Universal Types And Relational Substitutions

Or: How to get free theorems from typing property

Meven Bertrand, after Lau Skorstengaard

May 2nd 2017

Plan

1. System F
2. Contextual Equivalence
3. A Nice Logical Relation
4. How to get a free theorem?

Terms, types and values

System F = simply-typed λ-calculus + polymorphism/universal types

Types

$T ::= bool | \tau \to \tau | \forall\, \alpha,\ \tau | \alpha$

Terms

$e ::= x |\ \mathrm{true}\ |\ \mathrm{false}\ |\ \mathrm{if}\ e\ \mathrm{then}\ e\ \mathrm{else}\ e | \lambda x : \tau.e | e\ e | \Lambda\alpha.e | e[\tau]$

Values

$v ::= \mathrm{true}\ |\ \mathrm{false}\ | \lambda x : \tau.e | \Lambda\alpha.e$

Terms, types and values

System F = simply-typed λ-calculus + polymorphism/universal types

Types

$T ::= bool | \tau \to \tau | \forall\, \alpha,\ \tau | \alpha$

Terms

$e ::= x |\, \text{true} \,|\, \text{false} \,|\, \text{if}\, e\, \text{then}\, e\, \text{else}\, e | \lambda x : \tau.e | e\ e | \Lambda\alpha.e | e[\tau]$

Values

$v ::= \text{true} \,|\, \text{false} \,| \lambda x : \tau.e | \Lambda\alpha.e$

Evaluation

Evaluation context

$E ::= [] | \text{if } E \text{ then } e \text{ else } e | E\ e | v\ E | E[\tau]$

Evaluation

$$\text{if true then } e_1 \text{ else } e_2 \to e_1$$

$$\text{if false then } e_1 \text{ else } e_2 \to e_2$$

$$(\lambda x : \tau . e)\ v \to e[v/x]$$

$$(\Lambda \alpha . e)[\tau] \to e[\tau/\alpha]$$

$$\frac{e \to e'}{E[e] \to E[e']}$$

Typing - I

(Term) Context

$\Gamma ::= \bullet | \Gamma, x : \tau$

Type Context

$\Delta ::= \bullet | \Delta, \alpha$

Context correctness

$\Delta \vdash \tau$ if $\mathrm{FV}(\tau) \subseteq \Delta$
$\Delta \vdash \Gamma$ if $\forall\, x \in \mathrm{dom}(\Gamma),\ \Delta \vdash \Gamma(x)$

Typing - II

Typing rules

$$\frac{}{\Delta; \Gamma \vdash \text{false} : \mathit{bool}}$$

$$\frac{}{\Delta; \Gamma \vdash \text{true} : \mathit{bool}}$$

$$\frac{\Gamma(x) = \tau}{\Delta; \Gamma \vdash x : \tau}$$

$$\frac{\Delta; \Gamma, x : \tau_1 \vdash e : \tau_2}{\Delta; \Gamma \vdash x; \tau_1 . e : \tau_1 \to \tau_2}$$

$$\frac{\Delta; \Gamma \vdash e : \mathit{bool} \qquad \Delta; \Gamma \vdash e_1 : \tau \qquad \Delta; \Gamma \vdash e_2 : \tau}{\Delta; \Gamma \vdash \text{if}\, e\, \text{then}\, e_1\, \text{else}\, e_2 : \tau}$$

$$\frac{\Delta; \Gamma \vdash e_1 : \tau_2 \to \tau \qquad \Delta; \Gamma \vdash e_2 : \tau_2}{\Delta; \Gamma \vdash e_1\ e_2 : \tau}$$

$$\frac{\Delta; \Gamma \vdash e : \forall\, \alpha,\ \tau \qquad \Delta \vdash \tau'}{\Delta; \Gamma \vdash e[\tau'] : e[\tau'/\alpha]}$$

$$\frac{\Delta, \alpha; \Gamma \vdash e :\ \tau}{\Delta; \Gamma \vdash \Lambda \alpha . e : \forall\, \alpha,\ \tau}$$

And $\Gamma; \Delta \vdash e : \tau$ requires $\Delta \vdash \Gamma$

Plan

1. System F
2. Contextual Equivalence
3. A Nice Logical Relation
4. How to get a free theorem?

Why?

Typical theorems we want to prove:
if $\bullet; \bullet \vdash e : \forall\, \alpha,\ \alpha \to \alpha$ then e must "be" the identity
if $\bullet \vdash \tau$, $\bullet; \bullet \vdash e : \forall\, \alpha,\ \alpha \to bool$, $\bullet; \bullet \vdash v_1 : \tau$ and $\bullet; \bullet \vdash v_2 : \tau$ then $e[\tau]\ v_1$ and $e[\tau]\ v_2$ must be "the same"

Formal definition

Context

Context = term with a hole:

$$C ::= [\cdot] | \text{if } C \text{ then } e \text{ else } e | \text{if } e \text{ then } C \text{ else } e | \text{if } e \text{ then } e \text{ else } C | \lambda x : \tau . C | C\ e | e\ C | \Lambda \alpha x . C | C[\tau]$$

Context Typing

$$\frac{\Delta; \Gamma \vdash e : \tau \qquad \Delta'; \Gamma' \vdash C[e] : \tau'}{C : (\Delta; \Gamma \vdash \tau) \to (\Delta'; \Gamma' \vdash \tau')}$$

Contextual equivalence

$\Delta; \Gamma \vdash e_1 \approx_{ctx} e_2 : \tau$ if
$\forall \tau'$ base type, $\forall C : (\Delta; \Gamma \vdash \tau) \to (\bullet; \bullet \vdash \tau'),\ C[e_1] \Downarrow v \leftrightarrow C[e_2] \Downarrow v$

Plan

1. System F
2. Contextual Equivalence
3. A Nice Logical Relation
4. How to get a free theorem?

The general idea

Relation $\mathcal{V}[\![\tau]\!]$ on **values** of type τ: $\mathcal{V}[\![\tau]\!] \subseteq \{(v_1, v_2) | \bullet ; \bullet \vdash v_1 : \tau \wedge \bullet ; \bullet \vdash v_2 : \tau\}$, imitating the definition of SN_τ, and use

$$\mathcal{E}[\![\tau]\!] = \{(e_1, e_2) | \bullet ; \bullet \vdash e_1 : \tau \wedge \bullet ; \bullet \vdash e_2 : \tau \wedge \exists\, (v_1, v_2) \in \mathcal{V}[\![\tau]\!],\ e_1 \Downarrow v_1 \wedge e_2 \Downarrow v_2\}$$

The cases bool and $\rightarrow$

bool

$\mathcal{V}[\![bool]\!] = \{(\text{true}, \text{true}), (\text{false}, \text{false})\}$

$\rightarrow$

$\mathcal{V}[\![\tau \rightarrow \tau']\!] = \{(\lambda x : \tau.e_1, \lambda x : \tau.e_2) | \forall\, (v_1, v_2) \in \mathcal{V}[\![\tau]\!],\ (e_1[v_1/x], e_2[v_2/x]) \in \mathcal{E}[\![\tau']\!]\}$

Recall $SN_{\tau \rightarrow \tau'} = \{e | \bullet \vdash e : \tau \rightarrow \tau' \wedge \forall\, e' \in SN_\tau,\ e\ e' \in SN_{\tau'}\}$

Polymorphic term: the problems begin

First attempt:

$$\mathcal{V}[\![\forall\, \alpha,\ \tau]\!] = \{(\Lambda\alpha.e_1, \Lambda\alpha.e_2) | \forall\, \tau_1\tau_2,\ (e_1[\tau_1/\alpha], e_2[\tau_2/\alpha]) \in \mathcal{E}[\![\tau[\mathbf{??}/\alpha]]\!]\}$$

Different types for $e_1[\tau_1/\alpha]$ and $e_2[\tau_2/\alpha]$!

Keep α and add a substitution:

$$\mathcal{V}[\![\forall\, \alpha,\ \tau]\!]_\rho = \{(\Lambda\alpha.e_1, \Lambda\alpha.e_2) | \forall\, \tau_1\tau_2,\ (e_1[\tau_1/\alpha], e_2[\tau_2/\alpha]) \in \mathcal{E}[\![\tau]\!]_{\rho[\alpha \to (\tau_1,\tau_2)]}\}$$

Radboud University Nijmegen

Polymorphic term: the problems begin

First attempt:

$$\mathcal{V}[\![\forall\, \alpha,\ \tau]\!] = \{(\Lambda\alpha.e_1, \Lambda\alpha.e_2) | \forall\, \tau_1\tau_2,\ (e_1[\tau_1/\alpha], e_2[\tau_2/\alpha]) \in \mathcal{E}[\![\tau[\mathbf{??}/\alpha]]\!]\}$$

Different types for $e_1[\tau_1/\alpha]$ and $e_2[\tau_2/\alpha]$!

Keep α and add a substitution:

$$\mathcal{V}[\![\forall\, \alpha,\ \tau]\!]_\rho = \{(\Lambda\alpha.e_1, \Lambda\alpha.e_2) | \forall\, \tau_1\tau_2,\ (e_1[\tau_1/\alpha], e_2[\tau_2/\alpha]) \in \mathcal{E}[\![\tau]\!]_{\rho[\alpha \to (\tau_1, \tau_2)]}\}$$

Free variable: still not working

First attempt:

$$\mathcal{V}[\![\alpha]\!]_\rho = \{(v_1, v_2) | \rho(\alpha) = (\tau_1, \tau_2) \wedge \textbf{??}\}$$

We need a relation $R \subseteq \text{Rel}[\tau_1, \tau_2] = \{(v_1, v_2) | \bullet; \bullet \vdash v_1 : \tau \wedge \bullet; \bullet \vdash v_2 : \tau\}$, just add it as a third component to ρ:

$$\mathcal{V}[\![\alpha]\!]_\rho = \rho_R(\alpha)$$

with $\rho = (\rho_1, \rho_2, \rho_R)$ and $\rho_R(\alpha) \in \text{Rel}(\rho_1(\alpha), \rho_2(\alpha))$

Free variable: still not working

First attempt:

$$\mathcal{V}[\![\alpha]\!]_\rho = \{(v_1, v_2) | \rho(\alpha) = (\tau_1, \tau_2) \wedge \textbf{??}\}$$

We need a relation $R \subseteq \mathrm{Rel}[\tau_1, \tau_2] = \{(v_1, v_2) | \bullet; \bullet \vdash v_1 : \tau \wedge \bullet; \bullet \vdash v_2 : \tau\}$, just add it as a third component to ρ:

$$\mathcal{V}[\![\alpha]\!]_\rho = \rho_R(\alpha)$$

with $\rho = (\rho_1, \rho_2, \rho_R)$ and $\rho_R(\alpha) \in \mathrm{Rel}(\rho_1(\alpha), \rho_2(\alpha))$

All definitions together

First,

$$\mathcal{V}[\![\tau]\!]_\rho \subseteq \{(v_1, v_2) | \bullet; \bullet \vdash v_1 : \rho_1(\tau) \wedge \bullet; \bullet \vdash v_2 : \rho_2(\tau)\}$$

and

$$\mathcal{E}[\![\tau]\!] = \{(e_1, e_2) | \bullet; \bullet \vdash e_1 : \rho_1(\tau) \wedge \bullet; \bullet \vdash e_2 : \rho_2(\tau) \wedge \exists (v_1, v_2) \in \mathcal{V}[\![\tau]\!]_\rho,\ e_1 \Downarrow v_1 \wedge e_2 \Downarrow v_2\}$$

Then, the updated definitions

$$\mathcal{V}[\![bool]\!]_\rho = \{(\text{true}, \text{true}), (\text{false}, \text{false})\}$$

$$\mathcal{V}[\![\tau \to \tau']\!]_\rho = \{(\lambda x : \rho_1(\tau).e_1, \lambda x : \rho_2(\tau).e_2) | \forall (v_1, v_2) \in \mathcal{V}[\![\tau]\!]_\rho,\ (e_1[v_1/x], e_2[v_2/x]) \in \mathcal{E}[\![\tau']\!]_\rho\}$$

$$\mathcal{V}[\![\forall \alpha,\ \tau]\!]_\rho = \{(\Lambda \alpha.e_1, \Lambda \alpha.e_2) | \forall \tau_1 \tau_2 R \in \text{Rel}[\tau_1, \tau_2],\ (e_1[\tau_1/\alpha], e_2[\tau_2/\alpha]) \in \mathcal{E}[\![\tau]\!]_{\rho[\alpha \to (\tau_1, \tau_2, R)]}\}$$

$$\mathcal{V}[\![\alpha]\!]_\rho = \rho_R(\alpha)$$

Interpretation for a context and the relation

Context interpretation

$\mathcal{D}[\![\bullet]\!] = \{\emptyset\}$ and $\mathcal{D}[\![\Delta, \alpha]\!] = \{\rho[\alpha \to (\tau_1, \tau_2, R)], \rho \in \mathcal{D}[\![\Delta]\!] \wedge R \in \mathrm{Rel}[\tau_1, \tau_2]\}$
$\mathcal{G}[\![\bullet]\!]_\rho = \{\emptyset\}$ and $\mathcal{G}[\![\Gamma, x : \tau]\!]_\rho = \{\gamma[x \to (v_1, v_2)], \gamma \in \mathcal{G}[\![\Gamma]\!] \wedge (v_1, v_2) \in \mathcal{V}[\![\tau]\!]_\rho\}$

The relation (finally!)

$\Delta; \Gamma \vdash e_1 \approx e_2 : \tau$ if $\Delta; \Gamma \vdash e_1 : \tau$ and $\Delta; \Gamma \vdash e_2 : \tau$ and
$\forall\, \rho \in \mathcal{D}[\![\Delta]\!],\ \forall\, \gamma \in \mathcal{G}[\![\Gamma]\!]_\rho,\ (\rho_1(\gamma_1(e_1)), \rho_2(\gamma_2(\rho_2(e_2)))) \in \mathcal{E}[\![\tau]\!]_\rho$

Plan

1. System F
2. Contextual Equivalence
3. A Nice Logical Relation
4. How to get a free theorem?

Properties of $\approx$

Big property of $\approx$: $\Delta; \Gamma \vdash e_1 \approx e_2 : \tau$ implies $\Delta; \Gamma \vdash e_1 \approx_{ctx} e_2 : \tau$
Hard to prove, not always needed: often another theorem suffices:

Fundamental property of $\approx$

If $\Delta; \Gamma \vdash e : \tau$, then $\Delta; \Gamma \vdash e \approx e : \tau$

Proof using compatibility lemmas mimicking induction definitions:

$$\frac{}{\Delta;\ \Gamma \vdash \text{true} \approx \text{true} : \mathit{bool}}$$

$$\frac{\Delta; \Gamma, x :\ \tau \vdash e_1 \approx e_2 : \tau'}{\Delta; \Gamma \vdash (\lambda x : \tau . e_1) \approx (\lambda x : \tau . e_2) : \tau \to \tau'}$$

and so on ...

An example

The theorem

If $\bullet;\ \bullet \vdash e : \forall\, \alpha,\ \alpha \to \alpha$, $\bullet \vdash \tau$ and $\bullet; \bullet \vdash v : \tau$ is a value, then $e[\tau]\ v \Downarrow v$.

The Proof

We start with $\bullet; \bullet \vdash e \approx e : \forall\, \alpha,\ \alpha \to \alpha$, so $(e, e) \in \mathcal{E}[\![\forall\, \alpha,\ \alpha \to \alpha]\!]_{\emptyset}$.
Get F s.t. $e \Downarrow F$, then $(F, F) \in \mathcal{V}[\![\forall\, \alpha,\ \alpha \to \alpha]\!]_{\emptyset}$, say $F = \Lambda\alpha.e_1$. Take $\tau_1 = \tau_2 = \tau$ and $R = \{(v, v)\}$, then $(e_1[\tau/\alpha], e_1[\tau/\alpha]) \in \mathcal{E}[\![\alpha \to \alpha]\!]_{\emptyset[\alpha \to (\tau,\tau,R)]}$.
Again $e_1[\tau/\alpha] \Downarrow \lambda x : \tau.e_2$ and $(\lambda x :\ \tau.e_2, \lambda x :\ \tau.e_2) \in \mathcal{V}[\![\alpha \to \alpha]\!]_{\emptyset[\alpha \to (\tau,\tau,R)]}$, instantiate with v: $(e_2[v/x], e_2[v/x]) \in \mathcal{E}[\![\alpha]\!]_{\emptyset[\alpha \to (\tau,\tau,R)]}$.
Once again, $e_2[v/x] \to^* v_f$ with $v_f : \tau$ and $(v_f, v_f) \in \mathcal{V}[\![\alpha]\!]_{\emptyset[\alpha \to (\tau,\tau,R)]} = R$.
Thus $(v_f, v_f) = (v, v)$ since $R = \{(v, v)\}$.
Now

$$e[\tau]\ v \to^* (\Lambda\alpha.e_1) \to e_1[\tau/\alpha]\ v \to^* (\lambda x : \tau.e_2)\ v \to e_2[v/x] \to^* v$$

so $e[\tau]\ v \Downarrow v$ as v is a value.