

Deriving structural labelled transitions for mobile ambients

Julian Rathke and Paweł Sobociński*

ECS, University of Southampton

Abstract. We present a new labelled transition system (LTS) for the ambient calculus on which ordinary bisimilarity coincides with contextual equivalence. The key feature of this LTS is that it is the fruit of ongoing work on developing a systematic procedure for deriving LTSS in the structural style from the underlying reduction semantics and observability. Notably, even though we have derived our LTS for ambients systematically it compares very favourably with existing transition systems for the same calculus.

Introduction

Since the introduction of the archetypal process calculi, CCS [20], CSP [14] and ACP [2], and the π -calculus [9, 21] some years ago there has been a proliferation of calculus extensions and variants which address assorted computational features. One concern that is often voiced regarding these extended calculi is that their semantics, particularly their labelled transition semantics, are often ad hoc and heavily locally optimised. This state of affairs is unsatisfactory and initial attempts to address the issue were made in [17, 26] where it was proposed that labelled transitions should be *derived* (rather than defined) by means of considering underlying reduction rules for the language and taking labels to be suitably minimal contexts which induce reductions. The rationale for this is that for any new computational feature, its reduction rules are generally easier to define uncontentionally and can be taken to be definitional. Consequently the derived labelled transitions would be given without further design. This approach is appealing but Sewell's early results [26] were limited in their scope. Leifer and Milner generalised the approach with some degree of success [17]. A general definition of contexts-as-labels was provided using the universal property of (relative) pushouts to obtain a suitable notion of minimality. Even so, this work still has its problems, the chief of which is that the derived labelled transition systems are not presented in an inductive manner and are therefore often difficult to describe and reason with.

It is easy to lose sight of the fact that the original intention of structural operational semantics [23] and labelled transition systems [20] was to provide an inductive definition of the reduction relation for a language. Their subsequent

* Research partially supported by EPSRC grant EP/D066565/1.

use as points of comparison of interaction in bisimulation equivalences has allowed focus to drift away from inductively defined labelled transition systems and on to labels as the contextually observable parts of interaction. Our general research goal is to provide a method by which structurally defined labelled transition systems can be derived from an underlying reduction semantics. For this derived transition system, bisimulation equivalence must also correspond to a contextually defined equivalence. This task is difficult and we have begun by evaluating our ideas for simple process calculi. The results of such an experiment for the π -calculus appear in [24].

In this paper we apply our method to the ambient calculus of Cardelli and Gordon [7]. The ambient calculus has enjoyed success as a foundational model of distributed processes. It essentially comprises hierarchically arranged processes which can migrate, as well as dynamically modify the structure of their location. Our interest is not in the ambient calculus as a model of distributed computing per se but simply as a small calculus with an interesting set of reduction rules for which it has thus far proven difficult to provide a definitive labelled transition system and bisimulation equivalence [5, 18]. Our purpose is not necessarily to improve or undermine the existing labelled transition systems but to systematically derive one.

The approach we take is to consider the underlying ground rewrite rules of the language as structural rewrites supplied with suitable ground parameters. For any term which partially matches the left-hand-side (LHS) of a rewrite, we identify the parameters supplied by the term to the match. A label represents the remaining structure of the LHS of the rewrite rule along with the missing parameters which will be supplied by an interacting context. This separation of the structure of the rewrite and the parameters to the rule allows us to build our labelled transition systems in three steps: we define the process-view transitions, whose main purpose is to provide an inductively defined reduction relation, then the context-view transitions, which allows for a context to supply parameters to an interaction, and finally rules to combine them. Technically, we make use of the simply typed λ -calculus as a meta-language for abstracting parameters.

Structure of the paper. We present the syntax and semantics of the ambient calculus, along with a suitable contextually defined equivalence, in the next section. We then give an account of our method of deriving labelled transitions and show its instantiation for the ambient calculus in Section 2. Section 3 lists the properties of the LTS: bisimulation equivalence is proved to be sound for reduction barbed congruence and, after the addition of Honda-Tokoro [15, 25] style rules to account for unobservability of certain actions, complete. We include a comparison with related work in Section 4 and close with some concluding remarks regarding future work. Due to space constraints, detailed proofs have been omitted.

1 Ambients: syntax, metasyntax and inductive semantics

The grammars for types and terms, together with the type rules are given in Fig. 1. Note that the λ -calculus operators are part of the meta-language. They

$$\begin{array}{c}
\sigma ::= \mathbf{N} \mid \mathbf{Pr} \mid \sigma \rightarrow \sigma \\
M ::= \mathbf{X} \mid m \mid 0 \mid M \parallel M \mid M[M] \mid \nu m M \mid \text{out } M.M \mid \text{in } M.M \mid \text{open } M.M \mid [\lambda X:\sigma. M \mid M(M)] \\
\frac{}{\Gamma \vdash m : \mathbf{N}} (: \mathbf{NM}) \quad \frac{\Gamma(\mathbf{X}) = \sigma}{\Gamma \vdash \mathbf{X} : \sigma} (: \mathbf{VAR}) \quad \frac{}{\Gamma \vdash 0 : \mathbf{Pr}} (: \mathbf{0}) \\
\frac{\Gamma \vdash M : \mathbf{Pr} \quad \Gamma \vdash N : \mathbf{Pr}}{\Gamma \vdash M \parallel N : \mathbf{Pr}} (: \parallel) \quad \frac{\Gamma \vdash k : \mathbf{N} \quad \Gamma \vdash M : \mathbf{Pr}}{\Gamma \vdash k[M] : \mathbf{Pr}} (: \mathbf{AMB}) \quad \frac{\Gamma \vdash k : \mathbf{N} \quad \Gamma \vdash M : \mathbf{Pr}}{\Gamma \vdash \nu k M : \mathbf{Pr}} (: \nu) \\
\frac{\Gamma \vdash k : \mathbf{N} \quad \Gamma \vdash M : \mathbf{Pr}}{\Gamma \vdash \text{out } k.M : \mathbf{Pr}} (: \mathbf{OuPR}) \quad \frac{\Gamma \vdash k : \mathbf{N} \quad \Gamma \vdash M : \mathbf{Pr}}{\Gamma \vdash \text{in } k.M : \mathbf{Pr}} (: \mathbf{INPR}) \quad \frac{\Gamma \vdash k : \mathbf{N} \quad \Gamma \vdash M : \mathbf{Pr}}{\Gamma \vdash \text{open } k.M : \mathbf{Pr}} (: \mathbf{OpPR}) \\
\frac{\Gamma, X:\sigma \vdash M : \sigma'}{\Gamma \vdash \lambda X:\sigma. M : \sigma \rightarrow \sigma'} (: \lambda) \quad \frac{\Gamma \vdash M : \sigma \rightarrow \sigma' \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M(N) : \sigma'} (: \mathbf{APP})
\end{array}$$

Fig. 1. Types, syntax and type rules.

$$\begin{array}{c}
(P \parallel Q) \parallel R \equiv P \parallel (Q \parallel R) \quad P \parallel Q \equiv Q \parallel P \quad P \parallel 0 \equiv P \quad \nu m P \equiv \nu n P[n/m] \quad \nu m \nu n P \equiv \nu n \nu m P \\
\nu m 0 \equiv 0 \quad \nu m (P \parallel Q) \equiv P \parallel \nu m Q \quad (m \notin \text{fr}(P)) \quad \nu m (n[P]) \equiv n[\nu m P] \quad (m \neq n) \\
(\lambda X:\sigma. M)(N) \equiv M[N/X] \quad \lambda X:\sigma. M \equiv \lambda Y:\sigma. M[Y/X] \quad (Y \notin \text{fr}(M))
\end{array}$$

Fig. 2. Structural congruence.

$$\begin{array}{c}
\frac{}{m[\text{in } n.P \parallel Q] \parallel n[R] \rightarrow n[m[P \parallel Q] \parallel R]} \quad \frac{}{n[m[\text{out } n.P \parallel Q] \parallel R] \rightarrow m[P \parallel Q] \parallel n[R]} \\
\frac{}{\text{open } n.P \parallel n[Q] \rightarrow P \parallel Q} \quad \frac{P \rightarrow P'}{P \parallel Q \rightarrow P' \parallel Q} \quad \frac{P \rightarrow P'}{\nu n P \rightarrow \nu n P'} \quad \frac{P \rightarrow P'}{n[P] \rightarrow n[P']}
\end{array}$$

Fig. 3. Reduction semantics, inductively.

$$M ::= \dots \mid -\sigma \mid (M, \dots, M) \quad \frac{}{\Gamma \vdash -\sigma : \sigma} (: \mathbf{HOLE}) \quad \frac{\vdash V_1 : \sigma_1 \dots \vdash V_n : \sigma_n \quad (n \in \mathbf{N})}{\vdash (V_1, \dots, V_n) : [\sigma_1 \dots \sigma_n]} (: \mathbf{TUP}).$$

Fig. 4. Interface types.

$$\begin{array}{c}
Sk_n^{in} \stackrel{\text{def}}{=} (l_n^{in} \stackrel{\text{def}}{=} 1_{\mathbf{N}}[\text{in } n.2_{\mathbf{Pr}} \parallel 3_{\mathbf{Pr}}] \parallel n[4_{\mathbf{Pr}}], r_n^{in} \stackrel{\text{def}}{=} n[1_{\mathbf{N}}[2_{\mathbf{Pr}} \parallel 3_{\mathbf{Pr}}] \parallel 4_{\mathbf{Pr}}]) \\
Sk_n^{out} \stackrel{\text{def}}{=} (l_n^{out} \stackrel{\text{def}}{=} n[1_{\mathbf{N}}[\text{out } n.2_{\mathbf{Pr}} \parallel 3_{\mathbf{Pr}}] \parallel 4_{\mathbf{Pr}}], r_n^{out} \stackrel{\text{def}}{=} 1_{\mathbf{N}}[2_{\mathbf{Pr}} \parallel 3_{\mathbf{Pr}}] \parallel n[4_{\mathbf{Pr}}]) \\
Sk_n^{open} \stackrel{\text{def}}{=} (l_n^{open} \stackrel{\text{def}}{=} \text{open } n.1_{\mathbf{Pr}} \parallel n[2_{\mathbf{Pr}}], r_n^{open} \stackrel{\text{def}}{=} 1_{\mathbf{Pr}} \parallel 2_{\mathbf{Pr}})
\end{array}$$

Fig. 5. Skeletons $l_n^{in}, r_n^{in} : [\mathbf{N}, \mathbf{Pr}^3] \rightarrow [\mathbf{Pr}]$, $l_n^{out}, r_n^{out} : [\mathbf{N}, \mathbf{Pr}^3] \rightarrow [\mathbf{Pr}]$ and $l_n^{open}, r_n^{open} : [\mathbf{Pr}^2] \rightarrow [\mathbf{Pr}]$.

are used here solely to define the labelled transition system and should not be considered as a language extension. We assume distinct countable supplies of names (ranged over by n, m) and variables (ranged over by X, Y, x, y). By convention, we will use x, y for variables of type \mathbf{N} , X, Y for variables of type \mathbf{Pr} , k, l for terms of type \mathbf{N} and P, Q, R for closed terms of type \mathbf{Pr} . M, N will be used for arbitrary terms of type \mathbf{Pr} . A type context Γ is a finite map from variable names to types. We consider only typeable terms. The axioms of structural congruence are given in *Fig. 2*. It is straightforward to show that any term N structurally congruent to M is typeable iff M is and that they have the same type.

Our transition systems are presented in the structural style. We make one non-standard assumption: we always assume the implicit presence of the rule

$$\frac{P' \equiv P \quad P \xrightarrow{\alpha} Q \quad Q \equiv Q'}{P' \xrightarrow{\alpha} Q'} \text{ (STRCNG)}.$$

The reduction semantics is given in *Fig. 3*. It is easy to show that subject reduction holds.

Before we proceed it is worth pointing out that our language does not contain any replication or recursion operator and is thus finite. This however is not a significant restriction though as the crafting of a labelled transition system relies on a study of the immediate interactions of a process with a context. We could easily, but pointlessly, include a replication operator with negligible impact on the LTS rules. Given an LTS \mathcal{L} the only labelled equivalence we shall consider is standard strong bisimilarity $\sim_{\mathcal{L}}$, which is defined as the largest bisimulation on \mathcal{L} . Because we wanted to focus on the systematic derivation procedure of LTSS, we have not considered weak equivalences in this paper; our feeling is that the study of weak equivalences examines largely orthogonal issues. We will come back to this issue in the section on future work.

To explain our derivation procedure we will need a general notion of context. Contexts are defined in two stages.

Definition 1 (Precontext) For each type σ , we add σ -annotated holes $-_{\sigma}$ and n -tuples (for any $n \in \mathbb{N}$) to the syntax, together with two additional type rules, given in *Fig. 4*, where $[\vec{\sigma}]$ is called an *interface type*. A precontext is a typeable term of the form (v_1, \dots, v_n) . Note that each v_i is a closed term in that it does not contain free variables; holes and variables are separate syntactic entities.

Definition 2 (Context) Suppose that a precontext $(\vec{v}) : [\vec{\sigma}_1]$ contains m holes. A 1-1 enumeration of the holes with numbers from 1 to m uniquely determines a word $\vec{\sigma}_2$ over types, where $\sigma_{2,i}$ is the type of the i th numbered hole. Syntactically replacing each hole with its number yields a *context* of type $[\vec{\sigma}_2] \rightarrow [\vec{\sigma}_1]$. Ordinary terms of type \mathbf{Pr} will be identified with contexts of type $\square \rightarrow [\mathbf{Pr}]$. Given contexts $f : [\vec{\sigma}_1] \rightarrow [\vec{\sigma}_2]$ and $g : [\vec{\sigma}_2] \rightarrow [\vec{\sigma}_3]$, there is a context $g \circ f : [\vec{\sigma}_1] \rightarrow [\vec{\sigma}_3]$ which is obtained by substitution of the i th component of f for the i th hole of g . This operation may be capturing. Given $f : [\vec{\sigma}_1] \rightarrow [\vec{\sigma}_2]$ and $g : [\vec{\sigma}_3] \rightarrow [\vec{\sigma}_4]$ let

$f \otimes g : [\vec{\sigma}_1 \vec{\sigma}_3] \rightarrow [\vec{\sigma}_2 \vec{\sigma}_4]$ be the context which puts f and g “side-by-side”, where the numbering of all the holes in g are incremented by the length of $\vec{\sigma}_1$. Moreover:

- for any word $\vec{\sigma} = \sigma_1 \dots \sigma_k$, the *identity* context $\text{id}_{[\vec{\sigma}]} : [\vec{\sigma}] \rightarrow [\vec{\sigma}]$ is $(1_{\sigma_1}, \dots, k_{\sigma_k})$;
- if, given $\vec{\sigma}$ and $\vec{\sigma}'$, there exists a permutation $\rho : k \rightarrow k$ such that $\forall 1 \leq i \leq k. \sigma_{\rho i} = \sigma'_i$, it induces a *permutation context* $\rho : [\vec{\sigma}_1] \rightarrow [\vec{\sigma}_2]$ of the form $(\rho 1_{\sigma'_1}, \dots, \rho k_{\sigma'_k})$;
- a *language context* is a context which does not contain instances of the metalanguage; those of type $[\text{Pr}] \rightarrow [\text{Pr}]$ will be denoted by \mathcal{C} ;

For language contexts we will write $\mathcal{C}[M]$ for $\mathcal{C} \circ M$.

Definition 3 (Barbs) We say that a term P *barbs* on an ambient m , written $P \downarrow_m$, if there is an instance of an ambient m at the “top level”. More formally, $P \equiv \nu \vec{n} (m[Q] \parallel R)$ for some \vec{n}, Q, R such that m does not appear in \vec{n} .

Definition 4 (Reduction barb congruence) Reduction barb congruence (\simeq) is the largest symmetric relation \mathcal{R} such that if $P \mathcal{R} Q$ then:

- (i) If $P \rightarrow P'$ then there exists $Q \rightarrow Q'$ such that $P' \mathcal{R} Q'$;
- (ii) if $P \downarrow_m$ then $Q \downarrow_m$;
- (iii) for all language contexts \mathcal{C} we have that $\mathcal{C}[P] \mathcal{R} \mathcal{C}[Q]$.

2 Derivation of a structural LTS

The chief novelty of our paper is the *systematic* presentation of a novel LTS for the ambient calculus. Therefore, before we present the LTS we give an account of our derivation procedure. First, we consider the reduction rules of *Fig. 3* as parameterised rules. *Fig. 5* contains a rendering of these parameterised reduction rules, referred to as *skeletons*. Essentially, a skeleton is a pair of contexts (l_n^α, r_n^α) which describe the structural changes in passing from l_n^α to r_n^α .

Our LTS is organised into three components: the process-view, in *Fig. 6*, the context-view, in *Fig. 7*, and the combined system in *Fig. 8*. The context-view is the simplest of these and consists of a single “applicative” rule. In the remainder of this section we describe how to analyse the skeletons in order to obtain the process-view rules and how this combines with the context-view.

2.1 Derivation procedure: axioms

Treating the skeletons of *Fig. 5* as syntax trees, we say that a *match* for Sk_n^α is a subtree with root of type Pr of the LHS l_n^α . More formally, if $l_n^\alpha : [\vec{\sigma}] \rightarrow [\text{Pr}]$ is the LHS of a skeleton, a match is a term $\mu_n^\alpha : [\vec{\sigma}_1] \rightarrow [\text{Pr}]$ such that there exists $\vec{\sigma}_2$ so that $\vec{\sigma}_1 \vec{\sigma}_2$ is a permutation of $\vec{\sigma}$, and there exists a context $\chi : [\text{Pr}, \vec{\sigma}_2] \rightarrow [\text{Pr}]$ satisfying (\dagger) where $\rho : [\vec{\sigma}_1 \vec{\sigma}_2] \rightarrow [\vec{\sigma}]$ is the permutation context. A match is said to be *active* if there *does not* exist a context $\chi' : [\text{Pr}, \vec{\sigma}_2] \rightarrow [\text{Pr}]$ satisfying (\ddagger) .

$$\chi \circ (\mu_n^\alpha \otimes \text{id}_{[\vec{\sigma}_2]}) = l_n^\alpha \circ \rho \quad (\dagger) \qquad \chi' \circ (\mu_n^\alpha \otimes \text{id}_{[\vec{\sigma}_2]}) = r_n^\alpha \circ \rho \quad (\ddagger)$$

Intuitively, an active match is a part of the LHS of the skeleton that is modified as a result of the reduction. Clearly any match which has an active match as a subtree is itself active. Of particular interest are those active matches which are locally *minimal* with respect to the subtree relation.

Observation 5 *The minimal active matches for the skeletons in Fig. 5 are:*

- for $Sk_n^{in} : \text{in } n.1_{Pr} \text{ and } n[1_{Pr}]$;
- for $Sk_n^{out} : \text{out } n.1_{Pr}$;
- for $Sk_n^{open} : \text{open } n.1_{Pr} \text{ and } n[1_{Pr}]$.

The axioms of our process-view LTS are determined by the minimal active matches. Indeed, their LHSS are the *instantiated* minimal active matches: given a minimal active match $\mu_n^\alpha : [\vec{\sigma}] \rightarrow [Pr]$ they are the terms $\mu_n^\alpha \circ \iota$ where $\iota : [] \rightarrow [\vec{\sigma}]$. The result is then the RHS of the skeleton instantiated with the parameters ι of the minimal match together with that remaining parameters ι' required by χ :

$$\mu_n^\alpha \circ \iota \xrightarrow{\chi \circ (1_{Pr} \otimes \iota')} r_n^\alpha \circ \rho \circ (\iota \otimes \iota'). \quad (1)$$

This is clearly an RPO-like observation: the context provides $\chi \circ (1_{Pr} \otimes \iota')$ and enables a reduction $\chi \circ (1_{Pr} \otimes \iota') \circ \mu_n^\alpha \circ \iota = \chi \circ (\mu_n^\alpha \circ \text{id}) \circ (\iota \otimes \iota') = l_n^\alpha \circ \rho \circ (\iota \otimes \iota') \rightarrow r_n^\alpha \circ \rho \circ (\iota \otimes \iota')$.

Note that each χ is uniquely determined by the particular minimal active match μ_n^α . For this reason in the label of the transition we will use a textual abbreviation $\alpha_i n | \vec{M}$ where $\alpha_i n$ represents the i th minimal active match of Sk_n^α , and \vec{M} the list of the remaining parameters (cf ι' in (1)). Following this procedure, we obtain the following labelled transitions:

$$\text{in } n.P \xrightarrow{\text{in}_1 n | Q k R} n[k[P||Q]||R] \quad n[P] \xrightarrow{\text{in}_2 n | Q R k} n[k[Q||R]||P] \quad (2)$$

$$\text{out } n.P \xrightarrow{\text{out}_1 n | Q k R} k[P||Q]||n[R] \quad (3)$$

$$\text{open } n.P \xrightarrow{\text{open}_1 n | Q} P||Q \quad n[P] \xrightarrow{\text{open}_2 n | Q} Q||P \quad (4)$$

The main obstacle in giving a structural derivation of an LTS with labels of the kind given above is that in the results of the above transitions, the distinction between the parts provided by the process and the parts provided by the context is lost. Our solution is to delay the instantiation of the context components. Technically this is done with the use of the meta-syntax – the context contributions are initially replaced with lambda abstracted variables.

The SOS rules are thus naturally divided into three parts - rules for the *process-view* LTS \mathcal{C} for deriving the part of the label to the left of the \downarrow symbol, rules for the *context-view* LTS \mathcal{A} for deriving the remainder of the label, and rules for the *combined* LTS \mathcal{CA} which juxtapose the two contributions to form “complete” labelled transitions. Following this nomenclature, the process-view contribution to the transitions in (2) is

$$\frac{}{\text{in } n.P \xrightarrow{\text{in}_1 n} \lambda X \lambda Y. n[x[P||X]||Y]} \text{(IN1)} \quad \frac{}{n[P] \xrightarrow{\text{in}_2 n} \lambda X \lambda Y. n[x[X||Y]||P]} \text{(IN2)} \quad (5)$$

while the context parts are given by rule (INST) of Fig. 7. The rule which juxtaposes them is (CA) of Fig. 8. We take (IN1), (IN2) (cf. 5), (OU1) (obtained from (3)), (OP1), (OP2) (obtained from (4)) as provisional axioms for the process-view LTS.

$$\begin{array}{c}
\frac{}{\text{in } n.P \xrightarrow{\text{in } n} \lambda X.XY. n[x[P\|X]\|Y]} \text{(IN)} \quad \frac{P \xrightarrow{\text{in } n} T}{P\|Q \xrightarrow{\text{in } n} \lambda X.T(Q\|X)} (\|\text{IN}) \quad \frac{P \xrightarrow{\text{in } n} T \quad m \neq n}{\nu m P \xrightarrow{\text{in } n} \nu m T} (\nu\text{IN}) \\
\\
\frac{P \xrightarrow{\text{in } n} T}{m[P] \xrightarrow{[\text{in } n]} T(0)(m)} \text{(INAMB)} \quad \frac{P \xrightarrow{[\text{in } n]} U}{P\|Q \xrightarrow{[\text{in } n]} U\|Q} (\|\text{INAMB}) \quad \frac{P \xrightarrow{[\text{in } n]} U \quad m \neq n}{\nu m P \xrightarrow{[\text{in } n]} \nu m U} (\nu\text{INAMB}) \\
\\
\frac{}{n[P] \xrightarrow{[\text{in } n]} \lambda Z.Z(P)} \text{(COIN)} \quad \frac{P \xrightarrow{[\text{in } n]} A}{P\|Q \xrightarrow{[\text{in } n]} A\|Q} (\|\text{COIN}) \quad \frac{P \xrightarrow{[\text{in } n]} A \quad m \neq n}{\nu m P \xrightarrow{[\text{in } n]} \nu m A} (\nu\text{COIN})
\end{array}$$

$$\begin{array}{c}
\frac{}{\text{out } n.P \xrightarrow{\text{out } n} \lambda X.XY. x[P\|X]\|n[Y]} \text{(OU)} \quad \frac{P \xrightarrow{\text{out } n} T}{P\|Q \xrightarrow{\text{out } n} \lambda X.T(Q\|X)} (\|\text{OU}) \quad \frac{P \xrightarrow{\text{out } n} T \quad m \neq n}{\nu m P \xrightarrow{\text{out } n} \nu m T} (\nu\text{OU}) \\
\\
\frac{P \xrightarrow{\text{out } n} T}{m[P] \xrightarrow{[\text{out } n]} T(0)(m)} \text{(OUAMB)} \quad \frac{P \xrightarrow{[\text{out } n]} U}{P\|Q \xrightarrow{[\text{out } n]} \lambda Y.U(Q\|Y)} (\|\text{OUAMB}) \quad \frac{P \xrightarrow{[\text{out } n]} U \quad m \neq n}{\nu m P \xrightarrow{[\text{out } n]} \nu m U} (\nu\text{OUAMB})
\end{array}$$

$$\begin{array}{c}
\frac{}{\text{open } n.P \xrightarrow{\text{open } n} \lambda X. P\|X} \text{(OP)} \quad \frac{P \xrightarrow{\text{open } n} U}{P\|Q \xrightarrow{\text{open } n} U\|Q} (\|\text{OP}) \quad \frac{P \xrightarrow{\text{open } n} U \quad m \neq n}{\nu m P \xrightarrow{\text{open } n} \nu m U} (\nu\text{OP}) \\
\\
\frac{}{n[P] \xrightarrow{\text{open } n} \lambda Z.Z(P)} \text{(COOP)} \quad \frac{P \xrightarrow{\text{open } n} A}{P\|Q \xrightarrow{\text{open } n} A\|Q} (\|\text{COOP}) \quad \frac{P \xrightarrow{\text{open } n} A \quad m \neq n}{\nu m P \xrightarrow{\text{open } n} \nu m A} (\nu\text{COOP})
\end{array}$$

$$\begin{array}{c}
\frac{P \xrightarrow{[\text{in } n]} U \quad Q \xrightarrow{[\text{in } n]} A}{P\|Q \xrightarrow{\tau} A(U)} \text{(INTAU)} \quad \frac{P \xrightarrow{[\text{out } n]} U}{n[P] \xrightarrow{\tau} U(0)} \text{(OUTAU)} \quad \frac{P \xrightarrow{\text{open } n} U \quad Q \xrightarrow{\text{open } n} A}{P\|Q \xrightarrow{\tau} A(U)} \text{(OPTAU)} \\
\\
\frac{P \xrightarrow{\tau} P'}{P\|Q \xrightarrow{\tau} P'\|Q} (\|\text{TAU}) \quad \frac{P \xrightarrow{\tau} P'}{\nu m P \xrightarrow{\tau} \nu m P'} (\nu\text{TAU}) \quad \frac{P \xrightarrow{\tau} P'}{n[P] \xrightarrow{\tau} n[P']} \text{(TAUAMB)}
\end{array}$$

Fig. 6. Process-view fragment (\mathcal{C}). By convention $T : \text{Pr} \rightarrow \mathbf{N} \rightarrow \text{Pr} \rightarrow \text{Pr}$, $U : \text{Pr} \rightarrow \text{Pr}$, $A : (\text{Pr} \rightarrow \text{Pr}) \rightarrow \text{Pr}$.

$$\frac{\overline{M} : \overline{\sigma}}{\lambda \overline{X} : \overline{\sigma}. P \xrightarrow{\overline{M} \downarrow} (\lambda \overline{X} : \overline{\sigma}. P)(\overline{M})} \text{(INST)}$$

Fig. 7. Context-view fragment (\mathcal{A}).

$$\begin{array}{c}
\frac{P \xrightarrow{\alpha}_C A \quad A \xrightarrow{\overline{M}\downarrow}_A P'}{P \xrightarrow{\alpha|\overline{M}} P'} \quad (\text{C}\lambda) \qquad \frac{P \xrightarrow{[\text{in } n]}_C A \quad A(\lambda XYZx. n[x[Y\|Z]\|X]) \xrightarrow{Q Rk\downarrow}_A P'}{P \xrightarrow{[\text{in } n]\downarrow Q Rk} P'} \quad (\text{coIN}\lambda) \\
\frac{P \xrightarrow{\tau}_C P'}{P \xrightarrow{\tau} P'} \quad (\text{CTAU}) \qquad \frac{P \xrightarrow{\overline{\text{open } n}}_C A \quad A(\lambda XY. Y\|X) \xrightarrow{Q\downarrow}_A P'}{P \xrightarrow{\overline{\text{open } n}\downarrow Q} P'} \quad (\text{coOP}\lambda)
\end{array}$$

Fig. 8. Combined system of complete actions (\mathcal{CA}).

2.2 Derivation procedure: structure

Once the axioms are determined, we can attempt to provide the structural rules. There are three kinds:

- (i) a *substructural* modification: the added structure takes part in the reduction but the match, and therefore the label, remain unchanged. The structure is added to the appropriate parameter in the RHS. A particular kind of substructural transition used here concerns the situation where the current match is in parallel with a hole of type Pr in the skeleton; *e.g.* the minimal active match of Sk_n^{out} . Using the fact that structural congruence ensures that $(\|, 0)$ is a commutative monoid, introducing a parallel component does not mean that we must expand the match, instead we add the component to the parameter representing the aforementioned hole;
- (ii) a *superstructural* modification: the match, and therefore the label, remain unchanged and the added structure does not take part in the reduction; it is added to the result at top level. This situation is common and therefore we will make use of the following abbreviations which deal with lambda abstractions $T = \lambda \vec{X}. P$:

$$T\|Q \stackrel{\text{def}}{=} \lambda \vec{X}. (T(\vec{X})\|Q) \quad \text{and} \quad \nu m T \stackrel{\text{def}}{=} \lambda \vec{X}. \nu m T(\vec{X}) \quad ;$$

- (iii) an *observational* modification: the extra structure forces the enlargement of the match as a subtree of its skeleton – here the label itself has to be changed. Once enough structure is added to cover the entire LHS of a skeleton, a τ -labelled transition should be derived. This can occur in two ways, depending on the number of the minimal active matches in the skeleton. These two cases are analysed in the two paragraphs below for the setting of the ambient calculus.

In Sk_n^{out} which has only one minimal active match, the procedure is relatively straightforward. The axiom (Ov) in *Fig. 6* is just (Ov1) as described previously, with the numeral omitted. The rule $(\|\text{Ov})$ is a substructural modification as described above. The rule (νOv) is a superstructural modification since the ν binder has to first migrate outside, using structural congruence, before the reduction can take place. The side condition enables this emigration. Note that because substitution that is part of β -reduction is capture avoiding, the binder in the result

will not bind any names when instantiated by combining with the context-view; the correct behaviour. The rule (OUAMB) is an observational modification, here the structure (the ambient n) forces us to expand the match within the skeleton, meaning that we can now instantiate the first two parameters. The rule $(\|\text{OUAMB})$ is substructural while (νOUAMB) is superstructural. Finally, (OUTAU) is an observational modification which completes the skeleton, meaning that we derive a τ -labelled transition.

Skeletons with two (or more) minimal active matches lead to a more complicated situation. Indeed, consider the two minimal active matches of Sk_n^{in} and the two corresponding provisional axioms given in (5). Starting with either one, structure can be added extending the match. Indeed, consider (IN) of *Fig. 6* which is obtained from (IN1) of (5) by omitting the numeral. The rule $(\|\text{IN})$ is substructural and (νIN) superstructural. The rule (INAMB) is observational and extends the minimal match with the surrounding ambient. No further extension of the match is possible without including the contribution of the second minimal active match. The structural approach requires the combination of observations of the two matches in order to cover the entire LHS of the skeleton and derive a τ . However, in our two provisional axioms (IN1) , (IN2) we have included the RHS of the skeleton in result of the transitions, and it is not obvious how to “merge” the two, collecting the appropriate parameters. Our solution is to use *co-actions*, borrowing continuation-passing style. Indeed, we discard (IN2) and instead use the axiom (COIN) of *Fig. 6*. The idea is that instead of using the actual skeleton in the result, we use an abstract skeleton and apply that to the parameter (of the minimal active match). Merging actions and co-actions is now easy as the abstract skeleton can be replaced by the the actual skeleton provided by the action. Superstructural rules $(\|\text{COOP})$ and (νCOOP) are straightforward and we are able to use (INTAU) to collect the parameters to the RHS of the skeleton using a simple application. A similar approach is used to deal with the *open* reduction.

The use of co-actions gives one final complication. Because the result of a co-action transition does not have the shape which would result from using the RHS of the skeleton, we cannot simply use the the combination of (INST) of *Fig. 7* and $(\text{C}\lambda)$ of *Fig. 8*. Instead, we use the rules $(\text{COIN}\lambda)$ and $(\text{COOP}\lambda)$ which insist that the context provided by the environment conforms to the skeleton.

It is worth clarifying as to what extent the procedure, as described above, is systematic. As we have explained, we have chosen to include the RHS in the right hand side of (IN1) , resulting in (IN) . Differently, and in seemingly ad-hoc fashion, we have not done this for (IN2) , using instead a co-action (COIN) . A more systematic presentation would consist in using the co-action style for *all* the labels. Following this approach, the actual skeleton would never actually be instantiated in the RHS of the process-view transitions. The main price for this is that the rule (INST) needs to be replaced with specific rules for each co-action, in the spirit of $(\text{COIN}\lambda)$ and $(\text{COOP}\lambda)$ of *Fig. 8*. Such an “all-co-action” SOS rule set would derive the same LTS as the rule set presented in this paper. We believe that this approach could be mechanised. We have chosen to present the rules as in *Fig. 6* because we believe that they are easier to understand, and more

importantly, they correspond more closely to rules in previously published SOS rule sets for the ambient calculus (*cf.* Section 4).

The following lemma provides a sanity-check for our LTS which ensures that the transitions obtained from our structural rules are justified by a reduction in a context; the point with which we started our discussion in (1) on page 6.

Lemma 6 *If $P \xrightarrow{\alpha|\vec{M}}_{\mathcal{CA}} P'$, then there exists a context χ_α such that $\chi_\alpha \circ (1_{Pr}, \vec{M}) \circ P \rightarrow P'$. We list the corresponding χ_α s below:*

$$\begin{aligned} \chi_{in\ n} &\stackrel{\text{def}}{=} 3_{\mathbb{N}}[1_{Pr} \parallel 2_{Pr}] \parallel n[4_{Pr}] & \chi_{[in\ n], \chi_{open\ n}} &\stackrel{\text{def}}{=} 1_{Pr} \parallel n[2_{Pr}] & \chi_{[\overline{in\ n}]} &\stackrel{\text{def}}{=} 4_{\mathbb{N}}[in\ n.2_{Pr} \parallel 3_{Pr}] \parallel 1_{Pr} \\ \chi_{open\ n} &\stackrel{\text{def}}{=} open\ n.2_{Pr} \parallel 1_{Pr} & \chi_{out\ n} &\stackrel{\text{def}}{=} n[3_{\mathbb{N}}[1_{Pr} \parallel 2_{Pr}] \parallel 4_{Pr}] & \chi_{[out\ n]} &\stackrel{\text{def}}{=} n[1_{Pr} \parallel 2_{Pr}] & \chi_\tau &\stackrel{\text{def}}{=} 1_{Pr} \end{aligned}$$

3 Soundness and Completeness

Having presented our new labelled transition system we must demonstrate that it is fit for purpose. Specifically, the τ labelled transitions must characterise the reductions. Moreover, we also require that bisimulation equivalence is sound for reduction barb congruence. The fact that $\xrightarrow{\tau} \subseteq \rightarrow$ is implied already by the conclusion of Lemma 6. The converse follows by a straightforward inductive analysis of the structural forms of processes which may generate τ transitions.

Proposition 7 (Tau and Reduction) $P \xrightarrow{\tau} P'$ iff $P \rightarrow P'$. □

The chief property that needs to be established for the latter requirement (the soundness of $\sim_{\mathcal{CA}}$ with respect to \simeq) is congruence of bisimilarity with respect to language contexts. As a consequence of the fact that the LTS follows from the construction outlined in §2, this is straightforward to establish. The case of observational modifications which combine two separate derivations is the most interesting; here this concerns the rules (INTAU) and (OPTAU). Because the combination occurs via the \parallel operator, these rules are considered within a sub-case of the proof that bisimilarity is a congruence with respect to $1_{Pr} \parallel P$ contexts. The argument is roughly the following: the target of the derived τ -labelled transition, an application of the targets of two process-view transitions, can also be obtained by completing one of the transitions with the result of the other. The inductive hypothesis can then be used in order to match this complete transition, resulting in a bisimilar state, which can then be again deconstructed.

Proposition 8 (Congruence) *If $P \sim_{\mathcal{CA}} Q$ then $\mathcal{C}[P] \sim_{\mathcal{CA}} \mathcal{C}[Q]$ for all language contexts \mathcal{C} .* □

Theorem 9 (Soundness) $P \sim_{\mathcal{CA}} Q$ implies $P \simeq Q$. □

With soundness of bisimilarity established, it is a natural question as to whether the converse property of completeness holds. This is complicated by the issue of observability of actions. As encapsulated by the statement of Lemma 6,

$$\boxed{
\begin{array}{cc}
\frac{P \xrightarrow{\tau} P'}{P \xrightarrow{[\text{in } n] \downarrow R} P' \parallel n[R]} \text{ (A[IN])} & \frac{P \xrightarrow{\tau} P'}{P \xrightarrow{[\text{out } n] \downarrow R} n[P' \parallel R]} \text{ (A[OUT])}
\end{array}
}$$

Fig. 9. Honda-Tokoro rules \mathcal{HT} for unobservable actions

the labels of our LTS have corresponding underlying context-triggered reductions. Completeness relies on the converse relationship; a context-triggered reduction (or series of reductions and barb-observations) implying the existence of a transition.

Completeness needs to be checked manually; our systematic derivation technique as outlined in §2 does not guarantee that it holds. To prove it, one needs to show that each kind of label has a context which characterises it. This is a stronger requirement than that of Lemma 6 which exhibits a relationship between contexts and labels in one direction only: every labelled transition has a corresponding context in which there is a reduction to the right hand side. However, a reduction in this context does not necessarily imply the existence of the labelled transition. In order to do this, contexts have to contain more information.

It is unclear whether the LTS is complete with respect to reduction barb congruence in our finite language. Simply adding replication to the language does result in a language for which the LTS is not complete. Indeed, in the full ambient calculus, an ambient's ability to migrate is unobservable. This fact has been observed in [18] and a suitable adaptation of the definition of bisimulation is given to account for this. For aesthetic reasons we prefer to use ordinary bisimulation and thus use, a suitable modification of the Honda-Tokoro [15] style rules for strong equivalences instead. We have begun a more general investigation of such rules in [25]. Interestingly, here they are needed for $[\text{in } n]$ and $[\text{out } n]$ transitions only (*cf.* Fig. 9) and account for the following situation: the context provides the appropriate χ (*cf.* Lemma 6) but the process does not make use of it, thus χ is retained in the result. The rules are added to the combining rules of Fig. 8. Bisimilarity $\sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}}$ on the obtained LTS remains sound for contextual equivalence.

As an example of the necessity of the \mathcal{HT} rules for completeness, consider:

$$T_1 \stackrel{\text{def}}{=} !n[0] \parallel \nu k(k[\text{in } n.0]) \quad \text{and} \quad T_2 \stackrel{\text{def}}{=} !n[0] \parallel \tau$$

where $\tau \stackrel{\text{def}}{=} \nu m(\text{open } m.0 \parallel m[0])$. Processes T_1 and T_2 are reduction barb congruent. It is not difficult to check this directly using the fact that $\nu k k[0] \sim_{\mathcal{C}\mathcal{A}} 0$.

Nevertheless $T_1 \not\sim_{\mathcal{C}\mathcal{A}} T_2$ because the T_1 can do a $[\text{in } n] \downarrow R$ transition which cannot be matched by T_2 . Instead, it *does* hold that $T_1 \sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}} T_2$:

$$T_1 \xrightarrow{[\text{in } n] \downarrow R} !n[0] \parallel \nu k(n[k[0] \parallel R]) \text{ is matched by } T_2 \xrightarrow{[\text{in } n] \downarrow R} !n[0] \parallel n[R].$$

Concerning the remaining possible labels not considered by \mathcal{HT} -rules, we need to show that each complete labelled transition can be characterised by a predicate which is stable under reduction barbed congruence. For example, to characterise the transition labelled with $\text{open } n \downarrow R$ we use the context $\xi \stackrel{\text{def}}{=} \mathbf{1}_{\text{Pr}} \parallel n[i[0] \parallel \text{open } i.R]$ (with i fresh) and then show that $P \xrightarrow{\text{open } n \downarrow R} P'$ iff there exists P'' such that $\xi[P] \rightarrow P''$ with $P'' \downarrow_i$ and $P'' \rightarrow P'$ with $P' \not\downarrow_i$.

Theorem 10 (Completeness) $P \simeq Q$ implies $P \sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}} Q$. □

4 Conclusions, related and future work.

The introduction of the ambient calculus in [7] has spawned an enormous amount of research on the topic regarding variants of the calculus (*e.g.* [3, 10, 11]), type systems (*e.g.* [4, 6, 19]) and implementation details (*e.g.* [13, 22]). However, there has been relatively little work on labelled characterisations. An early attempt by Cardelli and Gordon [5] was abandoned in favour of a simpler approach in [8]. Interestingly, the structural rules and use of abstractions in the meta-language was already present in [5] where the authors seemed to encounter difficulty lay in relating their structural labels to contexts. This was particularly true for co-actions. The approach that we take in this paper resolves this issue.

Subsequent to [5, 8], Merro and Zappa-Nardelli [18] designed an LTS and established a full abstraction result using a form of context bisimilarity. Their paper is ostensibly the approach most closely related to ours in terms of results but the emphasis in our research is on a *systematic* derivation of the LTS model to achieve a similar result. We were fortunate in having had the model in [18] to use as a comparison and sanity check for our own semantics.

We hope that the benefits of our approach will become clear once one has compared the two LTS models: Merro and Zappa-Nardelli produced an LTS which built on the initial attempts by Cardelli and Gordon [5] (which already contained a reasonable account of the structural transitions towards an inductive definition of the τ -reduction relation) by analysing the contextual interactions provided by an arbitrary environment. Doing this necessitated a restriction to *system* level ambients – that is, ambients which were all boxed at top level – and a use of a piece of meta-syntax \circ to allow arbitrary environmental processes to be re-inserted into terms. The latter of these requirements resurfaces in our work through the use of the λ -calculus meta-language but the former, the restriction to systems, is avoided by providing context-oriented structural transitions in the LTS \mathcal{C} . The effect of this is that *all* of our (completed) labelled transitions are suitable for use in the definition of bisimulation as opposed to only the class of env-actions in [18]. Notice, for example, that our base rules (IN) and (OU) of *Fig. 6* retain the structure of the interacting context and term. This structure is carried in the rules (INAMB) and (OUAMB) whereas Merro and Zappa-Nardelli’s related rules, (ENTER SHH) and (EXIT SHH), in [18] serve primarily to recover this necessary structure. Our treatment of co-actions, in rules (COIN λ) and (COOPEN λ) of *Fig. 8*, by completing them with skeletal structure as well as missing parameters,

is mirrored in the rules (CO-ENTER) and (OPEN) of [18] although the restriction to systems complicates the latter of those. The remaining difference lies in the use of the name enclosing the migrating ambient in the (ENTER) and (EXIT) rules. They are included as part of the label in [18] and therefore reflect a slightly finer analysis of observability in ambients. However, rules (ENTER SHH) and (EXIT SHH) are then necessary because this name is not always observable. Our equivalent rules (INAMB) and (OUTAMB) do not record the name of the enclosing ambient in the label because this information is not determined by the context and the visibility of this name to be discovered by context parameter processes instead. Unlike [18] we deal with the unobservability of $[\text{in } n]$ and $[\text{out } n]$ actions using Honda Tokoro style [15] rules in *Fig. 9* rather than adopting a non-standard definition of bisimulation in the style of [1]. In conclusion, our derived LTS is pleasingly similar to, and, we believe, conceptually cleaner than its counterpart in [18] which represents the state of the art for this language to date.

In addition to the work mentioned above there have been a number of LTS models for variants of the ambient calculus [3, 10–12]. These models all use a variant of the language for which the contextual observations of co-actions are much clearer than in the pure ambient model and therefore the co-action labelled transitions are more easily defined. It will be interesting to see how our methodology fares when applied to these variants.

Finally, it is interesting to note that Sewell has already considered applying his contexts-as-labels approach [26] to the ambient calculus. We note that this work already suggests using (non-inductive versions of) our rules (IN) , (OUT) , and (OPEN) . Similarly, Jensen and Milner [16] use the context-as-labels approach to provide a derived LTS for the ambient calculus via an encoding to bigraphs. This LTS is also non-inductive and the lack of a detailed analysis of the resulting RPOs in [16] makes it difficult for us to find any striking similarities with our SOS rule-set and LTS.

In this paper and in [24] we have considered strong bisimilarity. It is interesting to observe that because Proposition 7 holds and because our bisimulation equivalence is defined over complete actions \mathcal{CA} , in principle it should be possible to smoothly lift our soundness and completeness results to weak bisimilarity. Notably, for weak transitions

$$P \xrightarrow{\tau_{\mathcal{CA}}} \dots \xrightarrow{\alpha_{\mathcal{CA}}} \dots \xrightarrow{\tau_{\mathcal{CA}}} P'$$

we will only ever need to decompose the *strong* α transition in to its Process and Context views. In particular, to characterise the weak equivalences, it is **not** the case that we will need to consider weak transitions from the \mathcal{C} and \mathcal{A} transitions systems separately. The difficulties which may arise in the weak case lie in providing contexts which witness weak transitions for the proof of completeness. We do not anticipate problems here but we have not yet checked the details of this.

The separation of process- and context-views in our approach means that our bisimulation equivalences are context bisimulations. This is due to the completion of labels by considering arbitrary context processes. As shown in [24], it is

sometimes possible to exploit this separation in order to refine the context-view so that only certain archetypal context processes need be supplied. It would be interesting to attempt to design an analogous refinement for ambients and we believe that our LTS serves as a good basis from which to do this.

Having experimented on the π -calculus [24] and the ambient calculus, we now intend to develop our method for deriving transition systems in a general setting and establish soundness and completeness results for a wide range of calculi.

Acknowledgment. We would like to thank the anonymous referees for their useful comments which have helped to improve the presentation of the paper.

References

1. R. Amadio, I. Castellani, and D. Sangiorgi. On bisimulations for the asynchronous pi-calculus. *Theor. Comput. Sci.*, 195(2):291–324, 1998.
2. J. Bergstra and J. Klop. Algebra of communicating processes with abstraction. *Theor. Comput. Sci.*, 37(1):77–121, 1985.
3. M. Bugliesi, S. Crafa, M. Merro, and V. Sassone. Communication interference in mobile boxed ambients. *Information and Computation*, 205:1235–1273, 2007.
4. L. Cardelli, G. Ghelli, and A. Gordon. Mobility types for mobile ambients. In *Proc. ICALP*, volume 1644 of *LNCS*, pages 230–239. Springer, 1999.
5. L. Cardelli and A. Gordon. A commitment relation for the ambient calculus., 1996. Unpublished notes.
6. L. Cardelli and A. Gordon. Types for mobile ambients. In *Proc. POPL*, pages 79–92. ACM Press, 1999.
7. L. Cardelli and A. Gordon. Mobile ambients. *Theoretical Computer Science*, 240/1:177–213, 2000.
8. L. Cardelli and A. Gordon. Equational properties of mobile ambients. *Math. Struct. Comput. Sci.*, 13(3):371–408, 2003.
9. U. Engberg and M. Nielsen. A calculus of communicating systems with label passing. Technical Report DAIMI PB-208, University of Aarhus, May 1986.
10. Y. Fu. Fair ambients. *Acta Inf.*, 43(8):535–594, 2007.
11. P. Garralda, E. Bonelli, A. Compagnoni, and M. Dezani-Ciancaglini. Boxed Ambients with Communication Interfaces. *MSCS*, 17:1–59, 2007.
12. M. Hennessy and M. Merro. Bisimulation congruences in safe ambients. *ACM Transactions on Programming Languages and Systems*, 28(2):290–330, 2006.
13. D. Hirschhoff, D. Pous, and D. Sangiorgi. An efficient abstract machine for safe ambients. *J. Logic and Algebraic Programming*, 71:114–149, 2007.
14. C. Hoare. *Communicating Sequential Processes*. Prentice Hall, Englewood Cliffs, 1985.
15. K. Honda and M. Tokoro. An object calculus for asynchronous communication. In *Proc. ECOOP*, volume 512 of *LNCS*, pages 133–147. Springer, 1991.
16. O. Jensen and R. Milner. Bigraphs and mobile processes. Technical Report 570, Computer Laboratory, University of Cambridge, 2003.
17. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. In *Proc. Concur*, volume 1877 of *LNCS*, pages 243–258. Springer, 2000.
18. M. Merro and F. Z. Nardelli. Behavioural theory for mobile ambients. *J. ACM*, 52(6):961–1023, 2005.

19. M. Merro and V. Sassone. Typing and subtyping mobility in boxed ambients. In *Proc. CONCUR*, volume 2421 of *LNCS*, pages 304–320. Springer, 2002.
20. R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.
21. R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, II. *Inf. Comput.*, 100(1):41–77, 1992.
22. A. Phillips. *Specifying and Implementing Secure Mobile Applications in the Channel Ambient System*. PhD thesis, Imperial College London, April 2006.
23. G. Plotkin. A structural approach to operational semantics. Technical Report FN-19, DAIMI, Computer Science Department, Aarhus University, 1981.
24. J. Rathke and P. Sobociński. Deconstructing behavioural theories of mobility. In *Proc. IFIP TCS 5th International Conference on Theoretical Computer Science*. Springer Science and Business Media, 2008. To appear.
25. J. Rathke and P. Sobociński. Making the unobservable unobservable. In *Proc. ICE '08*, ENTCS. Elsevier, 2008. To appear.
26. P. Sewell. From rewrite rules to bisimulation congruences. *Theor. Comput. Sci.*, 274(1-2):183–230, 2002. Extended version of Concur '98 conference paper.