Pirouette

Shruti Gupta Prashant Godhwani

September 2023

1 Syntax

Locations	ℓ	$\in$	$\mathcal{L}$
Synchronization Labels	d	$::=$	$AllVariables$
Choreography	C	$::=$	$() \mid X \mid \ell.e \mid \ell_1.e \rightsquigarrow \ell_2.x;\ C \mid \textbf{if}\ \ell.e\ \textbf{then}\ C_1\ \textbf{else}\ C_2$
		$\mid$	$\ell_1[d] \rightsquigarrow \ell_2;\ C \mid \textbf{let}\ \ell.x \coloneqq C_1\ \textbf{in}\ C_2 \mid \textbf{fun}\ F(X) \coloneqq C \mid C_1\ C_2$
		$\mid$	$(C_1, C_2) \mid \textbf{fst}\ C \mid \textbf{snd}\ C \mid \textbf{inl}\ C \mid \textbf{inr}\ C$
		$\mid$	$\textbf{match}\ C\ \textbf{with}\ inl\ X \to C_1\textbf{;}\ inr\ Y \to C_2$
Control Expressions	E	$::=$	$\ldots \mid () \mid (E_1, E_2) \mid \textbf{fst}\ E \mid \textbf{snd}\ E \mid \textbf{inl}\ E \mid \textbf{inr}\ E$
		$\mid$	$\textbf{match}\ E\ \textbf{with}\ inl\ X \to E_1\textbf{;}\ inr\ Y \to E_2$
Choreographic Types	τ	$::=$	$\textbf{unit} \mid \ell.t \mid \tau_1 \to \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 + \tau_2$
Local Types	t	$::=$	$\textbf{unit} \mid \textbf{int} \mid \textbf{string} \mid \textbf{bool} \mid t_1 \to t_2 \mid t_1 \times t_2 \mid t_1 + t_2$

2 Type System

$$\frac{}{\Gamma, \Delta \vdash (\ ) : \mathsf{unit}}\ \textsc{Unit} \qquad \frac{\Gamma, \Delta \vdash C_1 : \tau_1 \quad \Gamma, \Delta \vdash C_2 : \tau_2}{\Gamma, \Delta \vdash (C_1, C_2) : \tau_1 \times \tau_2}\ \textsc{Pair} \qquad \frac{\Gamma, \Delta \vdash (C_1, C_2) : \tau_1 \times \tau_2}{\Gamma, \Delta \vdash \mathsf{fst}\ (C_1, C_2) : \tau_1}\ \textsc{Fst}$$

$$\frac{\Gamma, \Delta \vdash (C_1, C_2) : \tau_1 \times \tau_2}{\Gamma, \Delta \vdash \mathsf{snd}\ (C_1, C_2) : \tau_2}\ \textsc{Snd} \qquad \frac{\Gamma, \Delta \vdash C : \tau_1}{\Gamma, \Delta \vdash \mathsf{inl}\ C : \tau_1 + \tau_2}\ \textsc{Inl} \qquad \frac{\Gamma, \Delta \vdash C : \tau_2}{\Gamma, \Delta \vdash \mathsf{inr}\ C : \tau_1 + \tau_2}\ \textsc{Inr}$$

$$\frac{\Gamma, \Delta \vdash C : \tau_1 + \tau_2 \quad \Gamma, \Delta, \mathsf{X} : \tau_1 \vdash C_1 : \tau_3 \quad \Gamma, \Delta, \mathsf{Y} : \tau_2 \vdash C_2 : \tau_3}{\Gamma, \Delta \vdash (\mathsf{match\ C\ with\ inl\ X} \to C_1\ ;\ \mathsf{inr\ Y} \to C_2) : \tau_3}\ \textsc{Match}$$

3 Operational Semantics

3.1 Control Language

$$\mathsf{fst}(E_1, E_2) \to E_1 \qquad \mathsf{snd}(E_1, E_2) \to E_2$$

$$(\mathsf{match\ \ inl\ \ E\ \ with\ \ inl\ X} \to E_1;\ \mathsf{inr\ Y} \to E_2) \to E_1\ [X \mapsto E]$$

$$(\mathsf{match\ \ inr\ \ E\ \ with\ \ inl\ X} \to E_1;\ \mathsf{inr\ Y} \to E_2) \to E_2\ [Y \mapsto E]$$

3.2 Choreography

$$\mathsf{fst}(C_1, C_2) \to C_1 \qquad\qquad \mathsf{snd}(C_1, C_2) \to C_2$$

$$(\mathsf{match\ \ inl\ \ C\ \ with\ \ inl\ X} \to C_1;\ \mathsf{inr\ Y} \to C_2) \to C_1\ [X \mapsto C]$$

$$(\mathsf{match\ \ inr\ \ C\ \ with\ \ inl\ X} \to C_1;\ \mathsf{inr\ Y} \to C_2) \to C_2\ [Y \mapsto C]$$

4 Glossary

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

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

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

5 Endpoint Projection

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

$$[\![\textbf{fst}\ C]\!]_\ell = \begin{cases} \textbf{fst}\ [\![C]\!]_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C, \tau_1 \text{ and } \tau_2 \\ [\![C]\!]_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C \\ () & \text{otherwise} \end{cases}$$

$$[\![\textbf{snd} C]\!]_\ell = \begin{cases} \textbf{snd}\ [\![C]\!]_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C, \tau_1 \text{ and } \tau_2 \\ [\![C]\!]_\ell & \text{if } C : \tau_1 \times \tau_2 \text{ and } \ell \text{ is involved in } C \\ () & \text{otherwise} \end{cases}$$

$$[\![\textbf{inl}\ C]\!]_\ell = \begin{cases} \textbf{inl}\ [\![C]\!]_\ell & \text{if } C : \tau_1 \text{ and } \ell \text{ is involved in } C \text{ and } \tau_1 \\ [\![C]\!]_\ell & \text{if } C : \tau_1 \text{ and } \ell \text{ is involved in } C \text{ but not in } \tau_1 \\ () & \text{if } C : \tau_1 \text{ and } \ell \text{ is not involved in } C \text{ and } \tau_1 \end{cases}$$

$$[\![\textbf{inr}\ C]\!]_\ell = \begin{cases} \textbf{inr}\ [\![C]\!]_\ell & \text{if } C : \tau_2 \text{ and } \ell \text{ is involved in } C \text{ and } \tau_2 \\ [\![C]\!]_\ell & \text{if } C : \tau_2 \text{ and } \ell \text{ is involved in } C \text{ but not in } \tau_2 \\ () & \text{if } C : \tau_2 \text{ and } \ell \text{ is not involved in } C \text{ and } \tau_2 \end{cases}$$

$$[\![\textbf{match } C \textbf{ in inl X } \rightarrow C_1\textbf{; inr Y } \rightarrow C_2]\!]_\ell$$

$$= \begin{cases} \textbf{match } [\![C]\!]_\ell \textbf{ in inl X } \rightarrow [\![C_1]\!]_\ell\textbf{; inr Y } \rightarrow [\![C_2]\!]_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is involved in both } \tau_1 \text{ and } \tau_2 \\ [\![C]\!]_\ell\textbf{;}\ [\![C_1]\!]_\ell \sqcup [\![C_2]\!]_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or } C \\ [\![C_1]\!]_\ell \sqcup [\![C_2]\!]_\ell & \text{if } C : \tau_1 + \tau_2 \text{ and } \ell \text{ is not involved in } C,\ \tau_1 \text{ and } \tau_2 \end{cases}$$

6 Type Projection

$$[\![\textbf{unit}]\!]_\ell = \ \textbf{unit}$$

$$[\![\ell_1.t]\!]_{\ell_2} = \begin{cases} t & \text{if } \ell_1 = \ell_2 \\ \textbf{unit} & \text{otherwise} \end{cases}$$

$$[\![\tau_1 \rightarrow \tau_2]\!]_\ell = \begin{cases} [\![\tau_1]\!]_\ell \rightarrow [\![\tau_2]\!]_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or both} \\ \textbf{unit} & \text{otherwise} \end{cases}$$

$$[\![\tau_1 + \tau_2]\!]_\ell = \begin{cases} [\![\tau_1]\!]_\ell + [\![\tau_2]\!]_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ or } \tau_2 \text{ or both} \\ \textbf{unit} & \text{otherwise} \end{cases}$$

$$[\![\tau_1 \times \tau_2]\!]_\ell = \begin{cases} [\![\tau_1]\!]_\ell \times [\![\tau_2]\!]_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ and } \tau_2 \\ [\![\tau_1]\!]_\ell & \text{if } \ell \text{ is involved in } \tau_1 \text{ but not } \tau_2 \\ [\![\tau_2]\!]_\ell & \text{if } \ell \text{ is involved in } \tau_2 \text{ but not } \tau_1 \\ \textbf{unit} & \text{otherwise} \end{cases}$$

7 Lemmas

Lemma 1 : If ℓ is not involved in τ then $[\![\tau]\!]_\ell = \textbf{unit}$
Proof : By Induction on τ

Case $\tau = \textbf{unit}$:
From type projection we know, $[\![\textbf{unit}]\!]_\ell = \text{unit}$

Case $\tau = \ell.t$:
Here we know that, $\ell_1 \neq \ell$ as ℓ is not involved in τ
so using type projection for $[\![\ell_1.t]\!]_\ell$
we can say, $[\![\ell_1.t]\!]_\ell = \textbf{unit}$

Case $\tau = [\![\tau_1 \rightarrow \tau_2]\!]_\ell$:
By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2
so using type projection for $[\![\tau_1 \rightarrow \tau_2]\!]_\ell$
we can say, $[\![\tau_1 \rightarrow \tau_2]\!]_\ell = \textbf{unit}$

Case $\tau = [\![\tau_1 + \tau_2]\!]_\ell$:
By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2
so using type projection for $[\![\tau_1 + \tau_2]\!]_\ell$
we can say, $[\![\tau_1 + \tau_2]\!]_\ell = \textbf{unit}$

Case $\tau = [\![\tau_1 \times \tau_2]\!]_\ell$:
By IH, ℓ is not involved in τ_1 and ℓ is not involved in τ_2
so using type projection for $[\![\tau_1 \times \tau_2]\!]_\ell$
we can say, $[\![\tau_1 * \tau_2]\!]_\ell = \textbf{unit}$

Lemma 2 : If $\ell \notin locs(C)$ then $[\![C]\!]_\ell = ()$

Lemma 3 : If $\vdash C : \tau,$ then $\vdash [\![C]\!]_\ell : [\![\tau]\!]_\ell$

8 ALPS Syntax

Locations	ℓ	$\in$	$\mathcal{L}$
Synchronization Labels	d	$::=$	**All Variables**
Integers	i	$::=$	**All Integers**
Strings	s	$::=$	**All Strings**
Boolean	b	$::=$	$true \mid false$
Variables	x	$::=$	**All Variables**
Binary Operations	$binop$	$::=$	$+ \mid - \mid * \mid / \mid = \mid <= \mid >= \mid != \mid > \mid < \mid \&\& \mid \|\|$
Value	val	$::=$	$i \mid b \mid s$
Local Types	t	$::=$	**unit** $\mid$ **int** $\mid$ **string** $\mid$ **bool**
Local Expressions	e	$::=$	$() \mid val \mid x \mid e_1\ binop\ e_2 \mid$ **let** $x = e_1\ in\ e_2 \mid (e_1, e_2) \mid$ **fst** e
		$\mid$	**snd** $e \mid$ **left** $e \mid$ **right** $e \mid$ **match** e **with** $[\ \mid p \to e_1\ ]^*$
Comments	$comments$	$::=$	**- -** $\mid$ **{- -}**
Declarations	D	$::=$	$F : \tau_1 \to \tau_2 \mid X : \tau \mid \ell.x : \ell.t \mid$ **type** $name = \tau$
Assignment	A	$::=$	$X = C \mid F\ P_1 \ldots P_n = C \mid \ell.x = C$
Declaration Block	$decl\_block$	$::=$	$\cdot \mid D\ decl\_block \mid A\ decl\_block$
Local Patterns	p	$::=$	$\_ \mid val \mid x \mid (p_1, p_2) \mid left\ p \mid right\ p$
Patterns	P	$::=$	$\_ \mid x \mid (P_1, P_2) \mid \ell.p \mid left\ P \mid right\ P$
Choreographic Types	τ	$::=$	**unit** $\mid \ell.t \mid \tau_1 \to \tau_2 \mid \tau_1 \times \tau_2 \mid \tau_1 + \tau_2$
Choreography	C	$::=$	$() \mid X \mid \ell.e \mid \ell_1.e \rightsquigarrow \ell_2.x;\ C \mid C \rightsquigarrow \ell$
		$\mid$	**if** C_1 **then** C_2 **else** C_3
		$\mid$	$\ell_1[d] \rightsquigarrow \ell_2;\ C \mid$ **let** $decl\_block$ **in** C
		$\mid$	**fun** $X \to C \mid C_1\ C_2$
		$\mid$	$(C_1, C_2) \mid$ **fst** $C \mid$ **snd** $C \mid$ **left** $C \mid$ **right** C
		$\mid$	**match** C **with** $[\ \mid P \to C_1\ ]^*$
Network Types	t_N	$::=$	$t \mid t_{N1} \to t_{N2} \mid t_{N1} \times t_{N2} \mid t_{N1} + t_{N2}$
Network Expressions	E	$::=$	$X \mid () \mid$ **fun** $X \to E \mid E_1\ E_2 \mid \mathsf{ret}(e)$
		$\mid$	let ret$(x) = E_1$ in $E_2 \mid$ send e to $\ell;\ E \mid$ receive x from $\ell;\ E$
		$\mid$	if E_1 then E_2 else $E_3 \mid$ choose d for $\ell;\ E$
		$\mid$	allow ℓ choice $[\ \mid d \to E\ ]^* \mid (E_1, E_2)$
		$\mid$	**fst** $E \mid$ **snd** $E \mid$ **left** $E \mid$ **right** E
		$\mid$	**match** E **with** $[\ \mid p \to E_1\ ]^*$
Program	ρ	$::=$	$decl\_block$