Lambda Calculus + Binary Sessions (à la FuSe)

Hernán Melgratti

ICC University of Buenos Aires-Conicet

Formal FuSe

Syntax

Expression	e ::=	x	variable
	|	$\lambda x.e$	abstraction
	|	$e_1 e_2$	application
	|	$\texttt{let}\ x = e_1\ \texttt{in}\ e_2$	let
	|	$\texttt{c}$	constant
	|	a^p	endpoint
Process	P, Q ::=	$\langle e \rangle$	thread
	|	$P \parallel Q$	composition
	|	$(\nu a)P$	session
	|	$\texttt{error}$	run-time error

Constant

$\texttt{c}$ ::= () | pair | fst | snd | inl | inr | cases | fix
| fork | create | send | receive | branch | left | right

Polarity

p ::= $+ \mid - \mid \star$

$\star$ denotes *invalid endpoint* that should not be used. Moreover, $\overline{\star} = \star$.

Formal FuSe

Values

Values $v, w ::= \lambda x.e \mid \texttt{c} \mid \texttt{c}^1 v \mid \texttt{c}^2 v\ w \mid a^p$

$\texttt{c}^n$	n max	Sugared	Description
()	0		unit
pair	2	(v, w)	pair creation
fst, snd	0		pair projection
inl, inr	1		left/right injection
cases	2		sum deconstructor
fix	0		fixpoint combinator
fork	1		forking a process
create	0		session creation
close	0		terminate session
receive	0		input
send	1		output
branch	0		offer choice
left	0		choose left
right	0		choose right

Formal FuSe

Reduction of expressions (1)

[r − beta]	$(\lambda x.e)v \to e\{v/x\}$
[r − let]	$\texttt{let}\ x = v\ \texttt{in}\ e \to e\{v/x\}$
[r − fix]	$\texttt{fix}\ v \to v(\lambda x.\texttt{fix}\ v\ x)$
[r − fst]	$\texttt{fst}\ (\texttt{pair}\ v\ w) \to v$
[r − snd]	$\texttt{snd}\ (\texttt{pair}\ v\ w) \to w$
[r − inl]	$\texttt{cases}\ (\texttt{inl}\ v)\ v\ w \to v$
[r − inr]	$\texttt{cases}\ (\texttt{inr}\ v)\ v\ w \to w$

Formal FuSe

Evaluation Contexts

- Contextual rules are handled by defining **Evaluation Contexts**

$$\mathscr{E} ::= [\,] \mid \mathscr{E}\ e \mid v\ \mathscr{E} \mid \texttt{let}\ x = \mathscr{E}\ \texttt{in}\ e$$

- $\mathscr{E}[e]$ stands for the result of replacing the hole $[\,]$ in $\mathscr{E}$ with e

Semantics

Reduction of processes(1)

$$P \xrightarrow{\ell} Q$$

where

Labels ℓ	$::=$	τ	internal action
	$\mid$	ca	session a has been closed
	$\mid$	map	message exchange from a^p to $a^{\bar{p}}$

Enpoints associated with a label $\mathsf{ep}(\_)$

$$\begin{aligned} \mathsf{ep}(\tau) &= \{\} \\ \mathsf{ep}(\mathit{ca}) &= \{a^{+}, a^{-}\} \\ \mathsf{ep}(\mathit{map}) &= \{a^{+}, a^{-}\} \end{aligned}$$

Semantics

Reduction of processes(2)

$$\begin{array}{lll} [\mathsf{r}-\texttt{fork}] & \langle \mathscr{E}[\texttt{fork}\ v\ w] \rangle \xrightarrow{\tau} \langle \mathscr{E}[()] \rangle \parallel \langle v\ w \rangle & \\ [\mathsf{r}-\texttt{create}] & \langle \mathscr{E}[\texttt{create}\ ()] \rangle \xrightarrow{\tau} (\nu a)\langle \mathscr{E}[\texttt{pair}\ a^{+}\ a^{-}] \rangle & a \text{ fresh} \\ [\mathsf{r}-\texttt{close}] & \langle \mathscr{E}[\texttt{close}\ a^{p}] \rangle \parallel \langle \mathscr{E}'[\texttt{close}\ a^{\bar{p}}] \rangle \xrightarrow{\mathsf{ca}} \langle \mathscr{E}_{\mathsf{ca}}[()] \rangle \parallel \langle \mathscr{E}'_{\mathsf{ca}}[()] \rangle & \end{array}$$

$$\mathscr{E}_{\mathsf{ca}} = \mathscr{E}\{a^{\star}/a^{+}, a^{\star}/a^{-}\}$$

Semantics

Reduction of processes(3)

$$\begin{array}{ll} [\mathsf{r}-\texttt{comm}] & \langle \mathscr{E}[\texttt{send}\ a^{p}\ v] \rangle \parallel \langle \mathscr{E}'[\texttt{receive}\ a^{\bar{p}}] \rangle \xrightarrow{\mathsf{map}} \langle \mathscr{E}_{\mathsf{map}}[a^{p}] \rangle \parallel \langle \mathscr{E}'_{\mathsf{map}}[\texttt{pair}\ v_{\mathsf{map}}\ a^{\bar{p}}] \rangle \\ [\mathsf{r}-\texttt{left}] & \langle \mathscr{E}[\texttt{left}\ a^{p}] \rangle \parallel \langle \mathscr{E}'[\texttt{branch}\ a^{\bar{p}}] \rangle \xrightarrow{\mathsf{map}} \langle \mathscr{E}_{\mathsf{map}}[a^{p}] \rangle \parallel \langle \mathscr{E}'_{\mathsf{map}}[\texttt{inl}\ a^{\bar{p}}] \rangle \\ [\mathsf{r}-\texttt{rigth}] & \langle \mathscr{E}[\texttt{right}\ a^{p}] \rangle \parallel \langle \mathscr{E}'[\texttt{branch}\ a^{\bar{p}}] \rangle \xrightarrow{\mathsf{map}} \langle \mathscr{E}_{\mathsf{map}}[a^{p}] \rangle \parallel \langle \mathscr{E}'_{\mathsf{map}}[\texttt{inr}\ a^{\bar{p}}] \rangle \end{array}$$

$$\begin{aligned} \mathscr{E}_{\mathsf{map}} &= \mathscr{E}\{a^{\star}/a^{+}, a^{\star}/a^{-}\} \\ v_{\mathsf{map}} &= v\{a^{\star}/a^{+}, a^{\star}/a^{-}\} \end{aligned}$$

Semantics

Reduction of processes(4)

$$[\mathtt{r-thread}]\quad \frac{e \xrightarrow{\tau} e'}{\langle \mathscr{E}[e] \rangle \xrightarrow{\tau} \langle \mathscr{E}[e'] \rangle}$$

$$[\mathtt{r-par}]\quad \frac{P \xrightarrow{\ell} Q}{P \parallel R \xrightarrow{\ell} Q \parallel R}$$

$$[\mathtt{r-new}]\quad \frac{P \xrightarrow{\ell} Q}{(\nu a)P \xrightarrow{\ell \backslash a} (\nu a)Q}$$

$$[\mathtt{r-struct}]\quad \frac{P \equiv P' \qquad P' \xrightarrow{\ell} Q' \qquad Q' \equiv Q}{P \xrightarrow{\ell} Q}$$

$$\ell \backslash a = \begin{cases} \tau & \text{if } \ell = \mathsf{map} \text{ or } \ell = \mathsf{c}a \\ \ell & \text{otherwise} \end{cases}$$

$\equiv$ is an equivalence relation such that:

- $\parallel$ is associative and commutative, with $\langle () \rangle$ as the identity
- $(\nu a)\langle () \rangle = \langle () \rangle$
- $(\nu a)P \parallel Q = (\nu a)(P \parallel Q)$ if $a \notin \mathrm{fn}(Q)$

Semantics

Reduction of processes(5)

$$[\mathtt{r-error}]\ \langle \mathscr{E}[K\ a^\star] \rangle \xrightarrow{\tau} \mathtt{error}$$

$$K \ ::= \ \mathtt{close} \mid \mathtt{send}\ v \mid \mathtt{receive} \mid \mathtt{left} \mid \mathtt{right} \mid \mathtt{branch}$$

Typing

Types

Type Schemes	σ	$::=$	$t \mid \forall \alpha.\sigma \mid \forall A.\sigma$
Types	t, s	$::=$	$\alpha \mid \mathtt{unit} \mid t \to s \mid t + s \mid t \times s \mid T$
Session Types	T, S	$::=$	$A \mid \overline{A} \mid \mathtt{end} \mid !t.T \mid ?t.T \mid T \oplus S \mid T \& S$

Equations are interpreted *coinductively*: Infinite terms instead of concrete syntax for infinite types

Instantiation of schemes

$$t \succ t \qquad\qquad \frac{\sigma \succ t}{\forall \alpha.\sigma \succ t\{s/\alpha\}}$$

Typing

Type schemes of constants

$$\begin{aligned}
() &: \mathtt{unit} \\
\mathtt{pair} &: \forall \alpha, \beta.\alpha \to \beta \to \alpha \times \beta \\
\mathtt{fst} &: \forall \alpha, \beta.\alpha \times \beta \to \alpha \\
\mathtt{snd} &: \forall \alpha, \beta.\alpha \times \beta \to \beta \\
\mathtt{inl} &: \forall \alpha, \beta.\alpha \to \alpha + \beta \\
\mathtt{inr} &: \forall \alpha, \beta.\beta \to \alpha + \beta \\
\mathtt{cases} &: \forall \alpha, \beta, \gamma.(\alpha + \beta) \to (\alpha \to \gamma) \to (\beta \to \gamma) \to \gamma \\
\mathtt{fix} &: \forall \alpha, \beta.((\alpha \to \beta) \to \alpha \to \beta) \to \alpha \to \beta \\
\mathtt{fork} &: \forall \alpha.((\alpha \to ()) \to \alpha \to \mathtt{unit} \\
\mathtt{create} &: \forall A.() \to A \times \overline{A} \\
\mathtt{close} &: \mathtt{end} \to \mathtt{unit} \\
\mathtt{send} &: \forall \alpha, A.\alpha \to !\alpha.A \to A \\
\mathtt{receive} &: \forall \alpha, A.?\alpha.A \to \alpha \times A \\
\mathtt{left} &: \forall A, B.A \oplus B \to A \\
\mathtt{right} &: \forall A, B.A \oplus B \to B \\
\mathtt{branch} &: \forall A, B.A \& B \to A + B
\end{aligned}$$

Typing

Typing

Typing rules for expressions $\boxed{\Gamma \vdash e : t}$

$$\frac{\mathsf{typeof}(\mathtt{c}) \succ t}{\Gamma \vdash \mathtt{c} : t}\text{[t-const]} \qquad \frac{\sigma \succ t}{\Gamma, u : \sigma \vdash u : t}\text{[t-name]}$$

$$\frac{\Gamma, x : t \vdash e : s}{\Gamma \vdash \lambda x.e : t \to^{\iota} s}\text{[t-fun]} \qquad \frac{\Gamma \vdash e_1 : t \to^{\iota} s \qquad \Gamma \vdash e_2 : t}{\Gamma \vdash e_1 e_2 : s}\text{[t-app]}$$

$$\frac{\Gamma \vdash e_1 : t_1 \qquad \Gamma, x : \mathsf{Close}(t_1, \Gamma) \vdash e_2 : t}{\Gamma \vdash \mathtt{let}\ x = e_1\ \mathtt{in}\ e_2 : t}\text{[t-let]}$$

$\mathsf{Close}(t, \Gamma) = \forall \alpha_1, \dots, \alpha_n$ where $\{\alpha_1, \dots, \alpha_n\} = \mathit{fv}(t) \setminus \mathit{fv}(\Gamma)$

Typing

Typing

Typing rules for processes $\boxed{\Gamma \vdash P}$

$$\frac{\Gamma \vdash e : \mathtt{unit}}{\Gamma \vdash \langle e \rangle}\text{[t-thread]} \qquad \frac{\Gamma \vdash P_1 \qquad \Gamma \vdash P_2}{\Gamma \vdash P_1 \parallel P_2}\text{[t-par]}$$

$$\frac{\Gamma, a^{+} : T, a^{-} : \overline{T}, a^{\star} : \forall A.A \vdash P}{\Gamma \vdash (\nu a)P}\text{[t-session]}$$

Properties

Endpoint affine/linear process

Let $\mathscr{A}$, $\mathscr{L}$ be the largest predicates such that:

- if either $\mathscr{A}(P)$ or $\mathscr{L}(P)$ and $P \equiv (\nu a_1 \dots a_n)(\mathscr{E}[K\ a^p] \parallel Q)$, then $p \in \{+, -\}$;
- if $\mathscr{L}(P)$ and $P \equiv (\nu a)Q$ and $a^p \in \mathsf{fv}(Q)$ and $p \in \{+, -\}$, then $a^{\overline{p}} \in \mathsf{fv}(Q)$
- if $\mathscr{L}(P)$ and $P \xrightarrow{\ell} Q$, then $\mathscr{L}(Q)$;
- if $\mathscr{A}(P)$ and $P \xrightarrow{\ell} Q$, then $\mathscr{A}(Q)$;

Properties

Balanced type environment

Γ is *balanced* if:

- for every $a^p \in \mathsf{dom}(\Gamma)$ with $p \in \{+, -\}$ we have $a^{\overline{p}}, a^{\star} \in \mathsf{dom}(\Gamma)$ and $\Gamma(a^p) \perp \Gamma(a^{\overline{p}})$
- for every $a^{\star} \in \mathsf{dom}(\Gamma)$ we have $\Gamma(a^p) = \forall A.A$

Properties

Subject reduction

- If $\Gamma \vdash P$ with Γ balanced and $\mathscr{A}(P)$ and $P \xrightarrow{\ell} Q$, then there exists Γ' such that $\Gamma \xrightarrow{\ell} \Gamma'$ and $\Gamma' \vdash Q$.

Semantics of types

$$\begin{array}{rcl}
\Gamma & \xrightarrow{\ell} & \Gamma \\
\Gamma, a^p : \mathbf{!}t\,\mathbf{.}\,T, a^{\overline{p}} : \mathbf{?}t\,\mathbf{.}\,\overline{T} & \xrightarrow{\text{map}} & \Gamma, a^p : T, a^{\overline{p}} : \overline{T} \\
\Gamma, a^p : T \oplus S, a^{\overline{p}} : \overline{T} \,\&\, \overline{S} & \xrightarrow{\text{map}} & \Gamma, a^p : T, a^{\overline{p}} : \overline{T} \\
\Gamma, a^p : T \oplus S, a^{\overline{p}} : \overline{T} \,\&\, \overline{S} & \xrightarrow{\text{map}} & \Gamma, a^p : S, a^{\overline{p}} : \overline{S} \\
\Gamma, a^p : \text{end}, a^{\overline{p}} : \text{end} & \xrightarrow{\text{ca}} & \Gamma
\end{array}$$

Properties

Type safety

Let $\Gamma \vdash P$ and $\mathscr{A}$. Then,

- If $P \equiv (\nu a_1 \dots a_n)(\mathscr{E}[\texttt{send}\ v\ w] \parallel Q)$, then there exists Γ' such that $\Gamma, \Gamma' \vdash v : t$ and $\Gamma, \Gamma' \vdash w : \mathbf{!}t\,\mathbf{.}\,T$.
- If $P \equiv (\nu a_1 \dots a_n)(\mathscr{E}[\texttt{left}\ v\ ] \parallel Q)$ or $P \equiv (\nu a_1 \dots a_n)(\mathscr{E}[\texttt{right}\ v\ ] \parallel Q)$, then there exists Γ' such that $\Gamma, \Gamma' \vdash v : S \oplus T$