Parametricity and GADTs

Dimitrios Vytiniotis
Stephanie Weirich

Computer and Information Science Department
University of Pennsylvania

Boston, July 2006

A very simple GADT example

```
data R :: * -> * where
  Rint  :: R Int
  Rbool :: R Bool

inc :: forall a. R a -> a -> a
inc Rint  x = x + 1
inc Rbool x = True
```

A very simple GADT example

```
inc :: forall a. R a -> a -> a
inc Rint  x = x + 1
inc Rbool x = True
```

This is a strange function:

- Can't apply `inc` to all types.
- The argument of type `a` is not treated parametrically.
- So, what does *parametricity* mean in this language?

Overview

1. System F + this GADT
2. Parametricity theorem for this language
3. Free theorems
4. Other GADTs

Overview

1. System F + this GADT
2. Parametricity theorem for this language
3. Free theorems
4. Other GADTs

This is all work in progress.

System F

$$
\begin{array}{lcl}
\tau, \sigma & ::= & int \mid bool \mid \alpha \mid \sigma \rightarrow \sigma \mid \forall a.\sigma \\
e & ::= & i \mid b \mid \lambda x.e \mid e_1\ e_2 \mid \Lambda\alpha.e \mid e[\sigma] \mid \ \ldots \\
 & & \\
v & ::= & i \mid \lambda x.e
\end{array}
$$

System F + R

$$
\begin{array}{lcl}
\tau, \sigma & ::= & int \mid bool \mid \alpha \mid \sigma \to \sigma \mid \forall a.\sigma \mid R\ \tau \\
e & ::= & i \mid b \mid \lambda x.e \mid e_1\ e_2 \mid \Lambda\alpha.e \mid e[\sigma] \mid \ldots \\
 & \mid & R_{int} \mid R_{bool} \mid case\ e\ e_{int}\ e_{bool} \\
v & ::= & i \mid \lambda x.e \mid R_{int} \mid R_{bool}
\end{array}
$$

System F + R

$$
\begin{array}{lcl}
\tau, \sigma & ::= & int \mid bool \mid \alpha \mid \sigma \rightarrow \sigma \mid \forall a.\sigma \mid R\ \tau \\
e & ::= & i \mid b \mid \lambda x.e \mid e_1\ e_2 \mid \Lambda\alpha.e \mid e[\sigma] \mid \ldots \\
 & \mid & R_{int} \mid R_{bool} \mid case\ e\ e_{int}\ e_{bool} \\
v & ::= & i \mid \lambda x.e \mid R_{int} \mid R_{bool}
\end{array}
$$

$$
\begin{array}{l}
inc :: \forall a.R\ \alpha \rightarrow \alpha \rightarrow \alpha \\
inc = \lambda x.case\ x\ (\lambda y.y + 1)\ (\lambda z.true)
\end{array}
$$

Typing rules

$$\frac{}{\Gamma \vdash R_{int} : R\ int}$$

$$\frac{}{\Gamma \vdash R_{bool} : R\ bool}$$

$$\frac{\Gamma \vdash e : R\ \tau \quad \Gamma \vdash e_{int} : \sigma\{int/\alpha\} \quad \Gamma \vdash e_{bool} : \sigma\{bool/\alpha\}}{\Gamma \vdash case\ e\ e_{int}\ e_{bool} : \sigma\{\tau/\alpha\}}$$

Bigstep, CBN Operational Semantics

$$\frac{}{v \Downarrow v}$$

$$\frac{e_1 \Downarrow \lambda x.e_1' \qquad e_1'\{e_2/x\} \Downarrow v}{e_1\, e_2 \Downarrow v} \qquad \frac{e_1 \Downarrow \Lambda a.e_1' \qquad e_1'\{\sigma/\alpha\} \Downarrow v}{e_1[\sigma] \Downarrow v}$$

$$\frac{e \Downarrow R_{int} \qquad e_{int} \Downarrow v}{case\;\; e\;\; e_{int}\;\; e_{bool} \Downarrow v} \qquad \frac{e \Downarrow R_{bool} \qquad e_{bool} \Downarrow v}{case\;\; e\;\; e_{int}\;\; e_{bool} \Downarrow v}$$

Some Definitions

Definition (Typed value relations)

Let $\mathcal{V}(\tau_1, \tau_2)$ be the set of relations between closed values of closed type τ_1 and τ_2.

Some Definitions

Definition (Typed value relations)

Let $\mathcal{V}(\tau_1, \tau_2)$ be the set of relations between closed values of closed type τ_1 and τ_2.

Definition (Type substitution)

A type substitution η is a map from type variables to (τ_1, τ_2, r) where τ_1 and τ_2 are closed types and $r \in \mathcal{V}(\tau_1, \tau_2)$. If $\eta(\alpha) = (\tau_1, \tau_2, r)$, then let $\eta_1(\alpha) = \tau_1$, $\eta_2(\alpha) = \tau_2$ and $\eta_r(\alpha) = r$.

Some Definitions

Definition (Typed value relations)

Let $\mathcal{V}(\tau_1, \tau_2)$ be the set of relations between closed values of closed type τ_1 and τ_2.

Definition (Type substitution)

A type substitution η is a map from type variables to (τ_1, τ_2, r) where τ_1 and τ_2 are closed types and $r \in \mathcal{V}(\tau_1, \tau_2)$. If $\eta(\alpha) = (\tau_1, \tau_2, r)$, then let $\eta_1(\alpha) = \tau_1$, $\eta_2(\alpha) = \tau_2$ and $\eta_r(\alpha) = r$.

Definition (Computational closure)

If $r \in \mathcal{V}(\tau_1, \tau_2)$, then define r° as
$\{(e_1, e_2) \mid \emptyset \vdash e_1 : \tau_1 \wedge \emptyset \vdash e_2 : \tau_2 \wedge e_1 \Downarrow v_1 \wedge e_2 \Downarrow v_2 \wedge (v_1, v_2) \in r\}$.

Logical Relation (System F)

$$
\begin{array}{lcl}
[\![int]\!]_\eta & = & \{(i, i)\} \\
[\![bool]\!]_\eta & = & \{(b, b)\} \\
[\![\sigma_1 \to \sigma_2]\!]_\eta & = & \{(v_1, v_2) \mid \\
& & \quad \emptyset \vdash v_1 : \eta_1(\sigma_1 \to \sigma_2) \wedge \emptyset \vdash v_2 : \eta_2(\sigma_1 \to \sigma_2) \\
& & \quad \forall (e_1, e_2) \in [\![\sigma_1]\!]_\eta{}^\circ \Rightarrow \\
& & \qquad (v_1\ e_1, v_2\ e_2) \in [\![\sigma_2]\!]_\eta{}^\circ\} \\
[\![\forall \alpha.\sigma]\!]_\eta & = & \{(v_1, v_2) \mid \emptyset \vdash v_1 : \eta_1(\forall \alpha.\sigma) \wedge \emptyset \vdash v_2 : \eta_2(\forall \alpha.\sigma) \\
& & \quad \forall \tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2), \\
& & \qquad (v_1[\tau_1], v_2[\tau_2]) \in [\![\sigma]\!]_{\eta, \alpha \mapsto (\tau_1, \tau_2, r)}{}^\circ\} \\
[\![\alpha]\!]_\eta & = & \eta_r(\alpha)
\end{array}
$$

Parametricity Theorem

Definition (Related substitution)

Let γ be a mapping from term variables to pairs of closed expressions. Say $\Gamma, \eta \vdash \gamma$ iff $\forall x : \sigma \in \Gamma, (\gamma_1(x), \gamma_2(x)) \in [\![\sigma]\!]_\eta{}^\circ$.

Parametricity Theorem

Definition (Related substitution)

Let γ be a mapping from term variables to pairs of closed expressions. Say $\Gamma, \eta \vdash \gamma$ iff $\forall x : \sigma \in \Gamma, (\gamma_1(x), \gamma_2(x)) \in [\![\sigma]\!]_\eta{}^\circ$.

Theorem (Fundamental theorem)

If $\Gamma \vdash e : \sigma$ *and* $ftv(\Gamma, e, \sigma) = dom(\eta)$ *and* $\Gamma, \eta \vdash \gamma$ *then* $(\gamma_1(e), \gamma_2(e)) \in [\![\sigma]\!]_\eta{}^\circ$.

Relation for R types

$$
\begin{array}{lcl}
[\![R\ int]\!]_\eta & = & \{(R_{int}, R_{int})\} \\
[\![R\ bool]\!]_\eta & = & \{(R_{bool}, R_{bool})\} \\
[\![R\alpha]\!]_\eta & = & \begin{cases} [\![R\tau]\!]_\emptyset & \text{when } \eta_1(\alpha) = \eta_2(\alpha) = \tau \\ & \quad \text{and } \eta_r(\alpha) = [\![\tau]\!]_\emptyset \\ & \quad \text{and } \tau \text{ is a closed monotype} \\ \emptyset & \text{otherwise} \end{cases} \\
[\![R\ \tau]\!]_\eta & = & \emptyset \text{ otherwise}
\end{array}
$$

Relation for R types

$$
\begin{array}{lcl}
[\![R\ int]\!]_\eta & = & \{(R_{int}, R_{int})\} \\
[\![R\ bool]\!]_\eta & = & \{(R_{bool}, R_{bool})\} \\
[\![R\alpha]\!]_\eta & = & \begin{cases} \mathcal{C}[\![R\tau]\!] & \text{when } \eta_1(\alpha) = \eta_2(\alpha) = \tau \\ & \quad \text{and } \eta_r(\alpha) = \mathcal{C}[\![\tau]\!] \\ & \quad \text{and } \tau \text{ is a closed monotype} \\ \emptyset & \text{otherwise} \end{cases} \\
[\![R\ \tau]\!]_\eta & = & \emptyset \text{ otherwise}
\end{array}
$$

Closed relation

$$
\begin{array}{lcl}
\mathcal{C}[\![int]\!] & = & \{(i, i)\} \\
\mathcal{C}[\![bool]\!] & = & \{(b, b)\} \\
\mathcal{C}[\![\sigma_1 \to \sigma_2]\!] & = & \{(v_1, v_2) \mid \\
 & & \quad \emptyset \vdash v_1 : \eta_1(\sigma_1 \to \sigma_2) \wedge \emptyset \vdash v_2 : \eta_2(\sigma_1 \to \sigma_2) \\
 & & \quad \forall (e_1, e_2) \in \mathcal{C}[\![\sigma_1]\!]^\circ \Rightarrow \\
 & & \qquad (v_1\ e_1, v_2\ e_2) \in \mathcal{C}[\![\sigma_2]\!]^\circ\} \\
\mathcal{C}[\![R\ int]\!] & = & \{(R_{int}, R_{int})\} \\
\mathcal{C}[\![R\ bool]\!] & = & \{(R_{bool}, R_{bool})\} \\
\mathcal{C}[\![\sigma]\!] & = & \emptyset \text{ otherwise}
\end{array}
$$

Lemma

If τ is a closed monotype then $[\![\tau]\!]_\emptyset = \mathcal{C}[\![\tau]\!]$

A free theorem

Consider a closed expression f of type $\forall\alpha.\alpha \to \alpha$. The free theorem for this type is:

$$\begin{aligned}&\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),\\&\qquad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]x, f[\tau_2]y) \in r^\circ\end{aligned}$$

A free theorem

Consider a closed expression f of type $\forall\alpha.\alpha \to \alpha$. The free theorem for this type is:

$$\begin{aligned}&\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),\\ &\qquad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]x, f[\tau_2]y) \in r^\circ\end{aligned}$$

We can use this theorem to show that forall values v of type τ, $f[\tau]v \Downarrow v$.

A free theorem

Consider a closed expression f of type $\forall\alpha.\alpha \to \alpha$. The free theorem for this type is:

$$\begin{aligned}&\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),\\ &\qquad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]x, f[\tau_2]y) \in r^\circ\end{aligned}$$

We can use this theorem to show that forall values v of type τ, $f[\tau]v \Downarrow v$.
Let r be the relation $\{(v, v)\}$.

A free theorem

Consider a closed expression f of type $\forall\alpha.\alpha \rightarrow \alpha$. The free theorem for this type is:

$$\begin{array}{l}\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2), \\ \qquad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]x, f[\tau_2]y) \in r^\circ\end{array}$$

We can use this theorem to show that forall values v of type τ, $f[\tau]v \Downarrow v$.
Let r be the relation $\{(v, v)\}$.
Now, $\forall(x, y) \in r^\circ \Rightarrow (f[\tau]x, f[\tau]y) \in r^\circ$.

A free theorem

Consider a closed expression f of type $\forall\alpha.\alpha \to \alpha$. The free theorem for this type is:

$$\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),$$
$$\forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]x, f[\tau_2]y) \in r^\circ$$

We can use this theorem to show that forall values v of type τ, $f[\tau]v \Downarrow v$.
Let r be the relation $\{(v, v)\}$.
Now, $\forall(x, y) \in r^\circ \Rightarrow (f[\tau]x, f[\tau]y) \in r^\circ$.
So $(v, v) \in r^\circ \Rightarrow (f[\tau]v, f[\tau]v) \in r^\circ$.

Free theorem for inc

Now consider a closed expression f of type $\forall\alpha.R\alpha \to \alpha \to \alpha$.

Free theorem for inc

Now consider a closed expression f of type $\forall\alpha.R\alpha \to \alpha \to \alpha$.

$$\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),$$

Free theorem for inc

Now consider a closed expression f of type $\forall\alpha.R\alpha \to \alpha \to \alpha$.

$$
\begin{array}{l}
\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2), \\
\quad (\tau_1 = \tau_2 \wedge r = [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow \forall(v, w) \in [\![R\tau]\!]_\emptyset{}^\circ, \\
\qquad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]\ v\ x, f[\tau_2]\ w\ y) \in r^\circ)
\end{array}
$$

Free theorem for inc

Now consider a closed expression f of type $\forall\alpha.R\alpha \rightarrow \alpha \rightarrow \alpha$.

$$
\begin{array}{l}
\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2), \\
\quad (\tau_1 = \tau_2 \wedge r = [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow \forall(v, w) \in [\![R\tau]\!]_\emptyset{}^\circ, \\
\quad\quad \forall(x, y) \in r^\circ \Rightarrow (f[\tau_1]\ v\ x, f[\tau_2]\ w\ y) \in r^\circ) \\
\quad \wedge (\tau_1 \neq \tau_2 \vee r \neq [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow \forall(v, w) \in \emptyset^\circ, \\
\quad\quad \forall(x, y) \in r^\circ, (f[\tau_1]\ v\ x, f[\tau_2]\ w\ y) \in r^\circ)
\end{array}
$$

You get what you pay for

Now consider a closed expression f of type $\forall\alpha.R\alpha \to R\alpha$, which *is* an identity function.
The free theorem for this type is:

$$
\begin{array}{l}
\forall\tau_1,\tau_2,r \in \mathcal{V}(\tau_1,\tau_2), \\
\quad (\tau_1 = \tau_2 \wedge r = [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow \\
\qquad \forall(x,y) \in [\![R\tau_1]\!]_\emptyset{}^\circ, (f[\tau_1]\ x, f[\tau_2]\ y) \in [\![R\tau]\!]_\emptyset{}^\circ) \\
\quad \wedge(\tau_1 \neq \tau_2 \vee r \neq [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow \\
\qquad \forall(x,y) \in \emptyset^\circ, (f[\tau_1]\ x, f[\tau_2]\ y) \in \emptyset^\circ)
\end{array}
$$

This theorem is also uninteresting—all it says is that when given equal arguments, f will produce equal results.

Not always useless

Consider a closed expression f of type $\forall\alpha.R\alpha$. The free theorem for this type is:

Not always useless

Consider a closed expression f of type $\forall\alpha.R\alpha$. The free theorem for this type is:

$$\forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2),$$

Not always useless

Consider a closed expression f of type $\forall\alpha.R\alpha$. The free theorem for this type is:

$$\begin{array}{l}\forall\tau_1,\tau_2,r\in\mathcal{V}(\tau_1,\tau_2),\\ \quad(\tau_1=\tau_2\wedge r=[\![\tau_1]\!]_\emptyset{}^\circ\Rightarrow(f[\tau_1],f[\tau_2])\in[\![R\tau_1]\!]_\emptyset{}^\circ)\end{array}$$

Not always useless

Consider a closed expression f of type $\forall\alpha.R\alpha$. The free theorem for this type is:

$$
\begin{aligned}
&\forall\tau_1,\tau_2, r\in\mathcal{V}(\tau_1,\tau_2),\\
&\quad(\tau_1=\tau_2\wedge r=[\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow (f[\tau_1],f[\tau_2])\in[\![R\tau_1]\!]_\emptyset{}^\circ)\\
&\quad\wedge(\tau_1\neq\tau_2\vee r\neq[\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow (f[\tau_1],f[\tau_2])\in\emptyset^\circ)
\end{aligned}
$$

Not always useless

Consider a closed expression f of type $\forall\alpha.R\alpha$. The free theorem for this type is:

$$\begin{array}{l} \forall\tau_1, \tau_2, r \in \mathcal{V}(\tau_1, \tau_2), \\ \quad (\tau_1 = \tau_2 \wedge r = [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow (f[\tau_1], f[\tau_2]) \in [\![R\tau_1]\!]_\emptyset{}^\circ) \\ \quad \wedge (\tau_1 \neq \tau_2 \vee r \neq [\![\tau_1]\!]_\emptyset{}^\circ \Rightarrow (f[\tau_1], f[\tau_2]) \in \emptyset^\circ) \end{array}$$

By this theorem, $(f[int], f[bool]) \in \emptyset^\circ$. So there cannot be any such f.

Alternative reasoning

Lemma (Canonical forms)

1. *If* $\emptyset \vdash v : R\ int$ *then* $v = R_{int}$.
2. *If* $\emptyset \vdash v : R\ bool$ *then* $v = R_{bool}$.
3. *There are no closed values of type* $R\ \sigma$, *when* σ *is not int or bool.*

Alternative reasoning

Lemma (Canonical forms)

1. *If* $\emptyset \vdash v : R\ int$ *then* $v = R_{int}$*.*
2. *If* $\emptyset \vdash v : R\ bool$ *then* $v = R_{bool}$*.*
3. *There are no closed values of type* $R\ \sigma$*, when* σ *is not int or bool.*

Using this this lemma, we can show that if $f : \forall\alpha.R\alpha \rightarrow R\alpha$ then for all $\emptyset \vdash v : R\tau$, $f[\tau]\ v \Downarrow v$.

Vector GADT

Consider another GADT.

```
data Z :: *
data S :: * -> *

data Vec :: * -> * -> * where
  Nil  :: Vec Z a
  Cons :: a -> Vec n a -> Vec (S n) a
```

More formally

$$\Gamma \vdash Nil : \forall\alpha.Vec\ Z\ \alpha$$

$$\Gamma \vdash Cons : \forall\alpha\beta.\alpha \to Vec\ \beta\ \alpha \to Vec\ (S\ \beta)\alpha$$

$$\frac{\begin{array}{l}\Gamma \vdash e : Vec\ \sigma_{ind}\ \sigma \\ \Gamma \vdash e_n : \sigma'\{Z/\alpha\} \\ \Gamma \vdash e_c : \forall\beta.\sigma \to \sigma'\{\beta/\alpha\} \to \sigma'\{S\ \beta/\alpha\}\end{array}}{\Gamma \vdash case\ e\ e_n\ e_c : \sigma'\{n/\alpha\}}$$

Logical relation

$$
\begin{array}{lcl}
[\![Z]\!]_\eta & = & \emptyset \\
[\![S\sigma]\!]_\eta & = & \emptyset \\
[\![Vec\ Z\ \sigma]\!]_\eta & = & \{(Nil, Nil)\} \\
[\![Vec\ (S\ \sigma_i)\ \sigma]\!]_\eta & = & \{(Cons[\eta_1(\sigma)][\eta_1(\sigma_i)]\ x_1\ y_1, \\
 & & \quad Cons[\eta_2(\sigma)][\eta_2(\sigma_i)]\ x_2\ y_2) \mid \\
 & & \quad (x_1, x_2) \in [\![\sigma]\!]_\eta, \quad (y_1, y_2) \in [\![Vec\ \sigma_i\ \sigma]\!]_\eta\} \\
[\![Vec\ \alpha\ \sigma]\!]_\eta & = & \begin{cases} [\![Vec\ \tau\ \sigma]\!]_\eta & \text{when } \eta_1(\alpha) = \eta_2(\alpha) = \tau \\ \emptyset & \text{otherwise} \end{cases} \\
[\![Vec\ \sigma_i\ \sigma]\!]_\eta & = & \emptyset \text{ otherwise}
\end{array}
$$

Logical relation

$$
\begin{array}{lcl}
[\![Z]\!]_\eta & = & \emptyset \\
[\![S\sigma]\!]_\eta & = & \emptyset \\
[\![Vec\ Z\ \sigma]\!]_\eta & = & \{(Nil, Nil)\} \\
[\![Vec\ (S\ \sigma_i)\ \sigma]\!]_\eta & = & \{(Cons[\eta_1(\sigma)][\eta_1(\sigma_i)]\ x_1\ y_1, \\
 & & \quad Cons[\eta_2(\sigma)][\eta_2(\sigma_i)]\ x_2\ y_2) \mid \\
 & & \quad (x_1, x_2) \in [\![\sigma]\!]_\eta, \quad (y_1, y_2) \in [\![Vec\ \sigma_i\ \sigma]\!]_\eta\} \\
[\![Vec\ \alpha\ \sigma]\!]_\eta & = & \begin{cases} [\![Vec\ \tau\ \sigma]\!]_\eta & \text{when } \eta_1(\alpha) = \eta_2(\alpha) = \tau \\ \emptyset & \text{otherwise} \end{cases} \\
[\![Vec\ \sigma_i\ \sigma]\!]_\eta & = & \emptyset \text{ otherwise}
\end{array}
$$

Note: Because the index type is empty, don't need to restrict $\eta_r(\alpha)$.

Where to next?

- More free theorems.
- Leave the "pure" world.
- Parametricity for general GADTs.
- Mechanize everything in a theorem prover. Dimitrios has a good start in Isabelle/HOL.