{c}.\mathbf{0}) \,|\, \underline{\overline{c}})$. The reason for this problem is that we pre-empt transitions because the considered process can *potentially* engage in a higher prioritized communication from a comparable location. However, this potential communication cannot take place if the communication partner is itself pre-empted. The same problem also arises when extending $\mathsf{CCS}^{\mathsf{prio}}$ to multiple priority levels, even if communication is only allowed on complementary actions of the same priority. This can be illustrated using a slight adaptation of the previous example, where priorities are given by natural numbers, with lower numbers denoting higher priority values: $(b{:}2.\mathbf{0} + a{:}1.\mathbf{0}) \,|\, (\overline{a}{:}1.\mathbf{0} + c{:}0.\mathbf{0}) \,|\, \overline{c}{:}0.\mathbf{0}$.

One can imagine two approaches to fixing the problems with the first (and second) alteration to our theory. One is to change the operational semantics; in particular, we could weaken the side conditions so that an unprioritized transition is only pre-empted when a prioritized action from a comparable location can *actually* engage in a communication. To do so we can adopt a conservative view of pre-emption by replacing the complementary initial action sets in the side conditions of parallel composition by ones that only include actions having the highest priority value. Hence, only *actual* communication partners

are considered. One may observe that in this setting additional transitions should be pre-empted whenever actions are restricted. For instance, in the process $P =_{\mathrm{df}} (b.\mathbf{0} + \underline{a}.\mathbf{0}) \,|\, (\overline{a}.\mathbf{0} + \underline{c}.\mathbf{0})$ the b-transition is not pre-empted since $\overline{a}$ does not have the highest priority in the right parallel component. However, when plugging P in the context $[\,] \setminus \{\underline{c}, c\}$, which restricts the $\underline{c}$-transition, the communication on port a may in fact take place and, consequently, the b-transition should be pre-empted. This additional potential of pre-emption needs to be taken care of by the side conditions of the operational rules for restriction. Unfortunately, it turns out that these are hard to formalize in a compact fashion. It is also not clear to us how to establish the statements of our main theorems for the resulting semantic theory.

The second solution follows an approach developed in [6] for a different setting and involves the use of a syntax restriction on processes prohibiting output actions, i.e. actions in $\overline{\Lambda}$, from occurring as initial actions of processes that are in the scope of $+$. Hence, all potential communication partners are also actual ones, and our side conditions for parallel composition are sufficient to encode the desired notion of pre-emption. It is important to mention that the proposed syntax restriction still allows one to specify many practically relevant examples within the calculus. Indeed, a similar restriction may be found in the programming language `occam`. For the new calculus all results presented for $\mathsf{CCS}^{\mathsf{prio}}$ in this paper can be carried over.

8 Discussion and Related Work

Several proposals have been made for extending traditional process algebras with priorities. They differ in the aspects of computation, such as interrupts [2], programming constructs like the `PRIALT` construct of `occam` [6,21] and constructs of `Ada` [14], or real-time [16], that they aim to capture.

An extension of CCS [24] with priorities has been proposed in [7], where priorities are assigned to actions in a *globally dynamic* way, i.e. in one state of a system action α may have priority over action β while the situation may be reversed in another state of the system. For that process algebra a complete semantic theory has been developed in an analogous fashion to [24] which includes congruences based on strong and weak bisimulation and their axiomatic characterizations [29]. Our process algebra $\mathsf{CCS}^{\mathsf{prio}}$ is based on the approach in [7,29], where we adopt all design decisions except the notion of *global* pre-emption. Therefore, $\mathsf{CCS}^{\mathsf{prio}}$ has the following characteristics. Only transitions labeled by complementary actions with the same priority may engage in a synchronization. As in [7], we consider actions with different priorities as *different* channels which is sufficient for most cases occurring in practice [9]. The strong relation of $\mathsf{CCS}^{\mathsf{prio}}$ to the process algebra proposed in [7,28,29] can be

made precise by the following fact. If we leave out the distributed summation operator and globalize pre-emption in our framework by defining $[m] =_{\mathrm{df}} \mathcal{L}oc$ for all $m \in \mathcal{L}oc$, our operational semantics and our behavioral relations reduce to the corresponding notions presented in [7,28,29].

For a comparison with other work one should note that existing approaches that assign priorities to actions are provided with a semantics dealing with *global* pre-emption. In contrast, we consider a notion of *local* pre-emption. This idea is also presented in [17], where a CSP-based calculus is extended with priorities. However, this process algebra suffers from a complicated semantics, especially for the hiding operator, and the authors only conjecture that their strong bisimulation is a congruence. They also do not provide an axiomatization for their equivalence and do not present a theory for observational congruence. Prasad's *Calculus of Broadcasting Systems with Priorities* (PCBS) [33] deals with a distributed notion of priorities. For PCBS a nice semantic theory based on bisimulation has been developed. However, our process algebra $\mathsf{CCS}^{\mathsf{prio}}$ is concerned with a different model for communication.

As shown in the previous section, having a notion of *local* pre-emption enables one to avoid some problems arising in the assignment of priority values to synchronizations involving actions at different priority levels. In traditional approaches, which assign priorities to actions [16,17], several proposals for adjustment functions, e.g. taking the maximum, minimum, or the sum of priority values, have been made. Unfortunately, in settings involving *global* pre-emption each solution is only intuitive for certain (classes of) examples and works only for certain frameworks without violating the congruence property of the considered behavioral relation.

In the remainder of this section we compare our work in detail with [6,21] where also a CCS-based framework is chosen but where priorities are *not* assigned to actions. Instead, there exists a special summation operator $+\rangle$ which favors its left argument over its right argument. Also in that approach a prioritized τ, i.e. a τ-action in which the left argument of $+\rangle$ can initially engage in, has pre-emptive power over unprioritized actions, i.e. actions in which the right argument of $+\rangle$ can initially engage in. Thus, the prioritized summation operator $+\rangle$ of [6] corresponds to the summation operator $+$ in our framework. In [6] the operator $+$ stands for nondeterministic choice where priorities arising from the left and the right argument are incomparable. This operator is matched by the distributed summation operator $\oplus$ in $\mathsf{CCS}^{\mathsf{prio}}$.

As in our framework, priorities arising from different sides of the parallel operator are considered to be incomparable in [6,21]. However, a prioritized parallel operator $\|\rangle$ is introduced in [21] which favors its left argument over its right one. It can be used in descriptions of simple scheduling algorithms. We say "simple" because most scheduling algorithms deal with priority values that may

change as the system evolves. This *dynamic* behavior cannot be described in the framework of [21], which considers *static* priorities only. In contrast, we concentrate on modeling interrupts and prioritized choice and show that this requires the concept of *local pre-emption*. However, scheduling is a global task and, therefore, it is based on a notion of *global* pre-emption. We feel that describing interrupts and scheduling algorithms should be done using two different priority concepts. More precisely, each action should be assigned with two priority values, the first interpreted as global priority value for scheduling purposes and the second interpreted as local priority value for modeling interrupts where the first priority value has more weight than the second one.

After stressing similarities and differences of $\mathsf{CCS}^{\mathsf{prio}}$ to the process algebra presented in [6,21] with respect to design decisions we focus on the algebraic results established in these frameworks. In [6,21] the pre-emption potential is directly encoded in the transition relation. By plugging in this transition relation into the definition of standard strong bisimulation one obtains a congruence relation immediately. In contrast, we start off defining *naive* strong bisimulation using the naive transition relation and consider the pre-emption potential subsequently (by introducing the prioritized initial action set condition). Then we show that the so obtained congruence is the largest congruence in the naive strong bisimulation. Similarly, a naive weak bisimulation is defined in [21] by using the above mentioned transition relation, which already reflects some pre-emption potential. More precisely, this transition relation corresponds to our prioritized weak transition relation where we drop the parameter M . Thus, our naive weak transition relation is more abstract than the one in [21]. In both approaches, the cited and the presented one, the obtained prioritized observational congruence is shown to be the largest congruence in the naive weak bisimulation. However, our result is tighter since our naive weak bisimulation is coarser.

We turn to some remarks about our notion of prioritized strong and weak bisimulation. Since our semantic theory reflects local pre-emption, locations are implicitly occurring in our semantic equivalences. However, in contrast to [5,27] we do not consider locations explicitly. Our objective is not to observe locations but to observe local pre-emption which is necessary for causal reasoning in process algebras with priorities.

Priorities have also been investigated in other concurrent frameworks, most notably in *Petri Nets* [34,37]. In this setting priorities are either expressed explicitly by priority relations over transitions [3] or implicitly via *inhibitor arcs* [20]. In the latter work priorities are modeled via the absence of tokens, i.e. a transition can fire if some predecessor places do not contain any tokens. In contrast to Petri Nets, the focus of process algebras lies in examining properties of behavioral relations with respect to the operators included in the considered algebra. Thus, it is difficult to compare priority approaches of both areas, Petri

Nets and process algebras, from a semantic point of view.

Finally, priorities can implicitly arise when studying causality for *mobile processes* (see e.g. [13]). In these approaches priorities cut off superfluous paths that only present new temporal but not causal dependencies of systems. Hence, this kind of priorities is equipped with a global nature of pre-emption. In contrast, the local view of pre-emption in $\mathsf{CCS}^{\mathsf{prio}}$ is used for restricting the causal, not the temporal, behavior of distributed systems.

9 Conclusions and Future Work

We have presented a process algebra, $\mathsf{CCS}^{\mathsf{prio}}$, that is capable of modeling interrupts and other prioritized behavior in distributed systems. The key idea for $\mathsf{CCS}^{\mathsf{prio}}$ is to take the distribution of the considered system into account in order to define a notion of *local* pre-emption. We have developed a semantic theory for this algebra and have shown its algebraic suitability by an example.

In order to investigate the practical benefits of our approach to distributed priorities, $\mathsf{CCS}^{\mathsf{prio}}$ and its semantic theory need to be implemented in an automated verification tool (e.g. [10]) that allow one to carry out larger case studies. It would also be useful to enhance the practical utility of our calculus by introducing *value-passing*. This can be done in an orthogonal fashion along the lines of [11]. From a theoretical point of view, we intend to axiomatize prioritized observational congruence in order to gain a better understanding of the relationship of our approach to the one presented in [6,21]. Moreover, having studied the semantic concept of distributed priorities within the simple CCS-based framework, it would be interesting to see if our approach can be carried over to more expressive calculi involving *mobile processes* [15,25].

Acknowledgments. We would like to thank the anonymous referees for their detailed comments and suggestions as well as Michael Mendler for carefully proof reading a preliminary version of this paper.

References

[1] J.C.M. Baeten, editor. *Applications of Process Algebra*, volume 17 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, England, 1990.

[2] J.C.M. Baeten, J.A. Bergstra, and J.W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae IX*, pages 127–168, 1986.

[3] E. Best and M. Koutny. Petri net semantics of priority systems. *Theoretical Computer Science*, 96:175–215, 1992.

[4] G. Boudol and I. Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fundamenta Informaticae*, XI(4):433–452, 1988.

[5] G. Boudol, I. Castellani, M. Hennessy, and A. Kiehn. Observing localities. *Theoretical Computer Science*, 114(1):31–61, June 1993.

[6] J. Camilleri and G. Winskel. CCS with priority choice. *Information and Computation*, 116(1):26–37, January 1995.

[7] R. Cleaveland and M.C.B. Hennessy. Priorities in process algebra. *Information and Computation*, 87(1/2):58–77, July/August 1990.

[8] R. Cleaveland, G. Lüttgen, and V. Natarajan. A process algebra with distributed priorities. Technical report, North Carolina State University, Raleigh, NC, USA, 1997. To appear.

[9] R. Cleaveland, V. Natarajan, S. Sims, and G. Lüttgen. Modeling and verifying distributed systems using priorities: A case study. *Software–Concepts and Tools*, 17(2):50–62, 1996.

[10] R. Cleaveland and S. Sims. The NCSU Concurrency Workbench. In R. Alur and T. Henzinger, editors, *Computer Aided Verification (CAV '96)*, Lecture Notes in Computer Science, pages 394–397, New Brunswick, New Jersey, July 1996. Springer-Verlag.

[11] R. Cleaveland and D. Yankelevich. An operational framework for value-passing processes. In *21st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '94)*, pages 326–338, Portland, Oregon, January 1994. IEEE Computer Society Press.

[12] P. Degano, R. De Nicola, and U. Montanari. A partial ordering semantics for CCS. *Theoretical Computer Science*, 75:223–262, 1990.

[13] P. Degano and C. Priami. Causality of mobile processes. In Z. Fülöp and F. Gécseg, editors, *International Conference on Automata, Languages and Programming (ICALP '95)*, volume 944 of *Lecture Notes in Computer Science*, pages 660–671, Szeged, Hungary, July 1995. Springer-Verlag.

[14] C. J. Fidge. A formal definition of priority in CSP. *ACM Transactions on Programming Languages and Systems*, 15(4):681–705, September 1993.

[15] C. Fournet, G. Gonthier, J. Lévy, L. Maranget, and D. Rémy. A calculus of mobile agents. In U. Montanari and V. Sassone, editors, *CONCUR '96 (Concurrency Theory)*, volume 1119 of *Lecture Notes in Computer Science*, pages 406–421, Pisa, Italy, August 1996. Springer-Verlag.

[16] R. Gerber and I. Lee. A resourced-based prioritized bisimulation for real-time systems. *Information and Computation*, 113:102–142, 1994.

[17] H. Hansson and F. Orava. A process calculus with incomparable priorities. In *Proceedings of the North American Process Algebra Workshop*, Workshops in Computing, pages 43–64, Stony Brook, New York, August 1992. Springer-Verlag.

[18] M.C.B. Hennessy. *Algebraic Theory of Processes.* MIT Press, 1988.

[19] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[20] R. Janicki and M. Koutny. Semantics of inhibitor nets. *Information and Computation*, 123(1):1–17, 1995.

[21] C.-T. Jensen. *Prioritized and Independent Actions in Distributed Computer Systems.* PhD thesis, Aarhus University, August 1994.

[22] P. Kanellakis and S.A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, May 1990.

[23] A. Mazurkiewicz. Trace theory. In M.P. Chytil et al., editor, *11th Symposium on Mathematical Foundations of Computer Science*, volume 176 of *Lecture Notes in Computer Science*, pages 115–133. Springer-Verlag, 1984.

[24] R. Milner. *Communication and Concurrency.* Prentice-Hall, London, 1989.

[25] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–77, September 1992.

[26] U. Montanari and D. Yankelevich. Location equivalence in a parametric setting. *Theoretical Computer Science*, 149:299–332, 1995.

[27] M. Mukund and M. Nielsen. CCS, locations and asynchronous transition systems. In *Foundations of Software Technology and Theoretical Computer Science*, volume 652 of *Lecture Notes in Computer Science*, pages 328–341. Springer-Verlag, 1992.

[28] V. Natarajan. *Degrees of Delay: Semantic Theories for Priority, Efficiency, Fairness, and Predictability in Process Algebras.* PhD thesis, North Carolina State University, August 1996.

[29] V. Natarajan, L. Christoff, I. Christoff, and R. Cleaveland. Priorities and abstraction in process algebra. In P.S. Thiagarajan, editor, *Foundations of Software Technology and Theoretical Computer Science*, volume 880 of *Lecture Notes in Computer Science*, pages 217–230, Madras, India, December 1994. Springer-Verlag.

[30] M. Nielsen, G. Plotkin, and G. Winskel. Petri nets, event structures and domains, Part 1. *Theoretical Computer Science*, 13:85–108, 1981.

[31] R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal of Computing*, 16(6):973–989, December 1987.

[32] D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings of 5th G.I. Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[33] K. V. S. Prasad. Broadcasting with priority. In *Proceedings of the 5th European Symposium on Programming*, volume 788 of *Lecture Notes in Computer Science*, pages 469–484, Edinburgh, U.K., April 1994. Springer-Verlag.

[34] W. Reisig. *Petri Nets: An Introduction*. Springer, 1985.

[35] S. A. Smolka and B. Steffen. Priority as extremal probability. *Formal Aspects of Computing*, 8:585–606, 1996.

[36] C. Verhoef. A congruence theorem for structured operational semantics with predicates and negative premises. *Nordic Journal of Computing*, 2:274–302, 1995.

[37] W. Vogler. *Modular Construction and Partial Order Semantics of Petri Nets*, volume 625 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[38] G. Winskel and M. Nielsen. Models for concurrency. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 4, pages 1–148. Oxford Science Publications, 1995.

A Computing the Behavioral Relations

In this section we show how to compute prioritized strong bisimulation $\simeq^{\mathrm{l}}$ and prioritized weak bisimulation $\underline{\approx}^{\mathrm{l}}$. Existing algorithms for computing standard strong bisimulation [24] for finite-state processes are based on the idea of partition refinement [22,31]. One approach to computing the prioritized equivalences would involve characterizing them as "standard" bisimulations over enriched transition systems and then applying these algorithms. In the next two subsections, we provide these characterizations; we then briefly discuss how they may be used as bases for computing the relations.

A.1 Alternative Characterization of Prioritized Strong Bisimulation

The following definition introduces an equivalence $\simeq^*$ which characterizes $\simeq^{\mathrm{l}}$ as standard strong bisimulation. It uses the notation $P \xrightarrow{\alpha} P'$ for some $P, P' \in \mathcal{P}$, $\alpha \in A$, and $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$ whenever $\exists m \in \mathcal{L}oc.\, P \xrightarrow{m,\alpha}{}^{L} P'$ and $\underline{\mathbb{I}}_{[m]}(P) \subseteq L$. Note that these enriched transitions take local pre-emption potential into account, thereby avoiding the explicit annotation of transitions with locations.

Definition 26 (Alternative Prioritized Strong Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is an alternative prioritized strong bisimu-

lation *if for every* $\langle P,Q\rangle \in \mathcal{R}$, $\alpha \in A$, $\underline{\alpha} \in \underline{A}$, *and* $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$ *the following conditions hold.*

(i) $P \stackrel{\underline{\alpha}}{\longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\underline{\alpha}}{\longrightarrow} Q'$ *and* $\langle P',Q'\rangle \in \mathcal{R}$.
(ii) $P \stackrel{\alpha}{\underset{L}{\longrightarrow}} P'$ *implies* $\exists Q'.\, Q \stackrel{\alpha}{\underset{L}{\longrightarrow}} Q'$ *and* $\langle P',Q'\rangle \in \mathcal{R}$.

We write $P \simeq^* Q$ *if there exists an alternative prioritized strong bisimulation* $\mathcal{R}$ *such that* $\langle P,Q\rangle \in \mathcal{R}$.

Proposition 27 (Characterization of $\simeq^{\mathrm{l}}$) *We have* $\simeq^{\mathrm{l}} = \simeq^*$.

The proof of this proposition follows traditional lines and is omitted.

A.2 Alternative Characterization of Prioritized Weak Bisimulation

We now characterize prioritized weak bisimulation as standard bisimulation over an appropriately defined transition system. To begin with, we introduce a family of relations $\stackrel{}{\underset{M}{\Longrightarrow}}$ on processes, where $M \subseteq \underline{A} \setminus \{\underline{\tau}\}$, by defining $P \stackrel{}{\underset{M}{\Longrightarrow}} P'$ if $\exists P''.\, P \stackrel{\epsilon}{\Longrightarrow} P'' \stackrel{\epsilon}{\Longrightarrow} P'$ and $\underline{\mathbb{I}}(P'') \subseteq M$. Moreover, we make the prioritized weak transition relation independent of locations by writing $P \stackrel{\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} P'$ whenever $\exists m \in \mathcal{L}oc.\, P \stackrel{m,\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} P'$.

Definition 28 (Alternative Prioritized Weak Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is an* alternative prioritized weak bisimulation *if for every* $\langle P,Q\rangle \in \mathcal{R}$, $\alpha \in A$, $\underline{\alpha} \in \underline{A}$, *and* $L, M \subseteq \underline{A} \setminus \{\underline{\tau}\}$ *the following conditions hold.*

(i) $P \stackrel{}{\underset{M}{\Longrightarrow}} P'$ *implies* $\exists Q'.\, Q \stackrel{}{\underset{M}{\Longrightarrow}} Q'$ *and* $\langle P',Q'\rangle \in \mathcal{R}$.
(ii) $P \stackrel{\hat{\alpha}}{\Longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\hat{\alpha}}{\Longrightarrow} Q'$ *and* $\langle P',Q'\rangle \in \mathcal{R}$.
(iii) $P \stackrel{\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} P'$ *implies* $\exists Q'.\, Q \stackrel{\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} Q'$ *and* $\langle P',Q'\rangle \in \mathcal{R}$.

We write $P \underline{\approx}^* Q$ *if there exists an alternative prioritized weak bisimulation* $\mathcal{R}$ *such that* $\langle P,Q\rangle \in \mathcal{R}$.

Proposition 29 (Characterization of $\underline{\approx}$) *We have* $\underline{\approx} = \underline{\approx}^*$.

In order to prove the above proposition we need the following two properties of prioritized weak bisimulation.

Lemma 30 *Let* $P, P', Q \in \mathcal{P}$ *such that* $P \underline{\approx} Q$ *and* $P \stackrel{\epsilon}{\Longrightarrow} P'$. *Then there exists some* $Q' \in \mathcal{P}$ *satisfying* $Q \stackrel{\epsilon}{\Longrightarrow} Q'$ *and* $P' \underline{\approx} Q'$.

Lemma 31 *Let $P, P', Q \in \mathcal{P}$ and $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$ such that $P \underline{\approx} Q$ and $P \stackrel{\epsilon}{\underset{L}{\Longrightarrow}} P'$. Then there exists some $Q' \in \mathcal{P}$ satisfying $Q \stackrel{\epsilon}{\underset{L}{\Longrightarrow}} Q'$ and $P' \underline{\approx} Q'$.*

Both lemmata can easily be established by induction on the length of the transition from process P to process P'.

Proof of Prop. 29 We first establish the inclusion $\underline{\approx} \subseteq \underline{\approx}^*$ by showing that $\underline{\approx}$ is an alternative prioritized weak bisimulation. Let $P, Q \in \mathcal{P}$ be such that $P \underline{\approx} Q$.

(i) Let $P \stackrel{}{\underset{M}{\Longrightarrow}} P'$ for some $P' \in \mathcal{P}$ and $M \subseteq \underline{A} \setminus \{\underline{\tau}\}$, i.e. $P \stackrel{\epsilon}{\Longrightarrow} P'' \stackrel{\epsilon}{\Longrightarrow} P'$ and $\underline{\mathbb{I}}(P'') \subseteq M$ for some $P'' \in \mathcal{P}$. Because of $P \underline{\approx} Q$ we know by Lemma 30 of the existence of some $Q'' \in \mathcal{P}$ such that $Q \stackrel{\epsilon}{\Longrightarrow} Q''$ and $P'' \underline{\approx} Q''$. By Cond. (i) of Def. 19 we obtain processes $\overline{Q}', \overline{Q}'' \in \mathcal{P}$ such that $Q'' \stackrel{\epsilon}{\Longrightarrow} \overline{Q}' \stackrel{\epsilon}{\Longrightarrow} \overline{Q}''$, $\underline{\mathbb{I}}(\overline{Q}'') \subseteq \underline{\mathbb{I}}(P'')$, and $P'' \underline{\approx} \overline{Q}''$. Because of the latter, there also exists some $Q' \in \mathcal{P}$ such that $\overline{Q}'' \stackrel{\epsilon}{\Longrightarrow} Q'$ and $P' \underline{\approx} Q'$. Summarizing, we have $Q \stackrel{\epsilon}{\Longrightarrow} Q'' \stackrel{\epsilon}{\Longrightarrow} \overline{Q}' \stackrel{\epsilon}{\Longrightarrow} \overline{Q}'' \stackrel{\epsilon}{\Longrightarrow} Q'$, $\underline{\mathbb{I}}(\overline{Q}'') \subseteq \underline{\mathbb{I}}(P'') \subseteq M$, and $P' \underline{\approx} Q'$. Thus, we have established the existence of some $Q' \in \mathcal{P}$ satisfying $Q \stackrel{}{\underset{M}{\Longrightarrow}} Q'$ and $P' \underline{\approx} Q'$, as desired.

(ii) Let $P \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} P'$ for some $P' \in \mathcal{P}$ and some $\underline{\alpha} \in \underline{A}$, i.e. there exist some $P'', P''' \in \mathcal{P}$ such that $P \stackrel{\epsilon}{\Longrightarrow} P'' \stackrel{\hat{\underline{\alpha}}}{\rightarrow} P''' \stackrel{\epsilon}{\Longrightarrow} P'$ by the definition of the prioritized weak transition relation. In the following, we assume $\alpha \neq \tau$ since the other case follows similar lines but is simpler. Because of $P \underline{\approx} Q$ and Lemma 30 we conclude the existence of some $Q'' \in \mathcal{P}$ satisfying $Q \stackrel{\epsilon}{\Longrightarrow} Q''$ and $P'' \underline{\approx} Q''$. The latter implies $Q'' \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} Q'''$ and $P''' \underline{\approx} Q'''$ for some $Q''' \in \mathcal{P}$. Finally, $P''' \underline{\approx} Q'''$ and Lemma 30 implies the existence of some $Q' \in \mathcal{P}$ such that $Q''' \stackrel{\epsilon}{\Longrightarrow} Q'$ and $P' \underline{\approx} Q'$. Summarizing, we conclude $Q \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} Q'$ and $P' \underline{\approx} Q'$ for some $Q' \in \mathcal{P}$.

(iii) Let $P \stackrel{\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} P'$ for some $P' \in \mathcal{P}$, $\alpha \in A$, and $L, M \subseteq \underline{A} \setminus \{\underline{\tau}\}$. Hence by definition, $P \stackrel{m,\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} P'$ for some $m \in \mathcal{L}oc$, i.e. $P \stackrel{\epsilon}{\underset{L}{\Longrightarrow}} P'' \stackrel{m,\hat{\alpha}}{\Longrightarrow} P''' \stackrel{\epsilon}{\Longrightarrow} P'$, $\underline{\mathbb{I}}_{[m]}(P'') \subseteq L$, and $\underline{\mathbb{I}}(P'') \subseteq M$ for some $P'', P''' \in \mathcal{P}$ according to the definition of the prioritized weak transition relation. A similar reasoning as in the previous case, using Lemma 31 instead of Lemma 30, leads to the existence of some $Q', Q'', Q''' \in \mathcal{P}$ and $n \in \mathcal{L}oc$ such that $Q \stackrel{\epsilon}{\underset{L}{\Longrightarrow}} Q'' \stackrel{n,\hat{\alpha}}{\underset{L',M'}{\Longrightarrow}} Q''' \stackrel{\epsilon}{\Longrightarrow} Q'$, where $L' = \underline{\mathbb{I}}_{[m]}(P'') \subseteq L$, $M' = \underline{\mathbb{I}}(P'') \subseteq M$, $P'' \underline{\approx} Q''$, $P''' \underline{\approx} Q'''$, and $P' \underline{\approx} Q'$. Thus, we obtain $Q \stackrel{n,\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} Q'$, i.e. $Q \stackrel{\hat{\alpha}}{\underset{L,M}{\Longrightarrow}} Q'$, and $P' \underline{\approx} Q'$ by the definition of the prioritized weak transition relation, as desired.

Hence, $\underline{\approx}$ is an alternative prioritized weak bisimulation, i.e. $\underline{\approx} \subseteq \underline{\approx}^*$ by Def. 28. For proving the reverse inclusion $\underline{\approx}^* \subseteq \underline{\approx}$, let $P, Q \in \mathcal{P}$ be such that

$P \underline{\approx}^* Q$. In the following we show that $\underline{\approx}^*$ is a prioritized weak bisimulation.

(i) Because of $P \Longrightarrow_M P$ for $M = \underline{\mathbb{I}}(P)$ and the premise $P \underline{\approx}^* Q$ we may conclude the existence of some $Q' \in \mathcal{P}$ such that $Q \Longrightarrow_M Q'$ and $P \underline{\approx}^* Q'$. Thus, Cond. (i) of Def. 19 holds.

(ii) Let $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ for some $P' \in \mathcal{P}$ and $\underline{\alpha} \in \underline{A}$. By the definition of the prioritized weak transition relation $P \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} P'$ also holds. Now we may conclude the existence of some $Q' \in \mathcal{P}$ satisfying $Q \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} Q'$ and $P' \underline{\approx}^* Q'$ because of the premise $P \underline{\approx}^* Q$. Hence, Cond. (ii) of Def. 19 is established.

(iii) Let $P \stackrel{m,\alpha}{\longrightarrow} P'$ for some $P' \in \mathcal{P}$, $\alpha \in A$, and $m \in \mathcal{L}oc$. This implies $P \stackrel{\hat{\alpha}}{\Longrightarrow}_{L,M} P'$, where $L = \underline{\mathbb{I}}_{[m]}(P)$ and $M = \underline{\mathbb{I}}(P)$, according to the definitions in this section. By the premise $P \underline{\approx}^* Q$ we may conclude the existence of some $Q' \in \mathcal{P}$ such that $Q \stackrel{\hat{\alpha}}{\Longrightarrow}_{L,M} Q'$ and $P' \underline{\approx}^* Q'$. Therefore, $Q \stackrel{n,\hat{\alpha}}{\Longrightarrow}_{L,M} Q'$ for some $n \in \mathcal{L}oc$ and $P' \underline{\approx}^* Q'$, i.e. Cond. (iii) of Def. 19 is satisfied, as desired.

Finally, we conclude by Def. 19 that $\underline{\approx}^*$ is a prioritized weak bisimulation, i.e. $\underline{\approx}^* \subseteq \underline{\approx}$ holds. $\square$

A.3 Algorithms

On the basis of these characterizations we may compute $\simeq^{\mathrm{l}}$ and $\underline{\approx}$ for finite-state processes using the following general technique. First build transition systems for the processes in question that have transitions of the form specified above; then apply a standard bisimulation algorithm. Thus, for example, to determine if $P \simeq^{\mathrm{l}} Q$ we would build transition systems for P and Q having transitions $\stackrel{\underline{\alpha}}{\Longrightarrow}$ and $\stackrel{\alpha}{\underset{L}{\longrightarrow}}$ and then use e.g. the Paige-Tarjan partition-refinement algorithm [31] to see if P and Q wind up in the same equivalence class.

Regarding efficiency, we begin by noting that the time complexity of the most efficient algorithm by Paige and Tarjan [31] is linear in the size of the transition relations of the considered processes and logarithmic in the size of their state spaces. Also, our enriched transition relations for unprioritized actions are parameterized by subsets of the prioritized visible alphabet of interest, i.e. the union of the finite sorts of the considered processes, leading to a potential exponential blow-up in the number of transitions. This is unlikely to be an issue in practice, however, since most actions used in a system definition are internal and only a few of them remain visible for an external observer. It should be noted that the local view of pre-emption does not allow us to eliminate the prioritized action set parameters of our transition relations, as has been done in [28] with respect to a priority framework dealing with global pre-emption.

B Logical Characterizations

In this section we provide a logical characterization of $\simeq^{\mathrm{l}}$ by adapting the well-known Hennessy-Milner Logic [24] to the (strong) enriched transition relation presented in the previous section. The syntax of our logic is defined by the following BNF where $\underline{\alpha} \in \underline{A}$, $\alpha \in A$, and $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$.

$$\Phi \quad ::= \quad tt \quad | \quad \neg\Phi \quad | \quad \Phi \wedge \Phi \quad | \quad \langle\underline{\alpha}\rangle\Phi \quad | \quad \langle\alpha, L\rangle\Phi$$

The set of all formulae is denoted by $\mathcal{F}$ and ranged over by $\Phi, \Psi, \ldots$. We define the satisfaction relation $\models \subseteq \mathcal{P} \times \mathcal{F}$ between processes and formulae inductively on the structure of formulae.

$$\begin{array}{lll} & & P \models tt \\ P \models \neg\Phi & \text{if} & \text{not } P \models \Phi \\ P \models \Phi \wedge \Psi & \text{if} & P \models \Phi \text{ and } P \models \Psi \\ P \models \langle\underline{\alpha}\rangle\Phi & \text{if} & \exists P' \in \mathcal{P}.\ P \stackrel{\underline{\alpha}}{\Longrightarrow} P' \text{ and } P' \models \Phi \\ P \models \langle\alpha, L\rangle\Phi & \text{if} & \exists P' \in \mathcal{P}.\ P \stackrel{\alpha}{\underset{L}{\longrightarrow}} P' \text{ and } P' \models \Phi \end{array}$$

Intuitively, P satisfies $\langle\alpha, L\rangle\Phi$ if P possesses an α-transition with parameter L to a process satisfying Φ.

Theorem 32 (Characterization of $\simeq^{\mathrm{l}}$)
Let $P, Q \in \mathcal{P}$. Then $P \simeq^{\mathrm{l}} Q$ if and only if $\{\Phi \in \mathcal{F} \mid P \models \Phi\} = \{\Phi \in \mathcal{F} \mid Q \models \Phi\}$.

Most proof parts of this theorem are similar to the corresponding ones presented in [24]. First, we define yet another characterization of prioritized strong bisimulation.

Definition 33 *Let $\simeq^{\mathrm{l}}_0 = \mathcal{P} \times \mathcal{P}$ and $P \simeq^{\mathrm{l}}_{i+1} Q$ for some $i \in \mathbb{N}$ if the following properties and their symmetric counterparts hold for all $\underline{\alpha} \in \underline{A}$, $\alpha \in A$, and $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$.*

(i) $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ *implies* $\exists Q'.\ Q \stackrel{\underline{\alpha}}{\Longrightarrow} Q'$ *and* $P' \simeq^{\mathrm{l}}_i Q'$.
(ii) $P \stackrel{\alpha}{\underset{L}{\longrightarrow}} P'$ *implies* $\exists Q'.\ Q \stackrel{\alpha}{\underset{L}{\longrightarrow}} Q'$ *and* $P' \simeq^{\mathrm{l}}_i Q'$.

The proof of the next proposition follows the lines in [24]. Note that for all processes in $\mathcal{P}$ their corresponding transition systems are *finite-branching* [24] because $\mathsf{CCS}^{\mathsf{prio}}$-processes are guarded, and the summation operators are binary.

Proposition 34 *Let $P, Q \in \mathcal{P}$. Then we have $P \simeq^{\mathrm{l}} Q$ if and only if $P \simeq^{\mathrm{l}}{}_i Q$ for all $i \in \mathbb{N}$.*

Now we are able to prove Theorem 32. By Prop. 34 it is sufficient to establish the following two lemmata.

Lemma 35 *Let $P, Q \in \mathcal{P}$, $i \in \mathbb{N}$, and $\Phi \in \mathcal{F}$ such that $P \simeq^{\mathrm{l}}{}_i Q$ and $P \models \Phi$. Then $Q \models \Phi$ holds.*

PROOF. We prove the lemma by induction on i where the induction step is divided into several cases according to the structure of Φ. The only non-standard case is $\Phi \equiv \langle \alpha, L \rangle \Psi$ for $\alpha \in A$ and $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$. By definition of $\models$ we conclude the existence of a process $P' \in \mathcal{P}$ such that $P \xrightarrow[L]{\alpha} P'$ and $P' \models \Psi$. Since $P \simeq^{\mathrm{l}}{}_i Q$ we also know of the existence of some $Q' \in \mathcal{P}$ such that $Q \xrightarrow[L]{\alpha} Q'$, and $P' \simeq^{\mathrm{l}}{}_{i-1} Q'$. By induction hypothesis, $Q' \models \Psi$. Therefore, $Q \models \langle \alpha, L \rangle \Psi$, as desired. $\square$

Lemma 36 *Let $P, Q \in \mathcal{P}$ and $i \in \mathbb{N}$ such that $P \not\simeq^{\mathrm{l}}{}_i Q$ holds. Then there exists a formula $\Phi \in \mathcal{F}$ satisfying $P \models \Phi$ but $Q \not\models \Phi$.*

PROOF. We prove this lemma by induction on i. The induction base is trivial since the premise $P \not\simeq^{\mathrm{l}}{}_0 Q$ does not hold. Now, let $i > 0$ and $P \not\simeq^{\mathrm{l}}{}_i Q$. We have to find a formula $\Phi \in \mathcal{P}$ such that $P \models \Phi$ and $Q \not\models \Phi$. Since $P \not\simeq^{\mathrm{l}}{}_i Q$ we either have $P \stackrel{\underline{\alpha}}{\Longrightarrow} P'$ for some $\underline{\alpha} \in \underline{A}$ and $P' \in \mathcal{P}$, or $P \xrightarrow[L]{\alpha} P'$ for some $\alpha \in A$, $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$, and $P' \in \mathcal{P}$. The first case follows the standard lines. In the second case we know that whenever $Q \xrightarrow[L]{\alpha} Q'$ then $P' \not\simeq^{\mathrm{l}}{}_{i-1} Q'$. Let $\{Q' \mid Q \xrightarrow[L]{\alpha} Q'\} = \{Q_j \mid j \in J\}$ for some index set J. By induction hypothesis we conclude the existence of formulae Ψ_j, for $j \in J$, such that $P' \models \Psi_j$ and $Q_j \not\models \Psi_j$. Now, define $\Phi =_{\mathrm{df}} \langle \alpha, L \rangle \bigwedge_{j \in J} \Psi_j$. It is easy to see that $P \models \Phi$. Since no α-derivative of Q parameterized by L satisfies $\bigwedge_{j \in I} \Psi_j$, we also have $Q \not\models \Phi$, as desired. $\square$

We conclude this section with a remark on the logical characterization of $\underline{\approx}$. Defining a suitable logic can be done by replacing the $\langle \alpha, L \rangle$ operators of the logic presented above by new operators $\langle\!\langle \alpha, L, M \rangle\!\rangle$ for $M \subseteq \underline{A} \setminus \{\underline{\tau}\}$ where a process $P \in \mathcal{P}$ satisfies the formula $\langle\!\langle \alpha, L, M \rangle\!\rangle \Phi$ if there exists a process $P' \in \mathcal{P}$ such that $P \stackrel{\hat{\alpha}}{\Longrightarrow}_{L,M} P'$ and $P' \models \Phi$. The operators $\langle \underline{\alpha} \rangle$ have also to be replaced by operators $\langle\!\langle \underline{\alpha} \rangle\!\rangle$ where $P \models \langle\!\langle \underline{\alpha} \rangle\!\rangle \Phi$ if there exists a process $P' \in \mathcal{P}$ such that $P \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} P'$ and $P' \models \Phi$. Finally, the logic has to be extended by new unary operators $\Uparrow_M$ for $M \subseteq \underline{A} \setminus \{\underline{\tau}\}$ in order to match the first requirement of Def. 19 where $P \models \Uparrow_M \Phi$ if $\exists P'. P \Longrightarrow_M P'$ and $P' \models \Phi$. Using these definitions a characterization of $\underline{\approx}$ can be done along the lines of [24].