PLC syntax

Types

$$\begin{array}{rcll} \tau & ::= & \alpha & \text{type variable} \\ & | & \tau \to \tau & \text{function type} \\ & | & \forall\, \alpha\, (\tau) & \forall\text{-type} \end{array}$$

Expressions

$$\begin{array}{rcll} M & ::= & x & \text{variable} \\ & | & \lambda\, x : \tau\, (M) & \text{function abstraction} \\ & | & M\, M & \text{function application} \\ & | & \Lambda\, \alpha\, (M) & \text{type generalisation} \\ & | & M\, \tau & \text{type specialisation} \end{array}$$

(α and x range over fixed, countably infinite sets $\mathbf{TyVar}$ and $\mathbf{Var}$ respectively.)

Datatypes in PLC [Sect. 4.4]

- define a suitable PLC type for the data
- define suitable PLC expressions for values & operations on the data
- show PLC expressions have correct typings & computational behaviour

need to give PLC an operational semantics

Functions on types

In PLC, $\boxed{\Lambda\,\alpha\,(M)}$ is an anonymous notation for the function F mapping each type τ to the value of $M[\tau/\alpha]$ (of some particular type). $\boxed{F\,\tau}$ denotes the result of applying such a function to a type.

Computation in PLC involves beta-reduction for such functions on types

$$(\Lambda\,\alpha\,(M))\,\tau \to M[\tau/\alpha]$$

as well as the usual form of beta-reduction from λ-calculus

$$(\lambda\,x : \tau\,(M_1))\,M_2 \to M_1[M_2/x]$$

Beta-reduction of PLC expressions

M *beta-reduces to* M' *in one step*, $\boxed{M \to M'}$, means M' can be obtained from M (up to alpha-conversion, of course) by replacing a subexpression which is a *redex* by its corresponding *reduct*. The redex-reduct pairs are of two forms:

capture-avoiding substitution

$$(\lambda\, x : \tau\, (M_1))\, M_2 \to M_1[M_2/x]$$
$$(\Lambda\, \alpha\, (M))\, \tau \to M[\tau/\alpha].$$

$M \to^* M'$ indicates a chain of finitely† many beta-reductions.

(† possibly zero—which just means M and M' are alpha-convertible).

M is in *beta-normal form* if it contains no redexes.

[p53]

$$(\lambda x:\alpha_1\to\alpha_1\ (x\,y))\ \left((\Lambda\alpha_2\,(\lambda z:\alpha_2\,(z)))\,(\alpha_1\to\alpha_1)\right)$$

[p53]

$$(\lambda x : \alpha_1 \to \alpha_1\ (x\, y))\ \boxed{(\Lambda \alpha_2\, (\lambda z : \alpha_2\, (z)))(\alpha_1 \to \alpha_1)}$$

$$(\lambda z : \alpha_1 \to \alpha_1\, (z))$$

[p53]

$$(\lambda x:\alpha_1\to\alpha_1\,(xy))\ \boxed{(\Lambda\alpha_2(\lambda z:\alpha_2(z)))(\alpha_1\to\alpha_1)}$$

$$(\lambda x:\alpha_1\to\alpha_1(xy))(\lambda z:\alpha_1\to\alpha_1(z))$$

[p53]

$$(\lambda x:\alpha_1\to\alpha_1\,(xy))\ \boxed{(\Lambda\alpha_2\,(\lambda z:\alpha_2\,(z)))(\alpha_1\to\alpha_1)}$$

$$\boxed{(\lambda x:\alpha_1\to\alpha_1\,(xy))(\lambda z:\alpha_1\to\alpha_1\,(z))}$$

$$(\lambda z:\alpha_1\to\alpha_1\,(z))\,y$$

[p53]

$$
\begin{tikzcd}[column sep=tiny]
& \boxed{(\lambda x:\alpha_1\to\alpha_1\,(xy))\;\boxed{(\Lambda\alpha_2(\lambda z:\alpha_2(z)))(\alpha_1\to\alpha_1)}} \arrow[dl, bend right] \arrow[dr, bend left] & \\
(\Lambda\alpha_2(\lambda z:\alpha_2(z)))(\alpha_1\to\alpha_1)\;y & & \boxed{(\lambda x:\alpha_1\to\alpha_1(xy))(\lambda z:\alpha_1\to\alpha_1(z))} \arrow[dl, bend left] \\
& (\lambda z:\alpha_1\to\alpha_1(z))\,y &
\end{tikzcd}
$$

[p53]

$$\begin{tikzcd}
& \boxed{(\lambda x:\alpha_1\to\alpha_1\,(xy))\ \boxed{(\Lambda\alpha_2(\lambda z:\alpha_2(z)))(\alpha_1\to\alpha_1)}} \arrow[dl] \arrow[dr] & \\
\boxed{(\Lambda\alpha_2(\lambda z:\alpha_2(z)))(\alpha_1\to\alpha_1)}\ y \arrow[dr] & & \boxed{(\lambda x:\alpha_1\to\alpha_1(xy))(\lambda z:\alpha_1\to\alpha_1(z))} \arrow[dl] \\
& (\lambda z:\alpha_1\to\alpha_1(z))\,y &
\end{tikzcd}$$

[p53]

$$
\begin{tikzcd}[column sep=small]
\boxed{(\lambda x:\alpha_1\to\alpha_1\,(xy))\ \boxed{(\Lambda\alpha_2\,(\lambda z:\alpha_2\,(z)))(\alpha_1\to\alpha_1)}} \arrow[d] \arrow[dr] & \\
& \boxed{(\lambda x:\alpha_1\to\alpha_1\,(xy))(\lambda z:\alpha_1\to\alpha_1\,(z))} \arrow[dl] \\
\boxed{(\Lambda\alpha_2\,(\lambda z:\alpha_2\,(z)))(\alpha_1\to\alpha_1)}\ y \arrow[d] & \\
(\lambda z:\alpha_1\to\alpha_1\,(z))\,y \arrow[d] & \\
y &
\end{tikzcd}
$$

Properties of PLC beta-reduction on typeable expressions

Suppose $\Gamma \vdash M : \tau$ is provable in the PLC type system. Then the following properties hold:

Subject Reduction. If $M \to M'$, then $\Gamma \vdash M' : \tau$ is also a provable typing.

Church Rosser Property. If $M \to^* M_1$ and $M \to^* M_2$, then there is M' with $M_1 \to^* M'$ and $M_2 \to^* M'$.

Strong Normalisation Property. There is no infinite chain $M \to M_1 \to M_2 \to \dots$ of beta-reductions starting from M.

Properties of PLC beta-reduction on typeable expressions

Suppose $\Gamma \vdash M : \tau$ is provable in the PLC type system. Then the following properties hold:

Subject Reduction. If $M \to M'$, then $\Gamma \vdash M' : \tau$ is also a provable typing.

Church Rosser Property. If $M \to^* M_1$ and $M \to^* M_2$, then there is M' with $M_1 \to^* M'$ and $M_2 \to^* M'$.

Strong Normalisation Property. There is no infinite chain $M \to M_1 \to M_2 \to \ldots$ of beta-reductions starting from M.

N.B. for the untypeable expression $M \triangleq \lambda x : \alpha (x\,x)$,

$MM = (\lambda x : \alpha (x\,x)) M \to MM \to MM \to \cdots$

Theorem 4.4.2 [p54]

Church Rosser (CR) + Strong Normalization (SN)
$\Rightarrow$ exist unique beta-normal forms for typeable PLC expressions

Existence: start from M & reduce any old way ... must eventually stop by SN

Uniqueness: if

$$\begin{tikzcd} & N_1 \arrow[r, "/" marking] & {} \\ M \arrow[ur, "*"] \arrow[dr, "*"'] & & \\ & N_2 \arrow[r, "/" marking] & {} \end{tikzcd}$$

Theorem 4.4.2 [p54]

> Church Rosser (CR) + Strong Normalization (SN)
> $\Rightarrow$ exist unique beta-normal forms for typeable PLC expressions

Existence: start from M & reduce any old way… must eventually stop by SN

Uniqueness: if

$$\begin{tikzcd} & N_1 \arrow[dr, "*"] & \\ M \arrow[ur, "*"] \arrow[dr, "*"'] & & M' \\ & N_2 \arrow[ur, "*"'] & \end{tikzcd}$$

by CR

Theorem 4.4.2 [p54]

Church Rosser (CR) + Strong Normalization (SN)
$\Rightarrow$ exist unique beta-normal forms for typeable PLC expressions

Existence: start from M & reduce any old way... must eventually stop by SN

Uniqueness: if

$$\begin{tikzcd}
& N_1 \arrow[r, "/" marking] \arrow[dr, "*"] & {} \\
M \arrow[ur, "*"] \arrow[dr, "*"'] & & M' \\
& N_2 \arrow[ur, "*"'] \arrow[r, "/" marking] & {}
\end{tikzcd}$$

so

$$\begin{matrix} N_1 \\ \| \\ M' \\ \| \\ N_2 \end{matrix}$$

PLC beta-conversion, $=_\beta$

By definition, $\boxed{M =_\beta M'}$ holds if there is a finite chain

$$M - \cdot - \cdots - \cdot - M'$$

where each $-$ is either $\rightarrow$ or $\leftarrow$, i.e. a beta-reduction in one direction or the other. (A chain of length zero is allowed—in which case M and M' are equal, up to alpha-conversion, of course.)

Church Rosser + Strong Normalisation properties imply that, for typeable PLC expressions, $M =_\beta M'$ holds if and only if there is some beta-normal form N with

$$M \rightarrow^* N \;{}^*\!\!\leftarrow M'$$

in other words, $=_\beta$ is the <u>smallest</u> equivalence relation containing $\rightarrow$

Polymorphic booleans

$$bool \stackrel{\text{def}}{=} \forall \alpha \, (\alpha \to (\alpha \to \alpha))$$

$$True \stackrel{\text{def}}{=} \Lambda \alpha \, (\lambda \, x_1 : \alpha, x_2 : \alpha \, (x_1))$$

$$False \stackrel{\text{def}}{=} \Lambda \alpha \, (\lambda \, x_1 : \alpha, x_2 : \alpha \, (x_2))$$

$$if \stackrel{\text{def}}{=} \Lambda \alpha \, (\lambda \, b : bool, x_1 : \alpha, x_2 : \alpha \, (b \, \alpha \, x_1 \, x_2))$$

PLC operator association

[Rmk 4.2.2 p 45]

$M_1 M_2 M_3$ means $(M_1 M_2) M_3$

$M_1 M_2 \tau$ means $(M_1 M_2) \tau$, etc.

$\forall \alpha_1, \alpha_2 (\tau)$ means $\forall \alpha_1 (\forall \alpha_2 (\tau))$

$\lambda x_1 : \tau_1 , x_2 : \tau_2 (M)$ means $\lambda x_1 : \tau_1 (\lambda x_2 : \tau_2 (M))$

$\Lambda \alpha_1, \alpha_2 (M)$ means $\Lambda \alpha_1 (\Lambda \alpha_2 (M))$

Polymorphic booleans

$$bool \stackrel{\text{def}}{=} \forall\,\alpha\,(\alpha \to (\alpha \to \alpha))$$

have type bool

$$True \stackrel{\text{def}}{=} \Lambda\,\alpha\,(\lambda\,x_1 : \alpha, x_2 : \alpha\,(x_1))$$

$$False \stackrel{\text{def}}{=} \Lambda\,\alpha\,(\lambda\,x_1 : \alpha, x_2 : \alpha\,(x_2))$$

$$if \stackrel{\text{def}}{=} \Lambda\,\alpha\,(\lambda\,b : bool, x_1 : \alpha, x_2 : \alpha\,(b\,\alpha\,x_1\,x_2))$$

has type $\forall\alpha\,(bool \to (\alpha \to (\alpha \to \alpha)))$

PLC type system

(**var**) $\quad \Gamma \vdash x : \tau \quad$ if $(x : \tau) \in \Gamma$

(**fn**) $$\frac{\Gamma, x : \tau_1 \vdash M : \tau_2}{\Gamma \vdash \lambda\, x : \tau_1\, (M) : \tau_1 \to \tau_2} \quad \text{if } x \notin \mathit{dom}(\Gamma)$$

(**app**) $$\frac{\Gamma \vdash M_1 : \tau_1 \to \tau_2 \quad \Gamma \vdash M_2 : \tau_1}{\Gamma \vdash M_1\, M_2 : \tau_2}$$

(**gen**) $$\frac{\Gamma \vdash M : \tau}{\Gamma \vdash \Lambda\, \alpha\, (M) : \forall\, \alpha\, (\tau)} \quad \text{if } \alpha \notin \mathit{ftv}(\Gamma)$$

(**spec**) $$\frac{\Gamma \vdash M : \forall\, \alpha\, (\tau_1)}{\Gamma \vdash M\, \tau_2 : \tau_1[\tau_2/\alpha]}$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$\mathrm{if}_\tau\, M_1\, M_2\, M_3 \to^* \mathrm{if}_\tau\, \mathrm{True}\, M_2\, M_3$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$\mathrm{if}\,\tau\, M_1\, M_2\, M_3 \to^* \mathrm{if}\,\tau\,\mathrm{True}\, M_2\, M_3$$

$$\qquad\qquad\qquad\qquad || $$

$$\qquad\qquad\qquad\qquad \Lambda\alpha(\ldots)\,\tau\,\mathrm{True}\, M_2\, M_3$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$\mathrm{if}\ \tau\ M_1\ M_2\ M_3 \to^* \mathrm{if}\ \tau\ \mathrm{True}\ M_2\ M_3$$

$$\boxed{\Lambda\alpha(\ldots)\ \tau}\ \mathrm{True}\ M_2\ M_3$$

$$(\lambda b{:}\mathrm{bool}, x_1{:}\tau, x_2{:}\tau\,(b\ \tau\ x_1\ x_2))\ \mathrm{True}\ M_2\ M_3$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$\mathrm{if}\,\tau\, M_1\, M_2\, M_3 \to^* \mathrm{if}\,\tau\,\mathrm{True}\, M_2\, M_3$$

$$\begin{aligned} &\mathrm{if}\,\tau\,\mathrm{True}\, M_2\, M_3 \\ &= \Lambda\alpha(\ldots)\,\tau\,\mathrm{True}\, M_2\, M_3 \\ &\to (\lambda b{:}\mathrm{bool}, x_1{:}\tau, x_2{:}\tau\,(b\,\tau\, x_1\, x_2))\,\mathrm{True}\, M_2\, M_3 \\ &\to^* \mathrm{True}\,\tau\, M_2\, M_3 \end{aligned}$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$
\begin{aligned}
\mathrm{if}\,\tau\, M_1\, M_2\, M_3 &\to^* \mathrm{if}\,\tau\, \mathrm{True}\, M_2\, M_3 \\
&= \Lambda\alpha(\ldots)\,\tau\, \mathrm{True}\, M_2\, M_3 \\
&\to (\lambda b{:}\mathrm{bool}, x_1{:}\tau, x_2{:}\tau\, (b\,\tau\, x_1\, x_2))\, \mathrm{True}\, M_2\, M_3 \\
&\to^* \mathrm{True}\,\tau\, M_2\, M_3 \\
&= \Lambda\alpha\,(\lambda x_1{:}\alpha, x_2{:}\alpha\,(x_1))\,\tau\, M_2\, M_3
\end{aligned}
$$

[p56] If $\begin{cases} M_1 \to^* \mathrm{True} \\ M_2 \to^* N \end{cases}$, then

$$\mathrm{if}\ \tau\ M_1\ M_2\ M_3 \to^* \mathrm{if}\ \tau\ \mathrm{True}\ M_2\ M_3$$

$$\begin{aligned}
&\mathrm{if}\ \tau\ \mathrm{True}\ M_2\ M_3 \\
&\quad\| \\
&\Lambda\alpha(\ldots)\ \tau\ \mathrm{True}\ M_2\ M_3 \\
&\quad\downarrow \\
&(\lambda b:\mathrm{bool}, x_1:\tau, x_2:\tau\,(b\ \tau\ x_1\ x_2))\ \mathrm{True}\ M_2\ M_3 \\
&\quad\downarrow^* \\
&\mathrm{True}\ \tau\ M_2\ M_3 \\
&\quad\| \\
&\Lambda\alpha\,(\lambda x_1:\alpha, x_2:\alpha\,(x_1))\ \tau\ M_2\ M_3
\end{aligned}$$

$$N \ {}^*\!\!\nwarrow\ M_2 \ {}^*\!\!\longleftarrow\ \Lambda\alpha\,(\lambda x_1:\alpha, x_2:\alpha\,(x_1))\ \tau\ M_2\ M_3$$

FACT: $\mathrm{True} \triangleq \Lambda\alpha\,(\lambda x_1, x_2 : \alpha\,(x_1))$

$\mathrm{False} \triangleq \Lambda\alpha\,(\lambda x_1, x_2 : \alpha\,(x_2))$

are the only closed expressions in β-normal form of type $\mathrm{bool} \triangleq \forall\alpha\,(\alpha \to (\alpha \to \alpha))$.