Mobile Ambients
–Annex–

Luca Cardelli
Digital Equipment Corporation
Systems Research Center
luca@pa.dec.com

Andrew D. Gordon
University of Cambridge
Computer Laboratory
adg@cl.cam.ac.uk

June 23, 1997

Abstract

We develop some theory of the ambient calculus, based on a labeled transition system, strong bisimilarity and contextual equivalence.

The goal of this Annex to the main paper is to recast notions of observational equivalence from the π-calculus in the setting of the ambient calculus, so as to reason about mobile computation. In Section 1 we review the technical definitions of the ambient calculus from the main part of the paper. In Section 2 we define the *commitment relation*, a labeled transition system for the calculus, and *strong bisimilarity*. In Section 3 we define *contextual equivalence* and prove an instance, the Perfect Firewall Equation. Section 4 concludes.

1 Review of the Ambient Calculus

In this section we review the syntax of the ambient calculus, and the structural congruence and reduction relations.

1.1 Capabilities and Processes

We assume an infinite set of *names*. We let m and n range over names. Moreover, we assume there is an infinite collections of *variables* ranged over by metavariables x, y, z. The sets of *capabilities* and *processes* are defined by the grammars:

Mobility and Communication Primitives

$M ::=$	capability
$\quad x$	$\quad$ variable
$\quad n$	$\quad$ name
$\quad \mathit{in}\ M$	$\quad$ can enter into M
$\quad \mathit{out}\ M$	$\quad$ can exit out of M
$\quad \mathit{open}\ M$	$\quad$ can open M
$\quad \epsilon$	$\quad$ null
$\quad M.M'$	$\quad$ path
$P, Q, R ::=$	process
$\quad (\nu n)P$	$\quad$ restriction
$\quad \mathbf{0}$	$\quad$ inactivity
$\quad P \mid Q$	$\quad$ composition
$\quad !P$	$\quad$ replication
$\quad M[P]$	$\quad$ ambient
$\quad M.P$	$\quad$ action
$\quad (x).P$	$\quad$ input action
$\quad \langle M \rangle$	$\quad$ async output action

The general forms $\mathit{in}\ M$, $\mathit{out}\ M$ and $\mathit{open}\ M$ 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 $\mathit{fn}(M)$ and $\mathit{fv}(M)$ of *free names* and *free variables* of a capability M, and the sets $\mathit{fn}(P)$ and $\mathit{fv}(P)$ of *free names* and *free variables* of a process P.

Free Names and Variables of Capabilities and Processes

$\mathit{fn}(x) = \varnothing$	$\mathit{fv}(x) = \{x\}$
$\mathit{fn}(n) = \{n\}$	$\mathit{fv}(n) = \varnothing$
$\mathit{fn}(\mathit{in}\ M) = \mathit{fn}(M)$	$\mathit{fv}(\mathit{in}\ M) = \mathit{fv}(M)$
$\mathit{fn}(\mathit{out}\ M) = \mathit{fn}(M)$	$\mathit{fv}(\mathit{out}\ M) = \mathit{fv}(M)$
$\mathit{fn}(\mathit{open}\ M) = \mathit{fn}(M)$	$\mathit{fv}(\mathit{open}\ M) = \mathit{fv}(M)$
$\mathit{fn}(\epsilon) = \varnothing$	$\mathit{fv}(\epsilon) = \varnothing$
$\mathit{fn}(M.M') = \mathit{fn}(M) \cup \mathit{fn}(M')$	$\mathit{fv}(M.M') = \mathit{fv}(M) \cup \mathit{fv}(M')$
$\mathit{fn}((\nu n)P) = \mathit{fn}(P) - \{n\}$	$\mathit{fv}((\nu n)P) = \mathit{fv}(P)$
$\mathit{fn}(\mathbf{0}) = \varnothing$	$\mathit{fv}(\mathbf{0}) = \varnothing$
$\mathit{fn}(P \mid Q) = \mathit{fn}(P) \cup \mathit{fn}(Q)$	$\mathit{fv}(P \mid Q) = \mathit{fv}(P) \cup \mathit{fv}(Q)$
$\mathit{fn}(!P) = \mathit{fn}(P)$	$\mathit{fv}(!P) = \mathit{fv}(P)$

$fn(M[P]) = fn(M) \cup fn(P)$	$fv(M[P]) = fv(M) \cup fv(P)$
$fn(M.P) = fn(M) \cup fn(P)$	$fv(M.P) = fv(M) \cup fv(P)$
$fn((x).P) = fn(P)$	$fv((x).P) = fv(P) - \{x\}$
$fn(\langle M \rangle) = fn(M)$	$fv(\langle M \rangle) = fv(M)$

We say a capability M is *closed* if and only if $fv(M) = \varnothing$; similarly, a process P is *closed* if and only if $fv(M) = \varnothing$.

If phrase ϕ is a capability or a process, we write $\phi\{x \leftarrow M\}$ for the outcome of a capture-avoiding substitution of the capability M for each free occurrence of the variable x in ϕ. We identify processes up to renaming of bound names and variables: $(\nu n)P = (\nu m)(P\{n \leftarrow m\})$ if $m \notin fn(P)$; $(x).P = (y).(P\{x \leftarrow y\})$ if $y \notin fn(P)$.

1.2 Structural Congruence

We let *structural congruence*, $\equiv$, be the least relation on processes that satisfies the following equations and rules:

Structural Congruence

$P \equiv P$	(Struct Refl)
$Q \equiv P \Rightarrow P \equiv Q$	(Struct Symm)
$P \equiv Q, Q \equiv R \Rightarrow P \equiv R$	(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 .)
$\rightarrow^*$	reflexive, transitive closure of $\rightarrow$

1.3 Reduction

We let the *reduction relation*, $\rightarrow$, be the least relation on processes to satisfy the following rules:

Reduction

$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$	(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 $\equiv$)

1.4 Basic Properties

Lemma 1 *If $P \equiv Q$ then $fn(P) = fn(Q)$.*

Proof By induction on the derivation of $P \equiv Q$. $\Box$

Lemma 2 *If $P \equiv Q$ then $P\{x \leftarrow M\} \equiv Q\{x \leftarrow M\}$.*

Proof By induction on the derivation of $P \equiv Q$. $\Box$

Lemma 3 *If $P \rightarrow Q$ then $fn(Q) \subseteq fn(P)$.*

Proof By induction on the derivation of $P \rightarrow Q$. $\Box$

2 Commitment and Strong Bisimilarity

This section introduces a labeled transition system, the *commitment relation*, for the ambient calculus and reviews the *strong bisimilarity* relation—a standard notion of behavioral equivalence—induced by commitment. The use of structural congruence to define reduction results in an elegant and compact definition, but also complicates proofs about reduction. As in the π-calculus [1, 4, 7], the purpose of the commitment relation is to characterise reduction independently of structural congruence.

Section 2.1 introduces the commitment relation. In π-calculus jargon our relation is an "early" transition system; we discuss how to obtain a "late" variant of our transition system in Section 2.2. In Section 2.3, we define and prove basic properties of the form of strong bisimilarity induced by the commitment relation. Finally, in Section 2.4 we prove that the commitment relation corresponds precisely to the reduction relation of Section 1.3.

2.1 Commitment

This section presents the commitment relation. It is a relation indexed by a set of *actions*, ranged over by α, given as follows.

Actions

$\alpha ::=$	actions
$\quad \tau$	internal action
$\quad \mathit{in}\ n$	enter ambient n
$\quad \mathit{enter}\ n\ A$	invite sibling A to enter
$\quad \overline{\mathit{enter}\ n}$	move ambient into sibling n
$\quad \mathit{out}\ n$	exit ambient n
$\quad \mathit{exit}\ n$	exit parent ambient
$\quad \mathit{open}\ n$	open ambient n
$\quad \overline{\mathit{open}\ n}$	open self
$\quad \mathit{anon}\ M$	anonymous input of M
$\quad \overline{\mathit{anon}}$	anonymous output

The commitment relation relates a process to an *outcome*, O, which is either a process or a *concretion*. Here are the grammars for ambients, concretions and outcomes.

Ambients, Concretions, Outcomes

$A ::= n[P]$	ambients
$C, D ::= (\nu n)C \mid \langle A\rangle P \mid \langle M\rangle P$	concretions
$O ::= P \mid C$	outcomes

In $(\nu n)C$, the name n is bound in C. We extend the notions of free names, free variables and substitution to concretions in the obvious way. We identify concretions up to renaming of bound names: $(\nu n)C = (\nu m)C\{n \leftarrow m\}$ if $m \notin \mathit{fn}(C)$.

The commitment relation is written as follows:

The Commitment Relation

$P \xrightarrow{\alpha} O$	P α-commits to outcome O

The commitment relation is the union of ten different judgments, one for each of the different kinds of action. Judgments of the form $P \xrightarrow{\tau} Q$ indicate that process P proceeds to Q by an internal computation step. We will later prove that this judgment corresponds to the reduction relation on processes. The role of the other nine judgments making up the commitment relation is to represent intermediate stages in the derivation of a τ-commitment.

We introduce the rules that inductively define the commitment relation in groups, beginning with a group of congruence rules. As a convenient shorthand for writing (Comm Par 1) and (Comm Par 2), we define the *composition*, $O \parallel O'$ of outcomes O and O', at least one of which is a process, as follows, where ϕ is either an ambient or a capability:

$$\begin{array}{rcll} P \parallel Q & \triangleq & P \mid Q & \\ ((\nu\vec{m})\langle\phi\rangle P) \parallel Q & \triangleq & (\nu\vec{m})\langle\phi\rangle(P \mid Q) & \text{if } \{\vec{m}\} \cap \mathit{fn}(Q) = \varnothing \\ P \parallel ((\nu\vec{m})\langle\phi\rangle Q) & \triangleq & (\nu\vec{m})\langle\phi\rangle(P \mid Q) & \text{if } \{\vec{m}\} \cap \mathit{fn}(P) = \varnothing \end{array}$$

Congruence

(Comm Amb)
$$\frac{P \xrightarrow{\tau} Q}{n[P] \xrightarrow{\tau} n[Q]}$$

(Comm Res)
$$\frac{P \xrightarrow{\alpha} O \quad n \notin \mathit{fn}(\alpha)}{(\nu n)P \xrightarrow{\alpha} (\nu n)O}$$

(Comm Par 1)
$$\frac{P \xrightarrow{\alpha} O}{P \mid Q \xrightarrow{\alpha} O \parallel Q}$$

(Comm Par 2)
$$\frac{Q \xrightarrow{\alpha} O}{P \mid Q \xrightarrow{\alpha} P \parallel O}$$

The next three groups of rules explain the behavior of processes of the form *in* $n.P$, *out* $n.P$ and *open* $n.P$.

Entering an Ambient

(Comm *in*)
$$\frac{}{\mathit{in}\; n.P \xrightarrow{\mathit{in}\; n} P}$$

(Comm $\overline{\mathit{enter}}$)
$$\frac{P \xrightarrow{\mathit{in}\; n} Q}{m[P] \xrightarrow{\overline{\mathit{enter}\; n}} \langle m[Q]\rangle \mathbf{0}}$$

(Comm *enter*)
$$\frac{}{n[P] \xrightarrow{\mathit{enter}\; n\; A} n[A \mid P]}$$

(Comm τ-*enter* 1) (where $\{\vec{m}\} \cap \mathit{fn}(Q) = \varnothing$)
$$\frac{P \xrightarrow{\overline{\mathit{enter}\; n}} (\nu\vec{m})\langle A\rangle P' \quad Q \xrightarrow{\mathit{enter}\; n\; A} Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P' \mid Q')}$$

(Comm τ-*enter* 2) (where $\{\vec{m}\} \cap \mathit{fn}(P) = \varnothing$)
$$\frac{P \xrightarrow{\mathit{enter}\; n\; A} P' \quad Q \xrightarrow{\overline{\mathit{enter}\; n}} (\nu\vec{m})\langle A\rangle Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P' \mid Q')}$$

We begin an enumeration of the nine judgments that represent intermediate stages in deriving τ-commitments, with the three used in the rules for entering an ambient:

(1) $P \xrightarrow{in\, n} Q$: process P instructs its enclosing ambient to move into sibling ambient n.

(2) $P \xrightarrow{enter\, n\, A} Q$: an ambient n within P is ready to accept entry of a sibling ambient, A; in this event, the residue of P is Q.

(3) $P \xrightarrow{\overline{enter\, n}} (\nu\vec{m})\langle A\rangle Q$: an ambient A within P is ready to enter a sibling ambient n; in this event, process Q is the residue of P.

Exiting an Ambient

(Comm *out*)
$$\frac{}{out\; n.P \xrightarrow{out\, n} P}$$

(Comm *exit*)
$$\frac{P \xrightarrow{out\, n} Q}{m[P] \xrightarrow{exit\, n} \langle m[Q]\rangle \mathbf{0}}$$

(Comm τ-*exit*) (where $n \notin \{\vec{m}\}$)
$$\frac{P \xrightarrow{exit\, n} (\nu\vec{m})\langle A\rangle Q}{n[P] \xrightarrow{\tau} (\nu\vec{m})(A \mid n[Q])}$$

Apart from τ-commitments, the rules above allow the derivation of the following judgments:

(4) $P \xrightarrow{out\, n} Q$: process P instructs its enclosing ambient to move out of its parent ambient n.

(5) $P \xrightarrow{exit\, n} (\nu\vec{n})\langle A\rangle Q$: ambient A within P is ready to exit its parent ambient n; in this event, process Q is the residue of P.

Opening an Ambient

(Comm *open*)
$$\frac{}{open\; n.P \xrightarrow{open\, n} P}$$

(Comm $\overline{open}$)
$$\frac{}{n[P] \xrightarrow{\overline{open\, n}} P}$$

(Comm τ-*open* 1)
$$\frac{P \xrightarrow{\overline{open\, n}} P' \quad Q \xrightarrow{open\, n} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

(Comm τ-*open* 2)
$$\frac{P \xrightarrow{open\, n} P' \quad Q \xrightarrow{\overline{open\, n}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Apart from τ-commitments, the rules above allow the derivation of the following judgments:

(6) $P \xrightarrow{open\, n} Q$: process P opens a sibling ambient n.

(7) $P \xrightarrow{\overline{open\, n}} Q$: an ambient n within P is opened.

The following group of rules explains the behavior of processes of the form $\langle M \rangle$ and $(x).P$.

Anonymous I/O

(Comm $anon$)

$$\frac{}{(x).P \xrightarrow{anon\, M} P\{x \leftarrow M\}}$$

(Comm $\overline{anon}$)

$$\frac{}{\langle M \rangle \xrightarrow{\overline{anon}} \langle M \rangle \mathbf{0}}$$

(Comm τ-$anon$ 1) (where $\{\vec{m}\} \cap fn(Q) = \varnothing$)

$$\frac{P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M \rangle P' \quad Q \xrightarrow{anon\, M} Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P' \mid Q')}$$

(Comm τ-$anon$ 2) (where $\{\vec{m}\} \cap fn(P) = \varnothing$)

$$\frac{P \xrightarrow{anon\, M} P' \quad Q \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M \rangle Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P' \mid Q')}$$

We can now conclude our enumeration of the nine judgments that represent intermediate stages in the derivation of a τ-commitment:

(8) $P \xrightarrow{anon\, M} Q$: a process within P is ready to input a capability, M; in this event, the residue of P is Q.

(9) $P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M \rangle Q$: a process within P is ready to output capability M; in this event, process Q is the residue of P.

Finally, we complete the definition of the commitment relation with two groups of commitment rules that explain the behavior of processes of the form $(M.M').P$, $\epsilon.P$ and $!P$:

Capability Sequencing

(Comm ϵ)

$$\frac{P \xrightarrow{\alpha} O}{\epsilon.P \xrightarrow{\alpha} O}$$

(Comm .)

$$\frac{M.(M'.P) \xrightarrow{\alpha} O}{(M.M').P \xrightarrow{\alpha} O}$$

Replication

(Comm Repl)

$$\frac{P \mid !P \xrightarrow{\alpha} O}{!P \xrightarrow{\alpha} O}$$

Commitment enjoys the following basic properties:

Lemma 4 *Whenever $P \xrightarrow{\alpha} O$:*

(1) *If $\alpha = enter\ n\ A$ then $fn(O) = fn(P) \cup fn(A)$ and $n \in fn(P)$.*

(2) *If $\alpha = anon M$ then $fn(O) \subseteq fn(P) \cup fn(M)$.*

(3) *Otherwise, $fn(\alpha) \cup fn(O) \subseteq fn(P)$.*

Proof By induction on the derivation of $P \xrightarrow{\alpha} O$. $\square$

Lemma 5 *$(\nu n)P \xrightarrow{\alpha} O$ if and only if there is Q with $P \xrightarrow{\alpha} Q$, $O = (\nu n)Q$ and $n \notin fn(\alpha)$.*

Proof (Comm Res) asserts the right-to-left direction. For the left-to-right direction, only (Comm Res) may derive $(\nu n)P \xrightarrow{\alpha} O$; therefore O can only take the form $(\nu n)Q$ where $P \xrightarrow{\alpha} Q$ and $n \notin fn(\alpha)$. $\square$

Lemma 6 *$P \mid Q \xrightarrow{\alpha} O$ if and only if one of the following holds:*

(1) *$O = O' \parallel Q$ where $P \xrightarrow{\alpha} O'$.*

(2) *$O = P \parallel O'$ where $Q \xrightarrow{\alpha} O'$.*

(3) *$O = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$*
where $P \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P'$ and $Q \xrightarrow{enter\ n\ A} Q'$ with $\{\vec{m}\} \cap fn(Q) = \varnothing$.

(4) *$O = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$*
where $P \xrightarrow{enter\ n\ A} P'$ and $Q \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle Q'$ with $\{\vec{m}\} \cap fn(P) = \varnothing$.

(5) *$O = P' \mid Q'$ and $\alpha = \tau$*
where $P \xrightarrow{\overline{open\ n}} P'$ and $Q \xrightarrow{open\ n} Q'$.

(6) *$O = P' \mid Q'$ and $\alpha = \tau$*
where $P \xrightarrow{open\ n} P'$ and $Q \xrightarrow{\overline{open\ n}} Q'$.

(7) $O = (\nu\vec{m})(P' \mid Q')$ *and* $\alpha = \tau$
where $P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P'$ *and* $Q \xrightarrow{anon\,M} Q'$ *with* $\{\vec{m}\} \cap fn(Q) = \varnothing$.

(8) $O = (\nu\vec{m})(P' \mid Q')$ *and* $\alpha = \tau$
where $P \xrightarrow{anon\,M} P'$ *and* $Q \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle Q'$ *with* $\{\vec{m}\} \cap fn(P) = \varnothing$.

Proof By inspection of the commitment rules. The eight possibilities correspond to rules (Comm Par 1), (Comm Par 2), (Comm τ-*enter* 1), (Comm τ-*enter* 2), (Comm τ-*open* 1), (Comm τ-*open* 2), (Comm τ-*anon* 1) and (Comm τ-*anon* 2), in that order. □

2.2 Early versus Late

The commitment relation of the previous section is said to be an early commitment relation because of its treatment of incoming data in judgment (2), $P \xrightarrow{enter\,n\,A} Q$, and judgment (8), $P \xrightarrow{anon\,M} Q$. Such judgments must originate with rules (Comm *enter*) and (Comm *anon*) respectively. Rule (Comm *enter*) allows an arbitrary incoming ambient A in a judgment $P \xrightarrow{enter\,n\,A} Q$, irrespective of whether A may actually be provided by the environment. Rules (Comm τ-*enter* 1) and (Comm τ-*enter* 2) only allow $P \xrightarrow{enter\,n\,A} P'$ to contribute to a τ-commitment if a judgment $Q \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle Q'$ proves that the ambient A is in fact present in the environment. Similarly, (Comm *anon*) chooses the incoming capability M irrespective of whether M may actually be provided by the environment. This is known as 'early instantiation' [5].

The contrast is with 'late instantiation' where incoming data is represented by variables which are later instantiated to actual data. To obtain a late version of our commitment relation, we could replace our early I/O rules with the following:

Late I/O

(Comm Late *anon*) (Comm Late $\overline{anon}$)

$$\frac{}{(x).P \xrightarrow{anon} (x)P} \qquad \frac{}{\langle M\rangle \xrightarrow{\overline{anon}} \langle M\rangle\mathbf{0}}$$

(Comm Late τ-*anon* 1) (where $\{\vec{m}\} \cap fn(P) = \varnothing$)

$$\frac{P \xrightarrow{anon} (x)P' \quad Q \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P'\{x \leftarrow M\} \mid Q')}$$

(Comm Late τ-*anon* 2) (where $\{\vec{m}\} \cap fn(Q) = \varnothing$)

$$\frac{P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P' \quad Q \xrightarrow{anon} (x)Q'}{P \mid Q \xrightarrow{\tau} (\nu\vec{m})(P' \mid Q'\{x \leftarrow M\})}$$

The phrase $(x)P$ in (Comm Late *anon*) is called an *abstraction*. The bound variable x is a placeholder that is instantiated to the incoming capability by rule (Comm Late τ-*anon* 1) or (Comm Late τ-*anon* 2). There are late versions of the rules for ambient entry, which need abstractions of the form $(X)P$ where X is a variable standing for ambients. We use an early rather than a late commitment relation simply to avoid introducing abstractions and variables standing for ambients.

2.3 Strong Bisimilarity

Strong bisimilarity [3, 5, 6] is a standard form of operational equivalence that may readily be adapted to any labeled transition system. To adapt it to our commitment relation, we begin with auxiliary definitions given in the following two tables.

First, let *structural congruence of concretions*, $\equiv$, be the smallest relation given by the following rules:

Structural Congruence of Concretions

$(\nu n)(\nu m)C \equiv (\nu m)(\nu n)C$	(Conc Res Res)
$n \notin fn(P) \Rightarrow (\nu n)\langle\phi\rangle P \equiv \langle(\nu n)\phi\rangle P$	(Conc Res 1)
$n \notin fn(\phi) \Rightarrow (\nu n)\langle\phi\rangle P \equiv \langle\phi\rangle(\nu n)P$	(Conc Res 2)
$C \equiv C$	(Conc Refl)
$C \equiv D \Rightarrow D \equiv C$	(Conc Symm)
$C \equiv C', C' \equiv C'' \Rightarrow C \equiv C''$	(Conc Trans)
$C \equiv D \Rightarrow (\nu n)C \equiv (\nu n)D$	(Conc Res)
$A \equiv B, P \equiv Q \Rightarrow \langle A\rangle P \equiv \langle B\rangle Q$	(Conc Amb)
$P \equiv Q \Rightarrow \langle M\rangle P \equiv \langle M\rangle Q$	(Conc Capa)

Lemma 7 *If $C \equiv D$ then $fn(C) = fn(D)$.*

Proof By induction on the derivation of $C \equiv D$. □

Second, if $\mathcal{S}$ is a relation on processes, let $\overline{\mathcal{S}}$ be the relation on outcomes given by the following rules:

Relation $\overline{\mathcal{S}}$ on Outcomes

(Outcome Process)	(Outcome Conc Amb)	(Outcome Conc Capa)
$\dfrac{P \; \mathcal{S} \; Q}{P \; \overline{\mathcal{S}} \; Q}$	$\dfrac{A \; \mathcal{S} \; B \qquad P \; \mathcal{S} \; Q}{(\nu\vec{n})\langle A\rangle P \; \overline{\mathcal{S}} \; (\nu\vec{n})\langle B\rangle Q}$	$\dfrac{P \; \mathcal{S} \; Q}{(\nu\vec{n})\langle M\rangle P \; \overline{\mathcal{S}} \; (\nu\vec{n})\langle M\rangle Q}$

Now we introduce strong bisimilarity:

- We write $O \equiv O'$ to mean that outcomes O and O' are either related by structural congruence of concretions or structural congruence of processes.
- If $\mathcal{S}$ is a relation on processes, let $\equiv\overline{\mathcal{S}}\equiv$ be the relation on outcomes such that $O_1 \equiv\overline{\mathcal{S}}\equiv O_2$ if and only if there are outcomes O_1' and O_2' such that $O_1 \equiv O_1'$, $O_1' \; \overline{\mathcal{S}} \; O_2'$ and $O_2' \equiv O_2$.
- Let a relation $\mathcal{S}$ on closed processes be a *strong simulation* if and only if $P \; \mathcal{S} \; Q$ implies that whenever $P \xrightarrow{\alpha} O$ there is O' such that $Q \xrightarrow{\alpha} O'$ and $O \equiv\overline{\mathcal{S}}\equiv O'$.
- Let a relation $\mathcal{S}$ on closed processes be a *strong bisimulation* if and only if both $\mathcal{S}$ and its converse, $\mathcal{S}^{-1}$, are strong simulations.
- Let *strong bisimilarity*, $\sim$, be the the union of all strong bisimulations.

By standard arguments, strong bisimilarity is the greatest strong bisimulation, and is an equivalence relation [3].

2.4 Relating Reduction and Commitment

The goal of this section is to prove Proposition 15, which asserts that the τ-commitment relation corresponds to the reduction relation in the sense that $P \to Q$ if and only if there is R with $P \xrightarrow{\tau} R$ and $R \equiv Q$.

We begin with a lemma proving the easier of the two directions of Proposition 15, that any τ-commitment implies a reduction step.

Lemma 8

(1) *If* $P \xrightarrow{M} Q$
then $P \equiv (\nu\vec{m})(P_1 \mid M.P_2)$ *with* $fn(M) \cap \{\vec{m}\} = \varnothing$
and $Q \equiv (\nu\vec{m})(P_1 \mid P_2)$.

(2) *If* $P \xrightarrow{enter\ n\ A} Q$
then $P \equiv (\nu\vec{m})(P_1 \mid n[P_2])$ *with* $n \notin \{\vec{m}\}$ *and* $fn(A) \cap \{\vec{m}\} = \varnothing$
and $Q \equiv (\nu\vec{m})(P_1 \mid n[A \mid P_2])$.

(3) *If* $P \xrightarrow{\overline{enter\, n}} (\nu\vec{m})\langle m[R']\rangle Q$
then $P \equiv (\nu\vec{m})(m[R] \mid Q)$ *with* $n \notin \{\vec{m}\}$ *and* $R \xrightarrow{in\, n} R'$.

(4) *If* $P \xrightarrow{exit\, n} (\nu\vec{m})\langle m[R']\rangle Q$
then $P \equiv (\nu\vec{m})(m[R] \mid Q)$ *with* $n \notin \{\vec{m}\}$ *and* $R \xrightarrow{out\, n} R'$.

(5) *If* $P \xrightarrow{\overline{open\, n}} Q$
then $P \equiv (\nu\vec{m})(P_1 \mid n[P_2])$ *with* $n \notin \{\vec{m}\}$
and $Q \equiv (\nu\vec{m})(P_1 \mid P_2)$.

(6) *If* $P \xrightarrow{anon\, M} Q$
then $P \equiv (\nu\vec{m})(P_1 \mid (x).P_2)$ *with* $fn(M) \cap \{\vec{m}\} = \varnothing$
and $Q \equiv (\nu\vec{m})(P_1 \mid P_2\{x \leftarrow M\})$.

(7) *If* $P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle Q$
then $P \equiv (\nu\vec{m})(\langle M\rangle \mid Q)$.

(8) *If* $P \xrightarrow{\tau} Q$ *then* $P \to Q$.

Proof We prove the proposition with part (3) strengthened to assert that the depth of derivation of $R \xrightarrow{in\, n} R'$ is less than that of $P \xrightarrow{\overline{enter\, n}} (\nu\vec{m})\langle m[R']\rangle Q$, and part (4) strengthened to assert that the depth of derivation of $R \xrightarrow{out\, n} R'$ is less than that of $P \xrightarrow{exit\, n} (\nu\vec{m})\langle m[R']\rangle Q$. We prove this strengthened proposition by simultaneous inductions on the derivations of the commitment relations.

We show four of the rules for deriving τ-commitments:

(Comm τ-*enter* 1)

$$\frac{P \xrightarrow{\overline{enter\, n}} (\nu\vec{p})\langle A\rangle P' \quad Q \xrightarrow{enter\, n\, A} Q' \quad \{\vec{p}\} \cap fn(Q) = \varnothing}{P \mid Q \xrightarrow{\tau} (\nu\vec{p})(P' \mid Q')}$$

Suppose $A = m[R']$. By induction hypothesis, $P \xrightarrow{\overline{enter\, n}} (\nu\vec{p})\langle m[R']\rangle P'$ implies $P \equiv (\nu\vec{p})(m[R] \mid P')$ with $n \notin \{\vec{p}\}$ and $R \xrightarrow{in\, n} R'$, with depth of derivation less than that of $P \mid Q \xrightarrow{\tau} (\nu\vec{p})(P' \mid Q')$. By induction hypothesis, $Q \xrightarrow{enter\, n\, A} Q'$ implies that $Q \equiv (\nu\vec{q})(Q_1 \mid n[Q_2])$ with $n \notin \{\vec{q}\}$ and $fn(A) \cap \{\vec{q}\} = \varnothing$ and $Q' \equiv (\nu\vec{q})(Q_1 \mid n[A \mid Q_2])$. Therefore we may apply the induction hypothesis to $R \xrightarrow{in\, n} R'$ to obtain that $R \equiv (\nu\vec{r})(R_1 \mid in\, n.R_2)$ with $fn(in\, n) \cap \{\vec{r}\} = \varnothing$ and $R' \equiv (\nu\vec{r})(R_1 \mid R_2)$.

Since the names $\vec{p}$, $\vec{q}$ and $\vec{r}$ are bound, we may assume that they are all distinct and that $\{\vec{p}\} \cap fn(Q) = \varnothing$, $\{\vec{q}\} \cap fn(P) = \varnothing$ and $\{\vec{r}\} \cap (fn(P') \cup \{m\} \cup fn(Q)) = \varnothing$.

We may calculate:

$$\begin{aligned}
P \mid Q &\equiv (\nu\vec{p})(m[R] \mid P') \mid (\nu\vec{q})(Q_1 \mid n[Q_2]) \\
&\equiv (\nu\vec{p})(m[(\nu\vec{r})(R_1 \mid in\ n.R_2)] \mid P') \mid (\nu\vec{q})(Q_1 \mid n[Q_2]) \\
&\equiv (\nu\vec{p})(\nu\vec{q})(\nu\vec{r})(m[R_1 \mid in\ n.R_2] \mid P' \mid Q_1 \mid n[Q_2]) \\
&\equiv (\nu\vec{p})(\nu\vec{q})(\nu\vec{r})(P' \mid m[in\ n.R_2 \mid R_1] \mid n[Q_2] \mid Q_1)
\end{aligned}$$

The second last step is by appeal to (Struct Res Par), (Struct Res Amb) and (Struct Res Res) and the facts that $\{\vec{p}\} \cap fn(Q) = \varnothing$, $\{\vec{r}\} \cap (\{m\} \cup fn(Q)) = \varnothing$ and $\{\vec{q}\} \cap fn(P) = \varnothing$.

By (Red In) we get:

$$\begin{aligned}
&(\nu\vec{p})(\nu\vec{r})(\nu\vec{q})(P' \mid m[in\ n.R_2 \mid R_1] \mid n[Q_2] \mid Q_1) \\
&\quad \rightarrow (\nu\vec{p})(\nu\vec{r})(\nu\vec{q})(P' \mid n[m[R_2 \mid R_1] \mid Q_2] \mid Q_1)
\end{aligned}$$

We may calculate:

$$\begin{aligned}
&(\nu\vec{p})(\nu\vec{r})(\nu\vec{q})(P' \mid n[m[R_2 \mid R_1] \mid Q_2] \mid Q_1) \\
&\equiv (\nu\vec{p})(\nu\vec{r})(\nu\vec{q})(P' \mid Q_1 \mid n[m[R_1 \mid R_2] \mid Q_2]) \\
&\equiv (\nu\vec{p})(P' \mid (\nu\vec{q})(Q_1 \mid n[m[(\nu\vec{r})(R_1 \mid R_2)] \mid Q_2])) \\
&\equiv (\nu\vec{p})(P' \mid (\nu\vec{q})(Q_1 \mid n[m[R'] \mid Q_2])) \\
&\equiv (\nu\vec{p})(P' \mid Q')
\end{aligned}$$

The second step is by appeal to (Struct Res Par) and (Struct Res Amb) and the facts that $\{\vec{q}\} \cap fn(P') = \varnothing$ and $\{\vec{r}\} \cap (fn(P') \cup fn(Q_1) \cup \{m, n\}) = \varnothing$. That $\{\vec{q}\} \cap fn(P') = \varnothing$ follows from the facts that $fn(P') \subseteq fn(P) \cup \{\vec{p}\}$, $\{\vec{q}\} \cap fn(P) = \varnothing$ and $\{\vec{p}\} \cap \{\vec{q}\} = \varnothing$. That $\{\vec{r}\} \cap fn(Q_1) = \varnothing$ follows from $fn(Q_1) \subseteq fn(Q) \cup \{\vec{q}\}$, $\{\vec{r}\} \cap fn(Q) = \varnothing$, $\{\vec{q}\} \cap \{\vec{r}\} = \varnothing$. That $\{\vec{r}\} \cap \{m, n\} = \varnothing$ follows from our assumptions that $m \notin \{\vec{r}\}$ and $\{\vec{r}\} \cap fn(Q) = \varnothing$, and the fact that $Q \xrightarrow{enter\ n\ A} Q'$ implies that $n \in fn(Q)$, by Lemma 4(1).

Combining the calculations above, we get $P \mid Q \rightarrow (\nu\vec{p})(P' \mid Q')$ by (Red $\equiv$).

(Comm τ-*exit*)

$$\frac{P \xrightarrow{exit\ n} (\nu\vec{m})\langle A\rangle Q \quad n \notin \{\vec{m}\}}{n[P] \xrightarrow{\tau} (\nu\vec{m})(A \mid n[Q])}$$

Suppose $A = m[R']$. By induction hypothesis, $P \xrightarrow{exit\ n} (\nu\vec{m})\langle A\rangle Q$ implies there is R with $P \equiv (\nu\vec{m})(m[R] \mid Q)$ with $n \notin \{\vec{m}\}$ and $R \xrightarrow{out\ n} R'$, where the depth of derivation of the latter is less than that of $n[P] \xrightarrow{\tau} (\nu\vec{m})(A \mid n[Q])$. By induction hypothesis, $R \xrightarrow{out\ n} R'$ implies that $R \equiv (\nu\vec{r})(R_1 \mid out\ n.R_2)$ with $n \notin \{\vec{r}\}$.

Since the names $\vec{r}$ are bound, we may assume that $\{\vec{r}\}\cap(\{m\}\cup fn(Q)) = \varnothing$.

We may calculate:

$$\begin{aligned} n[P] &\equiv n[(\nu\vec{m})(m[(\nu\vec{r})(R_1 \mid out\ n.R_2)] \mid Q)] \\ &\equiv (\nu\vec{m})(\nu\vec{r})n[m[out\ n.R_2 \mid R_1] \mid Q] \end{aligned}$$

The second step is by appeal to (Struct Par Comm), (Struct Par Assoc), (Struct Res Par), (Struct Res Amb) and the facts that $n \notin \{\vec{m}\}$ and $\{\vec{r}\} \cap \{m, n\} = \varnothing$.

By (Red Out) we get:

$$(\nu\vec{m})(\nu\vec{r})n[m[out\ n.R_2 \mid R_1] \mid Q] \quad\rightarrow\quad (\nu\vec{m})(\nu\vec{r})(m[R_2 \mid R_1] \mid n[Q])$$

We may calculate:

$$(\nu\vec{m})(\nu\vec{r})(m[R_2 \mid R_1] \mid n[Q]) \quad\equiv\quad (\nu\vec{m})(m[(\nu\vec{r})(R_1 \mid R_2)] \mid n[Q])$$

using (Struct Par Comm), (Struct Par Assoc), (Struct Res Par), (Struct Res Amb) and the fact that $\{\vec{r}\} \cap (\{m, n\} \cap fn(Q)) = \varnothing$.

Combining the calculations above, we get $n[P] \rightarrow (\nu\vec{m})(A \mid n[Q])$ by (Red $\equiv$).

(Comm τ-*open* 1)

$$\frac{P \xrightarrow{\overline{open\ n}} P' \quad Q \xrightarrow{open\ n} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

By induction hypothesis, $P \xrightarrow{\overline{open\ n}} P'$ implies that $P \equiv (\nu\vec{p})(P_1 \mid n[P_2])$ with $n \notin \{\vec{p}\}$ and $P' \equiv (\nu\vec{p})(P_1 \mid P_2)$. By induction hypothesis, $Q \xrightarrow{open\ n} Q'$ implies that $Q \equiv (\nu\vec{q})(Q_1 \mid open\ n.Q_2)$ and $Q' \equiv (\nu\vec{q})(Q_1 \mid Q_2)$ with $n \notin \{\vec{q}\}$.

Since the names $\vec{p}$ and $\vec{q}$ are bound, we may assume that $\{\vec{p}\}\cap fn(Q_1) = \varnothing$ and $\{\vec{q}\} \cap fn(P_1) = \varnothing$.

We may calculate:

$$\begin{aligned} P \mid Q &\equiv (\nu\vec{p})(P_1 \mid n[P_2]) \mid (\nu\vec{q})(Q_1 \mid \mathit{open}\, n.Q_2) \\ &\equiv (\nu\vec{p})(\nu\vec{q})(\mathit{open}\, n.Q_2 \mid n[P_2] \mid P_1 \mid Q_1) \end{aligned}$$

The second step uses (Struct Par Comm), (Struct Par Assoc), (Struct Res Par) and the facts that $\{\vec{p}\} \cap \mathit{fn}(Q_1) = \varnothing$ and $\{\vec{q}\} \cap \mathit{fn}(P_1) = \varnothing$.

By (Red Open) we get:

$$(\nu\vec{p})(\nu\vec{q})(\mathit{open}\, n.Q_2 \mid n[P_2] \mid P_1 \mid Q_1) \;\rightarrow\; (\nu\vec{p})(\nu\vec{q})(Q_2 \mid P_2 \mid P_1 \mid Q_1)$$

We may calculate:

$$(\nu\vec{p})(\nu\vec{q})(Q_2 \mid P_2 \mid P_1 \mid Q_1) \;\equiv\; (\nu\vec{p})(P_1 \mid P_2) \mid (\nu\vec{q})(Q_1 \mid Q_2)$$

Again we are using (Struct Par Comm), (Struct Par Assoc), (Struct Res Par) and the facts that $\{\vec{p}\} \cap \mathit{fn}(Q_1) = \varnothing$ and $\{\vec{q}\} \cap \mathit{fn}(P_1) = \varnothing$.

Combining the calculations above, we get $P \mid Q \rightarrow P' \mid Q'$ by (Red $\equiv$).

(Comm τ-*anon* 1)

$$\frac{P \xrightarrow{\overline{anon}} (\nu\vec{p})\langle M\rangle P' \quad Q \xrightarrow{anon\, M} Q' \quad \{\vec{p}\} \cap \mathit{fn}(Q) = \varnothing}{P \mid Q \xrightarrow{\tau} (\nu\vec{p})(P' \mid Q')}$$

By induction hypothesis, $P \xrightarrow{\overline{anon}} (\nu\vec{p})\langle M\rangle P'$ implies $P \equiv (\nu\vec{p})(\langle M\rangle \mid P')$. By induction hypothesis, $Q \xrightarrow{anon\, M} Q'$ implies $Q \equiv (\nu\vec{q})(Q_1 \mid (x).Q_2)$ with $\mathit{fn}(M) \cap \{\vec{q}\} = \varnothing$ and $Q' \equiv (\nu\vec{q})(Q_1 \mid Q_2\{x \leftarrow M\})$.

Since the names $\vec{p}$ and $\vec{q}$ are bound, we may assume that $\{\vec{p}\} \cap \mathit{fn}(Q_1 \mid (x).Q_2) = \varnothing$ and $\{\vec{q}\} \cap \mathit{fn}(tupleM \mid P') = \varnothing$.

We may calculate:

$$\begin{aligned} P \mid Q &\equiv (\nu\vec{p})(\langle M\rangle \mid P') \mid (\nu\vec{q})(Q_1 \mid (x).Q_2) \\ &\equiv (\nu\vec{p})(\nu\vec{q})(\langle M\rangle \mid (x).Q_2 \mid P' \mid Q_1) \end{aligned}$$

The second step appeals to (Struct Par Comm), (Struct Par Assoc), (Struct Res Par) and the facts that $\{\vec{p}\} \cap \mathit{fn}(Q_1 \mid (x).Q_2) = \varnothing$ and $\{\vec{q}\} \cap \mathit{fn}(\langle M\rangle \mid P') = \varnothing$.

By (Red Res), (Red Par) and (Red I/O) we get:

$$(\nu\vec{p})(\nu\vec{q})(\langle M\rangle \mid (x).Q_2 \mid P' \mid Q_1) \;\rightarrow\; (\nu\vec{p})(\nu\vec{q})(Q_2\{x \leftarrow M\} \mid P' \mid Q_1)$$

We may calculate:

$$(\nu\vec{p})(\nu\vec{q})(Q_2\{x \leftarrow M\} \mid P' \mid Q_1) \equiv (\nu\vec{p})(P' \mid (\nu\vec{q})(Q_1 \mid Q_2\{x \leftarrow M\}))$$

Again, we appeal to (Struct Par Comm), (Struct Par Assoc), (Struct Res Par) and the facts that $\{\vec{p}\} \cap fn(Q_1 \mid (x).Q_2) = \varnothing$ and $\{\vec{q}\} \cap fn(\langle M \rangle \mid P') = \varnothing$.

Combining the calculations above, we get $P \mid Q \rightarrow (\nu\vec{p})(P' \mid Q')$ by (Red $\equiv$).

Cases (Comm τ-*enter* 2), (Comm τ-*open* 2) and (Comm τ-*anon* 2) are symmetric. All the other cases follow easily. $\square$

Next, we have a lemma that asserts a uniformity property of commitments of the form $P \xrightarrow{anon\ M} Q$: given such a commitment, and another capability N there is R such that $P \xrightarrow{anon\ N} R$.

Lemma 9 *Whenever $P \xrightarrow{anon\ M} Q$ there is a variable $x \notin fv(P)$ and a process R with $fv(R) = fv(P) \cup \{x\}$ and $fn(R) = fn(P)$, such that $Q = R\{x \leftarrow M\}$ and, for any N, $P \xrightarrow{anon\ N} R\{x \leftarrow N\}$.*

Proof By induction on the derivation of $P \xrightarrow{enter\ n\ A} Q$.

To state an analogous lemma for commitments of the form $P \xrightarrow{enter\ n\ A} Q$ we need to introduce the notion of an evaluation context. Let an *evaluation context*, $\mathcal{E}$, be a phrase given as follows.

Evaluation Contexts

$\mathcal{E}$ ::=	evaluation context
$\bullet$	hole
$(\nu n)\mathcal{E}$	restriction
$P \mid \mathcal{E}$	left composition
$\mathcal{E} \mid Q$	right composition
$n[\mathcal{E}]$	ambient

We identify evaluation contexts up to renaming of bound names and variables. Every evaluation context is a process with exactly one occurrence of a hole, $\bullet$. We write $\mathcal{E}\{P\}$ for the outcome of the capture-avoiding substitution of the process P for the single occurrence of the hole, $\bullet$, in the evaluation context, $\mathcal{E}$.

The introduction of evaluation contexts allows us to formulate the following uniformity property of commitments of the form $P \xrightarrow{enter\ n\ A} Q$.

Lemma 10 *Whenever* $P \xrightarrow{enter\ n\ A} Q$ *there is an evaluation context,* $\mathcal{E}$*, such that* $P \equiv \mathcal{E}\{\mathbf{0}\}$*,* $Q = \mathcal{E}\{A\}$*, and, for any ambient* B*,* $P \xrightarrow{enter\ n\ B} \mathcal{E}\{B\}$*.*

Proof By induction on the derivation of $P \xrightarrow{enter\ n\ A} Q$. We show one case:

(Comm Repl) We have $!P \xrightarrow{enter\ n\ A} Q$ derived from $P \mid !P \xrightarrow{enter\ n\ A} Q$. By induction hypothesis, there is an evaluation context $\mathcal{E}$ such that $P \mid !P \equiv \mathcal{E}\{\mathbf{0}\}$, $Q = \mathcal{E}\{A\}$, and, for any B, $P \mid !P \xrightarrow{enter\ n\ B} \mathcal{E}\{B\}$. By (Struct Repl) and (Struct Trans), $P \equiv \mathcal{E}\{\mathbf{0}\}$. By (Comm Repl), $P \xrightarrow{enter\ n\ B} \mathcal{E}\{B\}$, for any B. $\square$

Next are three lemmas concerning structural congruence of outcomes.

Lemma 11

(1) *Suppose* $(\nu\vec{p})\langle A\rangle P \equiv (\nu\vec{q})\langle B\rangle Q$*. Then* $(\nu\vec{p})(A \mid P) \equiv (\nu\vec{q})(B \mid Q)$*. Moreover, if* $\mathcal{E}_1$ *and* $\mathcal{E}_2$ *are evaluation contexts with* $(fn(\mathcal{E}_1) \cup fn(\mathcal{E}_2)) \cap \{\vec{p}, \vec{q}\} = \varnothing$ *then* $(\nu\vec{p})\langle \mathcal{E}_1\{A\}\rangle(\mathcal{E}_2\{P\}) \equiv (\nu\vec{q})\langle \mathcal{E}_1\{B\}\rangle(\mathcal{E}_2\{Q\})$*.*

(2) *Suppose* $(\nu\vec{p})\langle M\rangle P \equiv (\nu\vec{q})\langle N\rangle Q$*. Then* $M = N$ *and* $(\nu\vec{p})(R\{x \leftarrow M\} \mid P) \equiv (\nu\vec{q})(R\{x \leftarrow M\} \mid Q)$ *for any* R *with* $fn(R) \cap \{\vec{p}, \vec{q}\} = \varnothing$*. Moreover if* $\mathcal{E}$ *is an evaluation context with* $fn(\mathcal{E}) \cap \{\vec{p}, \vec{q}\} = \varnothing$ *then* $(\nu\vec{p})\langle M\rangle(\mathcal{E}\{P\}) \equiv (\nu\vec{q})\langle M\rangle(\mathcal{E}\{Q\})$*.*

Proof By inductions on the derivations of $(\nu\vec{p})\langle A\rangle P \equiv (\nu\vec{q})\langle B\rangle Q$ and $(\nu\vec{p})\langle M\rangle P \equiv (\nu\vec{q})\langle N\rangle Q$. $\square$

Lemma 12

(1) *If* $O \equiv O'$ *then* $P \parallel O \equiv P \parallel O'$*.*

(2) *If* $P \equiv Q$ *then* $P \parallel O \equiv Q \parallel O$*.*

(3) $P \parallel O \equiv O \parallel P$*.*

(4) $(P \mid Q) \parallel O \equiv P \parallel (Q \parallel O)$*,* $(P \parallel O) \parallel Q \equiv P \parallel (O \parallel Q)$ *and* $(O \parallel P) \parallel Q \equiv O \parallel (P \mid Q)$*.*

(5) *If* $n \notin fn(O)$ *then* $(\nu n)(O \parallel P) \equiv O \parallel (\nu n)P$*. If* $p \notin fn(P)$ *then* $(\nu n)(P \parallel O) \equiv P \parallel (\nu n)O$*.*

Proof For part (1), if $O \equiv O'$ either both O and O' are processes, or they are both concretions. In the former case, part (1) holds trivially. Otherwise, suppose $O = (\nu\vec{p})\langle\phi\rangle\hat{P}$ and $O' = (\nu\vec{q})\langle\psi\rangle\hat{Q}$, where ϕ and ψ are either both the same capability or are both processes. Since the names $\vec{p}, \vec{q}$ are bound, we may assume that $\{\vec{p}, \vec{q}\} \cap (fn(P) \cup fn(Q)) = \varnothing$. Therefore:

$$\begin{array}{rcl} P \parallel O & = & (\nu\vec{p})\langle\phi\rangle(P \mid \hat{P}) \\ P \parallel O' & = & (\nu\vec{q})\langle\psi\rangle(P \mid \hat{Q}) \end{array}$$

Since $P \mid \bullet$ is an evaluation context whose free names are disjoint from $\{\vec{p}, \vec{q}\}$, parts (1) and (2) of Lemma 11 assert that

$$(\nu\vec{p})\langle\phi\rangle(P \mid \hat{P}) \quad \equiv \quad (\nu\vec{q})\langle\psi\rangle(P \mid \hat{Q}),$$

which is to say that $P \parallel O \equiv P \parallel O'$. This completes part (1).

If O is a process, parts (2), (3), (4) and (5) hold directly from the rules of structural congruence. Otherwise, suppose $O = (\nu\vec{p})\langle\phi\rangle R$. Since the names $\vec{p}, \vec{q}$ are bound, we may assume that $\{\vec{p}, \vec{q}\} \cap (fn(P) \cup fn(Q)) = \varnothing$.

For part (2):

$$\begin{array}{rcl} P \parallel O & = & (\nu\vec{p})\langle\phi\rangle(P \mid R) \\ Q \parallel O & = & (\nu\vec{p})\langle\psi\rangle(Q \mid R) \end{array}$$

Hence $P \parallel O \equiv Q \parallel O$ follows from $P \equiv Q$ using (Conc Res), either (Conc Amb) or (Conc Capa), and (Struct Par). similar reasoning.

For part (3):

$$\begin{array}{rcl} P \parallel O & = & (\nu\vec{p})\langle\phi\rangle(P \mid R) \\ O \parallel P & = & (\nu\vec{p})\langle\phi\rangle(R \mid P) \end{array}$$

Hence $P \parallel O \equiv O \parallel P$ follows using (Conc Res), either (Conc Amb) or (Conc Capa), and (Struct Comm).

For part (4):

$$\begin{array}{rcl} (P \mid Q) \parallel O & = & (\nu\vec{p})\langle\phi\rangle((P \mid Q) \mid R) \\ P \parallel (Q \parallel O) & = & P \parallel (\nu\vec{p})\langle\phi\rangle(Q \mid R) \\ & = & (\nu\vec{p})\langle\phi\rangle(P \mid (Q \mid R)) \end{array}$$

Hence $(P \mid Q) \parallel O \equiv P \parallel (Q \parallel O)$ follows from (Conc Res), either (Conc Amb) or (Conc Capa), and (Struct Assoc). We can compute $(P \parallel O) \parallel Q \equiv P \parallel (O \parallel Q)$ and $(O \parallel P) \parallel Q \equiv O \parallel (P \mid Q)$ via similar reasoning.

For part (5), we assume $n \notin fn(O)$. Since the names $\vec{p}$ are bound, we may assume $n \notin \{\vec{p}\}$. We compute:

$$\begin{array}{rcl} (\nu n)(O \parallel P) & = & (\nu n)(\nu \vec{p})\langle \phi \rangle (R \mid P) \\ O \parallel (\nu n)P & = & (\nu \vec{p})\langle \phi \rangle (R \mid (\nu n)P) \end{array}$$

From $n \notin fn(O) = (fn(\phi) \cup fn(R)) - \{\vec{p}\}$ and $n \notin \{\vec{p}\}$ we get that $n \notin fn(\phi)$ and $n \notin fn(R)$. By (Conc Res Res), $(\nu n)(O \parallel P) \equiv (\nu \vec{p})(\nu n)\langle \phi \rangle (R \mid P)$. Hence, by (Conc Res 2), $n \notin fn(\phi)$ implies $(\nu \vec{p})(\nu n)\langle \phi \rangle (R \mid P) \equiv (\nu \vec{p})\langle \phi \rangle (\nu n)(R \mid P)$. By (Struct Par Res), $n \notin fn(R)$ implies $(\nu \vec{p})\langle \phi \rangle (\nu n)(R \mid P) \equiv (\nu \vec{p})\langle \phi \rangle (R \mid (\nu n)P)$. Finally, by (Conc Trans), we get $(\nu n)(O \parallel P) \equiv O \parallel (\nu n)P$.

By similar reasoning we may establish that $p \notin fn(P)$ implies $(\nu n)(P \parallel O) \equiv P \parallel (\nu n)O$. $\square$

Lemma 13 *For all outcomes O_1, O_2, $O_1 \equiv\!\overline{\equiv}\!\equiv O_2$ if and only if $O_1 \equiv O_2$.*

Proof Either $O_1 \equiv\!\overline{\equiv}\!\equiv O_2$ or $O_1 \equiv O_2$ implies that O_1 and O_2 are either both processes or both concretions. Therefore, if they are both processes, the lemma follows at once. Otherwise, both are concretions and we examine each direction separately.

For the left-to-right direction, we have $O_1 \equiv O_1'$, $O_1' \overline{\equiv} O_2'$ and $O_2' \equiv O_2$. By definition of $O_1' \overline{\equiv} O_2'$, (Conc Res) and either (Conc Amb) or (Conc Capa) imply $O_1' \equiv O_2'$. By (Conc Trans), $O_1 \equiv O_1'$, $O_1' \equiv O_2'$ and $O_2' \equiv O_2$ imply $O_1 \equiv O_2$.

For the right-to-left direction, we have $O_1 \overline{\equiv} O_1$ by (Struct Refl) and either (Outcome Conc Amb) or (Outcome Conc Capa). This together with $O_1 \equiv O_1$ and $O_1 \equiv O_2$ imply $O_1 \equiv\!\overline{\equiv}\!\equiv O_2$. $\square$

The next lemma—that structural congruence of closed processes is a strong bisimulation—is the crux of showing the harder of the two directions of Proposition 15, that any reduction step implies a τ-commitment.

Lemma 14 *For all P and Q, $P \equiv Q$ implies that:*

(1) *whenever $P \xrightarrow{\alpha} O$ there is O' such that $Q \xrightarrow{\alpha} O'$ and $O \equiv O'$;*

(2) *whenever $Q \xrightarrow{\alpha} O'$ there is O such that $P \xrightarrow{\alpha} O$ and $O \equiv O'$.*

Hence, by Lemma 13, structural congruence, restricted to closed processes, is a strong bisimulation.

Proof By induction on the derivation of $P \equiv Q$.

(Struct Refl), (Struct Symm), (Struct Trans) Easy.

(Struct Res) We have $P = (\nu n)P_1 \equiv (\nu n)Q_1 = Q$ derived from $P_1 \equiv Q_1$. We prove part (1); part (2) follows by symmetry. Only (Comm Res) may derive a commitment from a restriction, so $P \xrightarrow{\alpha} O$ implies there is O_1 such that $P_1 \xrightarrow{\alpha} O_1$, $n \notin fn(\alpha)$ and $O = (\nu n)O_1$. By induction hypothesis, $P_1 \equiv Q_1$ and $P_1 \xrightarrow{\alpha} O_1$ imply there is O_1' such that $Q \xrightarrow{\alpha} O_1'$ and $O_1 \equiv O_1'$. Let $O' = (\nu n)O_1'$. Given $n \notin fn(\alpha)$, by (Comm Res), we obtain $Q = (\nu n)Q_1 \xrightarrow{\alpha} O'$. By (Conc Res), $O_1 \equiv O_1'$ implies $O = (\nu n)O_1 \equiv (\nu n)O_1' = O'$.

(Struct Par) We have $P = P_1 \mid R \equiv Q_1 \mid R = Q$ derived from $P_1 \equiv Q_1$. For part (1), we need to consider all the ways in which $P_1 \mid R \xrightarrow{\alpha} O$ and in each case produce O' such that $Q_1 \mid R \xrightarrow{\alpha} O'$ with $O \equiv O'$.

(Comm Par 1) In this case, $P_1 \xrightarrow{\alpha} O_1$ and $O = O_1 \parallel R$. By induction hypothesis, $P_1 \equiv Q_1$ implies there is O_1' such that $Q_1 \xrightarrow{\alpha} O_1'$ and $O_1 \equiv O_1'$. Let $O' = O_1' \parallel R$. By (Comm Par 1), $Q \xrightarrow{\alpha} O'$. By Lemma 12(1,3), $O_1 \equiv O_1'$ implies that $O \equiv O'$.

(Comm Par 2) In this case, $R \xrightarrow{\alpha} O''$ and $O = P_1 \parallel O''$. Let $O' = Q_1 \parallel O''$. By (Comm Par 2), $Q \xrightarrow{\alpha} O'$. By Lemma 12(2), $P_1 \equiv Q_1$ implies $P_1 \parallel O'' \equiv Q_1 \parallel O''$, that is, $O \equiv O'$.

(Comm τ-*enter* 1) In this case, $P_1 \mid R \xrightarrow{\tau} (\nu\vec{p})(P_1' \mid R')$ derives from $P_1 \xrightarrow{\overline{enter\,n}} (\nu\vec{p})\langle A\rangle P_1'$ and $R \xrightarrow{enter\,n\,A} R'$ with $\{\vec{p}\} \cap fn(R) = \varnothing$. Let $O_1 = (\nu\vec{p})\langle A\rangle P_1'$. By induction hypothesis, $P_1 \equiv Q_1$ and $P_1 \xrightarrow{\overline{enter\,n}} O_1$ imply there is O_1' such that $Q_1 \xrightarrow{\overline{enter\,n}} O_1'$ and $O_1 \equiv O_1'$. Suppose $O_1' = (\nu\vec{q})\langle B\rangle Q_1'$. Since the names $\vec{q}$ are bound, we may assume that $\{\vec{q}\} \cap fn(R) = \varnothing$. By Lemma 10, $R \xrightarrow{enter\,n\,A} R'$ implies there is an evaluation context, $\mathcal{E}$, such that $R \equiv \mathcal{E}\{\mathbf{0}\}$, $R' = \mathcal{E}\{A\}$, and $R \xrightarrow{enter\,n\,B} \mathcal{E}\{B\}$. By Lemma 1, $R \equiv \mathcal{E}\{\mathbf{0}\}$ implies $fn(\mathcal{E}) = fn(R)$. Therefore $fn(\mathcal{E}) \cap \{\vec{p},\vec{q}\} = \varnothing$. By (Comm τ-*enter* 1), $Q_1 \xrightarrow{\overline{enter\,n}} O_1'$, $R \xrightarrow{enter\,n\,B} \mathcal{E}\{B\}$ and $\{\vec{q}\} \cap fn(R) = \varnothing$ imply that $Q_1 \mid R \xrightarrow{\tau} (\nu\vec{q})(Q_1' \mid \mathcal{E}\{B\})$. By Lemma 11(1), $(\nu\vec{p})\langle A\rangle P_1' \equiv (\nu\vec{q})\langle B\rangle Q_1'$ and $fn(\mathcal{E}) \cap \{\vec{p},\vec{q}\} = \varnothing$ imply $(\nu\vec{p})(P_1' \mid \mathcal{E}\{A\}) \equiv (\nu\vec{q})(Q_1' \mid \mathcal{E}\{B\})$.

(Comm τ-*enter* 2) In this case, $P_1 \mid R \xrightarrow{\tau} (\nu\vec{r})(P_1' \mid R')$ derives from $P_1 \xrightarrow{enter\,n\,A} P_1'$ and $R \xrightarrow{\overline{enter\,n}} (\nu\vec{r})\langle A\rangle R'$ with $\{\vec{r}\} \cap fn(P_1) = \varnothing$. By induction hypothesis, $P_1 \xrightarrow{enter\,n\,A} P_1'$ and $P_1 \equiv Q_1$ implies there is Q_1' with $Q_1 \xrightarrow{enter\,n\,A} Q_1'$ and $P_1' \equiv Q_1'$. Since the names $\vec{r}$ are bound, we may assume that $\{\vec{r}\} \cap fn(Q_1) = \varnothing$. By (Comm τ-*enter* 2), $Q_1 \xrightarrow{enter\ n\ A} Q_1'$ and $R \xrightarrow{\overline{enter\ n}} (\nu\vec{r})\langle A\rangle R'$ and $\{\vec{r}\} \cap fn(Q_1) = \varnothing$ imply $Q_1 \mid R \xrightarrow{\tau} (\nu\vec{r})(Q_1' \mid R')$. Since $\equiv$ is a congruence, $P_1' \equiv Q_1'$ implies that $(\nu\vec{r})(P_1' \mid R') \equiv (\nu\vec{r})(Q_1' \mid R')$.

(Comm τ-*open* 1) In this case, $P_1 \mid R \xrightarrow{\tau} P_1' \mid R'$ derives from $P_1 \xrightarrow{\overline{open\ n}} P_1'$ and $R \xrightarrow{open\ n} R'$. By induction hypothesis, $P_1 \xrightarrow{\overline{open\ n}} P_1'$ and $P_1 \equiv Q_1$ implies there is Q_1' such that $Q_1 \xrightarrow{\overline{open\ n}} Q_1'$ and $P_1' \equiv Q_1'$. By (Comm τ-*open* 1), $Q_1 \xrightarrow{\overline{open\ n}} Q_1'$ and $R \xrightarrow{open\ n} R'$ imply $Q_1 \mid R \xrightarrow{\tau} Q_1' \mid R'$. Since $\equiv$ is a congruence, $P_1' \equiv Q_1'$ implies $P_1' \mid R' \equiv Q_1' \mid R'$.

(Comm τ-*open* 2) In this case, $P_1 \mid R \xrightarrow{\tau} P_1' \mid R'$ derives from $P_1 \xrightarrow{open\ n} P_1'$ and $R \xrightarrow{\overline{open\ n}} R'$. By induction hypothesis, $P_1 \xrightarrow{open\ n} P_1'$ and $P_1 \equiv Q_1$ implies there is Q_1' such that $Q_1 \xrightarrow{open\ n} Q_1'$ and $P_1' \equiv Q_1'$. By (Comm τ-*open* 2), $Q_1 \xrightarrow{open\ n} Q_1'$ and $R \xrightarrow{\overline{open\ n}} R'$ imply $Q_1 \mid R \xrightarrow{\tau} Q_1' \mid R'$. Since $\equiv$ is a congruence, $P_1' \equiv Q_1'$ implies $P_1' \mid R' \equiv Q_1' \mid R'$.

(Comm τ-*anon* 1) In this case, $P_1 \mid R \xrightarrow{\tau} (\nu\vec{p})(P_1' \mid R')$ derives from $P_1 \xrightarrow{\overline{anon}} (\nu\vec{p})\langle M\rangle P_1'$ and $R \xrightarrow{anon\ M} R'$ with $\{\vec{p}\} \cap fn(R) = \varnothing$. By Lemma 9, $R \xrightarrow{anon\ M} R'$ implies there is a variable $x \notin fv(R)$ and a process R'' with $fv(R'') = fv(R') \cup \{x\}$ and $fn(R'') = fn(R')$ such that $R' = R''\{x \leftarrow M\}$. By induction hypothesis, $P_1 \xrightarrow{\overline{anon}} (\nu\vec{p})\langle M\rangle P_1'$ and $P_1 \equiv Q_1$ imply there is O such that $Q_1 \xrightarrow{\overline{anon}} O$ and $(\nu\vec{p})\langle M\rangle P_1' \equiv O$. The latter implies that $O = (\nu\vec{q})\langle M\rangle Q_1'$. Since the names $\vec{q}$ are bound, we may assume that $\{\vec{q}\} \cap fn(R) = \varnothing$. Therefore $fn(R'') \cap \{\vec{p}, \vec{q}\} = \varnothing$. By (Comm τ-*anon* 1), $Q_1 \xrightarrow{\overline{anon}} (\nu\vec{q})\langle M\rangle Q_1'$, $R \xrightarrow{anon\ M} R'$ and $\{\vec{q}\} \cap fn(R) = \varnothing$ imply $Q_1 \mid R \xrightarrow{\tau} (\nu\vec{q})(Q_1' \mid R')$. By Lemma 11(2), $(\nu\vec{p})\langle M\rangle P_1' \equiv (\nu\vec{q})\langle M\rangle Q_1'$ and $fn(R'') \cap \{\vec{p}, \vec{q}\} = \varnothing$ implies $(\nu\vec{p})(P_1' \mid R') \equiv (\nu\vec{q})(Q_1' \mid R')$.

(Comm τ-*anon* 2) In this case, $P_1 \mid R \xrightarrow{\tau} (\nu\vec{r})(P_1' \mid R')$ derives from $P_1 \xrightarrow{anon\ M} P_1'$ and $R \xrightarrow{\overline{anon}} (\nu\vec{r})\langle M\rangle R'$ and $\{\vec{r}\} \cap fn(P_1) = \varnothing$. By induction hypothesis, $P_1 \xrightarrow{anon\ M} P_1'$ and $P_1 \equiv Q_1$ imply there is Q_1' such that $Q_1 \xrightarrow{anon\ M} Q_1'$ and $P_1' \equiv Q_1'$. Since the names $\vec{r}$ are bound, we may assume that $\{\vec{r}\} \cap fn(Q_1) = \varnothing$. By (Comm τ-*anon* 2), $Q_1 \xrightarrow{anon\ M} Q_1'$, $R \xrightarrow{\overline{anon}} (\nu\vec{r})\langle M\rangle R'$ and $\{\vec{r}\} \cap fn(Q_1) = \varnothing$ imply $Q_1 \mid R \xrightarrow{\tau} (\nu\vec{r})(Q_1' \mid R')$. Since $\equiv$ is a congruence, $P_1' \equiv Q_1'$ implies $(\nu\vec{r})(P_1' \mid R') \equiv (\nu\vec{r})(Q_1' \mid R')$.

This completes part (1) of the case for (Struct Par). Part (2) follows by a symmetric argument.

(Struct Repl) We have $!P \equiv !Q$ derived from $P \equiv Q$. For part (1), we need to prove that whenever $!P \xrightarrow{\alpha} O$ there is O' such that $!Q \xrightarrow{\alpha} O'$ and $O \equiv O'$. The only rule that can derive $!P \xrightarrow{\alpha} O$ is (Comm Repl). Therefore $P \mid !P \xrightarrow{\alpha} O$. If $P \equiv Q$ then $P \mid !P \equiv Q \mid !Q$ too. By induction hypothesis, $P \mid !P \xrightarrow{\alpha} O$ and $P \mid !P \equiv Q \mid !Q$ imply there is O' such that $Q \mid !Q \xrightarrow{\alpha} O'$ and $O \equiv O'$. By (Comm Repl), this implies $!Q \xrightarrow{\alpha} O'$. Part (2) follows by symmetry.

(Struct Amb) We have $P = n[P_1] \equiv n[Q_1] = Q$ derived from $P_1 \equiv Q_1$. We prove part (1); part (2) follows by symmetry. Suppose $n[P_1] \xrightarrow{\alpha} O$. This can only be obtained from one of six rules:

(Comm Amb) Here $\alpha = \tau$ and $P_1 \xrightarrow{\tau} P_1'$ with $O = n[P_1']$. By induction hypothesis, there is Q_1' such that $Q_1 \xrightarrow{\tau} Q_1'$ and $P_1' \equiv Q_1'$. Let $O' = n[Q_1']$. By (Comm Amb), we obtain $Q = n[Q_1] \xrightarrow{\tau} O'$. By (Struct Res), $P_1' \equiv Q_1'$ implies $O = n[P_1'] \equiv n[Q_1'] = O'$.

(Comm $\overline{enter}$) Here $\alpha = \overline{enter\ m}$, $P_1 \xrightarrow{in\ m} P_1'$ and $O = \langle n[P_1'] \rangle \mathbf{0}$. By induction hypothesis, there is Q_1' such that $Q_1 \xrightarrow{in\ m} Q_1'$ and $P_1' \equiv Q_1'$. Let $O' = \langle n[Q_1'] \rangle \mathbf{0}$. By (Comm $\overline{enter}$), we obtain $Q = n[Q_1] \xrightarrow{in\ m} O'$. By (Conc Amb), $P_1' \equiv Q_1'$ implies $\langle n[P_1'] \rangle \mathbf{0} \equiv \langle n[Q_1'] \rangle \mathbf{0}$, that is, $O \equiv O'$.

(Comm *enter*) Here $\alpha = enter\, n\, A$ and $O = n[A \mid P_1]$. Let $O' = n[A \mid Q_1]$. By (Comm *enter*), we obtain $Q \xrightarrow{enter\ n\ A} O'$. By (Struct Refl), (Struct Trans), (Struct Amb) and (Struct Par), $P_1 \equiv Q_1$ implies $O \equiv O'$.

(Comm *exit*) Here $\alpha = exit\ m$ and $P_1 \xrightarrow{out\ m} P_1'$ with $O = \langle n[P_1'] \rangle \mathbf{0}$. By induction hypothesis, there is Q_1' such that $Q_1 \xrightarrow{out\ m} Q_1'$ and $P_1' \equiv Q_1'$. Let $O' = \langle n[Q_1'] \rangle \mathbf{0}$. By (Comm *exit*) we obtain $Q = n[Q_1] \xrightarrow{exit\ m} O'$. By (Conc Amb), $P_1' \equiv Q_1'$ implies $O \equiv O'$.

(Comm τ-*exit*) Here $\alpha = \tau$ and $P_1 \xrightarrow{exit\ n} (\nu \vec{p}) \langle A \rangle P_1'$ with $n \notin \{\vec{p}\}$ and $O = (\nu \vec{p})(A \mid n[P_1'])$. By induction hypothesis, there is a concretion $(\nu \vec{q}) \langle B \rangle Q_1'$ such that $Q_1 \xrightarrow{exit\ n} (\nu \vec{q}) \langle B \rangle Q_1'$ with $(\nu \vec{p}) \langle A \rangle P_1' \equiv (\nu \vec{q}) \langle B \rangle Q_1'$. Since the names $\{\vec{q}\}$ are bound, we may assume $n \notin \{\vec{q}\}$. Let $O' = (\nu \vec{q})(B \mid n[Q_1'])$. By (Comm τ-*exit*), we obtain $Q = n[Q_1] \xrightarrow{\tau} O'$. By Lemma 11(1), $(\nu \vec{p}) \langle A \rangle P_1' \equiv (\nu \vec{q}) \langle B \rangle Q_1'$ and $n \notin \{\vec{p}\} \cup \{\vec{q}\}$ imply $O \equiv O'$.

(Comm $\overline{open}$) Here $\alpha = \overline{open\, n}$ and $O = P_1$. We have $P_1 \equiv Q_1$ and by (Comm $\overline{open}$) we obtain $Q \xrightarrow{\overline{open\, n}} Q_1$.

(Struct Action) We have $M.P \equiv M.Q$ derived from $P \equiv Q$. For part (1), we need to prove that whenever $M.P \xrightarrow{\alpha} O$ there is O' such that $M.Q \xrightarrow{\alpha} O'$ and $O \equiv O'$. The only rules that can derive $M.P \xrightarrow{\alpha} O$ are (Comm *in*), (Comm *out*) and (Comm *open*). Therefore $\alpha = M$ and $O = P$. By one of (Comm *in*), (Comm *out*) or (Comm *open*) we get $M.Q \xrightarrow{M} Q$, to complete part (1). Part (2) follows by symmetry.

(Struct Input) We have $(x).P \equiv (x).Q$ derived from $P \equiv Q$. For part (1), we need to prove that whenever $(x).P \xrightarrow{\alpha} O$ there is O' such that $(x).Q \xrightarrow{\alpha} O'$ and $O \equiv O'$. The only rule that can derive $(x)P \xrightarrow{\alpha} O$ is (Comm *anon*). Therefore $\alpha = anon\ M$, for some M, and $O = P\{x \leftarrow M\}$. Let $O' = Q\{x \leftarrow M\}$. By (Comm *anon*), $(x).Q \xrightarrow{anon\ M} Q\{x \leftarrow M\}$, that is, $(x).Q \xrightarrow{\alpha} O'$. By Lemma 2, $P \equiv Q$ implies $P\{x \leftarrow M\} \equiv Q\{x \leftarrow M\}$, that is, $O \equiv O'$. Part (2) follows by symmetry.

(Struct Par Comm) Easy, since each rule for deriving a commitment from a composition has a symmetric twin.

(Struct Par Assoc) We have $(P_1 \mid P_2) \mid P_3 \equiv P_1 \mid (P_2 \mid P_3)$. For part (1), we must show that whenever $(P_1 \mid P_2) \mid P_3 \xrightarrow{\alpha} O$ there is O' with $P_1 \mid (P_2 \mid P_3) \xrightarrow{\alpha} O'$ with $O \equiv O'$.

Lemma 6 enumerates the eight ways in which $(P_1 \mid P_2) \mid P_3 \xrightarrow{\alpha} O$ may be derived. We consider each possibility in turn.

(1) $O = O'' \parallel P_3$ where $P_1 \mid P_2 \xrightarrow{\alpha} O''$. By Lemma 6, there are a further eight cases to consider, based on the derivation of $P_1 \mid P_2 \xrightarrow{\alpha} O''$.

(a) $O'' = O''' \parallel P_2$ where $P_1 \xrightarrow{\alpha} O'''$.
Here $O = (O''' \parallel P_2) \parallel P_3$. Let $O' = O''' \parallel (P_2 \mid P_3)$. By (Comm Par 1), $P_1 \xrightarrow{\alpha} O'''$ implies $P_1 \mid (P_2 \mid P_3) \xrightarrow{\alpha} O'$. By Lemma 12(4), $O \equiv O'$.

(b) $O'' = P_1 \parallel O'''$ where $P_2 \xrightarrow{\alpha} O'''$.
Here $O = (P_1 \parallel O''') \parallel P_3$. Let $O' = P_1 \parallel (O''' \parallel P_3)$. By (Comm Par 1), $P_2 \xrightarrow{\alpha} O'''$ implies $P_2 \mid P_3 \xrightarrow{\alpha} O''' \parallel P_3$. By (Comm Par 2), this implies $P_1 \mid (P_2 \mid P_3) \xrightarrow{\alpha} O'$. By Lemma 12(4), $O \equiv O'$.

(c) $O'' = (\nu\vec{m})(P_1' \mid P_2')$ and $\alpha = \tau$ where $P_1 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_1'$ and $P_2 \xrightarrow{enter\ n\ A} P_2'$ with $\{\vec{m}\} \cap \mathit{fn}(P_2) = \varnothing$.
Here $O = (\nu\vec{m})(P_1' \mid P_2') \mid P_3$. Since the names $\vec{m}$ are bound, we may assume that $\mathit{fn}(P_3) \cap \{\vec{m}\} = \varnothing$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2' \mid P_3))$. By (Comm Par 1), $P_2 \xrightarrow{enter\ n\ A} P_2'$ implies $P_2 \mid P_3 \xrightarrow{enter\ n\ A} P_2' \mid P_3$. By (Comm τ-*enter* 1), this and $P_1 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_1'$ with $\{\vec{m}\} \cap \mathit{fn}(P_2 \mid P_3) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. By (Struct Assoc) and (Struct Res Par), in particular, $O \equiv O'$.

(d) $O'' = (\nu\vec{m})(P_1' \mid P_2')$ and $\alpha = \tau$ where $P_1 \xrightarrow{enter\ n\ A} P_1'$ and $P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_2'$ with $\{\vec{m}\} \cap \mathit{fn}(P_1) = \varnothing$.
Here $O = (\nu\vec{m})(P_1' \mid P_2') \mid P_3$. Since the names $\vec{m}$ are bound, we may assume that $\mathit{fn}(P_3) \cap \{\vec{m}\} = \varnothing$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2' \mid P_3))$. By (Comm Par 1), $P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_2'$ implies $P_2 \mid P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle(P_2' \mid P_3)$. By (Comm τ-*enter* 2), this and $P_1 \xrightarrow{enter\ n\ A} P_1'$ with $\{\vec{m}\} \cap \mathit{fn}(P_1) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. By (Struct Assoc) and (Struct Res Par), in particular, $O \equiv O'$.

(e) $O'' = P_1' \mid P_2'$ and $\alpha = \tau$ where $P_1 \xrightarrow{\overline{open\ n}} P_1'$ and $P_2 \xrightarrow{open\ n} P_2'$.
Here $O = (P_1' \mid P_2') \mid P_3$. Let $O' = P_1' \mid (P_2' \mid P_3)$. By (Comm Par 1), $P_2 \xrightarrow{open\ n} P_2'$ implies $P_2 \mid P_3 \xrightarrow{open\ n} P_2' \mid P_3$. By (Comm τ-*open* 1), this and $P_1 \xrightarrow{\overline{open\ n}} P_1'$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. (Struct Assoc) asserts $O \equiv O'$.

(f) $O'' = P_1' \mid P_2'$ and $\alpha = \tau$ where $P_1 \xrightarrow{open\ n} P_1'$ and $P_2 \xrightarrow{\overline{open\ n}} P_2'$.
Here $O = (P_1' \mid P_2') \mid P_3$. Let $O' = P_1' \mid (P_2' \mid P_3)$. By (Comm Par 1), $P_2 \xrightarrow{\overline{open\ n}} P_2'$ implies $P_2 \mid P_3 \xrightarrow{\overline{open\ n}} P_2' \mid P_3$. By (Comm τ-*open* 2), this and $P_1 \xrightarrow{open\ n} P_1'$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. (Struct Assoc) asserts $O \equiv O'$.

(g) $O'' = (\nu\vec{m})(P_1' \mid P_2')$ and $\alpha = \tau$ where $P_1 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_1'$ and $P_2 \xrightarrow{anon\ M} P_2'$ with $\{\vec{m}\} \cap \mathit{fn}(P_2) = \varnothing$.
Here $O = (\nu\vec{m})(P_1' \mid P_2') \mid P_3$. Since the names $\vec{m}$ are bound, we may assume that $\mathit{fn}(P_3) \cap \{\vec{m}\} = \varnothing$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2' \mid P_3))$. By (Comm Par 1), $P_2 \xrightarrow{anon\ M} P_2'$ implies $P_2 \mid P_3 \xrightarrow{anon\ M} P_2' \mid P_3$. By (Comm τ-*anon* 1), this and $P_1 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_1'$ with $\{\vec{m}\} \cap \mathit{fn}(P_2 \mid P_3) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. By (Struct Res Par) and (Struct

Assoc), in particular, $O \equiv O'$.

(h) $O'' = (\nu\vec{m})(P_1' \mid P_2')$ and $\alpha = \tau$ where $P_1 \xrightarrow{anon\ M} P_1'$ and $P_2 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_2'$ with $\{\vec{m}\} \cap fn(P_1) = \varnothing$.
Here $O = (\nu\vec{m})(P_1' \mid P_2') \mid P_3$. Since the names $\vec{m}$ are bound, we may assume that $fn(P_3) \cap \{\vec{m}\} = \varnothing$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2' \mid P_3))$. By (Comm Par 1), $P_2 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_2'$ implies $P_2 \mid P_3 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_2' \mid P_3$. By (Comm τ-*anon* 2), this and $P_1 \xrightarrow{anon\ M} P_1'$ with $\{\vec{m}\} \cap fn(P_1) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. By (Struct Res Par) and (Struct Assoc), in particular, $O \equiv O'$.

(2) $O = (P_1 \mid P_2) \parallel O''$ where $P_3 \xrightarrow{\alpha} O''$. By (Comm Par 2), $P_3 \xrightarrow{\alpha} O''$ implies $P_2 \mid P_3 \xrightarrow{\alpha} P_2 \parallel O''$. Let $O' = P_1 \parallel (P_2 \parallel O'')$. By (Comm Par 2), $P_2 \mid P_3 \xrightarrow{\alpha} P_2 \parallel O''$ implies $P_1 \mid (P_2 \mid P_3) \xrightarrow{\alpha} O'$. By Lemma 12(4), $O \equiv O'$.

(3) $O = (\nu\vec{m})(P_{12}' \mid P_3')$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_{12}'$ and $P_3 \xrightarrow{enter\ n\ A} P_3'$ with $\{\vec{m}\} \cap fn(P_3) = \varnothing$. Only (Comm Par 1) or (Comm Par 2) may derive $P_1 \mid P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_{12}'$.

In the first case, $P_1 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_1'$ and $P_{12}' = P_1' \mid P_2$ with $\{\vec{m}\} \cap fn(P_2) = \varnothing$. By (Comm Par 2), $P_3 \xrightarrow{enter\ n\ A} P_3'$ implies $P_2 \mid P_3 \xrightarrow{enter\ n\ A} P_2 \mid P_3'$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2 \mid P_3'))$. By (Comm τ-*enter* 1), $P_1 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_1'$ and $P_2 \mid P_3 \xrightarrow{enter\ n\ A} P_2 \mid P_3'$ with $\{\vec{m}\} \cap fn(P_2 \mid P_3) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. We have $O = (\nu\vec{m})((P_1' \mid P_2) \mid P_3')$. By (Struct Res) and (Struct Par Assoc), $O \equiv O'$.

In the second case, $P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_2'$ and $P_{12}' = P_1 \mid P_2'$ with $\{\vec{m}\} \cap fn(P_1) = \varnothing$. By (Comm τ-*enter* 1), $P_2 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_2'$, and $P_3 \xrightarrow{enter\ n\ A} P_3'$ with $\{\vec{m}\} \cap fn(P_3) = \varnothing$ imply $P_2 \mid P_3 \xrightarrow{\tau} (\nu\vec{m})(P_2' \mid P_3')$. By (Comm Par 2), this implies $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$ where $O' = P_1 \mid (\nu\vec{m})(P_2' \mid P_3')$. We have $O = (\nu\vec{m})((P_1 \mid P_2') \mid P_3')$. Since $\{\vec{m}\} \cap fn(P_1) = \varnothing$, we get $O \equiv O'$ using, in particular, (Struct Par Assoc) and (Struct Res Par).

(4) $O = (\nu\vec{m})(P_{12}' \mid P_3')$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{enter\ n\ A} P_{12}'$ and $P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_3'$ with $\{\vec{m}\} \cap fn(P_1 \mid P_2) = \varnothing$. Only (Comm Par 1) or (Comm Par 2) may derive $P_1 \mid P_2 \xrightarrow{enter\ n\ A} P_{12}'$.

In the first case, $P_1 \xrightarrow{enter\ n\ A} P_1'$ and $P_{12}' = P_1' \mid P_2$. By (Comm Par

2), $P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_3'$ and $\{\vec{m}\} \cap fn(P_2) = \varnothing$ imply $P_2 \mid P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle(P_2 \mid P_3')$. Let $O' = (\nu\vec{m})(P_1' \mid (P_2 \mid P_3'))$. By (Comm τ-*enter* 2), $P_1 \xrightarrow{enter\ n\ A} P_1'$ and $P_2 \mid P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle(P_2 \mid P_3')$ with $\{\vec{m}\} \cap fn(P_1) = \varnothing$ imply $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. We have $O = (\nu\vec{m})((P_1' \mid P_2) \mid P_3')$. By (Struct Res) and (Struct Par Assoc), $O \equiv O'$.

In the second case, $P_2 \xrightarrow{enter\ n\ A} P_2'$ and $P_{12}' = P_1 \mid P_2'$. By (Comm τ-*enter* 2), $P_2 \xrightarrow{enter\ n\ A} P_2'$ and $P_3 \xrightarrow{\overline{enter\ n}} (\nu\vec{m})\langle A\rangle P_3'$ with $\{\vec{m}\} \cap fn(P_2) = \varnothing$ imply $P_2 \mid P_3 \xrightarrow{\tau} (\nu\vec{m})(P_2' \mid P_3')$. Let $O' = P_1 \mid (\nu\vec{m})(P_2' \mid P_3')$. By (Comm Par 2), this implies $P_1 \mid (P_2 \mid P_3) \xrightarrow{\tau} O'$. We have $O = (\nu\vec{m})((P_1 \mid P_2') \mid P_3')$. Since $\{\vec{m}\} \cap fn(P_1) = \varnothing$, we get $O \equiv O'$ using, in particular, (Struct Par Assoc) and (Struct Res Par).

(5) $O = P_{12}' \mid P_3'$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{\overline{open\ n}} P_{12}'$ and $P_3 \xrightarrow{open\ n} P_3'$. The proof is analogous to that for case (3).

(6) $O = P_{12}' \mid P_3'$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{open\ n} P_{12}'$ and $P_3 \xrightarrow{\overline{open\ n}} P_3'$. The proof is analogous to that for case (4).

(7) $O = (\nu\vec{m})(P_{12}' \mid P_3')$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_{12}'$ and $P_3 \xrightarrow{anon\ M} P_3'$ with $\{\vec{m}\} \cap fn(P_3') = \varnothing$. The proof is analogous to that for case (3).

(8) $O = (\nu\vec{m})(P_{12}' \mid P_3')$ and $\alpha = \tau$ where $P_1 \mid P_2 \xrightarrow{anon\ M} P_{12}'$ and $P_3 \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P_3'$ with $\{\vec{m}\} \cap fn(P_1 \mid P_2) = \varnothing$. The proof is analogous to that for case (4).

This completes the proof for part (1). Part (2) follows by symmetric reasoning.

(Struct Repl Par) Here $P = !R$ and $Q = R \mid !R$. In part (1), there is O such that $P \xrightarrow{\alpha} O$. This can only be obtained using (Comm Repl). Therefore $R \mid !R \xrightarrow{\alpha} O$, which satisfies part (1). In part (2), there is O such that $Q = R \mid !R \xrightarrow{\alpha} O$. Therefore by (Comm Repl) we obtain $P = !R \xrightarrow{\alpha} O$, which satisfies part (2).

(Struct Res Res) We have $(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$. In part (1), we must show that whenever $(\nu n)(\nu m)P \xrightarrow{\alpha} O$ there is O' with $(\nu m)(\nu n)P \xrightarrow{\alpha} O'$ with $O \equiv O'$. Only (Comm Res) may derive $(\nu n)(\nu m)P \xrightarrow{\alpha} O$, so $O = (\nu n)O_1$ with $(\nu m)P \xrightarrow{\alpha} O_1$ and $n \notin fn(\alpha)$. Again, only (Comm Res) may derive $(\nu m)P \xrightarrow{\alpha} O_1$, so $O_1 = (\nu m)O_2$ with $P \xrightarrow{\alpha} O_2$

with $P \xrightarrow{\alpha} O_2$ and $m \notin fn(\alpha)$. By (Comm Res), $P \xrightarrow{\alpha} O_2$ and $n \notin fn(\alpha)$ imply $(\nu n)P \xrightarrow{\alpha} (\nu n)O_2$. Let $O' = (\nu m)(\nu n)O_2$. By (Comm Res) again, $(\nu n)P \xrightarrow{\alpha} (\nu n)O_2$ and $m \notin fn(\alpha)$ imply $(\nu m)(\nu n)P \xrightarrow{\alpha} (\nu m)(\nu n)O_2$. By (Conc Res Res), $O \equiv O'$. Part (2) follows by symmetric reasoning.

(Struct Res Par) We have $(\nu p)(P \mid Q) \equiv P \mid (\nu p)Q$, with $p \notin fn(P)$. In part (1), we must show that whenever $(\nu p)(P \mid Q) \xrightarrow{\alpha} O$ there is O' with $P \mid (\nu p)Q \xrightarrow{\alpha} O'$ and $O \equiv O'$. Since (Comm Res) is the only rule that may derive $(\nu p)(P \mid Q) \xrightarrow{\alpha} O$ it must be that $p \notin fn(\alpha)$ and there is O'' such that $O = (\nu p)O''$ and $P \mid Q \xrightarrow{\alpha} O''$. According to Lemma 6 there are eight ways in which the latter commitment may be derived.

(1) $O'' = O''' \parallel Q$ where $P \xrightarrow{\alpha} O'''$.
Here $O = (\nu p)(O''' \parallel Q)$. Let $O' = O''' \parallel (\nu p)Q$. By (Comm Par 1), $P \mid (\nu p)Q \xrightarrow{\alpha} O'$. By Lemma 4, $P \xrightarrow{\alpha} O'''$ and $p \notin fn(\alpha) \cup fn(P)$ implies $p \notin fn(O''')$. By Lemma 12(5), $p \notin fn(O''')$ implies $O \equiv O'$.

(2) $O'' = P \parallel O'''$ where $Q \xrightarrow{\alpha} O'''$.
Here $O = (\nu p)(P \parallel O''')$. Let $O' = P \parallel (\nu p)O'''$. By (Comm Res), $p \notin fn(\alpha)$ implies $(\nu p)Q \xrightarrow{\alpha} (\nu p)O'''$. By (Comm Par 2), $P \mid (\nu p)Q \xrightarrow{\alpha} O'$. By Lemma 12(5), $p \notin fn(P)$ implies $O \equiv O'$.

(3) $O'' = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$ where $P \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle P'$ and $Q \xrightarrow{enter\,n\,A} Q'$ with $\{\vec{m}\} \cap fn(Q) = \varnothing$.
Here $O = (\nu p)(\nu\vec{m})(P' \mid Q')$. Let $O' = (\nu\vec{m})(P' \mid (\nu p)Q')$. Since the names $\vec{m}$ are bound, we may assume $p \notin \{\vec{m}\}$. By Proposition 4, $P \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle P'$ implies $\{n\} \cup ((fn(A) \cup fn(P')) - \{\vec{m}\}) \subseteq fn(P)$. Therefore $p \notin fn(P) \cup \{\vec{m}\}$ implies $p \neq n$, $p \notin fn(A)$ and $p \notin fn(P')$. By (Comm Res), $p \notin \{n\} \cup fn(A)$ and $Q \xrightarrow{enter\,n\,A} Q'$ imply $(\nu p)Q \xrightarrow{enter\,n\,A} (\nu p)Q'$. By (Comm τ-*enter* 1), $P \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle P'$ and $(\nu p)Q \xrightarrow{enter\,n\,A} (\nu p)Q'$ with $\{\vec{m}\} \cap fn((\nu p)Q) = \varnothing$ imply $P \mid (\nu p)Q \xrightarrow{\tau} O'$. By (Struct Res Res), $O \equiv (\nu\vec{m})(\nu p)(P' \mid Q')$. By (Struct Res Par), $p \notin fn(P')$ implies $(\nu\vec{m})(\nu p)(P' \mid Q') \equiv O'$. By (Struct Trans), $O \equiv O'$.

(4) $O'' = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$ where $P \xrightarrow{enter\,n\,A} P'$ and $Q \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle Q'$ with $\{\vec{m}\} \cap fn(P) = \varnothing$.
Here $O = (\nu p)(\nu\vec{m})(P' \mid Q')$. By Proposition 4, $P \xrightarrow{enter\,n\,A} P'$ implies $fn(P') = fn(P) \cup fn(A)$ and $n \in fn(P)$. Since $p \notin fn(P)$

this implies $p \neq n$. By (Comm Res), $Q \xrightarrow{\overline{enter\,n}} (\nu\vec{m})\langle A\rangle Q'$ and $p \neq n$ imply $(\nu p)Q \xrightarrow{\overline{enter\,n}} (\nu p)(\nu\vec{m})\langle A\rangle Q'$. By (Comm τ-*enter* 2), $P \xrightarrow{enter\,n\;A} P'$ and $(\nu p)Q \xrightarrow{\overline{enter\,n}} (\nu p)(\nu\vec{m})\langle A\rangle Q'$ with $\{p, \vec{m}\} \cap \mathit{fn}(P) = \varnothing$ imply $P \mid (\nu p)Q \xrightarrow{\tau} (\nu p)(\nu\vec{m})(P' \mid Q')$, that is, $P \mid (\nu p)Q \xrightarrow{\tau} O$.

(5) $O'' = P' \mid Q'$ and $\alpha = \tau$ where $P \xrightarrow{\overline{open\,n}} P'$ and $Q \xrightarrow{open\,n} Q'$.
Here $O = (\nu p)(P' \mid Q')$. Let $O' = P' \mid (\nu p)Q'$. By Proposition 4, $P \xrightarrow{\overline{open\,n}} P'$ implies $\{n\} \cup \mathit{fn}(P') \subseteq \mathit{fn}(P)$. Hence $p \notin \mathit{fn}(P)$ implies $p \neq n$. By (Comm Res), $Q \xrightarrow{open\,n} Q'$ and $p \neq n$ imply $(\nu p)Q \xrightarrow{open\,n} (\nu p)Q'$. By (Comm τ-*open* 1), this and $P \xrightarrow{\overline{open\,n}} P'$ imply $P \mid (\nu p)Q \xrightarrow{\tau} O'$. By (Struct Res Par), $p \notin \mathit{fn}(P')$ implies $O \equiv O'$.

(6) $O'' = P' \mid Q'$ and $\alpha = \tau$ where $P \xrightarrow{open\,n} P'$ and $Q \xrightarrow{\overline{open\,n}} Q'$.
Here $O = (\nu p)(P' \mid Q')$. Let $O' = P' \mid (\nu p)Q'$. By Proposition 4, $P \xrightarrow{open\,n} P'$ implies $\{n\} \cup \mathit{fn}(P') \subseteq \mathit{fn}(P)$. Hence $p \notin \mathit{fn}(P)$ implies $p \neq n$. By (Comm Res), $Q \xrightarrow{\overline{open\,n}} Q'$ and $p \neq n$ imply $(\nu p)Q \xrightarrow{\overline{open\,n}} (\nu p)Q'$. By (Comm τ-*open* 2), this and $P \xrightarrow{open\,n} P'$ imply $P \mid (\nu p)Q \xrightarrow{\tau} O'$. By (Struct Res Par), $p \notin \mathit{fn}(P')$ implies $O \equiv O'$.

(7) $O'' = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$ where $P \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle P'$ and $Q \xrightarrow{anon\,M} Q'$ with $\{\vec{m}\} \cap \mathit{fn}(Q) = \varnothing$.
Similar to part (3).

(8) $O'' = (\nu\vec{m})(P' \mid Q')$ and $\alpha = \tau$ where $P \xrightarrow{anon\,M} P'$ and $Q \xrightarrow{\overline{anon}} (\nu\vec{m})\langle M\rangle Q'$ with $\{\vec{m}\} \cap \mathit{fn}(P) = \varnothing$.
Similar to part (4).

In part (2), we must show that whenever $P \mid (\nu n)Q \xrightarrow{\alpha} O'$ there is O with $(\nu n)(P \mid Q) \xrightarrow{O}$ and $O \equiv O'$. This may be proved by an argument similar to that for part (1).

(Struct Res Amb) We have $(\nu n)m[P] \equiv m[(\nu n)P]$ where $m \neq n$. In part (1), we must show that whenever $(\nu n)m[P] \xrightarrow{\alpha} O$ there is O' with $m[(\nu n)P] \xrightarrow{\alpha} O'$ and $O \equiv O'$. Only (Comm Res) may derive $(\nu n)m[P] \xrightarrow{\alpha} O$. Therefore $O = (\nu n)O_1$ with $m[P] \xrightarrow{\alpha} O_1$ and $n \notin \mathit{fn}(\alpha)$. Commitment $m[P] \xrightarrow{\alpha} O_1$ may be derived in any of the following ways.

(Comm Amb) Here $\alpha = \tau$ and $P \xrightarrow{\tau} P'$ with $O_1 = m[P']$. By (Comm Res), $P \xrightarrow{\tau} P'$ implies $(\nu n)P \xrightarrow{\tau} (\nu n)P'$. By (Comm

Amb), this implies $m[(\nu n)P] \xrightarrow{\tau} m[(\nu n)P']$. By (Struct Res Amb), $m \neq n$ implies $O \equiv (\nu n)m[P] \equiv m[(\nu n)P']$.

(Comm $\overline{enter}$) Here $\alpha = \overline{enter\ p}$, $P \xrightarrow{in\ p} P'$ and $O_1 = \langle m[P'] \rangle \mathbf{0}$. Since $n \notin fn(\alpha)$, $n \notin fn(in\ p)$. Hence by (Comm Res), $P \xrightarrow{in\ p} P'$ implies $(\nu n)P \xrightarrow{in\ p} (\nu n)P'$. By (Comm $\overline{enter}$), this implies $m[(\nu n)P] \xrightarrow{\overline{enter\ p}} m[(\nu n)P']$. By (Struct Res Amb), $m \neq n$ implies $O \equiv m[(\nu n)P']$.

(Comm *enter*) Here $\alpha = enter\ m\ A$ and $O_1 = m[A \mid P]$. Let $O' = m[A \mid (\nu n)P]$. By (Comm *enter*), $m[(\nu n)P] \xrightarrow{enter\ m\ A} O'$. By (Struct Res Amb), $m \neq n$ implies $O = (\nu n)m[A \mid P] \equiv m[(\nu n)(A \mid P)]$. Since $n \notin fn(\alpha)$, $n \notin fn(A)$. By (Struct Res Par), $n \notin fn(A)$ implies $m[(\nu n)(A \mid P)] \equiv m[A \mid (\nu n)P]$. By transitivity, $O \equiv O'$.

(Comm *exit*) Here $\alpha = exit\ p$ and $P \xrightarrow{out\ p} P'$ with $O_1 = \langle m[P'] \rangle \mathbf{0}$. Since $n \notin fn(\alpha)$, $n \neq p$. By (Comm Res), $P \xrightarrow{out\ p} P'$ and $n \neq p$ imply $(\nu n)P \xrightarrow{out\ p} (\nu n)P'$. We have $O = (\nu n)\langle m[P'] \rangle \mathbf{0}$. Let $O' = \langle m[(\nu n)P'] \rangle \mathbf{0}$. By (Comm *exit*), this implies $m[(\nu n)P] \xrightarrow{exit\ p} O'$. By (Conc Res 1), $(\nu n)\langle m[P'] \rangle \mathbf{0} \equiv \langle (\nu n)m[P'] \rangle \mathbf{0}$. By (Conc Res), (Conc Amb) and (Struct Res Amb), $m \neq n$ implies $\langle (\nu n)m[P'] \rangle \mathbf{0} \equiv \langle m[(\nu n)P'] \rangle \mathbf{0}$. By (Conc Trans), we get $(\nu n)\langle m[P'] \rangle \mathbf{0} \equiv \langle m[(\nu n)P'] \rangle \mathbf{0}$, that is, $O \equiv O'$.

(Comm τ-*exit*) Here $\alpha = \tau$ and $P \xrightarrow{exit\ m} (\nu \vec{p})\langle A \rangle P'$ with $m \notin \{\vec{p}\}$ and $O_1 = (\nu \vec{p})(A \mid m[P'])$. So $O = (\nu n)(\nu \vec{p})(A \mid m[P'])$. By (Comm Res), $P \xrightarrow{exit\ m} (\nu \vec{p})\langle A \rangle P'$ and $m \neq n$ imply $(\nu n)P \xrightarrow{exit\ m} (\nu n)(\nu \vec{p})\langle A \rangle P'$. By (Comm τ-*exit*), this implies $m[(\nu n)P] \xrightarrow{\tau} (\nu n)(\nu \vec{p})(A \mid m[P'])$.

(Comm $\overline{open}$) Here $\alpha = \overline{open\ m}$ and $O_1 = P$. So $O = (\nu n)P$. By (Comm $\overline{open}$), $m[(\nu n)P] \xrightarrow{\overline{open\ m}} (\nu n)P$.

In part (2), we must show that whenever $m[(\nu n)P] \xrightarrow{\alpha} O'$ there is O with $(\nu n)m[P] \xrightarrow{\alpha} O'$ and $O \equiv O'$. This may be showed by arguments symmetric to those for part (1).

(Struct Zero Par) Here $P = Q \mid \mathbf{0}$. In part (1), the commitment $P \xrightarrow{\alpha} O$ can only be obtained by (Comm Par 1), since $\mathbf{0}$ has no commitments. Therefore there is O' such that $Q \xrightarrow{\alpha} O'$ and $O = O' \parallel \mathbf{0}$. By definition of $\parallel$ and (Struct Zero Par) we obtain $O \equiv O'$, to complete part (1).

In part (2), from $Q \xrightarrow{\alpha} O'$ we obtain by (Comm Par 1) that $P = Q \mid \mathbf{0} \xrightarrow{\alpha} O' \parallel \mathbf{0}$. As in part (1), we get $O' \parallel \mathbf{0} \equiv O'$.

(Struct Zero Res) Here $P = (\nu n)\mathbf{0}$ and $Q = \mathbf{0}$. Therefore (1) and (2) follow vacuously, since neither P nor Q has any commitments.

(Struct Zero Repl) Here $P = !\mathbf{0}$ and $Q = \mathbf{0}$. Again, (1) and (2) follow vacuously, since neither P nor Q has any commitments.

(Struct ϵ) Here $P = \epsilon.Q$. Suppose $P \xrightarrow{\alpha} O$. This commitment may only be obtained by (Comm ϵ), from $Q \xrightarrow{\alpha} O$. On the other hand, suppose $Q \xrightarrow{\alpha} O$. Then by (Comm ϵ) we obtain $P \xrightarrow{\alpha} O$ also. Hence parts (1) and (2) are satisfied.

(Struct .) Here $P = (M.M').R$ and $Q = M.M'.R$. Suppose $P \xrightarrow{\alpha} O$. This can only be obtained by (Comm .), from $Q \xrightarrow{\alpha} O$. On the other hand, suppose $Q \xrightarrow{\alpha} O$. By (Comm .) we obtain $P \xrightarrow{\alpha} O$ also. Hence parts (1) and (2) are satisfied. $\square$

A corollary is that for all closed processes P and Q, $P \equiv Q$ implies $P \sim Q$. The purpose of all the lemmas in this section is to prove the following:

Proposition 15 *$P \to Q$ if and only if there is R with $P \xrightarrow{\tau} R$ and $R \equiv Q$.*

Proof For the backwards direction suppose $P \xrightarrow{\tau} R$ and $R \equiv Q$. By Lemma 8, $P \to R$, and then $P \to Q$ by (Red $\equiv$).

We can show that $P \to Q$ implies that there exists R such that $P \xrightarrow{\tau} R$ and $R \equiv Q$ by induction on the derivation of $P \to Q$. The only interesting case is (Red $\equiv$). Suppose that $P \to Q$ follows from $P \equiv P'$, $P' \to Q'$, and $Q' \equiv Q$. By induction hypothesis, $P' \xrightarrow{\tau} Q''$ with $Q'' \equiv Q'$. By Lemma 14, there is R such that $P \xrightarrow{\tau} R$ and $R \equiv Q''$. This with the previous equations gives $R \equiv Q$ as required. $\square$

3 Convergence and Contextual Equivalence

Strong bisimilarity is rather a fine-grained equivalence; instead of strong bisimilarity, in this section, we adopt Morris-style contextual equivalence as our notion of observational equivalence for the ambient calculus. We introduce a notion of a process converging to a name in Section 3.1, and use this notion in Section 3.2 to define contextual equivalence for the ambient calculus. In Section 3.3 we prove a non-trivial instance of testing equivalence, the Perfect Firewall Equation, that $(\nu n)n[P]$ is contextually equivalent to $\mathbf{0}$, if $n \notin fn(P)$.

3.1 Convergence

Let us say that a process P *exhibits* n just if P is a process with a top-level ambient named n. Formally, we define the predicate $P \downarrow n$, by the rules:

Exhibition of a Name

(Exh Amb) $\dfrac{}{n[P] \downarrow n}$ (Exh Par) $\dfrac{P \downarrow n}{P \mid Q \downarrow n}$ (Exh Res) $\dfrac{P \downarrow n \quad m \neq n}{(\nu m)P \downarrow n}$ (Exh $\equiv$) $\dfrac{P \equiv Q \quad Q \downarrow n}{P \downarrow n}$

Lemma 16 *$P \downarrow n$ if and only if there are distinct names $\vec{m}$ with $n \notin \{\vec{m}\}$ and processes P_1 and P_2 such that $P \equiv (\nu\vec{m})(P_1 \mid n[P_2])$.*

Proof The left-to-right implication follows by an easy induction on the derivation of $P \downarrow n$. The right-to-left implication is obvious. $\Box$

Let us say that a process P *converges to a name* n just if after some number of reductions, P exhibits n. Formally, we define the predicate $P \Downarrow n$, by the rules:

Convergence to a Name

(Conv Exh) $\dfrac{P \downarrow n}{P \Downarrow n}$ (Conv Red) $\dfrac{P \rightarrow Q \qquad Q \Downarrow n}{P \Downarrow n}$

Structural congruence preserves exhibition of or convergence to a name.

Lemma 17 *Suppose $P \equiv Q$.*

(1) *If $P \downarrow n$ then $Q \downarrow n$.*

(2) *If $P \Downarrow n$ then $Q \Downarrow n$.*

Proof Part (1) follows from (Exh $\equiv$). Part (2) is by induction on the derivation of $P \Downarrow n$. $\Box$

Next, we relate exhibition of a name, and convergence to a name, to the commitment relation.

Proposition 18 *$P \downarrow n$ if and only if there is Q such that $P \xrightarrow{\overline{open\ n}} Q$.*

Proof For the left-to-right direction, by Lemma 16, $P \downarrow n$ implies there are names $\vec{p}$ with $n \notin \{\vec{p}\}$ such that $P \equiv (\nu\vec{p})(P_1 \mid n[P_2])$. By (Comm Res), (Comm Par) and (Comm $\overline{open}$) we get $(\nu\vec{p})(P_1 \mid n[P_2]) \xrightarrow{\overline{open}} (\nu\vec{p})(P_1 \mid P_2)$. By Lemma 14, this and $P \equiv (\nu\vec{p})(P_1 \mid n[P_2])$ imply there is Q such that $P \xrightarrow{\overline{open}} Q$.

For the right-to-left direction, we may prove by induction on the derivation of $P \xrightarrow{\overline{open\ n}} Q$ that $P \downarrow n$. $\square$

Lemma 19 *Suppose $P \to P_1 \to \cdots \to P_n$. Then there are $Q_1, \ldots, Q_n$ such that $P \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} Q_n \equiv P_n$.*

Proof By induction on n. The base case is trivial. For the inductive case, suppose that $P \to P_1 \to \cdots \to P_n \to P_{n+1}$. By induction hypothesis, there are $Q_1, \ldots, Q_n$ such that $P \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} Q_n \equiv P_n$. By (Red $\equiv$), $Q_n \equiv P_n$ and $P_n \to P_{n+1}$ imply $Q_n \to P_{n+1}$. By Proposition 15, there is Q_{n+1} such that $Q_n \xrightarrow{\tau} Q_{n+1}$ and $Q_{n+1} \equiv P_{n+1}$, to complete the proof. $\square$

Proposition 20 *$P \Downarrow n$ if and only if there are processes Q and R such that $P \xrightarrow{\tau}{}^* Q$ and $Q \xrightarrow{\overline{open\ n}} R$.*

Proof For the left-to-right direction, $P \Downarrow n$ implies there are processes P_1, $\ldots$, P_n such that $P \to P_1 \to \cdots \to P_n \downarrow n$. By Lemma 19, $P \to P_1 \to \cdots \to P_n$ implies there are $Q_1, \ldots, Q_n$ such that $P \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} Q_n \equiv P_n$. By (Exh $\equiv$), $Q_n \equiv P_n$ and $P_n \downarrow n$ imply that $Q_n \downarrow n$. By Proposition 18, this implies there is R such that $Q_n \xrightarrow{\overline{open\ n}} R$.

For the right-to-left direction, $Q \xrightarrow{\overline{open\ n}} R$ implies $Q \downarrow n$ by Proposition 18. By (Conv Exh), $Q \downarrow n$ implies $Q \Downarrow n$. By repeated use of Lemma 8, $P \xrightarrow{\tau}{}^* Q$ implies $P \to^* Q$. By repeated use of (Conv Red), $P \to^* Q$ and $Q \Downarrow n$ imply $P \Downarrow n$. $\square$

3.2 Contextual Equivalence

Let a *context*, $\mathcal{C}$ or $\mathcal{D}$, be a process containing some number of *holes* in place of subprocesses, where a hole is written $\bullet$. Unlike processes, contexts are not identified up to renaming of bound names (or variables). If $\mathcal{C}$ is a context and P is a process, let $\mathcal{C}\{P\}$ be the process obtained by filling each hole in $\mathcal{C}$ with P. Hole-filling may result in a name free in P becoming bound in $\mathcal{C}\{P\}$. For example, n is free in $n[]$ but becomes bound in $\mathcal{C}\{n[]\}$ where $\mathcal{C} = (\nu n)\bullet$. Let $fn(\mathcal{C})$ be the set of names free in $\mathcal{C}$.

- Let us say that process P *converges*, written $P\Downarrow$ if and only if there is a name n such that $P \Downarrow n$.
- Let *contextual order*, $\sqsubseteq$, be the relation on processes such that $P \simeq Q$ if and only if for all contexts $\mathcal{C}$, $\mathcal{C}\{P\}\Downarrow$ implies $\mathcal{C}\{Q\}\Downarrow$.
- Let *contextual equivalence*, $\simeq$, be the relation on processes such that $P \simeq Q$ if and only if for all contexts $\mathcal{C}$, $\mathcal{C}\{P\}\Downarrow \Leftrightarrow \mathcal{C}\{Q\}\Downarrow$.

Clearly $P \simeq Q$ if and only if $P \sqsubseteq Q$ and $Q \sqsubseteq P$. Moreover:

Proposition 21

(1) *Relation $\simeq$ is reflexive, transitive and symmetric.*

(2) *For all P, Q and $\mathcal{C}$, $P \simeq Q$ implies $\mathcal{C}\{P\} \simeq \mathcal{C}\{Q\}$.*

(3) *For all P and Q, $P \equiv Q$ implies $P \simeq Q$.*

Proof Parts (1) and (2) are standard. For (3), suppose $P \equiv Q$ and consider any context $\mathcal{C}$ and any name n such that $\mathcal{C}\{P\} \Downarrow n$. Since $\equiv$ is a congruence, $P \equiv Q$ implies $\mathcal{C}\{P\} \equiv \mathcal{C}\{Q\}$. By Lemma 17, this and $\mathcal{C}\{P\} \Downarrow n$ implies $\mathcal{C}\{Q\} \Downarrow n$. Similarly, we can show that for all $\mathcal{C}$ and n, $\mathcal{C}\{Q\} \Downarrow n$ implies $\mathcal{C}\{P\} \Downarrow n$. Hence $P \simeq Q$. $\square$

The next proposition provides an alternative characterisation of contextual equivalence. First, we need a technical lemma.

Lemma 22 *Suppose $m \notin fn(P)$. Then, whenever $m[P \mid open\ n.Q] \xrightarrow{\tau} R$, there is P' such that either:*

(1) *$P \xrightarrow{\tau} P'$ and $R = m[P' \mid open\ n.Q]$, or*

(2) *$P \xrightarrow{\overline{open\ n}} P'$ and $R = m[P' \mid Q]$.*

Proof Suppose that $m[P \mid open\ n.Q] \xrightarrow{\tau} R$. The only rules that allow a τ-commitment to be derived from an ambient are (Comm Amb) or (Comm τ-*exit*). But (Comm τ-*exit*) cannot apply, since, by Lemma 4, $m \notin fn(P)$ implies that $P \mid open\ n.Q$ has no $(exit\ m)$-commitments. So $m[P \mid open\ n.Q] \xrightarrow{\tau} R$ must derive from (Comm Amb). Hence there is R' such that $P \mid open\ n.Q \xrightarrow{\tau} R'$ and $R = m[R']$. The only commitment of $open\ n.Q$ is $open\ n.Q \xrightarrow{open\ n} Q$. Therefore, the only rules that may derive $P \mid open\ n.Q \xrightarrow{\tau} R'$ are as follows.

(Comm Par 1) There is P' such that $P \xrightarrow{\tau} P'$ and $R' = P' \mid \mathit{open}\, n.Q$. In this case, $R = m[R'] = m[P' \mid \mathit{open}\, n.Q]$, thus fulfilling condition (1).

(Comm τ-*open* 1) There is P' such that $P \xrightarrow{\overline{\mathit{open}\, n}} P'$ and $R' = P' \mid Q$. In this case, $R = m[P' \mid Q]$, thus fulfilling condition (2). $\Box$

Proposition 23 *For all P and Q, $P \simeq Q$ if and only if for all n, $\mathcal{C}\{P\} \Downarrow n \Leftrightarrow \mathcal{C}\{Q\} \Downarrow n$.*

Proof The easy direction is that $P \simeq Q$ holds if for all n, $\mathcal{C}\{P\} \Downarrow n \Leftrightarrow \mathcal{C}\{Q\} \Downarrow n$. For the other direction, suppose that $P \simeq Q$ and that for some n, $\mathcal{C}\{P\} \Downarrow n$. Pick names m and o such that $\{m, o\} \cap (\mathit{fn}(P) \cup \mathit{fn}(Q) \cup \mathit{fn}(\mathcal{C})) = \varnothing$ and $m \neq o$. Let context $\mathcal{D}$ be:

$$\mathcal{D} \quad = \quad (\nu m)m[\mathcal{C} \mid \mathit{open}\, n.o[\mathit{out}\, m]]$$

By Lemma 16, $\mathcal{C}\{P\} \Downarrow n$ implies there are distinct names $\vec{p}$ with $n \notin \{\vec{p}\}$ and processes P_1 and P_2 such that $\mathcal{C}\{P\} \rightarrow^* (\nu\vec{p})(P_1 \mid n[P_2])$. Hence we may calculate:

$$\begin{array}{rcl}
\mathcal{D}\{P\} & = & (\nu m)m[\mathcal{C}\{P\} \mid \mathit{open}\, n.o[\mathit{out}\, m]] \\
& \rightarrow^* & (\nu m)m[(\nu\vec{p})(P_1 \mid n[P_2]) \mid \mathit{open}\, n.o[\mathit{out}\, m]] \\
& \rightarrow & (\nu m)m[(\nu\vec{p})(P_1 \mid P_2) \mid o[\mathit{out}\, m]] \\
& \rightarrow & (\nu m)m[(\nu\vec{p})(P_1 \mid P_2)] \mid o[] \\
& \downarrow & o
\end{array}$$

By Proposition 21(2), $P \simeq Q$ implies $\mathcal{D}\{P\} \simeq \mathcal{D}\{Q\}$. Therefore $\mathcal{D}\{P\}\Downarrow$ implies $\mathcal{D}\{Q\}\Downarrow$. By Proposition 20, $\mathcal{D}\{Q\}\Downarrow$ implies there are processes R and R', and a name p, such that $\mathcal{D}\{Q\} \xrightarrow{\tau}{}^* R \xrightarrow{\mathit{open}\, p} R'$. By Lemma 5, this implies $p \neq m$ and there are $\hat{R}$ and $\hat{R}'$ with $R = (\nu m)\hat{R}$ and $R' = (\nu m)\hat{R}'$ such that:

$$m[\mathcal{C}\{Q\} \mid \mathit{open}\, n.o[\mathit{out}\, m]] \xrightarrow{\tau}{}^* \hat{R} \xrightarrow{\mathit{open}\, p} \hat{R}'$$

Since $m \notin \mathit{fn}(\mathcal{C}\{Q\})$, Lemma 22 implies there are Q' and Q'' such that

$$\begin{array}{c}
\mathcal{C}\{Q\} \xrightarrow{\tau}{}^* Q' \xrightarrow{\overline{\mathit{open}\, n}} Q'' \\
m[Q'' \mid o[\mathit{out}\, m]] \xrightarrow{\tau}{}^* \hat{R} \xrightarrow{\mathit{open}\, p} \hat{R}'
\end{array}$$

By Proposition 20, $\mathcal{C}\{Q\} \xrightarrow{\tau}{}^* Q' \xrightarrow{\overline{\mathit{open}\, n}} Q''$ implies $\mathcal{C}\{Q\} \Downarrow n$, as required. $\Box$

In the π-calculus, if we define $P \downarrow n$ to mean that P is ready to input or output on channel n, then our definition of contextual equivalence coincides with De Nicola and Hennessy's may-testing equivalence [2] since the latter is preserved by all contexts. In the ambient calculus, however, testing equivalence is weaker than contextual equivalence, and is not preserved by all contexts. To see this, let processes P and Q be *testing equivalent* if and only if for all processes R and all names n, $(P \mid R) \Downarrow n \Leftrightarrow (Q \mid R) \Downarrow n$. Testing equivalence is not a congruence; $\mathit{out}\ n.m[]$ and $\mathbf{0}$ are testing equivalent, vacuously, but $(\nu n)n[\mathit{out}\ n.m[]]$ and $(\nu n)n[\mathbf{0}]$ are not, since $(\nu n)n[\mathit{out}\ n.m[]]$ but not $(\nu n)n[\mathbf{0}]$ may converge to m.

We appeal to the following fact in the next section.

Lemma 24 *For all P, $\mathbf{0} \sqsubseteq P$.*

Proof First, note the following facts, which may be proved by simultaneous induction on the derivation of the three judgments $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\alpha} (\nu\vec{n})\langle A\rangle Q$, $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\mathit{enter}\ a\ \mathcal{D}\{\mathbf{0}\}} Q$ and $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\alpha} Q$.

(1) $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\alpha} (\nu\vec{n})\langle A\rangle Q$ implies there are $\mathcal{D}$ and $\mathcal{C}'$ such that $A = \mathcal{D}\{\mathbf{0}\}$ and $Q = \mathcal{C}'\{\mathbf{0}\}$, and for any P, $\mathcal{C}\{P\} \xrightarrow{\alpha} (\nu\vec{n})\langle\mathcal{D}\{P\}\rangle\mathcal{C}'\{P\}$.

(2) $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\mathit{enter}\ a\ \mathcal{D}\{\mathbf{0}\}} Q$ implies there is $\mathcal{C}'$ such that $Q = \mathcal{C}'\{\mathbf{0}\}$ and, for any P, $\mathcal{C}\{P\} \xrightarrow{\mathit{enter}\ a\ \mathcal{D}\{P\}} \mathcal{C}'\{P\}$.

(3) $\mathcal{C}\{\mathbf{0}\} \xrightarrow{\alpha} Q$ and $\alpha \neq \mathit{enter}\ a\ A$ implies there is $\mathcal{C}'$ such that $Q = \mathcal{C}'\{\mathbf{0}\}$ and, for any P, $\mathcal{C}\{P\} \xrightarrow{\alpha} \mathcal{C}'\{P\}$.

By definition, we need to show, for any context $\mathcal{C}$, that $\mathcal{C}\{\mathbf{0}\}\Downarrow$ implies $\mathcal{C}\{P\}\Downarrow$. Now, pick an arbitrary context such that $\mathcal{C}\{\mathbf{0}\}\Downarrow$. By Proposition 20, $\mathcal{C}\{\mathbf{0}\}\Downarrow$ implies there are processes $Q_1, \ldots, Q_n, Q_{n+1}$ such that:

$$\mathcal{C}\{\mathbf{0}\} \xrightarrow{\tau} Q_1 \xrightarrow{\tau} \cdots \xrightarrow{\tau} Q_n \xrightarrow{\overline{\mathit{open}\ n}} Q_{n+1}$$

By fact (3) above, there are contexts $\mathcal{C}_1, \ldots, \mathcal{C}_n, \mathcal{C}_{n+1}$ such that:

$$\mathcal{C}\{\mathbf{0}\} \xrightarrow{\tau} \mathcal{C}_1\{\mathbf{0}\} \xrightarrow{\tau} \cdots \xrightarrow{\tau} \mathcal{C}_n\{\mathbf{0}\} \xrightarrow{\overline{\mathit{open}\ n}} \mathcal{C}_{n+1}\{\mathbf{0}\}$$

and

$$\mathcal{C}\{P\} \xrightarrow{\tau} \mathcal{C}_1\{P\} \xrightarrow{\tau} \cdots \xrightarrow{\tau} \mathcal{C}_n\{P\} \xrightarrow{\overline{\mathit{open}\ n}} \mathcal{C}_{n+1}\{P\}$$

By Proposition 20, we may conclude $\mathcal{C}_n\{P\}\Downarrow$. We have shown for an arbitrary $\mathcal{C}$ that $\mathcal{C}\{\mathbf{0}\}\Downarrow$ implies $\mathcal{C}\{P\}\Downarrow$. Therefore $\mathbf{0} \sqsubseteq P$. $\Box$

3.3 The Perfect Firewall Equation

We prove in this section a Perfect Firewall Equation, that for any process P and any name $n \notin fn(P)$, $(\nu n)n[P] \simeq \mathbf{0}$.

We begin by defining a family of relations, $\mathcal{S}_e$, where e is a finite set of names. We intend that $P \mathrel{\mathcal{S}_e} Q$ implies P and Q are almost identical, except that for any name $n \in e$, any occurrence of n in P takes the form $n[R]$, with $n \notin fn(R)$, and the corresponding position in Q is filled with $\mathbf{0}$. We can prove the firewall equation by showing that the relationship $P \mathrel{\mathcal{S}_e} Q$ is preserved by structural congruence and reduction.

Let e range over finite sets of names. Let $\mathcal{S}_e$ be the smallest relation on processes given by the following rules.

Relation $\mathcal{S}_e$ on Processes

($\mathcal{S}_e$ Firewall)
$$\frac{n \in e \quad fn(P) \cap e = \varnothing}{n[P] \mathrel{\mathcal{S}_e} \mathbf{0}}$$

($\mathcal{S}_e$ Res) (e' either e or $e \cup \{m\}$)
$$\frac{P \mathrel{\mathcal{S}_{e'}} Q \quad m \notin e}{(\nu m)P \mathrel{\mathcal{S}_e} (\nu m)Q}$$

($\mathcal{S}_e$ $\mathbf{0}$)
$$\frac{}{\mathbf{0} \mathrel{\mathcal{S}_e} \mathbf{0}}$$

($\mathcal{S}_e$ Par)
$$\frac{P \mathrel{\mathcal{S}_e} Q \quad P' \mathrel{\mathcal{S}_e} Q'}{P \mid P' \mathrel{\mathcal{S}_e} Q \mid Q'}$$

($\mathcal{S}_e$ Repl)
$$\frac{P \mathrel{\mathcal{S}_e} Q}{!P \mathrel{\mathcal{S}_e} !Q}$$

($\mathcal{S}_e$ Amb)
$$\frac{P \mathrel{\mathcal{S}_e} Q \quad m \notin e}{m[P] \mathrel{\mathcal{S}_e} m[Q]}$$

($\mathcal{S}_e$ Action)
$$\frac{P \mathrel{\mathcal{S}_e} Q \quad fn(M) \cap e = \varnothing}{M.P \mathrel{\mathcal{S}_e} M.Q}$$

($\mathcal{S}_e$ Input)
$$\frac{P \mathrel{\mathcal{S}_e} Q}{(x).P \mathrel{\mathcal{S}_e} (x).Q}$$

($\mathcal{S}_e$ Output)
$$\frac{fn(M) \cap e = \varnothing}{\langle M \rangle \mathrel{\mathcal{S}_e} \langle M \rangle}$$

Lemma 25 *If $P \mathrel{\mathcal{S}_e} Q$ and $fn(M) \cap e = \varnothing$ then $P\{x \leftarrow M\} \mathrel{\mathcal{S}_e} Q\{x \leftarrow M\}$.*

Proof By induction on the derivation of $P \mathrel{\mathcal{S}_e} Q$. □

Let $\equiv'$ be the smallest relation to include all the equations of $\equiv$, and symmetric versions of all of the equations of $\equiv$, and to be closed under all the inference rules of $\equiv$, minus (Struct Symm), plus a symmetric version of (Struct Par).

Lemma 26 *$P \equiv Q$ if and only if $P \equiv' Q$.*

Proof The easy direction is that $P \equiv' Q$ implies $P \equiv Q$. The other direction, that $P \equiv Q$ implies $P \equiv' Q$, is a corollary of proving by induction on the derivation of $P \equiv Q$ that both $P \equiv' Q$ and $Q \equiv' P$. □

We need this alternative characterization of $\equiv$ simply for the following lemma.

Lemma 27 *Suppose $P \; \mathcal{S}_e \; Q$. Then:*

(1) *$P \equiv' P'$ implies there is Q' with $Q \equiv' Q'$ and $P' \; \mathcal{S}_e \; Q'$.*

(2) *$Q \equiv' Q'$ implies there is P' with $P \equiv' P'$ and $P' \; \mathcal{S}_e \; Q'$.*

Proof We show part (1) by induction on the derivation of $P \equiv' P'$.

(Struct Refl) This case, in which $P' = P$, is solved by setting $Q' = Q$.

(Struct Trans) In this case, $P \equiv' P''$ and $P'' \equiv' P'$. By induction hypothesis, $P \; \mathcal{S}_e \; Q$ and $P \equiv' P''$ imply there is Q'' such that $Q \equiv' Q''$ and $P'' \; \mathcal{S}_e \; Q''$. By induction hypothesis, $P'' \; \mathcal{S}_e \; Q''$ and $P'' \equiv' P'$ imply there is Q' such that $Q'' \equiv' Q'$ and $P' \; \mathcal{S}_e \; Q'$. By (Struct Trans), $Q \equiv' Q'$.

(Struct Res) In this case, $P = (\nu m)\hat{P}$, $\hat{P} \equiv' \hat{P}'$ and $P' = (\nu m)\hat{P}'$. Only ($\mathcal{S}_e$ Res) can derive $(\nu m)\hat{P} \; \mathcal{S}_e \; Q$. Therefore $m \notin e$ and there is e', either e or $e \cup \{m\}$, and $\hat{Q}$ such that $Q = (\nu m)\hat{Q}$ with $\hat{P} \; \mathcal{S}_{e'} \; \hat{Q}$. By induction hypothesis, $\hat{P} \; \mathcal{S}_{e'} \; \hat{Q}$ and $\hat{P} \equiv' \hat{P}'$ imply there is $\hat{Q}'$ such that $\hat{Q} \equiv' \hat{Q}'$ and $\hat{P}' \; \mathcal{S}_{e'} \; \hat{Q}'$. Let $Q' = (\nu m)\hat{Q}'$. By ($\mathcal{S}_e$ Res), $\hat{P}' \; \mathcal{S}_{e'} \; \hat{Q}'$ and $m \notin e$ imply $P' \; \mathcal{S}_e \; Q'$. By (Struct Res), $\hat{Q} \equiv' \hat{Q}'$ implies $Q \equiv' Q'$.

(Struct Par) This rule, and its symmetrization, follow by an argument similar to that for (Struct Res).

(Struct Amb) In this case, $P = m[\hat{P}]$, $\hat{P} \equiv' \hat{P}'$ and $P' = m[\hat{P}']$. Either ($\mathcal{S}_e$ Firewall) or ($\mathcal{S}_e$ Amb) can derive $m[\hat{P}] \; \mathcal{S}_e \; Q$.

In case ($\mathcal{S}_e$ Amb), $Q = m[\hat{Q}]$, $\hat{P} \; \mathcal{S}_e \; \hat{Q}$ and $m \notin e$. By induction hypothesis, $\hat{P} \; \mathcal{S}_e \; \hat{Q}$ and $\hat{P} \equiv' \hat{P}'$ imply there is $\hat{Q}'$ such that $\hat{Q} \equiv' \hat{Q}'$ and $\hat{P}' \; \mathcal{S}_e \; \hat{Q}'$. Let $Q' = m[\hat{Q}']$. By ($\mathcal{S}_e$ Amb), $\hat{P}' \; \mathcal{S}_e \; \hat{Q}'$ and $m \notin e$ imply $P' \; \mathcal{S}_e \; Q'$. By (Struct Amb), $\hat{Q} \equiv' \hat{Q}'$ implies $Q \equiv' Q'$.

In case ($\mathcal{S}_e$ Firewall), $m \in e$, $fn(\hat{P}) \cap e = \varnothing$ and $Q = \mathbf{0}$. By Lemma 1, $\hat{P} \equiv' \hat{P}'$ implies $fn(\hat{P}) = fn(\hat{P}')$. Therefore $fn(\hat{P}') \cap e = \varnothing$, so by ($\mathcal{S}_e$ Firewall), $P' = m[\hat{P}'] \; \mathcal{S}_e \; Q$.

(Struct Res Amb) We treat the equation and then its symmetrization:

- In this case, $P = (\nu m)p[\hat{P}]$, $P' = p[(\nu m)\hat{P}]$ and $m \neq p$. Only ($\mathcal{S}_e$ Res) can derive $(\nu m)p[\hat{P}]\ \mathcal{S}_e\ Q$. Therefore $m \notin e$ and there is e', either e or $e \cup \{m\}$, and R, such that $Q = (\nu m)R$ and $p[\hat{P}]\ \mathcal{S}_{e'}\ R$. Either ($\mathcal{S}_{e'}$ Firewall) or ($\mathcal{S}_{e'}$ Amb) can derive $p[\hat{P}]\ \mathcal{S}_{e'}\ R$.

  In case ($\mathcal{S}_{e'}$ Firewall), $p \in e'$, $fn(\hat{P}) \cap e' = \varnothing$ and $R = \mathbf{0}$. By (Struct Zero Res), $Q = (\nu m)\mathbf{0} \equiv' \mathbf{0}$. Since $fn(\hat{P}) \cap e' = \varnothing$ we get $fn((\nu m)\hat{P}) \cap e = \varnothing$ too. Moreover, $m \neq p$ and $p \in e'$ implies $p \in e$, whether e' is e or $e \cup \{m\}$. Therefore, by ($\mathcal{S}_e$ Firewall), $p \in e$ and $fn((\nu m)\hat{P}) \cap e = \varnothing$ implies $P' = p[(\nu m)\hat{P}]\ \mathcal{S}_e\ \mathbf{0}$.

  In case ($\mathcal{S}_{e'}$ Amb), $p \notin e'$ and $R = p[\hat{Q}]$, $\hat{P}\ \mathcal{S}_{e'}\ \hat{Q}$, and $Q = (\nu m)p[\hat{Q}]$. Let $Q' = p[(\nu m)\hat{Q}]$. By (Struct Res Amb), $m \neq p$ implies $Q \equiv' Q'$. By ($\mathcal{S}_e$ Res), $\hat{P}\ \mathcal{S}_{e'}\ \hat{Q}$ and $m \notin e$ imply $(\nu m)\hat{P}\ \mathcal{S}_e\ (\nu m)\hat{Q}$. Since $e \subseteq e'$, $p \notin e'$ implies $p \notin e$. By ($\mathcal{S}_{e'}$ Amb), $(\nu m)\hat{P}\ \mathcal{S}_e\ (\nu m)\hat{Q}$ and $p \notin e$ imply $p[(\nu m)\hat{P}]\ \mathcal{S}_e\ p[(\nu m)\hat{Q}]$, that is, $P'\ \mathcal{S}_e\ Q'$.

- In this case, $P = p[(\nu m)\hat{P}]$, $P' = (\nu m)p[\hat{P}]$, and $m \neq p$. Either ($\mathcal{S}_e$ Firewall) or ($\mathcal{S}_e$ Amb) can derive $p[(\nu m)\hat{P}]\ \mathcal{S}_e\ Q$.

  In case ($\mathcal{S}_e$ Firewall), $p \in e$, $fn((\nu m)\hat{P}) \cap e = \varnothing$ and $Q = \mathbf{0}$. We may assume the bound name $m \notin e$ and hence $p \neq m$. Let $Q' = (\nu m)\mathbf{0}$. By the symmetrization of (Struct Zero Res), $Q \equiv Q'$. By ($\mathcal{S}_e$ Firewall), $p[\hat{P}]\ \mathcal{S}_e\ \mathbf{0}$, since we have $p \in e$, and $fn((\nu m)\hat{P}) \cap e = \varnothing$ and $m \notin e$ imply that $fn(\hat{P}) \cap e = \varnothing$. By ($\mathcal{S}_e$ Res), $m \notin e$ implies that $(\nu m)p[\hat{P}]\ \mathcal{S}_e\ (\nu m)\mathbf{0}$, that is, $P'\ \mathcal{S}_e\ Q'$.

  In case ($\mathcal{S}_e$ Amb), $Q = p[R]$, $p \notin e$ and $(\nu m)\hat{P}\ \mathcal{S}_e\ R$. Only ($\mathcal{S}_e$ Res) can derive $(\nu m)\hat{P}\ \mathcal{S}_e\ R$. Therefore $m \notin e$ and there is e', either e or $e \cup \{m\}$, and $\hat{Q}$, such that $R = (\nu m)\hat{Q}$ and $\hat{P}\ \mathcal{S}_{e'}\ \hat{Q}$. So $Q = p[(\nu m)\hat{Q}]$. Let $Q' = (\nu m)p[\hat{Q}]$. By the symmetrization of (Struct Res Amb), $m \neq p$ implies $Q \equiv' Q'$. Since $p \notin e$ and $m \neq p$, $p \notin e'$. By ($\mathcal{S}_{e'}$ Amb), $\hat{P}\ \mathcal{S}_{e'}\ \hat{Q}$ and $p \notin e'$ imply $p[\hat{P}]\ \mathcal{S}_{e'}\ p[\hat{Q}]$. By ($\mathcal{S}_e$ Res), this and $m \notin e$ imply $(\nu m)p[\hat{P}]\ \mathcal{S}_e\ (\nu m)p[\hat{Q}]$, that is, $P'\ \mathcal{S}_e\ Q'$.

The omitted cases are for equations, and their symmetrizations, that do not mention ambients. These cases are all straightforward.

This completes the proof of part (1); part (2), which is symmetric, follows by symmetric reasoning. $\square$

Lemma 28 *Whenever $P\ \mathcal{S}_e\ Q$ and $P \to P'$ there is Q' such that $P'\ \mathcal{S}_e\ Q'$ and either $Q \to Q'$ or $Q \equiv Q'$.*

Proof By induction on the derivation of $P \to P'$.

(Red In) In this case $P = m[in\ p.P_1 \mid P_2] \mid p[P_3]$ and $P' = p[m[P_1 \mid P_2] \mid P_3]$. Only ($\mathcal{S}_e$ Par) can derive $P\ \mathcal{S}_e\ Q$, so there are R_1 and R_2 with $m[in\ p.P_1 \mid P_2]\ \mathcal{S}_e\ R_1$, $p[P_3]\ \mathcal{S}_e\ R_2$ and $Q = R_1 \mid R_2$. Either ($\mathcal{S}_e$ Firewall) or ($\mathcal{S}_e$ Amb) can derive $m[in\ p.P_1 \mid P_2]\ \mathcal{S}_e\ R_1$.

In case ($\mathcal{S}_e$ Firewall), $m \in e$, $fn(in\ p.P_1 \mid P_2) \cap e = \varnothing$ (and therefore $p \notin e$) and $R_1 = \mathbf{0}$. Since $p \notin e$, $p[P_3]\ \mathcal{S}_e\ R_2$ must be derived by ($\mathcal{S}_e$ Amb), and not by ($\mathcal{S}_e$ Firewall), so there is Q_3 such that $R_2 = p[Q_3]$ and $P_3\ \mathcal{S}_e\ Q_3$. Therefore $Q = \mathbf{0} \mid p[Q_3]$. Let $Q' = p[\mathbf{0} \mid Q_3]$ and we have $Q \equiv Q'$. By ($\mathcal{S}_e$ Firewall), $m[P_1 \mid P_2]\ \mathcal{S}_e\ \mathbf{0}$, since $m \in e$ and since we get $fn(P_1 \mid P_2) \cap e = \varnothing$ from $fn(in\ p.P_1 \mid P_2) \cap e = \varnothing$. By ($\mathcal{S}_e$ Par), $m[P_1 \mid P_2]\ \mathcal{S}_e\ \mathbf{0}$ and $P_3\ \mathcal{S}_e\ Q_3$ imply $m[P_1 \mid P_2] \mid P_3\ \mathcal{S}_e\ \mathbf{0} \mid Q_3$. By ($\mathcal{S}_e$ Amb), this and $p \notin e$ imply $P' = p[m[P_1 \mid P_2] \mid P_3]\ \mathcal{S}_e\ Q'$.

In case ($\mathcal{S}_e$ Amb), $m \notin e$, $R_1 = m[R_3]$ and $in\ p.P_1 \mid P_2\ \mathcal{S}_e\ R_3$. By ($\mathcal{S}_e$ Par) and ($\mathcal{S}_e$ Action), there are Q_1 and Q_2 such that $P_1\ \mathcal{S}_e\ Q_1$, $P_2\ \mathcal{S}_e\ Q_2$, $R_3 = in\ p.Q_1 \mid Q_2$ and $p \notin e$. The latter implies that ($\mathcal{S}_e$ Amb), but not ($\mathcal{S}_e$ Firewall), can derive $p[P_3]\ \mathcal{S}_e\ R_2$. Therefore there is Q_3 such that $R_2 = p[Q_3]$ and $P_3\ \mathcal{S}_e\ Q_3$. In summary, we have shown that $Q = n[in\ p.Q_1 \mid Q_2] \mid p[Q_3]$. Let $Q' = p[m[Q_1 \mid Q_2] \mid Q_3]$. By (Red In), $Q \to Q'$. By ($\mathcal{S}_e$ Amb) and ($\mathcal{S}_e$ Par), $P_i\ \mathcal{S}_e\ Q_i$ for $i \in 1..3$ implies that $P'\ \mathcal{S}_e\ Q'$.

(Red I/O) In this case, $P = \langle M \rangle \mid (x).P_3 \to P_3\{x \leftarrow M\} = P'$. Since only ($\mathcal{S}_e$ Par) can derive $P\ \mathcal{S}_e\ Q$, $Q = Q_1 \mid Q_2$ with $\langle M \rangle\ \mathcal{S}_e\ Q_1$ and $(x).P_3\ \mathcal{S}_e\ Q_2$. Since these two relationships may only be derived by ($\mathcal{S}_e$ Output) and ($\mathcal{S}_e$ Input), respectively, it must be that $Q_1 = \langle M \rangle$ with $fn(M) \cap e = \varnothing$, and $Q_2 = (x).Q_3$ with $P_3\ \mathcal{S}_e\ Q_3$. In summary, $Q = \langle M \rangle \mid (x).Q_3$. Let $Q' = Q_3\{x \leftarrow M\}$. By (Red I/O), $Q \to Q'$. By Lemma 25, $P_3\ \mathcal{S}_e\ Q_3$ implies $P_3\{x \leftarrow M\}\ \mathcal{S}_e\ Q_3\{x \leftarrow M\}$, that is, $P'\ \mathcal{S}_e\ Q'$.

(Red Par) In this case, $P = P_1 \mid P_2$, $P_1 \to P_1'$ and $P' = P_1' \mid P_2$. Since only ($\mathcal{S}_e$ Par) can derive $P\ \mathcal{S}_e\ Q$, $Q = Q_1 \mid Q_2$ with $P_1\ \mathcal{S}_e\ Q_1$ and $P_2\ \mathcal{S}_e\ Q_2$. By induction hypothesis, $P_1\ \mathcal{S}_e\ Q_1$ and $P_1 \to P_1'$ imply there is Q_1' with $P_1'\ \mathcal{S}_e\ Q_1'$ and either $Q_1 \to Q_1'$ or $Q_1 \equiv Q_1'$. Let $Q' = Q_1' \mid Q_2$. By ($\mathcal{S}_e$ Par), $P'\ \mathcal{S}_e\ Q'$. If $Q_1 \to Q_1'$, (Red Par) implies $Q \to Q'$. If $Q_1 \equiv Q_1'$, (Struct Par) implies $Q \equiv Q'$.

(Red Amb) In this case, $P = m[P_1]$, $P_1 \to P_1'$ and $P' = m[P_1']$. Either ($\mathcal{S}_e$ Firewall) or ($\mathcal{S}_e$ Amb) can derive $m[P_1]\ \mathcal{S}_e\ Q$. In case ($\mathcal{S}_e$ Amb), the proof is similar to the proof for (Red Par). In case ($\mathcal{S}_e$ Firewall), $m \in e$, $fn(P_1) \cap e = \varnothing$ and $Q = \mathbf{0}$. Let $Q' = Nil$. By Lemma 1,

$P_1 \rightarrow P_1'$ implies $fn(P_1') \subseteq fn(P_1)$, so $fn(P_1') \cap e = \varnothing$. By ($\mathcal{S}_e$ Firewall), $P' = m[P_1']\ \mathcal{S}_e\ \mathbf{0} = Q'$ and $Q = Q'$.

(Red $\equiv$) In this case, $P \equiv P''$, $P'' \rightarrow P'''$ and $P''' \equiv P'$. By Lemma 27, $P\ \mathcal{S}_e\ Q$ and $P \equiv P''$ imply there is Q'' such that $Q \equiv Q''$ and $P''\ \mathcal{S}_e\ Q''$. By induction hypothesis, $P''\ \mathcal{S}_e\ Q''$ and $P'' \rightarrow P'''$ imply there is Q''' such that $P'''\ \mathcal{S}_e\ Q'''$ and either $Q'' \rightarrow Q'''$ or $Q'' \equiv Q'''$. By Lemma 27, $P'''\ \mathcal{S}_e\ Q'''$ and $P''' \equiv P'$ imply there is Q' such that $Q''' \equiv Q'$ and $P'\ \mathcal{S}_e\ Q'$. From $Q \equiv Q''$, either $Q'' \rightarrow Q'''$ or $Q'' \equiv Q'''$, and $Q''' \equiv Q'$, we obtain either $Q \equiv Q'$ by (Struct Trans), or $Q \rightarrow Q'$ by (Red $\equiv$).

We have omitted cases (Red Out), (Red Open) and (Red Res). Cases (Red Out) and (Red Open) have proofs similar to (Red In). Case (Red Res) has a proof similar to (Red Par). □

Lemma 29 *$P\ \mathcal{S}_e\ Q$, $P \downarrow m$ and $m \notin e$ implies $Q \downarrow m$.*

Proof An easy induction on the derivation of $P \downarrow m$, using Lemma 27 in case (Exh $\equiv$). □

Theorem 1 *For any process P and any name $n \notin fn(P)$, $(\nu n)n[P] \simeq \mathbf{0}$.*

Proof Given Lemma 24, it suffices to show that $(\nu n)n[P] \sqsubseteq \mathbf{0}$, that is, for all $\mathcal{C}$ and m that $\mathcal{C}\{(\nu n)n[P]\} \Downarrow m$ implies $\mathcal{C}\{\mathbf{0}\} \Downarrow m$. Since the name n is bound, we may assume that $n \notin fn(\mathcal{C})$. By ($\mathcal{S}_{\{n\}}$ Firewall), $n \notin fn(P)$ implies $n[P]\ \mathcal{S}_{\{n\}}\ \mathbf{0}$. By ($\mathcal{S}_\varnothing$ Res), this implies $(\nu n)n[P]\ \mathcal{S}_\varnothing\ (\nu n)\mathbf{0}$. Therefore we may establish $\mathcal{C}\{(\nu n)n[P]\}\ \mathcal{S}_\varnothing\ \mathcal{C}\{(\nu n)\mathbf{0}\}$, by induction on the size of $\mathcal{C}$. An induction on the derivation of $\mathcal{C}\{(\nu n)n[P]\} \Downarrow m$, using Lemma 28 and Lemma 29 establishes that $\mathcal{C}\{(\nu n)\mathbf{0}\} \Downarrow m$ too. (Struct Zero Res) asserts that $(\nu n)\mathbf{0} \equiv \mathbf{0}$. Since $\equiv$ is a congruence, we get $\mathcal{C}\{(\nu n)\mathbf{0}\} \equiv \mathcal{C}\{\mathbf{0}\}$. By Lemma 17, $\mathcal{C}\{(\nu n)\mathbf{0}\} \Downarrow m$ and $\mathcal{C}\{(\nu n)\mathbf{0}\} \equiv \mathcal{C}\{\mathbf{0}\}$ imply $\mathcal{C}\{\mathbf{0}\} \Downarrow m$. Hence $(\nu n)n[P] \sqsubseteq \mathbf{0}$, as required. □

A corollary is that for any P and Q such that $n \notin fn(P) \cup fn(Q)$, $(\nu n)n[P] \simeq (\nu n)n[Q]$. Since $\simeq$ is preserved by all contexts and implied by structural congruence,

$$(\nu K)c[K[P]] \simeq (\nu K)c[K[Q]]$$

for all P and Q such that $K \notin fn(P) \cup fn(Q)$. This equation is an ambient calculus counterpart of the spi calculus equation,

$$(\nu K)\overline{c}\{M\}_K \simeq (\nu K)\overline{c}\{N\}_K$$

for all capabilities M and N.

4 Conclusion

In this Annex to the main paper, we have begun to develop a theory of mobile computation in the ambient calculus by transferring ideas from the theory of the π-calculus. Our next task is to apply this theory to discharging various equivalences conjectured in the main paper.

References

[1] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. Technical Report 414, University of Cambridge Computer Laboratory, January 1997.

[2] R. De Nicola and M. C. B. Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–133, 1984.

[3] Robin Milner. *Communication and Concurrency.* Prentice-Hall International, 1989.

[4] Robin Milner. The π-calculus. Undergraduate lecture notes, Cambridge University, 1995.

[5] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, parts I and II. *Information and Computation*, 100:1–40 and 41–77, 1992.

[6] David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Theoretical Computer Science: 5th GI-Conference, Karlsruhe*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, March 1981.

[7] Davide Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms.* PhD thesis, University of Edinburgh, 1992. Available as Technical Report CST–99–93, Computer Science Department, University of Edinburgh.