On the Relative Expressiveness of Higher-Order Session Processes[1]

Dimitrios Kouzapas Jorge A. Pérez Nobuko Yoshida

University of Glasgow, UK

University of Groningen, The Netherlands

Imperial College London, UK

To appear in ESOP'16.

[1] This work has been sponsored by EPSRC EP/K011715/1, EPSRC EP/K034413/1, EPSRC EP/L00058X/1, EU project FP7-612985, and EU COST Action IC1201 BETTY.

Expressiveness of Higher-Order Sessions

- The HOπ, higher order session typed π-calculus
- Combines the π-calculus and the λ-calculus
- Bridge the gap between processes and functions
- Expressiveness in the presence of session types
- The encodings are *forced* to follow session typed properties

The λ and the π

λ			π		
$\lambda x.\, N\, m$	$\longrightarrow$	$N\{^m/_x\}$	$n?(x).P \mid n!\langle m\rangle.\mathbf{0}$	$\longrightarrow$	$P\{^m/_x\}$
$\lambda x.\, N\, M$	$\longrightarrow$	$N\{^M/_x\}$	?		
?			$n?(x).P \mid n!\langle m\rangle.\mathbf{0} \mid Q$	$\longrightarrow$	$P\{^m/_x\} \mid Q$

The π-calculus - Syntax

$$
\begin{array}{lcl}
n & ::= & a, b \mid s, \overline{s} \qquad\qquad u, w \;::=\; n \mid x, y, z \\
V, W & ::= & n \mid x \\
P, Q & ::= & u?(x).P \mid u!\langle V\rangle.P \mid u \triangleright \{l_i : P_i\}_{i \in I} \\
 & \mid & u \triangleleft l.P \mid P \mid Q \mid (\nu\, n)P \mid \mathbf{0} \\
 & \mid & X \mid \mu X.P
\end{array}
$$

The HO-calculus - Syntax

$$
\begin{array}{lcl}
n & ::= & a, b \mid s, \overline{s} \qquad\qquad u, w \;::=\; n \mid x, y, z \\
V, W & ::= & x \mid \lambda x.\, P \\
P, Q & ::= & u?(x).P \mid u!\langle V \rangle.P \mid u \triangleright \{l_i : P_i\}_{i \in I} \\
 & \mid & u \triangleleft l.P \mid P \mid Q \mid (\nu\, n)P \mid \mathbf{0} \\
 & \mid & V\, u
\end{array}
$$

The HOπ-calculus - Syntax

$$
\begin{array}{lcl}
n & ::= & a, b \mid s, \overline{s} \qquad\qquad u, w \;::=\; n \mid x, y, z \\
V, W & ::= & n \mid x \mid \lambda x.\, P \\
P, Q & ::= & u?(x).P \mid u!\langle V \rangle.P \mid u \triangleright \{l_i : P_i\}_{i \in I} \\
 & \mid & u \triangleleft l.P \mid P \mid Q \mid (\nu\, n)P \mid \mathbf{0} \\
 & \mid & X \mid \mu X.P \mid V\, u
\end{array}
$$

The HOπ typing system

$$
\begin{array}{lcl}
C & ::= & S \mid \langle S \rangle \mid \langle L \rangle \\
U & ::= & C \mid L \\
S & ::= & !\langle U \rangle; S \mid ?(U); S \mid \oplus\{l_i : S_i\}_{i \in I} \mid \&\{l_i : S_i\}_{i \in I} \\
 & \mid & \mu \mathtt{t}.S \mid \mathtt{t} \mid \mathtt{end}
\end{array}
\qquad
L ::= C \rightarrow \diamond \mid C \multimap \diamond
$$

Γ : Shared environment

Λ, Δ : Linear environments

Typing Judgements:

$$\Gamma; \Lambda; \Delta \vdash P \triangleright \diamond \qquad \Gamma; \Lambda; \Delta \vdash V \triangleright U$$

The HOπ labelled transitions semantics

Definition (Labelled Transition Semantics)

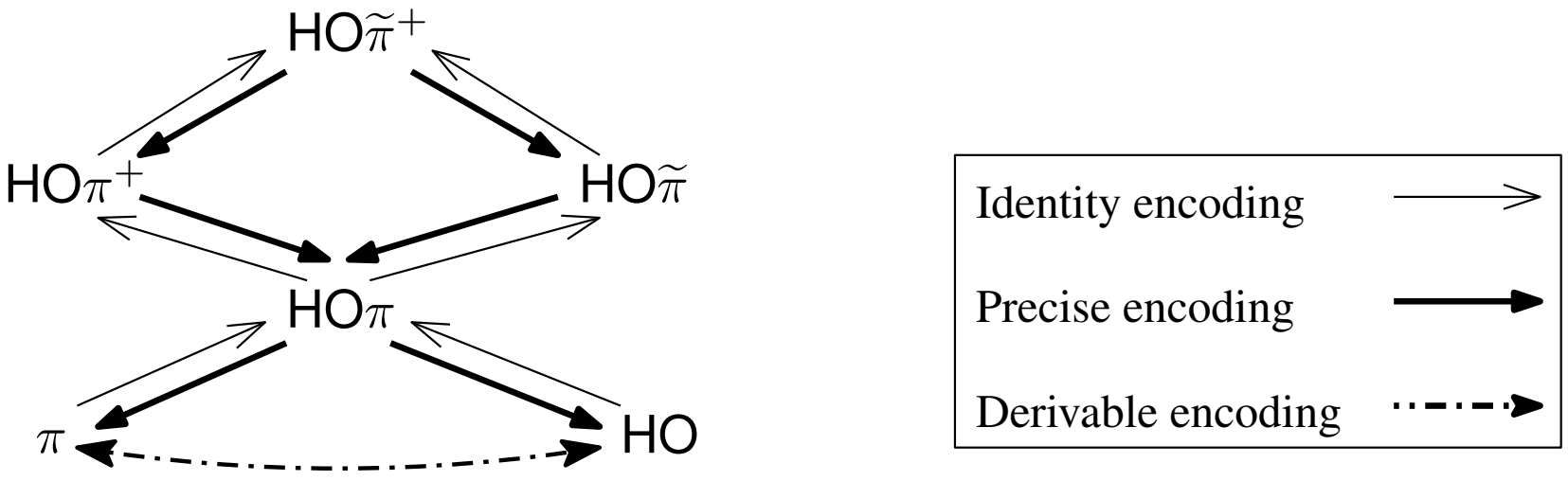
$$[\textsc{App}]\ \frac{}{\lambda x.\, P\, n \xrightarrow{\tau} P\{n/x\}} \qquad [\textsc{Out}]\ \frac{}{n!\langle V\rangle.P \xrightarrow{n!\langle V\rangle} P} \qquad \cdots$$

$$[\textsc{Tau}]\ \frac{P \xrightarrow{\ell_1} P' \qquad Q \xrightarrow{\ell_2} Q' \qquad \ell_1 \asymp \ell_2}{P \mid Q \xrightarrow{\tau} (\nu\, \mathtt{bn}(\ell_1) \cup \mathtt{bn}(\ell_2))(P' \mid Q')}$$

Definition (Typed Labelled Transition Semantics)

If $P \xrightarrow{\ell} P'$ and $(\Gamma, \Delta) \xrightarrow{\ell} (\Gamma, \Delta')$ then $\Gamma; \emptyset; \Delta \vdash P \xrightarrow{\ell} \Delta' \vdash P'$.

Expressiveness in the HOπ



Encodability in Higher-Order Sessions. (Precise encoding is defined later)

Why typed expressiveness?

Let (untyped) mapping $[\cdot] : \mathrm{HO}\pi \to \mathrm{HO}$ such that:

$$[s!\langle n\rangle.P] \stackrel{\mathtt{def}}{=} s?(x).([P] \mid x\, n)$$

$$[s?(n).Q] \stackrel{\mathtt{def}}{=} s!\langle \lambda x.\, [Q]\rangle.\mathbf{0}$$

Semantic correspondence:

$$s!\langle n\rangle.P \mid s?(n).Q \longrightarrow P \mid Q\{^{n}/_{x}\}$$

$$\begin{aligned}
[s!\langle n\rangle.P \mid s?(n).Q] &\stackrel{\mathtt{def}}{=} s?(x).([P] \mid x\, n) \mid s!\langle \lambda x.\, [Q]\rangle.\mathbf{0} \\
&\longrightarrow [P] \mid \lambda x.\, [Q]\, n \\
&\longrightarrow [P] \mid [Q]\{^{n}/_{x}\}
\end{aligned}$$

Why typed expressiveness?

Let (untyped) mapping $[\cdot] : \mathrm{HO}\pi \to \mathrm{HO}$ such that:

$$[s!\langle n\rangle.P] \stackrel{\mathtt{def}}{=} s?(x).([P] \mid x\ n)$$

$$[s?(n).Q] \stackrel{\mathtt{def}}{=} s!\langle \lambda x.\,[Q]\rangle.\mathbf{0}$$

The session type behaviour is reversed:

$$\Gamma_1; \emptyset; \Delta_1 \cdot s :!\langle U_1\rangle; S_1 \ \vdash\ P \ \triangleright \diamond$$

$$\Gamma_2; \emptyset; \Delta_2 \cdot s :?(U_2); S_2 \ \vdash\ [P] \ \triangleright \diamond$$

Need encoding criteria for <u>behaviour preservation</u>.

Typed Encoding

Definition (Typed Calculus)

A typed calculus $\mathcal{L}$ is a tuple $\langle \mathsf{C}, \mathcal{T}, \longrightarrow, \approx, \vdash \rangle$.

Definition (Typed Encoding)

Let:

- $\mathcal{L}_i = \langle \mathsf{C}_i, \mathcal{T}_i, \longrightarrow_i, \approx_i, \vdash_i \rangle \quad i \in 1, 2.$
- $[\![\cdot]\!] : \mathsf{C}_1 \to \mathsf{C}_2$
- $\langle\!\langle \cdot \rangle\!\rangle : \mathcal{T}_1 \to \mathcal{T}_2$

We write:

$$\langle [\![\cdot]\!], \langle\!\langle \cdot \rangle\!\rangle \rangle : \mathcal{L}_1 \to \mathcal{L}_2$$

for the type encoding of $\mathcal{L}_1$ into $\mathcal{L}_2$.

Encoding Criteria - Syntax Preservation

Definition (Syntax Preservation)

Typed encoding $\langle [\cdot], (\!|\cdot|\!) \rangle : \mathcal{L}_1 \to \mathcal{L}_2$ is *syntax preserving* whenever:

1. *Homomorphic wrt parallel*, if $(\!|\Gamma|\!); \emptyset; (\!|\Delta_1 \cdot \Delta_2|\!) \vdash_2 [P_1 \mid P_2] \triangleright \diamond$ then $(\!|\Gamma|\!); \emptyset; (\!|\Delta_1|\!) \cdot (\!|\Delta_2|\!) \vdash_2 [P_1] \mid [P_2] \triangleright \diamond$.

2. *Compositional wrt restriction*, if $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!) \vdash_2 [(\nu\, n)P] \triangleright \diamond$ then $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!) \vdash_2 (\nu\, n)[P] \triangleright \diamond$.

3. *Name invariant*, if $(\!|\sigma(\Gamma)|\!); \emptyset; (\!|\sigma(\Delta)|\!) \vdash_2 [\sigma(P)] \triangleright \diamond$ then $\sigma((\!|\Gamma|\!)); \emptyset; \sigma((\!|\Delta|\!)) \vdash_2 \sigma([P]) \triangleright \diamond$, for any injective renaming of names σ.

Encoding Criteria - Type Preservation

Definition (Type Preservation)

The typed encoding $\langle [\cdot], (\!|\cdot|\!) \rangle : \mathcal{L}_1 \to \mathcal{L}_2$ is typed preserving if for every k-ary type op operator in $\mathcal{T}_1$ it holds that:

$$(\!|\mathrm{op}(T_1, \ldots, T_k)|\!) = \mathrm{op}((\!|T_1|\!), \ldots, (\!|T_k|\!))$$

Example

Let mapping $(\!|\cdot|\!) : \mathcal{T}_1 \to \mathcal{T}_2$ that includes definition:

$$(\!|!\langle U \rangle; T|\!) \stackrel{\mathtt{def}}{=} ?((\!|U|\!)); (\!|T|\!)$$

Then $(\!|\cdot|\!)$ is not typed preserving.

Encoding Criteria - Semantic Preservation

Definition (Semantic Preservation)

Typed Encoding $\langle [\cdot], (\!|\cdot|\!) \rangle : \mathcal{L}_1 \to \mathcal{L}_2$ is *semantic preserving* if:

- *Type Soundness*,
  if $\Gamma; \emptyset; \Delta \vdash_1 P \triangleright \diamond$ then $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!) \vdash_2 [P] \triangleright \diamond$
- *Barb Preserving*,
  if $\Gamma, \emptyset; \Delta \vdash_1 P \downarrow_n$ then $(\!|\Gamma|\!), \emptyset; (\!|\Delta|\!) \vdash_1 [P] \Downarrow_n$
- *Operational Correspondence*, if $\Gamma; \emptyset; \Delta \vdash_1 P \triangleright \diamond$ then
  - *Complete*, if $\Gamma; \emptyset; \Delta \vdash_1 P \longrightarrow \Gamma; \emptyset; \Delta' \vdash_1 P'$ then $\exists Q, \Delta''$ such that $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!)[P] \longrightarrow^* (\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q$ and $(\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q \approx_2 (\!|\Delta'|\!) \vdash_2 [P']$.
  - *Sound*, if $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!)[P] \longrightarrow^* (\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q$ then $\exists \Delta', P'$ such that $\Gamma; \emptyset; \Delta \vdash_1 P \longrightarrow \Gamma; \emptyset; \Delta' \vdash_1 P'$ and $(\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q \approx_2 (\!|\Delta'|\!) \vdash_2 [P']$
- *Full Abstraction*,
  $\Gamma; \emptyset; \Delta \vdash_1 P \approx_1 \Delta' \vdash_1 Q$ iff $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!) \vdash_2 [P] \approx_2 (\!|\Delta'|\!) \vdash_2 [Q]$

Encoding Criteria: Precise Encoding

Definition (Precise Encoding)

A typed encoding is called *precise* if it is syntax preserving, type preserving, and semantic preserving.

Definition (Minimal Encoding)

A typed encoding is called *minimal* if it is parallel homomorphic, barb preserving, and operationally complete.

Encoding: HO$\pi \to$ HO

Let $([\cdot], (\!|\cdot|\!)) : \mathrm{HO}\pi \to \mathrm{HO}$ such that:

$$
\begin{array}{rcl}
[u!\langle w\rangle.P]_f & \stackrel{\mathtt{def}}{=} & u!\langle \lambda z.\,(z?(x).x\,w)\rangle.[P] \\
[u?(x).P]_f & \stackrel{\mathtt{def}}{=} & u?(y).(\nu\, s)(y\, s \mid \overline{s}!\langle \lambda x.\,[P]\rangle.\mathbf{0}) \\
(\!|!\langle S\rangle; S'|\!) & \stackrel{\mathtt{def}}{=} & !\langle ?((\!|S|\!) \multimap \diamond); \mathtt{end}\rangle; (\!|S'|\!) \\
(\!|?(S); S'|\!) & \stackrel{\mathtt{def}}{=} & ?(?((\!|S|\!) \multimap \diamond); \mathtt{end}); (\!|S'|\!)
\end{array}
$$

Pack a name w into abstraction $\lambda z.\,(z?(x).x\,w)$

Unpack and substitute using process passing and application.

Typing mapping lifts the type of the name by an order.

Encoding: HO$\pi \to$ HO

Let $([\cdot], (\!|\cdot|\!)) : \mathrm{HO}\pi \to \mathrm{HO}$ such that:

$$
\begin{array}{rcl}
[\mu X.P]_f & \stackrel{\mathtt{def}}{=} & (\nu\, s)(\overline{s}!\langle \lambda(\|\tilde{n}\|, y).\ y?(z_X).\left\|\!\left[P\right]_{f,\{X\to\tilde{n}\}}\right\|_{\emptyset}\rangle.\mathbf{0} \mid \\
 & & s?(z_X).[P]_{f,\{X\to\tilde{n}\}}) \quad (\tilde{n} = \mathtt{fn}(P)) \\
[X]_f & \stackrel{\mathtt{def}}{=} & (\nu\, s)(z_X\,(\tilde{n}, s) \mid \\
 & & \overline{s}!\big\langle \lambda(\|\tilde{n}\|, y).\, z_X\,(\|\tilde{n}\|, y)\big\rangle.\mathbf{0}) \quad (\tilde{n} = f(X)) \\
(\!|\Gamma \cdot X : \Delta|\!) & \stackrel{\mathtt{def}}{=} & (\!|\Gamma|\!) \cdot z_X : (\tilde{S}, \mu \mathrm{t}.?(\tilde{S}, \mathrm{t} \multimap \diamond); \mathtt{end}) \to \diamond
\end{array}
$$

Recursion is encoded into a higher order session typed replicator.

Type encoding reveals a non-tail recursive type.

Encoding: HO$\pi \to \pi$

Let $([\cdot], (\!|\cdot|\!)) : \mathrm{HO}\pi \to \pi$ such that

$$
\begin{array}{rcll}
[n!\langle \lambda x.\, Q\rangle.P] & \stackrel{\text{def}}{=} & (\nu\, a)(n!\langle a\rangle.[P] \mid *\ a?(y).y?(x).[Q]) & s \notin \mathtt{fn}(Q) \\
[n!\langle \lambda x.\, Q\rangle.P] & \stackrel{\text{def}}{=} & (\nu\, a)(n!\langle a\rangle.[P] \mid a?(y).y?(x).[Q]) & s \in \mathtt{fn}(Q) \\
[x\ u] & \stackrel{\text{def}}{=} & (\nu\, s)(x!\langle s\rangle.\overline{s}!\langle u\rangle.\mathbf{0}) & \\
[\lambda x.\, P\ u] & \stackrel{\text{def}}{=} & (\nu\, s)(s?(x).[P] \mid \overline{s}!\langle u\rangle.\mathbf{0}) & \\
(\!|!\langle S_1 \multimap \diamond\rangle; S_2|\!) & \stackrel{\text{def}}{=} & !\langle ?((\!|S_1|\!)); \mathtt{end}\rangle; (\!|S_2|\!) & \\
(\!|?(S_1 \multimap \diamond); S_2|\!) & \stackrel{\text{def}}{=} & ?(?((\!|S_1|\!)); \mathtt{end}); (\!|S_2|\!) &
\end{array}
$$

Negative Result: $\mathrm{HO}\pi \not\rightarrow \mathrm{HO}\pi^{-\mathrm{sh}}$

Definition

Let $\mathrm{HO}\pi^{-\mathrm{sh}}$ be the $\mathrm{HO}\pi$ without shared names.

Lemma (Determinacy)

Let P a $\mathrm{HO}\pi^{-\mathrm{sh}}$ process. If $P \longrightarrow^* P'$ then $P \approx P'$ (only session transitions).

Theorem

There exists no typed, minimal encoding $\langle [\![\cdot]\!], (\!|\cdot|\!) \rangle : \mathrm{HO}\pi \rightarrow \mathrm{HO}\pi^{-\mathrm{sh}}$.

Proof (Sketch).

Let $\langle [\![\cdot]\!], (\!|\cdot|\!) \rangle : \mathrm{HO}\pi \rightarrow \mathrm{HO}\pi^{-\mathrm{sh}}$ and $P = a!\langle s \rangle.\mathbf{0} \mid a?(x).n!\langle l_1 \rangle.\mathbf{0} \mid a?(x).\mathbf{0}$.

$$P \longrightarrow S_1 \downarrow_n \text{ implies } [\![S_1]\!] \Downarrow_n \qquad \textit{(barb preservation)} \tag{1}$$

$$P \longrightarrow S_2 \not\Downarrow_n \tag{2}$$

From *operational completeness* $[\![P]\!] \longrightarrow^* [\![S_1]\!]$ and $[\![P]\!] \longrightarrow^* [\![S_2]\!]$, so

$$[\![S_1]\!] \approx [\![S_2]\!] \qquad \textit{(determinacy)} \tag{3}$$

(1) and (3) implies $[\![S_2]\!] \Downarrow_n$, which contradicts with (2). $\square$

Extensions - The HOπ^+-calculus

$$
\begin{array}{rcl}
n & ::= & a, b \;\mid\; s, \overline{s} \qquad\qquad u, w \;::=\; n \;\mid\; x, y, z \\
V, W & ::= & u \;\mid\; \lambda x.\, P \\
P, Q & ::= & u?(x).P \;\mid\; u!\langle V\rangle.P \;\mid\; u \triangleright \{l_i : P_i\}_{i\in I} \\
 & \mid & u \triangleleft l.P \;\mid\; P \mid Q \;\mid\; (\nu\, n)P \;\mid\; \mathbf{0} \\
 & \mid & X \;\mid\; \mu X.P \;\mid\; \boxed{\mathrm{V}\ \mathrm{W}}
\end{array}
$$

$$
\begin{array}{rcl}
[\lambda x.\, P\, \lambda y.\, Q] & \stackrel{\text{def}}{=} & (\nu\, s)(s?(x).[P] \mid \overline{s}!\langle \lambda y.\, [Q]\rangle.\mathbf{0}) \\
(\!| L \to \diamond |\!) & \stackrel{\text{def}}{=} & (?(L); \mathtt{end}) \to \diamond
\end{array}
$$

Extensions - The HO$\tilde{\pi}$-calculus

$$
\begin{array}{rcl}
n & ::= & a, b \;\mid\; s, \overline{s} \qquad\qquad u, w \;::=\; n \;\mid\; x, y, z \\
V, W & ::= & \boxed{\tilde{u} \;\mid\; \lambda\tilde{x}.\, P} \\
P, Q & ::= & u?(x).P \;\mid\; u!\langle V\rangle.P \;\mid\; u \triangleright \{l_i : P_i\}_{i\in I} \\
 & \mid & u \triangleleft l.P \;\mid\; P \mid Q \;\mid\; (\nu\, n)P \;\mid\; \mathbf{0} \\
 & \mid & X \;\mid\; \mu X.P \;\mid\; V\, u
\end{array}
$$

$$
\begin{array}{rcl}
[\lambda\tilde{x}.\, P\, \tilde{u}] & \stackrel{\text{def}}{=} & (\nu\, s)(s?(\tilde{x}).[P] \mid \overline{s}?(\tilde{u}).\mathbf{0}) \\
\langle\!\langle \tilde{C} \rightarrow \diamond \rangle\!\rangle & \stackrel{\text{def}}{=} & (?(\tilde{C}); \mathtt{end}) \rightarrow \diamond \\
[s!\langle \tilde{n}\rangle.P] & \stackrel{\text{def}}{=} & s!\langle n_1\rangle.s!\langle n_2\rangle. \ldots [P] \\
\langle\!\langle !\langle \tilde{U}\rangle; S \rangle\!\rangle & \stackrel{\text{def}}{=} & !\langle\!\langle U_1 \rangle\!\rangle\rangle; !\langle\!\langle U_2 \rangle\!\rangle\rangle; \ldots \langle\!\langle S \rangle\!\rangle
\end{array}
$$

Extensions - The HO $\widetilde{\pi}^{+}$-calculus

$$
\begin{array}{rcl}
n & ::= & a, b \;\mid\; s, \overline{s} \qquad\qquad u, w \;::=\; n \;\mid\; x, y, z \\
V, W & ::= & \boxed{\tilde{u} \;\mid\; \lambda \tilde{x}.\, P} \\
P, Q & ::= & u?(x).P \;\mid\; u!\langle V \rangle.P \;\mid\; u \triangleright \{l_i : P_i\}_{i \in I} \\
 & \mid & u \triangleleft l.P \;\mid\; P \mid Q \;\mid\; (\nu\, n)P \;\mid\; \mathbf{0} \\
 & \mid & X \;\mid\; \mu X.P \;\mid\; \boxed{\mathrm{V}\ \mathrm{W}}
\end{array}
$$

Comparing HO and π

Definition (Labelled Correspondence)

Assume $\mathcal{L}_i = \langle C_i, \mathcal{T}_i, \xrightarrow{\ell_i}, \approx_i, \vdash_i \rangle$, encoding $\langle [\![\cdot]\!], (\!|\cdot|\!) \rangle : \mathcal{L}_1 \to \mathcal{L}_2$, and $\{\!\{\cdot\}\!\} : \mathcal{A}_1 \to \mathcal{A}_2$ with $\ell_i \in \mathcal{A}_i$. If $\Gamma; \emptyset; \Delta \vdash_1 P \triangleright \diamond$ then

- *Complete*, if $\Gamma; \emptyset; \Delta \vdash_1 P \xrightarrow{\ell} \Gamma; \emptyset; \Delta' \vdash_1 P'$ then $\exists Q, \Delta''$ such that $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!)[P] \overset{\{\!\{\ell\}\!\}}{\Longrightarrow} (\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q$ and $(\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q \approx_2 (\!|\Delta'|\!) \vdash_2 [P']$.
- *Sound*, if $(\!|\Gamma|\!); \emptyset; (\!|\Delta|\!)[P] \overset{\{\!\{\ell\}\!\}}{\Longrightarrow} (\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q$ then $\exists \Delta', P'$ such that $\Gamma; \emptyset; \Delta \vdash_1 P \xrightarrow{\ell} \Gamma; \emptyset; \Delta' \vdash_1 P'$ and $(\!|\Gamma|\!); \emptyset; (\!|\Delta''|\!) \vdash_2 Q \approx_2 (\!|\Delta'|\!) \vdash_2 [P']$

Definition (Tight Encoding)

An encoding $\langle [\![\cdot]\!], (\!|\cdot|\!) \rangle : \mathcal{L}_1 \to \mathcal{L}_2$ is *tight* if it is precise and for some $\{\!\{\cdot\}\!\} : \mathcal{A}_1 \to \mathcal{A}_2$ it satisfies labelled correspondence.

Comparing HO and π

- Encoding $\langle [\cdot]_f, (\!|\cdot|\!) \rangle : \mathrm{HO}\pi \to \mathrm{HO}$ is tight
- Encoding $\langle [\cdot], (\!|\cdot|\!) \rangle : \mathrm{HO}\pi \to \pi$ is not tight
- Encoding $\langle [\cdot], (\!|\cdot|\!) \rangle : \mathrm{HO}\pi \to \pi$ often creates the garbage process $*\, a?(y).y?(x).[Q]$
- The HO-calculus *includes* (most of) the λ-calculus
- The π-calculus needs to encode the λ-calculus.
- But sending names is more *economical* than sending processes

Summary

- Expressiveness results for higher-order process calculi in the presence of (session) types
- New encoding criteria: behaviour preservation
- Precise and tight encodings: syntactic, typing, semantic preservation
- The HO is a minimal higher-order process calculus with higher order message passing and name abstraction/application that can *tightly* express name passing and recursion
- Minimal Encoding: negative results - cannot encode shared names into session names