Deriving structural labelled transitions for mobile ambients

Julian Rathke, Paweł Sobociński*,[1]

ECS, University of Southampton

Abstract

We present a new labelled transition system (LTS) for the ambient calculus. Its most important property is that ordinary (strong) bisimulation coincides with (strong) contextual equivalence. The LTS is the outcome of the authors' ongoing work towards developing general techniques and systematic procedures for deriving LTSs in the structural (SOS) style from the underlying reduction semantics and observability.

Key words: labelled transition systems (LTS), structural operational semantics (SOS), ambient calculus, bisimulation

Introduction

Since the introduction of archetypal process calculi (CCS [21], CSP [15], ACP [2] and the π-calculus [10, 22]) there has been a proliferation of new languages, extensions and assorted variants of earlier calculi. Each addresses some computational feature and/or enjoys specific properties. One concern that is often voiced regarding this field is that the semantics, usually a labelled transition system, is often ad hoc and heavily locally optimised. This state of affairs is unsatisfactory and initial attempts to address the issue were made in [33, 18] where it was proposed that labelled transitions should be *derived* (rather than defined) from underlying reduction rules for the language, the justification being that reduction rules are generally easier to define uncontentiously and can be taken to be definitional. Specifically, it

*Corresponding author

[1]Research partially supported by EPSRC grant EP/D066565/1

was proposed that labels ought to be 'suitably minimal' contexts that trigger reductions.

Sewell's seminal results [33] in this direction were limited in their scope. Leifer and Milner generalised the approach with some degree of success [18]. A general definition of 'contexts as labels' was provided using the universal property of relative pushouts (RPOs) 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 characterise and reason about. Indeed, compositionality in the sense of "the semantics of a compound phrase is a function of the semantics of its subphrases" is lost. It is thus easy to lose sight of the fact that an original intention of structural operational semantics [26] and labelled transition systems [21] was to provide an inductive definition of the reduction relation for a language. Their subsequent 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 long-term 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 characterise a (canonical, if possible) contextually defined equivalence. This task is difficult and we have begun by evaluating our ideas for well-known process calculi. The results of such an experiment for the π-calculus appear in [28]. The present paper concerns the ambient calculus of Cardelli and Gordon [8] and is an extended version of a conference paper [29]. Another recent work in this area is [30].

The ambient calculus has enjoyed some success as a foundational model of spatially distributed, concurrent processes that are hierarchically arranged, can migrate and dynamically modify the structure of their location. For our purposes, however, it is merely a small calculus with an interesting set of reduction rules. Moreover, endowing it with a labelled transition system and bisimulation equivalence was historically viewed as a challenging and worthwhile goal in its own right [6, 19]. It is, therefore, an ideal place to develop, apply and hone *generally applicable* syntactic techniques for the derivation of structural labelled transition systems. Indeed, the ambient calculus contrasts nicely with the π-calculus, the subject of our companion paper [28] on derivation of SOS rules: it is almost as well-known but its reduction semantics has a markedly different nature; whereas the π-calculus

has a single reduction rule schema that is structurally very simple but features a non-trivial use of meta-types (name substitution), the ambient calculus has three reduction rule schemas with quite intricate structure, but which reference only base types (see Section 2). Our purpose, as in [28], is not necessarily to improve or undermine an existing labelled transition system but to *derive one*, identifying principles and techniques that we hope will prove to be more generally applicable.

Roughly, the approach we take is to consider the underlying reduction rules of the language as open rewrite rules, which we dub *skeletons*. If a term partially matches the left-hand side of a (partially instantiated) rule, it will be the source of a labelled transition. The transition's label represents the remaining structure of the left-hand side of the reduction rule along with any missing parameters that must be supplied by an interacting context. This separation of a rule's structure and parameters allows us to build our labelled transition systems in three steps: we derive *process-view transitions* whose main purpose is to provide an inductively defined reduction relation, then the *context-view transitions* that allow for a context to supply parameters to an interaction, and finally global rules that combine them into a complete labelled transition. Technically, we make use of the simply typed λ-calculus as a powerful meta-language for syntax manipulation.

In addition to the inclusion of proofs and a more complete account of the derivation process, the current paper differs from the previous conference version [29] in that the *structural* nature of the LTS is emphasised. Previously we have used structural congruence to simplify the derivation process and, subsequently, the theorems about the resulting LTS. The price we paid for enhanced simplicity was obscured syntactic structure that made it harder to claim that our LTS was "truly" structural. Our justification was that the use of structural congruence was not ineradicable. In this paper we put this into practice. This fact should be contrasted with other recent work [3], in which the use of structural congruence is unavoidable.

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. In Section 3 we list technical lemmas about the derived LTS that allow us to connect labels with contexts—a necessary step in order for a satisfactory comparison of bisimilarity with a contextual equivalence. In Section 4

we prove that bisimilarity is sound for reduction barbed congruence. In Section 5 we add suitable Honda-Tokoro [16, 30] rules and show that bisimilarity on the resulting LTS is both sound and complete. We include a comparison with related work in Section 6 and close with concluding remarks regarding future work.

1. Ambients: syntax, metasyntax and reduction semantics

We give the grammar for sorts/types below (1). Expressions in the ambient calculus will be either names (of sort N) or processes (of sort Pr).

$$\sigma \quad ::= \quad \mathsf{N} \quad | \quad \mathsf{Pr} \tag{1}$$

The grammar for terms is specified below (2). As is usual, ordinary terms derived from the grammar will be considered as abstract syntax.

$$\begin{array}{rcl} M & ::= & m \mid \mathsf{X} \mid 0 \mid M \parallel M \mid M[\,M\,] \mid \nu m\, M \\ & & \quad \mid \mathsf{out}\, M.M \mid \mathsf{in}\, M.M \mid \mathsf{open}\, M.M \end{array} \tag{2}$$

We assume distinct countable supplies of names (ranged over by m, n; first syntactic category in (2)) and variables (ranged over by X, Y, x, y; second syntactic category in (2)). The syntactic construct 'ν' is a binder—it binds a *name* within its scope. To avoid unnecessary bookkeeping we assume that the syntax is quotiented with respect to α-equivalence, that is, we treat α-equivalent terms as equal. Indeed, in this paper and in our work on the π-calculus [28] we never examine the "syntactic identity" of bound names within a term. We shall need to be careful, however, when talking about contexts—these, in general, have the ability to bind.

Types are assigned to terms in the standard way. The type inference rules are listed in Fig. 1. A type context Γ is a finite map from variable names to types. Following the standard practice, we shall consider only typeable terms. By convention, we shall use x, y for variables of type N, X, Y for variables of type Pr, k, l for terms of type N and P, Q, R for closed terms of type Pr. M, N will be used for arbitrary terms of type Pr.

Given an LTS $\mathcal{L}$ the only labelled equivalence we shall consider is standard (strong) bisimilarity $\sim_{\mathcal{L}}$. It is the largest bisimulation on $\mathcal{L}$. Because we wanted to focus on the systematic derivation procedure of LTSs, we have not

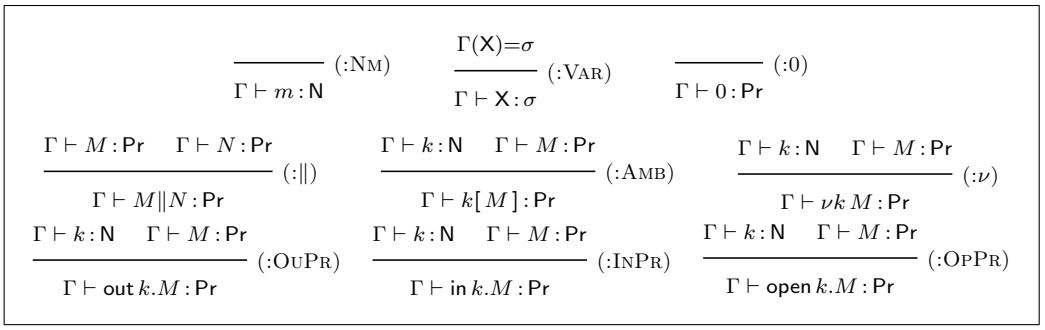


Figure 1: Type inference rules for typing terms generated by grammar (2).

considered weak equivalences in this paper; our feeling is that the study of weak equivalences examines largely orthogonal issues. We shall come back to this issue in the section on future work.

1.1. Replication and infinite processes

Before we proceed, it is worth noticing that our language does not contain replication or recursion operators and is thus finite. This is not a significant restriction because the crafting of a labelled transition system relies mainly on the characterisation of the immediately possible *interactions* of a process with a context—where an interaction means that the process and the context *together* (*i.e.* with non-trivial participation from each) trigger a reduction. In other words, labels themselves usually do not carry any information about the future behaviour (whether finite or not) of a process, only its current capability for interaction.

Technically this observation manifests itself through the usual semantics of replication: it is normally handled purely with structural congruence and *not* with its own reduction rule: $!P$ is, roughly, interpreted as an 'infinite parallel composition' of P's as evidenced by the structural congruence axiom $!P \equiv P \,\|!P$. The ramification is that the syntactic construct ! does not have its own inherent dynamics.

For sake of concreteness we could easily include a replication operator with negligible impact on the LTS rules; it would suffice to include the rule

$$\frac{P\|!P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'} \text{ (Rep)}$$

that, in any derivation, simply 'unfolds' enough P's in parallel in order to derive the desired labelled transition—this is possible because $\|$ does not inhibit behaviour, see the rules (L$\|*$) of Fig. 2.

More generally, (Rep) above can be seen as an instance of Plotkin's [26] well-known SOS rule

$$\frac{P[\mu X.P/X] \xrightarrow{\alpha} P'}{\mu X.P \xrightarrow{\alpha} P'} \text{ (Rec)}$$

for recursion and hence the above discussion is not limited just to replication. We chose not to consider these extensions in order to keep these details, irrelevant from the point of view of the derivation process, from increasing the complexity and specificity of the presentation. It is worth keeping in mind, however, that the presence of infinite processes *can* sometimes have an effect on the completeness of bisimilarity for contextual equivalence. The reasons for this are related to the reasons for why reduction congruence can sometimes coincide with barbed congruence in a finite language [23]. See Remark 17 for further elaboration on this point.

1.2. Contexts

A general notion of context is vital for a satisfactory exposition of the techniques harnessed in this paper. Contexts are defined using preliminary constructs that we shall refer to as 'precontexts'.

Definition 1 (Precontext)**.** Syntactically, precontexts are generated by the grammar obtained by adding a *σ-annotated hole* $-_\sigma$ for each type σ and a constructor for n-tuples (for any $n \in \mathbb{N}$) to the grammar (2):

$$M \quad ::= \quad \ldots \quad | \quad -_\sigma \quad | \quad (M, \ldots, M),$$

with the proviso that the ν-binder now has a different nature depending on whether its scope includes a hole—if this is the case then the resulting syntactic construct is not subject to α-equivalence, if not then the resulting construct is treated as within an ordinary term, up to α-equivalence. While this schizophrenic nature of the ν-binder may seem peculiar, there are no technical problems with its usage. In order to type precontexts, we add two additional type rules to the set of inference rules presented in Fig. 1:

$$\frac{}{\Gamma \vdash -_\sigma : \sigma} \text{ (:Hole)} \qquad \frac{\vdash V_1 : \sigma_1 \ \ldots \ \vdash V_n : \sigma_n \ (n \in \mathbb{N})}{\vdash (V_1, \ldots, V_n) : [\sigma_1 \ldots \sigma_n]} \text{ (:Tup).}$$

where $[\vec{\sigma}]$ is called an *interface type*. A precontext is then a typeable term of the form $(V_1, \ldots, V_n)$. Note that as a consequence of the fact that we require empty type contexts in rule (:TUP), any precontext of this form must have each V_i a closed term (no free variables). Indeed, it is worth emphasising that holes and variables are separate syntactic entities.

Definition 2 (Context)**.** Suppose that a precontext $(\vec{V}) : [\vec{\sigma}]$ contains m instances of type-annotated holes. A 1-1 enumeration of its holes with natural numbers from 1 to m uniquely determines a word $\vec{\tau}$ over types, where τ_i is the type of the ith-numbered hole. Syntactically replacing each hole with its number yields a *context* of type $[\vec{\tau}] \to [\vec{\sigma}]$. Ordinary terms of type Pr will be identified with contexts of type $[] \to [\mathsf{Pr}]$. Given contexts $f : [\vec{\sigma_1}] \to [\vec{\sigma_2}]$ and $g : [\vec{\sigma_2}] \to [\vec{\sigma_3}]$, there is a context $g \circ f : [\vec{\sigma_1}] \to [\vec{\sigma_3}]$ obtained by substitution of the ith component of f for the ith hole of g. This operation may involve capture if the hole is in the scope of a binder. Given $f : [\vec{\sigma_1}] \to [\vec{\sigma_2}]$ and $g : [\vec{\sigma_3}] \to [\vec{\sigma_4}]$ let $f \otimes g : [\vec{\sigma_1}\vec{\sigma_3}] \to [\vec{\sigma_2}\vec{\sigma_4}]$ be the context that 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 $\vec{\sigma} = \sigma_1 \ldots \sigma_k$, the *identity* context $\mathrm{id}_{[\vec{\sigma}]} : [\vec{\sigma}] \to [\vec{\sigma}]$ is $(\mathsf{1}_{\sigma_1}, \ldots, \mathsf{k}_{\sigma_k})$;
- if, given $\vec{\sigma}$ and $\vec{\tau}$ of equal length k, there exists a permutation $\rho : k \to k$ such that $\forall 1 \leq i \leq k.\ \sigma_{\rho i} = \tau_i$, there is an induced *permutation context* $\rho : [\vec{\sigma}] \to [\vec{\tau}]$ of the form $(\rho\mathsf{1}_{\tau_1}, \ldots, \rho\mathsf{k}_{\tau_k})$;
- a *language context* is a context of type $[\mathsf{Pr}] \to [\mathsf{Pr}]$. These will be denoted by $\mathcal{C}$;
- an *evaluation context* is simply a language context of type $[\mathsf{Pr}] \to [\mathsf{Pr}]$ in which the hole does not appear under a prefix. We shall denote these by $\mathcal{D}$;
- an *interaction context* is an evaluation context in which the hole does not appear within an ambient—it must appear at "top level". We shall denote these by $\mathcal{E}, \mathcal{F}$;
- given a language context we shall write $\mathcal{C} \# k$ if the hole of $\mathcal{C}$ is not within the scope of a νk.

To denote substitution of a term M in a language context $\mathcal{C}$ we shall often write $\mathcal{C}[\![M]\!]$ instead of $\mathcal{C} \circ M$. A relation R on terms is said to be a congruence if $M\ R\ N$ implies $\mathcal{C}[\![M]\!]\ R\ \mathcal{C}[\![N]\!]$ for all language contexts $\mathcal{C}$.

1.3. Structural congruence

Structural congruence is the smallest relation $\equiv$ on (possibly open) Pr-typed terms of the language that contains the axioms below and is a congruence. We write $\Gamma \vdash P \equiv Q$ as shorthand for $\Gamma \vdash P : \mathsf{Pr}$, $\Gamma \vdash Q : \mathsf{Pr}$ and $P \equiv Q$. Roughly, the axioms ensure that $\|$ can be thought of as an associative and commutative operator with identity 0 and the syntactic ν binder can migrate throughout the term without changing the set of bound names.

$$\Gamma \vdash (P \parallel Q) \parallel R \ \equiv\ P \parallel (Q \parallel R) \quad \Gamma \vdash P \parallel Q \ \equiv\ Q \parallel P \quad \Gamma \vdash P \parallel 0 \ \equiv\ P$$

$$\Gamma \vdash \nu m\, \nu n\, P \ \equiv\ \nu n\, \nu m\, P \qquad \Gamma \vdash \nu m\, 0 \ \equiv\ 0$$

$$\Gamma \vdash \nu m\, (P \parallel Q) \ \equiv\ P \parallel \nu m\, Q \quad (m \text{ not free in } P)$$

$$\Gamma \vdash \nu m\, (n[\, P\,]) \ \equiv\ n[\, \nu m\, P\,] \quad (m \neq n)$$

Structurally congruent terms should be considered as being 'intensionally' equal although clearly not 'syntactically' equal.

Remark 3. In previous expositions [28, 29] we have essentially quotiented the syntax by structural congruence when presenting our labelled transition systems, with the proviso that the use of structural congruence was *not essential* and its purpose was solely to simplify the presentation. This is a subtle issue because quotienting syntax by structural congruence 'blurs' structure to some degree and thus it may no longer be clear what a "structural operational semantics" means on syntax up to structural congruence. Indeed, in [3] a simpler LTS has been obtained with a price: the use of structural congruence in a derivation is an ineradicable component. Here we shall refrain from using structural congruence in the presentation of the LTS in order to emphasise its structural nature.

1.4. Meta-syntax

We shall use a meta-syntax for simple syntactic manipulation of terms. The meta-syntax is a simply typed λ-calculus and can be thought of as a primitive system of higher order abstract syntax [24]. In (3) below we extend

the base types (1) with function types that will be necessary in order to type terms in the metasyntax.

$$\sigma \quad ::= \quad \ldots \quad | \quad \sigma \to \sigma \tag{3}$$

The λ-calculus operators added to the signature (2) in (4) below constitute the syntactic aspect of the meta-language. Their function is solely to make the structural definition of a labelled transition system possible and they should not be considered as a language extension, having no computational meaning. They are to be thought of as meta-operators on syntactic phrases of the language.

$$M \quad ::= \quad \ldots \quad | \quad \lambda \mathsf{X} : \sigma . M \quad | \quad M(M) \tag{4}$$

In the above, λ-abstraction binds variables and we do not distinguish α-equivalent terms, analogously to our treatment of the ν-binder.

Terms in the metalanguage are typed with aid of the standard type rules for simply typed λ given in (5) below; these are added to the set of type rules presented in Fig. 1.

$$\frac{\Gamma, \mathsf{X}{:}\sigma \vdash M : \sigma'}{\Gamma \vdash \lambda \mathsf{X}{:}\sigma . M : \sigma \to \sigma'} \; (:\lambda) \qquad \frac{\Gamma \vdash M : \sigma \to \sigma' \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M(N) : \sigma'} \; (:\textsc{App}) \tag{5}$$

We quotient the terms of the metasyntax by the smallest congruence that contains (6) and (7). Substitution is the usual capture-avoiding notion—it is capture avoiding with respect to *all binders* and thus also the ν-binder of the underlying language.

$$(\lambda \mathsf{X} : \sigma . M)(N) \;=\; M[N/\mathsf{X}] \tag{6}$$

$$\lambda \mathsf{X} : \sigma . M(\mathsf{X}) \;=\; M \tag{7}$$

In the remainder of the paper, when we write a metasyntax phrase of base type, say Pr, we mean *the syntactic phrase that corresponds to the complete evaluation of the metasyntactic term*. This technique is very useful because it avoids many bureaucratic difficulties of binding scopes. Notice that this convention does not contradict our quotienting of general metasyntactic terms by (6) and (7)—its only consequence is that when we speak of a generalised term of base type we can assume that it is a bona fide syntactic term, not an equivalence class of meta-syntactic terms.

We can extend structural congruence to terms containing λ-abstractions by letting $\Gamma \vdash \lambda \mathsf{X}. P \equiv \lambda \mathsf{X}. Q$ whenever $\Gamma, \mathsf{X} \vdash P \equiv Q$. Structural congruence is compatible with (6) in the following sense:

Lemma 4. *If $P \equiv Q : \sigma \to \tau$ and for some $R \equiv S : \sigma$ then it follows that $P(R) \equiv Q(S) : \tau$.* □

1.5. Reductions

The inductive presentation of the reduction semantics is given below. It is easy to show that subject reduction holds. Note that the presence of (StrCong) makes this not a structural presentation (since structure can be changed with the aid of $\equiv$). In the conference version of this paper [29] we included this rule also in our main LTS with the proviso that it could be removed (see Remark 3). Herein we shall use the rule (StrCong) solely for the purpose of defining the reduction relation, for which purpose it is intrinsic.

$$\frac{}{m[\,\mathsf{in}\,n.P \| Q\,] \| n[\,R\,] \to n[\,m[\,P \| Q\,] \| R\,]}\ (\textsc{In}) \qquad \frac{}{n[\,m[\,\mathsf{out}\,n.P \| Q\,] \| R\,] \to m[\,P \| Q\,] \| n[\,R\,]}\ (\textsc{Out})$$

$$\frac{}{\mathsf{open}\,n.P \| n[\,Q\,] \to P \| Q}\ (\textsc{Open}) \qquad \frac{P \to P'}{P \| Q \to P' \| Q}\ (\textsc{Par}) \qquad \frac{P \to P'}{\nu n\, P \to \nu n\, P'}\ (\textsc{Nu}) \qquad \frac{P \to P'}{n[\,P\,] \to n[\,P'\,]}\ (\textsc{Amb})$$

$$\frac{P' \equiv P \quad P \to Q \quad Q \equiv Q'}{P' \to Q'}\ (\textsc{StrCng}).$$

The 'touchstone' equivalence for our purposes is reduction barbed congruence. It is outside of the scope of this paper to give a systematic explanation of how the correct barbs are to be chosen in general. Some progress towards this goal has been made in [27].

Below we recall a suitable definition of barb for the ambient calculus and the definition of the equivalence itself.

Definition 5 (Barbs)**.** We say that a term P *barbs* on an ambient m, written $P{\downarrow_m}$, if there is a "top level" instance of an ambient m in P. More formally, $P = \mathcal{E}[\![m[\,P'\,]]\!]$ for some P' and interaction context $\mathcal{E}$ such that $\mathcal{E} \# m$.

Definition 6 (Reduction barb congruence)**.** Reduction barb congruence ($\simeq$) is the largest symmetric relation $\mathcal{R}$ such that if $P\ \mathcal{R}\ Q$ then:

(i) If $P \to P'$ then there exists $Q \to Q'$ 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 contribution of this paper is a *systematic derivation procedure* of a novel structurally-defined LTS for the ambient calculus. First, we consider the reduction axioms (In), (Out) and (Open) as parameterised rules, referred to as *skeletons*. A skeleton is a pair of contexts (l_n^α, r_n^α) that describe the structural changes in passing from l_n^α to r_n^α. There are three skeletons: Sk_n^{in}, Sk_n^{out} and Sk_n^{open} with components:

$$(\; l_n^{in} \stackrel{\text{def}}{=} \mathsf{1_N}[\, \mathsf{in}\, n.\mathsf{2_{Pr}} \parallel \mathsf{3_{Pr}}\,] \parallel n[\,\mathsf{4_{Pr}}\,],\; r_n^{in} \stackrel{\text{def}}{=} n[\,\mathsf{1_N}[\,\mathsf{2_{Pr}} \parallel \mathsf{3_{Pr}}\,] \parallel \mathsf{4_{Pr}}\,] \;)$$

$$(\; l_n^{out} \stackrel{\text{def}}{=} n[\,\mathsf{1_N}[\,\mathsf{out}\, n.\mathsf{2_{Pr}} \parallel \mathsf{3_{Pr}}\,] \parallel \mathsf{4_{Pr}}\,],\; r_n^{out} \stackrel{\text{def}}{=} \mathsf{1_N}[\,\mathsf{2_{Pr}} \parallel \mathsf{3_{Pr}}\,] \parallel n[\,\mathsf{4_{Pr}}\,] \;)$$

$$(\; l_n^{open} \stackrel{\text{def}}{=} \mathsf{open}\, n.\mathsf{1_{Pr}} \parallel n[\,\mathsf{2_{Pr}}\,],\; r_n^{open} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{2_{Pr}} \;)$$

that are typed $l_n^{in}, r_n^{in} : [\mathsf{N}, \mathsf{Pr}^3] \to [\mathsf{Pr}]$, $l_n^{out}, r_n^{out} : [\mathsf{N}, \mathsf{Pr}^3] \to [\mathsf{Pr}]$ and $l_n^{open}, r_n^{open} : [\mathsf{Pr}^2] \to [\mathsf{Pr}]$ respectively. Using skeletons and contexts we can give an alternative "global" presentation of the reduction semantics of the calculus.

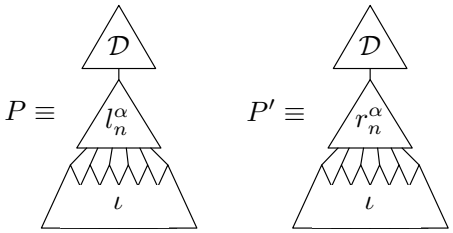
Proposition 7. *Let* $\to_g$ *be the following relation on pairs of closed terms of type* Pr*:*

$P \to_g P'$ *iff* $\exists \alpha \in \{in, out, open\},\, n,\, \mathcal{D}, \iota.\; P \equiv \mathcal{D}[\![l_n^\alpha \circ \iota]\!] \wedge P' \equiv \mathcal{D}[\![r_n^\alpha \circ \iota]\!]$

where $\mathcal{D}$ *is an evaluation context and* ι *are parameters of the appropriate type. Then* $\to_g = \to$ *.*

□

The condition on P and P' in Proposition 7 is illustrated in the diagram below.



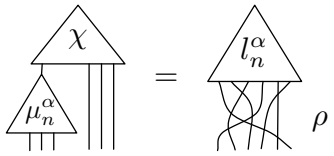
The derivation rules of our LTS are organised into three subsets: those defining the 'process view', in Fig. 2, the 'context view' in Fig. 3, and the 'combined' system in Fig. 4. The context view is the simplest of these and consists of a single applicative rule. In the remainder of this section we describe how to analyse the skeletons in order to obtain process-view rules and how this combines with the context view.

2.1. Derivation procedure: axioms

Considering the left-hand side $l : [\vec{\sigma}] \to [\mathsf{Pr}]$ of a skeleton Sk as a syntax tree, we say that a *match* is a subtree with root of type Pr. More formally, a match for l is any $\mu : [\vec{\sigma_1}] \to [\mathsf{Pr}]$ such for some $\vec{\sigma_2}$ there is a permutation $\rho : \vec{\sigma_1}\vec{\sigma_2} \to \vec{\sigma}$, and there exists a context $\chi : [\mathsf{Pr}, \vec{\sigma_2}] \to [\mathsf{Pr}]$ satisfying (8) below.

$$\chi \circ (\mu_n^\alpha \otimes \mathrm{id}_{[\vec{\sigma_2}]}) = l_n^\alpha \circ \rho \tag{8}$$

The intuition for the above equation is given by the diagram below.



A match is said to be *active* if there *does not* exist a context $\chi' : [\mathsf{Pr}, \vec{\sigma_2}] \to [\mathsf{Pr}]$ satisfying (9).

$$\chi' \circ (\mu_n^\alpha \otimes \mathrm{id}_{[\vec{\sigma_2}]}) = r_n^\alpha \circ \rho \tag{9}$$

Intuitively, an active match is a part of the left-hand side of the skeleton that is modified as a result of the reduction. Clearly any match that has an active match as a subtree is itself active. Of particular interest are those active matches that are locally *minimal* with respect to the subtree relation.[2]

Observation 8. *The minimal active matches are:*

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

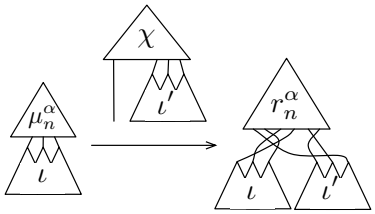
The axioms of our process-view LTS are determined by the minimal active matches. Indeed, their left-hand sides are the *instantiated* minimal active matches: given a minimal active match $\mu_n^\alpha : [\vec{\sigma}] \to [\mathsf{Pr}]$ they are the terms $\mu_n^\alpha \circ \iota$ where $\iota : [] \to [\vec{\sigma}]$. The result is the right-hand side of the skeleton

[2]Choosing active matches allows us to consider only those contexts in which the term under consideration interacts non-trivially. The definition given here also gives the right SOS axioms when applied in the setting of π-calculus.

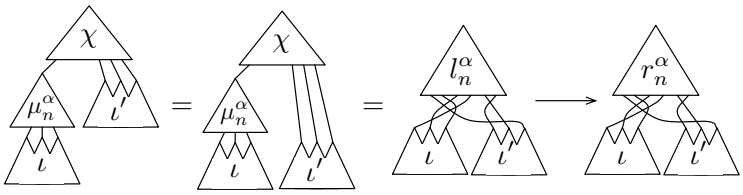
instantiated with the parameters ι of the minimal match together with that remaining parameters ι' required by χ:

$$\mu_n^\alpha \circ \iota \xrightarrow{\chi\circ(1_{\mathsf{Pr}}\otimes\iota')} r_n^\alpha \circ \rho \circ (\iota \otimes \iota'). \tag{10}$$

For the sake of intuition, it may be of use examining a graphical representation of the above, which we provide below.



The label is clearly a minimal context that triggers a reduction and as such is related to the early work of Sewell [33] and later work in this direction [18, 31, 32]. Indeed, the context provides $\chi \circ (1_{\mathsf{Pr}} \otimes \iota')$ and enables a reduction $\chi\circ(1_{\mathsf{Pr}}\otimes\iota')\circ\mu_n^\alpha\circ\iota = \chi\circ(\mu_n^\alpha\otimes\mathrm{id})\circ(\iota\otimes\iota') = l_n^\alpha\circ\rho\circ(\iota\otimes\iota') \to r_n^\alpha\circ\rho\circ(\iota\otimes\iota')$; this is illustrated below with aid of diagrams.



The main challenge to resolve in the sequel is to understand how to derive a transition such as (10) *compositionally* using SOS.

Note that each χ is uniquely determined by the particular minimal active match μ_α^n. For this reason in the label of the transition we shall use a textual representation $\alpha_i n \downarrow \vec{M}$ where $\alpha_i n$ represents the ith minimal active match of Sk_i^α, and $\vec{M}$ the list of the remaining parameters (see ι' in (10)). Following this procedure, we obtain the following labelled transitions:

$$\mathsf{in}\, n.P \xrightarrow{\mathsf{in}_1\, n\downarrow QkR} n[\,k[\,P\|Q\,]\|R\,] \qquad n[\,P\,] \xrightarrow{\mathsf{in}_2\, n\downarrow QRk} n[\,k[\,Q\|R\,]\|P\,] \tag{11}$$

$$\mathsf{out}\, n.P \xrightarrow{\mathsf{out}_1\, n\downarrow QkR} k[\,P\|Q\,]\|n[\,R\,] \tag{12}$$

$$\mathsf{open}\, n.P \xrightarrow{\mathsf{open}_1\, n\downarrow Q} P\|Q \qquad n[\,P\,] \xrightarrow{\mathsf{open}_2\, n\downarrow Q} Q\|P \tag{13}$$

An obstacle in giving a structural derivation of such an LTS is that in the results of the above transitions the distinction between ingredients for the interaction provided by the left-hand side term and ingredients provided by the context is lost. Our solution is to delay instantiation of the context components. Technically this is done with 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 $\mathcal{CA}$ that juxtapose these two views to form "complete" labelled transitions. Following this nomenclature, the process view's contribution to the transitions in (11) is

$$\frac{}{\mathsf{in}\, n.P \xrightarrow{\mathsf{in}_1\, n} \lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, n[\mathsf{x}[\, P \| \mathsf{X}\,] \| \mathsf{Y}\,]}\ (\textsc{In}1) \qquad \frac{}{n[\, P\,] \xrightarrow{\mathsf{in}_2\, n} \lambda \mathsf{X}\mathsf{Y}\mathsf{x}.\, n[\mathsf{x}[\mathsf{X} \| \mathsf{Y}\,] \| P\,]}\ (\textsc{In}2) \tag{14}$$

while the context parts are given by rule (Inst) of Fig. 3.

The rule that juxtaposes them is (Cλ) of Fig. 4. We take (In1), (In2) (see 14), (Ou1) (obtained from (12)), (Op1), (Op2) (obtained from (13)) as provisional axioms for the process-view LTS. By 'provisional' we mean that they are not the 'official' axioms (given in Fig. 3) of the LTS: they are given here as a starting point to aid the explanations below.

2.2. Derivation procedure: structure

Once the (provisional) axioms are determined, we can attempt to provide structural rules. There are three kinds, depending on the role that the added structure plays in the interaction that the label represents:

(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 right-hand side. 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 new component to the appropriate parameter;

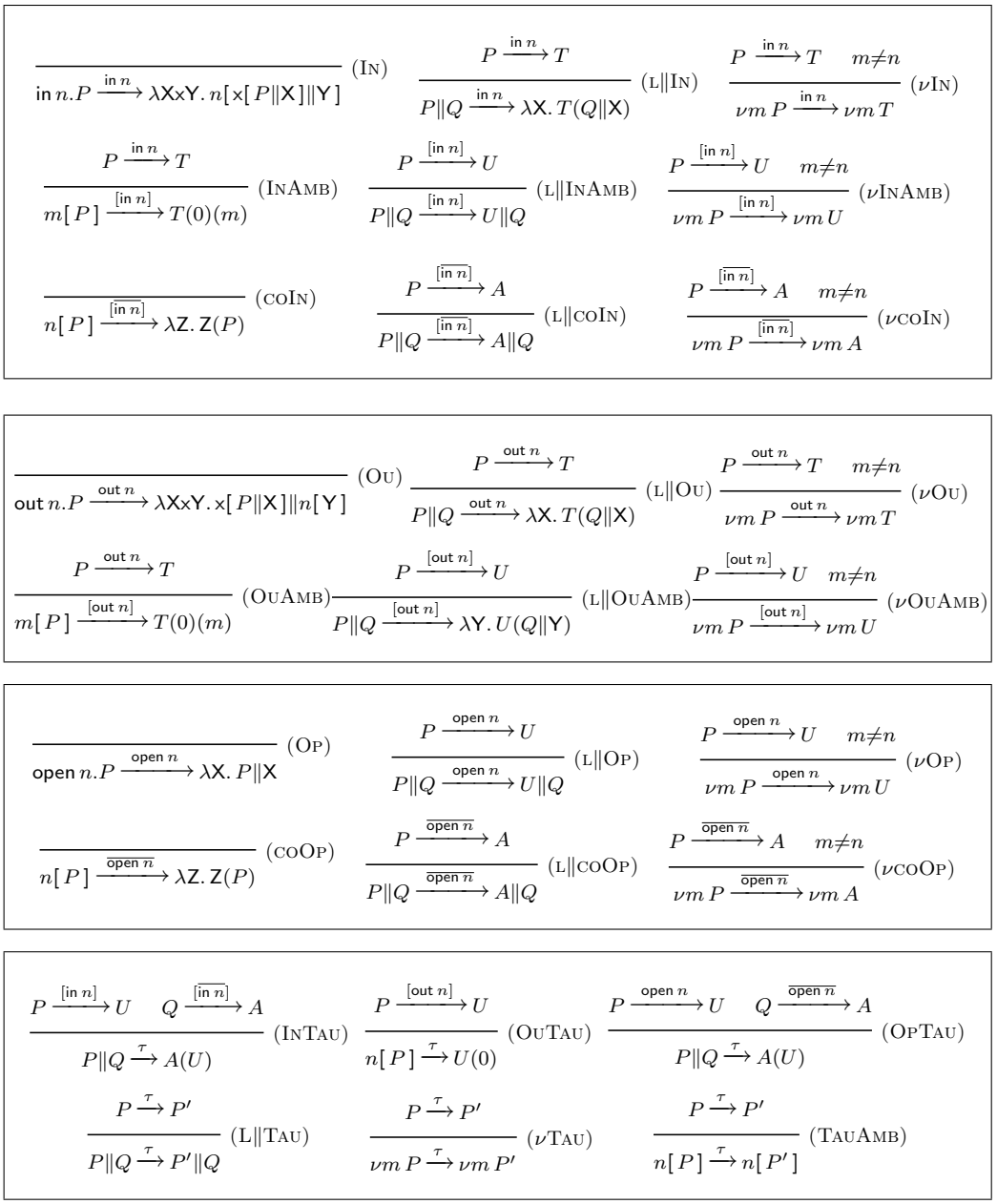


Figure 2: Process view ($\mathcal{C}$). By convention $T : \mathsf{Pr} \to \mathsf{N} \to \mathsf{Pr} \to \mathsf{Pr}$, $U : \mathsf{Pr} \to \mathsf{Pr}$, $A : (\mathsf{Pr} \to \mathsf{Pr}) \to \mathsf{Pr}$. Symmetric rules $(\mathrm{R}\|*)$ omitted. When $T = \lambda\vec{\mathsf{X}}.\, P$ we use $T \parallel Q \stackrel{\text{def}}{=} \lambda\vec{\mathsf{X}}.\, (T(\vec{\mathsf{X}}) \parallel Q)$ and $\nu m\, T \stackrel{\text{def}}{=} \lambda\vec{\mathsf{X}}.\, \nu m\, T(\vec{\mathsf{X}})$

$$\frac{\vec{M}{:}\vec{\sigma}}{\lambda\vec{\mathsf{X}}{:}\vec{\sigma}.\, P \xrightarrow{\downarrow\vec{M}} (\lambda\vec{\mathsf{X}}{:}\vec{\sigma}.\, P)(\vec{M})}\ (\textsc{Inst})$$

Figure 3: Context-view fragment ($\mathcal{A}$).

$$\frac{P \xrightarrow{\alpha}_{\mathcal{C}} A \qquad A \xrightarrow{\downarrow\overline{M}}_{\mathcal{A}} P'}{P \xrightarrow{\alpha\downarrow\overline{M}} P'}\ (\mathrm{C}\lambda) \qquad \frac{P \xrightarrow{\overline{[\mathsf{in}\ n]}}_{\mathcal{C}} A}{P \xrightarrow{\overline{[\mathsf{in}\ n]}\downarrow RSk} A(\lambda\mathsf{X}.\, n[\,k[\,R \| S\,] \| \mathsf{X}\,])}\ (\textsc{coIn}\lambda)$$

$$\frac{P \xrightarrow{\tau}_{\mathcal{C}} P'}{P \xrightarrow{\tau} P'}\ (\textsc{CTau}) \qquad \frac{P \xrightarrow{\overline{\mathsf{open}\ n}}_{\mathcal{C}} A}{P \xrightarrow{\overline{\mathsf{open}\ n}\downarrow R} A(\lambda\mathsf{X}.\, R \| \mathsf{X})}\ (\textsc{coOp}\lambda)$$

Figure 4: Combined system of complete actions ($\mathcal{CA}$).

(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 shall make use of the following abbreviations that deal with lambda abstractions $T = \lambda\vec{\mathsf{X}}.\, P$:

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

(iii) an *observational* modification: the extra structure forces an 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 left-hand side 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 (Ou) in Fig. 2 is just (Ou1) as described previously, with the numeral omitted. The rule (L‖Ou) is a substructural modification as described above. The rule (νOu) 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 right-hand side of the transition will not bind any names from the context when the context is instantiated via the context view rule. This is the correct behaviour and illustrates the import of capture-avoiding substitution and hence the suitability of using simply typed λ as a metalanguage. 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 (L‖OuAmb) is substructural while (νOuAmb) is superstructural. Finally, (OuTau) is an observational modification that completes the skeleton, meaning that a τ-labelled transition is derived.

Skeletons with two (or more) minimal active matches lead to a more involved situation. Consider the two minimal active matches of Sk_n^{in} and the two corresponding provisional axioms given in (14). Starting with either one, structure can be added, extending the match. Indeed, consider (In) of Fig. 2 obtained from (In1) of (14) by omitting the numeral. The rule (L‖In) is substructural and (νIn) superstructural. The rule (InAmb) is observational and extends the minimal match with a surrounding ambient. No further extension of the match is possible without including a contribution of the second minimal active match. The structural approach requires a combination of observations of the two matches in order to cover the entire left-hand side of the skeleton and derive a τ. However, in our two provisional axioms (In1), (In2) we have *included the right-hand side of the skeleton in result of the transitions*; a consequence that it is not obvious how to 'merge' the two by collecting 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. 2. The idea is that rather than using a concrete 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 actual skeleton provided by the action. Superstructural rules (L‖coOp) and (νcoOp) are straightforward and we are able to use (InTau) to collect the parameters to the right-hand side 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 that would result from using the right-hand side of the skeleton, we cannot simply use the combination

of (Inst) of Fig. 3 and (Cλ) of Fig. 4. Instead, we use rules (coInλ) and (coOpλ), which ensure that any context provided by the environment conforms to the appropriate 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 right-hand side of the skeleton in the result of the transition derived by (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 uniform presentation would consist in using the co-action style for *all* the labels. Following this approach, actual skeletons would never actually be instantiated in the right-hand side of the process-view transitions. The main price for this is that the rule (Inst) would need to be replaced with specific rules for each co-action, in the spirit of (CoInλ) and (CoOpλ) of Fig. 4. 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. 2 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 (see Section 6).

3. Properties of the LTS

Many of the proofs in the proceeding sections rely on a structural decomposition that, given a labelled transition, gives us some of the relevant structure of the left-hand side. This is the role of Lemmas 9 and 10 below. The first (Lemma 9) deals with the process-view LTS $\mathcal{C}$ and the second (Lemma 10) pertains to the complete LTS $\mathcal{CA}$.

Lemma 9 (Structural, $\mathcal{C}$)**.** *In each of the following choices for (α, Q, B), if $P \xrightarrow{\alpha}_{\mathcal{C}} A$ $(\alpha \neq \tau)$ then there exists a name n and an interaction context $\mathcal{E}$ with $\mathcal{E} \# n$ so that $P = Q$ and $A \equiv B$. Conversely, if $P = Q$ for some interaction context $\mathcal{E} \# n$ then there exists $A \equiv B$ such that $P \xrightarrow{\alpha}_{\mathcal{C}} A$.*

(i) $\alpha = \mathsf{in}\, n$, $Q = \mathcal{E}[\![\mathsf{in}\, n.P_1]\!]$ *and* $B = \lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, n[\,\mathsf{x}[\,\mathcal{E}[\![P_1]\!] \parallel \mathsf{X}\,] \parallel \mathsf{Y}\,]$;

(ii) $\alpha = [\mathsf{in}\, n]$, *for some* $m, \mathcal{E}'$ $\#$ n $Q = \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{in}\, n.P_1]\!]\,]]\!]$ *and* $B = \lambda \mathsf{Y}.\, \mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![P_1]\!]\,] \parallel \mathsf{Y}\,]]\!]$;

(iii) $\alpha = [\overline{\mathsf{in}\, n}]$, $Q = \mathcal{E}[\![n[\,P_1\,]]\!]$ *and* $B = \lambda \mathsf{Z}.\, \mathcal{E}[\![\mathsf{Z}P_1]\!]$;

(iv) $\alpha = \mathsf{open}\, n$, $Q = \mathcal{E}[\![\mathsf{open}\, n.P_1]\!]$ *and* $B = \lambda \mathsf{Y}.\, \mathcal{E}[\![P_1 \parallel \mathsf{Y}]\!]$;

(v) $\alpha = \overline{\mathsf{open}\, n}$, $Q = \mathcal{E}[\![n[\, P_1 \,]]\!]$ *and* $B = \lambda \mathsf{Z}.\, \mathcal{E}[\![\mathsf{Z} P_1]\!]$;

(vi) $\alpha = \mathsf{out}\, n$, $Q = \mathcal{E}[\![\mathsf{out}\, n.P_1]\!]$ *and* $B = \lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, \mathsf{x}[\, \mathcal{E}[\![P_1]\!] \parallel \mathsf{X} \,] \parallel n[\, \mathsf{Y} \,]$;

(vii) $\alpha = [\mathsf{out}\, n]$, *there exist* m *and* $\mathcal{E}' \# n$ *such that* $Q = \mathcal{E}[\![m[\, \mathcal{E}'[\![\mathsf{out}\, n.P_1]\!] \,]]\!]$. *Let names* $\vec{l}$ *and term* P_2 *be such that* $\mathcal{E}[\![\mathsf{X}]\!] \equiv \nu \vec{l} (\mathsf{X} \parallel P_2)$, *then* $B = \lambda \mathsf{Y}.\, \nu \vec{l}\ (m[\, \mathcal{E}'[\![P_1]\!] \,] \parallel n[\, P_2 \parallel \mathsf{Y} \,])$.

Proof. In each case, first one reasons by induction over the derivation of the labelled transition. For the converse, one argues by induction on the structure of $\mathcal{E}$. □

The following lemma is easily proved using the conclusions of Lemma 9 and the construction of the LTS. It is useful when reasoning about the complete LTS $\mathcal{CA}$ (see Fig. 4) and will be referred to often.

Lemma 10 (Structural, $\mathcal{CA}$)**.** *In each of the following choices for* (α, Q, Q'), *if* $P \xrightarrow{\alpha}_{\mathcal{CA}} P'$ $(\alpha \neq \tau)$ *then there exists a name* n *and an interaction context* $\mathcal{E}$ *with* $\mathcal{E} \# n$ *so that* $P = Q$ *and* $Q \equiv Q'$. *Conversely, if* $P = Q$ *for some interaction context* $\mathcal{E} \# n$ *then there exists* $P' \equiv Q'$ *such that* $P \xrightarrow{\alpha}_{\mathcal{CA}} P'$.

(i) $\alpha = \mathsf{in}\, n \downarrow RkS$, $Q = \mathcal{E}[\![\mathsf{in}\, n.P_1]\!]$ *and* $Q' = (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, n[\, \mathsf{x}[\, \mathcal{E}[\![P_1]\!] \parallel \mathsf{X} \,] \parallel \mathsf{Y} \,])(RkS)$;

(ii) $\alpha = [\mathsf{in}\, n] \downarrow R$, $Q = \mathcal{E}[\![m[\, \mathcal{E}'[\![\mathsf{in}\, n.P_1]\!] \,]]\!]$ *and* $Q' = (\lambda \mathsf{Y}.\, \mathcal{E}[\![n[\, m[\, \mathcal{E}'[\![P_1]\!] \,] \parallel \mathsf{Y} \,]]\!])(R)$;

(iii) $\alpha = [\overline{\mathsf{in}\, n}] \downarrow RSk$, $Q = \mathcal{E}[\![n[\, P_1 \,]]\!]$ *and* $Q' = (\lambda \mathsf{Y}\mathsf{Z}\mathsf{x}.\, \mathcal{E}[\![n[\, x[\, \mathsf{Y} \parallel \mathsf{Z} \,] \parallel P_1 \,]]\!])(RSk)$;

(iv) $\alpha = \mathsf{open}\, n \downarrow R$, $Q = \mathcal{E}[\![\mathsf{open}\, n.P_1]\!]$ *and* $Q' = (\lambda \mathsf{Y}.\, \mathcal{E}[\![P_1 \parallel \mathsf{Y}]\!])(R)$;

(v) $\alpha = \overline{\mathsf{open}\, n} \downarrow R$, $Q = \mathcal{E}[\![n[\, P_1 \,]]\!]$ *and* $Q' = (\lambda \mathsf{Y}.\, \mathcal{E}[\![\mathsf{Y} \parallel P_1]\!])(R)$;

(vi) $\alpha = \mathsf{out}\, n \downarrow RkS$, $Q = \mathcal{E}[\![\mathsf{out}\, n.P_1]\!]$ *and* $Q' = (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, \mathsf{x}[\, \mathcal{E}[\![P_1]\!] \parallel \mathsf{X} \,] \parallel n[\, \mathsf{Y} \,])(RkS)$;

(vii) $\alpha = [\mathsf{out}\, n] \downarrow R$, *there exist* m *and* $\mathcal{E}' \# n$ *s.t.* $Q = \mathcal{E}[\![m[\, \mathcal{E}'[\![\mathsf{out}\, n.P_1]\!] \,]]\!]$. *Let names* $\vec{l}$ *and term* P_2 *be such that* $\mathcal{E}[\![\mathsf{X}]\!] \equiv \nu \vec{l} (\mathsf{X} \parallel P_2)$, *then* $Q' = (\lambda \mathsf{Y}.\, \nu \vec{l}\ (m[\, \mathcal{E}'[\![P_1]\!] \,] \parallel n[\, P_2 \parallel Y \,]))(R)$.

Labelled transitions are compatible with structural congruence in the following sense:

Lemma 11. *Suppose that* $P \xrightarrow{\alpha}_{\mathcal{CA}} P'$ *and that* $P \equiv Q$. *Then there exists* Q' *such that* $P' \equiv Q'$ *and* $Q \xrightarrow{\alpha}_{\mathcal{CA}} Q'$.

Proof. If $\alpha \neq \tau$ then we argue by cases using the conclusions of Lemma 10. The argument is roughly similar in each case. Roughly, we use the fact that while for each kind of label α the structural congruence $P \equiv P'$ 'blurs' structure, it preserves the 'triggers' of labels (the Q's of Lemma 10).

Case $\alpha = \mathsf{in}\, n \downarrow RkS$**:** $P = \mathcal{E}[\![\mathsf{in}\, n.P_1]\!]$ and since $P \equiv Q$ we must have $Q = \mathcal{F}[\![\mathsf{in}\, n.Q_1]\!]$ s.t. $\mathcal{E}[\![P_1]\!] \equiv \mathcal{F}[\![Q_1]\!]$ (1). Now $P' \equiv (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, n[\,\mathsf{x}[\,\mathcal{E}[\![P_1]\!] \parallel \mathsf{X}\,] \parallel \mathsf{Y}\,])(RkS)$ and $Q \xrightarrow{\alpha} Q' \equiv (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, n[\,\mathsf{x}[\,\mathcal{F}[\![Q_1]\!] \parallel \mathsf{X}\,] \parallel \mathsf{Y}\,])(RkS)$ whence from (1) it follows that $P' \equiv Q'$.

Case $\alpha = [\mathsf{in}\, n] \downarrow R$**:** $P = \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{in}\, n.P_1]\!]\,]]\!]$, thus $Q = \mathcal{F}[\![m'[\,\mathcal{F}'[\![\mathsf{in}\, n.Q_1]\!]\,]]\!]$ ($\mathcal{F}$, $\mathcal{F}' \# n$) such that $\mathcal{E}[\![m[\,\mathcal{E}'[\![P_1]\!]\,]]\!] \equiv \mathcal{F}[\![m'[\,\mathcal{F}'[\![Q_1]\!]\,]]\!]$ (2). Now $P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![P_1]\!]\,] \parallel \mathsf{Y}\,]]\!])(R)$ and $Q \xrightarrow{\alpha} Q' \equiv (\lambda \mathsf{Y}.\, \mathcal{F}[\![n[\,m'[\,\mathcal{F}'[\![Q_1]\!]\,] \parallel \mathsf{Y}\,]]\!])(R)$ so $Q \equiv Q'$ by (2) and the fact that $\mathcal{E}$, $\mathcal{F} \# n$.

Case $\alpha = [\overline{\mathsf{in}\, n}] \downarrow RkS$**:** $P = \mathcal{E}[\![n[\,P_1\,]]\!]$ and so $Q = \mathcal{F}[\![n[\,Q_1\,]]\!]$ with $\mathcal{E}[\![P_1]\!] \equiv \mathcal{F}[\![Q_1]\!]$ (3). Now $P' \equiv (\lambda \mathsf{Y}\mathsf{Z}\mathsf{x}.\, \mathcal{E}[\![n[\,\mathsf{x}[\,\mathsf{Y} \parallel \mathsf{Z}\,] \parallel P_1\,]]\!])(RSk)$ and $Q \xrightarrow{\alpha} Q' \equiv (\lambda \mathsf{Y}\mathsf{Z}\mathsf{x}.\, \mathcal{F}[\![n[\,\mathsf{x}[\,\mathsf{Y} \parallel \mathsf{Z}\,] \parallel P_1\,]]\!])(RSk)$ whence by (3) we have $P' \equiv Q'$;

Case $\alpha = \mathsf{open}\, n \downarrow R$**:** $P = \mathcal{E}[\![\mathsf{open}\, n.P_1]\!]$ and so $Q = \mathcal{F}[\![\mathsf{open}\, n.Q_1]\!]$ such that $\mathcal{E}[\![P_1]\!] \equiv \mathcal{F}[\![Q_1]\!]$ (4). Now $P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![P_1 \parallel \mathsf{Y}]\!])(R)$ and $Q \xrightarrow{\alpha} Q' \equiv (\lambda Y.\, \mathcal{F}[\![Q_1 \parallel \mathsf{Y}]\!])(R)$, so by (4) we have $P' \equiv Q'$.

Case $\alpha = \overline{\mathsf{open}\, n} \downarrow R$**:** $P = \mathcal{E}[\![n[\,P_1\,]]\!]$ and so $Q = \mathcal{F}[\![n[\,Q_1\,]]\!]$ such that $\mathcal{E}[\![P_1]\!] \equiv \mathcal{F}[\![Q_1]\!]$ (5). Now $P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![\mathsf{Y} \parallel P_1]\!])(R)$ and $Q' \equiv (\lambda \mathsf{Y}.\, \mathcal{F}[\![\mathsf{Y} \parallel P_1]\!])(R)$ whence by (5) it follows that $P' \equiv Q'$.

Case $\alpha = \mathsf{out}\, n \downarrow RkS$**:** $P = \mathcal{E}[\![\mathsf{out}\, n.P_1]\!]$ and so $Q = \mathcal{F}[\![\mathsf{out}\, n.Q_1]\!]$ such that $\mathcal{E}[\![P_1]\!] \equiv \mathcal{F}[\![Q_1]\!]$ (6). Now $P' \equiv (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, \mathsf{x}[\,\mathcal{E}[\![P_1]\!] \parallel \mathsf{X}\,] \parallel n[\,\mathsf{Y}\,])(RkS)$ and $Q \xrightarrow{\alpha} Q' \equiv (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, \mathsf{x}[\,\mathcal{F}[\![Q_1]\!] \parallel \mathsf{X}\,] \parallel n[\,\mathsf{Y}\,])(RkS)$; by (6) it follows that $P' \equiv Q'$.

Case $\alpha = [\mathsf{out}\, n] \downarrow R$**:** $P = \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{out}\, n.P_1]\!]\,]]\!]$, $Q = \mathcal{F}[\![m'[\,\mathcal{F}'[\![\mathsf{out}\, n.Q_1]\!]\,]]\!]$ such that $\mathcal{E}[\![m[\,\mathcal{E}'[\![P_1]\!]\,]]\!] \equiv \mathcal{F}[\![m'[\,\mathcal{F}'[\![Q_1]\!]\,]]\!]$ (7). Now, letting $\vec{l}$ and P_2 be

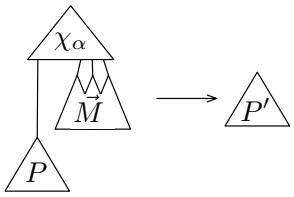
such that $\mathcal{E}[\![\mathsf{X}]\!] = \nu\vec{l}(\mathsf{X} \parallel P_2)$, we have that $P' \equiv (\lambda\mathsf{Y}.\,\nu\vec{l}(m[\,\mathcal{E}'[\![P_1]\!]\,] \parallel n[\,P_2 \parallel \mathsf{Y}\,]))(R)$ and for $\vec{k}, Q_2$ such that $\mathcal{F}[\![\mathsf{X}]\!] \equiv \nu\vec{k}\,(\mathsf{X} \parallel Q_2)$ we have $Q \xrightarrow{\alpha} Q' \equiv (\lambda\mathsf{Y}.\,\nu\vec{k}\,(m'[\,\mathcal{F}'[\![Q_1]\!]\,] \parallel n[\,Q_2 \parallel \mathsf{Y}\,]))(R)$. Equation (7) can be used to show that $\nu\vec{l}(m[\,\mathcal{E}'[\![P_1]\!]\,] \parallel P_2) \equiv \nu\vec{k}\,(m'[\,\mathcal{F}'[\![Q_1]\!]\,] \parallel Q_2)$ which implies that $P' \equiv Q'$.

Case $\alpha = \tau$**:** By induction on the derivation of the labelled transition we obtain the possible structural decompositions of P that are preserved by $\equiv$, similarly to the other cases. □

Notice that the conclusion of Lemma 11 implies that $\equiv \subseteq \sim_{\mathcal{CA}}$.

The following lemma provides a sanity check for our LTS that ensures that transitions obtained from our structural rules are justified by a reduction in a context—the point with which we started our discussion in (10) on page 13.

Lemma 12. *If* $P \xrightarrow{\alpha\downarrow\vec{M}}_{\mathcal{CA}} P'$*, then there exists a context* χ_α *such that* $\chi_\alpha \circ (\mathsf{1_{Pr}}, \vec{M}) \circ P \rightarrow P'$*; this is illustrated graphically below.*



The corresponding χ_α *contexts are listed below:*

$$\chi_{\mathsf{in}\,n} \stackrel{\text{def}}{=} \mathsf{3_N}[\,\mathsf{1_{Pr}} \parallel \mathsf{2_{Pr}}\,] \parallel n[\,\mathsf{4_{Pr}}\,], \qquad \chi_{[\mathsf{in}\,n]}, \chi_{\mathsf{open}\,n} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel n[\,\mathsf{2_{Pr}}\,],$$

$$\chi_{\overline{[\mathsf{in}\,n]}} \stackrel{\text{def}}{=} \mathsf{4_N}[\,\mathsf{in}\,n.\mathsf{2_{Pr}} \parallel \mathsf{3_{Pr}}\,] \parallel \mathsf{1_{Pr}}, \qquad \chi_{\overline{\mathsf{open}\,n}} \stackrel{\text{def}}{=} \mathsf{open}\,n.\mathsf{2_{Pr}} \parallel \mathsf{1_{Pr}},$$

$$\chi_{\mathsf{out}\,n} \stackrel{\text{def}}{=} n[\,\mathsf{3_N}[\,\mathsf{1_{Pr}} \parallel \mathsf{2_{Pr}}\,] \parallel \mathsf{4_{Pr}}\,], \qquad \chi_{[\mathsf{out}\,n]} \stackrel{\text{def}}{=} n[\,\mathsf{1_{Pr}} \parallel \mathsf{2_{Pr}}\,], \qquad \chi_\tau \stackrel{\text{def}}{=} \mathsf{1_{Pr}}$$

Proof. We include the case $\mathsf{in}\,n \downarrow RkS$, the other cases where $(\alpha \neq \tau)$ are similar. If $P \xrightarrow{\mathsf{in}\,n\downarrow RkS} P'$ then $\exists \mathcal{E} \,\#\, n$, P_1 such that $P = \mathcal{E}[\![\mathsf{in}\,n.P_1]\!]$ and $P' \equiv \lambda\mathsf{X}\mathsf{x}\mathsf{Y}.\,n[\,\mathsf{x}[\,\mathcal{E}[\![P_1]\!] \parallel \mathsf{X}\,] \parallel \mathsf{Y}\,](RkS)$. Also $\chi_{\mathsf{in}\,n} \circ (\mathsf{1_{Pr}}, R, k, S) \circ P = (k[\,\mathsf{1_{Pr}} \parallel R\,] \parallel n[\,S\,]) \circ P = k[\,\mathcal{E}[\![\mathsf{in}\,n.P_1]\!] \parallel R\,] \parallel n[\,S\,] \rightarrow n[\,k[\,\mathcal{E}[\![P_1]\!] \parallel R\,] \parallel S\,] = P'$. For $\alpha = \tau$ one argues by induction over the derivation of the transition in order to obtain a redex within P.

□

4. Soundness

We shall first note that τ-labelled transitions characterise reductions. Secondly, we shall prove that bisimilarity is sound for reduction barb congruence, *i.e.* $\sim_{\mathcal{CA}} \subseteq \simeq$.

We have already verified that $\xrightarrow{\tau} \subseteq \rightarrow$: this is implied by the conclusion of Lemma 12. The converse follows by a straightforward inductive analysis of the structural forms of processes that are the sources of a τ transition.

Proposition 13 (Tau and Reduction)**.** *If $P \xrightarrow{\tau} P'$ then $P \rightarrow P'$. If $P \rightarrow P'$ then $\exists P''.\, P'' \equiv P'$ such that $P \xrightarrow{\tau} P''$.* □

The chief property that needs to be established for soundness of $\sim_{\mathcal{CA}}$ with respect to $\simeq$ is congruence of bisimilarity with respect to language contexts. As a consequence of the construction outlined in §2, this is fairly straightforward to establish. The case of observational modifications that combine two separate derivations is the most involved; here this concerns the rules (InTau) and (OpTau). Because the combination occurs via the $\|$ operator, these rules are considered within a subcase of the proof that bisimilarity is a congruence with respect to $\mathsf{1}_{\mathsf{Pr}} \parallel P$ 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 used in order to match this complete transition, resulting in a bisimilar state, which can then be again deconstructed.

We shall use the following result:

Lemma 14 (Interaction contexts commute)**.** *For all interaction contexts $\mathcal{E}, \mathcal{E}'$ and terms P, Q we have:*

1. $\mathcal{E}[\![\mathcal{E}'[\![P]\!]]\!] \equiv \mathcal{E}'[\![\mathcal{E}[\![P]\!]]\!]$
2. $\mathcal{E}'[\![P]\!] \parallel \mathcal{E}[\![Q]\!] \equiv \mathcal{E}'[\![\mathcal{E}[\![P \parallel Q]\!]]\!]$

up to α-conversion on $\mathcal{E}$ and $\mathcal{E}'$.

Proof. We first show (by structural induction on $\mathcal{E}$) the subsidiary statements that (A) $\mathcal{E}[\![P]\!] \parallel R \equiv \mathcal{E}[\![P \parallel R]\!]$ and (B) $\nu n\, \mathcal{E}[\![P]\!] \equiv \mathcal{E}[\![\nu n\, P]\!]$ (for $n \notin \mathcal{E}$). Then (i) follows by an induction over $\mathcal{E}$ and (ii) follows by two applications of (A). □

Proposition 15 (Congruence)**.** *If* $P \sim_{\mathcal{CA}} Q$ *then* $\mathcal{C}[\![P]\!] \sim_{\mathcal{CA}} \mathcal{C}[\![Q]\!]$ *for all language contexts* $\mathcal{C}$*.*

Proof. Let $\hat{\sim}$ be the congruence relation defined:

$$\mathcal{C}[\![P]\!] \hat{\sim} \mathcal{C}[\![Q]\!] \text{ whenever } P \sim_{\mathcal{CA}} Q$$

for any language context $\mathcal{C}$. To prove the theorem it suffices to show that $\hat{\sim}$ is a bisimulation up to $\equiv$. By induction over $\mathcal{C}$ we will show that:

$\mathcal{C}[\![P]\!] \hat{\sim} \mathcal{C}[\![Q]\!]$ and $\mathcal{C}[\![P]\!] \xrightarrow{\gamma}_{\mathcal{CA}} P'$ implies $\mathcal{C}[\![Q]\!] \xrightarrow{\gamma}_{\mathcal{CA}} Q'$ such that $P' \equiv \hat{\sim} \equiv Q'$

for some Q'.

Case 1_{Pr}: the base case $\mathcal{C} = 1_{\mathsf{Pr}}$ holds trivially.

Case $\mathsf{in}\, n.\mathcal{C}'$, $\mathsf{out}\, n.\mathcal{C}'$, and $\mathsf{open}\, n.\mathcal{C}'$: for contexts of the form $\mathsf{in}\, n.\mathcal{C}$, $\mathsf{out}\, n.\mathcal{C}$ and $\mathsf{open}\, n.\mathcal{C}$ it is easy to see that there can be no contribution to the transition $\mathcal{C}[\![P]\!] \xrightarrow{\gamma}_{\mathcal{CA}} P'$ from P and hence $\mathcal{C}[\![Q]\!]$ easily matches this.

Case $\nu n\, \mathcal{C}'$: suppose that $\mathcal{C} = \nu n\, \mathcal{C}'$ and suppose also that $\mathcal{C}[\![P]\!] \xrightarrow{\gamma}_{\mathcal{CA}} P'$. We know that γ is either τ or $\alpha \downarrow \vec{R}$. For τ the result follows directly from the inductive hypothesis and so we consider the latter case. We know that $\mathcal{C}[\![P]\!] \xrightarrow{\alpha\downarrow\vec{R}}_{\mathcal{CA}} P'$ implies that $P' \equiv (\nu n\, A)(\vec{R})$ for some A. By analysis of the LTS rules we also see that $\mathcal{C}'[\![P]\!] \xrightarrow{\alpha\downarrow\vec{R}}_{\mathcal{CA}} P''$ where $P'' \equiv A(\vec{R})$. In particular $P' \equiv \nu n\, P''$. The inductive hypothesis tells us that $\mathcal{C}'[\![Q]\!] \xrightarrow{\alpha\downarrow\vec{R}}_{\mathcal{CA}} Q''$ for some Q'' such that $P'' \equiv \hat{\sim} \equiv Q''$. It follows that $\nu n\, \mathcal{C}'[\![Q]\!] \xrightarrow{\gamma}_{\mathcal{CA}} \nu n\, Q''$ and further, because $\hat{\sim}$ and $\equiv$ are congruence relations, that $P' \equiv \nu n\, P'' \equiv \hat{\sim} \equiv \nu n\, Q''$ as required.

Case $n[\,\mathcal{C}'\,]$: suppose now that $\mathcal{C} = n[\,\mathcal{C}'\,]$ and that $\mathcal{C}[\![P]\!] \xrightarrow{\gamma}_{\mathcal{CA}} P'$. There are a number of subcases to consider:

> **Subcase** $\gamma = \tau$: here there are two possibilities: either the last LTS rule used was (TauAmb) and the desired result follows easily from the inductive hypothesis or rule (OuTau) was used and $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{out}\, n]}_{\mathcal{C}} U$ where $P' \equiv U(0)$ and hence $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{out}\, n]\Downarrow 0}_{\mathcal{CA}} U(0)$. By the inductive hypothesis we see that $\mathcal{C}'[\![Q]\!] \xrightarrow{[\mathsf{out}\, n]\Downarrow 0}_{\mathcal{CA}} Q''$ for Q'' such that $P' \equiv U(0) \equiv \hat{\sim} \equiv Q''$. Therefore $\mathcal{C}[\![Q]\!] \xrightarrow{\tau}_{\mathcal{CA}} Q''$ by (OuTau) and (CTau).

SubCase $\gamma = [\mathsf{out}\, m] \downarrow R$: Suppose that γ is of the form $[\mathsf{out}\, m] \downarrow R$, that is we have the transitions $\mathcal{C}[\![P]\!] \xrightarrow{[\mathsf{out}\, m]\downarrow R} A(R)$ for some A such that $A(R) = P'$. By rule (OuAmb) we know that $\mathcal{C}'[\![P]\!] \xrightarrow{\mathsf{out}\, m}_{\mathcal{C}} T$ where $A \equiv T(0)(n)$. This means that we have

$$\mathcal{C}'[\![P]\!] \xrightarrow{\mathsf{out}\, m\downarrow 0nR}_{\mathcal{CA}} T(0)(n)(R) \equiv A(R) \equiv P'$$

and thus by the inductive hypothesis

$$\mathcal{C}'[\![Q]\!] \xrightarrow{\mathsf{out}\, m\downarrow 0nR}_{\mathcal{CA}} Q' \text{ with } P' \equiv\hat{\sim}\equiv Q'.$$

From this we obtain $\mathcal{C}'[\![Q]\!] \xrightarrow{\mathsf{out}\, m}_{\mathcal{C}} T'$ for some T' such that $Q' \equiv T'(0)(n)(R)$ and then, by (OuAmb), we have

$$\mathcal{C}[\![Q]\!] \xrightarrow{[\mathsf{out}\, m]}_{\mathcal{C}} T'(0)(n) \xrightarrow{R}_{\mathcal{A}} Q'$$

which yields $\mathcal{C}[\![Q]\!] \xrightarrow{\gamma}_{\mathcal{CA}} Q'$ as required.

Subcase $\gamma = [\mathsf{in}\, m] \downarrow R$: is similar to the previous subcase so we omit details of this.

Subcase $\gamma = [\overline{\mathsf{in}\, n}] \downarrow RSk$: suppose instead that γ is $[\overline{\mathsf{in}\, n}] \downarrow RSk$ derived using rule (CoInλ). We see that $\mathcal{C}[\![P]\!] \xrightarrow{[\overline{\mathsf{in}\, n}]}_{\mathcal{C}} A$ for some A such that $A(r_n^{in}(R,S,k)) \equiv P'$ (where by an abuse of notation, r_n^{in} refers to a suitably abstracted version of this skeleton). Furthermore, by Lemma 9, we see that, up to $\equiv$, A is necessarily of the form $\lambda \mathsf{Z}.\, \mathsf{Z}(\mathcal{C}'[\![P]\!])$. This means that

$$P' \equiv r_n^{in}(R,S,k)(\mathcal{C}'[\![P]\!]) \equiv n[\, k[\, R \parallel S\,] \parallel \mathcal{C}'[\![P]\!]\,].$$

Now it follows similarly that $\mathcal{C}[\![Q]\!] \xrightarrow{[\overline{\mathsf{in}\, n}]\downarrow RSk}_{\mathcal{CA}} Q'$ where

$$Q' \equiv r_n^{in}(R,S,k)(\mathcal{C}'[\![Q]\!]) \equiv n[\, k[\, R \parallel S\,] \parallel \mathcal{C}'[\![Q]\!]\,].$$

Note though that by definition of $\hat{\sim}$ we have $\mathcal{C}'[\![P]\!] \hat{\sim} \mathcal{C}'[\![Q]\!]$ and so $P' \hat{\sim} Q'$ also holds, as required.

Subcase $\gamma = \overline{\mathsf{open}\, n} \downarrow R$: this subcase is similar to the previous one and, again, we omit the details here.

Case $\mathcal{C}' \parallel R$: finally, suppose that $\mathcal{C} = \mathcal{C}' \parallel R$ and $\mathcal{C}[\![P]\!] \xrightarrow{\gamma}_{\mathcal{CA}} P'$.

Subcase $\gamma \neq \tau$: if γ is not τ then this transition must have been generated using one of the $\parallel$ rules. For those that are superstructural rules the matching transition is easily obtained by applying the inductive hypothesis. The remaining $\parallel$ rules are substructural: (L‖In),(L‖Ou), (L‖OuAmb) and their symmetric right versions. Suppose then that γ is $\alpha \downarrow \vec{R}$ and that the transition has been derived using one of the substructural rules. We know that

$$\mathcal{C}[\![P]\!] = (\mathcal{C}'[\![P]\!] \parallel R) \xrightarrow{\alpha}_{\mathcal{C}} A$$

with $P' \equiv A(\vec{R})$ and that this is derived from

$$\mathcal{C}'[\![P]\!] \xrightarrow{\alpha}_{\mathcal{C}} A' \text{ with } A \equiv \lambda \mathsf{X}.\, A'(R \parallel \mathsf{X}).$$

Notice that $\mathcal{C}'[\![P]\!] \xrightarrow{\alpha\downarrow\vec{R}^+}_{\mathcal{CA}} A'(\vec{R}^+)$ where, if $\vec{R}$ is $R_1, \ldots, R_n$ then $\vec{R}^+$ is defined to be $(R \parallel R_1), R_2, \ldots R_n$. Moreover, notice that $P' \equiv A(\vec{R}) \equiv A'(\vec{R}^+)$. Therefore we can apply the inductive hypothesis to see that $\mathcal{C}'[\![Q]\!] \xrightarrow{\alpha\downarrow\vec{R}^+}_{\mathcal{CA}} Q'$ with $P' \equiv\hat{\sim}\equiv Q'$. We also have $Q' \equiv A''(\vec{R}^+)$ for some A'' such that $\mathcal{C}'[\![Q]\!] \xrightarrow{\alpha}_{\mathcal{C}} A''$. The substructural rules then allow us to obtain

$$\mathcal{C}[\![Q]\!] \xrightarrow{\alpha}_{\mathcal{C}} \lambda \mathsf{X}.\, A''(R \parallel \mathsf{X}) \xrightarrow{\vec{R}}_{\mathcal{A}} (\lambda \mathsf{X}.\, A''(R \parallel \mathsf{X}))(\vec{R}) \equiv A''(\vec{R}^+)$$

which yields $\mathcal{C}[\![Q]\!] \xrightarrow{\gamma}_{\mathcal{CA}} Q'$ as required.

Subcase $\gamma = \tau$: we can now assume that γ is τ. If γ is derived from $\mathcal{C}'[\![P]\!]$ alone, independently of R, then it is easy to use the inductive hypothesis to obtain the required match. The more interesting cases arise through an interaction between $\mathcal{C}'[\![P]\!]$ and R derived using the (InTau) and (OpTau) rules. By commutativity of $\equiv$, there are two such cases for each of these rules. We only show the proof for the (InTau) rule as the details for the (OpTau) rule are similar.

> **Subcase** $\mathcal{C}'[\![P]\!]$ provides an $[\mathsf{in}\, n]$ action: suppose that $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{in}\, n]}_{\mathcal{C}} U$ and $R \xrightarrow{\overline{[\mathsf{in}\, n]}}_{\mathcal{C}} A$ where $P' \equiv A(U)$. We know by Lemma 9 that $A \equiv \lambda \mathsf{Z}.\, \mathcal{E}[\![\mathsf{Z}(R')]\!]$ for some R'. We can derive $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{in}\, n]\downarrow R'}_{\mathcal{CA}} U(R')$ and then apply the inductive hypothesis to see that $\mathcal{C}'[\![Q]\!] \xrightarrow{[\mathsf{in}\, n]\downarrow R'}_{\mathcal{CA}} Q''$ with $U(R') \equiv\hat{\sim}\equiv Q''$. It must be the case, however, that $Q'' \equiv U'(R')$ for

some U' such that $\mathcal{C}'[\![Q]\!] \xrightarrow{[\mathsf{in}\,n]}_{\mathcal{C}} U'$. This tells us, by applying (InTau), that

$$\mathcal{C}[\![Q]\!] = (\mathcal{C}'[\![Q]\!] \parallel R) \xrightarrow{\tau}_{\mathcal{CA}} A(U') \equiv \mathcal{E}[\![U'(R')]\!]$$

Now we know that, by congruence of $\hat{\sim}$ and $\equiv$, that

$$P' \equiv A(U) \equiv \mathcal{E}[\![U(R')]\!] \equiv \hat{\sim} \equiv \mathcal{E}[\![Q'']\!] \equiv \mathcal{E}[\![U'(R')]\!] \equiv A(U')$$

as required.

Subcase $\mathcal{C}'[\![P]\!]$ provides an $[\overline{\mathsf{in}\,n}]$ action: for the final case, suppose that $\mathcal{C}'[\![P]\!] \xrightarrow{[\overline{\mathsf{in}\,n}]}_{\mathcal{C}} A$ and $R \xrightarrow{[\mathsf{in}\,n]}_{\mathcal{C}} U$ with $P' \equiv A(U)$. In this case, by Lemma 9 we know that

$$A \equiv \lambda \mathsf{Z}.\, \mathcal{E}'[\![\mathsf{Z}(P'')]\!] \text{ and } U \equiv \lambda \mathsf{Y}.\, \mathcal{E}[\![n[\, m[\, R' \parallel R''\,] \parallel \mathsf{Y}\,]]\!]$$

for some P'', m, R, R''. Thus, writing $P_0 \stackrel{\text{def}}{=} A(\lambda \mathsf{Y}.\, n[\, m[\, R' \parallel R''\,] \parallel \mathsf{Y}\,])$,

$$\begin{aligned} P' \equiv A(U) &\equiv \mathcal{E}'[\![\mathcal{E}[\![n[\, m[\, R' \parallel R''\,] \parallel P''\,]]\!]]\!] \\ &\equiv \mathcal{E}[\![\mathcal{E}'[\![n[\, m[\, R' \parallel R''\,] \parallel P''\,]]\!]]\!] \qquad \text{(by Lemma 14)} \\ &\equiv \mathcal{E}[\![A(\lambda \mathsf{Y}.\, n[\, m[\, R' \parallel R''\,] \parallel \mathsf{Y}\,])]\!] \\ &\equiv \mathcal{E}[\![P_0]\!] \end{aligned}$$

Now, by rule (CoInλ), we know that $\mathcal{C}'[\![P]\!] \xrightarrow{[\overline{\mathsf{in}\,n}]\downarrow R' R'' m}_{\mathcal{CA}} P_0$ therefore we can apply the inductive hypothesis to obtain $\mathcal{C}'[\![Q]\!] \xrightarrow{[\overline{\mathsf{in}\,n}]\downarrow R' R'' m}_{\mathcal{CA}} Q_0$ with $P_0 \equiv \hat{\sim} \equiv Q_0$. In fact, $Q_0 \equiv A'(\lambda \mathsf{Y}.\, n[\, m[\, R' \parallel R''\,] \parallel \mathsf{Y}\,])$ for some A' such that $\mathcal{C}'[\![Q]\!] \xrightarrow{[\overline{\mathsf{in}\,n}]}_{\mathcal{C}} A'$. By applying the rule (InTau) to this and the co-action from R we get

$$\mathcal{C}[\![Q]\!] = (\mathcal{C}'[\![Q]\!] \parallel R) \xrightarrow{\tau}_{\mathcal{CA}} A'(U).$$

Reasoning as above we also obtain $A'(U) \equiv \mathcal{E}[\![Q_0]\!]$. Hence $P' \equiv \mathcal{E}[\![P_0]\!] \equiv \hat{\sim} \equiv \mathcal{E}[\![Q_0]\!] \equiv A'(U)$ as required.

This takes care of all possible cases for $\mathcal{C}$ and hence concludes the proof. □

Theorem 16 (Soundness)**.** *$P \sim_{\mathcal{CA}} Q$ implies $P \simeq Q$.*

Proof. We shall show that $\sim_{\mathcal{CA}}$ satisfies the defining properties of barbed congruence. These are (i) preservation of reduction, (ii) preservation of barbs, and (iii) congruence.

Suppose that $P \sim_{\mathcal{CA}} Q$. We begin by showing (i). Suppose that $P \rightarrow P'$. We know by Proposition 13, that there exists P'' such that $P'' \equiv P'$ and $P \xrightarrow{\tau} P'$. Then there exists a Q'' such that $Q \xrightarrow{\tau} Q''$ and $P'' \sim_{\mathcal{CA}} Q''$. Using Proposition 13 we have $Q \rightarrow Q''$ and, using the fact that $\equiv \subseteq \sim_{\mathcal{CA}}$ (see Lemma 11) and transitivity, we are done.

To show (ii), we suppose that $P \downarrow_m$. By definition, this tells us that $P = \mathcal{E}[\![m[\,P'\,]]\!]$ for some P' such that m is not captured by $\mathcal{E}$. We know then that $P \xrightarrow{\overline{\mathsf{open}\, m}\downarrow 0} R$ for some R. By bisimilarity, we have $Q \xrightarrow{\overline{\mathsf{open}\, m}\downarrow 0} R'$ for some R'. By Lemma 9, $Q = \mathcal{F}[\![m[\,Q'\,]]\!]$ for some Q' and $\mathcal{F} \# m$, hence $Q \downarrow_m$.

For (iii) we need to demonstrate that $\mathcal{C}[\![P]\!] \sim_{\mathcal{CA}} \mathcal{C}[\![Q]\!]$ holds for all $\mathcal{C}$. This however is the precisely the remit of Proposition 15. □

5. Completeness

With soundness of bisimilarity established we shall now consider the converse property: completeness. The central issue here is the *observability* of actions. As encapsulated by the statement of Lemma 12, 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. Indeed, for an action α to be *observable* there must exist a suitable predicate on terms that (i) characterises when a term is the source of an α-labelled transition and (ii) is preserved by contextual equivalence (see Proposition 20). Whether or not this is the case for particular α depends on the language at hand.

Essentially, one needs to show that each kind of label has a context that characterises it. This is a stronger requirement than that of Lemma 12 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 for this to occur, contexts must contain more information.

When such a contextual condition does not exist for a particular label, one can use an additional Honda-Tokoro ($\mathcal{HT}$) rule in the SOS specification to ensure its existence and hence completeness of bisimilarity for contextual equivalence. For the background and examples of such rules see [30]. In the setting of ambients they are needed for $[\mathsf{in}\, n]$ and $[\mathsf{out}\, n]$ transitions only (see Fig. 5) and account for the following situation: the environment provides an appropriate context χ (as in Lemma 12) but the process does not make use of it, thus χ is retained in the result of the interaction. As an example of the necessity, in general, of the $\mathcal{HT}$ rules for completeness, consider:

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

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

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

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

Remark 17 (Completeness and finite processes)**.** It is unclear whether bisimilarity on $\mathcal{CA}$ is complete with respect to reduction barb congruence in the *finite* language—similar questions have been studied in [30] in a simpler setting. Indeed, based on the examples therein, it is likely that $\mathcal{CA}$ is already complete. Simply adding replication, however, results in a language for which the $\mathcal{CA}$ is not complete, as illustrated by the preceding example. Indeed, in the full ambient calculus, an ambient's ability to migrate is unobservable.[3]

We would thus consider completeness of bisimilarity on $\mathcal{CA}$ for the finite language, speaking colloquially, as 'completeness by accident' and not an important fact: for us the 'essence' of completeness lies in a local contextual characterisation of actions in the spirit of Proposition 20. Indeed, an LTS on which bisimilarity is proved complete in this sense enjoys the property that its actions remain observable under various language extensions.

[3]This fact has been observed in [19] 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 [16] style rules for strong equivalences instead.

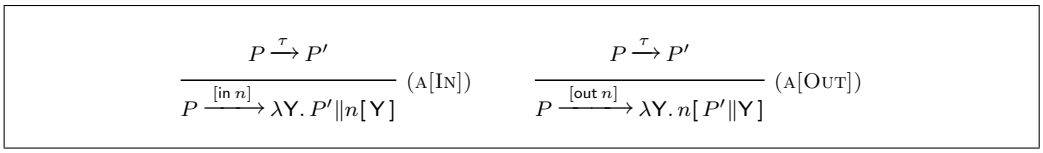


Figure 5: Honda-Tokoro rules $\mathcal{HT}$ for unobservable actions

The following lemma states that adding the $\mathcal{HT}$ rules (see Fig. 5). does not generate new τ-labelled transitions. It is needed to show the soundness and completeness of the extension.

Lemma 18. *τ-labelled transitions in $\mathcal{C} + \mathcal{HT}$ agree with reductions.*

Proof. Induction on the number of $\mathcal{HT}$ rules present in the derivation of a τ-transition, relying on the conclusion of Proposition 13. Essentially, we show that any use of the $\mathcal{HT}$ rules can be cut from any given derivation.

For example, we shall show that any derivation with $k+1$ applications of (A[IN]) can be replaced with an equivalent derivation with k applications. Let us consider the final use of (A[IN]). To be discharged, the resulting $[\mathsf{in}\, n]$ needs to be combined with a $[\overline{\mathsf{in}\, n}]$ in essentially the following derivation snippet:

$$\dfrac{\dfrac{P \xrightarrow{\tau} P'}{P \xrightarrow{[\mathsf{in}\, n]} \lambda \mathsf{X}.\, P' \| n[\,\mathsf{X}\,]}\ (\text{A[IN]}) \qquad \dfrac{}{n[\,Q\,] \xrightarrow{[\overline{\mathsf{in}\, n}]} \lambda \mathsf{Z}.\, \mathsf{Z}(Q)}\ (\text{CoIn})}{P \| n[\,Q\,] \xrightarrow{\tau} P' \| n[\,Q\,]}\ (\text{InTau})$$

which can be replaced simply by

$$\dfrac{P \xrightarrow{\tau} P'}{P \| n[\,Q\,] \xrightarrow{\tau} P' \| n[\,Q\,]}\ (\text{L}\| \text{ Tau})$$

□

Indeed, bisimilarity $\sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}}$ on the obtained LTS remains sound for contextual equivalence.

Proposition 19 (Soundness, $(\mathcal{C} + \mathcal{HT})\mathcal{A}$)**.** *$P \sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}} Q$ implies $P \simeq Q$.*

Proof. It is enough to show that bisimilarity remains a congruence, since the remainder of the proof of Theorem 16 is unaffected. Moreover the conclusion

of Lemma 18 tells us that τ-labelled transitions coincide with reductions, and so in particular with the τ-labelled transitions in $\mathcal{CA}$.

It suffices then to consider the possible ways in which $\mathcal{HT}$ rules affect the proof of Proposition 15. We must inspect each case where the rules (A[In]) and (A[Out]) can be applied.

First, for any context $\mathcal{C}$, if $\mathcal{C}[\![P]\!] \xrightarrow{\alpha\downarrow R} \mathcal{D}[\![P']\!]$ is derived from either (A[In]) or (A[Out]) then also $\mathcal{C}[\![P]\!] \xrightarrow{\tau} P'$. But this is matched by $\mathcal{C}[\![Q]\!] \xrightarrow{\tau} Q'$ such that $P' \mathrel{\hat{\sim}} Q'$; then $\mathcal{C}[\![P]\!] \xrightarrow{\alpha\downarrow R} \mathcal{D}[\![Q']\!]$ and, by definition of $\hat{\sim}$, $\mathcal{D}[\![P']\!] \mathrel{\hat{\sim}} \mathcal{D}[\![Q']\!]$.

Now consider the case $\mathcal{C} = \mathcal{C}' \parallel R$ where $\mathcal{C}[\![P]\!] \xrightarrow{\tau} P'$ and the sub case where $\mathcal{C}'[\![P]\!]$ provides a $[\mathsf{in}\, n]$ action:

$$\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{in}\, n]} U \quad \text{and} \quad R \xrightarrow{\overline{[\mathsf{in}\, n]}} A \quad \text{with} \quad P' \equiv A(U)$$

But then $R = \mathcal{E}[\![n[\, R'\,]]\!]$ and $A = \lambda \mathsf{Z}.\, \mathcal{E}[\![\mathsf{Z}(R')]\!]$; and so $A(U) = \mathcal{E}[\![U(R')]\!]$. Now $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{in}\, n]\downarrow R'} U(R')$ and with the ind. hyp. $\mathcal{C}'[\![Q]\!] \xrightarrow{[\mathsf{in}\, n]\downarrow R'} Q' \mathrel{\hat{\sim}} U(R')$. Suppose that the latter transition was derived with (A[In]): then $\mathcal{C}'[\![Q]\!] \xrightarrow{\tau} Q''$ and $Q' = Q'' \parallel n[\, R'\,]$. But now

$$\begin{aligned}\mathcal{C}[\![Q]\!] &= \mathcal{C}'[\![Q]\!] \parallel R \\ &= \mathcal{C}'[\![Q]\!] \parallel \mathcal{E}[\![n[\, R'\,]]\!] \xrightarrow{\tau} Q'' \parallel \mathcal{E}[\![n[\, R'\,]]\!] \equiv \mathcal{E}[\![Q']\!] \mathrel{\hat{\sim}} \mathcal{E}[\![U(R')]\!] = P\end{aligned}$$

A similar, if technically simpler, argument applies in the subcase where $\mathcal{C}'[\![P]\!]$ provides a $[\mathsf{out}\, n]$ action.

The final case is $\mathcal{C} = n[\, \mathcal{C}'\,]$ with $\mathcal{C}[\![P]\!] \xrightarrow{\tau} P'$ deriving from a $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{out}\, n]} U$ action. Then $P' = U(0)$ and thus we have $\mathcal{C}'[\![P]\!] \xrightarrow{[\mathsf{out}\, n]\downarrow 0} P'$. This, using the ind. hyp., is matched with $\mathcal{C}'[\![Q]\!] \xrightarrow{[\mathsf{out}\, n]\downarrow 0} Q'$ with $P' \mathrel{\hat{\sim}} Q'$. If the latter arises from an application of (A[Out]) then we have $\mathcal{C}'[\![Q]\!] \xrightarrow{\tau} Q''$ and $Q' = n[\, Q'' \parallel 0\,] \equiv n[\, Q''\,]$. But then $\mathcal{C}[\![Q]\!] = n[\, \mathcal{C}'[\![Q]\!]\,] \xrightarrow{\tau} n[\, Q''\,] \equiv Q' \mathrel{\hat{\sim}} P'$. □

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 that is stable under reduction barbed congruence. This, unfortunately, is technical, calculus-specific work and is not particularly illuminating. The appendix is devoted to obtaining the right predicates, as summarised by the following:

Proposition 20. *Lemmas 22, 23, 24, 25 and 26 in the appendix concern all the non-τ labels:*

$$\alpha \in \{\mathsf{open}\, n \downarrow R,\ \overline{\mathsf{open}\, n} \downarrow R,\ [\overline{\mathsf{in}\, n}] \downarrow RSk,\ \mathsf{in}\, n \downarrow RkS,\ \mathsf{out}\, n \downarrow RkS,\ [\mathsf{in}\, n] \downarrow R,\ [\mathsf{out}\, n] \downarrow R\}$$

and for each such α introduce a binary predicate $\Phi_\alpha(P, P')$ on terms that characterises labelled transitions in the following sense:

- *if $P \xrightarrow{\alpha} P'$ then $\Phi_\alpha(P, P')$;*
- *if $\Phi_\alpha(P, P')$ then $\exists P'' \equiv P'$ such that $P \xrightarrow{\alpha} P''$.*

Additionally, the predicates Φ_α are stable under barbed congruence in the following sense: if $P \simeq Q$ and there exists P' such that $\Phi_\alpha(P, P')$ then there exists Q' such that $P' \simeq Q'$ and $\Phi_\alpha(Q, Q')$. □

These individual characterisations allow us to easily prove completeness.

Theorem 21 (Completeness)**.** *$P \simeq Q$ implies $P \sim_{(\mathcal{C}+\mathcal{HT})\mathcal{A}} Q$.*

Proof. We need to show that $\simeq$ is a bisimulation. Suppose that $P \xrightarrow{\alpha} P'$:

- if $\alpha = \tau$ then the result follows immediately from Lemma 18;
- otherwise we use the appropriate predicate Φ_α (see Proposition 20). Indeed, if $P \xrightarrow{\alpha} P'$ then $\Phi_\alpha(P, P')$. Then since $P \simeq Q$, there exists Q' such that $Q \simeq Q'$ and $\Phi_\alpha(Q, Q')$. Then there exists $Q'' \equiv Q'$ such that $Q \xrightarrow{\alpha} Q''$. Using the fact that $\equiv \subseteq \simeq$ and the transitivity of $\simeq$ we have $P' \simeq Q''$.

□

6. Conclusions, related and future work.

The introduction of the ambient calculus in [8] has spawned a considerable amount of research on the topic regarding variants of the calculus (*e.g.* [12, 11, 4]), type systems (*e.g.* [20, 7, 5]) and implementation details (*e.g.* [14, 25]). However, there has been relatively little work on labelled characterisations. An early attempt by Cardelli and Gordon [6] was abandoned in favour of a simpler approach in [9]. Interestingly, the structural rules and

use of abstractions in the meta-language was already present in [6] where the authors seemed to encounter difficulties 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 [6, 9], Merro and Zappa-Nardelli [19] 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. Indeed, our belief is that the main significance of our contribution is not the introduction of a new LTS for ambients but rather a step towards generally applicable techniques for the derivation of labelled transition systems. In this we were fortunate in having had the model in [19] 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 that built on the initial attempts by Cardelli and Gordon [6] (that 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 that 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 [19]. Notice, for example, that our base rules (In) and (Ou) of Fig. 2 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 [19] serve primarily to recover this necessary structure. Our treatment of co-actions, in rules (coInλ) and (coOpenλ) of Fig. 4, by completing them with skeletal structure as well as missing parameters, is mirrored in the rules (Co-Enter) and (Open) of [19] 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 [19] 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 an enclosing ambient in the label because this information is not determined by the context and the name's identity must be subsequently discovered by some context parameter processes. Unlike [19] we deal with the unobservability of $[\mathsf{in}\, n]$ and $[\mathsf{out}\, n]$ actions using Honda Tokoro style [16] rules in Fig. 5 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 [19] that 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 [13, 11, 12, 4]. 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 worthwhile 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 [33] 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 [17] 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 [17] makes it difficult for us to find any striking similarities with our SOS rule-set and LTS.

In this paper and in [28] we have considered strong bisimilarity. Because Proposition 13 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

$$P \xrightarrow{\tau}_{\mathcal{CA}} \cdots \xrightarrow{\alpha}_{\mathcal{CA}} \cdots \xrightarrow{\tau}_{\mathcal{CA}} P'$$

we shall 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 shall need to consider weak transitions from the $\mathcal{C}$ and $\mathcal{A}$ transitions systems separately. The difficulties that may arise in the weak case lie in providing contexts that witness weak transitions for the proof of completeness.

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 [28], 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. An analogous refinement for ambients would be desirable, albeit very difficult; we believe that our LTS serves as a good basis from which to do this.

Having experimented on the π-calculus [28] 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 wider range of calculi.

Acknowledgment. We thank the anonymous referees for their useful remarks, which have helped us to significantly improve the presentation of the paper.

References

[1] R. Amadio, I. Castellani, D. Sangiorgi, On bisimulations for the asynchronous pi-calculus, Theor. Comput. Sc. 195 (2) (1998) 291–324.

[2] J. Bergstra, J. Klop, Algebra of communicating processes with abstraction, Theor. Comput. Sc. 37 (1) (1985) 77–121.

[3] F. Bonchi, F. Gadducci, G. Monreale, Reactive systems, barbed semantics, and the mobile ambients, in: Proc. FoSSaCS, vol. 5504 of LNCS, Springer, 2009, pp. 272–287.

[4] M. Bugliesi, S. Crafa, M. Merro, V. Sassone, Communication interference in mobile boxed ambients, Inf. Comput. 205 (2007) 1235–1273.

[5] L. Cardelli, G. Ghelli, A. Gordon, Mobility types for mobile ambients, in: Proc. ICALP, vol. 1644 of LNCS, Springer, 1999, pp. 230–239.

[6] L. Cardelli, A. Gordon, A commitment relation for the ambient calculus, unpublished notes (1996).

[7] L. Cardelli, A. Gordon, Types for mobile ambients, in: Proc. PoPL, ACM Press, 1999, pp. 79–92.

[8] L. Cardelli, A. Gordon, Mobile ambients, Theor. Comput. Sc. 240/1 (2000) 177–213.

[9] L. Cardelli, A. Gordon, Equational properties of mobile ambients, Math. Struct. Comput. Sc. 13 (3) (2003) 371–408.

[10] U. Engberg, M. Nielsen, A calculus of communicating systems with label passing, Tech. Rep. DAIMI PB-208, University of Aarhus (May 1986).

[11] Y. Fu, Fair ambients, Acta Inf. 43 (8) (2007) 535–594.

[12] P. Garralda, E. Bonelli, A. Compagnoni, M. Dezani-Ciancaglini, Boxed Ambients with Communication Interfaces, Math. Struct. Comput. Sc. 17 (2007) 1–59.

[13] M. Hennessy, M. Merro, Bisimulation congruences in safe ambients, ACM T. Progr. Lang. Sys. 28(2) (2006) 290–330.

[14] D. Hirschkoff, D. Pous, D. Sangiorgi, An efficient abstract machine for safe ambients, J. Logic Algebr. Progr. 71 (2007) 114–149.

[15] C. A. R. Hoare, Communicating Sequential Processes, Prentice Hall, 1985.

[16] K. Honda, M. Tokoro, An object calculus for asynchronous communication, in: Proc. ECOOP, vol. 512 of LNCS, Springer, 1991, pp. 133–147.

[17] O. Jensen, R. Milner, Bigraphs and mobile processes, Tech. Rep. 570, Computer Laboratory, University of Cambridge (2003).

[18] J. Leifer, R. Milner, Deriving bisimulation congruences for reactive systems, in: Proc. Concur, vol. 1877 of LNCS, Springer, 2000, pp. 243–258.

[19] M. Merro, F. Z. Nardelli, Behavioural theory for mobile ambients, J. ACM 52 (6) (2005) 961–1023.

[20] M. Merro, V. Sassone, Typing and subtyping mobility in boxed ambients, in: Proc. Concur, vol. 2421 of LNCS, Springer, 2002, pp. 304–320.

[21] R. Milner, A Calculus of Communicating Systems, vol. 92 of LNCS, Springer, 1980.

[22] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, II, Inf. Comput. 100 (1) (1992) 41–77.

[23] R. Milner, D. Sangiorgi, Barbed bisimulation, in: Proc. ICALP, No. 623 in LNCS, 1992, pp. 685–695.

[24] F. Pfenning, C. Elliott, Higher-order abstract syntax, in: Proc. PLDI, vol. 23(7) of SIGPLAN Notices, 1988, pp. 199–208.

[25] A. Phillips, Specifying and implementing secure mobile applications in the channel ambient system, Ph.D. thesis, Imperial College London (April 2006).

[26] G. D. Plotkin, A structural approach to operational semantics, J. Logic Algebr. Progr. 60-61 (2004) 17–139, originally appeared as Technical Report DAIMI FN-19, University of Aarhus, 1981.

[27] J. Rathke, V. Sassone, P. Sobociński, Semantic barbs and biorthogonality, in: Proc. FoSSaCS, vol. 4423 of LNCS, Springer, 2007, pp. 302–316.

[28] J. Rathke, P. Sobociński, Deconstructing behavioural theories of mobility, in: Proc. TCS, vol. 273 of IFIP, Springer, 2008, pp. 507–520.

[29] J. Rathke, P. Sobociński, Deriving structural labelled transitions for mobile ambients, in: Proc. Concur, vol. 5201 of LNCS, Springer, 2008, pp. 462–476.

[30] J. Rathke, P. Sobociński, Making the unobservable, unobservable, Electron. Notes Theor. Comput. Sc. 229 (3) (2009) 131–144.

[31] V. Sassone, P. Sobociński, Deriving bisimulation congruences using 2-categories, Nordic J. Comput. 10 (2) (2003) 163–183.

[32] V. Sassone, P. Sobociński, Reactive systems over cospans, in: Proc. LiCS, IEEE Press, 2005, pp. 311–320.

[33] P. Sewell, From rewrite rules to bisimulation congruences, Theor. Comput. Sc. 274 (1-2) (2002) 183–230, extended version of a Concur '98 conference paper.

Appendix

Here we obtain the necessary predicates described in Proposition 20. While the proofs are fairly technical, there are no sophisticated techniques used and most of the work is done by the structural lemmas (Lemmas 9 and 10).

The only real difficulty is in ensuring that the context-view contribution in the label does not take part in the interaction—we solve this problem by inhibiting its participation. Basically, the contributed processes are not used directly but are guarded by an open prefix that destroys a fresh-named ambient. Let $\{P\}^i$ (read "inhibit P") $\stackrel{\text{def}}{=} i[\,] \parallel \mathsf{open}\, i.P$. Clearly $\{P\}^i{\downarrow_i}$ and $\{P\}^i \rightarrow P$. To ascertain that such an *inhibited* P does not take part, one checks for the presence of the ambient i after the interaction takes place.

Lemma 22 (Contextually characterising open and $\overline{\mathsf{open}}$)**.** *Let:*

$$\chi_{\mathsf{open}\, n\downarrow R} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel n[\,\{R\}^i\,], \qquad \chi_{\overline{\mathsf{open}\, n}\downarrow R} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{open}\, n.\{R\}^i, \text{ and}$$

$$\Phi(P,P') \stackrel{\text{def}}{=} \exists P_1.\, \chi_\alpha[\![P]\!] \rightarrow P_1,\, P_1{\downarrow_i},\, P_1 \rightarrow P',\, P'{\not\downarrow_i}.$$

Then for $\alpha \in \{\mathsf{open}\, n \downarrow R, \overline{\mathsf{open}\, n} \downarrow R\}$*:*

(i) if $P \xrightarrow{\alpha} P'$ *then* $\Phi(P,P')$*;*

(ii) if $\Phi(P,P')$ *then* $\exists P'' \equiv P'$ *such that* $P \xrightarrow{\alpha} P''$*.*

Proof. (i) If $P \xrightarrow{\mathsf{open}\, n\downarrow R} P'$ then $P = \mathcal{E}[\![\mathsf{open}\, n.Q]\!]$ and

$$P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![P' \parallel \mathsf{Y}]\!])(R) \qquad \text{(Lemma 10)}.$$

Now $\chi_{\mathsf{open}\, n\downarrow R}[\![P]\!] = \mathcal{E}[\![\mathsf{open}\, n.Q]\!] \parallel n[\,\{R\}^i\,] = \mathcal{E}'[\![\mathsf{open}\, n.Q' \parallel n[\,\{R\}^i\,]]\!] \rightarrow \mathcal{E}'[\![Q' \parallel \{R\}^i]\!] = (\mathcal{E}[\![Q]\!] \parallel \{R\}^i)\downarrow_i$ and $\mathcal{E}[\![Q]\!] \parallel \{R\}^i \rightarrow P'$. Similarly if $P \xrightarrow{\overline{\mathsf{open}\, n}\downarrow R} P'$ then $P = \mathcal{E}[\![n[\,Q\,]]\!]$ and $P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![\mathsf{Y} \parallel Q]\!])(R)$. Then we have $\chi_{\overline{\mathsf{open}\, n}\downarrow R}[\![P]\!] = \mathcal{E}[\![n[\,Q\,]]\!] \parallel \mathsf{open}\, n.\{R\}^i = \mathcal{E}'[\![n[\,Q'\,] \parallel \mathsf{open}\, n.\{R\}^i]\!] \rightarrow \mathcal{E}'[\![Q' \parallel \{R\}^i]\!] = \mathcal{E}[\![Q]\!] \parallel \{R\}^i\downarrow_i$. Moreover $\mathcal{E}[\![Q]\!] \parallel \{R\}^i \rightarrow P'$. In both instances, P' does not contain i and so cannot barb on it in any interaction context.

(ii) Suppose that $\chi_{\mathsf{open}\, n\downarrow R}[\![P]\!] = P \parallel n[\,\{R\}^i\,] \rightarrow P_1$ such that $P_1\downarrow_i$. Since $n[\,\{R\}^i\,]$ cannot reduce without destroying the barb i and $n[\,\{R\}^i\,]{\not\downarrow_i}$, the reduction must involve both P and $n[\,\{R\}^i\,]$. The unique possibility that

leaves i at top level is $P = \mathcal{E}[\![\mathsf{open}\, n.Q]\!]$ and so we have $P_1 \equiv \mathcal{E}[\![Q]\!] \parallel \{R\}^i$. Now if $P_1 \rightarrow P'$ such that $P' \not\downarrow_i$ then it follows that the unique possible reduction is $P_1 \rightarrow \mathcal{E}[\![Q]\!] \parallel R = (\lambda \mathsf{Y}.\, \mathcal{E}[\![Q \parallel \mathsf{Y}]\!])(R)$. Then by Lemma 10, we have $P \xrightarrow{\mathsf{open}\, n \downarrow R} P''$ for some $P'' \equiv P'$.

Similarly, if $\chi_{\overline{\mathsf{open}\, n}\downarrow R}[\![P]\!] = P \parallel \mathsf{open}\, n.\{R\}^i \rightarrow P_1\downarrow_i$ then $P = \mathcal{E}[\![n[\, Q\,]]\!]$ and $P_1 = \mathcal{E}[\![Q]\!] \parallel \{R\}^i$. Now if $P_1 \rightarrow P' \not\downarrow_i$ then $P' = \mathcal{E}[\![Q]\!] \parallel R \equiv \lambda \mathsf{Y}.\, \mathcal{E}[\![\mathsf{Y} \parallel Q]\!](R)$. We thus have $P \xrightarrow{\overline{\mathsf{open}\, n}\downarrow R} P'$. □

Lemma 23 (Contextually characterising $[\overline{\mathsf{in}}]$)**.** *Let:*

$$\chi \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel k[\, \mathsf{in}\, n.\{R \parallel S\}^{i_1}\,], \qquad \xi \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{open}\, n.\mathsf{open}\, k.\{0\}^{i_2}, \text{ and}$$

$$\begin{aligned}\Phi(P, P') \stackrel{\text{def}}{=}\ & \exists P_1,\, P_2.\, \chi[\![P]\!] \rightarrow P_1,\, \xi[\![P_1]\!] \rightarrow {}^2P_2, \\ & P_2\downarrow_{i_1,i_2},\, P_1 \rightarrow P',\, \forall \theta.\, (i_1 \notin \theta) \rightarrow (\theta[\![P']\!] \not\downarrow_{i_1}).\end{aligned}$$

Then:

(i) if $P \xrightarrow{[\overline{\mathsf{in}}\, n]\downarrow RSk} P'$ *then* $\Phi(P, P')$*;*

(ii) if $\Phi(P, P')$ *then* $\exists P'' \equiv P'$ *such that* $P \xrightarrow{[\overline{\mathsf{in}}\, n]\downarrow RSk} P''$*.*

Proof. (*i*) If $P \xrightarrow{[\overline{\mathsf{in}}\, n]\downarrow RSk} P'$ then $P = \mathcal{E}[\![n[\, P^\dagger\,]]\!]$ and

$$P' \equiv (\lambda \mathsf{YZx}.\, \mathcal{E}[\![n[\, \mathsf{x}[\, \mathsf{Y} \parallel \mathsf{Z}\,] \parallel P^\dagger\,]]\!])(RSk).$$

Then

$$\begin{aligned}\chi[\![\mathcal{E}[\![n[\, P^\dagger\,]]\!]]\!] &= \mathcal{E}[\![n[\, P^\dagger\,]]\!] \parallel k[\, \mathsf{in}\, n.\{R \parallel S\}^{i_1}\,] \\ &= \mathcal{E}'[\![n[\, P^\ddagger\,] \parallel k[\, \mathsf{in}\, n.\{R \parallel S\}^{i_1}\,]]\!] \rightarrow \mathcal{E}'[\![n[\, P^\ddagger \parallel k[\, \{R \parallel S\}^{i_1}\,]\,]]\!].\end{aligned}$$

Now

$$\begin{aligned}&\xi[\![\mathcal{E}'[\![n[\, P^\ddagger \parallel k[\, \{R \parallel S\}^i\,]\,]]\!]]\!] \\ &\quad \equiv \mathcal{E}'[\![n[\, P^\ddagger \parallel k[\, \{R \parallel S\}^{i_1}\,]\,] \parallel \mathsf{open}\, n.\mathsf{open}\, k.\{0\}^{i_2}]\!] \\ &\quad\quad \rightarrow \mathcal{E}[\![P^\dagger \parallel k[\, \{R \parallel S\}^i\,] \parallel \mathsf{open}\, k.\{0\}^j]\!] \\ &\quad\quad\quad \rightarrow \mathcal{E}[\![P^\dagger \parallel \{R \parallel S\}^i \parallel \{0\}^j]\!]\downarrow_{i_1,i_2}.\end{aligned}$$

Clearly also $\mathcal{E}'[\![n[\,P^{\ddagger} \parallel k[\,\{R \parallel S\}^{i_1}\,]\,]]\!] \rightarrow P'$ and P' does not contain instances of i_1.

(ii) Suppose $\chi[\![P]\!] = P \parallel k[\,\mathsf{in}\,n.\{R \parallel S\}^{i_1}\,] \rightarrow P_1$ and $\xi[\![P_1]\!] = P_1 \parallel \mathsf{open}\,n.\mathsf{open}\,k.\{0\}^{i_2} \rightarrow {}^2P_2$ such that $P_2 \downarrow_{i_1,i_2}$. First notice that $P_1 \not\downarrow_{i_1}$, since the only reduction that "unlocks" the barb is the insertion of the k ambient into an n ambient, and in that case the ambient is not at the top level. Now because $P_2 \downarrow_{i_2}$ we must have either

$$P_1 \equiv \mathcal{E}'[\![n[\,Q_1\,] \parallel k[\,Q_2\,]]\!] \qquad \text{or} \qquad P_1 \equiv \mathcal{E}'[\![n[\,k[\,Q_1\,] \parallel Q_2\,]]\!]$$

Note that, since $P_1 \not\downarrow_{i_1}$, we have that $\mathcal{E}'[\![0]\!] \not\downarrow_{i_1}$. In the first case we have $P_2 = \mathcal{E}'[\![Q_1 \parallel Q_2 \parallel \{0\}^{j}]\!]$. Since also $P_2 \downarrow_{i_1}$ we must have either $Q_1 \downarrow_{i_1}$ or $Q_2 \downarrow_{i_1}$. Then an examination of the possible targets of the first reduction then confirms that the first choice for P_1 is impossible. Hence $P_1 \equiv \mathcal{E}'[\![n[\,k[\,Q_1\,] \parallel Q_2\,]]\!]$ and moreover, $Q_1 \equiv \mathcal{E}''[\{R \parallel S\}^{i_1}]$. This leaves just one possible reduction, and we conclude that $P_1 \equiv \mathcal{E}[\![n[\,k[\,\{R \parallel S\}^{i}\,] \parallel Q_2\,]]\!]$, which means that $P \equiv \mathcal{E}[\![n[\,Q_2\,]]\!]$. Finally, it is again easy to see that the only possible reduction from P_1 which renders i_1 invisible to any context is the reduction that destroys it, ie $P' \equiv \mathcal{E}[\![n[\,k[\,R \parallel S\,] \parallel Q_2\,]]\!]$. It follows from the Lemma 10 that $P \xrightarrow{\overline{[\mathsf{in}\,n]}\downarrow RSk} (\lambda \mathsf{YZx}.\,\mathcal{E}[\![n[\,\mathsf{x}[\,\mathsf{Y} \parallel \mathsf{Z}\,] \parallel Q_2\,]]\!])(RSk) \equiv P'$. □

Lemma 24 (Contextually characterising in)**.** *Let:*

$$\chi \stackrel{\text{def}}{=} k[\,\mathsf{1_{Pr}} \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,],$$

$$\xi^{+} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{open}\,n.\mathsf{open}\,k.\{0\}^{i_3}, \quad \xi^{-} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{open}\,k.\{0\}^{i_3}$$

$$\begin{aligned}\Phi(P,P') \stackrel{\text{def}}{=}\ & \exists P_1,\, P_2.\, \chi[\![P]\!] \rightarrow P_1,\, \xi^{+}[\![P_1]\!] \rightarrow {}^2P_2{\downarrow_{i_1,i_2,i_3}},\\ & \forall T.\, \xi^{-}[\![P_1]\!] \rightarrow T,\, T \not\downarrow_{i_2,i_3},\, P_1 \rightarrow {}^2P',\, \forall\theta.\, (i_1,\, i_2 \notin \theta) \rightarrow (\theta[\![P']\!] \not\downarrow_{i_1,i_2})\end{aligned}$$

Then:

(i) if $P \xrightarrow{\mathsf{in}\,n\downarrow RkS} P'$ *then* $\Phi(P,P')$*;*

(ii) if $\Phi(P,P')$ *then* $\exists P'' \equiv P'$ *such that* $P \xrightarrow{\mathsf{in}\,n\downarrow RkS} P''$*.*

Proof. *(i)* If $P \xrightarrow{\mathsf{in}\,n\downarrow RkS} P'$ then $P = \mathcal{E}[\![\mathsf{in}\,n.Q]\!]$ and

$$P' \equiv (\lambda \mathsf{XxY}.\, n[\,\mathsf{x}[\,\mathcal{E}[\![Q]\!] \parallel \mathsf{X}\,] \parallel \mathsf{Y}\,])(RkS).$$

So $\chi[\![P]\!] = k[\,\mathcal{E}[\![\mathsf{in}\,n.Q]\!] \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,] \rightarrow n[\,k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2}\,] \parallel \{S\}^{i_1}\,]$. Now substituting the right-hand side into ξ^+ gives $n[\,k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2}\,] \parallel \{S\}^{i_1}\,] \parallel \mathsf{open}\,n.\mathsf{open}\,k.\{0\}^{i_3} \rightarrow {}^2\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2} \parallel \{S\}^{i_1} \parallel \{0\}^{i_3} \downarrow_{i_1,i_2,i_3}$. Using ξ^- instead gives $n[\,k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2}\,] \parallel \{S\}^{i_1}\,] \parallel \mathsf{open}\,k.\{0\}^{i_3}$ which cannot reduce to reveal i_3 since there is no top-level k ambient. Finally, $n[\,k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2}\,] \parallel \{S\}^{i_1}\,] \rightarrow {}^2P'$ which does not contain instances of i_1, i_2.

(*ii*) Consider the possible ways of finding a redex in $\chi[\![P]\!] = k[\,P \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,]$, with the caveat that no part of $\{R\}^{i_2}$ and $\{S\}^{i_1}$ may be used, as this would destroy, respectively, i_2 and i_1 which need to be observed subsequently.

If the reduction is internal to P then the result is of the form $k[\,P^\dagger \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,]$. But $\xi^-[\![k[\,P^\dagger \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,]]\!] \rightarrow P^\dagger \parallel \{R\}^{i_2} \parallel n[\,\{S\}^{i_1}\,] \parallel \{0\}^{i_3} \downarrow i_2, i_3$. Alternatively, P may contain a top level ambient of the form $l[\,\mathcal{E}[\![\mathsf{out}\,k.P^\dagger]\!]\,]$, in that case the reactum is $l[\,\mathcal{E}[\![P^\dagger]\!]\,] \parallel k[\,P^\ddagger \parallel \{R\}^{i_2}\,] \parallel n[\,\{S\}^{i_1}\,]$ which again exposes the barb i_2 after interaction with ξ^-. The only remaining possibility is $P = \mathcal{E}[\![\mathsf{in}\,n.Q]\!]$ which implies that $P_1 \equiv n[\,k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_2}\,] \parallel \{S\}^{i_1}\,]$ which clearly has the correct behaviour wrt to ξ^+ and ξ^-. Also, the only possibility to hide i_1, i_2 with two reductions is to destroy them, hence $P' \equiv n[\,k[\,\mathcal{E}[\![Q]\!] \parallel R\,] \parallel S\,]$ and the correct labelled transition follows via Lemma 10. □

Lemma 25 (Contextually characterising out)**.** *Let:*

$$\chi \stackrel{\text{def}}{=} n[\,k[\,\mathsf{1}_{\mathsf{Pr}} \parallel \{R\}^{i_1}\,] \parallel \{S\}^{i_2}\,], \quad \xi \stackrel{\text{def}}{=} \mathsf{1}_{\mathsf{Pr}} \parallel \mathsf{open}\,k.\mathsf{open}\,n.\{0\}^{i_3}$$

$$\begin{aligned}\Phi(P,P') \stackrel{\text{def}}{=} \exists P_1,\, P_2,\, P_3.\, \chi[\![P]\!] \rightarrow P_1,\, \xi[\![P_1]\!] \rightarrow P_2,\, P_2 \rightarrow P_3,\\ P_2{\downarrow_{i_1}},\, P_3{\downarrow_{i_1,i_2,i_3}},\, P_1 \rightarrow {}^2P',\, \forall\theta.\, (i_1,i_2 \notin \theta) \rightarrow (\theta[P'] \not\downarrow_{i_1,i_2}).\end{aligned}$$

Then:

(i) if $P \xrightarrow{\mathsf{out}\,n\downarrow RkS} P'$ *then* $\Phi(P,P')$*;*

(ii) if $\Phi(P,P')$ *then* $\exists P'' \equiv P'$ *such that* $P \xrightarrow{\mathsf{out}\,n\downarrow RkS} P''$*.*

Proof. (*i*) if $P \xrightarrow{\mathsf{out}\,n\downarrow RkS} P'$ then $P = \mathcal{E}[\![\mathsf{out}\,n.Q]\!]$ and

$$P' \equiv (\lambda \mathsf{X}\mathsf{x}\mathsf{Y}.\, x[\,\mathcal{E}[\![Q]\!] \parallel \mathsf{X}\,] \parallel n[\,\mathsf{Y}\,])(RkS).$$

Then

$$\chi[\![P]\!] = n[\,k[\,\mathcal{E}[\![\mathsf{out}\,n.Q]\!] \parallel \{R\}^{i_1}\,] \parallel \{S\}^{i_2}\,] \\ \rightarrow k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1}\,] \parallel n[\,\{S\}^{i_2}\,].$$

Inserting the right-hand side into ξ yields

$$k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1}\,] \parallel n[\,\{S\}^{i_2}\,] \parallel \mathsf{open}\,k.\mathsf{open}\,n.\{0\}^{i_3} \\ \rightarrow \mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1} \parallel n[\,\{S\}^{i_2}\,] \parallel \mathsf{open}\,n.\{0\}^{i_3} \downarrow_{i_1} \\ \rightarrow \mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1} \parallel \{S\}^{i_2} \parallel \{0\}^{i_3} \downarrow_{i_1,i_2,i_3}.$$

Also, $k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1}\,] \parallel n[\,\{S\}^{i_2}\,] \rightarrow {}^2P'$

($\Leftarrow$) Consider the possible ways finding a redex in $\chi[\![P]\!] = n[\,k[\,P \parallel \{R\}^{i_1}\,] \parallel \{S\}^{i_2}\,]$. Note that R and S cannot take any part in the first interaction as this would involve the destruction of barbs i_1 and i_2. If the redex is entirely contained in P (ie $P \rightarrow P^\dagger$) then, after the reduction we have a term $n[\,k[\,P^\dagger \parallel \{R\}^{i_1}\,] \parallel \{S\}^{i_2}\,]$ which does not have k at top level and hence cannot interact with ξ. The second possibility is that P contains a top-level ambient of the form $l[\,\mathcal{E}[\![\mathsf{out}\,k.P^\dagger]\!]\,]$, in this case after the reduction we have $n[\,l[\,\mathcal{E}[\![P^\dagger]\!]\,] \parallel k[\,P^\ddagger \parallel \{R\}^{i_1}\,] \parallel \{S\}^{i_2}\,]$ that fails to interact with ξ for the same reason. The final possibility is that $P = \mathcal{E}[\![\mathsf{out}\,n.Q]\!]$ in which case $P_1 \equiv k[\,\mathcal{E}[\![Q]\!] \parallel \{R\}^{i_1}\,] \parallel n[\,\{S\}^{i_2}\,]$—the only choice that has the correct behaviour wrt ξ. The only two reductions which hide the barbs i_1 and i_2 from any context, therefore, are those which destroy them, hence $P' \equiv k[\,\mathcal{E}[\![Q]\!] \parallel R\,] \parallel n[\,S\,]$ whence the required transition follows from the Lemma 10. □

Lemma 26 (Contextually characterising [in] and [out])**.** *Let:*

$$\chi_{[\mathsf{in}\,n]\!\downarrow R} \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel n[\,\{R\}^{i_1}\,],\ \chi_{[\mathsf{out}\,n]\!\downarrow R} \stackrel{\text{def}}{=} n[\,\mathsf{1_{Pr}} \parallel \{R\}^{i_1}\,],\ \xi \stackrel{\text{def}}{=} \mathsf{1_{Pr}} \parallel \mathsf{open}\,n.\{0\}^{i_2}$$

$$\Phi_\alpha \stackrel{\text{def}}{=} \exists P_1,\, P_2.\, \chi_\alpha[\![P]\!] \rightarrow P_1 \not\downarrow_{i_1},\, \xi[\![P_1]\!] \rightarrow P_2 \downarrow_{i_1,i_2}, \\ P_1 \rightarrow P',\, \forall\theta.(i_1 \notin \theta) \rightarrow (\theta[\![P']\!] \not\downarrow_{i_1})$$

Then, for $\alpha \in \{[\mathsf{in}\,n]\!\downarrow R,\, [\mathsf{out}\,n]\!\downarrow R\}$ *we have:*

(i) if $P \xrightarrow{\alpha} P'$ *then* $\Phi_\alpha(P, P')$*;*

(ii) if $\Phi_\alpha(P, P')$ *then there exists* $P'' \equiv P'$ *such that* $P \xrightarrow{\alpha} P''$.

Proof. (*i*) $P \xrightarrow{[\mathsf{in}\, n]\!\downarrow R} P'$: if the derivation does not feature the use of a $\mathcal{HT}$-rule then Lemma 10 applies: for some m, $P = \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{in}\, n.Q]\!]\,]]\!]$ and

$$P' \equiv (\lambda \mathsf{Y}.\, \mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![Q]\!]\,] \parallel \mathsf{Y}\,]]\!])(R).$$

We have

$$\chi_{[\mathsf{in}\, n]\!\downarrow R}[\![\mathcal{E}[\![n[\,\mathcal{E}'[\![\mathsf{in}\, n.Q]\!]\,]]\!]\!] = \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{in}\, n.Q]\!]\,]]\!] \parallel n[\,\{R\}^{i_1}\,] \\ \rightarrow \mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![Q]\!]\,] \parallel \{R\}^{i_1}\,]]\!].$$

Plugging in ξ:

$$\mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![Q]\!]\,] \parallel \{R\}^{i_1}\,]]\!] \parallel \mathsf{open}\, n.\{0\}^{i_2} \rightarrow m[\,\mathcal{E}'[\![Q]\!]\,] \parallel \{R\}^{i_1} \parallel \{0\}^{i_2} \downarrow_{i_1,i_2}$$

and $\mathcal{E}[\![n[\,m[\,\mathcal{E}'[\![Q]\!]\,] \parallel \{R\}^{i_1}\,]]\!] \rightarrow P'$.

On the other hand, if ([AIN]) is used then $P \xrightarrow{\tau} Q$ and $P' = Q \parallel n[\,R\,]$. Hence, using Lemma 18, $\chi_{[\mathsf{in}\, n]\!\downarrow R}[\![P]\!] \rightarrow \chi_{[\mathsf{in}\, n]\!\downarrow R}[\![Q]\!] = Q \parallel n[\,\{R\}^{i_1}\,]$. Now $\xi[\![Q \parallel n[\,\{R\}^{i_1}\,]]\!] = Q \parallel n[\,\{R\}^{i_1}\,] \parallel \mathsf{open}\, n.\{0\}^{i_2} \rightarrow Q \parallel \{R\}^{i_1} \parallel \{0\}^{i_2} \downarrow_{i_1,i_2}$. Also, $Q \parallel n[\,\{R\}^{i_1}\,] \rightarrow P'$.

A similar calculation can be carried for the case $\alpha = [\mathsf{out}\, n]\!\downarrow R$.

(*ii*) Suppose that $\chi_{[\mathsf{in}\, n]\!\downarrow R}[\![P]\!] \rightarrow P_1$, $\xi[\![P_1]\!] \rightarrow P_2 \downarrow_{i_1,i_2}$, $P_1 \rightarrow P'$, and $\forall\theta.(i_1 \notin \theta) \rightarrow (\theta[P'] \not\downarrow_{i_1})$. Examining $\chi_{[\mathsf{in}\, n]\!\downarrow R}[\![P]\!] = P \parallel n[\,\{R\}^{i_1}\,]$, the only possible reductions that do not involve $\{R\}^{i_1}$ and that do not result in the barb i_1 at top level are: (1) an internal reduction in $P \rightarrow Q$ in which case $P_1 \equiv Q \parallel n[\,\{R\}^{i_1}\,]$; the only subsequent reduction that hides the i_1 ambient from any context is the reduction that destroys it, hence $P' \equiv Q \parallel n[\,R\,]$. But then there is $Q' \equiv Q$ such that $P \xrightarrow{\tau} Q'$ and so via an application of ([AIN]), $P \xrightarrow{[\mathsf{in}\, n]\!\downarrow R} P''$, where $P'' = Q' \parallel n[\,R\,] \equiv Q \parallel n[\,R\,] \equiv P'$. (2) $P \equiv \mathcal{E}[\![m[\,\mathcal{E}'[\![\mathsf{in}\, n.Q]\!]\,]]\!]$ and $P_1 \equiv n[\,\mathcal{E}[\![m[\,\mathcal{E}'[\![Q]\!]\,]]\!] \parallel \{R\}^{i_1}\,]$. Then we must have $P' \equiv n[\,\mathcal{E}[\![m[\,\mathcal{E}'[\![Q]\!]\,]]\!] \parallel R\,]$ and so there exists $P'' \equiv P'$ such that $P \xrightarrow{[\mathsf{in}\, n]\!\downarrow R} P''$.

Again, a similar calculation can be carried out for $\alpha = [\mathsf{out}\, n]\!\downarrow R$. □