Barbed Bisimulation

Robin Milner Davide Sangiorgi

Laboratory for Foundations of Computer Science
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ, U.K.

1 Introduction

Process algebras have emerged as key model to reason about nondeterministic and concurrent computations. The *transition semantics* on which they are based is given by defining the appropriate congruence on the top of the *labelled* transition system describing the possible evolutions of a process.

This contrasts with what happens in term rewriting systems, as based on a *reduction semantics* where the equivalence is defined on the top of an *unlabelled* transition system (the *reduction system*). It is indeed the simplicity of their semantics which makes term rewriting systems attractive, and in turn facilitates a satisfactory mathematical analysis.

In the lambda calculus, probably the most well-known term rewriting system, what makes a reduction system possible is that two terms having to interact are naturally in contiguous positions. This is not the case in process calculi, where interaction is not dependent upon physical contiguity. To put this another way, a *redex* of a lambda term is a subterm, while a "redex" in a process calculus is distributed over the term.

A guideline for the definition of reduction systems in process algebras is offered by Milner [11], inspired by Berry and Boudol' Chemical Abstract Machine [8]. In this technique, axioms for a *structural congruence relation* are introduced prior to the reduction system, in order to to break a rigid, geometrical vision of concurrency; then reduction rules can easily be presented in which redexes are indeed subterms again. It can then be shown that modulo structural congruence the reduction relation exactly represents the silent action of the transition semantics.

It is left as an open problem in [11] how to recover from such a formulation the familiar congruences which are based upon a labelled transition system. It turns out that this is not a trivial problem. We tackle it in this paper for the simple case of CCS and strong observational equivalence ($\sim$). Because the reduction relation coincides with the silent action $\stackrel{\tau}{\longrightarrow}$ of the labelled transition system (as mentioned above), we can remain within the latter framework. But we wish to retain the spirit of the reduction semantics as far as possible, in the sense that we wish to equip a global observer with a minimal ability to observe actions and/or process states. We then obtain an equivalence, namely indistinguishability under global observations. This in turn induces a congruence over agents, namely equivalence in all contexts. The question is: what minimal power of observation characterizes in this way the congruences with which we are already familiar?

If we look for a characterization of $\sim$ in this way, then it is reasonable to examine first the case in which silent actions are the *only* observables. Unfortunately the *reduction congruence* which is so obtained is in general less discriminatory than $\sim$ (it is the same as $\sim$ only for the divergence-free processes). Therefore it is necessary to add the capability of observing some more properties of the states. It seems natural in concurrency that the extra power provided is in term of action observability; in this sense the minimal capability is given by the adoption of a convergency predicate detecting the possibility of performing *some* visible actions. It is therefore a "may" convergency predicate, in this sense the same used by Boudol in [7]. We call the resulting bisimulation relation and its induced congruence respectively *barbed bisimulation* and *barbed congruence*. This paper is mainly devoted to proving that barbed congruence and $\sim$ coincide.

The barbed bisimulation strongly recalls Abramsky's applicative bisimulation for the lambda calculus [9], as both are defined in terms of reduction and convergency. In fact the generality of such predicates, which are independent of the underlying transition system, allows one to use barbed bisimulation as a tool to describe *uniformly* bisimulation-based equivalences in calculi which syntactically may be quite far from each other. This is useful in different aspects. First of all it helps a lot in understanding the relationship between two different calculi. It is by using barbed bisimulation, for instance, that we were able to conduct the study in [4] on the expressive power of the π-calculus for modeling lambda calculus and higher-order π-calculus .

Barbed equivalence also provides for free any "new" calculus with an important equivalence; it is then an excellent test for the calculus itself and its operators to find out whether barbed congruence can be characterized in a direct way, not involving quantification over all contexts.

An important point emerging in [11] is that in some cases a reduction semantics may be much more enlighting and simple than a transition semantics. Our study in [4] suggests that in higher-order process calculi such benefits can become decisive.

Structure of the paper. We review the basic concepts of pure CCS in section 2; we study the reduction bisimulation/congruence and the barbed bisimulation/congruence respectively in sections 3 and 4; in section 5 we look at other issues and uses of the barbed bisimulations.

2 Background

We review here the syntax and the operational semantics of 'pure' CCS [10] together with the concept of strong bisimulation.

Let $N = \{a, b, \ldots\}$ be a set of names, $\Lambda = \{a, \overline{a} | a \in \Lambda\}$ be the set of labels built on N and $Act = \Lambda \cup \{\tau\}$ where τ is a distinguished action (the silent action) not in Λ . We use the convention that if $a \in N$, $\overline{\overline{a}} = a$ and we let ℓ to range over Λ and α over Act.

We will also use a set Ψ of variables ranged over by X and a set Φ of agent constant ranged over by A.

The class of the CCS expressions with names in N is the smallest class $\Im$ which includes Ψ and Φ and such that contains the following expressions, where E, E_i are already in $\Im$:

$\alpha.E$, $E \setminus L\ (L \subseteq N)$, $E_1 | E_2$, $\sum_{i \in I} E_i$ (I is an indexing set)

It is supposed that for each constant A there is a defining equation of the form $A \stackrel{def}{=} E$, $E \in \Im$.

A CCS expression with no free variable in it is a *process*. In the following P, Q, R, T represent processes. A CCS expression with a variable in it is a *context*. Given a context C and a process P, we write $C(P)$ to denote the closed term obtained by substituting P for the variable occurring in C.

$$\frac{}{\alpha.E \xrightarrow{\alpha} E} \qquad \frac{E_j \xrightarrow{\alpha} E'_j}{\sum_{i\in I} E_i \xrightarrow{\alpha} E'_j}\ (j \in I)$$

$$\frac{E \xrightarrow{\alpha} E'}{E \setminus L \xrightarrow{\alpha} E' \setminus L}\ (\alpha, \overline{\alpha} \notin L) \qquad \frac{P \xrightarrow{\alpha} P'}{A \xrightarrow{\alpha} P'}\ (A \stackrel{def}{=} P\ )$$

$$\frac{E \xrightarrow{\alpha} E'}{E|F \xrightarrow{\alpha} E'|F} \qquad \frac{E \xrightarrow{\alpha} E'}{F|E \xrightarrow{\alpha} F|E'} \qquad \frac{E \xrightarrow{\ell} E' \quad F \xrightarrow{\overline{\ell}} F'}{E|F \longrightarrow E'|F'}$$

Table 1: The operational semantics of CCS

The operational semantics of a CCS process is modelled by a labelled transition system. It is shown in table 1. The transition $P \xrightarrow{\alpha} Q$, for $\alpha \in Act$ has to be interpreted as 'P may perform the action α and become Q in doing so'.

We will write **0** to denote the inactive agent, defined as $0 \stackrel{def}{=} \sum_{i\in\emptyset} E_i$. We will also write $P \xrightarrow{\alpha}$ to mean that P' exists such that $P \xrightarrow{\alpha} P'$; we will abbreviate $P \xrightarrow{\tau} P'$ as $P \longrightarrow P'$ and $\alpha.0$ as α; we will use $\equiv$ to denote syntactical identity.

Definition 1 *A process Q is a* descendant *of a process P if a sequence of actions α_1, α_2 ..., α_n $\alpha_i \in Act$, $1 \leq i \leq n$, $n \geq 0$ exist such that $P \xrightarrow{\alpha_1}\xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} Q$ (where for $\alpha, \alpha' \in Act$, $\xrightarrow{\alpha}\xrightarrow{\alpha'}$ is defined in the obvious way as composition of relations). The set $L \subseteq N$ is a* sort *for the process P if whenever Q is a descendant of P and $Q \xrightarrow{\ell}$, then $\ell \in L$. Then we write $P : L$.*

Strong Bisimulation is commonly accepted as the finest extensional or behavioral equivalence on processes one would like to impose.

Definition 2 *A relation $S \subseteq Pr^2$ is a* strong simulation *if for each $(P, Q) \in S$, whenever $P \xrightarrow{\alpha} P'$ for some action α, then $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in S$; S is a* strong bisimulation *if S and S^{-1} are strong simulations. Two processes P and Q are* strongly bisimilar, *written $P \sim Q$ if $(P, Q) \in S$, for some strong bisimulation S.*

3 Reduction Bisimulation

In the definition of strong bisimulation, all the actions in *Act* are treated uniformly. Can we get a characterization for $\sim$ if instead we focus only on the silent action, the simplest form of action? Consider the following definition:

Definition 3 *A relation* $S \subseteq Pr^2$ *is a* reduction simulation *if for each* $(P, Q) \in S$, *whenever* $P \longrightarrow P'$ *then* $Q \longrightarrow Q'$ *and* $(P', Q') \in S$. S *is a* reduction bisimulation *if* S *and* S^{-1} *are reduction simulation. Two processes* P *and* Q *are* reduction bisimilar, *written* $P \mathrel{\dot\sim^\tau} Q$ *iff* $(P, Q) \in S$, *for some reduction bisimulation* S.

By itself, $\dot\sim^\tau$ is not very interesting; it is rather weak and for instance it is not preserved by parallel composition.

Example 1 *Consider* $P \stackrel{def}{=} a.0$ *and* $Q \stackrel{def}{=} 0$. *Then we have* $P \mathrel{\dot\sim^\tau} Q$, *but* $P \mid \overline{a} \mathrel{\dot{\not\sim}^\tau} Q \mid \overline{a}$.

It is natural then to consider the congruence induced by $\dot\sim^\tau$.

Definition 4 *Two processes* P *and* Q *are* reduction congruent, *written* $P \sim^\tau Q$ *iff for each context* C, *it holds* $C(P) \mathrel{\dot\sim^\tau} C(Q)$.

Remark: We are trying to use a dot to denote an equivalence which is not necessarily a congruence; so in this case, and for future equivalences, the absence of a dot means a congruence.

Proposition 1 $\sim \subseteq \sim^\tau$

Proof. Clearly, $\sim \subseteq \dot\sim^\tau$; since $\sim$ is a congruence, $P \sim Q$ implies $C(P) \sim C(Q)$, which in turns implies $C(P) \mathrel{\dot\sim^\tau} C(Q)$, for every context C. □

Remark: The proof of the above proposition would still be correct if we allowed non standard CCS operators in the context C, as long as such operators are "well-formed"; intuitively an operator is well-formed if its behavior depends only on the semantics - not on the syntax - of its operands. In fact,

$\sim$ is a congruence over well-formed operators. The meaning of well-formed operators and well-formed transition systems is formally studied in [3] and G.

Unfortunately the converse of Proposition 1 not true.

Definition 5 *P is* always divergent *if for every descendant P' of P, we have $P' \longrightarrow$.*

Processes which are always divergent are not necessarily strong bisimilar (see Proposition 3); however they are reduction congruent; intuitively since any state that they can reach is divergent, no CCS context can make the distinction between them as long as the τ actions are the only observed.

Proposition 2 *If P and Q are always divergent, then $P \sim^{\tau} Q$.*

Proof (outline) We shall prove that
$S = \{(C(P), C(Q)) \mid C$ is a context, and P, Q are always divergent$\}$
is a reduction bisimulation.

Let $(C(P), C(Q)) \in S$. Suppose that $C(P) \longrightarrow C'(P)$ and C itself has produced the τ-action (i.e. P has not been "used"): then also $C(Q) \longrightarrow C'(Q)$ and $(C(P), C(Q) \in S$.

Suppose now that $C(P) \longrightarrow C'(P')$ and P has contributed to such action (either with an interaction with C or by simply performing itself the τ-action): then a context C'' exists such that for any P'', Q' such that $P \longrightarrow P''$ and $Q \longrightarrow Q'$, it holds that $C(P) \longrightarrow C''(P'')$ and $C(Q) \longrightarrow C''(Q')$. Moreover $C'(P')$, $C''(P'')$, $C''(Q')$ are all always divergent. It follows that $(C'(P'), C''(Q')) \in S$. □

Proposition 3 $\sim \neq \sim^{\tau}$.

Proof. Take $Q \stackrel{def}{=} \tau.Q$ and $P \stackrel{def}{=} \tau.P + a.Q$. By Proposition 1, $P \sim^{\tau} Q$; however $P \not\sim Q$. □

We will prove however in section 3 that under some restrictions, namely divergence-freeness, we do have $\sim^{\tau} = \sim$.

Notice that we could distinguish the processes P and Q in the proof of Proposition 3 if we added a disabling operator $\rhd$, very similar in the intent to Lotos's desrupting operator [5], described by the following rules:

i) if $Q \xrightarrow{\alpha} Q'$ then $P \rhd Q \xrightarrow{\alpha} P \rhd Q'$;

ii) if $P \xrightarrow{\ell} P'$ and $Q \xrightarrow{\overline{\ell}} Q'$, then $P \rhd Q \longrightarrow P'$.

Then for P and Q as defined in Proposition 3 it would indeed hold that $\overline{a}.0 \rhd P \not\dot{\sim}^{\tau} \overline{a}.0 \rhd Q$.

It looks that, at least when image-finiteness is guaranteed, the addition of $\rhd$ would make $\sim^{\tau}$ and $\sim$ coincide. We have not pursued further this direction. Instead we wanted to stick to CCS-contexts; what we have tried to do is to increase as little as possible the power of the global observer, but enough to induce a congruence identical with $\sim$.

4 The Barbed Bisimulation

Let us write $P \downarrow$ to mean that ℓ exists such that $P \xrightarrow{\ell}$ (that is, P can perform some visible action) and $P \not\downarrow$ if it is not the case that $P \downarrow$.

Definition 6 *A process is called* inactive *if $P : \emptyset$ (i.e. P can only perform τ actions).*

Definition 7 *A relation $S \subseteq Pr^2$ is a* barbed simulation *if for each $(P, Q) \in S$,*
1) whenever $P \longrightarrow P'$ then $Q \longrightarrow Q'$ and $(P', Q') \in S$;
2) $P \downarrow$ implies $Q \downarrow$.
S is a barbed bisimulation *if S and S^{-1} are barbed simulation. Two processes P and Q are* barbed-bisimilar*, written $P \dot{\sim}^{\square} Q$ iff $(P, Q) \in S$, for some barbed bisimulation S.*

Notice that the global observer has visibility of the silent actions produced and of the state reached; he can also recognize the production of a visible action, but in this case he cannot see neither the identity of the action produced nor the state reached.

Definition 8 *Two processes P and Q are* barbed-congruent*, written $P \sim^{\square} Q$ iff for each context C, it holds that $C(P) \dot{\sim}^{\square} C(Q)$.*

Again, clearly, it holds that $\sim^{\square} \subseteq \sim$. The remaining of this section is devoted to the proof that $\sim \subseteq \sim^{\square}$.

Let b be a name. Consider the sequence $\{N^b_\gamma\}_{\gamma\ ordinal}$ of processes so defined:
$N^b_0 \stackrel{def}{=} \tau.b$,
$N^b_\gamma \stackrel{def}{=} \tau.(b + \tau.N^b_{\gamma-1})$, γ ordinal successor,
$N^b_\gamma \stackrel{def}{=} \sum_{\delta<\gamma} N^b_\delta$, γ ordinal limit.

Lemma 1 *Let P, Q, be two inactive processes; then for all ordinals γ and δ, $\gamma \neq \delta$ implies $P|N^b_\gamma \not\dot\sim^\Box Q|N^b_\delta$.*

Proof. Use transfinite induction. □

Now we can prove the main theorem:

Theorem 1 $\sim^\Box \subseteq \sim$

Proof. Let $P \sim^\Box Q$. We will define a strong bisimulation S containing the pair (P, Q). Let $P, Q : L$, where $L = \{\ell_i | i \in I\}$ and $b \notin L$.

Because I is a set and $\{N^b_\gamma\}$ is built over the ordinals, we are sure that it is possible to associate a distinguished N^b_i for each $i \in I$.

Define the context $C(X) \stackrel{def}{=} (X|V) \setminus L$ where
$V \stackrel{def}{=} \sum_{i\in I} \overline{\ell}_i.\tau.V_i$, $\qquad V_i \stackrel{def}{=} (b + \tau.\tau.(\tau.N^b_i + b) + \tau.V)$.
Now take $S = \{(P, Q) | P, Q : L \ \ and \ \ C(P) \dot\sim^\Box C(Q)\}$.

We claim that S is a strong bisimulation. Let $(P, Q) \in S$ and let $P \stackrel{\ell_i}{\longrightarrow} P'$. We have to find Q' such that $Q \stackrel{\ell_i}{\longrightarrow} Q'$ and $(P', Q') \in S$.

First note that $C(P) \longrightarrow\longrightarrow R$, for $R \equiv (P'|V_i)\setminus L$. Since $C(P) \dot\sim^\Box C(Q)$, we must have $C(Q) \longrightarrow\longrightarrow T$, for some T such that $R \dot\sim^\Box T$. But $R\downarrow$; hence for some j, $Q \stackrel{\ell_j}{\longrightarrow} Q'$ and $T \equiv (Q'|V_j) \setminus L$.

Also, it must be that $i = j$. For if not, then $R \longrightarrow\longrightarrow R' \equiv (P'|(\tau.N^b_i + b)) \setminus L$, where after the first move b is not visible, while it is after the second. This forces $T \longrightarrow\longrightarrow T' \equiv (Q'|(\tau.N^b_j + b))\setminus L$; but then $R' \longrightarrow R'' \equiv (P'|N^b_i)\setminus L$ and $R'' \not\downarrow$, which could only be matched with $T' \longrightarrow T'' \equiv (Q'|N^b_j) \setminus L$. However we can prove that $R'' \not\dot\sim^\Box T''$. In fact, since $b \notin L$ and for any γ it is $N^b_\gamma : \{b\}$, we have that $R'' \sim (P' \setminus L)|N^b_i$ and using Proposition 1, also $R'' \dot\sim^\Box (P' \setminus L)|N^b_i$; similarly, $T'' \dot\sim^\Box (Q' \setminus L)|N^b_j$. Both $(P' \setminus L)|N^b_i$ and $(Q' \setminus L)|N^b_j$ are inactive; then the conclusion follows from Lemma 1.

It therefore remains to prove that $(P', Q') \in S$, i.e. that $C(P') \dot{\sim}^{\Box} C(Q)$. Using analogous reasoning as above, one can prove that the move $R \longrightarrow (P'|V)\backslash L \not\downarrow$ can only be matched by $T \longrightarrow (Q'|V)\backslash L$ obtaining $(P'|V)\backslash L \dot{\sim}^{\Box} (Q' \mid V) \setminus L$, which is exactly $C(P') \dot{\sim}^{\Box} C(Q)$.

The case when $P \longrightarrow P'$ is simpler (V does not need to move). $\Box$

Proposition 3 asserted that in general $\sim \neq \sim^{\tau}$. This was shown by using divergent processes. We can prove now that divergence is just what makes the two relations different.

Definition 9 *Let Div be the largest class of processes such that*

$$\forall P \in Div\ \exists P' \in Div \text{ and } P \longrightarrow P'$$

The processes in Div are usually called divergent*. A process P is* divergent free *if no descendant of P is divergent.*

Definition 10 *Let TotDiv be the largest class of processes such that whenever $P \in TotDiv$ and Q is a descendant of P, then $Q \longrightarrow$. We will call* totally divergent *the processes in TotDiv.*

The process $\Omega \stackrel{def}{=} \tau.\Omega$ is the classical totally divergent process. Notice that a process may be divergent but not totally divergent. As an example, take the process $\tau.b + \Omega$.

Theorem 2 *If P and Q are divergent free, then* $\sim = \sim^{\tau}$

Proof. (outline) Because of Proposition 1 we have only to prove that $\sim^{\tau} \subset \sim$.

Let $\{N^{\Omega}_{\gamma}\}_{\gamma\ ordinal}$, be defined as $\{N^{b}_{\gamma}\}_{\gamma\ ordinal}$ with $\Omega + \tau$ replacing b. Now, when P and Q are divergent free, for N^{Ω}_{i} instead of N^{b}_{i}, the predicate "$R \longrightarrow R'$, for some R' totally divergent" instead of "$R \downarrow$", and $\dot{\sim}^{\tau}$, $\sim^{\tau}$ instead of $\dot{\sim}^{\Box}$, $\sim^{\Box}$, the proofs of Lemma 1 and Theorem 1 remain correct, therefore yielding $\sim^{\tau} \subset \sim$. $\Box$

5 Other issues on the Barbed Bisimulations

We mention here some other issues on barbed bisimulations that we have examined; in particular the use of barbed bisimulation in the weak case and in higher-order process calculi. We refer to [4] for the details.

5.1 The Weak case

In the weak case the relation $\stackrel{\alpha}{\Longrightarrow}$ replaces $\stackrel{\alpha}{\longrightarrow}$, where $\Longrightarrow$ is defined as reflexive and transitive closure of $\longrightarrow$, and $\stackrel{\ell}{\Longrightarrow}$ is $\Longrightarrow\stackrel{\ell}{\longrightarrow}\Longrightarrow$. Then $P \Downarrow$, which replaces $P \downarrow$, has to be interpreted as "for some ℓ, $P \stackrel{\ell}{\Longrightarrow}$".

Notice that since in the weak case τ-actions have no weight, weak reduction congruence corresponds to the universal relation (i.e. for every P and Q, $P \approx^{\tau} Q$).

Let $\dot{\approx}^{\Box}$ be the obvious generalization of $\dot{\sim}^{\Box}$ in the weak case. We have tried to understand if it is possible to recover the CCS observational equivalence ($\approx$), probably the most important bisimulation-based equivalence for CCS, using $\dot{\approx}^{\Box}$. First of all, since $\approx$ is not a congruence over the sum operator, we cannot hope to get $\approx$ simply by parametrizing $\dot{\approx}^{\Box}$ over *all* the possible contexts (this would lead more towards the CCS observational congruence).

Instead we need to parametrize $\dot{\approx}^{\Box}$ over a *subclass* of all the possible contexts. The obvious representative is the subclass $\Re$ of the *structural* contexts, intuitively the contexts which are built by composing variables and processes by means of any operator but sum. We have called barbed congruence over $\Re$ (written $\approx^{\Box}_{\Re}$) the resulting relation (i.e. $P \approx^{\Box}_{\Re} Q$ if for every $C \in \Re$, $C(P) \dot{\approx}^{\Box} C(Q)$).

So far we have been able to prove that $\approx^{\Box}_{\Re} = \approx$ only for processes which are image-finite w.r.t. the relation $\stackrel{\ell}{\Longrightarrow}$. It is still an open problem whether the equality still holds when the image-finiteness requirement is dropped.

Because $\dot{\approx}^{\Box}$ seems not easy to handle, we have also considered the possibility of increasing the power of the observer. One way to obtain this is by partitioning the visible actions into subsets, say $S_1, \ldots S_n$, and then allowing the observer to be able to discriminate between visible actions from different subsets (but still preventing him/her from discriminating actions in the same subset). Then, one gets $\approx$ just by using a partition into two subsets and by the parametrization of the resulting relation over the contexts in $\Re$.

As an aside let us just point out that the possibility of parametrizing the barbed-based bisimulations over a specific class of contexts seems to have other interesting applications. In [4] it has been used to obtain some fully abstract model for the lazy lambda calculus from its encoding into the π-calculus. We are just now considering also the use of parametrization in the framework of action refinement. Intuitively, since the refinement of an action modifies the communication protocol of a process, only contexts which "respect" such protocol should be considered when verifying the equivalence between two refined processes. The barbed bisimulations allow us to make this idea into a nice piece of theory.

5.2 Higher Order Process Calculi

We have considered π-calculus, Plain CHOCS and HOπ as case study.

In the π-calculus. The π-calculus [1] is a generalization of CCS where names can be exchanged as a result of a communication. Although it is perhaps questionable whether the π-calculus should be considered an higher order process calculus, it has been proved that name passing is enough to give a nice encoding of higher-order communications like process passing.

There are two major philosophy for defining in the π-calculus the correspondent of the CCS $\sim$ and $\approx$, respectively referred as "early version" and "late version" (see [6]); it is not clear yet which one should be preferred. We were interested in finding out which of the two – if any – could be captured using the barbed bisimulation machinery. The answer has been the early version, a good point in favor of this one; in fact, the same results described here for CCS can be recovered for the corresponding early equivalences in the π-calculus.

In Plain CHOCS and HOπ. Plain CHOCS ([12]) is a CCS-based language where process passing is adopted; HOπ ([4]) is an enrichment of the π-calculus where - besides names - also processes and parametrized processes of arbitrarily high order can be transmitted.

The adoption of the barbed bisimulations/congruences seems to solve the non trivial problem of obtaining a natural bisimulation-based equivalence for such calculi. In particular, in Plain CHOCS the adoption of the barbed bisimulations avoid some counterintuitive equalities induced by the higher-order bisimulation proposed by Thomsen for his Plain CHOCS.

References

[1] Milner, R., Parrow, J.G. and Walker, D.J., *A Calculus of Mobile Pro-*

cesses, Parts I and II, Report ECS-LFCS-89-85 and -86, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University.

[2] Groote, J.F., *Transition System Specifications with Negative Premises*, Report CS-R8950, Centrum voor Wiskunde in Informatica, Amsterdam.

[3] Groote, J.F. and Vaandrager, F.W., *Structured Operational Semantics and Bisimulation as a Congruence*, Report CS-R8845, Centrum voor Wiskunde in Informatica, Amsterdam.

[4] Sangiorgi, D., forthcoming PhD thesis, Computer Science Department, Edinburgh University.

[5] ISO DIS 8807, Information Processing Systems - Open System Interconnection, LOTOS, A Formal Description Technique Based on the Temporal Ordering of Observational Behaviour, 1987.

[6] Milner, R., Parrow, J.G. and Walker, D.J., *Modal Logics for Mobile Processes*, Report ECS-LFCS-91-136, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University.

[7] Boudol, G., *A Lambda-Calculus for (strict) Parallel Function*, Report N.1387, INRIA-Sophia Antipolis, 1991.

[8] Berry, G. and Boudol, G., *The Chemical Abstract Machine*, Proc 17th Annual Symposium on Principle of Programming Languages, 1990.

[9] Abramsky, S., *The Lazy Lambda Calculus*, in Declarative Programming, ed. Addison Wesley, 1988.

[10] Milner, R., *A Calculus of Communicating Systems*, Prentice Hall, 1989.

[11] Milner, R., *Functions as Processes*, Report N.1154, INRIA-Roquencourt, 1990.

[12] Thomsen, B., *Plain CHOCS*, Report DOC 89/4, Department of Computing, Imperial College, London, 1989.