Properties of the Timed Operational and Denotational Semantics of Orc

Ian Wehrman, David Kitchin, William R. Cook, Jayadev Misra
Department of Computer Sciences
The University of Texas at Austin
email: {iwehrman,dkitchin,wcook,misra}@cs.utexas.edu

December 31, 2007

Preface

Orc is a language for structured concurrent programming. Orc provides three powerful combinators that define the structure of a concurrent computation. These combinators support sequential and concurrent execution, and concurrent execution with blocking and termination.

Orc is particularly well-suited for task orchestration, a form of concurrent programming with applications in workflow, business process management, and web service orchestration. Orc provides constructs to orchestrate the concurrent invocation of services while managing time-outs, priorities, and failures of services or communication.

Previous work on the semantics of Orc has focused on its asynchronous behavior. In this report, we define a relative-time operational semantics of Orc that allows reasoning about delays, which are introduced explicitly by time-based constructs or implicitly by network delays. We develop a number of identities among Orc expressions and define an equality relation that is a congruence. We also develop a denotational semantics for Orc, in which the meaning of an Orc expression is a set of traces. A number of properties about the semantics are shown here, including equivalence of the operational and denotational semantics.

Contents

Chapter 1

Introduction

1.1 Introduction

This monograph establishes a number of semantic properties of Orc expressions. An operational semantics of Orc is given elsewhere and abbreviated in Section 1.2. Executions and traces of Orc expressions are defined (see Section 1.2.3) based on this operational semantics.

In Section 2.2, page 36, we define combinators, corresponding to each Orc combinator, that can be applied to sets of executions (and traces). Then we can write $U \mid V$, $U >x> V$ and $U <x< V$, where U and V are sets of executions. Denoting the set of executions of f by $[\![f]\!]$, we establish in Chapter 2 that $[\![f * g]\!] = [\![f]\!] * [\![g]\!]$, where $*$ is any orc combinator, $\mid$, $>x>$ or $<x<$.

The results of Chapter 2 are used in Chapter 3 to establish that the traces of $f * g$ can be determined from the traces of f and g. Denoting the traces of f by $\langle\!\langle f \rangle\!\rangle$, we write $f \cong g$ to mean $\langle\!\langle f \rangle\!\rangle = \langle\!\langle g \rangle\!\rangle$. We show that relation $\cong$ is an equality relation; given $f \cong g$, f and g can replace each other in all contexts. We establish even finer results by defining a partial order over expressions; $f \subseteq g$ means that $\langle\!\langle f \rangle\!\rangle \subseteq \langle\!\langle g \rangle\!\rangle$. Given $f \subseteq g$, we prove that $f * h \subseteq g * h$ and $h * f \subseteq h * g$; i.e., replacing f by g in any context results in at least the same set of traces.

Finally, in Chapter 4 we present a meaning function μ for arbitrary Orc expressions, including those with recursively defined expressions. The meaning $\mu(f)$ of expression f is a set of traces. The final theorem establishes that $\langle\!\langle f \rangle\!\rangle = \mu(f)$; i.e., that the operational and denotational semantics are equivalent.

The main theorems we establish are:

Theorem 1 $\langle\!\langle f * g \rangle\!\rangle = \overline{\langle\!\langle f \rangle\!\rangle * \langle\!\langle g \rangle\!\rangle}$
See Theorem 26, page 114.

Theorem 2 Suppose $U \subseteq U'$ and $V \subseteq V'$. Then, $U * V \subseteq U' * V'$.
See Theorem 9, page 38.

Theorem 3 For any U and V,

1. (Left Distributivity) $(\cup i :: P_i * V) = (\cup i :: P_i) * V$, for a family of sets P_i.
2. (Right Distributivity) $(\cup i :: U * Q_i) = U * (\cup i :: Q_i)$, for a sequence of sets Q_i, where $Q_0 \subseteq Q_1 \subseteq \cdots$.

See Theorem 10, page 41.

Theorem 4 $\langle\!\langle f \rangle\!\rangle = \mu(f)$ See Theorem 27, page 120.

1.2 Timed Operational Semantics

1.2.1 Time-shifted Expressions

A time-shifted expression, written f^t, is the expression that results from f after t units have elapsed *without occurrence of an event.* When it is not possible for t time units to elapse without f engaging in an event we write $f^t = \perp$, where $\perp$ is an unreachable expression described later. The time-shifted expression f^t, for $t \geq 0$, is defined in Figure 1.1 based on the structure of f.

$$
\begin{array}{rcl}
(f \mid g)^t & \underline{\Delta} & f^t \mid g^t \\
(f > x > g)^t & \underline{\Delta} & f^t > x > g \\
(f < x < g)^t & \underline{\Delta} & f^t < x < g^t \\
M(x)^t & \underline{\Delta} & M(x) \\
M(m)^t & \underline{\Delta} & \begin{cases} M(m) & \text{if } t = 0 \\ \perp & \text{otherwise.} \end{cases} \\
?k^t & \underline{\Delta} & ?\{(s, m) \mid (t + s, m) \in k\} \cup \{x \in ?k \mid x = \omega\}
\end{array}
$$

Figure 1.1: Definition of Time-shifted Expressions

The first three cases, for each of the combinators, are easy to justify informally. Expression $M(x)^t$, where x is a variable, is simply $M(x)$ because the site cannot be invoked until the parameter has a value. Expression $M(m)$, where m is a value, must be invoked at time 0; therefore, $M(m)^0 = M(m)$, whereas $M(m)^t = \perp$ for $t > 0$. The time-shifted handle $?k^t$ may publish m at time s iff $?k$ may publish m at $t + s$; and $?k^t$ includes ω iff $?k$ does.

The definitions for $M(x)^t$ and $M(m)^t$ in Figure 1.1 also encompass local sites $\mathit{if}(\mathit{true})^t$, Signal^t, $\mathit{let}(m)^t$, etc. Of particular importance is *Rtimer*. Consider the handle $?k$ that results from a call to *Rtimer*(3). It is easily seen that $?k^2 = ?j$, where $?j$ is a handle resulting from a call to *Rtimer*(1), i.e., *Rtimer*(3) behaves like *Rtimer*(1) after 2 times units have elapsed.

$$\frac{[E(x) \;\underline{\Delta}\; f] \in \mathcal{D}}{E(p) \overset{0,\tau}{\to} [p/x].f} \quad (\textsc{Def})$$

$$\frac{f \overset{t,a}{\to} f' \qquad a \neq\, !m}{f >x> g \overset{t,a}{\to} f' >x> g} \quad (\textsc{Seq1N})$$

$$\frac{k \in \Sigma(M,m)}{M(m) \overset{0,\tau}{\to} ?k} \quad (\textsc{Call})$$

$$\frac{f \overset{t,!m}{\to} f'}{f >x> g \overset{t,\tau}{\to} (f' >x> g) \mid [m/x].g} \quad (\textsc{Seq1V})$$

$$\frac{(t,m) \in k}{?k \overset{t,!m}{\to} \mathbf{0}} \quad (\textsc{Return})$$

$$\frac{f \overset{t,a}{\to} f'}{f <x< g \overset{t,a}{\to} f' <x< g^t} \quad (\textsc{Asym1})$$

$$\frac{f \overset{t,a}{\to} f'}{f \mid g \overset{t,a}{\to} f' \mid g^t} \quad (\textsc{Sym1})$$

$$\frac{g \overset{t,!m}{\to} g'}{f <x< g \overset{t,\tau}{\to} [m/x].f^t} \quad (\textsc{Asym2V})$$

$$\frac{g \overset{t,a}{\to} g'}{f \mid g \overset{t,a}{\to} f^t \mid g'} \quad (\textsc{Sym2})$$

$$\frac{g \overset{t,a}{\to} g' \qquad a \neq\, !m}{f <x< g \overset{t,a}{\to} f^t <x< g'} \quad (\textsc{Asym2N})$$

Figure 1.2: Timed Semantics of Orc

1.2.2 Transition Rules

We present a timed operational semantics of Orc based on a labeled transition system. The labels of the transition system are time-event pairs (t,a). The relation $f \overset{t,a}{\to} f'$, defined in Figure 1.2, states that expression f may transition exactly t time units after its evaluation starts with event a to expression f'.

Events are either *publication* events, written $!m$, or *internal* events, written τ. Publication events correspond to the communication of value m to the environment during a transition. Internal events correspond to state changes not intended to be observable by the environment. We refer to both publication and internal events as *base* events.

The times in the transition relation are *relative* to the start of evaluation of the expression. Furthermore, $f \overset{t,a}{\to} f'$ specifies that no other events have occurred in the t units that have passed since the beginning of the evaluation of f. Times are natural numbers (though we can use any totally-ordered set with a least element, such as the non-negative reals).

Notation Henceforth, expressions are denoted by f, g, h; variables by x, y, z; events by a, b; and times by t, s. Sets of objects are denoted by the uppercase versions of their corresponding letters. We write $[m/x].f$ for the expression obtained from f by replacing every free occurrence of x by value m. Parameters, which are either variables or values, are denoted by p.

1.2.3 Executions and Traces

In this section, we formalize the notions of *executions* and *traces* for expressions. An execution of f is a sequence of timed events in which f may engage. A trace is an execution with the τ events removed. We write $f \stackrel{u}{\Rightarrow} g$, where u is a sequence of timed events of the form (t, a), to denote that f may engage in event a exactly t units after the start of its evaluation, and transition to g immediately after engaging in all the events in u.

Execution relation $\Rightarrow$ is derived from the reflexive and transitive closure of the transition relation $\rightarrow$ of Figure 1.2. However, we need to shift the times in forming the transitive closure. Given $f \stackrel{(s,a)}{\rightarrow} f'$ and $f' \stackrel{(t,b)}{\rightarrow} f''$, we can not claim that $f \stackrel{(s,a)(t,b)}{\Rightarrow} f''$, because b occurs $s + t$ units after the evaluation of f starts. We define u_t as the sequence that results from increasing each time component of u by t. The definition of u_t is also lifted to sets pointwise: $U_t \;\underline{\Delta}\; \{u_t \mid u \in U\}$.

Define relation $\Rightarrow$ as the reflexive-transitive closure of relation $\rightarrow$ except that the time components accumulate.

$$f \stackrel{\epsilon}{\Rightarrow} f \text{ (Ex-Refl)} \qquad \frac{f \stackrel{(t,a)}{\rightarrow} f'',\; f'' \stackrel{u}{\Rightarrow} f'}{f \stackrel{(t,a)u_t}{\Rightarrow} f'} \text{ (Ex-Trans)}$$

Call u an *execution* of f if $f \stackrel{u}{\Rightarrow} f'$ for some $f' \neq \perp$. Note that the empty sequence ϵ is an execution of any expression by rule (Ex-Refl).

The definition of executions requires $f' \neq \perp$ so that all intermediate expressions in an execution (such as f'') are reachable—if any intermediate expression is unreachable, the final expression, f', would be unreachable because $\perp$ has no transitions.

A *trace* $\overline{u}$ is obtained from execution u by removing each internal event (t, τ). The definition is also lifted pointwise to sets: $\overline{U} \;\underline{\Delta}\; \{\overline{u} \mid u \in U\}$.

Notation The *execution set* and *trace set* of f are written $[\![f]\!]$ and $\langle\!\langle f \rangle\!\rangle$ respectively:

$$[\![f]\!] \;\underline{\Delta}\; \{u \mid f \stackrel{u}{\Rightarrow} f', \text{for some } f'\}, \text{ and } \langle\!\langle f \rangle\!\rangle \;\underline{\Delta}\; \overline{[\![f]\!]}.$$

1.2.4 Substitution Events

Substitution Rules

$$
\begin{array}{lcl}
[m/y].(?k) & = & ?k \\
[m/y].(M(p)) & = & \begin{cases} M(m) & \text{if } p = y \\ M(p) & \text{otherwise} \end{cases} \\
[m/y].(E(p)) & = & \begin{cases} E(m) & \text{if } p = y \\ E(p) & \text{otherwise} \end{cases} \\
[m/y].(f \mid g) & = & ([m/y].f) \mid ([m/y].g) \\
[m/y].(f >x> g) & = & \begin{cases} ([m/y].f) >x> g & \text{if } x = y \\ ([m/y].f) >x> ([m/y].g) & \text{otherwise} \end{cases} \\
[m/y].(f <x< g) & = & \begin{cases} f <x< ([m/y].g) & \text{if } x = y \\ ([m/y].f) <x< ([m/y].g) & \text{otherwise} \end{cases}
\end{array}
$$

We have the following rule with *substitution event*:

$$f \overset{t,[m/x]}{\longrightarrow} [m/x].(f^t) \qquad \text{(Subst)}$$

Henceforth, we write $[m/x].f^t$ to mean $[m/x].(f^t)$, i.e., the time-shift operator binds more strongly than substitution.

1.2.5 Summary of Notation

A summary of notation used in the sequel is shown in Figure 1.3.

$f \overset{t,a}{\rightarrow} g$	:	f evaluates in one step to g with event a at time t
$f \overset{u}{\Rightarrow} g$	:	f evaluates in multiple steps to g with execution u
f^t	:	expression f shifted forward in time by t units
u_t, U_t	:	execution or trace u (or set U) delayed by t units
$\overline{u}$, $\overline{U}$	:	trace of an execution u (or set U)
$[\![f]\!]$	:	the set of executions of f
$\langle\!\langle f \rangle\!\rangle$	:	$\overline{[\![f]\!]}$, the set of traces of f
$[m/x].f$	:	replace all free occurrences of x by m in f
$f \cong g$	:	$\langle\!\langle f \rangle\!\rangle = \langle\!\langle g \rangle\!\rangle$

Figure 1.3: Summary of Notation

1.3 Basic Operators on Sequences

A *sequence* is a finite sequence of tuples of the form (t, b), where b is an *event* and t is its associated time, $t \geq 0$. The times of events in a sequence are monotone non-decreasing. An event is either *base* or *substitution* event. There is a special base event, τ. As is customary, the empty sequence is denoted by ϵ.

Notation For event a, $a.time$ is its associated time; for sequence x, $x.time$ is the time of its last event if x is non-empty and 0 if x is empty. Note that

$$\begin{array}{l} \epsilon.time = 0, \\ p_t.time = p.time + t, \text{ for } p \neq \epsilon, \\ (ap).time = a.time, \text{ if } p = \epsilon, \text{ and } p.time, \text{ if } p \neq \epsilon \end{array}$$

We define a few basic operations on sequences (and sets of sequences) in this section.

1.3.1 Definitions of Operators

Henceforth, u, v, p and q denote sequences. And, uppercase equivalents of these symbols denote sets of sequences. Symbols a and b denote events.

The *time-shift* of u by t, where $t \geq 0$, is written as u_t; it is the same sequence as u except that the associated time of each event is increased by t. Formally, time-shift for an individual event is given by $(s, b)_t = (s + t, b)$. And, for a sequence,

$$\begin{array}{l} \epsilon_t = \epsilon \\ (au)_t = a_t u_t \end{array}$$

It is customary to write a non-empty sequence v as au_t, where t is the time associated with event a. Here, u_t is the suffix of v containing all events except a, and u is obtained from this suffix by decreasing associated times by t.

Observation 1 For sequence u, $(u_s)_t = u_{s+t}$.

The *prefix-closure* of u, written as u^*, is the set of prefixes of u. Formally,

$$\begin{array}{l} \epsilon^* = \{\epsilon\}, \\ (au)^* = \{\epsilon\} \cup au^* \end{array}$$

Note that $\{\epsilon\} \subseteq u^*$, for all u. Therefore, $(au)^* = \{\epsilon\} \cup au^*$ holds (vacuously) even when $a = \epsilon$. Set U is *prefix-closed* if $u^* \subseteq U$, for every u in U.

The *trace* of u, $\overline{u}$, is the subsequence obtained from u by dropping all τ events from it. Formally,

$$\begin{array}{l} \overline{\epsilon} = \epsilon, \\ \overline{\tau} = \epsilon, \\ \overline{a} = a \text{ where } a \neq \tau, \\ \overline{au} = \overline{a}\,\overline{u} \end{array}$$

Event Removal from front of a sequence Define $u \backslash a$, where a is any event, as follows:

$$u \backslash a = \begin{cases} \{v\} & \text{if } u = av_t \text{ where } t = a.time \\ \phi & \text{otherwise} \end{cases}$$

1.3.2 Coercion

We coerce time-shift, trace and removal operators, α, to sets of sequences:

$$\alpha(U) = (\cup u : u \in U : \alpha(u))$$

Thus, for example,

$$U \backslash a = (\cup u : u \in U : u \backslash a) = \{v|\ av_t \in U\}, \text{ where } t = a.time$$

We use the convention that if $\alpha(u)$ is a sequence (as in u_t or $\overline{u}$), it is treated as the set $\{\alpha(u)\}$. If $U = \phi$, the empty set, $\alpha(U) = \phi$.

An operator α is *coercive* if it satisfies $\alpha(U) = (\cup u : u \in U : \alpha(u))$. Operators time-shift, trace and removal are, by definition, coercive. A coercive operator distributes over set union.

Prefix-closure is coercive over non-empty sets. For empty set ϕ, $\phi^* = \{\epsilon\}$, not ϕ. Prefix-closure distributes over set union even when some of the sets are empty.

Observation 2 A coercive operator α satisfies the monotonicity condition:

$$U \subseteq V \ \Rightarrow\ \alpha(U) \subseteq \alpha(V)$$

Observation 3 Composition of coercive operators is coercive.

Observation 4 For coercive α and γ, where u ranges over all elements of U,

$$\begin{aligned} &\alpha(u) \subseteq \gamma(u) \ \Rightarrow\ \alpha(U) \subseteq \gamma(U), \\ &\alpha(u) = \gamma(u) \ \Rightarrow\ \alpha(U) = \gamma(U) \end{aligned}$$

Idempotence Operator α is idempotent if $\alpha(\alpha(u)) = \alpha(u)$.

1.3.3 Some Simple Facts

Lemma 1

$$\begin{aligned} &(u_t)^* = (u^*)_t, \\ &\overline{(u_t)} = (\overline{u})_t, \\ &\overline{u^*} = \overline{u}^*. \end{aligned}$$

Proof: Each of these may be proved by induction on the length of u. We give a detailed proof for the last case, $\overline{u^*} = \overline{u}^*$. If $u = \epsilon$, the result follows easily. Next, let $u = av$.

$$\begin{aligned} & \overline{u^*} \\ = \quad & \{u = av\} \\ & \overline{(av)^*} \\ = \quad & \{\text{definition of } ^*\} \\ & \overline{\{\epsilon\} \cup av^*} \\ = \quad & \{\text{distribute trace over union and concatenation}\} \end{aligned}$$

$$
\begin{array}{ll}
 & \{\epsilon\} \cup \overline{a}\,\overline{v^*} \\
= & \{\text{induction}\} \\
 & \{\epsilon\} \cup \overline{a}\,\overline{v}^* \\
= & \{\text{definition of } {}^*\} \\
 & (\overline{a}\,\overline{v})^* \\
= & \{\text{distribute trace over concatenation}\} \\
 & (\overline{av})^* \\
= & \{u = av\} \\
 & (\overline{u})^*
\end{array}
$$

A number of results over sequences can be coerced to sets of sequences using these observations. For example, we can derive $(U_t)^* = (U^*)_t$, as follows. From Lemma 1, page 7, $(u_t)^* = (u^*)_t$. The operator on each side of the identity is coercive, since it is a composition of two coercive operators (see Observation 3, page 7). Applying Observation 4, page 7, the result follows.

Henceforth, we will state results mostly over sequences, and derive the corresponding results over sets using coercion.

Observation 5 Operators prefix-closure and trace are idempotent, i.e.,

$$
\begin{array}{l}
(u^*)^* = u^*, \\
\overline{\overline{u}} = \overline{u}
\end{array}
$$

Note that time-shift is not idempotent. Also note that for coercive and idempotent α, $\alpha(\alpha(U)) = \alpha(U)$, by applying Observation 4, page 7 to the definition of idempotence.

Observation 6 Let $f \overset{a}{\rightarrow} f'$ where a is a substitution at time t. Then, $[\![f']\!] = [\![f]\!]\backslash a$, and $a[\![f']\!]_t \subseteq [\![f]\!]$.

Proof: Given $f \overset{a}{\rightarrow} f'$, for any $u \in [\![f']\!]$, i.e., $f' \overset{u}{\Rightarrow}$, $au_t \in [\![f]\!]$. Therefore, $a[\![f']\!]_t \subseteq [\![f]\!]$. We show $[\![f']\!] = [\![f]\!]\backslash a$ by mutual inclusion.

- $[\![f']\!] \subseteq [\![f]\!]\backslash a$:

$$
\begin{array}{ll}
 & u \in [\![f']\!] \\
\subseteq & \{a[\![f']\!]_t \subseteq [\![f]\!]\} \\
 & au_t \in [\![f]\!] \\
\Rightarrow & \{[\![f]\!]\backslash a = \{v \mid av_t \in [\![f]\!]\}\} \\
 & u \in [\![f]\!]\backslash a
\end{array}
$$

- $[\![f]\!]\backslash a \subseteq [\![f']\!]$:

$$
\begin{array}{ll}
 & u \in [\![f]\!]\backslash a \\
\Rightarrow & \{[\![f]\!]\backslash a = \{v \mid av_t \in [\![f]\!]\}\} \\
 & au_t \in [\![f]\!] \\
\Rightarrow & \{\text{meaning of execution}\}
\end{array}
$$

$$
\begin{array}{ll}
 & f \xrightarrow{a} f'' \stackrel{u}{\Rightarrow} \text{, for some } f'' \\
\Rightarrow & \{\text{since } a \text{ is a substitution and } f \xrightarrow{a} f', \text{ we get } f' = f''\} \\
 & f \xrightarrow{a} f' \stackrel{u}{\Rightarrow} \\
\Rightarrow & \{\text{obviously}\} \\
 & u \in [\![f']\!]
\end{array}
$$

1.4 A Specific Set of Sequences

We define set $A(t)$, for any t, $t \geq 0$, to contain sequences of substitutions, as follows. For $t \geq 0$,

$$A(t) = \{u_r |\ u_r \text{ is a finite sequence of substitutions at time } r,\ 0 \leq r \leq t\}.$$

Similarly, we define set $D(t)$, for any $t \geq 0$ to contain sequences of substitutions as follows:

$$D(t) = \{p \mid p \text{ is a finite sequence of substitutions with nondecreasing time} \leq t\}$$

For sets of times T, $A(T)$ and $D(T)$ are defined coercively. Observe that in any sequence of $A(t)$, all events occur at the same time. Also, $A(t)$ and $D(t)$ contain the empty sequence.

For sets of sequences U and V, their *concatenation* and *partial concatenation*, written UV and $U \cdot V$ respectively, are defined by

$$UV = \{uv \mid u \in U, v \in V\}, \text{ and } U \cdot V = U \cup UV.$$

Partial concatenation is right-associative: $U \cdot V \cdot W = U \cdot (V \cdot W)$.

Observation 7 *1.* $A(t)^* = A(t)$ *and* $D(t)^* = D(t)$*.*

2. $\overline{A(t)} = A(t)$ *and* $\overline{D(t)} = D(t)$*.*

3. $A(s+t) = A(s) \cup A(t)_s$ *and* $D(s+t) = D(s) \cdot D(t)_s$*.*

4. For $s \leq t$*,* $A(s) \subseteq A(t)$ *and* $D(s) \subseteq D(t)$*.*

5. $A(s) \subseteq D(t)$

6. $A(0)\backslash[m/x] = A(0)$ *and* $D(0)\backslash[m/x] = D(0)$

The sets $A(t)$ and $D(t)$ are first used in Sections 3.1.1 and 2.1, respectively.

1.5 Basic Theorems on Executions

We derive two basic theorems on executions in this section.

1.5.1 Evolution

Theorem 5 (Evolution) $f^s \stackrel{t,a}{\rightarrow} h \equiv f \stackrel{s+t,a}{\rightarrow} h$

Proof: First, we dispose of the case where $h = \bot$. In that case, both sides of the equivalence are true (because $f \stackrel{t,a}{\rightarrow} \bot$, for all t, a and f). Henceforth, assume that $h \neq \bot$.

If $s = 0$, the result follows by appealing to the proposition $f^0 = f$. Henceforth, let $s > 0$. If f is $\mathbf{0}$, $M(m)$, *if* or *let*, then f^s is $\bot$, which has transition only to $\bot$. Since $h \neq \bot$, f is not one of the given expressions. If f is $Rtimer(u)$, then $Rtimer(u)^s \stackrel{t,a}{\rightarrow} h$ where $h \neq \bot$ arises only when $u \geq s + t$. Then, the result is easy to see.

We give proofs by structural induction for expressions of the form $(f \mid g)$, $(f > x > g)$ and $(f < x < g)$ in place of f.

$(f \mid g)$: Suppose $(f \mid g)^s \stackrel{t,a}{\rightarrow} h$. From definition, that is $f^s \mid g^s \stackrel{t,a}{\rightarrow} h$. Assume, without loss in generality, that this is deduced by applying (Sym1), i.e.,

$$f^s \stackrel{t,a}{\rightarrow} f', \text{ and } h = f' \mid (g^s)^t$$

Now,

$$\begin{array}{ll} & f^s \stackrel{t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{induction}\} \\ & f \stackrel{s+t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{Apply (Sym1)}\} \\ & f \mid g \stackrel{s+t,a}{\rightarrow} f' \mid g^{s+t} \\ \Rightarrow & \{g^{s+t} = (g^s)^t\} \\ & f \mid g \stackrel{s+t,a}{\rightarrow} f' \mid (g^s)^t \\ \Rightarrow & \{h = f' \mid (g^s)^t\} \\ & f \mid g \stackrel{s+t,a}{\rightarrow} h \end{array}$$

In the other direction, suppose that $f \mid g \stackrel{s+t,a}{\rightarrow} h$. Assume, without loss in generality, that this is deduced by applying (Sym1), i.e.,

$$f \stackrel{s+t,a}{\rightarrow} f', \text{ and } h = f' \mid g^{s+t}$$

Now,

$$\begin{array}{ll} & f \stackrel{s+t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{induction}\} \\ & f^s \stackrel{t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{Apply (Sym1)}\} \\ & f^s \mid g^s \stackrel{t,a}{\rightarrow} f' \mid (g^s)^t \\ \Rightarrow & \{g^{s+t} = (g^s)^t\} \end{array}$$

$$
\begin{array}{ll}
 & f^s \mid g^s \overset{t,a}{\to} f' \mid g^{s+t} \\
\Rightarrow & \{h = f' \mid g^{s+t}\} \\
 & f^s \mid g^s \overset{t,a}{\to} h \\
\Rightarrow & \{\text{definition}\} \\
 & (f \mid g)^s \overset{t,a}{\to} h
\end{array}
$$

$(f \ {>}x{>}\ g)$**:** Suppose $(f \ {>}x{>}\ g)^s \overset{t,a}{\to} h$. From definition, that is $f^s \ {>}x{>}\ g \overset{t,a}{\to} h$. There are two cases, depending on whether or not a is a publication event. First assume $a \neq !v$, and by rule (SEQ1N):

$$f^s \overset{t,a}{\to} f', \text{ and } h = f' \ {>}x{>}\ g.$$

Now,

$$
\begin{array}{ll}
 & f^s \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{induction}\} \\
 & f \overset{s+t,a}{\to} f' \\
\Rightarrow & \{\text{Apply (SEQ1N)}\} \\
 & f \ {>}x{>}\ g \overset{s+t,a}{\to} f' \ {>}x{>}\ g \\
\Rightarrow & \{h = f' \ {>}x{>}\ g\} \\
 & f \ {>}x{>}\ g \overset{s+t,a}{\to} h
\end{array}
$$

Next assume $a = !v$, and by rule (SEQ1V):

$$f^s \overset{t,!m}{\to} f', \text{ and } h = (f' \ {>}x{>}\ g) \mid [m/x].g.$$

Now,

$$
\begin{array}{ll}
 & f^s \overset{t,!m}{\to} f' \\
\Rightarrow & \{\text{induction}\} \\
 & f \overset{s+t,!v}{\to} f' \\
\Rightarrow & \{\text{Apply (SEQ1V)}\} \\
 & f \ {>}x{>}\ g \overset{s+t,!v}{\to} (f' \ {>}x{>}\ g) \mid [m/x].g \\
\Rightarrow & \{h = (f' \ {>}x{>}\ g) \mid [m/x].g\} \\
 & f \ {>}x{>}\ g \overset{s+t,!v}{\to} h
\end{array}
$$

In the other direction, suppose that $f \ {>}x{>}\ g \overset{s+t,a}{\to} h$. Again there are two cases corresponding to the presence of a publication. First assume $a \neq !v$, and by rule (SEQ1N):

$$f \overset{s+t,a}{\to} f', \text{ and } h = f' \ {>}x{>}\ g$$

Now,

$$
\begin{array}{ll}
& f \overset{s+t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{induction}\} \\
& f^s \overset{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Apply } (\textsc{Seq1N})\} \\
& f^s > x > g \overset{t,a}{\rightarrow} f' > x > g \\
\Rightarrow & \{h = f' > x > g\} \\
& f^s > x > g \overset{t,a}{\rightarrow} h \\
\Rightarrow & \{\text{definition}\} \\
& (f > x > g)^s \overset{t,a}{\rightarrow} h
\end{array}
$$

Next assume $a = !v$, and by rule (Seq1V):

$$f \overset{s+t,\,!v}{\rightarrow} f', \text{ and } h = (f' > x > g) \mid [m/x].g$$

Now,

$$
\begin{array}{ll}
& f \overset{s+t,\,!v}{\rightarrow} f' \\
\Rightarrow & \{\text{induction}\} \\
& f^s \overset{t,!m}{\rightarrow} f' \\
\Rightarrow & \{\text{Apply } (\textsc{Seq1V})\} \\
& f^s > x > g \overset{t,!m}{\rightarrow} (f' > x > g) \mid [m/x].g \\
\Rightarrow & \{h = (f' > x > g) \mid [m/x].g\} \\
& f^s > x > g \overset{t,!m}{\rightarrow} h \\
\Rightarrow & \{\text{definition}\} \\
& (f > x > g)^s \overset{t,!m}{\rightarrow} h
\end{array}
$$

$(f < x < g)$**:** Suppose $(f < x < g)^s \overset{t,a}{\rightarrow} h$. From definition, that is $f^s < x < g^s \overset{t,a}{\rightarrow} h$. First assume the transition is due to a transition of f^s by (Asym1N), i.e.,

$$f^s \overset{t,a}{\rightarrow} f', \text{ and } h = f' < x < (g^s)^t.$$

Now,

$$
\begin{array}{ll}
& f^s \overset{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{induction}\} \\
& f \overset{s+t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Apply } (\textsc{Asym1N})\} \\
& f < x < g \overset{s+t,a}{\rightarrow} f' < x < g^{s+t} \\
\Rightarrow & \{g^{s+t} = (g^s)^t\} \\
& f < x < g \overset{s+t,a}{\rightarrow} f' < x < (g^s)^t \\
\Rightarrow & \{h = f' < x < (g^s)^t\} \\
& f < x < g \overset{s+t,a}{\rightarrow} h
\end{array}
$$

Next assume the transition is due to a transition of g^s. There are two cases, depending on whether or not a is a publication event. First assume $a \neq !v$, and by rule (ASYM2):

$$g^s \stackrel{t,a}{\rightarrow} g' \text{ and } h = (f^s)^t <x< g'.$$

Now,

$$\begin{array}{ll}
 & g^s \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{induction}\} \\
 & g \stackrel{s+t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Apply (ASYM2)}\} \\
 & f <x< g \stackrel{s+t,a}{\rightarrow} f^{s+t} <x< g' \\
\Rightarrow & \{f^{s+t} = (f^s)^t\} \\
 & f <x< g \stackrel{s+t,a}{\rightarrow} (f^s)^t <x< g' \\
\Rightarrow & \{h = (f^s)^t <x< g'\} \\
 & f <x< g \stackrel{s+t,a}{\rightarrow} h
\end{array}$$

Finally assume $a = !v$, and by rule (ASYM1V):

$$g^s \stackrel{t,!m}{\rightarrow} g' \text{ and } h = [m/x].(f^s)^t.$$

Now,

$$\begin{array}{ll}
 & g^s \stackrel{t,!m}{\rightarrow} g' \\
\Rightarrow & \{\text{induction}\} \\
 & g \stackrel{s+t,!v}{\rightarrow} g' \\
\Rightarrow & \{\text{Apply (ASYM1V)}\} \\
 & f <x< g \stackrel{s+t,!v}{\rightarrow} [m/x].f^{s+t} \\
\Rightarrow & \{f^{s+t} = (f^s)^t\} \\
 & f <x< g \stackrel{s+t,!v}{\rightarrow} [m/x].(f^s)^t \\
\Rightarrow & \{h = [m/x].(f^s)^t\} \\
 & f <x< g \stackrel{s+t,!v}{\rightarrow} h
\end{array}$$

In the other direction, suppose that $f <x< g \stackrel{s+t,a}{\rightarrow} h$. First assume the transition is due to a transition of f by (ASYM1N), i.e.,

$$f \stackrel{s+t,a}{\rightarrow} f', \text{ and } h = f' <x< g^{s+t}$$

Now,

$$
\begin{array}{ll}
 & f \stackrel{s+t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{induction}\} \\
 & f^s \stackrel{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Apply (}\textsc{Asym1N}\text{)}\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} f' <x< (g^s)^t \\
\Rightarrow & \{(g^s)^t = g^{s+t}\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} f' <x< g^{s+t} \\
\Rightarrow & \{h = f' <x< g^{s+t}\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} h \\
\Rightarrow & \{\text{definition}\} \\
 & (f <x< g)^s \stackrel{t,a}{\rightarrow} h
\end{array}
$$

Next assume the transition is due to g. There are two cases depending on whether or not a is a publication event. First assume $a \neq !v$, and by rule (Asym2):

$$g \stackrel{s+t,a}{\rightarrow} g', \text{ and } h = f^{s+t} <x< g'$$

Now,

$$
\begin{array}{ll}
 & g \stackrel{s+t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{induction}\} \\
 & g^s \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Apply (}\textsc{Asym2}\text{)}\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} (f^s)^t <x< g' \\
\Rightarrow & \{(f^s)^t = f^{s+t}\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} f^{s+t} <x< g' \\
\Rightarrow & \{h = f^{s+t} <x< g'\} \\
 & f^s <x< g^s \stackrel{t,a}{\rightarrow} h \\
\Rightarrow & \{\text{definition}\} \\
 & (f <x< g)^s \stackrel{t,a}{\rightarrow} h
\end{array}
$$

Finally assume $a = !v$, and by rule (Asym1V):

$$g \stackrel{s+t,!v}{\rightarrow} g', \text{ and } h = [m/x].f^{s+t}.$$

Now,

$$
\begin{array}{ll}
 & g \stackrel{s+t,!v}{\rightarrow} g' \\
\Rightarrow & \{\text{induction}\} \\
 & g^s \stackrel{t,!m}{\rightarrow} g' \\
\Rightarrow & \{\text{Apply (}\textsc{Asym1V}\text{)}\}
\end{array}
$$

$$
\begin{array}{ll}
 & f^s <x< g^s \stackrel{t,!m}{\to} [m/x].(f^s)^t \\
\Rightarrow & \{(f^s)^t = f^{s+t}\} \\
 & f^s <x< g^s \stackrel{t,!m}{\to} [m/x].f^{s+t} \\
\Rightarrow & \{h = [m/x].f^{s+t}\} \\
 & f^s <x< g^s \stackrel{t,!m}{\to} h \\
\Rightarrow & \{\text{definition}\} \\
 & (f <x< g)^s \stackrel{t,!m}{\to} h
\end{array}
$$

Theorem 6 (Shift) $f^t \stackrel{u}{\Rightarrow} g$ *iff* $f \stackrel{u_t}{\Rightarrow} g$.

Proof:
If $u = \epsilon$, $u_t = \epsilon$ and $f = f^t = g$. Otherwise, $u = (s,a)u'_s$, and

$$
\begin{array}{ll}
 & f^t \stackrel{s,a}{\to} f' \stackrel{u'}{\Rightarrow} g \\
\equiv & \{\text{Thm. 5 (Evolution) on page 10}\} \\
 & f \stackrel{t+s,a}{\to} f' \stackrel{u'}{\Rightarrow} g \\
\equiv & \{\text{definition of executions}\} \\
 & f \stackrel{(t+s,a)u'_{t+s}}{\Rightarrow} g \\
\equiv & \{\text{Obs. 1 on page 6}\} \\
 & f \stackrel{(t+s,a)(u'_s)_t}{\Rightarrow} g \\
\equiv & \{\text{definition of shifting}\} \\
 & f \stackrel{((s,a)u'_s)_t}{\Rightarrow} g
\end{array}
$$

Observation 8 $u \in [\![f^t]\!] \equiv u_t \in [\![f]\!]$

Proof: Follows from Theorem 6, page 15.

1.5.2 Substitution Independence

The goal of this section is to show that in an execution of an Orc expression, a pair of adjacent events, $(t,a)(t,b)$, can be swapped, given that a is not a substitution and b is a substitution. First, we prove a lemma.

Lemma 2 Suppose $f \stackrel{(0,a)}{\to} f'$, where a is not a substitution. Then, $[m/x].f \stackrel{(0,a)}{\to} [m/x].(f')$.

Proof: Proof is by induction on the structure of f.

- $\mathbf{0}$: The expression $\mathbf{0}$ only transitions as a result of rule (Subst).

- $?k$: By the operational semantics, the only transition of $?k$ is by rule (Return), where $?k \stackrel{t,!m}{\to} \mathbf{0}$. The result follows because $[m/x].?k = ?k$ and $[m/x].\mathbf{0} = \mathbf{0}$.

• $M(m)$: By the operational semantics, the only transition of $M(m)$ is by rule (Call), where $M(m) \xrightarrow{0,\tau} ?k$. The result follows because $[m/x].M(m) = M(m)$ and $[m/x].?k = ?k$.

• $M(x)$: The expression $M(x)$ only transitions as a result of rule (Subst).

• $f \mid g$: Without loss in generality, suppose that $f \xrightarrow{(0,a)} f'$, so that $f \mid g \xrightarrow{(0,a)} f' \mid g$. We show $[m/x].(f \mid g) \xrightarrow{(0,a)} [m/x].(f' \mid g)$.

$$
\begin{array}{ll}
 & [m/x].(f \mid g) \\
= & \{\text{from substitution rules}\} \\
 & [m/x].f \mid [m/x].g \\
\xrightarrow{(0,a)} & \{\text{from } f \xrightarrow{(0,a)} f'\text{, inductively, } [m/x].f \xrightarrow{(0,a)} [m/x].f'; \\
 & \quad \text{apply rule (SYM1) from operational semantics}\} \\
 & [m/x].f' \mid ([m/x].g)^0 \\
= & \{\text{simplify the last term}\} \\
 & [m/x].f' \mid [m/x].g \\
= & \{\text{from substitution rules}\} \\
 & [m/x].(f' \mid g)
\end{array}
$$

• $f >x> g$: We have two proofs for the two rules (SEQ1N) and (SEQ1V).

Case 1) Suppose $f \xrightarrow{(0,a)} f'$ and (SEQ1N) was applied in deducing $f >x> g \xrightarrow{(0,a)} f' >x> g$.

First we consider the case where substitution is made to the bound variable x. We show $[m/x].(f >x> g) \xrightarrow{(0,a)} [m/x].(f' >x> g)$.

$$
\begin{array}{ll}
 & [m/x].(f >x> g) \\
= & \{\text{from substitution rules}\} \\
 & ([m/x].f) >x> g \\
\xrightarrow{(0,a)} & \{\text{from } f \xrightarrow{(0,a)} f'\text{, inductively, } [m/x].f \xrightarrow{(0,a)} [m/x].f'; \\
 & \quad \text{apply rule (SEQ1N) from operational semantics}\} \\
 & ([m/x].f') >x> g \\
= & \{\text{from substitution rules}\} \\
 & [m/x].(f' >x> g)
\end{array}
$$

Next, consider the case where substitution is made to variable y, $y \neq x$. We show $[m/y].(f >x> g) \xrightarrow{(0,a)} [m/y].(f' >x> g)$.

$$
\begin{array}{ll}
 & [m/y].(f >x> g) \\
= & \{\text{from substitution rules}\} \\
 & ([m/y].f) >x> ([m/y].g) \\
\xrightarrow{(0,a)} & \{\text{from } f \xrightarrow{(0,a)} f'\text{, inductively, } [m/y].f \xrightarrow{(0,a)} [m/y].f'; \\
 & \quad \text{apply rule (SEQ1N) from operational semantics}\}
\end{array}
$$

$$
\begin{array}{ll}
 & ([m/y].f') >x> ([m/y].g) \\
= & \{\text{from substitution rules}\} \\
 & [m/y].(f' >x> g)
\end{array}
$$

Case 2) Suppose $f \stackrel{(0,!n)}{\rightarrow} f'$ and (SEQ1V) was applied in deducing $f >x> g \stackrel{(0,\tau)}{\rightarrow} f' >x> g \mid [n/x].g$.

First we consider the case where substitution is made to the bound variable x. We show $[m/x].(f >x> g) \stackrel{(0,\tau)}{\rightarrow} [m/x].(f' >x> g \mid [n/x].g)$.

$$
\begin{array}{ll}
 & [m/x].(f >x> g) \\
= & \{\text{from substitution rules}\} \\
 & ([m/x].f) >x> g \\
\stackrel{(0,\tau)}{\rightarrow} & \{\text{from } f \stackrel{(0,!n)}{\rightarrow} f'\text{, inductively, } [m/x].f \stackrel{(0,!n)}{\rightarrow} [m/x].f'; \\
 & \quad \text{apply rule (SEQ1V) from operational semantics}\} \\
 & ([m/x].f') >x> g \mid [n/x].g \\
= & \{[n/x].g \text{ has no free occurrence of } x\text{; so } [m/x].([n/x].g) = [n/x].g\} \\
 & ([m/x].f') >x> g \mid [m/x].([n/x].g) \\
= & \{\text{from substitution rules, } ([m/x].f') >x> g = [m/x].(f' >x> g)\} \\
 & ([m/x].(f' >x> g) \mid [m/x].([n/x].g) \\
= & \{\text{from substitution rules}\} \\
 & [m/x].(f' >x> g \mid [n/x].g)
\end{array}
$$

Next, consider the case where substitution is made to variable y, $y \neq x$. We show $[m/y].(f >x> g) \stackrel{(0,\tau)}{\rightarrow} [m/y].(f' >x> g \mid [n/x].g)$.

$$
\begin{array}{ll}
 & [m/y].(f >x> g) \\
= & \{\text{from substitution rules}\} \\
 & ([m/y].f) >x> [m/y].g \\
\stackrel{(0,\tau)}{\rightarrow} & \{\text{from } f \stackrel{(0,!n)}{\rightarrow} f'\text{, inductively, } [m/y].f \stackrel{(0,!n)}{\rightarrow} [m/y].f'; \\
 & \quad \text{apply rule (SEQ1V) from operational semantics}\} \\
 & ([m/y].f') >x> [m/y].g \mid [n/x].([m/y].g) \\
= & \{\text{from substitution rules, } ([m/y].f') >x> [m/y].g = [m/y].(f' >x> g)\} \\
 & [m/y].(f' >x> g) \mid [n/x].([m/y].g) \\
= & \{x \neq y\text{; so } [n/x].([m/y].g) = [m/y].([n/x].g)\} \\
 & [m/y].(f' >x> g) \mid [m/y].([n/x].g) \\
= & \{\text{from substitution rules}\} \\
 & [m/y].(f' >x> g \mid [n/x].g)
\end{array}
$$

• $f <x< g$: We have three proofs for the three rules (ASYM1), (ASYM2N), (ASYM2V).

Case 1) Suppose $f \stackrel{(0,a)}{\rightarrow} f'$ and (ASYM1) was applied in deducing $f <x< g \stackrel{(0,a)}{\rightarrow} f' <x< g$.

First, we consider the case where substitution is made to the bound variable x. We show $[m/x].(f <x< g) \stackrel{(0,a)}{\rightarrow} [m/x].(f' <x< g)$.

$$
\begin{array}{ll}
 & [m/x].(f <x< g) \\
= & \{\text{from substitution rules}\} \\
 & f <x< [m/x].g \\
\stackrel{(0,a)}{\rightarrow} & \{\text{given } f \stackrel{(0,a)}{\rightarrow} f'; \\
 & \quad \text{apply rule (ASYM1) from operational semantics}\} \\
 & f' <x< [m/x].g \\
= & \{\text{from substitution rules}\} \\
 & [m/x].(f' <x< g)
\end{array}
$$

Next, consider the case where substitution is made to variable y, $y \neq x$. We show $[m/y].(f <x< g) \stackrel{(0,a)}{\rightarrow} [m/y].(f' <x< g)$.

$$
\begin{array}{ll}
 & [m/y].(f <x< g) \\
= & \{\text{from substitution rules}\} \\
 & [m/y].f <x< [m/y].g \\
\stackrel{(0,a)}{\rightarrow} & \{\text{using induction on } f \stackrel{(0,a)}{\rightarrow} f', [m/y].f \stackrel{(0,a)}{\rightarrow} [m/y].f'; \\
 & \quad \text{apply rule (ASYM1) from operational semantics}\} \\
 & [m/y].f' <x< [m/y].g \\
= & \{\text{from substitution rules}\} \\
 & [m/y].(f' <x< g)
\end{array}
$$

Case 2) Suppose $g \stackrel{(0,a)}{\rightarrow} g'$, a is not a publication, and (ASYM2N) was applied in deducing $f <x< g \stackrel{(0,a)}{\rightarrow} f <x< g'$.

First, we consider the case where substitution is made to the bound variable x. We show $[m/x].(f <x< g) \stackrel{(0,a)}{\rightarrow} [m/x].(f <x< g')$.

$$
\begin{array}{ll}
 & [m/x].(f <x< g) \\
= & \{\text{from substitution rules}\} \\
 & f <x< [m/x].g \\
\stackrel{(0,a)}{\rightarrow} & \{\text{using induction on } g \stackrel{(0,a)}{\rightarrow} g', [m/x].g \stackrel{(0,a)}{\rightarrow} [m/x].g'; \\
 & \quad \text{apply rule (ASYM2N) from operational semantics}\} \\
 & f <x< [m/x].g' \\
= & \{\text{from substitution rules}\} \\
 & [m/x].(f <x< g')
\end{array}
$$

Next, we consider the case where substitution is made to variable y, $y \neq x$. We show $[m/y].(f <x< g) \stackrel{(0,a)}{\rightarrow} [m/y].(f <x< g')$.

$$
\begin{array}{ll}
 & [m/y].(f <x< g) \\
= & \{\text{from substitution rules}\} \\
 & [m/y].f <x< [m/y].g \\
\stackrel{(0,a)}{\rightarrow} & \{\text{using induction on } g \stackrel{(0,a)}{\rightarrow} g', [m/y].g \stackrel{(0,a)}{\rightarrow} [m/y].g'; \\
 & \quad \text{apply rule (ASYM2N) from operational semantics}\}
\end{array}
$$

$$\begin{array}{cl} & [m/y].f <\!x\!< [m/y].g' \\ = & \{\text{from substitution rules}\} \\ & [m/y].(f <\!x\!< g') \end{array}$$

Case 3) Suppose $g \overset{(0,!n)}{\rightarrow} g'$ and (ASYM2V) was applied in deducing $f <\!x\!< g \overset{(0,\tau)}{\rightarrow} [n/x].f$.

First, we consider the case where substitution is made to the bound variable x. We show $[m/x].(f <\!x\!< g) \overset{(0,\tau)}{\rightarrow} [m/x].([n/x].f)$.

$$\begin{array}{cl} & [m/x].(f <\!x\!< g) \\ = & \{\text{from substitution rules}\} \\ & f <\!x\!< [m/x].g \\ \overset{(0,\tau)}{\rightarrow} & \{\text{using induction on } g \overset{(0,!n)}{\rightarrow} g',\ [m/x].g \overset{(0,!n)}{\rightarrow} [m/x].g'; \\ & \quad \text{apply rule (ASYM2V) from operational semantics}\} \\ & [n/x].(f^0) \\ = & \{\text{simplify}\} \\ & [n/x].f \\ = & \{[n/x].f \text{ has no free occurrence of } x\} \\ & [m/x].([n/x].f) \end{array}$$

Next, we consider the case where substitution is made to variable y, $y \neq x$. We show $[m/y].(f <\!x\!< g) \overset{(0,\tau)}{\rightarrow} [m/y].([n/x].f)$.

$$\begin{array}{cl} & [m/y].(f <\!x\!< g) \\ = & \{\text{from substitution rules}\} \\ & [m/y].f <\!x\!< [m/y].g \\ \overset{(0,\tau)}{\rightarrow} & \{\text{using induction on } g \overset{(0,!n)}{\rightarrow} g',\ [m/y].g \overset{(0,!n)}{\rightarrow} [m/y].g'; \\ & \quad \text{apply rule (ASYM2V) from operational semantics}\} \\ & [n/x].(([m/y].f)^0) \\ = & \{\text{simplify}\} \\ & [n/x].([m/y].f) \\ = & \{[n/x] \text{ and } [m/y] \text{ are substitutions to different variables}\} \\ & [m/y].([n/x].f) \end{array}$$

□

Theorem 7 Let $p(t,a)(t,b)q$ be an execution of expression g, where a is not a substitution and b is a substitution $[m/x]$. Then, $p(t,b)(t,a)q$ is also an execution of g.

Proof: Let $g \overset{p}{\Rightarrow} f \overset{(t,a)}{\rightarrow} f' \overset{(0,b)}{\rightarrow} f''$, where a is not a substitution and b is. We show that $g \overset{p}{\Rightarrow} f \overset{(t,b)}{\rightarrow} \overset{(0,a)}{\rightarrow} f''$. It is is sufficient to show for any expression f that $f \overset{(t,a)}{\rightarrow} f' \overset{(0,b)}{\rightarrow} f''$ implies $f \overset{(t,b)}{\rightarrow} \overset{(0,a)}{\rightarrow} f''$.

$$\begin{array}{cl} & f \overset{(t,a)}{\rightarrow} f' \overset{(0,b)}{\rightarrow} f'' \\ \Rightarrow & \{\text{from the Evolution theorem, Theorem 5, page 10, } f \overset{(t,a)}{\rightarrow} f' \text{ implies } f^t \overset{(0,a)}{\rightarrow} f'\} \end{array}$$

$$
\begin{array}{ll}
 & f^t \overset{(0,a)}{\rightarrow} f' \overset{(0,b)}{\rightarrow} f'' \\
\Rightarrow & \{\text{from Lemma 2, page 15, } f^t \overset{(0,a)}{\rightarrow} f' \text{ implies } [m/x].f^t \overset{(0,a)}{\rightarrow} [m/x].(f')\} \\
 & [m/x].f^t \overset{(0,a)}{\rightarrow} [m/x].(f') \text{ and } f' \overset{(0,b)}{\rightarrow} f'' \\
\Rightarrow & \{b = [m/x]. \text{ So, from } f' \overset{(0,b)}{\rightarrow} f'',\ f'' = [m/x].(f')\} \\
 & [m/x].f^t \overset{(0,a)}{\rightarrow} f'' \\
\Rightarrow & \{\text{Given } b = [m/x],\ f \overset{(t,b)}{\rightarrow} [m/x].f^t\} \\
 & f \overset{(t,b)}{\rightarrow} [m/x].f^t \overset{(0,a)}{\rightarrow} f''
\end{array}
$$

We can prove this theorem under a weaker restriction: a and b are not substitutions to the same variable. In that case, two substitutions, being applied at the same time to different variables, may be executed in either order. We don't, however, need this generality for our subsequent proofs.

Substitution Independent Set Set U is substitution independent if

$p(t,a)(t,b)q \in U$ implies $p(t,b)(t,a)q \in U$,
whenever a is not a substitution and b is a substitution.

Observation 9 1. For any expression f, $[\![f]\!]$ is substitution independent.

2. Let U be substitution independent, and $p(0,b)q \in U$, where no event in p is a substitution and b is a substitution. Then, $(0,b)pq \in U$.

3. Union of substitution independent sets is substitution independent.

Proof: Part (1) follows from Theorem 7, page 19. Part (2) follows by applying induction on the length of p. For $p = \epsilon$, the result is immediate. Given $p = r(0,a)$, from the definition of substitution independent set, $r(0,b)(0,a)q \in U$, and inductively, $(0,b)r(0,a)q = (0,b)pq \in U$. Part (3) follows from the definition of substitution independent set.

Lemma 3 Let U be a substitution independent set and a be a substitution at time 0. Then, $\overline{U\backslash a} = \overline{U}\backslash a$.

Proof: We show $\overline{U\backslash a} \subseteq \overline{U}\backslash a$ and $\overline{U}\backslash a \subseteq \overline{U\backslash a}$.

- $\overline{U\backslash a} \subseteq \overline{U}\backslash a$: We show that for any p, where $p \in U\backslash a$, $\overline{p} \in \overline{U}\backslash a$.

$$
\begin{array}{ll}
 & p \in U\backslash a \\
\Rightarrow & \{\text{definition of } \backslash a\text{; note that } a.time = 0\} \\
 & ap \in U \\
\Rightarrow & \{\text{definition of trace; } a \text{ is a substitution, so } \overline{a} = a\} \\
 & a\overline{p} \in \overline{U} \\
\Rightarrow & \{\text{definition of } \backslash a\text{; note that } a.time = 0\} \\
 & \overline{p} \in \overline{U}\backslash a
\end{array}
$$

- $\overline{U}\backslash a \subseteq \overline{U\backslash a}$:

$$
\begin{array}{ll}
& p \in \overline{U}\backslash a \\
\Rightarrow & \{\text{definition of } \backslash a\text{; note that } a.time = 0\} \\
& ap \in \overline{U} \\
\Rightarrow & \{\text{definition of trace}\} \\
& (\exists u, q : u \text{ is a sequence of } \tau \text{ at time } 0 \ \wedge \overline{q} = p : \ uaq \in U) \\
\Rightarrow & \{\text{from Observation 9, page 20, part(2), } auq \in U\} \\
& (\exists u, q : u \text{ is a sequence of } \tau \text{ at time } 0 \ \wedge \overline{q} = p : \ auq \in U) \\
\Rightarrow & \{\text{definition of } U\backslash a\text{; note that } a.time = 0\} \\
& (\exists u, q : u \text{ is a sequence of } \tau \wedge \overline{q} = p : \ uq \in U\backslash a) \\
\Rightarrow & \{uq \in U\backslash a \text{ implies } \overline{uq} \in \overline{U\backslash a}\} \\
& (\exists u, q : u \text{ is a sequence of } \tau \wedge \overline{q} = p : \ \overline{uq} \in \overline{U\backslash a}) \\
\Rightarrow & \{\overline{u} = \epsilon, \overline{q} = p\} \\
& p \in \overline{U\backslash a}
\end{array}
$$

Corollary 1 For any substitution a at time 0, $\langle\!\langle a.f \rangle\!\rangle = \langle\!\langle f \rangle\!\rangle \backslash a$.

Proof:

$$
\begin{array}{ll}
& \langle\!\langle f \rangle\!\rangle \backslash a \\
= & \{\text{definition of trace}\} \\
& \overline{[\![f]\!]}\backslash a \\
= & \{\text{from Observation 9, page 20, } [\![f]\!] \text{ is substitution independent;} \\
& \quad \text{applying Lemma 3, page 20}\} \\
& \overline{[\![f]\!]\backslash a} \\
= & \{\text{from Observation 6, page 8, } [\![f]\!]\backslash a = [\![a.f]\!]\} \\
& \overline{[\![a.f]\!]} \\
= & \{\text{definition of trace}\} \\
& \langle\!\langle a.f \rangle\!\rangle
\end{array}
$$

1.6 Identities

1.6.1 Strong Bisimilarity

In this section, we list certain identities over arbitrary expressions (i.e., with or without free variables), some of them similar to the laws of Kleene algebra. Proofs of the identities, using strong bisimulation, are given below. Other identities such as $f > x > let(x) \cong f$, can also be proved using weak bisimulation.

Below, "f is x-free" means that x is not a free variable of f.

Lemma 4

1. $f \mid \mathbf{0} \sim f$
2. $f \mid g \sim g \mid f$
3. $f \mid (g \mid h) \sim (f \mid g) \mid h$

4. $f \ {>}x{>}\ (g \ {>}y{>}\ h) \sim (f \ {>}x{>}\ g) \ {>}y{>}\ h$, *if h is x-free.*
5. $\mathbf{0} \ {>}x{>}\ f \sim \mathbf{0}$
6. $(f \mid g) \ {>}x{>}\ h \sim f \ {>}x{>}\ h \mid g \ {>}x{>}\ h$
7. $(f \mid g) \ {<}x{<}\ h \sim (f \ {<}x{<}\ h) \mid g$, *if g is x-free.*
8. $(f \ {>}y{>}\ g) \ {<}x{<}\ h \sim (f \ {<}x{<}\ h) \ {>}y{>}\ g$, *if g is x-free.*
9. $(f \ {<}x{<}\ g) \ {<}y{<}\ h \sim (f \ {<}y{<}\ h) \ {<}x{<}\ g$,
   if g is y-free and h is x-free.
10. $\mathbf{0} \ {<}x{<}\ b \sim b \gg \mathbf{0}$, *where b is a site call or handle.*

Proof:

1. $f \mid \mathbf{0} \sim f$.

   The only subexpression is f. Subexpression $\mathbf{0}$ has no transition.

$$
\begin{array}{ll}
 & f \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{Sym1}\} \\
 & f \mid \mathbf{0} \overset{t,a}{\to} f' \mid \mathbf{0}^t \\
\Rightarrow & \{\text{definition of } \mathbf{0}^t\} \\
 & f \mid \mathbf{0} \overset{t,a}{\to} f' \mid \mathbf{0}
\end{array}
$$

   And,

$$
f \overset{t,a}{\to} f'
$$

   Assumed: $f' \mid \mathbf{0} \sim f'$.

2. $f \mid g \sim g \mid f$.

   First, we consider the transitions of f.

$$
\begin{array}{ll}
 & f \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{Sym1}\} \\
 & f \mid g \overset{t,a}{\to} f' \mid g^t
\end{array}
$$

$$
\begin{array}{ll}
 & f \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{Sym2}\} \\
 & g \mid f \overset{t,a}{\to} g^t \mid f'
\end{array}
$$

   Assumed: $f' \mid g^t \sim g^t \mid f'$

   The derivation with g's transition is symmetric.

3. $f \mid (g \mid h) \sim (f \mid g) \mid h$. We consider the transitions of f, g and h in turn.

(a) (Transition of f: $f \overset{t,a}{\to} f'$)

$$
\begin{array}{ll}
 & f \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{Sym1}\} \\
 & f \mid (g \mid h) \overset{t,a}{\to} f' \mid (g \mid h)^t \\
\Rightarrow & \{\text{definition of } (g \mid h)^t\} \\
 & f \mid (g \mid h) \overset{t,a}{\to} f' \mid (g^t \mid h^t)
\end{array}
$$

And,

$$
\begin{array}{ll}
 & f \overset{t,a}{\to} f' \\
\Rightarrow & \{\text{Sym1}\} \\
 & f \mid g \overset{t,a}{\to} f' \mid g^t \\
\Rightarrow & \{\text{Sym1}\} \\
 & (f \mid g) \mid h \overset{t,a}{\to} (f' \mid g^t) \mid h^t
\end{array}
$$

Assumed: $f' \mid (g^t \mid h^t) \sim (f' \mid g^t) \mid h^t$

(b) (Transition of g: $g \overset{t,a}{\to} g'$)

$$
\begin{array}{ll}
 & g \overset{t,a}{\to} g' \\
\Rightarrow & \{\text{Sym1}\} \\
 & g \mid h \overset{t,a}{\to} g' \mid h^t \\
\Rightarrow & \{\text{Sym2}\} \\
 & f \mid (g \mid h) \overset{t,a}{\to} f^t \mid (g' \mid h^t)
\end{array}
$$

And,

$$
\begin{array}{ll}
 & g \overset{t,a}{\to} g' \\
\Rightarrow & \{\text{Sym2}\} \\
 & f \mid g \overset{t,a}{\to} f^t \mid g' \\
\Rightarrow & \{\text{Sym1}\} \\
 & (f \mid g) \mid h \overset{t,a}{\to} (f^t \mid g') \mid h^t
\end{array}
$$

Assumed: $f^t \mid (g' \mid h^t) \sim (f^t \mid g') \mid h^t$

(c) (Transition of h: $h \overset{t,a}{\to} h'$)

$$
\begin{array}{ll}
 & h \overset{t,a}{\to} h' \\
\Rightarrow & \{\text{Sym2}\} \\
 & g \mid h \overset{t,a}{\to} g^t \mid h' \\
\Rightarrow & \{\text{Sym2}\} \\
 & f \mid (g \mid h) \overset{t,a}{\to} f^t \mid (g^t \mid h')
\end{array}
$$

And,

$$
\begin{array}{ll}
 & h \overset{t,a}{\to} h' \\
\Rightarrow & \{\text{Sym2}\} \\
 & (f \mid g) \mid h \overset{t,a}{\to} (f \mid g)^t \mid h' \\
\Rightarrow & \{\text{definition of } (f \mid g)^t\} \\
 & (f \mid g) \mid h \overset{t,a}{\to} (f^t \mid g^t) \mid h'
\end{array}
$$

Assumed: $f^t \mid (g^t \mid h') \sim (f^t \mid g^t) \mid h'$

4. $f >x> (g >y> h) \sim (f >x> g) >y> h$, provided h is x-free.

   Only the transitions of f have corresponding transitions in $f >x> (g >y> h)$. And, only the transitions of f have corresponding transitions in $f >x> g$, and hence in $(f >x> g) >y> h$. Therefore, we consider only the transitions of f, publications and non-publications.

   (a) $(f \stackrel{t,!m}{\rightarrow} f')$

   $$\begin{array}{ll} & f \stackrel{t,!m}{\rightarrow} f' \\ \Rightarrow & \{\text{Seq1V}\} \\ & f >x> (g >y> h) \stackrel{t,\tau}{\rightarrow} f' >x> (g >y> h) \mid [m/x].(g >y> h) \end{array}$$

   And,

   $$\begin{array}{ll} & f \stackrel{t,!m}{\rightarrow} f' \\ \Rightarrow & \{\text{Seq1V}\} \\ & f >x> g \stackrel{t,\tau}{\rightarrow} f' >x> g \mid ([m/x].g) \\ \Rightarrow & \{\text{Seq1N}\} \\ & (f >x> g) >y> h \stackrel{t,\tau}{\rightarrow} (f' >x> g \mid [m/x].g) >y> h \end{array}$$

   We show $f' >x> (g >y> h) \mid [m/x].(g >y> h) \sim (f' >x> g \mid [m/x].g) >y> h$

   $$\begin{array}{ll} & f' >x> (g >y> h) \mid [m/x].(g >y> h) \\ = & \{\text{substitution distributes}\} \\ & f' >x> (g >y> h) \mid ([m/x].g) >y> ([m/x].h) \\ = & \{h \text{ is } x\text{-free}\} \\ & f' >x> (g >y> h) \mid ([m/x].g) >y> h \end{array}$$

   And,

   $$\begin{array}{ll} & (f' >x> g \mid [m/x].g) >y> h \\ \sim & \{\text{distributivity law}\} \\ & (f' >x> g) >y> h \mid [m/x].g >y> h \\ \sim & \{\text{Associativity}\} \\ & f' >x> (g >y> h) \mid ([m/x].g) >y> h \end{array}$$

   (b) $(f \stackrel{t,a}{\rightarrow} f',\ a \neq !m)$

   $$\begin{array}{ll} & f \stackrel{t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{Seq1N}\} \\ & f >x> (g >y> h) \stackrel{t,a}{\rightarrow} f' >x> (g >y> h) \end{array}$$

   And,

   $$\begin{array}{ll} & f \stackrel{t,a}{\rightarrow} f' \\ \Rightarrow & \{\text{Seq1N}\} \\ & f >x> g \stackrel{t,a}{\rightarrow} f' >x> g \\ \Rightarrow & \{\text{Seq1N}\} \\ & (f >x> g) >y> h \stackrel{t,a}{\rightarrow} (f' >x> g) >y> h \end{array}$$

Assumed: $f' >x> (g >y> h) \sim (f' >x> g) >y> h$, given h is x-free.

Corollary: $f \gg (g >y> h) \sim (f \gg g) >y> h$

5. $\mathbf{0} \gg f \sim \mathbf{0}$, $\mathbf{0} >x> f \sim \mathbf{0}$.

   Only transitions of $\mathbf{0} \gg f$ and $\mathbf{0}$ correspond to those of $\mathbf{0}$, and $\mathbf{0}$ has no transition.

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

   We consider only the transitions of f and g because transitions of h do not have corresponding transitions for either expression. By symmetry and the commutativity of $\mid$ we need consider only the transitions of f.

   (a) $(f \overset{t,a}{\to} f',\ a \neq !m)$

   $$\begin{array}{ll} & f \overset{t,a}{\to} f' \\ \Rightarrow & \{\text{Sym1}\} \\ & f \mid g \overset{t,a}{\to} f' \mid g^t \\ \Rightarrow & \{\text{Seq1N}\} \\ & (f \mid g) >x> h \overset{t,a}{\to} (f' \mid g^t) >x> h \end{array}$$

   And,

   $$\begin{array}{ll} & f \overset{t,a}{\to} f' \\ \Rightarrow & \{\text{Seq1N}\} \\ & f >x> h \overset{t,a}{\to} f' >x> h \\ \Rightarrow & \{\text{Sym1}\} \\ & f >x> h \mid g >x> h \overset{t,a}{\to} f' >x> h \mid (g >x> h)^t \\ \Rightarrow & \{\text{definition of } (g >x> h)^t\} \\ & f >x> h \mid g >x> h \overset{t,a}{\to} f' >x> h \mid g^t >x> h \end{array}$$

   Assumed: $(f' \mid g^t) >x> h \sim f' >x> h \mid g^t >x> h$.

   (b) $(f \overset{t,!m}{\to} f')$

   $$\begin{array}{ll} & f \overset{t,!m}{\to} f' \\ \Rightarrow & \{\text{Seq1V}\} \\ & f >x> h \overset{t,\tau}{\to} f' >x> h \mid [m/x].h \\ \Rightarrow & \{\text{Sym1}\} \\ & f >x> h \mid g >x> h \overset{t,\tau}{\to} (f' >x> h \mid [m/x].h) \mid (g >x> h)^t \\ \Rightarrow & \{\text{definition of } (g >x> h)^t\} \\ & f >x> h \mid g >x> h \overset{t,\tau}{\to} (f' >x> h \mid [m/x].h) \mid (g^t >x> h) \end{array}$$

   And,

   $$\begin{array}{ll} & f \overset{t,!m}{\to} f' \\ \Rightarrow & \{\text{Sym1}\} \end{array}$$

$$
\begin{array}{ll}
 & f \mid g \stackrel{t,!m}{\rightarrow} f' \mid g^t \\
\Rightarrow & \{\text{Seq1V}\} \\
 & (f \mid g) >x> h \stackrel{t,\tau}{\rightarrow} (f' \mid g^t) >x> h \mid [m/x].h
\end{array}
$$

To see $(f' >x> h \mid [m/x].h) \mid g^t >x> h \sim (f' \mid g^t) >x> h \mid [m/x].h$

$$
\begin{array}{ll}
 & (f' \mid g^t) >x> h \mid [m/x].h \\
\sim & \{\text{distributivity}\} \\
 & (f' >x> h \mid g^t >x> h) \mid [m/x].h \\
\sim & \{\text{associativity and commutativity of } \mid \} \\
 & (f' >x> h \mid [m/x].h) \mid g^t >x> h
\end{array}
$$

7. $(f \mid g) <x< h \sim (f <x< h) \mid g$, provided g is x-free.

   There are four different kinds of transitions for each of the expressions: transitions of f, g, publication of h and non-publication of h.

   (a) $f \stackrel{t,a}{\rightarrow} f'$:

$$
\begin{array}{ll}
 & f \stackrel{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Sym1}\} \\
 & f \mid g \stackrel{t,a}{\rightarrow} f' \mid g^t \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f \mid g) <x< h \stackrel{t,a}{\rightarrow} (f' \mid g^t) <x< h^t
\end{array}
$$

   And,

$$
\begin{array}{ll}
 & f \stackrel{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Asym2}\} \\
 & f <x< h \stackrel{t,a}{\rightarrow} f' <x< h^t \\
\Rightarrow & \{\text{Sym1}\} \\
 & (f <x< h) \mid g \stackrel{t,a}{\rightarrow} (f' <x< h^t) \mid g^t
\end{array}
$$

   Assumed: $(f' \mid g^t) <x< h^t \sim (f' <x< h^t) \mid g^t$

   (b) $g \stackrel{t,a}{\rightarrow} g'$:

$$
\begin{array}{ll}
 & g \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Sym2}\} \\
 & f \mid g \stackrel{t,a}{\rightarrow} f^t \mid g' \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f \mid g) <x< h \stackrel{t,a}{\rightarrow} (f^t \mid g') <x< h^t
\end{array}
$$

   And,

$$
\begin{array}{ll}
 & g \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Sym2}\} \\
 & (f <x< h) \mid g \stackrel{t,a}{\rightarrow} (f <x< h)^t \mid g' \\
\Rightarrow & \{\text{definition of } (f <x< h)^t\} \\
 & (f <x< h) \mid g \stackrel{t,a}{\rightarrow} (f^t <x< h^t) \mid g'
\end{array}
$$

Since g is x-free, so is g'. Assumed: $(f^t \mid g')$ <x< $h^t \sim (f^t$ <x< $h^t) \mid g'$.

(c) $h \overset{t,a}{\to} h'$, $a \neq !m$:

$$\begin{array}{ll} & h \overset{t,a}{\to} h' \\ \Rightarrow & \{\text{Asym1N}\} \\ & (f \mid g) \mathrel{<}x\mathrel{<} h \overset{t,a}{\to} (f \mid g)^t \mathrel{<}x\mathrel{<} h' \\ \Rightarrow & \{\text{definition } (f \mid g)^t\} \\ & (f \mid g) \mathrel{<}x\mathrel{<} h \overset{t,a}{\to} (f^t \mid g^t) \mathrel{<}x\mathrel{<} h' \end{array}$$

And,

$$\begin{array}{ll} & h \overset{t,a}{\to} h' \\ \Rightarrow & \{\text{Asym1N}\} \\ & f \mathrel{<}x\mathrel{<} h \overset{t,a}{\to} f^t \mathrel{<}x\mathrel{<} h' \\ \Rightarrow & \{\text{Sym1}\} \\ & (f \mathrel{<}x\mathrel{<} h) \mid g \overset{t,a}{\to} (f^t \mathrel{<}x\mathrel{<} h') \mid g^t \end{array}$$

Given g is x-free, and assumed $(f^t \mid g^t) \mathrel{<}x\mathrel{<} h' \sim (f^t \mathrel{<}x\mathrel{<} h') \mid g^t$.

(d) $h \overset{t,!m}{\to} h'$:

$$\begin{array}{ll} & h \overset{t,!m}{\to} h' \\ \Rightarrow & \{\text{Asym1V}\} \\ & (f \mid g) \mathrel{<}x\mathrel{<} h \overset{t,\tau}{\to} [m/x].(f \mid g)^t \\ \Rightarrow & \{\text{definition of } (f \mid g)^t\} \\ & (f \mid g) \mathrel{<}x\mathrel{<} h \overset{t,\tau}{\to} [m/x].(f^t \mid g^t) \end{array}$$

And,

$$\begin{array}{ll} & h \overset{t,!m}{\to} h' \\ \Rightarrow & \{\text{Asym1V}\} \\ & f \mathrel{<}x\mathrel{<} h \overset{t,\tau}{\to} [m/x].f^t \\ \Rightarrow & \{\text{Sym1}\} \\ & (f \mathrel{<}x\mathrel{<} h) \mid g \overset{t,\tau}{\to} [m/x].f^t \mid g^t \end{array}$$

To see that $[m/x].(f^t \mid g^t) \sim [m/x].f^t \mid g^t$, we show they are equal.

$$\begin{array}{ll} & [m/x].(f^t \mid g^t) \\ = & \{\text{substitution distributes}\} \\ & ([m/x].f^t) \mid ([m/x].g^t) \\ = & \{g \text{ is } x\text{-free, and so is } g^t\} \\ & ([m/x].f^t) \mid g^t \end{array}$$

8. $(f \mathrel{>}y\mathrel{>} g) \mathrel{<}x\mathrel{<} h \sim (f \mathrel{<}x\mathrel{<} h) \mathrel{>}y\mathrel{>} g$, provided g is x-free.

The transitions of the left side expression correspond to those of $(f \mathrel{>}y\mathrel{>} g)$ and h, i.e., of f and h. Similarly for the right side expression. We consider publication and non-publication transitions of f and h separately.

(a) $(f \overset{t,a}{\to} f'$, $a \neq !m)$

$$
\begin{array}{ll}
 & f \stackrel{t,a}{\to} f' \\
\Rightarrow & \{\text{Seq1N}\} \\
 & f \mathbin{>y>} g \stackrel{t,a}{\to} f' \mathbin{>y>} g \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f \mathbin{>y>} g) \mathbin{<x<} h \stackrel{t,a}{\to} (f' \mathbin{>y>} g) \mathbin{<x<} h^t
\end{array}
$$

And,

$$
\begin{array}{ll}
 & f \stackrel{t,a}{\to} f' \\
\Rightarrow & \{\text{Asym2}\} \\
 & f \mathbin{<x<} h \stackrel{t,a}{\to} f' \mathbin{<x<} h^t \\
\Rightarrow & \{\text{Seq1N}\} \\
 & (f \mathbin{<x<} h) \mathbin{>y>} g \stackrel{t,a}{\to} (f' \mathbin{<x<} h^t) \mathbin{>y>} g
\end{array}
$$

Assumed: $(f' \mathbin{>y>} g) \mathbin{<x<} h^t \sim (f' \mathbin{<x<} h^t) \mathbin{>y>} g$.

(b) $(f \stackrel{t,!m}{\to} f')$

$$
\begin{array}{ll}
 & f \stackrel{t,!m}{\to} f' \\
\Rightarrow & \{\text{Seq1V}\} \\
 & f \mathbin{>y>} g \stackrel{t,\tau}{\to} f' \mathbin{>y>} g \mid [m/y].g \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f \mathbin{>y>} g) \mathbin{<x<} h \stackrel{t,\tau}{\to} (f' \mathbin{>y>} g \mid [m/y].g) \mathbin{<x<} h^t
\end{array}
$$

And,

$$
\begin{array}{ll}
 & f \stackrel{t,!m}{\to} f' \\
\Rightarrow & \{\text{Asym2}\} \\
 & f \mathbin{<x<} h \stackrel{t,!m}{\to} f' \mathbin{<x<} h^t \\
\Rightarrow & \{\text{Seq1V}\} \\
 & (f \mathbin{<x<} h) \mathbin{>y>} g \stackrel{t,\tau}{\to} (f' \mathbin{<x<} h^t) \mathbin{>y>} g \mid [m/y].g
\end{array}
$$

To see that $(f' \mathbin{>y>} g \mid [m/y].g) \mathbin{<x<} h^t \sim (f' \mathbin{<x<} h^t) \mathbin{>y>} g \mid [m/y].g$

$$
\begin{array}{ll}
 & (f' \mathbin{>y>} g \mid [m/y].g) \mathbin{<x<} h^t \\
\sim & \{g \text{ is } x\text{-free. So, is } [m/y].g\} \\
 & (f' \mathbin{>y>} g \mathbin{<x<} h^t) \mid [m/y].g \\
\sim & \{\text{this law}\} \\
 & (f' \mathbin{<x<} h^t) \mathbin{>y>} g \mid [m/y].g
\end{array}
$$

(c) $(h \stackrel{t,a}{\to} h',\, a \neq\, !m)$

$$
\begin{array}{ll}
 & h \stackrel{t,a}{\to} h' \\
\Rightarrow & \{\text{Asym1N}\} \\
 & (f \mathbin{>y>} g) \mathbin{<x<} h \stackrel{t,a}{\to} (f \mathbin{>y>} g)^t \mathbin{<x<} h' \\
\Rightarrow & \{\text{definition of } (f \mathbin{>y>} g)^t\} \\
 & (f \mathbin{>y>} g) \mathbin{<x<} h \stackrel{t,a}{\to} (f^t \mathbin{>y>} g) \mathbin{<x<} h'
\end{array}
$$

And,

$$
\begin{array}{rl}
& h \stackrel{t,a}{\rightarrow} h' \\
\Rightarrow & \{\text{Asym1N}\} \\
& f <x< h \stackrel{t,a}{\rightarrow} f^t <x< h' \\
\Rightarrow & \{\text{Seq1N}\} \\
& (f <x< h) >y> g \stackrel{t,a}{\rightarrow} (f^t <x< h') >y> g
\end{array}
$$

Assumed: $(f^t >y> g) <x< h' \sim (f^t <x< h') >y> g$

(d) $(h \stackrel{t,!m}{\rightarrow} h')$

$$
\begin{array}{rl}
& h \stackrel{t,!m}{\rightarrow} h' \\
\Rightarrow & \{\text{Asym1V}\} \\
& (f >y> g) <x< h \stackrel{t,\tau}{\rightarrow} [m/x].(f >y> g)^t \\
\Rightarrow & \{\text{definition of } (f >y> g)^t\} \\
& (f >y> g) <x< h \stackrel{t,\tau}{\rightarrow} [m/x].(f^t >y> g)
\end{array}
$$

And,

$$
\begin{array}{rl}
& h \stackrel{t,!m}{\rightarrow} h' \\
\Rightarrow & \{\text{Asym1V}\} \\
& f <x< h \stackrel{t,\tau}{\rightarrow} [m/x].f^t \\
\Rightarrow & \{\text{Seq1N}\} \\
& (f <x< h) >y> g \stackrel{t,\tau}{\rightarrow} ([m/x].f^t) >y> g
\end{array}
$$

To see that $[c/x](f^t >y> g) \sim ([c/x]f^t) >y> g$:

$$
\begin{array}{rl}
& [c/x](f^t >y> g) \\
= & \{\text{substitution distributes}\} \\
& ([c/x]f^t) >y> ([c/x]g) \\
= & \{g \text{ is } x\text{-free. So, } [c/x]g = g\} \\
& ([c/x]f^t) >y> g
\end{array}
$$

9. $(f <x< g) <y< h \sim (f <y< h) <x< g$, provided g is y-free and h is x-free.

We have to consider the transitions corresponding to those of f , g and h. The roles of g and h are symmetric; so, we consider only the transitions of g.

(a) $(f \stackrel{t,a}{\rightarrow} f')$

$$
\begin{array}{rl}
& f \stackrel{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Asym2}\} \\
& f <x< g \stackrel{t,a}{\rightarrow} f' <x< g^t \\
\Rightarrow & \{\text{Asym2}\} \\
& (f <x< g) <y< h \stackrel{t,a}{\rightarrow} (f' <x< g^t) <y< h^t
\end{array}
$$

And,

$$
\begin{array}{rl}
& f \stackrel{t,a}{\rightarrow} f' \\
\Rightarrow & \{\text{Asym2}\}
\end{array}
$$

$$
\begin{array}{ll}
 & f <y< h \stackrel{t,a}{\rightarrow} f' <y< h^t \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f <y< h) <x< g \stackrel{t,a}{\rightarrow} (f' <y< h^t) <x< g^t
\end{array}
$$

Assumed: $(f' <x< g^t) <y< h^t \sim (f' <y< h^t) <x< g^t$

(b) $(g \stackrel{t,a}{\rightarrow} g',\ a \neq !m)$

$$
\begin{array}{ll}
 & g \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Asym1N}\} \\
 & f <x< g \stackrel{t,a}{\rightarrow} f^t <x< g' \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f <x< g) <y< h \stackrel{t,a}{\rightarrow} (f^t <x< g') <y< h^t
\end{array}
$$

And,

$$
\begin{array}{ll}
 & g \stackrel{t,a}{\rightarrow} g' \\
\Rightarrow & \{\text{Asym1N}\} \\
 & (f <y< h) <x< g \stackrel{t,a}{\rightarrow} (f <y< h)^t <x< g' \\
\Rightarrow & \{\text{definition of } (f <y< h)^t\} \\
 & (f <y< h) <x< g \stackrel{t,a}{\rightarrow} (f^t <y< h^t) <x< g'
\end{array}
$$

Assumed: $(f^t <x< g') <y< h^t \sim (f^t <y< h^t) <x< g'$.

(c) $(g \stackrel{t,!m}{\rightarrow} g')$

$$
\begin{array}{ll}
 & g \stackrel{t,!m}{\rightarrow} g' \\
\Rightarrow & \{\text{Asym1V}\} \\
 & f <x< g \stackrel{t,\tau}{\rightarrow} [c/x]f^t \\
\Rightarrow & \{\text{Asym2}\} \\
 & (f <x< g) <y< h \stackrel{t,\tau}{\rightarrow} [c/x]f^t <y< h^t
\end{array}
$$

And,

$$
\begin{array}{ll}
 & g \stackrel{t,!m}{\rightarrow} g' \\
\Rightarrow & \{\text{Asym1V}\} \\
 & (f <y< h) <x< g \stackrel{t,\tau}{\rightarrow} [c/x](f <y< h)^t \\
\Rightarrow & \{\text{definition of } (f <y< h)^t\} \\
 & (f <y< h) <x< g \stackrel{t,\tau}{\rightarrow} [c/x](f^t <y< h^t)
\end{array}
$$

To see that $[c/x]f^t <y< h^t \sim [c/x](f^t <y< h^t)$,

$$
\begin{array}{ll}
 & [c/x](f^t <y< h^t) \\
= & \{\text{substitution distributes}\} \\
 & [c/x]f^t <y< [c/x]h^t \\
\Rightarrow & \{h \text{ is } x\text{-free and so is } h^t\} \\
 & [c/x]f^t <y< h^t
\end{array}
$$

10. $\mathbf{0} <x< M(v) \sim M(v) \gg \mathbf{0}$, for any site M and value v.

The only transition of the constituent expression $M(v)$ is $M(v) \stackrel{0,\tau}{\rightarrow} ?k$.

$$
\begin{array}{rl}
 & M(v) \overset{0,\tau}{\to} ?k \\
\Rightarrow & \{\text{Asym1N}\} \\
 & \mathbf{0} <x< M(v) \overset{0,\tau}{\to} \mathbf{0}^t <x< ?k \\
\Rightarrow & \{\text{definition of } \mathbf{0}^t\} \\
 & \mathbf{0} <x< M(v) \overset{0,\tau}{\to} \mathbf{0} <x< ?k
\end{array}
$$

And,

$$
\begin{array}{rl}
 & M(v) \overset{0,\tau}{\to} ?k \\
\Rightarrow & \{\text{Seq1N}\} \\
 & M(v) \gg \mathbf{0} \overset{0,\tau}{\to} ?k \gg \mathbf{0}
\end{array}
$$

Assumed: $\mathbf{0} <x< ?k \sim ?k \gg \mathbf{0}$

Corollaries

(a) $f <x< M(v) \sim f \mid M(v) \gg \mathbf{0}$, if f is x-free

$$
\begin{array}{rl}
 & f <x< M(v) \\
\sim & \{\text{proved}\} \\
 & (f \mid \mathbf{0}) <x< M(v) \\
\sim & \{f \text{ is } x\text{-free}\} \\
 & f \mid (\mathbf{0} <x< M(v)) \\
\sim & \{\text{proved}\} \\
 & f \mid M(v) \gg \mathbf{0}
\end{array}
$$

(b) $f <x< \mathbf{0} \sim f$, if f is x-free

$$
\begin{array}{rl}
 & f <x< \mathbf{0} \\
\sim & \{\text{from above}\} \\
 & f \mid \mathbf{0} \gg \mathbf{0} \\
\sim & \{\text{proved}\} \\
 & f \mid \mathbf{0} \\
\sim & \{\text{proved}\} \\
 & f
\end{array}
$$

1.6.2 Weak Bisimulation

Lemma 5

1. $f >x> let(x) \cong f$
2. $E(p) \cong [p/x].g$, *where* $E(x) \ \underline{\Delta}\ g$

Proof:

1. $f >x> let(x) \cong f$:

   The only transitions of $f >x> let(x)$ are from the subexpression f.

- (Seq1N): Then $f \overset{t,a}{\to} f'$, where a is not a publication event. And

$$\begin{array}{ll} & f \overset{t,a}{\to} f' \\ \Rightarrow & \{(\textsc{Seq1N})\} \\ & f \mathbin{>x>} let(x) \overset{t,\tau}{\to} f' \mathbin{>x>} let(x) \end{array}$$

Assumed: $f' \mathbin{>x>} let(x) \cong f'$

- (Seq1V): Then $f \overset{t,!m}{\to} f'$. And

$$\begin{array}{ll} & f \mathbin{>x>} let(x) \\ \overset{t,\tau}{\to} & \{(\textsc{Seq1V})\} \\ & f' \mathbin{>x>} let(x) \mid let(m) \\ \overset{0,!m}{\to} & \{let(m) \overset{0,!m}{\to} \mathbf{0} \text{ and } (\textsc{Sym2})\} \\ & f' \mathbin{>x>} let(x) \mid \mathbf{0} \\ \sim & \{\text{Proved above}\} \\ & f' \mathbin{>x>} let(x) \end{array}$$

Assumed: $f' \mathbin{>x>} let(x) \cong f'$

2. $E(p) \cong [p/x].g$, where $E(x) \ \underline{\Delta}\ g$: Suppose $[p/x].g \overset{t,a}{\to} g'$. Then

$$\begin{array}{ll} & E(p) \\ \overset{0,\tau}{\to} & \{(\text{Def}),\ E(x) \ \underline{\Delta}\ g\} \\ & [p/x].g \\ \overset{t,a}{\to} & \{\text{assumed}\} \\ & g' \end{array}$$

Chapter 2

Combinators applied to Executions

In Section 2.1 we characterize the execution sets of the base expressions. In Section 2.2, we define $U * V$ where U and V are sets of executions and $*$ is any Orc combinator: $\;|\;$, $\;>x>\;$, and $\;<x<\;$. These definitions give the meaning function for each combinator when applied to sets. We prove results about monotonicity (Section 2.2.4, page 38) and distributivity (Section 2.2.5, page 40) of the combinators.

In subsequent sections of this chapter, we prove that for expressions f and g, $[\![f * g]\!] = [\![f]\!] * [\![g]\!]$.

Throughout we assume a fixed environment mapping Σ and set of definitions $\mathcal{D}$. We denote the set of times by $\mathcal{T}$.

2.1 Base Expressions

For variable x and set of sequences A, the *exclusion of* x *from* A, written $A \backslash x$, is defined by

$$A \backslash x \;\;\underline{\Delta}\;\; \{p \in A \mid [m/x] \text{ does not occur in } p\}.$$

Theorem 8 *The following sets characterize the executions of the base Orc expressions:*

- $[\![\mathbf{0}]\!] = D(\mathcal{T})$
- $[\![?k]\!] = (\cup (t, m) : (t, m) \in k : D(t) \cdot (t, !m) \cdot [\![\mathbf{0}]\!]_t) \cup (\cup \omega : \omega \in k : [\![\mathbf{0}]\!])$
- $[\![M(m)]\!] = (\cup k : k \in \Sigma(M, m) : D(0) \cdot (0, \tau) \cdot [\![?k]\!])$
- $[\![M(x)]\!] = (\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t) \backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t)$

Proof:

- $[\![\,\mathbf{0}\,]\!] = D(\mathcal{T})$: Only the rule (Subst) applies to $\mathbf{0}$, hence every execution is a finite sequence of substitution events with nondecreasing time.

- $[\![\,?k\,]\!] = (\cup(t,m) : (t,m) \in k : D(t) \cdot (t,\, !m) \cdot [\![\,\mathbf{0}\,]\!]_t) \cup (\cup\omega : \omega \in k : [\![\,\mathbf{0}\,]\!])$: The proof is by mutual inclusion.

  - $[\![\,?k\,]\!] \subseteq (\cup(t,m) : (t,m) \in k : D(t)\cdot(t,\, !m)\cdot[\![\,\mathbf{0}\,]\!]_t)\cup(\cup\omega : \omega \in k : [\![\,\mathbf{0}\,]\!])$: Assume $p \in [\![\,?k\,]\!]$. If $p = \epsilon$ the result follows because $\epsilon \in D(t)$ and $\epsilon \in [\![\,\mathbf{0}\,]\!]$. Otherwise let $p = aq_t$, so

    $$?k \overset{t,a}{\to} f \overset{q}{\Rightarrow} f'.$$

    By the operational semantics, the transition $\overset{t,a}{\to}$ must be due to either rule (Return) or (Subst).

    In the (Return) case, we have $(t,m) \in k$, $a = !m$ and $f = \mathbf{0}$. Since $z \in [\![\,\mathbf{0}\,]\!]$, it follows that $(t,\, !m)q_t \in (t,\, !m)[\![\,\mathbf{0}\,]\!]_t$ and also, since $\epsilon \in D(t)$, that $D(t)(t,\, !m)q_t \in (t,\, !m)[\![\,\mathbf{0}\,]\!]_t$.

    In the (Subst) case, we have

    $$?k \overset{t,[m/x]}{\to} ?k^t \overset{q}{\Rightarrow} f'.$$

    By Thm. 6 on page 15 we have $q_t \in [\![\,?k\,]\!]$, and so by induction we have either $q_t \in (\cup(t,m) : (t,m) \in k : D(t) \cdot (t,\, !m) \cdot [\![\,\mathbf{0}\,]\!]_t)$ or $q_t \in (\cup\omega : \omega \in k : [\![\,\mathbf{0}\,]\!])$. In either case prepending a substitution event at time t preserves the inclusion because $aD(t) \subseteq D(t)$ and $a[\![\,\mathbf{0}\,]\!] \subseteq [\![\,\mathbf{0}\,]\!]$.

  - $(\cup(t,m) : (t,m) \in k : D(t)\cdot(t,\, !m)\cdot[\![\,\mathbf{0}\,]\!]_t)\cup(\cup\omega : \omega \in k : [\![\,\mathbf{0}\,]\!]) \subseteq [\![\,?k\,]\!]$: First assume $p \in (\cup\omega : \omega \in k : [\![\,\mathbf{0}\,]\!])$. Since $\omega \in k$, the only rule that applies to $?k$ is (Subst), and $?k^t = ?k$ for any t. The result follows by induction on the length of p.

    Otherwise, assume $p \in (\cup(t,m) : (t,m) \in k : D(t) \cdot (t,\, !m) \cdot [\![\,\mathbf{0}\,]\!]_t)$. Hence, for some $(t,m) \in k$, $p \in D(t) \cdot (t,\, !m) \cdot [\![\,\mathbf{0}\,]\!]_t$. If $p = \epsilon$, then the result follows by prefix closure of execution sets. Otherwise $p \neq \epsilon$. Suppose $p = q(t,\, !m)r_t$, where $r \in [\![\,\mathbf{0}\,]\!]$ and $p \in D(t)$; the other cases follow by prefix-closure of execution sets. If $q = \epsilon$, then $p = (t,\, !m)r_t$ and $?k \overset{t,\,!m}{\to} \mathbf{0} \overset{r}{\Rightarrow}$ by the operational semantics. Otherwise, $q = aq_s$ and $p = ap'_s$, where a is a substitution event at time s, for some $s \leq t$, and $p' = q_t(t,\, !m)r_t$. By induction, $p'_s \in [\![\,?k\,]\!]$, and so by Thm 6 on page 15 $p' \in [\![\,?k^s\,]\!]$. Finally,

    $$?k \overset{a}{\to} ?k^s \overset{p'}{\Rightarrow} .$$

- $[\![\,M(m)\,]\!] = (\cup k : k \in \Sigma(M,m) : D(0) \cdot (0,\tau) \cdot [\![\,?k\,]\!])$: The proof is by mutual inclusion.

– $[\![M(m)]\!] \subseteq (\cup k : k \in \Sigma(M,m) : D(0) \cdot (0,\tau) \cdot [\![?k]\!])$: Consider $p \in [\![M(m)]\!]$. If $p = \epsilon$ the result follows because $\epsilon \in D(0)$. Otherwise

$$M(m) \overset{t,a}{\rightarrow} f \overset{q}{\Rightarrow} f'.$$

By the operational semantics, the transition $\overset{t,a}{\rightarrow}$ must be due to either rule (Call) or (Subst).

In the (Call) case, we have

$$M(m) \overset{0,\tau}{\rightarrow} ?k \overset{q}{\Rightarrow} f'.$$

Since $q \in [\![?k]\!]$, it follows that $(0,\tau)q \in (0,\tau)[\![?k]\!]$, and hence $(0,\tau)q \in D(0)(0,\tau)[\![?k]\!]$.

In the (Subst) case, we have

$$M(m) \overset{0,[m/x]}{\rightarrow} M(m) \overset{q}{\Rightarrow} f'.$$

Note that the substitution must occur at time 0 because $M(m)^t = \perp$ for $t > 0$. By induction, $q \in (\cup k : k \in \Sigma(M,m) : D(0) \cdot (0,\tau) \cdot [\![?k]\!])$. The result follows because $(0,[m/x])D(0) \subseteq D(0)$.

– $(\cup k : k \in \Sigma(M,m) : D(0) \cdot (0,\tau) \cdot [\![?k]\!]) \subseteq [\![M(m)]\!]$: Consider $p \in (\cup k : k \in \Sigma(M,m) : D(0) \cdot (0,\tau) \cdot [\![?k]\!])$. Assume $p = q(0,\tau)r$, where $q \in D(0)$ and $r \in [\![?k]\!]$; the other cases follow by prefix-closure of execution sets. If $p = \epsilon$, the result follows by prefix-closure of execution sets. Otherwise $p \neq \epsilon$. If $q = \epsilon$, then $p = (0,\tau)r$. Since $r \in [\![?k]\!]$, we have by rule (Call)

$$M(m) \overset{0,\tau}{\rightarrow} ?k \overset{r}{\Rightarrow} .$$

Otherwise $q = aq'$ and $p = ap'$, where a is a substitution event at time 0. By induction we have $p' \in [\![M(m)]\!]$, and the result follows from

$$M(m) \overset{a}{\rightarrow} M(m) \overset{p'}{\Rightarrow} .$$

- $[\![M(x)]\!] = (\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t)\backslash x \cdot (t,[m/x]) \cdot [\![M(m)]\!]_t)$: The proof is by mutual inclusion.

  – $[\![M(x)]\!] \subseteq (\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t)\backslash x \cdot (t,[m/x]) \cdot [\![M(m)]\!]_t)$: Consider $p \in [\![M(x)]\!]$. If $p = \epsilon$ the result follows because $\epsilon \in D(t)$ for all times t. Otherwise

$$M(x) \overset{t,a}{\rightarrow} f \overset{q}{\Rightarrow} f'.$$

By the operational semantics, this transition must be by rule (Subst).

If the substitution is to a variable $y \neq x$, then

$$M(x) \stackrel{t,[m/y]}{\rightarrow} M(x) \stackrel{q}{\Rightarrow} f'.$$

By induction, $q \in (\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t)\backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t)$. Then for some $s \in \mathcal{T}$ and value m, $q \in D(s)\backslash x \cdot (s, [m/y]) \cdot [\![M(m)]\!]_s$. Since $q_t \in D(s+t)\backslash x \cdot (s+t, [m/y]) \cdot [\![M(m)]\!]_{s+t}$. The result follows because $(t, [m/y])D(s+t)\backslash x \subseteq D(s+t)\backslash x$.

Otherwise we have

$$M(x) \stackrel{t,[m/x]}{\rightarrow} M(m) \stackrel{q}{\Rightarrow} .$$

Then $q \in [\![M(m)]\!]$ and so $(t, [m/x])q_t \in (t, [m/x])[\![M(m)]\!]_t$, from which the result follows.

– $(\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t)\backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t) \subseteq [\![M(x)]\!]$: Consider $p \in (\cup t, m : t \in \mathcal{T}, m \in \mathcal{V} : D(t)\backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t)$. So, for some $t \in \mathcal{T}$ and value m, $p \in D(t)\backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t$. Assume $p = q(t, [m/x])r_t$, where $q \in D(t)\backslash x$ and $r \in [\![M(m)]\!]$; the other cases follow by prefix-closure of execution sets. If $p = \epsilon$, the result follows by prefix-closure of execution sets. Otherwise, $p \neq \epsilon$.

If $q = \epsilon$, then by $r \in [\![M(m)]\!]$ and rule (SUBST),

$$M(x) \stackrel{t,[m/x]}{\rightarrow} M(m) \stackrel{r}{\Rightarrow} .$$

Otherwise, $q = (s, [m/y])q'$ and $p = (s, [m/y])p'$, where $s \leq t$ and $y \neq x$. By induction, $p' \in [\![M(x)]\!]$, and the result follows from

$$M(x) \stackrel{s,[m/y]}{\rightarrow} M(x) \stackrel{p'}{\Rightarrow} .$$

2.2 Meanings of Execution Combinators

2.2.1 Meaning of Symmetric Composition

We introduce *guarded set*, a notational device, which simplifies our definition and subsequent algebraic manipulations. Let p be a predicate and S a set. Then

$$[p \ \rightarrow \ S] = \begin{cases} S & \text{if } p \\ \{\epsilon\} & \text{otherwise} \end{cases}$$

We call $[p \ \rightarrow \ S]$ a *guarded set*, and predicate p its *guard*.

We define *merge* over two sequences that yields a non-empty set of sequences. The merge of u and v, written as $u \mid v$ is defined by the following two rules. Henceforth, $a \simeq b$ means that a and b are identical substitution events, and $a \preceq b$ means that a is a base event and $a.time \leq b.time$.

$$\epsilon \mid v = \{\epsilon\}, \ u \mid \epsilon = \{\epsilon\},$$
$$au \mid bv = [a \simeq b \rightarrow a(u \mid v)] \cup [a \preceq b \rightarrow a(u \mid bv)] \cup [b \preceq a \rightarrow b(au \mid v)]$$

We define $|$ to be coercive so that

$$U \mid V = (\cup u, v:\ u \in U,\ v \in V:\ u \mid v)$$

Therefore $|$ distributes over set union, and observation 2, as well as observations 3 and 4, page 7, apply.

2.2.2 Meaning of Sequential Composition

In this section, we deal with expressions of the form $f > x > g$; variable x will be treated specially in this section. We write *own-substitution* for a substitution to x and *other-substitution* for any other substitution, i.e., made to a variable other than x.

We define $p > x > V$, for sequence p and set V, as a set of sequences.

$$\epsilon > x > \phi = \phi,$$
$$\epsilon > x > V = \{\epsilon\}, \text{ for } V \neq \phi$$

$$ap > x > V = \begin{cases} a(p > x > V) & \text{if } c_1(a) \quad \text{(SCD1)} \\ a(p > x > V') & \text{if } c_2(a) \quad \text{(SCD2)} \\ (t,\tau)(p > x > V \mid V_t'') & \text{if } c_3(a) \quad \text{(SCD3)} \end{cases}$$

where

$c_1(a)$	is	"a is a non-publication base event or an own-substitution",
$c_2(a)$	is	"a is an other-substitution (t,b)"; here $V' = V \backslash (0,b)$,
$c_3(a)$	is	"a is publication $(t, !m)$"; here $V'' = V \backslash (0, [m/x])$.

Coerce the definition for set U:

$$U > x > V = (\cup u:\ u \in U:\ u > x > V)$$

The form of coercion for sequential composition is different from that for merge. Merge is defined with two sequences as arguments, whereas sequential composition has a set as a second argument.

2.2.3 Meaning of Asymmetric Composition

In this section, we deal with expressions of the form $f < x < g$; variable x will be treated specially in this section. We write *own-substitution* for a substitution to x and *other-substitution* for any other substitution, i.e., made to a variable other than x.

Constrained Partial and Full Merge

Let $a \approx_x b$ mean that a and b are identical other-substitutions. As before, $a \preceq b$ means that a is a base event and $a.time \leq b.time$. Let $b \preceq\!\!\prec_x a$ denote that: (either b is a base event or an own-substitution) and $b.time \leq a.time$. Define *partial merge*, $|_x$, an extension of merge, over a pair of sequences.

$\epsilon|_x v = \{\epsilon\}$, $u|_x \epsilon = \{\epsilon\}$
$au|_x bv = [a \approx_x b \rightarrow a(u|_x v)] \cup [a \preceq b \rightarrow a(u|_x bv)] \cup [b \precneqq_x a \rightarrow b(au|_x v)]$

Coerce the definition to sets U and V:

$$U|_x V = (\cup u, v :\ u \in U,\ v \in V :\ u|_x v)$$

Next, we define *full merge*, using a notation similar to guarded sets. Let $\langle p \ \longrightarrow \ S\rangle$ be set S if p is *true* and the empty set, ϕ, if p is *false*. Note that, unlike the guarded sets of Section 3.2, page 77, the default value here is the empty set, not $\{\epsilon\}$. Therefore, $\langle p \ \longrightarrow \ S\rangle \cup \langle \mathit{false} \ \longrightarrow \ S'\rangle = \langle p \ \longrightarrow \ S\rangle$.

Constrained full merge of u and v, written as $u +_x v$, is a set of sequences defined as follows.

$$u +_x \epsilon = \begin{cases} \{u\} & \text{if } u \text{ contains no substitution event} \\ \phi & \text{otherwise} \end{cases}$$
$$\epsilon +_x v = \begin{cases} \{v\} & \text{if } v \text{ contains no other-substitution} \\ \phi & \text{otherwise} \end{cases}$$
$$au +_x bv = \langle a \approx_x b \rightarrow a(u +_x v)\rangle \cup \langle a \preceq b \rightarrow a(u +_x bv)\rangle \cup \langle b \precneqq_x a \rightarrow b(au +_x v)\rangle$$

Coerce the definition to sets U and V:

$$U +_x V = (\cup u, v :\ u \in U,\ v \in V :\ u +_x v)$$

Definition of Asymmetric Composition

Define

$d_0(u, v) \ \underline{\Delta} \ \ u$ and v have the same sequence of other-substitutions
$d_1(u, v) \ \underline{\Delta} \ \ u$ has no own-substitution and v has no publication
$d_2(u, v) \ \underline{\Delta}$
u is of the form $u'(t, [m/x])u''$, v is of the form $v'(t, !m)v''$, and $d_0(u', v')$, $d_1(u', v')$

We now define $u <x< v$.

$$u <x< v = \begin{cases} u|_x v & \text{if } d_1(u, v) \\ (u' +_x v')(t, \tau)u'' & \text{if } d_2(u, v) \\ \phi & \text{otherwise} \end{cases}$$
$$U <x< V = (\cup u, v : u \in U, v \in V : u <x< v)$$

2.2.4 Monotonicity

Theorem 9 For any Orc combinator $*$, $U * V$ is ϕ if either U or V is ϕ. Further, suppose $U \subseteq U'$ and $V \subseteq V'$. Then, $U * V \subseteq U' * V'$.

Proof: The first part follows from the definitions of meaning functions where $*$ is either $|$ or $<x<$. Combinator $>x>$ is coercive in its left argument; so, $\phi >x> V = \phi$. To show $U >x> \phi = \phi$, we show $p >x> \phi = \phi$, for any p. This can be proved by applying induction on the length of p.

For the next part, combinator $\mid$ is coercive in both of its arguments; so, $U \subseteq U'$ and $V \subseteq V'$ implies $U \mid V \subseteq U' * V'$. Similar remarks apply for $<x<$. Combinator $>x>$ is coercive in its left argument; so, $U \subseteq U'$ implies $U >x> V \subseteq U' >x> V$. Next, we prove that $V \subseteq W$ implies $U >x> V \subseteq U >x> W$.

If $U = \phi$, from the definition, $U >x> V = \phi = U >x> W$. For $U \neq \phi$, let $p \in U$. We show $p >x> V \subseteq p >x> W$. Proof is by induction on the length of p.

For $p = \epsilon$, the result follows by inspection of the definition. Next, we prove $ap >x> V \subseteq ap >x> W$. Consider three cases:

- $c_1(a)$:

$$\begin{array}{ll}
 & ap >x> V \\
= & \{\text{definition of meaning function}\} \\
 & a(p >x> V) \\
\subseteq & \{V \subseteq W\text{; inductively, } p >x> V \subseteq p >x> W\} \\
 & a(p >x> W) \\
= & \{\text{definition of meaning function}\} \\
 & ap >x> W
\end{array}$$

- $c_2(a)$:

$$\begin{array}{ll}
 & ap >x> V \\
= & \{\text{definition of meaning function}\} \\
 & a(p >x> V') \\
\subseteq & \{V \subseteq W\text{; from the definition of } \backslash(0,b) \text{ from Section 1.3.1, page 6, } V' \subseteq W'\text{;} \\
 & \quad \text{apply induction}\} \\
 & a(p >x> W') \\
= & \{\text{definition of meaning function}\} \\
 & ap >x> W
\end{array}$$

- $c_3(a)$:

$$\begin{array}{ll}
 & ap >x> V \\
= & \{\text{definition of meaning function}\} \\
 & a(p >x> V \mid (V'')_t) \\
\subseteq & \{V \subseteq W\text{; inductively, } p >x> V \subseteq p >x> W \\
 & \quad \mid \text{ is monotonic in both arguments}\} \\
 & a(p >x> W \mid (V'')_t) \\
= & \{\text{from the definition of } \backslash(0,b) \text{ from Section 1.3.1, page 6, } V'' \subseteq W''\text{;} \\
 & \quad \text{given } V'' \subseteq W''\text{, } (V'')_t \subseteq (W'')_t\} \\
 & a(p >x> W \mid (W'')_t) \\
= & \{\text{definition of meaning function}\} \\
 & ap >x> W
\end{array}$$

2.2.5 Distributivity

Lemma 6 Let $V_0 \subseteq V_1 \cdots$. Then, $(\cup i :: p >x> V_i) = p >x> (\cup i :: V_i)$, for any p.

Proof:
A note on notation: we use subscripts i, j, k and n (in addition to a 0 and 1) to designate sets such as V_0. At thesame time, we will write V_t, for time-shift of V. And, they will be combined a few times as in $(V_i)_t$; we use context to differentiate the two usages.

Proof is by mutual inclusion.

- $(\cup i :: p >x> V_i) \subseteq p >x> (\cup i :: V_i)$: For any i,

$$
\begin{array}{ll}
 & p >x> V_i \\
\subseteq & \{V_i \subseteq (\cup i :: V_i);\ >x> \text{ is monotonic from Theorem 9, page 38}\} \\
 & p >x> V
\end{array}
$$

Therefore, $(\cup i :: p >x> V_i) \subseteq p >x> (\cup i :: V_i)$.

- $p >x> (\cup i :: V_i) \subseteq (\cup i :: p >x> V_i)$:

If $(\cup i :: V_i) = \phi$ then, $V_i = \phi$, for all i. So, $p >x> (\cup i :: V_i) = \phi = (\cup i :: p >x> V_i)$. Assume, henceforth, that $(\cup i :: V_i) \neq \phi$. Then, there is some V_j, $V_j \neq \phi$. The proof is by induction on the length of p.

If $p = \epsilon$, then $p >x> (\cup i :: V_i) = \{\epsilon\}$ and $p >x> V_j = \{\epsilon\} \subseteq (\cup i :: p >x> V_i)$.

Next, we show that $ap >x> (\cup i :: V_i) \subseteq (\cup i :: ap >x> V_i)$, for any ap. We consider three cases, depending on a.

Case 1) $c_1(a)$:

$$
\begin{array}{ll}
 & ap >x> (\cup i :: V_i) \\
= & \{c_1(a)\} \\
 & a(p >x> (\cup i :: V_i)) \\
\subseteq & \{\text{induction on } p >x> (\cup i :: V_i)\} \\
 & a(\cup i :: p >x> V_i) \\
= & \{\text{concatenation distributes over set union}\} \\
 & (\cup i :: a(p >x> V_i)) \\
= & \{c_1(a)\} \\
 & (\cup i :: ap >x> V_i)
\end{array}
$$

Case 2) $c_2(a)$:

$$
\begin{array}{ll}
 & ap >x> (\cup i :: V_i) \\
= & \{c_2(a)\} \\
 & a(p >x> (\cup i :: V_i)') \\
= & \{\text{removal operator distributes over set union}\} \\
 & a(p >x> (\cup i :: V_i')) \\
\subseteq & \{\text{induction on } p >x> (\cup i :: V_i')\}
\end{array}
$$

$$
\begin{array}{ll}
 & a(\cup i :: p >x> V_i') \\
= & \{\text{concatenation distributes over set union}\} \\
 & (\cup i :: a(p >x> V_i')) \\
= & \{c_2(a)\} \\
 & (\cup i :: ap >x> V_i)
\end{array}
$$

Case 3) $c_3(a)$:

$$
\begin{array}{ll}
 & ap >x> (\cup i :: V_i) \\
= & \{c_3(a)\} \\
 & a(p >x> (\cup i :: V_i) \mid ((\cup i :: V_i)'')_t) \\
= & \{\text{removal operator distributes over set union}\} \\
 & a(p >x> (\cup i :: V_i) \mid (\cup i :: V_i'')_t) \\
= & \{\text{time-shift distributes over set union}\} \\
 & a(p >x> (\cup i :: V_i) \mid (\cup i :: (V_i'')_t)) \\
\subseteq & \{\text{induction on } p >x> (\cup i :: V_i); \\
 & \quad \text{merge and concatenation are montonic wrt set union}\} \\
 & a(\cup i :: p >x> V_i) \mid (\cup i :: (V_i'')_t)
\end{array}
$$

Now it is sufficient to show that for any $q \in (\cup i :: p >x> V_i)$ and $r \in (\cup i :: (V_i'')_t)$, $a(q \mid r) \subseteq (\cup i :: ap >x> V_i)$.

Since $q \in (\cup i :: p >x> V_i)$, for some j, $q \in p >x> V_j$.
Since $r \in (\cup i :: (V_i'')_t)$, for some k, $r \in (V_k'')_t$.
Let $n = \max(j, k)$. Then $V_j \subseteq V_n$ and $V_k \subseteq V_n$.

$$
\begin{array}{ll}
 & q \in p >x> V_j \\
\Rightarrow & \{V_j \subseteq V_n \text{ and } >x> \text{ is monotonic from Theorem 9, page 38}\} \\
 & q \in p >x> V_n
\end{array}
$$

Similarly, from $r \in (V_k'')_t$ and $V_k \subseteq V_n$, we get $r \in (V_n'')_t$. Then,

$$
\begin{array}{ll}
 & a(q \mid r) \\
\subseteq & \{q \in p >x> V_n; \\
 & \quad \text{merge and concatenation are montonic wrt set union}\} \\
 & a(p >x> V_n \mid r) \\
\subseteq & \{r \in (V_n'')_t\} \\
 & a(p >x> V_n \mid (V_n'')_t) \\
= & \{c_3(a)\} \\
 & ap >x> V_n \\
\subseteq & \{\text{set theory}\} \\
 & (\cup i :: ap >x> V_i)
\end{array}
$$

Next, we establish that every Orc combinator distributes over set union in in its left argument and in its right argument under a certain condition. Below $*$ is any Orc combinator: $\mid$, $>x>$ or $<x<$.

Theorem 10 For any U and V,

1. (Left Distributivity) $(\cup i :: P_i * V) = (\cup i :: P_i) * V$, for a family of sets P_i.

2. (Right Distributivity) $(\cup i :: U * Q_i) = U * (\cup i :: Q_i)$, for a sequence of sets Q_i, where $Q_0 \subseteq Q_1 \subseteq \cdots$.

Proof: Left Distributivity follows from the definitions of the combinators over sets.

$$
\begin{array}{cl}
 & (\cup i :: P_i * V) \\
= & \{\text{expanding } P_i * V\} \\
 & (\cup i :: (\cup p : p \in P_i : p * V)) \\
= & \{\text{set theory}\} \\
 & (\cup p : p \in (\cup i :: P_i) * V) \\
= & \{\text{definition of coercion}\} \\
 & (\cup i :: P_i) * V
\end{array}
$$

Right distributivity for $\mid$ and $<\!x\!<$ follow similarly, because they are coercive in both arguments. Now, we show that $(\cup i :: U >\!x\!> Q_i) = U >\!x\!> (\cup i :: Q_i)$, for a sequence of sets Q_i, where $Q_0 \subseteq Q_1 \subseteq \cdots$.

$$
\begin{array}{cl}
 & (\cup i :: U >\!x\!> Q_i) \\
= & \{\text{expanding } U >\!x\!> Q_i\} \\
 & (\cup i :: (\cup p :\ p \in U :\ p >\!x\!> Q_i)) \\
= & \{\text{set theory}\} \\
 & (\cup p :\ p \in U :\ (\cup i ::\ p >\!x\!> Q_i)) \\
= & \{(\cup i ::\ p >\!x\!> Q_i) = p >\!x\!> (\cup i ::\ Q_i), \text{ from Lemma 6, page 40}\} \\
 & (\cup p :\ p \in U :\ p >\!x\!> (\cup i ::\ Q_i)) \\
= & \{\text{definition of coercion}\} \\
 & U >\!x\!> (\cup i :: Q_i)
\end{array}
\qquad \square
$$

We note that right distributivity holds for the combinators $\mid$ and $<\!x\!<$ for arbitrary sets Q_is; the additional condition $Q_0 \subseteq Q_1 \subseteq \cdots$ is not required. This condition is needed only for $>\!x\!>$, as we outline below.

We show that $U >\!x\!> (V \cup W) \neq U >\!x\!> V \cup U >\!x\!> W$, in general; in fact, $ap >\!x\!> (V \cup W) \neq (ap >\!x\!> V) \cup (ap >\!x\!> W)$ when $c_3(a)$ holds. Let $V_1 = p >\!x\!> V$ and $V_2 = (V'')_t$, and W_1 and W_2 are similarly defined.

$$
\begin{array}{cl}
 & ap >\!x\!> (V \cup W) \\
= & \{c_3(a)\} \\
 & a(p >\!x\!> (V \cup W) \mid ((V \cup W)'')_t) \\
\supseteq & \{p >\!x\!> (V \cup W) \supseteq p >\!x\!> V \cup p >\!x\!> W\} \\
 & a((p >\!x\!> V \cup p >\!x\!> W) \mid ((V \cup W)'')_t) \\
= & \{V_1 = p >\!x\!> V \text{ and } V_2 = (V'')_t\text{; similarly for } W_1 \text{ and } W_2\} \\
 & a((V_1 \cup W_1) \mid (V_2 \cup W_2)) \\
= & \{\text{coercion}\} \\
 & a((V_1 \mid V_2) \cup (V_1 \mid W_2) \cup (W_1 \mid V_2) \cup (W_1 \mid W_2))
\end{array}
$$

And,

$$
\begin{array}{ll}
 & (ap >x> V) \cup (ap >x> W) \\
= & \{c_3(a)\} \\
 & a(p >x> V \mid (V'')_t) \cup a(p >x> W \mid (W'')_t) \\
= & \{V_1 = p >x> V \text{ and } V_2 = (V'')_t\text{; similarly for } W_1 \text{ and } W_2\} \\
 & a(V_1 \mid V_2) \cup a(W_1 \mid W_2) \\
= & \{\text{rewriting}\} \\
 & a((V_1 \mid V_2) \cup (W_1 \mid W_2))
\end{array}
$$

Thus, in general, $ap >x> (V \cup W) \supseteq (ap >x> V) \cup (ap >x> W)$.

2.3 Characterization of Symmetric Composition

The goal of this section is to show that $[\![f \mid g]\!] = [\![f]\!] \mid [\![g]\!]$. We prove this result in two parts: $[\![f \mid g]\!] \subseteq [\![f]\!] \mid [\![g]\!]$ and $[\![f]\!] \mid [\![g]\!] \subseteq [\![f \mid g]\!]$. For the proof we employ the operational semantics of $\mid$, from Section 1.2, page 2, and the meaning function given in Section 2.2.1, page 36.

We note some preliminary facts.

1. $a \simeq b \equiv b \simeq a$.
2. $a \simeq b \Rightarrow \neg(a \preceq b)$, $a \simeq b \Rightarrow \neg(b \preceq a)$.
3. It is possible for $a \simeq b$, $a \preceq b$ and $b \preceq a$ to be all false at the same time.

Lemma 7 $(u \mid v)_t = u_t \mid v_t$.

Proof: Apply the definition of $\mid$ to both sides. Note that $a_t \simeq b_t \equiv a \simeq b$, $a_t \preceq b_t \equiv a \preceq b$. The result follows by applying induction.

2.3.1 $[\![f \mid g]\!] \subseteq [\![f]\!] \mid [\![g]\!]$

Theorem 11 $[\![f \mid g]\!] \subseteq [\![f]\!] \mid [\![g]\!]$

Proof: Given $p \in [\![f \mid g]\!]$, we show that $p \in [\![f]\!] \mid [\![g]\!]$. Proof is by induction on the length of p.

• $p = \epsilon$: then, $p \in \{\epsilon\} \mid \{\epsilon\} \subseteq [\![f]\!] \mid [\![g]\!]$, since $\{\epsilon\} \subseteq [\![f]\!]$, and $\{\epsilon\} \subseteq [\![g]\!]$.

• $p = aq_t$, where a is base: Given $p \in [\![f \mid g]\!]$, without loss in generality, assume that $f \overset{a}{\rightarrow} f'$ and $f' \mid g^t \overset{q}{\Rightarrow}$.

Case 1) $q = \epsilon$: Since $g^t \neq \perp$, there is some $y \in [\![^t]\!]$, that is, $y_t \in [\![g]\!]$.

$$
\begin{array}{ll}
 & aq_t \\
\in & \{q = \epsilon\} \\
 & a(\epsilon \mid y_t) \\
\subseteq & \{a \text{ is a base event at time } t\text{; use definition of } \mid\}
\end{array}
$$

$$
\begin{array}{ll}
 & a\epsilon \mid y_t \\
= & \{a\epsilon = a\} \\
 & a \mid y_t \\
\subseteq & \{\text{from } f \overset{a}{\rightarrow},\ a \in [\![f]\!];\ \text{we have } y_t \in [\![g]\!].\} \\
 & [\![f \mid g]\!]
\end{array}
$$

Case 2) $q \neq \epsilon$: From $f' \mid g^t \overset{q}{\Rightarrow}$, inductively, $q \in x \mid y$, where $x \in [\![f']\!]$ and $y \in [\![g^t]\!]$. Since $q \neq \epsilon$, we get $y \neq \epsilon$.

$$
\begin{array}{ll}
 & aq_t \\
\in & \{q \in x \mid y\} \\
 & a(x \mid y)_t \\
= & \{\text{distribute time-shift}\} \\
 & a(x_t \mid y_t) \\
\subseteq & \{a.time = t \leq (y_t).time, \text{ and } a \text{ is base}\} \\
 & ax_t \mid y_t \\
\subseteq & \{f \overset{a}{\rightarrow} f' \overset{x}{\Rightarrow},\ a.time = t;\ \text{so } f \overset{ax_t}{\Rightarrow},\ \text{or } ax_t \in [\![f]\!] \\
 & \ \ y \in [\![g^t]\!],\ \text{so, } y_t \in [\![g]\!]\} \\
 & [\![f]\!] \mid [\![g]\!]
\end{array}
$$

• $p = aq_t$, where a is a substitution event: Since $aq_t \in [\![f \mid g]\!]$, from the operational semantics, $f \overset{a}{\rightarrow} f'$, $g \overset{a}{\rightarrow} g'$, and $f' \mid g' \overset{q}{\Rightarrow}$. Inductively, from $f' \mid g' \overset{q}{\Rightarrow}$, there exists x and y, where $x \in [\![f']\!]$, $y \in [\![g']\!]$ and $q \in x \mid y$.

$$
\begin{array}{ll}
 & q \in x \mid y \\
\Rightarrow & \{\text{simple algebra}\} \\
 & aq_t \in a(x \mid y)_t \\
\Rightarrow & \{a(x \mid y)_t = a(x_t \mid y_t)\} \\
 & aq_t \in a(x_t \mid y_t) \\
\Rightarrow & \{\text{since } a \text{ is a substitution event, } a(x_t \mid y_t) = ax_t \mid ay_t\} \\
 & aq_t \in (ax_t \mid ay_t) \\
\Rightarrow & \{f \overset{a}{\rightarrow} f' \overset{x}{\Rightarrow};\ \text{so, } ax_t \in [\![f]\!].\ \text{Similarly, } ay_t \in [\![g]\!]\} \\
 & aq_t \in [\![f]\!] \mid [\![g]\!]
\end{array}
$$

2.3.2 $[\![f]\!] \mid [\![g]\!] \subseteq [\![f \mid g]\!]$

Theorem 12 $[\![f]\!] \mid [\![g]\!] \subseteq [\![f \mid g]\!]$

Proof: Let $p \in [\![f]\!] \mid [\![g]\!]$; we show that $p \in [\![f \mid g]\!]$. Proof is by induction on the length of p.

• $p = \epsilon$: The execution set of every expression, hence $[\![f \mid g]\!]$, contains ϵ.

• $p = aq_t$, where a is a base event:

$$
\begin{array}{ll}
 & aq_t \in [\![f]\!] \mid [\![g]\!] \\
\Rightarrow & \{\text{assume that } a \text{ is an event from } [\![f]\!];\ \text{definition of } \mid\}
\end{array}
$$

$$
\begin{array}{ll}
& aq_t \in au_t \mid v_t, \text{ where } au_t \in [\![f]\!], v_t \in [\![g]\!] \\
\Rightarrow & \{a \text{ is a base event; so, } au_t \mid v_t = a(u_t \mid v_t)\} \\
& aq_t \in a(u_t \mid v_t), \text{ where } au_t \in [\![f]\!], v_t \in [\![g]\!] \\
\Rightarrow & \{u_t \mid v_t = (u \mid v)_t\} \\
& q \in u \mid v, \text{ and } au_t \in [\![f]\!], v_t \in [\![g]\!] \\
\Rightarrow & \{au_t \in [\![f]\!] \text{ means } f \xrightarrow{a} f' \stackrel{u}{\Rightarrow}, \text{ for some } f', \\
& \quad v_t \in [\![g]\!] \text{ means } g^t \stackrel{v}{\Rightarrow} \} \\
& q \in [\![f']\!] \mid [\![g^t]\!] \\
\Rightarrow & \{\text{induction}\} \\
& q \in [\![f' \mid g^t]\!] \\
\Rightarrow & \{f \xrightarrow{a} f' \text{ implies } f \mid g \xrightarrow{a} f' \mid g^t \text{ and } f' \mid g^t \stackrel{q}{\Rightarrow} \} \\
& aq_t \in [\![f \mid g]\!]
\end{array}
$$

- $p = aq_t$, where a is a substitution event:

$$
\begin{array}{ll}
& aq_t \in [\![f]\!] \mid [\![g]\!] \\
\Rightarrow & \{a \text{ is a substitution event, use definition of merge}\} \\
& au_t \in [\![f]\!], av_t \in [\![g]\!], \text{ where } q \in u \mid v \\
\Rightarrow & \{\text{rewriting}\} \\
& f \xrightarrow{a} f' \stackrel{u}{\Rightarrow}, \text{ for some } f', \text{ and } g \xrightarrow{a} g' \stackrel{v}{\Rightarrow}, \text{ for some } g' \\
\Rightarrow & \{q \in u \mid v\} \\
& q \in [\![f']\!] \mid [\![g']\!] \\
\Rightarrow & \{f \xrightarrow{a} f' \text{ and } g \xrightarrow{a} g' \text{ implies } f \mid g \xrightarrow{a} f' \mid g'\} \\
& f \mid g \xrightarrow{a} f' \mid g' \stackrel{q}{\Rightarrow} \\
\Rightarrow & \{\text{rewriting}\} \\
& aq_t \in [\![f \mid g]\!]
\end{array}
$$

2.4 Characterization of Sequential Composition

The goal of this section is to show that $[\![f >x> g]\!] = [\![f]\!] >x> [\![g]\!]$. We prove this result in two parts: $[\![f >x> g]\!] \subseteq [\![f]\!] >x> [\![g]\!]$ and $[\![f >x> g]\!] \subseteq [\![f]\!] >x> [\![g]\!]$. For the proof we employ the operational semantics of $>x>$, from Section 1.2, page 2, and the meaning function given in Section 2.2.2, page 37.

2.4.1 Preliminary Results

Lemma 8 For any p and V, $p_t >x> V = (p >x> V)_t$

Proof: If $V = \phi$, the result is trivial. Assume $V \neq \phi$. Proof is by induction on the length of p.

- $\epsilon_t >x> V = (\epsilon >x> V)_t$:

$\epsilon_t >x> V = \epsilon >x> V = \{\epsilon\}$ and $(\epsilon >x> V)_t = \{\epsilon\}_t = \{\epsilon\}$.

- $(ap)_t >x> V = (ap >x> V)_t$:

The proof is similar for the first two cases in the definition (i.e., when a is not a publication). So, we show just one proof, for the first case in the definition where $c_1(a)$ holds.

$$
\begin{array}{ll}
 & (ap)_t >x> V \\
= & \{(ap)_t = a_t p_t\} \\
 & a_t p_t >x> V \\
= & \{\text{definition; note that } c_1(a) \text{ holds, so } c_1(a_t) \text{ holds}\} \\
 & a_t(p_t >x> V) \\
= & \{\text{induction: } p_t >x> V = (p >x> V)_t\} \\
 & a_t(p >x> V)_t \\
= & \{\text{time-shift of } t \text{ distributes over concatenation}\} \\
 & (a(p >x> V))_t \\
= & \{\text{definition: } ap >x> V = a(p >x> V)\text{, given } c_1(a)\} \\
 & (ap >x> V)_t
\end{array}
$$

For the last case in the definition, let a be a publication at time s.

$$
\begin{array}{ll}
 & (ap)_t >x> V \\
= & \{(ap)_t = a_t p_t\} \\
 & a_t p_t >x> V \\
= & \{\text{definition; note that } a \text{ is a publication at } s; \\
 & \quad \text{so } c_3(a) \text{ and } c_3(a_t) \text{ hold}\} \\
 & (s+t,\tau)(p_t >x> V \mid (V'')_{s+t}) \\
= & \{\text{induction: } p_t >x> V = (p >x> V)_t\} \\
 & (s+t,\tau)((p >x> V)_t \mid (V'')_{s+t}) \\
= & \{\text{time-shift of } t \text{ distributes over merge}\} \\
 & ((s+t,\tau)(p >x> V \mid V''_s)_t) \\
= & \{\text{move time-shift over concatenation}\} \\
 & ((s,\tau)(p >x> V \mid V''_s))_t \\
= & \{\text{from definition, } ap >x> V = (s,\tau)(p >x> V \mid V''_s)\text{, given } c_3(a)\} \\
 & (ap >x> V)_t
\end{array}
$$

Observation 10 For $y \neq x$, $f >x> g \overset{(t,[m/y])}{\longrightarrow} f' >x> g'$, where $f \overset{(t,[m/y])}{\longrightarrow} f'$ and $g \overset{(0,[m/y])}{\longrightarrow} g'$

Proof:

$$
\begin{array}{ll}
 & f >x> g \\
\overset{(t,[m/y])}{\longrightarrow} & \{\text{definition of substitution}\} \\
 & [m/y].(f >x> g)^t \\
= & \{\text{definition of } (f >x> g)^t\} \\
 & [m/y].(f^t >x> g) \\
= & \{\text{substitution rules, note that } y \neq x\} \\
 & [m/y].(f^t) >x> [m/y].g \\
= & \{f \overset{(t,[m/y])}{\longrightarrow} [m/y].(f^t) = f',\ g \overset{(0,[m/y])}{\longrightarrow} [m/y].g = g'\} \\
 & f' >x> g'
\end{array}
$$

Observation 11 $f >x> g \overset{(t,[m/x])}{\rightarrow} f' >x> g$, where $f \overset{(t,[m/x])}{\rightarrow} f'$.

Proof:

$$
\begin{array}{cl}
 & f >x> g \\
\overset{(t,[m/x])}{\rightarrow} & \{\text{definition of substitution}\} \\
 & [m/x].(f >x> g)^t \\
= & \{\text{definition of } (f >x> g)^t\} \\
 & [m/x].(f^t >x> g) \\
= & \{\text{substitution rules}\} \\
 & [m/x].(f^t) >x> g \\
= & \{f \overset{(t,[m/x])}{\rightarrow} [m/x].(f^t) = f'\} \\
 & f' >x> g
\end{array}
$$

2.4.2 $[\![f >x> g]\!] \subseteq [\![f]\!] >x> [\![g]\!]$

Theorem 13 $[\![f >x> g]\!] \subseteq [\![f]\!] >x> [\![g]\!]$

Proof: Given $p \in [\![f >x> g]\!]$, we show that $p \in [\![f]\!] >x> [\![g]\!]$. Proof is by induction on the length of p.

- $p = \epsilon$: We have $\epsilon \in [\![f]\!]$ and $\{\epsilon\} \subseteq [\![g]\!]$. Therefore, $\{\epsilon\} = \epsilon >x> \{\epsilon\} \subseteq [\![f]\!] >x> [\![g]\!]$.

- $p = aq_t$, where a is an other-substitution:

$$
\begin{array}{cl}
 & aq_t \in [\![f >x> g]\!] \\
\Rightarrow & \{\text{operational semantics of } [\![f >x> g]\!]\} \\
 & f >x> g \overset{a}{\rightarrow} f' >x> g' \overset{q}{\Rightarrow} \\
\Rightarrow & \{\text{from Observation 10, page 46, } f \overset{a}{\rightarrow} f',\ g \overset{(0,b)}{\rightarrow} g', \text{ where } a = (t,b)\} \\
 & f \overset{a}{\rightarrow} f',\ g \overset{(0,b)}{\rightarrow} g', \text{ where } a = (t,b) \text{ and } f' >x> g' \overset{q}{\Rightarrow} \\
\Rightarrow & \{\text{induction on } q \in [\![f' >x> g']\!]\} \\
 & q \in [\![f']\!] >x> [\![g']\!] \\
\Rightarrow & \{\text{rewriting}\} \\
 & aq_t \in a([\![f']\!] >x> [\![g']\!])_t \\
\Rightarrow & \{\text{see sublemma below}\} \\
 & aq_t \in (a[\![f']\!]_t) >x> [\![g]\!] \\
\Rightarrow & \{f \overset{a}{\rightarrow} f'; \text{ from Observation 6, page 8, } a[\![f']\!]_t \subseteq [\![f]\!]\} \\
 & aq_t \in [\![f]\!] >x> [\![g]\!]
\end{array}
$$

Sublemma: We show that $a([\![f']\!] >x> [\![g']\!])_t = (a[\![f']\!]_t) >x> [\![g]\!]$, given $g \overset{(0,b)}{\rightarrow} g'$, and $a = (t,b)$.

$$
\begin{array}{cl}
 & (a[\![f']\!]_t) >x> [\![g]\!] \\
= & \{\text{from definition (SCD2), since } c_2(a)\}
\end{array}
$$

$$
\begin{array}{ll}
 & a(\llbracket f' \rrbracket_t >x> (\llbracket g \rrbracket \backslash (0,b))) \\
= & \{g \overset{(0,b)}{\to} g'. \text{ So, } \llbracket g \rrbracket \backslash (0,b) = \llbracket g' \rrbracket, \text{ from Observation 6, page 8}\} \\
 & a(\llbracket f' \rrbracket_t >x> \llbracket g' \rrbracket) \\
\Rightarrow & \{\text{from Lemma 8, page 45, } \llbracket f' \rrbracket_t >x> \llbracket g' \rrbracket = (\llbracket f' \rrbracket >x> \llbracket g' \rrbracket)_t\} \\
 & \\
= & \{g \overset{(0,b)}{\to} g'. \text{ So, } \llbracket g \rrbracket \backslash (0,b) = \llbracket g' \rrbracket\} \\
 & a(\llbracket f' \rrbracket >x> \llbracket g' \rrbracket)_t
\end{array}
$$

- $p = aq_t$, where a is an own-substitution:

$$
\begin{array}{ll}
 & aq_t \in \llbracket f >x> g \rrbracket \\
\Rightarrow & \{\text{from Observation 11, page 47}\} \\
 & f >x> g \overset{a}{\to} f' >x> g \overset{q}{\Rightarrow}, \text{ where } f \overset{a}{\to} f' \\
\Rightarrow & \{\text{induction on } q \in \llbracket\, ' >x> g \rrbracket\} \\
 & q \in \llbracket f' \rrbracket >x> \llbracket g \rrbracket \\
\Rightarrow & \{\text{rewriting}\} \\
 & aq_t \in a(\llbracket f' \rrbracket >x> \llbracket g \rrbracket)_t \\
\Rightarrow & \{\text{from Lemma 8, page 45, } (\llbracket f' \rrbracket >x> \llbracket g \rrbracket)_t = \llbracket f' \rrbracket_t >x> \llbracket g \rrbracket\} \\
 & aq_t \in a(\llbracket f' \rrbracket_t >x> \llbracket g \rrbracket) \\
= & \{\text{from definition (SCD1), since } c_1(a)\} \\
 & aq_t \in (a\llbracket f' \rrbracket_t) >x> \llbracket g \rrbracket \\
\Rightarrow & \{f \overset{a}{\to} f'; \text{ so, } a\llbracket f' \rrbracket_t \subseteq \llbracket f \rrbracket\} \\
 & aq_t \in \llbracket f \rrbracket >x> \llbracket g \rrbracket
\end{array}
$$

- $p = aq_t$, where $a \neq \tau$ and a is not a substitution:

Since $aq_t \in \llbracket f >x> g \rrbracket$, a can not be a publication. Hence, $c_1(a)$ holds. Also, $a \neq \tau$ means

$$f \overset{a}{\to} f', \text{ and } f >x> g \overset{a}{\to} f' >x> g \overset{q}{\Rightarrow}$$

The proof is similar to the case where a is an own-substitution.

- $p = aq_t$, where $a = (t, \tau)$:

If a is not due to a publication, that is, $f \overset{a}{\to} f'$ so that $f >x> g \overset{a}{\to} f' >x> g$, the proof is similar to the case where a is an own-substitution.

If a is due to a publication $(t, !m)$, i.e., $f \overset{(t,!m)}{\to} f'$,

$$
\begin{array}{ll}
 & f \overset{(t,!m)}{\to} f' \\
\Rightarrow & \{\text{operational semantics}\} \\
 & f >x> g \overset{(t,\tau)}{\to} f' >x> g \mid [m/x].g \\
\Rightarrow & \{p = aq_t \in \llbracket f >x> g \rrbracket \text{ and } a = (t,\tau)\} \\
 & q \in \llbracket f' >x> g \mid [m/x].g \rrbracket \\
\Rightarrow & \{\text{from theorems Theorem 11, page 43 and Theorem 12, page 44:} \\
 & \quad \llbracket f' >x> g \mid [m/x].g \rrbracket = \llbracket f' >x> g \rrbracket \mid \llbracket [m/x].g \rrbracket\}
\end{array}
$$

$$
\begin{array}{ll}
& q \in [\![f' \mathbin{>x>} g]\!] \mid [\![[m/x].g]\!] \\
\Rightarrow & \{\text{rewriting}\} \\
& q \in u \mid v, \text{ where } u \in [\![f' \mathbin{>x>} g]\!], \text{ and } v \in [\![[m/x].g]\!] \\
\Rightarrow & \{\text{induction on } u \in [\![f' \mathbin{>x>} g]\!]\} \\
& u \in [\![f']\!] \mathbin{>x>} [\![g]\!],\ q \in u \mid v,\ v \in [\![[m/x].g]\!] \\
\Rightarrow & \{q \in u \mid v. \text{ So } q_t \in (u \mid v)_t = u_t \mid v_t\} \\
& aq_t \in a(u_t \mid v_t),\ v \in [\![[m/x].g]\!] \\
\Rightarrow & \{\text{from Observation 6, page 8, } [\![[m/x].g]\!] = [\![g]\!]\backslash(0, [m/x])\} \\
& aq_t \in a(u_t \mid v_t),\ v \in ([\![g]\!]\backslash(0, [m/x])) \\
\Rightarrow & \{a = (t, \tau),\ u \in [\![f']\!] \mathbin{>x>} [\![g]\!]\} \\
& aq_t \in (t, \tau)(([\![f']\!] \mathbin{>x>} [\![g]\!])_t \mid ([\![g]\!]\backslash(0, [m/x]))_t) \\
\Rightarrow & \{\text{from Lemma 8, page 45, } ([\![f']\!] \mathbin{>x>} [\![g]\!])_t = ([\![f']\!]_t \mathbin{>x>} [\![g]\!])\} \\
& aq_t \in (t, \tau)(([\![f']\!]_t \mathbin{>x>} [\![g]\!]) \mid ([\![g]\!]\backslash(0, [m/x]))_t) \\
\Rightarrow & \{\text{from definition (SCD3) given } c_3(a)\} \\
& aq_t \in ((t, !m)[\![f']\!]_t) \mathbin{>x>} [\![g]\!] \\
\Rightarrow & \{\text{from } f \overset{(t,!m)}{\rightarrow} f',\ (t, !m)[\![f']\!]_t \subseteq [\![f]\!]\} \\
& aq_t \in [\![f]\!] \mathbin{>x>} [\![g]\!]
\end{array}
$$

2.4.3 $[\![f]\!] \mathbin{>x>} [\![g]\!] \subseteq [\![f \mathbin{>x>} g]\!]$

Theorem 14 $[\![f]\!] \mathbin{>x>} [\![g]\!] \subseteq [\![f \mathbin{>x>} g]\!]$

Proof: For $p \in [\![f]\!] \mathbin{>x>} [\![g]\!]$, we show $p \in [\![f \mathbin{>x>} g]\!]$. If $p = \epsilon$, the result follows from $\epsilon \in [\![f \mathbin{>x>} g]\!]$.

Let $p = aq_t$. From definition (SCD1–SCD3), a can not be a publication. We consider four cases: (1) a is an own-substitution, (2) a is an other-substitution, and (3) a is not a substitution and $a \neq (t, \tau)$, for any t, and (4) $a = (t, \tau)$, for some t.

Case 1) a is an own-substitution: So, $c_1(a)$ holds.

$$
\begin{array}{ll}
& aq_t \in [\![f]\!] \mathbin{>x>} [\![g]\!] \\
\Rightarrow & \{\text{from definition (SCD1), since } c_1(a)\} \\
& aq_t \in a(r_t \mathbin{>x>} [\![g]\!]), \text{ where } ar_t \in [\![f]\!] \\
\Rightarrow & \{\text{simplify and rewrite}\} \\
& q_t \in r_t \mathbin{>x>} [\![g]\!], \text{ and } f \overset{a}{\rightarrow} f' \overset{r}{\Rightarrow} \\
\Rightarrow & \{\text{from Lemma 8, page 45, } r_t \mathbin{>x>} [\![g]\!] = (r \mathbin{>x>} [\![g]\!])_t\} \\
& q_t \in (r \mathbin{>x>} [\![g]\!])_t, \text{ and } f' \overset{r}{\Rightarrow} \\
\Rightarrow & \{\text{simplify}\} \\
& q \in r \mathbin{>x>} [\![g]\!], \text{ and } f' \overset{r}{\Rightarrow} \\
\Rightarrow & \{r \in [\![f']\!]\} \\
& q \in ([\![f']\!] \mathbin{>x>} [\![g]\!]) \\
\Rightarrow & \{\text{induction}\} \\
& q \in [\![f' \mathbin{>x>} g]\!] \\
\Rightarrow & \{\text{from operational semantics and Observation 11, page 47} \\
& \quad f \overset{a}{\rightarrow} f' \text{ and } c_1(a) \text{ implies } f \mathbin{>x>} g \overset{a}{\rightarrow} f' \mathbin{>x>} g\}
\end{array}
$$

$$f >x> g \overset{a}{\rightarrow} f' >x> g \overset{q}{\Rightarrow}$$

$\Rightarrow$ {rewrite}

$$aq_t \in [\![f >x> g]\!]$$

Case 2) a is an other-substitution: So, $c_2(a)$ holds. Let $a = (t, b)$ and $a_0 = (0, b)$.

$$aq_t \in [\![f]\!] >x> [\![g]\!]$$

$\Rightarrow$ {from definition (SCD2), since $c_2(a)$}

$aq_t \in a(r_t >x> ([\![g]\!] \backslash a_0))$, where $ar_t \in [\![f]\!]$

$\Rightarrow$ {Let $g \overset{a_0}{\Rightarrow} g'$; then, from Observation 6, page 8, $([\![g]\!] \backslash a_0) = [\![g']\!]$}

$aq_t \in a(r_t >x> [\![g']\!])$, where $ar_t \in [\![f]\!]$ and $g \overset{a_0}{\Rightarrow} g'$

$\Rightarrow$ {simplify and rewrite}

$q_t \in r_t >x> [\![g']\!]$, $f \overset{a}{\rightarrow} f' \overset{r}{\Rightarrow}$ and $g \overset{a_0}{\Rightarrow} g'$

$\Rightarrow$ {$r_t >x> [\![g']\!] = (r >x> [\![g']\!])_t$, from Lemma 8, page 45}

$q \in r >x> [\![g']\!]$, $f \overset{a}{\rightarrow} f' \overset{r}{\Rightarrow}$, and $g \overset{a_0}{\Rightarrow} g'$

$\Rightarrow$ {$r \in [\![f']\!]$}

$q \in [\![f']\!] >x> [\![g']\!]$, $f \overset{a}{\rightarrow} f' \overset{r}{\Rightarrow}$, and $g \overset{a_0}{\Rightarrow} g'$

$\Rightarrow$ {induction on $q \in [\![f']\!] >x> [\![g']\!]$}

$q \in [\![f' >x> g']\!]$, $f \overset{a}{\rightarrow} f' \overset{r}{\Rightarrow}$, and $g \overset{a_0}{\Rightarrow} g'$

$\Rightarrow$ {from Observation 10, page 46, $f >x> g \overset{a}{\rightarrow} f' >x> g'$}

$$f >x> g \overset{a}{\rightarrow} f' >x> g' \overset{q}{\Rightarrow}$$

$\Rightarrow$ {rewrite}

$$aq_t \in [\![f >x> g]\!]$$

Case 3) a is not a substitution and $a \neq (t, \tau)$, for any t: So, $c_1(a)$ holds and the proof is similar to that for Case (1).

Case 4) $a = (t, \tau)$, for some t: Then, the third case in the definition, (SCD3), applies.

$$aq_t \in [\![f]\!] >x> [\![g]\!]$$

$\Rightarrow$ {$a = (t, \tau)$}

$$(t, \tau)q_t \in [\![f]\!] >x> [\![g]\!]$$

$\Rightarrow$ {from definition (SCD3)}

$(t, \tau)q_t \in (t, \tau)(r_t >x> [\![g]\!] \mid ([\![g]\!] \backslash (0, [m/x]))_t)$, where $f \overset{(t,!m)}{\rightarrow} f' \overset{r}{\Rightarrow}$, for some m

$\Rightarrow$ {let $g \overset{(0,[m/x])}{\rightarrow} g''$; from Observation 6, page 8, $[\![g]\!] \backslash (0, [m/x]) = [\![g'']\!]$}

$q_t \in (r_t >x> [\![g]\!] \mid [\![g'']\!]_t)$, $f \overset{(t,!m)}{\rightarrow} f' \overset{r}{\Rightarrow}$, $g \overset{(0,[m/x])}{\rightarrow} g''$

$\Rightarrow$ {from Lemma 8, page 45, $r_t >x> [\![g]\!] = (r >x> [\![g]\!])_t$; simplify}

$q \in (r >x> [\![g]\!] \mid [\![g'']\!])$, $f \overset{(t,!m)}{\rightarrow} f' \overset{r}{\Rightarrow}$, $g \overset{(0,[m/x])}{\rightarrow} g''$

$\Rightarrow$ {$r \in [\![f']\!]$}

$q \in [\![f']\!] >x> [\![g]\!] \mid [\![g'']\!]$, $f \overset{(t,!m)}{\rightarrow} f' \overset{r}{\Rightarrow}$ and $g \overset{(0,[m/x])}{\rightarrow} g''$

$\Rightarrow$ {definition of symmetric composition}

$q \in u \mid v$, where $u \in [\![f']\!] >x> [\![g]\!]$ and $v \in [\![g'']\!]$

$\Rightarrow$ {induction on $u \in [\![f']\!] >x> [\![g]\!]$}

$$
\begin{array}{ll}
 & q \in u \mid v, \text{ where } u \in [\![f' >x> g]\!] \text{ and } v \in [\![g'']\!] \\
\Rightarrow & \{\text{rewrite}\} \\
 & q \in [\![f' >x> g]\!] \mid [\![g'']\!] \\
\Rightarrow & \{\text{theorems on Merge: } [\![f' >x> g]\!] \mid [\![g'']\!] = [\![f' >x> g \mid g'']\!]\} \\
 & q \in [\![f' >x> g \mid g'']\!] \\
\Rightarrow & \{\text{given } f \overset{(t,!m)}{\rightarrow} f'\} \\
 & f >x> g \overset{(t,\tau)}{\rightarrow} f' >x> g \mid [m/x].g \\
\Rightarrow & \{[m/x].g = g''\} \\
 & f >x> g \overset{(t,\tau)}{\rightarrow} f' >x> g \mid g'' \\
\Rightarrow & \{q \in [\![f' >x> g \mid g'']\!]\} \\
 & f >x> g \overset{(t,\tau)}{\rightarrow} f' >x> g \mid g'' \overset{q}{\Rightarrow} \\
\Rightarrow & \{\text{rewrite}\} \\
 & (t,\tau)q_t \in [\![f >x> g]\!]
\end{array}
$$

2.5 Characterization of Asymmetric Composition

The goal of this section is to show that $[\![f <x< g]\!] = [\![f]\!] <x< [\![g]\!]$. We prove this result in two parts: $[\![f <x< g]\!] \subseteq [\![f]\!] <x< [\![g]\!]$ and $[\![f <x< g]\!] \subseteq [\![f]\!] <x< [\![g]\!]$. For the proof we employ the operational semantics of $<x<$, from Section 1.2, page 2, and the meaning function given in Section 2.2.3, page 37.

2.5.1 Preliminary Results

We often write ap_t to mean that a is an event at time t, (and p_t is the execution following a). It is possible for p to be ϵ.

Observation 12 $f <x< g \overset{p}{\Rightarrow}$ implies $f <x< g \overset{pc}{\Rightarrow}$, where c is an other-substitution, and $c.time = p.time$.

Proof: From the operational semantics.

Observation 13 $(ap_t).time = t + p.time$

Proof: For $p = \epsilon$, $(ap_t).time = t = t + p.time$.
$p \neq \epsilon$, $(ap_t).time = t + p.time$.

Observation 14 $(ap_t).time - (aq_t).time = p.time - q.time$.
$(ap_t).time - (q_t).time = p.time - q.time$, provided $q \neq \epsilon$.

Proof: The first part follows directly from Observation 13, page 51. The second part follows similarly, by noting that $(q_t).time = t + q.time$, for $q \neq \epsilon$.

Observation 15 Let

$u.time = t + u'.time$,
$v.time = t + v'.time$,
$T = \max(u.time, v.time)$,
$T' = \max(u'.time, v'.time)$.

Then, $T - u.time = T' - u'.time$ and $T - v.time = T' - v'.time$.

Proof: The two proofs are similar; we prove only the first one.

$$\begin{array}{cl}
 & T \\
= & \{\text{definition}\} \\
 & \max(u.time, v.time) \\
= & \{u.time = t + u'.time,\ v.time = t + v'.time\} \\
 & \max(t + u'.time, t + v'.time) \\
= & \{\text{arithmetic}\} \\
 & t + \max(u'.time, v'.time) \\
= & \{T' = \max(u'.time, v'.time)\} \\
 & t + T' \\
= & \{u.time = t + u'.time, \text{ so, } t = u.time - u'.time\} \\
 & u.time - u'.time + T'
\end{array}$$

Therefore, $T - u.time = T' - u'.time$.

Observation 16 $\epsilon \in u +_x v \ \equiv\ u = \epsilon \wedge v = \epsilon$

Proof: $\epsilon \in u +_x v$ by application of the base rule only, because the inductive rule creates items starting with an item. The result follows by considering the base rule.

Lemma 9 $p \in u +_x v \ \Rightarrow\ d_0(u, v) \ \wedge\ p.time = \max(u.time, v.time)$.

Proof: Proof is by induction on the combined lengths of u and v.

$u = \epsilon$ and $v = \epsilon$: then, $d_0(u, v)$. And, $p = \epsilon$, so, $p.time = 0 = \max(u.time, v.time)$.

$u = \epsilon$ and $v \neq \epsilon$: Since $u +_x v \neq \phi$, v contains no other-substitution. So, $d_0(u, v)$. And, $p = v$, so, $p.time = v.time = \max(u.time, v.time)$.

$u \neq \epsilon$ and $v = \epsilon$: Since $u +_x v \neq \phi$, u contains no substitution. So, $d_0(u, v)$. And, $p = u$, so, $p.time = u.time = \max(u.time, v.time)$.

$u \neq \epsilon$ and $v \neq \epsilon$: We rename the terms and consider $cp \in au +_x bv$. We will show that $d_0(au, bv)$ and $(cp).time = \max(au.time, bv.time)$.

Case 1) $a \approx_x b$, $a = b = c$ and $p \in u +_x v$: Inductively, $d_0(u, v)$; so, $d_0(au, bv)$. Let $a.time = b.time = c.time = t$. And, $u = u'_t$, $v = v'_t$, $p = p'_t$. From $p \in u +_x v$, we get $p'_t \in u'_t +_x v'_t$, or $p'_t \in (u' +_x v')_t$, or $p' \in u' +_x v'$.

$$\begin{array}{cl}
 & \max(au.time, bv.time) \\
= & \{u = u'_t,\ v = v'_t\} \\
 & \max((au'_t).time, (bv'_t).time) \\
= & \{a.time = t,\ b.time = t; \text{ from Observation 13, page 51}\}
\end{array}$$

$$
\begin{array}{ll}
 & \max(t + u'.time, t + v'.time) \\
= & \{\text{arithmetic}\} \\
 & t + \max(u'.time, v'.time) \\
= & \{\text{from } p' \in u' +_x v', \text{ inductively, } p'.time = \max(u'.time, v'.time)\} \\
 & t + p'.time \\
= & \{a.time = t, \text{ from Observation 13, page 51}\} \\
 & (ap'_t).time \\
= & \{p = p'_t,\ a = c\} \\
 & (cp).time
\end{array}
$$

Case 2) $a \preceq b$, $c = a$ and $p \in u +_x bv$: Inductively, $d_0(u, bv)$; so, $d_0(au, bv)$, because a is base, from $a \preceq b$. Let $a.time = c.time = t$. And, $u = u'_t$, $bv = (b'v')_t$, $p = p'_t$. From $p \in u +_x bv$, we get $p'_t \in u'_t +_x (b'v')_t$, or $p'_t \in (u' +_x b'v')_t$, or $p' \in u' +_x b'v'$.

$$
\begin{array}{ll}
 & \max(au.time, bv.time) \\
= & \{u = u'_t,\ bv = (b'v')_t\} \\
 & \max((au'_t).time, (b'v')_t.time) \\
= & \{a.time = t; \text{ from Observation 13, page 51}\} \\
 & \max(t + u'.time, t + (b'v').time) \\
= & \{\text{arithmetic}\} \\
 & t + \max(u'.time, (b'v').time) \\
= & \{\text{from } p' \in u' +_x b'v', \text{ inductively, } p'.time = \max(u'.time, (b'v').time)\} \\
 & t + p'.time \\
= & \{a.time = t, \text{ from Observation 13, page 51}\} \\
 & (ap'_t).time \\
= & \{p = p'_t,\ a = c\} \\
 & (cp).time
\end{array}
$$

Case 3) $b \precneqq_x a$, $c = b$ and $p \in au +_x v$: Similar to Case (2).

The *prefix-closure* of u, written as u^*, is the set of prefixes of u. Formally,

$$
\begin{array}{l}
\epsilon^* = \{\epsilon\}, \\
(au)^* = \{\epsilon\} \cup au^*
\end{array}
$$

Note that $\{\epsilon\} \subseteq u^*$, for all u. Therefore, $(au)^* = \{\epsilon\} \cup au^*$ holds (vacuously) even when $a = \epsilon$. Set U is *prefix-closed* if $u^* \subseteq U$, for every u in U.

Lemma 10 $u|_x v$ is prefix-closed.

Proof: We first observe that $\epsilon \in (u|_x v)$, for any u and v. If either u or v is empty, the result follows from definition. For $au|_x bv$, $\neg((a \simeq b) \wedge (a \preceq b))$; so, at least one of these conditions is *false*, and the corresponding guarded set contributes $\{\epsilon\}$.

The proof of prefix-closure is by induction on the combined lengths of u and v. For empty u or empty v, the result is obvious.

$$
\begin{array}{ll}
 & (au|_x bv)^* \\
= & \{\text{definition}\} \\
 & [a \approx_x b \rightarrow a(u|_x v)] \cup [a \preceq b \rightarrow a(u|_x bv)] \cup [b \underline{\prec\!\prec}_x a \rightarrow b(au|_x v)]^* \\
= & \{\text{prefix-closure distributes over set union and guarded sets}\} \\
 & [a \approx_x b \rightarrow (a(u|_x v))^*] \cup [a \preceq b \rightarrow (a(u|_x bv))^*] \cup [b \underline{\prec\!\prec}_x a \rightarrow (b(au|_x v))^*] \\
= & \{\text{expand prefix-closure}\} \\
 & [a \approx_x b \rightarrow \{\epsilon\} \cup a(u|_x v)^*] \cup [a \preceq b \rightarrow \{\epsilon\} \cup a(u|_x bv)^*] \cup [b \underline{\prec\!\prec}_x a \rightarrow \{\epsilon\} \cup b(au|_x v)^*] \\
= & \{\text{guarded set property}\} \\
 & \{\epsilon\} \cup [a \approx_x b \rightarrow a(u|_x v)^*] \cup [a \preceq b \rightarrow a(u|_x bv)^*] \cup [b \underline{\prec\!\prec}_x a \rightarrow b(au|_x v)^*] \\
= & \{\text{induction}\} \\
 & \{\epsilon\} \cup [a \approx_x b \rightarrow a(u|_x v)] \cup [a \preceq b \rightarrow a(u|_x bv)] \cup [b \underline{\prec\!\prec}_x a \rightarrow b(au|_x v)] \\
= & \{\text{definition of } au|_x bv\} \\
 & \{\epsilon\} \cup (au|_x bv) \\
= & \{\epsilon \in (au|_x bv)\} \\
 & au|_x bv
\end{array}
$$

Lemma 11 Suppose $p \in u|_x v$. Then there are prefixes u' of u and v' of v such that

1. $p \in u' +_x v'$,

2. $u'.time \leq v.time$ and $v'.time \leq u.time$

Proof: Proof is by induction on the length of p.

• $p = \epsilon$: Let $u' = v' = \epsilon$. Then, $p \in u' +_x v'$. Also, $u'.time = 0 \leq v.time$ and $v'.time = 0 \leq u.time$.

• $p \neq \epsilon$: We rename the terms to get $cp \in au|_x bv$. We consider the three cases in the inductive definition of $au|_x bv$.

Case 1) $a \approx_x b \approx_x c$ and $p \in u|_x v$:
Inductively, for some prefix u' of u and v' of v, we have $p \in u' +_x v'$, $u'.time \leq v.time$ and $v'.time \leq u.time$. Then, au' is a prefix of au and bv' of bv. From $a \approx_x b \approx_x c$ and $p \in u' +_x v'$, $cp \in au' +_x bv'$. Further, from $u'.time \leq v.time$ and $a = b$, $(au').time \leq (bv).time$. Symmetrically, $(bv').time \leq (au).time$.

Case 2) $a \preceq b$, $c = a$ and $p \in u|_x bv$:
Inductively, for some prefix u' of u and w' of bv, we have $p \in u' +_x w'$, $u'.time \leq (bv).time$ and $w'.time \leq u.time$. We consider two cases, (2.1) $w' = \epsilon$ and (2.2) $w' \neq \epsilon$.

Case 2.1) $w' = \epsilon$: From $p \in u' +_x w'$, $p = u'$. Then, $cp = ap = au' \in (au' +_x w')$, and au' is a prefix of au and w' of bv. Next we show $(au').time \leq (bv).time$ and $w'.time \leq (au).time$. The latter one is trivial since $w' = \epsilon$.

To show $(au').time \leq (bv).time$, consider two cases.

$u' = \epsilon$: $(au').time = a.time$ $\{\text{from } a \preceq b\}$ $\leq b.time \leq (bv.time)$.

$u' \neq \epsilon$: $(au').time = u'.time$ {given} $\leq (bv.time)$.

Case 2.2) $w' \neq \epsilon$: Since w' is a prefix of bv, $w' = bv'$ for some prefix v' of v.

$$\begin{array}{ll} & cp \\ \in & \{c = a,\ p \in u' +_x w' = u' +_x bv'\} \\ & a(u' +_x bv') \\ \subseteq & \{a \preceq b;\ \text{apply definition of } +_x\} \\ & au' +_x bv' \end{array}$$

To show $(au').time \leq (bv).time$, consider two cases.
$u' = \epsilon$: $(au').time = a.time$ {from $a \preceq b$} $\leq b.time \leq (bv.time)$.
$u' \neq \epsilon$: $(au').time = u'.time$ {given} $\leq (bv.time)$.

To show $(bv').time \leq (au).time$:
$(bv').time = w'.time$ {given} $\leq u.time \leq (au).time$.

Case 3) $b \underline{\prec\!\!\prec}_x b$, $c = b$ and $p \in au|_x v$:
Similar to case (2).

Lemma 12 $uc +_x vc \subseteq uc|_x vc$, where c is an other-substitution.

Proof: We prove the result using induction on the combined length of u and v.

• $u = \epsilon$ and $v = \epsilon$: $uc +_x vc = c +_x c = c(\epsilon +_x \epsilon) = \{c\}$, and $uc|_x vc = c|_x c = \{\epsilon, c\}$.

• $u = \epsilon$ and $v = bv'$:

$$\begin{array}{ll} & uc +_x vc \\ = & \{u = \epsilon \text{ and } v = bv'\} \\ & c +_x bv'c \\ = & \{\text{since } c \text{ is an other-substitution, } \neg(c \preceq b);\ \text{from definition of } +_x\} \\ & \langle c \approx_x b \rightarrow c(\epsilon +_x v'c)\rangle \cup \langle b \underline{\prec\!\!\prec}_x c \rightarrow b(c +_x v'c)\rangle \\ = & \{v'c \text{ contains an other-substitution; so, } \epsilon +_x v'c = \phi\} \\ & \langle b \underline{\prec\!\!\prec}_x c \rightarrow b(c +_x v'c)\rangle \\ \subseteq & \{\text{apply induction}\} \\ & \langle b \underline{\prec\!\!\prec}_x c \rightarrow b(c|_x v'c)\rangle \\ \subseteq & \{\text{definitions of the two kinds of guarded sets}\} \\ & [b \underline{\prec\!\!\prec}_x c \rightarrow b(c|_x v'c)] \\ \subseteq & \{\text{definition of } |_x\} \\ & c|_x bv'c \\ = & \{u = \epsilon \text{ and } v = bv'\} \\ & uc|_x vc \end{array}$$

• $v = \epsilon$ and $a = au'$: The proof is similar to the previous case.

$$
\begin{array}{ll}
 & uc +_x vc \\
= & \{v = \epsilon \text{ and } a = au'\} \\
 & au'c +_x c \\
= & \{\text{since } c \text{ is an other-substitution, } \neg(c \preceq\!\!\prec_x a)\text{; from definition of } +_x\} \\
 & \langle a \approx_x c \rightarrow c(u'c +_x \epsilon)\rangle \cup \langle a \preceq c \rightarrow a(u'c +_x c)\rangle \\
= & \{u'c \text{ contains a substitution; so, } u'c +_x \epsilon = \phi\} \\
 & \langle a \preceq c \rightarrow a(u'c +_x c)\rangle \\
\subseteq & \{\text{apply induction}\} \\
 & \langle a \preceq c \rightarrow a(u'c|_x c)\rangle \\
\subseteq & \{\text{definitions of the two kinds of guarded sets}\} \\
 & [a \preceq c \rightarrow a(u'c|_x c)] \\
\subseteq & \{\text{definition of } |_x\} \\
 & au'c|_x c \\
= & \{v = \epsilon \text{ and } a = au'\} \\
 & uc|_x vc
\end{array}
$$

• $u = au'$ and $v = bv'$:

$$
\begin{array}{ll}
 & au'c +_x bv'c \\
= & \{\text{definition of } +_x\} \\
 & \langle a \approx_x b \rightarrow a(u'c +_x v'c)\rangle \cup \langle a \preceq b \rightarrow a(u'c +_x bv'c)\rangle \cup \langle b \preceq\!\!\prec_x a \rightarrow b(au'c +_x v'c)\rangle \\
\subseteq & \{\text{induction}\} \\
 & \langle a \approx_x b \rightarrow a(u'c|_x v'c)\rangle \cup \langle a \preceq b \rightarrow a(u'c|_x bv'c)\rangle \cup \langle b \preceq\!\!\prec_x a \rightarrow b(au'c|_x v'c)\rangle \\
\subseteq & \{\text{definitions of the two kinds of guarded sets}\} \\
 & [a \approx_x b \rightarrow a(u'c|_x v'c)] \cup [a \preceq b \rightarrow a(u'c|_x bv'c)] \cup [b \preceq\!\!\prec_x a \rightarrow b(au'c|_x v'c)] \\
= & \{\text{definition of } |_x\} \\
 & au'c|_x bv'c
\end{array}
$$

Lemma 13 Let c be an other-substitution and $pc \in u +_x v$. Then, $u = u'c$ and $v = v'c$, for some u' and v'.

Proof: Proof is by induction on the length of p.

• $p = \epsilon$: Then, $c \in u +_x v$. This can not be derived by the base rule, because c is an other-substitution. So, $c \in u +_x v$ is derived by the inductive rule. We examine each term in that rule.

$a \approx_x b$: Then, $a = b = c$ and $\epsilon \in u +_x v$. From Observation 16, page 52, $u = \epsilon \wedge v = \epsilon$. Hence, $au = c$ and $bv = c$, as required.

$a \preceq b$: Since a is a base event, $a \neq c$. Therefore, this term can not derive c.

$b \preceq\!\!\prec_x a$ Since b is base or own-substitution, $b \neq c$. Therefore, this term can not derive c.

• $p \neq \epsilon$: Let $pc = dp'c$. As before, the base rule can not be used for deriving pc. In the inductive rule, $p'c$ has to be generated by either $u +_x v$, $au +_x v$ or $u +_x bv$. Inductively, in each case, $p'c \in u'c +_x v'c$, for some u' and v'.

Lemma 14 $(u +_x v)_t = u_t +_x v_t$.

Proof: Apply the definition of $|$ to both sides. Note that $a_t \approx_x b_t \equiv a \approx_x b$, $a_t \preceq b_t \equiv a \preceq b$ and $b_t \precsim_x a_t \equiv b \precsim_x a$. The result follows by applying induction.

2.5.2 $[\![f \ {<}x{<}\ g]\!] \subseteq [\![f]\!] \ {<}x{<}\ [\![g]\!]$

Lemma 15 Let $f \ {<}x{<}\ g \stackrel{q}{\Rightarrow} h$, where the publication rule, (ASYM2V), was *not* used in forming q. Then,

(*1) there exist $f \stackrel{u}{\Rightarrow} f'$ and $g \stackrel{v}{\Rightarrow} g'$, such that

(*2) $d_1(u, v)$,

(*3) $q \in u +_x v$, and

(*4) $h = f'' \ {<}x{<}\ g''$, where $f'' = (f')^{q.time-u.time}$, $g'' = (g')^{q.time-v.time}$

Proof: Proof is by induction on the length of q.

- $q = \epsilon$: Then $f \ {<}x{<}\ g \stackrel{\epsilon}{\Rightarrow} f \ {<}x{<}\ g$. So, $h = f \ {<}x{<}\ g$. Let $u = \epsilon$, $v = \epsilon$, $f' = f$, $g' = g$. Then, $q.time = u.time = v.time = 0$. Now,

1. $f \stackrel{u}{\Rightarrow} f'$ and $g \stackrel{v}{\Rightarrow} g'$, from $f \stackrel{\epsilon}{\rightarrow} f$, $g \stackrel{\epsilon}{\rightarrow} g$
2. $d_1(u, v)$, from $d_1(\epsilon, \epsilon)$,
3. $q \in u +_x v$, from $\epsilon \in \epsilon +_x \epsilon$
4. $h = (f')^{q.time-u.time} \ {<}x{<}\ (g')^{q.time-v.time}$, from $h = f \ {<}x{<}\ g = f^0 \ {<}x{<}\ g^0$

- $q = ap_t$ where a is an other-substitution: Then, $f \stackrel{a}{\rightarrow} f_1$, $g \stackrel{a}{\rightarrow} g_1$ and $f_1 \ {<}x{<}\ g_1 \stackrel{p}{\Rightarrow} h$. Applying induction on $f_1 \ {<}x{<}\ g_1 \stackrel{p}{\Rightarrow} h$, we get

1. there exist $f_1 \stackrel{u'}{\Rightarrow} f_2$, and $g_1 \stackrel{v'}{\Rightarrow} g_2$, such that
2. $d_1(u', v')$,
3. $p \in u' +_x v'$
4. $h = f_3 \ {<}x{<}\ g_3$ where $f_3 = (f_2)^{p.time-u'.time}$, $g_3 = (g_2)^{p.time-v'.time}$

Let $u = au'_t$, $v = av'_t$, $f' = f_2$, $g' = g_2$.
Now, we show the required items under (*).

(*1) $f \stackrel{u}{\Rightarrow} f'$ and $g \stackrel{v}{\Rightarrow} g'$:
$f \stackrel{u}{\Rightarrow} f'$: $f \stackrel{a}{\rightarrow} f_1 \stackrel{u'}{\Rightarrow} f_2 = f'$. Since $u = au'_t$, $f \stackrel{u}{\Rightarrow} f'$.
$g \stackrel{v}{\Rightarrow} g'$: similarly.

(*2) $d_1(u, v)$: From $d_1(u', v')$ and that a is an other-substitution.

(*3) $q \in u +_x v$:

$$
\begin{array}{ll}
 & q \\
= & \{\text{definition of } q\} \\
 & ap_t \\
\in & \{p \in u' +_x v' \text{ implies } p_t \in u'_t +_x v'_t\} \\
 & a(u'_t +_x v'_t) \\
\subseteq & \{a \text{ is an other-substitution; apply definition of } +_x\} \\
 & au'_t +_x av'_t \\
= & \{u = au'_t,\ v = av'_t\} \\
 & u +_x v
\end{array}
$$

(*4) $h = (f')^{q.time-u.time} <x< (g')^{q.time-v.time}$:
We are given
$h = f_3 <x< g_3$ where $f_3 = (f_2)^{p.time-u'.time}$, $g_3 = (g_2)^{p.time-v'.time}$.
Since $f' = f_2$, $g' = g_2$, it is sufficient to show that

$$
\begin{array}{l}
q.time - u.time = p.time - u'.time \\
q.time - v.time = p.time - v'.time
\end{array}
$$

Since $q = ap_t$, $u = au'_t$ and $v = av'_t$, the results follow from Observation 14, page 51.

• $q = ap_t$, where a is own-substitution:
Then, $f <x< g \overset{a}{\rightarrow} f_1 <x< g_1 \overset{p}{\Rightarrow} h$.
From the definition of own-substitution, $f_1 = f^t$ and $g \overset{a}{\rightarrow} g_1$.
Applying induction on $f_1 <x< g_1 \overset{p}{\Rightarrow} h$, we get:

1. there exist $f_1 \overset{u'}{\Rightarrow} f_2$, and $g_1 \overset{v'}{\Rightarrow} g_2$, such that
2. $d_1(u', v')$,
3. $p \in u' +_x v'$
4. $h = f_3 <x< g_3$ where $f_3 = (f_2)^{p.time-u'.time}$, $g_3 = (g_2)^{p.time-v'.time}$

Case 1) Suppose $u' \neq \epsilon$: Let $u = u'_t$, $v = av'_t$, $f' = f_2$, $g' = g_2$. We show the required items under (*).
(*1) $f \overset{u}{\Rightarrow} f'$ and $g \overset{v}{\Rightarrow} g'$:
$f \overset{v}{\Rightarrow} f'$: Given

$$
\begin{array}{ll}
 & f_1 \overset{u'}{\Rightarrow} f_2 \\
\Rightarrow & \{f_1 = f^t\} \\
 & f^t \overset{u'}{\Rightarrow} f_2 \\
\Rightarrow & \{\text{time-shift}\} \\
 & f \overset{u'_t}{\Rightarrow} f_2 \\
\Rightarrow & \{u = u'_t,\ f' = f_2\} \\
 & f \overset{u}{\Rightarrow} f'
\end{array}
$$

$g \overset{v}{\Rightarrow} g'$: Given

$$\begin{aligned}
& g \overset{a}{\rightarrow} g_1 \overset{v'}{\Rightarrow} g_2 \\
\Rightarrow \quad & \{\text{rewriting}\} \\
& g \overset{av'_t}{\Rightarrow} g_2 \\
\Rightarrow \quad & \{av'_t = v,\ g_2 = g'\} \\
& g \overset{v}{\Rightarrow} g'
\end{aligned}$$

(*2) $d_1(u,v)$: Since $u = u'_t$, and u'_t contains no own-substitution, from $d_1(u',v')$, we conclude that u has no own-substitution. And, $v = av'_t$ contains no publication, because v' does not contain any, from $d_1(u',v')$.

(*3) $q \in u +_x v$:

$$\begin{aligned}
& q \\
= \quad & \{\text{definition of } q\} \\
& ap_t \\
\in \quad & \{p \in u' +_x v' \text{ implies } p_t \in u'_t +_x v'_t\} \\
& a(u'_t +_x v'_t) \\
\subseteq \quad & \{a \text{ is an own-substitution at time } t\text{; apply definition of } +_x\} \\
& u'_t +_x av'_t \\
= \quad & \{u = u'_t,\ v = av'_t\} \\
& u +_x v
\end{aligned}$$

(*4) $h = (f')^{q.time - u.time} <x< (g')^{q.time - v.time}$:
We are given
$h = f_3 <x< g_3$ where $f_3 = (f_2)^{p.time - u'.time}$, $g_3 = (g_2)^{p.time - v'.time}$. Since $f' = f_2$, $g' = g_2$, it is sufficient to show that

$$\begin{aligned}
& q.time - u.time = p.time - u'.time \\
& q.time - v.time = p.time - v'.time
\end{aligned}$$

Since $q = ap_t$, $u = u'_t$ where $u' \neq \epsilon$, and $v = av'_t$, the results follow from Observation 14, page 51.

Case 2) Suppose $u' = \epsilon$: From $p \in u' +_x v'$ and $u' = \epsilon$, we conclude that $p = v'$ and v' has no other-substitution. We are given $f_1 \overset{u'=\epsilon}{\Rightarrow} f_2$, so $f_1 = f_2$, and $g_1 \overset{v'=p}{\Rightarrow} g_2$.

Let $u = \epsilon$, $v = av'_t = ap_t = q$, $f' = f$, and $g' = g_2$. We show the required items under (*).

(*1) $f \overset{u}{\Rightarrow} f'$ and $g \overset{v}{\Rightarrow} g'$:
$f \overset{u}{\Rightarrow} f'$: Follows from $f \overset{u=\epsilon}{\Rightarrow} f = f'$
$g \overset{v}{\Rightarrow} g'$: Given

$$\begin{aligned}
& g \overset{a}{\rightarrow} g_1 \overset{v'}{\Rightarrow} g_2 \\
\Rightarrow \quad & \{\text{rewriting}\}
\end{aligned}$$

$$
\begin{array}{ll}
 & g \overset{av'_t}{\Rightarrow} g_2 \\
\Rightarrow & \{av'_t = v,\ g_2 = g'\} \\
 & g \overset{v}{\Rightarrow} g'
\end{array}
$$

(*2) $d_1(u, v)$: Since $u = \epsilon$, it contains no own-substitution. From $d_1(u', v')$, there is no publication in v'; further, a is a substitution; so, v has no publication.

(*3) $q \in u +_x v$: Given

$$
\begin{array}{ll}
 & p = v' \\
\Rightarrow & \{q = ap_t = av'_t = v, \\
 & \quad v \text{ has no other-substitution, because } v' \text{ has none and } a \text{ is own-substitution}\} \\
 & q \in \epsilon +_x v \\
\Rightarrow & \{u = \epsilon\} \\
 & q \in u +_x v
\end{array}
$$

(*4) $h = (f')^{q.time-u.time} <x< (g')^{q.time-v.time}$:
We are given
$h = f_3 <x< g_3$ where $f_3 = (f_2)^{p.time-u'.time}$, $g_3 = (g_2)^{p.time-v'.time}$.
First, we show $f_3 = (f')^{q.time-u.time}$.

$$
\begin{array}{ll}
 & f_3 \\
= & \{\text{given}\} \\
 & (f_2)^{p.time-u'.time} \\
= & \{\text{from } f_1 \overset{\epsilon}{\Rightarrow} f_2,\ f_2 = f_1, \text{ and } f_1 = f^t\} \\
 & (f^t)^{p.time-u'.time} \\
= & \{\text{arithmetic}\} \\
 & f^{t+p.time-u'.time} \\
= & \{q.time = (ap_t).time = t + p.time,\ u.time = u'.time = 0; \\
 & \quad \text{so } t + p.time - u'.time = q.time - u.time\} \\
 & f^{q.time-u.time} \\
= & \{f = f'\} \\
 & (f')^{q.time-u.time}
\end{array}
$$

Next, we show $g_3 = (g')^{q.time-v.time}$.

$$
\begin{array}{ll}
 & g_3 \\
= & \{\text{given}\} \\
 & (g_2)^{p.time-v'.time} \\
= & \{g_2 = g'\} \\
 & (g')^{p.time-v'.time} \\
= & \{p.time - v'.time = 0, \text{ from } p = v'; \text{ also, } q.time - v.time = 0, \text{ from } q = v\} \\
 & (g')^{q.time-v.time}
\end{array}
$$

• $q = ap_t$, where a is base:
Let $g \overset{a}{\rightarrow} g_1$ so that $f <x< g \overset{a}{\rightarrow} f^t <x< g_1 \overset{p}{\Rightarrow} h$. The proof for this case is identical to the last case where a was an own-substitution.

The proof is similar for the case where $f \xrightarrow{a} f_1$ so that $f <x< g \xrightarrow{a} f_1 <x< g^t \overset{p}{\Rightarrow} h$.

Theorem 15 $[\![f <x< g]\!] \subseteq [\![f]\!] <x< [\![g]\!]$

Proof: Let $p \in [\![f <x< g]\!]$. We show $p \in [\![f]\!] <x< [\![g]\!]$. We consider two cases for the execution p of $f <x< g$: (1) rule (ASYM2V) was not used in the execution p, and (2) (ASYM2V) was used.

• (ASYM2V) was not used in the execution p: Since $f <x< g \overset{p}{\Rightarrow}$, from Observation 12, page 51, $f <x< g \overset{pc}{\Rightarrow}$, where c is an other-substitution, and $c.time = p.time$. Then, applying Lemma 15, page 57,

1. there exist $f \overset{u}{\Rightarrow} f'$ and $g \overset{v}{\Rightarrow} g'$, such that
2. $d_1(u, v)$,
3. $pc \in u +_x v$

Now,

$$\begin{array}{ll} & pc \in u +_x v \\ \Rightarrow & \{\text{from Lemma 13, page 56}\} \\ & pc \in u'c +_x v'c, \text{ where } u = u'c \text{ and } v = v'c \\ \Rightarrow & \{\text{from Lemma 12, page 55, } u'c +_x v'c \subseteq u'c|_x v'c = u|_x v\} \\ & pc \in u|_x v \\ \Rightarrow & \{u|_x v \text{ is prefix-closed, from Lemma 10, page 53}\} \\ & p \in u|_x v \\ \Rightarrow & \{\text{given } d_1(u, v),\ u <x< v = u|_x v\} \\ & p \in u <x< v \\ \Rightarrow & \{\text{given } f \overset{u}{\Rightarrow},\ u \in [\![f]\!]; \text{ similarly, } v \in [\![g]\!]\} \\ & p \in [\![f]\!] <x< [\![g]\!] \end{array}$$

• (ASYM2V) was used in the execution p: Then, $p = q(t, \tau)r_t$, where
$f <x< g \overset{q}{\Rightarrow} h \xrightarrow{s,\tau} \overset{r}{\Rightarrow}$, and $t = q.time + s$.
Applying Lemma 15, page 57, on $f <x< g \overset{q}{\Rightarrow} h$,

1. there exist $f \overset{u}{\Rightarrow} f'$ and $g \overset{v}{\Rightarrow} g'$, such that
2. $d_1(u, v)$,
3. $q \in u +_x v$, and
4. $h = f'' <x< g''$, where $f'' = (f')^{q.time-u.time}$, $g'' = (g')^{q.time-v.time}$

Also, $g'' \xrightarrow{s,!m}$, and $[m/x].(f'')^s \overset{r}{\Rightarrow}$.

Let $j = u(t, [m/x])r_t$, and $k = v(t, !m)$. We first show that (1) $j \in [\![f]\!]$, (2) $k \in [\![g]\!]$, and (3) $d_2(j, k)$, from which we have an easy proof of $p \in [\![f]\!] <x< [\![g]\!]$.

(1) $j \in [\![f]\!]$: We show

$$f \overset{u}{\Rightarrow} f' \overset{(t_1,[m/x])}{\rightarrow} [m/x].(f'')^s \overset{r}{\Rightarrow}, \text{ where } t_1 + u.time = t$$

Hence, $j = u(t, [m/x])r_t \in [\![f]\!]$.

To prove the result, we already have $f \overset{u}{\Rightarrow} f'$ and $[m/x].(f'')^s \overset{r}{\Rightarrow}$. So, we need only prove $f' \overset{(t_1,[m/x])}{\rightarrow} [m/x].(f'')^s$.

$$\begin{array}{ll} & f' \\ \overset{(t_1,[m/x])}{\rightarrow} & \{\text{application of substitution}\} \\ & [m/x].(f')^{t_1} \\ = & \{t_1 = t - u.time = \{t = q.time + s\}\ q.time - u.time + s\} \\ & [m/x].(f')^{q.time-u.time+s} \\ = & \{f'' = (f')^{q.time-u.time}\} \\ & [m/x].(f'')^s \end{array}$$

(2) $k \in [\![g]\!]$: we have to show $g \overset{v(t,!m)}{\Rightarrow}$. We show $g \overset{v}{\Rightarrow} g' \overset{t_2,!m}{\rightarrow}$, where $v.time + t_2 = t$.

We are given $g \overset{v}{\Rightarrow} g'$. To show, $g' \overset{t_2,!m}{\rightarrow}$,

$$\begin{array}{ll} & g'' \overset{s,!m}{\rightarrow} \\ \Rightarrow & \{g'' = (g')^{q.time-v.time}\} \\ & g' \overset{q.time-v.time+s,!m}{\rightarrow} \\ \Rightarrow & \{\text{from } t = q.time + s,\ q.time - v.time + s = t - v.time = t_2\} \\ & g' \overset{t_2,!m}{\rightarrow} \end{array}$$

(3) $d_2(j,k)$: Both j and k are of the required form. We are given $d_1(u,v)$. To see $d_0(u,v)$:

$$\begin{array}{ll} & q \in u +_x v \\ \Rightarrow & \{\text{set theory}\} \\ & u +_x v \neq \phi \\ \Rightarrow & \{\text{from Lemma 9, page 52}\} \\ & d_0(u,v) \end{array}$$

Now, we show that $p \in [\![f]\!] <x< [\![g]\!]$.

$$\begin{array}{ll} & p \\ = & \{\text{given}\} \\ & q(t,\tau)r_t \\ \in & \{q \in u +_x v\} \\ & (u +_x v)(t,\tau)r_t \\ = & \{j = u(t,[m/x])r_t,\ k = v(t,!m) \text{ and } d_2(j,k) \text{ holds}\} \\ & j \triangleright k \\ = & \{d_2(j,k) \text{ holds}\} \end{array}$$

$$
\begin{array}{ll}
 & j \;{<}x{<}\; k \\
\subseteq & \{j \in [\![f]\!],\, k \in [\![g]\!]\} \\
 & [\![f]\!] \;{<}x{<}\; [\![g]\!]
\end{array}
$$

2.5.3 $[\![f]\!] \;{<}x{<}\; [\![g]\!] \subseteq [\![f \;{<}x{<}\; g]\!]$

Lemma 16 Suppose $f \overset{u}{\Rightarrow} f'$, $g \overset{v}{\Rightarrow} g'$ and $d_0(u, v)$.
Let $T = \max(u.time, v.time)$, $f'' = (f')^{T-u.time}$, $g'' = (g')^{T-v.time}$.
Then, for any p, where $p \in u +_x v$, $f \;{<}x{<}\; g \overset{p}{\Rightarrow} f'' \;{<}x{<}\; g''$.

Note: The lemma does not assert that under the given conditions, p is an execution of $f \;{<}x{<}\; g$. This is because f'' or g'' may be $\perp$. In order to show that p is an execution of $f \;{<}x{<}\; g$, it has to be shown that neither of these expressions is $\perp$.
Proof: Proof is by induction on the combined lengths of u and v.

- $u = \epsilon$ and $v = \epsilon$: Then, $f' = f$ and $g' = g$. Also, $u.time = v.time = T = 0$, and $f'' = f$ and $g'' = g$. Since $p = \epsilon$, we have to show $f \;{<}x{<}\; g \overset{\epsilon}{\Rightarrow} f \;{<}x{<}\; g$, which follows.

- $u \neq \epsilon$ and $v = \epsilon$: $p \in u +_x v$ means that $p = u$ and u has no substitutions. Given $u \neq \epsilon$, we may write $u = au'_t$. Here a is a base event because u has no substitution. Then, $f \overset{a}{\rightarrow} f_1 \overset{u'}{\Rightarrow} f'$.

$$
\begin{array}{ll}
 & f \;{<}x{<}\; g \\
\overset{a}{\rightarrow} & \{f \overset{a}{\rightarrow} f_1, \text{ and } a \text{ is a base event}\} \\
 & f_1 \;{<}x{<}\; g^t \\
\overset{u'}{\Rightarrow} & \{\text{Induction on } f_1 \overset{u'}{\Rightarrow} f' \text{ and } g^t \overset{\epsilon}{\Rightarrow} g^t; \\
 & \quad \text{let } T' = \max(u'.time, 0) = u'.time\} \\
 & (f')^{T'-u'.time} \;{<}x{<}\; (g^t)^{T'-v'.time} \\
= & \{T - u.time = 0 = T' - u'.time; \\
 & \quad T - v.time = T = u.time = t + u'.time = t + T', \text{ so,} \\
 & \quad (g^t)^{T'-v'.time} = (g')^{T-v.time}\} \\
 & (f')^{T-u.time} \;{<}x{<}\; (g')^{T-v.time}
\end{array}
$$

- $u = \epsilon$ and $v \neq \epsilon$: Similar to the above.

- $u \neq \epsilon$ and $v \neq \epsilon$: Let $p = aq_t$. We consider three cases, (1) a is base (2) a is an other-substitution, and (3) a is an own-substitution.

Case (1) a is base: We have $p = aq_t \in u +_x v$, where a is base. Without loss in generality, assume that
$u = u'_t$ and $v = av'_t$, so that $f \overset{u=u'_t}{\Rightarrow} f'$, $g \overset{a}{\rightarrow} g_1 \overset{v'}{\Rightarrow} g'$.

$$
\begin{array}{ll}
 & p \in u +_x v \\
\Rightarrow & \{p = aq_t,\, u = u'_t \text{ and } v = av'_t\}
\end{array}
$$

$$
\begin{array}{ll}
 & aq_t \in u'_t +_x av'_t \\
\Rightarrow & \{a \text{ is base; so, } u'_t +_x av'_t = a(u'_t +_x v'_t) = a(u' +_x v')_t\} \\
 & aq_t \in a(u' +_x v')_t \\
\Rightarrow & \{\text{obviously}\} \\
 & q \in u' +_x v'
\end{array}
$$

From $d_0(u, v)$, we have $d_0(u', v')$. Let $T' = \max(u'.time, v'.time)$.

$$
\begin{array}{ll}
 & f <x< g \\
\xrightarrow{a} & \{g \xrightarrow{a} g_1\} \\
 & f^t <x< g_1 \\
\overset{q}{\Rightarrow} & \{\text{induction on } f^t \overset{u'}{\Rightarrow} f',\, g_1 \overset{v'}{\Rightarrow} g' \text{ using } q \in u' +_x v' \text{ and } d_0(u', v')\} \\
 & f_2 <x< g_2, \text{ where } f_2 = (f')^{T'-u'.time},\, g_2 = (g')^{T'-v'time} \\
\Rightarrow & \{\text{Using Observation 15, page 51,} \\
 & \quad T' - u'.time = T - u.time,\, T' - v'.time = T - v.time\} \\
 & f_2 <x< g_2, \text{ where } f_2 = (f')^{T-u.time},\, g_2 = (g')^{T-v.time}
\end{array}
$$

Case (2) a is an other-substitution: We have $p = aq_t \in u +_x v$, which means $u = au'_t$, $v = av'_t$ and $q \in u' +_x v'$.
Then, $f \xrightarrow{a} f_1 \overset{u'}{\Rightarrow} f'$, $g \xrightarrow{a} g_1 \overset{v'}{\Rightarrow} g'$. Let $T' = \max(u'.time, v'.time)$. From $d_0(u, v)$, we have $d_0(u', v')$. Let $T' = \max(u'.time, v'.time)$.

$$
\begin{array}{ll}
 & f <x< g \\
\xrightarrow{a} & \{a \text{ is an other-substitution; } f \xrightarrow{a} f_1 \text{ and } g \xrightarrow{a} g_1\} \\
 & f_1 <x< g_1 \\
\overset{q}{\Rightarrow} & \{\text{induction on } f_1 \overset{u'}{\Rightarrow} f',\, g_1 \overset{v'}{\Rightarrow} g' \text{ using } q \in u' +_x v' \text{ and } d_0(u', v')\} \\
 & f_2 <x< g_2, \text{ where } f_2 = (f')^{T'-u'.time},\, g_2 = (g')^{T'-v'.time} \\
\Rightarrow & \{\text{Using Observation 15, page 51,} \\
 & \quad T' - u'.time = T - u.time,\, T' - v'.time = T - v.time\} \\
 & f_2 <x< g_2, \text{ where } f_2 = (f')^{T-u.time},\, g_2 = (g')^{T-v.time}
\end{array}
$$

Case (3) a is an own-substitution: this case is similar to Case (1) where a is base.

Lemma 17 $u \in [\![f]\!]$, $v \in [\![g]\!]$, and $d_1(u, v)$ implies $u <x< v \subseteq [\![f <x< g]\!]$

Proof:

$$
\begin{array}{ll}
 & p \in u <x< v \\
\Rightarrow & \{\text{definition of } u <x< v \text{ given } d_1(u, v)\} \\
 & p \in u|_x v \\
\Rightarrow & \{\text{from Lemma 11, page 54}\} \\
 & p \in u' +_x v', \text{ where } u' \text{ and } v' \text{ are prefixes of } u \text{ and } v, \\
 & u'.time \le v.time \text{ and } v'.time \le u.time \\
\Rightarrow & \{\text{from Lemma 9, page 52, } p \in u' +_x v' \;\Rightarrow\; d_0(u', v')\} \\
 & p \in u' +_x v', \text{ for prefixes } u' \text{ of } u \text{ and } v' \text{ of } v, \text{ and } d_0(u', v'),
\end{array}
$$

$$
\begin{array}{ll}
 & u'.time \le v.time \text{ and } v'.time \le u.time \\
\Rightarrow & \{u \in [\![f]\!],\, v \in [\![g]\!];\ u' \text{ and } v' \text{ are prefixes of } u \text{ and } v\} \\
 & p \in u' +_x v',\ f \overset{u'}{\Rightarrow} f',\ g \overset{v'}{\Rightarrow} g', \text{ for some } f' \text{ and } g',\ d_0(u', v'), \\
 & u'.time \le v.time \text{ and } v'.time \le u.time \\
\Rightarrow & \{\text{from Lemma 16, page 63}\} \\
 & f <x< g \overset{p}{\Rightarrow} (f')^{T-u'.time} <x< (g')^{T-v'.time}, \text{ where} \\
 & T = \max(u'.time, v'.time),\ u'.time \le v.time \text{ and } v'.time \le u.time
\end{array}
$$

Next, we show that $(f')^{T-u'.time} \neq \bot$ and $(g')^{T-v'.time} \neq \bot$; hence, that $p \in [\![f <x< g]\!]$, i.e., $u <x< v \subseteq [\![f <x< g]\!]$.

Given that $u \in [\![f]\!]$, u' is a prefix of u, and $f \overset{u'}{\Rightarrow} f'$, we have $f \overset{u'}{\Rightarrow} f' \overset{u''}{\Rightarrow}$, where $u = u'(u'')_{u'.time}$.

$$
\begin{array}{ll}
 & (f')^{u''.time} \neq \bot \\
\Rightarrow & \{\text{from } u = u'(u'')_{u'.time},\ u.time = u'.time + u''.time\} \\
 & (f')^{u.time-u'.time} \neq \bot \\
\Rightarrow & \{u' \text{ is a prefix of } u; \text{ so, } u'.time \le u.time \\
 & \text{Given, } v'.time \le u.time. \text{ So, } T = \max(u'.time, v'.time) \le u.time. \\
 & \text{Hence, } T - u'.time \le u.time - u'.time\} \\
 & (f')^{T-u'.time} \neq \bot
\end{array}
$$

Similarly, $(g')^{T-v'.time} \neq \bot$.

Theorem 16 $[\![f]\!] <x< [\![g]\!] \subseteq [\![f <x< g]\!]$

Proof: We show

1. $u \in [\![f]\!]$, $v \in [\![g]\!]$, and $d_1(u, v)$ implies $u <x< v \subseteq [\![f <x< g]\!]$,
2. $u \in [\![f]\!]$, $v \in [\![g]\!]$, and $d_2(u, v)$ implies $u <x< v \subseteq [\![f <x< g]\!]$, and
3. $u \in [\![f]\!]$, $v \in [\![g]\!]$, $\neg d_1(u, v)$, and $\neg d_2(u, v)$ implies $u <x< v \subseteq [\![f <x< g]\!]$

The first case follows from Lemma 17, page 64. The last case is trivial, since $u <x< v$ in that case is ϕ, a subset of any set. So, we prove only the second case.

• $u \in [\![f]\!]$, $v \in [\![g]\!]$, and $d_2(u, v)$ implies $u <x< v \subseteq [\![f <x< g]\!]$: Given $d_2(u, v)$ we may assume that

$$u = u'(t, [m/x])u''_t \in [\![f]\!],\ v = v'(t, !m)v'' \in [\![g]\!] \qquad \text{(F1)}$$

Then $u <x< v = (u' +_x v')(t, \tau)u''_t$.
We have to show that for any p, where $p \in u' +_x v'$, $p(t, \tau)u''_t \subseteq [\![f <x< g]\!]$.
We prove this by showing

$$f <x< g \overset{p}{\Rightarrow} f' <x< g' \overset{t-T,\tau}{\rightarrow} [m/x].(f')^{t-T} \overset{u''}{\Rightarrow}, \text{ where } T = p.time, \text{ or}$$

$$f <x< g \overset{p}{\Rightarrow} f' <x< g', \text{ for some } f' \text{ and } g', \qquad (1)$$

$$f' <x< g' \overset{t-T,\tau}{\rightarrow} [m/x].(f')^{t-T}, \qquad (2)$$

$$[m/x].(f')^{t-T} \overset{u''}{\Rightarrow} \qquad (3)$$

Note that, given $p \in u' +_x v'$, using Lemma 9, page 52, $p.time = \max(u'.time, v'.time)$. Since $T = p.time$, $T = \max(u'.time, v'.time)$.

Now,

$u \in [\![f]\!]$ means $f \stackrel{u'}{\Rightarrow} f_1 \stackrel{t_1,[m/x]}{\rightarrow} [m/x].(f_1)^{t_1} \stackrel{u''}{\Rightarrow}$, where
$t_1 + u'.time = t$. Also, $(f_1)^{t_1} \neq \perp$. (F2)

$v \in [\![g]\!]$ means $g \stackrel{v'}{\Rightarrow} g_1 \stackrel{t_2,!m}{\rightarrow}$, where
$t_2 + v'.time = t$. Also, $(g_1)^{t_2} \neq \perp$. (F3)

(1) $f <x< g \stackrel{p}{\Rightarrow} f' <x< g'$: We are given $f \stackrel{u'}{\Rightarrow} f_1$, $g \stackrel{v'}{\Rightarrow} g_1$. Also, $d_0(u', v')$ follows from $d_2(u, v)$, and $p \in u' +_x v'$. Applying Lemma 16, page 63, we get

$f <x< g \stackrel{p}{\Rightarrow} f' <x< g'$, where
$f' = (f_1)^{T-u'.time}$, $g' = (g_1)^{T-v'.time}$ (Recall $T = \max(u'.time, v'.time)$)

Next, we show that $f' \neq \perp$ and $g' \neq \perp$. First, from $u = u'(t, [m/x])u''_t$, $u'.time \leq t$, and from $v = v'(t, !m)v''$, $v'.time \leq t$. Therefore, $T = \max(u'.time, v'.time) \leq t$. Now, $t_1 = t - u'.time \geq T - u'.time$. Since $(f_1)^{t_1} \neq \perp$, from (F2), $(f_1)^{T-u'.time} = f' \neq \perp$. Similarly, $g' \neq \perp$.

(2) $f' <x< g' \stackrel{t-T,\tau}{\rightarrow} [m/x].(f')^{t-T}$: From (F2),

$$
\begin{array}{cl}
 & g_1 \stackrel{t_2,!m}{\rightarrow} \\
\Rightarrow & \{t_2 = t - v'.time = T - v'.time + t - T\} \\
 & (g_1)^{T-v'.time} \stackrel{t-T,!m}{\rightarrow} \\
\Rightarrow & \{g' = (g_1)^{T-v'.time}\} \\
 & (g') \stackrel{t-T,!m}{\rightarrow}
\end{array}
$$

Hence, from the operational semantics, using (ASYM2V),

$$f' <x< g' \stackrel{t-T,\tau}{\rightarrow} [m/x].(f')^{t-T}$$

(3) $[m/x].(f')^{t-T} \stackrel{u''}{\Rightarrow}$:

$$
\begin{array}{cl}
 & (f')^{t-T} \\
= & \{f' = (f_1)^{T-u'.time}\} \\
 & (f_1)^{T-u'.time+t-T} \\
= & \{\text{simplify exponent}\} \\
 & (f_1)^{t-u'.time} \\
= & \{t_1 = t - u'.time, \text{ from (F2)}\} \\
 & (f_1)^{t_1}
\end{array}
$$

Given $[m/x].(f_1)^{t_1} \stackrel{u''}{\Rightarrow}$, we have $[m/x].(f')^{t-T} \stackrel{u''}{\Rightarrow}$. This completes the proof.

Chapter 3

Breadth and Trace Preservation

The goal of this chapter is to show that the traces of $f * g$ can be determined from the traces of f and g. Specifically, we show

$$\langle\!\langle f * g \rangle\!\rangle = \overline{\langle\!\langle f \rangle\!\rangle * \langle\!\langle g \rangle\!\rangle}. \tag{P1}$$

where $*$ is any orc combinator, $\mid$, $>x>$ or $<x<$.

We prove (P1) by first showing, for sets U and V,

$$\overline{U * V} = \overline{\overline{U} * \overline{V}}. \tag{P2}$$

Then (P1) follows,

$$\begin{array}{cl} & \langle\!\langle f * g \rangle\!\rangle \\ = & \{\text{definition of } \langle\!\langle f * g \rangle\!\rangle\} \\ & \overline{[\![f * g]\!]} \\ = & \{\text{from Characterization Theorems in Chapter 2, } [\![f * g]\!] = [\![f]\!] * [\![g]\!]\} \\ & \overline{[\![f]\!] * [\![g]\!]} \\ = & \{\text{Use (P2) with } U = [\![f]\!] \text{ and } V = [\![g]\!]\} \\ & \overline{\overline{[\![f]\!]} * \overline{[\![g]\!]}} \\ = & \{\text{from definition, } \langle\!\langle f \rangle\!\rangle = \overline{[\![f]\!]} \text{ and similarly for } g\} \\ & \overline{\langle\!\langle f \rangle\!\rangle * \langle\!\langle g \rangle\!\rangle} \end{array}$$

The sets U and V in (P2) are not arbitrary, however. In particular, we call set U to be *broad* (1) if $x \in U$, then $xc \in U$, for any substitution c where $c.time = x.time$, and (2) if $xb \in U$, then $xc \in U$, for any substitution c where $x.time \leq c.time \leq b.time$. The formal definition, given in Section 3.1.1, page 68, is inductively defined to facilitate algebraic manipulations, though it is equivalent to the definition given here. Additionally, we will require that the sets be substitution independent, see Section 1.5.2, page 15. We establish (P2) under these conditions.

Clearly, we have to show that every $[\![f]\!]$ is broad and substitution independent. The latter result has been proved in Observation 9, page 20. We prove that $[\![f]\!]$ is broad by induction on the structure of f. Base orc expressions are broad, from Lemma 29, page 74. And we show that each combinator preserves breadth, i.e., if U and V are broad then so is $U*V$. In this chapter, we discharge both sets of proof obligations: (1) $U*V$ is broad given U and V are broad, and (2) $\overline{U*V} = \overline{\overline{U}*\overline{V}}$, given U and V are broad and substitution independent. We prove these results separately for each combinator, after establishing some preliminary results in the following section.

3.1 Additional Operators on Sequences

3.1.1 Breadth

The *breadth* of p, $\beta(p)$, is the set of sequences that can be generated from p by applying the substitution rule. Formally,

$\beta(\epsilon) = A(0)$
$\beta(ap_t) = A(t) \cup a(\beta(p))_t$, where $t = a.time$

Notation: Henceforth, we write $\beta(p)_t$ for $(\beta(p))_t$. Note that $\beta(p)_t$ is different from $\beta(p_t)$ (see Lemma 22, page 71, for a relationship between the two). We define $\beta()$ to be coercive, i.e.,

$\beta(P) = (\cup p : p \in P : \beta(p))$.

Note that $\beta(\phi) = \phi$.

Observation 17 *$\beta(\epsilon) \subseteq \beta(p)$, for any p.*

Proof: $\beta(\epsilon) = A(0)$, and from the definition of $\beta()$, $A(0) \subseteq \beta(p)$, for any p.

Broad Define set P to be *broad* iff $P = \beta(P)$.

It follows that if P and Q are broad, $P \cup Q$ is broad:
$\beta(P \cup Q) = \beta(P) \cup \beta(Q) = P \cup Q$. Also, that the empty set is broad.

Lemma 18 *$A(r)$ is broad for any r.*

Proof: We observe that for p, any finite sequence of substitutions at time 0, $\beta(p) = A(0)$ (proof is by induction on the length of p). Therefore, $\beta(A(0)) = A(0)$, i.e., $A(0)$ is broad.

Next, observe that any sequence of $A(r)$ is of the form ap_t, where $0 \le t \le r$, a is a substitution at t, and p is a finite sequence of substitutions at time 0.

$\quad\ \beta(ap_t)$
$= \quad \{\text{definition of } \beta()\}$

$$
\begin{array}{ll}
& A(t) \cup a\beta(p)_t \\
= & \{p \in A(0). \text{ Hence, } \beta(p) = A(0), \text{ from the the proof above}\} \\
& A(t) \cup A(0)_t \\
= & \{A(0)_t \subseteq A(t)\} \\
& A(t)
\end{array}
$$

We now show that $A(r)$ is broad, i.e., $\beta(A(r)) = A(r)$.

$$
\begin{array}{ll}
& \beta(A(r)) \\
= & \{\text{coercion}\} \\
& (\cup q : q \in A(r) : \beta(q)) \\
= & \{\text{from above, } \beta(q) = A(t), \text{ where } t = q.time\} \\
& (\cup t : 0 \leq t \leq r : A(t)) \\
= & \{\text{for } t \leq r,\ A(t) \subseteq A(r)\} \\
& A(r)
\end{array}
$$

Lemma 19 $D(t)$ is broad for any t.

Proof: The proof is by induction on the length of $p \in D(t)$. For $p = \epsilon$, $\beta(\epsilon) = A(0)$ and $A(0) \subseteq D(t)$ by Obs. 7 on page 9. Otherwise $p = aq_s$, for substitution event a with $a.time = s \leq t$.

$$
\begin{array}{ll}
& \beta(aq_s) \\
= & \{\text{definition of } \beta()\} \\
& A(s) \cup a\beta(q)_s \\
\subseteq & \{\text{induction on } q \in D(t-s)\} \\
& A(s) \cup aD(t-s)_s \\
\subseteq & \{aD(t-s)_s \subseteq D(t)\} \\
& A(s) \cup D(t) \\
\subseteq & \{A(s) \subseteq A(t) \subseteq D(t) \text{ by Obs. 7}\} \\
& D(t)
\end{array}
$$

Lemma 20 $\beta(\beta(p)) = \beta(p)$. So, $\beta(p)$ is broad.

Proof: Proof is by induction on the length of p. For $p = \epsilon$, $\beta(\epsilon) = A(0)$, and $A(0)$ is broad from Lemma 18, page 68.

$$
\begin{array}{ll}
& \beta(\beta(ap_t)), \text{ where } t = a.time \\
= & \{\text{definition of } \beta()\} \\
& \beta(A(t) \cup a\beta(p)_t) \\
= & \{\beta() \text{ distributes over union}\} \\
& \beta(A(t)) \cup \beta(a\beta(p)_t) \\
= & \{\beta(A(t)) = A(t), \text{ from Lemma 18, page 68}\} \\
& A(t) \cup \beta(a\beta(p)_t) \\
= & \{\text{definition of } \beta()\} \\
& A(t) \cup A(t) \cup a\beta(\beta(p))_t \\
= & \{\text{induction}\} \\
& A(t) \cup a\beta(p)_t \\
= & \{\text{definition of } \beta()\} \\
& \beta(ap_t)
\end{array}
$$

Lemma 21 $p^* \subseteq \beta(p)$.

Proof: Proof is by induction on the length of p.

First, we show $\epsilon^* \subseteq \beta(\epsilon)$. $\epsilon^* = \{\epsilon\}$, and $\beta(\epsilon) = A(0)$. And, $\{\epsilon\} \subseteq A(0)$, from the definition of $A(0)$.

Next, we show that $(ap_t)^* \subseteq \beta(ap_t)$, where $t = a.time$.

$$
\begin{array}{ll}
 & (ap_t)^* \\
= & \{\text{definition of prefix-closure}\} \\
 & \{\epsilon\} \cup a(p_t)^* \\
= & \{(p_t)^* = (p^*)_t, \text{ from Lemma 1, page 7}\} \\
 & \{\epsilon\} \cup a(p^*)_t \\
\subseteq & \{\{\epsilon\} \subseteq A(t) \text{ and inductively, } p^* \subseteq \beta(p), \text{ so } (p^*)_t \subseteq \beta(p)_t\} \\
 & A(t) \cup a\beta(p)_t \\
= & \{\text{definition of breadth}\} \\
 & \beta(ap_t)
\end{array}
$$

Corollary 2 $p \in \beta(p)$. And $P \subseteq \beta(P)$.

Proof: $p \in p^*$. From Lemma 21, page 70, $p^* \subseteq \beta(p)$. Therefore, $p \in \beta(p)$. And, $P \subseteq \beta(P)$ follows by applying coercion.

Corollary 3 A broad set is prefix-closed.

Proof: For broad set P and any sequence q, we show that $q \in P \Rightarrow q^* \subseteq P$.

$$
\begin{array}{ll}
 & q \in P \\
\Rightarrow & \{\text{apply } \beta() \text{ to both sides}\} \\
 & \beta(q) \subseteq \beta(P) \\
\Rightarrow & \{\beta(P) = P, \text{ since } P \text{ is broad}\} \\
 & \beta(q) \subseteq P \\
\Rightarrow & \{q^* \subseteq \beta(q), \text{ from Lemma 21, page 70}\} \\
 & q^* \subseteq P
\end{array}
$$

□

To prove that set P is broad, we can employ any of the following characterizations of a broad set.

Corollary 4 P is broad iff

1. $P = \beta(Q)$, for some set Q.
2. $\beta(P) \subseteq P$.
3. for every p, where $p \in P$, $\beta(p) \subseteq P$.
4. P is a union of broad sets.

Proof: (1) If P is broad, $P = \beta(P)$, by definition. And, if $P = \beta(Q)$, for some set Q, then $\beta(P) = \beta(\beta(Q)) = \beta(Q) = P$; hence P is broad. (2) From the definition of broad, and Corollary 2, page 70. (3) is a rewriting of (2): $\beta(P) \subseteq P$ is same as, for all p, $p \in P$, $\beta(p) \subseteq P$. (4) If P is a union of broad sets, say P_is, then $\beta(P) = \beta(\cup i :: P_i) = (\cup i :: \beta(P_i)) = (\cup i :: P_i) = P$. Conversely, if P is a broad set then it is a union of broad sets vacuously (to make it non-vacuous, take the union of P and the empty set, which are both broad).

Lemma 22 *For non-empty* q, $\beta(q_s) = A(s) \cup \beta(q)_s$

Proof: Let $q = ap_t$, where $t = a.time$. Then $q_s = a_s p_{s+t} = bp_{s+t}$, where $b = a_s$.

$$
\begin{array}{ll}
 & \beta(q_s) \\
= & \{q_s = bp_{s+t}\} \\
 & \beta(bp_{s+t}) \\
= & \{b.time = a_s.time = a.time + s = s + t;\ \text{expand } \beta(bp_{s+t})\} \\
 & A(s+t) \cup b\beta(p)_{s+t} \\
= & \{A(s+t) = A(s) \cup A(t)_s,\ \text{from Observation 7, page 9; rewrite } b\} \\
 & A(s) \cup A(t)_s \cup a_s\beta(p)_{s+t}
\end{array}
$$

And,

$$
\begin{array}{ll}
 & A(s) \cup \beta(q)_s \\
= & \{q = ap_t\} \\
 & A(s) \cup \beta(ap_t)_s \\
= & \{\text{definition of } \beta()\} \\
 & A(s) \cup (A(t) \cup a\beta(p)_t)_s \\
= & \{\text{rewriting}\} \\
 & A(s) \cup A(t)_s \cup a_s\beta(p)_{s+t}
\end{array}
$$

Lemma 23 $uc \in \beta(u)$, *where* c *is any substitution and* $c.time = u.time$.

Proof: By induction on the length of u.

• $u = \epsilon$: We write simply c for the sequence containing just c, in the following proof. We have to show that $c \in \beta(\epsilon) = A(0)$, where c is any substitution at time 0. This follows from the definition of $A(0)$.

• $au_tc_t \in \beta(au_t)$ where $c.time = u.time$: Inductively, $uc \in \beta(u)$, and

$$
\begin{array}{ll}
 & uc \in \beta(u) \\
\Rightarrow & \{\text{time shift applied to both sides}\} \\
 & (uc)_t \in \beta(u)_t \\
\Rightarrow & \{\text{concatenation applied to both sides}\} \\
 & a(uc)_t \in a\beta(u)_t \\
\Rightarrow & \{a\beta(u)_t \subseteq \beta(au_t),\ \text{from definition of } \beta()\} \\
 & a(uc)_t \in \beta(au_t) \\
\Rightarrow & \{a(uc)_t = au_tc_t\} \\
 & au_tc_t \in \beta(au_t)
\end{array}
$$

3.1.2 Visible sequences and Traces

A sequence is *visible* if it is empty or its last event is non-τ.

Lemma 24 For visible sequence p, $\overline{\beta(p)} = \beta(\overline{p})$.

Proof: Proof is by induction on the length of p.

$\overline{\beta(\epsilon)} = \overline{A(0)}$, and from Observation 7, page 9, $\overline{A(0)} = A(0)$, and
$\beta(\overline{\epsilon}) = \beta(\epsilon) = A(0)$

Consider ap_t, where ap_t is visible and $t = a.time$. Note that p is visible (p could be ϵ).

$$
\begin{array}{ll}
 & \overline{\beta(ap_t)} \\
= & \{\text{definition of } \beta()\} \\
 & \overline{A(t) \cup a\beta(p)_t} \\
= & \{\text{distribute trace over union and concatenation}\} \\
 & \overline{A(t)} \cup \overline{a}\,\overline{(\beta(p)_t)} \\
= & \{\overline{(\beta(p)_t)} = (\overline{\beta(p)})_t, \text{ from Lemma 1, page 7}\} \\
 & \overline{A(t)} \cup \overline{a}(\overline{\beta(p)})_t \\
= & \{\text{induction using } p \text{ is visible; also use } \overline{A(t)} = A(t)\} \\
 & A(t) \cup \overline{a}\beta(\overline{p})_t
\end{array}
$$

We show that $A(t) \cup \overline{a}\beta(\overline{p})_t = \beta(\overline{ap_t})$. Consider: (1) $a \neq \tau$, (2) $a = \tau$.
(1) $a \neq \tau$: Then $a = \overline{a}$.

$$
\begin{array}{ll}
 & A(t) \cup \overline{a}\beta(\overline{p})_t \\
= & \{\text{definition of } \beta() \text{ and } \overline{a}.time = t\} \\
 & \beta(\overline{a}\,\overline{p}_t) \\
= & \{(\overline{p})_t = \overline{(p_t)}, \text{ from Lemma 1, page 7}\} \\
 & \beta(\overline{a}\,\overline{(p_t)}) \\
= & \{\text{distribute trace over concatenation}\} \\
 & \beta(\overline{ap_t})
\end{array}
$$

(2) $a = \tau$: then $\overline{a} = \epsilon$ and p is non-empty and visible, because ap_t is visible.

$$
\begin{array}{ll}
 & A(t) \cup \overline{a}\beta(\overline{p})_t \\
= & \{\overline{a} = \epsilon\} \\
 & A(t) \cup \beta(\overline{p})_t \\
= & \{p \text{ is non-empty and visible; so } \overline{p} \text{ is non-empty. from Lemma 22, page 71}\} \\
 & \beta((\overline{p})_t) \\
= & \{(\overline{p})_t = \overline{(p_t)}, \text{ from Lemma 1, page 7}\} \\
 & \beta(\overline{p_t}) \\
= & \{\overline{a} = \epsilon. \text{ So, } \beta(\overline{p_t}) = \beta(\overline{a}\,\overline{p_t}) = \beta(\overline{ap_t})\} \\
 & \beta(\overline{ap_t})
\end{array}
$$

Lemma 25 Given that U is broad, $\overline{U}$ is broad.

Proof: We show that $\beta(x) \subseteq \overline{U}$ for every $x \in \overline{U}$. Then $\overline{U}$ is broad, from Corollary 4, page 70.

Given that $x \in \overline{U}$ there is $y \in U$ such that $x = \overline{y}$. We can assume that y is visible (otherwise, z, the longest prefix of y that is visible satisfies $x = \overline{z}$ and $z \in U$, from prefix-closure of U). Note that if $x = \epsilon$, then $y = \epsilon$ satisfies the requirements.

$$
\begin{array}{ll}
 & y \in U \\
\Rightarrow & \{U \text{ is broad; use Corollary 4, page 70}\} \\
 & \beta(y) \subseteq U \\
\Rightarrow & \{\text{apply trace to both sides}\} \\
 & \overline{\beta(y)} \subseteq \overline{U} \\
\Rightarrow & \{y \text{ is visible; from Lemma 24, page 72, } \overline{\beta(y)} = \beta(\overline{y})\} \\
 & \beta(\overline{y}) \subseteq \overline{U} \\
\Rightarrow & \{x = \overline{y}\} \\
 & \beta(x) \subseteq \overline{U}
\end{array}
$$

Event Removal from front of a sequence

We have defined the removal operator in page 6. Repeating the definition,

$$
u \backslash a = \begin{cases} \{v\} & \text{if } u = av_t \text{ where } t = a.time \\ \phi & \text{otherwise} \end{cases}
$$

$$
U \backslash a = \{v \mid av_t \in U\}, \text{ where } t = a.time
$$

Lemma 26 Given that U is broad, $U \backslash a$ is broad.

Proof: We show that for any p, $p \in U \backslash a$, $\beta(p) \subseteq U \backslash a$. The result follows from Corollary 4, page 70 part(3).

$$
\begin{array}{ll}
 & p \in U \backslash a \\
\Rightarrow & \{\text{definition of } U \backslash a\} \\
 & ap_t \in U, \text{ where } t = a.time \\
\Rightarrow & \{U \text{ broad; use Corollary 4, page 70}\} \\
 & \beta(ap_t) \subseteq U \\
\Rightarrow & \{a\beta(p)_t \subseteq \beta(ap_t), \text{ from the definition of breadth}\} \\
 & a\beta(p)_t \subseteq U \\
\Rightarrow & \{\text{definition of } U \backslash a\} \\
 & \beta(p) \subseteq U \backslash a
\end{array}
$$

Reducing times in a sequence

Given set V, define V_{-t}, for $t \geq 0$, to be

$$
V_{-t} = \{v \mid v_t \in V\}
$$

Thus, every sequence in V that starts at or after t has all its event times reduced by t, and all other non-empty sequences are discarded. The empty sequence, if it is in V, is retained because $\epsilon_t = \epsilon$.

Observation 18 $A(t)_{-t} = A(0)$, from definition.

Lemma 27 Given that V is broad and some $u_t \in V$, where $u \neq \epsilon$, V_{-t} is broad.

Proof: First, note the essential requirement that some sequence u_t, whose first event starts at or after t, is in V. Otherwise, take $V = A(0)$, which is broad, and choose some positive t. Then $A(0)_{-t} = \{\epsilon\}$. But $\{\epsilon\}$ is not broad.

The proof follows by showing that for any v, $v \in V_{-t}$, $\beta(v) \in V_{-t}$. Proof is by induction on the length of v.

- $v = \epsilon$: We have to show that $\beta(\epsilon) = A(0) \subseteq V_{-t}$. Given,

$$
\begin{array}{ll}
 & u_t \in V \\
\Rightarrow & \{V \text{ is broad}\} \\
 & \beta(u_t) \subseteq V \\
\Rightarrow & \{A(t) \subseteq \beta(u_t), \text{ from definition of } \beta()\} \\
 & A(t) \subseteq V \\
\Rightarrow & \{\text{definition of } V_{-t}\} \\
 & A(t)_{-t} \subseteq V_{-t} \\
\Rightarrow & \{A(t)_{-t} = A(0), \text{ from Observation 18, page 74}\} \\
 & A(0) \subseteq V_{-t}
\end{array}
$$

- $v \neq \epsilon$:

$$
\begin{array}{ll}
 & v \in V_{-t} \\
\Rightarrow & \{\text{definition of } V_{-t}\} \\
 & v_t \in V \\
\Rightarrow & \{V \text{ is broad}\} \\
 & \beta(v_t) \in V \\
\Rightarrow & \{\text{from Lemma 22, page 71, } \beta(v)_t \subseteq \beta(v_t)\} \\
 & \beta(v)_t \in V \\
\Rightarrow & \{\text{definition of } V_{-t}\} \\
 & \beta(v) \in V_{-t}
\end{array}
$$

3.1.3 Base Expressions are Broad

Lemma 28 *$A(t) \subseteq [\![M(x)]\!]$, for any time t.*

Proof: Consider $p \in A(t)$. If p does not contain a substitution to x, then $p \in [\![M(x)]\!]$ follows easily from Theorem 8 on page 33. Otherwise, let $p = q(t, [m/x])r$, where q has no substitution to x. By Theorem 8 it suffices to show that $p \in D(t)\backslash x \cdot (t, [m/x]) \cdot [\![M(m)]\!]_t$. Since $A(t) \subseteq D(t)$ by Observation 7 on page 9, we have $A(t)\backslash x \subseteq D(t)\backslash x$. Since $q \in A(t)\backslash x$ it follows that $q \in D(t)\backslash x$. And $r \in A(t) \subseteq D(t)$, and so $r \in [\![M(m)]\!]_t$ by Theorem 8.

Lemma 29 Base expressions are broad.

Proof:

- $\beta([\![\,\mathbf{0}\,]\!]) \subseteq [\![\,\mathbf{0}\,]\!]$:

$$\begin{array}{ll} & \beta([\![\,\mathbf{0}\,]\!]) \\ = & \{\text{Theorem 8 on page 33}\} \\ & \beta((\cup t : t \in \mathcal{T} : D(t))) \\ = & \{\beta() \text{ is coercive}\} \\ & (\cup t : t \in \mathcal{T} : \beta(D(t))) \\ \subseteq & \{\text{Lemma 19 on page 69}\} \\ & (\cup t : t \in \mathcal{T} : D(t)) \\ = & \{\text{Theorem 8 on page 33}\} \\ & [\![\,\mathbf{0}\,]\!] \end{array}$$

- $\beta([\![\,?k\,]\!]) \subseteq [\![\,?k\,]\!]$: We show that, for $p \in [\![\,?k\,]\!]$, $\beta(p) \subseteq [\![\,?k\,]\!]$. By Theorem 8 on page 33, either $p \in [\![\,\mathbf{0}\,]\!]$ or, for some time t and value m, $p \in D(t) \cdot (t, !m) \cdot [\![\,\mathbf{0}\,]\!]_t$. The first case follows from the proof above that $\beta([\![\,\mathbf{0}\,]\!]) \subseteq [\![\,\mathbf{0}\,]\!]$. Otherwise it suffices to consider $p = q(t, !m)r_t$, where $q \in D(t)$ and $r \in [\![\,\mathbf{0}\,]\!]$; the other cases follow from Corollary 3 on page 70.

  Suppose $q = \epsilon$.

$$\begin{array}{ll} & \beta((t, !m)r_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup (t, !m)\beta(r)_t \\ \subseteq & \{r \in [\![\,\mathbf{0}\,]\!] \text{ by assumption}\} \\ & A(t) \cup (t, !m)\beta([\![\,\mathbf{0}\,]\!])_t \\ \subseteq & \{\text{above}\} \\ & A(t) \cup (t, !m)[\![\,\mathbf{0}\,]\!]_t \\ \subseteq & \{A(t) \subseteq D(t) \text{ by Observation 7 on page 9}\} \\ & D(t) \cup (t, !m)[\![\,\mathbf{0}\,]\!]_t \\ \subseteq & \{D(t) \subseteq [\![\,?k\,]\!] \text{ and } (t, !m)[\![\,\mathbf{0}\,]\!]_t \subseteq [\![\,?k\,]\!] \text{ by Theorem 8 on page 33}\} \\ & [\![\,?k\,]\!] \end{array}$$

  Otherwise $q = aq'_s$ and $p = ap'_s$.

$$\begin{array}{ll} & \beta(ap'_s) \\ = & \{\text{definition of } \beta()\} \\ & A(s) \cup a\beta(p')_s \\ \subseteq & \{\text{induction on } p' \in [\![\,?k\,]\!]\} \\ & A(s) \cup a[\![\,?k\,]\!]_s \\ \subseteq & \{A(s) \subseteq D(s)\} \\ & D(s) \cup a[\![\,?k\,]\!]_s \\ \subseteq & \{D(s) \subseteq [\![\,?k\,]\!] \text{ and } a[\![\,?k\,]\!]_s \subseteq [\![\,?k\,]\!] \text{ by Theorem 8 on page 33}\} \\ & [\![\,?k\,]\!] \end{array}$$

- $\beta([\![\,M(m)\,]\!]) \subseteq [\![\,M(m)\,]\!]$: We show that, for $p \in [\![\,M(m)\,]\!]$, $\beta(p) \subseteq [\![\,M(m)\,]\!]$. By Theorem 8 on page 33, it suffices to consider $p = q(0, \tau)r \in D(0) \cdot (0, \tau) \cdot [\![\,?k\,]\!]$, for some $k \in \Sigma(M, m)$ where $q \in D(0)$ and $r \in [\![\,?k\,]\!]$; the other cases follow from Corollary 3 on page 70.

Suppose $q = \epsilon$.

$$\begin{array}{ll} & \beta((0,\tau)r) \\ = & \{\text{definition of } \beta()\} \\ & A(0) \cup (0,\tau)\beta(r) \\ \subseteq & \{r \in [\![\, ?k \,]\!], \text{ which is broad}\} \\ & A(0) \cup (0,\tau)?k \\ \subseteq & \{A(0) \subseteq D(0) \text{ by Observation 7 on page 9}\} \\ & D(0) \cup (0,\tau)?k \\ \subseteq & \{D(0) \subseteq [\![\, M(m) \,]\!] \text{ and } (0,\tau)[\![\, ?k \,]\!] \subseteq [\![\, M(m) \,]\!] \text{ by Theorem 8 on page 33}\} \\ & [\![\, M(m) \,]\!] \end{array}$$

Otherwise $q = aq'$ and $p = ap'$, where $a.time = 0$.

$$\begin{array}{ll} & \beta(ap') \\ = & \{\text{definition of } \beta()\} \\ & A(0) \cup a\beta(p') \\ \subseteq & \{\text{induction on } p' \in [\![\, M(m) \,]\!]\} \\ & A(0) \cup a[\![\, M(m) \,]\!] \\ \subseteq & \{A(0) \subseteq D(0) \text{ by Observation 7 on page 9}\} \\ & D(0) \cup a[\![\, M(m) \,]\!] \\ \subseteq & \{D(0) \subseteq [\![\, M(m) \,]\!] \text{ and } a \subseteq [\![\, M(m) \,]\!] \text{ by Theorem 8 on page 33}\} \\ & [\![\, M(m) \,]\!] \end{array}$$

- $\beta([\![\, M(x) \,]\!]) \subseteq [\![\, M(x) \,]\!]$: Consider $p \in [\![\, M(x) \,]\!]$. By Theorem 8 on page 8, it suffices to consider $p = q(t, [m/x])r_t$, where $q \in D(t)\backslash x$ and $r \in [\![\, M(m) \,]\!]$; the other cases follow from Corollary 3 on page 70.

  Suppose $q = \epsilon$.

$$\begin{array}{ll} & \beta((t,[m/x])r_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup (t,[m/x])\beta(r)_t \\ \subseteq & \{r \in [\![\, M(m) \,]\!],\ \beta(M(m)) \subseteq M(m) \text{ as above}\} \\ & A(t) \cup (t,[m/x])[\![\, M(m) \,]\!]_t \\ \subseteq & \{A(t) \subseteq [\![\, M(x) \,]\!] \text{ by Lemma 28 on page 74, } (t,[m/x])[\![\, M(m) \,]\!]_t \subseteq [\![\, M(x) \,]\!] \text{ by Theorem 8}\} \\ & [\![\, M(x) \,]\!] \end{array}$$

  Otherwise $q = aq'_s$ and $p = ap'_s$.

$$\begin{array}{ll} & \beta(ap'_s) \\ = & \{\text{definition of } \beta()\} \\ & A(s) \cup a\beta(p')_s \\ \subseteq & \{\text{induction on } p' \in [\![\, M(x) \,]\!]\} \\ & A(s) \cup a[\![\, M(x) \,]\!]_s \\ \subseteq & \{A(s) \subseteq [\![\, M(x) \,]\!] \text{ by Lemma 28 on page 74, } a[\![\, M(x) \,]\!]_s \subseteq [\![\, M(x) \,]\!] \text{ by Theorem 8}\} \\ & [\![\, M(x) \,]\!] \end{array}$$

3.2 Symmetric Composition

We use the definition of $\mid$ applied to sets, as given in Section 2.2.1, page 36.

3.2.1 Preliminary Results

We use the following algebraic properties of guarded sets.

Observation 19

1. $[true \;\rightarrow\; S] = S$, $[false \;\rightarrow\; S] = \{\epsilon\}$.
2. Given that $\epsilon \in S'$, $[false \;\rightarrow\; S] \cup [p' \;\rightarrow\; S'] = [p' \;\rightarrow\; S']$.
3. Given that $S \subseteq S'$, $[p \;\rightarrow\; S] \subseteq [p \;\rightarrow\; S']$.
4. Suppose $f(\epsilon) = \{\epsilon\}$. Then, $f[p \;\rightarrow\; S] = [p \;\rightarrow\; f(S)]$. Thus,

$$\begin{array}{l}\overline{[p \;\rightarrow\; S]} = [p \;\rightarrow\; \overline{S}],\\ [p \;\rightarrow\; S]^* = [p \;\rightarrow\; S^*],\\ \beta([p \;\rightarrow\; S]) = [p \;\rightarrow\; \beta(S)]\end{array}$$

Lemma 30 $\mid$ is commutative.

Proof: Proof is by induction on the combined length of the arguments. If either u or v is empty, then $u \mid v = \{\epsilon\} = v \mid u$. Now, we show that $bv \mid au = au \mid bv$.

$$\begin{array}{cl} & bv \mid au \\ = & \{\text{definition of } \mid\} \\ & [b \simeq a \rightarrow b(v \mid u)] \cup [b \preceq a \rightarrow b(v \mid au)] \cup [a \preceq b \rightarrow a(bv \mid u)] \\ = & \{b \simeq a \equiv a \simeq b.\text{ Also, } a \simeq b \Rightarrow a = b\} \\ & [a \simeq b \rightarrow a(v \mid u)] \cup [b \preceq a \rightarrow b(v \mid au)] \cup [a \preceq b \rightarrow a(bv \mid u)] \\ = & \{\text{induction: } v \mid u = u \mid v,\; v \mid au = au \mid v,\; bv \mid u = u \mid bv\} \\ & [a \simeq b \rightarrow a(u \mid v)] \cup [b \preceq a \rightarrow b(au \mid v)] \cup [a \preceq b \rightarrow a(u \mid bv)] \\ = & \{\text{rearranging the terms around set union}\} \\ & au \mid bv \end{array}$$

Observation 20 $\epsilon \in (u \mid v)$, for any u and v.

Proof: If either u or v is empty, the result follows from definition. For $au \mid bv$, $\neg((a \simeq b) \wedge (a \preceq b))$; so, at least one of these conditions is *false*, and the corresponding guarded set contributes $\{\epsilon\}$.

We can prove a much stronger result, that $u \mid v$ is (non-empty and) prefix-closed. We do not need this result in developing the theory.

Observation 21 $[false \;\rightarrow\; S] \cup [p \;\rightarrow\; u \mid v] = [p \;\rightarrow\; u \mid v]$

Proof: This follows from $[false \;\rightarrow\; S] = \{\epsilon\}$ and that $u \mid v$ includes ϵ, from (Observation 20, page 77).

This observation allows us to simplify expressions by dropping terms whose guards are *false*, provided that one of the sets that is retained contains ϵ.

3.2.2 Symmetric Composition Preserves Breadth

We show that for broad sets U and V, $U \mid V$ is broad.

Lemma 31 Given $s \leq t$, $A(s) \mid A(t) = A(s)$.

Proof: Let u be a sequence of substitutions all at some time r and v a sequence of substitutions all at time r'. Prove by induction that $u \mid v$ is the set of common prefixes of u and v. Considering the sequences in $A(s)$ and $A(t)$, $A(s) \mid A(t) = A(s)$.

Corollary 5 $A(t) \mid A(t) = A(t)$.
$\beta(\epsilon) \mid \beta(\epsilon) = \beta(\epsilon)$

Corollary 6 Let U and V be broad sets, and $a_t \in U$ and $b_t \in V$, for some a and b. Then, $A(t) \subseteq U \mid V$.

Proof:

$$
\begin{array}{ll}
 & a_t \in U \\
\Rightarrow & \{U \text{ broad}\} \\
 & \beta(a_t) \subseteq U \\
\Rightarrow & \{A(t) \subseteq \beta(a_t), \text{ by the definition of } \beta()\} \\
 & A(t) \subseteq U \\
\Rightarrow & \{\text{similarly, } A(t) \subseteq V\} \\
 & A(t) \subseteq U,\ A(t) \subseteq V \\
\Rightarrow & \{\text{apply } \mid\ \} \\
 & A(t) \mid A(t) \subseteq U \mid V \\
\Rightarrow & \{A(t) \mid A(t) = A(t), \text{ from Corollary 5, page 78}\} \\
 & A(t) \subseteq U \mid V
\end{array}
$$

Lemma 32 $(U \mid V)\backslash a = U\backslash a \mid V\backslash a$, where a is a substitution.

Proof: Since $\backslash a$ is coercive, it is sufficient to prove that $(u \mid v)\backslash a = u\backslash a \mid v\backslash a$.

If both u and v do not start with a, then $u \mid v$ does not start with a, from the definition of $\mid$ and that a is a substitution. Then $(u \mid v)\backslash a = \phi$. Also, at least one of $u\backslash a$ and $v\backslash a$ is ϕ, so $u\backslash a \mid v\backslash a = \phi$.

If both u and v start with a, then $u = ap_t$ and $v = aq_t$, where $t = a.time$.

$$
\begin{array}{ll}
 & (u \mid v)\backslash a \\
= & \{u = ap_t \text{ and } v = aq_t\} \\
 & (ap_t \mid aq_t)\backslash a \\
= & \{\text{from definition of } \mid,\ ap_t \mid aq_t = a(p_t \mid q_t) = a(p \mid q)_t\} \\
 & (a(p \mid q)_t)\backslash a \\
= & \{\text{definition of } \backslash a\} \\
 & p \mid q \\
= & \{u = ap_t \text{ and } v = aq_t. \text{ So, } \{p\} = u\backslash a \text{ and } \{q\} = v\backslash a\} \\
 & u\backslash a \mid v\backslash a
\end{array}
$$

Lemma 33 Given U and V broad, a a base event at t, $a \in U$, and $b_t \in V$ for some event b. Then, $u \in U \backslash a \mid V_{-t} \;\Rightarrow\; au_t \in U \mid V$.

Proof: Proof is by case analysis on u.

• $u = \epsilon$ We have to show that $a \in U \mid V$.

$$
\begin{array}{ll}
 & a \in U \\
\Rightarrow & \{b_t \in V\} \\
 & a \mid b_t \subseteq U \mid V \\
\Rightarrow & \{\text{from definition of } \mid, [a \preceq b_t \to a(\epsilon \mid b_t)] \subseteq a \mid b_t\} \\
 & [a \preceq b_t \to a(\epsilon \mid b_t)] \subseteq U \mid V \\
\Rightarrow & \{a \preceq b_t \text{ holds given that } a \text{ is base event at } t;\ \epsilon \mid b_t = \{\epsilon\}\} \\
 & a \in U \mid V
\end{array}
$$

• $u \neq \epsilon$ Given $u \in U \backslash a \mid V_{-t}$, $u \in p \mid q$, where $p \in U \backslash a$, and $q \in V_{-t}$. Neither p nor q is empty because u in non-empty.

$$
\begin{array}{ll}
 & u \in p \mid q \\
\Rightarrow & \{\text{apply time-shift}\} \\
 & u_t \in p_t \mid q_t \\
\Rightarrow & \{\text{concatenation}\} \\
 & au_t \in a(p_t \mid q_t) \\
\Rightarrow & \{\text{let } q = cr. \text{ Then, } ap_t \mid q_t = ap_t \mid c_t r_t \supseteq \{a \preceq c_t\}\ a(p_t \mid c_t r_t) = a(p_t \mid q_t)\} \\
 & au_t \in ap_t \mid q_t \\
\Rightarrow & \{p \in U \backslash a \Rightarrow ap_t \in U;\ q \in V_{-t} \Rightarrow q_t \in V\} \\
 & au_t \in U \mid V
\end{array}
$$

Theorem 17 Given that U and V are broad, $U \mid V$ is broad.

Proof: If either of U or V is the empty set, then $U \mid V$ is the empty set, which is broad. Now, assume that both sets are non-empty. We show for any u and v, where $u \in U$ and $v \in V$, that $\beta(u \mid v) \subseteq U \mid V$. Then from Corollary 4, page 70, $U \mid V$ is broad.

The proof of $\beta(u \mid v) \subseteq U \mid V$ is by induction on the combined length of u and v.

• u or v is empty: Then $u \mid v = \{\epsilon\}$, and we have to show $\beta(\epsilon) \subseteq U \mid V$. From Observation 17, page 68, using $\beta(U) = U$,

$$
\begin{array}{ll}
 & \beta(\epsilon) \subseteq U \\
\Rightarrow & \{\text{similarly with } V\} \\
 & \beta(\epsilon) \subseteq U,\ \beta(\epsilon) \subseteq V \\
\Rightarrow & \{\text{taking } \mid\} \\
 & \beta(\epsilon) \mid \beta(\epsilon) \subseteq U \mid V \\
\Rightarrow & \{\beta(\epsilon) \mid \beta(\epsilon) = \beta(\epsilon), \text{ from Corollary 5, page 78}\} \\
 & \beta(\epsilon) \subseteq U \mid V
\end{array}
$$

• $au_t \in U$ and $av_t \in V$, where $a.time = t$ and $a \simeq b$: we show $\beta(au_t \mid av_t) \subseteq U \mid V$.

$$\begin{array}{ll} & \beta(au_t \mid av_t) \\ = & \{\text{definition of } \mid\} \\ & \beta(a(u \mid v)_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup a\beta(u \mid v)_t \end{array}$$

Now, $A(t) \subseteq U \mid V$ follows from Corollary 6, page 78, because $a \in U$, $a \in V$, and $a.time = t$. We show $a\beta(u \mid v)_t \subseteq U \mid V$.

$$\begin{array}{ll} & au_t \in U,\ av_t \in V \\ \Rightarrow & \{\text{definition}\} \\ & u \in U\backslash a,\ v \in V\backslash a \\ \Rightarrow & \{\text{apply } \mid\} \\ & u \mid v \subseteq U\backslash a \mid V\backslash a \\ \Rightarrow & \{U\backslash a \text{ and } V\backslash a \text{ are broad from Lemma 26, page 73; apply induction}\} \\ & \beta(u \mid v) \subseteq U\backslash a \mid V\backslash a \\ \Rightarrow & \{U\backslash a \mid V\backslash a = (U \mid V)\backslash a, \text{ from Lemma 32, page 78}\} \\ & \beta(u \mid v) \subseteq (U \mid V)\backslash a \\ \Rightarrow & \{\text{definition of } (U \mid V)\backslash a\} \\ & a\beta(u \mid v)_t \subseteq U \mid V \end{array}$$

• $au \in U$ and $bv \in V$, where $\neg(a \simeq b)$: we show $\beta(au \mid bv) \subseteq U \mid V$.

$$\begin{array}{ll} & \beta(au \mid bv) \\ = & \{\text{definition of } \mid \text{ given } \neg(a \simeq b)\} \\ & \beta([a \preceq b \rightarrow a(u \mid bv)] \cup [b \preceq a \rightarrow b(au \mid v)] \cup \{\epsilon\}) \\ = & \{\beta() \text{ distributes over set union and guarded sets}\} \\ & [a \preceq b \rightarrow \beta(a(u \mid bv))] \cup [b \preceq a \rightarrow \beta(b(au \mid v))] \cup \beta(\epsilon) \end{array}$$

In the earlier proof with $u = \epsilon$ or $v = \epsilon$, we showed $\beta(\epsilon) \subseteq U \mid V$. The remaining two terms are symmetric in au and bv, using commutativity of $\mid$. Therefore, it is sufficient to show that $[a \preceq b \rightarrow \beta(a(u \mid bv))] \subseteq U \mid V$.

If $\neg(a \preceq b)$, then $[a \preceq b \rightarrow \beta(a(u \mid bv))] = \{\epsilon\}$, which is trivially in $U \mid V$. Assume $a \preceq b$. We rename the terms as au_t and b_tv_t. Our goal is to show $\beta(a(u_t \mid b_tv_t)) \subseteq U \mid V$, where $t = a.time$, $au_t \in U$, $b_tv_t \in V$.

$$\begin{array}{ll} & \beta(a(u_t \mid b_tv_t)) \\ = & \{\text{distribute time-shift}\} \\ & \beta(a(u \mid bv)_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup a\beta(u \mid bv)_t \end{array}$$

From Corollary 6, page 78, $A(t) \subseteq U \mid V$. The remaining task is to show $a\beta(u \mid bv)_t \subseteq U \mid V$.

$$
\begin{array}{ll}
& au_t \in U, \text{ and } b_t v_t \in V \\
\Rightarrow & \{\text{definitions}\} \\
& u \in U \backslash a,\ bv \in V_{-t} \\
\Rightarrow & \{U \backslash a \text{ is broad, from Lemma 26, page 73,} \\
& \quad \text{given } b_t v_t \in V, \text{ and } V \text{ broad, from Lemma 27, page 74, } V_{-t} \text{ is broad,} \\
& \quad \text{apply induction (combined length of } u \text{ and } bv \text{ is less than } au_t \text{ and } b_t v_t)\} \\
& \beta(u \mid bv) \subseteq U \backslash a \mid V_{-t} \\
\Rightarrow & \{\text{Apply Lemma 33, page 79, for each element in } \beta(u \mid bv): \\
& \quad a \text{ is base, from } a \preceq b, \\
& \quad a \in U, \text{ from } au_t \in U, \text{ and } U \text{ prefix-closed,} \\
& \quad b_t \in V, \text{ from } b_t v_t \in V, \text{ and } V \text{ prefix-closed}\} \\
& a\beta(u \mid bv)_t \subseteq U \mid V
\end{array}
$$

3.2.3 Symmetric Composition Preserves Traces

We show that for broad sets U and V, $\overline{U \mid V} = \overline{U} \mid \overline{V}$.

Lemma 34 Let u and v be visible. Then, $\overline{u \mid v} = \overline{u} \mid \overline{v}$

Proof: We prove the result by induction on the combined length of u and v.

If either u or v is ϵ, both sides are $\{\epsilon\}$, from the definition. Next, we take au and bv which are both visible, and show that $\overline{au \mid bv} = \overline{au} \mid \overline{bv}$. Note that u and v are visible, given au and bv are visible; either or both of u and v may be ϵ.

$$
\begin{array}{ll}
& \overline{au \mid bv} \\
= & \{\text{definition of } au \mid bv\} \\
& \overline{[a \simeq b \to a(u \mid v)] \cup [a \preceq b \to a(u \mid bv)] \cup [b \preceq a \to b(au \mid v)]} \\
= & \{\text{distribute trace over set union, guarded sets and concatenation}\} \\
& [a \simeq b \to \overline{a}(\overline{u \mid v})] \cup [a \preceq b \to \overline{a}(\overline{u \mid bv})] \cup [b \preceq a \to \overline{b}(\overline{au \mid v})] \\
= & \{\text{induction. Note that } au,\ bv,\ u \text{ and } v \text{ are visible}\} \\
& [a \simeq b \to \overline{a}(\overline{u} \mid \overline{v})] \cup [a \preceq b \to \overline{a}(\overline{u} \mid \overline{bv})] \cup [b \preceq a \to \overline{b}(\overline{au} \mid \overline{v})] \\
= & \{\text{distribute trace over concatenation}\} \\
& [a \simeq b \to \overline{a}(\overline{u} \mid \overline{v})] \cup [a \preceq b \to \overline{a}(\overline{u} \mid \overline{b}\,\overline{v})] \cup [b \preceq a \to \overline{b}(\overline{a}\,\overline{u} \mid \overline{v})] \quad (*)
\end{array}
$$

We show that $\overline{au \mid bv} = \overline{au} \mid \overline{bv}$ for each of these cases: (1) $a, b = \tau, \tau$, (2) $a \neq \tau$ and $b \neq \tau$ (3) $a \neq \tau$ and $b = \tau$, (4) $a = \tau$ and $b \neq \tau$.

Case 1) $a, b = \tau, \tau$: $\overline{au} \mid \overline{bv} = \overline{u} \mid \overline{v}$

$$
\begin{array}{ll}
& \overline{au \mid bv} \\
= & \{\text{use } \neg(a \simeq b) \text{ in } (*) \text{ since } a, b = \tau, \tau; \text{ apply Observation 21, page 77}\} \\
& [a \preceq b \to (\overline{u} \mid \overline{v})] \cup [b \preceq a \to (\overline{u} \mid \overline{v})] \\
= & \{a, b = \tau, \tau, \text{ so } a \preceq b \equiv a.time \leq b.time, \text{ and } b \preceq a \equiv b.time \leq a.time\} \\
& [a.time \leq b.time \ \longrightarrow\ (\overline{u} \mid \overline{v})] \cup [b.time \leq a.time \ \longrightarrow\ (\overline{u} \mid \overline{v})] \\
= & \{\text{condition in at least one of the guarded sets applies and } \epsilon \in \overline{u} \mid \overline{v}\} \\
& \overline{u} \mid \overline{v}
\end{array}
$$

Case 2) $a \neq \tau$ and $b \neq \tau$:

$$\begin{array}{rl} & \overline{au} \mid \overline{bv} \\ = & \{\text{distribute trace over concatenation}\} \\ & a\overline{u} \mid b\overline{v} \\ = & \{\text{apply definition}\} \\ & [a \simeq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [a \preceq b \rightarrow a(\overline{u} \mid b\overline{v})] \cup [b \preceq a \rightarrow b(a\overline{u} \mid \overline{v})] \end{array}$$

And, this matches (*) given $\overline{a} = a$ and $\overline{b} = b$.

Case 3) $a \neq \tau$ and $b = \tau$:
Then, $\overline{au} \mid \overline{bv} = a\overline{u} \mid \overline{v}$ (L)
And,

$$\begin{array}{rl} & \overline{au \mid bv} \\ = & \{\text{use } \neg(a \simeq b) \text{ in (*) since } b = \tau\} \\ & [a \preceq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [b \preceq a \rightarrow (a\overline{u} \mid \overline{v})] \cup \{\epsilon\} \\ = & \{\text{drop } \{\epsilon\}\text{, using Observation 21, page 77, on the second term}\} \\ & [a \preceq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [b \preceq a \rightarrow (a\overline{u} \mid \overline{v})] \qquad \text{(R)} \end{array}$$

Since bv is visible and $b = \tau$, v is non-empty and visible. Therefore $\overline{v} \neq \epsilon$. Let c be the first event of $\overline{v}$. Then, considering bv, $b.time \leq c.time$. We show (L) = (R) for 3 cases: (1) $a.time < b.time$, (2) $a.time = b.time$, and (3) $a.time > b.time$.

Case 3.1) $a.time < b.time$:

$$\begin{array}{rl} & \text{(L)} = a\overline{u} \mid \overline{v} \\ = & \{a.time < b.time \leq c.time\text{. Hence, } \neg(a \simeq c) \text{ and } \neg(c \preceq a) \\ & \quad \text{Apply definition of } \mid \} \\ & [a \preceq c \rightarrow a(\overline{u} \mid \overline{v})] \cup \{\epsilon\} \\ = & \{a.time \leq c.time \text{ means } a \preceq c \equiv a \text{ is base }\} \\ & [a \text{ is base } \longrightarrow a(\overline{u} \mid \overline{v})] \cup \{\epsilon\} \end{array}$$

And,

$$\begin{array}{rl} & \text{(R)} = [a \preceq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [b \preceq a \rightarrow (a\overline{u} \mid \overline{v})] \\ = & \{a.time < b.time\text{. So, } \neg(b \preceq a)\text{. Also, } a \preceq b \equiv a \text{ is base }\} \\ & [a \text{ is base } \longrightarrow a(\overline{u} \mid \overline{v})] \cup \{\epsilon\} \\ = & \{\text{from above derivation}\} \\ & \text{(L)} \end{array}$$

Case 3.2) $a.time = b.time$:

$$\begin{array}{rl} & \text{(R)} = [a \preceq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [b \preceq a \rightarrow (a\overline{u} \mid \overline{v})] \\ = & \{\text{from } a.time = b.time,\ a \preceq b \equiv a \text{ is base and } b \preceq a \text{ holds given } b = \tau\} \end{array}$$

$$
\begin{array}{rl}
 & [a \text{ is base } \;\longrightarrow\; a(\overline{u} \mid \overline{v})] \cup (a\overline{u} \mid \overline{v}) \\
= & \{\text{If } a \text{ is not base :} \\
 & \quad [a \text{ is base } \;\longrightarrow\; a(\overline{u} \mid \overline{v})] \cup (a\overline{u} \mid \overline{v}) = \{\epsilon\} \cup (a\overline{u} \mid \overline{v}) = (a\overline{u} \mid \overline{v}) \\
 & \text{If } a \text{ is base :} \\
 & \quad [a \text{ is base } \;\longrightarrow\; a(\overline{u} \mid \overline{v})] \cup (a\overline{u} \mid \overline{v}) = a(\overline{u} \mid \overline{v}) \cup (a\overline{u} \mid \overline{v}) \\
 & \quad a.time \leq c.time \text{ implies, from } \mid \text{ definition, } a\overline{u} \mid \overline{v} \supseteq a(\overline{u} \mid \overline{v}) \\
 & \text{In all cases, } [a \text{ is base } \;\longrightarrow\; a(\overline{u} \mid \overline{v})] \cup (a\overline{u} \mid \overline{v}) = (a\overline{u} \mid \overline{v})\} \\
 & a\overline{u} \mid \overline{v} = \text{(L)}
\end{array}
$$

Case 3.3) $a.time > b.time$:

$$
\begin{array}{rl}
 & \text{(R)} = [a \preceq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [b \preceq a \rightarrow (a\overline{u} \mid \overline{v})] \\
= & \{\neg(a \preceq b) \text{ and } b \preceq a \text{ hold}\} \\
 & (a\overline{u} \mid \overline{v}) \cup \{\epsilon\} \\
= & \{\text{from Observation 20, page 77, } \{\epsilon\} \subseteq (a\overline{u} \mid \overline{v})\} \\
 & a\overline{u} \mid \overline{v} = \text{(L)}
\end{array}
$$

Lemma 35 Let u and v be visible. Then, $\overline{u \mid v} = \overline{u} \mid \overline{v}$

Proof: We prove the result by induction on the combined length of u and v.

If either u or v is ϵ, both sides are $\{\epsilon\}$, from the definition. Next, we take au and bv which are both visible, and show that $\overline{au \mid bv} = \overline{au} \mid \overline{bv}$. Note that u and v are visible, given au and bv are visible; either or both of u and v may be ϵ.

$$
\begin{array}{rl}
 & \overline{au \mid bv} \\
= & \{\text{definition of } au \mid bv\} \\
 & \overline{[a \simeq b \rightarrow a(u \mid v)] \cup [a \preceq b \rightarrow a(u \mid bv)] \cup [b \preceq a \rightarrow b(au \mid v)]} \\
= & \{\text{distribute trace over set union, guarded sets and concatenation}\} \\
 & [a \simeq b \rightarrow \overline{a}(\overline{u \mid v})] \cup [a \preceq b \rightarrow \overline{a}(\overline{u \mid bv})] \cup [b \preceq a \rightarrow \overline{b}(\overline{au \mid v})] \\
= & \{\text{induction. Note that } au,\ bv,\ u \text{ and } v \text{ are visible}\} \\
 & [a \simeq b \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup [a \preceq b \rightarrow \overline{a}(\overline{u} \mid \overline{bv})] \cup [b \preceq a \rightarrow \overline{b}(\overline{au} \mid \overline{v})] \quad \text{(R)}
\end{array}
$$

We show that $\overline{au \mid bv} = \overline{au} \mid \overline{bv}$ for each of these cases: (1) $a \neq \tau$ and $b \neq \tau$ (2) $b = \tau$, (3) and $a = \tau$.

Case 1) $a \neq \tau$ and $b \neq \tau$:

$$
\begin{array}{rl}
 & \overline{au} \mid \overline{bv} \\
= & \{\text{distribute trace over concatenation}\} \\
 & a\overline{u} \mid b\overline{v} \\
= & \{\text{apply definition}\} \\
 & [a \simeq b \rightarrow a(\overline{u} \mid \overline{v})] \cup [a \preceq b \rightarrow a(\overline{u} \mid b\overline{v})] \cup [b \preceq a \rightarrow b(a\overline{u} \mid \overline{v})]
\end{array}
$$

And, this matches (R) given $\overline{a} = a$ and $\overline{b} = b$.

Case 2) $b = \tau$: Since bv is visible and $b = \tau$, it follows that v is non-empty and visible. Let $v = cv'$. Then, considering bv, $b.time \leq c.time$.

$$
\begin{array}{ll}
 & \overline{au} \mid \overline{bv} \\
= & \{b = \tau\} \\
 & \overline{au} \mid \overline{v} \qquad \text{(L1)} \\
= & \{\text{induction}\} \\
 & \overline{au \mid v} \\
= & \{v = cv';\ \text{expand } \overline{au \mid cv'} \text{ from (R) replacing } b \text{ by } c \text{ and } v \text{ by } v'\} \\
 & [a \simeq c \rightarrow \overline{a}(\overline{u} \mid \overline{v'})] \cup [a \preceq c \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup [c \preceq a \rightarrow \overline{c}(\overline{au} \mid \overline{v'})] \quad \text{(L2)}
\end{array}
$$

We consider two cases and show that (R) equals (L1) or (L2) in each case.

Case 2.1) $b.time \leq a.time$:

$$
\begin{array}{ll}
 & \text{(R)} \\
= & \{\neg(a \simeq b) \text{ from } b = \tau, \text{ and } b \preceq a \text{ using } b.time \leq a.time\} \\
 & [a \preceq b \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup (\overline{au} \mid \overline{v}) \cup \{\epsilon\} \\
= & \{ \text{ given } b.time \leq c.time,\ a \preceq b \Rightarrow a \preceq c; \\
 & \quad \text{so, } [a \preceq b \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \subseteq [a \preceq c \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup \{\epsilon\}; \\
 & \quad \text{from (L1,L2), } [a \preceq c \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \subseteq (\overline{au} \mid \overline{v}); \\
 & \quad \text{so, } [a \preceq b \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \subseteq (\overline{au} \mid \overline{v}) \cup \{\epsilon\}\} \\
 & (\overline{au} \mid \overline{v}) \cup \{\epsilon\} \\
= & \{\text{from Observation 20, page 77, } \{\epsilon\} \subseteq (\overline{au} \mid \overline{v})\} \\
 & \text{(L1)}
\end{array}
$$

Case 2.2) $\neg(b.time \leq a.time)$:

$$
\begin{array}{ll}
 & \text{(R)} \\
= & \{\text{given } a.time < b.time. \text{ So, } \neg(a \simeq b),\ \neg(b \preceq a)\} \\
 & [a \preceq b \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup \{\epsilon\} \\
= & \{\text{given } a.time < b.time \leq c.time,\ a \preceq c \equiv \{a \text{ is base}\}\ a \preceq b\} \\
 & [a \preceq c \rightarrow \overline{a}(\overline{u} \mid \overline{v})] \cup \{\epsilon\} \\
= & \{\text{given } a.time < b.time \leq c.time. \text{ So, } \neg(a \simeq c),\ \neg(c \preceq a)\} \\
 & \text{(L2)}
\end{array}
$$

Case 3) $a = \tau$: Similar to case (2).

Lemma 36 $u \mid v \subseteq uc \mid vd$, for any events c and d.

Proof: : We prove $u \mid v \subseteq uc \mid v$. Similarly, using commutativity, it can be shown that $u \mid v \subseteq u \mid vd$. Then, $u \mid v \subseteq uc \mid v \subseteq uc \mid vd$.

Now, we prove $u \mid v \subseteq uc \mid v$ by induction on the combined length of u and v.

- $u = \epsilon$ or $v = \epsilon$: Left side is $\{\epsilon\}$, and the right side, being a merge, contains ϵ, from Observation 20, page 77.

- $au \mid bv \subseteq auc \mid bv$:

$$\begin{array}{ll}
 & au \mid bv \\
= & \{\text{definition of } \mid \} \\
 & [a \simeq b \to a(u \mid v)] \cup [a \preceq b \to a(u \mid bv)] \cup [b \preceq a \to b(au \mid v)] \\
\subseteq & \{\text{induction: } u \mid v \subseteq uc \mid v\text{; similarly for the other terms}\} \\
 & [a \simeq b \to a(uc \mid v)] \cup [a \preceq b \to a(uc \mid bv)] \cup [b \preceq a \to b(auc \mid v)] \\
= & \{\text{definition of } \mid \} \\
 & auc \mid bv
\end{array}$$

Theorem 18 Let U and V be broad sets. Then $\overline{U \mid V} = \overline{U} \mid \overline{V}$.

Proof: The proof is in two parts: $\overline{U \mid V} \subseteq \overline{U} \mid \overline{V}$, and $\overline{U} \mid \overline{V} \subseteq \overline{U \mid V}$.

- $\overline{U \mid V} \subseteq \overline{U} \mid \overline{V}$: For any u in U and v in V. We show $\overline{u \mid v} \subseteq \overline{U} \mid \overline{V}$.

$$\begin{array}{ll}
 & \overline{u \mid v} \\
\subseteq & \{\text{from Lemma 23, page 71, there is } uc \in \beta(u) \subseteq U \text{ and } vd \in V \\
 & \quad \text{where } c \text{ and } d \text{ are substitutions} \\
 & \quad \text{apply Lemma 36, page 84}\} \\
 & \overline{(uc \mid vd)} \\
= & \{uc \text{ and } vd \text{ are visible; apply Lemma 34, page 81}\} \\
 & (\overline{uc} \mid \overline{vd}) \\
\subseteq & \{\text{from } uc \in U,\ \overline{uc} \in \overline{U}\text{; similarly, } \overline{vd} \in \overline{V}\} \\
 & \overline{U} \mid \overline{V}
\end{array}$$

- $\overline{U} \mid \overline{V} \subseteq \overline{U \mid V}$: We show that for u in U and v in V, $\overline{u} \mid \overline{v} \subseteq \overline{U \mid V}$.

Let p be the longest visible prefix of u and q of v. Then $\overline{u} = \overline{p}$ and $\overline{v} = \overline{q}$. From prefix-closure, $p \in U$ and $q \in V$.

$$\begin{array}{ll}
 & \overline{u} \mid \overline{v} \\
= & \{\text{from above}\} \\
 & \overline{p} \mid \overline{q} \\
= & \{p \text{ and } q \text{ are visible}\} \\
 & \overline{p \mid q} \\
\subseteq & \{p \in U \text{ and } q \in V\} \\
 & \overline{U \mid V}
\end{array}$$

3.3 Sequential Composition

We use the definition of $>x>$ applied to sets, as given in Section 2.2.2, page 37.

3.3.1 Preliminary Results

Lemma 37 For p that has no publication and $V \neq \phi$, $p >x> V = \{p\}$.

Proof: By induction on the length of p.

Corollary 7 $A(t) >x> V = A(t)$, for $V \neq \phi$.

Proof: From above Lemma, because no sequence in $A(t)$ has a publication.

Notational Simplification Define

$$\hat{a} = \begin{cases} a & \text{if } a \text{ is not a publication} \\ (t, \tau) & \text{if } a \text{ is a publication at time } \ t \end{cases}$$

Then, replacing p by p_t in the definition of sequential composition, and using Lemma 8, page 45, $ap_t >x> V = \hat{a}P_t$, where P is

$$\text{(SCD)} \begin{cases} p >x> V & \text{if } c_1(a) \\ p >x> V' & \text{if } c_2(a) \\ p >x> V \mid V'' & \text{if } c_3(a) \end{cases}$$

3.3.2 Sequential Composition Preserves Breadth

Theorem 19 For broad sets P and V, $P >x> V$ is broad.

Proof: We have $P >x> \phi = \phi$, from Lemma 2.2.4, page 38, and ϕ is broad, vacuously. Assume, henceforth, that $V \neq \phi$. We prove below that for any sequence q, $\beta(q) >x> V$ is broad. Then,

$$\begin{array}{ll} & P >x> V \\ = & \{P \text{ is broad; so, } P = \beta(P)\} \\ & \beta(P) >x> V \\ = & \{\beta(P) = (\cup q: \ q \in P: \ \beta(q))\} \\ & (\cup q: \ q \in P: \ \beta(q)) >x> V \\ = & \{\text{distribute } >x> \text{ over set union}\} \\ & (\cup q: \ q \in P: \ \beta(q) >x> V) \end{array}$$

Each $\beta(q) >x> V$ is broad (to be shown) and union of broad sets is broad, from Corollary 4, page 70. Hence the result.

We prove $\beta(q) >x> V$ is broad by induction on the length of q.

• $q = \epsilon$:
$\beta(\epsilon) >x> V = A(0) >x> V = A(0)$, from Corollary 7, page 85. And, $A(0)$ is broad, from Lemma 18, page 68.

• $q = ap_t$, where $a.time = t$:

$$\begin{array}{ll} & \beta(ap_t) >x> V \\ = & \{\text{definition of breadth}\} \\ & (A(t) \cup a\beta(p)_t) >x> V \\ = & \{\text{distribute } >x> \text{ over set union}\} \\ & (A(t) >x> V) \cup (a\beta(p)_t >x> V) \\ = & \{A(t) >x> V = A(t)\} \\ & A(t) \cup (a\beta(p)_t >x> V) \\ = & \{\text{replace } p \text{ by } \beta(p) \text{ in (SCD), page 86, to get } U\} \end{array}$$

$$
\begin{array}{ll}
 & A(t) \cup \hat{a}U_t \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (*1) \\
= & \{\text{from induction hypothesis} \\
 & \quad \beta(p) >x> V \text{ is broad, given } V \text{ is broad} \\
 & \quad \beta(p) >x> V' \text{ is broad, because } V' \text{ is broad from Lemma 26, page 73} \\
 & \quad \beta(p) >x> V \mid V'' \text{ is broad, because } V'' \text{ is broad, from Lemma 26, page 73, and} \\
 & \quad \text{merge of two broad sets is broad, from Theorem 17, page 79;} \\
 & \quad \text{therefore, } U \text{ is broad; hence } U = \beta(U)\} \\
 & A(t) \cup \hat{a}\beta(U)_t \\
= & \{\text{definition of } \beta(); \text{ note that } \hat{a}.time = t\} \\
 & \beta(\hat{a}U_t)
\end{array}
$$

It follows from Corollary 4, page 70 Part (1), that $\beta(ap_t) >x> V$ is broad since it is $\beta(Q)$, for some Q.

Lemma 38 *$(U \backslash a >x> V) = (U >x> V)\backslash a$, where a is an own substitution.*

Proof: We show for arbitrary $u \in U$ that $(u\backslash a >x> V) = (u >x> V)\backslash a$. The result then follows by coercion.

If $u = \epsilon$, then

$$
\begin{array}{ll}
 & (\epsilon\backslash a >x> V) \\
= & \{\text{definition of } \backslash\} \\
 & (\emptyset >x> V) \\
= & \{\text{coercion}\} \\
 & \emptyset,
\end{array}
$$

and

$$
\begin{array}{ll}
 & (\epsilon >x> V)\backslash a \\
= & \{\text{definition of } \epsilon >x> V\} \\
 & \{\epsilon\}\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset.
\end{array}
$$

Next suppose $u = bu'$ and $b \neq a$. Then

$$
\begin{array}{ll}
 & (bu'\backslash a >x> V) \\
= & \{\text{definition of } \backslash, \, b \neq a\} \\
 & (\emptyset >x> V) \\
= & \{\text{coercion}\} \\
 & \emptyset,
\end{array}
$$

Then if b is an other-substitution:

$$
\begin{array}{ll}
 & (bu' >x> V)\backslash a \\
= & \{\text{condition } c_2 \text{ holds}\} \\
 & (b(u' >x> V\backslash b))\backslash a \\
= & \{\text{definition of } \backslash, \, b \neq a\} \\
 & \emptyset.
\end{array}
$$

If b is a non-publication base event:

$$\begin{array}{cl} & (bu' >x> V)\backslash a \\ = & \{\text{condition } c_1 \text{ holds}\} \\ & (b(u' >x> V))\backslash a \\ = & \{\text{definition of } \backslash,\ b \neq a\} \\ & \emptyset. \end{array}$$

Finally, if b is a publication event $(t, !m)$:

$$\begin{array}{cl} & ((t, !m)u' >x> V)\backslash a \\ = & \{\text{condition } c_3 \text{ holds}\} \\ & ((t,\tau)(u' >x> V \mid (V\backslash a)_t))\backslash a \\ = & \{\text{definition of } \backslash\} \\ & \emptyset. \end{array}$$

Otherwise, suppose $b = a$.

$$\begin{array}{cl} & (au'\backslash a >x> V) \\ = & \{\text{definition of } \backslash\} \\ & (u' >x> V) \end{array}$$

and

$$\begin{array}{cl} & (au' >x> V)\backslash a \\ = & \{\text{condition } c_1 \text{ holds}\} \\ & (a(u' >x> V))\backslash a \\ = & \{\text{definition of } \backslash\} \\ & (u' >x> V). \end{array}$$

Lemma 39 $(U\backslash a >x> V\backslash a) = (U >x> V)\backslash a$, *where a is an other substitution.*

Proof: We show for arbitrary $u \in U$ that $(u\backslash a >x> V\backslash a) = (u >x> V)\backslash a$. The result then follows by coercion.

If $u = \epsilon$, then

$$\begin{array}{cl} & (\epsilon\backslash a >x> V\backslash a) \\ = & \{\text{definition of } \backslash\} \\ & (\emptyset >x> V\backslash a) \\ = & \{\text{coercion}\} \\ & \emptyset, \end{array}$$

and

$$\begin{array}{cl} & (\epsilon >x> V)\backslash a \\ = & \{\text{definition of } \epsilon >x> V\} \\ & \{\epsilon\}\backslash a \\ = & \{\text{definition of } \backslash\} \\ & \emptyset. \end{array}$$

Next suppose $u = bu'$ and $b \neq a$. Then

$$
\begin{array}{cl}
 & (bu' \backslash a >x> V \backslash a) \\
= & \{\text{definition of } \backslash,\ b \neq a\} \\
 & (\emptyset >x> V) \\
= & \{\text{coercion}\} \\
 & \emptyset,
\end{array}
$$

Then if b is an other-substitution:

$$
\begin{array}{cl}
 & (bu' >x> V) \backslash a \\
= & \{\text{condition } c_2 \text{ holds}\} \\
 & (b(u' >x> V \backslash b)) \backslash a \\
= & \{\text{definition of } \backslash,\ b \neq a\} \\
 & \emptyset.
\end{array}
$$

If b is an own-substitution or a non-publication base event:

$$
\begin{array}{cl}
 & (bu' >x> V) \backslash a \\
= & \{\text{condition } c_1 \text{ holds}\} \\
 & (b(u' >x> V)) \backslash a \\
= & \{\text{definition of } \backslash,\ b \neq a\} \\
 & \emptyset.
\end{array}
$$

Finally, if b is a publication event $(t, !m)$:

$$
\begin{array}{cl}
 & ((t, !m)u' >x> V) \backslash a \\
= & \{\text{condition } c_3 \text{ holds}\} \\
 & ((t, \tau)(u' >x> V \mid (V \backslash [m/x])_t)) \backslash a \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset.
\end{array}
$$

Otherwise, suppose $b = a$.

$$
\begin{array}{cl}
 & (au' \backslash a >x> V \backslash a) \\
= & \{\text{definition of } \backslash\} \\
 & (u' >x> V \backslash a)
\end{array}
$$

and

$$
\begin{array}{cl}
 & (au' >x> V) \backslash a \\
= & \{\text{condition } c_2 \text{ holds}\} \\
 & (a(u' >x> V \backslash a)) \backslash a \\
= & \{\text{definition of } \backslash\} \\
 & (u' >x> V \backslash a).
\end{array}
$$

3.3.3 Sequential Composition Preserves Traces

Theorem 20 For broad P and V, where V is substitution independent (see page 20), $\overline{P >x> V} = \overline{\overline{P} >x> \overline{V}}$.

Proof: As in Theorem 19, page 86, it is sufficient to show that for any sequence q and non-empty broad V, $\overline{\beta(q) >x> V} = \overline{(\overline{\beta(q)} >x> \overline{V})}$. Proof is by induction on the length of q.

• $q = \epsilon$:
$\overline{\beta(\epsilon) >x> V}$= {from Corollary 7, page 85} $\overline{\beta(\epsilon)}$, and
$\overline{(\overline{\beta(\epsilon)} >x> \overline{V})}$ = $\{\overline{\beta(\epsilon)} = \beta(\epsilon)\}$ $\overline{\beta(\epsilon) >x> \overline{V}}$ = {Corollary 7, page 85} $\overline{\beta(\epsilon)}$.

• $q = ap_t$, where $a.time = t$: We show $\overline{\beta(ap_t) >x> V} = \overline{\overline{\beta(ap_t)} >x> \overline{V}}$.
First, we prove a sublemma.

Sublemma Let

$$U = \begin{cases} \beta(p) >x> V & \text{if } c_1(a) \\ \beta(p) >x> V' & \text{if } c_2(a) \\ \beta(p) >x> V \mid V'' & \text{if } c_3(a) \end{cases}$$

and

$$W = \begin{cases} \overline{\beta(p)} >x> \overline{V} & \text{if } c_1(a) \\ \overline{\beta(p)} >x> (\overline{V})' & \text{if } c_2(a) \\ \overline{\beta(p)} >x> \overline{V} \mid (\overline{V})'' & \text{if } c_3(a) \end{cases}$$

Then $\overline{W} = \overline{U}$. (*2)
Note: We get U from (SCD), page 86, replacing p by $\beta(p)$. And, W is obtained from U by replacing $\beta(p)$ and V by $\overline{\beta(p)}$ and $\overline{V}$, respectively.

Proof: Observe that

$$\overline{U} = \begin{cases} \overline{\beta(p) >x> V} & \text{if } c_1(a) \\ \overline{\beta(p) >x> V'} & \text{if } c_2(a) \\ \overline{\beta(p) >x> V \mid V''} & \text{if } c_3(a) \end{cases}$$

We consider the three cases for $\overline{U}$, as given above, and show that $\overline{U} = \overline{W}$.

(Subcase 1: $c_1(a)$ holds) Applying induction, $\overline{\beta(p) >x> V} = \overline{\overline{\beta(p)} >x> \overline{V}}$, or $\overline{U} = \overline{W}$.

(Subcase 2: $c_2(a)$ holds)

$$\begin{array}{ll} & \overline{U} \\ = & \{\text{from the definition of } \overline{U} \text{ in the second case}\} \end{array}$$

$\overline{\beta(p) >x> V'}$
= {V' is broad, from Lemma 26, page 73; apply induction}
$\overline{\overline{\beta(p)} >x> \overline{V'}}$
= {given V is substitution independent, from Lemma 3, page 20, $\overline{V'} = \overline{V}'$}
$\overline{\overline{\beta(p)} >x> \overline{V}'}$
= {definition of W}
$\overline{W}$

(Subcase 3: $c_3(a)$ holds)

$\overline{U}$
= {from the definition of $\overline{U}$ in the third case}
$\overline{\beta(p) >x> V \mid V''}$
= {$\beta(p) >x> V$ is broad, from Theorem 19, page 86
V'' is broad, from Lemma 26, page 73
apply Theorem 18, page 85}
$\overline{\beta(p) >x> V} \mid \overline{V''}$
= {given V is substitution independent, from Lemma 3, page 20, $\overline{V''} = \overline{V}''$}
$\overline{\beta(p) >x> V} \mid \overline{V}''$
= {induction on the first term}
$\overline{\overline{\beta(p)} >x> \overline{V}} \mid \overline{V}''$
= {definition of W}
$\overline{W}$ (End of Sublemma)

Next, we simplify $\overline{\beta(ap_t) >x> V}$ and $(\overline{\beta(ap_t)} >x> \overline{V})$.

$\overline{\beta(ap_t) >x> V}$
= {from the derivation in Theorem 19, page 86, see (*1)}
$\overline{A(t) \cup \hat{a}U_t}$, where U is as given in the Sublemma
= {distribute trace over set union, concatenation and time-shift;
$\overline{A(t)} = A(t)$; $\overline{\hat{a}} = \hat{\overline{a}}$}
$A(t) \cup \hat{\overline{a}}\overline{U}_t$ (*3)

Next,

$\overline{\beta(ap_t)} >x> \overline{V}$
= {expanding $\beta(ap_t)$}
$\overline{A(t) \cup a\beta(p)_t} >x> \overline{V}$
= {distribute trace over set union; $\overline{A(t)} = A(t)$}
$(A(t) \cup \overline{a\beta(p)_t}) >x> \overline{V}$
= {distribute $>x>$ over set union; $A(t) >x> \overline{V} = A(t)$}
$A(t) \cup (\overline{a\beta(p)_t} >x> \overline{V})$
= {distribute trace over concatenation and time-shift}
$A(t) \cup ((\overline{a}\,\overline{\beta(p)}_t) >x> \overline{V})$ (*4)

To complete the proof that $\overline{\beta(ap_t) >x> V} = \overline{\overline{\beta(ap_t)} >x> \overline{V}}$, we consider two cases.
Case 1) $a \neq \tau$:

$$
\begin{array}{cl}
 & \overline{\overline{\beta(ap_t)} >x> \overline{V}} \\
= & \{\text{from } (*4)\} \\
 & \overline{A(t) \cup ((\overline{a}\,\overline{\beta(p)}_t) >x> \overline{V})} \\
= & \{a \neq \tau, \text{ so } \overline{a} = a\} \\
 & \overline{A(t) \cup ((a\overline{\beta(p)}_t) >x> \overline{V})} \\
= & \{\text{from (SCD), page 86, where } W \text{ is as given in the Sublemma}\} \\
 & \overline{A(t) \cup \hat{a}W_t} \\
= & \{\overline{A(t)} = A(t),\ \overline{\hat{a}} = \hat{\overline{a}}\} \\
 & A(t) \cup \hat{\overline{a}}\overline{W}_t \\
= & \{\overline{W} = \overline{U}, \text{ from } (*2)\} \\
 & A(t) \cup \hat{\overline{a}}\overline{U}_t \\
= & \{\text{from } (*3)\} \\
 & \overline{\beta(ap_t) >x> V}
\end{array}
$$

Case 2) $a = \tau$:

$$
\begin{array}{cl}
 & \overline{\overline{\beta(ap_t)} >x> \overline{V}} \\
= & \{\text{from } (*4)\} \\
 & \overline{A(t) \cup ((\overline{a}\,\overline{\beta(p)}_t) >x> \overline{V})} \\
= & \{\overline{A(t)} = A(t),\ \overline{a} = \epsilon, \text{ distribute trace over set union}\} \\
 & A(t) \cup \overline{\overline{\beta(p)}_t >x> \overline{V}} \\
= & \{\overline{\overline{\beta(p)}_t >x> \overline{V}} = (\overline{\overline{\beta(p)} >x> \overline{V}})_t, \text{ from Lemma 8, page 45;} \\
 & \quad \text{distribute time-shift over trace}\} \\
 & A(t) \cup (\overline{\overline{\beta(p)} >x> \overline{V}})_t \\
= & \{\text{induction}\} \\
 & A(t) \cup (\overline{\beta(p) >x> V})_t \\
= & \{\text{given } a = \tau, \text{ i.e., } c_1(a), \text{ and definition of } U\} \\
 & A(t) \cup \overline{U}_t \\
= & \{a = \tau\} \\
 & A(t) \cup \hat{\overline{a}}\overline{U}_t \\
= & \{\text{from } (*3)\} \\
 & \overline{\beta(ap_t) >x> V}
\end{array}
$$

3.4 Asymmetric Composition

We use the definition of $<x<$ applied to sets, as given in Section 2.2.3, page 37.

3.4.1 Preliminary Results on Constrained Partial Merge

Observation 22 $\epsilon \in (u|_x v)$, for any u and v.

Proof: If either u or v is empty, the result follows from definition. For $au|_x bv$, $\neg((a \approx_x b) \wedge (a \preceq b))$; so, at least one of these conditions is *false*, and the corresponding guarded set contributes $\{\epsilon\}$.

Observation 23 $[\mathit{false} \;\rightarrow\; S] \cup [p \;\rightarrow\; u|_x v] = [p \;\rightarrow\; u|_x v]$

Proof: This follows from $[\mathit{false} \;\rightarrow\; S] = \{\epsilon\}$ and that $u|_x v$ includes ϵ, from (Observation 22, page 92).

This observation allows us to simplify expressions by dropping terms whose guards are *false*, provided that one of the sets that is retained contains ϵ.

Lemma 40 $(u|_x v)_t = u_t|_x v_t$

Proof: Apply the definition of $|_x$ to both sides. Note that $a_t \approx_x b_t \equiv a \approx_x b$, $a_t \preceq b_t \equiv a \preceq b$ and $b_t \underline{\prec\!\!\prec}_x a_t \equiv b \underline{\prec\!\!\prec}_x a$. The result follows by applying induction on the combined length of u and v.

Lemma 41 $A(t)|_x A(t) = A(t)$.

Proof: Proof is by mutual inclusion.

- $A(t)|_x A(t) \subseteq A(t)$: Let u be a sequence of substitutions, all at time r, and v be a sequence of substitutions, all at time s. If the base rule is used in forming $u|_x v$, then $u|_x v = \{\epsilon\} \subseteq A(t)$. If the inductive rule is used, then each term starts with an event at $\min(r, s)$. Using induction, it can be shown that $u|_x v$ then contains sequences of substitutions all at time $\min(r, s)$. Therefore, $u|_x v \subseteq A(t)$.

- $A(t) \subseteq A(t)|_x A(t)$: Let $p \in A(t)$. We show $u \in A(t)$ and $v \in A(t)$, such that $p \in u|_x v$, $u.\mathit{time} = v.\mathit{time} = p.\mathit{time}$ and either both u and v are empty or neither is. Proof is by induction on the length of p.

Case 1) $p = \epsilon$: Let $u = v = \epsilon$. All the conditions are met.

Case 2) $p = aq$: From the definition of $A(t)$, $q \in A(t)$.
Case 2.1) a is an own-substitution:

If $q = \epsilon$, let $v = a$, $u = c$, where c is an other-substitution at $a.\mathit{time}$. All the conditions are met.

If $q \neq \epsilon$, inductively, $q \in u'|_x v'$, where $u' \in A(t)$, $v' \in A(t)$, $u'.\mathit{time} = v'.\mathit{time} = q.\mathit{time}$; further, since $q \neq \epsilon$ neither u' nor v' is empty. Let $u = u'$ and $v = av'$. Since $aq \in A(t)$, $a.\mathit{time} = q.\mathit{time}$, hence, $a.\mathit{time} = v'.\mathit{time}$, and $av' = v \in A(t)$.

$$
\begin{array}{ll}
 & aq \\
\in & \{q \in u'|_x v'\} \\
 & a(u'|_x v') \\
\subseteq & \{a \text{ is own-substitution, } a.\mathit{time} = u'.\mathit{time}\} \\
 & u'|_x av' \\
= & \{u = u' \text{ and } v = av'\} \\
 & u|_x v
\end{array}
$$

Case 2.2) a is an other-substitution: Inductively, $q \in u'|_x v'$, where $u' \in A(t)$, $v' \in A(t)$, $u'.time = v'.time = q.time$. And either both u' and v' are empty or neither is. Let $u = au'$ and $v = av'$.

If both u' and v' are empty: then $q = \epsilon$. Hence, $p = aq = a \in A(t)$. From the definition of $|_x$, $a \in a|_x a = u|_x v$. The other conditions are met.

If neither of u' and v' is empty: Since $aq \in A(t)$, $a.time = q.time$. Given $q.time = u'.time = v'.time$, we get $a.time = u'.time$ and $a.time = v'.time$. So, $u = au' \in A(t)$ and $v = av' \in A(t)$.

$$
\begin{array}{ll}
 & aq \\
\in & \{q \in u'|_x v'\} \\
 & a(u'|_x v') \\
\subseteq & \{a \text{ is other-substitution}\} \\
 & au'|_x av' \\
= & \{u = au' \text{ and } v = av'\} \\
 & u|_x v
\end{array}
$$

Corollary 8 Let U and V be broad sets, and $a_t \in U$ and $b_t \in V$, for some a and b. Then, $A(t) \subseteq U|_x V$.

Proof:

$$
\begin{array}{ll}
 & a_t \in U \\
\Rightarrow & \{U \text{ broad}\} \\
 & \beta(a_t) \subseteq U \\
\Rightarrow & \{A(t) \subseteq \beta(a_t), \text{ from the definition of } \beta()\} \\
 & A(t) \subseteq U \\
\Rightarrow & \{\text{similarly, } A(t) \subseteq V\} \\
 & A(t) \subseteq U,\ A(t) \subseteq V \\
\Rightarrow & \{\text{apply } |_x\} \\
 & A(t)|_x A(t) \subseteq U|_x V \\
\Rightarrow & \{A(t)|_x A(t) = A(t), \text{ from Lemma 41, page 93}\} \\
 & A(t) \subseteq U|_x V
\end{array}
$$

Lemma 42 $(U|_x V)\backslash a = U\backslash a|_x V\backslash a$, where a is an other-substitution.

Proof: Since $\backslash a$ is coercive, it is sufficient to prove that $(u|_x v)\backslash a = u\backslash a|_x v\backslash a$.

If both u and v do not start with a, then $u|_x v$ does not start with a, from the definition of $|_x$ and that a is an other-substitution. Then $(u|_x v)\backslash a = \phi$. Also, at least one of $u\backslash a$ and $v\backslash a$ is ϕ, so $u\backslash a|_x v\backslash a = \phi$.

If both u and v start with a, then $u = ap_t$ and $v = aq_t$, where $t = a.time$.

$$
\begin{array}{ll}
 & (u|_x v)\backslash a \\
= & \{u = ap_t \text{ and } v = aq_t\} \\
 & (ap_t|_x aq_t)\backslash a \\
= & \{\text{from the definition of } |_x,\ ap_t|_x aq_t = a(p_t|_x q_t) = a(p|_x q)_t\} \\
 & (a(p|_x q)_t)\backslash a \\
= & \{\text{from the definition of } \backslash a; \text{ see Section 3.1.2, page 73}\}
\end{array}
$$

$$
\begin{array}{ll}
 & p|_x q \\
= & \{u = ap_t \text{ and } v = aq_t\text{. So, } \{p\} = u\backslash a \text{ and } \{q\} = v\backslash a\} \\
 & u\backslash a|_x v\backslash a
\end{array}
$$

Lemma 43 Given U and V broad, a is a base event at t, $a \in U$, and $b_t \in V$ for some event b. Then, $u \in U\backslash a|_x V_{-t} \;\Rightarrow\; au_t \in U|_x V$.

Proof: Proof is by case analysis on u.

• $u = \epsilon$: We have to show that $a \in U|_x V$. Given,

$$
\begin{array}{ll}
 & a \in U \\
\Rightarrow & \{b_t \in V\} \\
 & a|_x b_t \subseteq U|_x V \\
\Rightarrow & \{\text{apply the definition of } |_x \text{ to } a|_x b_t, \\
 & \quad [a \preceq b_t \to a(\epsilon|_x b_t)] \subseteq a|_x b_t\} \\
 & [a \preceq b_t \to a(\epsilon|_x b_t)] \subseteq U|_x V \\
\Rightarrow & \{a \preceq b_t \text{ holds given that } a \text{ is base event at } t;\ \epsilon|_x b_t = \{\epsilon\}\} \\
 & a \in U|_x V
\end{array}
$$

• $u \neq \epsilon$: Given $u \in U\backslash a|_x V_{-t}$, we have $u \in p|_x q$, where $p \in U\backslash a$, and $q \in V_{-t}$. Neither p nor q is empty because u in non-empty.

$$
\begin{array}{ll}
 & u \in p|_x q \\
\Rightarrow & \{\text{apply time-shift}\} \\
 & u_t \in p_t|_x q_t \\
\Rightarrow & \{\text{concatenation}\} \\
 & au_t \in a(p_t|_x q_t) \\
\Rightarrow & \{\text{let } q = cr\text{. Then, } ap_t|_x q_t = ap_t|_x c_t r_t \supseteq \{a \preceq c_t\}\ a(p_t|_x c_t r_t) = a(p_t|_x q_t)\} \\
 & au_t \in ap_t|_x q_t \\
\Rightarrow & \{p \in U\backslash a \Rightarrow ap_t \in U;\ q \in V_{-t} \Rightarrow q_t \in V\} \\
 & au_t \in U|_x V
\end{array}
$$

Corollary 9 Given that U and V are broad, b a base event or an own-substitution at t, $b \in V$, $a_t \in U$, for some event a. Then, $u \in U_{-t}|_x V\backslash b \;\Rightarrow\; bu_t \in U|_x V$.

Proof: Similar to Lemma 43, page 95.

Theorem 21 Given that U and V are broad, $U|_x V$ is broad.

Proof: If either of U or V is the empty set, then $U|_x V$ is the empty set, which is broad. Now, assume that both sets are non-empty. We show for any u and v, where $u \in U$ and $v \in V$, that $\beta(u|_x v) \subseteq U|_x V$. Then from Corollary 4, page 70, $U|_x V$ is broad.

The proof of $\beta(u|_x v) \subseteq U|_x V$ is by induction on the combined length of u and v.

• $u = \epsilon$ or $v = \epsilon$: Then $u|_x v = \{\epsilon\}$, and we have to show $\beta(\epsilon) \subseteq U|_x V$. From Observation 17, page 68, using $\beta(U) = U$,

$$
\begin{array}{ll}
 & \beta(\epsilon) \subseteq U \\
\Rightarrow & \{\text{similarly with } V\} \\
 & \beta(\epsilon) \subseteq U,\ \beta(\epsilon) \subseteq V \\
\Rightarrow & \{\text{taking } |_x\} \\
 & \beta(\epsilon)|_x\beta(\epsilon) \subseteq U|_x V \\
\Rightarrow & \{\beta(\epsilon) = A(0);\ \text{from Lemma 41, page 93, } \beta(\epsilon)|_x\beta(\epsilon) = \beta(\epsilon)\} \\
 & \beta(\epsilon) \subseteq U|_x V
\end{array}
$$

• $au_t \in U$ and $av_t \in V$, where $a.time = t$ and $a \approx_x b$: we show $\beta(au_t|_x av_t) \subseteq U|_x V$.

$$
\begin{array}{ll}
 & \beta(au_t|_x av_t) \\
= & \{\text{definition of } |_x\} \\
 & \beta(a(u|_x v)_t) \\
= & \{\text{definition of } \beta()\} \\
 & A(t) \cup a\beta(u|_x v)_t
\end{array}
$$

Now, $A(t) \subseteq U|_x V$ follows from Corollary 8, page 94, because $a \in U$, $a \in V$, and $a.time = t$. We show $a\beta(u|_x v)_t \subseteq U|_x V$.

$$
\begin{array}{ll}
 & au_t \in U,\ av_t \in V \\
\Rightarrow & \{\text{definition}\} \\
 & u \in U\backslash a,\ v \in V\backslash a \\
\Rightarrow & \{\text{apply } |_x\} \\
 & u|_x v \subseteq U\backslash a|_x V\backslash a \\
\Rightarrow & \{U\backslash a \text{ and } V\backslash a \text{ are broad from Lemma 26, page 73; apply induction}\} \\
 & \beta(u|_x v) \subseteq U\backslash a|_x V\backslash a \\
\Rightarrow & \{a \text{ is other-substitution; so, } U\backslash a|_x V\backslash a = (U|_x V)\backslash a, \text{ from Lemma 42, page 94}\} \\
 & \beta(u|_x v) \subseteq (U|_x V)\backslash a \\
\Rightarrow & \{\text{definition of } (U|_x V)\backslash a\} \\
 & a\beta(u|_x v)_t \subseteq U|_x V
\end{array}
$$

• $au \in U$ and $bv \in V$, where $\neg(a \approx_x b)$: we show $\beta(au|_x bv) \subseteq U|_x V$.

$$
\begin{array}{ll}
 & \beta(au|_x bv) \\
= & \{\text{definition of } |_x \text{ given } \neg(a \approx_x b)\} \\
 & \beta([a \preceq b \rightarrow a(u|_x bv)] \cup [b \preceq\!\!\!\prec_x a \rightarrow b(au|_x v)] \cup \{\epsilon\}) \\
= & \{\beta() \text{ distributes over set union and guarded sets}\} \\
 & [a \preceq b \rightarrow \beta(a(u|_x bv))] \cup [b \preceq\!\!\!\prec_x a \rightarrow \beta(b(au|_x v))] \cup \beta(\epsilon)
\end{array}
$$

In the earlier proof with $u = \epsilon$ or $v = \epsilon$, we showed $\beta(\epsilon) \subseteq U|_x V$. We now show that $[a \preceq b \rightarrow \beta(a(u|_x bv))] \subseteq U|_x V$ and $[b \preceq\!\!\!\prec_x a \rightarrow \beta(b(au|_x v))] \subseteq U|_x V$.

Case 1) $[a \preceq b \rightarrow \beta(a(u|_x bv))] \subseteq U|_x V$:
If $\neg(a \preceq b)$, then $[a \preceq b \rightarrow \beta(a(u|_x bv))] = \{\epsilon\}$, which is trivially in $U|_x V$. Assume $a \preceq b$. We rename the terms as au_t and $b_t v_t$. Our goal is to show $\beta(a(u_t|_x b_t v_t)) \subseteq U|_x V$, where $t = a.time$, $au_t \in U$, $b_t v_t \in V$.

$$\begin{array}{ll} & \beta(a(u_t|_x b_t v_t)) \\ = & \{\text{distribute time-shift}\} \\ & \beta(a(u|_x bv)_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup a\beta(u|_x bv)_t \end{array}$$

From Corollary 8, page 94, $A(t) \subseteq U|_x V$ (using $au_t \in U$, $b_t v_t \in V$). The remaining task is to show $a\beta(u|_x bv)_t \subseteq U|_x V$.

$$\begin{array}{ll} & au_t \in U, \text{ and } b_t v_t \in V \\ \Rightarrow & \{\text{definitions}\} \\ & u \in U \backslash a,\ bv \in V_{-t} \\ \Rightarrow & \{U\backslash a \text{ is broad, from Lemma 26, page 73,} \\ & \quad \text{given } b_t v_t \in V, \text{ and } V \text{ broad, from Lemma 27, page 74, } V_{-t} \text{ is broad,} \\ & \quad \text{apply induction (combined length of } u \text{ and } bv \text{ is less than } au_t \text{ and } b_t v_t)\} \\ & \beta(u|_x bv) \subseteq U\backslash a|_x V_{-t} \\ \Rightarrow & \{\text{Apply Lemma 43, page 95, for each element in } \beta(u|_x bv): \\ & \quad a \text{ is base, from } a \preceq b; \text{ also, } a.time = t, \\ & \quad a \in U, \text{ from } au_t \in U, \text{ and } U \text{ is prefix-closed,} \\ & \quad b_t \in V, \text{ from } b_t v_t \in V, \text{ and } V \text{ is prefix-closed}\} \\ & a\beta(u|_x bv)_t \subseteq U|_x V \end{array}$$

Case 2) $[b \underline{\preceq\!\!\!\prec}_x\ a \to \beta(b(au|_x v))] \subseteq U|_x V$:
The proof is similar to that of Case (1); we include it here for completeness.

If $\neg(b \underline{\preceq\!\!\!\prec}_x\ a)$, then $[b \underline{\preceq\!\!\!\prec}_x\ a \to \beta(a(u|_x bv))] = \{\epsilon\}$, which is in $U|_x V$, from Observation 22, page 92. Assume $b \underline{\preceq\!\!\!\prec}_x\ a$. We rename the terms as $a_t u_t$ and bv_t. Our goal is to show $\beta(b(a_t u_t|_x v_t)) \subseteq U|_x V$, where $t = b.time$, $a_t u_t \in U$, $bv_t \in V$.

$$\begin{array}{ll} & \beta(b(a_t u_t|_x v_t)) \\ = & \{\text{distribute time-shift}\} \\ & \beta(b(au|_x v)_t) \\ = & \{\text{definition of } \beta()\} \\ & A(t) \cup b\beta(au|_x v)_t \end{array}$$

From Corollary 8, page 94, $A(t) \subseteq U|_x V$ (using $a_t u_t \in U$, $bv_t \in V$). The remaining task is to show $b\beta(au|_x v)_t \subseteq U|_x V$.

$$\begin{array}{ll} & a_t u_t \in U, \text{ and } bv_t \in V \\ \Rightarrow & \{\text{definitions}\} \\ & au \in U_{-t},\ v \in V\backslash a \\ \Rightarrow & \{\text{given } a_t u_t \in U, \text{ and } U \text{ broad, from Lemma 27, page 74, } U_{-t} \text{ is broad,} \\ & \quad \text{given } V \text{ broad, from Lemma 26, page 73, } V\backslash b \text{ is broad,} \\ & \quad \text{apply induction (combined length of } au \text{ and } v \text{ is less than } a_t u_t \text{ and } bv_t)\} \\ & \beta(au|_x v) \subseteq U_{-t}|_x V\backslash a \\ \Rightarrow & \{\text{Apply Corollary 9, page 95, for each element in } \beta(au|_x v): \\ & \quad b \text{ is base or own-substitution, from } b \underline{\preceq\!\!\!\prec}_x\ a, \text{ and } b.time = t \\ & \quad a_t \in U, \text{ from } a_t u_t \in U, \text{ and } U \text{ is prefix-closed,} \\ & \quad b \in V, \text{ from } bv_t \in V, \text{ and } V \text{ is prefix-closed}\} \end{array}$$

$b\beta(au|_x v)_t \subseteq U|_x V$

Lemma 44 Let u and v be visible. Then, $\overline{u|_x v} = \overline{u}|_x \overline{v}$

Proof: We prove the result by induction on the combined length of u and v.

If either u or v is ϵ, both sides are $\{\epsilon\}$, from the definition. Next, we take au and bv which are both visible, and show that $\overline{au|_x bv} = \overline{au}|_x \overline{bv}$. Note that u and v are visible, given au and bv are visible; either or both of u and v may be ϵ.

$$
\begin{array}{ll}
 & \overline{au|_x bv} \\
= & \{\text{definition of } au|_x bv\} \\
 & \overline{[a \approx_x b \rightarrow a(u|_x v)] \cup [a \preceq b \rightarrow a(u|_x bv)] \cup [b \preceq\!\!\prec_x a \rightarrow b(au|_x v)]} \\
= & \{\text{distribute trace over set union, guarded sets and concatenation}\} \\
 & [a \approx_x b \rightarrow \overline{a}(\overline{u|_x v})] \cup [a \preceq b \rightarrow \overline{a}(\overline{u|_x bv})] \cup [b \preceq\!\!\prec_x a \rightarrow \overline{b}(\overline{au|_x v})] \\
= & \{\text{apply induction. Note that } u,\ v,\ au \text{ and } bv \text{ are visible}\} \\
 & [a \approx_x b \rightarrow \overline{a}(\overline{u}|_x \overline{v})] \cup [a \preceq b \rightarrow \overline{a}(\overline{u}|_x \overline{bv})] \cup [b \preceq\!\!\prec_x a \rightarrow \overline{b}(\overline{au}|_x \overline{v})] \\
= & \{\text{distribute trace over concatenation}\} \\
 & [a \approx_x b \rightarrow \overline{a}(\overline{u}|_x \overline{v})] \cup [a \preceq b \rightarrow \overline{a}(\overline{u}|_x \overline{b}\,\overline{v})] \cup [b \preceq\!\!\prec_x a \rightarrow \overline{b}(\overline{a}\,\overline{u}|_x \overline{v})]\ (*)
\end{array}
$$

We show that $\overline{au|_x bv} = \overline{au}|_x \overline{bv}$ for each of these cases: (1) $a, b = \tau, \tau$, (2) $a \neq \tau$ and $b \neq \tau$ (3) $a \neq \tau$ and $b = \tau$, (4) $a = \tau$ and $b \neq \tau$.

Case 1) $a, b = \tau, \tau$: $\overline{au}|_x \overline{bv} = \overline{a}\,\overline{u}|_x \overline{b}\,\overline{v} = \overline{u}|_x \overline{v}$

$$
\begin{array}{ll}
 & \overline{au|_x bv} \\
= & \{\text{use } \neg(a \approx_x b) \text{ in } (*) \text{ since } a, b = \tau, \tau\text{; apply Observation 23, page 93}\} \\
 & [a \preceq b \rightarrow (\overline{u}|_x \overline{v})] \cup [b \preceq\!\!\prec_x a \rightarrow (\overline{u}|_x \overline{v})] \\
= & \{a, b = \tau, \tau, \text{ so } a \preceq b \equiv a.time \leq b.time, \text{ and } b \preceq\!\!\prec_x a \equiv b.time \leq a.time\} \\
 & [a.time \leq b.time \ \longrightarrow\ (\overline{u}|_x \overline{v})] \cup [b.time \leq a.time \ \longrightarrow\ (\overline{u}|_x \overline{v})] \\
= & \{\text{condition in at least one of the guarded sets applies, and} \\
 & \quad \epsilon \in (u|_x v)\text{, from Observation 22, page 92}\} \\
 & \overline{u}|_x \overline{v}
\end{array}
$$

Case 2) $a \neq \tau$ and $b \neq \tau$:

$$
\begin{array}{ll}
 & \overline{au}|_x \overline{bv} \\
= & \{\text{distribute trace over concatenation}\} \\
 & a\overline{u}|_x b\overline{v} \\
= & \{\text{apply definition}\} \\
 & [a \approx_x b \rightarrow a(\overline{u}|_x \overline{v})] \cup [a \preceq b \rightarrow a(\overline{u}|_x b\overline{v})] \cup [b \preceq\!\!\prec_x a \rightarrow b(a\overline{u}|_x \overline{v})]
\end{array}
$$

And, this matches (*) given $\overline{a} = a$ and $\overline{b} = b$.

Case 3) $a \neq \tau$ and $b = \tau$:
Then, $\overline{au}|_x \overline{bv} = a\overline{u}|_x \overline{v}$ (L)
And,

$$
\begin{array}{ll}
 & \overline{au|_x bv} \\
= & \{\text{since } b = \tau,\ \neg(a \approx_x b).\ \text{Simplify } (*)\} \\
 & [a \preceq b \rightarrow a(\overline{u}|_x \overline{v})] \cup [b \precneqq_x a \rightarrow (a\overline{u}|_x \overline{v})] \cup \{\epsilon\} \\
= & \{\text{drop } \{\epsilon\},\ \text{using Observation 23, page 93, on the second term}\} \\
 & [a \preceq b \rightarrow a(\overline{u}|_x \overline{v})] \cup [b \precneqq_x a \rightarrow (a\overline{u}|_x \overline{v})] \qquad \text{(R)}
\end{array}
$$

Since bv is visible and $b = \tau$, v is non-empty and visible. Therefore $\overline{v} \neq \epsilon$. Let c be the first event of $\overline{v}$. Then, considering bv, $b.time \leq c.time$. We show (L) = (R) for 3 cases: (1) $a.time < b.time$, (2) $a.time = b.time$, and (3) $a.time > b.time$.

Case 3.1) $a.time < b.time$:

$$
\begin{array}{ll}
 & \text{(L)} = a\overline{u}|_x \overline{v} \\
= & \{a.time < b.time \leq c.time.\ \text{Hence, } \neg(a \approx_x c) \text{ and } \neg(c \precneqq_x a) \\
 & \quad \text{Apply definition of } |_x\} \\
 & [a \preceq c \rightarrow a(\overline{u}|_x \overline{v})] \cup \{\epsilon\} \\
= & \{a.time \leq c.time \text{ means } a \preceq c \equiv a \text{ is base }\} \\
 & [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup \{\epsilon\}
\end{array}
$$

And,

$$
\begin{array}{ll}
 & \text{(R)} = [a \preceq b \rightarrow a(\overline{u}|_x \overline{v})] \cup [b \precneqq_x a \rightarrow (a\overline{u}|_x \overline{v})] \\
= & \{a.time < b.time.\ \text{So, } \neg(b \precneqq_x a).\ \text{Also, } a \preceq b \equiv a \text{ is base }\} \\
 & [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup \{\epsilon\} \\
= & \{\text{from above derivation}\} \\
 & \text{(L)}
\end{array}
$$

Case 3.2) $a.time = b.time$:

$$
\begin{array}{ll}
 & \text{(R)} = [a \preceq b \rightarrow a(\overline{u}|_x \overline{v})] \cup [b \precneqq_x a \rightarrow (a\overline{u}|_x \overline{v})] \\
= & \{\text{from } a.time = b.time,\ a \preceq b \equiv a \text{ is base and } b \precneqq_x a \text{ holds, given } b = \tau\} \\
 & [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup (a\overline{u}|_x \overline{v}) \\
= & \{\text{If } a \text{ is not base :} \\
 & \quad [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup (a\overline{u}|_x \overline{v}) = \{\epsilon\} \cup (a\overline{u}|_x \overline{v}) = (a\overline{u}|_x \overline{v}) \\
 & \text{If } a \text{ is base :} \\
 & \quad [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup (a\overline{u}|_x \overline{v}) = a(\overline{u}|_x \overline{v}) \cup (a\overline{u}|_x \overline{v}) \\
 & \quad a.time \leq c.time \text{ implies, from } |_x \text{ definition, } a\overline{u}|_x \overline{v} \supseteq a(\overline{u}|_x \overline{v}) \\
 & \text{In all cases, } [a \text{ is base} \ \longrightarrow \ a(\overline{u}|_x \overline{v})] \cup (a\overline{u}|_x \overline{v}) = (a\overline{u}|_x \overline{v})\} \\
 & a\overline{u}|_x \overline{v} = \text{(L)}
\end{array}
$$

Case 3.3) $a.time > b.time$:

$$
\begin{array}{ll}
 & \text{(R)} = [a \preceq b \rightarrow a(\overline{u}|_x \overline{v})] \cup [b \precneqq_x a \rightarrow (a\overline{u}|_x \overline{v})] \\
= & \{\neg(a \preceq b) \text{ and } b \precneqq_x a \text{ hold}\} \\
 & (a\overline{u}|_x \overline{v}) \cup \{\epsilon\} \\
= & \{\text{from Observation 22, page 92, } \{\epsilon\} \subseteq (a\overline{u}|_x \overline{v})\} \\
 & a\overline{u}|_x \overline{v} = \text{(L)}
\end{array}
$$

Case 4) $a = \tau$ and $b \neq \tau$: Similar to Case (3)

Lemma 45 $u|_x v \subseteq uc|_x vd$, for other substitutions c and d.

Proof: : We prove $u|_x v \subseteq uc|_x v$. Similarly, it can be shown that $u|_x v \subseteq u|_x vd$. Then, $u|_x v \subseteq uc|_x v \subseteq uc|_x vd$.

Now, we prove $u|_x v \subseteq uc|_x v$ by induction on the combined length of u and v.

- $u = \epsilon$ or $v = \epsilon$: Left side is $\{\epsilon\}$, and the right side, being a merge, contains ϵ, from Observation 22, page 92.

- $au|_x bv \subseteq auc|_x bv$:

$$
\begin{array}{ll}
 & au|_x bv \\
= & \{\text{definition of } |_x\} \\
 & [a \approx_x b \rightarrow a(u|_x v)] \cup [a \preceq b \rightarrow a(u|_x bv)] \cup [b \underline{\prec\!\!\prec}_x a \rightarrow b(au|_x v)] \\
\subseteq & \{\text{induction: } u|_x v \subseteq uc|_x v\text{; similarly for the other terms}\} \\
 & [a \approx_x b \rightarrow a(uc|_x v)] \cup [a \preceq b \rightarrow a(uc|_x bv)] \cup [b \underline{\prec\!\!\prec}_x a \rightarrow b(auc|_x v)] \\
= & \{\text{definition of } |_x\} \\
 & auc|_x bv
\end{array}
$$

Theorem 22 Let U and V be broad sets. Then $\overline{U|_x V} = \overline{U}|_x \overline{V}$.

Proof: The proof is in two parts: $\overline{U|_x V} \subseteq \overline{U}|_x \overline{V}$, and $\overline{U}|_x \overline{V} \subseteq \overline{U|_x V}$.

- $\overline{U|_x V} \subseteq \overline{U}|_x \overline{V}$: For any u in U and v in V, we show $\overline{u|_x v} \subseteq \overline{U}|_x \overline{V}$.

$$
\begin{array}{ll}
 & \overline{u|_x v} \\
\subseteq & \{\text{from Lemma 23, page 71, there is } uc \in \beta(u) \subseteq U \text{ and, similarly, } vd \in V, \\
 & \quad \text{where we may pick } c \text{ and } d \text{ to be other-substitutions,} \\
 & \quad \text{apply Lemma 45, page 100}\} \\
 & \overline{(uc|_x vd)} \\
= & \{uc \text{ and } vd \text{ are visible; apply Lemma 44, page 98}\} \\
 & (\overline{uc}|_x \overline{vd}) \\
\subseteq & \{\text{from } uc \in U,\ \overline{uc} \in \overline{U}\text{; similarly } \overline{vd} \in \overline{V}\} \\
 & \overline{U}|_x \overline{V}
\end{array}
$$

- $\overline{U}|_x \overline{V} \subseteq \overline{U|_x V}$: We show that for u in U and v in V, $\overline{u}|_x \overline{v} \subseteq \overline{U|_x V}$.

Let p be the longest visible prefix of u and q of v. Then $\overline{u} = \overline{p}$ and $\overline{v} = \overline{q}$. From prefix-closure, $p \in U$ and $q \in V$.

$$
\begin{array}{ll}
 & \overline{u}|_x \overline{v} \\
= & \{\text{from above}\} \\
 & \overline{p}|_x \overline{q} \\
= & \{p \text{ and } q \text{ are visible}\} \\
 & \overline{p|_x q} \\
\subseteq & \{p \in U \text{ and } q \in V\} \\
 & \overline{U|_x V}
\end{array}
$$

3.4.2 Preliminary Results on Constrained Full Merge

Lemma 46 $\overline{u +_x v} = \overline{u} +_x \overline{v}$.

Proof: The proof is analogous to Lemma 44, page 98. Proof is by induction on the combined length of u and v.

Suppose u is ϵ: if v contains no other-substitution (then neither does $\overline{v}$), $\overline{u +_x v} = \{\overline{v}\}$ and $\overline{u} +_x \overline{v} = \{\overline{v}\}$. If v contains an other-substitution, then so does $\overline{v}$, and $\overline{u +_x v} = \overline{\phi} = \phi = \overline{u} +_x \overline{v}$. The proof for $v = \epsilon$ is analogous.

Next, we show that $\overline{au +_x bv} = \overline{au} +_x \overline{bv}$.

$$
\begin{array}{cl}
 & \overline{au +_x bv} \\
= & \{\text{definition of } au +_x bv\} \\
 & \overline{\langle a \approx_x b \to a(u +_x v)\rangle \cup \langle a \preceq b \to a(u +_x bv)\rangle \cup \langle b \underline{\prec\!\prec}_x a \to b(au +_x v)\rangle} \\
= & \{\text{distribute trace over set union, guarded sets and concatenation}\} \\
 & \langle a \approx_x b \to \overline{a}(\overline{u +_x v})\rangle \cup \langle a \preceq b \to \overline{a}(\overline{u +_x bv})\rangle \cup \langle b \underline{\prec\!\prec}_x a \to \overline{b}(\overline{au +_x v})\rangle \\
= & \{\text{induction}\} \\
 & \langle a \approx_x b \to \overline{a}(\overline{u} +_x \overline{v})\rangle \cup \langle a \preceq b \to \overline{a}(\overline{u} +_x \overline{bv})\rangle \cup \langle b \underline{\prec\!\prec}_x a \to \overline{b}(\overline{au} +_x \overline{v})\rangle \\
= & \{\text{distribute trace over concatenation}\} \\
 & \langle a \approx_x b \to \overline{a}(\overline{u} +_x \overline{v})\rangle \cup \langle a \preceq b \to \overline{a}(\overline{u} +_x \overline{b}\,\overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x a \to \overline{b}(\overline{a}\,\overline{u} +_x \overline{v})\rangle (*)
\end{array}
$$

We show that $\overline{au +_x bv} = \overline{au} +_x \overline{bv}$ for each of these cases: (1) $a, b = \tau, \tau$, (2) $a \neq \tau$ and $b \neq \tau$ (3) $a \neq \tau$ and $b = \tau$, (4) $a = \tau$ and $b \neq \tau$.

Case 1) $a, b = \tau, \tau$: Then, $\overline{au} +_x \overline{bv} = \overline{u} +_x \overline{v}$

$$
\begin{array}{cl}
 & \overline{au +_x bv} \\
= & \{\text{simplify (*), noting that } \neg(a \approx_x b)\text{, from } a, b = \tau, \tau\} \\
 & \langle a \preceq b \to (\overline{u} +_x \overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x a \to (\overline{u} +_x \overline{v})\rangle \\
= & \{\text{given } a, b = \tau, \tau,\ a \preceq b \equiv a.time \leq b.time \text{ and } b \underline{\prec\!\prec}_x a \equiv b.time \leq a.time, \\
 & \quad \text{so, at least one of } a \preceq b \text{ and } b \underline{\prec\!\prec}_x a \text{ holds }\} \\
 & \overline{u} +_x \overline{v}
\end{array}
$$

Case 2) $a \neq \tau$ and $b \neq \tau$:

$$
\begin{array}{cl}
 & \overline{au} +_x \overline{bv} \\
= & \{\text{distribute trace over concatenation}\} \\
 & a\overline{u} +_x b\overline{v} \\
= & \{\text{apply definition}\} \\
 & \langle a \approx_x b \to \overline{a}(\overline{u} +_x \overline{v})\rangle \cup \langle a \preceq b \to a(\overline{u} +_x b\overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x a \to b(a\overline{u} +_x \overline{v})\rangle
\end{array}
$$

And, this matches (*) given $\overline{a} = a$ and $\overline{b} = b$.

Case 3) $a \neq \tau$ and $b = \tau$:
Then, $\overline{au} +_x \overline{bv} = a\overline{u} +_x \overline{v}$ (L)
And,

$$\begin{array}{cl}
 & \overline{au +_x bv} \\
= & \{\text{from } (*) \text{, noting that } \neg(a \approx_x b) \text{, from } b = \tau\} \\
 & \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x\ a \rightarrow (a\overline{u} +_x \overline{v})\rangle
\end{array} \qquad \text{(R)}$$

Since bv is visible and $b = \tau$, v is non-empty and visible. Therefore $\overline{v} \neq \epsilon$. Let c be the first event of $\overline{v}$. Then, $b.time \leq c.time$. We show (L) = (R) for 3 cases: (1) $a.time < b.time$, (2) $a.time = b.time$, and (3) $a.time > b.time$.

Case 3.1) $a.time < b.time$:

$$\begin{array}{cl}
 & \text{(L)} = a\overline{u} +_x \overline{v} \\
= & \{a.time < b.time \leq c.time. \text{ Hence, } \neg(a \approx_x c) \text{ and } \neg(c \underline{\prec\!\prec}_x\ a) \\
 & \quad \text{Apply definition of } +_x\} \\
 & \langle a \preceq c \rightarrow a(\overline{u} +_x \overline{v})\rangle \\
= & \{\text{from } a.time < c.time,\ a \preceq c \equiv a \text{ is base}\} \\
 & \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle
\end{array}$$

And,

$$\begin{array}{cl}
 & \text{(R)} = \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x\ a \rightarrow (a\overline{u} +_x \overline{v})\rangle \\
= & \{a.time < b.time. \text{ So, } \neg(b \underline{\prec\!\prec}_x\ a)\} \\
 & \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \\
= & \{a.time < b.time. \text{ So, } a \preceq b \equiv a \text{ is base }\} \\
 & \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle \\
= & \{\text{above derivation}\} \\
 & \text{(L)}
\end{array}$$

Case 3.2) $a.time = b.time$:

$$\begin{array}{cl}
 & \text{(R)} = \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x\ a \rightarrow (a\overline{u} +_x \overline{v})\rangle \\
= & \{\text{given } a.time = b.time \text{ and } b = \tau,\ b \underline{\prec\!\prec}_x\ a \text{ holds}\} \\
 & \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \cup (a\overline{u} +_x \overline{v}) \\
= & \{\text{given } a.time = b.time,\ a \preceq b \equiv a \text{ is base }\} \\
 & \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle \cup (a\overline{u} +_x \overline{v}) \\
= & \{\text{If } a \text{ is not base :} \\
 & \quad \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle \cup (a\overline{u} +_x \overline{v}) = a\overline{u} +_x \overline{v}, \\
 & \text{If } a \text{ is base :} \\
 & \quad \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle \cup (a\overline{u} +_x \overline{v}) = a(\overline{u} +_x \overline{v}) \cup (a\overline{u} +_x \overline{v}), \\
 & \quad a.time \leq c.time \text{ implies, from } +_x \text{ definition, } a\overline{u} +_x \overline{v} \supseteq a(\overline{u} +_x \overline{v}) \\
 & \text{In all cases, } \langle a \text{ is base} \ \longrightarrow\ a(\overline{u} +_x \overline{v})\rangle \cup (a\overline{u} +_x \overline{v}) = a\overline{u} +_x \overline{v}\} \\
 & a\overline{u} +_x \overline{v} = \text{(L)}
\end{array}$$

Case 3.3) $a.time > b.time$:

$$\begin{array}{cl}
 & \text{(R)} = \langle a \preceq b \rightarrow a(\overline{u} +_x \overline{v})\rangle \cup \langle b \underline{\prec\!\prec}_x\ a \rightarrow (a\overline{u} +_x \overline{v})\rangle \\
= & \{\neg(a \preceq b) \text{ and } b \underline{\prec\!\prec}_x\ a \text{ hold}\} \\
 & a\overline{u} +_x \overline{v} = \text{(L)}
\end{array}$$

Case (4) $a = \tau$ and $b \neq \tau$: Similar to Case(3).

Call a sequence *pub-free* if it has no publication. Formally, ϵ is pub-free, and ap is pub-free iff a is not a publication and p is pub-free. A set is pub-free if all its sequences are.

Observation 24 $A(t)$ is pub-free.

Proof: From the definition of $A(t)$.

Lemma 47 Given p is pub-free, $\beta(p)$ is pub-free.

Proof: By induction on the length of p. For $p = \epsilon$, $\beta(\epsilon) = A(0)$ is pub-free, by Observation 24. Next, consider ap_t where a is not a publication and p is pub-free. Then, $\beta(ap_t) = A(t) \cup \beta(p)_t$, where $A(t)$ is pub-free, by Observation 24, and $\beta(p)$ is pub-free by induction hypothesis (then, so is $\beta(p)_t$).

Lemma 48 Let V be broad and W be its pub-free subset. Then, W is broad.

Proof: We show that for every v, where $v \in W$, $\beta(v) \subseteq W$.

$$
\begin{array}{ll}
 & v \in W \\
\Rightarrow & \{W \subseteq V;\ v \in W \text{ means } v \text{ is pub-free}\} \\
 & v \in V \text{ and } v \text{ is pub-free} \\
\Rightarrow & \{V \text{ is broad}\} \\
 & \beta(v) \subseteq V \text{ and } v \text{ is pub-free} \\
\Rightarrow & \{v \text{ is pub-free implies } \beta(v) \text{ is pub-free, from Lemma 47, page 103}\} \\
 & \beta(v) \subseteq V \text{ and } \beta(v) \text{ is pub-free} \\
\Rightarrow & \{W \text{ is the pub-free subset of } V\} \\
 & \beta(v) \subseteq W
\end{array}
$$

Lemma 49 $\beta(paq_t) = \beta(pc) \cup pa\beta(q)_t$, where $t = a.time$, and c is any substitution at time t.

Proof: Proof is by induction on the length of p.

- $p = \epsilon$: we have to show $\beta(aq_t) = \beta(c) \cup a\beta(q)_t$.

$$
\begin{array}{ll}
 & \beta(c) \cup a\beta(q)_t \\
= & \{c \text{ is at time } t\text{; from definition of } \beta(),\ \beta(c) = A(t) \cup c\beta(\epsilon)_t\} \\
 & A(t) \cup c\beta(\epsilon)_t \cup a\beta(q)_t \\
= & \{\beta(\epsilon)_t = A(0)_t;\ c \text{ being a substitution, } cA(0)_t \subseteq A(t)\} \\
 & A(t) \cup a\beta(q)_t \\
= & \{\text{definition of } \beta()\} \\
 & \beta(aq_t)
\end{array}
$$

- $\beta(bpaq_t) = \beta(bpc) \cup bpa\beta(q)_t$:

Let $s = b.time$, $p'_s = p$, $a'_s = a$, and $r = t - s$.

$$\begin{array}{ll}
& \beta(bpaq_t) \\
= & \{\text{transform using } p'_s = p,\ a'_s = a, \text{ and } r = t - s.\} \\
& \beta(b(p'a'q_r)_s) \\
= & \{\text{apply definition of } \beta(), \text{ using } s = b.time\} \\
& A(s) \cup b\beta(p'a'q_r)_s \\
= & \{\text{induction; note } a.time = t, \text{ so } (a'_s).time = t, \text{ or } a'.time = t - s = r\} \\
& A(s) \cup b(\beta(p'd) \cup p'a'\beta(q)_r)_s, \text{ where } d \text{ is any substitution at } a'.time = r \\
= & \{\text{rewrite using } p'_s a'_s = pa\} \\
& A(s) \cup b\beta(p'd)_s \cup bpa\beta(q)_{r+s} \\
= & \{\beta(b(p'd)_s) = A(s) \cup b\beta(p'd)_s\} \\
& \beta(b(p'd)_s) \cup bpa\beta(q)_{r+s} \\
= & \{p'_s = p;\ r + s = t\} \\
& \beta(bpd_s) \cup bpa\beta(q)_t
\end{array}$$

Now, $c = d_s$ is an arbitrary substitution at $d.time + s = r + s = t$ □

Lemma 50 Let $pa \in U$, where U is broad. Then $pc \in U$, where c is any substitution at $a.time$.

Proof:

$$\begin{array}{ll}
& pa \in U \\
\Rightarrow & \{U \text{ is broad}\} \\
& \beta(pa) \subseteq U \\
\Rightarrow & \{\text{from Lemma 49, page 103, with } q = \epsilon,\ \beta(pa) = \beta(pc) \cup pa\beta(\epsilon)_t \\
& \quad \text{So, } \beta(pc) \subseteq \beta(pa); \text{ note that } c.time = a.time\} \\
& \beta(pc) \subseteq U, \text{ where } c.time = a.time \\
\Rightarrow & \{\text{from Corollary 2, page 70, } pc \in \beta(pc)\} \\
& pc \in U, \text{ where } c.time = a.time
\end{array}$$

3.4.3 Asymmetric Composition Preserves Breadth

We show that for broad sets U and V, $U <x< V$ is broad.

Lemma 51 Let U and V be broad and V pub-free. For any p, $p \in U|_x V$, there exist u in U and v in V such that $d_1(u, v)$ and $p \in u|_x v$.

Proof: First, we prove a sublemma.

Sublemma Consider sequences u and v. Let $u = u'cu''$ where c is an own-substitution. Let d be an other-substitution, where $c.time = d.time$ and d does not occur in v. Then, $u|_x v = u'd|_x v$.
Proof: Proof is by induction on the combined length of u' and v.

- $v = \epsilon$: Then $u|_x v = \{\epsilon\} = u'd|_x v$.
- $u' = \epsilon$: We may assume that $v \neq \epsilon$, i.e., $v = bv'$.

$$
\begin{array}{cl}
 & u|_x v \\
= & \{u = cu'' \text{ and } v = bv'\} \\
 & cu''|_x bv' \\
= & \{\text{definition of } |_x\} \\
 & [c \approx_x b \rightarrow c(u''|_x v')] \cup [c \preceq b \rightarrow c(u''|_x bv')] \cup [b \underline{\prec\!\!\prec}_x\ c \rightarrow b(cu''|_x v')] \\
= & \{\neg(c \approx_x b), \text{ since } c \text{ is not an other-substitution} \\
 & \quad \neg(c \preceq b), \text{ since } c \text{ is not a base event}\} \\
 & [b \underline{\prec\!\!\prec}_x\ c \rightarrow b(cu''|_x v')] \cup \{\epsilon\} \\
= & \{\text{induction: } cu''|_x v' = d|_x v'\} \\
 & [b \underline{\prec\!\!\prec}_x\ c \rightarrow b(d|_x v')] \cup \{\epsilon\} \\
= & \{b \underline{\prec\!\!\prec}_x\ c \equiv b \underline{\prec\!\!\prec}_x\ d, \text{ because } c.time = d.time\} \\
 & [b \underline{\prec\!\!\prec}_x\ d \rightarrow b(d|_x v')] \cup \{\epsilon\}
\end{array}
$$

And,

$$
\begin{array}{cl}
 & u'd|_x v \\
= & \{u' = \epsilon \text{ and } v = bv'\} \\
 & d|_x bv' \\
= & \{\text{definition of } |_x\} \\
 & [d \approx_x b \rightarrow d(\epsilon|_x v')] \cup [d \preceq b \rightarrow d(\epsilon|_x bv')] \cup [b \underline{\prec\!\!\prec}_x\ d \rightarrow b(d|_x v')] \\
= & \{\neg(d \approx_x b), \text{ since } d \text{ does not occur in } v \\
 & \quad \neg(d \preceq b), \text{ since } d \text{ is not a base event}\} \\
 & [b \underline{\prec\!\!\prec}_x\ d \rightarrow b(d|_x v')] \cup \{\epsilon\} \\
= & \{\text{from above proof}\} \\
 & u|_x v
\end{array}
$$

- $u' = aw'$: Then $u = aw'cu''$. Abbreviate $w'cu''$ by w.

$$
\begin{array}{cl}
 & u|_x v \\
= & \{u = aw'cu'' = aw \text{ and } v = bv'\} \\
 & aw|_x bv' \\
= & \{\text{definition of } |_x\} \\
 & [a \approx_x b \rightarrow a(w|_x v')] \cup [a \preceq b \rightarrow a(w|_x bv')] \cup [b \underline{\prec\!\!\prec}_x\ a \rightarrow b(aw|_x v')] \\
= & \{\text{induction on every term; recall } w = w'cu''\text{; so, replace } w \text{ by } w'd\} \\
 & [a \approx_x b \rightarrow a(w'd|_x v')] \cup [a \preceq b \rightarrow a(w'd|_x bv')] \cup [b \underline{\prec\!\!\prec}_x\ a \rightarrow b(aw'd|_x v')] \\
= & \{\text{definition of } |_x\} \\
 & aw'd|_x bv' \\
= & \{u' = aw' \text{ and } v = bv'\} \\
 & u'd|_x v
\end{array}
$$

Now, we are ready to prove the main lemma. Given $p \in U|_x V$, there exist r and s, in U and V, respectively, such that $p \in r|_x s$. If r has no own-substitution, then $d_1(r, s)$, because s is pub-free (from V pub-free); thus $u, v = r, s$.

If r has a own-substitution c, let $r = r'cr''$, where r' has no own-substitution. Since U is broad, it is prefix-closed; therefore $r = r'cr'' \in U$ implies $r'c \in U$. Using Lemma 50, page 104, (set $p = r'$ and $a = c$), $r'd \in U$, where d is any substitution at $c.time$. Choose d to be an other-substitution that does not occur in s (this is possible because s has a finite number of substitutions). Let $u = r'd$ and $v = s$. Then, $p \in r|_x s =$ {from Sublemma} $r'd|_x s = u|_x v$. Further, u has no own substitution because r' has none and d is an other-substitution. Also, s is pub-free. So, $d_1(u, v)$.

Lemma 52 Let U and W be broad and W be pub-free. Then $U <x< W = U|_x W$.

Proof:

$$
\begin{array}{ll}
 & U|_x W \\
= & \{\text{definition of coercion}\} \\
 & (\cup u \in U,\ w \in W :\ u|_x w) \\
= & \{\text{from Lemma 51, page 104}\} \\
 & (\cup u \in U,\ w \in W,\ d_1(u, w) :\ u|_x w) \\
= & \{\text{definition of } u <x< w \text{ using } w \text{ is pub-free}\} \\
 & (\cup u \in U,\ w \in W,\ d_1(u, w) :\ u <x< w) \\
= & \{\text{given } w \text{ is pub-free}, \neg d_1(u, w) \text{ implies } u <x< w = \phi\} \\
 & (\cup u \in U,\ w \in W,\ d_1(u, w) :\ u <x< w) \cup (\cup u \in U,\ w \in W,\ \neg d_1(u, w) :\ u <x< w) \\
= & \{\text{set theory}\} \\
 & (\cup u \in U,\ w \in W :\ u <x< w) \\
= & \{\text{definition of } <x< \text{ over sets}\} \\
 & U <x< W
\end{array}
$$

Theorem 23 For broad sets U and V, $U <x< V$ is broad.

Proof: We show that for every p, where $p \in U <x< V$, $\beta(p) \subseteq U <x< V$. Let W be the pub-free subset of V. We consider two cases: (1) $p \in U <x< W$, and (2) $p \in U <x< (V - W)$, and show in each case that $\beta(p) \subseteq U <x< V$.

- $p \in U <x< W$: We first show that $U <x< W$ is broad.

$$
\begin{array}{ll}
 & W \text{ is the pub-free subset of } V \\
\Rightarrow & \{\text{from Lemma 48, page 103}\} \\
 & W \text{ is broad} \\
\Rightarrow & \{U \text{ is broad; from Theorem 21, page 95}\} \\
 & U|_x W \text{ is broad} \\
\Rightarrow & \{U \text{ and } W \text{ are broad, and } W \text{ is pub-free; from Lemma 52, page 106}\} \\
 & U <x< W = U|_x W, \text{ and } U|_x W \text{ is broad} \\
\Rightarrow & \{\text{obviously}\} \\
 & U <x< W \text{ is broad}
\end{array}
$$

Hence,

$$
\begin{array}{ll}
 & p \in U <x< W \\
\Rightarrow & \{U <x< W \text{ is broad}\} \\
 & \beta(p) \subseteq U <x< W \\
\Rightarrow & \{W \subseteq V\} \\
 & \beta(p) \subseteq U <x< V
\end{array}
$$

• $p \in U <x< (V - W)$: Any such p is generated by $u \in U$ and $v \in V$, where $d_2(u, v)$. So, $u = u'(t, [m/x])u''_t$ and $v = v'(t, !m)v''$, $d_0(u', v')$ and $d_1(u', v')$.

$$
\begin{array}{ll}
 & p \in (u' +_{x} v')(t, \tau)u''_t \\
\Rightarrow & \{\text{apply } \beta() \text{ to both sides}\} \\
 & \beta(p) \subseteq \beta((u' +_{x} v')(t, \tau)u''_t) \\
\Rightarrow & \{\text{from Lemma 49, page 103,} \\
 & \quad \beta((u' +_{x} v')(t, \tau)u''_t) = \beta((u' +_{x} v')c) \cup (u' +_{x} v')(t, \tau)\beta(u'')_t, \\
 & \quad \text{where } c \text{ is an other-substitution at time } t\} \\
 & \beta(p) \subseteq \beta((u' +_{x} v')c) \cup (u' +_{x} v')(t, \tau)\beta(u'')_t, \\
 & \text{where } c \text{ is an other-substitution at } t
\end{array}
$$

We show that each of $\beta((u' +_{x} v')c)$ and $(u' +_{x} v')(t, \tau)\beta(u'')_t$ are subsets of $U <x< V$.

Case 1) $\beta((u' +_{x} v')c) \subseteq U <x< V$:
Since U is broad, and hence prefix-closed, from $u'(t, [m/x])u''_t \in U$, we have $u'(t, [m/x]) \in U$, and using Lemma 50, page 104, $u'c \in U$; similarly, $v'c \in V$. From $d_1(u', v')$, u' has no own-substitution; therefore, $u'c$ has none either. From $d_1(u', v')$, v' is pub-free, and so is $v'c$. So, $d_1(u'c, v'c)$. Also, $v'c \in W$, since $v'c \in V$ and W is the pub-free subset of V.

$$
\begin{array}{ll}
 & \beta(u'c +_{x} v'c) \\
\subseteq & \{\text{from Lemma 12, page 55, } u'c +_{x} v'c \subseteq u'c|_{x} v'c\} \\
 & \beta(u'c|_{x} v'c) \\
\subseteq & \{u'c \in U;\ v'c \in W\} \\
 & \beta(U|_{x} W) \\
= & \{U \text{ is broad,} \\
 & \quad W \text{ is broad, from Lemma 48, page 103,} \\
 & \quad \text{so, } U|_{x} W \text{ is broad, from Theorem 21, page 95}\} \\
 & U|_{x} W \\
= & \{U \text{ and } W \text{ are broad, } W \text{ has no publication; from Lemma 52, page 106}\} \\
 & U <x< W \\
\subseteq & \{W \subseteq V\} \\
 & U <x< V
\end{array}
$$

Case 2) $(u' +_{x} v')(t, \tau)\beta(u'')_t \subseteq U <x< V$:

$$
\begin{array}{ll}
 & (u' +_{x} v')(t, \tau)\beta(u'')_t \\
= & \{d_2(u, v) \text{ implies } d_2(w, v), \text{ where } w = u'(t, [m/x])w', \text{ for any } w', \\
 & \quad \text{use each element of } \beta(u'')_t \text{ for } w'; \text{ apply definition of } <x<\ \} \\
 & u'(t, [m/x])\beta(u'')_t <x< v \\
\subseteq & \{\text{from Lemma 49, page 103, } u'(t, [m/x])\beta(u'')_t \subseteq \beta(u'(t, [m/x])u''_t) = \beta(u)\} \\
 & \beta(u) <x< v \\
\subseteq & \{u \in U,\ v \in V\} \\
 & \beta(U) <x< V \\
= & \{U \text{ is broad; so } \beta(U) = U\} \\
 & U <x< V
\end{array}
$$

Lemma 53 *$(U <x< V\backslash a) \subseteq (U <x< V)\backslash a$, where a is an own substitution.*

Proof: We show for arbitrary $u \in U$ and $v \in V$ that $(u <x< v\backslash a) \subseteq (u <x< v)\backslash a$. The result then follows by coercion.

Suppose v does not begin with a. Then

$$
\begin{array}{ll}
 & (u <x< v\backslash a) \\
= & \{\text{definition of } \backslash\} \\
 & (u <x< \emptyset) \\
= & \{\text{coercion}\} \\
 & \emptyset,
\end{array}
$$

Suppose $d_1(u, v)$ holds. Then

$$
\begin{array}{ll}
 & (u <x< v)\backslash a \\
= & \{\text{definition of } u <x< v,\ d_1(u, v)\} \\
 & (u|_x v)\backslash a \\
= & \{\text{if } v \text{ does not begin with } a\text{, then neither does any execution of } u|_x v\} \\
 & \emptyset.
\end{array}
$$

Suppose $d_2(u, v)$ holds. Then

$$
\begin{array}{ll}
 & (u <x< v)\backslash a \\
= & \{\text{definition of } u <x< v,\ d_2(u, v)\} \\
 & ((u' +_x v')(t, \tau)u'')\backslash a \\
= & \{\text{if } v \text{ does not begin with } a\text{, then neither does any execution of } u' +_x v'\} \\
 & \emptyset.
\end{array}
$$

Suppose neither $d_1(u, v)$ not $d_2(u, v)$ holds. Then

$$
\begin{array}{ll}
 & (u <x< v)\backslash a \\
= & \{\text{definition of } u <x< v\text{, neither } d_2(u, v) \text{ nor } d_2(u, v)\} \\
 & \emptyset\backslash a. \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset.
\end{array}
$$

Otherwise assume v starts with a, and $v = av'$.
Suppose $d_1(u, v)$ holds. Then

$$
\begin{array}{ll}
 & u <x< av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u <x< v' \\
= & \{\text{definition of } u <x< v',\ d_1(u, av') \text{ implies } d_1(u, v')\} \\
 & u|_x v'
\end{array}
$$

and

$$
\begin{array}{ll}
 & (u \mathbin{<\!x\!<} av')\backslash a \\
= & \{\text{definition of } u \mathbin{<\!x\!<} av',\ d_1(u, av')\} \\
 & (u|_x av')\backslash a \\
\subseteq & \{\text{definition of } u|_x av'\} \\
 & a(u|_x v')\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u|_x v'
\end{array}
$$

Suppose $d_2(u, v)$ holds. Then

$$
\begin{array}{ll}
 & u \mathbin{<\!x\!<} av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u \mathbin{<\!x\!<} v' \\
= & \{\text{definition of } u \mathbin{<\!x\!<} v',\ d_2(u, av') \text{ implies } d_2(u, v')\} \\
 & (u'' +_x v'')(t, \tau)u'''
\end{array}
$$

and

$$
\begin{array}{ll}
 & (u \mathbin{<\!x\!<} av')\backslash a \\
= & \{\text{definition of } u \mathbin{<\!x\!<} av',\ d_2(u, av')\} \\
 & (u'' +_x av'')(t, \tau)u'''\backslash a \\
\subseteq & \{\text{definition of } u'' +_x av''\} \\
 & a(u'' +_x v'')(t, \tau)u'''\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & (u'' +_x v'')(t, \tau)u'''
\end{array}
$$

Suppose neither $d_1(u, v)$ nor $d_2(u, v)$ holds. Then

$$
\begin{array}{ll}
 & u \mathbin{<\!x\!<} av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u \mathbin{<\!x\!<} v' \\
= & \{\text{definition of } u \mathbin{<\!x\!<} v',\ \neg d_1(u, av') \text{ implies } \neg d_1(u, v') \text{ and} \\
\multicolumn{2}{l}{\neg d_2(u, av') \text{ implies } \neg d_2(u, v')\}} \\
 & \emptyset
\end{array}
$$

and

$$
\begin{array}{ll}
 & (u \mathbin{<\!x\!<} av')\backslash a \\
= & \{\text{definition of } u \mathbin{<\!x\!<} av',\ \neg d_1(u, av') \text{ and } \neg d_2(u, av')\} \\
 & \emptyset\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset
\end{array}
$$

Lemma 54 $(U\backslash a \mathbin{<\!x\!<} V\backslash a) = (U \mathbin{<\!x\!<} V)\backslash a$*, where* a *is an other substitution.*

Proof: We show for arbitrary $u \in U$ and $v \in V$ that $(u\backslash a \mathbin{<\!x\!<} v\backslash a) = (u \mathbin{<\!x\!<} v)\backslash a$. The result then follows by coercion.

Suppose u does not begin with a. (The case when v does not begin with a is similar.) Then

$$\begin{array}{ll}
 & (u\backslash a \mathrel{<x<} v\backslash a) \\
= & \{\text{definition of } \backslash\} \\
 & (\emptyset \mathrel{<x<} v\backslash a) \\
= & \{\text{coercion}\} \\
 & \emptyset,
\end{array}$$

Suppose $d_1(u,v)$ holds. Then

$$\begin{array}{ll}
 & (u \mathrel{<x<} v)\backslash a \\
= & \{\text{definition of } u \mathrel{<x<} v,\ d_1(u,v)\} \\
 & (u|_{x} v)\backslash a \\
= & \{\text{if } u \text{ does not begin with } a\text{, then neither does any execution of } u|_{x} v\} \\
 & \emptyset.
\end{array}$$

Suppose $d_2(u,v)$ holds. Then

$$\begin{array}{ll}
 & (u \mathrel{<x<} v)\backslash a \\
= & \{\text{definition of } u \mathrel{<x<} v,\ d_2(u,v)\} \\
 & ((u' +_{x} v')(t,\tau)u'')\backslash a \\
= & \{\text{if } u \text{ does not begin with } a\text{, then neither does any execution of } u' +_{x} v'\} \\
 & \emptyset.
\end{array}$$

Suppose neither $d_1(u,v)$ not $d_2(u,v)$ holds. Then

$$\begin{array}{ll}
 & (u \mathrel{<x<} v)\backslash a \\
= & \{\text{definition of } u \mathrel{<x<} v\text{, neither } d_2(u,v) \text{ nor } d_2(u,v)\} \\
 & \emptyset\backslash a. \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset.
\end{array}$$

Otherwise, assume both u and v begin with a, and $u = au'$ and $v = av'$.
Suppose $d_1(u,v)$ holds. Then

$$\begin{array}{ll}
 & au'\backslash a \mathrel{<x<} av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u' \mathrel{<x<} v' \\
= & \{\text{definition of } u' \mathrel{<x<} v',\ d_1(au',av') \text{ implies } d_1(u',v')\} \\
 & u'|_{x} v'
\end{array}$$

and

$$\begin{array}{ll}
 & (au' \mathrel{<x<} av')\backslash a \\
= & \{\text{definition of } au' \mathrel{<x<} av',\ d_1(au',av')\} \\
 & (au'|_{x} av')\backslash a \\
= & \{\text{definition of } au'|_{x} av'\} \\
 & a(u'|_{x} v')\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u'|_{x} v'
\end{array}$$

Suppose $d_2(u,v)$ holds. Then

$$
\begin{array}{ll}
 & au'\backslash a <x< av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u' <x< v' \\
= & \{\text{definition of } u' <x< v',\ d_2(au',av') \text{ implies } d_2(u',v')\} \\
 & (u'' +_x v'')(t,\tau)u'''
\end{array}
$$

and

$$
\begin{array}{ll}
 & (au' <x< av')\backslash a \\
= & \{\text{definition of } au' <x< av',\ d_2(au',av')\} \\
 & (au'' +_x av'')(t,\tau)u'''\backslash a \\
= & \{\text{definition of } au'' +_x av''\} \\
 & a(u'' +_x v'')(t,\tau)u'''\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & (u'' +_x v'')(t,\tau)u'''
\end{array}
$$

Suppose neither $d_1(u,v)$ nor $d_2(u,v)$ holds. Then

$$
\begin{array}{ll}
 & au'\backslash a <x< av'\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & u' <x< v' \\
= & \{\text{definition of } u' <x< v',\ \neg d_1(au',av') \text{ implies } \neg d_1(u',v') \text{ and} \\
 & \neg d_2(au',av') \text{ implies } \neg d_2(u',v')\} \\
 & \emptyset
\end{array}
$$

and

$$
\begin{array}{ll}
 & (au' <x< av')\backslash a \\
= & \{\text{definition of } au' <x< av',\ \neg d_1(au',av') \text{ and } \neg d_2(au',av')\} \\
 & \emptyset\backslash a \\
= & \{\text{definition of } \backslash\} \\
 & \emptyset
\end{array}
$$

3.4.4 Asymmetric Composition Preserves Traces

We show that for broad sets U and V, $\overline{U <x< V} = \overline{U} <x< \overline{V}$.

Observation 25 $d_0(u,v) \equiv d_0(\overline{u},\overline{v})$, $d_1(u,v) \equiv d_1(\overline{u},\overline{v})$, and $d_2(u,v) \equiv d_2(\overline{u},\overline{v})$.

Proof: The results follow from the definitions of d_0, d_1 and d_2.

Theorem 24 Given that U and V are broad, $\overline{U <x< V} = \overline{\overline{U} <x< \overline{V}}$.

Proof: Let $V = W \cup R$, where W is the pub-free subset of V and every sequence in R has a publication. Then,

$$
\begin{array}{cl}
 & \overline{U \;<x<\; V} \\
= & \{\text{definition of } <x< \text{ over sets}\} \\
 & \overline{(U \;<x<\; W) \cup (U \;<x<\; R)} \\
= & \{\text{distribute trace over set union}\} \\
 & \overline{U \;<x<\; W} \cup \overline{U \;<x<\; R}
\end{array}
$$

Similarly, using $\overline{V} = \overline{W} \cup \overline{R}$, we get $\overline{\overline{U} \;<x<\; \overline{V}} = \overline{\overline{U} \;<x<\; \overline{W}} \cup \overline{\overline{U} \;<x<\; \overline{R}}$. We show that $\overline{U \;<x<\; W} = \overline{\overline{U} \;<x<\; \overline{W}}$, and $\overline{U \;<x<\; R} = \overline{\overline{U} \;<x<\; \overline{R}}$.

- $\overline{U \;<x<\; W} = \overline{\overline{U} \;<x<\; \overline{W}}$:

$$
\begin{array}{cl}
 & \overline{\overline{U} \;<x<\; \overline{W}} \\
= & \{\text{since } U \text{ and } W \text{ are broad, } \overline{U} \text{ and } \overline{W} \text{ are broad, from Lemma 25, page 72;} \\
 & \quad \overline{W} \text{ is pub-free since } W \text{ is; apply Lemma 52, page 106}\} \\
 & \overline{\overline{U}|_x\overline{W}} \\
= & \{\text{from Theorem 22, page 100}\} \\
 & \overline{\overline{U|_x W}} \\
= & \{\text{simplify}\} \\
 & \overline{U|_x W} \\
= & \{\text{from Lemma 52, page 106}\} \\
 & \overline{U \;<x<\; W}
\end{array}
$$

- $\overline{U \;<x<\; R} = \overline{\overline{U} \;<x<\; \overline{R}}$: For any $u \in U$ and $w \in R$, we first show that

$$\overline{u \;<x<\; w} = \overline{\overline{u} \;<x<\; \overline{w}} \tag{**}$$

Case 1) $\neg d_2(u, w)$:
Since $w \in R$, w has a publication. So, $\neg d_1(u, w)$. From $\neg d_1(u, w)$ and $\neg d_2(u, w)$, $u \;<x<\; w = \phi$. From Observation 25, page 111, $\neg d_1(\overline{u}, \overline{w})$ and $\neg d_2(\overline{u}, \overline{w})$; so, $\overline{u} \;<x<\; \overline{w} = \phi$.

Case 2) $d_2(u, w)$: Let $u = u'(t, [m/x])u''$ and $w = w'(t, !m)w''$. From Observation 25, page 111, $d_2(\overline{u}, \overline{w})$.

$$
\begin{array}{cl}
 & \overline{u \;<x<\; w} \\
= & \{\text{expanding } u'(t, [m/x])u'' \;<x<\; w'(t, !m)w''\} \\
 & \overline{(u' +_x w')(t, \tau)u''} \\
= & \{\text{simplify}\} \\
 & \overline{(u' +_x w')}\; \overline{u''} \\
= & \{\text{from Lemma 46, page 101, } \overline{(u' +_x w')} = \overline{u'} +_x \overline{w'}\} \\
 & (\overline{u'} +_x \overline{w'})\; \overline{u''}
\end{array}
$$

Now,

$$
\begin{array}{cl}
 & \overline{\overline{u} \;<x<\; \overline{w}} \\
= & \{\text{in the above derivation, replace } u', w' \text{ and } u'' \text{ by } \overline{u'}, \overline{w'} \text{ and } \overline{u''}\}
\end{array}
$$

$$
\begin{array}{ll}
 & (\overline{\overline{u'}} +_x \overline{\overline{w'}})\,\overline{\overline{u''}} \\
= & \{\text{simplify}\} \\
 & (\overline{u'} +_x \overline{w'})\,\overline{u''} \\
= & \{\text{from above derivation}\} \\
 & \overline{u <x< w}
\end{array}
$$

Now, we show that $\overline{U <x< R} = \overline{\overline{U} <x< \overline{R}}$, by mutual inclusion.

• $\overline{U <x< R} \subseteq \overline{\overline{U} <x< \overline{R}}$:
For any $u \in U$ and $w \in R$, we show $\overline{u <x< w} \subseteq \overline{\overline{U} <x< \overline{R}}$.

$$
\begin{array}{ll}
 & \overline{u <x< w} \\
= & \{\text{from (**)}\} \\
 & \overline{\overline{u} <x< \overline{w}} \\
\subseteq & \{\text{given } u \in U,\ \overline{u} \in \overline{U}\text{; similarly, } \overline{w} \in \overline{R}\} \\
 & \overline{\overline{U} <x< \overline{R}}
\end{array}
$$

• $\overline{\overline{U} <x< \overline{R}} \subseteq \overline{U <x< R}$:
For any $p \in \overline{U}$ and $q \in \overline{R}$, we show $\overline{p <x< q} \subseteq \overline{U <x< R}$. Since $p \in \overline{U}$, there is some $u \in U$, such that $\overline{u} = p$; similarly, there is some $w \in R$, such that $\overline{w} = q$.

$$
\begin{array}{ll}
 & \overline{p <x< q} \\
= & \{\overline{u} = p,\ \overline{w} = q\} \\
 & \overline{\overline{u} <x< \overline{w}} \\
\Rightarrow & \{\text{from (**)}\} \\
 & \overline{u <x< w} \\
\Rightarrow & \{u \in U,\ w \in R\} \\
 & \overline{U <x< R}
\end{array}
$$

Chapter 4

Traces are Denotations

We are finally ready to prove the main result, that the traces of an expression can be generated from the traces of its constituent expressions. Let $*$ denote any Orc combinator, $\mid$, $>x>$ or $<x<$.

Theorem 25 For any expression f, $[\![f]\!]$ is broad.

Proof: The proof is by induction on the structure of the expression. For base expression f, $[\![f]\!]$ is broad, from Lemma 29, page 74. For $f \mid g$, We have from Section 2.3, page 43

$$
\begin{array}{ll}
 & [\![f \mid g]\!] = [\![f]\!] \mid [\![g]\!] \\
\Rightarrow & \{\text{inductively, } [\![f]\!] \text{ and } [\![g]\!] \text{ are broad;} \\
 & \quad \text{apply Theorem 17, page 79}\} \\
 & [\![f \mid g]\!] = [\![f]\!] \mid [\![g]\!] \text{ and } [\![f]\!] \mid [\![g]\!] \text{ is broad} \\
\Rightarrow & \{\text{obviously}\} \\
 & [\![f \mid g]\!] \text{ is broad}
\end{array}
$$

Proofs for the other combinators are similar: for $>x>$, use the result from Section 2.4, page 45 and apply the breadth preservation Theorem 19, page 86; for $<x<$, use the result from Section 2.5, page 51 and Theorem 23, page 106.

Theorem 26 For any Orc combinator $*$, $\langle\!\langle f * g \rangle\!\rangle = \overline{\langle\!\langle f \rangle\!\rangle * \langle\!\langle g \rangle\!\rangle}$

Proof:

$$
\begin{array}{ll}
 & \langle\!\langle f * g \rangle\!\rangle \\
= & \{\text{definition of trace}\} \\
 & \overline{[\![f * g]\!]} \\
= & \{\text{from the characterization theorems, } [\![f * g]\!] = [\![f]\!] * [\![g]\!] \\
 & \quad \text{see Section 2.3, page 43 for Symmetric Composition,} \\
 & \quad \text{see Section 2.4, page 45 for Sequential Composition,} \\
 & \quad \text{see Section 2.5, page 51 for Asymmetric Composition.}\}
\end{array}
$$

$$
\begin{aligned}
& \overline{[\![f]\!] * [\![g]\!]} \\
= \;& \{\text{from Theorem 25, page 114, } [\![f]\!] \text{ and } [\![g]\!] \text{ are broad,} \\
& \quad [\![g]\!] \text{ is substitution independent, from Observation 9, page 20;} \\
& \quad \text{for broad sets } U \text{ and } V\text{, where } V \text{ is substitution independent:} \\
& \quad \overline{U \mid V} = \overline{\overline{U} \mid \overline{V}}\text{, from Theorem 18, page 85;} \\
& \quad \overline{U > x > V} = \overline{\overline{U} > x > \overline{V}}\text{, from Theorem 20, page 90;} \\
& \quad \overline{U * V} = \overline{\overline{U} * \overline{V}}\text{, from Theorem 24, page 111}\} \\
& \overline{\overline{[\![f]\!]} * \overline{[\![g]\!]}} \\
= \;& \{\text{definition of trace}\} \\
& \overline{\langle\!\langle f \rangle\!\rangle * \langle\!\langle g \rangle\!\rangle}
\end{aligned}
$$

Lemma 55 *Suppose* $E(x) \;\; \underline{\Delta} \;\; g$ *and* $F(x) \;\; \underline{\Delta} \;\; h$*. Then* $E(p) \cong F(p)$ *if* $g \cong h$*.*

Proof: It suffices to show that $\langle\!\langle E(p) \rangle\!\rangle = \langle\!\langle F(p) \rangle\!\rangle$.

$$
\begin{aligned}
& \langle\!\langle E(p) \rangle\!\rangle \\
= \;& \{\text{Lemma 5, page 31}\} \\
& \langle\!\langle [p/x].g \rangle\!\rangle \\
= \;& \{\text{Corollary 1, page 21}\} \\
& \langle\!\langle g \rangle\!\rangle \backslash [p/x] \\
= \;& \{g \cong h \text{ by assumption}\} \\
& \langle\!\langle h \rangle\!\rangle \backslash [p/x] \\
= \;& \{\text{Corollary 1, page 21}\} \\
& \langle\!\langle [p/x].h \rangle\!\rangle \\
= \;& \{\text{Lemma 5, page 31}\} \\
& \langle\!\langle F(p) \rangle\!\rangle
\end{aligned}
$$

4.1 The Denotation of an Orc Expression

A family of functions μ_i, for $i \geq 0$, is defined that maps recursive Orc expressions to trace sets. The denotation for recursive expression f is defined as the least upper bound of the trace sets $\mu_i(f)$, where μ_i is defined by:

- $\mu_0(f) = A(0)$
- $\mu_{i+1}(f) = \begin{cases} \langle\!\langle b \rangle\!\rangle & \text{if } f = b\text{, a base expression} \\ \overline{\mu_{i+1}(g) * \mu_{i+1}(h)} & \text{if } f = g * h \\ \mu_i([p/x].g) & \text{if } f = E(p) \text{ and } \mathcal{D}(E(x)) = g \end{cases}$

And $\mu(f) = (\cup i : i \geq 0 : \mu_i(f))$.

Lemma 56 $\overline{\mu(f)} = \mu(f)$.

Proof: By induction on i. The base case is trivial, because $A(0)$ contains no τ events. Assume $\overline{\mu_i(f)} = \mu_i(f)$, for an expression f. Then $\overline{\mu_{i+1}(b)} = \overline{\langle\!\langle b \rangle\!\rangle} = \langle\!\langle b \rangle\!\rangle$. For the combinator case,

$$
\begin{array}{cl}
 & \overline{\mu_{i+1}(f * g)} \\
= & \{\text{definition of } \mu\} \\
 & \overline{\overline{\mu_{i+1}(f) * \mu_{i+1}(g)}} \\
= & \{\text{trace is idempotent}\} \\
 & \overline{\mu_{i+1}(f) * \mu_{i+1}(g)} \\
= & \{\text{definition of } \mu\} \\
 & \mu_{i+1}(f * g)
\end{array}
$$

For a defined expression $E(p)$, where $E(x) \ \underline{\Delta} \ g$,

$$
\begin{array}{cl}
 & \overline{\mu_{i+1}(E(p))} \\
= & \{\text{definition of } \mu\} \\
 & \overline{\mu_i([p/x].g)} \\
= & \{\text{induction on } i\} \\
 & \mu_i([p/x].g) \\
= & \{\text{definition of } \mu\} \\
 & \mu_{i+1}(E(p))
\end{array}
$$

Lemma 57 *For any combinator* $*$, $A(0) * A(0) = A(0)$.

Proof: By Lemma 31, page 78, $A(0) \mid A(0) = A(0)$. By Corollary 7, page 85 $A(0) >x> A(0) = A(0)$. Finally

$$
\begin{array}{cl}
 & A(0) <x< A(0) \\
= & \{\text{Lemma 52, page 106}\} \\
 & A(0)|_x A(0) \\
= & \{\text{Lemma 41, page 93}\} \\
 & A(0)
\end{array}
$$

Lemma 58 $\mu(f)$ *is broad.*

Proof: We show by induction on i and the structure of f that $\mu_i(f)$ is broad for all i, where $i \geq 0$. For $i = 0$, the result follows from Lemma 18, page 68. Next, assume for all expressions f that $\mu_i(f)$ is broad. We show $\mu_{i+1}(f)$ is broad.

- $f = b$, a base expression: $\mu_{i+1}(b) = \langle\!\langle b \rangle\!\rangle$, and $\langle\!\langle b \rangle\!\rangle$ is broad by Lemma 29, page 74.

- $f = g * h$: $\mu_{i+1}(g * h) = \overline{\mu_{i+1}(g) * \mu_{i+1}(h)}$. By structural induction, both $\mu_{i+1}(g)$ and $\mu_{i+1}(h)$ are broad. The combinators preserve breadth by Theorem 17, page 79, Theorem 19, page 86 and Theorem 23, page 106. And trace preserves breadth by Lemma 25, page 72.

- $f = E(p)$, where $E(x) \ \underline{\Delta} \ g$: $\mu_{i+1}(E(p)) = \mu_i([p/x].g)$, which is broad by induction on i.

Lemma 59 $\mu_i(a.(f \mid g)) \subseteq \mu_i(f \mid g)\backslash a$, *for all* i, *where* $i \geq 0$.

Proof: The proof is by induction on i and the subterm ordering. $\mu_0(a.(f \mid g)) = A(0)$ by definition and $\mu_i(f \mid g)\backslash a = A(0)\backslash a = A(0)$ by Observation 7, page 9. Next, assume $\mu_i(a.(f \mid g)) \subseteq \mu_i(f \mid g)\backslash a$. We show $\mu_{i+1}(a.(f \mid g)) \subseteq \mu_{i+1}(f \mid g)\backslash a$.

$$
\begin{aligned}
& \mu_{i+1}(a.(f \mid g)) \\
= \quad & \{\text{definition of substitution}\} \\
& \mu_{i+1}(a.f \mid a.g) \\
= \quad & \{\text{definition of } \mu_{i+1}\} \\
& \overline{\mu_{i+1}(a.f) \mid \mu_{i+1}(a.g)} \\
\subseteq \quad & \{\text{subterm induction}\} \\
& \overline{\mu_{i+1}(f)\backslash a \mid \mu_{i+1}(g)\backslash a} \\
\subseteq \quad & \{\text{Lemma 32, page 78}\} \\
& \overline{(\mu_{i+1}(f) \mid \mu_{i+1}(g))\backslash a} \\
\subseteq \quad & \{\text{Lemma 3, page 20}\} \\
& \overline{(\mu_{i+1}(f) \mid \mu_{i+1}(g))}\backslash a \\
= \quad & \{\text{definition of } \mu_{i+1}\} \\
& \overline{(\mu_{i+1}(f \mid g))}\backslash a
\end{aligned}
$$

Lemma 60 *$\mu_i(a.(f >x> g)) \subseteq \mu_i(f >x> g)\backslash a$, for all i, where $i \geq 0$.*

Proof: The proof is by induction on i and the subterm ordering. $\mu_0(a.(f >x> g)) = A(0)$ by definition and $\mu_i(f >x> g)\backslash a = A(0)\backslash a = A(0)$ by Observation 7, page 9. Next, assume $\mu_i(a.(f >x> g)) \subseteq \mu_i(f >x> g)\backslash a$. We show $\mu_{i+1}(a.(f >x> g)) \subseteq \mu_{i+1}(f >x> g)\backslash a$.

- Case $a = [m/x]$:

$$
\begin{aligned}
& \mu_{i+1}(a.(f >x> g)) \\
= \quad & \{\text{definition of substitution}\} \\
& \mu_{i+1}(a.f >x> g) \\
= \quad & \{\text{definition of } \mu_{i+1}\} \\
& \overline{\mu_{i+1}(a.f) >x> \mu_{i+1}(g)} \\
\subseteq \quad & \{\text{subterm induction}\} \\
& \overline{\mu_{i+1}(f)\backslash a >x> \mu_{i+1}(g)} \\
= \quad & \{\text{Lemma 38, page 87}\} \\
& \overline{(\mu_{i+1}(f) >x> \mu_{i+1}(g))\backslash a} \\
\subseteq \quad & \{\text{Lemma 3, page 20}\} \\
& \overline{(\mu_{i+1}(f) >x> \mu_{i+1}(g))}\backslash a \\
= \quad & \{\text{definition of } \mu_{i+1}\} \\
& \mu_{i+1}(f >x> g))\backslash a
\end{aligned}
$$

- Case $a = [m/y]$, where $y \neq x$:

$$
\begin{aligned}
& \mu_{i+1}(a.(f >x> g)) \\
= \quad & \{\text{definition of substitution, } x \neq y\} \\
& \mu_{i+1}(a.f >x> a.g) \\
= \quad & \{\text{definition of } \mu_{i+1}\}
\end{aligned}
$$

$$
\begin{array}{ll}
 & \overline{\mu_{i+1}(a.f) >x> \mu_{i+1}(a.g)} \\
\subseteq & \{\text{subterm induction}\} \\
 & \overline{\mu_{i+1}(f)\backslash a >x> \mu_{i+1}(g)\backslash a} \\
= & \{\text{Lemma 39, page 88}\} \\
 & \overline{(\mu_{i+1}(f) >x> \mu_{i+1}(g))\backslash a} \\
\subseteq & \{\text{Lemma 3, page 20}\} \\
 & \overline{(\mu_{i+1}(f) >x> \mu_{i+1}(g))}\backslash a \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & (\mu_{i+1}(f >x> g))\backslash a
\end{array}
$$

Lemma 61 *$\mu_i(a.(f <x< g)) \subseteq \mu_i(f <x< g)\backslash a$, for all i, where $i \geq 0$.*

Proof: The proof is by induction on i and the subterm ordering. $\mu_0(a.(f <x< g)) = A(0)$ by definition and $\mu_i(f <x< g)\backslash a = A(0)\backslash a = A(0)$ by Observation 7, page 9. Next, assume $\mu_i(a.(f <x< g)) \subseteq \mu_i(f <x< g)\backslash a$. We show $\mu_{i+1}(a.(f <x< g)) \subseteq \mu_{i+1}(f <x< g)\backslash a$.

- Case $a = [m/x]$:

$$
\begin{array}{ll}
 & \mu_{i+1}(a.(f <x< g)) \\
= & \{\text{definition of substitution}\} \\
 & \mu_{i+1}(f <x< a.g) \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \overline{\mu_{i+1}(f) <x< \mu_{i+1}(a.g)} \\
\subseteq & \{\text{subterm induction}\} \\
 & \overline{\mu_{i+1}(f) <x< \mu_{i+1}(g)\backslash a} \\
\subseteq & \{\text{Lemma 53, page 108}\} \\
 & \overline{(\mu_{i+1}(f) <x< \mu_{i+1}(g))\backslash a} \\
\subseteq & \{\text{Lemma 3, page 20}\} \\
 & \overline{(\mu_{i+1}(f) <x< \mu_{i+1}(g))}\backslash a \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_{i+1}(f <x< g))\backslash a
\end{array}
$$

- Case $a = [m/y]$, where $y \neq x$:

$$
\begin{array}{ll}
 & \mu_{i+1}(a.(f <x< g)) \\
= & \{\text{definition of substitution, } x \neq y\} \\
 & \mu_{i+1}(a.f <x< a.g) \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \overline{\mu_{i+1}(a.f) <x< \mu_{i+1}(a.g)} \\
\subseteq & \{\text{subterm induction}\} \\
 & \overline{\mu_{i+1}(f)\backslash a <x< \mu_{i+1}(g)\backslash a} \\
= & \{\text{Lemma 54, page 109}\} \\
 & \overline{(\mu_{i+1}(f) <x< \mu_{i+1}(g))\backslash a} \\
\subseteq & \{\text{Lemma 3, page 20}\}
\end{array}
$$

$$
\begin{array}{ll}
 & \overline{(\mu_{i+1}(f) <x< \mu_{i+1}(g))}\backslash a \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & (\mu_{i+1}(f <x< g))\backslash a
\end{array}
$$

Lemma 62 *For expression f, substitution a and $i \geq 0$, $\mu_i(a.f) \subseteq \mu_i(f)\backslash a$.*

Proof: By induction on i.

- $\mu_0(a.f) = \mu_0(f)\backslash a$:

$$
\begin{array}{ll}
 & \mu_0(a.f) \\
= & \{\text{definition of } \mu_0\} \\
 & A(0) \\
= & \{\text{Observation 7, page 9}\} \\
 & A(0)\backslash a \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_0(f)\backslash a
\end{array}
$$

- $\mu_{i+1}(a.f) = \mu_{i+1}(f)\backslash a$: By induction on the structure of f.

  - Case $f = b$, a base expression:

$$
\begin{array}{ll}
 & \mu_{i+1}(a.b) \\
= & \{\text{definition of } \mu_i\text{, } a.b \text{ is a base expression}\} \\
 & \langle\!\langle a.b \rangle\!\rangle \\
= & \{\text{Corollary 1, page 21}\} \\
 & \langle\!\langle b \rangle\!\rangle\backslash a \\
= & \{\text{definition of } \mu_{i+1}(b)\} \\
 & \mu_{i+1}(b)\backslash a
\end{array}
$$

  - Case $f = g * h$:

$$
\begin{array}{ll}
 & \mu_{i+1}(a.(g * h)) \\
\subseteq & \{\text{Lemma 59, page 116, Lemma 60, page 117 and Lemma 61, page 118}\} \\
 & \mu_{i+1}(g * h)\backslash a
\end{array}
$$

  - Case $f = E(p)$, where $\mathcal{D}(E(x)) = g$: We consider two cases. First, assume the substitution is $[m/p]$:

$$
\begin{array}{ll}
 & \mu_{i+1}([m/p].E(p)) \\
= & \{\text{definition of substitution}\} \\
 & \mu_{i+1}(E(m)) \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_i([m/x].g) \\
= & \{\text{only } x \text{ is free in } g\} \\
 & \mu_i([m/p].([p/x].g)) \\
= & \{\text{induction on } i\} \\
 & \mu_i([p/x].g)\backslash[m/p] \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_{i+1}(E(p))\backslash[m/p]
\end{array}
$$

Next, assume the substitution is $[m/q]$, where $q \neq p$.

$$
\begin{array}{ll}
 & \mu_{i+1}([m/q].E(p)) \\
= & \{\text{definition of substitution}\} \\
 & \mu_{i+1}(E(p)) \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_i([p/x].g) \\
= & \{\text{only } x \text{ is free in } g\} \\
 & \mu_i([m/q].[p/x].g) \\
= & \{\text{induction on } i\} \\
 & \mu_i([p/x].g)\backslash[m/q] \\
= & \{\text{definition of } \mu_{i+1}\} \\
 & \mu_{i+1}(E(p))\backslash[m/q]
\end{array}
$$

Lemma 63 *For expression f and substitution event a, $\mu(a.f) \subseteq \mu(f)\backslash a$.*

Proof:

$$
\begin{array}{ll}
 & \mu([m/x].f) \\
= & \{\text{definition of } \mu\} \\
 & (\cup i : i \geq 0 : \mu_i([m/x].f)) \\
= & \{\text{Lemma 62, page 119}\} \\
 & (\cup i : i \geq 0 : \mu_i(f)\backslash[m/x]) \\
= & \{\text{operator } \backslash \text{ is coercive}\} \\
 & (\cup i : i \geq 0 : \mu_i(f))\backslash[m/x] \\
= & \{\text{definition of } \mu\} \\
 & \mu(f)\backslash[m/x]
\end{array}
$$

Lemma 64 *Suppose $\mathcal{D}(E(x)) = g$. Then $\mu(E(p)) = \mu([p/x].g)$.*

Proof:

$$
\begin{array}{ll}
 & \mu(E(p)) \\
= & \{\text{definition of } \mu\} \\
 & (\cup i : i \geq 0 : \mu_i(E(p))) \\
= & \{\text{set theory, definition of } \mu_0(E(p))\} \\
 & A(0) \cup (\cup i : i \geq 0 : \mu_{i+1}(E(p))) \\
= & \{\text{definition of } \mu_{i+1}(E(p))\} \\
 & A(0) \cup (\cup i : i \geq 0 : \mu_i([p/x].g)) \\
= & \{\text{definition of } \mu\} \\
 & A(0) \cup \mu([p/x].g) \\
= & \{A(0) \subset \mu([p/x].g)\} \\
 & \mu([p/x].g)
\end{array}
$$

Theorem 27 *(Equivalence of Semantics) For expression f, $\langle\!\langle f \rangle\!\rangle = \mu(f)$.*

Proof: By well founded induction on the product of the subterm ordering on the structure of f and the usual ordering on the natural numbers.

- $f = b$, a base expression

$$
\begin{array}{cl}
 & \mu(b) \\
= & \{\text{definition of } \mu\} \\
 & A(0) \cup (\cup i : i \geq 0 : \mu_{i+1}(b)) \\
= & \{\text{definition of } \mu_{i+1}(b)\} \\
 & A(0) \cup (\cup i : i \geq 0 : \langle\!\langle b \rangle\!\rangle) \\
= & \{A(0) \subseteq \langle\!\langle f \rangle\!\rangle, \text{ for any expression } f\} \\
 & \langle\!\langle b \rangle\!\rangle.
\end{array}
$$

- $f = g * h$:

$$
\begin{array}{cl}
 & \langle\!\langle g * h \rangle\!\rangle \\
= & \{\text{Theorem 26, page 114}\} \\
 & \overline{\langle\!\langle g \rangle\!\rangle * \langle\!\langle h \rangle\!\rangle} \\
= & \{\text{induction}\} \\
 & \overline{\mu(g) * \mu(h)} \\
= & \{\text{definition of } \mu\} \\
 & \overline{(\cup i : i \geq 0 : \mu_i(g)) * (\cup i : i \geq 0 : \mu_i(h))} \\
= & \{\text{Theorem 10, page 41, monotonicity of } \mu_i\} \\
 & \overline{(\cup i : i \geq 0 : \mu_i(g) * \mu_i(h))} \\
= & \{\text{definition of } \mu_{i+1}(g * h),\ A(0) * A(0) = A(0) \text{ by Lemma 57, page 116}\} \\
 & \overline{A(0) \cup (\cup i : i \geq 0 : \mu_{i+1}(g * h))} \\
= & \{\text{definition of } \mu,\ \overline{\mu(g * h)} = \mu(g * h) \text{ by Lemma 56, page 115}\} \\
 & \mu(g * h)
\end{array}
$$

- $f = E(p)$, where $E(x) \;\underline{\Delta}\; g$. The proof is by mutual inclusion.

  - $\mu(E(p)) \subseteq \langle\!\langle E(p) \rangle\!\rangle$: We show, for all $i \geq 0$, that $\mu_i(E(p)) \subseteq \langle\!\langle E(p) \rangle\!\rangle$. We proceed by induction on i.

    Suppose $i = 0$ and $u \in A(0)$. Then $u \in \langle\!\langle E(p) \rangle\!\rangle$ by definition. Otherwise assume that $\mu_i(E(p)) \subseteq \langle\!\langle E(p) \rangle\!\rangle$ and consider $u \in \mu_{i+1}(E(p))$.

$$
\begin{array}{cl}
 & u \in \mu_{i+1}(E(p)) \\
\Rightarrow & \{\text{definition}\} \\
 & u \in \mu_i([p/x].g) \\
\Rightarrow & \{\text{induction on } i\} \\
 & u \in \langle\!\langle [p/x].g \rangle\!\rangle \\
\Rightarrow & \{\text{by definition, for some } v \text{ such that } \overline{v} = u\} \\
 & v \in [\![\, [p/x].g \,]\!] \\
\Rightarrow & \{\text{operational semantics, } E(x) \;\underline{\Delta}\; g\} \\
 & (0, \tau)v \in [\![\, E(p) \,]\!] \\
\Rightarrow & \{\overline{(0, \tau)v} = \overline{v} = u,\ \overline{[\![\, E(p) \,]\!]} = \langle\!\langle E(p) \rangle\!\rangle\} \\
 & u \in \langle\!\langle E(p) \rangle\!\rangle
\end{array}
$$

  - $\langle\!\langle E(p) \rangle\!\rangle \subseteq \mu(E(p))$: Consider $u \in \langle\!\langle (E(p)) \rangle\!\rangle$. Let $v \in [\![\, E(p) \,]\!]$ such that $\overline{v} = u$. We show $\overline{v} \in \mu(E(p))$, which implies $u \in \mu(E(p))$. The proof proceeds by induction on v.

Suppose $v = \epsilon$, so $\overline{v} = \epsilon$. Then the result follows from Lemma 58, page 116, breadth of μ, because ϵ is in all broad sets. Otherwise $v = av'_t$. If a is a substitution event, then $t = 0$ because $E(p)^t = \perp$ for $t > 0$.

$$
\begin{array}{ll}
 & av' \in [\![\, E(p) \,]\!] \\
\Rightarrow & \{\text{operational semantics}\} \\
 & v' \in [\![\, a.E(p) \,]\!] \\
\Rightarrow & \{\text{definition of trace}\} \\
 & \overline{v'} \in \langle\!\langle a.E(p) \rangle\!\rangle \\
\Rightarrow & \{\text{induction on } v'\} \\
 & \overline{v'} \in \mu(a.E(p)) \\
\Rightarrow & \{\mu(a.E(p)) \subseteq \mu(E(p))\backslash a \text{ by Lemma 63, page 120}\} \\
 & \overline{v'} \in \mu(E(p))\backslash a \\
\Rightarrow & \{\text{definition of } \backslash\} \\
 & a\overline{v'} \in \mu(E(p)) \\
\Rightarrow & \{a\overline{v'} = \overline{av'}\} \\
 & \overline{av'} \in \mu(E(p))
\end{array}
$$

Otherwise, by rule (Def), $E(p) \overset{0,\tau}{\rightarrow} [p/x].g \overset{v'}{\Rightarrow}$, where $E(x) \;\underline{\Delta}\; g$. So $v = (0,\tau)v'$.

$$
\begin{array}{ll}
 & (0,\tau)v' \in [\![\, E(p) \,]\!] \\
\Rightarrow & \{\text{operational semantics}\} \\
 & v' \in [\![\, [p/x].g \,]\!] \\
\Rightarrow & \{\text{definition of trace}\} \\
 & \overline{v'} \in \langle\!\langle [p/x].g \rangle\!\rangle \\
\Rightarrow & \{\text{induction on } v'\} \\
 & \overline{v'} \in \mu([p/x].g) \\
\Rightarrow & \{\text{Lemma 64, page 120}\} \\
 & \overline{v'} \in \mu(E(p)) \\
\Rightarrow & \{\overline{v'} = \overline{(0,\tau)v'}\} \\
 & \overline{(0,\tau)v'} \in \mu(E(p))
\end{array}
$$