Active Evaluation Contexts for Reaction Semantics

Henrik Pilegaard Flemming Nielson Hanne Riis Nielson

{hepi,nielson,riis}@imm.dtu.dk

Informatics and Mathematical Modelling
The Technical University of Denmark

Abstract

In the context of process algebras it is customary to define semantics in the form of a reaction relation supported by a structural congruence relation. Recently process algebras have grown more expressive in order to meet the modelling demands of fields as diverse as business modelling and systems biology. This leads to combining various features, such as general choice and parallelism that were previously studied separately, and it often becomes difficult to define the reaction semantics. We present a general approach based on *active evaluation contexts* that allows the reaction semantics to be easily constructed.

Keywords: Structural Operational Semantics, Reaction Semantics, Process Algebra, General Choice, Calculus of Communicating Systems, BioAmbients.

1 Introduction

Since their proposal [7,11,1] process calculi have become the primary tool for researching paradigms of concurrent computation. In the three decades that have passed two types of semantics have emerged:

Structural operational semantics [15] describes how processes may interact with their immediate environment. As usual for structural operational semantics the immediate behaviour of a composite process is defined structurally in terms of the immediate behaviours of its component processes.

Behaviours are often expressed using labelled transition systems. The label languages have considerable potential, which ensures that the structural operational semantics approach is viable for more expressive calculi also. As calculi do become more expressive, however, the required label languages tend to grow complicated and somewhat obscure the intuition of concurrency.

Reaction semantics in the style of the chemical abstract machine (CHAM) [2] clearly expresses an intuitive (chemical) understanding of concurrency. Every process term is perceived as the description of a solution of (syntactic) reactive entities. The usual *structural congruence* is a magical stirring mechanism that

This paper is electronically published in
Electronic Notes in Theoretical Computer Science

allows syntactic entities to float and mix as required. The *reaction relation* then simply states how reactions happen when 'matching' reactive entities come sufficiently close to each other.

While very intuitive this type of semantics has the drawback that the structural congruence is subject to conflicting requirements. On the one hand, we demand that the congruence sharply distinguishes between semantically different process expressions. On the other hand, we require it to be able to move potential redex constituents, which are syntactically located arbitrarily far apart, close enough together for a reaction rule to identify them. The usual decidability problems aside, these two requirements seem to clash as calculi grow in expressiveness.

Due to their different strengths it is usual for calculi to have both types of semantics. However, for the more elaborate calculi this is often difficult *unless* syntactic restrictions are imposed.

Milner's Calculus of Communicating Systems (CCS) is a prime example. The more recent version [13], which we briefly describe in Section 2, restricts choice to guarded sum. This facilitates both types of semantics because a normal form

$$\sum_j \alpha_j.P_j \tag{1}$$

can always be assumed for the constituents of redexes.

This may be contrasted to the original calculus [11] that has unrestricted choice. For this reason the substantially more complex normal form

$$(\ldots(((\alpha.P + P')|P'') + P''')|P'''' \ldots) \tag{2}$$

needs to be assumed for the constituents of redexes. This normal form is hard to match syntactically and the structural congruence is of little help as it is semantically meaningless to allow choice and parallel to distribute freely over one another. Thus, traditionally, only structural operational semantics is defined for derivatives of this calculus.

In this paper we show how reaction semantics can be defined even for very expressive calculi. One may ask why it is of interest to be able to deal with a binary unrestricted choice as opposed to an indexed guarded sum (over some arbitrary finite index set). In doing so we follow one of the design principles used by Gordon Plotkin when devising Structural Operational Semantics [15]: that one should always strive to use unary or binary syntactic constructors rather than general n-ary constructors because the former choice assists machine readable formal semantics and also gives a deeper semantic understanding of the programming construct at hand [16].

The proposed approach is based on a novel notion of *active evaluation contexts*. These contexts arise naturally when one allows standard evaluation contexts, originally proposed by Felleisen [5], to evolve when reactions occur. In Section 3 we develop the active evaluation contexts and use them to define a reaction semantics for the recursion-free fragment of CCS with unrestricted choice. The main theoretical result of this paper is that the resulting reaction semantics agrees with Milner's original structural operational semantics for closed expressions.

$$P ::= \mathbf{0} \mid \alpha \,.\, P \mid \sum_{i \in I} \alpha_i \,.\, P_i \mid P \,\big|\, P \mid (\nu a)\, P \qquad\qquad \alpha ::= \tau \mid a \mid \overline{a}$$

Fig. 1. Syntax of finite core CCS with guarded sums (CCSgs).

Reordering of parallel processes:

$P \,\big|\, \mathbf{0} \equiv_{\mathsf{gs}} P$

$P \,\big|\, Q \equiv_{\mathsf{gs}} Q \,\big|\, P$

$P \,\big|\, (Q \,\big|\, R) \equiv_{\mathsf{gs}} (P \,\big|\, Q) \,\big|\, R$

Scope rules for name restrictions:

$(\nu a)\ \mathbf{0} \equiv_{\mathsf{gs}} \mathbf{0}$

$(\nu a)\,(\nu b)\, P \equiv_{\mathsf{gs}} (\nu b)\,(\nu a)\, P$

$(\nu a)\,(P \,\big|\, Q) \equiv_{\mathsf{gs}} P \,\big|\, (\nu a)\, Q \quad \textit{if } a \notin \mathrm{fn}(P)$

Alpha equivalence:

$P \equiv_\alpha Q \Rightarrow P \equiv_{\mathsf{gs}} Q$

Reordering of term in a summation:

Summands can be freely reordered.

Equivalence:

$P \equiv_{\mathsf{gs}} P$

$P \equiv_{\mathsf{gs}} Q \Rightarrow Q \equiv_{\mathsf{gs}} P$

$P \equiv_{\mathsf{gs}} Q \wedge Q \equiv_{\mathsf{gs}} R \Rightarrow P \equiv_{\mathsf{gs}} R$

Congruence:

$P \equiv_{\mathsf{gs}} Q \Rightarrow \alpha.P + M \equiv_{\mathsf{gs}} \alpha.Q + M$

$P \equiv_{\mathsf{gs}} Q \Rightarrow (\nu a)\, P \equiv_{\mathsf{gs}} (\nu a)\, Q$

$P \equiv_{\mathsf{gs}} Q \Rightarrow P \,\big|\, R \equiv_{\mathsf{gs}} Q \,\big|\, R$

$P \equiv_{\mathsf{gs}} Q \Rightarrow R \,\big|\, P \equiv_{\mathsf{gs}} R \,\big|\, Q$

Fig. 2. Structural congruence of CCSgs.

In the case of more complicated calculi the notions of active evaluation contexts and structural congruence combine nicely to give the desired semantics. We illustrate this in Section 4 where we define a reaction semantics for the full BioAmbients calculus extended with unrestricted choice.

2 CCS with Guarded Sums

In order to set the scene we start by considering CCS with guarded sums as defined by Milner [13]. In order to expose our contribution in Section 3 more clearly we shall focus on the finite fragment of the language; thus omitting recursion. This does not indicate a limitation in our framework - as we shall demonstrate later, in Section 4, recursion can easily be incorporated using a structural congruence.

Now, let $\mathcal{N}$, ranged over by $a, b, \cdots$, be a denumerable set of channel names and let the special symbol τ denote internal actions. The syntactical class of action prefixes, $\alpha \in \mathrm{Act}$, then contains all names $a \in \mathcal{N}$, all corresponding co-names $\overline{a} \in \mathcal{N}$, and the special symbol τ. In this context the class of finite core CCS processes with guarded sums, to be denoted CCSgs, is described by the grammar in Figure 1, where we assume the I in $\sum_{i \in I} \alpha_i \,.\, P_i$ to be finite and write $\mathbf{0}$ when $|I| = 0$ and $\alpha.P$ when $|I| = 1$.

Because the choice construct $\sum_{i \in I} \alpha_i \,.\, P_i$ is guarded it is possible to define a traditional (CHAM style) reaction semantics. As always, due to the syntactical nature of the reaction semantics, the definition relies on a structural congruence relation. If we let $\equiv_\alpha$ denote ordinary α-equivalence, the structural congruence, $\equiv_{\mathsf{gs}}$, is the least relation that satisfies the axioms and rules in Figure 2. Using the

$$\text{TAU:}\quad \tau\,.\,P + M \longrightarrow_{\mathsf{gs}} P \qquad \text{RES:}\quad \frac{P \longrightarrow_{\mathsf{gs}} P'}{(\nu a)\,P \longrightarrow_{\mathsf{gs}} (\nu a)\,P'} \qquad \text{PAR:}\quad \frac{P \longrightarrow_{\mathsf{gs}} P'}{P \,|\, Q \longrightarrow_{\mathsf{gs}} P' \,|\, Q}$$

$$\text{REACT:}\quad (a\,.\,P + M) \,|\, (\overline{a}\,.\,Q + N) \longrightarrow_{\mathsf{gs}} P \,|\, Q \qquad \text{STRUCT:}\quad \frac{P \equiv Q \quad Q \longrightarrow_{\mathsf{gs}} Q' \quad Q' \equiv P'}{P \longrightarrow_{\mathsf{gs}} P'}$$

Fig. 3. Reaction relation of CCSgs.

$$\text{SUM}_t:\quad M + \alpha\,.\,P + N \xrightarrow{\alpha}_{\mathsf{gs}} P \qquad \text{REACT}_t:\quad \frac{P \xrightarrow{\lambda}_{\mathsf{gs}} P' \quad Q \xrightarrow{\overline{\lambda}}_{\mathsf{gs}} Q'}{P \,|\, P \xrightarrow{\tau}_{\mathsf{gs}} P' \,|\, Q'}$$

$$\text{L-PAR}_t:\quad \frac{P \xrightarrow{\alpha}_{\mathsf{gs}} P'}{P \,|\, Q \xrightarrow{\alpha}_{\mathsf{gs}} P' \,|\, Q} \qquad \text{R-PAR}_t:\quad \frac{Q \xrightarrow{\alpha}_{\mathsf{gs}} Q'}{P \,|\, Q \xrightarrow{\alpha}_{\mathsf{gs}} P \,|\, Q'}$$

$$\text{RES}_t:\quad \frac{P \xrightarrow{\alpha}_{\mathsf{gs}} P'}{(\nu a)\,P \xrightarrow{\alpha}_{\mathsf{gs}} (\nu a)\,P'} \quad \text{if } \mathsf{n}(\alpha) \neq a$$

Fig. 4. Structural operational semantics of CCSgs.

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

Fig. 5. Syntax of finite core CCS with unrestricted choice (CCSuc).

congruence the reaction relation, $\longrightarrow_{\mathsf{gs}}$, defined by the axioms and rules of Figure 3 specifies the full reaction semantics of CCSgs. Note how this definition relies on the existence of the previously described normal forms of type (1).

Next we define a structural operational semantics specifying the process behaviour in terms of labelled transition systems. We assume the same class α of action prefixes as before but use the abbreviation λ to denote action prefixes that are not internal (i.e. $\lambda \in \mathrm{Act}\backslash\{\tau\}$); we shall write $\mathsf{n}(\alpha)$ to denote the base name of any action prefix. The transition relation $\xrightarrow{\alpha}_{\mathsf{uc}}$ defining the structural operational semantics is then the least relation satisfying the axioms and rules of Figure 4.

While these two formulations of the CCSgs semantics are often used for different purposes they are intended to express the same behaviour for closed process expressions. Thus the following result [13] is crucial:

Theorem 2.1 (For CCSgs reaction agrees with τ-transition) *For any CCSgs process P we have that $P \xrightarrow{\tau}_{\mathsf{gs}} \equiv P'$ if and only if $P \longrightarrow_{\mathsf{gs}} P'$.*

Proof. See Milner [13] Theorem 5.6 □

3 Active Evaluation Contexts for CCS

When the calculus is generalised to finite core CCS with *unrestricted choice* (CCSuc), as shown in Figure 5, the picture changes. Neither Milner nor other contributors have ever defined a classic (CHAM style) reaction semantics for a derivative of this language - and for good technical reasons. We believe that the technical means to deal with normal forms of type (2) have simply been lacking, and for this reason calculi descending from CCSuc are traditionally given only a structural operational semantics similar to the one shown in Figure 6 [12].

$$\text{PRE}_t: \quad \alpha\,.\,P \xrightarrow{\alpha}_{\mathsf{uc}} P \qquad \text{L-PAR}_t: \quad \frac{P \xrightarrow{\alpha}_{\mathsf{uc}} P'}{P \,|\, Q \xrightarrow{\alpha}_{\mathsf{uc}} P' \,|\, Q} \qquad \text{L-SUM}_t: \quad \frac{P \xrightarrow{a}_{\mathsf{uc}} P'}{P + Q \xrightarrow{\alpha}_{\mathsf{uc}} P'}$$

$$\text{R-PAR}_t: \quad \frac{Q \xrightarrow{\alpha}_{\mathsf{uc}} Q'}{P \,|\, Q \xrightarrow{\alpha}_{\mathsf{uc}} P \,|\, Q'} \qquad \text{R-SUM}_t: \quad \frac{Q \xrightarrow{a}_{\mathsf{uc}} Q'}{P + Q \xrightarrow{\alpha}_{\mathsf{uc}} Q'}$$

$$\text{RES}_t: \quad \frac{P \xrightarrow{\alpha}_{\mathsf{uc}} P'}{(\nu a)\,P \xrightarrow{\alpha}_{\mathsf{uc}} (\nu a)\,P'} \text{ if } \mathsf{n}(\alpha) \neq a \qquad \text{REACT}_t: \quad \frac{P \xrightarrow{\lambda}_{\mathsf{uc}} P' \quad Q \xrightarrow{\overline{\lambda}}_{\mathsf{uc}} Q'}{P \,|\, Q \xrightarrow{\tau}_{\mathsf{uc}} P' \,|\, Q'}$$

Fig. 6. Structural operational semantics of FcCSSuc.

$$\mathbb{C} ::= [\,] \mid (\nu a)\,\mathbb{C} \mid \mathbb{C} \,|\, P \mid P \,|\, \mathbb{C} \mid \mathbb{C} + P \mid P + \mathbb{C}$$

Fig. 7. The active evaluation contexts of CCSuc.

$$\text{EMP}_c: \quad [\,] \longrightarrow [\,] \qquad \text{NEW}_c: \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{(\nu a)\,\mathbb{C} \longrightarrow (\nu a)\,\mathbb{C}'} \qquad \text{L-PAR}_c: \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{\mathbb{C} \,|\, P \longrightarrow \mathbb{C}' \,|\, P}$$

$$\text{R-PAR}_c: \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{P \,|\, \mathbb{C} \longrightarrow P \,|\, \mathbb{C}'} \qquad \text{L-SUM}_c: \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{\mathbb{C} + P \longrightarrow \mathbb{C}'} \qquad \text{R-SUM}_c: \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{P + \mathbb{C} \longrightarrow \mathbb{C}'}$$

Fig. 8. Context reduction for active evaluation contexts in CCSuc.

$$\text{TAU:} \quad \tau\,.\,P \longrightarrow_{\mathsf{uc}} P$$

$$\text{REACT:} \quad \frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2'}{\mathbb{C}_1[\,\lambda\,.\,P\,] \,|\, \mathbb{C}_2[\,\overline{\lambda}\,.\,Q\,] \longrightarrow_{\mathsf{uc}} \mathbb{C}_1'[\,P\,] \,|\, \mathbb{C}_2'[\,Q\,]} \text{ if } \mathsf{n}(\lambda) \notin (\mathsf{masked}(\mathbb{C}_1) \cup \mathsf{masked}(\mathbb{C}_2))$$

$$\text{CONT:} \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}' \quad P \longrightarrow_{\mathsf{uc}} P'}{\mathbb{C}[\,P\,] \longrightarrow_{\mathsf{uc}} \mathbb{C}'[\,P'\,]}$$

Fig. 9. Reaction relation of CCSuc.

We shall now propose a semantics for CCSuc that retains the intuition of reaction semantics, but avoids the difficulties of previous approaches. For this purpose we shall introduce a notion of *active evaluation contexts* as defined in Figure 7. As usual for process/evaluation contexts, active evaluation contexts are process expressions with exactly one hole [5,13,8]. Contrary to ordinary contexts, however, we shall allow active contexts to evolve when reactive sub-processes occupying their hole engage in reactions.

To facilitate this we define the *context reduction relation* described in Figure 8. It specifies exactly what happens to contexts when reactive sub-processes engage in reaction. The reduction ability of contexts enables the compact and elegant definition of the reaction relation, $\longrightarrow$, shown in Figure 9. Here we use the auxiliary function $\mathsf{masked}(\mathbb{C})$ to determine the names and co-names that are restricted by a context $\mathbb{C}$. The function is given by:

$$\mathsf{masked}(\mathbb{C}) = \{\lambda \mid \textit{some } (\nu\lambda)\,\mathbb{C}' \textit{ occurs in } \mathbb{C}\}$$

In particular, $\mathsf{masked}((\nu a)\,\mathbb{C}) = \{a\} \cup \mathsf{masked}(\mathbb{C})$.

3.1 Correspondence of Semantics

It is evident that that the context reduction relation strongly resembles those rules of the structural operational semantics that encode the recursive descent into process terms. Consequently, structural congruence turns out to be unnecessary as was the case for the structural operational semantics.

In this favourable context the equivalence of the reaction semantics and the structural operational semantics for closed process expressions can be expressed simply as:

Theorem 3.1 (Reaction corresponds to τ transition) *$P \longrightarrow_{\mathsf{uc}} P'$ if and only if $P \xrightarrow{\tau}_{\mathsf{uc}} P'$.*

The proof has two parts, but first we establish the following useful result, which shows how the notion of active contexts relates to the structural operational semantics:

Lemma 3.2 (Contexts respect behaviour) *If $P \xrightarrow{\alpha}_{\mathsf{uc}} P'$, $\mathbb{C} \longrightarrow \mathbb{C}'$, and $\mathsf{n}(\alpha) \notin \mathsf{masked}(\mathbb{C})$ then $\mathbb{C}[\,P\,] \xrightarrow{\alpha}_{\mathsf{uc}} \mathbb{C}'[\,P'\,]$.*

Proof. The proof proceeds by structural induction on $\mathbb{C}$:

Base case $[\,]$: trivial.

Case $\mathbb{C} \mid R$:

From the premises we have $P \xrightarrow{\alpha}_{\mathsf{uc}} P'$, $\mathbb{C} \mid R \longrightarrow (\mathbb{C} \mid R)'$, and $\mathsf{n}(\alpha) \notin \mathsf{masked}(\mathbb{C})$.
By the shape of the inference of $\longrightarrow$ we have $(\mathbb{C} \mid R)' = \mathbb{C}' \mid R$ and $\mathbb{C} \longrightarrow \mathbb{C}'$ as a necessary premise. From the induction hypothesis it is now clear that $\mathbb{C}[\,P\,] \xrightarrow{\alpha}_{\mathsf{uc}} \mathbb{C}'[\,P'\,]$.
A single application of the L-PAR$_t$ rule now establishes the desired result: $\mathbb{C}[\,P\,] \mid R \xrightarrow{\alpha}_{\mathsf{uc}} \mathbb{C}'[\,P'\,] \mid R$

Cases $R \mid \mathbb{C}$, $R + \mathbb{C}$, and $\mathbb{C} + R$:

All similar.

Case $(\nu a)\,\mathbb{C}$:

From the premises we have $P \xrightarrow{\alpha}_{\mathsf{uc}} P'$, $(\nu a)\,\mathbb{C} \longrightarrow ((\nu a)\,\mathbb{C})'$, and $\mathsf{n}(\alpha) \notin \mathsf{masked}((\nu a)\,\mathbb{C})$.
By the shape of the inference of $\longrightarrow$ we have $((\nu a)\,\mathbb{C})' = (\nu a)\,\mathbb{C}'$ and $\mathcal{C} \longrightarrow \mathcal{C}'$ as a necessary premise and we know that $\mathsf{n}(\alpha) \notin \mathsf{masked}(\mathbb{C})$. From the induction hypothesis it is now clear that $\mathbb{C}[\,P\,] \xrightarrow{\alpha}_{\mathsf{uc}} \mathbb{C}'[\,P'\,]$.
Given that $\mathsf{n}(\alpha) \notin \mathsf{masked}((\nu a)\,\mathbb{C})$ we have $\mathsf{n}(\alpha) \neq a$ and a single application of the RES$_t$ rule now establishes the desired result: $(\nu a)\,\mathbb{C}[\,P\,] \xrightarrow{\alpha}_{\mathsf{uc}} (\nu a)\,\mathbb{C}'[\,P'\,]$. □

Given this lemma it is now easy to establish the 'if' part of Theorem 3.1:

Lemma 3.3 *If $P \longrightarrow_{\mathsf{uc}} P'$ then $P \xrightarrow{\tau}_{\mathsf{uc}} P'$.*

Proof. We proceed by induction on the inference of $\longrightarrow_{\mathsf{uc}}$:
Case TAU:

Given the process term $\tau.P$ rule PRE_t trivially instantiates to give us $\tau.P \xrightarrow{\tau}_{\mathsf{uc}} P$, just as desired.

Case REACT:

Rule PRE_t gives us $a\,.\,P_1 \xrightarrow{a}_{\mathsf{uc}} P_1$ and $\overline{a}\,.\,P_2 \xrightarrow{\overline{a}}_{\mathsf{uc}} P_2$, and from the rule premises we have that $\mathbb{C}_1 \longrightarrow \mathbb{C}_1'$, $\mathbb{C}_2 \longrightarrow \mathbb{C}_2'$, $\mathsf{n}(a) \notin (\mathsf{masked}(\mathbb{C}_1) \cup \mathsf{masked}(\mathbb{C}_2))$.
Using Lemma 3.2 we can establish that $\mathbb{C}_1[\,a\,.\,P_1\,] \xrightarrow{a}_{\mathsf{uc}} \mathbb{C}_1'[\,P_1\,]$ and $\mathbb{C}_2[\,\overline{a}\,.\,P_2\,] \xrightarrow{\overline{a}}_{\mathsf{uc}} \mathbb{C}_2'[\,P_2\,]$. By a single application of rule REACT_t we can now conclude $\mathbb{C}_1[\,a\,.\,P_1\,] \,|\, \mathbb{C}_2[\,\overline{a}\,.\,P_2\,] \xrightarrow{\tau}_{\mathsf{uc}} \mathbb{C}_1'[\,P_1\,] \,|\, \mathbb{C}_2'[\,P_2\,]$, as required.

Case CONT:

From the premises we have $\mathbb{C} \longrightarrow \mathbb{C}'$ and $P \longrightarrow_{\mathsf{uc}} P'$. Using the induction hypothesis on the latter we obtain $P \xrightarrow{\tau}_{\mathsf{uc}} P'$, where obviously $\mathsf{n}(\tau) \notin \mathsf{masked}(\mathbb{C})$.
Lemma 3.2 now tells us that $\mathbb{C}[\,P\,] \xrightarrow{\tau}_{\mathsf{uc}} \mathbb{C}'[\,P'\,]$, as required. □

We now turn to the 'only if' part of Theorem 3.1, which is a straightforward corollary of the following lemma:

Lemma 3.4 *If $P \xrightarrow{\tau}_{\mathsf{uc}} Q$ then $P \longrightarrow_{\mathsf{uc}} Q$ and if $P \xrightarrow{\lambda}_{\mathsf{uc}} Q$ then, for all contexts $\mathbb{C}, \mathbb{C}'$ such that $\mathbb{C} \longrightarrow \mathbb{C}'$ and $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C})$, we have $\mathbb{C}[\,P\,] \,|\, \overline{\lambda}\,.\,R \longrightarrow_{\mathsf{uc}} \mathbb{C}'[\,Q\,] \,|\, R$ and $\overline{\lambda}\,.\,R \,|\, \mathbb{C}[\,P\,] \longrightarrow_{\mathsf{uc}} R \,|\, \mathbb{C}'[\,Q\,]$.*

Proof. The proof proceeds by induction on the inference of $\xrightarrow{\alpha}_{\mathsf{uc}}$:
Base case PRE_t :

If α is τ then P is $\tau.P'$ and Q is P' and the transition $P \longrightarrow_{\mathsf{uc}} Q$ follows from TAU.
Otherwise P is $\lambda.P'$ and Q is P' and the required transitions both follow from REACT (taking one of $\mathbb{C}_1$ and $\mathbb{C}_2$ to be $\mathbb{C}$ and the other to be $[\,]$).

Case L-PAR_t :

If α is τ then P is $P' \,|\, Q'$ and Q is $P'' \,|\, Q'$ and the transition $P \longrightarrow_{\mathsf{uc}} Q$ follows from the induction hypothesis and CONT where $\mathbb{C}$ is taken to be $[\,] \,|\, Q'$.
Otherwise, P and Q are of a similar form, but the transitions have to follow from REACT. From premises we know that $P' \xrightarrow{\lambda}_{\mathsf{uc}} P''$, and the induction hypothesis then tells us that $\mathbb{C}[\,P'\,] \,|\, \overline{\lambda}\,.\,R \longrightarrow_{\mathsf{uc}} \mathbb{C}'[\,P''\,] \,|\, R$ for all suitable $\mathbb{C}, \mathbb{C}'$ (i.e. $\mathbb{C} \longrightarrow \mathbb{C}'$ with $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C})$). Given that $\mathbb{C}, \mathbb{C}'$ are suitable clearly $\mathbb{C}[[\,] \,|\, Q'], \mathbb{C}'[[\,] \,|\, Q']$ are also suitable, and then the required transitions both follow from REACT.

Case $\text{R-PAR}_t, \text{LSUM}_t,$ and R-SUM_t :

All similar.

Case RES_t :

If α is τ then P is $(\nu a)\,P'$ and Q is $(\nu a)\,P''$ and the transition $P \longrightarrow_{\mathsf{uc}} Q$ follows from the induction hypothesis and CONT where $\mathbb{C}$ is taken to be $(\nu a)\,[\,]$.

Otherwise, P and Q are of a similar form, but the transitions have to follow from REACT. From the premise we know that $P' \xrightarrow{\lambda}_{\mathsf{uc}} P''$, and the induction hypothesis then tells us that $\mathbb{C}[\,P'\,] \,|\, \overline{\lambda}\,.\,R \longrightarrow_{\mathsf{uc}} \mathbb{C}'[\,P''\,] \,|\, R$ for all suitable $\mathbb{C}, \mathbb{C}'$ (i.e. $\mathbb{C} \longrightarrow \mathbb{C}'$ with $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C})$). Given that $\mathbb{C}, \mathbb{C}'$ are suitable clearly $\mathbb{C}[\,(\nu a)\,[\,]\,], \mathbb{C}'[\,(\nu a)\,[\,]\,]$ are also suitable because we know from the side-condition that $\mathsf{n}(\lambda) \neq a$, which means that $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C}) \cup \{a\}$. The required transitions then both follow from REACT.

Case REACT$_t$:

Here α is τ and P is $P' \,|\, Q'$ and Q is $P'' \,|\, Q''$. By the premises and the induction hypothesis we have both $\mathbb{C}[\,P'\,] \,|\, \overline{\lambda}\,.\,R \longrightarrow_{\mathsf{uc}} \mathbb{C}'[\,P''\,] \,|\, R$ and $\overline{\lambda}\,.\,R \,|\, \mathbb{C}[\,P'\,] \longrightarrow_{\mathsf{uc}} R \,|\, \mathbb{C}'[\,P''\,]$. As these reactions can only arise by the use of REACT we may assume that $\mathbb{C}[\,P'\,]$ is of the form $\mathbb{C}[\,\mathbb{C}_{\mathsf{deep}}[\,\lambda\,.\,P_{\mathsf{deep}}\,]\,]$, where $\mathbb{C}_{\mathsf{deep}} \longrightarrow \mathbb{C}'_{\mathsf{deep}}$ and $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C}_{\mathsf{deep}})$.
Similar arguments for Q' allows us to assume that $\mathbb{C}[\,Q'\,]$ is of the form $\mathbb{C}[\,\mathbb{C}_{\mathsf{deep}}[\,\overline{\lambda}\,.\,Q_{\mathsf{deep}}\,]\,]$, where $\mathbb{C}_{\mathsf{deep}} \longrightarrow \mathbb{C}'_{\mathsf{deep}}$ and $\mathsf{n}(\lambda) \notin \mathsf{masked}(\mathbb{C}_{\mathsf{deep}})$. From this we obtain the desired reaction by a single application of REACT. □

4 Active Evaluation Contexts for BioAmbients

The use of process calculi as modelling languages for real-world domains, such as business modelling and systems biology, seems to be a current trend in language based technology. The trend combines many language features that were previously unstudied or only studied in isolation. This invariably leads to evermore expressive calculi that share the difficulties of CCSuc with respect to the definition of appropriate reaction semantics.

The BioAmbients calculus of Regev et al. [18,17,3] is a prime example. The language is a sibling of Mobile Ambients (Cardelli and Gordon [4]) designed to model biological systems. It preserves the notion of ambients as bounded mobile sites of activity; contrary to Mobile Ambients, however, bio-ambients are cast as nameless entities. The ambients are used to model chemically active sub-systems (compartments) bound by biological barriers (membranes) in an intuitive manner.

The calculus is quite extensive in terms of modelling primitives. Appropriate sets of capabilities and co-capabilities are devised for modelling a variety of biological reactions, such as movement and communication, that may happen between the sub-systems. Both communication and movement are facilitated by having capability/co-capability pairs react with each other as in [10,14]. As a consequence all reactions are synchronous in the sense that the process exposing the capability and the process exposing the corresponding co-capability must simultaneously agree on a reaction for it to happen. Such an agreement can be reached only if the two entities share the same (channel) name.

The set of control structures for processes is slightly larger than what is traditionally studied for Mobile Ambients. Besides the ambient construct it includes non-deterministic (external) choice as well as a general recursion construct in the manner of CCS [12] in order to facilitate the description of more faithful models of biological systems.

$P ::=$	$\mathbf{0}$	a terminal (stuck) process
$\mid$	$(\nu a)\, P$	restricting the scope of a to the process expression P
$\mid$	$\lfloor\!\lfloor P \rfloor\!\rfloor$	process P enclosed by ambient boundary
$\mid$	$P \mid P'$	process P in parallel with process P'
$\mid$	$P + P'$	non-deterministic external choice between P and P'
$\mid$	$M\,.\,P$	capability prefixed process
$\mid$	$\mathsf{rec}\, X.\, P$	recursive process definition ($X = P$)
$\mid$	X	process identifier
$M ::=$	$\mathsf{enter}\ a \mid \mathsf{accept}\ a$	enter movement
$\mid$	$\mathsf{exit}\ a \mid \mathsf{expel}\ a$	exit movement
$\mid$	$\mathsf{merge}\text{–}\ a \mid \mathsf{merge}\text{+}\ a$	merge movement
$\mid$	$a!\{b\} \mid a?\{c\}$	local communication binding the variable c
$\mid$	$a\_!\{b\} \mid a\hat{}?\{c\}$	parent to child communication binding the variable c
$\mid$	$a\hat{}!\{b\} \mid a\_?\{c\}$	child to parent communication binding the variable c
$\mid$	$a\#!\{b\} \mid a\#?\{c\}$	sibling communication binding the variable c

Fig. 10. Syntax of BioAmbients.

Following the tradition of ambient calculi BioAmbients is endowed by Regev with a (CHAM style) reaction semantics [18,17]. Arguably, this is a natural choice because it ensures a high degree of coherence between the inherently bio-chemical modelling domain and the operational model of the language. As for CCSgs, however, external choice is limited to guarded sums and, again, we believe that this is so because the technical means to combine parallelism and unrestricted choice was lacking at the time of definition.

In the following we present a BioAmbients variant where choice is unrestricted. We trust this to be a conservative extension of the original calculus, but a formal proof is besides the point of the present paper. Rather, we shall focus on defining a reaction semantics using our active evaluation contexts.

4.1 Syntax

The full syntax of BioAmbients is defined in Figure 10. Note that we use the heavy brackets $\lfloor\!\lfloor$ and $\rfloor\!\rfloor$ to represent ambient boundaries; the ordinary brackets [and] are reserved for substitutions and holes of contexts. We use $a, b, \cdots \in \mathcal{N}$ to denote channel names and $M \in \mathbf{Cap}$ for the notion of *(co-) capabilities*, which are based on names and generalise the notion of actions. As customary for BioAmbients we omit the notion of internal τ-actions. Also, since reactions are based on (co-)capabilities, we have no need for co-names.

In the following we shall write $P[a/b]$ to denote the process that is as P except that all free occurrences of the name b are replaced by a. Similarly, we shall use $P[Q/X]$ to identify the process that is as P except that all free occurrences of the process identifier X are replaced by the process expression Q. In both cases we take care to perform the necessary α-renamings to avoid capturing free names and process identifiers. Finally, we shall use $\mathsf{fn}(P)$ to pick out the free names of a process P and write $P \equiv_\alpha Q$ to state that two processes P and Q are identical up to α-renaming of names.

$$\mathbb{C} ::= [\,] \mid \mathbb{C} \mid P \mid P \mid \mathbb{C} \mid \mathbb{C} + P \mid P + \mathbb{C}$$

Fig. 11. The active evaluation contexts of BioAmbients.

$$\text{EMP}_c : \quad [\,] \longrightarrow [\,] \qquad \text{L-PAR}_c : \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{\mathbb{C} \mid P \longrightarrow \mathbb{C}' \mid P} \qquad \text{R-PAR}_c : \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{P \mid \mathbb{C} \longrightarrow P \mid \mathbb{C}'}$$

$$\text{L-SUM}_c : \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{\mathbb{C} + P \longrightarrow \mathbb{C}'} \qquad \text{R-SUM}_c : \quad \frac{\mathbb{C} \longrightarrow \mathbb{C}'}{P + \mathbb{C} \longrightarrow \mathbb{C}'}$$

Fig. 12. Reduction of BioAmbients active evaluation contexts.

4.2 Semantics

The active evaluation contexts of BioAmbients, shown in Figure 11 and Figure 12, are simpler than those of CCSuc. Their definition embodies three crucial choices, which we shall further substantiate below:

(i) The active contexts are *name restriction free*.

(ii) The active contexts are *ambient boundary free*.

(iii) The active contexts are *recursion free*.

The choice **(i)** is necessary because both π-style name passing and ambient style movement may cause extrusion of scope. This happens when restricted names are communicated to recipients or moved to positions outside of their original bounding box. Defining the active contexts to be name restriction free allows us to deal explicitly with all scope related issues in the usual way, i.e. using the structural congruence, shown in Figure 13, to migrate name restrictions in and out of redexes as required.

Contrary to the usual practice we allow constant introductions (νa) to migrate in and out of non-deterministic external choice constructs in much the same way as is customary for parallel composition. This is necessary because the rules of our reaction semantics are implicitly going to assume the normal form

$$(\ldots(((\lfloor(\ldots(((\mathbf{M}\,.\,P_i + P_i') \mid P_i'') + P_i''') \mid P_i'''' \ldots) \rfloor + P_o') \mid P_o'') + P_o''') \mid P_o'''' \ldots) \tag{3}$$

for the constituents of redexes of movement actions, and

$$(\ldots(((\lfloor(\ldots(((\mathbf{M}\,.\,P_i + P_i') \mid P_i'') + P_i''') \mid P_i'''' \ldots)\rfloor + P_o') \mid P_o'') + P_o''') \mid P_o'''' \ldots) \tag{4}$$

(where the grey symbols denote syntax that may, or may not, be present) for the constituents of redexes of communication actions. In each of these cases the congruence must be strong enough to migrate an obstructing name restriction out of the way, if appropriate.

The choice **(ii)** is required to ensure that rules of the reaction semantics, shown in Figure 14, recognise and alter redexes correctly. All redexes have two constituents, one exposing a capability prefix and another exposing the corresponding co-capability prefix. As mentioned, these constituents can always be assumed to be of one of the forms (3) or (4), which implies that there are some cases where exactly

Scope rules for namebindings:

$$(\nu a)\,\mathbf{0} \equiv \mathbf{0} \qquad (\nu a)\,(P \mid P') \equiv ((\nu a)\,P) \mid P' \quad \text{if } a \notin \mathsf{fn}(P')$$
$$(\nu a_1)\,(\nu a_2)\,P \equiv (\nu a_2)\,(\nu a_1)\,P \qquad (\nu a)\,(P + P') \equiv ((\nu a)\,P) + P' \quad \text{if } a \notin \mathsf{fn}(P')$$
$$(\nu a)\,(\lfloor\, P \,\rfloor) \equiv \lfloor\, (\nu a)\,P \,\rfloor$$

Unfolding of recursion:

$$\mathsf{rec}\,X.\,P \equiv P[\mathsf{rec}\,X.\,P/X]$$

α-renaming:

$$\frac{P \equiv_\alpha Q}{P \equiv Q}$$

Congruence requirements:

$$P \equiv P \qquad \frac{P \equiv Q}{Q \equiv P} \qquad \frac{P \equiv Q \quad Q \equiv R}{P \equiv R}$$
$$\frac{P \equiv Q}{\mathbb{C}[\,P\,] \equiv \mathbb{C}[\,Q\,]} \qquad \frac{P \equiv Q}{\lfloor\, P \,\rfloor \equiv \lfloor\, Q \,\rfloor} \qquad \frac{P \equiv Q}{(\nu a)\,P \equiv (\nu a)\,Q}$$

Fig. 13. Structural congruence $P \equiv Q$ for BioAmbients.

one boundary is demanded to enclose the exposed prefix, and other cases where no boundaries are allowed. Defining the active evaluation contexts to be ambient boundary free allows us to easily match each of these cases in the following manner:

(I) If no ambient boundary is allowed, the constituent is simply a capability prefixed process expression enclosed in an active evaluation context, which we match by $\mathbb{C}[\,\mathbf{M}\,.\,P\,]$.

(II) If exactly one ambient boundary is demanded, the constituent is an expression of the form **(I)** enclosed in an ambient boundary construct and a further active evaluation context, which we match by $\mathbb{C}_1\lfloor\, \mathbb{C}_2[\,\mathbf{M}\,.\,P\,] \,\rfloor$.

As illustrated by Figure 14, where the active contexts are toned down, systematic application of these patterns allows us to focus entirely on the high level structure of redexes and contractums while the contexts conveniently hide the details of redex constituents as well as reactions.

Finally, the choice **(iii)** completely separates the notion of recursion from that of the active evaluation contexts. As a result recursion is easily handled in the usual manner, i.e. using the structural congruence to unfold recursive processes as required.

5 Related Work

Employing evaluation contexts to express semantics of process calculi is not a new idea.

Berry and Boudol [2] use *program contexts* to denote the arbitrary testing environments that form the basis of semantic equivalence in CHAM. Later authors, such as Milner [13], use a similar (derived) notion of *process contexts*, primarily in order to extend equivalences to congruences. A few authors, such as Godskesen, Hildebrandt, and Sasone [6] for the Calculus of Mobile Resources, also use similar derived notions (*path contexts*, *evaluation contexts*, *resource contexts* etc.) to define the actual reaction relation of their calculi. In all cases, however, the involved notions of context are (standard) static ones and none of the authors address the issue of combining general choice with parallelism.

Sewell [20] makes a radically different use of contexts. He shows how to auto-

Movement of ambients:

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3' \quad \mathbb{C}_4 \longrightarrow \mathbb{C}_4'}{\mathbb{C}_1[\,\mathbb{C}_2[\,\mathsf{enter}\ a\,.\,P\,]\,]\,|\,\mathbb{C}_3[\,\mathbb{C}_4[\,\mathsf{accept}\ a\,.\,Q\,]\,] \longrightarrow \mathbb{C}_1'[\,\mathbf{0}\,]\,|\,\mathbb{C}_3'[\,\mathbb{C}_2'[\,P\,]\,|\,\mathbb{C}_4'[\,Q\,]\,]}$$

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3'}{[\,\mathbb{C}_1[\,\mathbb{C}_2[\,\mathsf{exit}\ a\,.\,P\,]\,]\,|\,\mathbb{C}_3[\,\mathsf{expel}\ a\,.\,Q\,]\,] \longrightarrow [\,\mathbb{C}_2'[\,P\,]\,]\,|\,[\,\mathbb{C}_1'[\,\mathbf{0}\,]\,|\,\mathbb{C}_3'[\,Q\,]\,]}$$

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3' \quad \mathbb{C}_4 \longrightarrow \mathbb{C}_4'}{\mathbb{C}_1[\,\mathbb{C}_2[\,\mathsf{merge}{-}\ a\,.\,P\,]\,]\,|\,\mathbb{C}_3[\,\mathbb{C}_4[\,\mathsf{merge}{+}\ a\,.\,Q\,]\,] \longrightarrow \mathbb{C}_1'[\,\mathbf{0}\,]\,|\,\mathbb{C}_3'[\,\mathbb{C}_2'[\,P\,]\,|\,\mathbb{C}_4'[\,Q\,]\,]}$$

Communication between ambients:

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2'}{\mathbb{C}_1[\,a!\{b\}\,.\,P\,]\,|\,\mathbb{C}_2[\,a?\{c\}\,.\,Q\,] \longrightarrow \mathbb{C}_1'[\,P\,]\,|\,\mathbb{C}_2'[\,Q[m/p]\,]}$$

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3'}{\mathbb{C}_1[\,a\_!\{b\}\,.\,P\,]\,|\,\mathbb{C}_2[\,\mathbb{C}_3[\,a\hat{}?\{c\}\,.\,Q\,]\,] \longrightarrow \mathbb{C}_1'[\,P\,]\,|\,\mathbb{C}_2'[\,\mathbb{C}_3'[\,Q[m/p]\,]\,]}$$

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3'}{\mathbb{C}_1[\,\mathbb{C}_2[\,a\hat{}!\{b\}\,.\,P\,]\,]\,|\,\mathbb{C}_3[\,a\_?\{c\}\,.\,Q\,] \longrightarrow \mathbb{C}_1'[\,\mathbb{C}_2'[\,P\,]\,]\,|\,\mathbb{C}_3'[\,Q[m/p]\,]}$$

$$\frac{\mathbb{C}_1 \longrightarrow \mathbb{C}_1' \quad \mathbb{C}_2 \longrightarrow \mathbb{C}_2' \quad \mathbb{C}_3 \longrightarrow \mathbb{C}_3' \quad \mathbb{C}_4 \longrightarrow \mathbb{C}_4'}{\mathbb{C}_1[\,\mathbb{C}_2[\,a\#!\{b\}\,.\,P\,]\,]\,|\,\mathbb{C}_3[\,\mathbb{C}_4[\,a\#?\{c\}\,.\,Q\,]\,] \longrightarrow \mathbb{C}_1'[\,\mathbb{C}_2'[\,P\,]\,]\,|\,\mathbb{C}_3'[\,\mathbb{C}_4'[\,Q[m/p]\,]\,]}$$

Execution in context:

$$\frac{\mathbb{C} \longrightarrow \mathbb{C}' \quad P \longrightarrow Q}{\mathbb{C}[\,P\,] \longrightarrow \mathbb{C}'[\,Q\,]} \qquad \frac{P \longrightarrow Q}{(\nu a)\,P \longrightarrow (\nu a)\,Q} \qquad \frac{P \longrightarrow Q}{[\,P\,] \longrightarrow [\,Q\,]}$$

Structural congruence:

$$\frac{P \equiv Q \quad Q \longrightarrow Q' \quad Q' \equiv P'}{P \longrightarrow P'}$$

Fig. 14. Reaction relation of BioAmbients.

matically derive labelled transition systems from a variety of rewrite semantics by simply using suitable contexts as transition labels whenever reaction occurs. This allows operational equivalences, as provided by the reaction semantics, to be investigated in a (presumably) nicer labelled setting. The involved notion of context is not related to ours and calculi with choice are not considered at all.

Larsen [8] uses contexts equipped with structural operational semantics to define a notion of context dependent equivalence. Larsen and Xinxin [9] extends this into a notion of compositionality that allows Hennesy-Milner properties of composite systems to be decomposed into joint properties of the sub-components. This use of active contexts has subsequently been adopted back into the realm of functional languages by Sands [19]. In all cases the contexts are, in some sense, active, but the associated semantics is defined using exactly the complicated label languages that reaction semantics strive to avoid and, in purpose, the approach is unrelated to ours.

6 Conclusion

We have developed the notion of *active evaluation contexts* that allows reaction semantics in the style of the Chemical Abstract Machine [2] to be defined for a larger class of process algebras than has previously been considered.

In line with previous work on reaction semantics for CCS [13] we have compared our approach to the more classical approach of structural operational semantics [11] and proved that the two types of semantics coincide when closed process expressions

are considered. This result indicates that the notion of active evaluation contexts constitutes a *sound* approach to reaction semantics.

In order to illustrate our approach on more expressive calculi, such as those that arise to meet the demands of domain specific modelling for complex domains, we have presented a full reaction semantics for an extension of Regev and Cardelli's comprehensive BioAmbients calculus [18] that includes unrestricted choice. The resulting semantics has two properties that we find very encouraging. Firstly the process of actually defining it was highly systematic and, thus, easy. Secondly we find that it is comparable in elegance to Regev's original semantics. This indicates that the notion of active evaluation contexts also constitutes a *sensible* approach to reaction semantics.

Thus, we believe that active evaluation contexts constitute a sound and sensible approach to defining reaction semantics in general. We can only fully substantiate this claim, however, by subjecting other advanced calculi, which combine various features in new ways, to the approach.

References

[1] Bergstra, J. A. and J. W. Klop, *Algebra of communicating processes*, in: *Proc. of CWI Symposium*, Mathematics and Computer Science (1986), pp. 89–138.

[2] Berry, G. and G. Boudol, *The chemical abstract machine*, in: *POPL '90: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages* (1990), pp. 81–94.

[3] Cardelli, L., *Bioware languages*, in: A. Herbert and K. S. Jones, editors, *Computer Systems: Theory, Technology, and Applications - A Tribute to Roger Needham*, Monographs in Computer Science, Springer, 2004 pp. 59–65.

[4] Cardelli, L. and A. D. Gordon, *Mobile Ambients*, Theoretical Computer Science **240** (2000), pp. 177–213.

[5] Felleisen, M. and D. P. Friedman, *Control operators, the SECD-machine, and the λ-calculus*, in: M. Wirsing, editor, *Formal Descriptions of Programming Concepts III*, North-Holland, 1986 pp. 193–219.

[6] Godskesen, J. C., T. Hildebrandt and V. Sassone, *A calculus of mobile resources*, in: L. Brim, P. Jancar, M. ir Kretinsky and A. in Kucera, editors, *CONCUR 13 – Concurrency Theory*, LNCS **2421** (2002), pp. 272–287.
URL citeseer.ist.psu.edu/article/godskesen02calculus.html

[7] Hoare, C. A. R., *Communicating sequential processes*, Commun. ACM **21** (1978), pp. 666–677.

[8] Larsen, K. G., *A context dependent equivalence between processes*, Theor. Comput. Sci. **49** (1987), pp. 185–215.

[9] Larsen, K. G. and L. Xinxin, *Compositionality through an operational semantics of contexts*, Journal of Logic and Computation **1** (1991), pp. 761–795.

[10] Levi, F. and D. Sangiorgi, *Controlling interference in ambients*, in: *Proc. of POPL'2000* (2000), pp. 352–364.

[11] Milner, R., "A Calculus of Communicating Systems," Lecture Notes in Computer Science Vol. 92, Springer-Verlag, 1980.

[12] Milner, R., "Communication and Concurrency," Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[13] Milner, R., "Communicating and Mobile Systems: The π-Calculus," Cambridge University Press, 1999.

[14] Nielson, H. R., F. Nielson and M. Buchholtz, *Security for mobility*, in: *FOSAD 2001/2002 Tutorial Lecture Notes*, LNCS **2946**, Springer, 2004 pp. 207–265.

[15] Plotkin, G. D., *A Structural Approach to Operational Semantics*, Technical Report DAIMI FN-19, University of Aarhus (1981).
URL citeseer.csail.mit.edu/plotkin81structural.html

[16] Plotkin, G. D., *Personal communication to flemming nielson*, Aarhus (1981).

[17] Regev, A., "Computational Systems Biology: A Calculus for Biomolecular Knowledge," Ph.D. thesis, Tel Aviv University (2003).

[18] Regev, A., E. M, Pamina, W. Silverman, L. Cardelli and E. Shapiro, *BioAmbients: An abstraction for biological compartments*, Theoretical Computer Science **325** (2004), pp. 141–167.

[19] Sands, D., *Towards operational semantics of contexts in functional languages*, in: *Proceedings of the 6th Nordic Workshop on Programming Theory*, 1994, pp. 378–385.
URL citeseer.ist.psu.edu/68354.html

[20] Sewell, P., *From rewrite rules to bisimulation congruences*, Theoretical Computer Science **274** (2002), pp. 183–230.
URL citeseer.ist.psu.edu/sewell98from.html