

# A Theory of May Testing for Asynchronous Calculi with Locality and No Name Matching

Prasannaa Thati, Reza Ziaei, and Gul Agha

University of Illinois at Urbana-Champaign  
{thati,ziaei,agha}@uiuc.edu

**Abstract.** We present a theory of may testing for asynchronous calculi with locality and no name matching. Locality is a non-interference property that is common in systems based on object-paradigm. Concurrent languages such as Join and Pict disallow name matching, which is akin to pointer comparison in imperative languages, to provide for an abstract semantics that would allow useful program transformations. May testing is widely acknowledged to be an effective notion for reasoning about safety properties. We provide a trace-based characterization of may testing for versions of asynchronous  $\pi$ -calculus with locality and no name matching, which greatly simplifies establishing equivalences between processes. We also exploit the characterization to provide a complete axiomatization for the finitary fragment of the calculi.

## 1 Introduction

Experience with applying the  $\pi$ -calculus [10] to distributed systems has shown that it is necessary to make additional ontological commitments. Specifically, variants of  $\pi$ -calculus with *asynchrony*, *locality*, and *absence* of name matching have received wide attention recently [2, 4, 7, 9]. Asynchronous message passing is more common in distributed systems than synchronous communication that is assumed primitive in the  $\pi$ -calculus. The discipline of locality, which disallows a process from receiving messages targeted to a name previously received by the process, is typical in systems based on an object paradigm [1]. Name matching is analogous to pointer comparison in imperative languages; disallowing it enables certain performance optimizations. In fact, name comparisons are disallowed by concurrent languages such as Pict [12]. In any case, comparing names is rarely useful in programming; the behavior observed while communicating at a name is all that matters and not the specific name used for communication. A variant of the  $\pi$ -calculus that embodies these three features is  $L\pi$  [9].

We develop a theory of *may* testing for two subcalculi of asynchronous  $\pi$ -calculus [3]: one with only locality, called  $L\pi_{=}$ , and the other with both locality and no name matching, called  $L\pi$ . May testing [11] is a specific instance of the general notion of behavioral equivalence where two processes are said to be equivalent if they have the same success properties in all contexts. A context in

may testing consists of an observing process that runs in parallel and interacts with the process being tested, and success is defined as the observer signaling a special event. The non-determinism in execution may give rise to different runs. A process is said to pass a test proposed by an observer, if there exists a run that leads to a success. By viewing a success as something bad happening, may testing can be used for reasoning about safety properties.

Because the definition of may testing involves a universal quantification over contexts, it is very difficult to prove equivalences directly from the definition. A typical approach to circumvent the problem, is to find an alternate characterization of the equivalence, which involves only the processes being compared. We provide an alternate characterization of may testing in  $L\pi_{=}$  and  $L\pi$ . The characterizations are *trace* based, and directly build on the known characterization for asynchronous  $\pi$ -calculus [3]. In fact, we generalize the usual definition of may testing to a parameterized version, where the parameter determines the set of observers that is used to decide the order.

Our second result is to provide complete axiomatizations of finitary  $L\pi_{=}$  and  $L\pi$  (for processes with no replication). The axiomatizations highlight the differences that arise due to locality and lack of name matching. In addition to laws that are true for asynchronous  $\pi$ -calculus, we obtain laws that are true only in the presence of locality and the absence of name matching. Further, the inference rules for parameterized may testing generalize the ones for the usual may testing. Complete proofs of lemmas and theorems can be found in [13].

## 2 The Calculus $L\pi_{=}$

We assume an infinite set of names  $\mathcal{N}$ , and let  $u, v, w, x, y, z, \dots$  range over  $\mathcal{N}$ . The set of processes, ranged over by  $P, Q, R$ , is defined by the following restricted  $\pi$ -calculus grammar.

$$P := 0 \mid \bar{x}y \mid x(y).P \mid P|P \mid (\nu x)P \mid [x = y]P \mid !x(y).P$$

The name  $x$  is said to be the subject of the output  $\bar{x}y$  and the input  $x(y).P$ . The locality property is enforced by requiring that for every subterm of the form  $x(y).P$ , the bound name  $y$  does not occur as the subject of an input in  $P$ .

For a tuple  $\tilde{x}$ , we denote the set of names occurring in  $\tilde{x}$  by  $\{\tilde{x}\}$ . We write  $\tilde{x}, \tilde{y}$  for the result of appending  $\tilde{y}$  to  $\tilde{x}$ . We let  $\hat{z}$  range over  $\{\emptyset, \{z\}\}$ . The term  $(\nu \hat{z})P$  is  $(\nu z)P$  if  $\hat{z} = \{z\}$ , and  $P$  otherwise. The functions for free names, bound names and names,  $fn(\cdot)$ ,  $bn(\cdot)$  and  $n(\cdot)$ , of a process, and alpha equivalence on processes are defined as usual. We use the usual definition and notational convention for name substitutions, and let  $\sigma$  range over them. Name substitution on processes is defined modulo alpha equivalence with the usual renaming of bound names to avoid captures. We write  $P\sigma$  and  $x\sigma$  to denote the result of applying  $\sigma$  to  $P$  and  $x$  respectively.

We use an early style labeled transition system for the operational semantics (see table 1). The transition system is defined modulo alpha-equivalence on processes in that alpha-equivalent processes have the same transitions. The

$INP: x(y).P \xrightarrow{xz} P\{z/y\}$	$OUT: \overline{x}y \xrightarrow{\overline{x}y} 0$
$PAR: \frac{P_1 \xrightarrow{\alpha} P'_1}{P_1 P_2 \xrightarrow{\alpha} P'_1 P_2} \quad bn(\alpha) \cap fn(P_2) = \emptyset$	$COM: \frac{P_1 \xrightarrow{\overline{x}y} P'_1 \quad P_2 \xrightarrow{xy} P'_2}{P_1 P_2 \xrightarrow{\tau} P'_1 P'_2}$
$RES: \frac{P \xrightarrow{\alpha} P'}{(\nu y)P \xrightarrow{\alpha} (\nu y)P'} \quad y \notin n(\alpha)$	$OPEN: \frac{P \xrightarrow{\overline{x}y} P'}{(\nu y)P \xrightarrow{\overline{x}(y)} P'} \quad x \neq y$
$CLOSE: \frac{P_1 \xrightarrow{\overline{x}(y)} P'_1 \quad P_2 \xrightarrow{xy} P'_2}{P_1 P_2 \xrightarrow{\tau} (\nu y)(P'_1 P'_2)} \quad y \notin fn(P_2)$	
$REP: \frac{P !P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P'}$	$MATCH: \frac{P \xrightarrow{\alpha} P'}{[x=x]P \xrightarrow{\alpha} P'}$

**Table 1.** An early style labeled transition system for  $L\pi_{=}$ .

symmetric versions of  $COM$ ,  $CLOSE$ , and  $PAR$  are not shown. Transition labels, which are also called actions, can be of five forms:  $\tau$  (a silent action),  $\overline{x}y$  (free output of a message with target  $x$  and content  $y$ ),  $\overline{x}(y)$  (bound output),  $xy$  (free input of a message) and  $x(y)$  (bound input). The relation  $\xrightarrow{x(y)}$  is defined by the additional rule  $P \xrightarrow{x(y)} Q$  if  $P \xrightarrow{xy} Q$  and  $y \notin fn(P)$ . We denote the set of all visible (non- $\tau$ ) actions by  $\mathcal{L}$ , let  $\alpha$  range over  $\mathcal{L}$ , and let  $\beta$  range over all the actions. The functions  $fn(\cdot)$ ,  $bn(\cdot)$  and  $n(\cdot)$  are defined on  $\mathcal{L}$  the usual way. As a uniform notation for free and bound actions we adopt the following convention from [3]:  $(\emptyset)\overline{x}y = \overline{x}y$ ,  $(\{y\})\overline{x}y = \overline{x}(y)$ , and similarly for input actions. We define a complementation function on  $\mathcal{L}$  as  $(\hat{y})xy = (\hat{y})\overline{x}y$ ,  $(\hat{y})\overline{x}y = (\hat{y})xy$ .

We let  $s, r, t$  range over  $\mathcal{L}^*$ . The functions  $fn(\cdot)$ ,  $bn(\cdot)$  and  $n(\cdot)$  are extended to  $\mathcal{L}^*$  the obvious way. Complementation on  $\mathcal{L}$  is extended to  $\mathcal{L}^*$  the obvious way. Alpha equivalence over traces is defined as expected, and alpha-equivalent traces are not distinguished. From now on, only *normal* traces  $s \in \mathcal{L}^*$  that satisfy the following hygiene condition are considered: if  $s = s_1.\alpha.s_2$ , then  $(n(s_1) \cup fn(\alpha)) \cap bn(\alpha.s_2) = \emptyset$ . For an action  $\alpha$  and a set of traces  $S$  we define  $\alpha.S = \{\alpha.s \mid s \in S\}$ .

We use  $\Rightarrow$  to denote the reflexive transitive closure of  $\xrightarrow{\tau}$ , and  $\xRightarrow{\beta}$  to denote  $\Rightarrow \xrightarrow{\beta} \Rightarrow$ . For  $s = l.s'$  we use  $P \xrightarrow{s} Q$  to denote  $P \xrightarrow{l} \xrightarrow{s'} Q$ , and similarly  $P \xRightarrow{s} Q$  to denote  $P \xRightarrow{l} \xRightarrow{s'} Q$ . We write  $P \xRightarrow{s}$  if  $P \xRightarrow{s} Q$  for some  $Q$ , and similarly for  $P \xrightarrow{s}$  and  $P \xrightarrow{\tau}$ . We say  $P$  exhibits the trace  $s$  if  $P \xRightarrow{s}$ .

We now instantiate the testing framework [5] on  $L\pi_{=}$ . In fact, by extending the notion of locality, we consider a generalized version of may testing that supports encapsulation. We define a parameterized may preorder  $\overline{\mathcal{E}}_{\rho}$ , where only observers that do not listen on names in  $\rho$  are used to decide the order. The set of names  $\rho$  can be interpreted as being “owned” by the process being tested, in that any testing context is assumed to have only the capability of sending messages to these names. The reader may note that  $\overline{\mathcal{E}}_{\emptyset}$  is the usual may preorder.

(L1)	$s_1.(\hat{y})s_2 \prec s_1.(\hat{y})xy.s_2$	if $(\hat{y})s_2 \neq \perp$
(L2)	$s_1.(\hat{y})(\alpha.xy.s_2) \prec s_1.(\hat{y})xy.\alpha.s_2$	if $(\hat{y})(\alpha.xy.s_2) \neq \perp$
(L3)	$s_1.(\hat{y})s_2 \prec s_1.(\hat{y})xy.\bar{x}y.s_2$	if $(\hat{y})s_2 \neq \perp$
(L4)	$s_1.\bar{x}w.(s_2\{w/y\}) \prec s_1.\bar{x}(y).s_2$	

**Table 2.** A preorder relation on traces.

**Definition 1 (may testing).** *Observers are processes that can emit a special message  $\bar{\mu}\mu$ . We let  $O$  range over the set of observers. We say  $O$  accepts a trace  $s$  if  $O \xrightarrow{\bar{s}.\bar{\mu}\mu}$ . For  $P, O$ , we say  $P$  may  $O$  if  $P|O \xrightarrow{\bar{\mu}\mu}$ . Let  $\text{rcp}(P)$  be the set of all free names in  $P$  that occur as the subject of an input in  $P$ . For any given  $\rho$  we say  $P \sqsubseteq_\rho Q$  if for every  $O$  such that  $\text{rcp}(O) \cap \rho = \emptyset$ ,  $P$  may  $O$  implies  $Q$  may  $O$ . We say  $P \sqsubseteq_\rho Q$  if  $P \sqsubseteq_\rho Q$  and  $Q \sqsubseteq_\rho P$ . Note that  $\sqsubseteq_\rho$  is reflexive and transitive, and  $\simeq_\rho$  is an equivalence relation.  $\square$*

The larger the parameter of a preorder, the smaller the observer set that is used to decide the order. Hence if  $\rho_1 \subset \rho_2$ , we have  $P \sqsubseteq_{\rho_1} Q$  implies  $P \sqsubseteq_{\rho_2} Q$ . However,  $P \sqsubseteq_{\rho_2} Q$  need not imply  $P \sqsubseteq_{\rho_1} Q$ . For instance,  $0 \simeq_{\{x\}} \bar{x}x$ , but only  $0 \sqsubseteq_\emptyset \bar{x}x$  and  $\bar{x}x \not\sqsubseteq_\emptyset 0$ . Similarly,  $\bar{x}x \simeq_{\{x,y\}} \bar{y}y$ , but  $\bar{x}x \not\sqsubseteq_\emptyset \bar{y}y$  and  $\bar{y}y \not\sqsubseteq_\emptyset \bar{x}x$ . However,  $P \sqsubseteq_{\rho_2} Q$  implies  $P \sqsubseteq_{\rho_1} Q$  if  $\text{fn}(P) \cup \text{fn}(Q) \subset \rho_1$ .

**Theorem 1.** *Let  $\rho_1 \subset \rho_2$ . Then  $P \sqsubseteq_{\rho_1} Q$  implies  $P \sqsubseteq_{\rho_2} Q$ . Further, if  $\text{fn}(P) \cup \text{fn}(Q) \subset \rho_1$  then  $P \sqsubseteq_{\rho_2} Q$  implies  $P \sqsubseteq_{\rho_1} Q$ .  $\square$*

We now build on the trace-based characterization of may testing for asynchronous  $\pi$ -calculus presented in [3] to obtain a characterization of may testing in  $\text{L}\pi_-$ . We note that  $\text{L}\pi_-$  is a proper subcalculus of the calculus in [3], i.e. every  $\text{L}\pi_-$  term is also an asynchronous  $\pi$ -calculus term, and the transition systems of the two calculi match on the common terms. Following is a summary of the alternate characterization of may testing in asynchronous  $\pi$ -calculus. To avoid infinitary branching, a transition system with synchronous inputs instead of asynchronous inputs is used. To account for asynchrony, the trace semantics is modified using a trace preorder  $\preceq$  that is defined as the reflexive transitive closure of the laws shown in table 2. The notation  $(\hat{y})\cdot$  is extended to traces as follows.

$$(\hat{y})s = \begin{cases} s & \text{if } \hat{y} = \emptyset \text{ or } y \notin \text{fn}(s) \\ s_1.x(y).s_2 & \text{if } \hat{y} = \{y\} \text{ and there are } s_1, s_2, x \text{ s.t.} \\ & s = s_1.xy.s_2 \text{ and } y \notin \text{fn}(s_1) \cup \{x\} \\ \perp & \text{otherwise} \end{cases}$$

The may preorder  $\sqsubseteq_\emptyset$  in asynchronous  $\pi$ -calculus is then characterized as:  $P \sqsubseteq_\emptyset Q$  if and only if  $P \xRightarrow{s}$  implies  $Q \xRightarrow{r}$  for some  $r \preceq s$ .

The intuition behind the preorder is that if an observer accepts a trace  $s$ , then it also accepts any trace  $r \preceq s$ . Laws *L1-L3* capture asynchrony, and *L4* captures

the inability to mismatch names. Laws  $L1$  and  $L2$  state that an observer cannot force inputs on the process being tested. Since outputs are asynchronous, the actions following an output in a trace exhibited by an observer need not be causally dependent on the output. Hence the observer's outputs can be delayed until a causally dependent action ( $L2$ ), or dropped if there are no such actions ( $L1$ ). Law  $L3$  states that an observer can consume its own outputs unless there are subsequent actions that depend on the output. Law  $L4$  states that without mismatch an observer cannot discriminate bound names from free names, and hence can receive any name in place of a bound name. The intuition behind the trace preorder is formalized in the following lemma. We note that, since  $L\pi_{=}$  is a subcalculus of asynchronous  $\pi$ -calculus, the lemma also holds for  $L\pi_{=}$ .

**Lemma 1.** *If  $P \xRightarrow{\bar{s}}$ , then  $r \preceq s$  implies  $P \xRightarrow{\bar{r}}$ .*  $\square$

May testing in  $L\pi_{=}$  is weaker than in asynchronous  $\pi$ -calculus. This is because the locality property reduces the number of observers that can be used to test processes. For example, the following two processes are distinguishable in asynchronous  $\pi$ -calculus but equivalent in  $L\pi_{=}$ .

$$P = (\nu x)(!x(z).0|\bar{x}x|\bar{y}x) \qquad Q = (\nu x)(!x(z).0|\bar{y}x)$$

The observer  $O = y(z).z(w).\bar{\mu}\mu$  can distinguish  $P$  and  $Q$  in asynchronous  $\pi$ -calculus, but is not a valid  $L\pi_{=}$  term as it violates locality. In fact, no  $L\pi_{=}$  term can distinguish  $P$  and  $Q$ , because the message  $\bar{x}x$  is not observable.

To account for locality we need to consider only the traces that correspond to interaction between  $L\pi_{=}$  processes. Note that the transition system does not by itself account for locality. For instance, in case of the example above, we have  $P \xrightarrow{\bar{y}x} \xrightarrow{\bar{x}x}$  although the message  $\bar{x}x$  is not observable. To counter this deficiency, we define the notion of *well-formed* traces.

**Definition 2.** *For a set of names  $\rho$  and trace  $s$  we define  $rcp(\rho, s)$  inductively as*

$$rcp(\rho, \epsilon) = \rho \qquad rcp(\rho, s.(\hat{y})xy) = rcp(\rho, s) \qquad rcp(\rho, s.(\hat{y})\bar{x}y) = rcp(\rho, s) \cup \hat{y}$$

*We say  $s$  is  $\rho$ -well-formed if  $s = s_1.(\hat{y})\bar{x}y.s_2$  implies  $x \notin rcp(\rho, s_1)$ . We say  $s$  is well-formed if it is  $\emptyset$ -well-formed.*  $\square$

Only  $\rho$ -well-formed traces correspond to an interaction between a process and an  $L\pi_{=}$  observer  $O$  such that  $rcp(O) \cap \rho = \emptyset$ . We are now ready to give the alternate characterization of  $\approx_{\rho}$  in  $L\pi_{=}$ .

**Definition 3.** *We say  $P \ll_{\rho} Q$ , if for every  $\rho$ -well-formed trace  $s$ ,  $P \xRightarrow{s}$  implies there is  $r \preceq s$  such that  $Q \xRightarrow{r}$ .*  $\square$

To prove the characterization, we define an observer  $O(s)$  for a well-formed trace  $s$ , such that  $P \text{ may } O(s)$  implies  $P \xRightarrow{r}$  for some  $r \preceq s$ . This construction is the same as the one used for asynchronous  $\pi$ -calculus [3].

**Definition 4 (canonical observer).** For a trace  $s$ , we define  $O(s)$  as follows:

$$\begin{aligned} O(\epsilon) &\stackrel{\text{def}}{=} \bar{\mu}\mu & O((\hat{y})xy.t) &\stackrel{\text{def}}{=} (\nu\hat{y})(\bar{x}y|O(t)) \\ O(\bar{x}(y).t) &\stackrel{\text{def}}{=} x(y).O(t) & O(\bar{x}y.s) &\stackrel{\text{def}}{=} x(u).[u=y]O(s) \quad u \text{ fresh} \quad \square \end{aligned}$$

Note that well-formedness of  $s$  guarantees that  $O(s)$  is an  $L\pi_{=}$  term. Furthermore, it is easy to show that if  $s$  is  $\rho$ -well-formed, then  $\text{rcp}(O(s)) \cap \rho = \emptyset$ . Since the canonical observer constructions match and  $L\pi_{=}$  is a subcalculus of asynchronous  $\pi$ -calculus, the following lemma proved for asynchronous  $\pi$ -calculus [3], also holds in  $L\pi_{=}$ .

**Lemma 2.** For a well-formed trace  $s$ ,  $O(s) \xrightarrow{\bar{r}.\bar{\mu}\mu}$  implies  $r \preceq s$ .  $\square$

Theorem 2 proves the equivalence of  $\stackrel{\sqsubset}{\sim}_{\rho}$  and  $\ll_{\rho}$  in  $L\pi_{=}$ . The proof is similar to that of Theorem 3 in Section 3.

**Lemma 3.** Let  $\rho$  be a set of names where  $\text{rcp}(O) \cap \rho = \emptyset$ . Then  $P|O \xrightarrow{\bar{\mu}\mu}$  can be unzipped into  $P \xrightarrow{s}$  and  $O \xrightarrow{\bar{s}.\bar{\mu}\mu}$  for some  $s$  that is  $\rho$ -well-formed.  $\square$

**Theorem 2.**  $P \stackrel{\sqsubset}{\sim}_{\rho} Q$  if and only if  $P \ll_{\rho} Q$ .  $\square$

### 3 The Calculus $L\pi$

We now investigate the effect of lack of name matching capability. The rules in table 1 except the *MATCH* rule, constitute the transition system for  $L\pi$ .

The lack of name matching capability further weakens may testing equivalence. For example, the processes  $(\nu u)(\bar{x}u|\bar{x}u)$  and  $(\nu u, v)(\bar{x}u|\bar{x}v)$  are equivalent in  $L\pi$ , but not in  $L\pi_{=}$ . For the alternate characterization of  $P \stackrel{\sqsubset}{\sim}_{\rho} Q$ , it is too stringent to require that for any trace  $s$  that  $P$  exhibits,  $Q$  exhibits a *single* trace  $r$  such that any observer accepting  $s$  also accepts  $r$ . In fact, there exist  $L\pi$  processes  $P$  and  $Q$  such that  $P \stackrel{\sqsubset}{\sim}_{\rho} Q$ , and if  $P$  exhibits  $s$ , then  $Q$  exhibits *different* traces to satisfy different observers that accept  $s$ . For instance, let  $P = \bar{x}u_1|\bar{y}u_1|u_1(w).\bar{w}w$  which can exhibit  $s = \bar{x}u_1.\bar{y}u_1.u_1(w).\bar{w}w$ . The following  $L\pi$  observers accept  $s$ .

$$\begin{aligned} O_1 &= (\nu w)(x(u).y(v).\bar{w}w|w(v).\bar{\mu}\mu) \\ O_2 &= (\nu w)(x(u).y(v).\bar{v}w|w(v).\bar{\mu}\mu) \\ O_3 &= (\nu w)(x(u).y(v).\bar{u}_1w|w(v).\bar{\mu}\mu) \\ O_4 &= (\nu w)(x(u).y(v).(\bar{v}v|\bar{u}u) \mid u_1(z).u_1(z).\bar{u}_1w \mid w(v).\bar{\mu}\mu) \end{aligned}$$

Now, the process  $Q = (\nu v)(v(z).v(z')).(\bar{x}z|\bar{y}z')|\bar{v}u_1|\bar{v}u_2|!u_2(z).\bar{u}_1z \mid u_1(w).\bar{w}w$  can satisfy

$$\begin{aligned} O_1 \text{ with } r_1 &= \bar{x}u_1.\bar{y}u_2.u_1(w).\bar{w}w & O_2 \text{ with } r_2 &= \bar{x}u_2.\bar{y}u_1.u_1(w).\bar{w}w \\ O_3 \text{ with } r_1 \text{ or } r_2, \text{ and} & & O_4 \text{ with } r_4 &= \bar{x}u_1.\bar{y}u_2.u_2u_2.\bar{u}_1u_2.u_1(w).\bar{w}w \end{aligned}$$

but cannot exhibit a single trace that can satisfy all four observers. In fact, it is the case that  $P \sqsubseteq_{\emptyset} Q$ . Intuitively, although unlike  $P$ ,  $Q$  always exports two different names at  $x$  and  $y$ , for each possible dataflow pattern of the received names inside an observer that  $P$  satisfies,  $Q$  exhibits a corresponding trace that can lead the observer to a success.

For the alternate characterization, we define *templates* which are a special kind of traces that can be used to represent dataflows in an observer. A template is a trace in which all outputs are bound. The binding relation between arguments of outputs and their subsequent free occurrences, represents the relevant dependencies between the output argument that is received by an observer and its subsequent use in the observer's computation. For a trace  $s$  and set of names  $\rho$ , we define a set  $T(s, \rho)$  that has a template for each possible dataflow in a computation  $O \xRightarrow{\bar{s}, \bar{\mu}} \mu$  with  $\text{rcp}(O) \cap \rho = \emptyset$ . Further, if  $t$  represents the dataflow in a computation  $O \xRightarrow{\bar{s}, \bar{\mu}} \mu$ , then it will be the case that  $O \xRightarrow{\bar{t}, \bar{\mu}} \mu$ . Thus, if an observer accepts a trace  $s$ , then it also accepts a template in  $T(s, \rho)$ . This template construction essentially captures the effect of lack of match operator. We will show that  $P \sqsubseteq_{\rho} Q$  if and only if for every  $\rho$ -well-formed trace  $s$  that  $P$  exhibits and for each  $t \in T(s, \rho)$ ,  $Q$  exhibits some  $r \preceq t$ .

Following is an informal description of how the set  $T(s, \rho)$  can be obtained. Due to the lack of name matching capability, an observer cannot fully discriminate between free inputs. Therefore, a process can satisfy an observer  $O$  that exhibits  $O \xRightarrow{\bar{s}, \bar{\mu}} \mu$ , by replacing free input arguments in  $\bar{s}$  with any name as long as it is able to account for changes to the subsequent computation steps that depend on the replaced name. Specifically, suppose  $O \xRightarrow{\bar{s}, \bar{\mu}} \mu$  abbreviates the following computation:

$$O \xRightarrow{\bar{s}_1} O_0 \xrightarrow{xy} O_1 \xrightarrow{\beta_1} O_2 \xrightarrow{\beta_2} \dots O_n \xrightarrow{\beta_n} \bar{\mu} \mu$$

Because of the locality property, the name  $y$  received in the input may be used only in output terms of  $O_1$ . We call such occurrences of  $y$  as *dependent* on the input. During subsequent computation, these output terms may appear either as an output action or are consumed internally. In the latter case,  $y$  may be the target of the internal communication, or the argument which in turn may generate further output terms with dependent occurrences of  $y$ . Therefore,  $O$  can do the following computation when  $y$  in the input is replaced with an arbitrary name  $w$ :

$$O \xRightarrow{\bar{s}_1} O_0 \xrightarrow{(\hat{w})xw} O_1 \xrightarrow{\gamma_1} O_2 \xrightarrow{\gamma_2} \dots O_n \xrightarrow{\gamma_n} \bar{\mu} \mu$$

where  $\gamma_i$  is obtained from  $\beta_i$  as follows. If  $\beta_i$  is an output action, then  $\gamma_i$  is obtained from  $\beta_i$  by substituting dependent occurrences of  $y$  with  $w$ . If  $\beta_i$  is an internal delivery of a message  $\bar{y}z$  with target  $y$  being a dependent occurrence, there are two possibilities. If  $z$  is a private name, then  $\gamma_i = \bar{w}(z).yz$  and the subsequent bound output  $\beta_j$  ( $j > i$ ) that exports  $z$  for the first time (if any), is changed to a free output. If  $z$  is not a private name, then  $\gamma_i = \bar{w}z'.yz'$ , where  $z'$

is  $w$  when  $z$  is a dependent occurrence of  $y$  and  $z$  otherwise. For all other cases,  $\gamma_i = \beta_i$ . Note that, if  $w$  is fresh, the input of  $w$  could be a bound input.

Clearly, any computation obtained by repeated application of the above construction can be performed by  $O$ . In particular, if we always replace free inputs with bound inputs, we will eventually obtain a computation in which all inputs are bound and the construction can not be applied any further. Let  $O \xRightarrow{\bar{t}, \bar{\mu}\mu}$  abbreviate a computation thus obtained. The trace  $t$  is a template that explicitly represents all dependencies between received names (bound input arguments) and subsequent computation steps (subsequent free occurrences of the argument). The set  $T(s, \rho)$  consists of all the templates that can be obtained by this construction starting from arbitrary computations of the form  $O \xRightarrow{\bar{s}, \bar{\mu}\mu}$  with  $rcp(O) \cap \rho = \emptyset$ .

We now formalize the ideas presented above, leading to a direct inductive definition of  $T(s, \rho)$ . Let  $O \xRightarrow{\bar{s}_1} xy \rightarrow O_1 \xRightarrow{\bar{s}_2} \bar{\mu}\mu$ . We first consider the simple case where  $y \notin rcp(O_1)$ . Due to locality, in the computation following input  $xy$ , there cannot be an internal message delivery with  $y$  as the target. Therefore, the following computation is possible.  $O \xRightarrow{\bar{s}_1} (\dot{w})xw \rightarrow O'_1 \xRightarrow{\bar{s}'_2} \bar{\mu}\mu$  where  $\bar{s}'_2$  is obtained from  $\bar{s}_2$  by renaming dependent occurrences of  $y$  in output actions to  $w$ . Specifically, it does not involve exposing internal actions that use dependent occurrences of  $y$ . When the computation steps above are not known, all we can say about  $\bar{s}'_2$  is that it is obtained from  $\bar{s}_2$  by renaming some occurrences of  $y$ . Similarly,  $O'_1$  is obtained from  $O_1$  by renaming some occurrences of  $y$  in output terms. These relations are formalized in Definition 5 and Lemma 4.

**Definition 5 (random output substitution).** For  $\sigma = \{\tilde{u}/\tilde{v}\}$  we define random output substitution (from now on just random substitution) on process  $P$ , denoted by  $P[\sigma]$ , modulo alpha equivalence as follows. We assume  $bn(P) \cap \{\tilde{v}\} = fn(P)\sigma \cap bn(P) = \emptyset$ . For a name  $x$  we define  $x[\sigma] = \{x, x\sigma\}$ .

$$\begin{aligned} 0[\sigma] &= \{0\} & (x(y).P)[\sigma] &= \{x(y).P' \mid P' \in P[\sigma]\} \\ (\bar{x}y)[\sigma] &= \{\bar{x}'y' \mid x' \in x[\sigma], y' \in y[\sigma]\} & (P|Q)[\sigma] &= \{P'|Q' \mid P' \in P[\sigma], Q' \in Q[\sigma]\} \\ ((\nu x)P)[\sigma] &= \{(\nu x)P' \mid P' \in P[\sigma]\} & (!x(y).P)[\sigma] &= \{!x(y).P' \mid P' \in P[\sigma]\} \end{aligned}$$

Random substitution on traces is defined modulo equivalence as follows. We assume  $bn(s) \cap \{\tilde{v}\} = fn(s)\sigma \cap bn(s) = \emptyset$ .

$$\begin{aligned} \epsilon[\sigma] &= \{\epsilon\} & ((\hat{y})\bar{x}y.s)[\sigma] &= \{(\hat{y})\bar{x}y.s' \mid s' \in s[\sigma]\} \\ (x(y).s)[\sigma] &= \{x'(y).s' \mid x' \in x[\sigma], s' \in s[\sigma]\} \\ (xy.s)[\sigma] &= \{x'y'.s' \mid x' \in x[\sigma], y' \in y[\sigma], s' \in s[\sigma]\} \end{aligned}$$

We will use  $[\tilde{u}/\tilde{v}]$  as a short form for  $\{\{\tilde{u}/\tilde{v}\}\}$ . □

**Lemma 4.** If  $P \xRightarrow{\bar{s}}$ ,  $P' \in P[w/y]$ , and  $y \notin rcp(P)$ , then  $P' \xRightarrow{\bar{s}'}$  for some  $s' \in s[w/y]$ . □

Now, suppose  $y \in rcp(O_1)$ . Then, in the computation  $O \xRightarrow{\bar{s}_1} xy \rightarrow O_1 \xRightarrow{\bar{s}_2} \bar{\mu}\mu$ , certain internal transitions may involve a message with a dependent occurrence of



$y$  as the target. Then, the following computation which exposes such transitions is also possible  $O \xrightarrow{\overline{s_1}} (\hat{w})xw \xrightarrow{\overline{s_2}} O'_1 \xrightarrow{\overline{\mu}} \overline{\mu}\mu$  where  $\overline{s'_2}$  is obtained from  $\overline{s_2}$  by not only renaming all dependent occurrences of  $y$  in output transitions to  $w$ , but also exposing each internal message delivery with a dependent occurrence of  $y$  as the message target. If the computation steps are not known, we can only say  $\overline{s'_2}$  is obtained from some  $r \in s_2[w/y]$  by exposing arbitrary number of internal transitions at any point in  $\overline{r}$ . The relation between  $s_2$  and  $s'_2$  is formalized in Definition 6 and Lemma 5. To account for the situation where an exposed pair of actions  $(\hat{z})\overline{w}z.yz$  export a private name  $z$ , we need the following function on traces.

$$[\hat{y}]s = \begin{cases} s & \text{if } \hat{y} = \emptyset \text{ or } y \notin n(s) \\ s_1.xy.s_2 & \text{if } \hat{y} = \{y\} \text{ and there are } s_1, s_2, x \text{ s.t.} \\ & s = s_1.x(y).s_2 \text{ and } y \notin n(s_1) \cup \{x\} \\ \perp & \text{otherwise} \end{cases}$$

**Definition 6.** For a trace  $s$  and a pair of names  $w, y$ , the set  $F(s, w, y)$  is the smallest set closed under the following rules:

1.  $\epsilon \in F(\epsilon, w, y)$
2.  $(\hat{v})uv.s' \in F((\hat{v})uv.s, w, y)$  if  $s' \in F(s, w, y)$
3.  $(\hat{v})\overline{w}v.s' \in F((\hat{v})\overline{w}v.s, w, y)$  if  $s' \in F(s, w, y)$
4.  $(\hat{z})wz.\overline{y}z.[\hat{z}]s' \in F(s, w, y)$  if  $s' \in F(s, w, y)$  and  $[\hat{z}]s' \neq \perp$

Note that  $s \in F(s, w, y)$ . For a set of traces  $S$ , we define  $F(S, w, y) = \cup_{s \in S} F(s, w, y)$ .  $\square$

**Lemma 5.** If  $P \xrightarrow{\overline{s}}$  and  $P' \in P[w/y]$ , then  $P' \xrightarrow{\overline{s'}}$  for some  $s' \in F(s[w/y], w, y)$ .  $\square$

For a trace  $s$  and a set of names  $\rho$ , we say  $s$  is  $\rho$ -normal, if  $s$  is normal and  $\rho \cap bn(s) = \emptyset$ . Now, let  $O$  be an arbitrary observer such that  $rcp(O) \cap \rho = \emptyset$ . Suppose

$$O \xrightarrow{\overline{s_1}} xy \xrightarrow{\overline{s_2}} O_1 \xrightarrow{\overline{\mu}} \overline{\mu}\mu$$

where  $\overline{s_1}.xy.\overline{s_2}$  is  $\rho$ -normal. If  $y \in \rho$  or  $y$  is the argument of a bound input in  $\overline{s_1}$ , then by locality  $y \notin rcp(O_1)$ . Otherwise, since  $O$  is arbitrary, it is possible that  $y \in rcp(O_1)$ . From this observation, we have that for an arbitrary observer  $O$  such that  $rcp(O) \cap \rho = \emptyset$ , if  $O$  accepts the  $\rho$ -normal trace  $s_1.\overline{x}y.s_2$ , then  $O$  also accepts  $s_1.(\hat{w})\overline{x}w.s'_2$  where  $w$  is an arbitrary name and  $s'_2 \in s_2[w/y]$  if  $y \in \rho$  or  $y$  is the argument of a bound output in  $s_1$ , and  $s'_2 \in F(s_2[w/y], w, y)$  otherwise.  $T(s, \rho)$  is precisely the set of all traces with no free outputs, that can be obtained by repeated application of this reasoning.  $T(s, \rho)$  is formally defined in Definition 7.

**Definition 7.** For a trace  $s$  and a set of names  $\rho$ , the set of templates  $T(s, \rho)$  is defined modulo alpha equivalence as follows. We assume that  $s$  is  $\rho$ -normal.

1.  $\epsilon \in T(\epsilon, \rho)$ .

2.  $(\hat{y})xy.s' \in T((\hat{y})xy.s, \rho)$  if  $s' \in T(s, \rho)$
3.  $\bar{x}(y).s' \in T(\bar{x}(y).s, \rho)$  if  $s' \in T(s, \rho \cup \{y\})$
4.  $\bar{x}(w).s' \in T(\bar{x}y.s, \rho)$  if  $w$  fresh,  $s' \in T(s'', \rho \cup \{w\})$ , and
 
$$s'' \in \begin{cases} s[w/y] & \text{if } y \in \rho \\ F(s[w/y], w, y) & \text{if } y \notin \rho \end{cases}$$

The reader may check that if  $t \in T(s, \rho)$ , then  $s \preceq t$  using only L3 and L4.  $\square$

**Lemma 6.** If  $P \xrightarrow{\bar{s}}$  and  $\rho \cap \text{rcp}(P) = \emptyset$ , then there is  $t \in T(s, \rho)$  such that  $P \xrightarrow{\bar{t}}$ .  $\square$

Lemma 7 states that template construction in Definition 7 preserves  $\rho$ -well-formedness.

**Lemma 7.** If  $s$  is  $\rho$ -well-formed then every  $t \in T(s, \rho)$  is  $\rho$ -well-formed.  $\square$

We are now ready to give the alternate characterization of  $\bar{\preceq}_\rho$  in  $L\pi$ .

**Definition 8.** We say  $P \ll_\rho Q$  if for every  $\rho$ -well-formed trace  $s$ ,  $P \xrightarrow{s}$  implies for each  $t \in T(s, \rho)$  there is  $r \preceq t$  such that  $Q \xrightarrow{r}$ .  $\square$

For  $t \in T(s, \rho)$ , where  $s$  is a  $\rho$ -well-formed trace, let  $O(t)$  be the canonical observer as defined in Definition 4. By Lemma 7, since  $s$  is  $\rho$ -well-formed  $t$  is also  $\rho$ -well-formed. Hence  $O(t)$  satisfies the locality property, and  $\text{rcp}(O(t)) \cap \rho = \emptyset$ . Further, since  $t$  is a template, the case  $t = \bar{x}y.t'$  does not arise in the construction of the observer. Hence  $O(t)$  is an  $L\pi$  term. Since  $L\pi$  is a subcalculus of asynchronous  $\pi$ -calculus, Lemma 1 holds for  $L\pi$ . Further, since the canonical observer construction is unchanged, the following lemma (which is a weaker version of Lemma 2) holds for  $L\pi$ .

**Lemma 8.** For  $t \in T(s, \rho)$ , where  $s$  is a  $\rho$ -well-formed trace,  $O(t) \xrightarrow{\bar{r}.\bar{\mu}\mu}$  implies  $r \preceq t$ .  $\square$

Lemma 3 holds for  $L\pi$  with formally the same proof. Now, we are ready to prove that  $\ll_\rho$  is an alternate characterization of  $\bar{\preceq}_\rho$ .

**Theorem 3.**  $P \bar{\preceq}_\rho Q$  if and only if  $P \ll_\rho Q$ .

*Proof.* (if) Let  $P \ll_\rho Q$  and  $P \underline{\text{may}} O$  for an observer  $O$  such that  $\text{rcp}(O) \cap \rho = \emptyset$ . From  $P \underline{\text{may}} O$  we have  $P|O \xrightarrow{\bar{\mu}\mu}$ . By Lemma 3, this computation can be unzipped into  $P \xrightarrow{s}$  and  $O \xrightarrow{\bar{s}.\bar{\mu}\mu}$  for some  $\rho$ -well-formed trace  $s$ . From Lemma 1 and 6 we deduce there is a  $t' \in T(s.\mu\mu, \rho)$  such that  $r' \preceq t'$  implies  $O \xrightarrow{\bar{r}'}$ . It is easy to show that  $t' \in T(s.\mu\mu, \rho)$  implies  $t' = t.\mu\mu$  for some  $t \in T(s, \rho)$ . From  $P \ll_\rho Q$ , there is a trace  $r \preceq t$  such that  $Q \xrightarrow{r}$ . Moreover,  $r \preceq t$  implies  $r.\mu\mu \preceq t.\mu\mu = t'$ . Therefore,  $O \xrightarrow{\bar{r}.\bar{\mu}\mu}$ . We can zip this with  $Q \xrightarrow{r}$  to obtain  $Q|O \xrightarrow{\bar{\mu}\mu}$ , which means  $Q \underline{\text{may}} O$ .

(**only if**): Let  $P \sqsubseteq_\rho Q$  and  $P \xRightarrow{s}$  where  $s$  is  $\rho$ -well-formed. We have to show for every  $t \in T(s, \rho)$  there is a trace  $r \preceq t$  such that  $Q \xRightarrow{r}$ . It is easy to show that if  $t \in T(s, \rho)$ , then  $O(t) \xRightarrow{\bar{s}.\bar{\mu}\mu}$ . This can be zipped with  $P \xRightarrow{s}$  to get  $P|O(t) \xRightarrow{\bar{\mu}\mu}$ , that is  $P \underline{\text{may}} O(t)$ . From  $P \sqsubseteq_\rho Q$ , we have  $Q \underline{\text{may}} O(t)$  and therefore  $Q|O(t) \xRightarrow{\bar{\mu}\mu}$ . This can be unzipped into  $Q \xRightarrow{r}$  and  $O(t) \xRightarrow{\bar{r}.\bar{\mu}\mu}$ . From Lemma 8, it follows that  $r \preceq t$ .  $\square$

For finitary processes we can obtain a simpler characterization based on a modified version of Definition 7 as given below.

**Definition 9.** For a trace  $s$  and a set of names  $\rho$ , the set  $T_f(s, \rho)$  is defined inductively using the first three rules of Definition 7 and the following two.

- 4  $\bar{x}(w).s' \in T_f(\bar{x}y.s, \rho)$  if  $y \in \rho, w$  fresh,  $s' \in T_f(s'', \rho \cup \{w\})$ , and  $s'' \in s[w/y]$
- 5  $\bar{x}y.s' \in T_f(\bar{x}y.s, \rho)$  if  $y \notin \rho$ , and,  $s' \in T_f(s, \rho)$   $\square$

The main difference from Definition 7 is that output arguments  $y$  that are not in  $\rho$  are not converted to bound arguments. According to rule 4 of Definition 7, such conversions introduce arbitrary number of pairs of input/output actions. But, since the length of traces that a finite process can exhibit is bounded, the only way the process can exhibit a trace  $r \preceq t$  for each of the resulting templates, is by emitting the same name  $y$ , so that  $L4$  and  $L3$  can be applied to annihilate some of these input/output pairs. The following lemma helps formalize this observation.

**Lemma 9.** For a trace  $s$ , a set of names  $\rho$ , and a prefixed closed set  $R$  of traces with bounded length, if for every  $t \in T(s, \rho)$  there exists  $r \in R$  such that  $r \preceq t$ , then for every  $t_f \in T_f(s, \rho)$  there exists  $r \in R$  such that  $r \preceq t_f$ .  $\square$

Using this lemma, we can show that for finitary processes we can use  $T_f(s, \rho)$  in Definition 8 instead of  $T(s, \rho)$ . The resulting characterization is equivalent to the earlier one for the following reason. Suppose  $P \xRightarrow{s}$  implies, for every  $t \in T(s, \rho)$ , there exists  $r \preceq t$  such that  $Q \xRightarrow{r}$ . Then, let  $R$  be the set of all traces that  $Q$  exhibits. Note that  $R$  is prefix closed. Further, since  $Q$  is finite, there is a bound on the length of traces in  $R$ . By Lemma 9, for every  $t_f \in T_f(s, \rho)$ , there exists  $r \preceq t_f$  such that  $Q \xRightarrow{r}$ . Conversely, suppose  $P \xRightarrow{s}$  implies that for every  $t \in T_f(s, \rho)$  there exists  $r \preceq t$  such that  $Q \xRightarrow{r}$ . It is easy to verify that for every  $t \in T(s, \rho)$  there exists a  $t_f \in T_f(s, \rho)$  such that  $t_f \preceq t$ , where the relation can be derived using only  $L3$  and  $L4$ . From transitivity of  $\preceq$ , it follows that  $P \xRightarrow{s}$  implies for every  $t \in T(s, \rho)$  there exists  $r \preceq t$  such that  $Q \xRightarrow{r}$ .

## 4 An Axiomatization of Finitary $L\pi_{=}$ and $L\pi$

We first give a sound and complete proof system for  $\sqsubseteq_\rho$  for the finitary fragment of  $L\pi$ , i.e. for  $L\pi$  processes that do not use replication. A simple adaptation of

the proof system gives us one for finitary  $L\pi_-$ . The proof system consists of the laws given in table 3 and the rules for reflexivity and transitivity. For a finite index set  $I$ , we use the macro  $\sum_{i \in I} P_i$  to denote,  $(\nu u)((|_{i \in I} u(u).P_i)|\bar{u}u)$  for  $u$  fresh if  $I \neq \emptyset$ , and 0 otherwise. For an index set that is a singleton, we omit  $I$  and simply write  $\sum P$  instead of  $\sum_{i \in I} P$ . We let the variable  $G$  range over processes of form  $\sum_{i \in I} P_i$ . We write  $\sum_{i \in I} P_i + \sum_{j \in J} P_j$  to denote  $\sum_{k \in I \uplus J} P_k$ . We write  $\sqsubseteq$  as a shorthand for  $\sqsubseteq_\emptyset$ , and  $=$  for  $=_\emptyset$ . Random input substitution on processes  $P[w/y]_i$  is defined similar to random output substitution (Definition 5), except that only the occurrences of  $y$  at the subject of input prefixes in  $P$  are randomly substituted with  $w$ .

---

<i>I1</i>	if $P \sqsubseteq_\rho Q$ and $rcp(R) \cap \rho = \emptyset$ , then $(\nu x)P \sqsubseteq_{\rho - \{x\}} (\nu x)Q$ , $P R \sqsubseteq_\rho Q R$ .	
<i>I2</i>	if for each $z \in fn(P, Q)$ $P\{z/y\} \sqsubseteq_\rho Q\{z/y\}$ then $x(y).P \sqsubseteq_\rho x(y).Q$	
<i>I3</i>	if for each $i \in I$ $P_i \sqsubseteq_\rho \sum_{j \in J} Q_{ij}$ then $\sum_{i \in I} P_i \sqsubseteq_\rho \sum_{i \in I, j \in J} Q_{ij}$	
<i>I4</i>	if $\rho_1 \subset \rho_2$ and $P \sqsubseteq_{\rho_1} Q$ then $P \sqsubseteq_{\rho_2} Q$ .	
<i>A1</i>	$G + G = G$	<i>A3</i> $P 0 = P$
<i>A2</i>	$G \sqsubseteq G + G'$	<i>A4</i> $P Q = Q P$
		<i>A5</i> $(P Q) R = P (Q R)$
<i>A6</i>	Let $G = \sum_{i \in I} \alpha_i.P_i$ and $G' = \sum_{j \in J} \alpha'_j.P'_j$ where each $\alpha_i$ (resp. $\alpha'_j$ ) does not bind free names of $G'$ (resp. $G$ ). Then $G G' = \sum_{i \in I} \alpha_i.(P_i G') + \sum_{j \in J} \alpha'_j.(G P'_j)$	
<i>A7</i>	$(\nu x)(\sum_{i \in I} P_i) = \sum_{i \in I} (\nu x)P_i$	
<i>A8</i>	$(\nu x)(P Q) = P (\nu x)Q$	$x \notin n(P)$
<i>A9</i>	$(\nu x)(\bar{x}y \alpha.P) = \alpha.(\nu x)(\bar{x}y P)$	$x \notin n(\alpha)$
<i>A10</i>	$(\nu x)(\bar{x}y x(z).P) = (\nu x)(P\{y/z\})$	
<i>A11</i>	$(\nu x)(y(z).P) = \begin{cases} y(z).(\nu x)P & \text{if } x \neq y, x \neq z \\ 0 & \text{if } x = y \end{cases}$	
<i>A12</i>	$\bar{x}y \sum_{i \in I} P_i = \sum_{i \in I} (\bar{x}y P_i)$	$I \neq \emptyset$
<i>A13</i>	$\alpha.\sum_{i \in I} P_i = \sum_{i \in I} \alpha.P_i$	$I \neq \emptyset$
<i>A14</i>	$P = \sum P$	
<i>A15</i>	$x(y).(\bar{u}v P) \sqsubseteq \bar{u}v x(y).P$	$y \neq u, y \neq v$
<i>A16</i>	$P\{y/z\} \sqsubseteq \bar{x}y x(z).P$	
<i>A17</i>	$x(u).y(v).P \sqsubseteq y(v).x(u).P$	$u \neq y, u \neq v$
<i>A18</i>	$x(y).(\bar{x}y P) \sqsubseteq P$	$y \notin n(P)$
<i>A19</i>	$(\nu x)P \sqsubseteq P\{y/x\}$	
<i>A20</i>	If $x \in \rho$ , $w \neq x$ and $w \neq y$ , then $\bar{x}y z(w).P \sqsubseteq_\rho \sum z(w).(\bar{x}y P) + \sum z(w).P + \sum Q$ , where $Q = \begin{cases} P\{y/w\} & \text{if } x = z \\ 0 & \text{otherwise} \end{cases}$	
<i>A21</i>	$\bar{x}y P \sqsubseteq_\rho (\nu w)(\bar{x}w \sum_{P' \in P[w/y]_i} P')$	$w$ fresh, $y \in \rho$ .

---

**Table 3.** Inference rules and axioms for  $L\pi_-$ .

While axioms  $A1$  to  $A19$  all hold in asynchronous  $\pi$ -calculus [3], axioms  $A20$  and  $A21$  are unique to  $L\pi$ .  $A20$  captures the fact that a message targeted to a name that an environment is prohibited from listening to, cannot escape to the environment. The axiom states that there are only two ways such a message can be handled in the next transition step: it can be consumed internally or delayed for later. The axiom also accounts for delaying the message forever by including dropping of the message as one of the possibilities. As an application of this axiom, if  $x \in \rho$ , we can prove  $\bar{x}y \sqsubseteq_\rho 0$  as follows. For  $w$  fresh,

$$\begin{aligned}
\bar{x}y &\sqsubseteq_\rho \bar{x}y | (\nu w)(w(w).0) && (A3, A11, I1) \\
&\sqsubseteq_\rho (\nu w)(\bar{x}y | w(w).0) && (A8) \\
&\sqsubseteq_\rho (\nu w)(\sum w(w).0 + \sum w(w).\bar{x}y + \sum 0) && (A20, I1) \\
&\sqsubseteq_\rho \sum (\nu w)(w(w).0) + \sum (\nu w)w(w).\bar{x}y + \sum (\nu w)0 && (A7) \\
&\sqsubseteq_\rho 0 && (A1, A11, A14, I3)
\end{aligned}$$

Axiom  $A21$  captures the effect of lack of match operator. It is directly motivated from rule 4 of Definition 9 for template construction.

The inference rules extend the rules for asynchronous  $\pi$ -calculus to handle parameterization of the may preorder. In fact, the rules for asynchronous  $\pi$ -calculus presented in [3] can be obtained by setting  $\rho = \emptyset$  in  $I1$ ,  $I2$  and  $I3$ .  $I4$  is a new rule that is motivated by Theorem 1. We make a few remarks about  $I1$  which is significantly different from its analogue for asynchronous  $\pi$ -calculus. First, using  $\bar{x}y \sqsubseteq_{\{x\}} 0$  (proved above) and  $I1$ , we get  $(\nu x)\bar{x}y \sqsubseteq (\nu x)0$ , and by axiom  $A19$  we have  $(\nu x)0 \sqsubseteq 0$ . Therefore,  $(\nu x)\bar{x}y \sqsubseteq 0$ . Note the use of the ability to contract the parameter  $\rho$  of the may preorder after applying a restriction. Second, the following example illustrates the necessity of the side condition  $rcp(R) \cap \rho = \emptyset$  for composition:  $\bar{x}y \sqsubseteq_{\{x\}} 0$  but not  $\bar{x}y | x(y).\bar{y}y \sqsubseteq_{\{x\}} x(y).\bar{y}y$ , for the LHS can satisfy the observer  $y(u).\bar{\mu}\mu$  and the RHS can not.

The soundness of rules  $I1$ - $I4$  can be easily proved directly from Definition 1. We only show the argument for  $I1$ , which is given in Lemma 10. Soundness of axioms  $A1$ - $A21$  is easy to check. For  $A1$ - $A19$ , whenever  $P \sqsubseteq Q$ , we have  $P \xRightarrow{s} \cdot$ , implies  $Q \xRightarrow{r} \cdot$  such that  $r \preceq s$ . For  $A20$ , both LHS and RHS exhibit the same  $\rho$ -well-formed traces. Proof of soundness of axiom  $A21$  is more involved, and is established in Lemma 10. The reader can verify that  $A20$  and  $A21$  would also be sound as equalities. For instance, the converse of  $A21$  can be shown using  $A19$ ,  $A1$ , and  $I1$ .

**Lemma 10.**

1. If  $P \sqsubseteq_\rho Q$  and  $rcp(R) \cap \rho = \emptyset$ , then  $(\nu x)P \sqsubseteq_{\rho - \{x\}} (\nu x)Q$ ,  $P | R \sqsubseteq_\rho Q | R$ .
2. For  $y \in \rho$  and  $w$  fresh,  $\bar{x}y | P \sqsubseteq_\rho (\nu w)(\bar{x}w | \sum_{P' \in P[w/y]_i} P')$ .  $\square$

We prove that the laws presented constitute a complete proof system for finite processes, i.e. for finite processes  $P, Q$ ,  $P \sqsubseteq_\rho Q$  if  $P \sqsubseteq_\rho Q$ . Inspired by the alternate characterization, the proof relies on existence of canonical forms for processes.

**Definition 10.** If  $s$  is a template, then we call  $\bar{s}$  a cotemplate. Thus, a cotemplate is a trace with no free inputs. If  $s$  is well-formed, we say  $\bar{s}$  is cowell-formed.

1. For a cowell-formed cotemplate  $s$ , the process  $e(s)$  is defined inductively as follows.

$$\begin{aligned} e(\epsilon) &\stackrel{\text{def}}{=} 0 & e(\bar{x}y.s') &\stackrel{\text{def}}{=} \bar{x}y|e(s') \\ e(\bar{x}(y).s') &\stackrel{\text{def}}{=} (\nu y)(\bar{x}y|e(s')) & e(x(y).s') &\stackrel{\text{def}}{=} x(y).e(s') \end{aligned}$$

Note that cowell-formedness of  $s$  implies that  $e(s)$  is an  $L\pi$  term. From now on we follow the convention that whenever we write  $e(s)$  it is implicit that  $s$  is a cowell-formed cotemplate.

2. The process  $\sum_{s \in S} e(s)$ , for a set of traces  $S$ , is said to be in canonical form.  $\square$

The proof of completeness relies on the following four lemmas. The first lemma states that every process has an equivalent canonical form.

**Lemma 11.** For a process  $P$  there is a canonical form  $C$  such that  $P = C$ .  $\square$

**Lemma 12.** (1) If  $e(s) \xrightarrow{r}$ , then  $e(r) \sqsubseteq e(s)$ . (2) If  $s \preceq r$  then  $e(r) \sqsubseteq e(s)$ .  $\square$

The proofs of the two lemmas above are formally the same as the proofs of the corresponding lemmas for asynchronous  $\pi$ -calculus [3]. This is because, the proofs of  $P = C$  and  $e(r) \sqsubseteq e(s)$  constructed using the proof system of [3], can be transformed into proofs in our proof system. This claim is justified by the following three observations. First, every  $L\pi$  term is also an asynchronous  $\pi$ -calculus term. Second, starting from  $L\pi$  terms, every term that appears in the proofs of [3] is also an  $L\pi$  term. (Note that any summation that appears is finite and can be interpreted as our macro.) Finally, every axiom and inference rule used in their proof is derivable in our proof system.

**Lemma 13.** Let  $R$  contain all the cowell-formed cotemplates  $r$  such that  $e(s) \xrightarrow{r}$  and  $r$  is  $\rho$ -well-formed. Then  $e(s) \sqsubseteq_\rho \sum_{r \in R} e(r)$ .  $\square$

**Lemma 14.**  $e(s) \sqsubseteq_\rho \sum_{t \in T_f(s, \rho)} e(t)$ .  $\square$

Note that the summations in the two lemmas above are finite because  $R$  and  $T_f(s, \rho)$  are finite modulo alpha equivalence. For instance, finiteness of  $R$  is a direct consequence of the following two observations. For every  $r \in R$ , we have  $fn(r) \subset fn(e(s))$ , and since  $e(s)$  is a finite process, the length of traces in  $R$  is bounded. We are now ready to establish the completeness of the proof system.

**Theorem 4.** For finite  $L\pi$  processes  $P, Q$  and a set of names  $\rho$ ,  $P \sqsubseteq_\rho Q$  if and only if  $P \stackrel{\sim}{\sqsubseteq}_\rho Q$ .

*Proof.* The only-if part follows from the soundness of laws in table 3. We prove the if part. By Lemma 11 and soundness of the proof system, without loss of generality, we can assume that both  $P$  and  $Q$  are in canonical form, i.e.  $P$  is of form  $\sum_{s \in S_1} e(s)$  and  $Q$  is of form  $\sum_{s \in S_2} e(s)$ . Using Lemma 13, and laws  $I3$ ,  $A1$ , we get  $P \sqsubseteq_\rho \sum_{r \in R} e(r)$ , where  $R$  is the set of  $\rho$ -well-formed cowell-formed cotemplates that  $P$  exhibits. Using Lemma 14 and laws  $I3$ ,  $A1$ , we have  $\sum_{r \in R} e(r) \sqsubseteq_\rho \sum_{t \in T} e(t)$ , where  $T = \cup_{r \in R} T_f(r, \rho)$ . Note that since every  $r \in R$  is a cotemplate, so is every  $t \in T$ . Let  $t \in T$ . Then  $t \in T_f(r, \rho)$  for some  $\rho$ -well-formed  $r$  that  $P$  exhibits. Using the characterization of may preorder based on  $T_f(r, \rho)$ , we have  $P \sqsubseteq_\rho Q$  implies there is  $s' \preceq t$  such that  $Q \xRightarrow{s'}$ . It follows that for some  $s \in S_2$ ,  $e(s) \xRightarrow{s'}$ . Since  $Q \xRightarrow{s'}$ , by locality,  $s'$  is cowell-formed. From the facts that  $s' \preceq t$  and  $t$  is a cotemplate, it follows that  $s'$  is a cotemplate. Then by Lemma 12.2 and law  $I4$ ,  $e(t) \sqsubseteq_\rho e(s')$ . Further, by Lemma 12.1 and law  $I4$ ,  $e(s') \sqsubseteq_\rho e(s)$ . Hence by transitivity of  $\sqsubseteq_\rho$ , we have  $e(t) \sqsubseteq_\rho e(s)$ . Since  $t \in T$  is arbitrary, using laws  $I3$ ,  $A1$ , and  $A2$ , we deduce  $\sum_{t \in T} e(t) \sqsubseteq_\rho \sum_{s \in S_2} e(s)$ . The result follows from transitivity of  $\sqsubseteq_\rho$ .  $\square$

We obtain a complete proof system for  $L\pi_{=}$  by dropping axiom  $A21$  and adding the following two for the match operator:  $[x = x]P = P$ , and  $[x = y]P = 0$  if  $x \neq y$ . Completeness of the resulting proof system can be established by simple modifications to the proofs above.

## 5 Related Work

We have provided an alternate characterization of a parameterized version of may testing for asynchronous variants of  $\pi$ -calculus with locality and no name matching. We have exploited the characterizations to obtain complete axiomatizations of the may preorder for finitary fragments of the calculi. Our results extend the ones obtained by Boreale, De Nicola, and Pugliese for asynchronous  $\pi$ -calculus [3]. We now compare our work with other related research.

Hennessy and Rathke [6] study typed versions of three behavioral equivalences, namely may and must equivalences, and barbed congruence in a typed  $\pi$ -calculus where the type system allows names to be tagged with input/output capabilities. In the typed calculus, one can express processes that selectively distribute different capabilities on names. The locality property is a special case in which only the output capability on names can be passed. A novel labeled transition system is defined over configurations which are process terms with two typed environments, one that constrains the process and the other the environment. It is shown that the standard definitions of trace and acceptance sets [5] defined over the new transition system characterize may and must preorders respectively. In comparison to our work, the typed calculus of Hennessy and Rathke is synchronous and is equipped with name matching, whereas  $L\pi_{=}$  is asynchronous, and  $L\pi$  is asynchronous with no name matching. Further,  $L\pi_{=}$  has no capability types and hence we obtain a simpler characterization of may testing for it, which is based on the usual early style labeled transition system.

Finally, we have also a given an axiomatization of may testing, which is not pursued by Hennessy and Rathke.

There have been extensive investigations of bisimulation-based behavioral equivalences on  $L\pi$  and related variants of  $\pi$ -calculus, which are properly contained in may testing which is trace based. Merro and Sangiorgi [9] investigate barbed congruence in  $L\pi$ , and show that a variant of asynchronous early bisimulation provides an alternate characterization for the congruence. Boreale and Sangiorgi [4] study typed barbed equivalence for typed (synchronous)  $\pi$ -calculus with capability types and no name matching, and show that the equivalence is characterized by a typed variant of bisimulation. Merro [8] characterizes barbed congruence in the more restricted setting of asynchronous  $\pi$ -calculus with no name matching (no capability types, and no locality in particular). He defines synonymous bisimulation and shows that it characterizes barbed congruence in this setting.

## References

- [1] G. Agha. *Actors: A Model of Concurrent Computation in Distributed Systems*. MIT Press, 1986.
- [2] R. Amadio, I. Castellani, and D. Sangiorgi. On Bisimulations for Asynchronous  $\pi$ -Calculus. In *Proceedings of CONCUR '96*. Springer-Verlag, 1996. LNCS 1119.
- [3] M. Boreale, R. De Nicola, and R. Pugliese. Trace and testing equivalence on asynchronous processes, 2002.
- [4] Michele Boreale and D. Sangiorgi. Bisimulation in Name Passing Calculi without Matching. *Proceedings of LICS*, 1998.
- [5] M. Hennessy. *Algebraic Theory of Processes*. MIT Press, 1988.
- [6] Matthew Hennessy and Julian Rathke. Typed behavioral equivalences for processes in the presence of subtyping. Technical report, University of Sussex, Computer Science, March 2001.
- [7] K. Honda and M. Tokoro. An Object Calculus for Asynchronous Communication. In *Fifth European Conference on Object-Oriented Programming*, July 1991. LNCS 512, 1991.
- [8] M. Merro. On equators in asynchronous name-passing calculi without matching. *Electronic Notes in Theoretical Computer Science*, 27, 1999.
- [9] M. Merro and D. Sangiorgi. On Asynchrony in Name-Passing Calculi. In *Proceeding of ICALP '98*. Springer-Verlag, 1998. LNCS 1443.
- [10] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes (Parts I and II). *Information and Computation*, 100:1–77, 1992.
- [11] R. De Nicola and M. Hennesy. Testing equivalence for processes. In *Theoretical Computer Science*, volume 34, pages 83–133, 1984.
- [12] B. C. Pierce and D. N. Turner. Pict: A programming Language Based on the  $\pi$ -Calculus. Technical Report CSCI-476, Indiana University, March 1997.
- [13] Prasanna Thati, Reza Ziaei, and Gul Agha. A theory of may testing for asynchronous calculi with locality and no name matching. Technical Report UIUCDCS-R-2002-2277, University of Illinois at Urbana Champaign, May 2002.



## A Appendix

The structural congruence relation on  $L\pi$  processes  $\equiv$  is the smallest congruence that is closed under the following rules.

1.  $P|0 \equiv P$ ,  $P|Q \equiv Q|P$ , and  $P|(Q|R) \equiv (P|Q)|R$
2.  $(\nu x)0 \equiv 0$ ,  $(\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$ , and  $P|(\nu x)Q \equiv (\nu x)(P|Q)$  if  $x \notin fn(P)$ .
3.  $!P \equiv P|!P$ .

It is the case that structurally congruent processes have the same transitions, i.e. if  $P \xrightarrow{\beta} Q$ ,  $P \equiv P'$  and  $Q \equiv Q'$  then  $P' \xrightarrow{\beta} Q'$ . Structural congruence rules for  $L\pi_{\equiv}$  contain the additional rule:  $[x = x]P \equiv P$ .

**Proof of Theorem 1:** Let  $P \lesssim_{\rho_1} Q$ . Suppose  $P \underline{\text{may}} O$  and  $rcp(O) \cap \rho_2 = \emptyset$ . Since  $\rho_1 \subset \rho_2$ , we have  $rcp(O) \cap \rho_1 = \emptyset$ . Then since  $P \lesssim_{\rho_1} Q$ , we have  $Q \underline{\text{may}} O$ . Hence  $P \lesssim_{\rho_2} Q$ .

Let  $fn(P) \cup fn(Q) \subset \rho_1$  and  $P \lesssim_{\rho_2} Q$ . Suppose  $P \underline{\text{may}} O$  and  $rcp(O) \cap \rho_1 = \emptyset$ . We have to show  $Q \underline{\text{may}} O$ . Let  $\{\tilde{x}\} = rcp(O)$ . Then, we have  $rcp((\nu \tilde{x})O) = \emptyset$ . Since  $fn(P) \cap \{\tilde{x}\} = \emptyset$ , we have  $P|(\nu \tilde{x})O \equiv (\nu \tilde{x})(P|O)$ . From this, *RES* rule, and  $P|O \xrightarrow{\bar{\mu}\mu}$ , we deduce  $P|(\nu \tilde{x})O \xrightarrow{\bar{\mu}\mu}$ , i.e.  $P \underline{\text{may}} (\nu \tilde{x})O$ . Now, since  $P \lesssim_{\rho_2} Q$  we have  $Q \underline{\text{may}} (\nu \tilde{x})O$ . From  $fn(Q) \cap \{\tilde{x}\} = \emptyset$  we have  $(\nu \tilde{x})(Q|O) \equiv Q|(\nu \tilde{x})O$ . It follows that  $(\nu \tilde{x})(Q|O) \xrightarrow{\bar{\mu}\mu}$ , and hence  $Q|O \xrightarrow{\bar{\mu}\mu}$ .  $\square$

**Lemma 15.** *Let  $P \xrightarrow{s}$ . Then  $s = s_1.(\hat{y})xy.s_2$  implies  $x \in rcp(rcp(P), s_1)$ .*

**Proof:** Suppose  $P \xrightarrow{s_1} P_1 \xrightarrow{(\hat{y})xy} P_2 \xrightarrow{s_2}$ . We observe that  $P_1 \xrightarrow{(\hat{y})xy}$  if and only if  $x \in rcp(P_1)$ . So we are done if we show that  $rcp(P_1) \subset rcp(rcp(P), s_1)$ . This can be shown by a simple induction on  $s$  using the locality property.  $\square$

**Proof of Lemma 3:** The sequence  $P|O \xrightarrow{\bar{\mu}\mu}$  can be unzipped into  $P \xrightarrow{s}$  and  $O \xrightarrow{\bar{s}.\bar{\mu}\mu}$  for some trace  $s$ . While unzipping, we can choose bound names such that  $bn(\bar{s}) \cap rcp(O) = \emptyset$ . To show that  $s$  is  $\rho$ -well-formed, we show that if  $s = s_1.(\hat{y})\bar{x}y.s_2$ , then  $x \notin rcp(\rho, s_1)$ . By applying Lemma 15 to  $O \xrightarrow{\bar{s}.\bar{\mu}\mu}$  we deduce that  $x \in rcp(rcp(O), \bar{s}_1)$ . If  $x \in rcp(O)$ , then  $x \notin rcp(\rho, s_1)$  because  $\rho \cap rcp(O) = bn(s_1) \cap rcp(O) = \emptyset$ . If  $x \notin rcp(O)$ , then  $x$  must be the bound argument of an output in  $\bar{s}_1$ , which again implies  $x \notin rcp(\rho, s_1)$ . Therefore,  $s$  is  $\rho$ -well-formed.  $\square$

**Lemma 16.** *If  $P'_0 \in P_0[w/y]$ , then*

1.  $P_0 \xrightarrow{(\hat{v})uv} P_1$  and  $w \notin \hat{v}$  implies there is  $P'_1 \in P_1[w/y]$  such that  $P'_0 \xrightarrow{(\hat{v})uv} P'_1$ .
2.  $P_0 \xrightarrow{(\hat{v})\bar{u}v} P_1$  and  $w \notin \hat{v}$  implies there is  $P'_1 \in P_1[w/y]$  such that  $P'_0 \xrightarrow{\alpha} P'_1$  where for some  $u' \in u[w/y]$  and  $v' \in v[w/y]$

$$\alpha = \begin{cases} \bar{u'}v' & \text{if } \hat{v} = \emptyset \\ \bar{u'}(v) & \text{otherwise} \end{cases}$$

3.  $P_0 \xrightarrow{\tau} P_1$  implies one of the following;
- (a) There is  $P'_1 \in P_1[w/y]$  such that  $P'_0 \xrightarrow{\tau} P'_1$ .
  - (b)  $P_1 \equiv (\nu \hat{z})Q$ ,  $w, y \notin \hat{z}$ , and there is  $P'_1 \in Q[w/y]$  such that  $P'_0 \xrightarrow{(\hat{z})\overline{wz}, yz} P'_1$ .

**Proof of Lemma 16.1:** Since we work modulo alpha equivalence on process, without loss of generality we may assume the hygiene condition  $w, y \notin \text{bn}(P_0)$ .

The proof is by induction on the derivation of  $P_0 \xrightarrow{(\hat{v})uv} P_1$ . For the base case, we have  $P_0 = u(z).Q$ ,  $P_1 = Q\{v/z\}$ , the last derivation step is an application of *INP* rule, and  $\hat{v} \cap \text{fn}(P_0) = \emptyset$ . By hygiene condition  $z \neq w, y$ . Let  $P'_0 \in P_0[w/y]$ . Then  $P'_0 = u(z).Q'$  for some  $Q' \in Q[w/y]$ . By *INP* rule we have,  $u(z).Q' \xrightarrow{uv} Q'\{v/z\}$ . Since  $w \notin \hat{v}$  it follows that  $u(z).Q' \xrightarrow{(\hat{v})uv} Q'\{v/z\}$ . Now, since  $z \neq w, y$ , we have  $Q'\{v/z\} \in Q\{v/z\}[w/y]$  and the lemma follows.

For the induction step, there are three cases.

1.  $P_0 = Q_0|R$ ,  $P_1 = Q_1|R$ , and the last derivation step is an application of *PAR* rule:

$$PAR: \frac{Q_0 \xrightarrow{(\hat{v})uv} Q_1}{Q_0|R \xrightarrow{(\hat{v})uv} Q_1|R} \hat{v} \cap \text{fn}(R) = \emptyset$$

Then  $P'_0 = Q'_0|R'$  for some  $Q'_0 \in Q_0[w/y]$  and  $R' \in R[w/y]$ . By induction hypothesis we have  $Q'_0 \xrightarrow{(\hat{v})uv} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . From  $w \notin \hat{v}$  and  $\hat{v} \cap \text{fn}(R) = \emptyset$  we deduce  $\hat{v} \cap \text{fn}(R') = \emptyset$ . Then by *PAR* rule we have  $Q'_0|R' \xrightarrow{(\hat{v})uv} Q'_1|R'$ , and the lemma follows from the fact that  $Q'_1|R' \in P_1[w/y]$ .

2.  $P_0 = (\nu z)Q_0$ ,  $P_1 = (\nu z)Q_1$  and the last derivation step is an application of *RES* rule:

$$RES: \frac{Q_0 \xrightarrow{(\hat{v})uv} Q_1}{(\nu z)Q_0 \xrightarrow{(\hat{v})uv} (\nu z)Q_1} z \notin \{u, v\}$$

By the hygiene condition, we have  $z \neq w, y$ . Then  $P'_0 = (\nu z)Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . By induction hypothesis,  $Q'_0 \xrightarrow{(\hat{v})uv} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then by *RES* rule we have  $(\nu z)Q'_0 \xrightarrow{(\hat{v})uv} (\nu z)Q'_1$ , and the lemma follows from the fact that  $(\nu z)Q'_1 \in P_1[w/y]$ .

3.  $P_0 = !Q_0$ ,  $P_1 = Q_1$  and the last derivation step is an application of *REP* rule:

$$REP: \frac{Q_0|!Q_0 \xrightarrow{(\hat{v})uv} Q_1}{!Q_0 \xrightarrow{(\hat{v})uv} Q_1}$$

Then  $P'_0 = !Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . Since  $Q'_0|!Q'_0 \in (Q_0|!Q_0)[w/y]$ , by induction hypothesis we have  $Q'_0|!Q'_0 \xrightarrow{(\hat{v})uv} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then by *REP* rule we have  $!Q'_0 \xrightarrow{(\hat{v})uv} Q'_1$ , and the lemma follows.  $\square$

**Proof of Lemma 16.2:** Since we work modulo alpha equivalence on process, without loss of generality we may assume the hygiene condition  $w, y \notin \text{bn}(P_0)$ .

The proof is by induction on the derivation of  $P_0 \xrightarrow{(\hat{v})\overline{uv}} P_1$ . There are two base cases.

1.  $P_0 = \overline{uv}$ ,  $P_1 = 0$  and the last derivation step is an application of *OUT* rule.

$$OUT: \overline{uv} \xrightarrow{\overline{uv}} 0$$

Then  $P'_0 = \overline{u'}v'$  for some  $u' \in u[w/y]$  and  $v' \in v[w/y]$ . By *OUT* rule we have  $\overline{u'}v' \xrightarrow{\overline{u'}v'} 0$ , and the lemma follows.

2.  $P_0 = (\nu v)Q_0$ ,  $P_1 = Q_1$  and the last derivation step is an application of *OPEN* rule.

$$OPEN: \frac{Q_0 \xrightarrow{\overline{uv}} Q_1}{(\nu v)Q_0 \xrightarrow{\overline{u(v)}} Q_1} u \neq v$$

By hygiene condition, we have  $v \neq w, y$ . Then  $P'_0 = (\nu v)Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . By induction hypothesis  $Q'_0 \xrightarrow{\overline{u'}v'} Q'_1$  for some  $u' \in u[w/y]$ ,  $v' \in v[w/y]$ , and  $Q'_1 \in Q_1[w/y]$ . Further, since  $y \neq v$  we have  $v' = v$ , and since  $w \neq v$  we have  $u' \neq v$ . Then by *RES* rule we have  $(\nu v)Q'_0 \xrightarrow{\overline{u'}(v)} Q'_1$ , and the lemma follows.

For the induction step there are three cases.

1.  $P_0 = Q_0|R$ ,  $P_1 = Q_1|R$ , and the last derivation step is an application of *PAR* rule.

$$PAR: \frac{Q_0 \xrightarrow{(\hat{v})\overline{uv}} Q_1}{Q_0|R \xrightarrow{(\hat{v})\overline{uv}} Q_1|R} \hat{v} \cap fn(R) = \emptyset$$

Then  $P'_0 = Q'_0|R'$  for some  $Q'_0 \in Q_0[w/y]$  and  $R' \in R[w/y]$ . By induction hypothesis,  $Q'_0 \xrightarrow{\alpha} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$  and  $\alpha$  as stated in the lemma. We have  $bn(\alpha) = \hat{v}$ . From  $w \notin \hat{v}$  and  $\hat{v} \cap fn(R) = \emptyset$  we deduce  $bn(\alpha) \cap fn(R') = \emptyset$ . Then by *PAR* rule  $Q'_0|R' \xrightarrow{\alpha} Q'_1|R'$ , and the lemma follows from the fact that  $Q'_1|R' \in P_1[w/y]$ .

2.  $P_0 = (\nu z)Q_0$ ,  $P_1 = (\nu z)Q_1$ , and the last derivation step is an application of *RES* rule:

$$RES: \frac{Q_0 \xrightarrow{(\hat{v})\overline{uv}} Q_1}{(\nu z)Q_0 \xrightarrow{(\hat{v})\overline{uv}} (\nu z)Q_1} z \neq u, v$$

By hygiene condition we have  $z \neq w, y$ . Then  $P'_0 = (\nu z)Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . By induction hypothesis,  $Q'_0 \xrightarrow{\alpha} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$  and  $\alpha$  as stated in the lemma. Further since  $z \notin \{w, y, u, v\}$  we have  $z \notin n(\alpha)$ . Then by *RES* rule we have  $(\nu z)Q'_0 \xrightarrow{\alpha} (\nu z)Q'_1$ , and the lemma follows from the fact that  $(\nu z)Q'_1 \in P_1[w/y]$ .

3.  $P_0 = !Q_0$ ,  $P_1 = Q_1$  and the last derivation step is an application of *REP* rule:

$$REP: \frac{Q_0|!Q_0 \xrightarrow{(\hat{v})\overline{uv}} Q_1}{!Q_0 \xrightarrow{(\hat{v})\overline{uv}} Q_1}$$

Then  $P'_0 = !Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . Since  $Q'_0 ! Q'_0 \in (Q_0 ! Q_0)[w/y]$ , by induction hypothesis we have  $Q'_0 ! Q'_0 \xrightarrow{\alpha} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$  and  $\alpha$  as stated in the lemma. Then by *REP* rule we have  $!Q'_0 \xrightarrow{\alpha} Q'_1$ , and the lemma follows.  $\square$

**Proof of Lemma 16.3:** Since we work modulo alpha equivalence on process, without loss of generality we may assume the hygiene condition  $w, y \notin \text{bn}(P_0, P_1)$ . The proof is by induction on derivation of  $P_0 \xrightarrow{\tau} P_1$ . There are two base cases depending on the last derivation step.

1.  $P_0 = Q_0|R_0$ ,  $P_1 = Q_1|R_1$  and the last derivation step is

$$COM: \frac{Q_0 \xrightarrow{\bar{u}v} Q_1 \quad R_0 \xrightarrow{uv} R_1}{Q_0|R_0 \xrightarrow{\tau} Q_1|R_1}$$

Then  $P'_0 = Q'_0|R'_0$  for some  $Q'_0 \in Q_0[w/y]$  and  $R'_0 \in R_0[w/y]$ . By Lemma 16.2,  $Q'_0 \xrightarrow{\bar{u}'v'} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ ,  $u' \in u[w/y]$  and  $v' \in v[w/y]$ . There are two subcases.

- (a)  $u' = u$ : Since random substitution on processes does not change input prefixes, it follows that  $R'_0 \xrightarrow{uv'} R'_1$ ,  $R'_1 \in R_1[w/y]$ . Using *COM* we have  $Q'_0|R'_0 \xrightarrow{\tau} Q'_1|R'_1$ . Then condition 1 of lemma holds with  $P'_1 = Q'_1|R'_1$ .
  - (b)  $u' = w$ : Then  $u = y$ . Since random substitution on processes does not change input prefixes, it follows that  $R'_0 \xrightarrow{yv'} R'_1$ ,  $R'_1 \in R_1[w/y]$ . Then by *PAR* rule we have  $P'_0 = Q'_0|R'_0 \xrightarrow{\bar{w}v'} Q'_1|R'_0 \xrightarrow{yv'} Q'_1|R'_1$ . Then condition 2 of lemma holds with  $z = v'$  and  $\hat{z} = \emptyset$ ,  $Q = Q_1|R_1$ , and  $P'_1 = Q'_1|R'_1$ .
2.  $P_0 = Q_0|R_0$ ,  $P_1 = (\nu v)(Q_1|R_1)$  and the last derivation step is

$$CLOSE: \frac{Q_0 \xrightarrow{\bar{u}(v)} Q_1 \quad R_0 \xrightarrow{uv} R_1}{Q_0|R_0 \xrightarrow{\tau} (\nu v)(Q_1|R_1)} \quad v \notin \text{fn}(R_0)$$

Then  $P'_0 = Q'_0|R'_0$  for some  $Q'_0 \in Q_0[w/y]$  and  $R'_0 \in R_0[w/y]$ . By hygiene condition  $v \neq w, y$ . Then by Lemma 16.2, we have  $Q'_0 \xrightarrow{\bar{u}'(v)} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$  and  $u' \in u[w/y]$ . There are two subcases.

- (a)  $u' = u$ : Since random substitution on processes does not change input prefixes, it follows that  $R'_0 \xrightarrow{uv} R'_1$ ,  $R'_1 \in R_1[w/y]$ . From  $v \neq w, y$  and  $v \notin \text{fn}(R_0)$  we deduce  $v \notin \text{fn}(R'_0)$ . Using *COM* we have  $Q'_0|R'_0 \xrightarrow{\tau} (\nu v)(Q'_1|R'_1)$ . Then condition 1 of lemma holds with  $P'_1 = (\nu v)(Q'_1|R'_1)$ .
- (b)  $u' = w$ : Then  $u = y$ . Since random substitution on processes does not change input prefixes, it follows that  $R'_0 \xrightarrow{yv} R'_1$ ,  $R'_1 \in R_1[w/y]$ . From  $v \neq w, y$  and  $v \notin \text{fn}(R_0)$  we deduce  $v \notin \text{fn}(R'_0)$ . Then by *PAR* rule we have  $P'_0 = Q'_0|R'_0 \xrightarrow{\bar{w}(v)} Q'_1|R'_0 \xrightarrow{yv} Q'_1|R'_1$ . Then condition 2 of the lemma holds with  $z = v$  and  $\hat{z} = \{v\}$ ,  $Q = Q_1|R_1$ , and  $P'_1 = Q'_1|R'_1$ .

For the induction step, there are three cases.

1.  $P_0 = Q_0|R$ ,  $P_1 = Q_1|R$ , and the last derivation step is an application of *PAR* rule:

$$PAR: \frac{Q_0 \xrightarrow{\tau} Q_1}{Q_0|R \xrightarrow{\tau} Q_1|R}$$

Then  $P'_0 = Q'_0|R'$  for some  $Q'_0 \in Q_0[w/y]$  and  $R' \in R[w/y]$ . By induction hypothesis we have two cases.

- (a)  $Q'_0 \xrightarrow{\tau} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then by *PAR* rule,  $P'_0 = Q'_0|R' \xrightarrow{\tau} Q'_1|R'$  and condition 1 of the lemma holds with  $P'_1 = Q'_1|R'$ .
  - (b)  $Q_1 \equiv (\nu \hat{z})S$ ,  $w, y \notin \hat{z}$ ,  $Q'_0 \xrightarrow{(\hat{z})\overline{wz}, yz} S'$  for some  $S' \in S[w/y]$ . Without loss of generality we may assume  $\hat{z} \cap fn(R) = \emptyset$ . Then  $P_1 \equiv (\nu \hat{z})(S|R)$ , and  $\hat{z} \cap fn(R') = \emptyset$ . Then by *PAR* rule  $P'_0 = Q'_0|R' \xrightarrow{(\hat{z})\overline{wz}, yz} S'|R'$ . Then condition 2 of the lemma holds.
2.  $P_0 = (\nu v)Q_0$ ,  $P_1 = (\nu v)Q_1$  and the last derivation step is an application of *RES* rule:

$$RES: \frac{Q_0 \xrightarrow{\tau} Q_1}{(\nu v)Q_0 \xrightarrow{\tau} (\nu v)Q_1}$$

By hygiene condition, we have  $v \neq w, y$ . Then  $P'_0 = (\nu v)Q'_0$  for some  $Q'_0 \in Q_0[w/y]$ . By induction hypothesis we have two cases.

- (a)  $Q'_0 \xrightarrow{\tau} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then by *RES* rule,  $(\nu v)Q'_0 \xrightarrow{\tau} (\nu v)Q'_1$  and condition 1 of the lemma holds with  $P'_1 = (\nu v)Q'_1$ .
  - (b)  $Q_1 \equiv (\nu \hat{z})S$ ,  $w, y \notin \hat{z}$ ,  $Q'_0 \xrightarrow{(\hat{z})\overline{wz}, yz} S'$  for some  $S' \in S[w/y]$ . Without loss of generality we may assume  $v \notin \hat{z}$ . There are two subcases:
    - $v = z$ . Then  $\hat{z} = \emptyset$ . Then by *OPEN* rule  $P'_0 \equiv (\nu v)Q'_0 \xrightarrow{\overline{w(v)}, yv} S'$ . Then condition 2 of the lemma holds with  $z = v$ ,  $\hat{z} = \{v\}$ ,  $P_1 \equiv (\nu v)S$  and  $P'_1 = S'$ .
    - $v \neq z$ : Then by *RES* rule  $P'_0 \equiv (\nu v)Q'_0 \xrightarrow{(\hat{z})\overline{wz}, yz} (\nu v)S'$ . Then condition 2 of the lemma holds with  $P_1 \equiv (\nu \hat{z})(\nu v)S$  and  $P'_1 = (\nu v)S'$ .
3. The case where the last derivation step is an application of *REP* rule is straightforward.  $\square$

**Proof of Lemma 4:** Since we work modulo alpha equivalence on traces we can assume the hygiene condition  $bn(s) \cap \{w, y\} = \emptyset$ . Proof is by induction on the number of steps in the transition sequence abbreviated by  $P \xrightarrow{\bar{s}}$ . The base case is obvious by letting  $s' = \epsilon$  and the fact that  $\epsilon \in \epsilon[w/y]$ . For the induction step, suppose  $P \xrightarrow{\bar{s}}$  can be written as  $P \xrightarrow{\bar{\beta}} P_1 \xrightarrow{\bar{\tau}}$ . There are three cases depending on  $\beta$ :

1.  $P \xrightarrow{(\hat{v})uv} P_1$ : Due to locality,  $rcp(P_1) \subset rcp(P)$ . By Lemma 16.1,  $P' \xrightarrow{(\hat{v})uv} P'_1$  for some  $P'_1 \in P_1[w/y]$ . Let  $\bar{\beta}' = (\hat{v})uv$ .

2.  $P \xrightarrow{(\hat{v})\overline{u}v} P_1$ : We have  $rcp(P_1) \subset rcp(P) \cup \hat{v}$ . Since by hygiene condition  $y \notin \hat{v}$ , we have  $y \notin rcp(P_1)$ . By Lemma 16.2,  $P' \xrightarrow{\overline{\beta'}} P'_1$  for some  $P'_1 \in P_1[w/y]$  and  $\overline{\beta'}$  is  $\alpha$  as stated in Lemma 16.2.
3.  $P \xrightarrow{\tau} P_1$ : By locality  $y \notin rcp(P_1)$ . Now, we apply Lemma 16.3. From  $y \notin rcp(P)$  and  $P' \in P[w/y]$  we deduce  $y \notin rcp(P')$ . It follows that only the first case of Lemma 16.3 applies, because in the second case  $y$  is used as the subject of an input action which implies  $y \in rcp(P')$ . Therefore,  $P' \xrightarrow{\tau} P'_1$  for some  $P'_1 \in P_1[w/y]$ .

In all cases,  $y \notin rcp(P_1)$ . Then by induction hypothesis,  $P'_1 \xrightarrow{\overline{r'}} r' \in r[w/y]$ . The result follows from the observation that in cases 1 and 2,  $\beta'.r' \in (\beta.r)[w/y]$ , and in case 3,  $r' \in r[w/y] = s[w/y]$ .  $\square$

**Proof of Lemma 5:** Since we work modulo alpha equivalence on traces we can assume  $bn(s) \cap \{w, y\} = \emptyset$ . Proof is by induction on the number of steps in transition sequence  $P \xRightarrow{\overline{s}}$ . The base case is obvious with  $s' = \epsilon$  because  $\epsilon \in F(\epsilon[w/y], w, y)$ . For the induction step, suppose  $P \xRightarrow{\overline{s}}$  can be written as  $P \xrightarrow{\overline{\beta}} P_1 \xRightarrow{\overline{\tau}}$ . We have three cases:

1.  $P \xrightarrow{(\hat{v})uv} P_1$ : Since  $w \notin \hat{v}$ , by Lemma 16.1, there is  $P'_1 \in P_1[w/y]$  such that  $P' \xrightarrow{(\hat{v})uv} P'_1$ . By induction hypothesis  $P'_1 \xrightarrow{\overline{r'}} r' \in F(r[w/y], w, y)$ . By letting  $s' = (\hat{v})\overline{u}v.r'$ , we have  $s' \in F((\hat{v})\overline{u}v.r[w/y], w, y) = F(s[w/y], w, y)$  and the lemma follows.
2.  $P \xrightarrow{(\hat{v})\overline{u}v} P_1$ : Since  $w \notin \hat{v}$ , by Lemma 16.2, there is  $P'_1 \in P_1[w/y]$  such that  $P' \xrightarrow{\alpha} P'_1$  where for some  $u' \in u[w/y]$  and  $v' \in v[w/y]$

$$\alpha = \begin{cases} \overline{u'}v' & \text{if } \hat{v} = \emptyset \\ \overline{u'}(v) & \text{otherwise} \end{cases}$$

By induction hypothesis  $P'_1 \xrightarrow{\overline{r'}} r' \in F(r[w/y], w, y)$ . By letting  $s' = \overline{\alpha}.r'$ , we have  $s' \in F(\overline{\alpha}.r[w/y], w, y) = F(s[w/y], w, y)$  and the lemma follows.

3.  $P \xrightarrow{\tau} P_1$ : Then  $r = s$ . By Lemma 16.3 we have two cases:
  - (a) There is  $P'_1 \in P_1[w/y]$  such that  $P' \xrightarrow{\tau} P'_1$ . By induction hypothesis,  $P'_1 \xrightarrow{\overline{s'}} s' \in F(s[w/y], w, y)$ . The lemma follows from  $P' \xrightarrow{\overline{s'}} s'$ .
  - (b)  $P_1 \equiv (\nu \hat{z})Q_1$ ,  $w, y \notin \hat{z}$ , and there is  $P'_1 \in Q_1[w/y]$  such that  $P' \xrightarrow{(\hat{z})\overline{w}z} P'_1$ . From  $w, y \notin \hat{z}$  we have  $(\nu \hat{z})P'_1 \in ((\nu \hat{z})Q_1)[w/y]$ . By induction hypothesis,  $(\nu \hat{z})P'_1 \xrightarrow{\overline{s''}} s'' \in F(s[w/y], w, y)$ . It is easy to show that  $P'_1 \xrightarrow{[\hat{z}]\overline{s''}} s''$ . The lemma follows from  $(\hat{z})wz.\overline{y}z.[\hat{z}]s'' \in F(s[w/y], w, y)$  and  $P' \xrightarrow{(\hat{z})\overline{w}z} yz \xrightarrow{[\hat{z}]\overline{s''}} s''$ .  $\square$

**Lemma 17.** *If  $P \xrightarrow{xy} P_1$  and  $w \notin fn(P)$ , then there is  $P'_1 \in P_1[w/y]$  such that  $P \xrightarrow{x(w)} P'_1$ .*

**Proof:** Since we work modulo alpha equivalence on processes, without loss of generality we may assume the hygiene condition  $w \notin bn(P)$ . The proof is by induction on the derivation of  $P \xrightarrow{xy} P_1$ . For the base case, we have  $P = x(z).Q$ ,  $P_1 = Q\{y/z\}$ , and the last derivation step is an application of *INP* rule. Then, by *INP* rule and  $w \notin fn(P)$ , we have  $x(z).Q \xrightarrow{x(w)} Q\{w/z\}$ . Furthermore, by locality,  $z$  occurs only in output terms in  $Q\{w/z\}$ . Therefore,  $Q\{w/z\} \in Q\{y/z\}[w/y]$ , from which the lemma follows.

For the induction step, there are three cases.

1.  $P = Q|R$ ,  $P_1 = Q_1|R$ , and the last derivation step is an application of *PAR* rule:

$$PAR: \frac{Q \xrightarrow{xy} Q_1}{Q|R \xrightarrow{xy} Q_1|R}$$

From  $w \notin fn(P)$ , it follows that  $w \notin fn(Q)$  and  $w \notin fn(R)$ . Since  $w \notin fn(Q)$ , by induction hypothesis we have  $Q \xrightarrow{x(w)} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then, since  $w \notin fn(R)$ , by *PAR* rule we have  $Q|R \xrightarrow{x(w)} Q'_1|R$ . By letting  $P'_1 = Q'_1|R$ , the lemma follows from  $Q'_1|R \in P_1[w/y]$ .

2.  $P = (\nu z)Q$ ,  $P_1 = (\nu z)Q_1$  and the last derivation step is an application of *RES* rule:

$$RES: \frac{Q \xrightarrow{xy} Q_1}{(\nu z)Q \xrightarrow{xy} (\nu z)Q_1} \quad z \notin \{x, y\}$$

By hygiene condition  $w \neq z$  and hence  $w \notin fn(Q)$ . Then by induction hypothesis,  $Q \xrightarrow{x(w)} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Now, since  $z \notin \{x, w\}$ , by *RES* rule we have  $(\nu z)Q \xrightarrow{x(w)} (\nu z)Q'_1$ , and the lemma follows from  $(\nu z)Q'_1 \in P_1[w/y]$ .

3.  $P = !Q$ ,  $P_1 = Q_1$  and the last derivation step is an application of *REP* rule:

$$REP: \frac{Q|!Q \xrightarrow{xy} Q_1}{!Q \xrightarrow{xy} Q_1}$$

Since  $w \notin fn(Q|!Q)$ , by induction hypothesis,  $Q|!Q \xrightarrow{x(w)} Q'_1$  for some  $Q'_1 \in Q_1[w/y]$ . Then by *REP* rule we have  $!Q \xrightarrow{x(w)} Q'_1$ , and the lemma follows.  $\square$

**Proof of Lemma 6:** Since we work modulo alpha equivalence on traces, we assume  $s$  is  $\rho$ -normal. The proof is by induction on the number of  $\tau$  transitions and the number of steps in the transition sequence  $P \xrightarrow{\bar{s}}$  ordered lexicographically. Base case is easy with  $t = \epsilon$ . For the induction step, we can write  $P \xrightarrow{\bar{\beta}} P_1 \xrightarrow{\bar{\tau}}$ . Now, there are four cases based on  $\beta$ .

1.  $\bar{\beta} = \tau$  : By locality,  $rcp(P_1) \subset rcp(P)$ , and hence  $\rho \cap rcp(P_1) = \emptyset$ . Further,  $r = s$  and the lemma follows from induction hypothesis.

2.  $\bar{\beta} = (\hat{y})\bar{x}y$  : From  $\rho$ -normality of  $s$ , we have  $\hat{y} \cap \rho = \emptyset$ . Since  $rcp(P_1) \subset rcp(P) \cup \hat{y}$ , we have  $\rho \cap rcp(P_1) = \emptyset$ . Now,  $r$  is  $\rho$ -normal and by induction hypothesis, there exists  $r' \in T(r, \rho)$  such that  $P_1 \xRightarrow{\bar{r}'} P_1$ . The lemma follows from  $(\hat{y})xy.r' \in T(s, \rho)$ .
3.  $\bar{\beta} = x(y)$  : By locality,  $(\rho \cup \{y\}) \cap rcp(P_1) = \emptyset$ . Furthermore,  $r$  is  $(\rho \cup \{y\})$ -normal. By induction hypothesis, there exists  $r' \in T(r, \rho \cup \{y\})$  such that  $P_1 \xRightarrow{\bar{r}'} P_1$ . The lemma follows from  $\bar{x}(y).r' \in T(s, \rho)$ .
4.  $\bar{\beta} = xy$  : Let  $w$  be fresh, that is  $w \notin fn(P) \cup n(s) \cup \rho$ . By Lemma 17, there is  $P'_1 \in P_1[w/y]$  such that  $P \xrightarrow{x(w)} P'_1$ . Because of locality,  $rcp(P'_1) \subset rcp(P)$  and therefore  $(\rho \cup \{w\}) \cap rcp(P'_1) = \emptyset$ . We have two subcases:
  - $y \in \rho$ : Then  $y \notin rcp(P)$  and by locality  $y \notin rcp(P_1)$ . Then by Lemma 4,  $P'_1 \xRightarrow{\bar{r}''} P'_1$  for some  $r'' \in r[w/y]$ . From the proof of Lemma 4, it is clear that the computation  $P'_1 \xRightarrow{\bar{r}''} P'_1$  has the same number of  $\tau$  transitions and computation steps as  $P_1 \xRightarrow{\bar{r}} P_1$ .
  - $y \notin \rho$ : Then by Lemma 5,  $P'_1 \xRightarrow{\bar{r}''} P'_1$  for some  $r'' \in F(r[w/y], w, y)$ . From the proof of Lemma 5, it is clear that if the number of  $\tau$  transitions in  $P'_1 \xRightarrow{\bar{r}''} P'_1$  is not less than that in  $P_1 \xRightarrow{\bar{r}} P_1$ , then both computations have exactly the same number of steps.
 In either case, without loss of generality we may assume  $r''$  is  $(\rho \cup \{w\})$ -normal. Then by induction hypothesis,  $P'_1 \xRightarrow{\bar{r}'} P'_1$  for some  $r' \in T(r'', \rho \cup \{w\})$ . The lemma follows from  $\bar{x}(w).r' \in T(s, \rho)$ .  $\square$

**Lemma 18.** *Let  $s$  be  $\rho$ -well-formed. Then for  $y \notin \rho$  all traces in  $F(s, x, y)$  are  $\rho$ -well-formed.*

**Proof:** Let  $r \in F(s, x, y)$ . We prove by induction on the derivation of  $r$ , that  $r$  is  $\rho$ -well-formed. The base case where  $r = \epsilon \in F(\epsilon, x, y)$  is obvious. For the induction step there are three cases one for each rule of Definition 6.

1.  $s = (\hat{v})uv.s', r = (\hat{v})uv.r'$  and  $r' \in F(s', x, y)$ . Suppose  $r = (\hat{v})uv.r_1.(\hat{w})\bar{z}w.r_2$ . Now,  $s'$  is  $\rho$ -well-formed, and by induction hypothesis  $r'$  is  $\rho$ -well-formed. Then we have  $z \notin rcp(r_1, \rho) = rcp((\hat{v})uv.r_1, \rho)$ . Hence  $r$  is  $\rho$ -well-formed.
2.  $s = (\hat{v})\bar{u}v.s', r = (\hat{v})\bar{u}v.r'$  and  $r' \in F(s', x, y)$ . Suppose  $r = (\hat{v})\bar{u}v.r_1.(\hat{w})\bar{z}w.r_2$ . Now,  $s'$  is  $(\rho \cup \hat{v})$ -well-formed. Then by induction hypothesis  $r'$  is  $(\rho \cup \hat{v})$ -well-formed. Then  $z \notin rcp(r_1, \rho \cup \hat{v}) = rcp((\hat{v})\bar{u}v.r_1, \rho)$ . Further, since  $s$  is  $\rho$ -well-formed,  $u \notin \rho$ . Hence  $r$  is  $\rho$ -well-formed.
3.  $r = (\hat{w})xw.\bar{y}w.[\hat{w}]r'$ , for some  $r' \in F(s, x, y)$ . Let  $r = (\hat{w})xw.\bar{y}w.[\hat{w}](r_1.(\hat{v})\bar{u}v.r_2)$ . Since  $[\hat{w}]r' \neq \perp$ , we have  $r = (\hat{w})xw.\bar{y}w.([\hat{w}]r_1).(\hat{v})\bar{u}v.r'_2$  for some  $r'_2$ . Now, by induction hypothesis  $r'$  is  $\rho$ -well-formed. Then  $u \notin rcp(r_1, \rho)$ . Since  $[\hat{w}]r_1$  changes only the first bound input with argument  $w$  in  $r_1$  (if any), it follows that  $rcp(r_1, \rho) = rcp([\hat{w}]r_1, \rho) = rcp((\hat{w})xw.\bar{y}w.[\hat{w}]r_1)$ . Now, since  $y \notin \rho$ , we conclude that  $r$  is  $\rho$ -well-formed.  $\square$



**Proof of Lemma 7:** We prove by induction on derivation of  $r \in T(s, \rho)$  that  $r$  is  $\rho$ -well-formed. The base case  $\epsilon \in T(\epsilon, \rho)$  is obvious. For the induction step there are four cases.

1.  $s = (\hat{y})xy.s'$ ,  $r = (\hat{y})xy.r'$  and  $r' \in T(s', \rho)$ . Suppose  $r = (\hat{y})xy.r_1.(\hat{v})\bar{u}v.r_2$ . Now  $s'$  is  $\rho$ -well-formed, and by induction hypothesis  $r'$  is  $\rho$ -well-formed. Then we have  $u \notin \text{rcp}(r_1, \rho) = \text{rcp}((\hat{y})xy.r_1, \rho)$ . Hence  $r$  is  $\rho$ -well-formed.
2.  $s = \bar{x}(y).s'$ ,  $r = \bar{x}(y).r'$  and  $r' \in T(s', \rho \cup \{y\})$ . Suppose  $r = \bar{x}(y).r_1.(\hat{v})\bar{u}v.r_2$ . Now,  $s'$  is  $\rho \cup \{y\}$ -well-formed, and by induction hypothesis  $r'$  is  $\rho$ -well-formed. Then we have  $u \notin \text{rcp}(r_1, \rho \cup \{y\}) = \text{rcp}(\bar{x}(y).r_1)$ . Further, since  $s$  is  $\rho$ -well-formed  $x \notin \rho$ . Hence  $r$  is  $\rho$ -well-formed.
3.  $s = \bar{x}y.s'$ ,  $y \in \rho$ ,  $r = \bar{x}(w).r'$  for some  $w$  fresh and  $r' \in T(r'', \rho \cup \{w\})$  for some  $r'' \in s'[w/y]$ . Now  $s'$  is  $\rho$ -well-formed. Since  $s$  is normal  $y \notin \text{bn}(s')$ . From this, and the facts that  $w$  is fresh and random substitution on traces does not change output actions, we have  $r''$  is  $\rho$ -well-formed. Moreover, since  $w$  is fresh we also have  $r''$  is  $\rho \cup \{w\}$ -well-formed. By induction hypothesis  $r'$  is  $\rho \cup \{w\}$ -well-formed. Further, since  $s$  is  $\rho$ -well-formed we have  $x \notin \rho$ . We conclude  $r$  is  $\rho$ -well-formed.
4.  $s = \bar{x}y.s'$ ,  $y \notin \rho$ ,  $r = \bar{x}(w).r'$  for some  $w$  fresh and  $r' \in T(r'', \rho)$  for some  $r'' \in F(s'[w/y], w, y)$ . Now,  $s'$  is  $\rho$ -well-formed, and by the argument in case 3 we have  $r''$  is  $\rho \cup \{w\}$ -well-formed. By Lemma 18,  $r'$  is  $\rho \cup \{w\}$ -well-formed. Further, since  $s$  is  $\rho$ -well-formed we have  $x \notin \rho$ . We conclude  $r$  is  $\rho$ -well-formed.  $\square$

**Lemma 19.**  $(\hat{y})\bar{x}y.r \preceq \bar{x}(w).s$  implies  $r \preceq s\{y/w\}$ .

**Proof:** By induction on the derivation of  $(\hat{y})\bar{x}y.r \preceq \bar{x}(w).s$ .  $\square$

**Lemma 20.**  $r \preceq s_1.s_2$  implies  $r = r_1.r_2$  for some  $r_1 \preceq s_1$ .

**Proof:** By induction on the derivation of  $r \preceq s_1.s_2$ .  $\square$

From now on, we use the following notation. For two sets of traces  $R$  and  $S$ , we say  $R \preceq S$ , if for every  $s \in S$  there is  $r \in R$  such that  $r \preceq s$ .

**Proof of Lemma 9:** The statement of the lemma can be stated as: for a prefix closed set  $R$  of traces with bounded length,  $R \preceq T(s, \rho)$  implies  $R \preceq T_f(s, \rho)$ .

The proof is by induction on the length of  $s$ . The base case is easy because  $T(\epsilon, \rho) = T_f(\epsilon, \rho) = \{\epsilon\}$ . For the induction step we have four cases, of which we only consider the one which is central to the proof, namely where  $s = \bar{x}y.s'$  and  $y \notin \rho$ . The others are routine. We are done if we construct a prefixed closed set  $R'$  of traces with bounded length such that  $R' \preceq T(s', \rho)$  and  $\bar{x}y.R' \subset R$ . For, by induction hypothesis,  $R' \preceq T(s', \rho)$  implies  $R' \preceq T_f(s', \rho)$ . Then,  $\bar{x}y.R' \preceq \bar{x}y.T_f(s', \rho) = T_f(s, \rho)$ , which together with  $\bar{x}y.R' \subset R$  implies  $R \preceq T_f(s, \rho)$ .

Suppose  $t' \in T(s', \rho)$  and  $l$  is the bound on the length of traces in  $R$ . Now, let  $t'' = w(z_1).\bar{y}(z'_1) \dots w(z_n).\bar{y}(z'_n)$  for some  $n > \text{len}(t') + l$ . Now, for  $w$  fresh,  $\bar{x}(w).t'.t'' \in T(\bar{x}y.s', \rho)$ , because  $s' \in s'[w/y]$ ,  $s'.t'' \in F(s', w, y)$  and  $t'.t'' \in T(s'.t'', \rho \cup \{w\})$ . Then, since  $R \preceq T(s, \rho)$ , there exists  $r \in R$  such that  $r \preceq \bar{x}(w).t'.t''$ . It is easy to see by inspecting *L1-L4* that  $r$  can only start with an output action, that is  $r = (\hat{z})\bar{x}z.r_1$  for some  $r_1$ . By Lemma 19,  $r_1 \preceq (t'.t'')\{z/w\}$ . Furthermore,  $(t'.t'')\{z/w\} = t'.(t''\{z/w\})$  because  $w$  does not occur in  $t'$ .

Since the number of outputs in  $t''$  is greater than  $\text{len}(r_1)$ , some of them have to be dropped, which is only possible by an application of *L3*. Further, since the number of these outputs is also greater than  $\text{len}(t') + \text{len}(r_1)$ , we conclude that some of these applications of *L3* must involve the inputs in  $t''$ . But, such annihilation are possible only if  $z = y$ , which implies  $r = \bar{x}y.r_1$ . Furthermore, since  $r_1 \preceq t'.t''\{y/w\}$ , by Lemma 20 we have  $r_1 = r'.r''$  for some  $r' \preceq t'$ . Let  $R'$  be the prefix closure of the set of all traces  $r'$  thus obtained for each  $t' \in T(s', \rho)$ . By construction,  $R' \preceq T(s', \rho)$ . Furthermore, from  $r = \bar{x}y.r'.r''$  and prefix closure of  $R$ , we have  $\bar{x}y.R' \subseteq R$ . Finally, since the length of traces in  $R$  is bounded, so is the length of traces in  $R'$ .  $\square$

**Lemma 21 (Boreale et al. [3]).** *If  $P \xRightarrow{s}$  then  $P\{z/y\} \xRightarrow{s\{z/y\}}$ .*  $\square$

We say  $s_1.\bar{x}w.s_2\{w/y\} \preceq_{\{w/y\}} s_1.\bar{x}(y).s_2$ . If  $s_3$  is normal and  $s_1 \preceq_{\sigma_1} s_2 \preceq_{\sigma_2} s_3$ , then we say  $s_1 \preceq_{\sigma_1 \oplus \sigma_2} s_3$ , where

$$\sigma_1 \oplus \sigma_2 = \begin{cases} \sigma_1(x) & \text{if } \sigma_1(x) \neq x \\ \sigma_2(x) & \text{if } \sigma_2(x) \neq x \\ x & \text{otherwise} \end{cases}$$

Note that, normality of  $s_3$  implies that  $\sigma_1 \oplus \sigma_2$  is well-defined. The reader may check the following simple lemma.

**Lemma 22.** *If  $s \preceq_{\sigma} r$  then  $\text{len}(s) = \text{len}(r)$ . Further, if  $s = s_1.s_2$ ,  $r$  is normal and  $r = r_1.r_2$  with  $\text{len}(r_1) = \text{len}(s_1)$ , then there exist  $\sigma_1, \sigma_2$  such that  $s_1 \preceq_{\sigma_1} r_1$ ,  $s_2 \preceq_{\sigma_2} r_2\sigma_1$ , and  $\sigma = \sigma_1 \oplus \sigma_2$ .*

**Proof:** By induction on the length of derivation of  $s \preceq_{\sigma} r$ .  $\square$

**Lemma 23.** *For a finite process  $P$  (with no replication), if  $P \xRightarrow{s.(\hat{z})yz} Q$  and  $\hat{z} \cap \{w\} = \emptyset$ , then there is  $P' \in P[w/y]_i$  such that  $P' \xRightarrow{s.(\hat{z})wz} Q$ .*

**Proof:** By induction on the derivation of  $P \xRightarrow{s.(\hat{z})yz} Q$ .  $\square$

We define random substitution on substitutions as follows

$$\sigma[w/y] = \{\sigma[u \mapsto v] \mid v \in \sigma(u)[w/y]\}$$

**Lemma 24.** *For clarity, in the following, we write  $P[w/y]_o$  for random output substitution on processes instead of  $P[w/y]$  (as in definition 5). Let  $y \in \rho$ ,  $s$  is  $\rho$ -well-formed,  $t \in T(s, \rho)$ ,  $s \preceq_\sigma t$ ,  $P$  be a finite process, and  $P \xRightarrow{s} Q$ . Then for every  $P_1 \in P[w/y]_o$  there is  $P' \in P_1[w/y]_i$ ,  $Q' \in Q[w/y]_o$ ,  $\sigma' \in \sigma[w/y]$ , such that  $P' \xRightarrow{s'} Q'$  and  $s' \preceq_{\sigma'} t$ .*

**Proof:** Without loss of generality we may assume that  $s$  and  $t$  are  $\rho \cup \{w\}$ -normal. Let  $P_1 \in P[w/y]_o$ . The proof is by induction on the length of computation  $P \xRightarrow{s} Q$ . The base case is obvious. For the induction step, let

$$P \xRightarrow{s_1} Q_1 \xrightarrow{\alpha} Q$$

There are two cases depending on  $\alpha$ .

- $\alpha \neq \tau$ : Since  $s_1.\alpha \preceq_\sigma t$ , by Lemma 22, we have  $t = t_1.\theta$ , and for some  $\sigma_1, \sigma_2$  such that  $\sigma = \sigma_1 \oplus \sigma_2$ ,  $s_1 \preceq_{\sigma_1} t_1$  and  $\alpha \preceq_{\sigma_2} \theta\sigma_1$ . By induction hypothesis there exist  $P' \in P_1[w/y]_i$ ,  $Q'_1 \in Q_1[w/y]_o$ ,  $\sigma'_1 \in \sigma_1[w/y]$ , such that  $P' \xRightarrow{s'_1} Q'_1$  such that  $s'_1 \preceq_{\sigma'_1} t_1$ . There are two subcases.
  - $\alpha = (\hat{v})\bar{u}v$ : Since  $s$  is  $\rho$ -well-formed and  $y \in \rho$  we deduce  $u \neq y$ . We only consider the case where  $\alpha = \bar{u}y$ . The case where  $v \neq y$  is simpler. Since  $s$  is  $\rho$ -well-formed, by Lemma 7, so is  $t$ . The subject of outputs in a  $\rho$ -well-formed template are not bound by previous bound outputs. Therefore,  $\theta\sigma_1 = \theta$ . Therefore, since  $\bar{u}y \preceq_{\sigma_2} \theta\sigma_1$ , we deduce  $\theta = \bar{u}(v_1)$  for some  $v_1$ , and  $\sigma_2 = \{y/v_1\}$ . Since  $Q_1 \xrightarrow{\bar{u}y}$ , by Lemma 16.2 we have  $Q'_1 \xrightarrow{\bar{u}y'} Q'$  for some  $Q' \in Q[w/y]_o$  and  $y' \in y[w/y]$ . Let  $\alpha' = \bar{u}y'$ ,  $\sigma'_2 = \{y'/v_1\}$ , and  $\sigma' = \sigma'_1 \oplus \sigma'_2$ . For the same reason as for  $\theta\sigma_1 = \theta$ , we have  $\theta\sigma'_1 = \theta$ . Then we have  $\alpha' \preceq_{\sigma'_2} \theta\sigma'_1$ . Then since  $s_1 \preceq_{\sigma'_1} t_1$ ,  $s'_1.\alpha' \preceq_{\sigma'} t_1.\theta$ . Further  $\sigma' \in \sigma[w/y]$ . Now the lemma holds with  $P' \xRightarrow{s'_1.\alpha'} Q'$ .
  - $\alpha = (\hat{v})uv$ : We only consider the cases where  $\alpha$  is  $uy$  and  $(\hat{v})yv$ ,  $y \notin \{u, v\}$ . The case where  $\alpha$  is  $yy$  is similar to these two, and the case where  $\alpha$  is  $uv$  for  $y \neq u, v$  is simple. Note that we have  $\sigma_2$  is identity, and therefore  $\sigma = \sigma_1$  and  $\alpha = \theta\sigma_1$ .
    - \*  $\alpha = uy$ : From  $\alpha = \theta\sigma_1$  we deduce  $\theta = u_1v_1$ ,  $\sigma_1(u_1) = u$ , and  $\sigma_1(v_1) = y$ . Let  $y' = \sigma'_1(v_1)$ . From  $\sigma'_1 \in \sigma_1[w/y]$  it follows that  $y' \in \{y, w\}$ . From Lemmas 16.1 and 17 it follows  $Q'_1 \xrightarrow{uy'} Q'$  for some  $Q' \in Q[w/y]_o$ . Then we have  $s'_1.uy' \preceq_{\sigma'_1} t_1.\theta$ , because  $\theta\sigma'_1 = uy'$ . The lemma holds with  $P' \xRightarrow{s'_1.uy'} Q'$ .
    - \*  $\alpha = (\hat{v})yv$ : From  $\alpha = \theta\sigma_1$  we deduce  $\theta = (\hat{v}_1)u_1v_1$ ,  $\sigma_1(u_1) = y$ ,  $\sigma_1(v_1) = v$ . By Lemma 16.1 we have  $Q'_1 \xrightarrow{(\hat{v})yv} Q'$  for some  $Q' \in Q[w/y]_o$ . Since  $\sigma'_1 \in \sigma_1[w/y]$ , we have  $\sigma'_1(u_1) \in \{w, y\}$ . There are two cases. If  $\sigma'_1(u_1) = y$ : Then we have  $s'_1.(\hat{v})yv \preceq_{\sigma'_1} t_1.\theta$ , because  $\theta\sigma'_1 = (\hat{v})yv$ . The lemma holds with  $P' \xRightarrow{s'_1.(\hat{v})yv} Q'$ . On the other hand, if  $\sigma'_1(u_1) = w$ , then by Lemma 23, there is  $P'' \in P'[w/y]_i$  such

that  $P'' \xrightarrow{s'_1.(\hat{v})wv} Q'$  (note that since  $t$  is  $\rho \cup \{w\}$ -normal  $\{w\} \cap \hat{v} = \emptyset$ ). Note that  $P'' \in P_1[w/y]_i$ . Then we have  $s'_1.(\hat{v})wv \preceq_{\sigma'_1} t_1.\theta$ , because

$\theta\sigma'_1 = (\hat{v})wv$ . The lemma holds with  $P'' \xrightarrow{s'_1.(\hat{v})wv} Q'$ .

–  $\alpha = \tau$ : Then  $s_1 = s$ , and  $s_1 \preceq_\sigma t$ . Then by induction hypothesis, there exist  $P' \in P_1[w/y]_i$ ,  $Q'_1 \in Q_1[w/y]_o$ ,  $\sigma' \in \sigma[w/y]$ , such that  $P' \xrightarrow{s'} Q'_1$  for  $s' \preceq_{\sigma'} t$ . From  $Q_1 \xrightarrow{\tau} Q$ , by Lemma 16.3, we have two cases:

- There is  $Q' \in Q[w/y]_o$  such that  $Q'_1 \xrightarrow{\tau} Q'$ . The lemma follows trivially with  $P' \xrightarrow{s'} Q'$ .
- We have  $Q \equiv (\nu \hat{z})R$ ,  $w, y \notin \hat{z}$ , and there is  $R' \in R[w/y]_o$  such that  $Q'_1 \xrightarrow{(\hat{z})\overline{wz}} R'$ . Then applying Lemma 23 to  $P' \xrightarrow{s'.(\hat{z})\overline{wz}.yz} R'$  we have, there is  $P'' \in P'[w/y]_i$  such that  $P'' \xrightarrow{s'.(\hat{z})\overline{wz}.wz} R'$ . But then  $P'' \xrightarrow{s'} (\nu \hat{z})R'$ . Now, since  $w, y \notin \hat{z}$ , we have  $(\nu \hat{z})R' \in Q[w/y]_o$ . The lemma holds with  $P'' \xrightarrow{s'} (\nu \hat{z})R'$ , because  $P'' \in P_1[w/y]_i$ .

□

We define  $\langle \hat{y} \rangle s$  as follows.

$$\langle \hat{y} \rangle s = \begin{cases} s & \text{if } \hat{y} = \emptyset \text{ or } y \notin fn(s) \\ s_1.\overline{x}(y).s_2 & \text{if } \hat{y} = \{y\} \text{ and there are } s_1, s_2, x \text{ s.t.} \\ & s = s_1.\overline{x}y.s_2 \text{ and } y \notin fn(s_1) \cup \{x\} \\ \perp & \text{otherwise} \end{cases}$$

**Proof of Lemma 10:**

1. First we prove  $P \sqsubseteq_\rho Q$  implies  $(\nu x)P \sqsubseteq_{\rho - \{x\}} (\nu x)Q$ . Suppose for an observer  $O$  such that  $rcp(O) \cap (\rho - \{x\}) = \emptyset$ , we have  $(\nu x)P|O \xrightarrow{\overline{\mu}\mu}$ . Let  $z$  be fresh. Using Lemma 21, we have  $((\nu x)P|O)\{z/x\} \xrightarrow{\overline{\mu}\mu}$ . Since  $x$  is not free in  $(\nu x)P$  we have  $(\nu x)P|O\{z/x\} \xrightarrow{\overline{\mu}\mu}$ . Now  $x$  is not free in  $O\{z/x\}$ , and so we have  $(\nu x)(P|O\{z/x\}) \xrightarrow{\overline{\mu}\mu}$ . This implies  $P|O\{z/x\} \xrightarrow{\overline{\mu}\mu}$ . But  $P \sqsubseteq_\rho Q$ , and  $rcp(O) \cap \rho = \emptyset$ . Therefore,  $Q|O\{z/x\} \xrightarrow{\overline{\mu}\mu}$ . It follows that  $(\nu x)(Q|O\{z/x\}) \xrightarrow{\overline{\mu}\mu}$ . And since  $x$  is not free in  $O\{z/x\}$ , we also have  $(\nu x)Q|O\{z/x\} \xrightarrow{\overline{\mu}\mu}$ . Since  $z$  is not free in  $O$ , we have  $O\{z/x\}\{x/z\} = O$ . Therefore, using Lemma 21 again, we deduce  $((\nu x)Q|O\{z/x\})\{x/z\} \xrightarrow{\overline{\mu}\mu}$ , i.e.  $(\nu x)Q|O \xrightarrow{\overline{\mu}\mu}$  since  $z$  is not free in  $(\nu x)Q$ .  
Now we prove  $rcp(R) \cap \rho = \emptyset$  and  $P \sqsubseteq_\rho Q$  imply  $P|R \sqsubseteq_\rho Q|R$ . Suppose for an observer  $O$  such that  $rcp(O) \cap \rho = \emptyset$ , we have  $(P|R)|O \xrightarrow{\overline{\mu}\mu}$ . Then  $P|(R|O) \xrightarrow{\overline{\mu}\mu}$ . Now,  $rcp(R|O) \cap \rho = \emptyset$ . Then since  $P \sqsubseteq_\rho Q$ , we have  $Q|(R|O) \xrightarrow{\overline{\mu}\mu}$ . This, in turn implies that  $(Q|R)|O \xrightarrow{\overline{\mu}\mu}$  and the lemma follows.
2. Let  $\overline{x}y|P \xrightarrow{s}$  where  $s$  is  $\rho$ -well-formed, and  $t \in T(s, \rho)$ . We have  $\overline{x}w|P \in (\overline{x}y|P)[w/y]$ . Then by Lemma 24, it follows that there is  $P' \in P[w/y]_i$  such

that  $\bar{x}w|P' \xRightarrow{s'}$  for some  $s' \preceq t$ . Now,  $(\nu w)(\bar{x}w|P') \xRightarrow{\langle\{w\}\rangle^{s'}}$ , and  $\langle\{w\}\rangle^{s'} \preceq t$ . Then it follows that  $(\nu w)(\bar{x}w|\sum_{P' \in P[w/y]_t} P') \xRightarrow{\langle\{w\}\rangle^{s'}}$ .  $\square$

**Lemma 25.** *Let  $y \notin \rho$  and  $y$  does not occur free as the subject of an input in  $s$ . Then, for every  $\rho$ -well-formed  $r$  such that  $e(s\{z/y\}) \xRightarrow{r}$  there is a  $\rho$ -well-formed cotemplate  $r'$  such that  $e(s) \xRightarrow{r'}$  and  $e(r'\{z/y\}) \xRightarrow{r}$ .*

**Proof:** Let  $e(s\{z/y\}) \xRightarrow{r}$  for a  $\rho$ -well-formed  $r$ . Without loss of generality we can assume  $bn(s) \cap \{z, y\} = \emptyset$ , and  $s$  is  $\rho$ -normal. The proof is by induction on the length of  $s$ . The base case  $s = \epsilon$  is obvious. Let  $s = \alpha.s_1$ , then  $s\{z/y\} = \alpha\{z/y\}.s_1\{z/y\}$ . For the induction step there are two cases depending on  $\alpha$ .

1.  $\alpha = (\hat{w})\bar{x}w$ : We only consider the case  $x = y$ , which is central to the proof; the case  $x \neq y$  is simpler. Then  $\alpha\{z/y\} = (\hat{w}')\bar{x}w'$  where  $w' = w\{z/y\}$ , and  $e(s\{z/y\}) = (\nu \hat{w}')(\bar{x}w'|e(s_1\{z/y\}))$ . We consider the case  $z \in \rho$  which is more interesting; the case  $z \notin \rho$  is similar. Since  $z \in \rho$  and  $r$  is  $\rho$ -well-formed, the message  $\bar{x}w'$  cannot fire in  $e(s\{z/y\}) \xRightarrow{r}$ . So there are two possibilities.
  - $\bar{x}w'$  is consumed internally. Then  $e(s_1\{z/y\}) \xRightarrow{r_1}$  for some  $r_1 = r_2.zw'.r_3$  such that  $r = \langle\hat{w}'\rangle(r_2.r_3)$ . Since  $r$  is  $\rho$ -well-formed, we have  $r_1$  is  $(\rho \cup \hat{w}')$ -well-formed. By induction hypothesis there is a  $(\rho \cup \hat{w}')$ -well-formed cotemplate  $r'_1$  such that  $e(s_1) \xRightarrow{r'_1}$  and  $e(r'_1\{z/y\}) \xRightarrow{r_1}$ . Now,  $e(s) \xRightarrow{r'}$  where  $r' = (\hat{w})\bar{y}w.r'_1$ . Note that  $r'$  is a cotemplate that is  $\rho$ -well-formed because  $y \notin \rho$ . Further,  $e(r'\{z/y\}) = (\nu \hat{w}')(\bar{x}w'|e(r'_1\{z/y\}))$ . Therefore,  $e(r'\{z/y\}) \xRightarrow{r}$ .
  - $\bar{x}w'$  is not consumed. Then  $e(s_1\{z/y\}) \xRightarrow{r_1}$  for some  $r_1$  such that  $r = \langle\hat{w}'\rangle r_1$ . Since  $r$  is  $\rho$ -well-formed, we have  $r_1$  is  $(\rho \cup \hat{w}')$ -well-formed. By induction hypothesis there is a  $(\rho \cup \hat{w}')$ -well-formed cotemplate  $r'_1$  such that  $e(s_1) \xRightarrow{r'_1}$  and  $e(r'_1\{z/y\}) \xRightarrow{r_1}$ . Now,  $e(s) \xRightarrow{r'}$  where  $r' = \langle\hat{w}\rangle r'_1$ . Note that  $r'$  is a  $\rho$ -well-formed cotemplate. Further,  $e(r'\{z/y\}) = (\nu \hat{w}')(e(r'_1\{z/y\}))$ . Therefore,  $e(r'\{z/y\}) \xRightarrow{r}$ .
2.  $\alpha = x(w)$ : We only consider the more interesting case where  $x = y$ . Then  $\alpha\{z/y\} = z(w)$  and  $e(s\{z/y\}) = z(w).e(s_1\{z/y\})$ . Then  $r = (\hat{v})zv.r_1$  for some  $r_1$  such that  $e(s_1\{z/y\}\{v/w\}) \xRightarrow{r_1}$ . Note that  $r_1$  is  $\rho$ -well-formed. Since  $s$  is cowell-formed, so is  $s\{z/y\}$ , and therefore  $w$  does not occur free as the subject of an input in  $s_1\{z/y\}$ . Further, since  $s$  is  $\rho$ -normal,  $w \notin \rho$ . Then, by induction hypothesis,  $e(s_1\{z/y\}) \xRightarrow{r_2}$  for some  $\rho$ -well-formed cotemplate  $r_2$  such that  $e(r_2\{v/w\}) \xRightarrow{r_1}$ . Now, applying the induction hypothesis again on  $s_1\{z/y\}$ , we get  $e(s_1) \xRightarrow{r_3}$  for some  $\rho$ -well-formed cotemplate  $r_3$  such that  $e(r_3\{z/y\}) \xRightarrow{r_2}$ . Now, the reader can verify the following claim: for cotemplates  $t_1, t_2, t_3$ , if  $e(t_1) \xRightarrow{t_2}$  and  $e(t_2) \xRightarrow{t_3}$  then  $e(t_1) \xRightarrow{t_3}$ . Using this claim and Lemma 21 we conclude  $e(r_3\{z/y\}\{v/w\}) \xRightarrow{r_1}$ . Now,  $e(s) \xRightarrow{r'}$

where  $r' = y(w).r_3$ . Since  $r_3$  is  $\rho$ -well-formed cotemplate, so is  $r'$ . Further,  $e(r'\{z/y\}) = z(w).e(r_3\{z/y\}) \xrightarrow{(\hat{y})zv} e(r_3\{z/y\}\{v/w\}) \xRightarrow{r_1}$ . Therefore,  $e(r'\{z/y\}) \xRightarrow{r}$ .  $\square$

**Lemma 26.** *If  $P \xRightarrow{s}$  and  $e(s) \xRightarrow{r}$  then  $P \xRightarrow{r}$ .*

**Proof:** Following is a proof sketch. Since we work modulo alpha equivalence on traces, we assume  $bn(r) \cap fn(P) = \emptyset$ . If  $s$  and  $s'$  are alpha equivalent then so are  $e(s)$  and  $e(s')$ . Then, since  $\alpha$  equivalent processes have the same transitions, we can assume  $s$  is normal. The proof is by induction on the length of the computation  $e(s) \xRightarrow{r}$ . The base case is trivial. For the induction step we can write

$$e(s) \xrightarrow{\alpha} Q \xRightarrow{r_1},$$

for some  $Q$  and  $r_1$ . We have three cases based on  $\alpha$ :

- $\alpha = (\hat{y})\bar{x}y$ : Then  $s = \langle \hat{y} \rangle(s_1.\bar{x}y.s_2)$  for some  $s_1, s_2$  where  $s_1$  does not contain any inputs, and  $Q = e(s_1.s_2)$ . Further, since  $P \xRightarrow{(\hat{y})(s_1.\bar{x}y.s_2)}$  we can show  $P \xRightarrow{(\hat{y})\bar{x}y} P_1 \xRightarrow{s_1.s_2}$ . Since  $e(s_1.s_2) \xRightarrow{r_1}$ , by induction hypothesis,  $P_1 \xRightarrow{r_1}$ . The lemma follows from  $P \xRightarrow{(\hat{y})\bar{x}y.r_1}$ .
- $\alpha = (\hat{y})xy$ : Then  $s = s_1.x(u).s_2$  for some  $s_1, s_2$  such that  $x \notin bn(s_1)$ , and  $s_1$  contains no inputs, and  $Q = e(s_1.s_2\{y/u\})$ . By normality of  $s$ ,  $u \notin n(s_1)$ , and hence we can write  $Q = e((s_1.s_2)\{y/u\})$ . Further, from  $P \xRightarrow{s_1.x(u).s_2}$ ,  $u \notin n(s_1)$ , and  $x \notin bn(s_1)$  we also have  $P \xRightarrow{x(u)} P_1 \xRightarrow{s_1.s_2}$ . Further, since  $\hat{y} \cap fn(P) = \emptyset$ ,  $P \xRightarrow{(\hat{y})xy} P_1\{y/u\}$ . Then by Lemma 21 we have  $P_1 \xRightarrow{(s_1.s_2)\{y/u\}}$ . By induction hypothesis  $P_1\{y/u\} \xRightarrow{r_1}$ , and the lemma follows from  $P \xRightarrow{(\hat{y})xy.r_1}$ .
- $\alpha = \tau$ : Then we have  $e(s) \xRightarrow{(\hat{y})\bar{x}y.xy} Q'$ ,  $s = \langle \hat{y} \rangle(s_1.\bar{x}y.s_2.x(u).s_3)$ ,  $s_1$  and  $s_2$  contain only outputs,  $Q' = e(s_1.s_2.s_3\{y/u\})$ , and  $Q = (\nu \hat{y})Q' = e(\langle \hat{y} \rangle(s_1.s_2.s_3\{y/u\}))$ . Since  $P \xRightarrow{(\hat{y})(s_1.\bar{x}y.s_2.x(u).s_3)}$ , we have  $P \xRightarrow{(\hat{y})(s_1.\bar{x}y.s_2.xy.s_3\{y/u\})}$  by Lemma 21. The complementary input and output actions can be preponed so that  $P \xRightarrow{(\hat{y})(s_1.s_2.s_3\{y/u\})} P_1$ . By induction hypothesis  $P_1 \xRightarrow{r_1}$ , and the lemma follows.  $\square$

**Lemma 27.** *Let  $r$  be  $\hat{y}$ -well-formed and  $\hat{y}$ -normal. Then  $(\nu \hat{y})e(r) = e(r')$ , where*

$$r' = \begin{cases} r_1 & \text{if } \hat{y} = \{y\}, r = r_1.y(z).r_2, \text{ and } y \notin n(r_1) \\ r_1.\bar{x}(y).r_2 & \text{if } \hat{y} = \{y\}, r = r_1.\bar{x}y.r_2, \text{ and } y \notin n(r_1) \\ r & \text{otherwise} \end{cases}$$

*Note that the conditions on  $r$  imply that the three cases above are exhaustive. Further,  $(\nu \hat{y})e(r) \xRightarrow{r'}$ .*

**Proof:** That  $(\nu\hat{y})e(r) \xrightarrow{r'}$  is immediate. The proof of  $(\nu\hat{y})e(r) = e(r')$  is by a straightforward induction on the length of  $r$ . The idea is to push  $(\nu\hat{y})$  inwards as far as possible. To push across a restriction, we can use *I1*, *A3*, *A8* and *A19*, and the fact that  $(\nu x)0 = 0$  which can be derived using *A2*, *A14*, *A19*. To push across a message we can use *A8*, and to push across an input we can use *A11*. If at any point,  $(\nu\hat{y})$  cannot be pushed further, either case 2 of the definition of  $r'$  applies, or *A11* can be used and case 1 applies. If  $(\nu\hat{y})$  can be pushed all the way in, we can use *A3* and  $(\nu x)0 = 0$ , and case 3 applies.  $\square$

**Proof of Lemma 13:** For convenience, we write  $R(s, \rho)$  to denote the set of all  $\rho$ -well-formed cowell-formed cotemplate traces  $r$  such that  $e(s) \xrightarrow{r}$ . The lemma can be stated as: for every set of names  $\rho$ ,  $e(s) \sqsubseteq_\rho \sum_{r \in R(s, \rho)} e(r)$ . We will be using the following property in the proof, which the reader can verify easily. If  $e(s) \xrightarrow{r}$  then  $\text{len}(r) \leq \text{len}(s)$ .

Without loss of generality, we can assume  $s$  is  $\rho$ -normal. The proof is by induction on the length of  $s$ . For the base case,  $s = \epsilon$ , we have  $e(\epsilon) = 0$ ,  $R(\epsilon, \rho) = \{\epsilon\}$ , and the lemma follows using *A14* and *I4*. For the induction step we have three cases:

1.  $s = x(y).s_1$ : By induction hypothesis we have  $e(s_1) \sqsubseteq_\rho \sum_{r' \in R(s_1, \rho)} e(r')$ . Clearly, for every  $r_1 \in R(s_1, \rho)$ ,  $\text{fn}(e(r_1)) = \text{fn}(r_1) \subset \text{fn}(s_1) = \text{fn}(e(s_1))$ . Therefore, we are done if we show that for all  $z \in \text{fn}(s_1)$ ,  $e(s_1)\{z/y\} \sqsubseteq_\rho \sum_{r' \in R(s_1, \rho)} e(r')\{z/y\}$ , for then using the fact that  $x(y).R(s_1, \rho) \subset R(s, \rho)$ , and laws *I2*, *A2*, we conclude  $e(s) \sqsubseteq_\rho \sum_{r \in R(s, \rho)} e(r)$ .  
Now,  $e(s_1)\{z/y\} = e(s_1\{z/y\})$ . By induction hypothesis,  $e(s_1\{z/y\}) \sqsubseteq_\rho \sum_{r' \in R(s_1\{z/y\}, \rho)} e(r')$ . Since  $s$  is cowell-formed,  $y$  does not occur free as the subject of an input in  $s_1$ , and since  $s$  is  $\rho$ -normal,  $y \notin \rho$ . Then, using Lemmas 25 and 12, and laws *I3*, *A1*, and *A2*, we conclude  $\sum_{r' \in R(s_1\{z/y\}, \rho)} e(r') \sqsubseteq_\rho \sum_{r' \in R(s_1, \rho)} e(r')\{z/y\}$ . By transitivity of  $\sqsubseteq_\rho$ , we have  $e(s_1\{z/y\}) \sqsubseteq_\rho \sum_{r' \in R(s_1, \rho)} e(r')\{z/y\}$ .
2.  $s = (\hat{y})\bar{x}y.s_1$ : By induction hypothesis we have  $e(s_1) \sqsubseteq_{\rho \cup \hat{y}} \sum_{r' \in R(s_1, \rho \cup \hat{y})} e(r')$ . From  $e(s) = (\nu\hat{y})(\bar{x}y|e(s_1))$  and using *I1*, *A12*, and *A7*, we have  $e(s) \sqsubseteq_\rho \sum_{r' \in R(s_1, \rho \cup \hat{y})} (\nu\hat{y})(\bar{x}y|e(r')) = \sum_{r' \in R(s_1, \rho \cup \hat{y})} e((\hat{y})\bar{x}y.r')$ . If  $x \notin \rho$ , then  $(\hat{y})\bar{x}y.R(s_1, \rho \cup \hat{y}) \subset R(s, \rho)$ , and therefore using *A2* we have  $e(s) \sqsubseteq_\rho \sum_{r \in R(s, \rho)} e(r)$  as required. For the case  $x \in \rho$ , we are done if we show for every  $(\hat{y})\bar{x}y.r_1 \in (\hat{y})\bar{x}y.R(s_1, \rho \cup \hat{y})$  that  $e((\hat{y})\bar{x}y.r_1) \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\bar{x}y.r_1, \rho)} e(r')$ . Following is the reason. If  $e(s_1) \xrightarrow{r_1}$ , we have  $e(s) \xrightarrow{(\hat{y})\bar{x}y.r_1}$ . Then by Lemma 26, we have  $R((\hat{y})\bar{x}y.r_1, \rho) \subset R(s, \rho)$ . Then using *I3*, *A1*, and *A2*, we conclude  $e(s) \sqsubseteq_\rho \sum_{r \in R(s, \rho)} e(r)$ . To show  $e((\hat{y})\bar{x}y.r_1) \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\bar{x}y.r_1, \rho)} e(r')$ , we have two cases based on  $r_1$ . Without loss of generality, we can assume  $r_1$  is  $\rho \cup \hat{y}$ -normal.
  - $r_1 = \epsilon$ . Then  $(\hat{y})\bar{x}y.r_1 = (\hat{y})\bar{x}y$ ,  $e((\hat{y})\bar{x}y) = (\nu\hat{y})\bar{x}y$ . Since  $x \in \rho$ ,  $R((\hat{y})\bar{x}y, \rho) = \{\epsilon\}$ , and  $e(\epsilon) = 0$ . The result follows because  $(\nu\hat{y})(\bar{x}y) \sqsubseteq_\rho \sum 0$ , which can be derived using the example in Section 4 and laws *A14*, *A19*, *I1*, *I4*.

- $r_1 = u(w).r_2$ : We only consider the case  $u = x$ , the other is simpler. Using *A20*, *I1*, *A7*, *A11*, we deduce

$$\begin{aligned} e((\hat{y})\overline{xy}.r_1) &= (\nu\hat{y})(\overline{xy}|x(w).e(r_2)) \sqsubseteq_\rho \sum x(w).e((\hat{y})\overline{xy}.r_2) \\ &\quad + \sum x(w).(\nu\hat{y})e(r_2) \\ &\quad + \sum (\nu\hat{y})e(r_2\{y/w\}) \end{aligned} \quad (1)$$

We are done if we show that for each summand  $Q$  in the RHS,  $Q \sqsubseteq_\rho \sum_{r' \in T} e(r')$ , for some set  $T$  of  $\rho$ -well-formed traces that  $Q$  exhibits. This is because, it is clear from (1) that if  $Q \xrightarrow{r'}$  then  $e((\hat{y})\overline{xy}.r_1) \xrightarrow{r'}$ . Therefore  $T \subset R((\hat{y})\overline{xy}.r_1, \rho)$ , and by using *I3*, *A1*, and *A2*, we can conclude

$$e((\hat{y})\overline{xy}.r_1) \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\overline{xy}.r_1, \rho)} e(r')$$

Now, we consider each summand separately.

- (a)  $x(w).e((\hat{y})\overline{xy}.r_2)$ : Since  $e(s_1) \xrightarrow{r_1}$ , we have  $\text{len}(r_1) \leq \text{len}(s_1)$ . And since  $\text{len}(r_2) < \text{len}(r_1) \leq \text{len}(s_1) < \text{len}(s)$ , we have  $\text{len}((\hat{y})\overline{xy}.r_2) < \text{len}(s)$ . Therefore, we can apply the induction hypothesis to conclude

$$e((\hat{y})\overline{xy}.r_2) \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\overline{xy}.r_2, \rho)} e(r')$$

Now, for every  $r' \in R((\hat{y})\overline{xy}.r_2, \rho)$ ,  $\text{fn}(e(r')) = \text{fn}(r') \subseteq \text{fn}(e((\hat{y})\overline{xy}.r_2))$ . Now, since  $r_1$  is cowell-formed,  $w$  does not occur free as the subject of an input in  $(\hat{y})\overline{xy}.r_2$ . Further, since  $r_1$  is  $\rho \cup \hat{y}$ -normal, and hence  $w \notin \rho$ . Then using arguments similar to that in case 1, we deduce that for each  $z \in \text{fn}(e((\hat{y})\overline{xy}.r_2))$ ,

$$e((\hat{y})\overline{xy}.r_2)\{z/w\} \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\overline{xy}.r_2, \rho)} e(r')\{z/w\}$$

Then using *I1* and *A13* we get

$$x(w).e((\hat{y})\overline{xy}.r_2) \sqsubseteq_\rho \sum_{r' \in R((\hat{y})\overline{xy}.r_2, \rho)} x(w).e(r')$$

Now, because  $x(w).R((\hat{y})\overline{xy}.r_2, \rho) \subset R(x(w).(\hat{y})\overline{xy}.r_2, \rho)$ , using *A2*, we have

$$x(w).e((\hat{y})\overline{xy}.r_2) \sqsubseteq_\rho \sum_{r' \in R(x(w).(\hat{y})\overline{xy}.r_2, \rho)} e(r')$$

- (b)  $x(w).(\nu\hat{y})e(r_2)$ : Since  $r_1$  is  $\rho \cup \hat{y}$ -well-formed, it also  $\hat{y}$ -well-formed. Then by Lemma 27,  $(\nu\hat{y})e(r_2) = e(r')$ , where  $r'$  is as defined in the



lemma and  $(\nu\hat{y})e(r_2) \xrightarrow{r'}$ . The reader can check that  $r'$  is  $\rho$ -well-formed. Now,  $fn(e(r')) \subset fn((\nu\hat{y})e(r_2))$ . The reader can also check that, for  $z \in fn((\nu\hat{y})e(r_2))$ , by Lemma 27,  $((\nu\hat{y})e(r_2))\{z/w\} = e(r')\{z/w\}$ . Then by *I2*, *I4* and *A14*,

$$x(w).(\nu\hat{y})e(r_2) =_{\rho} e(x(w).r') =_{\rho} \sum e(x(w).r')$$

Now,  $x(w).r'$  is a  $\rho$ -well-formed trace, and since  $(\nu\hat{y})e(r_2) \xrightarrow{r'}$ , we have

- (c)  $(\nu\hat{y})e(r_2\{y/w\})$ : Since  $len(r_2) < len(s)$ , by induction hypothesis and axiom *I1* we have

$$(\nu\hat{y})e(r_2\{y/w\}) \sqsubseteq_{\rho} \sum_{r'' \in R(r_2\{y/w\}, \rho \cup \hat{y})} (\nu\hat{y})e(r'')$$

Let  $r'' \in R(r_2\{y/w\}, \rho \cup \hat{y})$ . We have,  $r''$  is also  $\hat{y}$ -well-formed. Then by Lemma 27,  $(\nu\hat{y})e(r'') = e(r')$ , where  $r'$  is as defined in the lemma and  $(\nu\hat{y})e(r'') \xrightarrow{r'}$ . The reader can check  $r'$  is  $\rho$ -well-formed. Further, since  $e(r_2\{y/w\}) \xrightarrow{r''}$ , using Lemma 26 we can show  $(\nu\hat{y})e(r_2\{y/w\}) \xrightarrow{r'}$ . Let  $R$  be the set of all  $r'$  that are obtained for each  $r'' \in R(r_2\{y/w\}, \rho \cup \hat{y})$ . Then, using *I3*, *I4*, *A1*, *A14* and transitivity of  $\sqsubseteq_{\rho}$ , we conclude

$$(\nu\hat{y})e(r_2\{y/w\}) \sqsubseteq_{\rho} \sum_{r' \in R} e(r')$$

–  $r_1 = (\hat{v})\overline{u}v.r_2$ : Using axioms *A4*, *A5*, and *A8*, we deduce

$$e((\hat{y})\overline{x}y.r_1) = e((\hat{y})\overline{x}y.(\hat{v})\overline{u}v.r_2) = e((\hat{v}')\overline{u}v.(\hat{y}')\overline{x}y.r_2) \quad (2)$$

where (since  $r_1$  is  $\rho \cup \hat{y}$ -normal)

$$\hat{v}' = \begin{cases} \hat{y} & \text{if } \hat{y} = \{v\} \\ \hat{v} & \text{otherwise} \end{cases} \quad \text{and} \quad \hat{y}' = \hat{y} - \hat{v}'$$

By induction hypothesis, we have

$$e((\hat{y}')\overline{x}y.r_2) \sqsubseteq_{\rho \cup \hat{v}'} \sum_{r' \in R((\hat{y}')\overline{x}y.r_2, \rho \cup \hat{v}')} e(r')$$

Then, using *I1*, *A7* and *A12* we deduce

$$e((\hat{v}')\overline{u}v.(\hat{y}')\overline{x}y.r_2) \sqsubseteq_{\rho} \sum_{r' \in R((\hat{y}')\overline{x}y.r_2, \rho \cup \hat{v}')} e((\hat{v}')\overline{u}v.r') \quad (3)$$

Now,  $u \notin \rho \cup \hat{y}$ , because  $r_1 \in R(s_1, \rho \cup \hat{y})$ . Then for every  $r' \in R((\hat{y}')\bar{x}y.r_2, \rho \cup \hat{v}')$ ,  $(\hat{v}')\bar{u}v.r'$  is  $\rho$ -well-formed. Further,  $e((\hat{y}')\bar{x}y.r_2) \xrightarrow{r'}$  implies  $e((\hat{v}')\bar{u}v.(\hat{y}')\bar{x}y.r_2) \xrightarrow{(\hat{v}')\bar{u}v.r'}$ . Therefore,

$$(\hat{v}')\bar{u}v.R(\hat{y}'\bar{x}y.r_2, \rho \cup \hat{v}') \subset R((\hat{v}')\bar{u}v.(\hat{y}')\bar{x}y.r_2, \rho) = R((\hat{y}')\bar{x}y.(\hat{v}')\bar{u}v.r_2, \rho)$$

Finally, from (2), (3) and (4), and using  $I_4$ ,  $A_2$  we obtain

$$e((\hat{y}')\bar{x}y.r_1) \sqsubseteq_\rho \sum_{r' \in R((\hat{y}')\bar{x}y.r_1, \rho)} e(r')$$

□

**Lemma 28.** *Let  $y \notin \rho$ , and  $s$  be a trace such that  $y$  does not occur free in input actions of  $s$ . Then for every  $t' \in T(s\{z/y\}, \rho)$  there is a  $t \in T(s, \rho)$  such that  $t\{z/y\} \preceq t'$  using only  $L_4$ .*

**Proof:** The proof is by induction on the length of  $s$ . Without loss of generality, we may assume  $bn(s) \cap \{z, y\} = \emptyset$ . The base case  $s = \epsilon$  is obvious. For the induction step, there are three cases:

1.  $s = (\hat{v})uv.s_1$ : For  $t' \in T(s\{z/y\}, \rho)$ , we have  $t' = (\hat{v}')u'v'.t'_1$ , where  $u' = u\{z/y\}$ ,  $v' = v\{z/y\}$  and  $t'_1 \in T(s_1\{z/y\}, \rho)$ . By induction hypothesis, there is  $t_1 \in T(s_1, \rho)$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L_4$ . But we have  $(\hat{v})uv.t_1 \in T((\hat{v})uv.s_1, \rho)$ , and  $((\hat{v})uv.t_1)\{z/y\} = (\hat{v}')u'v'.t_1\{z/y\} \preceq (\hat{v}')u'v'.t'_1 = t'$ , using only  $L_4$ .
2.  $s = \bar{u}(v).s_1$ : For  $t' \in T(s\{z/y\}, \rho)$  we have  $t' = \bar{u}'(v).t'_1$ , where  $u' = u\{z/y\}$  and  $t'_1 \in T(s_1\{z/y\}, \rho \cup \{v\})$ . By induction hypothesis, there is  $t_1 \in T(s_1, \rho \cup \{v\})$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L_4$ . But we have  $\bar{u}(v).t_1 \in T(\bar{u}(v).s_1, \rho)$ , and  $(\bar{u}(v).t_1)\{z/y\} = \bar{u}'(v).t_1\{z/y\} \preceq \bar{u}'(v).t'_1 = t'$ , using only  $L_4$ .
3.  $s = \bar{u}v.s_1$ : There are two subcases.
  - $v = y$ : Then  $s\{z/y\} = \bar{u}'z.s_1\{z/y\}$  where  $u' = u\{z/y\}$ . There are two more subcases:
    - $z \in \rho$ : For  $t' \in T(s\{z/y\}, \rho)$  we have  $t' = \bar{u}'(w).t'_1$ , where  $w$  fresh, and  $t'_1 \in T(s_2, \rho \cup \{w\})$  for some  $s_2 \in s_1\{z/y\}[w/z]$ . Using the fact that  $y$  cannot occur free in the input actions of  $s$ , we can show  $s_2 = s_3\{z/y\}$  for some  $s_3 \in s_1[w/z]$ . Clearly,  $s_3$  does not contain free occurrences of  $y$  in input actions. Then by induction hypothesis, there is  $t_1 \in T(s_3, \rho \cup \{w\})$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L_4$ . It is easy to see that, since  $w$  is fresh,  $t_1\{z/w\} \in T(s_1, \rho)$ . Then we have  $\bar{u}y.t_1\{z/w\} \in T(s, \rho)$ . Then  $(\bar{u}y.t_1\{z/w\})\{z/y\} = \bar{u}'z.t_1\{z/w\}\{z/y\} = \bar{u}'z.t_1\{z/y\}\{z/w\} \prec \bar{u}'(w).t_1\{z/y\} \preceq \bar{u}'(w).t'_1 = t'$ , where the relation  $\prec$  is by  $L_4$ .
    - $z \notin \rho$ : For  $t' \in T(s\{z/y\}, \rho)$  we have  $t' = \bar{u}'z.t'_1$ , where  $t'_1 \in T(s_1\{z/y\}, \rho)$ . By induction hypothesis, there is  $t_1 \in T(s_1, \rho)$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L_4$ . But we have  $\bar{u}y.t_1 \in T(\bar{u}y.s_1, \rho)$ , and  $(\bar{u}y.t_1)\{z/y\} = \bar{u}'z.t_1\{z/y\} \preceq \bar{u}'z.t'_1 = t'$  using only  $L_4$ .

- $v \neq y$ : Then  $s\{z/y\} = \overline{u'}v.s_1\{z/y\}$  where  $u' = u\{z/y\}$ . There are two subcases.
  - $v \notin \rho$ : Then for  $t' \in T(s\{z/y\}, \rho)$  we have  $t' = \overline{u'}v.t'_1$ , where  $t'_1 \in T(s_1\{z/y\}, \rho)$ . By induction hypothesis, there is  $t_1 \in T(s_1, \rho)$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L4$ . But we have  $\overline{u}v.t_1 \in T(s, \rho)$ , and  $(\overline{u}v.t_1)\{z/y\} = \overline{u'}v.t_1\{z/y\} \preceq \overline{u'}v.t'_1 = t'$  using only  $L4$ .
  - $v \in \rho$ : Then for  $t' \in T(s\{z/y\}, \rho)$  we have  $t' = \overline{u'}(w).t'_1$ , where  $w$  fresh,  $t'_1 \in T(s_2, \rho \cup \{w\})$  for some  $s_2 \in s_1\{z/y\}[w/v]$ . Using the fact that  $y$  occurs free only in output actions of  $s_1$ , we can show  $s_2 = s_3\{z/y\}$  for some  $s_3 \in s_1[w/v]$ . Clearly,  $y$  does not occur free in input actions of  $s_3$ . Then by induction hypothesis, there is  $t_1 \in T(s_3, \rho \cup \{w\})$  such that  $t_1\{z/y\} \preceq t'_1$  using only  $L4$ . But we have  $\overline{u}(w).t_1 \in T(s, \rho)$ ,  $(\overline{u}(w).t_1)\{z/y\} = \overline{u'}(w).t_1\{z/y\} \preceq \overline{u'}(w).t'_1 = t'$ , using only  $L4$ .  $\square$

**Proof of Lemma 14:** The proof is by induction on the length of  $s$ . Without loss of generality we may assume  $s$  is  $\rho$ -normal. The base case follows from  $0 \sqsubseteq \sum_{t \in \{\epsilon\}} 0$  which holds by *A14*. For the induction step we have three cases:

1.  $s = \overline{x}y.s'$ : Then  $e(s) = \overline{x}y|e(s')$ . There are two subcases:
  - (a)  $y \notin \rho$ : From induction hypothesis we have

$$e(s') \sqsubseteq_{\rho} \sum_{t' \in T(s', \rho)} e(t')$$

Using laws *I1* and *A12* in that order we get

$$\overline{x}y|e(s') \sqsubseteq_{\rho} \overline{x}y| \sum_{t' \in T(s', \rho)} e(t') =_{\rho} \sum_{t' \in T(s', \rho)} \overline{x}y|e(t') = \sum_{t' \in T(s', \rho)} e(\overline{x}y.t') = \sum_{t \in T(s, \rho)} e(t)$$

- (b)  $y \in \rho$ : It is easy to check that for  $w$  fresh, every  $s'' \in s'[w/y]$  is a cowell-formed cotemplate. Further,  $\text{len}(s'') = \text{len}(s')$ . Hence by induction hypothesis, we have

$$e(s'') \sqsubseteq_{\rho} \sum_{t' \in T(s'', \rho)} e(t')$$

Then, by *I3*

$$\sum_{s'' \in s'[w/y]} e(s'') \sqsubseteq_{\rho} \sum_{t' \in T(s'[w/y], \rho)} e(t') \tag{5}$$

Now, since  $y \in \rho$ , by law *A21* we have

$$e(s) = \overline{x}y|e(s') \sqsubseteq_{\rho} (\nu w)(\overline{x}w| \sum_{P \in e(s')[w/y]_i} P)$$

The reader may check that for any  $P \in e(s')[w/y]_i$ ,  $P = e(s'')$  for some  $s'' \in s'[w/y]$  and vice versa. Using this we get

$$e(s) \sqsubseteq_\rho (\nu w)(\bar{x}w \mid \sum_{s'' \in s'[w/y]} e(s''))$$

Now using 5, laws *I1* and *I3*, we have

$$e(s) \sqsubseteq_\rho (\nu w)(\bar{x}w \mid \sum_{t' \in T(s'[w/y], \rho)} e(t'))$$

Now, using laws *A7* and *A12*, we get

$$e(s) \sqsubseteq_\rho \sum_{t' \in T(s'[w/y], \rho)} e(\bar{x}(w).t') = \sum_{t \in T(s, \rho)} e(t)$$

2.  $s = \bar{x}(y).s'$ : Then  $e(s) = (\nu y)(\bar{x}y \mid e(s'))$ . By induction hypothesis, we have

$$e(s') \sqsubseteq_{\rho \cup \{y\}} \sum_{t' \in T(s', \rho \cup \{y\})} e(t')$$

Since  $s$  is  $\rho$ -normal, we have  $y \notin \rho$ . Using this and law *I1*, we get

$$e(s) \sqsubseteq_\rho (\nu y)(\bar{x}y \mid \sum_{t' \in T(s', \rho \cup \{y\})} e(t'))$$

Now, using laws *A7* and *A12*, we get

$$e(s) \sqsubseteq_\rho \sum_{t' \in T(s', \rho \cup \{y\})} e(\bar{x}(y).t') = \sum_{t \in T(s, \rho)} e(t)$$

3.  $s = x(y).s'$ : Then  $e(s) = x(y).e(s')$ . We are done if we show

$$e(s')\{z/y\} \sqsubseteq_\rho \left( \sum_{t' \in T(s', \rho)} e(t') \right) \{z/y\} \quad (6)$$

for every  $z \in \text{fn}(e(s'))$ , because then by laws *I2*, *A13* and *I4* we have

$$e(s) \sqsubseteq_\rho x(y). \sum_{t' \in T(s', \rho)} e(t') =_\rho \sum_{t' \in T(s', \rho)} x(y).e(t') = \sum_{t' \in T(s', \rho)} e(x(y).t') = \sum_{t \in T(s, \rho)} e(t)$$

Now we prove 6. By applying the induction hypothesis to  $s'\{z/y\}$ , we have

$$e(s')\{z/y\} = e(s'\{z/y\}) \sqsubseteq_\rho \sum_{t'' \in T(s'\{z/y\}, \rho)} e(t'')$$

Since

$$\left( \sum_{t' \in T(s', \rho)} e(t') \right) \{z/y\} = \sum_{t' \in T(s', \rho)} e(t')\{z/y\} = \sum_{t' \in T(s', \rho)} e(t'\{z/y\})$$

So, we are done if we show

$$\sum_{t'' \in T(s'\{z/y\}, \rho)} e(t'') \sqsubseteq_{\rho} \sum_{t' \in T(s', \rho)} e(t'\{z/y\})$$

Since  $s$  is  $\rho$ -normal,  $y \notin \rho$ . Further, since  $s$  is a cowell-formed cotemplate,  $y$  cannot occur free in the input actions of  $s'$ . Then by Lemma 28, for every  $t'' \in T(s'\{z/y\}, \rho)$  there is a  $t' \in T(s', \rho)$  such that  $t'\{z/y\} \preceq t''$ . Then by Lemma 12.2 we have  $e(t'') \sqsubseteq e(t'\{z/y\})$ , and 6 follows from laws *I3*, *I4*, *A1* and *A2*.  $\square$