Polymorphic Contracts

João Belo, *Michael Greenberg*,
Atsushi Igarashi, Benjamin Pierce

NJPLS
To appear: ESOP 2011

$$
\begin{aligned}
\text{NAT} \; : \; \exists\alpha.\; & Z{:}\alpha \\
& S{:}\alpha \to \alpha \\
& \text{isZero}{:}\alpha \to \text{Bool} \\
& \text{pred}{:}\alpha \to \alpha \\
& \leq{:}\alpha \to \alpha \to \text{Bool} \\
& \text{sub}{:}(x{:}\alpha) \to \alpha \to \alpha
\end{aligned}
$$

$\text{pred } (S\ Z) \longmapsto^{*} Z$

$\text{pred } Z \longmapsto^{*} Z$

$$
\begin{aligned}
\mathrm{NAT} \;:\; \exists\alpha.\;\; & Z{:}\alpha \\
& S{:}\alpha \rightarrow \alpha \\
& \mathrm{isZero}{:}\alpha \rightarrow \mathrm{Bool} \\
& \mathrm{pred}{:}\{x{:}\alpha \mid \mathrm{not}\ (\mathrm{isZero}\ x)\} \rightarrow \alpha \\
& \leq{:}\alpha \rightarrow \alpha \rightarrow \mathrm{Bool} \\
& \mathrm{sub}{:}(x{:}\alpha) \rightarrow \{y{:}\alpha \mid y \leq x\} \rightarrow \{z{:}\alpha \mid z \leq x\}
\end{aligned}
$$

$$\frac{}{c_1 : T} \qquad \frac{e : T}{c_2\ e : T}$$

$$\frac{e_1 : T \quad e_2 : T \quad {\color{blue}P_1(e_1, e_2)}}{c_3\ e_1\ e_2 : T} \qquad \frac{e_1 : T \quad e_2 : T \quad {\color{blue}P_2(e_1) \quad P_2(e_2)}}{c_4\ e_1\ e_2 : T}$$

$\color{orange}{P_2(c_4\ e_1\ e_2)}$

$$
\begin{aligned}
T : \exists\alpha.\ & c_1{:}\alpha \\
& c_2{:}\alpha \to \alpha \\
& p_1{:}\alpha \to \alpha \to \mathrm{Bool} \\
& c_3{:}(x{:}\alpha) \to \{y{:}\alpha \mid p_1\ x\ y\} \to \alpha \\
& p_2{:}\alpha \to \mathrm{Bool} \\
& c_4{:}\{x{:}\alpha \mid p_2\ x\} \to \{y{:}\alpha \mid p_2\ y\} \to \{z{:}\alpha \mid p_2\ z\}
\end{aligned}
$$

First-order contracts

assert($n \geq 0$)

sqrt : $\{x{:}\mathrm{Float} \mid x \geq 0\} \rightarrow \mathrm{Float}$

sqrt : $x{:}\{x{:}\mathrm{Float} \mid x \geq 0\} \rightarrow \{y{:}\mathrm{Float} \mid \mathrm{abs}(y^2 - x) < \epsilon\}$

Higher-order contracts

$$\text{fixed} : (f{:}\{x{:}\text{Int} \mid x \geq 0\} \to \{x{:}\text{Int} \mid x \geq 0\}) \to \{y{:}\text{Int} \mid y = f\ y\}$$

You give a function *f* on nats, I return a fixed point of *f*

- If you don't get a fixed point of *f*, oops—you blame me
- If *f* is called with a negative number, oops—you blame me
- If *f* returns a negative, oops—I blame you

Manifest contracts

Contracts = Types

Manifest contracts

Types $B ::= \mathrm{Bool} \mid \mathrm{Int} \mid \ldots$

$T ::= \{x{:}B \mid e\} \mid x{:}T_1 \to T_2$

Terms $e ::= \ldots \mid \langle T_1 \Rightarrow T_2 \rangle^{\ell} \mid \Uparrow \ell$

$\mathrm{fixed} : (f{:}(\{x{:}\mathrm{Int} \mid x \geq 0\} \to \{x{:}\mathrm{Int} \mid x \geq 0\})) \to \{y{:}\mathrm{Int} \mid y = f\ y\}$

Refinements

$<\{x{:}\mathrm{Int} \mid \mathrm{true}\} \Rightarrow \{x{:}\mathrm{Int} \mid x \geq 0\}>^{\ell}\ 2 \longmapsto^{*} 2$

$<\{x{:}\mathrm{Int} \mid \mathrm{true}\} \Rightarrow \{x{:}\mathrm{Int} \mid x \geq 0\}>^{\ell}\ -5 \longmapsto^{*} \Uparrow \ell$

Functions

$\langle x{:}T_1 \to T_2 \Rightarrow x{:}U_1 \to U_2\rangle^\ell\, f \longmapsto$

$\lambda x{:}\,U_1.\ \langle T_2[{}^{\langle U_1 \Rightarrow T_1\rangle^\ell\, x}/_x] \Rightarrow U_2\rangle^\ell\ (f\ (\langle U_1 \Rightarrow T_1\rangle^\ell\ x))$

Unwind contravariantly; extra cast in the codomain

Motivated by typing rules; see *Greenberg, Pierce, and Weirich 2010*

Our work

Add polymorphism to a manifest calculus

Adding polymorphism

Types $B ::= \mathrm{Bool} \mid \mathrm{Int} \mid \dots$

$T ::= \{x{:}B \mid e\} \mid x{:}T_1 \to T_2 \mid$

$\alpha \mid \forall\alpha.T$

Terms $e ::= \dots \mid \langle T_1 \Rightarrow T_2\rangle^{\ell} \mid \Uparrow\ell \mid$

$\Lambda\alpha.\, e \mid e\; T$

No interaction:

need to put refinements on type variables!

Adding polymorphism

Types $T ::= \mathrm{Bool} \mid \mathrm{Int} \mid \ldots \mid$

$\{x{:}T \mid e\} \mid x{:}T_1 \to T_2 \mid$

$\alpha \mid \forall\alpha.T$

Terms $e ::= \ldots \mid \langle T_1 \Rightarrow T_2\rangle^{\ell} \mid \Uparrow\ell \mid$

$\Lambda\alpha.\ e \mid e\ T$

Adding polymorphism

In the paper:

- Op. sem. for general refinements
- Syntactic type soundness proof
- Proof of parametricity
- Proof of upcast elimination
  - See *Knowles and Flanagan 2010*

Using polymorphism

Using polymorphism

Standard encodings:

Existentials ($\exists\alpha.T$, pack, unpack)

Products ($T_1 \times T_2$ and $\Sigma x{:}T_1.T_2$, fst, snd)

Sums ($T_1 + T_2$, in_L, in_R)

Using polymorphism

$$
\begin{aligned}
\text{NAT} \;:\; \exists\alpha.\;& Z{:}\alpha \\
& S{:}\alpha \to \alpha \\
& \mathrm{isZero}{:}\alpha \to \mathrm{Bool} \\
& \mathrm{pred}{:}\{x{:}\alpha \mid \mathrm{not}\ (\mathrm{isZero}\ x)\} \to \alpha \\
& \leq{:}\alpha \to \alpha \to \mathrm{Bool} \\
& \mathrm{sub}{:}(x{:}\alpha) \to \{y{:}\alpha \mid y \leq x\} \to \{z{:}\alpha \mid z \leq x\}
\end{aligned}
$$

Using polymorphism

NAT = < Z=...,
S=...,
isZero=λn:Nat. ...,
pred=...,
≤=λm:Nat. λn:Nat. ...,
sub=...> pack as ∃α. ...

Interfaces

$\mathrm{sub} : (x{:}\, N) \to \{y{:}\, N \mid y \leq x\} \to \{z{:}\, N \mid z \leq x\}$

$\mathrm{sub} = \langle N \to N \to N \Rightarrow (x{:}N) \to \{y{:}N \mid y \leq x\} \to \{z{:}N \mid z \leq x\} \rangle^{\ell}\ \mathrm{sub}'$

$\mathrm{sub}' = \lambda m{:}N.\ \lambda n{:}N.\ \ldots$

Our contribution

Parametrically polymorphic manifest calculus

Same great theorems

New and improved metatheory

Outlook

Contracts + polymorphism

=

better guarantees for ADTs