

Pirouette - Theory

Shruti Gupta Prashant Godhwani

February 2024

1 Syntax

Locations	ℓ	\in	\mathcal{L}
Synchronization Labels	d	$::=$	$L \mid R$
Binary Operations	\odot	$::=$	$+ \mid - \mid * \mid / \mid = \mid < = \mid > = \mid ! = \mid > \mid < \mid \&\& \mid \parallel$
Choreography	C	$::=$	$() \mid X \mid \ell.e \mid C \rightsquigarrow \ell \mid \text{if } C \text{ then } C_1 \text{ else } C_2$ $\mid \ell_1[d] \rightsquigarrow \ell_2; C \mid \text{let } \ell.x := C_1 \text{ in } C_2 \mid \text{fun } X \Rightarrow C \mid C_1 \ C_2$ $\mid (C_1, C_2) \mid \text{fst } C \mid \text{snd } C \mid \text{left } C \mid \text{right } C$ $\mid \text{match } C \text{ with left } X \Rightarrow C_1; \text{right } Y \Rightarrow C_2$
Local Expressions	e	$::=$	$() \mid \text{num} \mid x \mid e_1 \text{ binop } e_2 \mid \text{let } x := e_1 \text{ in } e_2 \mid (e_1, e_2) \mid \text{fst } e$ $\mid \text{snd } e \mid \text{left } e \mid \text{right } e \mid \text{match } e \text{ with left } x \Rightarrow e_1; \text{right } y \Rightarrow e_2$
Network Expressions	E	$::=$	$X \mid () \mid \text{fun } X \Rightarrow E \mid E_1 \ E_2 \mid \text{ret}(e)$ $\mid \text{let ret}(x) := E_1 \text{ in } E_2 \mid \text{send } e \text{ to } \ell; E \mid \text{receive } x \text{ from } \ell; E$ $\mid \text{if } E_1 \text{ then } E_2 \text{ else } E_3 \mid \text{choose } d \text{ for } \ell; E$ $\mid \text{allow } \ell \text{ choice } L \Rightarrow E_1; R \Rightarrow E_2 \mid (E_1, E_2)$ $\mid \text{fst } E \mid \text{snd } E \mid \text{left } E \mid \text{right } E$ $\mid \text{match } E \text{ with left } X \Rightarrow E_1; \text{right } Y \Rightarrow E_2$
Choreographic Types	τ	$::=$	$\text{unit} \mid \ell.t \mid \tau_1 \rightarrow \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 + \tau_2$
Local Types	t	$::=$	$\text{unit} \mid \text{int} \mid \text{bool} \mid \text{string} \mid t_1 \times t_2 \mid t_1 + t_2$
Network Types	T	$::=$	$\text{unit} \mid \boxed{t} \mid T_1 \rightarrow T_2 \mid T_1 \times T_2 \mid T_1 + T_2$

2 Type System

2.1 Local Language

LOC - UNIT	LOC - VAR	LOC - PAIR	LOC - FST
$\frac{}{\Gamma \vdash () : \text{unit}}$	$\frac{x : t \in \Gamma}{\Gamma \vdash x : t}$	$\frac{\Gamma \vdash e_1 : t_1 \quad \Gamma \vdash e_2 : t_2}{\Gamma \vdash (e_1, e_2) : t_1 \times t_2}$	$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash \text{fst } e : t_1}$
LOC - SND	LOC - LEFT	LOC - RIGHT	
$\frac{\Gamma \vdash e : t_1 \times t_2}{\Gamma \vdash \text{snd } e : t_2}$	$\frac{\Gamma \vdash e_1 : t_1}{\Gamma \vdash \text{left } e_1 : t_1 + t_2}$	$\frac{\Gamma \vdash e_2 : t_2}{\Gamma \vdash \text{right } e_2 : t_1 + t_2}$	
LOC - MATCH			
$\frac{\Gamma \vdash e : t_1 + t_2 \quad \Gamma, x : t_1 \vdash e_1 : t_3 \quad \Gamma, y : t_2 \vdash e_2 : t_3}{\Gamma \vdash (\text{match } e \text{ with left } x \Rightarrow e_1 ; \text{right } y \Rightarrow e_2) : t_3}$			

2.2 Network Language

$\frac{\text{NETWORK - UNIT}}{\Gamma; \Delta \vdash () : \text{unit}}$	$\frac{\text{NETWORK - VAR} \quad X : T \in \Delta}{\Gamma; \Delta \vdash X : T}$	$\frac{\text{RET} \quad \Gamma; \Delta \vdash e : t}{\Gamma; \Delta \vdash \text{ret } (e) : \boxed{t}}$
$\frac{\text{NETWORK - FUN} \quad \Gamma; \Delta, X : T_1 \vdash E : T_2}{\Gamma; \Delta \vdash \text{fun } X \Rightarrow E : T_1 \rightarrow T_2}$	$\frac{\text{NETWORK - APP} \quad \Gamma; \Delta \vdash E_1 : T_1 \rightarrow T_2 \quad \Gamma; \Delta \vdash E_2 : T_1}{\Gamma; \Delta \vdash E_1 E_2 : T_2}$	
$\frac{\text{NETWORK - IF} \quad \Gamma; \Delta \vdash E_1 : T_1 \quad \Gamma; \Delta \vdash E_2 : T_2 \quad \Gamma; \Delta \vdash E_3 : T_2}{\Gamma; \Delta \vdash \text{if } E_1 \text{ then } E_2 \text{ else } E_3 : T_2}$		
$\frac{\text{NETWORK - DEF} \quad \Gamma; \Delta \vdash E_1 : \boxed{t} \quad \Gamma, x : t; \Delta \vdash E_2 : T_2}{\Gamma; \Delta \vdash \text{let ret } (x) = E_1 \text{ in } E_2 : T_2}$	$\frac{\text{NETWORK - SEND} \quad \Gamma; \Delta \vdash e : t \quad \Gamma; \Delta \vdash E : T}{\Gamma; \Delta \vdash \text{send } e \text{ to } \ell; E : T}$	
$\frac{\text{NETWORK - RCV} \quad \Gamma, x : t; \Delta \vdash E : T}{\Gamma; \Delta \vdash \text{receive } x \text{ from } \ell; E : T}$	$\frac{\text{NETWORK - CHOOSE} \quad \Gamma; \Delta \vdash E : T}{\Gamma; \Delta \vdash \text{choose } d \text{ for } \ell; E : T}$	
$\frac{\text{NETWORK - ALLOW} \quad \Gamma; \Delta \vdash E_1 : T \quad \Gamma; \Delta \vdash E_2 : T}{\Gamma; \Delta \vdash (\text{allow } \ell \text{ choice } L \Rightarrow E_1 ; R \Rightarrow E_2) : T}$		
$\frac{\text{NETWORK - PAIR} \quad \Gamma; \Delta \vdash E_1 : T_1 \quad \Gamma; \Delta \vdash E_2 : T_2}{\Gamma; \Delta \vdash (E_1, E_2) : T_1 \times T_2}$	$\frac{\text{NETWORK - FST} \quad \Gamma; \Delta \vdash E : T_1 \times T_2}{\Gamma; \Delta \vdash \text{fst } E : T_1}$	$\frac{\text{NETWORK - SND} \quad \Gamma; \Delta \vdash E : T_1 \times T_2}{\Gamma; \Delta \vdash \text{snd } E : T_2}$
$\frac{\text{NETWORK - LEFT} \quad \Gamma; \Delta \vdash E_1 : T_1}{\Gamma; \Delta \vdash \text{left } E_1 : T_1 + T_2}$	$\frac{\text{NETWORK - RIGHT} \quad \Gamma; \Delta \vdash E_2 : T_2}{\Gamma; \Delta \vdash \text{right } E_2 : T_1 + T_2}$	
$\frac{\text{NETWORK - MATCH} \quad \Gamma \vdash E : T_1 + T_2 \quad \Gamma; \Delta, X : T_1 \vdash E_1 : T_3 \quad \Gamma; \Delta, Y : T_2 \vdash E_2 : T_3}{\Gamma \vdash (\text{match } E \text{ with left } X \Rightarrow E_1 ; \text{right } Y \Rightarrow E_2) : T_3}$		

2.3 Choreography

$$\begin{array}{c}
\text{UNIT} \\
\frac{}{\Gamma; \Delta \vdash () : \text{unit}} \\
\\
\text{VAR} \\
\frac{X : \tau \in \Delta}{\Gamma; \Delta \vdash X : \tau} \\
\\
\text{DONE} \\
\frac{\Gamma|_\ell \vdash e : t}{\Gamma; \Delta \vdash \ell.e : \ell.t} \\
\\
\text{SEND} \\
\frac{\Gamma; \Delta \vdash C : \ell.t}{\Gamma; \Delta \vdash C \rightsquigarrow \ell_2 : \ell_2.t} \\
\\
\text{SYNC} \\
\frac{\Gamma; \Delta \vdash C : \tau}{\Gamma; \Delta \vdash \ell_1[d] \rightsquigarrow \ell_2; C : \tau} \\
\\
\text{IF} \\
\frac{\Gamma; \Delta \vdash C_1 : \ell.t \quad \Gamma; \Delta \vdash C_2 : \tau_2 \quad \Gamma; \Delta \vdash C_3 : \tau_2}{\Gamma; \Delta \vdash \text{if } C_1 \text{ then } C_2 \text{ else } C_3 : T_2} \\
\\
\text{DEF} \\
\frac{\Gamma; \Delta \vdash C_1 : \ell.t \quad \Gamma, \ell.x : t; \Delta \vdash C_2 : \tau_2}{\Gamma; \Delta \vdash \text{let } \ell.x = C_1 \text{ in } C_2 : \tau_2} \\
\\
\text{FUN} \\
\frac{\Gamma; \Delta, X : \tau_1 \vdash C : \tau_2}{\Gamma; \Delta \vdash \text{fun } X \Rightarrow C : \tau_1 \rightarrow \tau_2} \\
\\
\text{APP} \\
\frac{\Gamma; \Delta \vdash C_1 : \tau_1 \rightarrow \tau_2 \quad \Gamma; \Delta \vdash C_2 : \tau_1}{\Gamma; \Delta \vdash C_1 C_2 : \tau_2} \\
\\
\text{PAIR} \\
\frac{\Gamma; \Delta \vdash C_1 : \tau_1 \quad \Gamma; \Delta \vdash C_2 : \tau_2}{\Gamma; \Delta \vdash (C_1, C_2) : \tau_1 \times \tau_2} \\
\\
\text{FST} \\
\frac{\Gamma; \Delta \vdash C : \tau_1 \times \tau_2}{\Gamma; \Delta \vdash \text{fst } C : \tau_1} \\
\\
\text{SND} \\
\frac{\Gamma; \Delta \vdash C : \tau_1 \times \tau_2}{\Gamma; \Delta \vdash \text{snd } C : \tau_2} \\
\\
\text{LEFT} \\
\frac{\Gamma; \Delta \vdash C : \tau_1}{\Gamma; \Delta \vdash \text{left } C : \tau_1 + \tau_2} \\
\\
\text{RIGHT} \\
\frac{\Gamma; \Delta \vdash C : \tau_2}{\Gamma; \Delta \vdash \text{right } C : \tau_1 + \tau_2} \\
\\
\text{MATCH} \\
\frac{\Gamma; \Delta \vdash C : \tau_1 + \tau_2 \quad \Gamma; \Delta, X : \tau_1 \vdash C_1 : \tau_3 \quad \Gamma; \Delta, Y : \tau_2 \vdash C_2 : \tau_3}{\Gamma; \Delta \vdash (\text{match } C \text{ with left } X \Rightarrow C_1 ; \text{right } Y \Rightarrow C_2) : \tau_3}
\end{array}$$

3 Operational Semantics

3.1 Local Language

Local values $v ::= () \mid \text{num} \mid v_1 \odot v_2 \mid (v_1, v_2) \mid \text{left } v \mid \text{right } v$

$$\begin{array}{c}
\text{BINOP 1} \quad \frac{e_1 \rightarrow_l e'_1}{e_1 \odot e_2 \rightarrow_l e'_1 \tilde{\odot} e_2} \quad \text{BINOP 2} \quad \frac{e_2 \rightarrow_l e'_2}{v \odot e_2 \rightarrow_l v \tilde{\odot} e'_2} \quad \text{BINOP 3} \quad \frac{}{v_1 \odot v_2 \rightarrow_l v_1 \tilde{\odot} v_2 \text{ (/ } \notin \odot \text{)}} \\
\\
\text{LOC - LET 1} \quad \frac{e_1 \rightarrow_l e'_1}{\text{let } x := e_1 \text{ in } e_2 \rightarrow_l \text{let } x := e'_1 \text{ in } e_2} \quad \text{LOC - LET 2} \quad \frac{}{\text{let } x := v \text{ in } e \rightarrow_l e[x \mapsto v]} \\
\\
\text{LOC - PAIR 1} \quad \frac{e_1 \rightarrow_l e'_1}{(e_1, e_2) \rightarrow_l (e'_1, e_2)} \quad \text{LOC - PAIR 2} \quad \frac{e_2 \rightarrow_l e'_2}{(v, e_2) \rightarrow_l (v, e'_2)} \quad \text{LOC - FST} \quad \frac{e \rightarrow_l e'}{\text{fst } e \rightarrow_l \text{fst } e'} \\
\\
\text{LOC - PAIR ELIM 1} \quad \frac{}{\text{fst } (v_1, v_2) \rightarrow_l v_1} \\
\\
\text{LOC - SND} \quad \frac{e \rightarrow_l e'}{\text{snd } e \rightarrow_l \text{snd } e'} \quad \text{LOC - PAIR ELIM 2} \quad \frac{}{\text{snd } (v_1, v_2) \rightarrow_l v_2} \quad \text{LOC - LEFT} \quad \frac{e \rightarrow_l e'}{\text{left } e \rightarrow_l \text{left } e'} \quad \text{LOC - RIGHT} \quad \frac{e \rightarrow_l e'}{\text{right } e \rightarrow_l \text{right } e'} \\
\\
\text{LOC - MATCH} \quad \frac{e \rightarrow_l e'}{(\text{match } e \text{ with left } x \Rightarrow e_1 ; \text{right } y \Rightarrow e_2) \rightarrow_l (\text{match } e' \text{ with left } x \Rightarrow e_1 ; \text{right } y \Rightarrow e_2)} \\
\\
\text{LOC - SUM ELIM 1} \quad \frac{}{(\text{match } (\text{left } v) \text{ with left } x \Rightarrow e_1 ; \text{right } y \Rightarrow e_2) \rightarrow_l e_1 [x \mapsto v]} \\
\\
\text{LOC - SUM ELIM 2} \quad \frac{}{(\text{match } (\text{right } v) \text{ with left } x \Rightarrow e_1 ; \text{right } y \Rightarrow e_2) \rightarrow_l e_2 [y \mapsto v]}
\end{array}$$

3.2 NetIR

$$\begin{array}{l}
\text{fst}(E_1, E_2) \rightarrow E_1 \quad \text{snd}(E_1, E_2) \rightarrow E_2 \\
\\
(\text{match inl } E \text{ with inl } X \Rightarrow E_1; \text{inr } Y \Rightarrow E_2) \rightarrow E_1 [X \mapsto E] \\
\\
(\text{match inr } E \text{ with inl } X \Rightarrow E_1; \text{inr } Y \Rightarrow E_2) \rightarrow E_2 [Y \mapsto E]
\end{array}$$

3.3 Choreography

Choreographic Values $V ::= () \mid \ell.v \mid (V_1, V_2) \mid \mathbf{left} \ V \mid \mathbf{right} \ V \mid \mathbf{fun} \ X \Rightarrow C$

$$\begin{array}{c}
\text{ASSOC} \quad \frac{e \rightarrow_l e'}{\ell.e \rightarrow_g \ell.e'} \qquad \text{SEND 1} \quad \frac{C \rightarrow_g C'}{C \rightsquigarrow \ell \rightarrow_g C' \rightsquigarrow \ell} \qquad \text{SEND 2} \quad \frac{}{\ell.v \rightsquigarrow \ell' \rightarrow_g \ell'.v} \\
\\
\text{SYNC} \quad \frac{}{\ell_1[d] \rightsquigarrow \ell_2; C \rightarrow_g C} \qquad \text{LET 1} \quad \frac{C_1 \rightarrow_g C'_1}{\text{let } X := C_1 \text{ in } C_2 \rightarrow_g \text{let } X := C'_1 \text{ in } C_2} \\
\\
\text{LET 2} \quad \frac{}{\text{let } X := V \text{ in } C \rightarrow_g C[X \mapsto V]} \\
\\
\text{IF 1} \quad \frac{C \rightarrow_g C'}{\text{if } C \text{ then } C_1 \text{ else } C_2 \rightarrow_g \text{if } C' \text{ then } C_1 \text{ else } C_2} \\
\\
\text{IF 2} \quad \frac{}{\text{if } \ell.\text{true} \text{ then } C_1 \text{ else } C_2 \rightarrow_g C_1} \qquad \text{IF 3} \quad \frac{}{\text{if } \ell.\text{false} \text{ then } C_1 \text{ else } C_2 \rightarrow_g C_2} \\
\\
\text{APP 1} \quad \frac{C_1 \rightarrow_g C'_1}{C_1 C_2 \rightarrow_g C'_1 C_2} \qquad \text{APP 2} \quad \frac{C_2 \rightarrow_g C'_2}{(\text{fun } X \Rightarrow C) C_2 \rightarrow_g (\text{fun } X \Rightarrow C) C'_2} \\
\\
\text{APP 3} \quad \frac{}{(\text{fun } X \Rightarrow C) V \rightarrow_g C[X \mapsto V]} \qquad \text{PAIR 1} \quad \frac{C_1 \rightarrow_g C'_1}{(C_1, C_2) \rightarrow_g (C'_1, C_2)} \\
\\
\text{PAIR 2} \quad \frac{C_2 \rightarrow_g C'_2}{(V, C_2) \rightarrow_g (V, C'_2)} \qquad \text{FST} \quad \frac{C \rightarrow_g C'}{\text{fst } C \rightarrow_g \text{fst } C'} \qquad \text{PAIR ELIM 1} \quad \frac{}{\text{fst } (V_1, V_2) \rightarrow_g V_1} \\
\\
\text{SND} \quad \frac{C \rightarrow_g C'}{\text{snd } C \rightarrow_g \text{snd } C'} \qquad \text{PAIR ELIM 2} \quad \frac{}{\text{snd } (V_1, V_2) \rightarrow_g V_2} \qquad \text{LEFT} \quad \frac{C \rightarrow_g C'}{\text{left } C \rightarrow_g \text{left } C'} \\
\\
\text{RIGHT} \quad \frac{C \rightarrow_g C'}{\text{right } C \rightarrow_g \text{right } C'} \\
\\
\text{MATCH} \quad \frac{C \rightarrow_g C'}{(\text{match } C \text{ with left } X \Rightarrow C_1 ; \text{right } Y \Rightarrow C_2) \rightarrow_g (\text{match } C' \text{ with left } X \Rightarrow C_1 ; \text{right } Y \Rightarrow C_2)} \\
\\
\text{SUM ELIM 1} \quad \frac{}{(\text{match } (\text{left } V) \text{ with left } X \Rightarrow C_1 ; \text{right } Y \Rightarrow C_2) \rightarrow_g C_1 [X \mapsto V]} \\
\\
\text{SUM ELIM 2} \quad \frac{}{(\text{match } (\text{right } V) \text{ with left } X \Rightarrow C_1 ; \text{right } Y \Rightarrow C_2) \rightarrow_g C_2 [Y \mapsto V]}
\end{array}$$

4 Theorems

Theorem 1. (*Local Progress*): For every expression $\cdot \vdash e : t$ either $\exists e'. e \rightarrow_l e'$ or e is a value

Proof. We will start with induction on e

Case $e = ()$

$()$ is a value and we are done

Case $e = \text{num}$

num is a value and we are done

Case $e = e_1 \odot e_2$

IH1 e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then $e_1 \odot e_2 \rightarrow_l e'_1 \tilde{\odot} e_2$ using binop 1 rule

if e_1 is a value v_1 , IH2 e_2 is either a value or $\exists e'_2. e_2 \rightarrow_l e'_2$

if e_2 is a value v_2 , then $v_1 \odot v_2 \rightarrow_l v_1 \tilde{\odot} v_2$, which is a value using the binop 3 rule

if $e_2 \rightarrow_l e'_2$ then $v_1 \odot e_2 \rightarrow_l v_1 \tilde{\odot} e'_2$ using binop 2 rule

Case $e = \text{let } x := e_1 \text{ in } e_2$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then $\text{let } x := e_1 \text{ in } e_2 \rightarrow_l \text{let } x := e'_1 \text{ in } e_2$ using loc - let 1 rule

if e_1 is a value v , then $\text{let } x := v \text{ in } e_2 \rightarrow_l e_2[x \mapsto v]$ using loc - let 2 rule

Case $e = (e_1, e_2)$

IH1 e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then $(e_1, e_2) \rightarrow_l (e'_1, e_2)$ using loc - pair 1 rule

if e_1 is a value v_1 , IH2 e_2 is either a value or $\exists e'_2. e_2 \rightarrow_l e'_2$

if e_2 is a value v_2 , then (v_1, v_2) is a value

if $e_2 \rightarrow_l e'_2$ then $(v_1, e_2) \rightarrow_l (v_1, e'_2)$ using loc - pair 2 rule

Case $e = \text{fst } e_1$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then $\text{fst } e_1 \rightarrow_l \text{fst } e'_1$ using loc - fst rule

if e_1 is a value v , this means $v = (v_1, v_2)$ then $\text{fst } (v_1, v_2) \rightarrow_l v_1$ using loc - pair elim 1 rule

Case $e = \text{snd } e_1$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then $\text{snd } e_1 \rightarrow_l \text{snd } e'_1$ using loc - snd rule

if e_1 is a value v , this means $v = (v_1, v_2)$ then $\text{snd } (v_1, v_2) \rightarrow_l v_2$ using loc - pair elim 2 rule

Case $e = \text{left } e_1$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$

if $e_1 \rightarrow_l e'_1$ then left $e_1 \rightarrow_l$ left e'_1 using loc - left rule
 if e_1 is a value v , then left v is a value

Case $e = \text{right } e_1$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$
 if $e_1 \rightarrow_l e'_1$ then right $e_1 \rightarrow_l$ right e'_1 using loc - right rule
 if e_1 is a value v , then right v is a value

Case $e = \text{match } e_1 \text{ with left } x \Rightarrow e_2; \text{right } y \Rightarrow e_3$

IH e_1 is either a value or $\exists e'_1. e_1 \rightarrow_l e'_1$
 if $e_1 \rightarrow_l e'_1$ then match e_1 with left $x \Rightarrow e_2$; right $y \Rightarrow e_3 \rightarrow_l$ match e'_1 with left $x \Rightarrow e_2$; right $y \Rightarrow e_3$ using loc - match rule
 if e_1 is a value, then e_1 is either left v or right v
 if $e_1 = \text{left } v$ then match (left v) with left $x \Rightarrow e_2$; right $y \Rightarrow e_3 \rightarrow_l e_2[x \mapsto v]$ using loc - sum elim 1 rule
 if $e_1 = \text{right } v$ then match (right v) with left $x \Rightarrow e_2$; right $y \Rightarrow e_3 \rightarrow_l e_3[y \mapsto v]$ using loc - sum elim 2 rule

Theorem 2. (Local Preservation): If $\Gamma \vdash e : t$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t$
Proof. We will start with induction on $\Gamma \vdash e : t$

Case $\Gamma \vdash e : \text{unit}$

$\Gamma \vdash () : \text{unit}$. $()$ doesn't take a step and we are done

Case $x : t$

$\Gamma \vdash x : t$. x doesn't take a step and we are done

Case $\Gamma \vdash e : t_1 X t_2$

This means $e = (e_1, e_2)$

IH If $\Gamma \vdash e_1 : t_1$ and $e_1 \rightarrow e'_1$ then $\Gamma \vdash e'_1 : t_1$

Now we know, $(e_1, e_2) \rightarrow (e'_1, e_2)$ and $\Gamma \vdash (e_1, e_2) : t_1 X t_2$

Using IH we can say $\Gamma \vdash (e'_1, e_2) : t_1 X t_2$

Case $\Gamma \vdash \text{fst } e : t_1$

This means $e : t_1 X t_2$

IH If $\Gamma \vdash e : t_1 X t_2$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t_1 X t_2$

Now we know, $\text{fst } e \rightarrow \text{fst } e'$ and $\Gamma \vdash \text{fst } e : t_1$

Using IH we can say $\Gamma \vdash \text{fst } e' : t_1$

Case $\Gamma \vdash \text{snd } e : t_2$

This means $e : t_1 X t_2$

IH If $\Gamma \vdash e : t_1 X t_2$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t_1 X t_2$

Now we know, $\text{snd } e \rightarrow \text{snd } e'$ and $\Gamma \vdash \text{snd } e : t_2$

Using IH we can say $\Gamma \vdash \text{snd } e' : t_2$

Case $\Gamma \vdash \text{left } e : t_1 + t_2$

This means $e : t_1$

IH If $\Gamma \vdash e : t_1$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t_1$

Now we know, left $e \rightarrow \text{left } e'$ and $\Gamma \vdash \text{left } e : t_1 + t_2$

Using IH we can say $\Gamma \vdash \text{left } e' : t_1 + t_2$

Case $\Gamma \vdash \text{right } e : t_1 + t_2$

This means $e : t_2$

IH If $\Gamma \vdash e : t_2$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t_2$

Now we know, right $e \rightarrow \text{right } e'$ and $\Gamma \vdash \text{right } e : t_1 + t_2$

Using IH we can say $\Gamma \vdash \text{right } e' : t_1 + t_2$

Case $\Gamma; x : t_1, y : t_2 \vdash \text{match } e \text{ with left } x \Rightarrow e_2; \text{right } y \Rightarrow e_3 : t_3$

This means $e : t_1 + t_2$

IH If $\Gamma \vdash e : t_1 + t_2$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t_1 + t_2$

Now we know, match e with left $x \Rightarrow e_2$; right $y \Rightarrow e_3 \rightarrow \text{match } e' \text{ with left } x \Rightarrow e_2$; right $y \Rightarrow e_3$ and $\Gamma \vdash \text{match } e \text{ with left } x \Rightarrow e_2$; right $y \Rightarrow e_3 : t_3$

Using IH we can say $\Gamma \vdash \text{match } e' \text{ with left } x \Rightarrow e_2$; right $y \Rightarrow e_3 : t_3$

Theorem 3. (Progress): For every choreography $\cdot \vdash C : \tau$ either $\exists C'. C \rightarrow C'$ or C is a value

Proof. We will start with induction on C

Case $C = ()$

$()$ is a value and we are done

Case $C = \ell.e$

we know from local progress, e is either a value or $\exists e'. e \rightarrow e'$

So, if e is a value, $\ell.e$ is a value and we are done

If $\exists e'. e \rightarrow e'$ then, $\ell.e \rightarrow \ell.e'$

Case $C = C_1 \rightsquigarrow \ell$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$

if $C_1 \rightarrow C'_1$ then $C_1 \rightsquigarrow \ell \rightarrow C'_1 \rightsquigarrow \ell$

if C_1 is a value, $C_1 = \ell_1.v$ and we know that $\ell_1.v \rightsquigarrow \ell \rightarrow \ell.v$

Case $C = \ell_1[d] \rightsquigarrow \ell_2; C_1$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$

if $C_1 \rightarrow C'_1$ then $\ell_1[d] \rightsquigarrow \ell_2; C_1 \rightarrow \ell_1[d] \rightsquigarrow \ell_2; C'_1$

if C_1 is a value V , then $\ell_1[d] \rightsquigarrow \ell_2; C_1$ steps to the value V

Case $C = \text{if } C_1 \text{ then } C_2 \text{ else } C_3$

Case $C = (C_1, C_2)$

IH1 C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$
 if $C_1 \rightarrow C'_1$ then $(C_1, C_2) \rightarrow (C'_1, C_2)$
 if C_1 is a value, IH2 C_2 is either a value or $\exists C'_2. C_2 \rightarrow C'_2$
 if C_2 is a value, then (C_1, C_2) is a value
 if $C_2 \rightarrow C'_2$ then $(C_1, C_2) \rightarrow (C_1, C'_2)$

Case $C = \text{fst } C_1$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$
 if $C_1 \rightarrow C'_1$ then $\text{fst } C_1 \rightarrow \text{fst } C'_1$
 if C_1 is a value, this means C_1 is a pair of values as $C_1 : \tau_1 \times \tau_2$ so if $C_1 = (v_1, v_2)$
 then $\text{fst } C_1 \rightarrow v_1$

Case $C = \text{snd } C_1$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$
 if $C_1 \rightarrow C'_1$ then $\text{snd } C_1 \rightarrow \text{snd } C'_1$
 if C_1 is a value, this means C_1 is a pair of values as $C_1 : \tau_1 \times \tau_2$ so if $C_1 = (v_1, v_2)$
 then $\text{snd } C_1 \rightarrow v_2$

Case $C = \text{left } C_1$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$
 if $C_1 \rightarrow C'_1$ then $\text{left } C_1 \rightarrow \text{left } C'_1$
 if C_1 is a value, then $\text{left } C_1$ is a value

Case $C = \text{right } C_1$

IH C_1 is either a value or $\exists C'_1. C_1 \rightarrow C'_1$
 if $C_1 \rightarrow C'_1$ then $\text{right } C_1 \rightarrow \text{right } C'_1$
 if C_1 is a value, then $\text{right } C_1$ is a value

Theorem 4. (Preservation): If $\Gamma; \Delta \vdash C : \tau$ and $C \rightarrow C'$ then $\Gamma; \Delta \vdash C' : \tau$
Proof. We will start with induction on $\Gamma; \Delta \vdash C : \tau$

Case $\Gamma; \Delta \vdash C : \text{unit}$

$\Gamma; \Delta \vdash () : \text{unit}$. $()$ doesn't take a step and we are done

Case $\Gamma; \Delta \vdash X : \tau$

$\Gamma; \Delta \vdash X : \tau$. X doesn't take a step and we are done

Case $\Gamma; \Delta \vdash C : \ell.t$

This means $C = \ell.e$

Using local preservation we can say that, if $\Gamma \vdash e : t$ and $e \rightarrow e'$ then $\Gamma \vdash e' : t$

Now we know, $\ell.e \rightarrow \ell.e'$ and $\Gamma; \Delta \vdash C : \ell.t$

Using local preservation, $\Gamma; \Delta \vdash C' : \ell.t$ where $C' = \ell.e'$

Case $\Gamma; \Delta \vdash C : \tau_1 \times \tau_2$

This means $C = (C_1, C_2)$

IH If $\Gamma; \Delta \vdash C_1 : \tau_1$ and $C_1 \rightarrow C'_1$ then $\Gamma; \Delta \vdash C'_1 : \tau_1$
 Now we know, $(C_1, C_2) \rightarrow (C'_1, C_2)$ and $\Gamma; \Delta \vdash (C_1, C_2) : \tau_1 \times \tau_2$
 Using IH we can say $\Gamma; \Delta \vdash (C'_1, C_2) : \tau_1 \times \tau_2$

Case $\Gamma; \Delta \vdash \text{fst } C : \tau_1$

This means $C : \tau_1 \times \tau_2$

IH If $\Gamma; \Delta \vdash C : \tau_1 \times \tau_2$ and $C \rightarrow c'$ then $\Gamma; \Delta \vdash C' : \tau_1 \times \tau_2$

Now we know, $\text{fst } C \rightarrow \text{fst } C'$ and $\Gamma; \Delta \vdash \text{fst } C : \tau_1$

Using IH we can say $\Gamma; \Delta \vdash \text{fst } C' : \tau_1$

Case $\Gamma; \Delta \vdash \text{snd } C : \tau_2$

This means $C : \tau_1 \times \tau_2$

IH If $\Gamma; \Delta \vdash C : \tau_1 \times \tau_2$ and $C \rightarrow C'$ then $\Gamma; \Delta \vdash C' : \tau_1 \times \tau_2$

Now we know, $\text{snd } C \rightarrow \text{snd } C'$ and $\Gamma; \Delta \vdash \text{snd } C : \tau_2$

Using IH we can say $\Gamma; \Delta \vdash \text{snd } C' : \tau_2$

Case $\Gamma; \Delta \vdash \text{left } C : \tau_1 + \tau_2$

This means $C : \tau_1$

IH If $\Gamma; \Delta \vdash C : \tau_1$ and $C \rightarrow C'$ then $\Gamma; \Delta \vdash C' : \tau_1$

Now we know, $\text{left } C \rightarrow \text{left } C'$ and $\Gamma; \Delta \vdash \text{left } C : \tau_1 + \tau_2$

Using IH we can say $\Gamma; \Delta \vdash \text{left } C' : \tau_1 + \tau_2$

Case $\Gamma; \Delta \vdash \text{right } C : \tau_1 + \tau_2$

This means $C : \tau_2$

IH If $\Gamma; \Delta \vdash C : \tau_2$ and $C \rightarrow C'$ then $\Gamma; \Delta \vdash C' : \tau_2$

Now we know, $\text{right } C \rightarrow \text{right } C'$ and $\Gamma; \Delta \vdash \text{right } C : \tau_1 + \tau_2$

Using IH we can say $\Gamma; \Delta \vdash \text{right } C' : \tau_1 + \tau_2$

Case $\Gamma; \Delta \vdash \text{match } C \text{ with left } X \Rightarrow C_2; \text{right } Y \Rightarrow C_3 : \tau_3$

This means $C : \tau_1 + \tau_2$

IH If $\Gamma; \Delta \vdash C : \tau_1 + \tau_2$ and $C \rightarrow C'$ then $\Gamma; \Delta \vdash C' : \tau_1 + \tau_2$

Now we know, $\text{match } C \text{ with left } X \Rightarrow C_2; \text{right } Y \Rightarrow C_3 \rightarrow \text{match } C' \text{ with left } X \Rightarrow C_2; \text{right } Y \Rightarrow C_3$ and $\Gamma; \Delta \vdash \text{match } C \text{ with left } X \Rightarrow C_2; \text{right } Y \Rightarrow C_3 : \tau_3$

Using IH we can say, $\Gamma; \Delta \vdash \text{match } C' \text{ with left } X \Rightarrow C_2; \text{right } Y \Rightarrow C_3 : \tau_3$

5 Glossary

ℓ involved in $\tau = \ell \in \text{locs}(\tau)$

$\ell \in \text{locs}(\tau) = \text{getLoc}$ is a function that recursively traverses over τ to construct $\text{locs}(\tau)$

$$locs(\tau) = \begin{cases} \phi & \text{if } \tau = \mathbf{unit} \\ \{\ell\} & \text{if } \tau = \ell.e \\ \text{getLoc } \tau_1 \cup \text{getLoc } \tau_2 & \text{if } \tau = \tau_1 \rightarrow \tau_2 \text{ or } \tau_1 + \tau_2 \text{ or } \tau_1 \times \tau_2 \end{cases}$$

6 Endpoint Projection

$$\llbracket (C_1, C_2) \rrbracket_\ell = \begin{cases} (\llbracket C_1 \rrbracket_\ell, \llbracket C_2 \rrbracket_\ell) & \text{if } (C_1, C_2) : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C_1, C_2, \tau_1, \tau_2 \\ \text{let } x = \llbracket C_1 \rrbracket_\ell \text{ in} & \text{if } (C_1, C_2) : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C_1, C_2 \text{ and } \tau_1 \text{ but not in } \tau_2 \\ \text{let } _ = \llbracket C_2 \rrbracket_\ell \text{ in } x & \\ \llbracket C_1 \rrbracket_\ell; \llbracket C_2 \rrbracket_\ell & \text{if } (C_1, C_2) : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C_1 \text{ or } C_2 \\ () & \text{otherwise} \end{cases}$$

$$\llbracket \mathbf{fst } C \rrbracket_\ell = \begin{cases} \mathbf{fst } \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C, \tau_1 \text{ and } \tau_2 \\ \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C \\ () & \text{otherwise} \end{cases}$$

$$\llbracket \mathbf{snd } C \rrbracket_\ell = \begin{cases} \mathbf{snd } \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C, \tau_1 \text{ and } \tau_2 \\ \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C \\ () & \text{otherwise} \end{cases}$$

$$\llbracket \mathbf{inl } C \rrbracket_\ell = \begin{cases} \mathbf{inl } \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \text{ and } \ell \text{ is involved in } C \text{ and } \tau_1 \\ \llbracket C \rrbracket_\ell & \text{if } C : \tau_1 \text{ and } \ell \text{ is involved in } C \text{ but not in } \tau_1 \\ () & \text{if } C : \tau_1 \text{ and } \ell \text{ is not involved in } C \text{ and } \tau_1 \end{cases}$$

$$\llbracket \mathbf{inr } C \rrbracket_\ell = \begin{cases} \mathbf{inr } \llbracket C \rrbracket_\ell & \text{if } C : \tau_2 \text{ and } \ell \text{ is involved in } C \text{ and } \tau_2 \\ \llbracket C \rrbracket_\ell & \text{if } C : \tau_2 \text{ and } \ell \text{ is involved in } C \text{ but not in } \tau_2 \\ () & \text{if } C : \tau_2 \text{ and } \ell \text{ is not involved in } C \text{ and } \tau_2 \end{cases}$$

$$\llbracket \mathbf{match } C \text{ in } \mathbf{inl } \mathbf{X} \Rightarrow C_1; \mathbf{inr } \mathbf{Y} \Rightarrow C_2 \rrbracket_\ell$$

$$= \begin{cases} \mathbf{match } \llbracket C \rrbracket_\ell \text{ in } \mathbf{inl } \mathbf{X} \Rightarrow \llbracket C_1 \rrbracket_\ell; \mathbf{inr } \mathbf{Y} \Rightarrow \llbracket C_2 \rrbracket_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is involved in both } \tau_1 \text{ and } \tau_2 \\ \llbracket C \rrbracket_\ell; \llbracket C_1 \rrbracket_\ell \sqcup \llbracket C_2 \rrbracket_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or } C \\ \llbracket C_1 \rrbracket_\ell \sqcup \llbracket C_2 \rrbracket_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is not involved in } C, \tau_1 \text{ and } \tau_2 \end{cases}$$

7 Type Projection

$$\llbracket \mathbf{unit} \rrbracket_\ell = \mathbf{unit}$$

$$\llbracket \ell_1.t \rrbracket_{\ell_2} = \begin{cases} t & \text{if } \ell_1 = \ell_2 \\ \mathbf{unit} & \text{otherwise} \end{cases}$$

$$\llbracket \tau_1 \rightarrow \tau_2 \rrbracket_\ell = \begin{cases} \llbracket \tau_1 \rrbracket_\ell \rightarrow \llbracket \tau_2 \rrbracket_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or both} \\ \mathbf{unit} & \text{otherwise} \end{cases}$$

$$\llbracket \tau_1 + \tau_2 \rrbracket_\ell = \begin{cases} \llbracket \tau_1 \rrbracket_\ell + \llbracket \tau_2 \rrbracket_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or both} \\ \mathbf{unit} & \text{otherwise} \end{cases}$$

$$\llbracket \tau_1 \times \tau_2 \rrbracket_\ell = \begin{cases} \llbracket \tau_1 \rrbracket_\ell \times \llbracket \tau_2 \rrbracket_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ and } \tau_2 \\ \llbracket \tau_1 \rrbracket_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ but not } \tau_2 \\ \llbracket \tau_2 \rrbracket_\ell & \text{if } \ell \text{ is involved in } \tau_2 \text{ but not } \tau_1 \\ \mathbf{unit} & \text{otherwise} \end{cases}$$

8 Lemmas

Lemma 1 : If ℓ is not involved in τ then $\llbracket \tau \rrbracket_\ell = \mathbf{unit}$

Proof : By Induction on τ

Case $\tau = \mathbf{unit}$:

From type projection we know, $\llbracket \mathbf{unit} \rrbracket_\ell = \mathbf{unit}$

Case $\tau = \ell.t$:

Here we know that, $\ell_1 \neq \ell$ as ℓ is not involved in τ

so using type projection for $\llbracket \ell_1.t \rrbracket_\ell$

we can say, $\llbracket \ell_1.t \rrbracket_\ell = \mathbf{unit}$

Case $\tau = \llbracket \tau_1 \rightarrow \tau_2 \rrbracket_\ell$:

By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2

so using type projection for $\llbracket \tau_1 \rightarrow \tau_2 \rrbracket_\ell$

we can say, $\llbracket \tau_1 \rightarrow \tau_2 \rrbracket_\ell = \mathbf{unit}$

Case $\tau = \llbracket \tau_1 + \tau_2 \rrbracket_\ell$:

By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2

so using type projection for $\llbracket \tau_1 + \tau_2 \rrbracket_\ell$

we can say, $\llbracket \tau_1 + \tau_2 \rrbracket_\ell = \mathbf{unit}$

Case $\tau = \llbracket \tau_1 \times \tau_2 \rrbracket_\ell$:

By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2

so using type projection for $\llbracket \tau_1 \times \tau_2 \rrbracket_\ell$

we can say, $\llbracket \tau_1 \times \tau_2 \rrbracket_\ell = \mathbf{unit}$

Lemma 2 : If $\ell \notin \text{locs}(C)$ then $\llbracket C \rrbracket_\ell = ()$

Lemma 3 : If $\vdash C : \tau$, then $\vdash \llbracket C \rrbracket_\ell : \llbracket \tau \rrbracket_\ell$