Constructing specific SOS semantics for concurrency via abstract interpretation

EXTENDED ABSTRACT

Chiara Bodei[1], Pierpaolo Degano,[1] Corrado Priami[2]

[1] Dipartimento di Informatica, Università di Pisa
Corso Italia 40, I-56100 Pisa, Italy
{chiara,degano}@di.unipi.it
[2] Istituto Policattedra, Università di Verona
Ca' Vignal 2, Strada Le Grazie 1, I-37134 Verona, Italy
priami@sci.univr.it

Abstract. Most of the SOS semantics for concurrent systems can be derived by abstracting on the inference rules of a concrete transition system, namely the proved transition system. Besides the standard interleaving semantics we mechanically derive the causal transition system for CCS, whose definition is particularly difficult and paradigmatic. Its rules are shown to coincide with those presented in the literature. Also, the tree of its computations coincide with that obtained by abstracting the computations of the proved transition system.

Keywords. Concurrency, abstract interpretation, SOS semantics, causality, non-interleaving descriptions.

1 Introduction

We apply the theory of abstract interpretation [10] to calculi for concurrency. We carry out our investigation in a pure setting and consider the basic, foundational calculus CCS [26]. However, we are confident that more powerful calculi, e.g. those for mobile processes, like the π-calculus [27], require only some technical adjustments to the machinery proposed here. The operational semantics of CCS is a transition system, defined following the SOS approach. More in detail, a transition for a process P is deduced applying a set of inference rules, driven by the syntax of P. The original transition system describes the evolution of processes in an interleaving style, and a great deal of work has been devoted to define new transition systems expressing also other aspects of process evolutions, both qualitative and quantitative. Among the former, particularly relevant has been the study of transition systems that express the causality between transitions (see [7,14,24,28,6] to mention only a few of a long list of references). Another well-studied non-interleaving qualitative aspect concerns the description of the localities where processes are placed (among the other proposals, see [8,1,2,29]). Quantitative descriptions include transition systems that express temporal aspects [22,30,19,21,16], probabilistic aspects [34,25], and stochastic ones [20,23,3,9,31]. Besides its interest *in se*, an *SOS* semantics for causality is relevant because it is paradigmatic for others qualitative, non-interleaving description of concurrent systems, as well as for some of the quantitative ones. In particular, many non-interleaving *SOS* definitions have rules in a form similar to those of the causal transition system of [14].

Our starting point is the proved transition system of *CCS* [17], PTS for short. This very concrete representation of systems has been used to describe qualitative and quantitative non-interleaving aspects of concurrent computations [32,33,4]. The transitions are labelled by encodings of their proofs, that encode also most of the aspects briefly mentioned above. The first and the last author used abstract interpretation in [5] to extract the causal computations from the proved computations of the PTS, and sketched how other kinds of computations could be derived, as well. Our main goal here is to push further the use of abstract interpretation in order to mechanically obtain from the PTS the *SOS* inference rules for most of the non-interleaving transition systems mentioned above. We argue that small step operational semantics in *SOS* style for process calculi can be organized in a hierarchy by abstract interpretation, along the lines of abstraction of rule based specifications presented in [11–13].

We work out in full detail the derivation of the causal transition system for *CCS*, possibly the most difficult non-interleaving transition system. We define a (family of) abstraction functions that yield CTS, a variant of the original causal transition system of [14]. As a consequence, the computations of CTS coincide with the abstraction of the computations of PTS, as defined in [5]. The task is easier with Milner's original transition system [26], that we also derive through abstractions from both the proved and the causal transition systems.

These abstractions constitute a fragment of a hierarchy of transition systems. Our approach is constructive, because the causal *SOS* rules are formally derived by abstraction of the proved ones and the same happens for the interleaving ones. This is a first step towards giving evidence of the power of formal methods over empirical ones.

In the conclusions, we briefly sketch the simple modifications needed to cope with the locality transition system, that we omit here for lack of space. We also give hints on the derivation of temporal transition systems.

2 The concrete domain

We recall the pure *CCS* [26] that we use here as a test-bed. As usual, we denote the countable set of atomic actions by Λ, the set of co-actions by $\overline{\Lambda}$ and the invisible actions with τ. Then $\mathcal{A} = \Lambda \cup \overline{\Lambda} \cup \{\tau\}$. We also assume a set of agent identifiers with typical element A. Processes (denoted by $P, Q, R, \ldots \in \mathcal{P}$) are built from actions and agents according to the syntax

$$P ::= \mathbf{0} \mid \mu.P \mid P + P \mid P|P \mid (\nu a)P \mid A$$

where $\mu \in \mathcal{A}$. Hereafter, we omit the trailing $\mathbf{0}$.

The prefix μ is the first atomic action that the process $\mu.P$ can perform. Summation denotes nondeterministic choice. The operator $|$ describes parallel composition of processes. The restriction operator (νa) prevents P from performing a. A is the invocation of the agent identifier A, that has a unique defining equation of the form $A = P$.

Hereafter, we assume that $(\mathcal{P}/\equiv_P, +, \mathbf{0})$ is a commutative monoid. As in [17], the parallel operator $|$ is *neither* associative *nor* commutative, and this is the only difference with the original CCS definition, apart from the omission of the relabelling operator, which is irrelevant to our present study. The transition system of CCS is in Tab. 1.

$$act : \mu.P \stackrel{\mu}{\hookrightarrow} P \qquad\qquad ide : \frac{P \stackrel{\mu}{\hookrightarrow} P'}{A \stackrel{\mu}{\hookrightarrow} P'}, A = P$$

$$par_0 : \frac{P \stackrel{\mu}{\hookrightarrow} P'}{P|Q \stackrel{\mu}{\hookrightarrow} P'|Q} \qquad\qquad sum : \frac{P \stackrel{\mu}{\hookrightarrow} P'}{P + Q \stackrel{\mu}{\hookrightarrow} P'}$$

$$par_1 : \frac{P \stackrel{\mu}{\hookrightarrow} P'}{Q|P \stackrel{\mu}{\hookrightarrow} Q|P'} \qquad\qquad res : \frac{P \stackrel{\mu}{\hookrightarrow} P'}{(\nu a)P \stackrel{\mu}{\hookrightarrow} (\nu a)P'}, \mu \notin \{a, \overline{a}\}$$

$$com : \frac{P \stackrel{\overline{a}}{\hookrightarrow} P', Q \stackrel{a}{\hookrightarrow} Q'}{P|Q \stackrel{\tau}{\hookrightarrow} P'|Q'}$$

Table 1. Transition system for CCS.

As anticipated in the introduction, we enrich the labels of the standard interleaving transition system of CCS in the style of [7,15]. It is thus possible to derive different semantic models for CCS by extracting new kinds of labels from the enriched ones (see [17,18]). We start with the definition of the enriched labels and of a function (ℓ) that takes them to the corresponding action labels.

Definition 1. Let $\vartheta \in \{||_0, ||_1\}^*$. Then *proof terms* (with metavariable $\theta \in \Theta$) are defined by the following syntax

$$\theta ::= \vartheta\mu \mid \vartheta\langle ||_0\vartheta_0\mu_0, ||_1\vartheta_1\mu_1\rangle$$

with $\mu_0 = a$ if and only if $\mu_1 = \overline{a}$, or vice versa.
Function ℓ is defined as $\ell(\vartheta a) = a$ and $\ell(\vartheta\langle ||_0\vartheta_0 a, ||_1\vartheta_1 a'\rangle) = \tau$.

The proved transition system PTS for CCS is in Tab. 2, where the symmetric rule for communication (Com_1) is omitted. These rules are our concrete domain.

$$Act : \mu.P \xrightarrow{\mu} P \qquad Ide : \frac{P \xrightarrow{\theta} P'}{A \xrightarrow{\theta} P'}, A = P$$

$$Par_0 : \frac{P \xrightarrow{\theta} P'}{P|Q \xrightarrow{||_0\theta} P'|Q} \qquad Sum : \frac{P \xrightarrow{\theta} P'}{P + Q \xrightarrow{\theta} P'}$$

$$Par_1 : \frac{P \xrightarrow{\theta} P'}{Q|P \xrightarrow{||_1\theta} Q|P'} \qquad Res : \frac{P \xrightarrow{\theta} P'}{(\nu a)P \xrightarrow{\theta} (\nu a)P'}, \ell(\theta) \notin \{a, \overline{a}\}$$

$$Com_0 : \frac{P \xrightarrow{\vartheta\overline{a}} P', Q \xrightarrow{\vartheta' a} Q'}{P|Q \xrightarrow{\langle ||_0\vartheta\overline{a}, ||_1\vartheta' a\rangle} P'|Q'}$$

Table 2. Proved transition system for CCS.

The proved transition system differs from the one in Tab. 1 in the rules for parallel composition and communication. Rule Par_0 (Par_1) adds to the label a tag $||_0$ ($||_1$) to record that the left (right) component is moving. The rule Com_0 has in its conclusion a pair instead of a τ to record the components which interacted.

The standard interleaving semantics [26] is obtained from the proved one by relabelling each transition through function ℓ in Def. 1. The relabelling is an abstraction from the proved transition system to the interleaving transition system semantics. This result is made precise in Subsection 6.2.

3 The abstract domain

Causal CCS ($CCCS$) is a version of CCS introduced in [14] to give full account to causality relations between transitions. Processes are extended with a family of operators $K ::$, with $K \in \wp_{fin}(\mathbb{N})$. The intended meaning of K in $K :: P$ is to encode the causes of the process P, i.e. the transition occurrences enabling P.

The $CCCS$ processes, called *extended processes* and denoted by $t, t', \ldots \in \mathcal{T}$, are defined according to the syntax

$$t ::= K :: P \mid t|t \mid (\nu\ a)t.$$

We assume that the operators $K ::$ distribute over the restriction (νa) and the parallel operator $|$, i.e. we assume on extended processes the least congruence $\equiv$ that satisfies the following clauses

- $K :: (\nu r)t \equiv (\nu r)K :: t$
- $K :: (t|t') \equiv (K :: t)|(K :: t')$

Also we let $K :: K' :: P = (K \cup K') :: P$; and we write $\mathcal{K}(t)$ for the set of causes in t, k for the singleton $\{k\}$, and $K\{K'/k\}$ for $(K \backslash k) \cup K'$. Moreover, we let $(K :: P)\{K'/k\} = K\{K'/k\} :: P$ and we homomorphically extend this replacement of a set of causes K for a single cause k onto extended processes t.

Causal transitions have the form $t \xmapsto[K;k]{\mu} t'$, where μ is the standard action, $k \in I\!N$ is the unique name of the transition and K represents the set of the transitions that enable k. The causal transition system in Tab. 3 is a variant of the original one of [14] (see also [24]), where the function f is[1]

$$f(K, \mu) = \begin{cases} K; k \text{ if } \mu \neq \tau \\ \emptyset; \emptyset \ \text{ otherwise} \end{cases}.$$

We write $\pi_2(f(K, \mu))$ for the second component of $f(K, \mu)$ and, by abuse of notation, $\pi_2(f(K, \mu)) \notin K$, assuming it true if $\pi_2(f(K, \mu)) = \emptyset$. As usual, we omit in Tab. 3 the rule $Kcom_1$, symmetric to $Kcom_0$.

$$Kact : K :: \mu.P \xmapsto[f(K,\mu)]{\mu} \pi_2(f(K, \mu)) \cup K :: P, \ \pi_2(f(K, \mu)) \notin K$$

$$Kide : \frac{t \xmapsto[f(K,\mu)]{\mu} t'}{t'' \xmapsto[f(K,\mu)]{\mu} t'}, \ t'' = t \qquad Kres : \frac{t \xmapsto[f(K,\mu)]{\mu} t'}{(\nu a)t \xmapsto[f(K,\mu)]{\mu} (\nu a)t'}, \ \mu \notin \{a, \overline{a}\}$$

$$Kpar_0 : \frac{t \xmapsto[f(K,\mu)]{\mu} t'}{t|t'' \xmapsto[f(K,\mu)]{\mu} t'|t''}, \ \ \pi_2(f(K, \mu)) \notin \mathcal{K}(t'') \qquad Ksum : \frac{t \xmapsto[f(K,\mu)]{\mu} t'}{t + t'' \xmapsto[f(K,\mu)]{\mu} t'}$$

$$Kpar_1 : \frac{t \xmapsto[f(K,\mu)]{\mu} t'}{t''|t \xmapsto[f(K,\mu)]{\mu} t''|t'}, \ \ \pi_2(f(K, \mu)) \notin \mathcal{K}(t') \qquad Kcom_0 : \frac{t_0 \xmapsto[K_0;k]{a} t_0', \, t_1 \xmapsto[K_1;k]{\overline{a}} t_1'}{t_0|t_1 \xmapsto[\emptyset;\emptyset]{\tau} t_0'\{K_1/k\}|t_1'\{K_0/k\}}$$

Table 3. Causal transition system for CCS.

4 Rule-based inductive definitions

We first recall some definitions and results from [11]. Then, we instantiate them in our setting.

[1] This function is introduced to avoid duplicating the rules: one set to handle τ-transitions and the other for the visible transitions, as done in [24].

Definition 2. An *inductive definition* is a quadruple $\mathcal{F} = \langle U, \Phi, \bot, \sqcup \rangle$ such that:

- the *universe* U is a set;
- Φ is the set of *rule instances* $\frac{P}{c}$, with $P \subseteq U$ and $c \in U$;
- $\bot \in U$ is the *basis*;
- $\sqcup \in \wp(\wp(U)) \to \wp(U)$, is the (partial) *join* operator and the *induced ordering* $\sqsubseteq$ is $x \sqsubseteq y$ iff $x \sqcup y = y$ and is a partial order on $\wp(U)$.

Definition 3.
The *operator* $\overline{\Phi}$ induced by $\mathcal{F} = \langle U, \Phi, \bot, \sqcup \rangle$ is $\overline{\Phi} : \wp(U) \to \wp(U)$ such that

$$\overline{\Phi}(X) = \{c \in U | \exists P \subseteq X : \frac{P}{c} \in \Phi\}$$

The inductive definition $\mathcal{F}$ is *monotonic* whenever $\overline{\Phi}$ is monotonic and $\bot \sqsubseteq \overline{\Phi}(\bot)$. Moreover $\mathcal{F}$ is *on a cpo* (resp. on a complete lattice) iff $\langle \wp(U), \sqsubseteq, \bot, \sqcup \rangle$ (resp. $\langle \wp(U), \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$) is a cpo (resp. a complete lattice).

Definition 4. Given $\mathcal{F} = \langle U, \Phi, \bot, \sqcup \rangle$ and $\mathcal{F}' = \langle U', \Phi', \bot', \sqcup' \rangle$, we write $\mathcal{F} \overset{\alpha}{\underset{\gamma}{\Longleftrightarrow}} \mathcal{F}'$ for $\langle \wp(U), \sqsubseteq, \bot, \sqcup \rangle \overset{\alpha}{\underset{\gamma}{\Longleftrightarrow}} \langle \wp(U'), \sqsubseteq', \bot', \sqcup' \rangle$, whenever $\langle \alpha, \gamma \rangle$ is a Galois connection, i.e. if and only if

- $\langle \wp(U), \sqsubseteq \rangle$ and $\langle \wp(U'), \sqsubseteq' \rangle$ are posets;
- $\alpha \in \wp(U) \to \wp(U')$ and $\gamma \in \wp(U') \to \wp(U)$;
- $\forall x \in \wp(U), \forall y \in \wp(U')\ :\ [\alpha(x) \sqsubseteq' y]$ if and only if $[x \sqsubseteq \gamma(y)]$.

The following proposition (corresponding to Prop.53 in [11]), will be useful later.

Proposition 5. *Let* $\mathcal{F} = \langle U, \Phi, \bot, \sqcup \rangle$ *be such that*

1. $\mathcal{F}$ *is a monotonic inductive definition on the cpo* $\langle \wp(U), \sqsubseteq, \bot, \sqcup \rangle$*,*
2. $\langle \wp(U'), \sqsubseteq', \sqcup' \rangle$ *be a partial order,*
3. $\alpha \in \wp(U) \to \wp(U')$ *be a complete* $\sqcup$*- and* $\cup$*-morphism such that* $\forall \frac{P}{c} \in \Phi$, $\forall X \subseteq U : \alpha(P) \subseteq \alpha(X)$ *implies* $\exists \frac{P'}{c'} \in \Phi : P' \subseteq X$ *and* $\alpha(\{c\}) \subseteq \alpha(\{c'\})$*.*

Define $\Phi' = \{\frac{\alpha(P)}{c'} \mid \frac{P}{c} \in \Phi,$ *with* $c' \in \alpha(c)\}$ *and* $\bot = \alpha(\bot)$*.*
Then $\mathcal{F}' = \langle U', \Phi', \bot', \sqcup' \rangle$ *is a well formed inductive definition (i.e. a monotonic or extensive definition on a cpo) such that* $\alpha : \mathcal{F} \to \mathcal{F}'$*.*
If moreover $\langle \wp(U), \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ *is a complete lattice then* $\mathcal{F} \overset{\alpha}{\underset{\gamma}{\Longleftrightarrow}} \mathcal{F}'$*, where* $\gamma \in \wp(U') \to \wp(U)$ *is* $\gamma(Y) = \sqcup\{X \in U \mid \alpha(X) \sqsubseteq' Y\}$*.*

More precisely, under the conditions of the above proposition we have the following (for the notation used and more technical details see Def. 46 and Proposition 53 in [11]). The abstraction function α preserves least fixed points *lfp*, because $\mathcal{F} \overset{\alpha[=]}{\underset{\gamma}{\Longleftrightarrow}} \mathcal{F}'$ implies that $\alpha(lfp_{\sqsubseteq}(\overline{\Phi})) = lfp_{\sqsubseteq'}(\overline{\Phi'})$.

In the following we use a special form of inductive definition only, namely the positive one. These definitions are well-formed, hence they have always the fixed point $\overline{\Phi}_{\cup}^{\infty}(\emptyset)$.

Definition 6. The inductive definition $\mathcal{F}_P = \langle U_P, \Phi_P, \bot, \sqcup \rangle$ is *positive*, and is written as $\langle U, \Phi \rangle$, whenever $\bot$ is the emptyset $\emptyset$ and $\sqcup$ is the set union $\cup$.

The set of all the possible transitions involving CCS processes can be generated by a rule-based inductive definition. In this way, instead of reasoning about the classical schemata of rules and axioms, we consider a set Φ of rule instances, where axioms have empty premises. Some of the rules in Tables 2 and 3 have side-conditions, that can also be seen as additional premises. In fact, each rule with a side-condition stands for a set of rules, therefore we can safely use a rule-based inductive definition.

4.1 The concrete domain $\mathcal{F}_P$

Here we give a rule-based inductive definition of the proved transitions of CCS.

Definition 7. The inductive definition $\mathcal{F}_P = \langle U_P, \Phi_P \rangle$ is the positive inductive definition, where:

- $U_P = \{P \xrightarrow{\theta} P' | P, P' \in \mathcal{P}, \theta \in \Theta\}$ is the universe, and
- $\Phi_P = \{\frac{P}{c} \mid P \subseteq U_P, c \in U_P$ and $\frac{P}{c}$ is an instance of $Act, Ide, Res, Sum, Par_0, Par_1, Com_0, Com_1\}$ is the set of rule instances.

The following properties of $\mathcal{F}_P$ are easy to establish.

Proposition 8. *The positive inductive definition $\mathcal{F}_P$ is monotonic and it is well-formed.*

Proof. We only need to prove that the operator $\overline{\Phi}_P$ is monotonic and that $\emptyset \subseteq \overline{\Phi}_P(\emptyset)$. Let $X \subseteq Y \subseteq U_P$. Since $c \in \overline{\Phi}_P(X)$ implies $\exists P \subseteq X \;:\; \frac{P}{c} \in \Phi_P$ and $X \subseteq Y$, then $c \in \overline{\Phi}_P(Y)$. Finally, $\emptyset \subseteq \overline{\Phi}_P(\emptyset)$ holds because $\overline{\Phi_P}(\emptyset)$ is a set.

Proposition 9. $\langle \wp(U_P), \subseteq, \emptyset, U_P, \cup, \cap \rangle$ *is a complete lattice.*

4.2 The abstract domain $\mathcal{F}_C$

The set of all the possible causal transitions involving $CCCS$ processes can be generated by a rule-based positive inductive definition, as well.

Definition 10. $\mathcal{F}_C = \langle U_C, \Phi_C \rangle$ is the positive inductive definition, where:

- $U_C = \{t \underset{f(K,\mu)}{\overset{\mu}{\longmapsto}} t' \mid t, t' \in \mathcal{T}, \mu \in Act, K \in \wp_{fin}(I\!N)\}$ is the universe;
- $\Phi_C = \{\frac{P}{c} \mid P \subseteq U_C, c \in U_C$ and $\frac{P}{c}$ is an instance of$Kact, Kide, Kres, Ksum, Kpar_0, Kpar_1, Kcom_0, Kcom_1\}$ is the set of rule instances.

Just as for $\mathcal{F}_P$, we have the following.

Proposition 11. *The positive inductive definition $\mathcal{F}_C$ is monotonic and well-formed.*

Proposition 12. $\langle \wp(U_C), \subseteq, \emptyset, U_C, \cup, \cap \rangle$ *is a complete lattice.*

5 The abstract interpretation function

To abstract inductive definitions, we use an abstraction of subsets of the universe, according to Proposition 5. Our abstraction is based on a family of abstraction functions, which act on the components of the rule schemata and which are parameterized on a class S defined according to the syntax

$$S ::= K \mid (S|S).$$

We write S^i for the i^{th} (from the left) set $K \in \wp_{fin}(\mathbb{N})$ of (the frontier of the tree) S. For instance if $S = (K|(K'|K''))|(H|H')$, then $S^4 = H$.
The family of functions α_S is such that

- $\alpha_K(P) = K :: P$
- $\alpha_{S_0|S_1}(P \mid Q) = \alpha_{S_0}(P) \mid \alpha_{S_1}(Q)$.

Moreover we assume that

- $K :: \alpha_H(P) = \alpha_{K \cup H}(P)$;
- $K :: \alpha_{S_0|S_1}(P \mid Q) = K :: \alpha_{S_0}(P) \mid K :: \alpha_{S_1}(Q)$;
- $(S_0 \mid S_1)\{H/k\} = S_0\{H/k\} \mid S_1\{H/k\}$.

As an example of how an abstraction function works, consider the S introduced above and the process $R = (P|(P'|P''))|(a.Q|Q')$. Then

$$\begin{aligned}
\alpha_S(R) &= \alpha_{(K|(K'|K''))}(P|(P'|P'')) \mid \alpha_{(H|H')}(a.Q|Q') = \\
&\alpha_K(P) \mid (\alpha_{K'}(P') \mid \alpha_{K''}(P'')) \mid (\alpha_H(a.Q) \mid \alpha_{H'}(Q')) = \\
&(K :: P \mid (K' :: P' \mid K'' :: P'')) \mid (H :: a.Q \mid H' :: Q') = t_R.
\end{aligned}$$

Now, we can define the abstraction function α_c^p that maps proved transitions in causal transitions. It will be used to obtain the causal transition system by abstracting the rules of the proved one.

Definition 13. Let $\alpha_c^p : \wp(U_P) \to \wp(U_C)$ be defined as

$$\begin{aligned}
\alpha_c^p(X) = &\{\alpha_c^p(P \xrightarrow{\theta\tau} P') = \alpha_S(P) \xrightarrow[\emptyset;\emptyset]{\tau} \alpha_S(P') \mid P \xrightarrow{\vartheta\tau} P' \in X\} \cup \\
&\{\alpha_c^p(P \xrightarrow{\theta\langle\vartheta a,\vartheta' a'\rangle} P') = \alpha_{S_0}(P) \xrightarrow[\emptyset;\emptyset]{\tau} \alpha_{S_1}(P') \mid P \xrightarrow{\vartheta\langle\vartheta' a',\vartheta'' a''\rangle} P' \in X, \\
&\quad \exists i,j : S_1^j = S_1^i = S_0^j \cup S_0^i \text{ and } \forall p \neq i,j : S_1^p = S_0^p\} \cup \\
&\{\alpha_c^p(P \xrightarrow{\vartheta a} P') = \alpha_{S_0}(P) \xrightarrow[K;k]{a} \alpha_{S_1}(P') \mid P \xrightarrow{\vartheta a} P' \in X, \forall p : k \notin S_0^p \text{ and} \\
&\quad \exists i, K : S_0^i = K, S_1^i = K \cup \{k\},\ \forall j \neq i : S_1^j = S_0^j\}
\end{aligned}$$

To see our definition at work, consider again the α_S and R introduced above, and the following proved transition $R \xrightarrow{||_1||_0 a} R' = (P|(P'|P''))|(Q|Q')$. Then, its abstraction is $t_R \xmapsto[H;h]{a} t_{R'}$, for $h \notin K \cup K' \cup K'' \cup H \cup H'$ and $S' = (K|(K'|K''))|(H \cup \{h\}|H')$, where

$t_{R'} = \alpha_{S'}(R') = (K :: P \mid (K :: P' \mid K'' :: P'')) \mid (H \cup \{h\} :: Q \mid H' :: Q')$.

It is easy to deduce this causal transition with the rules in Tab. 3. The interested reader may wish to abstract step by step its whole proved derivation. We wish to

$$\alpha_r(Act) = \alpha_K(\mu.P) \xrightarrow[f(K,\mu)]{\mu} \pi_2(f(K,\mu)) :: \alpha_K(P), \quad \pi_2(f(K,\mu)) \notin K$$

$$\alpha_r(Ide) = \frac{\alpha_S(P) \xrightarrow[f(K,\ell(\theta))]{l(\theta)} \pi_2(f(K,\ell(\theta))) :: \alpha_{S'}(P')}{\alpha_S(A) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'}(P')}, \ \alpha_S(A) = \alpha_S(P)$$

$$\alpha_r(Res) = \frac{\alpha_S(P) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'}(P')}{\alpha_S(\nu a)P) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'}((\nu a)P')}, \ \ell(\theta) \notin \{a, \overline{a}\}$$

$$\alpha_r(Sum) = \frac{\alpha_S(P) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'}(P')}{\alpha_S(P+Q) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'}(P')}$$

$$\alpha_r(Par_0) = \frac{\alpha_{S_0}(P) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'_0}(P')}{\alpha_{S_0|S_1}(P \mid Q) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'_0|S_1}(P' \mid Q)}, \quad \forall i: \ \pi_2(f(K,\ell(\theta))) \notin S_1^i$$

$$\alpha_r(Par_1) = \frac{\alpha_{S_1}(Q) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S'_1}(Q')}{\alpha_{S_0|S_1}(P \mid Q) \xrightarrow[f(K,\ell(\theta))]{\ell(\theta)} \alpha_{S_0|S'_1}(P \mid Q')}, \quad \forall i: \ \pi_2(f(K,\ell(\theta))) \notin S_0^i$$

$$\alpha_r(Com_0) = \frac{\alpha_{S_0}(P_0) \xrightarrow[f(K_0;a)]{a} \alpha_{S'_0}(P'_0), \alpha_{S_1}(P_1) \xrightarrow[f(K_0;\overline{a})]{\overline{a}} \alpha_{S'_1}(P'_1)}{\alpha_{S_0|S_1}(P_0 \mid P_1) \xrightarrow[\emptyset;\emptyset]{\tau} \alpha_{S'_0\{K_1/k\} \ \mid \ S'_1\{K_0/k\}}(P'_0 \mid P'_1)}$$

Table 4. The abstraction interpretation α_r of rule instances.

use Proposition 5 to establish a Galois connection between the definitions of the proved and the causal transition systems. We already know that the abstract domain is a complete lattice (Proposition 12). We are left to show that our abstraction function α_c^p enjoys the properties required by item 3 of Proposition 5.

Proposition 14. *The function α_c^p a complete $\cup$-morphism,*

$$i.e.\ \forall X \subseteq U_P \ : \ \cup X \in U_P \quad implies \quad \cup\, \alpha_c^p(X) = \alpha_c^p(\cup X).$$

Proof. It is easy to prove that the function α_c^p is monotone. We now prove the double inclusion $\subseteq$ and $\supseteq$ (note that induced ordering is the same for both domains).
We start with $\subseteq$. By definition of $\cup$, $\forall X \subseteq U_P, \alpha_c^p(X) \subseteq \cup\alpha_c^p(X)$ and $X \subseteq \cup X$. Then, by monotonicity of α_c^p, it is $\alpha_c^p(X) \subseteq \alpha_c^p(\cup X)$. Since $\cup$ is the *least* upper bound, $\cup' \alpha_c^p(X) \subseteq \alpha_c^p(\cup X)$.

We now prove $\supseteq$. Let FK be the function introduced in Def. 25 that deletes the causes from an extended process, restoring a pure CCS process.

By hypothesis $\cup\alpha_c^p(X) = \{t_i \xrightarrow[K_i;k_i]{\mu_i} t_i' \mid i \in I\} \cup \{t_j \xrightarrow[\emptyset;\emptyset]{\tau} t_j' \mid j \in J\} \subseteq U_C$. Then,

by construction, there exists a set $Y = \{P_p \xrightarrow{\theta_p} P_p' \mid FK(t_p) = P_p, FK(t_p') = P_p', \ell(\theta_p) = \mu_p \mid p \in I \cup J\}$, such that $\alpha_c^p(Y) = \cup\alpha_c^p(X)$. First we prove $\forall \tilde{Y} \subseteq X : \tilde{Y} \subseteq Y$. Per absurdum, let $S \subseteq X$ such that $S \supset Y$. Then, by monotonicity of α_c^p, $\alpha_c^p(S) \supseteq \alpha_c^p(Y) = \cup\alpha_c^p(X)$, against the hypothesis of $\cup\alpha_c^p(X)$ to be the least upper bound. Then, $\forall \tilde{Y} \subseteq X : \tilde{Y} \subseteq \cup X$, as well. From this and from $\tilde{Y} \subseteq Y$, $\cup X \subseteq Y$, because $\cup X$ is the *least* upper bound. By monotonicity of α_c^p, we conclude $\alpha_c^p(\cup X) \subseteq \alpha_c^p(Y) = \cup\alpha_c^p(X)$.

Proposition 15. *The function α_c^p is such that $\forall \frac{P}{c} \in \Phi_P$, $\forall X \subseteq U_P$: $\alpha_c^p(P) \subseteq \alpha_c^p(X)$ implies $\exists \frac{P'}{c'} \in \Phi_P : P' \subseteq X$ and $\alpha_c^p(\{c\}) \subseteq \alpha_c^p(\{c'\})$.*

Proof. Since $\alpha_c^p(P) \neq \alpha_c^p(P')$, whenever $P \neq P'$, $\alpha_c^p(P) \subseteq \alpha_c^p(X)$ implies $P \subseteq X$. Hence, we only need to put $P = P'$.

To define the abstract interpretation of the operator Φ, we introduce in Tab. 4 the auxiliary family of functions α_r. Note that $\alpha_r(\frac{P}{c}) = \frac{\alpha_c^p(P)}{c'}$ with $c' \in \alpha_c^p(c)$ (cf. the definition of Φ' in Proposition 5). We can now establish the required Galois connection, that preserves least fixed points.

Theorem 16. *Let $\Phi' = \{\alpha_r(\frac{P}{c}) \mid \frac{P}{c} \in \Phi_P\}$ and $\emptyset = \alpha_r(\emptyset)$. The positive inductive definition $\mathcal{F}' = \{\alpha_c^p(U), \Phi'\}$ is well formed and such that $\mathcal{F}_\mathcal{P} \underset{\gamma_c^p}{\overset{\alpha_p^c[=]}{\longleftrightarrow}} \mathcal{F}'$, i.e. $\langle \alpha_c^p, \gamma_p^c \rangle$ is a Galois connection, where $\gamma_p^c : \wp(U_C) \to \wp(U_P)$ is*

$$\gamma_p^c(Y) = \begin{cases} \cup\{X \in U_P \mid \alpha_c^p(X) \subseteq Y\} & \textit{if } Y \in \alpha_c^p(U_P) \\ U_P & \textit{if } Y \notin \alpha_c^p(U_P) \end{cases}$$

Proof. By Proposition 5, whose hypotheses are satisfied because of Propositions 8, 9, 11, 12, 14, 15.

Although $\alpha_c^p(U_P) \subset U_C$, we have established exactly what we would like to obtain. In fact, it is immediate to prove the following property, because the rules in Tables 3 and 4 can be easily put in bijection.

Proposition 17. $\alpha_r(\Phi_P) = \Phi_C$.

The above correspondence suffices to establish that the computations of the causal transition system obtained via abstraction α_c^p coincide with the abstraction of the proved computations as defined in [5]. Actually, a stronger result holds: the trees of computations coincide, so the non deterministic aspects of computations are preserved, as well. Lack of space prevents us from recalling the relevant definitions and results of [5]. We simply introduce some notation.

Let $Ptree(P)$ be the tree defined as the set of the proved computations starting from P, ordered by prefix. Similarly, the tree of causal computations is $Ktree(P)$, with transitions derived by the rules in Tab. 3; and finally let $\alpha_c^p Ptree(P)$ be the tree of computations with transitions deduced according to $\alpha_c^p(\Phi_P)$. We denote by $\langle\overline{\alpha}, \overline{\gamma}\rangle$ the Galois connection of Def. 17 in [5], where $\overline{\alpha}$ maps the tree of proved computations $Ptree(P)$ to the tree of causal computations $Ktree(P)$. Now, we can state the following.

Theorem 18. *For all $P \in \mathcal{P}$, we have that $\alpha_c^p Ptree(P) = Ktree(P)$.*
Proof. The bijection between Tab. 3 and Tab. 4 suffices, because $\overline{\alpha}(Ptree(P)) = Ktree(P)$ by Theorem 18 of [5].

6 A fragment of a hierarchy

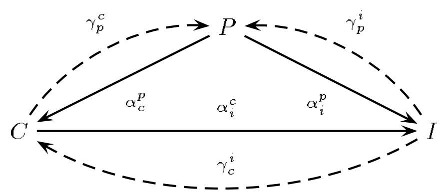

Fig. 1. The hierarchy of semantics.

We now define a simple hierarchy of transition systems involving the proved, the causal and the standard interleaving semantics of CCS. The hierarchy in Fig. 1 is established by studying three abstraction functions: from the proved to the causal model and to the interleaving model, and from the causal to the interleaving one. This fragment of a hierarchy is formally constructed, and in the conclusions we sketch how other models can be derived by further constructive abstractions of the proved model.

6.1 The abstract domain $\mathcal{F}_I$

We start with the definition of the abstract domain made of the standard interleaving transition system of CCS [26]. The set of all the possible transitions involving CCS processes can be generated by the following positive rule-based inductive definition.

Definition 19. $\mathcal{F}_I = \langle U_I, \Phi_I \rangle$ is the positive inductive definition, where:

- $U_I = \{P \overset{\mu}{\hookrightarrow} P' \mid P, P' \in \mathcal{P}, \mu \in Act\}$ is the universe;
- $\Phi_I = \{\frac{P}{c} \mid P \subseteq U_I, c \in U_I$ and $\frac{P}{c}$ is an instance of $act, ide, res, sum, par_0, par_1, com_0, com_1\}$ is the set of rule instances.

Just as for $\mathcal{F}_C$, we have the following.

Proposition 20. *The positive inductive definition $\mathcal{F}_I$ is monotonic and well-formed.*

Proposition 21. $\langle \wp(U_I), \subseteq, \emptyset, U_I, \cup, \cap \rangle$ *is a complete lattice.*

6.2 The abstraction from proved to interleaving transition systems

We make now precise the observation made at the end of Section 2, by defining a Galois connection between $\mathcal{F}_P$ and $\mathcal{F}_I$. So one can easily recover the standard interleaving transition system from the proved one.

Definition 22. Let $\alpha_i^p : \wp(U_P) \to \wp(U_I)$ be defined as
$\alpha_i^p(X) = \{\alpha_i^p(P \xrightarrow{\theta} P') = P \overset{\ell(\theta)}{\hookrightarrow} P' \mid P \xrightarrow{\theta} P' \in X\}$, where ℓ is as in Def. 1.

Also in this case is possible to reformulate the results relative to the Galois connection and the consequent correspondences. Note again that $\alpha_r''(\frac{P}{c}) = \frac{\alpha_i^p(P)}{c'}$ with $c' \in \alpha_i^p(c)$.

Theorem 23. *Let $\Phi'' = \{\alpha_r''(\frac{P}{c}) \mid \frac{P}{c} \in \Phi_P\}$ and $\emptyset = \alpha_r''(\emptyset)$. The positive inductive definition $\mathcal{F}'' = \{\alpha_i^p(U_P), \Phi''\}$ is well formed and is such that $\mathcal{F}_P \underset{\gamma_p^i}{\overset{\alpha_i^p[=]}{\Longleftrightarrow}} \mathcal{F}'_I$, i.e. $\langle \alpha_i^p, \gamma_p^i \rangle$ is a Galois connection, where $\gamma_p^i : \wp(U_I) \to \wp(U_P)$ is*

$$\gamma_p^i(Y) = \begin{cases} \cup\{X \in U_P \mid \alpha_i^p(X) \subseteq Y\} & \text{if } Y \in \alpha_i^p(U_P) \\ U_P & \text{if } Y \notin \alpha_i^p(U_P) \end{cases}$$

The following proposition says that the induced operator Φ'' above is indeed Φ_I, and justifies the use of $\overset{\mu}{\hookrightarrow}$ in Def. 22.

Proposition 24. $\alpha_r''(\Phi_P) = \Phi_I$.

6.3 The abstraction from causal to interleaving transition systems

Our next abstraction is the function α_i^c that maps causal transitions in standard interleaving transitions. It will be used to obtain the interleaving transition system by abstracting the rules of the causal one. The steps are the same as above.

Definition 25. Let $\alpha_i^c : \wp(U_C) \to \wp(U_I)$ be defined as

$$\alpha_i^c(X) = \{\alpha_i^c(t \xrightarrow[f(K,\mu)]{\mu} t') = FK(t) \stackrel{\mu}{\hookrightarrow} FK(t') \mid t \xrightarrow[f(K,\mu)]{\mu} t' \in X\}.$$

where FK deletes the causes from extended processes, defined as
$FK(K :: t) = FK(t)$, $FK(t|t') = FK(t)|FK(t')$, $FK((\nu a)t) = (\nu a)FK(t)$.

Theorem 26. *Let $\Phi''' = \{\alpha_r'''(\frac{P}{c}) \mid \frac{P}{c} \in \Phi_C\}$ and $\emptyset = \alpha_r'''(\emptyset)$. The positive inductive definition $\mathcal{F}''' = \{\alpha_i^c(U_C), \Phi'''\}$ is well formed and is such that $\mathcal{F}_C \overset{\alpha_i^c[=]}{\underset{\gamma_c^i}{\Longleftrightarrow}} \mathcal{F}'''_I$, i.e. $\langle \alpha_i^c, \gamma_c^i \rangle$ is a Galois connection, where $\gamma_c^i : \wp(U_I) \to \wp(U_C)$ is*

$$\gamma_c^i(Y) = \begin{cases} \cup\{X \in U_C \mid \alpha_i^c(X) \subseteq Y\} & \text{if } Y \in \alpha_i^c(U_C) \\ U_C & \text{if } Y \notin \alpha_i^c(U_C) \end{cases}$$

Note that $FK(t)$ is a standard CCS process and the following proposition justifies the presence of $\stackrel{\mu}{\hookrightarrow}$ in Def. 25.

Proposition 27. $\alpha_r'''(\Phi_C) = \Phi_I$.

6.4 The hierarchy

The commutativity of the diagram in Fig. 1 is now immediate.

Theorem 28. $\alpha_i^c(\alpha_c^p(\Phi_P)) = \alpha_i^p(\Phi_P) = \Phi_I$.

The above theorem suffices to show a correspondence between the trees of computations in the various models. The trees of computations obtained by abstracting proved computations to interleaving ones coincide with the trees of the computations based on the standard interleaving semantics. The same happens abstracting causal computations. Therefore, the non deterministic aspects of agent behaviour are fully respected. This supports our claim that formal constructive methods can be used to mechanically derive, clarify and compare various different models for concurrency.

Let $Stree(P)$ be the synchronization tree defined as the set of the interleaving CCS computations starting from P, ordered by prefix [26]. Also, let $\alpha_i^p Ptree(P)$ and $\alpha_i^c Ktree(P)$ be the trees of computations with transitions deduced with $\alpha_i^p(\Phi_P)$ and with $\alpha_i^c(\Phi_C)$. The following is immediate.

Theorem 29. *For all $P \in \mathcal{P}$, $\alpha_i^p Ptree(P) = Stree(P) = \alpha_i^c Ktree(P)$.*

7 Conclusions

We have constructed the causal and the interleaving transition systems for CCS by abstracting on the rules of the proved transition system. Lack of space prevents us from giving more evidence that quite similar constructions are possible, and give most of the interleaving and the non-interleaving SOS definitions presented in the literature for concurrency calculi. Their designers aimed at describing different qualitative and quantitative aspects of process evolutions, and a considerable effort has been spent in their actual *SOS* definitions.

Our proposal may help in mechanically deriving these *SOS* definitions, keeping in mind only the effect that a *single* transition has on the state of the whole system of concurrent processes. Consider for instance the effect that a visible action a has on the α_S in Def. 13: only the causes K of the process performing a are affected — this obvious fact, together with exchange of causes in communications, are sufficient to obtain the rules for the causal transition system.

Despite space limitation, we would like to give some hints of three more abstraction functions, thus showing that our construction is in a sense paradigmatic.

Consider the location transition system of [8]. It was designed to keep track of the sites (taken from a given set *Location*) where the visible actions actually occur, and it assumes that the sites of invisible ones cannot be detected. Having this in mind, it is easy to define the location abstraction functions α_l^p, mimicking Def. 13, as follows.

$$\alpha_l^p(X) = \{\alpha_l^p(P \xrightarrow{\theta} P') = \alpha_L(P) \xrightarrow{\tau} \alpha_L(P') \mid P \xrightarrow{\theta} P' \in X, \ell(\theta) = \tau\} \;\cup$$
$$\{\alpha_l^p(P \xrightarrow{\vartheta a} P') = \alpha_L(P) \xrightarrow[lu]{a} \alpha_{L'}(P') \mid P \xrightarrow{\vartheta a} P' \in X,$$
$$\forall p : l \notin L_0^p \text{ and } \exists i : L_0^i = u, L_1^i = lu, \text{ and } \forall j \neq k : L_1^j = L_0^j\},$$

where $l \in Location, u \in Location^*$ and L is either u or $L|L$ (i.e. L is a tree of strings of locations, rather than a tree of sets of natural numbers), and α_L is a family of abstraction functions defined just as α_S.

Indeed, the only difference between Def. 13 and the one sketched above is that communications do not exchange causes, a typical feature of causality, not shared by location models. It is easy then to establish a Galois connection $\langle \alpha_l^p, \gamma_p^l \rangle$ and results similar to Theorems 16 and 18.

At this point, the reader familiar with the local/global transition system of [24] can easily obtain it by defining an abstraction function $\alpha_{l/g}^p$ that combines α_c^p and α_l^p and that enjoys all the properties these abstraction functions have.

As a third example, we consider a *SOS* semantics that expresses quantitative aspects of concurrent computations. Consider the timed transition system of [19], whose transitions have labels of the form $\langle a, n \rangle \star w$, where n is the time required to complete the standard action a at location w. We can easily modify α_l^p to obtain the abstraction function α_t^p that renders the considered timed transition systems. Assume as given a function $time(a)$ that assigns a duration to actions (we only consider visible actions, because the calculus of [19] has no τ's, and we slightly change their notion of location). This abstraction function is as follows. $\alpha_t^p(X) = \{\alpha_t^p(P \xrightarrow{\vartheta a} P') = \alpha_L(P) \xrightarrow{\langle a,n \rangle \star w} \alpha_{L'}(P') \mid P \xrightarrow{\vartheta a} P' \in X, \forall p : l \notin L_0^p \text{ and } \exists i : L_0^i = u, L_1^i = lu, \text{ and } \forall j \neq k : L_1^j = L_0^j, n = time(a)\}$.

We have not yet considered transition systems expressing other qualitative aspects of concurrent processes. However, we are very confident that most of them can be cast in our framework, and that abstract interpretation is a flexible tool for constructing specific *SOS* semantics for concurrency calculi, including those for mobility, like the π-calculus [27].

Acknowledgment. The authors wish to thank the referees for their precise remarks and helpful comments.

References

1. L. Aceto. A static view of localities. *Formal Aspects of Computing*, 1992.
2. R.M. Amadio and S. Prasad. Localities and failures. In *Proceedings of FST-TCS'94*, pages 205–216. Springer-Verlag, 1994.
3. M. Bernardo, L. Donatiello, and R. Gorrieri. Integrating performance and functional analysis of concurrent systems with EMPA. Technical Report UBLCS-95-14, University of Bologna, Laboratory for Computer Science, 1995.
4. C. Bodei, P. Degano, and C. Priami. Mobile processes with a distributed environment. In *Proceedings of ICALP'96, LNCS 1099*, pages 490–501. Springer-Verlag, 1996. To appear in TCS.
5. C. Bodei and C. Priami. True concurrency via abstract interpretation. In *Proceedings of SAS'97, LNCS 1302*, pages 202–216, 1997.
6. M. Boreale and D. Sangiorgi. A fully abstract semantics of causality in the π-calculus. In *Proceedings of STACS'95, LNCS*. Springer Verlag, 1995.
7. G. Boudol and I. Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fundamenta Informaticae*, XI(4):433–452, 1988.
8. G. Boudol, I. Castellani, M. Hennessy, and A. Kiehn. A theory of processes with localities. *Theoretical Computer Science*, 114, 1993.
9. P. Buchholz. On a markovian process algebra. Technical report, Informatik IV, University of Dortmund, 1994.
10. P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, 1992.
11. P. Cousot and R. Cousot. Inductive definitions, semantics and abstract interpretation. In *Proceedings of POPL'92*, pages 83–94, 1992.
12. P. Cousot and R. Cousot. Higher-order abstract interpretation (and application to comportment analysis generalizing strictness, termination, projection and PER analysis of functional languages). In *Procs. ICCL'94, IEEE*, pages 95–112, 1994.
13. P. Cousot and R. Cousot. Compositional and inductive semantic definitions in fixpoint equational, constraint, closure-condition, rule-based and game-theoretic form. In *Proceedings of CAV'95, LNCS 939*, pages 293–308, 1995.
14. Ph. Darondeau and P. Degano. Causal trees. In *Proceedings of ICALP'89, LNCS 372*, pages 234–248. Springer-Verlag, 1989.
15. P. Degano, R. De Nicola, and U. Montanari. Partial ordering derivations for CCS. In *Proceedings of FCT, LNCS 199*, pages 520–533. Springer-Verlag, 1985.
16. P. Degano, J.-V. Loddo, and C. Priami. Mobile processes with local clocks. In *Proceedings of Workshop on Analysis and Verification of Multiple-Agent Languages*, Stockholm, Sweden, 1996.
17. P. Degano and C. Priami. Proved trees. In *Proceedings of ICALP'92, LNCS 623*, pages 629–640. Springer-Verlag, 1992.
18. P. Degano and C. Priami. Non interleaving semantics for mobile processes. Extended abstract. In *Proceedings of ICALP'95, LNCS 944*, pages 660–671. Springer-Verlag, 1995. To appear in TCS.

19. R. Gorrieri, M. Roccetti, and E. Stancapiano. A theory of processes with durational actions. *Theoretical Computer Science*, (140), 1994.
20. N. Götz, U. Herzog, and M. Rettelbach. TIPP- a language for timed processes and performance evaluation. Technical Report 4/92, IMMD VII, University of Erlangen-Nurnberg, 1992.
21. E. Goubault. Durations for truly concurrent transitions. In *Proceedings of ESOP96, LNCS 1058*, pages 173–187, 1995.
22. M. Hennessy and T. Regan. A temporal process algebra. Technical Report 2/90, University of Sussex, 1990.
23. J. Hillston. The nature of synchronization. In U. Herzog and M. Rettelbach, editors, *Proceedings of PAPM'94*, University of Erlangen, 1994.
24. A. Kiehn. Comparing causality and locality based equivalences. *Acta Informatica*, 31(8):697–718, 1994.
25. K.G. Larsen and A. Skou. Compositional verification of probabilistic processes. In *Proceedings of CONCUR'92*, volume 630 of *LNCS*. Springer-Verlag, 1992.
26. R. Milner. *Communication and Concurrency*. Prentice-Hall, London, 1989.
27. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (I and II). *Information and Computation*, 100(1):1–77, 1992.
28. U. Montanari and M. Pistore. Concurrent semantics for the π-calculus. In *Electronic Notes in Computer Science*, number 1. Elsevier, 1995.
29. U. Montanari and D. Yankelevich. A parametric approach to localities. In *Proceedings of ICALP'92, LNCS 623*, pages 617–628. Springer-Verlag, 1992.
30. X. Nicollin and J. Sifakis. An overview and synthesis on timed process algebras. In *Real Time: Theory in Practice, LNCS 600*, pages 526–548. Springer-Verlag, 1991.
31. C. Priami. Stochastic π-calculus. *The Computer Journal*, 38(6):578–589, 1995.
32. C. Priami. *Enhanced Operational Semantics for Concurrency*. PhD thesis, Dipartimento di Informatica, Università di Pisa, March 1996. Available as Tech. Rep. TD-08/96.
33. C. Priami. Interleaving-based partial odering semantics. In *Proceedings of Italian Conference on Theoretical Computer Science*, pages 264–278, Ravello, November 1995, 1996. World Scientific.
34. R.J. van Glabbeek, S.A. Smolka, B. Steffen, and C.M.N. Tofts. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 1995.