A compositional trace semantics for Orc

Dimitrios Vardoulakis and Mitchell Wand

Northeastern University
dimvar@ccs.neu.edu wand@ccs.neu.edu

Abstract. Orc [9] is a language for task orchestration. It has a small set of primitives, but is sufficient to express many useful programs succinctly. We show that the operational and denotational semantics given in Kitchin et al. [9] do not agree, by giving counterexamples to their Theorems 2 and 3. We remedy this situation by providing new operational and denotational semantics with a better treatment of variable binding, and proving an adequacy theorem to relate them. Our semantics validates some useful equivalences between Orc processes; since the semantics is compositional these automatically become congruences. Last, we consider an alternative semantics that is insensitive to internal events.

1 Introduction

Orc [9] is a small concurrent programming language, designed with web services in mind. It has few primitives, but they suffice to express many popular concurrent programming patterns (see [9], [11]). Orc uses autonomous computing units called *sites* to perform sequential computation and other basic services. It then provides operators to orchestrate the execution of sites and build larger processes. Kitchin et al. [9] have developed operational and denotational semantics for Orc.

In this paper, we address some shortcomings of the existing Orc semantics.

- We show that the operational and denotational semantics of Orc do not agree, by giving counterexamples to the Theorems 2 and 3 of [9]. We develop a new operational and denotational semantics and prove an adequacy theorem to relate them.
- In [9], the authors impose a unique naming constraint, namely that "free and bound variables of an expression have different names." Our semantics do not require this limitation.
- We can prove all the equivalences in [9] between Orc processes. Since our semantics is compositional these equivalences become congruences.

We present our semantics first, and then discuss the problems with the semantics of [9].

2 Overview of Orc

We now give an informal description of the language before we present its formal syntax and semantics in the next section. The simplest Orc program is a

site call. For example, *Factorize*(N) will compute and send back the prime factors of its argument. *RedditFeed*$(today)$ will respond with today's tech news. In Orc terminology, we use the word *publication* to refer to the result of a site call. A site may respond to a call at most once and it can also ignore the request. Note that the same site call at different times may publish different values.

In *symmetric composition* $(f \mid g)$ the two processes are evaluated in parallel and there is no interaction between them. The composite process publishes all the values published by f and g. For instance, the process (*Factorize*(N) | *RedditFeed*$(today)$) can publish at most two values.

The *sequencing* operator $(f >x> g)$ is used to spawn threads. It first evaluates f, and whenever f publishes some value v, it binds v to x in g and launches a new instance of g in parallel. For example, ((*Factorize*(N) | *RedditFeed*$(today)$) $>x>$ *Print*(x)) may print twice, if both *Factorize*(N) and *RedditFeed*$(today)$ publish. If f does not publish, g is not run.

Last, we can use the **where** operator to terminate a process after it publishes. The expression $(f$ **where** $x :\in g)$ starts evaluating f and g in parallel. However, the parts of f that depend on x block until x acquires a value. If g publishes, the value published is bound to x in f and g is terminated. Therefore, the expression (*Print*(x) **where** $x :\in$ (*Factorize*(N) | *RedditFeed*$(today)$)) will either print the prime factors of N or today's tech–news, maybe none, but not both. Site calls are strict, which means that *Print*(x) by itself has no transitions. However, placed in a context that can provide a value for x (as in this example), *Print*(x) is no longer inert.[1]

The operators we saw up to now do not allow us to write recursive processes. To do that, we can define expressions like the following:

$$DOS(x) \triangleq Ping(x) \mid DOS(x)$$

This is a simple denial-of-service attack; the process $DOS(ip)$ pings ip an unbounded number of times.

At this point, we have explained the features of Orc informally and we can proceed to discuss its formal syntax and operational semantics.

3 Syntax – Operational Semantics

3.1 Syntax

The syntax of Orc is shown in Fig. 1. An Orc program consists of a finite set of mutually recursive declarations and an expression (i.e. process) which is evaluated with these declarations in scope. To avoid dynamic binding of variables, we require that a declaration $E_i(x) \triangleq e$ satisfy f.v.$(e) \subseteq \{x\}$. The process $\mathbf{0}$ is the inert process. The actual parameter of a site call or a call to a defined expression is either a variable or a value. We will not assign types to our values; all values belong to some generic set *Val*. Orc is not higher-order: a process is not a value. In what follows, we assume that processes are *well-formed*, i.e. do not contain $E_i(p)$ when there are fewer than i declarations in the program (see Fig. 1).

[1] It is this behavior that requires the existence of an environment Γ in the operational semantics (section 3).

Program	$P ::= D_1, \ldots, D_k$ **in** e	
Expression	$e ::= \mathbf{0} \mid M(p) \mid let(p) \mid E_i(p) \mid (e_1 \mid e_2) \mid e_1 > x > e_2 \mid e_1$ **where** $x :\in e_2$	
Parameter	$p ::= x \mid v$	
Declaration	$D_i ::= E_i(x) \triangleq e$	

Fig. 1. Syntax of Orc

3.2 Operational Semantics

Our version of the operational semantics of Orc (Fig. 2) uses labeled transitions. The metavariables f, g range over processes. Every transition is of the form

$$\Delta, \Gamma \vdash f \xrightarrow{a} f'$$

In this transition, process f takes a step to f' with event a, when the set of declarations is Δ and the environment for variables is Γ. Note that Δ and Γ remain unchanged during the evaluation of an expression. The events that occur during transitions are listed below:

Event ::=	$!v$	*publication*
|	τ	*internal*
|	$M_k(v)$	*site call*
|	$k?v$	*site response*
|	$[v/x]$	*receive*

Let's take a closer look at the rules. When process $M(v)$ calls site M with value v, a site call event occurs and a fresh handle k is allocated to identify the call (rule SITEC). The resulting process $?k$ is just an idle thread waiting for an answer to the call with handle k. It is a necessary addition to the syntax to represent intermediate state.

If the site replies with some value w, $?k$ performs a site response event $k?w$ and becomes $let(w)$, as shown in rule SITERET. *Let* is a process that responds with the same value it was called. By rule LET, $let(w)$ publishes w and becomes $\mathbf{0}$, which has no further transitions.

None of the above steps is guaranteed to happen; $M(v)$ may delay the site call to M indefinitely, if the call happens M may never respond, and if it responds the value may not be published.

Site calls are strict, so $M(x)$ will block until x acquires a value. In an environment that can supply value v to x, $M(x)$ performs a receive event and becomes $M(v)$. This is reflected by the rule SITEC-VAR. If x is not in Γ, $M(x)$ behaves like $\mathbf{0}$. Rules SITEC-VAR, LET-VAR and DEF-VAR reflect the potential transition of a process in a suitable environment. It is the environment that makes us able to distinguish between $M(x)$ and $\mathbf{0}$.

When we call a defined expression $E_i(v)$, v is substituted for x in the body of E_i, which is an internal event (rule DEF). The process continues as $[v/x]f_i$.

The two rules for symmetric composition are self explanatory; process $f \mid g$ takes a step if either f or g takes a step. The steps of the sub-processes can be interleaved arbitrarily.

$$\text{(SITEC)}\quad \frac{}{\Delta, \Gamma \vdash\ M(v)\ \xrightarrow{M_k(v)}\ ?k}\ k \text{ fresh}$$

$$\text{(SITEC-VAR)}\quad \frac{}{\Delta, \Gamma \vdash\ M(x)\ \xrightarrow{[v/x]}\ M(v)}\ \Gamma(x) = v$$

$$\text{(SITERET)}\quad \frac{}{\Delta, \Gamma \vdash ?k\ \xrightarrow{k?v}\ let(v)}$$

$$\text{(LET)}\quad \frac{}{\Delta, \Gamma \vdash\ let(v)\ \xrightarrow{!v}\ \mathbf{0}}$$

$$\text{(LET-VAR)}\quad \frac{}{\Delta, \Gamma \vdash\ let(x)\ \xrightarrow{[v/x]}\ let(v)}\ \Gamma(x) = v$$

$$\text{(DEF)}\quad \frac{}{\Delta, \Gamma \vdash\ E_i(v)\ \xrightarrow{\tau}\ [v/x]f_i}\ (E_i(x) \triangleq f_i) \in \Delta$$

$$\text{(DEF-VAR)}\quad \frac{}{\Delta, \Gamma \vdash\ E_i(x)\ \xrightarrow{[v/x]}\ E_i(v)}\ \substack{(E_i(x) \triangleq f_i) \in \Delta,\\ \Gamma(x) = v}$$

$$\text{(SYM-L)}\quad \frac{\Delta, \Gamma \vdash\ f\ \xrightarrow{a}\ f'}{\Delta, \Gamma \vdash\ f \mid g\ \xrightarrow{a}\ f' \mid g}$$

$$\text{(SYM-R)}\quad \frac{\Delta, \Gamma \vdash\ g\ \xrightarrow{a}\ g'}{\Delta, \Gamma \vdash\ f \mid g\ \xrightarrow{a}\ f \mid g'}$$

$$\text{(ASYM-L)}\quad \frac{\Delta, \Gamma \vdash\ f\ \xrightarrow{a}\ f'}{\Delta, \Gamma \vdash\ f\ \mathbf{where}\ x :\in g\ \xrightarrow{a}\ f'\ \mathbf{where}\ x :\in g}\ a \neq [v/x]$$

$$\text{(ASYM-R)}\quad \frac{\Delta, \Gamma \vdash\ g\ \xrightarrow{a}\ g'}{\Delta, \Gamma \vdash\ f\ \mathbf{where}\ x :\in g\ \xrightarrow{a}\ f\ \mathbf{where}\ x :\in g'}\ a \neq !v$$

$$\text{(ASYM-P)}\quad \frac{\Delta, \Gamma \vdash\ g\ \xrightarrow{!v}\ g'}{\Delta, \Gamma \vdash\ f\ \mathbf{where}\ x :\in g\ \xrightarrow{\tau}\ [v/x]f}$$

$$\text{(SEQ)}\quad \frac{\Delta, \Gamma \vdash\ f\ \xrightarrow{a}\ f'}{\Delta, \Gamma \vdash\ f >x> g\ \xrightarrow{a}\ f' >x> g}\ a \neq !v$$

$$\text{(SEQ-P)}\quad \frac{\Delta, \Gamma \vdash\ f\ \xrightarrow{!v}\ f'}{\Delta, \Gamma \vdash\ f >x> g\ \xrightarrow{\tau}\ (f' >x> g) \mid [v/x]g}$$

Fig. 2. Operational Semantics

Asymmetric composition resembles symmetric composition. In f **where** $x :\in g$, f and g execute in parallel unless g publishes. Then, g is terminated and the published value v is communicated via x to f (rule ASYM-P). We can think of x as an implicit communication channel. Rule ASYM-R shows the non-publication steps of g, and ASYM-L shows the steps of f. Free occurences of x in f refer to the binding for x in f **where** $x :\in g$. Thus, even if x is in Γ, f cannot proceed with a receive event for x (though receives for other variables are allowed). Its parts that depend on x will block waiting for a publication from g.

Process $f >x> g$ takes a step if f takes a step (rule SEQ). If f publishes v the process performs an internal event and launches a new instance of g in parallel (rule SEQ-P). As in the asymmetric case, we can think of x as a communication channel between f and g. Thinking of variables as channels also justifies the

By LET-VAR, SEQ	$\Delta, \Gamma \vdash \ let(x) >x> M(x) \stackrel{[2/x]}{\rightarrow} let(2) >x> M(x)$
By LET, SEQ-P	$\Delta, \Gamma \vdash \ let(2) >x> M(x) \stackrel{\tau}{\rightarrow} (\mathbf{0} >x> M(x)) \mid M(2)$
By SITEC, SYM-R	$\Delta, \Gamma \vdash \ (\mathbf{0} >x> M(x)) \mid M(2) \stackrel{M_k(2)}{\rightarrow} (\mathbf{0} >x> M(x)) \mid ?k$
By SITERET, SYM-R	$\Delta, \Gamma \vdash \ (\mathbf{0} >x> M(x)) \mid ?k \stackrel{k?11}{\rightarrow} (\mathbf{0} >x> M(x)) \mid let(11)$
By LET, SYM-R	$\Delta, \Gamma \vdash \ (\mathbf{0} >x> M(x)) \mid let(11) \stackrel{!11}{\rightarrow} (\mathbf{0} >x> M(x)) \mid \mathbf{0}$

Fig. 3. Possible evaluation of $let(x) >x> M(x)$ when $\Gamma = \{(x, 2)\}$

name *receive* event for $[v/x]$. The example in Fig. 3 illustrates the use of some of the rules.

We use the following definitions to model consecutive transitions (also called *executions*) of a process:

Definition 1 (Execution). *t is an execution of f i.e. $\Delta, \Gamma \vdash \ f \stackrel{t}{\rightarrow}{}^* f'$ iff*

- $t = \varepsilon$ *and* $f \equiv f'$, *or*
- $t = a\,t'$ *and for some* f'' $\Delta, \Gamma \vdash \ f \stackrel{a}{\rightarrow} f''$ *and* $\Delta, \Gamma \vdash \ f'' \stackrel{t'}{\rightarrow}{}^* f'$

4 Denotational Semantics

We now present our denotational semantics for Orc, which is the main contribution of this paper. It is based on complete partial orders. The meaning of a process is a set of traces in the presence of environments for the declarations *Fenv* and variables *Env*:

$$[\![f]\!] : [Fenv \rightarrow [Env \rightarrow P]]$$

A trace is a (possibly empty) sequence of events. Trace sets are prefix-closed and ordered by inclusion. They are also non-empty because the empty trace ε is a trace of any process. Last, we consider traces of finite length only; an infinite trace is represented by the set of all its finite prefixes.

$$\begin{array}{l} Traces = Event^*, \text{ a discrete CPO.} \\ P = \{\, S \mid S \subseteq Traces \ \wedge \ S \neq \emptyset \ \wedge \ S \text{ is prefix-closed}\} \end{array}$$

The metavariable ρ ranges over *Env*. We use two kinds of bindings in ρ, in order to differentiate between a variable x bound in $(f \ \mathbf{where} \ x :\in g)$ versus $(f >x> g)$ or $(E_i(x) \triangleq f_i)$. In the former case $\rho(x) \in GetVal$ and in the latter case $\rho(x) \in GotVal$, where

$$\begin{array}{l} Val = \text{the set of all values, a discrete CPO.} \\ Var = \text{the set of all variable names, a discrete CPO.} \\ GetVal = \{\, \natural v \mid v \in Val\} \\ GotVal = \{\, \flat v \mid v \in Val\} \\ Env = [\, Var \rightarrow (GetVal \cup GotVal \cup \{\text{Absent}\})] \end{array}$$

$$
\begin{aligned}
&[\![\mathbf{0}]\!] = \lambda\varphi.\lambda\rho.\{\varepsilon\}\\
&[\![let(v)]\!] = \lambda\varphi.\lambda\rho.\{!v\}_{\mathrm{p}}\\
&[\![let(x)]\!] = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ \mathrm{Absent}.\{\varepsilon\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \flat v.\{!v\}_{\mathrm{p}}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \natural v.\{[v/x]\ !v\}_{\mathrm{p}}\\
&[\![M(v)]\!] = \lambda\varphi.\lambda\rho.\{\ M_k(v)\ k?w\ !w \mid k\ \mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}\\
&[\![M(x)]\!] = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ \mathrm{Absent}.\{\varepsilon\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \flat v.\{\ M_k(v)\ k?w\ !w \mid k\ \mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \natural v.\{\ [v/x]\ M_k(v)\ k?w\ !w \mid k\ \mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}\\
&[\![?k]\!] = \lambda\varphi.\lambda\rho.\{\ k?w\ !w \mid w \in \mathit{Val}\}_{\mathrm{p}}\\
&[\![E_i(v)]\!] = \lambda\varphi.\lambda\rho.\{\ \tau\ t \mid t \in \varphi_i(v)\}_{\mathrm{p}}\\
&[\![E_i(x)]\!] = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ \mathrm{Absent}.\{\varepsilon\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \flat v.\{\ \tau\ t \mid t \in \varphi_i(v)\}_{\mathrm{p}}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \natural v.\{\ [v/x]\ \tau\ t \mid t \in \varphi_i(v)\}_{\mathrm{p}}\\
&[\![h \mid g]\!] = \lambda\varphi.\lambda\rho.\ [\![h]\!]\varphi\rho \parallel [\![g]\!]\varphi\rho\\
&[\![h >\!x\!> g]\!] = \lambda\varphi.\lambda\rho.\textstyle\bigcup_{s\in[\![h]\!]\varphi\rho}\ s \gg \lambda v.[\![g]\!]\varphi\rho[x = \flat v]\\
&[\![h\ \mathbf{where}\ x :\in g]\!] = \lambda\varphi.\lambda\rho.\ (\textstyle\bigcup_{v\in \mathit{Val}}\ [\![h]\!]\varphi\rho[x = \natural v]) <_x [\![g]\!]\varphi\rho
\end{aligned}
$$

Fig. 4. Trace Semantics of Orc

The distinction is necessary because the evaluation of f in $(f\ \mathbf{where}\ x :\in g)$ can start before f has a value for x. Then, the parts of f that depend on x will block. We would like to know which events in a trace t of f happened before g sent a value v for x and which events followed. To do that, any sub-process of f that uses x reveals it by inserting a receive event $[v/x]$ in its trace. So, t may be of the form $t_1[v/x]t_2$ where $[v/x]$ is the first receive event in t and juxtaposition means trace concatenation. Now, we can deduce that the events in t_1 happen independently of x; they can also happen when g does not publish. On the other hand, there is no need to mark the use of x in the traces of $(f >\!x\!> g)$ or $(E_i(x) \triangleq f_i)$ because x always has a value when g or f_i is evaluated. Moreover, since only x can be free in f_i, the traces of the declarations do not contain any receive events.

$$
\begin{aligned}
&\mathit{NoRecv} = \{\ S \mid S \in P\ \wedge\ \forall t \in S.\ \text{no receives in}\ t\}\\
&\mathit{Fenv} = ([\mathit{Val} \to \mathit{NoRecv}])^k
\end{aligned}
$$

NoRecv is a CPO with bottom element $\{\varepsilon\}$ and *Fenv* inherits its order from *NoRecv* in the usual way. We do not need names to refer to the declared processes, we can index them by the order of declaration.

The definitions of the meaning functions can be found in Fig. 4. The inert process has no transitions, thus it has no traces but the empty trace ε.

The denotation of $let(v)$ is straightforward. The prefix-closure operator is defined, among others, in Fig. 5. The traces of $let(x)$ depend on the environment. If x is not in the environment (signified by $\rho(x) = \mathrm{Absent}$) then $let(x)$ behaves like $\mathbf{0}$. If $\rho(x) = \natural v$ then a receive event $[v/x]$ reveals the use of x. Last, if $\rho(x) = \flat v$, only the publication is observed.

Concatenate a trace and a trace-set:
$s\,T \triangleq \{\, s\,t \mid t \in T \}$

Remove event 'a' from a trace:
$$t\backslash a \triangleq \begin{cases} \varepsilon & t = \varepsilon \\ t'\backslash a & t = at' \\ a'\,(t'\backslash a) & t = a't' \text{ and } a \neq a' \end{cases}$$

Remove event from a trace-set:
$T\backslash a \triangleq \{\, t\backslash a \mid t \in T \}$

Merge:
$$t_1 \parallel t_2 \triangleq \begin{cases} \{t_1\} & t_2 = \varepsilon \\ \{t_2\} & t_1 = \varepsilon \\ a(t_1' \parallel t_2) \cup b(t_1 \parallel t_2') & t_1 = at_1' \text{ and } t_2 = bt_2' \end{cases}$$

Merge trace-sets:
$T_1 \parallel T_2 \triangleq \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \parallel t_2$

Prefix-closure:
$$t_{\mathrm{p}} \triangleq \begin{cases} \{\varepsilon\} & t = \varepsilon \\ \{\varepsilon, a\} \cup a\,t_{\mathrm{p}}' & t = at' \end{cases}$$

Prefix-closure for trace-sets:
$S_{\mathrm{p}} \triangleq \bigcup_{s \in S} s_{\mathrm{p}}$

Sequencing combinator:
$$s \gg F = \begin{cases} \{s\} & \text{no publ. in } s \\ s_1\,\tau\,((s_2 \gg F) \parallel F(v)) & s \equiv s_1 ! v s_2 \text{ , no publ. in } s_1 \end{cases}$$

Asymmetric combinator:
$$t_1 <_x t_2 = \begin{cases} t_1 \parallel t_2 & \text{no recv. for } x \text{ in } t_1 \text{ , no publ. in } t_2 \\ t_1 \parallel t_{21}\tau & \text{no recv. for } x \text{ in } t_1 \text{ , } t_2 \equiv t_{21} ! v\, t_{22} \text{ , no publ. in } t_{21} \\ (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x]) & t_1 \equiv t_{11}[v/x]t_{12} \text{ , no recv. for } x \text{ in } t_{11} \text{ , } \\ & t_2 \equiv t_{21} ! v\, t_{22} \text{ , no publ. in } t_{21} \\ \{\varepsilon\} & \text{otherwise} \end{cases}$$

Asymmetric combinator for trace-sets:
$T_1 <_x T_2 = \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 <_x t_2$

Empty environment ρ_0:
$\forall x.\, \rho_0(x) = \text{Absent}$

Fig. 5. Various Definitions

The meaning functions for site calls are quite similar. Note the many possible responses to the same call. For simple processes like $M(4)$ and $let(x)$, it is easy to see that their traces coincide with their executions. We will prove that true for all Orc processes.

The traces of $E_i(v)$ are independent of the environment (since only x can be free in the body of $E_i(x)$). They are the traces of the i^{th} declaration, preceded by τ.

In symmetric composition, we get the traces by merging the traces of the constituent processes.

The denotation of $h >x> g$ can be demystified by observing the operational behavior of this process. Every trace s of h that does not publish is also a trace of $h >x> g$. Moreover, if s contains a publication, an instance of g is launched in parallel and the remaining transitions of h may spawn more instances of g.

Last, we need to look at the denotation of h **where** $x :\in g$. Let t_1 be a trace of h and t_2 a trace of g. If t_1 does not contain receive events for x it is independent of x. Thus, if t_2 contains no publication, we just merge the two traces. If t_2 contains a publication $!v$ we know that the part that follows $!v$ will be discarded because g is terminated. That is why we only merge t_1 with the part of t_2 prior to $!v$. If t_1 contains a receive event for x, the part after this event depends on x. Consequently, if t_2 contains a matching publication, the traces are merged prior to the publication and concatenated with the rest of t_1. The fourth branch of the definition stops us from creating nonsensical traces, as when combining a t_1 that receives x with a t_2 that does not publish.

We can easily establish the following properties of the meaning functions:

Theorem 1 (Prefix Closure of Trace Sets). *For all* f, φ, ρ, $[\![f]\!]\varphi\rho \in P$

Theorem 2 (Continuity of Denotations). *For all* f, $[\![f]\!]$ *is continuous.*

The proofs of these and all subsequent theorems can be found in the appendix. Finally, we apply the usual fixed-point technique [13] to give the denotation of a set of declarations Δ. We define an *Fenv* transformer $\hat{\Delta}$ by

$$\hat{\Delta} = \lambda\varphi.(\lambda v.[\![f_1]\!]\varphi\rho_0[x = \flat v] \times \cdots \times \lambda v.[\![f_k]\!]\varphi\rho_0[x = \flat v])$$

$\hat{\Delta}$ is continuous, so we define $[\![\Delta]\!]$ as its least fixed point

$$[\![\Delta]\!] = \text{fix}(\hat{\Delta})$$

To prove the correctness of our semantics we need to show that the executions of a process match its traces.

Theorem 3 (Adequacy).
If $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$,
$\sigma = [w_1/y_1] \ldots [w_n/y_n]$,
$\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m][y_1 = \flat w_1] \ldots [y_n = \flat w_n]$,
where the x's and y's are all distinct, then

$$t \in [\![f]\!][\![\Delta]\!]\rho \quad \textit{iff} \quad \exists f'.\, \Delta, \Gamma \vdash \sigma f \xrightarrow{t}{}^{*} f'$$

$$\text{(SITECALL)}\quad \frac{k \text{ fresh}}{M(v) \overset{M_k(v)}{\rightarrow} ?k} \qquad \text{(SITERET)}\quad \frac{}{?k \overset{k?v}{\rightarrow} let(v)}$$

$$\text{(LET)}\quad \frac{}{let(v) \overset{!v}{\rightarrow} \mathbf{0}} \qquad \text{(ASYM1N)}\quad \frac{f \overset{a}{\rightarrow} f'}{f \textbf{ where } x :\in g \overset{a}{\rightarrow} f' \textbf{ where } x :\in g}$$

$$\text{(ASYM1V)}\quad \frac{g \overset{!v}{\rightarrow} g'}{f \textbf{ where } x :\in g \overset{\tau}{\rightarrow} [v/x]f} \qquad \text{(ASYM2)}\quad \frac{g \overset{a}{\rightarrow} g' \qquad a \neq !v}{f \textbf{ where } x :\in g \overset{a}{\rightarrow} f \textbf{ where } x :\in g'}$$

$$\text{(SEQ1N)}\quad \frac{f \overset{a}{\rightarrow} f' \qquad a \neq !v}{f >x> g \overset{a}{\rightarrow} f' >x> g} \qquad \text{(SEQ1V)}\quad \frac{f \overset{!v}{\rightarrow} f'}{f >x> g \overset{\tau}{\rightarrow} (f' >x> g) \mid [v/x]g}$$

$$\text{(SUBST)}\quad \frac{}{f \overset{[v/x]}{\rightarrow} [v/x]f}$$

Fig. 6. Operational Semantics of Orc_1

The theorem is proved by induction on the length of t. It relies on the following lemma, which is proved by structural induction on f.

Lemma 1. *If* $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$, $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$ *then*

$$a\, t \in [\![f]\!][\![\Delta]\!]\rho \quad \textit{iff} \quad \exists f'.\, \Delta, \Gamma \vdash f \overset{a}{\rightarrow} f' \textit{ and } t \in [\![f']\!][\![\Delta]\!]\rho$$

5 Deficiencies of the previous trace semantics

The operational semantics of section 3 differs slightly from the previously proposed operational semantics. The main difference is our use of an environment Γ for the variables. The previous semantics treats free variables more permissively, a fact which makes the denotational treatment in [9] incorrect. Here, we only present a subset of the semantics which suffices to show the error. Thus, we are not concerned with recursive definitions.

In this section we will refer to Orc as presented in [9] as Orc_1 and to our version as Orc_2. Fig. 6 shows the operational semantics of Orc_1. All but the last rule are self explanatory. The last rule says that a process can spontaneously decide to substitute a value v for a variable x. Any process f can perform any substitution step, even for variables not free in f (of course then $[v/x]f = f$). The constraint is that the SUBST rule cannot be applied to parts of an expression, in other words the event 'a' in the other rules cannot be a receive event for any variable.

The traces of an Orc_1 process are defined *operationally*. The trace-set $\langle f \rangle$ of a process f is

$$\langle f \rangle = \{\, t \mid \exists s, f'.\, f \overset{s}{\rightarrow}{}^{*} f' \;\wedge\; t = s \backslash \tau \}$$

Asymmetric combinator:

$$t_1 \textbf{ where } x :\in t_2 = \begin{cases} t_1 \mid t_2 & \text{no publ. in } t_2,\ \text{no recv. for } x \text{ in } t_2 \\ (t_{11} \mid t_{21}) t_{12} & t_1 \equiv t_{11}[v/x]t_{12}\ ,\ \text{no recv. for } x \text{ in } t_{11}\ , \\ & t_2 \equiv t_{21} ?v\, t_{22}\ ,\ \text{no publ. in } t_{21}, \\ & \text{no recv. for } x \text{ in } t_2 \\ \emptyset & \text{otherwise} \end{cases}$$

$T_1 \textbf{ where } x :\in T_2 = \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \textbf{ where } x :\in t_2$

Trace selection using one receive event: $\quad T \upharpoonright [v/x] = \{\, t \mid [v/x]\, t \in T \}$

Trace selection using many receive events:
$T \upharpoonright \varepsilon = T$
$T \upharpoonright ([v/x]\, \sigma) = (T \upharpoonright [v/x]) \upharpoonright \sigma$

Subsequence of receive events from a trace:

$$\mathit{Recv}(t) = \begin{cases} \varepsilon & t = \varepsilon \\ [v/x]\, \mathit{Recv}(t') & t = [v/x]\, t' \\ \mathit{Recv}(t') & t = a\, t' \text{ and } a \text{ not a receive event} \end{cases}$$

Sequencing combinator:

$$s >\!x\!> T = \begin{cases} \{s\} & \text{no publ. in } s,\ \text{no recv. for } x \text{ in } s \\ s_1((s_2 >\!x\!> T') \mid (T' \upharpoonright [u/x])) & s = s_1 ! u\, s_2,\ \text{no publ. in } s_1, \\ & \text{no recv. for } x \text{ in } s,\ T' = T \upharpoonright \mathit{Recv}(s_1) \\ \emptyset & \text{otherwise} \end{cases}$$

$S >\!x\!> T = \bigcup_{s \in S} s >\!x\!> T$

Fig. 7. Operators on traces in Orc_1

Their objective is to prove that $\langle f >\!x\!> g \rangle$ and $\langle f \textbf{ where } x :\in g \rangle$ can be obtained from $\langle f \rangle$ and $\langle g \rangle$ using suitable operators on trace sets (see Fig. 7).[2]

The asymmetric combinator is defined for traces and lifted for trace sets. The operator $\mid$ is a merge operator. Its precise definition is not needed; we only need to know that $t \mid \varepsilon = \{t\}$. The operator $T \upharpoonright \sigma$ selects the traces in T whose prefix is σ and removes that prefix. In the definition of $(s_1 ! u\, s_2 >\!x\!> T)$ this operator

[2] Receive events are called substitution events in [9]. The precise definition of their sequencing combinator is

$$s >\!x\!> T = \begin{cases} \{s\} & \text{no publ. in } s \\ s_1((s_2 >\!x\!> T') \mid (T' \upharpoonright [u/x])) & s = s_1 ! u\, s_2,\ \text{no publ. in } s_1, \\ & T' = T \upharpoonright \mathit{Recv}(s_1) \end{cases}$$

accompanied by the note "Any substitution event $[v/x]$ in s is unrelated to x in $(s >\!x\!> T)$." We interpret this to mean that no receive events for x are allowed in s, and hence, if s contains a receive event for x then $s >\!x\!> T = \emptyset$. We interpreted a similar note for t_2 in their definition of $(t_1 \textbf{ where } x :\in t_2)$ in the same way.

is used to make sure that the receive events in T agree with the receive events in s_1.

In order to prove compositionality, it suffices to prove Claims 1, 2. We provide counterexamples to refute both.

Claim 1 $\langle f \textbf{ where } x :\in g\rangle = \langle f\rangle \textbf{ where } x :\in \langle g\rangle$

Let $h = (let(x) \textbf{ where } x :\in \mathbf{0})$. By SUBST and LET, $t = ([2/x]\,!2) \in \langle let(x)\rangle$ and also $\varepsilon \in \langle \mathbf{0}\rangle$. Then, $(([2/x]\,!2) \textbf{ where } x :\in \varepsilon) = (([2/x]\,!2) \mid \varepsilon) = \{[2/x]\,!2\}$ which yields $([2/x]\,!2) \in (\langle let(x)\rangle \textbf{ where } x :\in \langle \mathbf{0}\rangle)$. However, the only operational rule that applies to h is SUBST, which takes h to itself. This means that a trace of h can consist only of receive events, hence $t \notin \langle h\rangle$. Therefore, $\langle f \textbf{ where } x :\in g\rangle \neq \langle f\rangle \textbf{ where } x :\in \langle g\rangle$

Claim 2 $\langle f >x> g\rangle = \langle f\rangle >x> \langle g\rangle$

Let $h = M(1) >x> let(x)$. By rules SUBST, SITECALL and SEQ1N we get $[3/x]\,M_k(1) \in \langle h\rangle$. Let $t = [3/x]\,M_k(1)$. Assume that $t \in (\langle M(1)\rangle >x> \langle let(x)\rangle)$. Then, there exists $s \in \langle M(1)\rangle$ such that $t \in (s >x> \langle let(x)\rangle)$. If s contains no publication and no receive for x then $t = s$ which is a contradiction because $([3/x]\,M_k(1)) >x> \langle let(x)\rangle = \emptyset$
If s publishes, it is of the form $(\sigma_1\, M_k(1)\, \sigma_2\, k?w\, \sigma_3\, !w\, \sigma_4)$ where $\sigma_1, \ldots, \sigma_4$ are arbitrary sequences of receive events. But then, $\sigma_1 = [3/x]$ which is a contradiction because $([3/x]\,M_k(1)\,\sigma_2\,k?w\,\sigma_3\,!w\,\sigma_4) >x> \langle let(x)\rangle = \emptyset$. We conclude that there is no $s \in \langle M(1)\rangle$ such that $t \in (s >x> \langle let(x)\rangle)$. Hence, $t \notin (\langle M(1)\rangle >x> \langle let(x)\rangle)$ and $\langle f >x> g\rangle \neq \langle f\rangle >x> \langle g\rangle$

Our intuition is that each of these counterexamples stems from the non-restrictive usage of substitutions in Orc_1 (rule SUBST). By allowing arbitrary receive events, they make a process unaware of its context. It can transition with a dummy receive event at any time, so its traces blow up and reasoning about them is hard. Our addition of an environment in the operational semantics of Orc_2 restricts the receive-transitions of a process to just the useful ones.

6 Strong Bisimulation Congruences

In [9], Kitchin et al. have proved some useful equivalences between processes using strong bisimulation [10]. For our semantics, we define a family of strong bisimulation relations indexed by Δ:

Definition 2 (Δ-bisimulation). *Let Δ be a set of declarations. Then, a binary relation $\mathfrak{R}$ on processes is a Δ-bisimulation iff*

1. *$\mathfrak{R}$ is symmetric*
2. *for any $(f, g) \in \mathfrak{R}$ and for any Γ, σ if $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$ then $\exists g'.\ \Delta, \Gamma \vdash \sigma g \xrightarrow{a} g'$ and $(f', g') \in \mathfrak{R}$*

Definition 3 (Largest Strong Bisimulation). $\sim_\Delta \triangleq \bigcup\{\mathfrak{R} \mid \mathfrak{R} \textit{ is a } \Delta\textit{-bisim.}\}$

For any Δ such that f, g, h are well-formed,

1. $f \mid \mathbf{0} \sim_\Delta f$
2. $f \mid g \sim_\Delta g \mid f$
3. $f \mid (g \mid h) \sim_\Delta (f \mid g) \mid h$
4. $(f \mid g) >x> h \sim_\Delta (f >x> h) \mid (g >x> h)$
5. $f >x> (g >y> h) \sim_\Delta (f >x> g) >y> h$ if $x \notin$ f.v.(h)
6. $(f \mid g)$ **where** $x :\in h \sim_\Delta (f$ **where** $x :\in h) \mid g$ if $x \notin$ f.v.(g)
7. $(f >y> g)$ **where** $x :\in h \sim_\Delta (f$ **where** $x :\in h) >y> g$ if $x \notin$ f.v.(g)
8. $(f$ **where** $x :\in g)$ **where** $y :\in h \sim_\Delta (f$ **where** $y :\in h)$ **where** $x :\in g$ if $y \notin$ f.v.(g) and $x \notin$ f.v.(h)

Fig. 8. Strongly Bisimilar Processes

For different declaration sets we get different bisimulations. For example,

$$E_1(v) \sim_{\Delta_1} (let(v) >x> M(x)) \quad \text{for} \quad \Delta_1 = \{E_1(x) \triangleq M(x)\}$$

but

$$E_1(v) \not\sim_{\Delta_2} (let(v) >x> M(x)) \quad \text{for} \quad \Delta_2 = \{E_1(x) \triangleq \mathbf{0}\}$$

We can prove the equivalences in [9] using our new operational semantics (see Fig 8). Naturally, symmetric composition is commutative and associative (equiv. 2, 3). Symmetric composition can be distributed over sequencing because symmetrically composed processes do not communicate with each other (equiv. 4). Equivalence 6 verifies our intuition that a (**where** x)-context does not influence a process g if x is not free in g.

Lemma 2. *For any Δ, $\sim_\Delta$ is a congruence relation*

The proof proceeds by induction on contexts. By Lemma 2, the equivalences of Fig. 8 become congruences automatically. Congruence is important in a concurrent setting, because we can replace a process in a system with a congruent process without affecting the behavior of the system. The following example illustrates congruences 1, 2 and 6 when $x \notin$ f.v.(g)

$$g \textbf{ where } x :\in h \ \sim_\Delta (\mathbf{0} \mid g) \textbf{ where } x :\in h \ \sim_\Delta (\mathbf{0} \textbf{ where } x :\in h) \mid g$$

Definition 2 is universally quantified over Γ, σ. This helps establish a connection between strong bisimulation and trace equivalence:

Theorem 4. *If $f \sim_\Delta g$ then for any ρ it holds that $[\![f]\!][\![\Delta]\!]\rho = [\![g]\!][\![\Delta]\!]\rho$*

The proof is straightforward. This result provides an additional way to prove trace equivalence, which will come in handy when we show Lemma 3 in the next section.

Compositional Semantics without τ's:

$$\{\!|\mathbf{0}|\!\} = \lambda\varphi.\lambda\rho.\{\varepsilon\}$$
$$\{\!|let(v)|\!\} = \lambda\varphi.\lambda\rho.\{!v\}_{\mathrm{p}}$$
$$\begin{aligned}\{\!|let(x)|\!\} = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ &\mathrm{Absent}.\{\varepsilon\}\\ &\flat v.\{!v\}_{\mathrm{p}}\\ &\natural v.\{[v/x]\ !v\}_{\mathrm{p}}\end{aligned}$$
$$\{\!|M(v)|\!\} = \lambda\varphi.\lambda\rho.\{\ M_k(v)\ k?w\ !w \mid k\,\mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}$$
$$\begin{aligned}\{\!|M(x)|\!\} = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ &\mathrm{Absent}.\{\varepsilon\}\\ &\flat v.\{\ M_k(v)\ k?w\ !w \mid k\,\mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}\\ &\natural v.\{\ [v/x]\ M_k(v)\ k?w\ !w \mid k\,\mathrm{fresh}\,, w \in \mathit{Val}\}_{\mathrm{p}}\end{aligned}$$
$$\{\!|?k|\!\} = \lambda\varphi.\lambda\rho.\{\ k?w\ !w \mid w \in \mathit{Val}\}_{\mathrm{p}}$$
$$\{\!|E_i(v)|\!\} = \lambda\varphi.\lambda\rho.\varphi_i(v)$$
$$\begin{aligned}\{\!|E_i(x)|\!\} = \lambda\varphi.\lambda\rho.\mathbf{case}\ \rho(x)\ \mathbf{of}\ \ &\mathrm{Absent}.\{\varepsilon\}\\ &\flat v.\varphi_i(v)\\ &\natural v.\{\ [v/x]\ t \mid t \in \varphi_i(v)\}_{\mathrm{p}}\end{aligned}$$
$$\{\!|h \mid g|\!\} = \lambda\varphi.\lambda\rho.\ \{\!|h|\!\}\varphi\rho\ \parallel\ \{\!|g|\!\}\varphi\rho$$
$$\{\!|h >x> g|\!\} = \lambda\varphi.\lambda\rho.\bigcup_{s\in\{\!|h|\!\}\varphi\rho}\ s \mathbin{\dot{\gg}} \lambda v.\{\!|g|\!\}\varphi\rho[x = \flat v]$$
$$\{\!|h\ \mathbf{where}\ x :\in g|\!\} = \lambda\varphi.\lambda\rho.\ (\bigcup_{v\in\mathit{Val}}\ \{\!|h|\!\}\varphi\rho[x = \natural v]) \mathbin{\dot{<}_x}\ \{\!|g|\!\}\varphi\rho$$

Changed definitions of section 4:

$$\mathit{NoRecv}' = \{\ S \mid S \in \mathit{NoRecv} \quad\wedge\quad \forall t \in S.\ \text{no } \tau\text{'s in } t\}$$
$$\mathit{Fenv}' = ([\mathit{Val} \to \mathit{NoRecv}'])^k$$

$$s \mathbin{\dot{\gg}} F = \begin{cases}\{s\} & \text{no publ. in } s\\ s_1((s_2 \gg F) \parallel F(v)) & s \equiv s_1 ! v s_2\,, \text{no publ. in } s_1\end{cases}$$

$$t_1 \mathbin{\dot{<}_x} t_2 = \begin{cases}t_1 \parallel t_2 & \text{no recv. for } x \text{ in } t_1\,, \text{no publ. in } t_2\\ t_1 \parallel t_{21} & \text{no recv. for } x \text{ in } t_1\,,\ t_2 \equiv t_{21} ! v\, t_{22}\,, \text{no publ. in } t_{21}\\ (t_{11} \parallel t_{21})(t_{12} \backslash [v/x]) & t_1 \equiv t_{11}[v/x]t_{12}\,, \text{no recv. for } x \text{ in } t_{11}\,,\\ & t_2 \equiv t_{21} ! v\, t_{22}\,, \text{no publ. in } t_{21}\\ \{\varepsilon\} & \text{otherwise}\end{cases}$$

Fig. 9. Semantics without internal events

7 Semantics insensitive to internal events

Any Orc process can be a building block of a larger process, e.g. $\mathit{Factorize}(N)$ in $(\mathit{Print}(x)\ \mathbf{where}\ x :\in (\mathit{Factorize}(N) \mid \mathit{RedditFeed}(\mathit{today})))$. In such situations, the internal events of a process are not observable by its context, in the sense that they do not entail communication between the process and the rest of the system. Instead, τ events represent communication that takes place *within* the process. Therefore, we would like to have a semantics insensitive to internal events. Such a semantics is shown in Fig. 9. It is quite similar to the semantics in section 4, without τ events. In fact, we can show that:

Theorem 5. *For all f, $\{\!|f|\!\} = \lambda\varphi.\lambda\rho.[\![f]\!]\varphi\rho\backslash\tau$*

The meaning functions are continuous, since they are composed of continuous functions. Obviously, $[\![f]\!] = [\![g]\!]$ implies $\{\!|f|\!\} = \{\!|g|\!\}$. Therefore, the semantics in this section is less discriminating than the semantics in section 4. We can now prove the following interesting equivalence, which is false in our original semantics:

Lemma 3. *For all* $f, \rho \quad \{\!|f|\!\}\{\!|\Delta|\!\}\rho = \{\!|f >x> let(x)|\!\}\{\!|\Delta|\!\}\rho$

Proof. We prove that each side is a subset of the other. In the forward direction, we proceed by induction on the number of publications in some trace t of f. For the reverse direction, we use Theorems 4 and 5. We define a new length for traces that ignores duplicate receive events and proceed by lexicographic induction on this new length of t and the size of f.

The statement of the lemma uses Δ instead of φ because the proof employs the congruences of section 6. We use the above lemma to prove that there is no causality between a publication and the events that follow it in a trace:

Lemma 4. *If* $s_1 \,!v\, s_2 \in \{\!|f|\!\}\{\!|\Delta|\!\}\rho$ *then* $s_1(!v \parallel s_2) \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$

8 Conclusions and Related Work

Task orchestration is related to various industrial standards for business transactions (e.g. WSBPEL [1], WSCDL [8]). Academics have also looked at other aspects of business transactions, such as compensations (see [6], [3], [4], [5], [2]). A formal specification for a subset of WSBPEL has been proposed as well [12].

In this paper we presented operational and denotational semantics for Orc, a language for task orchestration. We proved an adequacy theorem, showing that the operational transitions of a process coincide with its denotational traces. This was not the case in [9], as demonstrated in section 5. Our semantics offers a better treatment of variable binding because it does not allow a process to transition with an arbitrary receive event. In addition, we showed a number of useful congruences using strong bisimulation and proved that strongly bisimilar processes have the same traces. Finally, we presented a trace semantics that ignores internal events, thereby equating more processes than our original semantics. In the future we want to investigate the properties of processes in the presence of timeouts and propose a timed semantics for Orc.

The semantics in [9] and this paper are asynchronous. Misra et al. [11] augment the operational semantics of [9] with a synchronous semantics. This is an operational semantics that gives priority to internal events, thus allowing the possibility for processes to synchronize on external interactions. However, they do not give a denotational semantics, nor do they state any theorems. Hoare et al. [7] present a tree-based denotational semantics for Orc, and sketch an operational semantics based on the same trees. They prove a number of interesting denotational equivalences, but do not state any theorem relating the operational and denotational semantics.

References

1. Alexandre Alves, Assaf Arkin, et al. Web services business process execution language version 2.0. Technical report, April 2007. http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf.
2. Roberto Bruni, Michael J. Butler, Carla Ferreira, C. A. R. Hoare, Hernán C. Melgratti, and Ugo Montanari. Comparing two approaches to compensable flow composition. In Martín Abadi and Luca de Alfaro, editors, *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 383–397. Springer, 2005.
3. Roberto Bruni, Hernán C. Melgratti, and Ugo Montanari. Theoretical foundations for compensations in flow composition languages. In Jens Palsberg and Martín Abadi, editors, *POPL*, pages 209–220. ACM, 2005.
4. Michael J. Butler and Carla Ferreira. A process compensation language. In Wolfgang Grieskamp, Thomas Santen, and Bill Stoddart, editors, *IFM*, volume 1945 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2000.
5. Michael J. Butler and Carla Ferreira. An operational semantics for stac, a language for modelling long-running business transactions. In Rocco De Nicola, Gian Luigi Ferrari, and Greg Meredith, editors, *COORDINATION*, volume 2949 of *Lecture Notes in Computer Science*, pages 87–104. Springer, 2004.
6. Michael J. Butler, C. A. R. Hoare, and Carla Ferreira. A trace semantics for long-running transactions. In Ali E. Abdallah, Cliff B. Jones, and Jeff W. Sanders, editors, *25 Years Communicating Sequential Processes*, volume 3525 of *Lecture Notes in Computer Science*, pages 133–150. Springer, 2004.
7. Tony Hoare, Galen Menzel, and Jayadev Misra. A tree semantics for an orchestration language, August 2004. Lecture Notes for NATO summer school, Marktoberdorf.
8. Nickolas Kavantzas, David Burdett, et al. Web services choreography description language version 1.0. Technical report, November 2005. http://www.w3.org/TR/ws-cdl-10/.
9. David Kitchin, William R. Cook, and Jayadev Misra. A language for task orchestration and its semantic properties. In Christel Baier and Holger Hermanns, editors, *CONCUR*, volume 4137 of *Lecture Notes in Computer Science*, pages 477–491. Springer, 2006.
10. Robin Milner. Operational and algebraic semantics of concurrent processes. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 1201–1242. 1990.
11. Jayadev Misra and William R. Cook. Computation orchestration: A basis for wide-area computing. *Software and Systems Modeling*, 6(1):83–110, 2007.
12. Mirko Viroli. Towards a formal foundation to orchestration languages. *Electr. Notes Theor. Comput. Sci.*, 105:51–71, 2004.
13. Glynn Winskel. *The Formal Semantics of Programming Languages*. MIT Press, Cambridge, MA, 1993.

A Various Definitions

Definition 4. *Concatenate a trace and a trace-set*
$s\,T \triangleq \{\, s\,t \mid t \in T\}$

Definition 5. *Concatenate trace-sets*
$T_1\,T_2 \triangleq \{\, t_1 t_2 \mid t_1 \in T_1, t_2 \in T_2\}$

Definition 6. *Remove event 'a' from a trace*
$$t\backslash a \triangleq \begin{cases} \varepsilon & t = \varepsilon \\ t'\backslash a & t = at' \\ a'\,t'\backslash a & t = a't' \text{ and } a \neq a' \end{cases}$$

Definition 7. *Remove event from a trace-set*
$T\backslash a \triangleq \{\, t\backslash a \mid t \in T\}$

Definition 8. *Merge for traces*
$$t_1 \parallel t_2 \triangleq \begin{cases} \{t_1\} & t_2 = \varepsilon \\ \{t_2\} & t_1 = \varepsilon \\ a(t_1' \parallel t_2) \cup b(t_1 \parallel t_2') & t_1 = at_1' \text{ and } t_2 = bt_2' \end{cases}$$

Definition 9. *Merge for trace-sets*
$T_1 \parallel T_2 \triangleq \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \parallel t_2$

Definition 10. *Prefixing*
$$t_{\mathrm{p}} \triangleq \begin{cases} \{\varepsilon\} & t = \varepsilon \\ \{\varepsilon, a\} \cup a\,t_{\mathrm{p}}' & t = at' \end{cases}$$

Definition 11. *Prefixing for trace-sets*
$S_{\mathrm{p}} \triangleq \bigcup_{s \in S} s_{\mathrm{p}}$

Definition 12. *Extend-env:* $Env \times (Val \times (GetVal \cup GotVal)) \to Env$
$\rho[x = u] \triangleq (\rho - \{(x, w)\}) \cup \{(x, u)\}$ *,where* $\rho(x) = w$

Definition 13. *Alternate merge*
$$t_1 \mathbin{\breve{\parallel}} t_2 \triangleq \begin{cases} \{t_1\} & t_2 = \varepsilon \\ \{t_2\} & t_1 = \varepsilon \\ (t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b & t_1 = t_1'a \text{ and } t_2 = t_2'b \end{cases}$$

Definition 14. *Alternate merge for trace-sets*
$T_1 \mathbin{\breve{\parallel}} T_2 \triangleq \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \mathbin{\breve{\parallel}} t_2$

Definition 15. *Trace length that ignores duplicate receive events*
$$\lceil t \rceil = \begin{cases} 0 & t = \varepsilon \\ 1 + \lceil t'\backslash [v/x] \rceil & t = [v/x]t' \\ 1 + \lceil t' \rceil & t = a\,t' \quad \text{and for any} \quad v, x \quad a \neq [v/x] \end{cases}$$

Definition 16. $\rho_{-x}(y) = \begin{cases} \text{Absent} & y = x \\ \rho(y) & y \neq x \end{cases}$

Note 5 *ρ_0 is an environment such that* $\forall x.\rho_0(x) = \text{Absent}$

Note 6 *$a \hat{\in} t$ means that trace t contains event a. $a \hat{\notin} t$ means that trace t does not contain event a.*

Definition 17. *Ordering of pairs of integers*
$(i, j) \sqsubset (k, l)$ *when* $(i < k) \vee (i = k \wedge j < l)$

B Strong Bisimulation

To improve readability, in this section we use the notation $\langle a, b\rangle$ for ordered pairs instead of (a, b).

Definition 18 (Strong Bisimulation). *The binary relation $\mathfrak{R}$ on processes is a strong bisimulation if*

1. *$\mathfrak{R}$ is symmetric*
2. *for any $\langle f, g\rangle \in \mathfrak{R}$ and for any Γ, σ if $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$ then $\Delta, \Gamma \vdash \sigma g \xrightarrow{a} g'$ and $\langle f', g'\rangle \in \mathfrak{R}$*

Definition 19 (Largest Strong Bisimulation). $\sim \triangleq \bigcup\{\, \mathfrak{R} \mid \mathfrak{R}$ *is a strong bisim.*$\}$

Definition 20 (Strong Bisimulation up to $\sim$). *The binary relation $\mathfrak{R}$ on processes is a strong bisimulation up to $\sim$, if $\sim \mathfrak{R} \sim$ is a strong bisimulation*

Lemma 7. *$\sim$ is an equivalence relation*

Lemma 8. $f \mid 0 \sim f$

Lemma 9. $f \mid g \sim g \mid f$

Lemma 10. $f \mid (g \mid h) \sim (f \mid g) \mid h$

Proof. $\mathfrak{R}_1 = \{\langle f \mid (g \mid h), (f \mid g) \mid h\rangle\}$
$\mathfrak{R} = \mathfrak{R}_1 \cup \mathfrak{R}_1^{-1}$ is a strong bisimulation
If f takes a step, $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$ then
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma(f \mid (g \mid h)) \xrightarrow{a} f' \mid \sigma(g \mid h) \equiv f' \mid (\sigma g \mid \sigma h)$
Also,
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma(f \mid g) \xrightarrow{a} f' \mid \sigma g$
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma((f \mid g) \mid h) \xrightarrow{a} (f' \mid \sigma g) \mid \sigma h$
But $\langle f' \mid (\sigma g \mid \sigma h), (f' \mid \sigma g) \mid \sigma h\rangle \in \mathfrak{R}$. Similarly if g or h takes a step □

Lemma 11. $\sigma f \sim \sigma g$ *when* $f \sim g$

Lemma 12. $f \sim (f$ **where** $x :\in \mathbf{0})$ *when* $x \notin f.v.(f)$

Lemma 13. *$\sim$ is a congruence relation*

Proof. We show appropriate bisimulations for all possible contexts:

a) $\mathfrak{R} = \{\, \langle f \mid h, g \mid h\rangle \mid f \sim g\}$ is a strong bisimulation.
If h takes a step, trivial.
If f takes a step, $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma(f \mid h) \xrightarrow{a} f' \mid \sigma h$
But $f \sim g$ so, $\Delta, \Gamma \vdash \sigma g \xrightarrow{a} g'$ and $f' \sim g'$
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma(g \mid h) \xrightarrow{a} g' \mid \sigma h$
and $\langle f' \mid \sigma h, g' \mid \sigma h\rangle \in \mathfrak{R}$
Similarly for the transitions of g.

b) $\mathfrak{R} = \{\, \langle h \mid f, h \mid g \rangle \mid f \sim g \,\}$ is a strong bisimulation.
As above.

c) $\mathfrak{R} = \{\, \langle (f >x> h) \mid d, (g >x> h) \mid d \rangle \mid f \sim g \}$
$\mathfrak{R}$ is a strong bisimulation up to $\sim$.
The only interesting case is when f publishes, $\Delta, \Gamma \vdash \sigma f \xrightarrow{!v} f'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(f >x> h) \xrightarrow{\tau} (f' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f >x> h) \mid d) \xrightarrow{\tau} ((f' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h) \mid \sigma d$
But $f \sim g$ so, $\Delta, \Gamma \vdash \sigma g \xrightarrow{!v} g'$ and $f' \sim g'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(g >x> h) \xrightarrow{\tau} (g' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((g >x> h) \mid d) \xrightarrow{\tau} ((g' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h) \mid \sigma d$
Then,

$((f' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h) \mid \sigma d$	$\sim$	by Lemma 10
$(f' >x> \sigma_{-x} h) \mid ([v/x]\sigma_{-x} h \mid \sigma d)$	$\mathfrak{R}$	
$(g' >x> \sigma_{-x} h) \mid ([v/x]\sigma_{-x} h \mid \sigma d)$	$\sim$	by Lemma 10
$((g' >x> \sigma_{-x} h) \mid [v/x]\sigma_{-x} h) \mid \sigma d$		

Similarly for g's transitions.
The desired result follows by Lemma 8 when $d \equiv \mathbf{0}$.

d) $\mathfrak{R} = \{\, \langle (h >x> f) \mid d_1, (h >x> g) \mid d_2 \rangle \mid f \sim g,\ d_1 \sim d_2 \}$
$\mathfrak{R}$ is a strong bisimulation up to $\sim$.
The only interesting case is when h publishes, $\Delta, \Gamma \vdash \sigma h \xrightarrow{!v} h'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(h >x> f) \xrightarrow{\tau} (h' >x> \sigma_{-x} f) \mid [v/x]\sigma_{-x} f$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((h >x> f) \mid d_1) \xrightarrow{\tau} ((h' >x> \sigma_{-x} f) \mid [v/x]\sigma_{-x} f) \mid \sigma d_1$
Also, $\Delta, \Gamma \vdash \sigma(h >x> g) \xrightarrow{\tau} (h' >x> \sigma_{-x} g) \mid [v/x]\sigma_{-x} g$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((h >x> g) \mid d_2) \xrightarrow{\tau} ((h' >x> \sigma_{-x} g) \mid [v/x]\sigma_{-x} g) \mid \sigma d_2$
By Lemma 11, $[v/x]\sigma_{-x} f \sim [v/x]\sigma_{-x} g$ and $\sigma d_1 \sim \sigma d_2$.
Then,

$((h' >x> \sigma_{-x} f) \mid [v/x]\sigma_{-x} f) \mid \sigma d_1$	$\sim$	by case b
$((h' >x> \sigma_{-x} f) \mid [v/x]\sigma_{-x} f) \mid \sigma d_2$	$\sim$	by Lemma 10
$(h' >x> \sigma_{-x} f) \mid ([v/x]\sigma_{-x} f \mid \sigma d_2)$	$\sim$	by cases a,b
$(h' >x> \sigma_{-x} f) \mid ([v/x]\sigma_{-x} g \mid \sigma d_2)$	$\mathfrak{R}$	
$(h' >x> \sigma_{-x} g) \mid ([v/x]\sigma_{-x} g \mid \sigma d_2)$	$\sim$	by Lemma 10
$((h' >x> \sigma_{-x} g) \mid [v/x]\sigma_{-x} g) \mid \sigma d_2$		

The desired result follows by Lemma 8 when $d_1 \equiv \mathbf{0}, d_2 \equiv \mathbf{0}$.

e) $\mathfrak{R} = \{\, \langle f \textbf{ where } x :\in h, g \textbf{ where } x :\in h \rangle \mid f \sim g \}$
$\mathfrak{R}$ is a strong bisimulation up to $\sim$.
The only interesting case is when h publishes, $\Delta, \Gamma \vdash \sigma h \xrightarrow{!v} h'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(f \textbf{ where } x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x} f$
Also, $\Delta, \Gamma \vdash \sigma(g \textbf{ where } x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x} g$

$[v/x]\sigma_{-x} f$	$\sim$	by Lemma 12
$[v/x]\sigma_{-x} f \textbf{ where } x :\in \mathbf{0}$	$\mathfrak{R}$	by Lemma 11
$[v/x]\sigma_{-x} g \textbf{ where } x :\in \mathbf{0}$	$\sim$	by Lemma 12
$[v/x]\sigma_{-x} g$		

f) $\mathfrak{R}_1 = \{\, \langle h \textbf{ where } x :\in f, h \textbf{ where } x :\in g \rangle \mid f \sim g \}$
$\mathfrak{R} = \mathfrak{R}_1 \cup \mathcal{ID}$ is a strong bisimulation.

If f publishes, $\Delta, \Gamma \vdash \sigma f \xrightarrow{!v} f'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(h \textbf{ where } x :\in f) \xrightarrow{\tau} [v/x]\sigma_{-x}h$
But $f \sim g$ so, $\Delta, \Gamma \vdash \sigma g \xrightarrow{!v} g'$ and $f' \sim g'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(h \textbf{ where } x :\in g) \xrightarrow{\tau} [v/x]\sigma_{-x}h$
and $\langle [v/x]\sigma_{-x}h, [v/x]\sigma_{-x}h\rangle \in \mathfrak{R}$
If f takes a non-publication step, $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(h \textbf{ where } x :\in f) \xrightarrow{a} \sigma_{-x}h \textbf{ where } x :\in f'$
But $f \sim g$ so, $\Delta, \Gamma \vdash \sigma g \xrightarrow{a} g'$ and $f' \sim g'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(h \textbf{ where } x :\in g) \xrightarrow{a} \sigma_{-x}h \textbf{ where } x :\in g'$
and $\langle \sigma_{-x}h \textbf{ where } x :\in f', \sigma_{-x}h \textbf{ where } x :\in g'\rangle \in \mathfrak{R}$ □

Lemma 14. $(f \mid g) >x> h \sim (f >x> h) \mid (g >x> h)$

Proof. $\mathfrak{R}_1 = \{\langle (((f \mid g) >x> h) \mid d, ((f >x> h) \mid (g >x> h)) \mid d\rangle\}$
$\mathfrak{R} = \mathfrak{R}_1 \cup {\mathfrak{R}_1}^{-1}$ is a strong bisimulation up to $\sim$
The only interesting case is when f publishes, $\Delta, \Gamma \vdash \sigma f \xrightarrow{!v} f'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(f \mid g) \xrightarrow{!v} f' \mid \sigma g$
$\Longrightarrow \Delta, \Gamma \vdash \sigma((f \mid g) >x> h) \xrightarrow{\tau} ((f' \mid \sigma g) >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(((f \mid g) >x> h) \mid d) \xrightarrow{\tau}$
$(((f' \mid \sigma g) >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h) \mid \sigma d$
Also,
$\Longrightarrow \Delta, \Gamma \vdash \sigma(f >x> h) \xrightarrow{\tau} (f' >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h$
$\Longrightarrow \Delta, \Gamma \vdash \sigma((f >x> h) \mid (g >x> h)) \xrightarrow{\tau}$
$((f' >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h) \mid (\sigma g >x> \sigma_{-x}h)$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(((f >x> h) \mid (g >x> h)) \mid d) \xrightarrow{\tau}$
$(((f' >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h) \mid (\sigma g >x> \sigma_{-x}h)) \mid \sigma d$
But then,
$(((f' >x> \sigma_{-x}h) \mid [v/x]\sigma_{-x}h) \mid (\sigma g >x> \sigma_{-x}h)) \mid \sigma d \sim$ by Lemmas 9,10
$((f' >x> \sigma_{-x}h) \mid (\sigma g >x> \sigma_{-x}h)) \mid ([v/x]\sigma_{-x}h \mid \sigma d) \quad \mathfrak{R}$
$((f' \mid \sigma g) >x> \sigma_{-x}h) \mid ([v/x]\sigma_{-x}h \mid \sigma d)$
The desired result follows by Lemma 8 when $d \equiv \mathbf{0}$. □

Lemma 15. $f >x> (g >y> h) \sim (f >x> g) >y> h \quad$ *if* $x \notin f.v.(h)$

Proof. $\mathfrak{R}_1 = \{\langle (f >x> (g >y> h)) \mid d, ((f >x> g) >y> h) \mid d\rangle\}$
$\mathfrak{R} = \mathfrak{R}_1 \cup {\mathfrak{R}_1}^{-1}$ is a strong bisimulation up to $\sim$ if $x \notin$ f.v.(h)
The only interesting case is when f publishes, $\Delta, \Gamma \vdash \sigma f \xrightarrow{!v} f'$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(f >x> (g >y> h)) \xrightarrow{\tau}$
$(f' >x> \sigma_{-x}(g >y> h)) \mid [v/x]\sigma_{-x}(g >y> h)$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(((f >x> g) >y> h) \mid d) \xrightarrow{\tau}$
$((f' >x> \sigma_{-x}(g >y> h)) \mid [v/x]\sigma_{-x}(g >y> h)) \mid \sigma d$
Also,
$\Longrightarrow \Delta, \Gamma \vdash \sigma(f >x> g) \xrightarrow{\tau} (f' >x> \sigma_{-x}g) \mid [v/x]\sigma_{-x}g$
$\Longrightarrow \Delta, \Gamma \vdash \sigma((f >x> g) >y> h) \xrightarrow{\tau} ((f' >x> \sigma_{-x}g) \mid [v/x]\sigma_{-x}g) >y> \sigma_{-y}h$
$\Longrightarrow \Delta, \Gamma \vdash \sigma(((f >x> g) >y> h) \mid d) \xrightarrow{\tau}$
$(((f' >x> \sigma_{-x}g) \mid [v/x]\sigma_{-x}g) >y> \sigma_{-y}h) \mid \sigma d$

By Lemma 14,
$(((f' >x> \sigma_{-x}g) \mid [v/x]\sigma_{-x}g) >y> \sigma_{-y}h) \mid \sigma d \quad \sim$
$(((f' >x> \sigma_{-x}g) >y> \sigma_{-y}h) \mid ([v/x]\sigma_{-x}g >y> \sigma_{-y}h)) \mid \sigma d \quad \mathfrak{R}$
$((f' >x> \sigma_{-x}(g >y> h)) \mid [v/x]\sigma_{-x}(g >y> h)) \mid \sigma d$ □

Lemma 16. $(f \mid g)$ **where** $x :\in h \sim (f$ **where** $x :\in h) \mid g$ *if* $x \notin$ *f.v.*(g)

Proof. $\mathfrak{R}_1 = \{\langle (f \mid g)$ **where** $x :\in h, (f$ **where** $x :\in h) \mid g\rangle\}$
$\mathfrak{R} = \mathfrak{R}_1 \cup {\mathfrak{R}_1}^{-1} \cup \mathcal{ID}$ is a strong bisimulation if $x \notin$ f.v.(g)
We know that x is not free in g, so for any σ, $\sigma_{-x}g \equiv \sigma g$ (I)
Let $\Delta, \Gamma \vdash \sigma_{-x}g \xrightarrow{a} g'$ (II)
By Lemma 51, 'a' is not a receive for x. Then,
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma_{-x}(f \mid g) \xrightarrow{a} \sigma_{-x}f \mid g'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f \mid g)$ **where** $x :\in h) \xrightarrow{a} (\sigma_{-x}f \mid g')$ **where** $x :\in \sigma h$
By $I, II \quad \Delta, \Gamma \vdash \sigma g \xrightarrow{a} g'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f$ **where** $x :\in h) \mid g) \xrightarrow{a} (\sigma_{-x}f$ **where** $x :\in \sigma h) \mid g'$
But by Lemma 50, $x \notin$ f.v.(g')
$\Longrightarrow \quad ((\sigma_{-x}f \mid g')$ **where** $x :\in \sigma h) \quad \mathfrak{R} \quad ((\sigma_{-x}f$ **where** $x :\in \sigma h) \mid g')$
The only interesting case left is when h publishes, $\Delta, \Gamma \vdash \sigma h \xrightarrow{!v} h'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f \mid g)$ **where** $x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x}(f \mid g)$
$\equiv [v/x]\sigma_{-x}f \mid \sigma_{-x}g$ because $x \notin$ f.v.(g)
Also,
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(f$ **where** $x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x}f$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f$ **where** $x :\in h) \mid g) \xrightarrow{\tau} [v/x]\sigma_{-x}f \mid \sigma g$
$\equiv [v/x]\sigma_{-x}f \mid \sigma_{-x}g$ because $x \notin$ f.v.(g)
And obviously, $\langle [v/x]\sigma_{-x}f \mid \sigma_{-x}g, [v/x]\sigma_{-x}f \mid \sigma_{-x}g\rangle \in \mathcal{ID}$ □

Lemma 17. $(f >y> g)$ **where** $x :\in h \sim (f$ **where** $x :\in h) >y> g$
if $x \notin$ *f.v.*(g)

Proof.
$\mathfrak{R}_1 = \{\langle (((f >y> g)$ **where** $x :\in h) \mid d, ((f$ **where** $x :\in h) >y> g) \mid d\rangle\}$
$\mathfrak{R} = \mathfrak{R}_1 \cup {\mathfrak{R}_1}^{-1} \cup \mathcal{ID}$ is a strong bisimulation up to $\sim$ if $x \notin$ f.v.(g)
We look only at the publication steps of h and f.
$\Delta, \Gamma \vdash \sigma h \xrightarrow{!v} h'$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f >y> g)$ **where** $x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x}(f >y> g)$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(((f >y> g)$ **where** $x :\in h) \mid d) \xrightarrow{\tau}$
$[v/x]\sigma_{-x}(f >y> g) \mid \sigma d$
Also,
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(f$ **where** $x :\in h) \xrightarrow{\tau} [v/x]\sigma_{-x}f$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((f$ **where** $x :\in h) >y> g) \xrightarrow{\tau} [v/x]\sigma_{-x}f >y> \sigma_{-y}g$
$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(((f$ **where** $x :\in h) >y> g) \mid d) \xrightarrow{\tau}$
$([v/x]\sigma_{-x}f >y> \sigma_{-y}g) \mid \sigma d$
But $x \notin$ f.v.(g) so
$\langle [v/x]\sigma_{-x}(f >y> g) \mid \sigma d, ([v/x]\sigma_{-x}f >y> \sigma_{-y}g) \mid \sigma d\rangle \in \mathcal{ID}$
If f publishes, $\Delta, \Gamma \vdash \sigma_{-x}f \xrightarrow{!v} f'$ then,

$\Longrightarrow \quad \Delta, \Gamma \vdash \ \sigma_{-x}(f >y> g) \xrightarrow{\tau} (f' >y> \sigma_{-x,-y}g) \mid [v/y]\sigma_{-x,-y}g$
$\Longrightarrow \quad \Delta, \Gamma \vdash \ \sigma((f >y> g) \textbf{ where } x :\in h) \xrightarrow{\tau}$
$\qquad\qquad ((f' >y> \sigma_{-x,-y}g) \mid [v/y]\sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h$
$\Longrightarrow \quad \Delta, \Gamma \vdash \ \sigma(((f >y> g) \textbf{ where } x :\in h) \mid d) \xrightarrow{\tau}$
$\qquad\qquad (((f' >y> \sigma_{-x,-y}g) \mid [v/y]\sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h) \mid \sigma d$

Also,

$\Delta, \Gamma \vdash \ \sigma(f \textbf{ where } x :\in h) \xrightarrow{!v} f' \textbf{ where } x :\in \sigma h$
$\Delta, \Gamma \vdash \ \sigma((f \textbf{ where } x :\in h) >y> g) \xrightarrow{\tau}$
$\qquad ((f' \textbf{ where } x :\in \sigma h) >y> \sigma_{-y}g) \mid [v/y]\sigma_{-y}g$
$\Delta, \Gamma \vdash \ \sigma(((f \textbf{ where } x :\in h) >y> g) \mid d) \xrightarrow{\tau}$
$\qquad (((f' \textbf{ where } x :\in \sigma h) >y> \sigma_{-y}g) \mid [v/y]\sigma_{-y}g) \mid \sigma d$

But $x \notin \text{f.v.}(g)$ so by Lemma 16
$((f' >y> \sigma_{-x,-y}g) \mid [v/y]\sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h \quad \sim$
$((f' >y> \sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h) \mid [v/y]\sigma_{-x,-y}g$
which by Lemma 13 yields
$(((f' >y> \sigma_{-x,-y}g) \mid [v/y]\sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h) \mid \sigma d \quad \sim$
$(((f' >y> \sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h) \mid [v/y]\sigma_{-x,-y}g) \mid \sigma d$
By Lemma 10, the last process is strongly bisimilar to
$((f' >y> \sigma_{-x,-y}g) \textbf{ where } x :\in \sigma h) \mid ([v/y]\sigma_{-x,-y}g \mid \sigma d) \quad \Re$
$((f' \textbf{ where } x :\in \sigma h) >y> \sigma_{-x,-y}g) \mid ([v/y]\sigma_{-x,-y}g \mid \sigma d) \quad \sim$
$(((f' \textbf{ where } x :\in \sigma h) >y> \sigma_{-y}g) \mid [v/y]\sigma_{-y}g) \mid \sigma d$
The desired result follows by Lemma 8 when $d \equiv \mathbf{0}$. □

Lemma 18. $(f \textbf{ where } x :\in g) \textbf{ where } y :\in h \sim (f \textbf{ where } y :\in h) \textbf{ where } x :\in g$
if $x \notin f.v.(h), y \notin f.v.(g)$

Proof. The proof is similar to the previous proofs □

Lemma 19. *If* $\Gamma = \{\langle x_1, v_1\rangle, \ldots, \langle x_m, v_m\rangle\}, \quad \sigma = [w_1/y_1]\ldots[w_n/y_n]$,
x's and y's are all distinct, then

$$f \sim g \ \Rightarrow \ (\Delta, \Gamma \vdash \ \sigma f \xrightarrow{t}{}^{*} f' \ \Leftrightarrow \ \Delta, \Gamma \vdash \ \sigma g \xrightarrow{t}{}^{*} g')$$

Proof. By induction on $|t|$ □

Theorem 6. *If* $f \sim g$ *then for any* ρ *it holds that* $[\![f]\!][\![\Delta]\!]\rho = [\![g]\!][\![\Delta]\!]\rho$

Proof. Let $t \in [\![f]\!][\![\Delta]\!]\rho$
$\Longrightarrow \quad \Delta, \Gamma \vdash \ \sigma f \xrightarrow{t}{}^{*} f'$ by Theorem 10
$\Longrightarrow \quad \Delta, \Gamma \vdash \ \sigma g \xrightarrow{t}{}^{*} g'$ by Lemma 19
$\Longrightarrow \quad t \in [\![g]\!][\![\Delta]\!]\rho$ by Theorem 9
$\Longrightarrow \quad [\![f]\!][\![\Delta]\!]\rho \subseteq [\![g]\!][\![\Delta]\!]\rho$
In the same way we get $[\![g]\!][\![\Delta]\!]\rho \subseteq [\![f]\!][\![\Delta]\!]\rho$ □

C Continuity Proofs

Lemma 20. *The union of prefix-closed sets is prefix-closed* □

Lemma 21. *P is a CPO under inclusion*

Proof. Let $X \subseteq P$ be directed and $B = \bigcup_{S \in X} S$. Then, B is prefix-closed by Lemma 20 and is an ub of X. Let B' be an ub of X
$\implies \forall S \in X. S \subseteq B'$
$\implies \bigcup_{S \in X} S \subseteq B'$
$\implies \bigsqcup X = B$ □

Lemma 22. *Merge : $Pow(Traces) \times Pow(Traces) \to Pow(Traces)$ is continuous*

Proof. It suffices to show that it is continuous in each argument separately. Let $X \subseteq Pow(Traces)$ be directed, $T \in Pow(Traces)$
$(\bigsqcup X) \parallel T = (\bigcup_{S \in X} S) \parallel T$
$\triangleq \bigcup_{s \in (\bigcup_{S \in X} S)} \bigcup_{t \in T} s \parallel t$
$= \bigcup_{S \in X} \bigcup_{s \in S} \bigcup_{t \in T} s \parallel t$
$\triangleq \bigcup_{S \in X} (S \parallel T)$
$= \bigsqcup_{S \in X} (S \parallel T)$
The proof is similar for the right argument □

Lemma 23. *Extend-env is continuous* □

Note 24 *$[Val \to NoRecv]$ is a CPO and if $X \subseteq [Val \to NoRecv]$ is directed, then $\bigsqcup X = \lambda v. \bigsqcup_{f \in X} f(v) = \lambda v. \bigcup_{f \in X} f(v)$*

Note 25 *Fenv is a CPO and if $X \subseteq Fenv$ is directed, then*
$\bigsqcup X = (\lambda v. \bigcup_{\varphi \in X} \varphi_1(v)) \times \cdots \times (\lambda v. \bigcup_{\varphi \in X} \varphi_k(v))$

Note 26 *Similar results to Note 24 hold for $[Val \to P]$, $[Val \to Pow(Traces)]$*

Lemma 27. *$\gg$: $Traces \times [Val \to Pow(Traces)] \to Pow(Traces)$ is continuous*

Proof. Show continuity in each argument separately. Over the left argument it is trivial, since *Traces* is a discrete CPO.
Over the right argument:
Let $X \subseteq [Val \to Pow(Traces)]$ be directed and $s \in Traces$
Proceed by induction on the number of publications in s
If no publications in s,
$\implies s \gg \bigsqcup X = \{s\} = \bigsqcup_{F \in X} (s \gg F)$
If $s \equiv s_1 ! v s_2$ and no publications in s_1,
$s \gg \bigsqcup X = s_1 \tau ((s_2 \gg \bigsqcup X) \parallel \bigcup_{F \in X} F(v))$ by Note 26
$= s_1 \tau ((\bigcup_{F \in X} s_2 \gg F) \parallel \bigcup_{F \in X} F(v))$ by *IH*
$= s_1 \tau \bigcup_{F \in X} ((s_2 \gg F) \parallel F(v))$ by Lemma 22
$= \bigcup_{F \in X} s_1 \tau ((s_2 \gg F) \parallel F(v))$
$= \bigsqcup_{F \in X} s \gg F$ □

Corollary 1. *Let* $S \in Pow(Traces)$ *and* $F \in [Val \to Pow(Traces)]$. *Then,* $\bigcup_{s \in S} s \gg F$ *is continuous* □

Lemma 28. *Prefixing* : $Pow(Traces) \to P$ *is continuous* □

Note 29 $<_x$: $Traces \times Traces \to Pow(Traces)$ *is continuous* □

Corollary 2. $<_x$: $Pow(Traces) \times Pow(Traces) \to Pow(Traces)$ *is continuous* □

Note 30 *All the functions proved to be continuous are also monotonic*

Theorem 7. *For all* f, $[\![f]\!]$ *is continuous*

Proof. We know that $[\![f]\!] \in [Fenv \to [Env \to P]]$. We will show the continuity of $[(Fenv \times Env) \to P]$ and this is enough because currying is a continuous operation.
By structural induction on f.
Let X_φ, X_ρ be directed subsets of $Fenv$ and Env respectively.

a) $let(v)$
$\implies [\![f]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = \{!v\}_{\mathrm{p}} = \bigsqcup_{\varphi \in X_\varphi} \bigsqcup_{\rho \in X_\rho} [\![let(v)]\!]\varphi\rho$

b) $\mathbf{0}$ or $M(v)$ or $?k$
as above

c) $let(x)$
$[\![let(x)]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = \bigsqcup_{\varphi \in X_\varphi} [\![let(x)]\!]\varphi(\bigsqcup X_\rho)$ $(c1)$
Cases on X_ρ:
- If $\exists \rho \in X_\rho.\ \rho(x) =$ Absent then $\forall \rho \in X_\rho.\ \rho(x) =$ Absent because X_ρ is directed.
  $(c1) \Rightarrow \bigsqcup_{\varphi \in X_\varphi} \{\varepsilon\} = \bigsqcup_{\varphi \in X_\varphi} \bigsqcup_{\rho \in X_\rho} [\![let(x)]\!]\varphi\rho$
- If $\exists \rho \in X_\rho.\ \rho(x) = \flat v$ then $\forall \rho \in X_\rho.\ \rho(x) = \flat v$ because X_ρ is directed.
  $(c1) \Rightarrow \bigsqcup_{\varphi \in X_\varphi} \{!v\}_{\mathrm{p}} = \bigsqcup_{\varphi \in X_\varphi} \bigsqcup_{\rho \in X_\rho} [\![let(x)]\!]\varphi\rho$
- If $\exists \rho \in X_\rho.\ \rho(x) = \natural v$ similarly

d) $M(x)$
as above

e) $E_i(v)$
$[\![E_i(v)]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) =$
$= \bigsqcup_{\rho \in X_\rho} [\![E_i(v)]\!](\bigsqcup X_\varphi)\rho$
$= \bigsqcup_{\rho \in X_\rho} \{\, \tau\ t \mid t \in (\bigsqcup X_\varphi)_i(v) \}_{\mathrm{p}}$
$= \bigsqcup_{\rho \in X_\rho} \{\, \tau\ t \mid t \in \bigcup_{\varphi \in X_\varphi} \varphi_i(v) \}_{\mathrm{p}}$ by Note 26
$= \bigsqcup_{\rho \in X_\rho} \bigcup_{\varphi \in X_\varphi} \{\, \tau\ t \mid t \in \varphi_i(v) \}_{\mathrm{p}}$ by Lemma 28
$= \bigsqcup_{\varphi \in X_\varphi} \bigsqcup_{\rho \in X_\rho} [\![E_i(v)]\!]\varphi\rho$

f) $E_i(x)$
Cases on $\bigsqcup X_\rho$ and similar to the previous case

g) $h \mid g$

$[\![h \mid g]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = [\![h]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) \parallel [\![g]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho)$

$(\bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h]\!]\varphi\rho) \parallel (\bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![g]\!]\varphi\rho)$ by *IH*

$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} ([\![h]\!]\varphi\rho \parallel [\![g]\!]\varphi\rho)$ by Lemma 22

$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h \mid g]\!]\varphi\rho$

h) $h >x> g$

$[\![g]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho)[x = \flat v] =$

$= [\![g]\!](\bigsqcup X_\varphi)(\bigsqcup_{\rho\in X_\rho} \rho[x = \flat v])$ by Lemma 23

$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![g]\!]\varphi\rho[x = \flat v]$ by *IH*

Then, by Note 26,

$\bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} \lambda v.[\![g]\!]\varphi\rho[x = \flat v] = \lambda v. \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![g]\!]\varphi\rho[x = \flat v]$ $(h1)$

Also, $[\![h]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h]\!]\varphi\rho$ by *IH* $(h2)$

By $h1$, $h2$ and Corollary 1 we get the result

i) h **where** $x :\in g$

By Lemma 23 and *IH*,

$[\![h]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho)[x = \natural v] = \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h]\!]\varphi\rho[x = \natural v]$

$\Longrightarrow \quad \bigcup_{v\in Val} \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h]\!]\varphi\rho[x = \natural v] =$

$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} \bigcup_{v\in Val} [\![h]\!]\varphi\rho[x = \natural v]$

By this, *IH* for g and Corollary 2 we get the result □

D Prefix-Closure Proofs

Lemma 31. $t_1 \parallel t_2 = t_1 \mathbin{\breve{\parallel}} t_2$

Proof. By induction on $|t_1| + |t_2|$.
The only interesting case is when $|t_1| \geq 2$ and $|t_2| \geq 2$ i.e. $t_1 = a_1 t_1' a_2$ and $t_2 = b_1 t_2' b_2$
$\implies t_1 \parallel t_2 = a_1(t_1' a_2 \parallel t_2) \cup b_1(t_1 \parallel t_2' b_2)$
$= a_1(t_1' a_2 \mathbin{\breve{\parallel}} t_2) \cup b_1(t_1 \mathbin{\breve{\parallel}} t_2' b_2)$ by *IH*
$= a_1((t_1' \mathbin{\breve{\parallel}} t_2)a_2 \cup (t_1' a_2 \mathbin{\breve{\parallel}} b_1 t_2')b_2) \cup b_1((a_1 t_1' \mathbin{\breve{\parallel}} t_2' b_2)a_2 \cup (t_1 \mathbin{\breve{\parallel}} t_2')b_2)$
$= a_1(t_1' \mathbin{\breve{\parallel}} t_2)a_2 \cup a_1(t_1' a_2 \mathbin{\breve{\parallel}} b_1 t_2')b_2 \cup b_1(a_1 t_1' \mathbin{\breve{\parallel}} t_2' b_2)a_2 \cup b_1(t_1 \mathbin{\breve{\parallel}} t_2')b_2$
$= (a_1(t_1' \mathbin{\breve{\parallel}} t_2) \cup b_1(a_1 t_1' \mathbin{\breve{\parallel}} t_2' b_2))a_2 \cup (a_1(t_1' a_2 \mathbin{\breve{\parallel}} b_1 t_2') \cup b_1(t_1 \mathbin{\breve{\parallel}} t_2'))b_2$
$= (a_1(t_1' \parallel t_2) \cup b_1(a_1 t_1' \parallel t_2' b_2))a_2 \cup (a_1(t_1' a_2 \parallel b_1 t_2') \cup b_1(t_1 \parallel t_2'))b_2$ by *IH*
$= (a_1 t_1' \parallel t_2)a_2 \cup (t_1 \parallel b_1 t_2')b_2$
$= (a_1 t_1' \mathbin{\breve{\parallel}} t_2)a_2 \cup (t_1 \mathbin{\breve{\parallel}} b_1 t_2')b_2$ by *IH*
$= t_1 \mathbin{\breve{\parallel}} t_2$ □

By this lemma, we can use the operators $\parallel$ and $\mathbin{\breve{\parallel}}$ interchangeably.

Lemma 32. *$T_1, T_2 \in P$ implies $T_1 \parallel T_2 \in P$*

Proof. By Lemma 31, suffices to show that $T_1 \mathbin{\breve{\parallel}} T_2 \in P$, i.e. suffices to show that for all $t \in T_1 \mathbin{\breve{\parallel}} T_2$, $t_\mathrm{p} \subseteq T_1 \mathbin{\breve{\parallel}} T_2$
By induction on $|t|$
Since $t \in T_1 \mathbin{\breve{\parallel}} T_2$, then $\exists t_1 \in T_1, t_2 \in T_2.\ t \in t_1 \mathbin{\breve{\parallel}} t_2$ (1)
The only interesting case is when $|t| \geq 2$ and $t_1 = t_1' a$ and $t_2 = t_2' b$
$\implies t \in ((t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)$
$\implies t_\mathrm{p} \subseteq ((t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)_\mathrm{p}$
$\implies t_\mathrm{p} \subseteq ((t_1' \mathbin{\breve{\parallel}} t_2)_\mathrm{p} \cup (t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')_\mathrm{p} \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)$ (2)
But $T_1 \in P \Rightarrow t_1' \in T_1$ and $T_2 \in P \Rightarrow t_2' \in T_2$
$\implies$ by *IH*, $(t_1' \mathbin{\breve{\parallel}} t_2)_\mathrm{p} \subseteq T_1 \mathbin{\breve{\parallel}} T_2$ and $(t_1 \mathbin{\breve{\parallel}} t_2')_\mathrm{p} \subseteq T_1 \mathbin{\breve{\parallel}} T_2$
$\implies$ by 2, suffices to show that $((t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b) \subseteq T_1 \mathbin{\breve{\parallel}} T_2$
i.e. that $t_1 \mathbin{\breve{\parallel}} t_2 \subseteq T_1 \mathbin{\breve{\parallel}} T_2$ which holds by 1 □

Lemma 33. *If $F \in [Val \to P]$ and $s \in Traces$, then $(\bigcup_{s' \in s_\mathrm{p}} s' \gg F) \in P$*

Proof. By induction on the number of publications in s.
If no publications in s,
$\implies \bigcup_{s' \in s_\mathrm{p}} s' \gg F = \bigcup_{s' \in s_\mathrm{p}} \{s'\} = s_\mathrm{p} \in P$
If $s = s_1 ! v s_2$ and no publications in s_1,
$\implies \bigcup_{s' \in s_\mathrm{p}} s' \gg F = (\bigcup_{s' \in (s_1)_\mathrm{p}} s' \gg F) \cup (s_1 ! v \gg F) \cup (\bigcup_{s' \in s_1 ! v (s_2)_\mathrm{p}} s' \gg F)$
$= (s_1)_\mathrm{p} \cup \{s_1 \tau\} \cup s_1 \tau((\bigcup_{s' \in (s_2)_\mathrm{p}} s' \gg F) \parallel F(v))$
$= \{s_1 \tau\}_\mathrm{p} \cup s_1 \tau((\bigcup_{s' \in (s_2)_\mathrm{p}} s' \gg F) \parallel F(v))$
$\implies$ suffices to show that $((\bigcup_{s' \in (s_2)_\mathrm{p}} s' \gg F) \parallel F(v)) \in P$
which, by Lemma 32, follows by $(\bigcup_{s' \in (s_2)_\mathrm{p}} s' \gg F) \in P$ and $F(v) \in P$, which holds by *IH* for s_2

Corollary 3. *If $T \in P$ and $F \in [Val \to P]$, then $(\bigcup_{s \in T} s \gg F) \in P$* □

Lemma 34. *$T_1, T_2 \in P$ implies $T_1 <_x T_2 \in P$*

Proof. If $t \in T_1 <_x T_2$ then $\exists t_1 \in T_1, t_2 \in T_2.\ t \in t_1 <_x t_2$
We must show that $t_\mathrm{p} \subseteq T_1 <_x T_2$.
Cases depending on which branch of the definition of $<_x$ was used

a) $t \in t_1 \parallel t_2$, no recv. for x in t_1, no publ. in t_2 (1)
$\implies$ $t \in \bigcup_{t_1' \in (t_1)_\mathrm{p}, t_2' \in (t_2)_\mathrm{p}} t_1' \parallel t_2' = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$ by Note 30
$\implies$ $t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p})_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$ by Lemma 32
By 1, $(t_1)_\mathrm{p} <_x (t_2)_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$
$\implies$ $t_\mathrm{p} \subseteq (t_1)_\mathrm{p} <_x (t_2)_\mathrm{p}$
$\implies$ $t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 30

b) $t \in t_1 \parallel t_{21}\tau$, no recv. for x in t_1, $t_2 = t_{21}!v\, t_{22}$, no publ. in t_{21}
$\implies$ $t \in (t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p}$ by Note 30
$\implies$ $t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p})_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p}$ by Lemma 32
$\implies$ $t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_{21})_\mathrm{p}) \cup ((t_1)_\mathrm{p} \parallel \{t_{21}\tau\})$
$\implies$ $t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} <_x (t_{21})_\mathrm{p}) \cup ((t_1)_\mathrm{p} <_x \{t_{21}!v\})$
$\implies$ $t_\mathrm{p} \subseteq (t_1)_\mathrm{p} <_x (t_{21}!v)_\mathrm{p}$
$\implies$ $t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 30

c) $t \in (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])$, $t_1 = t_{11}[v/x]t_{12}$, no recv. for x in t_{11},
$t_2 = t_{21}!v\, t_{22}$, no publ. in t_{21}
$\implies$ $t_\mathrm{p} \in (t_{11} \parallel t_{21}\tau)_\mathrm{p} \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$
$\implies$ $t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} \parallel \{t_{21}\tau\}_\mathrm{p})_\mathrm{p} \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$ by Note 30
$\implies$ $t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} \parallel \{t_{21}\tau\}_\mathrm{p}) \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$ by Lemma 32
By the previous case, this can be written
$\implies$ $t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}) \cup (t_{11}[v/x]\{t_{12}\}_\mathrm{p} <_x \{t_{21}!v\})$
$\implies$ $t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}) \cup (t_{11}[v/x]\{t_{12}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p})$ by Note 30
$\implies$ $t_\mathrm{p} \subseteq \{t_1\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}$
$\implies$ $t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 30 □

Theorem 8. *For all f, $[\![f]\!]\varphi\rho \in P$*

Proof. By structural induction on f, using Lemmas 32, 34 and Corollary 3 □

E Denotational Lemmas

Lemma 35 (Weakening). *If x not free in f then $[\![f]\!]\varphi\rho = [\![f]\!]\varphi\rho[x = \flat v] = [\![f]\!]\varphi\rho[x = \natural w]$ for any v, w*

Proof. By structural induction on f

a) If f is $\mathbf{0}$, $let(v)$, $M(v)$, $?k$, $E_i(v)$ it holds because the traces are independent of the environment
b) If f is $let(y)$, $M(y)$, $E_i(y)$, it holds because the traces depend only on y
c) $f \equiv h \mid g$
   By *IH*, $[\![h]\!]\varphi\rho = [\![h]\!]\varphi\rho[x = \flat v] = [\![h]\!]\varphi\rho[x = \natural w]$
   and $[\![g]\!]\varphi\rho = [\![g]\!]\varphi\rho[x = \flat v] = [\![g]\!]\varphi\rho[x = \natural w]$
   Therefore, $[\![h \mid g]\!]\varphi\rho = [\![h]\!]\varphi\rho \parallel [\![g]\!]\varphi\rho = [\![h \mid g]\!]\varphi\rho[x = \flat v] =$
   $= [\![h \mid g]\!]\varphi\rho[x = \natural w]$
d) $f \equiv h >x> g$ (Similarly when $f \equiv h >y> g, x \neq y$)
   By the statement of the lemma, x is not free in h
   $\Longrightarrow$ $[\![h]\!]\varphi\rho = [\![h]\!]\varphi\rho[x = \flat v] = [\![h]\!]\varphi\rho[x = \natural w]$ by *IH* $(d1)$
   Then, $[\![h >x> g]\!]\varphi\rho[x = \natural w] = \bigcup_{s \in [\![h]\!]\varphi\rho[x=\natural w]} s \gg \lambda v.[\![g]\!]\varphi\rho[x = \natural w][x = \flat v]$
   $= \bigcup_{s \in [\![h]\!]\varphi\rho} s \gg \lambda v.[\![g]\!]\varphi\rho[x = \flat v]$ by $d1$ and def. of extend-env
   $= [\![h >x> g]\!]\varphi\rho$
   Similarly, $[\![h >x> g]\!]\varphi\rho[x = \flat v] = [\![h >x> g]\!]\varphi\rho$
e) $f \equiv h$ **where** $x :\in g$ (Similarly when $f \equiv h$ **where** $y :\in g, x \neq y$)
   By the statement of the lemma, x is not free in g
   $\Longrightarrow$ $[\![g]\!]\varphi\rho = [\![g]\!]\varphi\rho[x = \flat v] = [\![g]\!]\varphi\rho[x = \natural w]$ by *IH* $(e1)$
   Then, $[\![h$ **where** $x :\in g]\!]\varphi\rho[x = \natural w] =$
   $= (\bigcup_{v \in Val} [\![h]\!]\varphi\rho[x = \natural w][x = \natural v]) <_x [\![g]\!]\varphi\rho[x = \natural w]$
   $= (\bigcup_{v \in Val} [\![h]\!]\varphi\rho[x = \natural v]) <_x [\![g]\!]\varphi\rho$ by $e1$ and def. of extenv-env
   $= [\![h$ **where** $x :\in g]\!]\varphi\rho$
   Similarly, $[\![h$ **where** $x :\in g]\!]\varphi\rho[x = \flat v] = [\![h$ **where** $x :\in g]\!]\varphi\rho$ □

Lemma 36 (Substitution). $[\![[v/x]f]\!]\varphi\rho = [\![f]\!]\varphi\rho[x = \flat v]$

Proof. By structural induction on f

a) If x not free in f then $[v/x]f = f$ and the result holds by Lemma 35
b) $f \equiv let(x)$
   $\Longrightarrow$ $[v/x]f = let(v)$
   $\Longrightarrow$ $[\![[v/x]f]\!]\varphi\rho = \{!v\}_{\mathrm{p}} = [\![let(x)]\!]\varphi\rho[x = \flat v]$
c) f is $M(x)$ or $E_i(x)$, as above
d) $f \equiv h \mid g$
   $[\![[v/x]f]\!]\varphi\rho = [\![[v/x]h]\!]\varphi\rho \parallel [\![[v/x]g]\!]\varphi\rho$
   $= [\![h]\!]\varphi\rho[x = \flat v] \parallel [\![g]\!]\varphi\rho[x = \flat v]$ by *IH*
   $= [\![f]\!]\varphi\rho[x = \flat v]$
e) $f \equiv h >x> g$ (Similarly when $f \equiv h >y> g, x \neq y$)
   $[\![[v/x]f]\!]\varphi\rho = [\![([v/x]h) >x> g]\!]\varphi\rho$
   $= \bigcup_{s \in [\![[v/x]h]\!]\varphi\rho} s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat w]$
   $= \bigcup_{s \in [\![h]\!]\varphi\rho[x=\flat v]} s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat v][x = \flat w]$ by *IH*
   $= [\![h >x> g]\!]\varphi\rho[x = \flat v]$

f) $f \equiv h$ **where** $x :\in g$ (Similarly when $f \equiv h$ **where** $y :\in g, x \neq y$)
$[\![[v/x]f]\!]\varphi\rho = [\![h \textbf{ where } x :\in [v/x]g]\!]\varphi\rho$
$= \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \natural w] <_x [\![[v/x]g]\!]\varphi\rho$
$= \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \flat v][x = \natural w] <_x [\![g]\!]\varphi\rho[x = \flat v]$ by *IH*
$= [\![h \textbf{ where } x :\in g]\!]\varphi\rho[x = \natural v]$ □

Lemma 37. *If* $t \in [\![f]\!]\varphi\rho$ *and* $[v/x]\hat{\in} t$ *then* $\rho(x) = \natural v$ □

Corollary 4. *If* $t \in [\![f]\!]\varphi\rho$, $[v/x]\hat{\in} t$ *and* $v \neq w$ *then* $[w/x] \hat{\notin} t$ □

Lemma 38. *If* $t \in [\![f]\!]\varphi\rho[x = \natural v]$ *and* $[v/x] \hat{\notin} t$ *then* $t \in [\![f]\!]\varphi\rho$

Proof. By structural induction on f
If x not free in f, it holds by Lemma 35. If x is free in f,

a) $f \equiv let(x)$
$\Longrightarrow \quad [\![let(x)]\!]\varphi\rho[x = \natural v] = \{[v/x]\,!v\}_{\mathrm{p}}$
$[v/x] \hat{\notin} t \;\; \therefore \;\; t = \varepsilon \;\; \therefore \;\; t \in [\![let(x)]\!]\varphi\rho$
b) f is $M(x)$ or $E_i(x)$, as above
c) $f \equiv h \mid g$
If $t \in [\![h \mid g]\!]\varphi\rho[x = \natural v]$ then there exist $t_1 \in [\![h]\!]\varphi\rho[x = \natural v]$,
$t_2 \in [\![g]\!]\varphi\rho[x = \natural v]$ such that $t \in t_1 \parallel t_2$.
But $[v/x] \hat{\notin} t$, so $[v/x] \hat{\notin} t_1$ and $[v/x] \hat{\notin} t_2$
$\Longrightarrow$ by *IH* $t_1 \in [\![h]\!]\varphi\rho$ and $t_2 \in [\![g]\!]\varphi\rho$
$\Longrightarrow \quad t \in [\![h \mid g]\!]\varphi\rho$
d) $f \equiv h >x> g$ (Similarly when $f \equiv h >y> g, x \neq y$)
If $t \in [\![h >x> g]\!]\varphi\rho[x = \natural v]$ then there exists $s \in [\![h]\!]\varphi\rho[x = \natural v]$ such that
$t \in s \gg \lambda w.[\![g]\!]\varphi\rho[x = \natural v][x = \flat w]$
$\Longrightarrow \quad t \in s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat w]$ ($d1$)
By Lemma 37, $[v/x]$ not in the traces of $[\![g]\!]\varphi\rho[x = \flat w]$
$\Longrightarrow$ $[v/x] \hat{\notin} t$ means $[v/x] \hat{\notin} s$
$\Longrightarrow$ by *IH* $s \in [\![h]\!]\varphi\rho$, so by $d1$ we get the desired result
e) $f \equiv h$ **where** $x :\in g$ (Similarly when $f \equiv h$ **where** $y :\in g, x \neq y$)
If $t \in [\![h \textbf{ where } x :\in g]\!]\varphi\rho[x = \natural v]$ then there exist
$t_1 \in \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \natural w]$, $t_2 \in [\![g]\!]\varphi\rho[x = \natural v]$ such that $t \in t_1 <_x t_2$ ($e1$)
Cases depending on which branch of the definition of $<_x$ was used:
We consider only one case, the others are similar.
$t_1 = t_{11}[u/x]\,t_{12}$, $[u/x] \hat{\notin} t_{11}$ and
$t_2 = t_{21}!u\,t_{22}$, $!u' \hat{\notin} t_{21}$ for any u'
$\Longrightarrow \quad t \in (t_{11} \parallel t_{21}\tau)(t_{12} \backslash [u/x])$ ($e2$)
But then, $[v/x] \hat{\notin} t$ means $[v/x] \hat{\notin} t_{21}$
and by Theorem 8, $t_{21}\,!u \in [\![g]\!]\varphi\rho[x = \natural v]$
$\Longrightarrow \quad t_{21}\,!u \in [\![g]\!]\varphi\rho$ by *IH* ($e3$)
By $e1$, $e2$ and $e3$ we get the desired result □

Lemma 39. *If* $\rho(x) =$ Absent *then* $[\![f]\!]\varphi\rho \subseteq [\![f]\!]\varphi\rho[x = \natural v]$ □

Lemma 40. *If* $\rho(x) =$ Absent *then* $[\![f]\!]\varphi\rho \subseteq [\![f]\!]\varphi\rho[x = \flat v]$ □

Lemma 41. $(t_1 \parallel t_2)\backslash a = t_1\backslash a \parallel t_2\backslash a$

Proof. By induction on $|t_1| + |t_2|$
The interesting case is when $|t_1| + |t_2| \geq 2$ and $t_1 = bt_1'$, $t_2 = ct_2'$
Then, $(t_1 \parallel t_2)\backslash a = (b(t_1' \parallel t_2) \cup c(t_1 \parallel t_2'))\backslash a$
$= (b(t_1' \parallel t_2))\backslash a \cup (c(t_1 \parallel t_2'))\backslash a$
If $b \neq a$ and $c \neq a$ the above becomes
$= b(t_1' \parallel t_2)\backslash a \cup c(t_1 \parallel t_2')\backslash a$
$= b(t_1'\backslash a \parallel t_2\backslash a) \cup c(t_1\backslash a \parallel t_2'\backslash a)$ by *IH*
$= t_1\backslash a \parallel t_2\backslash a$
Similarly when b and/or c is equal to a □

Corollary 5. $(T_1 \parallel T_2)\backslash a = T_1\backslash a \parallel T_2\backslash a$ □

Lemma 42. *Let $s \in$ Traces and F : Val $\rightarrow$ Pow(Traces).*
Then, $(s \gg F)\backslash[v/x] = s\backslash[v/x] \gg \lambda w.F(w)\backslash[v/x]$

Proof. By induction on the number of publications in s
If no publ. in s then $(s \gg F)\backslash[v/x] = \{s\}\backslash[v/x] = s\backslash[v/x] \gg \lambda w.F(w)\backslash[v/x]$
If $s = s_1!us_2$ and no publ. in s_1 then
$(s \gg F)\backslash[v/x] = (s_1\tau)\backslash[v/x]((s_2 \gg F) \parallel F(u))\backslash[v/x]$
$= (s_1\tau)\backslash[v/x]((s_2 \gg F)\backslash[v/x] \parallel F(u)\backslash[v/x])$ by Corollary 5
$= (s_1\tau)\backslash[v/x]((s_2\backslash[v/x] \gg \lambda w.F(w)\backslash[v/x]) \parallel F(u)\backslash[v/x])$ by *IH* for s_2
$= (s_1!us_2)\backslash[v/x] \gg \lambda w.F(w)\backslash[v/x]$
$= s\backslash[v/x] \gg \lambda w.F(w)\backslash[v/x]$ □

Lemma 43. *Let $s \in$ Traces and F : Val $\rightarrow$ Pow(Traces).*
Then, $(s \gg F)\backslash\tau = s\backslash\tau \mathbin{\dot{\gg}} \lambda w.F(w)\backslash\tau$ □

Lemma 44. *For all s such that $s\backslash\tau = s$ we get $s \in s \mathbin{\dot{\gg}} \lambda v.\{!v\}_{\mathrm{p}}$*

Proof. Straightforward induction on the number of publications in s. □

Lemma 45. $(t_1 <_y t_2)\backslash[v/x] = t_1\backslash[v/x] <_y t_2\backslash[v/x]$, *when* $y \neq x$ *and* $(t_1 <_x t_2)\backslash[v/x] = t_1 <_x t_2\backslash[v/x]$

Proof. Assume a well-formedness constraint for t_1, t_2 similar to Corollary 4.
Cases depending on which branch of the definition of $<_x$ was used:

a) no recv. for x in t_1, no publ. in t_2, $t_1 <_y t_2 = t_1 \parallel t_2$
   $\Longrightarrow$ holds by Lemma 41
b) no recv. for x in t_1, $t_2 = t_{21}!w\, t_{22}$, no publ. in t_{21} $t_1 <_y t_2 = t_1 \parallel t_{21}\, \tau$
   $\Longrightarrow$ holds by Lemma 41
c) $t_1 = t_{11}[w/y]\, t_{12}$, $[w/y] \tilde{\notin} t_{11}$, $t_2 = t_{21}!w\, t_{22}$, no publ. in t_{21},
   $t_1 <_y t_2 = (t_{11} \parallel t_{21}\, \tau)(t_{12}\backslash[w/y])$ (*c*1)
   When $x \neq y$, by $c1 \Rightarrow ((t_{11} \parallel t_{21}\, \tau)(t_{12}\backslash[w/y]))\backslash[v/x]$
   $= (t_{11} \parallel t_{21}\, \tau)\backslash[v/x]\ (t_{12}\backslash[w/y])\backslash[v/x]$
   $= (t_{11}\backslash[v/x] \parallel (t_{21}\, \tau)\backslash[v/x])\ (t_{12}\backslash[w/y])\backslash[v/x]$ by Corollary 5

$= t_1\backslash[v/x] <_y t_2\backslash[v/x]$
When $x = y$, by $c1 \Rightarrow ((t_{11} \parallel t_{21}\ \tau)(t_{12}\backslash[w/x]))\backslash[v/x]$
$= (t_{11} \parallel t_{21}\ \tau)\backslash[v/x]\ (t_{12}\backslash[w/x])\backslash[v/x]$ $(c2)$
By the well-formedness constraint, $[v/x] \hat{\notin} t_{12}$ when $v \neq w$,
therefore $(t_{12}\backslash[w/x])\backslash[v/x] = t_{12}\backslash[w/x]$
$(c2) \Rightarrow (t_{11} \parallel (t_{21}\ \tau)\backslash[v/x])\ (t_{12}\backslash[w/x])$
$= t_1 <_x t_2\backslash[v/x]$ □

Lemma 46. $(t_1 <_x t_2)\backslash\tau = t_1\backslash\tau \mathbin{\dot{<}}_x t_2\backslash\tau$ □

Lemma 47. $[\![f]\!]\varphi\rho[x = \flat v] = ([\![f]\!]\varphi\rho[x = \natural v])\backslash[v/x]$

Proof. By structural induction on f
If x is not free in f, by Lemma 35 we get
$[\![f]\!]\varphi\rho = [\![f]\!]\varphi\rho[x = \natural v] = [\![f]\!]\varphi\rho[x = \flat v]$ (I)
But $\rho(x) = \flat v$ in $[\![f]\!]\varphi\rho[x = \flat v]$, so by (the contrapositive of) Lemma 37 we know that $[v/x]$ is not in the traces of $[\![f]\!]\varphi\rho[x = \flat v]$
$\Longrightarrow$ $([\![f]\!]\varphi\rho[x = \natural v])\backslash[v/x] = [\![f]\!]\varphi\rho[x = \natural v]$ by I
$\Longrightarrow$ $[\![f]\!]\varphi\rho[x = \flat v] = ([\![f]\!]\varphi\rho[x = \natural v])\backslash[v/x]$
If x is free in f,

a) f is $let(x)$ or $M(x)$ or $E_i(x)$,
by inspection of the trace definitions

b) $f \equiv h \mid g$
$[\![h \mid g]\!]\varphi\rho[x = \flat v] = [\![h]\!]\varphi\rho[x = \flat v] \parallel [\![g]\!]\varphi\rho[x = \flat v]$
$([\![h]\!]\varphi\rho[x = \natural v])\backslash[v/x] \parallel ([\![g]\!]\varphi\rho[x = \natural v])\backslash[v/x]$ by IH
$([\![h]\!]\varphi\rho[x = \natural v] \parallel [\![g]\!]\varphi\rho[x = \natural v])\backslash[v/x]$ by Corollary 5
$= ([\![h \mid g]\!]\varphi\rho[x = \natural v])\backslash[v/x]$

c) $f \equiv h >x> g$ (Similarly when $f \equiv h >y> g, x \neq y$)
$[\![h >x> g]\!]\varphi\rho[x = \flat v] = \bigcup_{s \in [\![h]\!]\varphi\rho[x = \flat v]} s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat v][x = \flat w]$
$= \bigcup_{s \in ([\![h]\!]\varphi\rho[x = \natural v])\backslash[v/x]} s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat w]$ by IH $(c1)$
By Lemma 37, $[v/x]$ is not in the traces of $[\![g]\!]\varphi\rho[x = \flat w]$
$\Longrightarrow$ $[\![g]\!]\varphi\rho[x = \flat w] = ([\![g]\!]\varphi\rho[x = \flat w])\backslash[v/x]$
$= ([\![g]\!]\varphi\rho[x = \natural v][x = \flat w])\backslash[v/x]$
$c1 \Rightarrow \bigcup_{s \in [\![h]\!]\varphi\rho[x = \natural v]} s\backslash[v/x] \gg \lambda w.([\![g]\!]\varphi\rho[x = \natural v][x = \flat w])\backslash[v/x]$
$= \bigcup_{s \in [\![h]\!]\varphi\rho[x = \natural v]} (s \gg [\![g]\!]\varphi\rho[x = \natural v][x = \flat w])\backslash[v/x]$ by Lemma 42
$= (\bigcup_{s \in [\![h]\!]\varphi\rho[x = \natural v]} s \gg [\![g]\!]\varphi\rho[x = \natural v][x = \flat w])\backslash[v/x]$
$= ([\![h >x> g]\!]\varphi\rho[x = \natural v])\backslash[v/x]$

d) $f \equiv h$ **where** $x :\in g$ (Similarly when $f \equiv h$ **where** $y :\in g, x \neq y$)
$[\![h$ **where** $x :\in g]\!]\varphi\rho[x = \flat v] =$
$= \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \flat v][x = \natural w] <_x [\![g]\!]\varphi\rho[x = \flat v]$
$= \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \natural w] <_x ([\![g]\!]\varphi\rho[x = \natural v])\backslash[v/x]$ by IH
Let $T_1 = \bigcup_{w \in Val} [\![h]\!]\varphi\rho[x = \natural w]$, $T_2 = [\![g]\!]\varphi\rho[x = \natural v]$
then the above becomes $T_1 <_x T_2\backslash[v/x]$
$= \bigcup_{t_1 \in T_1, t_2 \in T_2\backslash[v/x]} t_1 <_x t_2$
$= \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 <_x t_2\backslash[v/x]$

$= \bigcup_{t_1 \in T_1, t_2 \in T_2} (t_1 <_x t_2) \backslash [v/x]$ by Lemma 45
$= (\bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 <_x t_2) \backslash [v/x]$
$= (T_1 <_x T_2) \backslash [v/x]$
$= (\bigcup_{w \in Val} [\![h]\!] \varphi\rho[x = \natural w] <_x [\![g]\!] \varphi\rho[x = \natural v]) \backslash [v/x]$
$= ([\![h \textbf{ where } x :\in g]\!] \varphi\rho[x = \natural v]) \backslash [v/x]$ □

F Operational Lemmas

Lemma 48. *If* $\Delta, \Gamma \vdash f \xrightarrow{a} f'$ *and 'a' not a recv for* x*, then* $\Delta, \Gamma[x = v] \vdash f \xrightarrow{a} f'$ *for any* v

Proof. By induction on the height of the derivation

- (SITEC) $\dfrac{}{\Delta, \Gamma \vdash M(v) \xrightarrow{M_k(v)} ?k}\ k$ fresh
  This reduction is independent of Γ, thus the Lemma holds.
  Similarly for SITERET, LET, DEF
- (LET-VAR) $\dfrac{}{\Delta, \Gamma \vdash let(y) \xrightarrow{[w/y]} let(w)}\ \Gamma(y) = w$
  This reduction is independent of $\Gamma(x)$, thus the Lemma holds.
  Similarly for SITEC-VAR, DEF-VAR
- (SYM-L) $\dfrac{\Delta, \Gamma \vdash f \xrightarrow{a} f'}{\Delta, \Gamma \vdash f \mid g \xrightarrow{a} f' \mid g}\ a \neq [w/x]$
  By *IH*, $\Delta, \Gamma[x = v] \vdash f \xrightarrow{a} f'$
  $\implies \Delta, \Gamma[x = v] \vdash f \mid g \xrightarrow{a} f' \mid g$ by SYM-L
  Similarly for SYM-R
- (ASYM-L) $\dfrac{\Delta, \Gamma \vdash f \xrightarrow{a} f'}{\Delta, \Gamma \vdash f \textbf{ where } x :\in g \xrightarrow{a} f' \textbf{ where } x :\in g}\ a \neq [w/x]$
  By *IH*, $\Delta, \Gamma[x = v] \vdash f \xrightarrow{a} f'$
  $\implies \Delta, \Gamma[x = v] \vdash f \textbf{ where } x :\in g \xrightarrow{a} f' \textbf{ where } x :\in g$ by ASYM-L
  Also, consider the case when $x \neq y$ and
  (ASYM-L) $\dfrac{\Delta, \Gamma \vdash f \xrightarrow{a} f'}{\Delta, \Gamma \vdash f \textbf{ where } y :\in g \xrightarrow{a} f' \textbf{ where } x :\in g}\ a \neq [w/x], a \neq [w/y]$
  As above.

Similarly for the other rules. □

Lemma 49. *If* $\Delta, \Gamma[x = v] \vdash f \xrightarrow{a} f'$ *and* $a \neq [v/x]$ *then* $\Delta, \Gamma' \vdash f \xrightarrow{a} f'$ *where* $\Gamma'(y) = \begin{cases} \Gamma(y) & x \neq y \\ \textit{unspecified/anything} & x = y \end{cases}$

Proof. By induction on the height of the derivation.
The Lemma trivially holds for the reductions that are independent of the environment.

- (SITEC-VAR) $\dfrac{}{\Delta, \Gamma[x = v] \vdash M(y) \xrightarrow{[w/y]} M(w)}\ \Gamma(y) = w$ and $x \neq y$
  $\implies \dfrac{}{\Delta, \Gamma' \vdash M(y) \xrightarrow{[w/y]} M(w)}\ \Gamma'(y) = w$
  Similarly for LET-VAR, DEF-VAR

– (SYM-L) $\dfrac{\Delta, \Gamma[x = v] \vdash\ f \xrightarrow{a} f'}{\Delta, \Gamma[x = v] \vdash\ f \mid g \xrightarrow{a} f' \mid g}\ a \neq [v/x]$
By *IH* and SYM-L we get the result. Similarly for SYM-R, ASYM-R, ASYM-P, SEQ, SEQ-P

– (ASYM-L) $\dfrac{\Delta, \Gamma[x = v] \vdash\ f \xrightarrow{a} f'}{\Delta, \Gamma[x = v] \vdash\ f \textbf{ where } x :\in g \xrightarrow{a} f' \textbf{ where } x :\in g}\ a \neq [v/x]$
By *IH* and ASYM-L we get the result.
Consider also the case when $x \neq y$ and
(ASYM-L) $\dfrac{\Delta, \Gamma[x = v] \vdash\ f \xrightarrow{a} f'}{\Delta, \Gamma[x = v] \vdash\ f \textbf{ where } y :\in g \xrightarrow{a} f' \textbf{ where } y :\in g}\ \begin{smallmatrix} a \neq [v/x] \\ a \neq [v'/y] \end{smallmatrix}$
By *IH*, $\Delta, \Gamma' \vdash\ f \xrightarrow{a} f'$ and when $a \neq [w/y]$
$\Delta, \Gamma' \vdash\ f \textbf{ where } y :\in g \xrightarrow{a} f' \textbf{ where } y :\in g$ by ASYM-L □

Lemma 50. *$\Delta, \Gamma \vdash\ f \xrightarrow{a} f'$ implies $f.v.(f') \subseteq f.v.(f)$*

Proof. By induction on the height of the derivation. The interesting cases are

– (DEF) $\dfrac{}{\Delta, \Gamma \vdash\ E_i(v) \xrightarrow{\tau} [v/x]f_i}\ (E_i(x) = f_i) \in \Delta$
f.v.$(E_i(v)) = \emptyset =$ f.v.$([v/x]f_i)$ by the constraint f.v.$(f_i) \subseteq \{x\}$

– (ASYM-L) $\dfrac{\Delta, \Gamma \vdash\ h \xrightarrow{a} h'}{\Delta, \Gamma \vdash\ h \textbf{ where } x :\in g \xrightarrow{a} h' \textbf{ where } x :\in g}\ a \neq [v/x]$
f.v.$(h') \subseteq$ f.v.(h) by *IH* (I)
f.v.$(h' \textbf{ where } x :\in g) =$ (f.v.$(h') - \{x\}) \cup$ f.v.(g)
$\subseteq$ (f.v.$(h) - \{x\}) \cup$ f.v.(g) by I
= f.v.$(h \textbf{ where } x :\in g)$ □

Lemma 51. *If $x \notin f.v.(f)$ then for any v, Γ $\Delta, \Gamma \vdash\ f \overset{[v/x]}{\not\to} f'$*

Proof. By structural induction on f. □

G Soundness - Adequacy

Lemma 52. *If* $\Delta, \Gamma \vdash f \xrightarrow{a} f'$ *and* $t \in [\![f']\!][\![\Delta]\!]\rho$ *then* $at \in [\![f]\!][\![\Delta]\!]\rho$ *where* $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$ *and* $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$

Proof. By structural induction on f and cases on the reduction rule used

a) (SITEC) $\dfrac{}{\Delta, \Gamma \vdash M(v) \xrightarrow{M_k(v)} ?k}$ k fresh
$[\![?k]\!][\![\Delta]\!]\rho = \{\, k?w\ !w \mid w \in Val \,\}_{\mathrm{p}}$
Consider only the case when $t = k?w\ !w$
Then, $(M_k(v)\ k?w\ !w) \in [\![M(v)]\!][\![\Delta]\!]\rho$

b) (SITEC-VAR) $\dfrac{}{\Delta, \Gamma \vdash M(x) \xrightarrow{[v/x]} M(v)}$ $\Gamma(x) = v$
$[\![M(v)]\!][\![\Delta]\!]\rho = \{\, M_k(v)\ k?w\ !w \mid w \in Val\,, k \text{ fresh} \,\}_{\mathrm{p}}$
Consider only the case when $t = M_k(v)\ k?w$
We know $\Gamma(x) = v$, therefore $\rho(x) = \natural v$
$\Longrightarrow \quad ([v/x]\ M_k(v)\ k?w) \in [\![M(x)]\!][\![\Delta]\!]\rho$ when $\rho(x) = \natural v$

c) SITERET, LET, LET-VAR, DEF-VAR similarly

d) (DEF) $\dfrac{}{\Delta, \Gamma \vdash E_i(v) \xrightarrow{\tau} [v/x]f_i}$ $(E_i(x) \triangleq f_i) \in \Delta$

Let $t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho \overset{\text{Lem. } 36}{\Longrightarrow} t \in [\![f_i]\!][\![\Delta]\!]\rho[x = \flat v]$ $(d1)$
Also, $[\![E_i(v)]\!][\![\Delta]\!]\rho = \{\, \tau\ t' \mid t' \in [\![\Delta]\!]_i(v) \,\}_{\mathrm{p}}$
where $[\![\Delta]\!]_i(v) = [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]$ $(d2)$
By $d2$, it suffices to show that $t \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]$, which holds by $d1$ and Lemma 35, because $x_1, \ldots, x_m$ are not free in f_i

e) (SYM-L) $\dfrac{\Delta, \Gamma \vdash h \xrightarrow{a} h'}{\Delta, \Gamma \vdash h \mid g \xrightarrow{a} h' \mid g}$
Let $t \in [\![h' \mid g]\!][\![\Delta]\!]\rho$, then there exist $t_1 \in [\![h']\!][\![\Delta]\!]\rho$, $t_2 \in [\![g]\!][\![\Delta]\!]\rho$ such that $t \in t_1 \parallel t_2$ $(e1)$
By *IH* for h, $at_1 \in [\![h]\!][\![\Delta]\!]\rho \overset{e1}{\Longrightarrow} at \in at_1 \parallel t_2$
$\Longrightarrow \quad at \in [\![h \mid g]\!][\![\Delta]\!]\rho$

f) Similarly for (SYM-R)

g) (ASYM-L) $\dfrac{\Delta, \Gamma \vdash h \xrightarrow{a} h'}{\Delta, \Gamma \vdash h \textbf{ where } x :\in g \xrightarrow{a} h' \textbf{ where } x :\in g}$ $a \neq [v/x]$
Let $t \in [\![h' \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$, then there exist
$t_1 \in \bigcup_{v \in Val} [\![h']\!][\![\Delta]\!]\rho[x = \natural v], t_2 \in [\![g]\!][\![\Delta]\!]\rho$ such that $t \in t_1 <_x t_2$ $(g1)$
Also, by Lemma 48, $\Delta, \Gamma[x = w] \vdash h \xrightarrow{a} h'$ for any w $(g2)$
Cases depending on which branch of the definition of $<_x$ was used for t:

- 1^{st} branch was used,
  $\Longrightarrow$ no recv. for x in t_1, no publ. in t_2, $t \in t_1 \parallel t_2$ $(g3)$
  By $g1, g2$ and *IH* for h we get $at_1 \in \bigcup_{v \in Val} [\![h]\!][\![\Delta]\!]\rho[x = \natural v]$ $(g4)$
  $\Longrightarrow \quad at \in at_1 \parallel t_2$ by $g3$
  $\Longrightarrow \quad at \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$ by $g1, g4$

- 2^{nd} branch was used,
  $\Longrightarrow$ no recv. for x in t_1, $t_2 = t_{21}!u\,t_{22}$, no publ. in t_{21},
  $t \in t_1 \parallel t_{21}\tau$ $(g5)$
  By $g1, g2$ and *IH* for h we get $at_1 \in \bigcup_{v\in Val}[\![h]\!][\![\Delta]\!]\rho[x = \natural v]$ $(g6)$
  $\Longrightarrow$ $at \in at_1 <_x t_2$ by $g5$
  $\Longrightarrow$ $at \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$ by $g1, g6$
- 3^{rd} branch was used,
  $\Longrightarrow$ $t_1 = t_{11}[u/x]\,t_{12}$, no recv. for x in t_{11},
  $t_2 = t_{21}!u\,t_{22}$, no publ. in t_{21}, $t \in (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[u/x])$ $(g7)$
  $t_1 \in [\![h']\!][\![\Delta]\!]\rho[x = \natural u]$ by Lemma 37
  $\Longrightarrow$ by $g2$ and *IH* for h we get $at_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural u]$
  $\Longrightarrow$ $at_1 \in \bigcup_{v\in Val}[\![h]\!][\![\Delta]\!]\rho[x = \natural v]$ $(g8)$
  $\Longrightarrow$ $at \in at_1 <_x t_2$ by $g7$
  $\Longrightarrow$ $at \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$ by $g1, g8$
- 4^{th} branch was used, $t = \varepsilon$
  $\varepsilon \in [\![g]\!][\![\Delta]\!]\rho$, $\varepsilon \in \bigcup_{v\in Val}[\![h']\!][\![\Delta]\!]\rho[x = \natural v]$ by Thm. 8
  $\Longrightarrow$ $a \in \bigcup_{v\in Val}[\![h]\!][\![\Delta]\!]\rho[x = \natural v]$ by *IH*
  $\Longrightarrow$ $a \in a <_x \varepsilon$
  $\Longrightarrow$ $a \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$

h) (ASYM-R) Similar to the previous case

i) (ASYM-P) $\dfrac{\Delta, \Gamma \vdash\ g \xrightarrow{!v}\ g'}{\Delta, \Gamma \vdash\ h \textbf{ where } x :\in g \xrightarrow{\tau}\ [v/x]h}$

Let $t \in [\![[v/x]h]\!][\![\Delta]\!]\rho$
$\Longrightarrow$ $t \in [\![h]\!][\![\Delta]\!]\rho[x = \flat v]$ by Lemma 36
$\Longrightarrow$ $\exists\, t' \in [\![h]\!][\![\Delta]\!]\rho[x = \natural v].\ t = t'\backslash[v/x]$ by Lemma 47 $(i1)$
$\varepsilon \in [\![g']\!][\![\Delta]\!]\rho$ by Thm. 8
$\Longrightarrow$ $!v \in [\![g]\!][\![\Delta]\!]\rho$ by *IH* $(i2)$

- no recv. for x in t'
  $\Longrightarrow$ $t = t'$ and $\tau\, t \in t <_x !v$ by $i1$
  $\Longrightarrow$ $\tau\, t \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$ by $i1, i2$
- $t' = t'_1[v/x]t'_2$, no recv. for x in t'_1 $(i3)$
  $\Longrightarrow$ $\tau\, t'_1(t'_2\backslash[v/x]) \in t' <_x !v$
  $\Longrightarrow$ $\tau\, t \in t' <_x !v$ by $i1, i3$
  $\Longrightarrow$ $\tau\, t \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$ by $i1, i2$

j) (SEQ) $\dfrac{\Delta, \Gamma \vdash\ h \xrightarrow{a}\ h'}{\Delta, \Gamma \vdash\ h >x> g \xrightarrow{a}\ h' >x> g}\ a \neq !v$

Let $t \in [\![h' >x> g]\!][\![\Delta]\!]\rho$, then there exists $s \in [\![h']\!][\![\Delta]\!]\rho$
such that $t \in s \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]$ $(j1)$
Cases on s:

- no publ. in $s \Rightarrow t \in \{s\} \Rightarrow t = s$ $(j2)$
  By *IH* for h, $as \in [\![h]\!][\![\Delta]\!]\rho$
  $\Longrightarrow$ $at \in as \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]$ by $j2$
  $\Longrightarrow$ $at \in [\![h >x> g]\!][\![\Delta]\!]\rho$
- $s = s_1!u\,s_2$, no publ. in s_1
  $\Longrightarrow$ $t \in s_1\tau((s_2 \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]) \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat u])$ by $j1$

$\implies \quad at \in as \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]$ $(j3)$

By *IH* for h, $as \in [\![h]\!][\![\Delta]\!]\rho$

$\implies \quad at \in [\![h >\!x\!> g]\!][\![\Delta]\!]\rho$ by $j3$

k) (SEQ-P) $\dfrac{\Delta, \Gamma \vdash\ h \xrightarrow{!u} h'}{\Delta, \Gamma \vdash\ h >\!x\!> g \xrightarrow{\tau} (h' >\!x\!> g) \mid [u/x]g}$

Let $t \in [\![(h' >\!x\!> g) \mid [u/x]g]\!][\![\Delta]\!]\rho$, then there exist

$t_1 \in [\![h' >\!x\!> g]\!][\![\Delta]\!]\rho$, $t_2 \in [\![[u/x]g]\!][\![\Delta]\!]\rho$ such that $t \in t_1 \parallel t_2$ $(k1)$

By Lemma 36, $t_2 \in [\![g]\!][\![\Delta]\!]\rho[x = \flat u]$ $(k2)$

By $k1$, $\exists s \in [\![h']\!][\![\Delta]\!]\rho.\ t_1 \in s \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]$ $(k3)$

By *IH* for h, $!u\, s \in [\![h]\!][\![\Delta]\!]\rho$ $(k4)$

By $k1, k2, k3 \quad t \in (s \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]) \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat u]$

$\implies \quad \tau\, t \in \tau((s \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]) \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat u])$

$\implies \quad \tau\, t \in\ !u\, s \gg \lambda v.[\![g]\!][\![\Delta]\!]\rho[x = \flat v]$

$\implies \quad \tau\, t \in [\![h >\!x\!> g]\!][\![\Delta]\!]\rho$ by $k4$ □

Theorem 9 (Soundness). *If* $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$, $\sigma = [w_1/y_1] \ldots [w_n/y_n]$, $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m][y_1 = \flat w_1] \ldots [y_n = \flat w_n]$, *x's and y's are all distinct, then*

$$\Delta, \Gamma \vdash\ \sigma f \xrightarrow{t}{}^{*}\ f' \quad \textit{implies} \quad t \in [\![f]\!][\![\Delta]\!]\rho$$

Proof. By induction on $|t|$

- If $|t| = 0 \Leftrightarrow t = \varepsilon$
  $\implies \quad \varepsilon \in [\![f]\!][\![\Delta]\!]\rho$ by Thm. 8
- If $t = a\, t'$
  $\implies \quad \Delta, \Gamma \vdash\ \sigma f \xrightarrow{a} f'' \xrightarrow{t'}{}^{*} f'$
  By *IH* for t', $t' \in [\![f'']\!][\![\Delta]\!]\rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$
  and by Lemma 52, $a\, t' \in [\![\sigma f]\!][\![\Delta]\!]\rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$
  therefore $t \in [\![f]\!][\![\Delta]\!]\rho$ by Lemma 36 □

Lemma 53. *If* $at \in [\![f]\!][\![\Delta]\!]\rho$ *then* $\Delta, \Gamma \vdash\ f \xrightarrow{a} f'$ *and* $t \in [\![f']\!][\![\Delta]\!]\rho$ *where* $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$ *and* $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$

Proof. By structural induction on f

a) $f \equiv 0$ vacuously true

b) $f \equiv let(v)$
   $\implies \quad [\![let(v)]\!][\![\Delta]\!]\rho = \{!v\}_{\mathrm{p}}$
   $\implies \quad a = !v$ and $t = \varepsilon$
   Also, $\Delta, \Gamma \vdash\ let(v) \xrightarrow{!v} \mathbf{0}$ and $\varepsilon \in [\![\mathbf{0}]\!][\![\Delta]\!]\rho$

c) $f \equiv M(v)$ or $?k$ similarly

d) $f \equiv let(x)$
   For a non-empty trace of f, we know $\rho(x) = \natural v$
   $\implies \quad [\![let(x)]\!][\![\Delta]\!]\rho = \{[v/x]\, !v\}_{\mathrm{p}}$
   Consider only the case when $a = [v/x]$ and $t = !v$
   Then, by LET-VAR, $\Delta, \Gamma \vdash\ let(x) \xrightarrow{[v/x]} let(v)$ and $!v \in [\![let(v)]\!][\![\Delta]\!]\rho$

e) $f \equiv M(x)$ similarly

f) $f \equiv E_i(v)$

$\Longrightarrow \quad [\![E_i(v)]\!][\![\Delta]\!]\rho = \{\, \tau\, t \mid t \in [\![\Delta]\!]_i(v) \,\}_{\mathrm{p}}$

By DEF, $\quad \Delta, \Gamma \vdash E_i(v) \xrightarrow{\tau} [v/x]f_i$

$\Longrightarrow$ suffices to show that for any $t \in [\![\Delta]\!]_i(v)$ then $t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho$

We know $[\![\Delta]\!] = \mathrm{fix}(\hat{\Delta}) \Rightarrow \hat{\Delta}([\![\Delta]\!]) = [\![\Delta]\!]$

Then, $t \in [\![\Delta]\!]_i(v)$ implies $t \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]$

$\Longrightarrow \quad t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho_0$ by Lemma 36

$\Longrightarrow \quad t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho$ by Lemma 35 because f.v.$([v/x]f_i) = \emptyset$

g) $f \equiv h \mid g$

Let $a\, t \in [\![h \mid g]\!][\![\Delta]\!]\rho$, then there exist

$t_1 \in [\![h]\!][\![\Delta]\!]\rho, t_2 \in [\![g]\!][\![\Delta]\!]\rho$ such that $a\, t \in t_1 \parallel t_2$ $(g1)$

- 'a' is an event of t_1, i.e. $t_1 = a\, t_1'$, and by $g1$, $t \in t_1' \parallel t_2$ $(g2)$

  By *IH* for h, $\Delta, \Gamma \vdash h \xrightarrow{a} h'$ and $t_1' \in [\![h']\!][\![\Delta]\!]\rho$ $(g3)$

  $\Longrightarrow \quad \Delta, \Gamma \vdash h \mid g \xrightarrow{a} h' \mid g$ by SYM-L

  $\Longrightarrow \quad t \in [\![h' \mid g]\!][\![\Delta]\!]\rho$ by $g1, g2, g3$

- 'a' is an event of t_2, similarly

h) $f \equiv h >\!x\!> g$

Let $a\, t \in [\![h >\!x\!> g]\!][\![\Delta]\!]\rho$ then there exists

$s \in [\![h]\!][\![\Delta]\!]\rho$ such that $a\, t \in s \gg \lambda w.[\![g]\!][\![\Delta]\!]\rho[x = \flat w]$ $(h1)$

- no publ. in s

  $\Longrightarrow \quad a\, t = s$ by $h1$

  $\Longrightarrow \quad \Delta, \Gamma \vdash h \xrightarrow{a} h'$ and $t \in [\![h']\!][\![\Delta]\!]\rho$ by *IH* for h $(h2)$

  $\Longrightarrow \quad \Delta, \Gamma \vdash h >\!x\!> g \xrightarrow{a} h' >\!x\!> g$ by SEQ

  $\Longrightarrow$ suffices to show that $t \in [\![h' >\!x\!> g]\!][\![\Delta]\!]\rho$ which holds by $h2$

- $s = s_1\, !v\, s_2$, no publ. in s_1

  Then, by $h1$

  $a\, t \in s_1 \tau((s_2 \gg \lambda w.[\![g]\!][\![\Delta]\!]\rho[x = \flat w]) \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat v])$ $(h3)$

  * 'a' is the first event of s_1, $s_1 = a\, s_1'$

    $\Longrightarrow \quad \Delta, \Gamma \vdash h \xrightarrow{a} h'$ and $s_1' !v s_2 \in [\![h']\!][\![\Delta]\!]\rho$ by *IH* for h $(h4)$

    $\Longrightarrow \quad \Delta, \Gamma \vdash h >\!x\!> g \xrightarrow{a} h' >\!x\!> g$ by SEQ

    We know that, $t \in s_1' !v s_2 \gg \lambda w.[\![g]\!][\![\Delta]\!]\rho[x = \flat w]$ by $h3$

    $\Longrightarrow \quad t \in [\![h' >\!x\!> g]\!][\![\Delta]\!]\rho$ by $h4$

  * s_1 is empty, therefore $s = !v\, s_2$ and by $h3$ $a = \tau$

    $\Longrightarrow \quad \Delta, \Gamma \vdash h \xrightarrow{!v} h'$ and $s_2 \in [\![h']\!][\![\Delta]\!]\rho$ by *IH* for h $(h6)$

    Then, by SEQ-P

    $\Delta, \Gamma \vdash h >\!x\!> g \xrightarrow{\tau} (h' >\!x\!> g) \mid [v/x]g$

    By $h3$, $t \in (s_2 \gg \lambda w.[\![g]\!][\![\Delta]\!]\rho[x = \flat w]) \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat v]$

    $\Longrightarrow \quad t \in [\![h' >\!x\!> g]\!][\![\Delta]\!]\rho \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat v]$ by $h6$

    $\Longrightarrow \quad t \in [\![(h' >\!x\!> g) \mid [v/x]g]\!][\![\Delta]\!]\rho$ by Lem. 36

i) $f \equiv h$ **where** $x :\in g$

Let $a\, t \in [\![h \textbf{ where } x :\in g]\!][\![\Delta]\!]\rho$, then there exist

$t_1 \in \bigcup_{v \in Val} [\![h]\!][\![\Delta]\!]\rho[x = \natural v], t_2 \in [\![g]\!][\![\Delta]\!]\rho$ such that $a\, t \in t_1 <_x t_2$ $(i1)$

Cases on the branch of the definition of $<_x$ used for $a\, t$

- no recv. for x in t_1, no publ. in $t_2 \Longrightarrow a\, t \in t_1 \parallel t_2$ $(i2)$

∗ 'a' is an event of t_1, i.e. $t_1 = a\,t_1'$ and $t \in t_1' \parallel t_2$ $(i3)$
We know that, $a\,t_1' \in [\![h]\!][\![\Delta]\!]\rho$ by $i2$ and Lemma 38
$\Longrightarrow$ $\Delta, \Gamma \vdash h \xrightarrow{a} h'$ and $t_1' \in [\![h']\!][\![\Delta]\!]\rho$ by IH $(i4)$
$\overset{\text{ASYM-L}}{\Longrightarrow}$ $\Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h'$ **where** $x :\in g$
By $i1$, $\exists u \in Val.\ a\,t_1' \in [\![h]\!][\![\Delta]\!]\rho[x = \natural u]$
$\Longrightarrow$ $\Delta, \Gamma[x = u] \vdash h \xrightarrow{a} h'$ and $t_1' \in [\![h']\!][\![\Delta]\!]\rho[x = \natural u]$ by IH
$\Longrightarrow$ $t_1' \in \bigcup_{v \in Val} [\![h']\!][\![\Delta]\!]\rho[x = \natural v]$
$\Longrightarrow$ $t \in [\![h'$ **where** $x :\in g]\!][\![\Delta]\!]\rho$ by $i1, i3$
∗ 'a' is an event of t_2, i.e. $t_2 = a\,t_2'$ and $t \in t_1 \parallel t_2'$ $(i5)$
$\Delta, \Gamma \vdash g \xrightarrow{a} g'$ and $t_2' \in [\![g']\!][\![\Delta]\!]\rho$ by IH for g $(i6)$
$\overset{\text{ASYM-R}}{\Longrightarrow}$ $\Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h$ **where** $x :\in g'$
Also, $t \in t_1 <_x t_2'$ by $i2, i5$
$\Longrightarrow$ $t \in [\![h$ **where** $x :\in g']\!][\![\Delta]\!]\rho$ by $i1, i6$

- no recv. for x in t_1, $t_2 = t_{21}!w\,t_{22}$, no publ. in t_{21}
$\Longrightarrow a\,t \in t_1 \parallel t_{21}\,\tau$ $(i7)$
∗ 'a' is an event of t_1, i.e. $t_1 = a\,t_1'$ and $t \in t_1' \parallel t_{21}\,\tau$
. . . It's exactly the same as the previous case for t_1
∗ 'a' is an event of t_{21}, i.e. $t_{21} = a\,t_{21}'$ and $t \in t_1 \parallel t_{21}'\,\tau$
. . . It's exactly the same as the previous case for t_2
∗ t_{21} is empty, $a = \tau$ and $t = t_1$
$\Delta, \Gamma \vdash g \xrightarrow{!w} g'$ and $t_{22} \in [\![g']\!][\![\Delta]\!]\rho$ by IH for g
$\overset{\text{ASYM-P}}{\Longrightarrow}$ $\Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{\tau} [w/x]h$
Suffices to show that $t_1 \in [\![[w/x]h]\!][\![\Delta]\!]\rho$
We know that $t_1 \in [\![h]\!][\![\Delta]\!]\rho_{-x}$ by $i7$ and Lemma 38
$\Longrightarrow$ $t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \flat w]$ by Lemma 40
$\Longrightarrow$ $t_1 \in [\![[w/x]h]\!][\![\Delta]\!]\rho$ by Lemma 36

- $t_1 = t_{11}[w/x]\,t_{12}, [w/x] \hat{\notin} t_{11}$, $t_2 = t_{21}!w\,t_{22}$, no publ. in t_{21}
$\Longrightarrow$ $a\,t \in (t_{11} \parallel t_{21}\tau)(t_{12} \backslash [w/x])$
∗ 'a' is an event of t_{11},
i.e. $t_{11} = a\,t_{11}'$ and $t \in (t_{11}' \parallel t_{21}\,\tau)(t_{12} \backslash [w/x])$
$\Longrightarrow$ $t \in (t_{11}'[w/x]t_{12} <_x t_2)$ $(i8)$
By Lemma 37, $t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural w]$
$\Longrightarrow$ $\Delta, \Gamma[x = w] \vdash h \xrightarrow{a} h'$ and
$t_{11}'[w/x]t_{12} \in [\![h']\!][\![\Delta]\!]\rho[x = \natural w]$ by IH $(i9)$
$\Longrightarrow$ $\Delta, \Gamma \vdash h \xrightarrow{a} h'$ by Lemma 49
$\overset{\text{ASYM-L}}{\Longrightarrow}$ $\Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h'$ **where** $x :\in g$
Also, by $i8$ and $i9$ $t \in [\![h'$ **where** $x :\in g]\!][\![\Delta]\!]\rho$
∗ 'a' is an event of t_{21},
i.e. $t_{21} = a\,t_{21}'$ and $t \in (t_{11} \parallel t_{21}'\,\tau)(t_{12} \backslash [w/x])$
$\Longrightarrow$ $t \in (t_1 <_x t_{21}'!w\,t_{22})$ $(i10)$
$\Delta, \Gamma \vdash g \xrightarrow{a} g'$ and $t_{21}'!w\,t_{22} \in [\![g']\!][\![\Delta]\!]\rho$ by IH $(i11)$
$\overset{\text{ASYM-R}}{\Longrightarrow}$ $\Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h$ **where** $x :\in g'$
and $t \in [\![h$ **where** $x :\in g']\!][\![\Delta]\!]\rho$ by $i10, i11$

$*$ t_{21} is empty, $a = \tau$ and $t = t_{11}(t_{21}\backslash[w/x]) = t_1\backslash[w/x]$ ($i12$)
$\Delta, \Gamma \vdash g \xrightarrow{!w} g'$ and $t_{22} \in [\![g']\!][\![\Delta]\!]\rho$ by *IH*
$\overset{\text{ASYM-P}}{\Longrightarrow} \Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{\tau} [w/x]h$
By Lemma 37, $t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural w]$
$\Longrightarrow$ $t_1\backslash[w/x] \in [\![h]\!][\![\Delta]\!]\rho[x = \flat w]$ by Lemma 47
$\Longrightarrow$ $t \in [\![[w/x]h]\!][\![\Delta]\!]\rho$ by $i12$ and Lemma 36

- $a\,t = \varepsilon$
not applicable, $a\,t$ can not be empty □

Lemma 54. *If* $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$,
$\sigma = [w_1/y_1] \ldots [w_n/y_n]$, $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m][y_1 = \flat w_1] \ldots [y_n = \flat w_n]$,
x's and y's are all distinct, then

$$at \in [\![f]\!][\![\Delta]\!]\rho \quad \textit{implies} \quad \Delta, \Gamma \vdash \sigma f \xrightarrow{a} f' \textit{ and } t \in [\![f']\!][\![\Delta]\!]\rho$$

Proof. Straightforward, using lemmas 53, 36 and 50 □

Theorem 10 (Adequacy). *If* $\Gamma = \{(x_1, v_1), \ldots, (x_m, v_m)\}$,
$\sigma = [w_1/y_1] \ldots [w_n/y_n]$, $\rho = \rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m][y_1 = \flat w_1] \ldots [y_n = \flat w_n]$,
x's and y's are all distinct, then

$$t \in [\![f]\!][\![\Delta]\!]\rho \quad \textit{implies} \quad \Delta, \Gamma \vdash \sigma f \xrightarrow{t}{}^{*} f'$$

Proof. By induction on $|t|$

- If $|t| = 0 \Leftrightarrow t = \varepsilon$, then σf reduces to itself in 0 steps.
- If $t = a\,t'$ then
$a\,t' \in [\![f]\!][\![\Delta]\!]\rho$
$\Longrightarrow$ $a\,t' \in [\![\sigma f]\!][\![\Delta]\!]\rho[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$ by Lemma 36
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f'$ and
$t' \in [\![f']\!][\![\Delta]\!]\rho[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$ by Lemma 53
$\Longrightarrow$ $\Delta, \Gamma \vdash f' \xrightarrow{t'}{}^{*} f''$ by *IH* for t'
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma f \xrightarrow{a} f' \xrightarrow{t'}{}^{*} f''$
$\Longrightarrow$ $\Delta, \Gamma \vdash \sigma f \xrightarrow{t}{}^{*} f''$ □

H Removing the τ's

Theorem 11. *For all f, $\{\!|f|\!\} = \lambda\varphi.\lambda\rho.[\![f]\!]\varphi\rho\backslash\tau$*

Proof. Suffices to show that for any φ, ρ we get
$\{\!|f|\!\}\varphi\rho = [\![f]\!]\varphi\rho\backslash\tau$
Proceed by structural induction on f

a) If f is $\mathbf{0}$, $let(v)$, $let(x)$, $M(v)$, $M(x)$, $?k$, $E_i(v)$, $E_i(x)$ it is immediate by the definitions

b) $f \equiv h \mid g$
$\implies \{\!|f|\!\}\varphi\rho = \{\!|h|\!\}\varphi\rho \parallel \{\!|g|\!\}\varphi\rho =$
$= [\![h]\!]\varphi\rho\backslash\tau \parallel [\![g]\!]\varphi\rho\backslash\tau$ by *IH*
$= ([\![h]\!]\varphi\rho \parallel [\![g]\!]\varphi\rho)\backslash\tau$ by Corollary 5
$= [\![h \mid g]\!]\varphi\rho\backslash\tau$

c) $f \equiv h >x> g$
$\implies \{\!|f|\!\}\varphi\rho = \bigcup_{s\in\{\!|h|\!\}\varphi\rho} s \dot{\gg} \lambda v.\{\!|g|\!\}\varphi\rho[x = \flat v] =$
$= \bigcup_{s\in[\![h]\!]\varphi\rho\backslash\tau} s \dot{\gg} \lambda v.[\![g]\!]\varphi\rho[x = \flat v]\backslash\tau$ by *IH*
$= \bigcup_{s\in[\![h]\!]\varphi\rho} s\backslash\tau \dot{\gg} \lambda v.[\![g]\!]\varphi\rho[x = \flat v]\backslash\tau$
$= \bigcup_{s\in[\![h]\!]\varphi\rho}(s \gg \lambda v.[\![g]\!]\varphi\rho[x = \flat v])\backslash\tau$ by Lemma 43
$= [\![h >x> g]\!]\varphi\rho\backslash\tau$

d) $f \equiv h$ **where** $x :\in g$
As above, using Lemma 46 □

Theorem 12. *For all f, $\{\!|f|\!\}$ is continuous*

Proof. By theorem 11, $\{\!|f|\!\}$ is the composition of continuous functions, therefore it is continuous. □

Lemma 55. *For all f, ρ $[\![f >x> let(x)]\!][\![\Delta]\!]\rho\backslash\tau \subseteq [\![f]\!][\![\Delta]\!]\rho\backslash\tau$*

Proof.
Suffices to show that if $t \in [\![f >x> let(x)]\!][\![\Delta]\!]\rho$ then $t\backslash\tau \in [\![f]\!][\![\Delta]\!]\rho\backslash\tau$

$IH((k,l))$:
$\forall i, j, t, f, \rho.\ (\lceil t\rceil = i \wedge |f| = j \wedge (i,j) \sqsubset (k,l) \wedge t \in [\![f >x> let(x)]\!][\![\Delta]\!]\rho)$
$\implies t\backslash\tau \in [\![f]\!][\![\Delta]\!]\rho\backslash\tau$
Cases on f:

a) $f \equiv M(v)$
$\implies T_1 = [\![M(v)]\!][\![\Delta]\!]\rho = \{\ M_k(v)\ k?w\ !w \mid k \text{ fresh}, w \in Val\}_{\mathrm{p}}$
$\implies T_2 = [\![M(v) >x> let(x)]\!][\![\Delta]\!]\rho = \bigcup_{s\in T_1} s \gg \lambda u.\{!u\}_{\mathrm{p}}$
If $t \in T_2$ then there exists $s \in T_1$ such that $t \in s \gg \lambda u.\{!u\}_{\mathrm{p}}$
If s does not publish, trivial.
If $s = M_k(v)\ k?w\ !w$ then $t \in (M_k(v)\ k?w\ \tau((\varepsilon \gg \lambda u.\{!u\}_{\mathrm{p}}) \parallel \{!w\}_{\mathrm{p}}))$
$\implies t \in (M_k(v)\ k?w\ \tau\{\varepsilon, !w\})$
$\implies t\backslash\tau \in \{M_k(v)\ k?w,\ M_k(v)\ k?w\ !w\}$
$\implies t\backslash\tau \in T_1$
$\implies t\backslash\tau \in T_1\backslash\tau$

b) If f is $\mathbf{0}$, $let(x)$, $let(v)$, $M(x)$, $?k$ similarly

c) $f \equiv E_i(v)$
$T_1 = [\![E_i(v)]\!][\![\Delta]\!]\rho = \{\, \tau\, t_1 \mid t_1 \in [\![\Delta]\!]_i(v)\}_{\mathrm{p}}$
$= \{\, \tau\, t_1 \mid t_1 \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]\}_{\mathrm{p}}$ because $[\![\Delta]\!] = \hat{\Delta}([\![\Delta]\!])$
Then,
$T_2 = [\![E_i(v) >x> let(x)]\!][\![\Delta]\!]\rho = \bigcup_{s\in T_1} s \gg \lambda w.\{!w\}_{\mathrm{p}}$
$= \{\varepsilon\} \cup \tau(\bigcup_{s\in[\![f_i]\!][\![\Delta]\!]\rho_0[x=\flat v]}\ s \gg \lambda w.\{!w\}_{\mathrm{p}})$
$= \{\varepsilon\} \cup \tau\,([\![f_i >x> let(x)]\!][\![\Delta]\!]\rho_0[x = \flat v])$
If $t = \varepsilon$ then $t\backslash\tau \in T_1\backslash\tau$
else there exists $t' \in [\![f_i >x> let(x)]\!][\![\Delta]\!]\rho_0[x = \flat v]$ such that $t = \tau\, t'$
But $\lceil t' \rceil < \lceil t \rceil$
$\Longrightarrow$ $(t', f_i) \sqsubset (t, E_i(v))$
$\Longrightarrow$ $t'\backslash\tau \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]\backslash\tau$ by *IH*
$\Longrightarrow$ $t\backslash\tau \in \{\, \tau\, t_1 \mid t_1 \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]\}_{\mathrm{p}}\backslash\tau$
$\Longrightarrow$ $t\backslash\tau \in T_1\backslash\tau$

d) $f \equiv h \mid g$
$T = [\![(h \mid g) >x> let(x)]\!][\![\Delta]\!]\rho$
$= [\![(h >x> let(x)) \mid (g >x> let(x))]\!][\![\Delta]\!]\rho$ by Lemma 14, Theorem 6
$= [\![h >x> let(x)]\!][\![\Delta]\!]\rho \parallel [\![g >x> let(x)]\!][\![\Delta]\!]\rho$
If $t \in T$, $\exists t_1 \in [\![h >x> let(x)]\!][\![\Delta]\!]\rho, t_2 \in [\![g >x> let(x)]\!][\![\Delta]\!]\rho.\ t \in t_1 \parallel t_2$
But $(t_1, h) \sqsubset (t, h \mid g)$ and $(t_2, g) \sqsubset (t, h \mid g)$
$\Longrightarrow$ $t_1\backslash\tau \in [\![h]\!][\![\Delta]\!]\rho\backslash\tau$ and $t_2\backslash\tau \in [\![g]\!][\![\Delta]\!]\rho\backslash\tau$ by *IH* $(d1)$
We know that, $t\backslash\tau \in (t_1 \parallel t_2)\backslash\tau$
$\Longrightarrow$ $t\backslash\tau \in t_1\backslash\tau \parallel t_2\backslash\tau$ by Lemma 41
$\Longrightarrow$ $t\backslash\tau \in [\![h]\!][\![\Delta]\!]\rho\backslash\tau \parallel [\![g]\!][\![\Delta]\!]\rho\backslash\tau$ by $d1$
$\Longrightarrow$ $t\backslash\tau \in ([\![h]\!][\![\Delta]\!]\rho \parallel [\![g]\!][\![\Delta]\!]\rho)\backslash\tau$ by Corollary 5
$\Longrightarrow$ $t\backslash\tau \in [\![h \mid g]\!][\![\Delta]\!]\rho\backslash\tau$

e) $f \equiv h >y> g$
We can always assume $y \neq x$ because we can α-rename if needed.
$T = [\![(h >y> g) >x> let(x)]\!][\![\Delta]\!]\rho$
$= [\![h >y> (g >x> let(x))]\!][\![\Delta]\!]\rho$ by Lemma 15, Theorem 6
$= \bigcup_{s\in[\![h]\!][\![\Delta]\!]\rho}\ s \gg \lambda w.[\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat w]$
If $t \in T$, $\exists s \in [\![h]\!][\![\Delta]\!]\rho.\ t \in (s \gg \lambda w.[\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat w])$
If s contains no publications, trivial.
If $s = s_1!v s_2$ no publ. in s_1 then
$t \in s_1\,\tau\,((s_2 \gg \lambda w.[\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat w])$
$\quad \parallel [\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat v])$
Then,
$\exists t' \in (s_2 \gg \lambda w.[\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat w]) \parallel [\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat v]$
such that $t = s_1\,\tau\,t'$
and also there exist
$t_1 \in (s_2 \gg \lambda w.[\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat w])$
$t_2 \in [\![g >x> let(x)]\!][\![\Delta]\!]\rho[y = \flat v]$
such that $t' \in t_1 \parallel t_2$ $(e1)$
But then, $(t_2, g) \sqsubset (t, h >y> g)$

$\Longrightarrow \quad t_2\backslash\tau \in [\![g]\!][\![\Delta]\!]\rho[x = \flat v]\backslash\tau$ by IH $(e2)$

Also, by applying Lemma 54 $(|s_1| + 1)$ times we get

$\Delta, \Gamma \vdash \sigma h \xrightarrow{s_1}{}^{*} h'' \xrightarrow{!v} h'$ and $s_2 \in [\![h']\!][\![\Delta]\!]\rho$

$\Longrightarrow \quad t_1 \in [\![h' >y> (g >x> let(x))]\!][\![\Delta]\!]\rho$ by $e1$

$\Longrightarrow \quad t_1 \in [\![(h' >y> g) >x> let(x)]\!][\![\Delta]\!]\rho$ by Lemma 15, Theorem 6

Then, $\lceil t_1 \rceil < \lceil t \rceil \Rightarrow (t_1, h' >y> g) \sqsubset (t, h >y> g)$

$\Longrightarrow \quad t_1\backslash\tau \in [\![h' >y> g]\!][\![\Delta]\!]\rho\backslash\tau$ by IH $(e3)$

By $e1$, $t'\backslash\tau \in (t_1 \parallel t_2)\backslash\tau$

$\Longrightarrow \quad t'\backslash\tau \in t_1\backslash\tau \parallel t_2\backslash\tau$ by Lemma 41

$\Longrightarrow \quad t'\backslash\tau \in [\![h' >y> g]\!][\![\Delta]\!]\rho\backslash\tau \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat v]\backslash\tau$ by $e2, e3$

$\Longrightarrow \quad t'\backslash\tau \in ([\![h' >y> g]\!][\![\Delta]\!]\rho \parallel [\![g]\!][\![\Delta]\!]\rho[x = \flat v])\backslash\tau$ by Corollary 5

$\Longrightarrow \quad t'\backslash\tau \in [\![(h' >y> g) \mid [v/y]g]\!][\![\Delta]\!]\rho\backslash\tau$ by Lemma 36

$\Longrightarrow \quad \exists t'' \in [\![(h' >y> g) \mid [v/y]g]\!][\![\Delta]\!]\rho.\ t'\backslash\tau = t''\backslash\tau$

$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma((h' >y> g) \mid [v/y]g) \xrightarrow{t''}{}^{*} p$ by Theorem 10 $(e4)$

We know that $s_1\, !v \in [\![h]\!][\![\Delta]\!]\rho$ so by Theorem 10,

$\Delta, \Gamma \vdash \sigma h \xrightarrow{s_1}{}^{*} h'' \xrightarrow{!v} h'$

$\Longrightarrow \Delta, \Gamma \vdash \sigma(h >y> g) \xrightarrow{s_1}{}^{*} h'' >y> \sigma_{-y}g \xrightarrow{\tau} (h' >y> \sigma_{-y}g) \mid [v/y]\sigma_{-y}g$

But $\text{f.v.}(h') \subseteq \text{f.v.}(\sigma h) \Rightarrow \sigma h' \equiv h'$

$\Longrightarrow \quad \sigma((h' >y> g) \mid [v/y]g) \equiv (h' >y> \sigma_{-y}g) \mid [v/y]\sigma_{-y}g$

$\Longrightarrow \quad \Delta, \Gamma \vdash \sigma(h >y> g) \xrightarrow{t_3}{}^{*} p \quad$ where $\quad t_3 = s_1\tau\, t''$ by $e4$

$\Longrightarrow \quad s_1\tau\, t'' \in [\![h >y> g]\!][\![\Delta]\!]\rho$ by Theorem 9

$\Longrightarrow \quad (s_1\tau\, t'')\backslash\tau \in [\![h >y> g]\!][\![\Delta]\!]\rho\backslash\tau$

$\Longrightarrow \quad (s_1\tau\, t')\backslash\tau \in [\![h >y> g]\!][\![\Delta]\!]\rho\backslash\tau$

$\Longrightarrow \quad t\backslash\tau \in [\![h >y> g]\!][\![\Delta]\!]\rho\backslash\tau$

f) $f \equiv h$ **where** $y :\in g$

We can always assume $y \neq x$ because we can α-rename if needed.

$T = [\![(h \textbf{ where } y :\in g) >x> let(x)]\!][\![\Delta]\!]\rho =$

$= [\![(h >x> let(x)) \textbf{ where } y :\in g]\!][\![\Delta]\!]\rho$ by Lemma 17 and Theorem 6

If $t \in T$ then

$\exists\, t_1 \in \bigcup_{w \in Val} [\![h >x> let(x)]\!][\![\Delta]\!]\rho[y = \natural w],\ t_2 \in [\![g]\!][\![\Delta]\!]\rho.\ t \in t_1 <_y t_2$

$\Longrightarrow \quad t_2\backslash\tau \in [\![g]\!][\![\Delta]\!]\rho\backslash\tau$ $(f1)$

We look only at one case, the rest are similar:

Let $t_1 = t_{11}\,[v/y]\,t_{12}$, no recv. for y in t_{11}, $t_2 = t_{21}\,!v\,t_{22}$, no publ. in t_{21}

$\Longrightarrow \quad t \in (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/y])$

We know $\lceil t_1 \rceil = \lceil t_{11}\,[v/y]\,(t_{12}\backslash[v/y]) \rceil = \lceil t_{11}\,\tau\,(t_{12}\backslash[v/y]) \rceil \leqslant \lceil t \rceil$

$\Longrightarrow \quad (t_1, h) \sqsubset (t, h \textbf{ where } y :\in g)$

$\Longrightarrow \quad t_1\backslash\tau \in [\![h]\!][\![\Delta]\!]\rho[y = \natural v]\backslash\tau$ by IH

$\Longrightarrow \quad t_1\backslash\tau \in (\bigcup_{w \in Val} [\![h]\!][\![\Delta]\!]\rho[y = \natural w])\backslash\tau$

$\Longrightarrow \quad \exists\, t_1' \in \bigcup_{w \in Val} [\![h]\!][\![\Delta]\!]\rho[y = \natural w].\ t_1'\backslash\tau = t_1\backslash\tau$ $(f2)$

But $t \in t_1 <_y t_2$

$\Longrightarrow \quad t\backslash\tau \in (t_1 <_y t_2)\backslash\tau \quad \Longrightarrow \quad t\backslash\tau \in (t_1\backslash\tau <_y t_2\backslash\tau)\backslash\tau$

$\Longrightarrow \quad t\backslash\tau \in (t_1'\backslash\tau <_y t_2\backslash\tau)\backslash\tau$ by $f2$

$\Longrightarrow \quad t\backslash\tau \in (t_1' <_y t_2)\backslash\tau$ by Lemma 46

$\Longrightarrow \quad t\backslash\tau \in [\![h \textbf{ where } y :\in g]\!][\![\Delta]\!]\rho\backslash\tau$ $\square$

Lemma 56. *For all* s, f, ρ, Δ *such that*

1. f *is well-formed when the set of declarations is* Δ
2. s *is of the form* $s_1 !v\, s_2$ *and* $s \in \{\!|f|\!\}\{\!|\Delta|\!\}\rho$

we get $s_1(!v \parallel s_2) \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$

Proof. By induction on the number of publications in s.
Let $s = s_1 !v\, s_2$, no publ. in s_1
Then, $!v$ is the first publication in s.
By Lemma 55, $s \dot{\gg} \lambda u.\{!u\}_{\mathrm{p}} \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$

$\implies$ $s_1((s_2 \dot{\gg} \lambda u.\{!u\}_{\mathrm{p}}) \parallel \{!v\}_{\mathrm{p}}) \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$
$\implies$ $s_1(s_2 \parallel !v) \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$ by Lemma 44
We proved the lemma for the first publication of s, which may be the only publication.
If there are more publications in s, then s_2 is of the form $s_2 = s_{21}\, !w\, s_{22}$ where w is not necessarily the first publication in s_2.
Now, it suffices to show that $s_1\, !v\, s_{21}(!w \parallel s_{22}) \subseteq \{\!|f|\!\}\{\!|\Delta|\!\}\rho$ (I)
By Lemma 55, there exists $s' \in [\![f]\!][\![\Delta]\!]\rho$ such that $s = s'\backslash\tau$ so there exist s'_1, s'_{21}, s'_{22} such that $s_1 = s'_1\backslash\tau, s_{21} = s'_{21}\backslash\tau, s_{22} = s'_{22}\backslash\tau$.
By applying Lemma 54 $|s'_1| + 1$ times we get

$\Delta, \Gamma \vdash \sigma f \xrightarrow{s'_1}{}^{*} f'' \xrightarrow{!v} f'$ and $s'_{21}\, !w\, s'_{22} \in [\![f']\!][\![\Delta]\!]\rho$ (II)
Note that we can do that because, by Lemma 50, for all processes f_i during the reduction of σf it holds that $\sigma f_i \equiv f_i$
By II and Theorem 11 we get that $s_2 \in \{\!|f'|\!\}\{\!|\Delta|\!\}\rho$ and s_2 has fewer publications than s, so by IH, $s_{21}(!w \parallel s_{22}) \subseteq \{\!|f'|\!\}\{\!|\Delta|\!\}\rho$
By I, it suffices to show that
for all $t \in s_{21}(!w \parallel s_{22})$ it holds that $s_1\, !v\, t \in \{\!|f|\!\}\{\!|\Delta|\!\}\rho$ (III)
For any of these traces t we find by Theorem 11 that there exists $t' \in [\![f']\!][\![\Delta]\!]\rho$ such that $t'\backslash\tau = t$ and we already know that $\sigma f' \equiv f'$ so by Theorem 10
$\Delta, \Gamma \vdash f' \xrightarrow{t'}{}^{*} f'''$

$\implies$ $\Delta, \Gamma \vdash \sigma f \xrightarrow{t''}{}^{*} f'''$ where $t'' = s'_1\, !v\, t'$ by II
$\implies$ $s'_1\, !v\, t' \in [\![f]\!][\![\Delta]\!]\rho$ by Thm. 9
$\implies$ $s_1\, !v\, t \in \{\!|f|\!\}\{\!|\Delta|\!\}\rho$ which is what we wanted in III. □