What are Polymorphically-typed Ambients ?

Torben Amtoft, Assaf J. Kfoury, Santiago Pericas-G.

Boston University

October 6, 2000

Polymorphism ?

Case 1. (ML Polymorphism)

$$p[\ \mathsf{in}\ r.\langle \mathsf{even}, 3\rangle] \mid q[\ \mathsf{in}\ r.\langle \mathsf{not}, \mathsf{true}\rangle] \mid r[\ (f, x).\, n[\ \langle f\ \ x\rangle \mid \mathcal{P}\ ] \mid \mathsf{open}\ p \mid \mathsf{open}\ q\ ]$$

Case 2. (Subtyping)

$$p[\ \mathsf{in}\ r.\langle 3, 2\rangle] \mid q[\ \mathsf{in}\ r.\langle 3.6, 5.1\rangle] \mid r[\ (x, y).\, n[\ \langle \mathsf{mult}(x, y)\rangle \mid \mathcal{P}\ ] \mid \mathsf{open}\ p \mid \mathsf{open}\ q\ ]$$

More Polymorphism ?

Case 3. (Arity Polymorphism)

$$n[\ \langle \mathsf{true}, 5\rangle \mid \langle 5, 6, 3.6\rangle \mid (x, y).\mathcal{P} \mid (x, y, z).\mathcal{Q}\ ]$$

Case 4. (Orderly Communication)

$$m[\ \langle 7\rangle \mid (x).\mathsf{open}\ n.\langle x = 42\rangle \mid n[\ (y).\mathcal{P}\ ]\ ]$$

Example: Packet Routing

- We use $n\{\mathcal{P}\}$ as an abbreviation for

$$n\{\mathcal{P}\} \triangleq n[\ \mathsf{coopen}\ n \mid \mathcal{P}\ ]$$

- Packet Routing:

$$router[!route\{\mathsf{in}\ packet.(dst).\mathsf{open}\ hop.\langle \mathsf{lookup\text{-}route}(dst)\rangle\}] \ \mid$$

$$packet[\mathsf{in}\ router.\mathsf{open}\ route.\langle\text{“bu”}\rangle \mid hop\{(x).x\}]$$

Example: Code on Demand, Data-Driven Dispatch

- Given prime : int $\rightarrow$ bool and relative-prime : $\times$(int, int) $\rightarrow$ bool,

$$
\begin{array}{lll}
\text{server} & \triangleq & s[\ !tst[\ \text{open}\ p\ |\\
& & (x_1, x_2).\ tstres\{\ x_1.\langle \text{prime}(x_2)\rangle\ \}\ |\\
& & (x_1, x_2, x_3).\ tstres\{\ x_1.\langle \text{relative-prime}(x_2, x_3)\rangle\ \}\ ]\ \ ]
\end{array}
$$

$$
\begin{array}{llll}
\text{client} & \triangleq & \text{LET} & exitPath = \text{out}\ c.\text{in}\ s.\text{in}\ tst.\text{coopen}\ p\\
& & & returnPath = \text{out}\ tst.\text{out}\ s.\text{in}\ c\\
& & \text{IN} & c[\ \text{open}\ tstres.(v).Q\ |\\
& & & \quad p[\ exitPath.\langle returnPath, 1 + 2^{4096}\rangle\ ]\ ]
\end{array}
$$

Example: Code on Demand, Data-Driven Dispatch

- LET-IN desugaring:

$$\begin{array}{llll} \textsf{client} \triangleq \textsf{LET} & \mathit{exitPath} & = \textsf{out}\ c.\textsf{in}\ s.\textsf{in}\ \mathit{tst}.\textsf{coopen}\ p \\ & \mathit{returnPath} & = \textsf{out}\ \mathit{tst}.\textsf{out}\ s.\textsf{in}\ c \\ \textsf{IN} & c[\ \textsf{open}\ \mathit{tstres}.(v).Q\ \mid & \\ & \quad p[\ \mathit{exitPath}.\langle \mathit{returnPath}, 1 + 2^{4096}\rangle\ ]\ ] & \end{array}$$

$$\begin{array}{ll} \textsf{client} \triangleq c[\ \langle \mathit{exitPath}\rangle\ \mid \\ \quad (z_1).\langle z_1, \mathit{returnPath}\rangle\ \mid \\ \quad (z_1, z_2).(\ \textsf{open}\ \mathit{tstres}.(v).Q \mid p[\ z_1.\langle z_2, 1 + 2^{4096}\rangle\ ]\ )\ ] \end{array}$$

AC+

- Expressions: for $k \geqslant 0$,

$$
\begin{aligned}
M \in \mathsf{Exp} \quad ::= \quad & n \mid c \mid \lambda n : \sigma . M \mid M_1 M_2 \mid \times(M_1, \ldots, M_k) \\
& \mid \mathsf{if}\ M_0\ \mathsf{then}\ M_1\ \mathsf{else}\ M_2 \mid \ldots \mid \epsilon \mid M_1 . M_2 \\
& \mid \mathsf{in}\ M \mid \mathsf{out}\ M \mid \mathsf{open}\ M \mid \mathsf{coopen}\ M
\end{aligned}
$$

- Processes: for $k \geqslant 0$,

$$
\begin{aligned}
P \in \mathsf{Proc} \quad ::= \quad & \mathbf{0} \mid P_1 \mathbin{\pmb{|}} P_2 \mid !P \mid (\nu n : \sigma).P \mid M.P \mid M\,\pmb{[}P\pmb{]} \\
& \mid (n_1 : \sigma_1, \ldots, n_k : \sigma_k).P \mid \langle M \rangle
\end{aligned}
$$

Values and Evaluation Contexts

- Values: for $k \geqslant 0$,

$$V ::= n \mid c \mid \lambda n : \sigma.M \mid \times(V_1, \ldots, V_k) \mid \epsilon \mid V_1.V_2$$
$$\mid \mathsf{in}\ V \mid \mathsf{out}\ V \mid \mathsf{open}\ V \mid \mathsf{coopen}\ V$$

- Evaluation Contexts: for $k \geqslant 0$,

$$E ::= \square \mid EM \mid VE \mid \times(V_1, .., V_{i-1}, E, M_{i+1}, .., M_k)$$
$$\mid \mathsf{if}\ E\ \mathsf{then}\ M_1\ \mathsf{else}\ M_2 \mid E.M$$
$$\mid \mathsf{in}\ E \mid \mathsf{out}\ E \mid \mathsf{open}\ E \mid \mathsf{coopen}\ E$$

Operational Semantics

- Expressions:

$(\lambda n : \sigma . M)V \longrightarrow M[n := V]$	(Red Beta)
$cV \longrightarrow V' \quad \text{if } V' = \delta(c, V)$	(Red Delta)
$\text{if true then } M_1 \text{ else } M_2 \longrightarrow M_1$	(Red IfTrue)
$\text{if false then } M_1 \text{ else } M_2 \longrightarrow M_2$	(Red IfFalse)
If $M_1 \longrightarrow M_2$ then $E[M_1] \longrightarrow E[M_2]$	(Red Context)

Operational Semantics

- Mobility and Communication:

$$n[\mathsf{in}\ m.P \mid Q] \mid m[R] \xrightarrow{\epsilon} m[n[P \mid Q] \mid R] \qquad (\mathsf{Red\ In})$$

$$m[n[\mathsf{out}\ m.P \mid Q] \mid R] \xrightarrow{\epsilon} n[P \mid Q] \mid m[R] \qquad (\mathsf{Red\ Out})$$

$$\mathsf{open}\ n.P \mid n[\mathsf{coopen}\ n.Q \mid R] \xrightarrow{\epsilon} P \mid Q \mid R \qquad (\mathsf{Red\ Open})$$

$$(n_1 : \sigma_1, \ldots, n_k : \sigma_k).P \mid \langle \times(V_1, ..., V_k) \rangle$$
$$\xrightarrow{\text{comm}(\tau)} P[n_i := V_i] \text{ if } \tau = \times(\sigma_1, \ldots, \sigma_k) \qquad (\mathsf{Red\ Comm})$$

$$!P \xrightarrow{\epsilon} P \mid\, !P \qquad (\mathsf{Red\ Repl})$$

Operational Semantics

- Reduction Rules:

If $M_1 \longrightarrow M_2$ then $M_1[P] \xrightarrow{\epsilon} M_2[P]$	(Red Name)
If $M_1 \longrightarrow M_2$ then $M_1.P \xrightarrow{\epsilon} M_2.P$	(Red Cap)
If $M_1 \longrightarrow M_2$ then $\langle M_1 \rangle \xrightarrow{\epsilon} \langle M_2 \rangle$	(Red Put)
If $P \xrightarrow{\ell} Q$ then $(\nu n : \sigma).P \xrightarrow{\ell} (\nu n : \sigma).Q$	(Red Res)
If $P \xrightarrow{\ell} Q$ then $n[P] \xrightarrow{\epsilon} n[Q]$	(Red Amb)
If $P \xrightarrow{\ell} Q$ then $P \mid R \xrightarrow{\ell} Q \mid R$	(Red Par)
If $P' \equiv P, P \xrightarrow{\ell} Q, Q \equiv Q'$ then $P' \xrightarrow{\ell} Q'$	(Red $\equiv$)

Example: Subtyping

Case 2. By assigning x and y the type real, p the type $\mathsf{amb}[\mathsf{put}(\mathsf{int}, \mathsf{int})]$, and q the type $\mathsf{amb}[\mathsf{put}(\mathsf{real}, \mathsf{real})]$, we can construct a type derivation for

$$p\{\ \mathsf{in}\ r.\langle 3, 2\rangle\}\ |\ q\{\ \mathsf{in}\ r.\langle 3.6, 5.1\rangle\}\ |$$
$$r[\ (x, y).\ n[\ \langle \mathsf{mult}(x, y)\rangle\ |\ \mathcal{P}\ ]\ |\ \mathsf{open}\ p\ |\ \mathsf{open}\ q\ ]$$

where the process in r has behavior

$$\mathsf{get}(\mathsf{real}, \mathsf{real}).\varepsilon\ |\ \mathsf{put}(\mathsf{int}, \mathsf{int})\ |\ \mathsf{put}(\mathsf{real}, \mathsf{real}).$$

Example: Arity Polymorphism

Case 3. With b and b' the behaviors of P and Q, we can construct a type derivation for

$$n[\ \langle \mathsf{true}, 5\rangle \mid \langle 5, 6, 3.6\rangle \mid (x, y).\mathcal{P} \mid (x, y, z).\mathcal{Q}\ ]$$

where the process in n has behavior

$$\mathsf{put}(\mathsf{bool}, \mathsf{int}) \mid \mathsf{put}(\mathsf{int}, \mathsf{int}, \mathsf{real}) \mid \mathsf{get}(\mathsf{bool}, \mathsf{int}).b \mid \mathsf{get}(\mathsf{int}, \mathsf{int}, \mathsf{real}).b'.$$

Example: Orderly Communication

Case 4. By assigning n the type $\mathsf{amb}[\mathsf{get}(\mathsf{bool}).b]$ (with b the behavior of P), we can construct a type derivation for

$$m[\ \langle 7\rangle \mid (x).\mathsf{open}\ n.\langle x = 42\rangle \mid n\{\ (y).\mathcal{P}\ \}\ ]$$

where the process in m has behavior

$$\mathsf{put}(\mathsf{int}) \mid \mathsf{get}(\mathsf{int}).(\mathsf{put}(\mathsf{bool}) \mid \mathsf{get}(\mathsf{bool}).b) \mid \varepsilon.$$