Polymorphism → Dynamic Typing

FAVONIA | NICK BENTON | BOB HARPER

Polymorphism

Dynamic Typing

CORRECT

FAVONIA | NICK BENTON | BOB HARPER

Is this the shortest PLunch?

Bob and I wanted to make homework

Bob and I wanted to make homework and so we had a JFP Theoretical Pearl.

System F to PCF

SOURCE

type var. t

$\downarrow$

"top" type $\mathtt{dyn}$

TARGET

System F to PCF

SOURCE

type var. t

“top” type $\texttt{dyn}$

TARGET

$$\forall t.t \to t \quad \dashrightarrow \quad \texttt{dyn} \rightharpoonup \texttt{dyn}$$

System F to PCF

SOURCE

type var. t

"top" type dyn

TARGET

$\forall t.t \to t \dashrightarrow \texttt{dyn} \rightharpoonup \texttt{dyn}$

But keep other types!

$\texttt{nat} \to \texttt{nat} \dashrightarrow \texttt{nat} \rightharpoonup \texttt{nat}$

$\lambda x{:}\texttt{nat}.x \dashrightarrow \lambda x{:}\texttt{nat}.x$

System F to PCF

System F SOURCE

$$\tau ::= t \mid \texttt{nat} \mid \tau_1 \to \tau_2 \mid \forall t.\tau$$

$$e ::= x \mid \texttt{z} \mid \texttt{suc}(e) \mid \texttt{ifz}(e; e_0; x.e_1) \mid \lambda x{:}\tau.e \mid e_1\, e_2 \mid \Lambda t.e \mid e[\tau]$$

System F to PCF

System F SOURCE

$\tau ::= t \mid \texttt{nat} \mid \tau_1 \to \tau_2 \mid \forall t.\tau$

$e ::= x \mid \texttt{z} \mid \texttt{suc}(e) \mid \texttt{ifz}(e; e_0; x.e_1) \mid \lambda x{:}\tau.e \mid e_1\, e_2 \mid \Lambda t.e \mid e[\tau]$

PCF with dyn TARGET

$\sigma ::= \texttt{nat} \mid \texttt{dyn} \mid \sigma_1 \rightharpoonup \sigma_2$

$d ::= x \mid \cdots \mid \texttt{cast}[c](d) \mid \texttt{new}[c](d) \mid \cdots$

$c ::= \texttt{num} \mid \texttt{fun}$

System F to PCF

System F SOURCE

$\tau ::= t \mid \texttt{nat} \mid \tau_1 \to \tau_2 \mid \forall t.\tau$

$e ::= x \mid \texttt{z} \mid \texttt{suc}(e) \mid \texttt{ifz}(e; e_0; x.e_1) \mid \lambda x{:}\tau.e \mid e_1\, e_2 \mid \Lambda t.e \mid e[\tau]$

PCF with dyn TARGET

$\texttt{new}[\texttt{num}](3) : \texttt{dyn}$

$\texttt{new}[\texttt{fun}](f : \texttt{dyn} \rightharpoonup \texttt{dyn}) : \texttt{dyn}$

$\sigma ::= \texttt{nat} \mid \texttt{dyn} \mid \sigma_1 \rightharpoonup \sigma_2$

$d ::= x \mid \cdots \mid \texttt{cast}[c](d) \mid \texttt{new}[c](d) \mid \cdots$

$c ::= \texttt{num} \mid \texttt{fun}$

System F to PCF

Keep everything except variables

$$
\begin{aligned}
t^\dagger &:= \mathtt{dyn} \\
\mathtt{nat}^\dagger &:= \mathtt{nat} \\
(\tau_1 \to \tau_2)^\dagger &:= \tau_1^\dagger \to \tau_2^\dagger \\
(\forall t.\tau)^\dagger &:= \tau^\dagger
\end{aligned}
$$

System F to PCF

Keep everything except variables

$$t^\dagger := \texttt{dyn}$$

$$\texttt{nat}^\dagger := \texttt{nat}$$

$$(\tau_1 \to \tau_2)^\dagger := \tau_1^\dagger \to \tau_2^\dagger$$

$$(\forall t.\tau)^\dagger := \tau^\dagger$$

$$\frac{\Delta, t \text{ type } \Gamma \vdash e : \tau \Rightarrow d}{\Delta\ \Gamma \vdash \Lambda t.e : \forall t.\tau \Rightarrow d}$$

$$\frac{\Delta \vdash \tau_2 \text{ type} \qquad \Delta\ \Gamma \vdash e : \forall t.\tau_1 \Rightarrow d}{\Delta\ \Gamma \vdash e[\tau_2] : [\tau_2/t]\tau_1 \Rightarrow ???}$$

Type Application

$$f : \forall t.t \to t \xrightarrow{\text{type app}} f[\mathtt{nat}] : \mathtt{nat} \to \mathtt{nat}$$

Type Application

$$\begin{tikzcd} f : \forall t.t \to t \arrow[r, "\text{type app}"] \arrow[d, dashed] & f[\mathtt{nat}] : \mathtt{nat} \to \mathtt{nat} \\ f^{\dagger} : \mathtt{dyn} \rightharpoonup \mathtt{dyn} & \end{tikzcd}$$

Type Application

$$\begin{tikzcd}[column sep=huge] f : \forall t. t \to t \arrow[r, "\text{type app}"] \arrow[d, dashed] & f[\mathtt{nat}] : \mathtt{nat} \to \mathtt{nat} \arrow[d, dashed] \\ f^\dagger : \mathtt{dyn} \rightharpoonup \mathtt{dyn} \arrow[r, "?"] & (f[\mathtt{nat}])^\dagger = ? : \mathtt{nat} \rightharpoonup \mathtt{nat} \end{tikzcd}$$

Type Application

$$
\begin{tikzcd}
f : \forall t.t \to t \arrow[r, "\text{type app}"] \arrow[d, dashed] & f[\mathtt{nat}] : \mathtt{nat} \to \mathtt{nat} \arrow[d, dashed] \\
f^\dagger : \mathtt{dyn} \rightharpoonup \mathtt{dyn} \arrow[r, "?"] & (f[\mathtt{nat}])^\dagger =? : \mathtt{nat} \rightharpoonup \mathtt{nat}
\end{tikzcd}
$$

$$
\begin{tikzcd}
& \cdot \arrow[dl, "{\mathtt{cast}[\mathtt{num}]}"] & \cdot \arrow[l, "f^\dagger"'] & \\
\cdot & & & \cdot \arrow[ul, "{\mathtt{new}[\mathtt{num}]}"']
\end{tikzcd}
$$

$$\lambda x{:}\mathtt{nat}.\mathtt{cast}[\mathtt{num}](f^\dagger(\mathtt{new}[\mathtt{num}](x)))$$

Type Application

$$\begin{tikzcd}
f : \forall t.t \to t \arrow[r, "\text{type app}"] \arrow[d, dashed] & f[\tau] : \tau \to \tau \arrow[d, dashed] \\
f^\dagger : \mathtt{dyn} \rightharpoonup \mathtt{dyn} \arrow[r, "?"] & (f[\tau])^\dagger = ? : \tau^\dagger \rightharpoonup \tau^\dagger
\end{tikzcd}$$

Type Application

$$\begin{tikzcd}
f : \forall t.t \to t \arrow[r, "\text{type app}"] \arrow[d, dashed] & f[\tau] : \tau \to \tau \arrow[d, dashed] \\
f^\dagger : \mathtt{dyn} \rightharpoonup \mathtt{dyn} \arrow[r, "?"] & (f[\tau])^\dagger =? : \tau^\dagger \rightharpoonup \tau^\dagger
\end{tikzcd}$$

$$\lambda x{:}\tau^\dagger . j_{\tau^\dagger}(f^\dagger(i_{\tau^\dagger}(x)))$$

$$i_\sigma : \sigma \rightharpoonup \mathtt{dyn}$$
$$j_\sigma : \mathtt{dyn} \rightharpoonup \sigma$$

Embed and Project

$$i_{\texttt{nat}}(n) := \texttt{new}[\texttt{num}](n)$$

$$i_{\texttt{dyn}}(d) := d$$

$$i_{\sigma_1 \rightharpoonup \sigma_2}(f) := \texttt{new}[\texttt{fun}](\lambda x{:}\texttt{dyn}.i_{\sigma_2}(f(j_{\sigma_1}(x))))$$

$$j_{\texttt{nat}}(d) := \texttt{cast}[\texttt{num}](d)$$

$$j_{\texttt{dyn}}(d) := d$$

$$j_{\sigma_1 \rightharpoonup \sigma_2}(f) := \lambda x{:}\sigma_1.j_{\sigma_2}(\texttt{cast}[\texttt{fun}](f)(i_{\sigma_1}(x)))$$

Type Application

$$\begin{tikzcd}
e : \forall t.\tau_1 \arrow[r, "\text{type app}"] \arrow[d, dashed] & e[\tau_2] : [\tau_2/t]\tau_1 \arrow[d, dashed] \\
e^\dagger : \tau_1^\dagger \arrow[r] & (e[\tau_2])^\dagger = ? : ([\tau_2/t]\tau_1)^\dagger
\end{tikzcd}$$

Type Application

$$\begin{tikzcd}
e : \forall t.\tau_1 \arrow[r, "\text{type app}"] \arrow[d, dashed] & e[\tau_2] : [\tau_2/t]\tau_1 \arrow[d, dashed] \\
e^\dagger : \tau_1^\dagger \arrow[r] & (e[\tau_2])^\dagger = ? : ([\tau_2/t]\tau_1)^\dagger
\end{tikzcd}$$

Idea: lift the projection to handle arbitrary type operators.

Type Application

$$\begin{tikzcd}
e : \forall t.\tau_1 \arrow[r, "\text{type app}"] \arrow[d, dashed] & e[\tau_2] : [\tau_2/t]\tau_1 \arrow[d, dashed] \\
e^\dagger : \tau_1^\dagger \arrow[r] & (e[\tau_2])^\dagger = ? : ([\tau_2/t]\tau_1)^\dagger
\end{tikzcd}$$

Idea: lift the projection to handle arbitrary type operators.

$$[\mathtt{dyn}/\bigcirc][\tau_1]_t = \tau_1^\dagger$$

$$[\tau_2^\dagger/\bigcirc][\tau_1]_t = ([\tau_2/t]\tau_1)^\dagger$$

Type Application

$$\begin{tikzcd}
e : \forall t.\tau_1 \arrow[r, "\text{type app}"] \arrow[d, dashed] & e[\tau_2] : [\tau_2/t]\tau_1 \arrow[d, dashed] \\
e^\dagger : \tau_1^\dagger \arrow[r, "J^{\tau_2^\dagger}_{[\tau_1]_t}"] & (e[\tau_2])^\dagger =? : ([\tau_2/t]\tau_1)^\dagger
\end{tikzcd}$$

Idea: lift the projection to handle arbitrary type operators.

$$[\mathtt{dyn}/\bigcirc][\tau_1]_t = \tau_1^\dagger$$

$$[\tau_2^\dagger/\bigcirc][\tau_1]_t = ([\tau_2/t]\tau_1)^\dagger$$

$I^\sigma_\omega : [\sigma/\bigcirc]\omega \rightharpoonup [\mathtt{dyn}/\bigcirc]\omega$ $\quad$ $J^\sigma_\omega : [\mathtt{dyn}/\bigcirc]\omega \rightharpoonup [\sigma/\bigcirc]\omega$

*Mayer-Wand style

Embed and Project 2.0

$$
\begin{aligned}
[t]_t &:= \bigcirc \\
[u]_t &:= \texttt{dyn} \\
[\texttt{nat}]_t &:= \texttt{nat} \\
[\tau_1 \to \tau_2]_t &:= [\tau_1]_t \rightharpoonup [\tau_2]_t \\
[\forall u.\tau_1]_t &:= [\tau_1]_t
\end{aligned}
$$

Find all the t!

Embed and Project 2.0

$$I^{\sigma}_{\bigcirc}(x) := i_{\sigma}(x)$$

$$I^{\sigma}_{\texttt{nat}}(x) := x$$

$$I^{\sigma}_{\texttt{dyn}}(x) := x$$

$$I^{\sigma}_{\omega_1 \rightharpoonup \omega_2}(f) := \lambda x{:}[\texttt{dyn}/\bigcirc]\omega_1 . I^{\sigma}_{\omega_2}(f(J^{\sigma}_{\omega_1}(x)))$$

$$J^{\sigma}_{\bigcirc}(x) := j_{\sigma}(x)$$

$$J^{\sigma}_{\texttt{nat}}(x) := x$$

$$J^{\sigma}_{\texttt{dyn}}(x) := x$$

$$J^{\sigma}_{\omega_1 \rightharpoonup \omega_2}(f) := \lambda x{:}[\sigma/\bigcirc]\omega_1 . J^{\sigma}_{\omega_2}(f(I^{\sigma}_{\omega_1}(x)))$$

Problem Statement

$$\frac{\Delta, t \text{ type}\ \Gamma \vdash e : \tau \Rightarrow d}{\Delta\ \Gamma \vdash \Lambda t.e : \forall t.\tau \Rightarrow d}$$

$$\frac{\Delta \vdash \tau_2 \text{ type} \qquad \Delta\ \Gamma \vdash e : \forall t.\tau_1 \Rightarrow d}{\Delta\ \Gamma \vdash e[\tau_2] : [\tau_2/t]\tau_1 \Rightarrow J^{\tau_2^\dagger}_{[\tau_1]_t}(d)}$$

Problem Statement

$$\frac{\Delta, t \text{ type } \Gamma \vdash e : \tau \Rightarrow d}{\Delta\ \Gamma \vdash \Lambda t.e : \forall t.\tau \Rightarrow d}$$

$$\frac{\Delta \vdash \tau_2 \text{ type} \qquad \Delta\ \Gamma \vdash e : \forall t.\tau_1 \Rightarrow d}{\Delta\ \Gamma \vdash e[\tau_2] : [\tau_2/t]\tau_1 \Rightarrow J^{\tau_2^\dagger}_{[\tau_1]_t}(d)}$$

(Intended) homework assignment: no run-time errors

Problem Statement

$$\frac{\Delta, t \text{ type } \Gamma \vdash e : \tau \Rightarrow d}{\Delta\ \Gamma \vdash \Lambda t.e : \forall t.\tau \Rightarrow d} \qquad \frac{\Delta \vdash \tau_2 \text{ type} \qquad \Delta\ \Gamma \vdash e : \forall t.\tau_1 \Rightarrow d}{\Delta\ \Gamma \vdash e[\tau_2] : [\tau_2/t]\tau_1 \Rightarrow J^{\tau_2^\dagger}_{[\tau_1]_t}(d)}$$

(Intended) homework assignment: no run-time errors

Igarashi, Pierce and Wadler [2001] showed correctness in a minimal core calculus for Java with generics, relying on the class table.

Problem Statement

$$\frac{\Delta, t \text{ type}\ \Gamma \vdash e : \tau \Rightarrow d}{\Delta\ \Gamma \vdash \Lambda t.e : \forall t.\tau \Rightarrow d} \qquad \frac{\Delta \vdash \tau_2 \text{ type} \qquad \Delta\ \Gamma \vdash e : \forall t.\tau_1 \Rightarrow d}{\Delta\ \Gamma \vdash e[\tau_2] : [\tau_2/t]\tau_1 \Rightarrow J^{\tau_2^\dagger}_{[\tau_1]_t}(d)}$$

(Intended) homework assignment: no run-time errors

Igarashi, Pierce and Wadler [2001] showed correctness in a minimal core calculus for Java with generics, relying on the class table.

Our difficulty: composition of types

$$e \sim_\tau d \; [\eta : \delta \leftrightarrow \rho]$$

System F terms

$\cdot \vdash e : \hat{\delta}(\tau)$

System F types

PCF terms

$\cdot \vdash d : \tau^\dagger$

relations for type variables that respect obs. equiv.

$$e \sim_\tau d \; [\eta : \delta \leftrightarrow \rho]$$

System F terms

$\cdot \vdash e : \hat{\delta}(\tau)$

System F types

PCF terms

$\cdot \vdash d : \tau^\dagger$

relations for type variables that respect obs. equiv.

- $e \sim_t d \; [\eta : \delta \leftrightarrow \rho] := \eta(t)(e, j_{\rho(t)}(d))$
- $e \sim_{\forall t.\tau_1} d \; [\eta : \delta \leftrightarrow \rho] := \forall \tau_2, \sigma_2, \mathcal{R}_2$ such that $\mathcal{R}_2 : \tau_2 \leftrightarrow \sigma_2$, $e[\tau_2] \sim_{\tau_1} d \; [\eta \otimes (t \hookrightarrow \mathcal{R}_2) : \delta \otimes (t \hookrightarrow \tau_2) \leftrightarrow \rho \otimes (t \hookrightarrow \sigma_2)]$

no J here!

Logical Relation Recipe

STEP 1 weakening and exchange

STEP 2 the relation itself respects obs. equiv.
(and the properties you care)

STEP 3 composition of types/relations

STEP 4 fundamental lemma!

$$\dfrac{\Delta\ \Gamma \vdash e : \tau \Rightarrow d \qquad \gamma \sim_{\Gamma} \xi\ [\eta : \delta \leftrightarrow \rho]}{\hat{\gamma}(\hat{\delta}(e)) \sim_{\tau} \hat{\xi}(d)\ [\eta : \delta \leftrightarrow \rho]}$$

Logical Relation Recipe

STEP 1 weakening and exchange

STEP 2 the relation itself respects obs. equiv.
(and the properties you care)

STEP 3 composition of types/relations ?

STEP 4 fundamental lemma!

$$\Delta\ \Gamma \vdash e : \tau \Rightarrow d$$
$$\gamma \sim_\Gamma \xi\ [\eta : \delta \leftrightarrow \rho]$$
$$\hat{\gamma}(\hat{\delta}(e)) \sim_\tau \hat{\xi}(d)\ [\eta : \delta \leftrightarrow \rho]$$

Compositionality

$$[\tau_2/t]\tau_1$$

Compositionality

$$[\tau_2/t]\tau_1$$

$$\mathcal{R}_2(e,d) := e \sim_{\tau_2} d\ [\eta : \delta \leftrightarrow \rho]$$

1. $e \sim_{\tau_1} d\ [\eta \otimes (t \hookrightarrow \mathcal{R}_2) : \delta \otimes (t \hookrightarrow \hat{\delta}(\tau_2)) \leftrightarrow \rho \otimes (t \hookrightarrow \tau_2^{\dagger})]$
   implies $\quad e \sim_{[\tau_2/t]\tau_1} J^{\tau_2^{\dagger}}_{[\tau_1]_t}(d)\ [\eta : \delta \leftrightarrow \rho]$

2. $e \sim_{[\tau_2/t]\tau_1} d\ [\eta : \delta \leftrightarrow \rho]$ implies
   $e \sim_{\tau_1} I^{\tau_2^{\dagger}}_{[\tau_1]_t}(d)\ [\eta \otimes (t \hookrightarrow \mathcal{R}_2) : \delta \otimes (t \hookrightarrow \hat{\delta}(\tau_2)) \leftrightarrow \rho \otimes (t \hookrightarrow \tau_2^{\dagger})]$

Compositionality

$$[\tau_2/t]\tau_1$$

$$\mathcal{R}_2(e,d) := e \sim_{\tau_2} d\ [\eta : \delta \leftrightarrow \rho]$$

1. $e \sim_{\tau_1} d\ [\eta \otimes (t \hookrightarrow \mathcal{R}_2) : \delta \otimes (t \hookrightarrow \hat{\delta}(\tau_2)) \leftrightarrow \rho \otimes (t \hookrightarrow \tau_2^{\dagger})]$

   implies $e \sim_{[\tau_2/t]\tau_1} J^{\tau_2^{\dagger}}_{[\tau_1]_t}(d)\ [\eta : \delta \leftrightarrow \rho]$

2. $e \sim_{[\tau_2/t]\tau_1} d\ [\eta : \delta \leftrightarrow \rho]$ implies

   $e \sim_{\tau_1} I^{\tau_2^{\dagger}}_{[\tau_1]_t}(d)\ [\eta \otimes (t \hookrightarrow \mathcal{R}_2) : \delta \otimes (t \hookrightarrow \hat{\delta}(\tau_2)) \leftrightarrow \rho \otimes (t \hookrightarrow \tau_2^{\dagger})]$

not so clear if it's "iff."

FAILED ATTEMPT

Compositionality

$$e_1 \sim_{[\tau'/t]\tau_1 \to [\tau'/t]\tau_2} J(d_1) \text{ implies(?) } e_1 \sim_{\tau_1 \to \tau_2} d_1$$

FAILED ATTEMPT

Compositionality

$e_1 \sim_{[\tau'/t]\tau_1 \to [\tau'/t]\tau_2} J(d_1)$ implies(?) $e_1 \sim_{\tau_1 \to \tau_2} d_1$

Assume

$$e_2 \sim_{\tau_1} d_2$$

We want

$$e_1\, e_2 \sim_{\tau_2} d_1\, d_2$$

FAILED ATTEMPT

Compositionality

$$e_1 \sim_{[\tau'/t]\tau_1 \to [\tau'/t]\tau_2} J(d_1) \text{ implies(?) } e_1 \sim_{\tau_1 \to \tau_2} d_1$$

Assume

$$e_2 \sim_{\tau_1} d_2$$

We want

$$e_1\, e_2 \sim_{\tau_2} d_1\, d_2$$

We have

$$e_2 \sim_{[\tau'/t]\tau_1} J(d_2)$$

FAILED ATTEMPT

Compositionality

$$e_1 \sim_{[\tau'/t]\tau_1 \to [\tau'/t]\tau_2} J(d_1) \text{ implies(?) } e_1 \sim_{\tau_1 \to \tau_2} d_1$$

Assume

$$e_2 \sim_{\tau_1} d_2$$

We want

$$e_1\, e_2 \sim_{\tau_2} d_1\, d_2$$

We have

$$e_2 \sim_{[\tau'/t]\tau_1} J(d_2)$$

$$e_1\, e_2 \sim_{[\tau'/t]\tau_2} J(d_1)\, J(d_2) \cong J(d_1(I(J(d_2))))$$

FAILED ATTEMPT

Compositionality

$$e_1 \sim_{[\tau'/t]\tau_1 \to [\tau'/t]\tau_2} J(d_1) \text{ implies(?) } e_1 \sim_{\tau_1 \to \tau_2} d_1$$

Assume

$$e_2 \sim_{\tau_1} d_2$$

We want

$$e_1\, e_2 \sim_{\tau_2} d_1\, d_2$$

We have

$$e_2 \sim_{[\tau'/t]\tau_1} J(d_2)$$

$$e_1\, e_2 \sim_{[\tau'/t]\tau_2} J(d_1)\, J(d_2) \cong J(d_1(I(J(d_2))))$$

$$e_1\, e_2 \sim_{\tau_2} d_1(I(J(d_2)))$$

Logical Relation Recipe

STEP 1 weakening and exchange

STEP 2 the relation itself respects obs. equiv.
(and the properties you care)

STEP 3 quasi-composition of types/relations

STEP 4 fundamental lemma!

Thus, compiled programs
give the same numbers!
(no run-time errors)

Concluding Notes

Concluding Notes

A nice trick to deal with embedding/projection!

Concluding Notes

A nice trick to deal with embedding/projection!

We used refinements to make $I(J(d_2)) \cong d_2$
but rejected by POPL. :-(

Concluding Notes

A nice trick to deal with embedding/projection!

We used refinements to make $I(J(d_2)) \cong d_2$
but rejected by POPL. :-(

We have 3 different proofs with different setups.

Concluding Notes

A nice trick to deal with embedding/projection!

We used refinements to make $I(J(d_2)) \cong d_2$
but rejected by POPL. :-(

We have 3 different proofs with different setups.

Thunks to preserve values: $(\forall t.\tau)^\dagger := \texttt{unit} \rightharpoonup \tau^\dagger$

Concluding Notes

A nice trick to deal with embedding/projection!

We used refinements to make $I(J(d_2)) \cong d_2$
but rejected by POPL. :-(

We have 3 different proofs with different setups.

Thunks to preserve values: $(\forall t.\tau)^\dagger := \texttt{unit} \rightharpoonup \tau^\dagger$

You can find open problems by TAing!