Math. Struct. in Comp. Science (2003), *vol.* 13, *pp.* 371–408.
DOI: 10.1017/S0960129502003742 Printed in the United Kingdom

Equational properties of mobile ambients

ANDREW D. GORDON and LUCA CARDELLI

Microsoft Research Ltd., 7 J. J. Thompson Avenue, Cambridge CB3 0FB, United Kingdom

Received 29 October 1999; revised 20 December 2001

The ambient calculus is a process calculus for describing mobile computation. We develop a theory of Morris-style contextual equivalence for proving properties of mobile ambients. We prove a context lemma that allows derivation of contextual equivalences by considering contexts of a particular limited form, rather than all arbitrary contexts. We give an activity lemma that characterises the possible interactions between a process and a context. We prove several examples of contextual equivalence. The proofs depend on characterising reductions in the ambient calculus in terms of a labelled transition system.

1. Motivation

This paper develops tools for proving equations in the ambient calculus.

In earlier work (Cardelli and Gordon 2000b), we introduced the ambient calculus by adding *ambients* (mobile, hierarchical protection domains) to a framework for concurrency extracted from the π-calculus (Milner 1999; Sangiorgi and Walker 2001). The ambient calculus is an abstract model of mobile computation, including both mobile software agents and mobile hardware devices. The calculus models access control as well as mobility. For example, a process may move into or out of a particular ambient only if it possesses the appropriate capability.

This paper focuses on behavioural equivalence of mobile ambients. In particular, we study a form of Morris' contextual equivalence (Morris 1968) for ambients and develop some proof techniques. Our motivation is to prove a variety of equations. Some of these equations express and confirm some of the informal principles we had in mind when designing the calculus. As in other recent work (Abadi *et al.* 1998; Abadi and Gordon 1999), some of the equations establish security properties of systems modelled within the calculus.

The inclusion of primitives for mobility makes the theory of the ambient calculus more complex than that of its ancestor, the π-calculus. The main contribution of this paper is to demonstrate that some standard tools (a labelled transition system, a context lemma and an activity lemma) may be recast in the setting of the ambient calculus. Moreover, the paper introduces a new technique, based on what we call the hardening relation, for factoring the definition of the labelled transition system into a set of rules that identify the individual processes participating in a transition, and a set of rules that express how the participant processes interact.

We begin, in Section 2, by reviewing the syntax and reduction semantics of the ambient calculus. The semantics consists of a structural congruence relation $P \equiv Q$ (which says that P may be structurally rearranged to yield Q) and a reduction relation $P \rightarrow Q$ (which says that P may evolve in one step of computation to yield Q).

We introduce contextual equivalence $P \simeq Q$ in Section 3. We define a predicate, $P \Downarrow n$, which means intuitively that an observer may eventually detect an ambient named n at the top-level of the process P. Then we define $P \simeq Q$ to mean that, whenever P and Q are placed within an arbitrary context constructed from the syntax of the calculus, any observation made of P may also be made of Q, and *vice versa*. We give examples of pairs of processes that are equivalent and of pairs that are inequivalent.

In Section 4, we describe some techniques for proving contextual equivalence. We introduce a second operational semantics for the ambient calculus based on a hardening relation and a labelled transition system. The hardening relation identifies the subprocesses of a process that may participate in a computation step. We use the hardening relation both for defining the labelled transition system and for characterising whether an ambient of a particular name is present at the top-level of a process. Our first result, Theorem 9, asserts that the τ-labelled transition relation and the reduction relation are the same, up to structural congruence. So our two operational semantics are equivalent. The labelled transition system is useful for analysing the possible evolution of a process, since we may read off the possible labelled transitions of a process by inspecting its syntactic structure. Our second result, Theorem 12, is a context lemma that allows us to prove contextual equivalence by considering a limited set of contexts, known as harnesses, rather than all arbitrary contexts. A harness is a context with a single hole that is enclosed only within parallel compositions, restrictions and ambients. The third result of this section, Theorem 15, is an activity lemma that elaborates the ways in which a reduction may be derived when a process is inserted into a harness: either the process reduces by itself, or the harness reduces by itself, or there is an interaction between the harness and the process.

We exercise these proof techniques on examples in Section 5, and conclude in Section 6. Certain lemmas, propositions and theorems are stated without proof in the main text. Appendix A contains the omitted proofs.

Earlier versions of this article have appeared as a conference paper and as a technical report (Gordon and Cardelli 1999). The technical report includes some details omitted from proofs in Appendix A.

2. The ambient calculus (review)

We briefly describe the syntax and semantics of the calculus. We assume there are infinite sets of *names* and *variables*, ranged over by m, n, p, q and x, y, z, respectively. The syntax of the ambient calculus is based on categories of *expressions* and *processes*, ranged over by M, N and P, Q, R, respectively. The calculus inherits a core of concurrency primitives from the π-calculus: a restriction $(\nu n)P$ creates a fresh name n whose scope is P; a composition $P \mid Q$ behaves as P and Q running in parallel; a replication $!P$ behaves as unboundedly many replicas of P running in parallel; and the inactive process $\mathbf{0}$ does

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

nothing. We augment these π-calculus processes with primitives for mobility (ambients, $n[P]$ and the exercise of capabilities, $M.P$) and primitives for communication (input, $(x).P$ and asynchronous output, $\langle M \rangle$).

Here is an example process that illustrates the new primitives for mobility and communication:

$$m[p[out\ m.in\ n.\langle M \rangle]] \mid n[open\ p.(x).Q]$$

The effect of the mobility primitives in this example is to move the ambient p out of m and into n, and then to open it up. The input $(x).Q$ may then consume the output $\langle M \rangle$ to leave the residue $m[] \mid n[Q\{x \leftarrow M\}]$. We may regard the ambients m and n in this example as modelling two machines on a network, and the ambient p as modelling a packet sent from m to n. We will now describe the semantics of the new primitives in more detail.

An ambient $n[P]$ is a boundary, named n, around the process P. The boundary prevents direct interactions between P and any processes running in parallel with $n[P]$, but it does not prevent interactions within P. Ambients may be nested, so they induce a hierarchy. For example, in the process displayed above, the ambient named m is a parent of the ambient named p, and the ambients named m and n are siblings.

An action $M.P$ exercises the capabilities represented by M, and then behaves as P. The action either affects an enclosing ambient or one running in parallel. A capability is an expression derived from the name of an ambient. The three basic capabilities are *in* n, *out* n and *open* n. An action *in* $n.P$ moves its enclosing ambient into a sibling ambient named n. An action *out* $n.P$ moves its enclosing ambient out of its parent ambient, named n, to become a sibling of the former parent. An action *open* $n.P$ dissolves the boundary of an ambient $n[Q]$ running in parallel; the outcome is that the residue P of the action and the residue Q of the opened ambient run in parallel. In general, the expression M in $M.P$ may stand for a finite sequence of the basic capabilities, which are exercised one by one. Finite sequences are built up using concatenation, written $M.M'$. The empty sequence is written ϵ.

The final two process primitives allow communication of expressions. Expressions include names, variables and capabilities. An output $\langle M \rangle$ outputs the expression M. An input $(x).P$ blocks until it may consume an output running in parallel. Then it binds the expression being output to the variable x and runs P. In $(x).P$, the variable x is bound; its scope is P. Inputs and outputs are local to the enclosing ambient. Inputs and outputs may not interact directly through an ambient boundary. Hence we may think of there being an implicit input/output channel associated with each ambient.

We formally specify the syntax of the calculus as follows:

Expressions and Processes:

$M, N ::=$	expressions	$P, Q, R ::=$	processes
x	variable	$(\nu n)P$	restriction
n	name	$\mathbf{0}$	inactivity
in M	can enter M	$P \mid Q$	composition
out M	can exit M	$!P$	replication
open M	can open M	$M[P]$	ambient

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

ϵ	null	$M.P$	action
$M.M'$	path	$(x).P$	input
		$\langle M \rangle$	output

The general forms *in* M, *out* M and *open* M allow for the ambient to be an arbitrary capability M. The only useful cases are for M to be a name, or a variable that gets instantiated to a name. Similarly, the ambient syntax $M[P]$ allows M to be an arbitrary capability. The only useful case is for M to be a name, or a variable that gets instantiated to a name.

The following table defines the sets $fn(M)$ and $fv(M)$ of *free names* and *free variables* of a capability M, and the sets $fn(P)$ and $fv(P)$ of *free names* and *free variables* of a process P.

Free Names and Variables of Capabilities and Processes

$fn(x) = \emptyset$	$fv(x) = \{x\}$
$fn(n) = \{n\}$	$fv(n) = \emptyset$
$fn(in\ M) = fn(M)$	$fv(in\ M) = fv(M)$
$fn(out\ M) = fn(M)$	$fv(out\ M) = fv(M)$
$fn(open\ M) = fn(M)$	$fv(open\ M) = fv(M)$
$fn(\epsilon) = \emptyset$	$fv(\epsilon) = \emptyset$
$fn(M.M') = fn(M) \cup fn(M')$	$fv(M.M') = fv(M) \cup fv(M')$
$fn((\nu n)P) = fn(P) - \{n\}$	$fv((\nu n)P) = fv(P)$
$fn(\mathbf{0}) = \emptyset$	$fv(\mathbf{0}) = \emptyset$
$fn(P \mid Q) = fn(P) \cup fn(Q)$	$fv(P \mid Q) = fv(P) \cup fv(Q)$
$fn(!P) = fn(P)$	$fv(!P) = fv(P)$
$fn(M[P]) = fn(M) \cup fn(P)$	$fv(M[P]) = fv(M) \cup fv(P)$
$fn(M.P) = fn(M) \cup fn(P)$	$fv(M.P) = fv(M) \cup fv(P)$
$fn((x).P) = fn(P)$	$fv((x).P) = fv(P) - \{x\}$
$fn(\langle M \rangle) = fn(M)$	$fv(\langle M \rangle) = fv(M)$

In situations where a process is expected, we often just write M as a shorthand for the process $M.\mathbf{0}$. We also often just write $M[]$ as a shorthand for the process $M[\mathbf{0}]$. We write $(\nu \vec{p})P$ as a shorthand for $(\nu p_1) \cdots (\nu p_k)P$ where $\vec{p} = p_1, \ldots, p_k$.

If a phrase ϕ is an expression or a process, we write $\phi\{x \leftarrow M\}$ and $\phi\{n \leftarrow M\}$ for the outcomes of capture-avoiding substitutions of the expression M for each free occurrence of the variable x and the name n, respectively, in ϕ. We identify processes up to consistent renaming of bound names and variables. We say an expression M is *closed* if and only if $fv(M) = \emptyset$; similarly, a process P is *closed* if and only if $fv(P) = \emptyset$.

We formally define the operational semantics of ambient calculus in the chemical style, using structural congruence and reduction relations:

Structural Congruence: $P \equiv Q$

$P \mid Q \equiv Q \mid P$	$P \equiv P$
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	$Q \equiv P \Rightarrow P \equiv Q$
$!P \equiv P \mid !P$	$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$
$n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$
$n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P]$
$P \mid \mathbf{0} \equiv P$
$(\nu n)\mathbf{0} \equiv \mathbf{0}$
$!\mathbf{0} \equiv \mathbf{0}$
$\epsilon.P \equiv P$
$(M.M').P \equiv M.M'.P$

$P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$
$P \equiv Q \Rightarrow P \mid R \equiv Q \mid R$
$P \equiv Q \Rightarrow !P \equiv !Q$
$P \equiv Q \Rightarrow M[P] \equiv M[Q]$
$P \equiv Q \Rightarrow M.P \equiv M.Q$
$P \equiv Q \Rightarrow (x).P \equiv (x).Q$

Reduction: $P \rightarrow Q$

$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$
$m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$
$open\ n.P \mid n[Q] \rightarrow P \mid Q$
$\langle M \rangle \mid (x).P \rightarrow P\{x \leftarrow M\}$

$P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R$
$P \rightarrow Q \Rightarrow (\nu n)P \rightarrow (\nu n)Q$
$P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q]$
$P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$

For example, the process displayed earlier has the following reductions:

$$\begin{aligned} m[p[out\ m.in\ n.\langle M \rangle]] \mid n[open\ p.(x).P] &\rightarrow m[] \mid p[in\ n.\langle M \rangle] \mid n[open\ p.(x).P] \\ &\rightarrow m[] \mid n[p[\langle M \rangle] \mid open\ p.(x).P] \\ &\rightarrow m[] \mid n[\langle M \rangle \mid (x).P] \\ &\rightarrow m[] \mid n[P\{x \leftarrow M\}]. \end{aligned}$$

The syntax allows the formation of certain processes that may not participate in any reductions, such as the action $n.P$ and the ambient $(in\ n)[P]$. The presence of these nonsensical processes is harmless as far as the purposes of this paper are concerned, and they may be ruled out by a simple type system (Cardelli and Gordon 1999).

This concludes our brief review of the calculus. Earlier papers (Cardelli 1999; Cardelli and Gordon 2000b) explain in detail the motivation for our calculus, and give programming examples.

3. Contextual equivalence

Morris-style contextual equivalence (Morris 1968) is a standard way of saying that two processes have the same behaviour: two processes are contextually equivalent if and only if they admit the same elementary observations whenever they are inserted inside any arbitrary enclosing process. In the setting of the ambient calculus, we shall define contextual equivalence in terms of observing the presence, at the top-level of a process, of an ambient whose name is not restricted.

Let us say that a process P *exhibits a name* n just if P is a process with a top-level ambient named n, that is not restricted:

Exhibition of a Name: $P \downarrow n$

$P \downarrow n \quad \triangleq \quad$ there are $\vec{m}$, P', P'' with $n \notin \{\vec{m}\}$ and $P \equiv (\nu \vec{m})(n[P'] \mid P'')$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Let us say that a process P *converges to a name* n just if after some number of reductions, P exhibits n:

Convergence to a Name: $P \Downarrow n$

$$\frac{P \downarrow n}{P \Downarrow n}\ (\text{Conv Exh}) \qquad \frac{P \to Q \qquad Q \Downarrow n}{P \Downarrow n}\ (\text{Conv Red})$$

Next, let a *context*, $\mathscr{C}()$, be a process containing zero or more holes. We write a hole as $()$. We write $\mathscr{C}(P)$ for the outcome of filling each of the holes in the context $\mathscr{C}$ with the process P. Variables and names free in P may become bound in $\mathscr{C}(P)$. For example, if $P = n[\langle x \rangle]$ and $\mathscr{C}() = (\nu n)(x).()$, the variable x and the name n have become bound in $\mathscr{C}(P) = (\nu n)(x).n[\langle x \rangle]$. Hence, we do not identify contexts up to renaming of bound variables and names.

Now we can formally define contextual equivalence of processes:

Contextual Equivalence: $P \simeq Q$

$$P \simeq Q \quad \stackrel{\Delta}{=} \quad \text{for all } n, \mathscr{C}() \text{ with } \mathscr{C}(P), \mathscr{C}(Q) \text{ closed}, \mathscr{C}(P) \Downarrow n \Leftrightarrow \mathscr{C}(Q) \Downarrow n$$

This equivalence is a form of the may-testing equivalence studied in De Nicola and Hennessy (1984). De Nicola and Hennessy also study must-testing equivalence and the Egli–Milner equivalence; these also could be recast in the setting of the ambient calculus.

The following two propositions state some basic properties enjoyed by contextual equivalence. Let a relation $\mathscr{R}$ be a *precongruence* if and only if, for all P, Q and $\mathscr{C}()$, if $P \mathscr{R} Q$, then $\mathscr{C}(P) \mathscr{R} \mathscr{C}(Q)$. If, in addition, $\mathscr{R}$ is reflexive, symmetric and transitive, we say it is a *congruence*. For example, the structural congruence relation has these properties. Moreover, by a standard argument, so has contextual equivalence.

Proposition 1. Contextual equivalence is a congruence.

Structural congruence preserves exhibition of or convergence to a name, and hence is included in contextual equivalence.

Lemma 2. Suppose $P \equiv Q$. If $P \downarrow n$, then $Q \downarrow n$. Moreover, if $P \Downarrow n$, then $Q \Downarrow n$ with the same depth of inference.

Proof. For part (1), $P \downarrow n$, by definition, means that there are $\vec{m}$, P', P'' with $n \notin \{\vec{m}\}$ and $P \equiv (\nu \vec{m})(n[P'] \mid P'')$. Since $P \equiv Q$, we have $Q \equiv (\nu \vec{m})(n[P'] \mid P'')$, and hence $Q \downarrow n$. Part (2) follows by a case analysis of the derivation of $P \Downarrow n$. □

Proposition 3. If $P \equiv Q$, then $P \simeq Q$.

Proof. Consider any context $\mathscr{C}()$ and any name n, such that $\mathscr{C}(P) \Downarrow n$. Since $\equiv$ is a congruence, $P \equiv Q$ implies $\mathscr{C}(P) \equiv \mathscr{C}(Q)$. By Lemma 2, this and $\mathscr{C}(P) \Downarrow n$ imply $\mathscr{C}(Q) \Downarrow n$. Similarly, we can show that for all $\mathscr{C}$ and n, $\mathscr{C}(Q) \Downarrow n$ implies $\mathscr{C}(P) \Downarrow n$. Hence $P \simeq Q$. □

The following two examples illustrate that to show that two processes are contextually inequivalent, it suffices to find a context that distinguishes them.

Example 1. If $m \neq n$, then $m[] \not\simeq n[]$.

Proof. Consider the context $\mathscr{C}() = ()$. Since $\mathscr{C}(m[]) \equiv m[]$, we have $\mathscr{C}(m[]) \downarrow m$. By (Conv Exh) , we have $\mathscr{C}(m[]) \Downarrow m$. On the other hand, the process $n[]$ has no reductions, and does not exhibit m. Hence, we cannot derive $\mathscr{C}(n[]) \Downarrow m$. □

Example 2. If $m \neq n$, then $open\ m.\mathbf{0} \not\simeq open\ n.\mathbf{0}$.

Proof. Let $\mathscr{C}() = m[p[]] \mid ()$. Then $\mathscr{C}(open\ m.\mathbf{0}) \Downarrow p$ but not $\mathscr{C}(open\ n.\mathbf{0}) \Downarrow p$. □

On the other hand, it is harder to show that two processes are contextually equivalent, since one must consider their behaviour when placed in an arbitrary context. For example, consider the following contextual equivalence.

Example 3. $(\nu n)(n[] \mid open\ n.P) \simeq P$ if $n \notin fn(P)$.

The restriction of the name n in the process $(\nu n)(n[] \mid open\ n.P)$ implies that no context may interact with this process until it has reduced to P. Therefore, we would expect the equation to hold. But to prove this and other equations formally, we need some further techniques, which we develop in the next section. We will return to Example 3 in Section 5.

4. Tools for proving contextual equivalence

In this section we introduce some relations and theorems as tools that help prove contextual equivalence.

4.1. *A hardening relation*

In this subsection, we define a relation that explicitly identifies the top-level subprocesses of a process that may be involved in a reduction. This relation, the *hardening* relation, takes the form,

$$P > (\nu p_1, \ldots, p_k)\langle P' \rangle P''$$

where the phrase $(\nu p_1, \ldots, p_k)\langle P' \rangle P''$ is called a *concretion*. We say that P' is the *prime* of the concretion, and that P'' is the *residue* of concretion. Both P' and P'' lie in the scope of the restricted names $p_1, \ldots, p_k$. The intuition is that the process P, which may have many top-level subprocesses, may harden to a concretion that singles out a prime subprocess P', leaving behind the residue P''. By saying that P' has a top-level occurrence in P, we mean that P' is a subprocess of P not enclosed within any ambient boundaries. In Section 4.2, we use the hardening relation to define an operational semantics for the ambient calculus in terms of interactions between top-level occurrences of processes.

Concretions were introduced by Milner in the context of the π-calculus (Milner 1999). For the ambient calculus, we specify them as follows, where the prime of the concretion must be an action, an ambient, an input, or an output:

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Concretions:

$C, D ::=$	concretions
$(v\vec{p})\langle M.P\rangle Q$	action, $M \in \{in\ n, out\ n, open\ n\}$
$(v\vec{p})\langle n[P]\rangle Q$	ambient
$(v\vec{p})\langle (x).P\rangle Q$	input
$(v\vec{p})\langle\langle M\rangle\rangle Q$	output

The order of the bound names $p_1, \ldots, p_k$ in a concretion $(vp_1, \ldots, p_k)\langle P'\rangle P''$ does not matter and they may be renamed consistently. When $k = 0$, we may write the concretion as $(v)\langle P'\rangle P''$.

We now introduce the basic ideas of the hardening relation informally. If P is an action *in* $n.Q$, *out* $n.Q$, *open* $n.Q$, an ambient $n[Q]$, an input $(x).Q$, or an output $\langle M\rangle$, then P hardens to $(v)\langle P\rangle\mathbf{0}$. Consider two processes P and Q. If either of these hardens to a concretion, then their composition $P \mid Q$ may harden to the same concretion, but with the other process included in the residue of the concretion. For example, if $P > (v)\langle P_1\rangle P_2$, then $P \mid Q > (v)\langle P_1\rangle(P_2 \mid Q)$. If a process P hardens to a concretion, then the replication $!P$ may harden to the same concretion, but with $!P$ included in the residue of the concretion – a replication is not consumed by hardening. Finally, if a process P hardens to a concretion C, then the restriction $(vn)P$ hardens to a concretion written $\overline{(vn)}C$, which is the same as C but with the restriction (vn) included either in the list of bound names, the prime or the residue of C. We define $\overline{(vn)}C$ by:

Restricting a Concretion: $\overline{(vn)}C$ **where** $C = (v\vec{p})\langle P_1\rangle P_2$ **and** $n \notin \{\vec{p}\}$

(1) If $n \in fn(P_1)$ then:

(a) If $P_1 = m[P_1']$, $m \neq n$, $n \notin fn(P_2)$, let $\overline{(vn)}C \stackrel{\Delta}{=} (v\vec{p})\langle m[(vn)P_1']\rangle P_2$.

(b) Otherwise, let $\overline{(vn)}C \stackrel{\Delta}{=} (vn, \vec{p})\langle P_1\rangle P_2$.

(2) If $n \notin fn(P_1)$ let $\overline{(vn)}C \stackrel{\Delta}{=} (v\vec{p})\langle P_1\rangle(vn)P_2$.

Next, we define the hardening relation by the following:

Hardening: $P > C$

(Harden Action) $\dfrac{M \in \{in\ n, out\ n, open\ n\}}{M.P > (v)\langle M.P\rangle\mathbf{0}}$ (Harden ϵ) $\dfrac{P > C}{\epsilon.P > C}$ (Harden .) $\dfrac{M.(N.P) > C}{(M.N).P > C}$

(Harden Amb) $\dfrac{}{n[P] > (v)\langle n[P]\rangle\mathbf{0}}$ (Harden Input) $\dfrac{}{(x).P > (v)\langle (x).P\rangle\mathbf{0}}$ (Harden Output) $\dfrac{}{\langle M\rangle > (v)\langle\langle M\rangle\rangle\mathbf{0}}$

(Harden Par 1) (for $\{\vec{p}\} \cap fn(Q) = \emptyset$) $\dfrac{P > (v\vec{p})\langle P'\rangle P''}{P \mid Q > (v\vec{p})\langle P'\rangle(P'' \mid Q)}$

(Harden Par 2) (for $\{\vec{q}\} \cap fn(P) = \emptyset$) $\dfrac{Q > (v\vec{q})\langle Q'\rangle Q''}{P \mid Q > (v\vec{q})\langle Q'\rangle(P \mid Q'')}$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

$$\frac{P > (\nu\vec{p})\langle P'\rangle P''}{!P > (\nu\vec{p})\langle P'\rangle(P'' \mid !P)}\ (\text{Harden Repl}) \qquad \frac{P > C}{(\nu n)P > \overline{(\nu n)}C}\ (\text{Harden Res})$$

For example, the process $P = (\nu p)(\nu q)(n[p[]] \mid q[])$ may harden in two ways:

$$\begin{aligned} P &> (\nu)\langle n[(\nu p)p[]]\rangle(\nu q)(\mathbf{0} \mid q[]) \\ P &> (\nu q)\langle q[]\rangle(\nu p)(n[p[]] \mid \mathbf{0}) \end{aligned}$$

The following lemma gives a basic property of hardening.

Lemma 4. If $P > (\nu\vec{p})\langle P'\rangle P''$, then $\{\vec{p}\} \subseteq fn(P')$ and the names $\vec{p}$ are pairwise distinct.

Proof. The proof is by induction on the derivation of $P > (\nu\vec{p})\langle P'\rangle P''$. □

The next two results relate hardening and structural congruence.

Lemma 5. If $P > (\nu\vec{p})\langle P'\rangle P''$, then $P \equiv (\nu\vec{p})(P' \mid P'')$.

Proposition 6. If $P \equiv Q$ and $Q > (\nu\vec{r})\langle Q'\rangle Q''$, there are P' and P'' with $P > (\nu\vec{r})\langle P'\rangle P''$, $P' \equiv Q'$, and $P'' \equiv Q''$.

These results follow from inductions on the derivations of $P > (\nu\vec{p})\langle P'\rangle P''$ and $P \equiv Q$, respectively. Using them, we may characterise the exhibition of a name independently of structural congruence.

Proposition 7. $P \downarrow n$ if and only if there are $\vec{p}$, P', P'' such that $P > (\nu\vec{p})\langle n[P']\rangle P''$ and $n \notin \{\vec{p}\}$.

Now, we can show that the hardening relation is image-finite.

Lemma 8. For all P, $\{C : P > C\}$ is finite.

Proof. The proof is by induction on the structure of P. □

The proof suggests a procedure for enumerating the set $\{C : P > C\}$. Given Proposition 7, it follows that the predicate $P \downarrow n$ is decidable.

4.2. *A labelled transition system*

The labelled transition system presented in this section allows for an analysis of the possible reductions from a process P in terms of the syntactic structure of P. The definition of the reduction relation does not directly support such an analysis, because of the rule $P' \equiv P, P \to Q, Q \equiv Q' \Rightarrow P' \to Q'$, which allows for arbitrary structural rearrangements of a process during the derivation of a reduction.

We define a family of transition relations $P \xrightarrow{\alpha} Q$, indexed by a set of labels, ranged over by α, which is given in the following table:

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Labels:

$\alpha ::=$	label
τ	internal step
in n	enter ambient n
out n	exit ambient n
open n	dissolve ambient n

An M-transition $P \xrightarrow{M} Q$ means that the process P has a top-level process exercising the capability M; these transitions are defined by the rule (Trans Cap) below. A τ-transition $P \xrightarrow{\tau} Q$ means that P evolves in one step to Q; these transitions are defined by the other rules below.

Labelled Transitions: $P \xrightarrow{\alpha} P'$

(Trans Cap)

$$\frac{P > (\nu\vec{p})\langle M.P'\rangle P'' \quad fn(M) \cap \{\vec{p}\} = \emptyset}{P \xrightarrow{M} (\nu\vec{p})(P' \mid P'')}$$

(Trans Amb)

$$\frac{P > (\nu\vec{p})\langle n[Q]\rangle P' \quad Q \xrightarrow{\tau} Q'}{P \xrightarrow{\tau} (\nu\vec{p})(n[Q'] \mid P')}$$

(Trans In) (where $\{\vec{r}\} \cap fn(n[Q]) = \emptyset$ and $\{\vec{r}\} \cap \{\vec{p}\} = \emptyset$)

$$\frac{P > (\nu\vec{p})\langle n[Q]\rangle R \quad Q \xrightarrow{in\ m} Q' \quad R > (\nu\vec{r})\langle m[R']\rangle R''}{P \xrightarrow{\tau} (\nu\vec{p},\vec{r})(m[n[Q'] \mid R'] \mid R'')}$$

(Trans Out) (where $n \notin \{\vec{q}\}$)

$$\frac{P > (\nu\vec{p})\langle n[Q]\rangle P' \quad Q > (\nu\vec{q})\langle m[R]\rangle Q' \quad R \xrightarrow{out\ n} R'}{P \xrightarrow{\tau} (\nu\vec{p})((\nu\vec{q})(m[R'] \mid n[Q']) \mid P')}$$

(Trans Open)

$$\frac{P > (\nu\vec{p})\langle n[Q]\rangle P' \quad P' \xrightarrow{open\ n} P''}{P \xrightarrow{\tau} (\nu\vec{p})(Q \mid P'')}$$

(Trans I/O) (where $\{\vec{q}\} \cap fn(\langle M\rangle) = \emptyset$)

$$\frac{P > (\nu\vec{p})\langle\langle M\rangle\rangle P' \quad P' > (\nu\vec{q})\langle(x).P''\rangle P'''}{P \xrightarrow{\tau} (\nu\vec{p},\vec{q})(P''\{x \leftarrow M\} \mid P''')}$$

The rules (Trans In), (Trans Out) and (Trans Open) derive a τ-transition from an M-transition. We introduced the M-transitions to simplify the statement of these three rules. (Trans I/O) allows for exchange of messages. (Trans Amb) is a congruence rule for τ-transitions within ambients.

Given its definition in terms of the hardening relation, we can analyse the transitions

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

derivable from any process by inspection of its syntactic structure. This allows a structural analysis of the possible reductions from a process, since the τ-transition relation corresponds to the reduction relation as in the following theorem, where $P \stackrel{\tau}{\longrightarrow}\equiv Q$ means there is R with $P \stackrel{\tau}{\longrightarrow} R$ and $R \equiv Q$.

Theorem 9. $P \to Q$ if and only if $P \stackrel{\tau}{\longrightarrow}\equiv Q$.

As corollaries of Lemma 8 and Theorem 9, we get that the transition system is image-finite, and that the reduction relation is image-finite up to structural congruence.

Lemma 10. For all P and α, the set $\{R : P \stackrel{\alpha}{\longrightarrow} R\}$ is finite.

Proof. The proof is by induction on the depth of inference of $P \stackrel{\alpha}{\longrightarrow} R$, with appeal to Lemma 8, one can see that the set $\{R : P \stackrel{\alpha}{\longrightarrow} R\}$ is finite. □

Lemma 11. For all P, the set $\{\{R : Q \equiv R\} : P \to Q\}$ is finite.

Proof. By Lemma 10, the set $\{Q : P \stackrel{\tau}{\longrightarrow} Q\}$ is finite. Therefore, the set $\{\{R : Q \equiv R\} : P \stackrel{\tau}{\longrightarrow} Q\}$ is finite. But, by Theorem 9 and the transitivity of structural congruence, this set is the same as $\{\{R : Q \equiv R\} : P \to Q\}$. □

4.3. *A context lemma*

The context lemma presented in this section is a tool for proving contextual equivalence by considering only a limited set of contexts, rather than all contexts. Many context lemmas have been proved for a wide range of calculi, starting with Milner's context lemma for the combinatory logic form of PCF (Milner 1977).

Our context lemma is stated in terms of a notion of a *harness*:

Harnesses:

$H ::=$	harnesses
$-$	process variable
$(\nu n)H$	restriction
$P \mid H$	left composition
$H \mid Q$	right composition
$n[H]$	ambient

Harnesses are analogous to the evaluation contexts found in context lemmas for some other calculi. Unlike the contexts of Section 3, harnesses are identified up to consistent renaming of bound names. We let $fn(H)$ and $fv(H)$ be the sets of names and variables, respectively, occurring free in a harness H. There is exactly one occurrence of the process variable $-$ in any harness. If H is a harness, we write $H\{P\}$ for the outcome of substituting the process P for the single occurrence of the process variable $-$. Names restricted in H are renamed to avoid capture of free names of P. For example, if $H = (\nu n)(- \mid \mathit{open}\ n)$, then $H\{n[]\} = (\nu n')(n[] \mid \mathit{open}\ n')$ for some $n' \neq n$. Similarly, if H and H' are harnesses, we write $H\{H'\}$ for the harness obtained by substituting H' for the process variable $-$ in H.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Let a *substitution*, σ, be a list $x_1 \leftarrow M_1, \ldots, x_k \leftarrow M_k$, where the variables $x_1, \ldots, x_k$ are pairwise distinct, and $fv(M_i) = \emptyset$ for each $i \in 1..k$. Let $dom(\sigma) = \{x_1, \ldots, x_k\}$. Let $P\sigma$ be the process $P\{x_1 \leftarrow M_1\} \cdots \{x_k \leftarrow M_k\}$. Let a harness be *closed* if and only if it has no free variables (though it may have free names).

Here is our context lemma.

Theorem 12 (Context). For all processes P and Q, $P \simeq Q$ if and only if for all substitutions σ with $dom(\sigma) = fv(P) \cup fv(Q)$, and for all closed harnesses H and names n we have $H\{P\sigma\} \Downarrow n \Leftrightarrow H\{Q\sigma\} \Downarrow n$.

A corollary is that for all closed processes P and Q, we have $P \simeq Q$ if and only if for all closed harnesses H and names n we have $H\{P\} \Downarrow n \Leftrightarrow H\{Q\} \Downarrow n$.

In general, however, we need to consider the arbitrary closing substitution σ when using Theorem 12. This is because a variable free in a process may become bound to an expression once the process is placed in a context. For example, let $P = x[n[]] \mid open\, y.\mathbf{0}$ and $Q = \mathbf{0}$. Consider the context $\mathscr{C}() = \langle m, m\rangle \mid (x, y).()$. We have $\mathscr{C}(P) \Downarrow n$ but not $\mathscr{C}(Q) \Downarrow n$. So P and Q are not contextually equivalent, though they do satisfy $H\{P\} \Downarrow n \Leftrightarrow H\{Q\} \Downarrow n$ for all closed H and n.

Some process calculi enjoy stronger context lemmas. Let processes P and Q be *parallel testing equivalent* if and only if for all processes R and names n we have $P \mid R \Downarrow n \Leftrightarrow Q \mid R \Downarrow n$. We might hope to show that any two closed processes are contextually equivalent if and only if they are parallel testing equivalent. This would be a stronger result than Theorem 12 because it would avoid considering contexts that include ambients. Such a result is true for CCS (De Nicola and Hennessy 1984), for example, but it is false for the ambient calculus. To see this, let $P = out\, p.\mathbf{0}$ and $Q = \mathbf{0}$. We can show that $P \mid R \Downarrow n \Leftrightarrow Q \mid R \Downarrow n$ for all n and R. Now, consider the context $\mathscr{C}() = p[m[()]]$. We have $\mathscr{C}(P) \Downarrow m$ but not $\mathscr{C}(\mathbf{0}) \Downarrow m$. So P and Q are parallel testing equivalent but not contextually equivalent.

4.4. *An activity lemma*

When we come to apply Theorem 12 we need to analyse judgments of the form $H\{P\} \downarrow n$ or $H\{P\} \rightarrow Q$. In this section we formalise these analyses.

We begin by extending the structural congruence, hardening and reduction relations to harnesses as follows:

— Let $H \equiv H'$ hold if and only if $H\{P\} \equiv H'\{P\}$ for all P.
— Let $H > (v\vec{p})\langle n[H']\rangle Q$ hold if and only if $H\{P\} > (v\vec{p})\langle n[H'\{P\}]\rangle Q$ for all P such that $\{\vec{p}\} \cap fn(P) = \emptyset$.
— Let $H > (v\vec{p})\langle Q\rangle H'$ hold if and only if $H\{P\} > (v\vec{p})\langle Q\rangle(H'\{P\})$ for all P such that $\{\vec{p}\} \cap fn(P) = \emptyset$.
— Let $H \rightarrow H'$ hold if and only if, for all P, $H\{P\} \rightarrow H'\{P\}$.

We need the following lemma about hardening.

Lemma 13. If $H\{P\} > (v\vec{p})\langle P_1\rangle P_2$, then either:

(1) $H > (v\vec{p})\langle n[H']\rangle P_2$ and $P_1 = n[H'\{P\}]$, or

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(2) $H > (\nu\vec{p})\langle P_1\rangle H'$ and $P_2 = H'\{P\}$, or
(3) $P > (\nu\vec{p})\langle P_1\rangle P'$, $H \equiv - \mid R$, $P_2 \equiv P' \mid R$, and $\{\vec{p}\} \cap fn(R) = \emptyset$.

Intuitively, there are two ways in which $H\{P\} \downarrow n$ can arise: either the process P exhibits the name by itself, or the harness H exhibits the name n by itself. Proposition 14 formalises this analysis. Similarly, there are three ways in which a reduction $H\{P\} \rightarrow Q$ may arise: either (1) the process P reduces by itself, or (2) the harness H reduces by itself, or (3) there is an interaction between the process and the harness. Theorem 15 formalises this analysis. Such a result is sometimes known as an activity lemma (Plotkin 1977).

Proposition 14. If $H\{P\} \downarrow n$, then either (1) $H\{Q\} \downarrow n$ for all Q, or (2) both $P \downarrow n$ and also for all Q, $Q \downarrow n$ implies that $H\{Q\} \downarrow n$.

Proof. By Proposition 7, $H\{P\} \downarrow n$ means there are $\vec{p}$, P', P'' such that $H\{P\} > (\nu\vec{p})\langle n[P']\rangle P''$ with $n \notin \{\vec{p}\}$. Hence, the proposition follows from Lemma 13. □

Theorem 15 (Activity). $H\{P\} \rightarrow R$ if and only if:

(Act Proc) there is a reduction $P \rightarrow P'$ with $R \equiv H\{P'\}$, or
(Act Har) there is a reduction $H \rightarrow H'$ with $R \equiv H'\{P\}$, or
(Act Inter) there are H' and $\vec{r}$ with $\{\vec{r}\} \cap fn(P) = \emptyset$, and one of the following holds:

(Inter In) $H \equiv (\nu\vec{r})H'\{m[- \mid R'] \mid n[R'']\}$, $P \xrightarrow{in\ n} P'$,
and $R \equiv (\nu\vec{r})H'\{n[m[P' \mid R'] \mid R'']\}$

(Inter Out) $H \equiv (\nu\vec{r})H'\{n[m[- \mid R'] \mid R'']\}$, $P \xrightarrow{out\ n} P'$,
and $R \equiv (\nu\vec{r})H'\{m[P' \mid R'] \mid n[R'']\}$

(Inter Open) $H \equiv (\nu\vec{r})H'\{- \mid n[R']\}$, $P \xrightarrow{open\ n} P'$,
and $R \equiv (\nu\vec{r})H'\{P' \mid R'\}$

(Inter Input) $H \equiv (\nu\vec{r})H'\{- \mid \langle M\rangle\}$, $P > (\nu\vec{p})\langle(x).P'\rangle P''$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(P'\{x\leftarrow M\} \mid P'')\}$, with $\{\vec{p}\} \cap fn(M) = \emptyset$

(Inter Output) $H \equiv (\nu\vec{r})H'\{- \mid (x).R'\}$, $P > (\nu\vec{p})\langle\langle M\rangle\rangle P'$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(P' \mid R'\{x\leftarrow M\})\}$, with $\{\vec{p}\} \cap fn(R') = \emptyset$

(Inter Amb) $P > (\nu\vec{p})\langle n[Q]\rangle P'$ and one of the following holds:

(1) $Q \xrightarrow{in\ m} Q'$, $H \equiv (\nu\vec{r})H'\{- \mid m[R']\}$, $\{\vec{p}\} \cap fn(m[R']) = \emptyset$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(P' \mid m[n[Q'] \mid R'])\}$

(2) $Q \xrightarrow{out\ m} Q'$, $H \equiv (\nu\vec{r})H'\{m[- \mid R']\}$, $\{\vec{p}\} \cap fn(m[R']) = \emptyset$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(n[Q'] \mid m[P' \mid R'])\}$

(3) $H \equiv (\nu\vec{r})H'\{m[R' \mid in\ n.R''] \mid -\}$, $\{\vec{p}\} \cap fn(m[R' \mid in\ n.R'']) = \emptyset$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(n[Q \mid m[R' \mid R'']] \mid P')\}$

(4) $H \equiv (\nu\vec{r})H'\{- \mid open\ n.R'\}$, $n \notin \{\vec{p}\}$,
and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(Q \mid P') \mid R'\}$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

5. Examples of contextual equivalence

In this section we give three examples to demonstrate how we can apply Theorem 12 and Theorem 15 to establish contextual equivalence.

5.1. *Opening an ambient*

First, we return to and prove Example 3 from Section 3.

Lemma 16. If $H\{(\nu n)(n[] \mid \mathit{open}\ n.P)\} \Downarrow m$ and $n \notin \mathit{fn}(P)$, then $H\{P\} \Downarrow m$.

Proof. The proof is by induction on the derivation of $H\{(\nu n)(n[] \mid \mathit{open}\ n.P)\} \Downarrow m$:

(Conv Exh) Here $H\{(\nu n)(n[] \mid \mathit{open}\ n.P)\} \downarrow m$. By Proposition 14, either (1), for all Q, $H\{Q\} \downarrow m$, or (2), $(\nu n)(n[] \mid \mathit{open}\ n.P) \downarrow m$. In case (1), we have, in particular, that $H\{P\} \downarrow m$. Hence, $H\{P\} \Downarrow m$, by (Conv Exh). Case (2) cannot arise, since, by Proposition 7, $(\nu n)(n[] \mid \mathit{open}\ n.P) \downarrow m$ implies that $(\nu n)(n[] \mid \mathit{open}\ n.P) > (\nu \vec{p})\langle m[P']\rangle P''$ with $m \notin \{\vec{p}\}$. But the only hardenings of the process $(\nu n)(n[] \mid \mathit{open}\ n.P)$ are

$$(\nu n)(n[] \mid \mathit{open}\ n.P) > (\nu n)\langle n[]\rangle(\mathbf{0} \mid \mathit{open}\ n.P)$$
$$(\nu n)(n[] \mid \mathit{open}\ n.P) > (\nu n)\langle \mathit{open}\ n.P\rangle(n[] \mid \mathbf{0}).$$

So case (2) is impossible.

(Conv Red) Here $H\{(\nu n)(n[] \mid \mathit{open}\ n.P)\} \to R$ and $R \Downarrow m$. By Theorem 15, one of three cases pertains:

(Act Proc) Then $(\nu n)(n[] \mid \mathit{open}\ n.P) \to P'$ with $R \equiv H\{P'\}$. By inspection of the rules of the labelled transition system, it must be that (Trans Open) derives this transition, with $P' \equiv P$. Therefore $R \Downarrow m$ implies that $H\{P\} \Downarrow m$.

(Act Har) Then $H \to H'$ with $R \equiv H'\{(\nu n)(n[] \mid \mathit{open}\ n.P)\}$. By Lemma 2, we may derive $H'\{(\nu n)(n[] \mid \mathit{open}\ n.P)\} \Downarrow m$ by the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $H'\{P\} \Downarrow m$. From $H \to H'$ we obtain $H\{P\} \to H'\{P\}$ in particular. By (Act Har), we get $H\{P\} \Downarrow m$.

(Act Inter) Then there is an interaction between the process $(\nu n)(n[] \mid \mathit{open}\ n.P)$ and the harness H. Given the possible hardenings of $(\nu n)(n[] \mid \mathit{open}\ n.P)$ stated above, there are no transitions derivable from $(\nu n)(n[] \mid \mathit{open}\ n.P)$, so none of (Inter In), (Inter Out) or (Inter Open) is applicable. Similarly, neither (Inter Input) nor (Inter Output) is applicable. Given $(\nu n)(n[] \mid \mathit{open}\ n.P) > (\nu n)\langle n[]\rangle(\mathbf{0} \mid \mathit{open}\ n.P)$, clause (Inter Amb) might be applicable. Points (1) and (2) are impossible, because $\mathbf{0}$ has no transitions, and points (3) and (4) are impossible because n is restricted. We conclude that none of the possibilities stated in clause (Act Inter) of Theorem 15 pertains. So this case is impossible. □

Proof of Example 3. $(\nu n)(n[] \mid \mathit{open}\ n.P) \simeq P$ if $n \notin \mathit{fn}(P)$.

Proof. By Theorem 12, it suffices to prove $H\{((\nu n)(n[] \mid \mathit{open}\ n.P))\sigma\} \Downarrow m \Leftrightarrow H\{P\sigma\} \Downarrow m$ for all closed harnesses H and names m and for all substitutions σ with $\mathit{dom}(\sigma) = \mathit{fv}(P)$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Since the name n is bound, we may assume that $n \notin fn(\sigma(x))$ for all $x \in dom(\sigma)$. Therefore, we have to prove that $H\{(\nu n)(n[] \mid open\ n.P\sigma\} \Downarrow m \Leftrightarrow H\{P\sigma\} \Downarrow m$ where $n \notin fn(P\sigma)$.

We will prove each direction separately. First, suppose that $H\{P\sigma\} \Downarrow m$. Then, since $(\nu n)(n[] \mid open\ n.P\sigma) \rightarrow P\sigma$, we get $H\{(\nu n)(n[] \mid open\ n.P\sigma)\} \rightarrow H\{P\sigma\}$. By (Conv Red), we get $H\{(\nu n)(n[] \mid open\ n.P\sigma)\} \Downarrow m$. Second, suppose that $H\{(\nu n)(n[] \mid open\ n.P\sigma)\} \Downarrow m$. By Lemma 16, we get $H\{P\sigma\} \Downarrow m$. □

5.2. *The perfect firewall equation*

Consider a process $(\nu n)n[P]$, where n is not free in P. Since the name n is known neither inside the ambient $n[P]$, nor outside it, the ambient $n[P]$ is a 'perfect firewall' that neither allows another ambient to enter nor to exit. The following two lemmas allow us to prove that $(\nu n)n[P]$ is contextually equivalent to $\mathbf{0}$, when $n \notin fn(P)$, which is to say that no context can detect the presence of $(\nu n)n[P]$.

Lemma 17. If $H\{(\nu n)n[P]\} \Downarrow m$ and $n \notin fn(P)$, then $H\{\mathbf{0}\} \Downarrow m$.

Proof. The proof is by induction on the derivation of $H\{(\nu n)n[P]\} \Downarrow m$:

(Conv Exh) Here $H\{(\nu n)n[P]\} \downarrow m$. By Proposition 14, either (1), for all Q, $H\{Q\} \downarrow m$, or (2), $(\nu n)n[P] \downarrow m$. In case (1), we have, in particular, that $H\{\mathbf{0}\} \downarrow m$. Hence, $H\{\mathbf{0}\} \Downarrow m$, by (Conv Exh). Case (2) cannot arise, since, by Proposition 7, $(\nu n)n[P] \downarrow m$ implies that $(\nu n)n[P] > (\nu \vec{p})\langle m[P']\rangle P''$ with $m \notin \{\vec{p}\}$, which is impossible.

(Conv Red) Here $H\{(\nu n)n[P]\} \rightarrow R$ and $R \Downarrow m$. By Theorem 15, one of three cases pertains:

(Act Proc) Then $(\nu n)n[P] \rightarrow P''$ with $R \equiv H\{P''\}$. By Theorem 9, there is Q with $(\nu n)n[P] \xrightarrow{\tau} Q$ and $Q \equiv P''$. Since $(\nu n)n[P] > (\nu n)\langle n[P]\rangle\mathbf{0}$ is the only hardening derivable from $(\nu n)n[P]$, and since $n \notin fn(P)$, the transition $(\nu n)n[P] \xrightarrow{\tau} Q$ can only be derived using (Trans Amb), with $P \xrightarrow{\tau} P'$ and $Q = (\nu n)(n[P'] \mid \mathbf{0})$. Therefore, there is a reduction $P \rightarrow P'$ and $P'' \equiv (\nu n)n[P']$. By Lemma 21 stated in the Appendix, $P \rightarrow P'$ implies $fn(P') \subseteq fn(P)$ and so $n \notin fn(P')$. We have $R \equiv H\{(\nu n)n[P']\}$ with $n \notin fn(P')$. By Lemma 2, we may derive $H\{(\nu n)n[P']\} \Downarrow m$ by the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $H\{\mathbf{0}\} \Downarrow m$.

(Act Har) Then $H \rightarrow H'$ with $R \equiv H'\{(\nu n)n[P]\}$. By Lemma 2, we may derive $H'\{(\nu n)n[P]\} \Downarrow m$ by the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $H'\{\mathbf{0}\} \Downarrow m$. From $H \rightarrow H'$ we obtain $H\{\mathbf{0}\} \rightarrow H'\{\mathbf{0}\}$ in particular. By (Conv Red), we get $H\{\mathbf{0}\} \Downarrow m$.

(Act Inter) Then there are H' and $\vec{r}$ with $\{\vec{r}\} \cap fn(P) = \varnothing$ and one of several conditions must hold. Since the only hardening or transition from $(\nu n)n[P]$ is $(\nu n)n[P] > (\nu n)\langle n[P]\rangle\mathbf{0}$, only the rule (Inter Amb) applies. According to Theorem 15, there are four possibilities to consider.

(1) Here, $P \xrightarrow{in\ m} P'$, $H \equiv (\nu \vec{r})H'\{- \mid m[R']\}$, $\{n\} \cap fn(m[R']) = \varnothing$ and $R \equiv (\nu \vec{r})H'\{(\nu n)(\mathbf{0} \mid m[n[P'] \mid R'])\}$. We have $R \equiv (\nu \vec{r})H'\{m[R' \mid (\nu n)n[P']]\}$. By Lemma 23 (stated in the Appendix), $n \notin fn(P)$ and $P \xrightarrow{in\ m} P'$ imply $n \notin fn(P')$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

By Lemma 2, we get $(\nu\vec{r})H'\{m[R' \mid (\nu n)n[P']]\} \Downarrow m$ with the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $(\nu\vec{r})H'\{m[R' \mid \mathbf{0}]\} \Downarrow m$. Moreover, $H\{\mathbf{0}\} \equiv (\nu\vec{r})H'\{m[R' \mid \mathbf{0}]\}$, and therefore $H\{\mathbf{0}\} \Downarrow m$.

(2) Here, $P \xrightarrow{out\ m} P'$, $H \equiv (\nu\vec{r})H'\{m[- \mid R']\}$, $R \equiv (\nu\vec{r})H'\{(\nu n)(n[P'] \mid m[\mathbf{0} \mid R'])\}$, with $m \notin \{n\}$. Since n is bound, we may assume $n \notin fn(H) \cup \{\vec{r}\}$, so that $n \notin fn(R')$, and hence we can derive that $R \equiv (\nu\vec{r})H'\{m[R'] \mid (\nu n)n[P']\}$. By Lemma 23, $n \notin fn(P)$ and $P \xrightarrow{out\ m} P'$ imply $n \notin fn(P')$. By Lemma 2, we get $(\nu\vec{r})H'\{m[R'] \mid (\nu n)n[P']\} \Downarrow m$ with the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $(\nu\vec{r})H'\{m[R'] \mid \mathbf{0}\} \Downarrow m$. Moreover, $H\{\mathbf{0}\} \equiv (\nu\vec{r})H'\{m[R'] \mid \mathbf{0}\}$, and therefore $H\{\mathbf{0}\} \Downarrow m$.

The other possibilities, (3) and (4), are ruled out because the name n is restricted in the concretion $(\nu n)\langle n[P]\rangle\mathbf{0}$. □

Lemma 18. If $H\{\mathbf{0}\} \Downarrow m$, then $H\{P\} \Downarrow m$.

Proof. The proof is by induction on the derivation of $H\{\mathbf{0}\} \Downarrow m$:

(Conv Exh) Here $H\{\mathbf{0}\} \downarrow m$. By Proposition 14, either (1), for all Q, $H\{Q\} \downarrow m$, or (2), $\mathbf{0} \downarrow m$. Case (2) is impossible. In case (1), we get, in particular, that $H\{P\} \downarrow m$. Hence, $H\{P\} \Downarrow m$.

(Conv Red) Here $H\{\mathbf{0}\} \to Q$ and $Q \Downarrow m$. By Theorem 15, and the fact that $\mathbf{0}$ has no reductions and no hardenings, it must be that $H \to H'$ with $Q \equiv H'\{\mathbf{0}\}$. By Lemma 2, we get that $H'\{\mathbf{0}\} \Downarrow m$ is derivable with the same depth of inference as $Q \Downarrow m$. By the induction hypothesis, $H'\{P\} \Downarrow m$. From $H \to H'$ we get that $H\{P\} \to H'\{P\}$. By (Conv Red), $H\{P\} \to H'\{P\}$ and $H'\{P\} \Downarrow m$ imply $H\{P\} \Downarrow m$. □

Using these two lemmas we get the following example.

Example 4. If $n \notin fn(P)$, then $(\nu n)n[P] \simeq \mathbf{0}$.

Proof. By Theorem 12, it suffices to prove that

$$H\{((\nu n)n[P])\sigma\} \Downarrow m \Leftrightarrow H\{\mathbf{0}\sigma\} \Downarrow m$$

for all closed harnesses H and names m and for all substitutions σ with $dom(\sigma) = fv((\nu n)n[P])$. Since the name n is bound, we may assume that $n \notin fn(\sigma(x))$ for any $x \in dom(\sigma)$. Therefore, we are to prove that

$$H\{(\nu n)n[P\sigma]\} \Downarrow m \Leftrightarrow H\{\mathbf{0}\} \Downarrow m,$$

where $n \notin fn(P\sigma)$. This follows from Lemma 17 and Lemma 18. □

Our first proof of this equation (which was stated in an earlier paper (Cardelli and Gordon 2000b)) was by a direct quantification over all contexts. The proof above using the context lemma is simpler.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

5.3. *Crossing a firewall*

This example concerns an agent that crosses a firewall using previously arranged passwords. We explained this example, but did not give a proof, in an earlier paper (Cardelli and Gordon 2000b).

Lemma 19. Suppose that $(fn(P) \cup fn(Q)) \cap \{k, k', k''\} = \emptyset$ and $w \notin fn(Q)$. Consider the processes defined by

$$\begin{aligned}
R_1 &\triangleq (\nu k\, k'\, k'')(k'[open\ k.k''[Q]] \mid \\
&\qquad (\nu w)w[k[out\ w.in\ k'.in\ w] \mid open\ k'.open\ k''.P]) \\
R_2 &\triangleq (\nu k\, k'\, k''\, w)(k'[open\ k.k''[Q]] \mid k[in\ k'.in\ w] \mid w[open\ k'.open\ k''.P]) \\
R_3 &\triangleq (\nu k\, k'\, k''\, w)(k'[k[in\ w] \mid open\ k.k''[Q]] \mid w[open\ k'.open\ k''.P]) \\
R_4 &\triangleq (\nu k\, k'\, k''\, w)(k'[in\ w \mid k''[Q]] \mid w[open\ k'.open\ k''.P]) \\
R_5 &\triangleq (\nu k\, k'\, k''\, w)w[k'[k''[Q]] \mid open\ k'.open\ k''.P] \\
R_6 &\triangleq (\nu k\, k'\, k''\, w)w[k''[Q] \mid open\ k''.P] \\
R_7 &\triangleq (\nu w)w[Q \mid P].
\end{aligned}$$

For each $i \in 1..6$, $R_i \simeq R_{i+1}$.

Proof. Suppose that $i \in 1..6$. Without loss of generality, we may assume that the processes P and Q are closed, and hence that all the R_i are closed. By Theorem 12, we need to show for all H and m that $H\{R_i\} \Downarrow m \Leftrightarrow H\{R_{i+1}\} \Downarrow m$. We may calculate that $R_i \rightarrow R_{i+1}$, for each i. It follows that $H\{R_{i+1}\} \Downarrow m$ implies $H\{R_i\} \Downarrow m$.

We now prove that $H\{R_i\} \Downarrow m$ implies $H\{R_{i+1}\} \Downarrow m$ by induction on the derivation of $H\{R_i\} \Downarrow m$.

(Conv Exh) Here $H\{R_i\} \downarrow m$. By Proposition 14, either (1), for all Q, $H\{Q\} \downarrow m$, or (2), $R_i \downarrow m$. In case (1), we have, in particular, that $H\{R_{i+1}\} \downarrow m$. Hence, $H\{R_{i+1}\} \Downarrow m$, by (Conv Exh). Case (2) cannot arise, because of the outermost restrictions on each R_i.

(Conv Red) Here $H\{R_i\} \rightarrow R$ and $R \downarrow m$. By Theorem 9 and Theorem 15, one of three cases pertains:

- **(Act Proc)** Then $R_i \rightarrow R'$ with $R \equiv H\{R'\}$. By inspection of the definitions of R_i and the labelled transition system, it must be that $R' \equiv R_{i+1}$. Therefore $R \Downarrow m$ implies that $H\{R_{i+1}\} \Downarrow m$.
- **(Act Har)** Then $H \rightarrow H'$ with $R \equiv H'\{R_i\}$. By Lemma 2, we may derive $H'\{R_i\} \Downarrow m$ by the same depth of inference as $R \Downarrow m$. By the induction hypothesis, $H'\{R_{i+1}\} \Downarrow m$. From $H \rightarrow H'$ we obtain $H\{R_{i+1}\} \rightarrow H'\{R_{i+1}\}$ in particular. By (Conv Red), we get $H\{R_{i+1}\} \Downarrow m$.
- **(Act Inter)** Then there is an interaction between the process R_i and the harness H'. Given that $fn(Q) \cap \{k', k'', w\} = \emptyset$, none of the conditions stated in the rule (Act Inter) of Theorem 15 applies. Therefore this case is impossible.

This completes the proof by induction. □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Example 5. Let us define

$$\begin{array}{rcl} \mathit{Firewall} & \triangleq & (\nu w)w[k[\mathit{out}\ w.\mathit{in}\ k'.\mathit{in}\ w] \mid \mathit{open}\ k'.\mathit{open}\ k''.P] \\ \mathit{Agent} & \triangleq & k'[\mathit{open}\ k.k''[Q]]. \end{array}$$

If $(fn(P) \cup fn(Q)) \cap \{k, k', k''\} = \emptyset$ and $w \notin fn(Q)$, then

$$(\nu k\, k'\, k'')(\mathit{Agent} \mid \mathit{Firewall}) \quad \simeq \quad (\nu w)w[Q \mid P].$$

Proof. Recall the processes R_1 and R_7 from Lemma 19. By that lemma, $R_1 \simeq R_7$. This is exactly the desired equation, since $R_1 = (\nu k k' k'')(\mathit{Agent} \mid \mathit{Firewall})$ and $R_7 = (\nu w)w[Q \mid P]$. □

6. Conclusions

We have developed a theory of Morris-style contextual equivalence for the ambient calculus. We have shown that standard tools such as a labelled transition system, a context lemma, and an activity lemma, may be adapted to the ambient calculus. We have then introduced a new technique, based on a hardening relation, for defining the labelled transition system. We employed these tools to prove equational properties of mobile ambients.

We have adapted the concretions of Milner (1999) to highlight those subprocesses of a process that may participate in a computation. This is an alternative to the membranes and airlocks of the chemical abstract machine of Berry and Boudol (1992). Unlike these authors, in the definition of our transition relation we use the hardening relation, rather than the full structural congruence relation, to choose subprocesses to participate in a transition. In applications of the activity lemma, Theorem 15, and in other situations, our proof techniques depend on analysing the possible hardenings and the possible transitions of processes by examining their structure. This is possible because, unlike structural congruence, the hardening relation is not transitive. Therefore, the use of hardening rather than structural congruence in the definition of the transition relation is essential for the techniques we advocate here.

Our use of the hardening relation to define the transition relation for the ambient calculus is similar to the use by Vitek and Castagna (1999) of a heating relation to define reduction in their Seal calculus. A difference in style is that Vitek and Castagna use structural congruence as well as the heating relation to define their reduction relation.

Since the work presented in this paper was completed, several authors have advanced the study of labelled transition systems and bisimulation for ambient calculi.

— Vigliotti (1999) studies labelled transition systems for the ambient calculus, but not bisimulation. She proves a completeness result relating reductions and labelled transitions, which is akin to Theorem 9 of this paper.
— Fournet *et al.* (2000) describes the first distributed implementation of mobile ambients, based on a translation to the join-calculus (Fournet and Gonthier 1996). They verify its correctness by adapting the technique of barbed coupled simulations.
— Although in this paper we have developed some novel tools for proving equational

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

properties, we have found it difficult to state a very rich collection of equational properties. Levi and Sangiorgi (Levi and Sangiorgi 2000) attribute this difficulty to certain interferences between overlapping redexes in the original ambient calculus. To enable such interferences to be avoided, they add co-capabilities to obtain a calculus of Safe Ambients (SA). With this addition, they state and prove a richer set of equational properties than seems to be possible for the unmodified calculus.

— Sangiorgi (2001) studies the equivalence induced by the ambient logic (Cardelli and Gordon 2000a) on an ambient calculus without replication or restriction. He characterises this equivalence as the bisimulation induced by a certain labelled transition system, and shows that the equivalence is closely related to structural congruence.

— Merro and Hennessy (Merro and Hennessy 2002) were the first to study the bisimulation induced by a labelled transition system for an ambient calculus with replication and restriction. They work with a calculus of Safe Ambients with Passwords (SAP), which has co-capabilities like the SA calculus but synchronisation is additionally contingent on shared knowledge of a secret password. They define a barbed congruence for the SAP calculus, and show that it equals the bisimulation induced by their labelled transition system. Hence, they obtain a convenient co-inductive proof technique for equational reasoning about ambients.

Appendix A. Proofs

In this appendix, we prove all the propositions stated without proof in the main body of the paper. To do so, we need several auxiliary results.

The appendix consists of several sections.

(1) In Section A.1 we prove the statement in Section 3 that contextual equivalence is a congruence.
(2) In Section A.2, we prove three important facts about the hardening relation that were stated in Section 4.1, *viz*. Lemma 5 and Propositions 6 and Proposition 7.
(3) Section A.3 contains some auxiliary results and a proof of Theorem 9 from Section 4.2, which states that the reduction and τ-transition relations are the same up to structural congruence.
(4) In Section A.4 we prove the activity lemma, Theorem 15, stated in Section 4.4.
(5) In Section A.5, we prove some auxiliary results about replication.
(6) Section A.6 is devoted to proving our context lemma, Theorem 12, which was stated in Section 4.3.

In the main body of the paper, we stated Theorem 12 before Theorem 15, but in fact we use Theorem 15 in the proof of Theorem 12. Therefore, we will give the proof of Theorem 15 before the proof of Theorem 12.

Throughout this appendix, we shall refer to the rules of structural congruence and reduction using the names in the following tables:

Structural Congruence: $P \equiv Q$

$P \equiv P$	(Struct Refl)
$Q \equiv P \Rightarrow P \equiv Q$	(Struct Symm)

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$	(Struct Trans)
$P \equiv Q \Rightarrow (\nu n)P \equiv (\nu n)Q$	(Struct Res)
$P \equiv Q \Rightarrow P \mid R \equiv Q \mid R$	(Struct Par)
$P \equiv Q \Rightarrow !P \equiv !Q$	(Struct Repl)
$P \equiv Q \Rightarrow M[P] \equiv M[Q]$	(Struct Amb)
$P \equiv Q \Rightarrow M.P \equiv M.Q$	(Struct Action)
$P \equiv Q \Rightarrow (x).P \equiv (x).Q$	(Struct Input)
$P \mid Q \equiv Q \mid P$	(Struct Par Comm)
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	(Struct Par Assoc)
$!P \equiv P \mid !P$	(Struct Repl Par)
$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$	(Struct Res Res)
$n \notin fn(P) \Rightarrow (\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$	(Struct Res Par)
$n \neq m \Rightarrow (\nu n)m[P] \equiv m[(\nu n)P]$	(Struct Res Amb)
$P \mid \mathbf{0} \equiv P$	(Struct Zero Par)
$(\nu n)\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Res)
$!\mathbf{0} \equiv \mathbf{0}$	(Struct Zero Repl)
$\epsilon.P \equiv P$	(Struct ϵ)
$(M.M').P \equiv M.M'.P$	(Struct .)

Reduction: $P \rightarrow Q$

$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$	(Red In)
$m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$	(Red Out)
$open\ n.P \mid n[Q] \rightarrow P \mid Q$	(Red Open)
$\langle M \rangle \mid (x).P \rightarrow P\{x \leftarrow M\}$	(Red I/O)
$P \rightarrow Q \Rightarrow P \mid R \rightarrow Q \mid R$	(Red Par)
$P \rightarrow Q \Rightarrow (\nu n)P \rightarrow (\nu n)Q$	(Red Res)
$P \rightarrow Q \Rightarrow n[P] \rightarrow n[Q]$	(Red Amb)
$P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$	(Red $\equiv$)

Many of the proofs in the rest of the appendix depend on the following basic facts about structural congruence, reduction, hardening, and the transition relation.

Lemma 20. If $P \equiv Q$, then $fn(P) = fn(Q)$ and $fv(P) = fv(Q)$.

Lemma 21. If $P \rightarrow Q$, then $fn(P) \subseteq fn(Q)$ and $fv(P) \subseteq fv(Q)$.

Lemma 22. If $P > C$, then $fn(P) = fn(C)$ and $fv(P) = fv(C)$.

Lemma 23. If $P \xrightarrow{\alpha} P'$, then $fn(\alpha) \cup fn(P') \subseteq fn(P)$, $fv(\alpha) = \emptyset$, and $fv(P') \subseteq fv(P)$.

Lemma 24. If $n \notin fn(P)$, then $(\nu n)P \equiv P$.

Proof. Using the axioms (Struct Zero Par), (Struct Res Par) and (Struct Zero Res), we get $(\nu n)P \equiv (\nu n)(P \mid \mathbf{0}) \equiv P \mid (\nu n)\mathbf{0} \equiv P \mid \mathbf{0} \equiv P$. □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

A.1. *Proof omitted from Section 3*

Apart from proving transitivity, the proof that contextual equivalence is a congruence is easy.

Proof of Proposition 1. Contextual equivalence is a congruence.

Proof. Reflexivity and symmetry are trivial.

For transitivity, suppose that $P \simeq P'$ and $P' \simeq P''$. To show that $P \simeq P''$, consider any context $\mathscr{C}()$ and any name n such that $\mathscr{C}(P)$ and $\mathscr{C}(P'')$ are closed. It need not be the case that $\mathscr{C}(P')$ is closed. Suppose that $\{x_1, \ldots, x_k\} = fv(\mathscr{C}(P'))$, and suppose that $m_1, \ldots, m_k$ are fresh names. We define a new family of contexts $\mathscr{D}_0, \mathscr{D}_1, \ldots, \mathscr{D}_k$ by induction: $\mathscr{D}_0 = \mathscr{C}$ and $\mathscr{D}_{i+1} = \langle m_{i+1} \rangle \mid (x_{i+1}).\mathscr{D}_i$. The context $\mathscr{D}_k$ has two useful properties. First, for all Q and q,

$$\mathscr{D}_k(Q) \Downarrow q \Leftrightarrow \mathscr{C}(Q)\{x_1 \leftarrow m_1\} \cdots \{x_k \leftarrow m_k\} \Downarrow q.$$

Second, $\mathscr{D}_k(P')$ is closed. Now, suppose that $\mathscr{C}(P) \Downarrow n$. Since $\mathscr{C}(P)$ is closed, $\mathscr{C}(P) = \mathscr{C}(P)\{x_1 \leftarrow m_1\} \cdots \{x_k \leftarrow m_k\}$. Hence, by the first property of $\mathscr{D}_k$, we have $\mathscr{D}_k(P) \Downarrow n$. By the second property of $\mathscr{D}_k$, and $P \simeq P'$, we have $\mathscr{D}_k(P') \Downarrow n$. Since $\mathscr{C}(P'')$ is closed, it follows that $\mathscr{D}_k(P'')$ is closed too. Therefore, $P' \simeq P''$ implies that $\mathscr{D}_k(P'') \Downarrow n$. Since $\mathscr{C}(P'')$ is closed, $\mathscr{C}(P'') = \mathscr{C}(P'')\{x_1 \leftarrow m_1\} \cdots \{x_k \leftarrow m_k\}$. Hence, by the first property of $\mathscr{D}_k$, we have $\mathscr{C}(P'') \Downarrow n$. A symmetric argument establishes that $\mathscr{C}(P'') \Downarrow n$ implies $\mathscr{C}(P) \Downarrow n$. Hence $P \simeq P''$.

For precongruence, consider any P, Q and $\mathscr{C}()$ with $P \simeq Q$. To show that $\mathscr{C}(P) \simeq \mathscr{C}(Q)$, consider any context $\mathscr{D}()$ and any name n with $\mathscr{D}(\mathscr{C}(P)) \Downarrow n$. Since $\mathscr{D}(\mathscr{C}())$ is a context, $P \simeq Q$ implies $\mathscr{D}(\mathscr{C}(Q)) \Downarrow n$. Similarly, $\mathscr{D}(\mathscr{C}(Q)) \Downarrow n$ implies $\mathscr{D}(\mathscr{C}(P)) \Downarrow n$. It follows that $\mathscr{C}(P) \simeq \mathscr{C}(Q)$. □

A.2. *Proofs omitted from Section 4.1*

This section provides proofs of Lemma 5 and Propositions 6 and 7. The main lemma of the section, Lemma 31, asserts that the hardening relation preserves structural congruence. To state and prove it, we need three auxiliary definitions.

The first auxiliary definition is a relation $P \mathrel{\hat{\equiv}} Q$ on primes, where a *prime* is an ambient $m[P]$, an action $M.P$ where $M \in \{in\ n, out\ n, open\ n\}$, an input $(x).P$ or an output $\langle M \rangle$. The relation $P \equiv Q$ is the least to satisfy the following rules:

Structural Congruence of Primes: $P \mathrel{\hat{\equiv}} Q$

$n[P] \mathrel{\hat{\equiv}} n[Q]$ if $P \equiv Q$
$M.P \mathrel{\hat{\equiv}} M.Q$ if $M \in \{in\ n, out\ n, open\ n\}$ and $P \equiv Q$
$(x).P \mathrel{\hat{\equiv}} (x).Q$ if $P \equiv Q$
$\langle M \rangle \mathrel{\hat{\equiv}} \langle N \rangle$ if $M = N$

This relation is clearly reflexive, symmetric and transitive, and implies structural congruence.

Lemma 25. For all primes P, Q, R:

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(1) $P \triangleq P$.
(2) If $P \triangleq Q$, then $Q \triangleq P$.
(3) If $P \triangleq Q$ and $Q \triangleq R$, then $P \triangleq R$.
(4) If $P \triangleq Q$, then $P \equiv Q$.

We will prove the converse of part (4) at the end of this section.

The second auxiliary definition is a relation $C \equiv D$ on concretions:

Structural Congruence of Concretions: $C \equiv D$

$$C \equiv D \quad \stackrel{\Delta}{=} \quad C = (v\vec{r})\langle P\rangle P',\ D = (v\vec{r})\langle Q\rangle Q',\ P \triangleq Q \text{ and } P' \equiv Q'.$$

Lemma 26. If $C \equiv D$, then $\overline{(vn)}C \equiv \overline{(vn)}D$.

Proof. From $C \equiv D$, it follows that $C = (v\vec{r})\langle P\rangle P'$, $D = (v\vec{r})\langle Q\rangle Q'$, $P \triangleq Q$ and $P' \equiv Q'$. Now, either $n \in fn(P)$ or not. First, suppose $n \in fn(P)$.

— If $P = m[P'']$, $m \neq n$ and $n \notin fn(P')$, then $\overline{(vn)}C = (v\vec{r})\langle m[(vn)P'']\rangle P'$. Since $P \triangleq Q$, it follows that $Q = m[Q'']$ with $P'' \equiv Q''$. By Lemma 20, $n \notin fn(P')$ implies $n \notin fn(Q')$. Therefore, $\overline{(vn)}D = (v\vec{r})\langle m[(vn)Q'']\rangle Q'$, so $\overline{(vn)}C \equiv \overline{(vn)}D$.
— Otherwise, $\overline{(vn)}C = (vn,\vec{r})\langle P\rangle P'$, and $\overline{(vn)}D = (vn,\vec{r})\langle Q\rangle Q'$.

Second, if $n \notin fn(P)$, we have $n \notin fn(Q)$ by Lemma 20. Thus, $\overline{(vn)}C = (v\vec{r})\langle P\rangle (vn)P'$ and $\overline{(vn)}D = (v\vec{r})\langle Q\rangle (vn)Q'$. □

By a similar analysis, we can prove the following lemma.

Lemma 27. $\overline{(vm)(vn)}C \equiv \overline{(vn)(vm)}C$.

The third auxiliary definition is a relation $M > N$ on expressions, defined by the following rules:

Auxiliary Relation on Expressions: $M > N$

$M > M.\epsilon$	if $M \in \{in\ n, out\ n, open\ n\}$
$\epsilon > \epsilon$	
$M.N > M_1.(M_2.N)$	if $M > M_1.M_2$
$M.N > N'$	if $M > \epsilon$ and $N > N'$

Lemma 28. If $M.P > C$, then either:
(1) $M > M_1.M_2$, $C = (v)\langle M_1.R\rangle \mathbf{0}$, and $R \equiv M_2.P$, or
(2) $M > \epsilon$ and $P > C$.

Proof. The proof is by induction on the derivation of $M.P > C$. □

Lemma 29. If $M > \epsilon$ and $P > C$, then $M.P > C$.

Proof. The proof is by induction on the derivation of $M > \epsilon$. □

Lemma 30. If $M > M_1.M_2$, then $M.P > (v)\langle M_1.P'\rangle \mathbf{0}$ with $P' \equiv M_2.P$.

Proof. The proof is by induction on the derivation of $M > M_1.M_2$. □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Next, we prove the main lemma of the section.

Lemma 31. If $P \equiv Q$ and $Q > D$, there is C with $P > C$ and $C \equiv D$.

Proof. We show by induction on the derivation of $P \equiv Q$, that $P \equiv Q$ implies:

(1) Whenever $P > C$ there is D with $Q > D$ and $C \equiv D$;
(2) Whenever $Q > D$ there is C with $P > C$ and $C \equiv D$.

We proceed by a case analysis of the rule that derives $P \equiv Q$:

(Struct Refl) In this case $P = Q$. So parts (1) and (2) are trivial.

(Struct Symm) In this case $Q \equiv P$. Part (1) follows from part (2) of the induction hypothesis, and part (2) follows from part (1) of the induction hypothesis.

(Struct Trans) In this case $P \equiv R$ and $R \equiv Q$. For (1), suppose $P > (\nu\vec{r})\langle P_1\rangle P_2$. By the induction hypothesis, $R > (\nu\vec{r})\langle R_1\rangle R_2$ with $P_1 \mathrel{\hat{\equiv}} R_1$ and $P_2 \equiv R_2$. By the induction hypothesis, again, $Q > (\nu\vec{r})\langle Q_1\rangle Q_2$ with $R_1 \mathrel{\hat{\equiv}} Q_1$ and $R_2 \equiv Q_2$. By transitivity, $P_1 \mathrel{\hat{\equiv}} Q_1$ and $P_2 \equiv Q_2$. Part (2) follows by a symmetric argument.

(Struct Res) In this case $P = (\nu n)P'$, $Q = (\nu n)Q'$ and $P' \equiv Q'$. For (1), suppose $(\nu n)P' > C$. This can only be derived using (Harden Res), so $P' > C'$ with $C = \overline{(\nu n)}C'$. By the induction hypothesis, $Q' > D'$ with $C' \equiv D'$. By (Harden Res), $Q = (\nu n)Q' > \overline{(\nu n)}D'$. By Lemma 26, $\overline{(\nu n)}C' \equiv \overline{(\nu n)}D'$. Part (2) follows by a symmetric argument.

(Struct Par) In this case $P = P' \mid R$, $Q = Q' \mid R$ and $P' \equiv Q'$. For (1), suppose $P' \mid R > (\nu\vec{r})\langle P_1\rangle P_2$. This judgment must be derived from one of the following rules:

- **(Harden Par 1)** Here $P' > (\nu\vec{r})\langle P_1\rangle P_2'$ with $P_2 = P_2' \mid R$ and $\{\vec{r}\} \cap fn(R) = \emptyset$. By the induction hypothesis, $Q' > (\nu\vec{r})\langle Q_1\rangle Q_2'$ with $P_1 \mathrel{\hat{\equiv}} Q_1$ and $P_2' \equiv Q_2'$. Let $Q_2 = Q_2' \mid R$. By (Harden Par 1), $Q = Q' \mid R > (\nu\vec{r})\langle Q_1\rangle Q_2$. Moreover, $P_2' \mid R \equiv Q_2' \mid R$, that is, $P_2 \equiv Q_2$.
- **(Harden Par 2)** Here $R > (\nu\vec{r})\langle P_1\rangle P_2'$ with $P_2 = P' \mid P_2'$ and $\{\vec{r}\} \cap fn(P') = \emptyset$. By Lemma 20, $fn(P') = fn(Q')$, so $\{\vec{r}\} \cap fn(Q') = \emptyset$. Let $Q_2 = Q' \mid P_2'$. By (Harden Par 2), $Q' \mid R > (\nu\vec{r})\langle P_1\rangle Q_2$. Moreover, $P' \mid P_2' \equiv Q' \mid P_2'$, that is, $P_2 \equiv Q_2$.

Part (2) follows by a symmetric argument.

We omit the other cases. □

Proof of Proposition 6. If $P \equiv Q$ and $Q > (\nu\vec{r})\langle Q'\rangle Q''$, there are P' and P'' with $P > (\nu\vec{r})\langle P'\rangle P''$, $P' \equiv Q'$, and $P'' \equiv Q''$.

Proof. The proof follows by combining Lemmas 25 and 31. □

Proof of Proposition 7. $P \downarrow n$ if and only if there exist $\vec{p}$, P', P'' such that $P > (\nu\vec{p})\langle n[P']\rangle P''$ and $n \notin \{\vec{p}\}$.

Proof. First, suppose $P \downarrow n$, that is, there are $\vec{p}$, R', R'' with $n \notin \{\vec{p}\}$ and $P \equiv R$ where $R = (\nu\vec{p})(n[R'] \mid R'')$. Given (Struct Res Amb) and (Struct Res Par), we may assume that $\{\vec{p}\} \subseteq fn(R') \cap fn(R'')$. Therefore, we may derive $R > (\nu\vec{p})\langle n[R']\rangle(\mathbf{0} \mid R'')$. By Lemma 31, $P \equiv R$ implies there are P', P'' such that $P > (\nu\vec{p})\langle n[P']\rangle P''$, $P' \equiv R'$ and $P'' \equiv R''$.

Second, suppose $P > (\nu\vec{p})\langle n[P']\rangle P''$ and $n \notin \{\vec{p}\}$. By Lemma 5, $P \equiv (\nu\vec{p})(n[P'] \mid P'')$. Therefore, $P \downarrow n$. □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

We end this section by exploring another consequence of Lemma 31.

Proposition 32. For all primes P and Q, if $P \equiv Q$, then $P \mathrel{\hat{=}} Q$.

Proof. Since P and Q are primes, their only hardenings are $P > (\nu)\langle P\rangle\mathbf{0}$ and $Q > (\nu)\langle Q\rangle\mathbf{0}$. Then, by Lemma 31, $P \mathrel{\hat{=}} Q$. □

A corollary of Lemma 25 and Proposition 32 is that for all primes P and Q, $P \equiv Q$ if and only if $P \mathrel{\hat{=}} Q$. For example, it follows that $m[P] \equiv n[Q]$ if and only if $m = n$ and $P \equiv Q$.

A.3. *Proofs omitted from Section 4.2*

This section provides a proof of Theorem 9, that $P \to Q$ if and only if there is R with $P \xrightarrow{\tau} R$ and $R \equiv Q$. We prove each direction separately, starting with the right-to-left implication.

First, we need the following lemma.

Lemma 33. If $P \xrightarrow{M} P'$, then $P \equiv (\nu\vec{p})(P_1 \mid M.P_2)$ with $P' \equiv (\nu\vec{p})(P_1 \mid P_2)$ and $fn(M) \cap \{\vec{p}\} = \emptyset$.

Proof. Only (Trans Cap) may derive the judgment $P \xrightarrow{M} P'$. So we have $P > (\nu\vec{p})\langle M.P_1\rangle P_2$, $P' = (\nu\vec{p})(P_1 \mid P_2)$, $M \in \{in\ n, out\ n, open\ n\}$ and $n \notin \{\vec{p}\}$. By Proposition 5, $P \equiv (\nu\vec{p})(M.P_1 \mid P_2)$. Moreover, $fn(M) = \{n\}$, so the result follows. □

We use the following to establish the right-to-left direction of Theorem 9.

Proposition 34. If $P \xrightarrow{\tau} P'$, then $P \to P'$.

Proof. The proof is by induction on the derivation of $P \xrightarrow{\tau} P'$. We examine one case:

(Trans In) In this case we have $P > (\nu\vec{p})\langle n[Q]\rangle R$, $Q \xrightarrow{in\ m} Q'$, $R > (\nu\vec{r})\langle m[R']\rangle R''$ and $P' = (\nu\vec{p},\vec{r})(m[n[Q'] \mid R'] \mid R'')$ with $\{\vec{r}\} \cap fn(n[Q]) = \emptyset$. By Lemma 5, $P \equiv (\nu\vec{p})(n[Q] \mid R)$. By Lemma 33, $Q \equiv (\nu\vec{q})(Q_1 \mid in\ m.Q_2)$, with $Q' \equiv (\nu\vec{q})(Q_1 \mid Q_2)$ and $n \notin \{\vec{q}\}$. Since the names $\vec{q}$ are bound, we may assume that $\{\vec{q}\} \cap fn(m[R']) = \emptyset$. By Lemma 5, $R \equiv (\nu\vec{r})(m[R'] \mid R'')$. Hence, we have

$$\begin{aligned} P &\equiv (\nu\vec{p})(n[(\nu\vec{q})(Q_1 \mid in\ n.Q_2)] \mid (\nu\vec{r})(m[R'] \mid R'')) \\ &\equiv (\nu\vec{p},\vec{r})((\nu\vec{q})(n[Q_1 \mid in\ n.Q_2] \mid m[R']) \mid R'') \\ &\to (\nu\vec{p},\vec{r})((\nu\vec{q})(m[n[Q_1 \mid Q_2] \mid R']) \mid R'') \\ &\equiv (\nu\vec{p},\vec{r})(m[n[Q'] \mid R'] \mid R'') \\ &= P'. \end{aligned}$$

The other cases follow similarly. □

Next, we prove a couple of lemmas needed for proving the left-to-right direction of Theorem 9.

Lemma 35. If $P \equiv Q$ and $Q \xrightarrow{\alpha} Q'$, there is P' such that $P \xrightarrow{\alpha} P'$ and $P' \equiv Q'$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Proof. The proof is by induction on the derivation of $Q \xrightarrow{\alpha} Q'$:

(Trans Cap) We have $Q > (v\vec{r})\langle M.Q_1\rangle Q_2$, $Q' = (v\vec{r})(Q_1 \mid Q_2)$ and $M \in \{in\ n, out\ n, open\ n\}$, with $n \notin \{\vec{r}\}$. By Proposition 6, there are P_1 and P_2 with $P > (v\vec{r})\langle M.P_1\rangle P_2$, $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$. By (Trans Cap), $P \xrightarrow{M} (v\vec{r})(P_1 \mid P_2)$, and we have that $(v\vec{r})(P_1 \mid P_2) \equiv Q'$.

(Trans In) We have $Q > (v\vec{q})\langle n[Q_1]\rangle Q_2$, $Q_1 \xrightarrow{in\ m} Q_1'$, $Q_2 > (v\vec{r})\langle m[Q_2']\rangle Q_2''$, and $Q' = (v\vec{q},\vec{r})(m[n[Q_1'] \mid Q_2'] \mid Q_2'')$ with $\{\vec{r}\} \cap fn(n[Q_1]) = \emptyset$ and $\{\vec{r}\} \cap \{\vec{q}\} = \emptyset$. By Proposition 6, we have $P > (v\vec{q})\langle n[P_1]\rangle P_2$, with $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$. By the induction hypothesis, $P_1 \xrightarrow{in\ m} P_1'$ with $P_1' \equiv Q_1'$. By Proposition 6, $P_2 > (v\vec{r})\langle m[P_2']\rangle P_2''$, with $P_2' \equiv Q_2'$ and $P_2'' \equiv Q_2''$. By Lemma 20, $fn(n[P_1]) = fn(n[Q_1])$, and therefore $\{\vec{r}\} \cap fn(n[P_1]) = \emptyset$. Let $P' = (v\vec{q},\vec{r})(m[n[P_1'] \mid P_2'] \mid P_2'')$. By (Trans In), we have $P \xrightarrow{\tau} P'$. Moreover, $P' \equiv (v\vec{q},\vec{r})(m[n[Q_1'] \mid Q_2'] \mid Q_2'')$, that is, $P' \equiv Q'$.

(Trans Out) We have $Q > (v\vec{p})\langle n[Q_1]\rangle Q_2$, $Q_1 > (v\vec{q})\langle m[Q_3]\rangle Q_4$, and $Q_3 \xrightarrow{out\ n} Q_3'$, with $Q' = (v\vec{p})(Q_2 \mid (v\vec{q})(n[Q_4] \mid m[Q_3']))$ and $n \notin \{\vec{q}\}$. By Proposition 6, $P > (v\vec{p})\langle n[P_1]\rangle P_2$, with $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$. By Proposition 6, $P_1 > (v\vec{q})\langle m[P_3]\rangle P_4$, with $P_3 \equiv Q_3$ and $P_4 \equiv Q_4$. By the induction hypothesis, $P_3 \xrightarrow{out\ n} P_3'$ with $P_3' \equiv Q_3'$. Let $P' = (v\vec{p})(P_2 \mid (v\vec{q})(n[P_4] \mid m[P_3']))$. By (Trans Out), we have $P \xrightarrow{\tau} P'$. Moreover, $P' \equiv (v\vec{p})(Q_2 \mid (v\vec{q})(n[Q_4] \mid m[Q_3']))$, that is, $P' \equiv Q'$.

(Trans Amb) We have $Q > (v\vec{r})\langle n[Q_1]\rangle Q_2$, $Q_1 \xrightarrow{\tau} Q_1'$, $Q' = (v\vec{r})(n[Q_1'] \mid Q_2)$. By Proposition 6, $P > (v\vec{r})\langle n[P_1]\rangle P_2$ with $P_1 \equiv Q_1$ and $P_2 \equiv Q_2$. By the induction hypothesis, $P_1 \xrightarrow{\tau} P_1'$ with $P_1' \equiv Q_1'$. Let $P' = (v\vec{r})(n[P_1'] \mid P_2)$. By (Trans Amb), $P \xrightarrow{\tau} P'$. Moreover, $P' \equiv (v\vec{r})(n[Q_1'] \mid Q_2)$, that is, $P' \equiv Q'$.

The other cases, (Trans Open) and (Trans I/O), follow similarly. □

Lemma 36.

(1) If $P \xrightarrow{\alpha} P'$ and $n \notin fn(\alpha)$, there is Q with $(vn)P \xrightarrow{\alpha} Q$ and $Q \equiv (vn)P'$.
(2) If $(vn)P \xrightarrow{\alpha} Q$ and $n \notin fn(\alpha)$, there is P' with $P \xrightarrow{\alpha} P'$ and $Q \equiv (vn)P'$.

Proof. The proof is by inductions on the derivations of $P \xrightarrow{\alpha} P'$ and $(vn)P \xrightarrow{\alpha} Q$, respectively. We omit the details. □

The following establishes the left-to-right direction of Theorem 9.

Proposition 37. If $P \to Q$, then $P \xrightarrow{\tau} \equiv Q$.

Proof. The proof is by induction on the derivation of $P \to Q$. The only interesting case is (Red ≡). We omit the other cases, which are routine.

(Red ≡) Here, $P \equiv P'$, $P' \to Q'$ and $Q' \equiv Q$. By the induction hypothesis, $P' \xrightarrow{\tau} \equiv Q'$. By (Struct Trans), this and $Q' \equiv Q$ imply $P' \xrightarrow{\tau} \equiv Q$. By Lemma 35, $P \equiv P'$ and $P' \xrightarrow{\tau} \equiv Q$ imply that $P \xrightarrow{\tau} \equiv Q$. □

Proof of Theorem 9. $P \to Q$ if and only if $P \xrightarrow{\tau} \equiv Q$.

Proof. The proof follows by combining Propositions 34 and 37 and rule (Red ≡). □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

A.4. *Proofs omitted from Section 4.4*

We provide proofs for Lemma 13 and Theorem 15.

Proof of Lemma 13. If $H\{P\} > (\nu\vec{p})\langle P_1\rangle P_2$, then either:

(1) $H > (\nu\vec{p})\langle n[H']\rangle P_2$ and $P_1 = n[H'\{P\}]$, or
(2) $H > (\nu\vec{p})\langle P_1\rangle H'$ and $P_2 = H'\{P\}$, or
(3) $P > (\nu\vec{p})\langle P_1\rangle P'$, $H \equiv - \mid R$, $P_2 \equiv P' \mid R$, and $\{\vec{p}\} \cap fn(R) = \emptyset$.

Proof. The proof is by induction on the derivation of $H\{P\} > (\nu\vec{p})\langle P_1\rangle P_2$:

(Harden Par 1) Then $H\{P\} = Q_1 \mid Q_2$, $Q_1 > (\nu\vec{p})\langle P_1\rangle P_3$, and $P_2 = P_3 \mid Q_2$, with $\{\vec{p}\} \cap fn(Q_2) = \emptyset$. Given that $H\{P\} = Q_1 \mid Q_2$, there are three cases to consider:

— Here $H = -$ and $P = Q_1 \mid Q_2$. Case (3) of the lemma pertains, with $R = \mathbf{0}$.

— Here $H = Q_1 \mid H_2$ and $Q_2 = H_2\{P\}$. By (Harden Par 1) and $\{\vec{p}\} \cap fn(Q_2) = \emptyset$, we may derive $H\{R\} > (\nu\vec{p})\langle P_1\rangle(P_3 \mid H_2\{R\})$ for all R with $\{\vec{p}\} \cap fn(R) = \emptyset$. Let $H' = P_3 \mid H_2$. We have $H > (\nu\vec{p})\langle P_1\rangle H'$, and, moreover, $P_2 = P_3 \mid Q_2 = P_3 \mid H_2\{P\} = H'\{P\}$. So case (2) of the lemma pertains.

— Here $H = H_1 \mid Q_2$ and $Q_1 = H_1\{P\}$. By the induction hypothesis, $H_1\{P\} > (\nu\vec{p})\langle P_1\rangle P_3$ implies that one of three cases holds:

(1) $H_1 > (\nu\vec{p})\langle n[H']\rangle P_3$ and $P_1 = n[H'\{P\}]$. We can derive $H > (\nu\vec{p})\langle n[H']\rangle(P_3 \mid Q_2)$ since $\{\vec{p}\} \cap fn(Q_2) = \emptyset$. Therefore, $H > (\nu\vec{p})\langle n[H']\rangle P_2$, as required to establish case (1) of the lemma.

(2) $H_1 > (\nu\vec{p})\langle P_1\rangle H'$ and $P_3 = H'\{P\}$. We have $H > (\nu\vec{p})\langle P_1\rangle(H' \mid Q_2)$ since $\{\vec{p}\} \cap fn(Q_2) = \emptyset$. Moreover, $P_2 = P_3 \mid Q_2 = H'\{P\} \mid Q_2$. This establishes case (2) of the lemma.

(3) $P > (\nu\vec{p})\langle P_1\rangle P'$, $H_1 \equiv - \mid R$, $P_3 \equiv P' \mid R$ and $\{\vec{p}\} \cap fn(R) = \emptyset$. We have $H \equiv - \mid R \mid Q_2$, $P_2 \equiv P' \mid R \mid Q_2$ and $\{\vec{p}\} \cap fn(R \mid Q_2) = \emptyset$. This establishes case (3) of the lemma.

We omit the remaining cases. □

For the purposes of proving Theorem 15, we adopt the following notation.

Interaction between a harness and a process: $H \bullet P \rightsquigarrow R$

Let $H \bullet P \rightsquigarrow R$ if and only if there are H' and $\vec{r}$ with $\{\vec{r}\} \cap fn(P) = \emptyset$, and one of the following holds:

(Inter In) $H \equiv (\nu\vec{r})H'\{m[- \mid R'] \mid n[R'']\}$, $P \xrightarrow{in\ n} P'$, and $R \equiv (\nu\vec{r})H'\{n[m[P' \mid R'] \mid R'']\}$

(Inter Out) $H \equiv (\nu\vec{r})H'\{n[m[- \mid R'] \mid R'']\}$, $P \xrightarrow{out\ n} P'$, and $R \equiv (\nu\vec{r})H'\{m[P' \mid R'] \mid n[R'']\}$

(Inter Open) $H \equiv (\nu\vec{r})H'\{- \mid n[R']\}$, $P \xrightarrow{open\ n} P'$, and $R \equiv (\nu\vec{r})H'\{P' \mid R'\}$

(Inter Input) $H \equiv (\nu\vec{r})H'\{- \mid \langle M\rangle\}$, $P > (\nu\vec{p})\langle(x).P'\rangle P''$, and $R \equiv (\nu\vec{r})H'\{(\nu\vec{p})(P'\{x \leftarrow M\} \mid P'')\}$, with $\{\vec{p}\} \cap fn(M) = \emptyset$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(Inter Output) $H \equiv (v\vec{r})H'\{- \mid (x).R'\}$, $P > (v\vec{p})\langle\langle M\rangle\rangle P'$,
and $R \equiv (v\vec{r})H'\{(v\vec{p})(P' \mid R'\{x \leftarrow M\})\}$, with $\{\vec{p}\} \cap fn(R') = \emptyset$
(Inter Amb) $P > (v\vec{p})\langle n[Q]\rangle P'$ and one of the following holds:

(1) $Q \xrightarrow{in\ m} Q'$, $H \equiv (v\vec{r})H'\{- \mid m[R']\}$, $\{\vec{p}\} \cap fn(m[R']) = \emptyset$,
and $R \equiv (v\vec{r})H'\{(v\vec{p})(P' \mid m[n[Q'] \mid R'])\}$
(2) $Q \xrightarrow{out\ m} Q'$, $H \equiv (v\vec{r})H'\{m[- \mid R']\}$, $\{\vec{p}\} \cap fn(m[R']) = \emptyset$,
and $R \equiv (v\vec{r})H'\{(v\vec{p})(n[Q'] \mid m[P' \mid R'])\}$
(3) $H \equiv (v\vec{r})H'\{m[R' \mid in\ n.R''] \mid -\}$, $\{\vec{p}\} \cap fn(m[R' \mid in\ n.R'']) = \emptyset$,
and $R \equiv (v\vec{r})H'\{(v\vec{p})(n[Q \mid m[R' \mid R'']] \mid P')\}$
(4) $H \equiv (v\vec{r})H'\{- \mid open\ n.R'\}$, $n \notin \{\vec{p}\}$,
and $R \equiv (v\vec{r})H'\{(v\vec{p})(Q \mid P') \mid R'\}$

The following lemmas concerning the $H \bullet P \rightsquigarrow R$ notation may easily be checked. (Lemma 40 is not a simple consequence of Lemma 39 since n may occur free in H.)

Lemma 38. If $H \bullet P \rightsquigarrow R$ and $H \equiv H'$ and $R \equiv R'$, then $H' \bullet P \rightsquigarrow R'$.

Lemma 39. If $H \bullet P \rightsquigarrow R$, then $H'\{H\} \bullet P \rightsquigarrow H'\{R\}$.

Lemma 40. If $H \bullet P \rightsquigarrow R$ and $n \notin fn(P)$, then $(vn)H \bullet P \rightsquigarrow (vn)R$.

The following lemma is a simple specialisation of Lemma 13:

Lemma 41. If $H\{P\} > (v\vec{p})\langle n[P_1]\rangle P_2$, then either:
(1) $H \equiv (v\vec{p})(n[H'] \mid P_2)$ and $P_1 = H'\{P\}$, or
(2) $H \equiv (v\vec{p})(n[P_1] \mid H')$ and $P_2 = H'\{P\}$, or
(3) $P > (v\vec{p})\langle n[P_1]\rangle P'$, $H \equiv - \mid R$, $P_2 \equiv P' \mid R$, and $\{\vec{p}\} \cap fn(R) = \emptyset$.

The next two lemmas follow from the definition of the M-transitions in terms of hardening.

Lemma 42. If $H\{P\} \xrightarrow{M} R$ for $M \in \{in\ n, out\ n, open\ n\}$, then either:
(1) $H \equiv (v\vec{r})(M.R' \mid H')$, $R \equiv (v\vec{r})(R' \mid H'\{P\})$, $\{\vec{r}\} \cap (\{n\} \cup fn(P)) = \emptyset$, or
(2) $H \equiv - \mid R'$, $P \xrightarrow{M} P'$, and $R \equiv P' \mid R'$.

Lemma 43. If $P \mid Q \xrightarrow{M} R$, then either:
(1) $P \xrightarrow{M} P'$ and $R \equiv P' \mid Q$, or
(2) $Q \xrightarrow{M} Q'$ and $R \equiv P \mid Q'$.

The following proposition is the main fact we need to prove in order to establish Theorem 15.

Proposition 44. If $H\{P\} \xrightarrow{\tau} R$, then one of the following holds:
(1) there is a reduction $P \rightarrow P'$ with $R \equiv H\{P'\}$, or
(2) there is a reduction $H \rightarrow H'$ with $R \equiv H'\{P\}$, or
(3) $H \bullet P \rightsquigarrow R$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Proof. The proof is by induction on the derivation of $H\{P\} \xrightarrow{\tau} R$:

(Trans Open) Here, $H\{P\} > (\nu\vec{q})\langle n[Q_1]\rangle Q_2$ and $Q_2 \xrightarrow{open\ n} Q_2'$ and $R = (\nu\vec{q})(Q_1 \mid Q_2')$. We may assume that $\{\vec{q}\} \cap fn(P) = \emptyset$. By Lemma 41, $H\{P\} > (\nu\vec{q})\langle n[Q_1]\rangle Q_2$ implies there are three cases to consider:

(1) $H \equiv (\nu\vec{q})(n[H'] \mid Q_2)$ and $Q_1 = H'\{P\}$. Let $H'' = (\nu\vec{q})(H' \mid Q_2')$. In this case we can see, for all Q, that $H\{Q\} \to H''\{Q\}$, which is to say that $H \to H''$. Moreover, $R \equiv (\nu\vec{q})(H'\{P\} \mid Q_2') \equiv H''\{P\}$. Hence, case (2) pertains.

(2) $H \equiv (\nu\vec{q})(n[Q_1] \mid H_1)$ and $Q_2 = H_1\{P\}$. By Lemma 42, $H_1\{P\} \xrightarrow{open\ n} Q_2'$ implies either:

(a) $H_1 \equiv (\nu\vec{r})(open\ n.R' \mid H_2)$, $Q_2' \equiv (\nu\vec{r})(R' \mid H_2\{P\})$ and $\{\vec{r}\} \cap (\{n\} \cup fn(P)) = \emptyset$. Let $H' = (\nu\vec{q})(Q_1 \mid (\nu\vec{r})(R' \mid H_2))$. We have that $H\{Q\} \to H'\{Q\}$ for all Q, that is, $H \to H'$. Moreover, $R \equiv (\nu\vec{q})(Q_1 \mid (\nu\vec{r})(R' \mid H_2\{P\})) \equiv H'\{P\}$. Hence, case (2) pertains.

(b) $H_1 \equiv - \mid R'$, $P \xrightarrow{open\ n} P'$ and $Q_2' \equiv P' \mid R'$. From $H \equiv (\nu\vec{q})(R' \mid - \mid n[Q_1])$, $P \xrightarrow{open\ n} P'$ and $R \equiv (\nu\vec{q})(Q_1 \mid P' \mid R') \equiv (\nu\vec{q})(R' \mid P' \mid Q_1)$ we may derive $H \bullet P \rightsquigarrow R$ using (Inter Open). Hence, case (3) pertains.

(3) $P > (\nu\vec{q})\langle n[Q_1]\rangle P'$, $H \equiv - \mid R'$, $Q_2 \equiv P' \mid R'$ and $\{\vec{q}\} \cap fn(R') = \emptyset$. From $P > (\nu\vec{q})\langle n[Q_1]\rangle P'$ we get $P \equiv (\nu\vec{q})(n[Q_1] \mid P')$. By Lemma 35, $Q_2 \equiv P' \mid R'$ and $Q_2 \xrightarrow{open\ n} Q_2'$ imply there is Q_2'' such that $P' \mid R' \xrightarrow{open\ n} Q_2''$ and $Q_2'' \equiv Q_2'$. By Lemma 43, there are two cases to consider:

(a) $P' \xrightarrow{open\ n} P''$ and $Q_2'' \equiv P'' \mid R'$. We have $P \to (\nu\vec{q})(Q_1 \mid P'')$, $H \equiv - \mid R'$ and $R \equiv (\nu\vec{q})(Q_1 \mid Q_2') \equiv (\nu\vec{q})(Q_1 \mid P'' \mid R') \equiv (\nu\vec{q})(Q_1 \mid P'') \mid R'$. Hence, case (1) pertains.

(b) $R' \xrightarrow{open\ n} R''$ and $Q_2'' \equiv P' \mid R''$. From $R' \xrightarrow{open\ n} R''$ it follows that $R' \equiv (\nu\vec{r})(R_1 \mid open\ n.R_2)$ with $R'' \equiv (\nu\vec{r})(R_1 \mid R_2)$ and $n \notin \{\vec{r}\}$. We have:

$$\begin{aligned} H &\equiv (\nu\vec{r})(R_1 \mid - \mid open\ n.R_2) \\ R &\equiv (\nu\vec{q})(Q_1 \mid Q_2') \\ &\equiv (\nu\vec{q})(Q_1 \mid P' \mid R'') \\ &\equiv (\nu\vec{q})(Q_1 \mid P' \mid (\nu\vec{r})(R_1 \mid R_2)) \\ &\equiv (\nu\vec{r})(R_1 \mid (\nu\vec{q})(Q_1 \mid P') \mid R_2), \end{aligned}$$

since we may assume that $\{\vec{q}\} \cap fn(R_1 \mid R_2) = \emptyset$ and $\{\vec{r}\} \cap fn(Q_1 \mid P') = \emptyset$ and $\{\vec{q}\} \cap \{\vec{r}\} = \emptyset$. From $\{\vec{q}\} \cap fn(R') = \emptyset$ and $R' \xrightarrow{open\ n} R''$ it follows that $n \notin \{\vec{q}\}$. From $P > (\nu\vec{q})\langle n[Q_1]\rangle P'$, $n \notin \{\vec{q}\}$ and the two displayed equations, we may derive $H \bullet P \rightsquigarrow R$ using clause (4) of (Inter Amb). Hence, case (3) pertains.

(Trans Amb) Here, $H\{P\} > (\nu\vec{q})\langle n[Q_1]\rangle Q_2$, $Q_1 \xrightarrow{\tau} Q_1'$ and $R = (\nu\vec{q})(n[Q_1'] \mid Q_2)$. From $H\{P\} > (\nu\vec{q})\langle n[Q_1]\rangle Q_2$ it follows that $\{\vec{q}\} \cap fn(P) = \emptyset$, since $fn(P) \subseteq fn(H\{P\})$. By Lemma 13, $H\{P\} > (\nu\vec{q})\langle n[Q_1]\rangle Q_2$ implies there are three cases to consider:

(1) $H > (\nu\vec{q})\langle n[H']\rangle Q_2$ and $Q_1 = H'\{P\}$. By the induction hypothesis, we have that $Q_1 = H'\{P\} \xrightarrow{\tau} Q_1'$ implies one of the following:

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(a) Here $P \to P'$ with $Q_1' \equiv H'\{P'\}$. We have $R \equiv (\nu\vec{q})(n[H'\{P'\}] \mid Q_2)$ and $H \equiv (\nu\vec{q})(n[H'] \mid Q_2)$, so case (1) pertains.

(b) Here $H' \to H''$ with $Q_1' \equiv H''\{P\}$. From $H > (\nu\vec{q})\langle n[H']\rangle Q_2$ and $H' \to H''$ we can derive $H \to (\nu\vec{q})(n[H''] \mid Q_2)$. We have $R \equiv (\nu\vec{q})(n[H''\{P\}] \mid Q_2)$, so case (2) pertains.

(c) Here $H' \bullet P \rightsquigarrow Q_1'$. From $H > (\nu\vec{q})\langle n[H']\rangle Q_2$ we get that $H \equiv (\nu\vec{q})(n[H'] \mid Q_2)$. Also, $R \equiv (\nu\vec{q})(n[Q_1'] \mid Q_2)$. By Lemma 39, $H' \bullet P \rightsquigarrow Q_1'$ implies that $n[H'] \mid Q_2 \bullet P \rightsquigarrow n[Q_1'] \mid Q_2$. By Lemma 40, $\{\vec{q}\} \cap fn(P) = \varnothing$ implies that $(\nu\vec{q})(n[H'] \mid Q_2) \bullet P \rightsquigarrow (\nu\vec{q})(n[Q_1'] \mid Q_2)$. By Lemma 38, $H \bullet P \rightsquigarrow R$. Hence case (3) pertains.

(2) $H > (\nu\vec{q})\langle n[Q_1]\rangle H_1$ and $Q_2 = H_1\{P\}$. Let $H' = (\nu\vec{q})(n[Q_1'] \mid H_1)$. Since $H \equiv (\nu\vec{q})(n[Q_1] \mid H_1)$ and $Q_1 \xrightarrow{\tau} Q_1'$, we get that $H \to H'$. Moreover, $R \equiv (\nu\vec{q})(n[Q_1'] \mid H_1\{P\}) \equiv H'\{P\}$. Hence case (2) pertains.

(3) $P > (\nu\vec{q})\langle n[Q_1]\rangle P'$, $H \equiv - \mid R'$, $Q_2 \equiv P' \mid R'$ and $\{\vec{q}\} \cap fn(R') = \varnothing$. Let $P' = (\nu\vec{q})(n[Q_1'] \mid P')$. From $Q_1 \xrightarrow{\tau} Q_1'$ and $P \equiv (\nu\vec{q})(n[Q_1] \mid P')$, we get that $P \to P'$. Moreover, $R \equiv (\nu\vec{q})(n[Q_1'] \mid P' \mid R') \equiv H\{P'\}$. Hence case (1) pertains.

The cases for the rules (Trans In), (Trans Out) and (Trans I/O) are proved by arguments similar to that for (Trans Open). Since the rule (Trans Cap) cannot derive a τ-transition, this completes the analysis of all the rules that may derive $H\{P\} \xrightarrow{\tau} R$. □

We now prove Theorem 15, which we restate in terms of the interaction predicate, $H \bullet P \rightsquigarrow R$.

Proof of Theorem 15. $H\{P\} \to R$ if and only if:

(Act Proc) $P \to P'$ with $R \equiv H\{P'\}$, or
(Act Har) $H \to H'$ with $R \equiv H'\{P\}$, or
(Act Inter) $H \bullet P \rightsquigarrow R$.

Proof. The right-to-left direction is a routine calculation. For the left-to-right direction, suppose that $H\{P\} \to R$. By Theorem 9, there is Q with $H\{P\} \xrightarrow{\tau} Q$ and $Q \equiv R$. By Proposition 44, there are three cases to consider:

(1) There is a reduction $P \to P'$ with $Q \equiv H\{P'\}$. From $Q \equiv R$ we get $R \equiv H\{P'\}$, so (Act Proc) applies.
(2) There is a reduction $H \to H'$ with $Q \equiv H'\{P\}$. From $Q \equiv R$ we get $R \equiv H'\{P\}$, so (Act Har) applies.
(3) We have $H \bullet P \rightsquigarrow Q$. By Lemma 38, $Q \equiv R$ implies that $H \bullet P \rightsquigarrow R$. Therefore, (Act Inter) applies. □

A.5. *Proofs about replication*

In this section, we prove a series of lemmas about replicated processes. These lemmas are needed in the next section, in the proof of Proposition 61, that the equivalence implicit in the context lemma is a congruence with respect to replication.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

We use the notation P^k as an abbreviation for k copies of P running in parallel: we inductively define $P^0 \triangleq \mathbf{0}$ and $P^{k+1} \triangleq P \mid P^k$.

Lemma 45. If $!P > (\nu\vec{p})\langle Q\rangle R$, there is P' such that $P > (\nu\vec{p})\langle Q\rangle P'$ with $R = P' \mid !P$ and $\{\vec{p}\} \cap fn(P) = \emptyset$.

Proof. The judgment $!P > (\nu\vec{p})\langle Q\rangle R$ can only be derived using the rule (Harden Repl), from a judgment $P > (\nu\vec{p})\langle Q\rangle P'$ such that $R = P' \mid !P$. By Lemma 22, $P > (\nu\vec{p})\langle Q\rangle P'$ implies that $fn(P) = fn((\nu\vec{p})\langle Q\rangle P')$, and therefore that $\{\vec{p}\} \cap fn(P) = \emptyset$. □

Lemma 46. If $!P \xrightarrow{M} Q$, there is R such that $P \xrightarrow{M} R$ and $Q \equiv R \mid !P$.

Proof. The judgment $!P \xrightarrow{M} Q$ can only be derived using (Trans Cap) from a judgment $!P > (\nu\vec{p})\langle M.P'\rangle P''$ with $fn(M) \cap \{\vec{p}\} = \emptyset$ and $Q = (\nu\vec{p})(P' \mid P'')$. By Lemma 45, there is P''' with $P > (\nu\vec{p})\langle M.P'\rangle P'''$, $P'' = P''' \mid !P$ and $\{\vec{p}\} \cap fn(P) = \emptyset$. Let $R = (\nu\vec{p})(P' \mid P''')$. By (Trans Cap), we have $P \xrightarrow{M} R$. Moreover, $Q = (\nu\vec{p})(P' \mid P''' \mid !P) \equiv R \mid !P$. □

Lemma 47. If $!P \xrightarrow{\tau} Q$, there is R with $P \mid P \xrightarrow{\tau} R$ and $Q \equiv R \mid !P$.

Proof. We use a case analysis of the derivation of $!P \xrightarrow{\tau} Q$:

(Trans Amb) Here, $!P \xrightarrow{\tau} (\nu\vec{p})(n[Q'] \mid P')$ derives from $!P > (\nu\vec{p})\langle n[Q]\rangle P'$ and $Q \xrightarrow{\tau} Q'$. By Lemma 45, $!P > (\nu\vec{p})\langle n[Q]\rangle P'$ implies there is R' such that $P > (\nu\vec{p})\langle n[Q]\rangle R'$, $P' = R' \mid !P$ and $fn(P) \cap \{\vec{p}\} = \emptyset$. By (Harden Par 1), $P > (\nu\vec{p})\langle n[Q]\rangle R'$ and $\{\vec{p}\} \cap fn(P) = \emptyset$ imply that $P \mid P > (\nu\vec{p})\langle n[Q]\rangle(R' \mid P)$. By (Trans Amb), this and $Q \xrightarrow{\tau} Q'$ imply that $P \mid P \xrightarrow{\tau} R$, where $R = (\nu\vec{p})(n[Q'] \mid R' \mid P)$. Finally, we may calculate $(\nu\vec{p})(n[Q'] \mid P') = (\nu\vec{p})(n[Q'] \mid R' \mid !P) \equiv (\nu\vec{p})(n[Q'] \mid R' \mid P \mid !P) \equiv R \mid !P$.

(Trans In) Here, $!P \xrightarrow{\tau} (\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1] \mid R_2)$ derives from the judgments $!P > (\nu\vec{p})\langle n[Q]\rangle R$, $Q \xrightarrow{in\ m} Q'$ and $R > (\nu\vec{r})\langle m[R_1]\rangle R_2$, with $\{\vec{r}\} \cap fn(n[Q]) = \emptyset$ and $\{\vec{r}\} \cap \{\vec{p}\} = \emptyset$. By Lemma 45, $!P > (\nu\vec{p})\langle n[Q]\rangle R$ implies there is R' such that $P > (\nu\vec{p})\langle n[Q]\rangle R'$ and $R \equiv R' \mid !P$ with $\{\vec{p}\} \cap fn(P) = \emptyset$.

By Proposition 6 and (Struct Symm), $R > (\nu\vec{r})\langle m[R_1]\rangle R_2$ and $R \equiv R' \mid !P$ imply there are R_1' and R_2' such that $R' \mid !P > (\nu\vec{r})\langle m[R_1']\rangle R_2'$, $R_1 \equiv R_1'$ and $R_2 \equiv R_2'$. Only two rules may derive the judgment $R' \mid !P > (\nu\vec{r})\langle m[R_1']\rangle R_2'$:

(Harden Par 1) In this case $R' > (\nu\vec{r})\langle m[R_1']\rangle R''$ with $R_2' = R'' \mid !P$ and $\{\vec{r}\} \cap fn(!P) = \emptyset$. By (Harden Par 1), $P > (\nu\vec{p})\langle n[Q]\rangle R'$ and $\{\vec{p}\} \cap fn(P) = \emptyset$ imply that $P \mid P > (\nu\vec{p})\langle n[Q]\rangle(R' \mid P)$. By (Harden Par 1), $R' > (\nu\vec{r})\langle m[R_1']\rangle R''$ and $\{\vec{r}\} \cap fn(P) = \emptyset$ imply that $R' \mid P > (\nu\vec{r})\langle m[R_1']\rangle(R'' \mid P)$. By (Trans In), $P \mid P > (\nu\vec{p})\langle n[Q]\rangle(R' \mid P)$, $Q \xrightarrow{in\ m} Q'$ and $R' \mid P > (\nu\vec{r})\langle m[R_1']\rangle(R'' \mid P)$ imply that $P \mid P \xrightarrow{\tau} (\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R'' \mid P)$. We know that $fn(P) \cap \{\vec{p},\vec{r}\} = \emptyset$, and hence we may calculate

$$\begin{aligned}(\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1] \mid R_2) &\equiv (\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R_2') \\ &= (\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R'' \mid !P) \\ &\equiv (\nu\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R'' \mid P) \mid !P.\end{aligned}$$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(Harden Par 2) In this case $!P > (v\vec{r})\langle m[R_1']\rangle R''$ with $R_2' = R' \mid R''$ and $\{\vec{r}\} \cap fn(R') = \emptyset$. By Lemma 45, $!P > (v\vec{r})\langle m[R_1']\rangle R''$ implies there is R''' such that $P > (v\vec{r})\langle m[R_1']\rangle R'''$ with $R'' = R''' \mid !P$ and $\{\vec{r}\} \cap fn(P) = \emptyset$. By (Harden Par 1), $P > (v\vec{p})\langle n[Q]\rangle R'$ and $\{\vec{p}\} \cap fn(P) = \emptyset$ imply that $P \mid P > (v\vec{p})\langle n[Q]\rangle(R' \mid P)$. By (Harden Par 2), $\{\vec{r}\} \cap fn(R') = \emptyset$ and $P > (v\vec{r})\langle m[R_1']\rangle R'''$ imply that $R' \mid P > (v\vec{r})\langle m[R_1']\rangle(R' \mid R''')$. By (Trans In), $P \mid P > (v\vec{p})\langle n[Q]\rangle(R' \mid P)$, $Q \xrightarrow{in\ m} Q'$ and $R' \mid P > (v\vec{r})\langle m[R_1']\rangle(R' \mid R''')$ imply $P \mid P \xrightarrow{\tau} (v\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R' \mid R''')$. We know that $fn(P) \cap \{\vec{p},\vec{r}\} = \emptyset$, and hence we may calculate

$$\begin{aligned}(v\vec{p},\vec{r})(m[n[Q'] \mid R_1] \mid R_2) &\equiv (v\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R' \mid R'')\\ &= (v\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R' \mid (R''' \mid !P))\\ &\equiv (v\vec{p},\vec{r})(m[n[Q'] \mid R_1'] \mid R' \mid R''') \mid !P.\end{aligned}$$

The other cases – (Trans Out), (Trans Open) and (Trans I/O) – follow by similar arguments. □

Lemma 48. If $H\{!P\} \to R$, there is H' such that $R \equiv H'\{!P\}$ and for all k, $H\{P^{k+2}\} \to H'\{P^k\}$.

Proof. The proof is a case analysis induced by Theorem 15. We omit the details. □

Lemma 49. If $H\{!P\} \Downarrow n$, there is k such that $H\{P^k\} \Downarrow n$.

Proof. The proof is by induction on the derivation of $H\{!P\} \Downarrow n$:

(Conv Exh) Here, $H\{!P\} \downarrow n$. By Proposition 14, this implies that either (1) $H\{Q\} \downarrow n$ for all Q, or (2) $!P \downarrow n$, and for all Q, we have $Q \downarrow n$ implies that $H\{Q\} \downarrow n$. In case (1), let $k = 1$ and we have $H\{P\} \downarrow n$. In case (2), Proposition 7 implies that $!P > (v\vec{p})\langle n[P']\rangle P''$ with $n \notin \{\vec{p}\}$, for some names $\vec{p}$ and processes P' and P''. By Lemma 45, it follows that there is P''' such that $P > (v\vec{p})\langle n[P']\rangle P'''$ with $P'' = P''' \mid !P$ and $\{\vec{p}\} \cap fn(P) = \emptyset$. Proposition 7 now yields $P \downarrow n$. Let $k = 1$ and we get $H\{P\} \downarrow n$.

(Conv Red) Here, $H\{!P\} \to Q$ and $Q \Downarrow n$. By Lemma 48, $H\{!P\} \to Q$ implies there is H' such that $Q \equiv H'\{!P\}$ and, for all j, $H\{P^{j+2}\} \to H'\{P^j\}$. By Lemma 2, there is a derivation of $H'\{!P\} \Downarrow n$ with the same depth of inference as the derivation of $Q \Downarrow n$. By the induction hypothesis, there is k such that $H'\{P^k\} \Downarrow n$. Now, we have that $H\{P^{k+2}\} \to H'\{P^k\}$. By (Conv Red), this and $H'\{P^k\} \Downarrow n$ imply that $H\{P^{k+2}\} \Downarrow n$. □

A.6. *Proofs omitted from Section 4.3*

The purpose of this section is to prove our context lemma, Theorem 12. Roughly speaking, the context lemma asserts that the distinctions made by all contexts are the same as the distinctions made by harnesses. To prove the context lemma, it is convenient to introduce the following auxiliary equivalence, defined in terms of harnesses. Recall that a *substitution*, σ, is a list $x_1 \leftarrow M_1, \ldots, x_k \leftarrow M_k$, where the variables $x_1, \ldots, x_k$ are pairwise distinct and $fv(M_i) = \emptyset$ for each $i \in 1..k$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

The equivalence implicit in the context lemma: $P \sim Q$

Let $P \sim Q$ if and only if for all substitutions σ with $dom(\sigma) = fv(P) \cup fv(Q)$,
and for all closed harnesses H and names n, we have $H\{P\sigma\} \Downarrow n \Leftrightarrow H\{Q\sigma\} \Downarrow n$.

Next, we prove a series of lemmas, which taken together imply Proposition 64, that the auxiliary equivalence $P \sim Q$ is a congruence. The context lemma then follows easily.

Proposition 50. The relation $P \sim Q$ is an equivalence, that is, reflexive, transitive and symmetric. Moreover, if $P \equiv Q$, then $P \sim Q$.

Proof. That $P \sim Q$ is an equivalence follows easily from its definition. Suppose that $P \equiv Q$. Consider any substitution σ such that $fv(P) \cup fv(Q) = dom(\sigma)$. Structural congruence is preserved by substitutions, so $P\sigma \equiv Q\sigma$. Moreover, structural congruence is a congruence, so $H\{P\sigma\} \equiv H\{Q\sigma\}$. By Lemma 2, it follows that for all n, we have $H\{P\sigma\} \Downarrow n \Leftrightarrow H\{Q\sigma\} \Downarrow n$. Therefore, $P \sim Q$. □

Proposition 51. If $P \sim P'$, then $P \mid Q \sim P' \mid Q$.

Proof. Consider any substitution σ with $dom(\sigma) = fv(P \mid Q) \cup fv(P' \mid Q)$, and any closed harness H and any name n. Let $H' = H\{- \mid Q\sigma\}$. Since $fv(Q) \subset dom(\sigma)$, the harness H' is closed. Let σ' be the restriction of σ to the domain $fv(P) \cup fv(P')$. We have

$$\begin{aligned} H\{(P \mid Q)\sigma\} &= H'\{P\sigma'\} \\ H\{(P' \mid Q)\sigma\} &= H'\{P'\sigma'\}. \end{aligned}$$

Now, suppose $H\{(P \mid Q)\sigma\} \Downarrow n$, that is, $H'\{P\sigma'\} \Downarrow n$. This and $P \sim P'$ imply that $H'\{P'\sigma'\} \Downarrow n$, which is to say, $H\{(P' \mid Q)\sigma\} \Downarrow n$. A symmetric argument establishes that $H\{(P' \mid Q)\sigma\} \Downarrow n$ implies $H\{(P \mid Q)\sigma\} \Downarrow n$. Therefore, $P \mid Q \sim P' \mid Q$. □

Lemma 52. If $m \neq n$, then $(\nu n)P \Downarrow m \Leftrightarrow P \Downarrow m$.

Proof. An induction on the derivation of $P \Downarrow m$ establishes that $(\nu n)P \Downarrow m$, using (Harden Res) and Proposition 7. On the other hand, an induction on the derivation of $(\nu n)P \Downarrow m$ establishes that $P \Downarrow m$, using Lemma 2, Theorem 9 and Lemma 36. □

Proposition 53. If $P \sim P'$, then $(\nu n)P \sim (\nu n)P'$.

Proof. Consider any substitution σ with $dom(\sigma) = fv((\nu n)P) \cup fv((\nu n)P')$, that is, $dom(\sigma) = fv(P) \cup fv(P')$. Consider any closed harness H and any name m. Since the name n is bound, we may assume that $n \notin fn(\sigma(x))$ for all $x \in dom(\sigma)$, that $n \notin fn(H)$ and that $m \neq n$. We have

$$\begin{aligned} H\{((\nu n)P)\sigma\} &= (\nu n)(H\{P\sigma\}) \\ H\{((\nu n)P')\sigma\} &= (\nu n)(H\{P'\sigma\}). \end{aligned}$$

By definition of $P \sim P'$, it follows that $H\{P\sigma\} \Downarrow m \Leftrightarrow H\{P'\sigma\} \Downarrow m$. By Lemma 52, it follows that $(\nu n)(H\{P\sigma\}) \Downarrow m \Leftrightarrow (\nu n)(H\{P'\sigma\}) \Downarrow m$, which is to say that $H\{((\nu n)P)\sigma\} \Downarrow m \Leftrightarrow H\{((\nu n)P')\sigma\} \Downarrow m$. It follows that $(\nu n)P \sim (\nu n)P'$. □

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Lemma 54. If M is not a name and $H\{M[P]\} \Downarrow m$, then $H\{\mathbf{0}\} \Downarrow m$.

Proof. The proof is by induction on the derivation of $H\{M[P]\} \Downarrow m$, with appeal to Proposition 14, and the activity lemma, Theorem 15. An ambient $M[P]$, where M is not a name, cannot participate in any transitions. □

Proposition 55. If $P \sim P'$, then $M[P] \sim M[P']$.

Proof. Consider any substitution σ with $dom(\sigma) = fv(M[P]) \cup fv(M[P'])$, that is, $dom(\sigma) = fv(M) \cup fv(P) \cup fv(P')$. Consider any closed harness H and any name m. Either $M\sigma$ is a name n, or not. If not, we get that $H\{(M[P])\sigma\} \Downarrow m \Leftrightarrow H\{\mathbf{0}\} \Downarrow m \Leftrightarrow H\{(M[P'])\sigma\} \Downarrow m$ from Lemma 18 and Lemma 54. On the other hand, suppose that $M\sigma$ is the name n. Let $H' = H\{n[-]\}$. Given that H is closed, so is H'. We have

$$\begin{aligned} H\{(M[P])\sigma\} &= H'\{P\sigma\} \\ H\{(M[P'])\sigma\} &= H'\{P'\sigma\}. \end{aligned}$$

Now, suppose $H\{(M[P])\sigma\} \Downarrow m$, that is, $H'\{P\sigma\} \Downarrow m$. This and $P \sim P'$ imply that $H'\{P'\sigma\} \Downarrow m$, which is to say, $H\{(M[P'])\sigma\} \Downarrow m$. A symmetric argument establishes that $H\{(M[P'])\sigma\} \Downarrow m$ implies $H\{(M[P])\sigma\} \Downarrow m$. Therefore, whether or not M is a name, $M[P] \sim M[P']$. □

The relation $M > \epsilon$ in the following lemma is as defined in Appendix A.2.

Lemma 56. $M.P \to Q$ if and only if $M > \epsilon$ and $P \to Q$.

Proof. The right-to-left direction follows from the fact that $M > \epsilon$ implies that $M.P \equiv P$. For the other direction, $M.P \to Q$ implies, by Theorem 9, that there is R with $M.P \xrightarrow{\tau} R$ and $R \equiv Q$. An inspection of the rules for deriving τ-transitions reveals that the first step in deriving $M.P \xrightarrow{\tau} R$ is a hardening $M.P > C$, where the prime of the concretion C is either an ambient or an output. Therefore, the second case of Lemma 28 must hold, and we have that $M > \epsilon$ and $P > C$. It follows that $P \xrightarrow{\tau} R$, and therefore that $P \to Q$. □

Lemma 57. If $M.P \xrightarrow{N} P'$, then either:

(1) $M > N.N'$ and $P' \equiv N'.P$, or

(2) $M > \epsilon$ and $P \xrightarrow{N} P'$.

Proof. By definition, $M.P \xrightarrow{N} P'$ implies that $M.P > (\nu\vec{p})\langle N.P_1\rangle P_2$ with $P' = (\nu\vec{p})(P_1 \mid P_2)$ and $fn(N) \cap \{\vec{p}\} = \emptyset$. By Lemma 28, one of two cases arises. In the first case, $M > N.N'$, $(\nu\vec{p})\langle N.P_1\rangle P_2 = (\nu)\langle N.R\rangle\mathbf{0}$ and $R \equiv N'.P$. So $\vec{p} = \emptyset$, $P_1 = R$ and $P_2 = \mathbf{0}$. Therefore, $P' \equiv R \mid \mathbf{0} \equiv N'.P$. In the second case, $M > \epsilon$ and $P > (\nu\vec{p})\langle N.P_1\rangle P_2$. By (Trans Cap), $P \xrightarrow{N} (\nu\vec{p})(P_1 \mid P_2) = P'$. □

Lemma 58. Consider any closed P and P' such that $P \sim P'$. If $H\{M.P\} \Downarrow n$, then $H\{M.P'\} \Downarrow n$.

Proof. The proof is by induction on the derivation of $H\{M.P\} \Downarrow n$:

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

(Conv Exh) Here $H\{M.P\} \downarrow n$, and we are to show that $H\{M.P'\} \Downarrow n$. By Proposition 14, either (1) $H\{Q\} \downarrow n$ for all Q, or (2) $M.P \downarrow n$, and for all Q, we have $Q \downarrow n$ implies that $H\{Q\} \downarrow n$. In case (1), we immediately get that $H\{M.P'\} \downarrow n$, and hence $H\{M.P'\} \Downarrow n$ by (Conv Exh). In case (2), $M.P \downarrow n$ implies that $M.P > (\nu\vec{r})\langle n[R_1]\rangle R_2$ with $n \notin \{\vec{r}\}$ by Proposition 7. By Lemma 28, $M.P > (\nu\vec{r})\langle n[R_1]\rangle R_2$ implies that $M > \epsilon$ and $P > (\nu\vec{r})\langle n[R_1]\rangle R_2$. (The first clause of Lemma 28 cannot apply since the prime of the concretion $(\nu\vec{r})\langle n[R_1]\rangle R_2$ is an ambient and not an action.) By Proposition 7 and (Conv Exh), we get that $P \Downarrow n$. Since $P \sim P'$, it follows that $P' \Downarrow n$. So there is P'' such that $P' \rightarrow^* P''$ and $P'' \downarrow n$. We have $H\{M.P'\} \equiv H\{P'\}$ from $M > \epsilon$, and $H\{P'\} \rightarrow^* H\{P''\}$, and $H\{P''\} \downarrow n$, by the property of H obtained from Proposition 14 above. These three facts imply that $H\{M.P'\} \Downarrow n$.

(Conv Red) Here $H\{M.P\} \rightarrow R$ and $R \Downarrow n$. By Theorem 15, one of the following cases must hold:

(Act Proc) Then $M.P \rightarrow R'$ with $R \equiv H\{R'\}$. By Lemma 56, we have that $M > \epsilon$ and $P \rightarrow R'$. If $M > \epsilon$, then $H\{M.P\} \equiv H\{P\}$, so $H\{P\} \Downarrow n$. Since $P \sim P'$, $H\{P\} \Downarrow n$ implies that $H\{P'\} \Downarrow n$. From $M > \epsilon$, we get that $H\{M.P'\} \equiv H\{P'\}$, and therefore that $H\{M.P'\} \Downarrow n$.

(Act Har) Then $H \rightarrow H'$ with $R \equiv H'\{M.P\}$. By Lemma 2, $R \equiv H'\{M.P\}$ implies that $H'\{M.P\} \Downarrow n$ with the same depth of inference as $R \Downarrow n$. By the induction hypothesis, we get $H'\{M.P'\} \Downarrow n$ too. From $H \rightarrow H'$ we get that $H\{M.P'\} \rightarrow H'\{M.P'\}$, and hence that $H\{M.P'\} \Downarrow n$.

(Act Inter) Then there are H' and $\vec{r}$ with $\{\vec{r}\} \cap fn(M.P) = \emptyset$, and one of several cases holds. We consider just one; the others follow by similar arguments.

(Inter In) Here we have $H \equiv (\nu\vec{r})H'\{m[- \mid R'] \mid n[R'']\}$, $M.P \xrightarrow{in\ n} P''$ and $R \equiv (\nu\vec{r})H'\{n[m[P'' \mid R'] \mid R'']\}$. By Lemma 57, $M.P \xrightarrow{in\ n} P''$ implies that one of two cases must hold.

In the first case, $M > in\ n.N'$ and $P'' \equiv N'.P$. Here, $M.P' \xrightarrow{in\ n} N'.P'$, and therefore we have

$$\begin{array}{rcl} H\{M.P'\} & \xrightarrow{\tau} & (\nu\vec{r})H'\{n[m[N'.P' \mid R'] \mid R'']\} \\ R & \equiv & (\nu\vec{r})H'\{n[m[N'.P \mid R'] \mid R'']\}. \end{array}$$

By the induction hypothesis, $R \Downarrow n$ and Lemma 2 imply that

$$(\nu\vec{r})H'\{n[m[N'.P' \mid R'] \mid R'']\} \Downarrow n$$

and therefore that $H\{M.P'\} \Downarrow n$.

In the second case, $M > \epsilon$ and $P \xrightarrow{in\ n} P''$. In this case, $H\{M.P\} \equiv H\{P\}$ and $H\{M.P'\} \equiv H\{P'\}$. Therefore $H\{M.P\} \Downarrow n$ and $P \sim P'$ imply that $H\{M.P'\} \Downarrow n$. □

Proposition 59. If $P \sim P'$, then $M.P \sim M.P'$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Proof. Consider any substitution σ with $dom(\sigma) = fv(M.P) \cup fv(M.P')$, and any closed harness H and any name m. By Lemma 58, we get that $H\{M\sigma.P\sigma)\} \Downarrow m$ if and only if $H\{M\sigma.P'\sigma)\} \Downarrow m$. Hence, $M.P \sim M.P'$. □

Lemma 60. If $H\{P\} \Downarrow n$, then $H\{P \mid Q\} \Downarrow n$.

Proof. Suppose $H\{P\} \Downarrow n$. Let $H' = H\{P \mid -\}$. We have that $H\{P\} \equiv H\{P \mid \mathbf{0}\} = H'\{\mathbf{0}\}$. Hence, by Lemma 2, $H\{P\} \Downarrow n$ implies $H'\{\mathbf{0}\} \Downarrow n$. By Lemma 18, this implies $H'\{Q\} \Downarrow n$, which is to say that $H\{P \mid Q\} \Downarrow n$. □

Proposition 61. If $P \sim P'$, then $!P \sim !P'$.

Proof. Consider any substitution σ with $dom(\sigma) = fv(!P) \cup fv(!P')$, that is, $dom(\sigma) = fv(P) \cup fv(P')$. Consider any closed harness H and any name n. Suppose that $H\{(!P)\sigma\} \Downarrow n$. By Lemma 49, there is k such that $H\{(P\sigma)^k\} \Downarrow n$, which is to say $H\{P^k\sigma\} \Downarrow n$. By Proposition 51, $P^k \sim P'^k$. Therefore, $H\{P^k\sigma\} \Downarrow n$ implies $H\{P'^k\sigma\} \Downarrow n$, which is to say $H\{(P'\sigma)^k\} \Downarrow n$. By Lemma 60, this implies $H\{(P'\sigma)^k \mid !(P'\sigma)\} \Downarrow n$. Since $H\{!P'\sigma\} \equiv H\{(P'\sigma)^k \mid !(P'\sigma)\}$, it follows that $H\{!P'\sigma\} \Downarrow n$, that is, $H\{(!P')\sigma\} \Downarrow n$. By symmetric reasoning, $H\{(!P')\sigma\} \Downarrow n$ implies $H\{(!P)\sigma\} \Downarrow n$. □

Lemma 62. Consider any P and P' such that $P \sim P'$ and $fv(P) \cup fv(P') \subseteq \{x\}$. If $H\{(x).P\} \Downarrow n$, then $H\{(x).P'\} \Downarrow n$.

Proof. The proof is by induction on the derivation of $H\{(x).P\} \Downarrow n$:

(Conv Exh) Here $H\{(x).P\} \downarrow n$. By Proposition 14, either $H\{Q\} \downarrow n$ for all Q, or $(x).P \downarrow n$. In the first case, we get $H\{(x).P'\} \downarrow n$. In the second case, Proposition 7 implies that $(x).P$ hardens to a concretion whose prime is an ambient. This is impossible, so the second case cannot arise.

(Conv Red) Here $H\{(x).P\} \rightarrow R$ and $R \Downarrow n$. By Theorem 15, one of the following cases must hold:

- **(Act Proc)** Then $(x).P \rightarrow R'$ with $R \equiv H\{R'\}$. This case cannot arise, since $(x).P$ has no τ-transitions.
- **(Act Har)** Then $H \rightarrow H'$ with $R \equiv H'\{(x).P\}$. By Lemma 2, $R \equiv H'\{(x).P\}$ implies that $H'\{(x).P\} \Downarrow n$ with the same depth of inference as $R \Downarrow n$. By the induction hypothesis, we get $H'\{(x).P'\} \Downarrow n$ too. From $H \rightarrow H'$ we get that $H\{(x).P'\} \rightarrow H'\{(x).P'\}$, and hence that $H\{(x).P'\} \Downarrow n$.
- **(Act Inter)** Then $H \bullet (x).P \rightsquigarrow R$. By analysing the rules of interaction, $H \bullet (x).P \rightsquigarrow R$ can only be derived using (Inter Input) given that $H \equiv (v\vec{r})H'\{- \mid \langle M\rangle\}$, $(x).P > (v\vec{p})\langle(x).P_1\rangle P_2$ and $R \equiv (v\vec{r})H'\{(v\vec{p})(P_1\{x \leftarrow M\} \mid P_2)\}$, with $\{\vec{p}\} \cap fn(M) = \emptyset$ and $\{\vec{r}\} \cap fn(P) = \emptyset$. From $(x).P > (v\vec{p})\langle(x).P_1\rangle P_2$, it follows that $\vec{p} = \emptyset$, $P_1 = P$, $P_2 = \mathbf{0}$. Therefore, $R \equiv (v\vec{r})H'\{P\{x \leftarrow M\}\}$. We have that $(v\vec{r})H'\{P\{x \leftarrow M\}\} \Downarrow n$. By assumption, this implies that $(v\vec{r})H'\{P'\{x \leftarrow M\}\} \Downarrow n$. Now, $H\{(x).P'\} \equiv (v\vec{r})H'\{(x).P' \mid \langle M\rangle\} \rightarrow (v\vec{r})H'\{P'\{x \leftarrow M\}\}$. Therefore, $H\{(x).P'\} \Downarrow n$.

Proposition 63. If $P \sim P'$, then $(x).P \sim (x).P'$.

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

Proof. Consider any substitution σ with $dom(\sigma) = fv((x).P) \cup fv((x).P')$, that is, $dom(\sigma) = (fv(P) \cup fv(P')) - \{x\}$. From $P \sim P'$ it follows that $P\sigma \sim P'\sigma$ and that $fv(P\sigma) \cup fv(P'\sigma) \subseteq \{x\}$. Consider any closed harness H and any name n. By Lemma 62, we get $H\{(x).P\sigma)\} \Downarrow n$ if and only if $H\{(x).P'\sigma)\} \Downarrow n$. Hence, $(x).P \sim (x).P'$. □

Proposition 64. If $P \sim P'$, then $\mathscr{C}(P) \sim \mathscr{C}(Q)$.

Proof. The proof follows by combining Propositions 50, 51, 53, 55, 59, 61 and 63. □

Finally, we prove that the relations $P \sim Q$ and $P \simeq Q$ are one.

Proposition 65. If $P \sim Q$, then $P \simeq Q$.

Proof. We must show for all names n and contexts $\mathscr{C}$ with $\mathscr{C}(P)$ and $\mathscr{C}(Q)$ closed, that $\mathscr{C}(P) \Downarrow n \Leftrightarrow \mathscr{C}(Q) \Downarrow n$, assuming that $P \sim Q$. By Proposition 64, $P \sim Q$ implies that $\mathscr{C}(P) \sim \mathscr{C}(Q)$. Therefore $\mathscr{C}(P) \Downarrow n \Leftrightarrow \mathscr{C}(Q) \Downarrow n$ follows from the definition of $\mathscr{C}(P) \sim \mathscr{C}(Q)$, given that $\mathscr{C}(P)$ and $\mathscr{C}(Q)$ are closed. □

To show the converse implication, we need the following combinator.

A substitution combinator: *subst x M P*

$$subst\ x\ M\ P \triangleq (vm)(vn)(open\ n \mid m[\langle M\rangle \mid (x).n[out\ m.open\ m.P]]) \quad \text{for } \{m,n\} \cap fn(M.P) = \emptyset$$

Lemma 66. For all P and M, *subst x M P* $\sim P\{x \leftarrow M\}$.

Proof. Consider the processes defined by the following, where $\{m,n\} \cap fn(M.P) = \emptyset$.

$$\begin{aligned} R_1 &\triangleq (vm)(vn)(open\ n \mid m[\langle M\rangle \mid (x).n[out\ m.open\ m.P]]) \\ R_2 &\triangleq (vm)(vn)(open\ n \mid m[n[out\ m.open\ m.P\{x \leftarrow M\}]]) \\ R_3 &\triangleq (vm)(vn)(open\ n \mid n[open\ m.P\{x \leftarrow M\}] \mid m[]) \\ R_4 &\triangleq (vm)(open\ m.P\{x \leftarrow M\} \mid m[]) \\ R_5 &\triangleq P\{x \leftarrow M\}. \end{aligned}$$

We will omit the details, but using the activity lemma we can show that $R_i \sim R_{i+1}$ for $i \in 1..4$, much as in the proof of Lemma 19. By transitivity, we obtain $R_1 \sim R_5$, that is, *subst x M P* $\sim P\{x \leftarrow M\}$. □

Lemma 67. If $P \simeq Q$, then $P\{x \leftarrow M\} \simeq Q\{x \leftarrow M\}$.

Proof. From $P \simeq Q$ it follows that *subst x M P* $\simeq$ *subst x M Q*. By Lemma 66 and Proposition 65, we get that *subst x M P* $\simeq P\{x \leftarrow M\}$ and *subst x M Q* $\simeq Q\{x \leftarrow M\}$. Combining these equations yields $P\{x \leftarrow M\} \simeq Q\{x \leftarrow M\}$. □

Proposition 68. If $P \simeq Q$, then $P \sim Q$.

Proof. Suppose $P \simeq Q$. Consider any substitution σ with $dom(\sigma) = fv(P) \cup fv(Q)$, and any closed harness H and name n. By Lemma 67, $P \simeq Q$ implies that $P\sigma \simeq Q\sigma$. Since $\simeq$ is a congruence, Proposition 1, we get that $H\{P\sigma\} \simeq H\{Q\sigma\}$. By definition of $H\{P\sigma\} \simeq$

https://doi.org/10.1017/S0960129502003742 Published online by Cambridge University Press

$H\{Q\sigma\}$, the fact that $H\{P\sigma\}$ and $H\{Q\sigma\}$ are closed implies that $H\{P\sigma\} \Downarrow n \Leftrightarrow H\{Q\sigma\} \Downarrow n$. Therefore $P \sim Q$. □

Proof of Theorem 12. For all processes P and Q, $P \simeq Q$ if and only if for all substitutions σ with $dom(\sigma) = fv(P) \cup fv(Q)$, and for all closed harnesses H and names n, that $H\{P\sigma\} \Downarrow n \Leftrightarrow H\{Q\sigma\} \Downarrow n$.

Proof. By definition of $P \sim Q$, this is equivalent to showing that $P \simeq Q$ if and only if $P \sim Q$, for all P and Q, which follows from Propositions 65 and 68. □

Acknowledgement

Comments by Giuseppe Castagna, Cédric Fournet, Georges Gonthier, Tony Hoare and Jan Vitek were helpful. We thank the anonymous referees for their very thorough reading of the manuscript.

References

Abadi, M., Fournet, C. and Gonthier, G. (1998) Secure communications implementation of channel abstractions. *13th IEEE Symposium on Logic in Computer Science (LICS'98)* 105–116.

Abadi, M. and Gordon, A. D. (1999) A calculus for cryptographic protocols: The spi calculus. *Information and Computation* **148** 1–70.

Berry, G. and Boudol, G. (1992) The chemical abstract machine. *Theoretical Computer Science* **96** (1) 217–248.

Cardelli, L. (1999) Abstractions for mobile computation. In: Jensen, C. and Vitek, J. (eds.) Secure Internet Programming: Issues in Distributed and Mobile Object Systems. *Springer-Verlag Lecture Notes in Computer Science* **1603** 51–94.

Cardelli, L. and Gordon, A. D. (1999) Types for mobile ambients. *26th ACM Symposium on Principles of Programming Languages (POPL'99)* 79–92.

Cardelli, L. and Gordon, A. D. (2000a) Anytime, anywhere: Modal logics for mobile ambients. *27th ACM Symposium on Principles of Programming Languages (POPL'00)* 365–377.

Cardelli, L. and Gordon, A. D. (2000b) Mobile ambients. *Theoretical Computer Science* **240** 177–213.

De Nicola, R. and Hennessy, M. (1984) Testing equivalences for processes. *Theoretical Computer Science* **34** 83–133.

Fournet, C. and Gonthier, G. (1996) The reflexive CHAM and the Join-calculus. *23rd ACM Symposium on Principles of Programming Languages (POPL'96)* 372–385.

Fournet, C., Lévy, J.-J. and Schmitt, A. (2000) An asynchronous distributed implementation of mobile ambients. In: IFIP International Conference on Theoretical Computer Science (IFIP TCS 2000). *Springer-Verlag Lecture Notes in Computer Science* **1872** 348–364.

Gordon, A. D. and Cardelli, L. (1999) Equational properties of mobile ambients. Foundations of Software Science and Computation Structures (FoSSaCS'99). *Springer-Verlag Lecture Notes in Computer Science* **1578** 212–226. (An extended version appears as Microsoft Research Technical Report MSR–TR–99–11, April 1999.)

Levi, F. and Sangiorgi, D. (2000) Controlling interference in ambients. *27th ACM Symposium on Principles of Programming Languages (POPL'00)* 352–364.

Merro, M. and Hennessy, M. (2002) Bisimulation congruences in safe ambients. *29th ACM Symposium on Principles of Programming Languages (POPL'02)* 71–80.

Milner, R. (1977) Fully abstract models of typed lambda-calculi. *Theoretical Computer Science* **4** 1–23.

Milner, R. (1999) *Communicating and Mobile Systems: the π-Calculus*, Cambridge University Press.

Morris, J. H. (1968) *Lambda-Calculus Models of Programming Languages*, Ph. D. thesis, MIT.

Plotkin, G. D. (1977) LCF considered as a programming language. *Theoretical Computer Science* **5** 223–255.

Sangiorgi, D. (2001) Extensionality and intensionality of the ambient logics. *28th ACM Symposium on Principles of Programming Languages (POPL'01)* 4–13.

Sangiorgi, D. and Walker, D. (2001) *The π-calculus: a Theory of Mobile Processes*, Cambridge University Press.

Vigliotti, M. G. (1999) *Transition systems for the ambient calculus*, Master's thesis, Imperial College of Science, Technology and Medicine.

Vitek, J. and Castagna, G. (1999) Seal: A framework for secure mobile computations. In: Internet Programming Languages. *Springer-Verlag Lecture Notes in Computer Science* **1686** 47–77.