A compositional trace semantics for Orc

Dimitrios Vardoulakis and Mitchell Wand

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

Abstract. Orc [6] is a language for task orchestration. It has a small set of primitives, but it is sufficient to express many useful programs succinctly. We show that the operational and denotational semantics given in Kitchin et. al [6] do not agree, by giving a counterexample to their Theorem 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.

1 Introduction

Orc [6] 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 [6], [7]). 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[6] have developed operational and trace-based semantics for Orc.

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

– We show that the trace-based semantics of Orc is flawed, and develop a denotational semantics which we prove sound and adequate with respect to the operational semantics.
– In [6], 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.

Due to space limitations, we decided to present our semantics first, and then introduce only the necessary machinery from [6] to show the error.

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. Site calls are *strict*: the process $M(x)$ has no transitions. 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 $(\mathit{Factorize}(N) \mid \mathit{RedditFeed}(\mathit{today}))$ 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, $((\mathit{Factorize}(N) \mid \mathit{RedditFeed}(\mathit{today})) >x> \mathit{Print}(x))$ may print twice, if both $\mathit{Factorize}(N)$ and $\mathit{RedditFeed}(\mathit{today})$ 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 $(f \ \mathbf{where}\, x :\in g)$ 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 $(\mathit{Print}(x) \ \mathbf{where}\, x :\in (\mathit{Factorize}(N) \mid \mathit{RedditFeed}(\mathit{today})))$ will either print the prime factors of N or today's tech–news, maybe none, but not both. Recall that site calls are strict, thus $\mathit{Print}(x)$ has no transitions. However, placed in a context that can provide a value for x (as in this example), $\mathit{Print}(x)$ is no longer inert.[1]

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

$$\mathit{DOS}(x) \triangleq \mathit{Ping}(x) \mid \mathit{DOS}(x)$$

This is the implementation of a simple denial-of-service attack; the process $\mathit{DOS}(\mathit{ip})$ 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 $\text{f.v.}(e) \subseteq \{x\}$. The process $\mathbf{0}$ is the inert process, a site which never responds. 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.

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

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

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*
$\vert$	τ	*internal*
$\vert$	$M_k(v)$	*site call*
$\vert$	$k?v$	*site response*
$\vert$	$[v/x]$	*receive*

Let's take a closer look at the rules. 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, thus $M(x)$ 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 $\mathbf{0}$. Rule SITEC-VAR (and similarly LET-VAR and DEF-VAR) reflects 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.

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

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

(SITERET) $$\frac{}{\Delta, \Gamma \vdash ?k \overset{k?v}{\rightarrow}\ let(v)}$$

(LET) $$\frac{}{\Delta, \Gamma \vdash\ let(v) \overset{!v}{\rightarrow}\ \mathbf{0}}$$

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

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

(DEF-VAR) $$\frac{}{\Delta, \Gamma \vdash\ E_i(x) \overset{[v/x]}{\rightarrow}\ E_i(v)}\ \begin{matrix}(E_i(x) \triangleq f_i) \in \Delta, \\ \Gamma(x) = v\end{matrix}$$

(SYM-L) $$\frac{\Delta, \Gamma \vdash\ f \overset{a}{\rightarrow}\ f'}{\Delta, \Gamma \vdash\ f \mid g \overset{a}{\rightarrow}\ f' \mid g}$$

(SYM-R) $$\frac{\Delta, \Gamma \vdash\ g \overset{a}{\rightarrow}\ g'}{\Delta, \Gamma \vdash\ f \mid g \overset{a}{\rightarrow}\ f \mid g'}$$

(ASYM-L) $$\frac{\Delta, \Gamma \vdash\ f \overset{a}{\rightarrow}\ f'}{\Delta, \Gamma \vdash\ f \textbf{ where } x :\in g \overset{a}{\rightarrow}\ f' \textbf{ where } x :\in g}\ a \neq [v/x]$$

(ASYM-R) $$\frac{\Delta, \Gamma \vdash\ g \overset{a}{\rightarrow}\ g'}{\Delta, \Gamma \vdash\ f \textbf{ where } x :\in g \overset{a}{\rightarrow}\ f \textbf{ where } x :\in g'}\ a \neq !v$$

(ASYM-P) $$\frac{\Delta, \Gamma \vdash\ g \overset{!v}{\rightarrow}\ g'}{\Delta, \Gamma \vdash\ f \textbf{ where } x :\in g \overset{\tau}{\rightarrow}\ [v/x]f}$$

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

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

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). You 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, no matter if x is in Γ, f cannot proceed with a receive event for x (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

$$\begin{array}{l}
\Delta, \Gamma \vdash\ let(x) \overset{[2/x]}{\rightarrow} let(2) \\
\overset{\text{SEQ}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ let(x) >x> M(x) \overset{[2/x]}{\rightarrow} let(2) >x> M(x) \\
\overset{\substack{\text{LET} \\ \text{SEQ-P}}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ let(2) >x> M(x) \overset{\tau}{\rightarrow} (\mathbf{0} >x> M(x)) \mid M(2) \\
\overset{\substack{\text{SITEC} \\ \text{SYM-R}}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ (\mathbf{0} >x> M(x)) \mid M(2) \overset{M_k(2)}{\rightarrow} (\mathbf{0} >x> M(x)) \mid ?k \\
\overset{\substack{\text{SITERET} \\ \text{SYM-R}}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ (\mathbf{0} >x> M(x)) \mid ?k \overset{k?11}{\rightarrow} (\mathbf{0} >x> M(x)) \mid let(11) \\
\overset{\substack{\text{LET} \\ \text{SYM-R}}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ (\mathbf{0} >x> M(x)) \mid let(11) \overset{!11}{\rightarrow} (\mathbf{0} >x> M(x)) \mid \mathbf{0}
\end{array}$$

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.

4 Denotational Semantics

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

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

Trace sets are closed under prefix. Also, we are concerned with traces of finite length only; an infinite trace is represented by the set of all its finite prefixes.

Traces: $Event^*$, a discrete CPO.
P: the set of all non-empty prefix-closed sets of finite traces,
a CPO under inclusion.
Val: the set of all values, a discrete CPO.
Var: the set of all variable names, a discrete CPO

We use two different kinds of bindings for variables in *Env*. That is because we want to differentiate between a variable x bound in f **where** $x :\in g$ versus $f >x> g$ or $E_i(x) \triangleq f_i$. In asymmetric composition, the evaluation of f may start before f has a value for x. Then, the parts of f that depend on x will block. For a trace t of f, we want to know which part of it depends on x. For this reason, we "mark" the usage of x with a receive event, e.g. $t \equiv t_1[v/x]t_2$ where juxtaposition means trace concatenation. Now, we can deduce that the events in t_1 happen independently of x. There is no need for that in $f >x> g$ or $E_i(x) \triangleq f_i$ because x always has a value when g or f_i is evaluated.

$$\begin{array}{l}
GetVal = \{\, \natural v \mid v \in Val\} \\
GotVal = \{\, \flat v \mid v \in Val\} \\
Env = [\,Var \rightarrow (GetVal \cup GotVal \cup \{\bot\})] \\
NoRecv = \{\, S \mid S \in P \quad \wedge \quad \forall t \in S.\ \text{no receives in } t\} \\
Fenv = ([\,Val \rightarrow NoRecv])^k
\end{array}$$

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 prefixing 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 $\rho(x) = \perp$) then $let(x)$ behaves like $\mathbf{0}$. If the value for x was received "now", i.e. $\rho(x) = \natural v$, then a receive event precedes the publication. If the value for x was received "earlier", the trace does not contain a receive event.

The meaning functions for site calls are quite similar. Note the many possible responses to the same call. We invite you to check that, for simple processes like $M(4)$ and $let(x)$, the traces coincide with what we get from the operational semantics. We will prove that true for all Orc processes.

The traces of $E_i(v)$ are independent of the environment (remember that x *only* 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 interleaving (or else *merging*) the traces of the constituent processes.

The denotation of $h >x> g$ can be demystified by observing the operational behaviour 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

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

Fig. 4. Trace Semantics of Orc

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:
$T_1 \parallel T_2 \triangleq \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \parallel t_2$

Prefixing:

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

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

Sequencing combinator:

$$s \gg F = \begin{cases} \{s\} & \text{no publ. in } s \\ s_1\,\tau\,((s_2 \gg F) \parallel F(v)) & s \equiv s_1!vs_2\text{ , no publ. in } s_1 \end{cases}$$

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:
$T_1 <_x T_2 = \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 <_x t_2$

Note: ρ_0 is an environment such that $\forall x.\rho_0(x) = \perp$

Fig. 5. Various Definitions

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 now 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 only need talk about the denotation of the declarations. Consider $E_i(x) \triangleq f_i$. We can find its traces if we know the traces of f_i in a suitable *Fenv*:

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

$\hat{\Delta}$ is an *Fenv* transformer, since it consumes an *Fenv* and produces another *Fenv*. Additionally, it is a continuous function because it is composed of continuous functions. *Fenv* is a CPO with bottom element $(\lambda v.\{\varepsilon\})^k$. Therefore, $\hat{\Delta}$ has a least fixed point which we take to be the denotation of the declarations:

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

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

Theorem 3 (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*

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

Theorem 4 (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*

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

The relation $\rightarrow^*$ is the reflexive and transitive closure of $\rightarrow$.

5 Flaws of the previous trace semantics

The operational semantics of Section 3 differs slightly from the previously proposed operational semantics.[2] 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 [6] wrong. Here, we only

(LET) $$\frac{}{let(v) \xrightarrow{!v} \mathbf{0}}$$

(ASYM1N) $$\frac{f \xrightarrow{a} f'}{f \textbf{ where } x :\in g \xrightarrow{a} f' \textbf{ where } x :\in g}$$

(ASYM1V) $$\frac{g \xrightarrow{!v} g'}{f \textbf{ where } x :\in g \xrightarrow{\tau} [v/x]f}$$

(ASYM2) $$\frac{g \xrightarrow{a} g' \qquad a \neq !v}{f \textbf{ where } x :\in g \xrightarrow{a} f \textbf{ where } x :\in g'}$$

(SUBST) $$\frac{}{f \xrightarrow{[v/x]} [v/x]f}$$

Fig. 6. Operational Semantics of Orc$_1$

present a subset of the semantics which suffices to show the error. Thus, we are not concerned with recursive definitions.

All but the last rule in Fig. 6 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.* If $f \xrightarrow{s}{}^{*} f'$ then they obtain a trace of f by removing the τ events from s. Let $\langle f \rangle$ denote f's set of traces. The objective is to *prove* compositionality, i.e. that $\langle f \textbf{ where } x :\in g\rangle$ can be defined in terms of $\langle f \rangle$ and $\langle g \rangle$.

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

Constraint: No receive event for x in t_2

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

The **where** operator is defined for traces and lifted for trace sets. The $|$ operator is similar to our merge operator. Its precise definition is not needed, we only need to know that $t \mid \varepsilon = \{t\}$. The theorem to prove now is:

Theorem 5. $\langle f \textbf{ where } x :\in g\rangle = \langle f \rangle \textbf{ where } x :\in \langle g \rangle$

The following counterexample refutes this theorem:
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, thus $t \notin \langle h \rangle$
Therefore, $\langle f \textbf{ where } x :\in g\rangle \neq \langle f \rangle \textbf{ where } x :\in \langle g \rangle$

[2] For disambiguation, in this section we will refer to Orc as presented in [6] as Orc$_1$.

We saw that the trace set of an Orc_1 process is not defined correctly in terms of the traces of its sub-processes. Our intuition is that the error stems from the non-restrictive usage of substitutions (rule SUBST).

6 Conclusions

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

In this paper we presented a denotational trace-based semantics for Orc, a language for task orchestration. We pointed out the deficiencies of the previously proposed trace semantics [6] and proved the correctness of ours. Other semantic treatments for Orc can be found in [7], [4].

In a forthcoming paper, we use strong bisimulation to show various equivalences between Orc processes and we present a trace-based semantics insensitive to internal events. In the future we want to investigate the properties of processes in the presence of timeouts and propose a timed semantics for Orc.

References

1. Alexandre Alves, Assaf Arkin, et al. Web services business process execution language version 2.0. Technical report, April 2007.
2. 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.
3. 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.
4. Tony Hoare, Galen Menzel, and Jayadev Misra. A tree semantics for an orchestration language, August 2004. Lecture Notes for NATO summer school.
5. Nickolas Kavantzas, David Burdett, et al. Web services choreography description language version 1.0. Technical report, November 2005.
6. 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.
7. Jayadev Misra and William R. Cook. Computation orchestration: A basis for wide-area computing. *Software and Systems Modeling*, 6(1):83–110, 2007.
8. Mirko Viroli. Towards a formal foundation to orchestration languages. *Electr. Notes Theor. Comput. Sci.*, 105:51–71, 2004.

A Various Definitions

Definition 1. *Concatenate a trace and a trace-set*
$s\,T \triangleq \{\, s\,t \mid t \in T\}$

Definition 2. *Concatenate trace-sets*
$T_1\,T_2 \triangleq \{\, t_1 t_2 \mid t_1 \in T_1, t_2 \in T_2\}$

Definition 3. *Remove event 'a' from a trace*

$$t\backslash a \triangleq \begin{cases} \varepsilon & t = \varepsilon \\ t'\backslash a & t = at' \\ a'\,t'\backslash a & t = a't' \textit{ and } a \neq a' \end{cases}$$

Definition 4. *Remove event from a trace-set*
$T\backslash a \triangleq \{\, t\backslash a \mid t \in T\}$

Definition 5. *Merge for traces*

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

Definition 6. *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 7. *Prefixing*

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

Definition 8. *Prefixing for trace-sets*
$S_{\mathrm{p}} \triangleq \bigcup_{s \in S} s_{\mathrm{p}}$

Definition 9. *Extend-env: Env $\times$ (Val $\times$ (GetVal $\cup$ GotVal)) $\rightarrow$ Env*
$\rho[x = u] \triangleq (\rho - \{(x, w)\}) \cup \{(x, u)\}$ *,where* $\rho(x) = w$

Definition 10. *Alternate merge*

$$t_1 \breve{\parallel} t_2 \triangleq \begin{cases} \{t_1\} & t_2 = \varepsilon \\ \{t_2\} & t_1 = \varepsilon \\ (t_1' \breve{\parallel} t_2)a \cup (t_1 \breve{\parallel} t_2')b & t_1 = t_1'a \textit{ and } t_2 = t_2'b \end{cases}$$

Definition 11. *Alternate merge for trace-sets*
$T_1 \breve{\parallel} T_2 \triangleq \bigcup_{t_1 \in T_1, t_2 \in T_2} t_1 \breve{\parallel} t_2$

Definition 12. $\rho_{-x}(y) = \begin{cases} \bot & y = x \\ \rho(y) & y \neq x \end{cases}$

Note 1 *ρ_0 is an environment such that $\forall x.\rho_0(x) = \bot$*

Note 2 *$a \,\hat{\in}\, t$ means that trace t contains event a. $a \,\hat{\notin}\, t$ means that trace t does not contain event a.*

Definition 13. *Ordering of pairs of integers*
$(i, j) \sqsubset (k, l)$ *when* $(i < k) \vee (i = k \wedge j < l)$

B Continuity Proofs

Lemma 3. *The union of prefix-closed sets is prefix-closed* □

Lemma 4. *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 3 and is an ub of X. Let B' be an ub of X
$\implies \quad \forall S \in X . S \subseteq B'$
$\implies \quad \bigcup_{S \in X} S \subseteq B'$
$\implies \quad \bigsqcup X = B$ □

Lemma 5. *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 6. *Extend-env is continuous* □

Note 7 *$[Val \to NoRecv]$ 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 8 *$Fenv$ 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 9 *Similar results to Note 7 hold for $[Val \to P]$, $[Val \to Pow(Traces)]$*

Lemma 10. *$\gg$: $Traces \times [Val \to Pow(Traces)] \to Pow(Traces)$ 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,
$\implies \quad s \gg \bigsqcup X = \{s\} = \bigsqcup_{F \in X} (s \gg F)$
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 9
$= s_1 \tau \, ((\bigcup_{F \in X} s_2 \gg F) \parallel \bigcup_{F \in X} F(v))$ by *IH*
$= s_1 \tau \, \bigcup_{F \in X} ((s_2 \gg F) \parallel F(v))$ by Lemma 5
$= \bigcup_{F \in X} s_1 \tau \, ((s_2 \gg F) \parallel F(v))$
$= \bigsqcup_{F \in X} s \gg F$ □

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

Lemma 11. *Prefixing* : $Pow(Traces) \rightarrow P$ *is continuous* $\square$

Note 12 $<_x$: $Traces \times Traces \rightarrow Pow(Traces)$ *is continuous* $\square$

Corollary 2. $<_x$: $Pow(Traces) \times Pow(Traces) \rightarrow Pow(Traces)$
is continuous $\square$

Note 13 *All the functions proved to be continuous are also monotonic*

Theorem 6. *For all f, $[\![f]\!]$ is continuous*

Proof. We know that $[\![f]\!] \in [Fenv \rightarrow [Env \rightarrow P]]$. We will show the continuity of $[(Fenv \times Env) \rightarrow P]$ 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) $let(v)$
$\Longrightarrow\ [\![f]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = \{!v\}_{\mathrm{p}} = \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![let(v)]\!]\varphi\rho$

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

c) $let(x)$
$[\![let(x)]\!](\bigsqcup X_\varphi)(\bigsqcup X_\rho) = \bigsqcup_{\varphi\in X_\varphi} [\![let(x)]\!]\varphi(\bigsqcup X_\rho)$ $\quad (c1)$
Cases on X_ρ:
- If $\exists\rho\in X_\rho.\ \rho(x) = \perp$ then $\forall\rho\in X_\rho.\ \rho(x) = \perp$ 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 9
$= \bigsqcup_{\rho\in X_\rho} \bigcup_{\varphi\in X_\varphi} \{\tau\ t \mid t \in \varphi_i(v)\}_{\mathrm{p}}$ by Lemma 11
$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![E_i(v)]\!]\varphi\rho$

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 5
$= \bigsqcup_{\varphi\in X_\varphi} \bigsqcup_{\rho\in X_\rho} [\![h \mid g]\!]\varphi\rho$

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 6

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

Then, by Note 9,

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

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 6 and *IH*,

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

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

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

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

C Prefix-Closure Proofs

Lemma 14. $t_1 \parallel t_2 = t_1 \mathbin{\breve{\parallel}} t_2$

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 $t_2 = b_1 t_2' b_2$
$\implies \quad t_1 \parallel t_2 = a_1(t_1' a_2 \parallel t_2) \cup b_1(t_1 \parallel t_2' b_2)$
$= a_1(t_1' a_2 \mathbin{\breve{\parallel}} t_2) \cup b_1(t_1 \mathbin{\breve{\parallel}} t_2' b_2)$ 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 $\breve{\parallel}$ interchangeably.

Lemma 15. $T_1, T_2 \in P$ *implies* $T_1 \parallel T_2 \in P$

Proof. By Lemma 14, 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 $t_2 = t_2' b$
$\implies \quad t \in ((t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)$
$\implies \quad t_{\mathrm{p}} \subseteq ((t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)_{\mathrm{p}}$
$\implies \quad t_{\mathrm{p}} \subseteq ((t_1' \mathbin{\breve{\parallel}} t_2)_{\mathrm{p}} \cup (t_1' \mathbin{\breve{\parallel}} t_2)a \cup (t_1 \mathbin{\breve{\parallel}} t_2')_{\mathrm{p}} \cup (t_1 \mathbin{\breve{\parallel}} t_2')b)$ (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 16. *If* $F \in [\mathit{Val} \to P]$ *and* $s \in \mathit{Traces}$, *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,
$\implies \quad \bigcup_{s' \in s_{\mathrm{p}}} s' \gg F = \bigcup_{s' \in s_{\mathrm{p}}} \{s'\} = s_{\mathrm{p}} \in P$
If $s = s_1 !v s_2$ and no publications in s_1,
$\implies \quad \bigcup_{s' \in s_{\mathrm{p}}} s' \gg F = (\bigcup_{s' \in (s_1)_{\mathrm{p}}} s' \gg F) \cup (s_1 !v \gg F) \cup (\bigcup_{s' \in s_1 !v (s_2)_{\mathrm{p}}} s' \gg F)$
$= (s_1)_{\mathrm{p}} \cup \{s_1 \tau\} \cup s_1 \tau((\bigcup_{s' \in (s_2)_{\mathrm{p}}} s' \gg F) \parallel F(v))$
$= \{s_1 \tau\}_{\mathrm{p}} \cup s_1 \tau((\bigcup_{s' \in (s_2)_{\mathrm{p}}} s' \gg F) \parallel F(v))$
$\implies \quad$ suffices to show that $((\bigcup_{s' \in (s_2)_{\mathrm{p}}} s' \gg F) \parallel F(v)) \in P$
which, by Lemma 15, 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* $F \in [\mathit{Val} \to P]$, *then* $(\bigcup_{s \in T} s \gg F) \in P$ □

Lemma 17. $T_1, T_2 \in P$ *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)
$\Longrightarrow\quad t \in \bigcup_{t'_1 \in (t_1)_\mathrm{p}, t'_2 \in (t_2)_\mathrm{p}} t'_1 \parallel t'_2 = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$ by Note 13
$\Longrightarrow\quad t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p})_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$ by Lemma 15
By 1, $(t_1)_\mathrm{p} <_x (t_2)_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_2)_\mathrm{p}$
$\Longrightarrow\quad t_\mathrm{p} \subseteq (t_1)_\mathrm{p} <_x (t_2)_\mathrm{p}$
$\Longrightarrow\quad t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 13

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}
$\Longrightarrow\quad t \in (t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p}$ by Note 13
$\Longrightarrow\quad t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p})_\mathrm{p} = (t_1)_\mathrm{p} \parallel (t_{21}\tau)_\mathrm{p}$ by Lemma 15
$\Longrightarrow\quad t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} \parallel (t_{21})_\mathrm{p}) \cup ((t_1)_\mathrm{p} \parallel \{t_{21}\tau\})$
$\Longrightarrow\quad t_\mathrm{p} \subseteq ((t_1)_\mathrm{p} <_x (t_{21})_\mathrm{p}) \cup ((t_1)_\mathrm{p} <_x \{t_{21}!v\})$
$\Longrightarrow\quad t_\mathrm{p} \subseteq (t_1)_\mathrm{p} <_x (t_{21}!v)_\mathrm{p}$
$\Longrightarrow\quad t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 13

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}
$\Longrightarrow\quad t_\mathrm{p} \in (t_{11} \parallel t_{21}\tau)_\mathrm{p} \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$
$\Longrightarrow\quad t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} \parallel \{t_{21}\tau\}_\mathrm{p})_\mathrm{p} \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$ by Note 13
$\Longrightarrow\quad t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} \parallel \{t_{21}\tau\}_\mathrm{p}) \cup (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[v/x])_\mathrm{p}$ by Lemma 15
By the previous case, this can be written
$\Longrightarrow\quad t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}) \cup (t_{11}[v/x]\{t_{12}\}_\mathrm{p} <_x \{t_{21}!v\})$
$\Longrightarrow\quad t_\mathrm{p} \in (\{t_{11}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}) \cup (t_{11}[v/x]\{t_{12}\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p})$ by Note 13
$\Longrightarrow\quad t_\mathrm{p} \subseteq \{t_1\}_\mathrm{p} <_x \{t_{21}!v\}_\mathrm{p}$
$\Longrightarrow\quad t_\mathrm{p} \subseteq T_1 <_x T_2$ by Note 13 □

Theorem 7. *For all* f, $[\![f]\!]\varphi\rho \in P$

Proof. By structural induction on f, using Lemmas 15, 17 and Corollary 3 □

D Denotational Lemmas

Lemma 18 (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, $[\![h \mid g]\!]\varphi\rho = [\![h]\!]\varphi\rho \,\|\, [\![g]\!]\varphi\rho = [\![h \mid g]\!]\varphi\rho[x = \flat v] =$
$= [\![h \mid g]\!]\varphi\rho[x = \natural w]$

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
$\implies$ $[\![h]\!]\varphi\rho = [\![h]\!]\varphi\rho[x = \flat v] = [\![h]\!]\varphi\rho[x = \natural w]$ 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
$\implies$ $[\![g]\!]\varphi\rho = [\![g]\!]\varphi\rho[x = \flat v] = [\![g]\!]\varphi\rho[x = \natural w]$ by *IH* $(e1)$
Then, $[\![h \textbf{ where } x :\in g]\!]\varphi\rho[x = \natural w] =$
$= (\bigcup_{v \in Val} [\![h]\!]\varphi\rho[x = \natural w][x = \natural v]) <_x [\![g]\!]\varphi\rho[x = \natural w]$
$= (\bigcup_{v \in Val} [\![h]\!]\varphi\rho[x = \natural v]) <_x [\![g]\!]\varphi\rho$ by $e1$ and def. of extenv-env
$= [\![h \textbf{ where } x :\in g]\!]\varphi\rho$
Similarly, $[\![h \textbf{ where } x :\in g]\!]\varphi\rho[x = \flat v] = [\![h \textbf{ where } x :\in g]\!]\varphi\rho$ □

Lemma 19 (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 18

b) $f \equiv let(x)$
$\implies$ $[v/x]f = let(v)$
$\implies$ $[\![[v/x]f]\!]\varphi\rho = \{!v\}_{\text{p}} = [\![let(x)]\!]\varphi\rho[x = \flat v]$

c) f is $M(x)$ or $E_i(x)$, as above

d) $f \equiv h \mid g$
$[\![[v/x]f]\!]\varphi\rho = [\![[v/x]h]\!]\varphi\rho \,\|\, [\![[v/x]g]\!]\varphi\rho$
$= [\![h]\!]\varphi\rho[x = \flat v] \,\|\, [\![g]\!]\varphi\rho[x = \flat v]$ 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** $y :\in g, x \neq y$)
$[\![[v/x]f]\!]\varphi\rho = [\![h$ **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*
$= [\![h$ **where** $x :\in g]\!]\varphi\rho[x = \natural v]$ □

Lemma 20. *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 21. *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 18. If x is free in f,

a) $f \equiv let(x)$
$\Longrightarrow$ $[\![let(x)]\!]\varphi\rho[x = \natural v] = \{[v/x]\,!v\}_{\text{p}}$
$[v/x]\,\hat{\notin}\, t \;\therefore\; t = \varepsilon \;\therefore\; t \in [\![let(x)]\!]\varphi\rho$
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 $t_2 \in [\![g]\!]\varphi\rho$
$\Longrightarrow$ $t \in [\![h \mid g]\!]\varphi\rho$
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 $t \in s \gg \lambda w.[\![g]\!]\varphi\rho[x = \natural v][x = \flat w]$
$\Longrightarrow$ $t \in s \gg \lambda w.[\![g]\!]\varphi\rho[x = \flat w]$ $(d1)$
By Lemma 20, $[v/x]$ 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 $t \in [\![h$ **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'
$\Longrightarrow$ $t \in (t_{11} \parallel t_{21}\tau)(t_{12}\backslash[u/x])$ $(e2)$
But then, $[v/x]\,\hat{\notin}\, t$ means $[v/x]\,\hat{\notin}\, t_{21}$
and by Theorem 7, $t_{21}\,!u \in [\![g]\!]\varphi\rho[x = \natural v]$
$\Longrightarrow$ $t_{21}\,!u \in [\![g]\!]\varphi\rho$ by *IH* $(e3)$
By $e1$, $e2$ and $e3$ we get the desired result □

Lemma 22. *If* $\rho(x) =\perp$ *then* $[\![f]\!]\varphi\rho \subseteq [\![f]\!]\varphi\rho[x = \natural v]$ □

Lemma 23. *If* $\rho(x) =\perp$ *then* $[\![f]\!]\varphi\rho \subseteq [\![f]\!]\varphi\rho[x = \flat v]$ □

Lemma 24. $(t_1 \parallel t_2)\backslash a = t_1\backslash a \parallel t_2\backslash a$

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 25. *Let $s \in Traces$ and $F : Val \to Pow(Traces)$.
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 26. $(t_1 <_y t_2)\backslash[v/x] = t_1\backslash[v/x] <_y t_2\backslash[v/x]$, *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, $t_1 <_y t_2 = t_1 \parallel t_2$
$\implies$ holds by Lemma 24
b) no recv. for x in t_1, $t_2 = t_{21}!w\, t_{22}$, no publ. in t_{21} $t_1 <_y t_2 = t_1 \parallel t_{21}\,\tau$
$\implies$ holds by Lemma 24
c) $t_1 = t_{11}[w/y]\, t_{12}$, $[w/y] \hat{\notin} t_{11}$, $t_2 = t_{21}!w\, t_{22}$, no publ. in t_{21},
$t_1 <_y t_2 = (t_{11} \parallel t_{21}\,\tau)(t_{12}\backslash[w/y])$ $(c1)$
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 27. $[\![f]\!]\varphi\rho[x=\flat v] = ([\![f]\!]\varphi\rho[x=\natural v])\backslash[v/x]$

Proof. By structural induction on f
If x is not free in f, by Lemma 18 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 20 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 20, $[v/x]$ 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 25
$= (\bigcup_{s\in[\![h]\!]\varphi\rho[x=\natural v]} s \gg [\![g]\!]\varphi\rho[x=\natural v][x=\flat w])\backslash[v/x]$
$= ([\![h >x> g]\!]\varphi\rho[x=\natural v])\backslash[v/x]$

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 26
$= (\bigcup_{t_1\in T_1, t_2\in T_2} t_1 <_x t_2)\backslash[v/x]$
$= (T_1 <_x T_2)\backslash[v/x]$
$= (\bigcup_{w\in Val}[\![h]\!]\varphi\rho[x=\natural w] <_x [\![g]\!]\varphi\rho[x=\natural v])\backslash[v/x]$
$= ([\![h$ **where** $x :\in g]\!]\varphi\rho[x=\natural v])\backslash[v/x]$ □

E Operational Lemmas

Lemma 28. *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 29. *If* $\Delta, \Gamma[x = v] \vdash\ f \xrightarrow{a} f'$ *and* $a \neq [v/x]$ *then*

$$\Delta, \Gamma' \vdash\ f \xrightarrow{a} f' \quad \textit{where } \Gamma'(y) = \begin{cases} \Gamma(y) & x \neq y \\ \textit{unspecified/anything} & x = y \end{cases}$$

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) $\dfrac{\Delta, \Gamma[x = v] \vdash\ f \xrightarrow{a} f'}{\Delta, \Gamma[x = v] \vdash\ f \textbf{ where } y :\in g \xrightarrow{a} f' \textbf{ where } y :\in g}\ {\scriptstyle \begin{array}{l} a\neq[v/x] \\ a\neq[v'/y] \end{array}}$
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 30. *$\Delta, \Gamma \vdash\ f \xrightarrow{a} f'$ implies $f.v.(f') \subseteq f.v.(f)$*

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

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

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

Lemma 31. *If $x \notin f.v.(f)$ then for any v, Γ $\quad \Delta, \Gamma \vdash\ f \overset{[v/x]}{\not\to} f'$*

Proof. By structural induction on f. □

F Soundness - Adequacy

Lemma 32. *If* $\Delta, \Gamma \vdash f \xrightarrow{a} f'$ *and* $t \in [\![f']\!][\![\Delta]\!]\rho$ *then* $at \in [\![f]\!][\![\Delta]\!]\rho$ *where* $\rho = \rho_0[x = \natural v_1] \ldots [x_m = \natural v_m]$ *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
$[\![?k]\!][\![\Delta]\!]\rho = \{\, k?w\, !w \mid w \in \mathit{Val}\}_{\mathrm{p}}$
Consider only the case when $t = k?w\, !w$
Then, $(M_k(v)\, k?w\, !w) \in [\![M(v)]\!][\![\Delta]\!]\rho$

b) (SITEC-VAR) $\dfrac{}{\Delta, \Gamma \vdash M(x) \xrightarrow{[v/x]} M(v)}$ $\Gamma(x) = v$
$[\![M(v)]\!][\![\Delta]\!]\rho = \{\, M_k(v)\, k?w\, !w \mid w \in \mathit{Val}\,, k \text{ fresh}\}_{\mathrm{p}}$
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 $t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho \overset{\text{Lem. 19}}{\Longrightarrow} t \in [\![f_i]\!][\![\Delta]\!]\rho[x = \flat v]$ $\quad (d1)$
Also, $[\![E_i(v)]\!][\![\Delta]\!]\rho = \{\, \tau\, t' \mid t' \in [\![\Delta]\!]_i(v)\}_{\mathrm{p}}$
where $[\![\Delta]\!]_i(v) = [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]$ $\quad (d2)$
By $d2$, it suffices to show that $t \in [\![f_i]\!][\![\Delta]\!]\rho_0[x = \flat v]$, which holds by $d1$ and Lemma 18, 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 $t \in t_1 \parallel t_2$ $\quad (e1)$
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
$t_1 \in \bigcup_{v \in \mathit{Val}} [\![h']\!][\![\Delta]\!]\rho[x = \natural v], t_2 \in [\![g]\!][\![\Delta]\!]\rho$ such that $t \in t_1 <_x t_2$ $\quad (g1)$
Also, by Lemma 28, $\quad \Delta, \Gamma[x = w] \vdash h \xrightarrow{a} h' \quad$ for any w $\quad (g2)$
Cases depending on which branch of the definition of $<_x$ was used for t:

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

- 2nd 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 $g1, g6$
- 3rd 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 20
  $\Longrightarrow$ 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 $g1, g8$
- 4th branch was used, $t = \varepsilon$
  $\varepsilon \in [\![g]\!][\![\Delta]\!]\rho$, $\varepsilon \in \bigcup_{v\in Val}[\![h']\!][\![\Delta]\!]\rho[x = \natural v]$ by Thm. 7
  $\Longrightarrow$ $a \in \bigcup_{v\in Val}[\![h]\!][\![\Delta]\!]\rho[x = \natural v]$ 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 19
$\Longrightarrow$ $\exists\, t' \in [\![h]\!][\![\Delta]\!]\rho[x = \natural v].\ t = t'\backslash[v/x]$ by Lemma 27 $(i1)$
$\varepsilon \in [\![g']\!][\![\Delta]\!]\rho$ by Thm. 7
$\Longrightarrow$ $!v \in [\![g]\!][\![\Delta]\!]\rho$ 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 19, $t_2 \in [\![g]\!][\![\Delta]\!]\rho[x = \flat u]$ $(k2)$
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 8 (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. 7
- 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 32, $a\, t' \in [\![\sigma f]\!][\![\Delta]\!]\rho_0[x_1 = \natural v_1] \ldots [x_m = \natural v_m]$
  therefore $t \in [\![f]\!][\![\Delta]\!]\rho$ by Lemma 19 □

Lemma 33. *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 19

$\Longrightarrow \quad t \in [\![[v/x]f_i]\!][\![\Delta]\!]\rho$ by Lemma 18 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 $s_1'!vs_2 \in [\![h']\!][\![\Delta]\!]\rho$ by *IH* for h $(h4)$

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

    We know that, $t \in s_1'!vs_2 \gg \lambda w.[\![g]\!][\![\Delta]\!]\rho[x = \flat w]$ 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. 19

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

Let $a\,t \in [\![h$ **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 $t \in t'_1 \parallel t_2$ $\quad (i3)$

  We know that, $a\,t'_1 \in [\![h]\!][\![\Delta]\!]\rho$ by $i2$ and Lemma 21

  $\Longrightarrow \quad \Delta, \Gamma \vdash h \xrightarrow{a} h'$ and $t'_1 \in [\![h']\!][\![\Delta]\!]\rho$ by *IH* $\quad (i4)$

  $\overset{\text{ASYM-L}}{\Longrightarrow} \Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h'$ **where** $x :\in g$

  By $i1$, $\exists u \in \mathit{Val}.\ a\,t'_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural u]$

  $\Longrightarrow \quad \Delta, \Gamma[x = u] \vdash h \xrightarrow{a} h'$ and $t'_1 \in [\![h']\!][\![\Delta]\!]\rho[x = \natural u]$ by *IH*

  $\Longrightarrow \quad t'_1 \in \bigcup_{v \in \mathit{Val}} [\![h']\!][\![\Delta]\!]\rho[x = \natural v]$

  $\Longrightarrow \quad t \in [\![h'$ **where** $x :\in g]\!][\![\Delta]\!]\rho$ by $i1, i3$

- ∗ 'a' is an event of t_2, i.e. $t_2 = a\,t'_2$ and $t \in t_1 \parallel t'_2$ $\quad (i5)$

  $\Delta, \Gamma \vdash g \xrightarrow{a} g'$ and $t'_2 \in [\![g']\!][\![\Delta]\!]\rho$ by *IH* for g $\quad (i6)$

  $\overset{\text{ASYM-R}}{\Longrightarrow} \Delta, \Gamma \vdash h$ **where** $x :\in g \xrightarrow{a} h$ **where** $x :\in g'$

  Also, $t \in t_1 <_x t'_2$ by $i2, i5$

  $\Longrightarrow \quad t \in [\![h$ **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}
  $\Longrightarrow a\,t \in t_1 \parallel t_{21}\,\tau$ $\quad (i7)$

  - ∗ '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 21

    $\Longrightarrow \quad t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \flat w]$ by Lemma 23

    $\Longrightarrow \quad t_1 \in [\![[w/x]h]\!][\![\Delta]\!]\rho$ by Lemma 19

- $t_1 = t_{11}[w/x]\,t_{12}, [w/x] \hat{\notin} t_{11},\ t_2 = t_{21}!w\,t_{22}$, no publ. in t_{21}
  $\Longrightarrow \quad a\,t \in (t_{11} \parallel t_{21}\tau)(t_{12} \backslash [w/x])$

  - ∗ 'a' is an event of t_{11},

    i.e. $t_{11} = a\,t'_{11}$ and $t \in (t'_{11} \parallel t_{21}\,\tau)(t_{12} \backslash [w/x])$

    $\Longrightarrow \quad t \in (t'_{11}[w/x]t_{12} <_x t_2)$ $\quad (i8)$

    By Lemma 20, $t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural w]$

    $\Longrightarrow \quad \Delta, \Gamma[x = w] \vdash h \xrightarrow{a} h'$ and

    $t'_{11}[w/x]t_{12} \in [\![h']\!][\![\Delta]\!]\rho[x = \natural w]$ by *IH* $\quad (i9)$

    $\Longrightarrow \quad \Delta, \Gamma \vdash h \xrightarrow{a} h'$ by Lemma 29

    $\overset{\text{ASYM-L}}{\Longrightarrow} \Delta, \Gamma \vdash h$ **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 $t \in (t_{11} \parallel t'_{21}\,\tau)(t_{12} \backslash [w/x])$

    $\Longrightarrow \quad t \in (t_1 <_x t'_{21}!w\,t_{22})$ $\quad (i10)$

    $\Delta, \Gamma \vdash g \xrightarrow{a} g'$ and $t'_{21}!w\,t_{22} \in [\![g']\!][\![\Delta]\!]\rho$ by *IH* $\quad (i11)$

    $\overset{\text{ASYM-R}}{\Longrightarrow} \Delta, \Gamma \vdash h$ **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 $t = t_{11}(t_{21}\backslash[w/x]) = t_1\backslash[w/x]$ $\quad (i12)$
    $\Delta, \Gamma \vdash\ g \xrightarrow{!w} g'$ and $t_{22} \in [\![g']\!][\![\Delta]\!]\rho$ $\quad$ by *IH*
    $\overset{\text{ASYM-P}}{\Longrightarrow}\ \Delta, \Gamma \vdash\ h$ **where** $x :\in g \xrightarrow{\tau} [w/x]h$
    By Lemma 20, $t_1 \in [\![h]\!][\![\Delta]\!]\rho[x = \natural w]$
    $\Longrightarrow\ t_1\backslash[w/x] \in [\![h]\!][\![\Delta]\!]\rho[x = \flat w]$ $\quad$ by Lemma 27
    $\Longrightarrow\ t \in [\![[w/x]h]\!][\![\Delta]\!]\rho$ $\quad$ by $i12$ and Lemma 19
- $a\,t = \varepsilon$
  not applicable, $a\,t$ can not be empty □

Theorem 9 (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
  $a\,t' \in [\![f]\!][\![\Delta]\!]\rho$
  $\Longrightarrow\ a\,t' \in [\![\sigma f]\!][\![\Delta]\!]\rho[x_1 = \natural v_1]\ldots[x_m = \natural v_m]$ $\quad$ by Lemma 19
  $\Longrightarrow\ \Delta, \Gamma \vdash\ \sigma f \xrightarrow{a} f'$ and
  $t' \in [\![f']\!][\![\Delta]\!]\rho[x_1 = \natural v_1]\ldots[x_m = \natural v_m]$ $\quad$ by Lemma 33
  $\Longrightarrow\ \Delta, \Gamma \vdash\ f' \xrightarrow{t'}{}^{*} f''$ $\quad$ by *IH* for t'
  $\Longrightarrow\ \Delta, \Gamma \vdash\ \sigma f \xrightarrow{a} f' \xrightarrow{t'}{}^{*} f''$
  $\Longrightarrow\ \Delta, \Gamma \vdash\ \sigma f \xrightarrow{t}{}^{*} f''$ □