Priority in Process Algebra

Rance Cleaveland

Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794, USA, rance@cs.sunysb.edu

Gerald Lüttgen

Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681, USA, luettgen@icase.edu

V. Natarajan

Networking Hardware Division, IBM Corporation, Research Triangle Park, NC 27709, USA, nataraj@raleigh.ibm.com

Abstract

This chapter surveys the semantic ramifications of extending traditional process algebras with notions of *priority* that allow some transitions to be given precedence over others. The need for these enriched formalisms arises when one wishes to model system features such as *interrupts*, *prioritized choice*, or *real-time behavior*.

Approaches to priority in process algebras can be classified according to whether the induced notion of *pre-emption* on transitions is *global* or *local* and whether priorities are *static* or *dynamic*. Early work in the area concentrated on global pre-emption and static priorities and led to formalisms for modeling interrupts and aspects of real-time, such as *maximal progress*, in *centralized* computing environments. More recent research has investigated localized notions of pre-emption in which the *distribution* of systems is taken into account, as well as dynamic priority approaches. The latter allows priority values to change as systems evolve and enables an efficient encoding of real-time semantics.

Technically, this chapter studies the different models of priorities by presenting extensions of Milner's *Calculus of Communicating Systems* (CCS) with static and dynamic priority as well as with notions of global and local pre-emption. For each extension, the operational semantics of CCS is modified appropriately, behavioral theories based on *strong* and *weak bisimulation* are given, and related approaches for different process-algebraic settings are discussed.

Key words: priority, pre-emption, bisimulation.

Contents

1 Introduction

Traditional *process algebras* [?,?,?,?] provide a framework for reasoning about the *communication potential* of *concurrent* and *distributed systems*. Such theories typically consist of a simple *calculus* with a well-defined *operational semantics* [?,?] given as *labeled transition systems*; a *behavioral equivalence* is then used to relate implementations and specifications, which are both formalized as terms in the calculus. In order to facilitate *compositional reasoning*, in which systems are verified component by component, researchers have concentrated on developing *behavioral congruences*, which allow the substitution of "equals for equals" inside larger systems.

Over the past decade, process-algebraic approaches to system modeling and verification have been applied to a number of case studies (see, e.g., [?]). Nevertheless, many systems in practice cannot be modeled accurately within this framework. One reason is that traditional process algebras focus exclusively on expressing the potential *nondeterminism* that the interplay of concurrent processes may exhibit; they do not provide any means for encoding differing levels of *urgency* among the transitions that might be enabled from a given system state. In practice, however, some system transitions are intended to take precedence of others; examples include:

- **interrupts**, where non-urgent transitions at a state are *pre-empted* whenever an interrupt is raised;
- **programming language constructs**, such as the `PRIALT` construct in `occam` [?], that impose a priority order on transition choices;
- **real-time behavior** that is founded on the well-known *synchrony hypothesis* [?] or *maximal progress assumption* [?]; and
- **scheduling algorithms** which also rely on the concept of pre-emption.

In each of these cases urgency provides a means for *restricting nondeterminism*. This mechanism is simply ignored in traditional process algebras. Hence, the resulting system models are often not faithful since they contain spurious paths that cannot be traversed by the real-world systems themselves [?,?].

As an illustration of the need for integrating notions of urgency into process algebra consider the interrupt-based system depicted in Figure **??**. The system consists of two processes: A, which flips back and forth between two states, and B, which prompts A to report its status. Whenever B receives a check message from the environment it instructs A of this via *interrupt port* $\underline{i}$; A in turn issues its response on its ok channel. In the absence of an indication that a communication on $\underline{i}$ is more urgent than one on back and forth, process A is not required to respond immediately to a request from B. Indeed, the system's behavior is not even *fair* [?,?] since A may ignore such requests indefinitely.

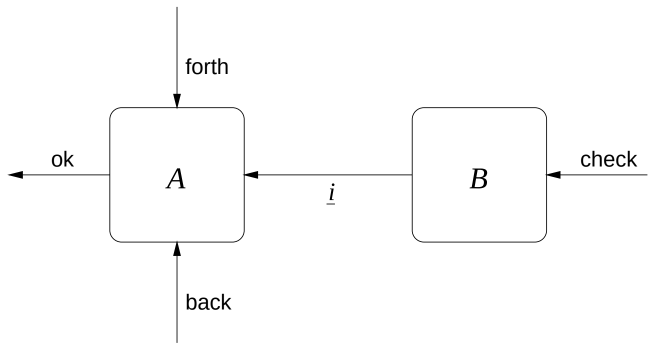

Fig. 1. Back-and-forth example

It should be noted that fairness is a weaker notion than priority [?], as it only requires *eventual* rather than *immediate* responses.

1.1 Classification of Approaches to Priority

A number of approaches have been proposed for introducing priority to process algebras [?,?]. One may classify these approaches according to the following two criteria.

Static priority vs. dynamic priority:

In *static* approaches transition priorities may be inferred from syntax of the system description before it commences execution; they do not change during computation. These schemes find application in the modeling of interrupts or prioritized choice constructs. In the former case, interrupts have a fixed urgency level associated with them; in the latter priorities of transitions are fixed by the static program syntax. Almost all priority approaches to process algebra published so far deal with static priorities. The exceptions are [?,?], which present models that allow priority values of transitions to change as systems evolve. Such *dynamic* behavior is useful in modeling scheduling algorithms and real-time semantics.

Global pre-emption vs. local pre-emption:

This criterion refers to the *scope* of priority values. In the case of centralized systems, priorities generally have a *global* scope in the sense that transitions in one process may pre-empt transitions in another. We refer to this kind of pre-emption, which has been advocated by Baeten, Bergstra, and Klop [?] and by Cleaveland and Hennessy [?] in the late eighties, as *global pre-emption*. In contrast, in a distributed system containing multiple sites, transitions should only be allowed to pre-empt those at the same site or *location*. This kind of pre-emption, which was first studied by Camilleri and Winskel [?] in the early nineties, is called *local pre-emption*.

Based on the above classification scheme, the body of this chapter investigates the following different semantics for a prototypical process-algebraic language: static/global, static/local, and dynamic/global. Only the combination of dynamic priority and local pre-emption is omitted, since research has not yet been carried out on this topic.

Some caveats about terminology are in order here. Other process algebra researchers have used the term "pre-emption" in a setting without priorities [?]; in their usage pre-emption occurs when the *execution* of one transition removes the possibility of another. In our priority-oriented framework, we say that pre-emption occurs when the *availability* of one transition disables others. Berry [?] refers to this latter notion as *must pre-emption* and to the former as *may pre-emption*. In this chapter, whenever we speak of "pre-emption," we mean "must pre-emption." It should also be noted that our concept of global pre-emption and dynamic priority differs from the notion of *globally dynamic* priority found in [?]; as the distinction is somewhat technical we defer further discussion on this point to later in the chapter.

1.2 Summary of this Chapter

This chapter surveys existing work on priority in process algebras. To illustrate the semantic issues involved, we introduce a single process calculus that we then equip with different notions of pre-emption. This calculus extends Milner's *Calculus of Communicating Systems* (CCS) [?] by attaching priority values to actions. For this language three different semantics are given: one reflecting static priorities and global pre-emption, one for static priorities and local pre-emption, and one capturing dynamic priorities. The common language allows for a detailed comparison of the semantic concepts; in addition, the classification scheme presented above helps categorize most published approaches to priority. These have been proposed for a variety of well-known process algebras, such as the already mentioned CCS, the *Algebra of Communicating Processes* (ACP) [?], *Communicating Sequential Processes* (CSP) [?], the *Calculus of Broadcasting Systems* (CBS) [?], *Synchronous CCS* (SCCS) [?], and *Asynchronous Communicating Shared Resources* (ACSR) [?].

For the static-priority settings in this chapter we develop semantic theories based on the notion of bisimulation [?,?]. Our aim is to carry over the standard algebraic results from CCS [?], including full-abstractness theorems as well as axiomatic, logical, and algorithmic characterizations. In particular, we investigate both *strong* and *weak bisimulations* that are based on naive adaptations of the standard definitions as given by Milner; we then focus on characterizing the *largest congruences* contained in these relations. Such full-abstractness results indicate that the behavioral relations are compositional and thus useful for formally reasoning about concurrent and distributed systems. Then we present sound and complete *axiomatizations* for the obtained strong bisimulations with respect to finite processes, i.e., those which do not contain recursion. These axiomatizations testify to the mathematical tractability of the semantic theories presented here. We also characterize the attendant notions of prioritized strong and weak bisimulations as standard bisimulations via enriched transition labels, so that well-known *partition-refinement algorithms* [?,?,?] for their computation become applicable. Such accounts also provide support for logical characterizations of the behavioral relations, as they give direct insights into how the *Hennessy–Milner logic* [?,?] should be adapted. In case of the dynamic priority semantics, we prove a one-to-one correspondence with traditional *real-time semantics* [?,?,?] in terms of strong bisimulation. Because of this close relationship, semantic theories developed for real-time process algebras can be carried over to the dynamic priority setting.

1.3 Organization of this Chapter

The remainder of this chapter is organized as follows. The next section introduces our extension of CCS, defines some formal notation used throughout the chapter, and discusses some basic design decisions we have taken. Section **??** presents a semantics of the language based on static priority and global pre-emption; Section **??** then develops a semantics based on static priority and local pre-emption. A dynamic priority approach is illustrated in Section **??**. Related work is referred to in each of the last three sections, while Section **??** surveys several priority approaches adopted for process-algebraic frameworks other than CCS. Section **??** contains our conclusions and suggestions for future work. The final two sections point to the sources of the research compiled in this chapter and summarize our notation, respectively. A summary of notation and symbols used may be found at the end.

2 Basic Language and Notation

As mentioned above, the language considered here is an extension of Milner's CCS [?], a process algebra characterized by *handshake communication* and *interleaving semantics* for parallel composition. Syntactically, CCS includes notations for *visible actions*, which are either sends or receives on ports, and a distinguished *invisible*, or *internal* action. The semantics of CCS is then given via a transition relation that labels execution steps with actions. When a sender and receiver synchronize, the resulting action is internal. Consequently, transitions labeled by visible actions can be seen as representing only "potential" computation steps, since in order for them to occur they require a contribution from the environment. Transitions labeled by internal actions describe complete synchronizations and, therefore, should be viewed as "actual" computation steps.

In order to capture priorities, the syntax of our language differs from CCS in that the *port* set exhibits a priority scheme, i.e., priorities are attached to ports. Our notion of pre-emption then stipulates that a system cannot engage in transitions labeled by actions with a given priority if it is able to perform a transition labeled by an *internal* action of higher priority. In this case we say that the lower-priority transition is pre-empted by the higher-priority internal transition. In accordance with the above discussion visible actions *never* have pre-emptive power over actions of lower priority, because visible actions only indicate the potential for execution. An algebraic justification of this design decision can be found in Section **??**.

Priority values are taken from some finite domain equipped with a total order. For the sake of simplicity we use finite initial intervals $\mathcal{N}$ of the natural numbers $\mathbb{N}$ in what follows. We adopt the convention that smaller numbers mean higher priorities; so 0 is the highest priority.

Let $\{\Lambda_k \,|\, k \in \mathcal{N}\}$ denote an $\mathcal{N}$-indexed family of countably infinite, pairwise disjoint sets of ports. Intuitively, Λ_k contains the ports with priority k over which processes may synchronize. Then the set of *actions* $\mathcal{A}_k$ with priority k may be defined by $\mathcal{A}_k =_{\text{df}} \Lambda_k \cup \overline{\Lambda}_k \cup \{\tau_k\}$, where $\overline{\Lambda}_k =_{\text{df}} \{\overline{\lambda}_k \,|\, \lambda_k \in \Lambda_k\}$ and $\tau_k \notin \Lambda_k$. An action $\lambda_k \in \Lambda_k$ may be thought of as representing the receipt of an input on port λ that has priority k, while $\overline{\lambda}_k \in \overline{\Lambda}_k$ constitutes the deposit of an output on λ. The invisible action τ_k represent internal computation steps with priority k. For better readability we write $\lambda\!:\!k$, if $\lambda_k \in \Lambda_k$, $\overline{\lambda}\!:\!k$, if $\overline{\lambda}_k \in \Lambda_k$, and $\tau\!:\!k$ for τ_k. The sets of all ports Λ, all complementary ports $\overline{\Lambda}$, and all actions $\mathcal{A}$ are defined by $\bigcup\{\Lambda_k \,|\, k \in \mathcal{N}\}$, $\bigcup\{\overline{\Lambda}_k \,|\, k \in \mathcal{N}\}$, and $\bigcup\{\mathcal{A}_k \,|\, k \in \mathcal{N}\}$, respectively. In what follows, we use $\alpha\!:\!k, \beta\!:\!k, \ldots$ to range over $\mathcal{A}$ and $a\!:\!k, b\!:\!k, \ldots$ to range over $\Lambda \cup \overline{\Lambda}$. We also extend complementation to all visible actions $a\!:\!k$ by $\overline{\overline{a}}\!:\!k =_{\text{df}} a\!:\!k$. Finally, if $L \subseteq \Lambda \cup \overline{\Lambda}$ then $\overline{L} = \{\overline{a}\!:\!k \,|\, a\!:\!k \in L\}$. The syntax of our language is defined by the following BNF.

$$
\begin{array}{lllllllllllll}
P & ::= & \mathbf{0} & | & x & | & \alpha\!:\!k.P & | & P + P & | & P \,|\, P & | \\
 & & & & P[f] & | & P \setminus L & | & \mu x.P & & &
\end{array}
$$

Here, f is a *finite relabeling*, i.e., a mapping on $\mathcal{A}$ with $f(\tau\!:\!k) = \tau\!:\!k$, for all $k \in \mathcal{N}$, $\overline{f(\overline{a}\!:\!k)} = f(a\!:\!k)$, for all $a\!:\!k \in \Lambda \cup \overline{\Lambda}$, and $|\{\alpha\!:\!k \,|\, f(\alpha\!:\!k) \neq \alpha\!:\!k\}| < \infty$. A relabeling is also required to preserve priority values, i.e., for all $a\!:\!k \in \Lambda \cup \overline{\Lambda}$, we have $f(a\!:\!k) = b\!:\!k$, for some $b\!:\!k \in \Lambda_k \cup \overline{\Lambda}_k$. The *restriction set* L is a subset of $\Lambda \cup \overline{\Lambda}$, and x is a *variable* taken from a set $\mathcal{V}$. Sometimes it is convenient to write $C \stackrel{\text{def}}{=} P$ for $\mu C.P$, where identifier C is interpreted as a variable. We adopt 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 use $P, Q, R, \ldots$ to range over the set $\mathcal{P}$ of processes. Finally, we denote syntactic equality by $\equiv$.

Although our framework allows for multi-level *priority schemes* we generally restrict ourselves to a two-level priority framework, i.e. we choose $\mathcal{N} = \{0, 1\}$. The reason is that even in this simple setting most semantic and technical issues regarding the introduction of priority to process algebra can be illustrated. However, we also discuss how the obtained results can be carried over to multi-level priority schemes. In order to improve readability within the two-level priority scheme we often write $\underline{\alpha}$ for the prioritized action $\alpha\!:\!0$, α for the unprioritized action $\alpha\!:\!1$, $\underline{A}$ for $\mathcal{A}_0$, and A for $\mathcal{A}_1$. We let δ and γ represent elements taken from $\underline{A} \cup A$. We emphasize that α and $\underline{\alpha}$ are considered to be different ports; i.e., the priority value is part of a port name. Thus, in a CCS-based framework only complementary actions having the *same* priority value can engage in a communication. We discuss the consequences of lifting this restriction in Section **??** for frameworks involving global pre-emption and in Section **??** for those involving local pre-emption. It should be noted that the dynamic priority approach presented in Section **??** also differs in its interpretation of ports, actions, and priority values. Finally, our language does not provide any operator for changing priority values of actions. However, we will discuss in Section **??** the effect of introducing such operators, called *prioritization* and *deprioritization*, which increase and decrease the priority values of actions, respectively.

3 Static Priority and Global Pre-emption

In this section we introduce a semantics of our language that features static priorities, global pre-emption and a two-level priority scheme on actions. We refer to our language with this semantics as $\mathsf{CCS}^{\mathsf{sg}}$ (CCS with $\mathsf{s}$tatic priority and $\mathsf{g}$lobal pre-emption) and develop its semantic theory along the lines mentioned in Section **??**. The organization of this section is as follows. Section **??** formally introduces the operational semantics for $\mathsf{CCS}^{\mathsf{sg}}$. Sections **??** and **??** show how to adapt the notions of strong bisimulation and observational congruence to $\mathsf{CCS}^{\mathsf{sg}}$, respectively. Section **??** applies the semantic theory to the introductory back-and-forth example illustrated in Figure **??**. The consequences of adding prioritization and deprioritization operators to $\mathsf{CCS}^{\mathsf{sg}}$ are discussed in Section **??**. Finally, Section **??** comments on the extension of $\mathsf{CCS}^{\mathsf{sg}}$ to multi-level priority schemes, and Section **??** discusses the design decisions in our theory and presents related work.

3.1 Operational Semantics

The *semantics* of a process $P \in \mathcal{P}$ is given by a labeled transition system $\langle \mathcal{P}, \mathcal{A}, \longrightarrow, P \rangle$, where $\mathcal{P}$ is the set of *states*, $\mathcal{A}$ is the *alphabet*, $\longrightarrow \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{P}$ is the *transition relation*, and P is the *start state*. We write $P \stackrel{\gamma}{\longrightarrow} P'$ instead of $\langle P, \gamma, P' \rangle \in \longrightarrow$ and say that P *may engage in action* γ *and thereafter behave like process* P'. Moreover, we let $P \stackrel{\gamma}{\longrightarrow}$ stand for $\exists P' \in \mathcal{P}.\, P \stackrel{\gamma}{\longrightarrow} P'$. The presentation of the operational rules requires *prioritized initial action sets* $\underline{\mathcal{I}}(P)$ which are defined as the least sets satisfying the equations in Table **??**. Intuitively, $\underline{\mathcal{I}}(P)$ denotes the set of all prioritized actions in which P can initially engage. For convenience, we also write $\underline{\mathcal{I\!I}}(P)$ for the set $\underline{\mathcal{I}}(P) \setminus \{\underline{\tau}\}$ of prioritized external actions.

Table 1
Prioritized initial action sets for $\mathsf{CCS}^{\mathsf{sg}}$

$\underline{\mathcal{I}}(\underline{\alpha}.P)$	$= \{\underline{\alpha}\}$	$\underline{\mathcal{I}}(\mu x.P)$	$= \underline{\mathcal{I}}(P[\mu x.P/x])$
$\underline{\mathcal{I}}(P+Q)$	$= \underline{\mathcal{I}}(P) \cup \underline{\mathcal{I}}(Q)$	$\underline{\mathcal{I}}(P \,\vert\, Q)$	$= \underline{\mathcal{I}}(P) \cup \underline{\mathcal{I}}(Q) \cup \{\underline{\tau} \,\vert\, \underline{\mathcal{I}}(P) \cap \overline{\underline{\mathcal{I}}(Q)} \neq \emptyset\}$
$\underline{\mathcal{I}}(P[f])$	$= \{f(\underline{\alpha}) \,\vert\, \underline{\alpha} \in \underline{\mathcal{I}}(P)\}$	$\underline{\mathcal{I}}(P \setminus L)$	$= \underline{\mathcal{I}}(P) \setminus (L \cup \overline{L})$

The rules in Plotkin-style notation [?] in Table **??** formally define the transition relation and capture the following operational intuitions. Process $\gamma.P$ may engage in action γ and then behave like P. The *summation operator* $+$ denotes *nondeterministic choice*. Process $P+Q$ may behave like process P (Q) if Q (P) does not pre-empt an unprioritized transition by performing a prioritized internal transition. $P[f]$ behaves exactly as process P with the actions renamed with respect to *relabeling* f. Process $P \,|\, Q$ represents the *parallel composition* of P and Q, which are executed according to an *interleaving semantics* with *synchronized communication* on complementary actions on the same priority level resulting in the internal action $\underline{\tau}$ or τ. However, if Q (P) is capable of engaging in a prioritized internal transition, then unprioritized transitions of P (Q) are pre-empted. The *restriction operator* $\setminus L$ prohibits in $P \setminus L$ the execution of transitions labeled by actions in $L \cup \overline{L}$ and, thus, permits the *scoping* of actions. Finally, $\mu x.P$ denotes a *recursively defined* process that is a distinguished solution to the equation $x = P$.

3.2 Semantic Theory Based on Strong Bisimulation

The semantic theory for $\mathsf{CCS}^{\mathsf{sg}}$ is based on the notion of *bisimulation* [?,?]. First, *strong bisimulation* [?] is adapted from CCS to our setting as follows; we refer to this relation as *prioritized strong bisimulation*.

Table 2
Operational semantics for $\mathsf{CCS}^{\mathsf{sg}}$

$\underline{\mathsf{Act}}\quad \dfrac{-}{\underline{\alpha}.P \xrightarrow{\alpha} P}$	$\mathsf{Act}\quad \dfrac{-}{\alpha.P \xrightarrow{\alpha} P}$
$\underline{\mathsf{Sum1}}\quad \dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$	$\mathsf{Sum1}\quad \dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}\ \underline{\tau} \notin \underline{\mathcal{I}}(Q)$
$\underline{\mathsf{Sum2}}\quad \dfrac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}$	$\mathsf{Sum2}\quad \dfrac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} Q'}\ \underline{\tau} \notin \underline{\mathcal{I}}(P)$
$\underline{\mathsf{Com1}}\quad \dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$	$\mathsf{Com1}\quad \dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ \underline{\tau} \notin \underline{\mathcal{I}}(P \mid Q)$
$\underline{\mathsf{Com2}}\quad \dfrac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}$	$\mathsf{Com2}\quad \dfrac{Q \xrightarrow{\alpha} Q'}{P \mid Q \xrightarrow{\alpha} P \mid Q'}\ \underline{\tau} \notin \underline{\mathcal{I}}(P \mid Q)$
$\underline{\mathsf{Com3}}\quad \dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$	$\mathsf{Com3}\quad \dfrac{P \xrightarrow{a} P' \quad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\ \underline{\tau} \notin \underline{\mathcal{I}}(P \mid Q)$
$\underline{\mathsf{Rel}}\quad \dfrac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$	$\mathsf{Rel}\quad \dfrac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$
$\underline{\mathsf{Res}}\quad \dfrac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L}\ \underline{\alpha} \notin L \cup \overline{L}$	$\mathsf{Res}\quad \dfrac{P \xrightarrow{\alpha} P'}{P \setminus L \xrightarrow{\alpha} P' \setminus L}\ \alpha \notin L \cup \overline{L}$
$\underline{\mathsf{Rec}}\quad \dfrac{P[\mu x.P/x] \xrightarrow{\alpha} P'}{\mu x.P \xrightarrow{\alpha} P'}$	$\mathsf{Rec}\quad \dfrac{P[\mu x.P/x] \xrightarrow{\alpha} P'}{\mu x.P \xrightarrow{\alpha} P'}$

Definition 3.1 (Prioritized Strong Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is called a* prioritized strong bisimulation *if for every* $\langle P, Q\rangle \in \mathcal{R}$ *and* $\gamma \in \mathcal{A}$ *the following condition holds.*

$$P \xrightarrow{\gamma} P' \textit{ implies } \exists Q'.\, Q \xrightarrow{\gamma} Q' \textit{ and } \langle P', Q'\rangle \in \mathcal{R}\ .$$

We write $P \simeq Q$ *if* $\langle P, Q\rangle \in \mathcal{R}$ *for some prioritized strong bisimulation* $\mathcal{R}$*.*

Table 3
Axiomatization of $\simeq$

(A1)	$t + u = u + t$	(A2)	$t + (u + v) = (t + u) + v$
(A3)	$t + t = t$	(A4)	$t + \mathbf{0} = t$

(E) Let $t = \sum_i \gamma_i.t_i$ and $u = \sum_j \delta_j.u_j$. Then

$$t \,|\, u = \sum_i \gamma_i.(t_i \,|\, u) + \sum_j \delta_j.(t \,|\, u_j) + \sum_{\gamma_i \equiv \overline{\delta_j}} \{\underline{\tau}.(t_i \,|\, u_j) \,|\, \gamma_i \in \underline{A}\}) + \sum_{\gamma_i \equiv \overline{\delta_j}} \{\tau.(t_i \,|\, u_j) \,|\, \gamma_i \in A\})$$

(Res1)	$\mathbf{0} \setminus L = \mathbf{0}$		(Rel1)	$\mathbf{0}[f] = \mathbf{0}$
(Res2)	$(\gamma.t) \setminus L = \mathbf{0}$	$(\gamma \in L \cup \overline{L})$	(Rel2)	$(\gamma.t)[f] = f(\gamma).(t[f])$
(Res3)	$(\gamma.t) \setminus L = \gamma.(t \setminus L)$	$(\gamma \notin L \cup \overline{L})$	(Rel3)	$(t + u)[f] = t[f] + u[f]$
(Res4)	$(t + u) \setminus L = (t \setminus L) + (u \setminus L)$		(P)	$\underline{\tau}.t + \alpha.u = \underline{\tau}.t$

It is easy to see that $\simeq$ is an equivalence and that it is the *largest* prioritized strong bisimulation. The following result, which enables compositional reasoning, can be proved using standard techniques [**?**,**?**,**?**].

Theorem 3.2 *$\simeq$ is a congruence.*

An *axiomatization* of $\simeq$ for *finite* processes, i.e., guarded and closed $\mathsf{CCS}^{\mathsf{sg}}$ terms not containing recursion, can be developed closely along the lines of [**?**]. We write $\vdash t = u$ if process term t can be rewritten to u using the axioms in Table **??**. These axioms correspond to the ones presented in [**?**], except that Axiom (P) dealing with global pre-emption is added. In the *Expansion Axiom* (E) the symbol $\sum$ stands for the indexed version of $+$, where the empty sum denotes the inaction process $\mathbf{0}$. The next theorem states that our equations characterize prioritized strong bisimulation for finite $\mathsf{CCS}^{\mathsf{sg}}$ processes. Its proof can be found in [**?**]; it uses techniques described in [**?**].

Theorem 3.3 *Let t, u be finite processes. Then $t \simeq u$ if and only if $\vdash t = u$.*

3.3 Semantic Theory Based on Weak Bisimulation

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. The process-algebraic remedy for this problem involves the development of a semantic congruence that abstracts from internal transitions. In pursuit of such a relation we start off with the definition of a *naive prioritized weak bisimulation*, which is an adaptation of Milner's *observational equivalence* [?].

Definition 3.4 (Naive Prioritized Weak Transitions)

(1) $\hat{\underline{\tau}} =_{df} \hat{\tau} =_{df} \epsilon$, $\hat{\underline{a}} =_{df} \underline{a}$, *and* $\hat{a} =_{df} a$
(2) $\stackrel{\epsilon}{\Longrightarrow}_{\times} =_{df} (\stackrel{\underline{\tau}}{\longrightarrow} \cup \stackrel{\tau}{\longrightarrow})^{*}$
(3) $\stackrel{\gamma}{\Longrightarrow}_{\times} =_{df} \stackrel{\epsilon}{\Longrightarrow}_{\times} \circ \stackrel{\gamma}{\longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}_{\times}$

Observe that this transition relation ignores priority levels for $\stackrel{\epsilon}{\Longrightarrow}_{\times}$. This is in accordance with the fact that a priority value is part of an action name and, thus, unobservable for internal actions.

Definition 3.5 (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.*

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

We write $P \approx_{\times} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for a naive prioritized weak bisimulation $\mathcal{R}$.

Naive prioritized weak bisimulation can be shown to be an equivalence. Unfortunately, $\approx_{\times}$ is not a congruence for $\mathsf{CCS}^{\mathsf{sg}}$ with respect to parallel composition and summation. Whereas the compositionality defect for summation is similar to the one for observational equivalence in CCS [?], the problem with parallel composition is due to pre-emption. As an example consider the processes $P \stackrel{\text{def}}{=} a.\mathbf{0} + \underline{b}.\mathbf{0}$ and $Q \stackrel{\text{def}}{=} a.\mathbf{0} + \tau.(a.\mathbf{0} + \underline{b}.\mathbf{0})$. It is easy to see that $P \approx_{\times} Q$. However, when composing these processes in parallel with process $\overline{\underline{b}}.\mathbf{0}$, $Q \,|\, \overline{\underline{b}}.\mathbf{0} \stackrel{a}{\longrightarrow} \mathbf{0} \,|\, \overline{\underline{b}}.\mathbf{0}$ whereas $P \,|\, \overline{\underline{b}}.\mathbf{0} \stackrel{a}{\not\Longrightarrow}_{\times}$, i.e., $P \,|\, \overline{\underline{b}}.\mathbf{0} \not\approx_{\times} Q \,|\, \overline{\underline{b}}.\mathbf{0}$. In order to obtain a congruence, more care must be taken when defining the prioritized weak transition relation. Transitions labeled by visible actions may turn into internal transitions when composed with an environment and, thus, may gain pre-emptive power. An adequate notion of weak transitions must take the processes' potential of engaging in visible prioritized transitions into account.

3.3.1 Prioritized Weak Bisimulation

Despite its lack of compositionality, the definition of $\approx_{\times}$ reflects an intuitive approach to abstracting from internal computation, and consequently we wish to develop a congruence that is as "close" to $\approx_{\times}$ as possible. That such a relation exists is due to the following fact from universal algebra.

Proposition 3.6 *For an equivalence $\mathcal{R}$ in an algebra $\Re$ a largest congruence $\mathcal{R}^{+} \subseteq \mathcal{R}$ exists and $\mathcal{R}^{+} = \{\langle P, Q\rangle \,|\, \forall\, \Re\text{-contexts } C[X].\, \langle C[P], C[Q]\rangle \in \mathcal{R}\}$.*

Here, a $\Re$-context $C[X]$ is a term in $\Re$ with one free occurrence of variable X.

We therefore know that $\approx_\times$ contains a largest congruence $\approx_\times^+$ for $\mathsf{CCS^{sg}}$ and devote the rest of this section to giving an operational characterization of this congruence. We begin by defining a new weak transition relation that takes pre-emptability into account.

Definition 3.7 (Prioritized Transitions) *Let $L \subseteq \underline{A} \setminus \{\underline{\tau}\}$.*

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

(ii) $P \xrightarrow[L]{\alpha} P'$ *if* $P \xrightarrow{\alpha} P'$ *and* $\underline{\mathbb{II}}(P) \subseteq L$

(iii) $\xRightarrow{\underline{\epsilon}} =_{df} (\xrightarrow{\underline{\tau}})^*$

(iv) $\xRightarrow[L]{\epsilon} =_{df} (\xrightarrow{\underline{\tau}} \cup \xrightarrow[L]{\tau})^*$

(v) $\xRightarrow{\underline{\alpha}} =_{df} \xRightarrow{\underline{\epsilon}} \circ \xrightarrow{\underline{\alpha}} \circ \xRightarrow{\underline{\epsilon}}$

(vi) $\xRightarrow[L]{\alpha} =_{df} \xRightarrow[L]{\epsilon} \circ \xrightarrow[L]{\alpha} \circ \xRightarrow{\underline{\epsilon}}$

Intuitively, we have made the transition relation sensitive to pre-emption by introducing conditions involving prioritized initial action sets and by preserving priority levels of internal actions. In particular, $P \xrightarrow[L]{\alpha} P'$ holds when P can evolve to P' by performing the unprioritized action α, provided the environment does not offer any prioritized communication involving an action in L, which contains the prioritized actions P may initially engage in. In the remainder, we show that such prioritized initial action sets are an adequate means for measuring pre-emption potentials.

Definition 3.8 (Prioritized Weak Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a prioritized weak bisimulation *if for every $\langle P, Q \rangle \in \mathcal{R}$, $\underline{\alpha} \in \underline{A}$, and $\alpha \in A$ the following conditions hold.*

(1) $\underline{\tau} \notin \underline{\mathbb{I}}(P)$ *implies* $\exists Q'.\, Q \xRightarrow[L]{\epsilon} Q'$, $\underline{\mathbb{II}}(Q') \subseteq L$ *where* $L = \underline{\mathbb{II}}(P)$, $\underline{\tau} \notin \underline{\mathbb{I}}(Q')$, *and* $\langle P, Q' \rangle \in \mathcal{R}$.

(2) $P \xrightarrow{\underline{\alpha}} P'$ *implies* $\exists Q'.\, Q \xRightarrow{\hat{\underline{\alpha}}} Q'$, *and* $\langle P, Q' \rangle \in \mathcal{R}$.

(3) $P \xrightarrow{\alpha} P'$ *implies* $\exists Q'.\, Q \xRightarrow[L]{\hat{\alpha}} Q'$, *where* $L = \underline{\mathbb{II}}(P)$, *and* $\langle P', Q' \rangle \in \mathcal{R}$.

We write $P \underline{\approx} Q$ if $\langle P, Q \rangle \in \mathcal{R}$ for some prioritized weak bisimulation $\mathcal{R}$.

This new version of weak bisimulation is algebraically more robust than the naive one, as the next result indicates. It should be noted that Condition (**??**) of Definition **??** is necessary for achieving compositionality with respect to parallel composition [**?**].

Proposition 3.9 *The equivalence $\underline{\approx}$ is a congruence with respect to prefixing, parallel composition, relabeling, and restriction. Moreover, $\underline{\approx}$ is the* largest congruence *contained in $\approx_\times$ with respect to the sub-algebra of $\mathsf{CCS^{sg}}$ induced by these operators and recursion.*

Although $\underline{\approx}$ is itself not a congruence for $\mathsf{CCS}^{\mathsf{sg}}$, this relation provides the basis for obtaining a congruence, as is made precise in the next section.

3.3.2 Prioritized Observational Congruence

The compositionality defect of $\underline{\approx}$ with respect to summation is handled by the following notion of *prioritized observational congruence*. Unfortunately, the summation fix presented in [**?**], which requires an initial internal transition to be matched by a nontrivial internal weak transition, is not sufficient in order to achieve a congruence based on prioritized weak bisimulation. To see why, let $D \stackrel{\mathrm{def}}{=} \underline{\tau}.E$ and $E \stackrel{\mathrm{def}}{=} \tau.D$. Now define $P \stackrel{\mathrm{def}}{=} \tau.D$ and $Q \stackrel{\mathrm{def}}{=} \underline{\tau}.E$. By Definition **??** we may observe $P \,\underline{\approx}\, Q$, but $P + a.\mathbf{0} \;\underline{\not\approx}\; Q + a.\mathbf{0}$ since the former can perform an a-transition whereas the latter cannot. It turns out that we must require that observationally congruent processes must possess the same initial action sets; a requirement which is stronger than Condition (**??**) of Definition **??**.

Definition 3.10 *Define* $P \,\underline{\approx}^{\mathrm{l}}\, Q$ *if for all* $\underline{\alpha} \in \underline{A}$ *and* $\alpha \in A$ *the following conditions and their symmetric counterparts hold.*

(1) $\underline{\mathcal{I}}(P) \supseteq \underline{\mathcal{I}}(Q)$
(2) $P \stackrel{\underline{\alpha}}{\longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\underline{\alpha}}{\Longrightarrow} Q'$ *and* $P' \,\underline{\approx}\, Q'$*.*
(3) $P \stackrel{\alpha}{\longrightarrow} P'$ *implies* $\exists Q'.\, Q \underset{L}{\stackrel{\alpha}{\Rightarrow}} Q'$*, where* $L = \underline{\underline{\mathcal{I}}}(P)$*, and* $P' \,\underline{\approx}\, Q'$*.*

The following theorem states the desired algebraic result for $\underline{\approx}^{\mathrm{l}}$.

Theorem 3.11 $\underline{\approx}^{\mathrm{l}}$ *is the largest congruence contained in* $\approx_{\times}$*, i.e.,* $\underline{\approx}^{\mathrm{l}} \;=\; \approx_{\times}^{+}$*.*

Whereas the proof of the congruence property of $\underline{\approx}^{\mathrm{l}}$ is standard (cf. [**?**]), the "largest" part is proved by using the following fact from universal algebra.

Proposition 3.12 *Let* $\mathcal{R}_1$ *and* $\mathcal{R}_2$ *be equivalences in an algebra* $\Re$ *such that* $\mathcal{R}_1^{+} \subseteq \mathcal{R}_2 \subseteq \mathcal{R}_1$*. Then* $\mathcal{R}_1^{+} = \mathcal{R}_2^{+}$*.*

For the purposes of this section, we choose $\mathcal{R}_1 \;=\; \approx_{\times}$ and $\mathcal{R}_2 \;=\; \underline{\approx}$. The following theorem, which establishes $\mathcal{R}_2^{+} = \underline{\approx}^{\mathrm{l}}$, can be proved along the lines of a corresponding theorem in [**?**]; for details see [**?**].

Theorem 3.13 $\underline{\approx}^{\mathrm{l}}$ *is the* largest *congruence contained in* $\underline{\approx}$*.*

In order to apply Proposition **??**, $\approx_{\times}^{+} \subseteq \underline{\approx} \subseteq \approx_{\times}$ needs to be established. The inclusion $\underline{\approx} \subseteq \approx_{\times}$ immediately follows from the definition of the naive prioritized weak and the prioritized weak transition relations. One is left with $\approx_{\times}^{+} \subseteq \underline{\approx}$. This inclusion turns out to be difficult to show directly. Instead, the auxiliary relation $\underline{\approx}_{\mathrm{a}} =_{\mathrm{df}} \{\langle P, Q\rangle \mid C_{PQ}[P] \approx_{\times} C_{PQ}[Q]\}$ satisfying the inclusions $\approx^+_\times \subseteq \underline{\approx}_{\mathrm{a}} \subseteq \underline{\approx}$ is defined. Using the abbreviation $\underline{S}$ for the union of the finite prioritized sorts of P and Q, we define $C_{PQ}[X] \stackrel{\mathrm{def}}{=} X \mid H_{PQ}$, where

$$H_{PQ} \stackrel{\mathrm{def}}{=} \underline{c}.\mathbf{0} + \sum_{\substack{L \subseteq \overline{\underline{S}}, \\ \underline{b} \in \underline{S}}} \underline{\tau}. \left(\begin{array}{c} \underline{d}_{L,\underline{b}}.H_{PQ} + \\ D_L + \underline{e}.H_{PQ} + \\ \overline{\underline{b}}.H_{PQ} \end{array} \right)$$

and D_L is defined as $\sum_{\underline{\alpha} \in L} \underline{\alpha}.\mathbf{0}$. Actions $\underline{c}, \underline{d}_{L,\underline{b}}, \underline{e}$, for all $L \subseteq \overline{\underline{S}}$ and $\underline{b} \in \underline{S}$, and their complements, are assumed to be "fresh" actions, i.e., not in $\underline{S} \cup \overline{\underline{S}}$. By Proposition **??** we may conclude $\approx^+_\times \subseteq \underline{\approx}_{\mathrm{a}}$. The other necessary inclusion $\underline{\approx}_{\mathrm{a}} \subseteq \underline{\approx}$ is established by showing that $\underline{\approx}_{\mathrm{a}}$ is a prioritized weak bisimulation. Summarizing, Theorem **??** is a consequence of Proposition **??** when assembling our results as illustrated in Figure **??**, where an arrow from relation R_1 to relation R_2 indicates that $R_1 \subseteq R_2$.

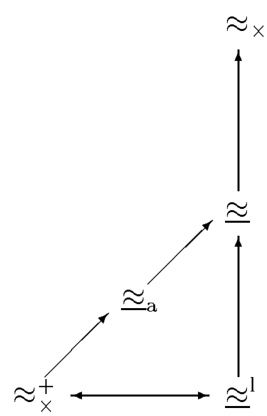

Fig. 2. Proof sketch of Theorem **??**

3.3.3 Operational Characterization

The aim of this section is to show how prioritized weak bisimulation can be computed by adapting standard *partition-refinement algorithms* [?,?,?] developed for strong bisimulation [?]. To this end, we provide an operational characterization of prioritized weak bisimulation as strong bisimulation by introducing an *alternative prioritized weak transition relation.*

Definition 3.14 *Let $P, P' \in \mathcal{P}$, $\alpha \in A$, and $\underline{\alpha} \in \underline{A}$.*
(1) $\stackrel{\hat{\alpha}}{\Longrightarrow}_* =_{df} \stackrel{\hat{\alpha}}{\Longrightarrow}$ *and*
(2) $P \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow}_* P'$ *if* $\exists P''.\ \underline{\tau} \notin \underline{\mathcal{I}}(P'')$ *and* $P \stackrel{\epsilon}{\Longrightarrow} P'' \stackrel{\hat{\underline{\alpha}}}{\underset{L}{\Longrightarrow}} P'$, *for* $L = \underline{\mathcal{II}}(P'')$.

Observe that the alternative prioritized weak transition relation is not parameterized by prioritized initial action sets. It can be computed efficiently by using *dynamic programming* techniques.

Definition 3.15 *A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is called an* alternative prioritized weak bisimulation *if for all $\langle P, Q \rangle \in \mathcal{R}$ and $\gamma \in \mathcal{A}$ the following condition holds.*

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

We write $P \underset{\sim}{\approx}_* Q$ if $\langle P, Q \rangle \in R$ for some alternative prioritized weak bisimulation $\mathcal{R}$.

Theorem 3.16 (Operational Characterization) $\underset{\sim}{\approx} = \underset{\sim}{\approx}_*$.

The proof of this characterization result is straightforward [**?**]. It should be mentioned that the presented characterization may also serve as a basis for defining a Hennessy–Milner logic for prioritized weak bisimulation along the lines of [**?**].

3.4 Example

As a simple example, we return to the back-and-forth system introduced in Section **??**, which can be formalized in $\mathsf{CCS}^{\mathsf{sg}}$ as follows: $\mathsf{Sys} \stackrel{\text{def}}{=} (A \,|\, B) \setminus \{\underline{i}\}$ where $A \stackrel{\text{def}}{=} \mathsf{back}.A' + \underline{i}.\tau.\overline{\mathsf{ok}}.\overline{\underline{i}}.A$, $A' \stackrel{\text{def}}{=} \mathsf{forth}.A + \underline{i}.\tau.\overline{\mathsf{ok}}.\overline{\underline{i}}.A'$, and $B \stackrel{\text{def}}{=} \mathsf{check}.\overline{\underline{i}}.\underline{i}.B$. Intuitively, $\underline{i}$ is an internal *interrupt*, and thus prioritized and restricted (via $\setminus\{\underline{i}\}$), which is invoked whenever check is executed. Hence, processes A and A' cannot engage in a transition labeled by back and forth, respectively, according to our pre-emptive operational semantics, but must accept the communication on prioritized port $\underline{i}$. One can think of the τ-action in the definition of process A as representing some internal activities determining the current status of the system. The $\mathsf{CCS}^{\mathsf{sg}}$ semantics of Sys is shown in Figure **??**.

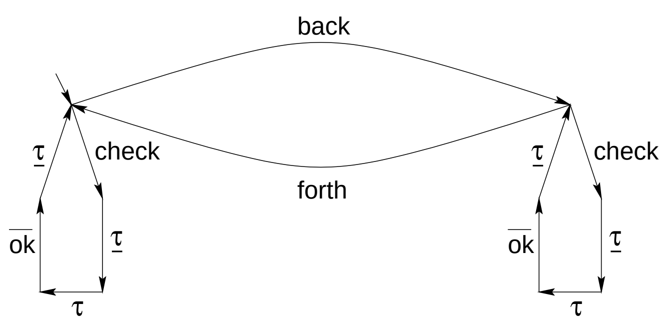

Fig. 3. Semantics of Sys

In the sequel, we prove that Sys meets its specification Spec, which is given by

$$\begin{aligned}\mathsf{Spec} &\stackrel{\text{def}}{=} \mathsf{back}.\mathsf{Spec}' + \mathsf{check}.\overline{\mathsf{ok}}.\mathsf{Spec}\\ \mathsf{Spec}' &\stackrel{\text{def}}{=} \mathsf{forth}.\mathsf{Spec} + \mathsf{check}.\overline{\mathsf{ok}}.\mathsf{Spec}' \,.\end{aligned}$$

First, the validity of $\mathsf{Sys} \approx \mathsf{Spec}$ is demonstrated by the relation presented in Table **??**, whose symmetric closure is a prioritized weak bisimulation that contains $\langle \mathsf{Sys}, \mathsf{Spec} \rangle$.

Table 4
A relation whose symmetric closure is a prioritized weak bisimulation

$$
\begin{array}{lll}
\{ \langle \mathsf{Sys} & , \mathsf{Spec} \;\rangle , & \langle (A' \,|\, B) \setminus \{\underline{i}\} \quad , \mathsf{Spec}' \;\rangle , \\
\;\langle (A \,|\, \overline{\underline{i}}.\underline{i}.B) \setminus \{\underline{i}\} & , \overline{\mathsf{ok}}.\mathsf{Spec} \;\rangle , & \langle (\tau.\overline{ok}.\overline{\underline{i}}.A \,|\, \underline{i}.B) \setminus \{\underline{i}\} \;, \overline{\mathsf{ok}}.\mathsf{Spec} \rangle , \\
\;\langle (\overline{ok}.\overline{\underline{i}}.A \,|\, \underline{i}.B) \setminus \{\underline{i}\} & , \overline{\mathsf{ok}}.\mathsf{Spec} \;\rangle , & \langle (\overline{\underline{i}}.A \,|\, \underline{i}.B) \setminus \{\underline{i}\} \quad , \mathsf{Spec} \;\rangle , \\
\;\langle (A' \,|\, \overline{\underline{i}}.\underline{i}.B) \setminus \{\underline{i}\} & , \overline{\mathsf{ok}}.\mathsf{Spec}' \rangle , & \langle (\tau.\overline{ok}.\overline{\underline{i}}.A' \,|\, \underline{i}.B) \setminus \{\underline{i}\} \;, \overline{\mathsf{ok}}.\mathsf{Spec} \rangle , \\
\;\langle (\overline{ok}.\overline{\underline{i}}.A' \,|\, \underline{i}.B) \setminus \{\underline{i}\} & , \overline{\mathsf{ok}}.\mathsf{Spec}' \rangle , & \langle (\overline{\underline{i}}.A' \,|\, \underline{i}.B) \setminus \{\underline{i}\} \quad , \mathsf{Spec}' \;\rangle \; \}
\end{array}
$$

In addition, both processes only possess visible initial actions, and their prioritized initial action sets are identical. Hence, we may conclude $\mathsf{Sys} \underline{\approx}^{\mathsf{l}} \mathsf{Spec}$.

3.5 Prioritization and Deprioritization Operators

There are several other language constructs of interest in the setting of priorities. Of particular interest are the *prioritization* and *deprioritization* operators introduced in [**?**]. The former operator may be used to raise the priority of a visible unprioritized action and is written $\lceil a$, for $a \not\equiv \tau$; the latter lowers the priority of a visible prioritized action and is written $\lfloor \underline{a}$, for $\underline{a} \not\equiv \underline{\tau}$. The operational semantics of these operators is formally defined in Table **??**. For these operators to be well-defined there must be a one-to-one correspondence between prioritized ports $\underline{a}$ and unprioritized ports a; in addition every relabeling f must satisfy $f(\underline{a}) = \underline{f(a)}$. Intuitively, $P\lceil a$ prioritizes all a-transitions that P can perform, while $P\lfloor \underline{a}$ deprioritizes all $\underline{a}$-transitions in which P can engage. Observe that the notion of priority is still static and not dynamic since the prioritization and deprioritization operators are static operators and therefore define a scope.

Including prioritization and deprioritization operators with $\mathsf{CCS}^{\mathsf{sg}}$ does not have any consequences for prioritized strong bisimulation since it is compositional with respect to these operators [**?**]. The axiomatization of prioritized strong bisimulation for finite processes can also be extended to cover the new operators. The necessary additional axioms are stated in Table **??**.

The presence of the prioritization and the deprioritization operators suggests a formal justufucation for the design decision that only prioritized internal actions have pre-emptive power over unprioritized actions. For this purpose assume that (i) pre-emption is not encoded in the side conditions of the operational rules but, equivalently, in the notion of bisimulation [**?**] and that

Table 5
Semantics for the prioritization and the deprioritization operator

$$\textsf{Prio1}\ \frac{P \xrightarrow{a} P'}{P\lceil a \xrightarrow{\underline{a}} P'\lceil a} \qquad \textsf{Deprio1}\frac{P \xrightarrow{\underline{a}} P'}{P\lfloor \underline{a} \xrightarrow{a} P'\lfloor \underline{a}}\ \underline{\tau} \notin \underline{\mathcal{I}}(P)$$

$$\textsf{Prio2}\ \frac{P \xrightarrow{\gamma} P'}{P\lceil a \xrightarrow{\gamma} P'\lceil a}\ \gamma \not\equiv a \qquad \textsf{Deprio2}\frac{P \xrightarrow{\underline{a}} P'}{P\lfloor \underline{a} \xrightarrow{\underline{a}} P'\lfloor \underline{a}}\ \underline{\tau} \in \underline{\mathcal{I}}(P)$$

$$\textsf{Deprio3}\frac{P \xrightarrow{\gamma} P'}{P\lfloor \underline{a} \xrightarrow{\gamma} P'\lfloor \underline{a}}\ \gamma \not\equiv \underline{a}$$

Table 6
Axioms for the prioritization and the deprioritization operator

(Prio1)	$\mathbf{0}\lceil a$	$=$	$\mathbf{0}$	
(Prio2)	$(a.t)\lceil a$	$=$	$\underline{a}.(t\lceil a)$	
(Prio3)	$(\gamma.t)\lceil a$	$=$	$\gamma.(t\lceil a)$	$\gamma \not\equiv a$
(Prio4)	$(t+\underline{\tau}.u+\underline{b}.v)\lceil a$	$=$	$(t+\underline{\tau}.u)\lceil a+\underline{b}.(v\lceil a)$	
(Prio5)	$(t+\delta.u+\gamma.v)\lceil a$	$=$	$(t+\delta.u)\lceil a+(t+\gamma.v)\lceil a$	$\delta,\gamma \in \mathcal{A}\setminus\{\underline{\tau}\}$
(Deprio1)	$\mathbf{0}\lfloor \underline{a}$	$=$	$\mathbf{0}$	
(Deprio2)	$(\underline{a}.t)\lfloor \underline{a}$	$=$	$a.(t\lfloor \underline{a})$	
(Deprio3)	$(\gamma.t)\lfloor \underline{a}$	$=$	$\gamma.(t\lceil \underline{a})$	$\gamma \not\equiv \underline{a}$
(Deprio4)	$(t+\underline{\tau}.u+\underline{b}.v)\lfloor \underline{a}$	$=$	$(t+\underline{\tau}.u)\lfloor \underline{a}+\underline{b}.(v\lfloor \underline{a})$	
(Deprio5)	$(t+\delta.u+\gamma.v)\lfloor \underline{a}$	$=$	$(t+\delta.u)\lfloor \underline{a}+(t+\gamma.v)\lfloor \underline{a}$	$\delta,\gamma \in \mathcal{A}\setminus\{\underline{\tau}\}$

(ii) the naive view of pre-emption gives all prioritized actions pre-emptive power. Thus, a naive bisimulation $\sim_{\rm n}$ demands the following condition for equivalent processes $P \sim_{\rm n} Q$ and unprioritized actions $\alpha \in A$:

$$(P \xrightarrow{\alpha} P' \text{ and } \not\exists \underline{\beta}.\, P \xrightarrow{\beta}) \text{ implies } (\exists Q'.\, Q \xrightarrow{\alpha} Q',\ \not\exists \underline{\beta}.\, Q \xrightarrow{\beta}, \text{ and } P' \sim_{\rm n} Q')$$

and vice versa. The condition for prioritized actions can be adopted from standard strong bisimulation. It turns out that $\sim_{\rm n}$ is not a congruence; for example, $a.\mathbf{0}+\underline{b}.\mathbf{0} \sim_{\rm n} \underline{b}.\mathbf{0}$ but $(a.\mathbf{0}+\underline{b}.\mathbf{0})\setminus\{\underline{b}\} \not\sim_{\rm n} (\underline{b}.\mathbf{0})\setminus\{\underline{b}\}$ since the former process can engage in an a-transition while the latter is deadlocked. Then the question arises what the largest congruence contained in $\sim_{\rm n}$ is; it turns out that it is prioritized strong congruence as defined above (see [?] for details). This shows that in a pre-emptive semantics only prioritized internal actions may pre-empt unprioritized actions. However, the above algebraic result is only correct if we include the deprioritization operator in our language. An operational characterization of the largest congruence contained in $\sim_{\rm n}$ with respect to $\mathsf{CCS}^{\mathsf{sg}}$ is an open problem.

For the language extended by the prioritization and the deprioritization operator, an observational congruence together with an axiomatic characterization with respect to finite processes has been developed in [?,?], which is briefly reviewed here. For this purpose, we need to refine the prioritized weak transition relation. First, we re-define $\stackrel{a}{\Longrightarrow}$ to $\stackrel{\epsilon}{\Longrightarrow} \circ \stackrel{a}{\longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}$, i.e., a weak unprioritized a-transition consists of an a-transition that is preceded and trailed by *prioritized* internal transitions only. Moreover, we replace $\underline{I\!I}(P)$ by $\underline{I\!I}(P) \cup I\!I(P)$ in the definition of $\stackrel{\epsilon}{\Longrightarrow}$ since one has to take into account the fact that unprioritized actions may turn into prioritized ones when applying the prioritization operator. Finally, we write $P \overset{\tau}{\underset{L}{\Rightarrow}} P'$ whenever $P \overset{\epsilon}{\underset{L}{\Rightarrow}} P'$ and $P \not\equiv P'$. Consequently, visible weak unprioritized transitions only abstract from prioritized internal actions. The reason for this restriction is that, otherwise, prioritized weak bisimulation would not be compositional with respect to the prioritization and the deprioritization operator. In contrast, the original prioritized weak transition relation allows an α-transition to be preceded by any sequence of $\underline{\tau}$- and τ-transitions (satisfying a condition on prioritized initial action sets) and only to be trailed by $\underline{\tau}$-transitions.

The notions of prioritized weak bisimulation and prioritized observational congruence are defined in [?,?] as follows, where $P\underline{\Downarrow}$ stands for $\exists P'.\, P \stackrel{\epsilon}{\Longrightarrow} P'$ and $P' \stackrel{\underline{\tau}}{\not\longrightarrow}$.

Definition 3.17 *A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is an* extended prioritized weak bisimulation *if for every $\langle P, Q\rangle \in \mathcal{R}$ and $\gamma \in \mathcal{A} \setminus \{\tau\}$ the following conditions hold.*

(1) $P\underline{\Downarrow}$ implies $Q\underline{\Downarrow}$.
(2) $P \stackrel{\gamma}{\longrightarrow} P'$ implies $\exists Q'.\, Q \stackrel{\hat{\gamma}}{\Longrightarrow} Q'$, and $\langle P, Q'\rangle \in \mathcal{R}$.
(3) $P \stackrel{\tau}{\longrightarrow} P'$ implies $\exists Q'.\, Q \overset{\epsilon}{\underset{L}{\Rightarrow}} Q'$, $L = \underline{I\!I}(P) \cup I\!I(P)$, and $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \approxeq_{\rm pd} Q$ if there exists an extended prioritized weak bisimulation $\mathcal{R}$ such that $\langle P, Q\rangle \in \mathcal{R}$.

Definition 3.18 *We define $P \approxeq^{\rm l}_{\rm pd} Q$ if for all $\gamma \in \mathcal{A} \setminus \{\tau\}$ the following conditions and their symmetric counterparts hold.*

Table 7

Axioms for the τ-laws

$(\underline{\tau}1)$	$\gamma.(1.t+t)$	$=$	$\gamma.t$	$1 \in \{\tau, \underline{\tau}\}$
$(\underline{\tau}2)$	$\underline{\tau}.t$	$=$	$\underline{\tau}.t+t$	
$(\underline{\tau}3)$	$\gamma.(t+\underline{\tau}.u)$	$=$	$\gamma.(t+\underline{\tau}.u)+\gamma.u$	
$(\tau 1)$	$t+\tau.(u+\tau.v)$	$=$	$t+\tau.(u+\tau.v)+\tau.v$	$\vdash_I t \sqsubseteq_{\mathrm{i}} v$

Table 8

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

(iC1) $\underline{\alpha}.t \sqsubseteq_{\mathrm{i}} \underline{\alpha}.u$	(iC2) $\mathbf{0} \sqsubseteq_{\mathrm{i}} \gamma.t \quad \gamma \in \mathcal{A} \setminus \{\underline{\tau}\}$	(iC3) $\alpha.t \sqsubseteq_{\mathrm{i}} \mathbf{0}$

(1) $P \xrightarrow{\gamma} P'$ *implies* $\exists Q'\, Q \overset{\gamma}{\Longrightarrow} Q'$ *and* $P' \approx_{\mathrm{pd}} Q'$.
(2) $P \xrightarrow{\tau} P'$ *implies* $\exists Q'.\, Q \underset{L}{\overset{\tau}{\Rightarrow}} Q'$, *where* $L = \underline{\mathbb{II}}(P) \cup \mathbb{II}(P)$, *and* $P' \approx_{\mathrm{pd}} Q'$.

The observational congruence $\approx^{\mathrm{l}}_{\mathrm{pd}}$ possesses nice algebraic properties for our language extended by the prioritization and the deprioritization operators, including a largest congruence result similar to Theorem **??** and a sound and complete axiomatization for finite processes. For the latter, the axiomatization for prioritized strong bisimulation is augmented with suitable τ-laws as shown in Table **??** (cf. [**?**]). The relation $\sqsubseteq_{\mathrm{i}}$, occurring in the side condition of Axiom $(\tau 1)$, is the pre-congruence on finite processes generated by the axioms presented in Table **??** using the laws of inequational reasoning. We write $\vdash_I t \sqsubseteq_{\mathrm{i}} u$ if t can be related to u by Axioms (iC1), (iC2), and (iC3). Intuitively, $\vdash_I t \sqsubseteq_{\mathrm{i}} u$ holds, whenever (i) $\underline{\tau} \in \underline{\mathcal{I}}(t)$ if and only if $\underline{\tau} \in \underline{\mathcal{I}}(u)$ and (ii) $\underline{\mathbb{II}}(t) \subseteq \underline{\mathbb{II}}(u)$.

Finally, it should be noted that applications underline the importance of the additional abstraction from internal transitions gained by leaving out the prioritization and deprioritization operators. Indeed, the observational congruence $\approx^{\mathrm{l}}_{\mathrm{pd}}$ does not relate processes Sys and Spec of our back-and-forth example [**?**]. This is due to the presence of action τ in Sys.

3.6 Extension to Multi-level Priority Schemes

We now remark on the extension of $\mathsf{CCS}^{\mathsf{sg}}$ to a multi-level priority scheme. We first alter the definition of prioritized initial action sets to capture the priority-level of actions by introducing sets $\mathrm{I}^k(P)$ for process P with respect to priority value k as shown in Table **??**.

Using this definition of initial action sets and the notation "$\tau \notin \mathrm{I}^{<k}(P)$ if

Table 9
Potential initial action sets for $\mathsf{CCS}^{\mathsf{sg}}$

$\mathrm{I}^k(\alpha{:}l.P) = \{\alpha{:}l \mid l = k\}$	$\mathrm{I}^k(P[f]) \;= \{f(\alpha{:}l) \mid \alpha{:}l \in \mathrm{I}^k(P)\}$
$\mathrm{I}^k(\mu x.P) \;= \mathrm{I}^k(P[\mu x.P/x])$	$\mathrm{I}^k(P{+}Q) = \mathrm{I}^k(P) \cup \mathrm{I}^k(Q)$
$\mathrm{I}^k(P \setminus L) = \mathrm{I}^k(P) \setminus (L \cup \overline{L})$	$\mathrm{I}^k(P \mid Q) \;= \mathrm{I}^k(P) \cup \mathrm{I}^k(Q) \cup \{\tau{:}k \mid \mathrm{I}^k(P) \cap \overline{\mathrm{I}^k(Q)} \neq \emptyset\}$

$\nexists l < k.\ \tau{:}l \in \mathrm{I}^l(P)$," the operational semantics depicted in Table **??** can be restated in the manner illustrated by the following modification to Rule ($\mathsf{Com3}$).

$$\mathsf{Com3}\ \frac{P \stackrel{a:k}{\longrightarrow} P' \quad Q \stackrel{\overline{a}:k}{\longrightarrow} Q'}{P \mid Q \stackrel{\tau:k}{\longrightarrow} P' \mid Q'}\ \tau \notin \mathrm{I}^{<k}(P \mid Q)$$

Observe that the sets $\mathrm{I}^k(P)$ may contain actions in which P cannot initially engage, since their definition does not consider pre-emption. In fact, the set $\mathcal{I}^k(P)$ of actions with priority value k in which P can indeed initially engage is given by $\{\alpha{:}k \in \mathrm{I}^k(P) \mid \tau{:}l \notin \mathrm{I}^l(P)$, for all $l < k\}$. However, it is easy to show that $\tau \notin \mathrm{I}^{<k}(P)$ if and only if $\tau \notin \mathcal{I}^{<k}(P)$ [**?**]. Thus, the side condition of Rule ($\mathsf{Com3}$) captures our intuition that $P \mid Q$ cannot engage in a more urgent internal transition.

Table 10
Prioritized weak transition relation

$\stackrel{\epsilon:0}{\Longrightarrow} \;=_{\mathrm{df}} (\stackrel{\tau:0}{\longrightarrow})^*$	$P \underset{L}{\stackrel{\alpha:k}{\to}} P'$ if $P \stackrel{\alpha:k}{\longrightarrow} P'$ and $\mathit{I\!I}^l(P) \subseteq L$ for all $l < k$
$\underset{L}{\stackrel{\epsilon:k}{\Rightarrow}} \;=_{\mathrm{df}} (\{ \underset{L}{\stackrel{\tau:l}{\to}} \mid l \leq k\})^*$	$\underset{L}{\stackrel{\alpha:k}{\Rightarrow}} \;=_{\mathrm{df}} \underset{L}{\stackrel{\epsilon:k}{\Rightarrow}} \circ \underset{L}{\stackrel{\alpha:k}{\to}} \circ \stackrel{\epsilon:0}{\Longrightarrow}$

The re-development of the bisimulation-based semantic theory proceeds along the lines of the above sections and does not raise any new semantic issues. For example, the notion of prioritized observational congruence is defined as follows, where (i) the prioritized weak transition relation is given by the rules in Table **??**, (ii) $\mathit{I\!I}^k(P) =_{\mathrm{df}} \mathcal{I}^k(P) \setminus \{\tau{:}k\}$, (iii) the relation $\approxeq_{\mathsf{ml}}$ is the adaption of prioritized weak bisimulation to the multi-level priority scheme, (iv) $\mathcal{I}(P) =_{\mathrm{df}} \bigcup\{\mathcal{I}^k(P) \mid k \in \mathcal{N}\}$, and (v) $\mathit{I\!I}^{<k}(P) =_{\mathrm{df}} \mathcal{I}^{<k}(P) \setminus \{\tau{:}l \mid l < k\}$.

Definition 3.19 *Processes P and Q are* prioritized observational congruent *if for all actions $\alpha{:}k$ the following conditions and their symmetric counterparts hold.*

(1) $\mathcal{I}(P) \supseteq \mathcal{I}(Q)$
(2) $P \stackrel{\alpha:k}{\longrightarrow} P'$ *implies* $\exists Q'.\, Q \underset{L}{\stackrel{\alpha:k}{\Rightarrow}} Q'$, *where* $L = \mathit{I\!I}^{<k}(P)$, *and* $P' \approxeq_{\mathsf{ml}} Q'$

Details of this extension of $\mathsf{CCS}^{\mathsf{sg}}$ can be found in [**?**].

3.7 *Concluding Remarks and Related Work*

We conclude by first commenting on the design decision that priority values are considered to be part of port names, which implies that only complementary actions having the same priority can synchronize. Lifting this design decision by allowing $a\!:\!k$ and $\overline{a}\!:\!l$, where $k \neq l$, to synchronize leads to the question of which priority value to assign to the resulting τ. One can imagine several obvious choices for this function, e.g., maximum or minimum. On the other hand, [?,?] recommend using the sum of the priority values of the actions involved. Unfortunately, while a specific function may be suitable for certain examples, it often appears *ad hoc* in general. In the next section we will see that such a function is unnecessary when dealing with *local pre-emption*.

Regarding related work, Gerber and Lee developed a real-time process algebra, the *Calculus of Communicating Shared Resources* (CCSR) [?], that explicitly takes the availability of system resources into account. Synchronizations between processes are modeled in an interleaving fashion using instantaneous transitions, whereas the access of resources is truly concurrent and consumes time. In CCSR a priority structure may be defined over resources in order to indicate their importance, e.g., for ensuring that deadlines are met. The underlying concept of priority is similar to that of $\mathsf{CCS}^{\mathsf{sg}}$ in that priorities are static and pre-emption is global. In [?] a resource-based prioritized (strong) bisimulation for CCSR together with axiomatizations for several classes of processes [?] is presented.

Prasad extended his *Calculus of Broadcasting Systems* (CBS) in [?] for dealing with a notion of static priority [?]. He refers to the priority calculus as PCBS. For PCBS nice semantic theories based on Milner's strong and weak bisimulation [?] have been developed. Remarkably, these theories do not suffer from the technical subtleties which have been encountered for $\mathsf{CCS}^{\mathsf{sg}}$, although the concept of pre-emption is basically the same. The reason is that PCBS uses a much simpler model for communication which is based on the principle of *broadcasting*. In this setting, priority values are only attached to output actions, which cannot be restricted or hidden as in traditional process algebras. Finally, it should be mentioned that PCBS contains an operator, called *translate*, which enables the prioritization and the deprioritization of actions.

4 Static Priority and Local Pre-emption

This section provides a new semantics, $\mathsf{CCS}^{\mathsf{sl}}$(CCS with $\mathsf{s}$tatic priority and $\mathsf{l}$ocal pre-emption), for the language introduced in Section ??. The new semantics is distinguished from the one developed in the previous section in that it only allows actions to pre-empt others at the same "location" and, therefore, captures a notion of *localized precedence*. 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 an interaction, e.g., a synchronization, between them.

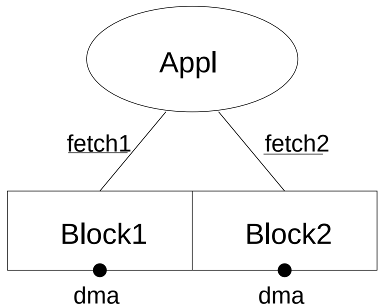

Fig. 4. Example system

The following example illustrates why locations should be taken into account when reasoning about priority within distributed systems. The system in question consists of an application that manipulates data from two blocks of memory (see Figure **??**). In order to improve efficiency, each memory block is connected to a direct-memory-access (DMA) controller. Assume further, that application Appl fetches pieces of data alternately from each memory block, i.e., $\mathsf{Appl} \stackrel{\text{def}}{=} \overline{\underline{\mathsf{fetch1}}}.\overline{\underline{\mathsf{fetch2}}}.\mathsf{Appl}$. Both $\mathsf{Block1}$ and $\mathsf{Block2}$, are continuously able to serve the application or to allow the external DMA controller to access the memory via channel dma, with the former having priority over the latter; whence, $\mathsf{Block1} \stackrel{\text{def}}{=} \underline{\mathsf{fetch1}}.\mathsf{Block1} + \mathsf{dma}.\mathsf{Block1}$ and $\mathsf{Block2} \stackrel{\text{def}}{=} \underline{\mathsf{fetch2}}.\mathsf{Block2} + \mathsf{dma}.\mathsf{Block2}$. The overall system Sys is given by:

$$\mathsf{Sys} \stackrel{\text{def}}{=} (\mathsf{Appl} \mid \mathsf{Block1} \mid \mathsf{Block2}) \setminus \{\underline{\mathsf{fetch1}}, \underline{\mathsf{fetch2}}\}$$

Since the application uses the memory blocks alternately, the DMA is expected to be allowed to access the one memory block which is currently not serving the application. However, when using the approach to priority involving global pre-emption presented in Section **??**, all dma-transitions in the labeled transition system of Sys are pre-empted, since the application can indefinitely engage in a prioritized communication, i.e., direct-memory-access is never granted.

Generally speaking, one would expect that priorities at different *sites* of a distributed system do not influence the behavior of each other, i.e., priorities at different sites are supposed to be incomparable. The semantics given in Section **??** does not permit this distinction to be made. Hence, it precludes too many transitions that one would expect to find in a distributed setting. In the following, we show how to remedy the shortcoming of the semantics of Section **??** for distributed systems. Our elaboration is based on the concept of *local pre-emption* [**?**,**?**].

The remainder of this section is organized as follows. Section **??** introduces a notion of locations that is used in Section **??** in the definition of the operational semantics of $\mathsf{CCS}^{\mathsf{sl}}$ for a two-level priority scheme. Sections **??** and **??** develop the semantic theories based on strong and weak bisimulation, respectively, while Section **??** re-considers the direct-memory-access example. The consequences of lifting some design decisions in $\mathsf{CCS}^{\mathsf{sl}}$ are discussed in Section **??**. After extending $\mathsf{CCS}^{\mathsf{sl}}$ to a multi-level priority scheme in Section **??** and presenting another calculus to priority taken from [?] in Section **??**, a formal comparison of the two calculi is given in Section **??**. Finally, Section **??** concludes with some additional remarks and with comments on related work.

4.1 Locations

A location represents the "address" of a subterm inside a larger term; when a system performs an action, $\mathsf{CCS}^{\mathsf{sl}}$ semantics will also note the locations of the subterm that "generate" this action. Because of the potential for synchronization more than one subterm may be involved in an action. Our account of locations closely follows that of [?,?].

Let $\mathcal{A}_{\mathrm{addr}} =_{\mathrm{df}} \{L, R, l, r\}$ be the *address alphabet*, and let $\bullet$ be a special symbol not in $\mathcal{A}_{\mathrm{addr}}$. Then $Addr =_{\mathrm{df}} \{\bullet s \,|\, s \in \mathcal{A}^*_{\mathrm{addr}}\}$ represents the set of (process) *addresses* ranged over by v and w. Intuitively, an element of $Addr$ 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 process $(a.\mathbf{0} \,|\, b.\mathbf{0}) + c.\mathbf{0}$ the address of $a.\mathbf{0}$ is $\bullet Ll$, of $b.\mathbf{0}$ is $\bullet Rl$, and of $c.\mathbf{0}$ is $\bullet r$. 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 $\mathcal{A}^*_{\mathrm{addr}}$). Further, if $V \subseteq Addr$ and $\zeta \in \mathcal{A}_{\mathrm{addr}}$, then we write $V \cdot \zeta$ for $\{v \cdot \zeta \,|\, v \in V\}$. Occasionally, we omit $\bullet$ from addresses.

As mentioned previously, we adopt the view that processes at different sides of the parallel composition operator are logically — not necessarily physically — executed on different processors. Thus, priorities on different sides of the parallel composition operator are distributed and, therefore, should be incomparable. However, priorities on different sides of the summation operator should be comparable since argument processes of summation are logically scheduled on the same processor. This intuition is formalized in the following *location comparability relation* on addresses which is adapted from [?].

Definition 4.1 (Location Comparability Relation)
The location comparability relation $\bowtie$ on addresses is the smallest reflexive and symmetric subset of $Addr \times Addr$ such that for all $v, w \in Addr$:

(1) $\langle v \cdot l, w \cdot r \rangle \in \bowtie$, *and*

Table 11
Distributed prioritized initial action sets for $\mathsf{CCS^{sl}}$

$\underline{\mathcal{I}}_m(\mu x.P)$	$= \underline{\mathcal{I}}_m(P[\mu x.P/x])$	$\underline{\mathcal{I}}_\bullet(\underline{\alpha}.P)$	$= \{\underline{\alpha}\}$
$\underline{\mathcal{I}}_{m\cdot l}(P+Q)$	$= \underline{\mathcal{I}}_m(P)$	$\underline{\mathcal{I}}_{n\cdot r}(P+Q)$	$= \underline{\mathcal{I}}_n(Q)$
$\underline{\mathcal{I}}_m(P[f])$	$= \{f(\underline{\alpha}) \mid \underline{\alpha} \in \underline{\mathcal{I}}_m(P)\}$	$\underline{\mathcal{I}}_m(P \setminus L)$	$= \underline{\mathcal{I}}_m(P) \setminus (L \cup \overline{L})$
$\underline{\mathcal{I}}_{m\cdot L}(P\vert Q)$	$= \underline{\mathcal{I}}_m(P)$	$\underline{\mathcal{I}}_{n\cdot R}(P\vert Q)$	$= \underline{\mathcal{I}}_n(Q)$
$\underline{\mathcal{I}}_{\langle m\cdot L, n\cdot R\rangle}(P\vert Q)$	$= \{\underline{\tau} \mid \underline{\mathcal{I}}_m(P) \cap \overline{\underline{\mathcal{I}}_n(Q)} \neq \emptyset\}$		

(2) $\langle v, w\rangle \in \bowtie$ *implies* $\langle v\cdot\zeta, w\cdot\zeta\rangle \in \bowtie$, *for* $\zeta \in \mathcal{A}_{addr}$.

In the sequel, we write $v \bowtie w$ instead of $\langle v, w\rangle \in \bowtie$. If $v \in Addr$ then we use $[v]$ to denote the set $\{w \in Addr \mid v \bowtie w\}$. Note that the location comparability relation is not transitive, e.g., we have $Ll \bowtie r$ and $r \bowtie Rl$, but $Ll \not\bowtie Rl$, since $L \not\bowtie R$.

We now define the set $\mathcal{L}oc$ of (transition) *locations* as $Addr \cup (Addr \times Addr)$. Intuitively, a transition location records the addresses of the components in a term that participate in the execution of a given action. In our language, 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 $v, w \in Addr$ and $\zeta \in \mathcal{A}_{\mathrm{addr}}$. We use $m, n, o, \ldots$ to range over $\mathcal{L}oc$ in what follows.

4.2 Operational Semantics

The operational semantics of a $\mathsf{CCS^{sl}}$ process P is given by a labeled transition system $\langle \mathcal{P}, \underline{A} \cup (\mathcal{L}oc \times A), \longrightarrow, P\rangle$. The transition relation $\longrightarrow \subseteq \mathcal{P} \times (\mathcal{L}oc \times A)) \times \mathcal{P}$ with respect to unprioritized actions is defined in Table **??** using Plotkin-style operational rules [**?**] whereas for prioritized actions the same rules as for $\mathsf{CCS^{sg}}$ apply (cf. Table **??**). We write $P \xrightarrow{m,\alpha} P'$ if $\langle P, \langle m, \alpha\rangle, P'\rangle \in \longrightarrow$ and $\alpha \in A$, and say that P *may engage in action* α *offered from location* m *and thereafter behave like process* P'.

The presentation of the operational rules requires *distributed prioritized initial action sets* which are defined as the least sets satisfying the rules in Table **??**. Intuitively, $\underline{\mathcal{I}}_m(P)$ 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 action. $\underline{\mathcal{I}}_m(P) = \emptyset$ 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 $\bigcup\{\underline{\mathcal{I}}_m(P) \mid m \in M\}$ of all distributed prioritized initial actions of P from

Table 12
Operational semantics for $\mathsf{CCS}^{\mathsf{sl}}$

$$\textsf{Act}\quad \frac{-}{\alpha.P \xrightarrow{\bullet,\alpha} P} \qquad\qquad \textsf{Sum1}\quad \frac{P \xrightarrow{m,\alpha} P'}{P+Q \xrightarrow{m\cdot l,\alpha} P'}\ \underline{\tau} \notin \underline{\mathcal{I}}(Q)$$

$$\textsf{Rel}\quad \frac{P \xrightarrow{m,\alpha} P'}{P[f] \xrightarrow{m,f(\alpha)} P'[f]} \qquad\qquad \textsf{Sum2}\quad \frac{Q \xrightarrow{n,\alpha} Q'}{P+Q \xrightarrow{n\cdot r,\alpha} Q'}\ \underline{\tau} \notin \underline{\mathcal{I}}(P)$$

$$\textsf{Res}\quad \frac{P \xrightarrow{m,\alpha} P'}{P \setminus L \xrightarrow{m,\alpha} P' \setminus L}\ \alpha \notin L \cup \overline{L} \qquad \textsf{Com1}\quad \frac{P \xrightarrow{m,\alpha} P'}{P|Q \xrightarrow{m\cdot L,\alpha} P'|Q}\ \underline{\mathit{II}}_{[m]}(P) \cap \overline{\underline{\mathit{II}}(Q)} = \emptyset$$

$$\textsf{Rec}\quad \frac{P[\mu x.P/x] \xrightarrow{m,\alpha} P'}{\mu x.P \xrightarrow{m,\alpha} P'} \qquad\qquad \textsf{Com2}\quad \frac{Q \xrightarrow{n,\alpha} Q'}{P|Q \xrightarrow{n\cdot R,\alpha} P|Q'}\ \underline{\mathit{II}}_{[n]}(Q) \cap \overline{\underline{\mathit{II}}(P)} = \emptyset$$

$$\textsf{Com3}\quad \frac{P \xrightarrow{m,\underline{a}} P' \quad Q \xrightarrow{n,\overline{\underline{a}}} Q'}{P|Q \xrightarrow{\langle m\cdot L, n\cdot R\rangle,\underline{\tau}} P'|Q'}\ \begin{array}{l} \underline{\mathit{II}}_{[m]}(P) \cap \overline{\underline{\mathit{II}}(Q)} = \emptyset \,\wedge \\ \underline{\mathit{II}}_{[n]}(Q) \cap \overline{\underline{\mathit{II}}(P)} = \emptyset \end{array}$$

locations $M \subseteq \mathcal{L}oc$ by $\underline{\mathcal{I}}_M(P)$ and the set $\underline{\mathcal{I}}_{\mathcal{L}oc}(P)$ of all distributed prioritized initial actions of P by $\underline{\mathcal{I}}(P)$. We also define analogous sets restricted to visible actions: $\underline{\mathit{II}}_M(P) =_{\text{df}} \underline{\mathcal{I}}_M(P) \setminus \{\underline{\tau}\}$ and $\underline{\mathit{II}}(P) =_{\text{df}} \underline{\mathcal{I}}(P) \setminus \{\underline{\tau}\}$, respectively.

The side conditions of the operational rules guarantee that a process does not perform an unprioritized action if it can engage in a prioritized synchronization or in an internal computation, i.e., a $\underline{\tau}$-transition, from a comparable location. In contrast to the global notion of pre-emption defined in Section **??**, the local notion here is much weaker since $\underline{\mathit{II}}_{[m]}(P) \subseteq \underline{\mathit{II}}(P)$, for all $m \in \mathcal{L}oc$ and $P \in \mathcal{P}$. In other words, local pre-emption does not pre-empt as many transitions as global pre-emption does. The difference between $\mathsf{CCS}^{\mathsf{sl}}$ and $\mathsf{CCS}^{\mathsf{sg}}$ semantics manifests itself in the side conditions of the rules for parallel composition with respect to unprioritized transitions. Since locations on different sides of parallel composition $P|Q$ are incomparable, $\underline{\tau}$'s arising from a location of P cannot pre-empt the execution of a transition, even an unprioritized one, of Q. Only if P engages in a prioritized synchronization with Q can unprioritized actions from a comparable location of P be pre-empted.

4.3 Semantic Theory Based on Strong Bisimulation

As in Section **??**, we present an equivalence relation for $\mathsf{CCS}^{\mathsf{sl}}$ processes based on bisimulation [**?**]. Our aim is to characterize the largest congruence contained in the "naive" adaptation of strong bisimulation [?], which we refer to as *naive distributed prioritized strong bisimulation*, to our framework obtained by ignoring location information. The definition of naive distributed prioritized strong bisimulation, $\simeq$, is identical to the one presented in Definition **??**, except that our transition relation $\xrightarrow{\gamma}$ is defined as mentioned above.

Although $\simeq$ is an equivalence, it is unfortunately — in contrast to Section **??** — not a congruence. The lack of compositionality is demonstrated by the following example, which embodies the traditional view that "*parallelism = nondeterminism.*" We have $a.\underline{b}.\mathbf{0}+\underline{b}.a.\mathbf{0} \simeq a.\mathbf{0} \,|\, \underline{b}.\mathbf{0}$ but $(a.\underline{b}.\mathbf{0}+\underline{b}.a.\mathbf{0}) \,|\, \overline{\underline{b}}.\mathbf{0} \not\simeq (a.\mathbf{0} \,|\, \underline{b}.\mathbf{0}) \,|\, \overline{\underline{b}}.\mathbf{0}$; the latter process can perform an a-transition, while the corresponding a-transition of the former 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. However, we know by Proposition **??** that $\simeq$ includes a largest congruence $\simeq^+$ for $\mathsf{CCS}^{\mathsf{sl}}$.

4.3.1 Distributed Prioritized Strong Bisimulation

To develop a characterization of $\simeq^+$ we need to take *local* pre-emption into account.

Definition 4.2 *A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a* distributed 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.*

(1) $P \xrightarrow{\underline{\alpha}} P'$ implies $\exists Q'.\, Q \xrightarrow{\underline{\alpha}} Q'$ and $\langle P', Q'\rangle \in \mathcal{R}$.
(2) $P \xrightarrow{m,\alpha} P'$ implies
$\exists Q', n.\, Q \xrightarrow{n,\alpha} Q'\,, \underline{\mathcal{I}\!\mathcal{I}}_{[n]}(Q) \subseteq \underline{\mathcal{I}\!\mathcal{I}}_{[m]}(P)\,,$ *and $\langle P', Q'\rangle \in \mathcal{R}$.*

We write $P \simeq^{\mathrm{l}} Q$ if $\langle P, Q\rangle \in \mathcal{R}$ for some distributed prioritized strong bisimulation $\mathcal{R}$.

Intuitively, the distributed 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 Definition **??** states that an unprioritized action α from some location m of 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. The next theorem is the main result of this section.

Theorem 4.3 $\simeq^{\mathrm{l}}$ *is the* largest congruence *contained in* $\simeq$*.*

We refer to [?] for the proof; the context needed in the largest congruence proof is similar to the one used in Section **??**.

Table 13
Axiomatization of $\simeq^{\mathsf{l}}$ (Axioms E)

(iA1)	$t \oplus u = u \oplus t$	(iA2)	$t \oplus (u \oplus v) = (t \oplus u) \oplus v$
(iA3)	$t \oplus t = t$	(iA4)	$t \oplus \mathbf{0} = t$

(E)
$$\begin{aligned} & t \equiv \textstyle\bigoplus_i \sum_j \gamma_{ij}.t_{ij} \text{ and } u \equiv \bigoplus_k \sum_l \delta_{kl}.u_{kl} \quad \text{implies} \quad t \,|\, u = \\ & \textstyle\bigoplus_i \sum_j (\gamma_{ij}.(t_{ij} \,|\, u) + \sum_k \sum_l \{\tau.(t_{ij} \,|\, u_{kl}) \,|\, \gamma_{ij} \equiv \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in A\} \\ & \qquad\qquad\qquad \textstyle + \sum_k \sum_l \{\underline{\tau}.(t_{ij} \,|\, u_{kl}) \,|\, \gamma_{ij} \equiv \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in \underline{A}\}) \oplus \\ & \textstyle\bigoplus_k \sum_l (\delta_{kl}.(t \,|\, u_{kl}) + \sum_i \sum_j \{\tau.(t_{ij} \,|\, u_{kl}) \,|\, \gamma_{ij} \equiv \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in A\} \\ & \qquad\qquad\qquad \textstyle + \sum_i \sum_j \{\underline{\tau}.(t_{ij} \,|\, u_{kl}) \,|\, \gamma_{ij} \equiv \overline{\delta}_{kl},\ \gamma_{ij}, \delta_{kl} \in \underline{A}\}) \end{aligned}$$

(iRes4)	$(t \oplus u) \setminus L = (t \setminus L) \oplus (u \setminus L)$	(iRel3)	$(t \oplus u)[f] = t[f] \oplus u[f]$

4.3.2 Axiomatic Characterization

In this section we present an axiomatization of $\simeq^{\mathsf{l}}$ for *finite* processes, for which we introduce a new binary summation operator $\oplus$ to the process algebra $\mathsf{CCS}^{\mathsf{sl}}$. This operator is called *distributed summation operator* and is needed for giving an *Expansion Axiom* (cf. Axiom (E) in Table **??**). Its operational semantics is defined below and differs form the nondeterministic choice operator $+$ in that a location in its left argument is never comparable to one in its right argument.

$$\underline{\mathsf{dSum1}}\ \frac{t \xrightarrow{\underline{\alpha}} t'}{t \oplus u \xrightarrow{\underline{\alpha}} t'} \qquad \mathsf{dSum1}\ \frac{t \xrightarrow{m,\alpha} t'}{t \oplus u \xrightarrow{m \cdot L, \alpha} t'}$$

$$\underline{\mathsf{dSum2}}\ \frac{u \xrightarrow{\underline{\alpha}} u'}{t \oplus u \xrightarrow{\underline{\alpha}} u'} \qquad \mathsf{dSum2}\ \frac{u \xrightarrow{n,\alpha} u'}{t \oplus u \xrightarrow{n \cdot R, \alpha} u'}$$

It can easily be checked that $\simeq^{\mathsf{l}}$ is compositional with respect to $\oplus$.

Now, we turn to the axiom system for distributed prioritized strong bisimulation. We write $\vdash_E t = u$ if term t can be rewritten to u using the axioms in Tables **??** and **??** as well as Axioms (A1)–(A4), Axioms (Res1)–(Res4), Axioms (Rel1)–(Rel3), and Axiom (P) from Table **??**. Axioms (Ic1), (D1), (S2), and (S3) involve side conditions. Regarding Axiom (Ic1), we introduce the unary predicate $\natural$ over processes (of the form $\sum_{j \in J} \gamma_j.t_j$, for some nonempty index set J) together with the following proof rules: (i) $\natural\underline{\alpha}.t$ and (ii) $\natural t$ and $\natural u$ implies $\natural(t+u)$. Intuitively, $\natural(\sum_{j \in J} \gamma_j.t_j)$ if and only if $\gamma_j \in \underline{A}$ for all $j \in J$. The relation $\sqsubseteq_{\mathsf{i}}$ is defined as in Section **??** (cf. Table **??**). The axioms in Table **??**

Table 14
Axioms E (continued)

(D1)	$(t \oplus t') + (u \oplus u') = ((t \oplus t') + u') \oplus ((u \oplus u') + t')$	
	$(\vdash_I t \sqsubseteq_{\mathrm{i}} t'\,,\ \vdash_I u \sqsubseteq_{\mathrm{i}} u')$	
(D2)	$(t \oplus u) + \underline{\alpha}.v = (t + \underline{\alpha}.v) \oplus (u + \underline{\alpha}.v)$	
(Ic1)	$t \oplus \underline{\alpha}.u = t + \underline{\alpha}.u$	$(\natural t)$
(Ic2)	$(\underline{\alpha}.t + u) = (\underline{\alpha}.t + u) \oplus \underline{\alpha}.t$	
(S1)	$(t + \underline{\alpha}.u) \oplus (t' + \underline{\alpha}.u') = (t + \underline{\alpha}.u + \underline{\alpha}.u') \oplus (t' + \underline{\alpha}.u')$	
(S2)	$(t + \alpha.v) \oplus (u + \alpha.v) = (t + \alpha.v) \oplus u$	$(\vdash_I t \sqsubseteq_{\mathrm{i}} u)$
(S3)	$t \oplus u = t + u$	$(\vdash_I t =_{\mathrm{i}} u)$

are those given in Table **??** 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 $\oplus$). Note that parallelism in $\mathsf{CCS}^{\mathsf{sl}}$ cannot be resolved in nondeterminism by using operator $+$ only, since priorities on different sides of $|$ are incomparable, but priorities on different sides of $+$ are comparable. This is the motivation for introducing operator $\oplus$. The axioms in Table **??** are new and show how we may "restructure" locations. They deal with the *distributivity* of the summation operators (Axioms (D1) and (D2)), the *interchangeability* of the summation operators (Axioms (Ic1) and (Ic2)), and the *saturation* of locations (Axioms (S1), (S2), and (S3)), respectively.

Theorem 4.4 *Let t, u be finite processes. Then $\vdash_E t = u$ if and only if $t \simeq^{\mathrm{l}} u$.*

The proof of this theorem can be found in [**?**,**?**].

4.3.3 Operational Characterization

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

Definition 4.5 *A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is an* alternative distributed prioritized strong bisimulation *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.*

(1) $P \xrightarrow{\alpha} P'$ implies $\exists Q'.\, Q \xrightarrow{\alpha} Q'$ and $\langle P', Q' \rangle \in \mathcal{R}$.

(2) $P \xrightarrow[L]{\alpha} P'$ *implies* $\exists Q'.\, Q \xrightarrow[L]{\alpha} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$.

We write $P \simeq_* Q$ *if* $\langle P, Q \rangle \in \mathcal{R}$ *for some alternative distributed prioritized strong bisimulation* $\mathcal{R}$.

Similar to Section **??** we obtain an operational characterization of our behavioral relation.

Theorem 4.6 (Operational Characterization) $\simeq^{\mathsf{l}} = \simeq_*$.

4.4 Semantic Theory Based on Weak Bisimulation

As for $\mathsf{CCS}^{\mathsf{sg}}$, we develop a coarser bisimulation-based congruence by abstracting from internal actions. We start off with the definition of a *naive distributed prioritized weak bisimulation* which is an adaptation of observational equivalence [**?**].

Definition 4.7 (Naive Distributed Prioritized Weak Transitions)

(1) $\hat{\gamma} =_{df} \epsilon$, *if* $\gamma \in \{\underline{\tau}, \tau\}$, *and* $\hat{\gamma} =_{df} \gamma$, *otherwise*
(2) $\stackrel{\epsilon}{\Longrightarrow}_\times \,=_{df} (\stackrel{\underline{\tau}}{\longrightarrow} \cup \bigcup\{\stackrel{m,\tau}{\longrightarrow} \mid m \in \mathcal{L}oc\})^*$
(3) $\stackrel{\underline{\alpha}}{\Longrightarrow}_\times \,=_{df} \stackrel{\epsilon}{\Longrightarrow}_\times \circ \stackrel{\underline{\alpha}}{\longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}_\times$
(4) $\stackrel{m,\alpha}{\Longrightarrow}_\times \,=_{df} \stackrel{\epsilon}{\Longrightarrow}_\times \circ \stackrel{m,\alpha}{\longrightarrow} \circ \stackrel{\epsilon}{\Longrightarrow}_\times$

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

Definition 4.8 (Naive Distributed Prioritized Weak Bisimulation) *A symmetric relation* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is a* naive distributed 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* $\langle P, Q \rangle \in \mathcal{R}$ *for some naive distributed prioritized weak bisimulation* $\mathcal{R}$.

It is fairly easy to see that $\approx_\times$ is not a congruence for $\mathsf{CCS}^{\mathsf{sl}}$. One compositionality defect arises with respect to parallel composition and is similar to the one mentioned for naive distributed prioritized strong bisimulation. Another defect, which is carried over from CCS, is concerned with the summation operators.

4.4.1 Distributed Prioritized Weak Bisimulation

We devote the rest of Section **??** to operationally characterizing the largest congruence contained in naive distributed prioritized weak bisimulation. To do so, we first re-define the weak transition relation.

Definition 4.9 (Distributed Prioritized Weak Transitions)
For $L, M \subseteq \underline{\mathcal{A}} \setminus \{\underline{\tau}\}$ we define the following notations.

(1) $\hat{\underline{\tau}} =_{df} \underline{\epsilon}$, $\hat{\underline{a}} =_{df} \underline{a}$, $\hat{\tau} =_{df} \epsilon$, *and* $\hat{a} =_{df} a$
(2) $P \overset{m,\alpha}{\underset{L}{\rightarrow}} P'$ *if* $P \xrightarrow{m,\alpha} P'$ *and* $\underline{\mathcal{II}}_{[m]}(P) \subseteq L$
(3) $\overset{\underline{\epsilon}}{\Longrightarrow} =_{df} (\xrightarrow{\underline{\tau}} \cup \bigcup\{\overset{m,\tau}{\underset{\emptyset}{\rightarrow}} \mid m \in \mathcal{L}oc\})^*$
(4) $\overset{\underline{\alpha}}{\Longrightarrow} =_{df} \overset{\underline{\epsilon}}{\Longrightarrow} \circ \xrightarrow{\underline{\alpha}} \circ \overset{\underline{\epsilon}}{\Longrightarrow}$
(5) $\overset{\epsilon}{\underset{L}{\Rightarrow}} =_{df} (\xrightarrow{\underline{\tau}} \cup \bigcup\{\overset{m,\tau}{\underset{L}{\rightarrow}} \mid m \in \mathcal{L}oc\})^*$
(6) $P \overset{m,\alpha}{\underset{L,M}{\Rightarrow}} P'$ *if* $\exists P'', P'''.\, P \overset{\epsilon}{\underset{L}{\Rightarrow}} P'' \overset{m,\alpha}{\underset{L}{\rightarrow}} P''' \overset{\underline{\epsilon}}{\Longrightarrow} P'$ *and* $\underline{\mathcal{II}}(P'') \subseteq M$.

$P \overset{m,\alpha}{\underset{L}{\rightarrow}} P'$ means that P can engage in action α at location m to P' *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 location comparable to m in P, which would pre-empt the α. Similarly, $P \overset{\underline{\epsilon}}{\Longrightarrow} P'$ 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, $P \overset{\epsilon}{\underset{L}{\Rightarrow}} P'$ means that, so long as the environment does not offer to synchronize with P on a prioritized action in L, process P may engage in a sequence of internal computation steps and become P'. Finally, the M-parameter in $\overset{m,\alpha}{\underset{L,M}{\Rightarrow}}$ provides a measure of the pre-emptive impact that a process has on its environment. $P \overset{m,\alpha}{\underset{L,M}{\Rightarrow}} P'$ 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 engage in 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.

Note that the definition of $P \overset{\epsilon}{\underset{L}{\Rightarrow}} P'$ is in accordance with our intuition that internal actions, and therefore their locations, are unobservable. Moreover, the environment of P is not influenced by internal actions performed by P, since priorities arising from different sides of the parallel composition operator are incomparable. Therefore, the parameter M is unnecessary in the definition of the relation $\overset{\epsilon}{\underset{L}{\Rightarrow}}$. Finally, for notational convenience, $\overset{m,\epsilon}{\underset{L,M}{\Rightarrow}}$ is interpreted as $\overset{\epsilon}{\underset{L}{\Rightarrow}}$.

Definition 4.10 (Distributed Prioritized Weak Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is a distributed 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.

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

We write $P \underline{\approx} Q$ *if* $\langle P, Q \rangle \in \mathcal{R}$ *for some distributed prioritized weak bisimulation* $\mathcal{R}$.

Condition (**??**) of Definition **??** guarantees that distributed prioritized weak bisimulation is compositional with respect to parallel composition. 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 Condition (**??**) were absent. However, the context $C[X] \stackrel{\text{def}}{=} X \,|(\underline{\overline{a}}.\mathbf{0} + b.\mathbf{0})$ distinguishes them. The following proposition is the $\mathsf{CCS}^{\mathsf{sl}}$ equivalent of Proposition **??**.

Proposition 4.11 *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 sub-algebra of* $\mathsf{CCS}^{\mathsf{sl}}$ *induced by these operators and recursion.*

4.4.2 Distributed Prioritized Observational Congruence

As in Section **??**, the summation fix presented in [**?**] is not sufficient in order to achieve a congruence relation.

Definition 4.12 *We define* $P \underline{\approx}^{\mathrm{l}} Q$ *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.*

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

Now, we can state the main result of this section, which can be proved by using the technique already presented in Section **??**; see [**?**] for details.

Theorem 4.13 $\underline{\approx}^{\mathrm{l}}$ *is the largest congruence contained in* $\approx_\times$.

4.4.3 Operational Characterization

We now characterize distributed prioritized weak bisimulation as standard bisimulation over an appropriately defined transition relation. To begin with, we introduce a family of relations $\underset{M}{\Rightarrow}$ on processes, where $M \subseteq \underline{A} \setminus \{\underline{\tau}\}$, by defining $P \underset{M}{\Rightarrow} P'$ if $\exists P''.\, P \stackrel{\epsilon}{\Longrightarrow} P'' \stackrel{\epsilon}{\Longrightarrow} P'$ and $\underline{I\!I}(P'') \subseteq M$. Moreover, we write $P \underset{L,M}{\overset{\hat{\alpha}}{\Rightarrow}} P'$ whenever there exists some $m \in \mathcal{L}oc$ such that $P \underset{L,M}{\overset{m,\hat{\alpha}}{\Rightarrow}} P'$.

Definition 4.14 *A symmetric relation* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is an* alternative distributed 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.*

(1) $P \underset{M}{\Rightarrow} P'$ *implies* $\exists Q'.\, Q \underset{M}{\Rightarrow} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$*.*
(2) $P \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} P'$ *implies* $\exists Q'.\, Q \stackrel{\hat{\underline{\alpha}}}{\Longrightarrow} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$*.*
(3) $P \underset{L,M}{\overset{\hat{\alpha}}{\Rightarrow}} P'$ *implies* $\exists Q'.\, Q \underset{L,M}{\overset{\hat{\alpha}}{\Rightarrow}} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$*.*

We write $P \underline{\approx}_* Q$ *if* $\langle P, Q \rangle \in \mathcal{R}$ *for some alternative distributed prioritized weak bisimulation* $\mathcal{R}$*.*

Theorem 4.15 (Operational Characterization) $\underline{\approx} = \underline{\approx}_*$.

The interested reader can find the proof of this theorem in [**?**].

4.5 Example

We now return to the direct-memory-access example system introduced above. The $\mathsf{CCS}^{\mathsf{sl}}$ semantics of Sys, which corresponds to our intuition regarding distributed systems, is given in Figure **??** where we leave out the locations in the transitions' labels.

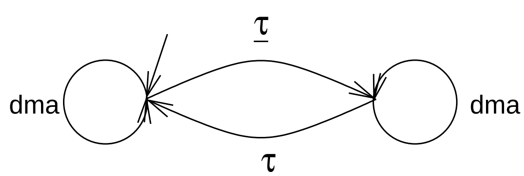

Fig. 5. Semantics of the dma-system Sys

Recall that the application uses the two memory blocks alternately. Thus, the DMA is expected to be allowed to access the free memory block. Accordingly, the specification of the system can be formalized by $\mathsf{Spec} \stackrel{\text{def}}{=} \mathsf{dma.Spec}$. It is easy to see that the symmetric closure of

$$\{\langle \mathsf{Spec}, \mathsf{Sys} \rangle, \langle \mathsf{Spec}, (\overline{\underline{\mathsf{fetch2}}}.\mathsf{Appl} \mid \mathsf{Block1} \mid \mathsf{Block2}) \setminus \{\underline{\mathsf{fetch1}}, \underline{\mathsf{fetch2}}\} \rangle\}$$

is a distributed prioritized weak bisimulation. Therefore, $\mathsf{Spec} \underline{\approx} \mathsf{Sys}$ as expected, i.e., system Sys meets its specification Spec.

Table 15
Modified operational rules

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

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

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

4.6 Discussion of Alternative Design Decisions

Up to now we have restricted the number of priority levels in $\mathsf{CCS}^{\mathsf{sl}}$ to two and communication to complementary actions having the same priority. In this section we study the implications of the removal of these restrictions, which leads to a new version of $\mathsf{CCS}^{\mathsf{sl}}$, called $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ ($\mathsf{CCS}^{\mathsf{sl}}$ with a $\mathsf{m}$ulti-$\mathsf{l}$evel priority scheme), that is formally defined in the next section.

Allowing communication between unprioritized actions and complementary prioritized actions raises the question of whether the resulting internal action should be τ or $\underline{\tau}$ (cf. Section **??**). 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 $(\underline{a}.\mathbf{0} \,|\, \overline{a}.\mathbf{0}) + b.\mathbf{0}$ 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., it cannot engage in a communication. This implies that the *minimal* priority of the complementary actions ought to be assigned to the internal action. To reflect this in the operational semantics, one could replace Rules ($\mathsf{Com1}$), ($\mathsf{Com2}$), and ($\mathsf{Com3}$) for parallel composition by the ones presented in Table **??** plus their symmetric counterparts. The side conditions involve sets $\mathit{\Pi}(P)$ that include all unprioritized visible actions in which P can initially engage.

It turns out that the largest congruence results concerning distributed prioritized strong bisimulation and distributed prioritized observational congruence 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 process $(b.\mathbf{0} + \underline{a}.\mathbf{0}) \,|(\overline{a}.\mathbf{0} + \underline{c}.\mathbf{0}) \,|\, \overline{\underline{c}}.\mathbf{0}$. In the left-associative case, the initial b-transition is pre-empted, according to Rule ($\mathsf{Com1}$) since $\underline{a}$ may potentially communicate with $\overline{a}$. In the right-associative case, on the other hand, the $\overline{a}$-transition in the second component is pre-empted, and consequently the b-transition is enabled by Rule ($\mathsf{Com1}$). The reason for this problem is that transitions are pre-empted because a 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 $\mathsf{CCS}^{\mathsf{sl}}$ to multiple priority levels, even if communication is only allowed on complementary actions of the same priority, as can be observed by using a slight adaptation of the previous example: $(b\!:\!2.\mathbf{0} + a\!:\!1.\mathbf{0}) \,|\, (\overline{a}\!:\!1.\mathbf{0} + c\!:\!0.\mathbf{0}) \,|\, \overline{c}\!:\!0.\mathbf{0}$.

One can imagine two approaches to fixing the problems with the first (and second) alteration to the theory. One is to change the operational semantics; in particular, the side conditions could be weakened such that an unprioritized transition is only pre-empted when a prioritized action from a comparable location can *actually* engage in a communication. This approach has not yet been investigated in the literature. The second solution follows an approach developed in [?] 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 that are in the scope of $+$. Hence, all potential communication partners are also actual ones, and the standard 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` [?].

4.7 Extension to Multi-level Priority Schemes

For $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ we allow a multi-level priority scheme and communication between complementary actions having potentially different priorities. As seen in the previous section, both of these relaxations yield a semantics for which parallel composition is not associative. However, we have also argued that this problem vanishes if the syntax is restricted such that output actions can never be pre-empted. We adapt the syntax restriction proposed by Camilleri and Winskel [?], which states that initial actions in the scope of a summation operator must be input actions. Therefore, input and output actions are explicitly distinguished in $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$, with the internal action τ being treated as an input action. In the following, we let $a, b, \ldots$ range over the set Λ of input ports and $\overline{a}, \overline{b}, \ldots$ over the set $\overline{\Lambda}$ of output ports. Moreover, we let γ stand for any input action and let α range over $\mathcal{A} =_{\mathrm{df}} \Lambda \cup \overline{\Lambda} \cup \{\tau\}$. Since in the restricted syntax priority values of output actions need never be compared with other priority values, there are no priority values associated with output actions at all. The syntax of $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ is formally defined by the following BNF for P.

$$I \quad ::= \quad \mathbf{0} \mid x \mid \gamma\!:\!k.I \mid I+I \mid I \oplus I \mid I \,|\, I \mid I[f] \mid I \setminus L \mid \mu x.I$$

$$P \quad ::= \quad \mathbf{0} \mid x \mid \alpha\!:\!k.P \mid I+I \mid P \oplus P \mid P \,|\, P \mid P[f] \mid P \setminus L \mid \mu x.P$$

Here, f is an *injective*, finite relabeling, $L \subseteq \Lambda \cup \overline{\Lambda}$ is a restriction set, and x is a variable taken from a countable domain $\mathcal{V}$. A relabeling satisfies the properties $f(\Lambda) \subseteq \Lambda$, $f(\overline{\Lambda}) \subseteq \overline{\Lambda}$, $f(\tau) = \tau$, and $f(\overline{a}) = \overline{f(a)}$. Thus, in addition to the requirements of a finite relabeling in CCS, relabelings in $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ may only map input ports to input ports and output ports to output ports. Since actions attached with different priority values do not represent different ports here, relabelings and restriction sets do not deal with priority values. Especially, the priority value of a relabeled transition remains the same, i.e., there is no explicit or implicit mechanism for prioritization or deprioritization (cf. Section **??**). In the remainder, we let $\mathcal{P}^{\mathsf{sl}}_{\mathsf{ml}}$ denote the set of all $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ processes.

Table 16
Initial output action sets for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$

$\overline{\mathbb{I}}(\mu x.P) = \overline{\mathbb{I}}(P[\mu x.P/x])$	$\overline{\mathbb{I}}(\overline{a}.P) = \{a\}$
$\overline{\mathbb{I}}(P \,|\, Q) = \overline{\mathbb{I}}(P) \cup \overline{\mathbb{I}}(Q)$	$\overline{\mathbb{I}}(P \oplus Q) = \overline{\mathbb{I}}(P) \cup \overline{\mathbb{I}}(Q)$
$\overline{\mathbb{I}}(P[f]) = \{f(a) \,|\, a \in \overline{\mathbb{I}}(P)\}$	$\overline{\mathbb{I}}(P \setminus L) = \overline{\mathbb{I}}(P) \setminus (L \cup \overline{L})$

The semantics of $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ processes are again labeled transition systems whose transition relations are specified by operational rules. Since transitions labeled by output actions are never pre-empted they do not need to be associated with locations. We first present two auxiliary sets that are useful for presenting the operational rules: (i) initial output action sets $\overline{\mathbb{I}}(P)$ of a process P and (ii) initial input action sets $\mathrm{I}^k_m(P)$ of P with respect to a priority value k and a location m, which are defined to be the least sets satisfying the equations presented in Tables **??** and **??**, respectively. For technical convenience, we remove the complement of output actions in the definition of $\overline{\mathbb{I}}(\cdot)$, and we use the following four abbreviations: (i) $\mathrm{I}^{<k}_M(P) =_{\mathrm{df}} \bigcup\{\mathrm{I}^l_m(P) \,|\, m \in M,\ l < k\}$, (ii) $\mathbb{I}^{<k}_M(P) =_{\mathrm{df}} \mathrm{I}^{<k}_M(P) \setminus \{\tau\}$, (iii) $\mathrm{I}(P) =_{\mathrm{df}} \bigcup\{\mathrm{I}^l_m(P) \,|\, m \in \mathcal{L}oc,\ l \in \mathcal{N}\}$, and (iv) $\mathbb{I}(P) =_{\mathrm{df}} \mathrm{I}(P) \setminus \{\tau\}$.

The operational rules for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ semantics are formally stated in Table **??** for *output transitions* and in Table **??** for *input transitions*. As expected, the rules for output transitions coincide with the ones for plain CCS [**?**], whereas the rules for input transitions take local pre-emption into account, thereby using location and priority value information in their side conditions. It is worth taking a closer look at the side conditions of Rules ($\mathsf{Sum1}$) and ($\mathsf{Sum2}$) which differ in principle from the corresponding ones in $\mathsf{CCS}^{\mathsf{sl}}$. They guarantee that an initial $\gamma\!:\!l$-transition of a process P is also pre-empted whenever there

Table 17
Initial input action sets for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$

$\mathrm{I}^k_m(\mu x.P)$	$= \mathrm{I}^k_m(P[\mu x.P/x])$	$\mathrm{I}^k_\bullet(\gamma : l.P)$	$= \{\gamma \mid k = l\}$
$\mathrm{I}^k_{m\cdot l}(P + Q)$	$= \mathrm{I}^k_m(P)$	$\mathrm{I}^k_{m\cdot L}(P \oplus Q)$	$= \mathrm{I}^k_m(P)$
$\mathrm{I}^k_{n\cdot r}(P + Q)$	$= \mathrm{I}^k_n(Q)$	$\mathrm{I}^k_{n\cdot R}(P \oplus Q)$	$= \mathrm{I}^k_n(Q)$
$\mathrm{I}^k_m(P[f])$	$= \{f(\gamma) \mid \gamma \in \mathrm{I}^k_m(P)\}$	$\mathrm{I}^k_{m\cdot L}(P \mid Q)$	$= \mathrm{I}^k_m(P) \cup \{\tau \mid \mathrm{I}^k_m(P) \cap \overline{I\!I}(Q) \neq \emptyset\}$
$\mathrm{I}^k_m(P \setminus L)$	$= \mathrm{I}^k_m(P) \setminus (L \cup \overline{L})$	$\mathrm{I}^k_{n\cdot R}(P \mid Q)$	$= \mathrm{I}^k_n(Q) \cup \{\tau \mid \mathrm{I}^k_n(Q) \cap \overline{I\!I}(P) \neq \emptyset\}$

Table 18
Operational semantics for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ wrt. output transitions

$$\overline{\mathsf{Act}}\ \frac{-}{\overline{a}.P \xrightarrow{\overline{a}} P} \qquad \overline{\mathsf{iSum1}}\ \frac{P \xrightarrow{\overline{a}} P'}{P \oplus Q \xrightarrow{\overline{a}} P'} \qquad \overline{\mathsf{Com1}}\ \frac{P \xrightarrow{\overline{a}} P'}{P \mid Q \xrightarrow{\overline{a}} P' \mid Q}$$

$$\overline{\mathsf{Rel}}\ \frac{P \xrightarrow{\overline{a}} P'}{P[f] \xrightarrow{f(\overline{a})} P'[f]} \qquad \overline{\mathsf{iSum2}}\ \frac{Q \xrightarrow{\overline{a}} Q'}{P \oplus Q \xrightarrow{\overline{a}} Q'} \qquad \overline{\mathsf{Com2}}\ \frac{Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\overline{a}} P \mid Q'}$$

$$\overline{\mathsf{Rec}}\ \frac{P[\mu x.P/x] \xrightarrow{\overline{a}} P'}{\mu x.P \xrightarrow{\overline{a}} P'} \qquad \overline{\mathsf{Res}}\ \frac{P \xrightarrow{\overline{a}} P'}{P \setminus L \xrightarrow{\overline{a}} P' \setminus L}\ \overline{a} \notin L \cup \overline{L}$$

exists a higher prioritized initial $\gamma : k$-transition of P, i.e., if $k < l$. This new kind of pre-emption reflects that output transitions can communicate with a complementary input transition regardless of its priority value; i.e., if more than one communication partner offering the matching input transition is available from a comparable location, then the one having the highest priority is taken. For this notion of pre-emption to be well-defined, relabelings must be restricted to injective ones, as is pointed out in [?].

The behavioral relations defined for $\mathsf{CCS}^{\mathsf{sl}}$ can be adapted to $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ in a straightforward fashion, as we demonstrate by the notion of distributed prioritized strong bisimulation.

Definition 4.16 *A symmetric relation* $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ *is a* distributed prioritized strong bisimulation for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ *if for every* $\langle P, Q \rangle \in \mathcal{R}$, $\overline{a} \in \overline{\Lambda}$, $\gamma \in \Lambda \cup \{\tau\}$, $k \in \mathcal{N}$, *and* $m \in \mathcal{L}oc$ *the following conditions hold.*

(1) $P \xrightarrow{\overline{a}} P'$ *implies* $\exists Q'. Q \xrightarrow{\overline{a}} Q'$ *and* $\langle P', Q' \rangle \in \mathcal{R}$, *and*

Table 19
Operational semantics for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ wrt. input transitions

$$\mathsf{Act}\ \frac{-}{\gamma:k.P \xrightarrow{\bullet,\gamma:k} P} \qquad \mathsf{Sum1}\ \frac{P \xrightarrow{m,\gamma:k} P'}{P+Q \xrightarrow{m\cdot l,\gamma:k} P'}\ \tau,\gamma \notin \mathrm{I}^{<k}(Q)$$

$$\mathsf{iSum1}\ \frac{P \xrightarrow{m,\gamma:k} P'}{P \oplus Q \xrightarrow{m\cdot L,\gamma:k} P'} \qquad \mathsf{Sum2}\ \frac{Q \xrightarrow{n,\gamma:k} Q'}{P+Q \xrightarrow{n\cdot r,\gamma:k} Q'}\ \tau,\gamma \notin \mathrm{I}^{<k}(P)$$

$$\mathsf{iSum2}\ \frac{Q \xrightarrow{n,\gamma:k} Q'}{P \oplus Q \xrightarrow{n\cdot R,\gamma:k} Q'} \qquad \mathsf{Com1}\ \frac{P \xrightarrow{m,\gamma:k} P'}{P\,|\,Q \xrightarrow{m\cdot L,\gamma:k} P'\,|\,Q}\ \mathbb{I}^{<k}_{[m]}(P) \cap \overline{\mathbb{I}}(Q) = \emptyset$$

$$\mathsf{Rel}\ \frac{P \xrightarrow{m,\gamma:k} P'}{P[f] \xrightarrow{m,f(\gamma):k} P'[f]} \qquad \mathsf{Com2}\ \frac{Q \xrightarrow{n,\gamma:k} Q'}{P\,|\,Q \xrightarrow{n\cdot R,\gamma:k} P\,|\,Q'}\ \mathbb{I}^{<k}_{[n]}(Q) \cap \overline{\mathbb{I}}(P) = \emptyset$$

$$\mathsf{Rec}\ \frac{P[\mu x.P/x] \xrightarrow{m,\gamma:k} P'}{\mu x.P \xrightarrow{m,\gamma:k} P'} \qquad \mathsf{Com3}\ \frac{P \xrightarrow{m,a:k} P' \quad Q \xrightarrow{\overline{a}} Q'}{P\,|\,Q \xrightarrow{m\cdot L,\tau:k} P'\,|\,Q'}\ \mathbb{I}^{<k}_{[m]}(P) \cap \overline{\mathbb{I}}(Q) = \emptyset$$

$$\mathsf{Com4}\ \frac{P \xrightarrow{\overline{a}} P' \quad Q \xrightarrow{n,a:k} Q'}{P\,|\,Q \xrightarrow{n\cdot R,\tau:k} P'\,|\,Q'}\ \mathbb{I}^{<k}_{[n]}(Q) \cap \overline{\mathbb{I}}(P) = \emptyset$$

$$\mathsf{Res}\ \frac{P \xrightarrow{m,\gamma:k} P'}{P \setminus L \xrightarrow{m,\gamma:k} P' \setminus L}\ \gamma \notin L \cup \overline{L}$$

(2) $P \xrightarrow{m,\gamma:k} P'$ *implies*
$\exists Q', l, n.\, Q \xrightarrow{n,\gamma:l} Q',\ \mathbb{I}^{<l}_{[n]}(Q) \subseteq \mathbb{I}^{<k}_{[m]}(P),$ *and* $\langle P', Q'\rangle \in \mathcal{R}$.

We write $P \simeq_{\mathsf{ml}} Q$ *if* $\langle P, Q\rangle \in \mathcal{R}$ *for some distributed prioritized strong bisimulation* $\mathcal{R}$ *for* $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$.

Proposition 4.17 *The relation* $\simeq_{\mathsf{ml}}$ *is compositional with respect to all operators except summation.*

The proof is by standard techniques [**?**] and, therefore, is omitted here. The reason for the lack of compositionality with respect to summation is illustrated by the following example: $a:0.\mathbf{0} \simeq_{\mathsf{ml}} a:1.\mathbf{0}$ holds, but $a:0.\mathbf{0}+\tau:0.\mathbf{0} \not\simeq_{\mathsf{ml}} a:1.\mathbf{0}+\tau:0.\mathbf{0}$ since the former process can engage in a transition labeled by action a whereas the latter cannot. This defect can easily be repaired (note the analogy with weak bisimulation [**?**]) but is not important in the remainder.

In this section we briefly review Camilleri and Winskel's approach to priority [?], which we refer to as $\mathsf{CCS^{cw}}$ (CCS with priority due to Camilleri and Winskel). In contrast to the approaches considered so far, this process algebra with priority does not assign priority values to actions. Instead, there exists a special summation operator $+\rangle$, called *prioritized choice*, which favors its left argument over its right one. The syntax of $\mathsf{CCS^{cw}}$ is given by the following BNF for P.

$$I \quad ::= \quad \mathbf{0} \mid x \mid \gamma.I \mid I +\rangle I \mid I + I \mid I \,|\, I \mid I[f] \mid I \setminus L \mid \mu x.I$$

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

Here, the action γ, the injective, finite relabeling f, and the restriction set L satisfy the constraints mentioned in Section **??**. Again, closed and guarded terms determine the set $\mathcal{P}^{\mathsf{cw}}$ of $\mathsf{CCS^{cw}}$ processes. Further, we introduce initial output and input action sets as depicted in Tables **??** and **??**, respectively, and write $\mathbb{I}^{\mathsf{cw}}(P)$ for $\mathrm{I}^{\mathsf{cw}}(P) \setminus \{\tau\}$.

Table 20
Initial output action sets for $\mathsf{CCS^{cw}}$

$\overline{\mathbb{I}}^{\mathsf{cw}}(\overline{a}.P) = \{a\}$	$\overline{\mathbb{I}}^{\mathsf{cw}}(\mu x.P) = \overline{\mathbb{I}}^{\mathsf{cw}}(P[\mu x.P/x])$
$\overline{\mathbb{I}}^{\mathsf{cw}}(P \,\|\, Q) = \overline{\mathbb{I}}^{\mathsf{cw}}(P) \cup \overline{\mathbb{I}}^{\mathsf{cw}}(Q)$	$\overline{\mathbb{I}}^{\mathsf{cw}}(P + Q) = \overline{\mathbb{I}}^{\mathsf{cw}}(P) \cup \overline{\mathbb{I}}^{\mathsf{cw}}(Q)$
$\overline{\mathbb{I}}^{\mathsf{cw}}(P[f]) = \{f(a) \,\|\, a \in \overline{\mathbb{I}}^{\mathsf{cw}}(P)\}$	$\overline{\mathbb{I}}^{\mathsf{cw}}(P \setminus L) = \overline{\mathbb{I}}^{\mathsf{cw}}(P) \setminus (L \cup \overline{L})$

Table 21
Initial input action sets for $\mathsf{CCS^{cw}}$

$\mathrm{I}^{\mathsf{cw}}(\gamma.P) = \{\gamma\}$	$\mathrm{I}^{\mathsf{cw}}(\mu x.P) = \mathrm{I}^{\mathsf{cw}}(P[\mu x.P/x])$
$\mathrm{I}^{\mathsf{cw}}(P +\rangle Q) = \mathrm{I}^{\mathsf{cw}}(P) \cup \mathrm{I}^{\mathsf{cw}}(Q)$	$\mathrm{I}^{\mathsf{cw}}(P + Q) = \mathrm{I}^{\mathsf{cw}}(P) \cup \mathrm{I}^{\mathsf{cw}}(Q)$
$\mathrm{I}^{\mathsf{cw}}(P[f]) = \{f(\gamma) \,\|\, \gamma \in \mathrm{I}^{\mathsf{cw}}(P)\}$	$\mathrm{I}^{\mathsf{cw}}(P \setminus L) = \mathrm{I}^{\mathsf{cw}}(P) \setminus (L \cup \overline{L})$
$\mathrm{I}^{\mathsf{cw}}(P \,\|\, Q) = \mathrm{I}^{\mathsf{cw}}(P) \cup \mathrm{I}^{\mathsf{cw}}(Q) \cup \{\tau \,\|\, \mathrm{I}^{\mathsf{cw}}(P) \cap \overline{\mathbb{I}}^{\mathsf{cw}}(Q) \neq \emptyset\}$	

The semantics of a $\mathsf{CCS^{cw}}$ process is given by a labeled transition system whose transition relation possesses transitions of the form $\vdash^{\mathsf{cw}}_{M} P \stackrel{\alpha}{\to} P'$, where $M \subseteq \Lambda$. Intuitively, process P can engage in an α-transition to P' whenever the environment does not offer communications on ports in M. Despite notational differences, this is the same underlying principle as for the transition relations defined in the previous sections, which are parameterized by initial action sets. Note that $\alpha \in \overline{\Lambda}$ implies $M = \emptyset$. The $\mathsf{CCS^{cw}}$ transition relation is formally defined in Table **??**, where $f(M)$ stands for $\{f(m) \mid m \in M\}$. Recall

Table 22
Operational semantics for $\mathsf{CCS}^{\mathsf{cw}}$

$$\mathsf{Act}\quad \frac{-}{\vdash^{\mathsf{cw}}_{\emptyset} \alpha.P \xrightarrow{\alpha} P} \qquad\qquad \mathsf{Res}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{M\setminus(L\cup\overline{L})} P\setminus L \xrightarrow{\alpha} P'\setminus L}\ \alpha\notin L\cup\overline{L}$$

$$\mathsf{Sum1}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{M} P +\rangle\, Q \xrightarrow{\alpha} P'} \qquad\qquad \mathsf{Sum2}\quad \frac{\vdash^{\mathsf{cw}}_{N} Q \xrightarrow{\alpha} Q'}{\vdash^{\mathsf{cw}}_{N\cup\mathbb{I}^{\mathsf{cw}}(P)} P +\rangle\, Q \xrightarrow{\alpha} Q'}\ \tau,\alpha\notin \mathbb{I}^{\mathsf{cw}}(P)$$

$$\mathsf{iSum1}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{M} P + Q \xrightarrow{\alpha} P'} \qquad\qquad \mathsf{Com1}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{M} P\,|\,Q \xrightarrow{\alpha} P'\,|\,Q}\ M\cap\overline{\mathbb{I}}^{\mathsf{cw}}(Q)=\emptyset$$

$$\mathsf{iSum2}\quad \frac{\vdash^{\mathsf{cw}}_{N} Q \xrightarrow{\alpha} Q'}{\vdash^{\mathsf{cw}}_{N} P + Q \xrightarrow{\alpha} Q'} \qquad\qquad \mathsf{Com2}\quad \frac{\vdash^{\mathsf{cw}}_{N} Q \xrightarrow{\alpha} Q'}{\vdash^{\mathsf{cw}}_{N} P\,|\,Q \xrightarrow{\alpha} P\,|\,Q'}\ N\cap\overline{\mathbb{I}}^{\mathsf{cw}}(P)=\emptyset$$

$$\mathsf{Rel}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{f(M)} P[f] \xrightarrow{f(\alpha)} P'[f]} \qquad \mathsf{Com3}\quad \frac{\vdash^{\mathsf{cw}}_{M} P \xrightarrow{a} P' \quad \vdash^{\mathsf{cw}}_{\emptyset} Q \xrightarrow{\overline{a}} Q'}{\vdash^{\mathsf{cw}}_{M} P\,|\,Q \xrightarrow{\tau} P'\,|\,Q'}\ M\cap\overline{\mathbb{I}}^{\mathsf{cw}}(Q)=\emptyset$$

$$\mathsf{Rec}\quad \frac{\vdash^{\mathsf{cw}}_{M} P[\mu x.P/x] \xrightarrow{\alpha} P'}{\vdash^{\mathsf{cw}}_{M} \mu x.P \xrightarrow{\alpha} P'} \qquad \mathsf{Com4}\quad \frac{\vdash^{\mathsf{cw}}_{\emptyset} P \xrightarrow{\overline{a}} P' \quad \vdash^{\mathsf{cw}}_{N} Q \xrightarrow{a} Q'}{\vdash^{\mathsf{cw}}_{N} P\,|\,Q \xrightarrow{\tau} P'\,|\,Q'}\ N\cap\overline{\mathbb{I}}^{\mathsf{cw}}(P)=\emptyset$$

that the initial actions of P in $P +\rangle\, Q$ are given preference over the initial actions of Q. As expected, a prioritized τ, i.e., an internal action in which the left argument of $+\rangle$ can initially engage, has pre-emptive power over unprioritized actions, i.e., actions in which the right argument of $+\rangle$ can initially engage. Thus, the prioritized choice operator $+\rangle$ in $\mathsf{CCS}^{\mathsf{cw}}$ is evocative of the summation operator $+$ in $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$. In [?] 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 $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$. We further investigate the correspondence of these operators in the next section.

Camilleri and Winskel also developed a bisimulation-based semantic theory for $\mathsf{CCS}^{\mathsf{cw}}$. Their notion of strong bisimulation for $\mathsf{CCS}^{\mathsf{cw}}$, as defined below, is shown to be a congruence [?].

Definition 4.18 *A symmetric relation* $\mathcal{R}\subseteq\mathcal{P}\times\mathcal{P}$ *is a* distributed prioritized strong bisimulation for $\mathsf{CCS}^{\mathsf{cw}}$ *if for every* $\langle P,Q\rangle\in\mathcal{R}$, $\alpha\in\mathcal{A}$, *and* $M\subseteq\Lambda$ *the* *following condition holds.*

$$\vdash^{cw}_{M} P \stackrel{\alpha}{\rightarrow} P' \ \textit{ implies } \ \exists Q', N.\ \vdash^{cw}_{N} Q \stackrel{\alpha}{\rightarrow} Q',\ N \subseteq M, \ \textit{ and } \langle P', Q' \rangle \in \mathcal{R}$$

We write $P \simeq_{\mathsf{cw}} Q$ if $\langle P, Q \rangle \in \mathcal{R}$ for some distributed prioritized strong bisimulation $\mathcal{R}$ for $\mathsf{CCS}^{\mathsf{cw}}$.

4.9 Relating $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ and $\mathsf{CCS}^{\mathsf{cw}}$

We show that the algebras $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ and $\mathsf{CCS}^{\mathsf{cw}}$ are closely related by providing an embedding of $\mathsf{CCS}^{\mathsf{cw}}$ in $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$. For this purposes we define $\mathcal{N} =_{\mathrm{df}} \{0,1\}^*$ and the strict order $<$ on priority values to be the lexicographical order on $\mathcal{N}$, where 1 is less than 0.

Table 23
Translation function

$\xi^k(\mathbf{0}) \quad =_{\mathrm{df}} \mathbf{0}$	$\xi^k(P+Q) =_{\mathrm{df}} \xi^k(P) \oplus \xi^k(Q)$	$\xi^k(P \setminus L) =_{\mathrm{df}} \xi^k(P) \setminus L$
$\xi^k(x) \quad =_{\mathrm{df}} x$	$\xi^k(P \,+\!\!\rangle\, Q) =_{\mathrm{df}} \xi^{k0}(P) + \xi^{k1}(Q)$	$\xi^k(P[f]) \ =_{\mathrm{df}} \xi^k(P)[f]$
$\xi^k(\gamma.P) =_{\mathrm{df}} \gamma\!:\!k.\xi^{\epsilon}(P)$	$\xi^k(P \,\vert\, Q) \ =_{\mathrm{df}} \xi^k(P) \mid \xi^k(Q)$	$\xi^k(\mu x.P) =_{\mathrm{df}} \mu x.\xi^k(P)$
$\xi^k(\overline{a}.P) =_{\mathrm{df}} \overline{a}.\xi^{\epsilon}(P)$		

We now introduce the *translation function* $\xi(\cdot) : \mathcal{P}^{\mathsf{cw}} \longrightarrow \mathcal{P}^{\mathsf{sl}}_{\mathsf{ml}}$ by defining $\xi(P) =_{\mathrm{df}} \xi^{\epsilon}(P)$, which maps $\mathsf{CCS}^{\mathsf{cw}}$ terms to $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ terms. The functions $\xi^k(P)$, for $k \in \mathcal{N}$, are inductively defined over the structure of $\mathsf{CCS}^{\mathsf{cw}}$ processes as shown in Table **??**. We note that the translation function is not *surjective*, e.g., consider process $(a\!:\!0.\mathbf{0}+b\!:\!2.\mathbf{0})+c\!:\!1.\mathbf{0}$ on which no $\mathsf{CCS}^{\mathsf{cw}}$ process is mapped. This example also shows that the notion of compositionality in $\mathsf{CCS}^{\mathsf{cw}}$ is more restrictive than the one in $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$, since a comparable summation can only be extended by summands that have a higher or a lower priority than the already considered summands. The following theorem, which is proved in [**?**], makes the semantic relationship between a $\mathsf{CCS}^{\mathsf{cw}}$ process P and its embedding $\xi(P)$ precise.

Theorem 4.19 *Let $P, Q \in \mathcal{P}^{\mathsf{cw}}$. Then $P \simeq_{\mathsf{cw}} Q$ if and only if $\xi(P) \simeq_{\mathsf{ml}} \xi(Q)$.*

Consequently, distributed prioritized strong bisimulation for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ is also compositional with respect to summation in the sub-calculus of $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ induced by $\mathsf{CCS}^{\mathsf{cw}}$.

A local concept of pre-emption is also considered by Hansson and Orava in [?], where CSP [?] is extended with priority by assigning natural numbers to actions. As for $\mathsf{CCS}^{\mathsf{sl}}$, they use a notion of location, and pre-emption in the calculus is sensitive to locations. Indeed, their work served as an inspiration for $\mathsf{CCS}^{\mathsf{sl}}$. However, the authors only conjecture that their version of strong bisimulation is a congruence, and they provide neither an axiomatization for their behavioral relation nor a theory for observational congruence. One may also criticize their semantics as not truly reflecting distributed computation. In particular, despite having a local pre-emptive semantics they compute a global priority for synchronizations.

After stressing the strong similarity of $\mathsf{CCS}^{\mathsf{sl}}$ to the process algebra $\mathsf{CCS}^{\mathsf{cw}}$ in the previous section, we focus on the algebraic results established in these frameworks. In [?,?] the transition relation is directly annotated with pre-emption potentials. By using this transition relation in the definition of standard strong bisimulation one immediately obtains a congruence. In contrast, [?] starts off by defining *naive* distributed prioritized strong bisimulation using the naive transition relation; the potential of pre-emption is then considered subsequently (by introducing the distributed prioritized initial action set condition). Then it is shown that the resulting congruence is the largest congruence in the naive equivalence. Similarly, Jensen [?] defines a naive distributed prioritized weak bisimulation based on the above-mentioned annotated transition relation. His naive weak transition relation corresponds to the distributed prioritized weak transition relation in $\mathsf{CCS}^{\mathsf{sl}}$ if the parameter M is dropped. Because of the difference in the naive transition relations the abstraction result presented here is somewhat stronger than Jensen's, although the observational congruences appear to coincide.

One may wonder about the relationship between $\mathsf{CCS}^{\mathsf{sl}}$ and $\mathsf{CCS}^{\mathsf{sg}}$, i.e., the static priority and global pre-emption language of Section **??**. If in $\mathsf{CCS}^{\mathsf{sl}}$ the distributed summation operator is left out and pre-emption is globalized by defining $[m] =_{\text{df}} \mathcal{L}oc$, for all $m \in \mathcal{L}oc$, the operational semantics and the behavioral relations reduce to the corresponding notions presented in Section **??**.

Like Camilleri and Winskel, Barrett [?] devises a semantics of `occam`'s priority mechanism that is additionally concerned with fairness aspects. His framework is based on a structural operational semantics augmented with *ready-guard* sets which model possible inputs from the environment. Intuitively, these sets characterize the nature of the contexts in which a transition is enabled. Thus, they correspond to the action sets with which the $\mathsf{CCS}^{\mathsf{sl}}$ and the $\mathsf{CCS}^{\mathsf{cw}}$ transition relations are parameterized. Barrett is not concerned with investigating behavioral relations, but focuses instead on implementing `occam`'s `PRIALT` and

PRIPAR constructs on the transputer platform.

Other researchers have also extended Hoare's *Communicating Sequential Processes* (CSP) [?] with a concept of static priority. Inspired by the notion of priority in Ada [?], Fidge [?] introduced new versions of the operators for external choice, parallel composition by interleaving, and parallel composition by intersection. Their operational semantics favors execution of the left-hand operands, as was also done by Jensen in [?]. The semantic theory in [?] is based on *failure semantics* which is made sensitive to local pre-emption. For this purpose, traces are augmented with a *preference function* that identifies the priority relation on the initial action sets of a given process. A related approach was presented by Lowe [?]. It differs from [?] in that the underlying algebra is a timed version of CSP [?]. Additionally, Lowe aims at obtaining a fully deterministic language by employing a similar notion of priority as the one proposed by Fidge.

Finally, we remark on the notions of strong and weak bisimulation for $\mathsf{CCS}^{\mathsf{sl}}$. Since the semantic theory reflects local pre-emption, locations occur implicitly in our semantic equivalences. In contrast to work on *location equivalences* [?,?,?], we only *indirectly* consider locations for defining a suitable notion of local pre-emption potential in form of prioritized initial action sets; our objective is not to observe locations but to capture local pre-emption.

5 Dynamic Priority and Global Pre-emption

This section develops a theory in which priorities are dynamic and pre-emption is global. The motivation for this theory originated in a desire to devise a compact model of real-time computation, and we devote significant space to establishing a tight connection between the seemingly different notions of priority and real-time [?]. For this purpose we equip our language with a dynamic priority semantics based on global pre-emption and refer to it as $\mathsf{CCS}^{\mathsf{dg}}$ (CCS with dynamic priority and global pre-emption). The connection with real-time arises when we interpret delays as priorities: the longer the delay preceding an action, the lower its priority. This approach contrasts significantly with more traditional accounts of real-time, where the only notion of pre-emption arises in the context of the *maximal progress assumption* [?], which states that time may only pass if the system under consideration cannot engage in any further internal computation. The main result of this section is the formalization of a one-to-one correspondence between the strong-bisimulation equivalences induced by dynamic priority semantics and real-time semantics.

Unlike the process algebras with priority considered so far, actions in $\mathsf{CCS}^{\mathsf{dg}}$ have priority values that may change as systems evolve. Accordingly, we alter our point of view regarding actions and priorities by separating action names from their priority values, i.e., an action's priority is no longer implicit in its port name. In this vein, we take the set of actions $\mathcal{A}$ to be $\{\alpha, \beta, \ldots\}$. We also allow priority values to come from the full set $\mathbb{N}$ of natural numbers rather than a finite set. Our syntax of processes will then require that each action is equipped with a priority value taken from $\mathbb{N}$.

The structure of this section is as follows. Section **??** briefly presents a real-time semantics for our language, whereas the dynamic priority semantics is introduced in Section **??**. The one-to-one correspondence between dynamic priority semantics and real-time semantics is established in Section **??**. Finally, Section **??** contains our concluding remarks and discusses related work.

5.1 Real-time Semantics

We first define a real-time semantics, $\mathsf{CCS}^{\mathsf{rt}}$, for the language introduced in Section **??**. $\mathsf{CCS}^{\mathsf{rt}}$ explicitly represents timing behavior: the semantics of a process is defined by a labeled transition system that contains explicit *clock transitions* — each representing a delay of one time unit — as well as *action transitions*. With respect to clock transitions, the operational semantics is such that processes willing to communicate with some process running in parallel are able to wait until the communication partner is ready. However, as soon as it is available, the communication must occur, i.e., further idling is prohibited. This assumption is usually referred to as the maximal progress assumption [**?**] or the *synchrony hypothesis* [**?**].

Table 24

Operational semantics for $\mathsf{CCS}^{\mathsf{rt}}$ (action transitions)

$$\mathsf{Act}\ \frac{-}{\alpha\!:\!0.P \stackrel{\alpha}{\longmapsto} P} \qquad \mathsf{Sum1}\ \frac{P \stackrel{\alpha}{\longmapsto} P'}{P+Q \stackrel{\alpha}{\longmapsto} P'} \qquad \mathsf{Rec}\ \frac{P[\mu x.P/x] \stackrel{\alpha}{\longmapsto} P'}{\mu x.P \stackrel{\alpha}{\longmapsto} P'}$$

$$\mathsf{Rel}\ \frac{P \stackrel{\alpha}{\longmapsto} P'}{P[f] \stackrel{f(\alpha)}{\longmapsto} P'[f]} \qquad \mathsf{Sum2}\ \frac{Q \stackrel{\alpha}{\longmapsto} Q'}{P+Q \stackrel{\alpha}{\longmapsto} Q'} \qquad \mathsf{Res}\ \frac{P \stackrel{\alpha}{\longmapsto} P'}{P \setminus L \stackrel{\alpha}{\longmapsto} P' \setminus L}\ \alpha \notin L \cup \overline{L}$$

$$\mathsf{Com1}\ \frac{P \stackrel{\alpha}{\longmapsto} P'}{P|Q \stackrel{\alpha}{\longmapsto} P'|Q} \qquad \mathsf{Com2}\ \frac{Q \stackrel{\alpha}{\longmapsto} Q'}{P|Q \stackrel{\alpha}{\longmapsto} P|Q'} \qquad \mathsf{Com3}\ \frac{P \stackrel{a}{\longmapsto} P' \quad Q \stackrel{\overline{a}}{\longmapsto} Q'}{P|Q \stackrel{\tau}{\longmapsto} P'|Q'}$$

Formally, the labeled transition system corresponding to a process P is a four-tuple $\langle \mathcal{P}, \mathcal{A} \cup \{1\}, \longmapsto, P \rangle$, where the alphabet $\mathcal{A} \cup \{1\}$ satisfies $1 \notin \mathcal{A}$. The transition relation for actions coincides with the one for traditional CCS (cf.

Table **??**). The transition relation $\longmapsto \subseteq \mathcal{P} \times \{1\} \times \mathcal{P}$ for clock transitions is defined in Table **??**. We use γ as representative of $\mathcal{A} \cup \{1\}$, and write $P \overset{\gamma}{\longmapsto} P'$ instead of $\langle P, \gamma, P' \rangle \in \longmapsto$. If $\gamma \in \mathcal{A}$ we speak of an *action transition*, otherwise of a *clock transition*. Sometimes it is convenient to write $P \overset{\gamma}{\longmapsto}$ for $\exists P' \in \mathcal{P}.\, P \overset{\gamma}{\longmapsto} P'$. In order to ensure maximal progress, our operational semantics is such that $P \overset{1}{\not\longmapsto}$ whenever $P \overset{\tau}{\longmapsto}$, i.e., clock transitions are preempted if P can engage in internal computation.

Table 25
Operational semantics for $\mathsf{CCS}^{\mathsf{rt}}$ (clock transitions)

tNil	$\dfrac{-}{\mathbf{0} \overset{1}{\longmapsto} \mathbf{0}}$	tRec	$\dfrac{P[\mu x.P/x] \overset{1}{\longmapsto} P'}{\mu x.P \overset{1}{\longmapsto} P'}$
tAct1	$\dfrac{-}{\alpha\!:\!k.P \overset{1}{\longmapsto} \alpha\!:\!(k-1).P}\ k > 0$	tAct2	$\dfrac{-}{a\!:\!0.P \overset{1}{\longmapsto} a\!:\!0.P}$
tSum	$\dfrac{P \overset{1}{\longmapsto} P' \quad Q \overset{1}{\longmapsto} Q'}{P + Q \overset{1}{\longmapsto} P' + Q'}$	tCom	$\dfrac{P \overset{1}{\longmapsto} P' \quad Q \overset{1}{\longmapsto} Q'}{P \vert Q \overset{1}{\longmapsto} P' \vert Q'}\ P \vert Q \overset{\tau}{\not\longmapsto}$
tRel	$\dfrac{P \overset{1}{\longmapsto} P'}{P[f] \overset{1}{\longmapsto} P'[f]}$	tRes	$\dfrac{P \overset{1}{\longmapsto} P'}{P \setminus L \overset{1}{\longmapsto} P' \setminus L}$

Intuitively, process $\alpha\!:\!k.P$, where $k > 0$, may engage in a clock transition and then behave like $\alpha\!:\!(k-1).P$. Process $\alpha\!:\!0.P$ performs an α-transition to become process P. Moreover, if $\alpha \not\equiv \tau$, it may also idle by executing a clock transition to itself. Time must proceed equally on both sides of summation, i.e., $P + Q$ can engage in a clock transition and, thus, delay the nondeterministic choice if and only if both P and Q can engage in a clock transition. Hence, time is a deterministic concept. Similar to summation, P and Q must synchronize on clock transitions according to Rule (tCom). Its side condition implements maximal progress by ensuring that there is no pending communication between P and Q. Although this condition is negative, our semantics is still well-defined [**?**]. A semantic theory based on the notion of bisimulation has been developed for $\mathsf{CCS}^{\mathsf{rt}}$(cf. [**?**]). For this section we restrict ourselves to (*strong*) temporal bisimulation, a congruence which is defined as follows.

Definition 5.1 (Temporal Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is called a temporal bisimulation *if for every $\langle P, Q \rangle \in \mathcal{R}$ and $\gamma \in \mathcal{A} \cup \{1\}$ the following holds: $P \overset{\gamma}{\longmapsto} P'$ implies $\exists Q'.\, Q \overset{\gamma}{\longmapsto} Q'$ and $\langle P', Q' \rangle \in \mathcal{R}$. We write $P \sim_{rt} Q$ if $\langle P, Q \rangle \in \mathcal{R}$ for some temporal bisimulation $\mathcal{R}$.*

Observe that $\mathsf{CCS}^{\mathsf{rt}}$ semantics unfolds every delay value into a sequence of elementary time units. For example, process $a\!:\!k.\mathbf{0}$ has $k+2$ states, namely $\mathbf{0}$ and $a\!:\!l.\mathbf{0}$, for $0 \leq l \leq k$ (see also Figure **??** in Section **??**). Representing $a\!:\!k.\mathbf{0}$ by a single transition, which is labeled by $a\!:\!k$ and leads to state $\mathbf{0}$, would yield labeled transition systems with fewer states. This idea of compacting the state space of real-time systems can be implemented by viewing k as *priority value* assigned to action a.

5.2 Dynamic Priority Semantics

We introduce $\mathsf{CCS}^{\mathsf{dg}}$, i.e., a dynamic priority semantics for our language presented in Section **??**. The notion of pre-emption incorporated in $\mathsf{CCS}^{\mathsf{dg}}$ is similar to $\mathsf{CCS}^{\mathsf{sg}}$, and it naturally encodes the maximal progress assumption employed in $\mathsf{CCS}^{\mathsf{rt}}$ semantics. Formally, the $\mathsf{CCS}^{\mathsf{dg}}$ semantics of a process P is given by a labeled transition system $\langle \mathcal{P}, \mathcal{A} \times \mathbb{N}, \longrightarrow, P \rangle$. The presentation of the operational rules for the transition relation $\longrightarrow$ requires two auxiliary definitions.

Table 26
Potential initial action sets for $\mathsf{CCS}^{\mathsf{dg}}$

$\mathrm{I}^k(\alpha\!:\!l.P)$	$= \{\alpha \,\vert\, l \leq k\}$	$\mathrm{I}^k(P \,\vert\, Q)$	$= \mathrm{I}^k(P) \cup \mathrm{I}^k(Q) \cup \{\tau \,\vert\, \mathrm{I}^k(P) \cap \overline{\mathrm{I}^k(Q)} \neq \emptyset\}$
$\mathrm{I}^k(P+Q)$	$= \mathrm{I}^k(P) \cup \mathrm{I}^k(Q)$	$\mathrm{I}^k(P[f])$	$= \{f(\alpha) \,\vert\, \alpha \in \mathrm{I}^k(P)\}$
$\mathrm{I}^k(\mu x.P)$	$= \mathrm{I}^k(P[\mu x.P/x])$	$\mathrm{I}^k(P \setminus L)$	$= \mathrm{I}^k(P) \setminus (L \cup \overline{L})$

First, we introduce *potential initial action sets* as defined in Table **??**, the actions in which a given process can potentially engage. Note that these sets are only supersets of the initial actions of processes because they do not take *pre-emption* into account. However, this is sufficient for our purposes regarding pre-emption, since $\tau \notin \mathrm{I}^{<k}(P)$ if and only if $\not\exists l < k.\, P \xrightarrow{\tau:l}$, where $\mathrm{I}^{<k}(P) =_{\mathrm{df}} \mathrm{I}^{k-1}(P)$, for $k > 0$, and $\mathrm{I}^{<0}(P) =_{\mathrm{df}} \emptyset$.

Table 27
Priority adjustment function

$[\mathbf{0}]^k$	$=_{\mathrm{df}} \mathbf{0}, \quad [x]^k =_{\mathrm{df}} x$	$[\mu x.P]^k$	$=_{\mathrm{df}} [P[\mu x.P/x]]^k$
$[\alpha\!:\!l.P]^k$	$=_{\mathrm{df}} \alpha\!:\!(l-k).P \;$ if $l > k$	$[\alpha\!:\!l.P]^k$	$=_{\mathrm{df}} \alpha\!:\!0.P \;$ if $l \leq k$
$[P+Q]^k$	$=_{\mathrm{df}} [P]^k + [Q]^k$	$[P \vert Q]^k$	$=_{\mathrm{df}} [P]^k \,\vert\, [Q]^k$
$[P[f]]^k$	$=_{\mathrm{df}} [P]^k[f]$	$[P \setminus L]^k$	$=_{\mathrm{df}} [P]^k \setminus L$

As second auxiliary definition for presenting the transition relation, we introduce a *priority adjustment function* as shown in Table **??**. Intuitively, if one parallel component of a process engages in a transition with priority k, then the priority values of all initial actions at every other parallel component have to be decreased by k, i.e., those actions become equally "more urgent." Thus, the semantics of parallel composition employs a kind of *fairness assumption*, and priorities have a *dynamic* character. More precisely, the priority adjustment function applied to a process P and a natural number k, denoted as $[P]^k$, returns a process term that is "identical" to P except that the priority values of the initial, top-level actions are decreased by k. Note that a priority value cannot become less than 0 and that "identical" does not mean syntactic equality but syntactic equality *up to unfolding* of recursion.

Table 28
Operational semantics for $\mathsf{CCS}^{\mathsf{dg}}$

$$\mathsf{Act1}\ \ \frac{-}{a\!:\!k.P \xrightarrow{a:l} P}\ l \geq k \qquad\qquad \mathsf{Act2}\ \ \frac{-}{\tau\!:\!k.P \xrightarrow{\tau:k} P}$$

$$\mathsf{Sum1}\ \ \frac{P \xrightarrow{\alpha:k} P'}{P+Q \xrightarrow{\alpha:k} P'}\ \tau \notin \mathrm{I}^{<k}(Q) \qquad\qquad \mathsf{Sum2}\ \ \frac{Q \xrightarrow{\alpha:k} Q'}{P+Q \xrightarrow{\alpha:k} Q'}\ \tau \notin \mathrm{I}^{<k}(P)$$

$$\mathsf{Com1}\ \ \frac{P \xrightarrow{\alpha:k} P'}{P|Q \xrightarrow{\alpha:k} P'|[Q]^k}\ \tau \notin \mathrm{I}^{<k}(P|Q) \qquad\qquad \mathsf{Rel}\ \ \frac{P \xrightarrow{\alpha:k} P'}{P[f] \xrightarrow{f(\alpha):k} P'[f]}$$

$$\mathsf{Com2}\ \ \frac{Q \xrightarrow{\alpha:k} Q'}{P|Q \xrightarrow{\alpha:k} [P]^k|Q'}\ \tau \notin \mathrm{I}^{<k}(P|Q) \qquad\qquad \mathsf{Res}\ \ \frac{P \xrightarrow{\alpha:k} P'}{P \setminus L \xrightarrow{\alpha:k} P' \setminus L}\ \alpha \notin L \cup \overline{L}$$

$$\mathsf{Com3}\ \ \frac{P \xrightarrow{a:k} P' \quad Q \xrightarrow{\overline{a}:k} Q'}{P|Q \xrightarrow{\tau:k} P'|Q'}\ \tau \notin \mathrm{I}^{<k}(P|Q) \qquad\qquad \mathsf{Rec}\ \ \frac{P[\mu x.P/x] \xrightarrow{\alpha:k} P'}{\mu x.P \xrightarrow{\alpha:k} P'}$$

The operational rules in Table **??** capture the following intuition. Process $a\!:\!k.P$ may engage in action a, with priority value $l \geq k$, yielding process P. The side condition $l \geq k$ means that k does not specify an exact priority but the *maximum* priority of the initial transition of $a\!:\!k.P$. Due to the notion of pre-emption incorporated in $\mathsf{CCS}^{\mathsf{dg}}$, $\tau\!:\!k.P$ may not perform the initial τ-transition with a lower priority than k. Process $P+Q$ may behave like P if Q does not pre-empt the considered transition by being able to engage in a higher prioritized internal transition. Process $P\,|\,Q$ denotes the *parallel composition* of P and Q according to an interleaving semantics with synchronized communication on complementary actions of P and Q having the same priority value k, which results in the internal action τ attached with priority value k (cf. Rule (Com3)). The interleaving Rules (Com1) and (Com2) incorporate the dynamic behavior of priority values as explained in the previous paragraph. The side conditions of Rules (Com$i$) implement global pre-emption. The semantics for *relabeling*, *restriction*, and *recursion* is straightforward. As for $\mathsf{CCS^{rt}}$, we introduce a notion of strong bisimulation, referred to as *prioritized bisimulation.*

Definition 5.2 (Prioritized Bisimulation)
A symmetric relation $\mathcal{R} \subseteq \mathcal{P} \times \mathcal{P}$ is called prioritized bisimulation *if for every $\langle P, Q \rangle \in \mathcal{R}$, $\alpha \in \mathcal{A}$, and $k \in \mathbb{N}$ the following holds: $P \xrightarrow{\alpha:k} P'$ implies $\exists Q'. Q \xrightarrow{\alpha:k} Q'$ and $\langle P', Q' \rangle \in \mathcal{R}$. We write $P \sim_{dg} Q$ if there exists a prioritized bisimulation $\mathcal{R}$ such that $\langle P, Q \rangle \in \mathcal{R}$.*

5.3 Relating Dynamic Priority and Real-time Semantics

We show that $\mathsf{CCS^{dg}}$ and $\mathsf{CCS^{rt}}$ semantics are closely related. The underlying intuition is best illustrated by a simple example dealing with the prefixing operator. Figure **??** depicts the dynamic priority semantics and real-time semantics of process $a\!:\!k.\mathbf{0}$. Both transition systems intuitively reflect that process $a\!:\!k.\mathbf{0}$ must delay at least k time units before it may engage in the a-transition. According to $\mathsf{CCS^{rt}}$ semantics, this process consecutively engages in k clock transitions passing the states $a\!:\!(k-l).\mathbf{0}$, for $0 \leq l \leq k$, before it may either continue idling in state $a\!:\!0.\mathbf{0}$ or perform the a-transition to inaction process $\mathbf{0}$. Thus, time is explicitly part of states and made visible by clock transitions, each representing a step consuming one time unit.

In contrast, the dynamic priority semantics encodes the delay of at least k time units in the transitions rather than in the states. Hence, it possesses only the two states $a\!:\!k.\mathbf{0}$ and $\mathbf{0}$ connected via transitions labeled by $a\!:\!l$ for $l \geq k$. Although at first sight it seems that the price for saving intermediate states is to be forced to deal with infinite-branching, an upper bound for l can be given. In our example this upper bound is k itself, since a delay by more than k time units only results in idling and does not enable new or disable existing system behavior. Therefore, the dynamic priority transition system of $a\!:\!k.\mathbf{0}$ just consists of the two states $a\!:\!k.\mathbf{0}$ and $\mathbf{0}$ and a *symbolic* transition labeled by $a\!:\!k$, whereas the real-time transition system has $k+2$ states and $k+2$ transitions. The following proposition formally states that $\mathsf{CCS^{dg}}$ semantics can indeed be understood as an efficient encoding of $\mathsf{CCS^{rt}}$ semantics. Here, $\stackrel{1}{\longmapsto}^k$ denotes k consecutive clock transitions.

Proposition 5.3 *Let $P, P' \in \mathcal{P}$, $\alpha \in \mathcal{A}$, and $k \in \mathbb{N}$. Then $P \xrightarrow{\alpha:k} P'$ if and only if $\exists P''. P \stackrel{1}{\longmapsto}^k P'' \stackrel{\alpha}{\longmapsto} P'$.*

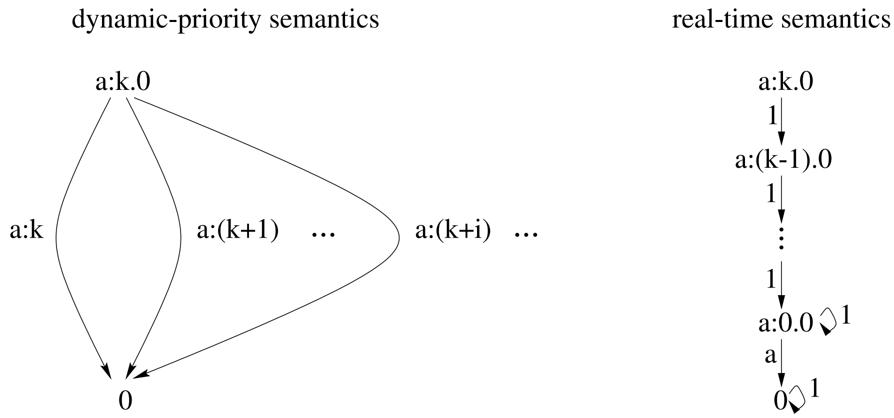

Fig. 6. Relating $\mathsf{CCS}^{\mathsf{dg}}$ semantics and $\mathsf{CCS}^{\mathsf{rt}}$ semantics

Proposition **??** is the key to prove the main result of this section.

Theorem 5.4 *Let $P, Q \in \mathcal{P}$. Then $P \sim_{dg} Q$ if and only if $P \sim_{rt} Q$.*

Consequently, prioritized and temporal bisimulation possess the same properties; in particular, prioritized bisimulation is a congruence for $\mathsf{CCS}^{\mathsf{dg}}$. Proof details can be found in [**?**].

5.4 Concluding Remarks and Related Work

As shown above, real-time semantics can be encoded by dynamic priority semantics. The utility of this encoding stems from the fact that the state space of $\mathsf{CCS}^{\mathsf{dg}}$ models is much smaller and the size of the transition relation is at least not worse, but in practice often better, than the one of corresponding $\mathsf{CCS}^{\mathsf{rt}}$ models. This has been demonstrated by formally modeling and verifying several aspects of the widely-used *SCSI-2 bus-protocol*, for which the state space of the dynamic priority model is almost an order of magnitude smaller than the one resulting from traditional real-time semantics [**?**].

Regarding related work, a similar approach to the one presented above has been investigated by Jeffrey [**?**]. He established a formal relationship between a quantitative real-time process algebra and a process algebra with static priority which is very similar to $\mathsf{CCS}^{\mathsf{sg}}$ introduced in Section **??**. Jeffrey also translates real-time to priority based on the idea of time-stamping. In contrast to $\mathsf{CCS}^{\mathsf{rt}}$ semantics, however, a process modeled in Jeffrey's framework may either immediately engage in an action transition or idle forever. This semantics does not allow a process to wait until a communication partner becomes available, but instead forces a "livelock" in such situations. It is only because of this design decision that Jeffrey does not need to choose a dynamic priority framework.

In [?] a variant of CCSR [?], called CCSR92, has been introduced, which allows for the modeling of not only static priority but also dynamic priority. The main focus of CCSR involves the specification and verification of real-time concurrent systems including scheduling behavior. Thus, a notion of dynamic priority, as applied in *priority-inheritance* and *earliest-deadline-first* scheduling algorithms, is important. In [?] dynamic priorities are given as a function of the *history* of the system under consideration. Accordingly, the operational semantics of CCSR92 is re-defined to include the historical context. The authors show that dynamic priorities do, in general, not lead to a compositional semantics and give a sufficient condition that ensures compositionality.

6 Priority in Other Process-algebraic Frameworks

This section completes the discussion of related work by focusing on approaches to priority which either (i) do not fit in our classification scheme presented in Section **??**, such as approaches for ACP [?], SCCS [?], and *stochastic* [?,?] or *probabilistic* [?,?,?] process algebras, or (ii) are concerned with process-algebraic descriptions of non-process-algebraic languages, such as *Esterel* [?,?] and *Statecharts* [?].

Baeten, Bergstra, and Klop were the first researchers who investigated priority in process algebras [?] by developing a notion of priority in the *Algebra of Communicating Processes* (ACP) [?]. Their work is inspired by the insight that it is essential to incorporate an interrupt mechanism teBerKlo85, a process algebra equipped with an axiomatic semantics. Their work is inspired by the insight that it is essential to incorporate an interrupt mechanism in process-algebraic frameworks, as it enhances their expressive power as specification and verification formalisms for concurrent systems. Therefore, a new unary operator θ together with semantics-defining equations is introduced in [?]. The definition of θ is based on a given partial order $<$ on actions. Intuitively, $\theta(P)$ is the context of P in which action a has precedence over action b, whenever $b < a$, i.e., nondeterministic choices between actions a and b are resolved within $\theta(P)$. Technically, the axiomatic semantics of the new language, notated as a *term rewrite system*, is shown to possess nice algebraic properties such as *confluence* and *termination*. The utility of the theory is demonstrated by simple examples dealing with interrupts, timeouts, and other aspects of system behavior. The approach in [?] differs from most other work presented in this chapter in that the partial order expressing priorities is fixed with respect to the system under consideration, i.e., the same priority relation holds at all states of the system. For example, if $a < b$ at some state of the system, then $a > b$ cannot be valid at another state, i.e., priorities in [?] are not *globally dynamic* in the sense of [?]. It should also be mentioned that the version of ACP used in [?] does not include a designated internal action, cf. action τ in CCS; a fact which simplifies the development of algebraic theories.

Stochastic process algebras [?,?], which enhance the expressiveness of classical process algebras by integrating *performance* descriptions of concurrent systems, also define notions of priority. One example of a popular stochastic process algebra is the *Extended Markovian Process Algebra* (EMPA) [?] whose semantics is given in terms of strong bisimulation and whose static priority approach is adapted from $\mathsf{CCS}^{\mathsf{sg}}$.

Smolka and Steffen [?] have introduced static priority to the *Synchronous Calculus of Communicating Systems* (SCCS) [?]. They extended a probabilistic version of this language, known as PCCS [?], whose semantics is given in terms of *probabilistic bisimulation*. Their work shows that the concept of priority is not only related to real-time, as investigated in Section ??, but also to probability. The main idea in [?] is to allow probability guards of value 0 to be associated with alternatives of a probabilistic summation expression. Such alternatives can be chosen only if the non-zero alternatives are precluded by contextual constraints. Thus, priority may be viewed as an extreme case of probability. Most remarkably, the semantics developed in [?] does not employ a notion of pre-emption as one would expect from any priority setting. A conjecture – which if true would justify this situation – is that the hiding operator in SCCS prohibits the pre-emption of any transition, if not compositionality is sacrificed.

Tofts has investigated another extension of SCCS, the *Weighted Synchronous Calculus of Communicating Systems* (WSCCS) [?]. Its semantics relies upon a notion of *relative frequency* that is suitable for specifying and reasoning about aspects of priority, probability, and time in concurrent systems. In this approach, priority is encoded by means of higher ordinals; a transition has priority over another if their weights are separated at least by a factor of ω. An operator similar to the θ-operator in [?] is defined, which extracts the highest priority transitions enabled at a process state by referring to a global notion of pre-emption. In contrast to [?], Toft's operator allows for different priority structures at different states. His concept of priority yields a simpler operational semantics than the one in [?], but a more complicated definition of bisimulation. For WSCCS, a congruence adapted from strong bisimulation together with an equational characterization, which is sound and complete for finite processes, has been developed.

The concept of pre-emption has also been studied by Berry in *Esterel's zero-delay process calculus* [?], a theoretical version of the Esterel synchronous programming language [?]. The calculus' semantics, which obey maximal progress [?], interprets processes as deterministic mappings from input sequences to output sequences. Berry emphasizes the importance of pre-emption in control-dominated reactive and real-time programming. He suggests that pre-emption operators be considered as first-class operators that are fully orthogonal with respect to all other primitives, such as concurrency and communication. This is in contrast to the approach chosen in this chapter, in which pre-emption is implicitly encoded as side conditions of operational rules involving non-determinism. Several examples of useful pre-emption operators are presented and axiomatized in [?], all of which are based on the ideas of *abortion* and *suspension*.

The specification language *Statecharts* [?], for which process-algebraic descriptions of Statecharts' semantics have been developed [?,?,?], extends communicating finite automata by concepts of *hierarchy*, *concurrency*, and *priority*. In Statecharts static priorities can be expressed via the absence of actions, which are called *events*, by permitting negated actions as guards, which are referred to as *triggers*. As an example, consider the Statecharts-like term $a\!:\!b.\,P + \neg b\!:\!c.\,Q$ consisting of a nondeterministic choice between a b-transition with guard a to process P, and a c-transition with guard $\neg b$ to Q. Intuitively, the Statechart may only engage in the latter transition if it cannot execute the former one. The reason is that the former transition produces event b which falsifies the guard of the c-transition. Thus, the b-transition is given precedence over the c-transition. Approaches to priority based on negated events (cf. [?]) do not go well with the concept of *hiding*, which is employed in many process algebras and also in an alternative language to Statecharts called Argos [?]. Hiding enables one to relabel a visible action into a distinguished invisible action. The difficulty with hiding arises when several events are hidden, i.e., all of them are relabeled to the same event and, thus, have the same implicit priority value attached to them. Hence, hiding may destroy priority schemes. In contrast, in the priority approaches considered in this chapter priorities are assigned to transitions, thereby allowing for a more fine-granular priority mechanism and avoiding the above-mentioned problem.

7 Conclusions and Directions for Future Work

This chapter investigated various aspects of priority in process algebras. The utility of introducing priority to traditional process algebras is to enhance their expressiveness and, thereby, making them more attractive to system designers. In a nutshell, priorities allow one to resolve potential nondeterminism.

Conclusions. We illustrated the most important aspects of priority in a prototypic language that extends Milner's CCS. This language was equipped with several semantics according to whether priorities are static or dynamic and whether the adopted notion of pre-emption is global or local.

In practice it is easy to determine when to use static priority semantics and when to use dynamic priority semantics. For modeling interrupts and prioritized choice constructs a static notion of priority is adequate, whereas for modeling real-time or scheduling behavior, dynamic priorities should be considered. However, static priority approaches also allow for the description of simple scheduling algorithms, as shown in [?] in the presence of a prioritized parallel composition operator. The dynamic priority approach yields a more efficient verification of real-time systems, since the sizes of system models with respect to dynamic priority semantics are often significantly smaller than the ones regarding real-time semantics [?]. If one needs to deal with both interrupt and real-time aspects at the same time, static and dynamic priority approaches should be combined. In this situation each action could be assigned two priority values, the first interpreted as a global priority value for scheduling purposes, and the second interpreted as a local priority value for modeling interrupts, where the first priority value has more weight than the second one.

Suitable guidelines for choosing between a global or a local notion of pre-emption are the following. A semantics obeying global pre-emption is favorable when modeling interrupts and prioritized-choice constructs in concurrent, centralized systems or when specifying real-time and scheduling aspects. Global pre-emption also provides a simple means for enforcing that action sequences are executed atomically. This supports the accurate modeling of certain system behavior and, moreover, helps one to keep system models small [?]. However, when dealing with interrupts or prioritized-choice constructs within distributed systems, the concept of global pre-emption is inadequate. Here, the use of local pre-emption does not only lead to an intuitive but also to an implementable semantics, since it does not require any knowledge about computations that are internal to other, potentially unknown sites (cf. [?]).

The three calculi presented in Sections **??**–**??** are equipped with a bisimulation-based semantics. The re-development of the semantic theory of CCS for the static priority calculi included: (i) characterizations of the largest congruences contained in the naive adaptations of the standard strong and weak bisimulations, (ii) encodings of the new behavioral relations as standard strong bisimulations on enriched transition relations, and (iii) axiomatic characterizations of the prioritized strong bisimulations for finite processes. For the dynamic priority calculus, strong bisimulation served as a semantic tool for establishing a one-to-one correspondence between dynamic priority and real-time semantics. We want to point out that the semantic theories presented here show that extensions of process algebras by priority do not need to sacrifice the simplicity and the elegance that have made traditional process-algebraic approaches successful.

This chapter also surveyed related approaches to priority which are concerned with process-algebraic calculi other than CCS and its derivatives. We have classified them according to whether priorities are considered to be static or dynamic and whether their concept of pre-emption is global or local. The concept of priority has also been investigated in other concurrency frameworks, most notably *Petri Nets* [?,?]. In this setting priorities are either expressed explicitly by priority relations over transitions [?] or implicitly via *inhibitor arcs* [?]. Finally, it should be mentioned that priorities can *implicitly* arise when studying causality for *mobile processes* (see, e.g., [?]). In these approaches, priorities eliminate superfluous paths that only present new temporal but not causal system dependencies.

Future Work. In addition to the fact that a calculus combining dynamic priority and local pre-emption has not yet been developed, the semantic theories for $\mathsf{CCS}^{\mathsf{sg}}$ and $\mathsf{CCS}^{\mathsf{sl}}$ need to be completed by axiomatizing their *observational* congruences. For *finite* processes, one should be able to establish these axiomatizations using standard techniques [?]. However, for *regular* processes — i.e., the class of finite-state processes that do not contain recursion through static operators — it is not clear how to obtain a completeness result. The point is that existing methods for proving completeness of axiomatizations with respect to observational congruences rely on the possibility to remove or to insert τ-cycles in processes. In the context of pre-emption, however, this would possibly change the pre-emption potential of processes and, thus, is semantically incompatible with the prioritized observational congruences presented here. Recently, a similar problem, which was identified for the temporal process algebra PMC [?] before, has been attacked in [?] for stochastic process calculi with priority and/or maximal progress and in [?] for a different, more classical process-algebraic setting. It remains to be seen whether these techniques can be employed for the priority settings investigated in this chapter.

Most process algebras that have been equipped with a notion of priority rely on an interleaving semantics, handshake communication, and a semantic theory based on bisimulation. Therefore, it should be investigated how the approaches and results presented here can be adapted to broadcasting calculi, such as Hoare's CSP [?]. Moreover, since for semantics based on local pre-emption the usual interleaving law is not valid, it is worth pursuing research regarding local pre-emption for non-interleaving semantic frameworks [?,?]. Preliminary considerations have been made in Jensen's thesis [?]. However, the insights obtained by Jensen are restricted to a structural operational semantics for a CCS-based calculus, which is defined using *asynchronous transition systems* [?]. Jensen's results do not involve a behavioral relation such as bisimulation (cf. [?]). Finally, we want to note that — to the best of our knowledge — extensions of higher-order process algebras [?,?] with concepts of priority do not yet exist. Thus, it would be interesting to see if some of the approaches presented here can be carried over.

8 Sources and Acknowledgments

Major parts of this chapter have been adapted from several publications by the authors, which include two PhD theses: the results of Section **??** are taken from [?,?,?,?] and the ones of Section **??** from [?,?]; Section **??** heavily borrows from material contained in [?,?].

This work was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the second author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), MS 132C, NASA Langley Research Center, Hampton, Virginia 23681-2199, USA. The first author acknowledges support by NSF grants CCR-9257963, CCR-9505662, CCR-9804091, and INT-9603441, AFOSR grant F49620-95-1-0508, and ARO grant P-38682-MA. Last but not least we would like to thank Girish Bhat, Matthew Hennessy, Michael Mendler, and Bernhard Steffen for many discussions about priority in process algebras, as well as Scott Smolka for carefully proofreading a draft of this chapter.

9 Symbols and Notation

The following table lists symbols used in this chapter and is organized according to the chapter's sections.

Notation introduced in Section ??

CCS	Calculus of Communicating Systems
$\mathcal{N}$	finite initial interval of $\mathbb{N}$
λ	port name
Λ_k	ports with priority k (receiving)
λ_k, $\lambda{:}k$	representative of Λ_k
$\overline{\Lambda}_k$	complementary ports with priority k (sending)
$\overline{\lambda}_k$, $\overline{\lambda}{:}k$	representative of $\overline{\Lambda}_k$
τ_k, $\tau{:}k$	invisible, internal action with priority k
$\mathcal{A}_k$	set of actions with priority k
Λ	set of (receiving or input) ports
$\overline{\Lambda}$	set of complementary (i.e., sending or output) ports
$a{:}k, b{:}k, \ldots$	representatives of $\Lambda \cup \overline{\Lambda}$
$\mathcal{A}$	set of actions
$\alpha{:}k, \beta{:}k, \ldots$	representatives of $\mathcal{A}$
A	set of unprioritized actions (i.e., $\mathcal{A}_0$ in 2-level priority-scheme)
α	representative of A
$\underline{A}$	set of prioritized actions (i.e., $\mathcal{A}_1$ in 2-level priority-scheme)
$\underline{\alpha}$	representative of $\underline{A}$

δ, γ	representatives of $A \cup \underline{A}$
L	subset of $\Lambda \cup \overline{\Lambda}$ (restriction set)
f	finite relabeling
$\mathcal{V}$	set of variables
x	representative of $\mathcal{V}$
$\mathcal{P}$	set of processes
$P, Q, R, \ldots$	representatives of $\mathcal{P}$
$C[X]$	representative of a context
$\mathbf{0}$	nil process
$\alpha{:}k.$	prefix operator
$+$	nondeterministic choice operator
$\mid$	parallel-composition operator
$[f]$	relabeling operator
$\backslash L$	restriction operator
$\mu x.P$, $x \stackrel{\text{def}}{=} P$	recursion
$\equiv$	syntactic equality
$\mathcal{R}^+$	largest congruence contained in equivalence $\mathcal{R}$

Notation for static priority and global pre-emption (cf. Section ??)

$\mathsf{CCS}^{\mathsf{sg}}$	CCS with static priority and global pre-emption
$\underline{\mathcal{I}}(P)$	prioritized initial action set of P
$\underline{\mathcal{II}}(P)$	visible prioritized initial action set of P (i.e., $\underline{\mathcal{I}}(P) \setminus \{\underline{\tau}\}$)
$\stackrel{\gamma}{\longrightarrow}$	(standard) transition relation
$\stackrel{\alpha}{\underset{L}{\longrightarrow}}$	parameterized transition relation
$\stackrel{\gamma}{\Longrightarrow}_{\times}$	naive prioritized weak transition relation
$\stackrel{\alpha}{\Longrightarrow}$, $\stackrel{\alpha}{\underset{L}{\Longrightarrow}}$	prioritized weak transition relation
$\stackrel{\gamma}{\Longrightarrow}_{*}$	alternative prioritized weak transition relation
$\simeq$	prioritized strong bisimulation
$\approx_{\times}$	naive prioritized weak bisimulation
$\underline{\approx}$	prioritized weak bisimulation
$\underline{\approx}^{\mathrm{l}}$	prioritized observational congruence
$\underline{\approx}_{*}$	alternative prioritized weak bisimulation
$\underline{\approx}_{\mathrm{pd}}$	extended prioritized weak bisimulation
$\underline{\approx}^{\mathrm{l}}_{\mathrm{pd}}$	extended prioritized observational congruence
$\lceil \cdot$	prioritization operator
$\lfloor \cdot$	deprioritization operator
$\vdash$	axiomatic derivation
$\sqsubseteq_{\mathrm{i}}$	auxiliary pre-congruence for axiomatization

Notation for static priority and local pre-emption (cf. Section ??)

$\mathsf{CCS}^{\mathsf{sl}}$	CCS with static priority and local pre-emption
$\mathcal{A}_{\mathrm{addr}}$	address alphabet $\{L, R, l, r\}$
$Addr$	set of addresses
$\bullet$	address pointing to current process term

$\mathcal{L}oc$	set of transition locations
m, n, o	representatives of $\mathcal{L}oc$
$\bowtie$	location comparability relation
$\oplus$	distributed-summation operator
$\vdash_E$	axiomatic derivation
$\natural$	auxiliary predicate for axiomatization
$\underline{\mathcal{I}}_m(P)$	distributed prioritized initial action set of P wrt. location m
$\underline{\mathcal{I}}_M(P)$	... wrt. locations $m \in M$
$\underline{\mathcal{I}}(P)$	distributed prioritized initial action set of P
$\underline{\mathcal{II}}(P)$	visible distributed prioritized initial action set of P
$\underline{\mathcal{II}}_M(P)$	... wrt. locations $m \in M$ (i.e., $\underline{\mathcal{I}}_M(P) \setminus \{\underline{\tau}\}$)
$\underset{L}{\overset{\alpha}{\longrightarrow}}$, $\underset{L}{\overset{m,\alpha}{\longrightarrow}}$	parameterized transition relation
$\overset{\underline{\alpha}}{\Longrightarrow}_\times$, $\overset{m,\alpha}{\Longrightarrow}_\times$	naive distributed prioritized weak transition relation
$\overset{\underline{\alpha}}{\Longrightarrow}$, $\underset{L,M}{\overset{m,\alpha}{\Longrightarrow}}$	distributed prioritized weak transition relation
$\underline{\simeq}$	naive distributed prioritized strong bisimulation
$\underline{\simeq}^{\mathrm{l}}$	distributed prioritized strong bisimulation
$\underline{\simeq}_*$	alternative distributed prioritized strong bisimulation
$\underline{\approx}_\times$	naive distributed prioritized weak bisimulation
$\underline{\approx}$	distributed prioritized weak bisimulation
$\underline{\approx}^{\mathrm{l}}$	distributed prioritized observational congruence
$\underline{\approx}_*$	alternative distributed prioritized weak bisimulation

Notation for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ (cf. Section **??***)*

$\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$	$\mathsf{CCS}^{\mathsf{sl}}$ with a multi-level priority-scheme
Λ	set of input ports
$a, b, \ldots$	representatives of Λ
γ	representative of $\Lambda \cup \{\tau\}$
$\overline{\Lambda}$	set of output ports
$\overline{a}, \overline{b}, \ldots$	representatives of $\overline{\Lambda}$
$\mathcal{A}$	set of actions (i.e., $\mathcal{A} =_{\mathrm{df}} \Lambda \cup \overline{\Lambda} \cup \{\tau\}$)
α	representative of $\mathcal{A}$
$\mathcal{P}^{\mathsf{sl}}_{\mathsf{ml}}$	set of $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$ processes
$\xrightarrow{m,\alpha}$	transition relation for $\mathsf{CCS}^{\mathsf{sl}}$
$\underline{\mathcal{II}}(P)$	visible unprioritized initial actions of P
$\overline{\underline{\mathcal{II}}}(P)$	initial output action set for P
$\mathrm{I}^k_m(P)$	initial input action set for P wrt. priority k and location m
$\mathrm{I}^{<k}_M(P)$, $\mathrm{II}^{<k}_M(P)$	variants of initial input action sets
$\mathrm{I}(P)$	initial input action set of P
$\mathrm{II}(P)$	initial visible input action set of P (i.e., $\mathrm{I}(P) \setminus \{\tau\}$)
$\underline{\simeq}_{\mathsf{ml}}$	distributed prioritized strong bisimulation for $\mathsf{CCS}^{\mathsf{sl}}_{\mathsf{ml}}$

Notation for $\mathsf{CCS}^{\mathsf{cw}}$ (cf. Section **??***)*

$\mathsf{CCS}^{\mathsf{cw}}$	CCS with priority due to Camilleri and Winskel
$+\!\rangle$	prioritized-choice operator

$\mathcal{P}^{\mathsf{cw}}$	set of $\mathsf{CCS}^{\mathsf{cw}}$ processes
$\vdash_M^{\mathsf{cw}} \stackrel{\alpha}{\rightarrow}$	transition relation for $\mathsf{CCS}^{\mathsf{cw}}$
$\overline{\mathbb{I}}^{\mathsf{cw}}(P)$	initial output action set for P
$\mathrm{I}^{\mathsf{cw}}(P)$	initial input action set for P
$\mathbb{I}^{\mathsf{cw}}(P)$	initial visible input action set for P (i.e., $\mathrm{I}^{\mathsf{cw}}(P) \setminus \{\tau\}$)
$\simeq_{\mathsf{cw}}$	distributed prioritized strong bisimulation for $\mathsf{CCS}^{\mathsf{cw}}$
$\xi(\cdot)$	translation function

Notation for dynamic priority and global pre-emption (cf. Section ??)

$\mathsf{CCS}^{\mathsf{dg}}$	CCS with dynamic priority and global pre-emption
$\mathcal{A}$	set of actions
$\alpha, \beta, \ldots$	representatives of $\mathcal{A}$
1	clock tick (i.e., one time unit)
γ	representative of $\mathcal{A} \cup \{1\}$
$\stackrel{\alpha}{\longmapsto}$	transition relation for action transitions
$\stackrel{1}{\longmapsto}$	transition relation for clock transitions
$\stackrel{\alpha:k}{\longrightarrow}$	transition relation for dynamic priorities
$\mathrm{I}^k(P)$	potential initial action sets for $\mathsf{CCS}^{\mathsf{dg}}$ wrt. P and priority k
$\sim_{\mathtt{rt}}$	temporal bisimulation
$\sim_{\mathtt{dg}}$	prioritized bisimulation
$[P]^k$	priority adjustment function