Relational Parametricity *for* Higher Kinds

Robert Atkey
robert.atkey@strath.ac.uk

University of Strathclyde
Glasgow, UK

5th September 2012

Higher Kinds

Higher Kinded Polymorphism

System F: Quantification over *types*:

$$\forall\alpha.\ \mathit{List}\,\alpha \to \mathit{List}\,\alpha$$

System Fω: Quantification over *type operators*:

$$\forall_{*\to*}f.\ \forall_{*}\alpha.\ f\,\alpha \to f\,\alpha$$

and type-level λ-abstraction:

$$\begin{aligned}
\mathit{List} &= \lambda\alpha : * \to *.\ \forall_{*}\beta.\ \beta \to (\alpha \to \beta \to \beta) \to \beta \\
\mathit{Monad} &= \lambda m : * \to *.\ (\forall_{*}\alpha.\ \alpha \to m\,\alpha)\times \\
&\qquad\qquad\qquad (\forall_{*}\alpha\beta.\ m\,\alpha \to (\alpha \to m\,\beta) \to m\,\beta)
\end{aligned}$$

Present in Haskell, Scala, and ML (via the module system)

Church Encodings

Booleans

$$Bool = \forall\alpha.\ \alpha \to \alpha \to \alpha$$

Naturals

$$Nat = \forall\alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$$

Lists

$$List\ \alpha = \forall\beta.\ \beta \to (\alpha \to \beta \to \beta) \to \beta$$

and initial algebras, (co)products, final coalgebras, existentials.

but only *weakly* initial or final : $\left\{ \begin{array}{l} \text{no uniqueness} \\ \text{no reasoning principle} \end{array} \right.$

Church Encodings with Higher Kinds

Lists

$$List = \lambda\alpha : *.\ \forall_* \beta.\ \beta \to (\alpha \to \beta \to \beta) \to \beta$$

Vectors

$$\begin{aligned} Vec = \lambda\alpha\ n : *.\ \ & \forall_{* \to *}\beta.\ \beta\ Z \to \\ & (\forall_* n.\ \alpha \to \beta\ n \to \beta\ (S\ n)) \to \beta\ n \end{aligned}$$

Equality

$$Eq_\kappa = \lambda\alpha\beta : \kappa.\ \forall_{\kappa \to \kappa \to *} f.\ (\forall_\kappa \gamma.\ f\gamma\gamma) \to f\alpha\beta$$

but only *weakly* initial (or final) : $\left\{ \begin{array}{l} \text{no uniqueness} \\ \text{no reasoning principle} \end{array} \right.$

Relational Parametricity

(Reynolds, 1983)

Relational Parametricity

For example,

$$e : \forall\alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha$$

let X and Y be sets, and let $R \subseteq X \times Y$

if we have $z_1 \in X$, $z_2 \in Y$ such that:

$$(z_1, z_2) \in R$$

and $s_1 : X \to X$, $s_2 : Y \to Y$ such that:

$$\forall(a, b) \in R.\ (s_1\ a, s_2\ b) \in R$$

then

$$(e\ [X]\ z_1\ s_1, e\ [Y]\ z_2\ s_2) \in R$$

Preservation of Relations

implies initiality, and $(\forall\alpha.\ \alpha \to (\alpha \to \alpha) \to \alpha) \cong \mathbb{N}$

Relational Parametricity

Relational interpretations of types

$$\mathcal{R}[\![\Theta \vdash A]\!]\ \theta\ \theta'\ \rho \subseteq \mathcal{T}[\![\Theta \vdash A]\!]\ \theta \times \mathcal{T}[\![\Theta \vdash A]\!]\ \theta'$$

$$\begin{aligned}
\mathcal{R}[\![\alpha]\!]\ \rho &= \rho(\alpha) \\
\mathcal{R}[\![A \to B]\!]\ \rho &= \{(f_1, f_2) \mid \forall (a_1, a_2) \in \mathcal{R}[\![A]\!]\rho.\ (f_1\ a_1, f_2\ a_2) \in \mathcal{R}[\![B]\!]\rho\} \\
\mathcal{R}[\![\forall \alpha.\ A]\!]\ \rho &= \{(x_1, x_2) \mid\ \ \forall X, Y, R \subseteq X \times Y. \\
&\qquad\qquad (x_1\ [X], x_2\ [Y]) \in \mathcal{R}[\![A]\!](\rho[\alpha \mapsto R])\}
\end{aligned}$$

Relational Parametricity

Identity Extension:

$$\forall x, y \in \mathcal{T}[\![\Theta \vdash A]\!]\theta \quad \Rightarrow \quad ((x, y) \in \mathcal{R}[\![\Theta \vdash A]\!]([\![\Theta]\!]^{\Delta}\theta) \Leftrightarrow x = y)$$

and *Abstraction*:

$$\Theta \mid - \vdash e : A \quad \Rightarrow \quad [\![e]\!] \in \mathcal{T}[\![\Theta \vdash A]\!]\theta$$

Manufacturing Relationally Parametric Models

Option I: find them

Operational Models (Pitts, 2000)

Manufacturing Relationally Parametric Models

Option II: force them

Mutually define base and relational interpretations of types

(Reynolds, 1983) (Bainbridge *et al.*, 1990)

$$
\begin{aligned}
\mathcal{T}[\![\alpha]\!]\theta &= \theta(\alpha) \\
\mathcal{T}[\![A \to B]\!]\theta &= \mathcal{T}[\![A]\!]\theta \to \mathcal{T}[\![B]\!]\theta \\
\mathcal{T}[\![\forall\alpha.A]\!]\theta &= \{\, x : \forall X.\ \mathcal{T}[\![A]\!](\theta[\alpha \mapsto X]) \\
&\qquad \mid \forall X, Y, R \subseteq X \times Y. \\
&\qquad\quad \mathcal{R}[\![\tau]\!](\Delta_\theta[\alpha \mapsto R])\ (x\, A_1)\ (x\, A_2) \ \} \\
\mathcal{R}[\![\alpha]\!]\rho &= \rho(\alpha) \\
\mathcal{R}[\![A \to B]\!]\rho &= \{(f_1, f_2) \mid \forall (a_1, a_2) \in \mathcal{R}[\![A]\!]\rho.\ (f_1\ a_1, f_2\ a_2) \in \mathcal{R}[\![B]\!]\rho\} \\
\mathcal{R}[\![\forall\alpha.\tau]\!]\rho\ x\ y &= \{(x_1, x_2) \mid \forall X, Y, R \subseteq X \times Y. \\
&\qquad (x\,[X], y\,[Y]) \in \mathcal{R}[\![\tau]\!](\rho[\alpha \mapsto R])\}
\end{aligned}
$$

$$
\text{then} : \begin{cases} \text{prove Identity Extension} \\ \text{prove Abstraction} \end{cases}
$$

Relational Parametricity
for
Higher Kinds

Relational Parametricity *for* Higher Kinds

How to interpret kinds?

Implicitly:

$$[\![*]\!] = \text{Set} \quad \text{and} \quad [\![*]\!]^R = (X, Y) \mapsto \mathcal{P}(X \times Y)$$

So let us try:

$$[\![*]\!] = \text{Set}$$
$$[\![\kappa_1 \to \kappa_2]\!] = [\![\kappa_1]\!] \to [\![\kappa_2]\!]$$

and

$$[\![\kappa]\!]^R \; : \; [\![\kappa]\!] \times [\![\kappa]\!] \to \text{Set}$$
$$[\![*]\!]^R = (X, Y) \mapsto \mathcal{P}(X, Y)$$
$$[\![\kappa_1 \to \kappa_2]\!]^R = (F, G) \mapsto \forall X, Y. [\![\kappa_1]\!]^R(X, Y) \to [\![\kappa_2]\!]^R(FX, GY)$$

Relational Parametricity *for* Higher Kinds

Identity extension?

Recall identity extension:

$$\forall x, y \in \mathcal{T}[\![\Theta \vdash A : *]\!]\theta \quad \Rightarrow \quad ((x, y) \in \mathcal{R}[\![\Theta \vdash A : *]\!]([\![\Theta]\!]^{\Delta}\theta) \Leftrightarrow x = y)$$

What is $[\![* \to *]\!]^{\Delta}(F)$?

No good answer in general.

Solution

build-in an "identity" for every semantic type operator

require every semantic type operator to preserve identities

Kinds as Reflexive Graphs

Reflexive Graph Categories

(Hasegawa, 1994)
(Robinson and Rosolini, 1994)
(Dunphy and Reddy, 2004)

Let $RG = \bullet \underset{\overset{\longleftarrow}{\delta_1}}{\overset{\overset{\delta_0}{\longleftarrow}}{\xrightarrow{\;i\;}}} \bullet$ such that $\delta_0 \circ i = id$ and $\delta_1 \circ i = id$.

Interpret kinds as elements of Set_1^{RG}.

Kinds as "Categories without Composition"

A kind is interpreted as a pair of (large) sets O and R, with maps:

$$
\begin{aligned}
id &: O \to R \\
src &: R \to O \\
tgt &: R \to O
\end{aligned}
$$

Higher kinds are interpreted using the cartesian-closed structure.

Interpretation of System Fω

Interpretation of Kinds

Kinds interpreted as "categories without composition"

$$[\![*]\!] = (\mathrm{Set}, \{(A, B, R \subseteq A \times B) \mid A, B \in \mathrm{Set}\})$$

Interpretation of Types $\Theta \vdash A : \kappa$

— interpreted as a functor "without composition"

— actually, natural transformations in Set_1^{RG}

— recreates the mutual induction used for System F

Interpretation of Terms $\Theta \mid \Gamma \vdash e : A$

— interpreted as a natural transformations "without composition"

— yields the standard abstraction theorem

Applications
of
Relational Parametricity
for
Higher Kinds

Equality Types

Specification

$\mathrm{Eq}_\kappa : \kappa \to \kappa \to *$, with

$$\begin{aligned} \mathrm{refl}_\kappa &: \forall_\kappa \alpha.\ \mathrm{Eq}_\kappa \alpha\alpha \\ \mathrm{elimEq}_\kappa &: \forall_\kappa \alpha\beta.\ \mathrm{Eq}_\kappa \alpha\beta \to \forall_{\kappa\to\kappa\to *}\rho.(\forall_\kappa \gamma.\ \rho\gamma\gamma) \to \rho\alpha\beta \end{aligned}$$

with β- and η-laws.

Implementation

$$\begin{aligned} Eq_\kappa &= \lambda\alpha\beta : \kappa.\ \forall_{\kappa\to\kappa\to *} f.\ (\forall_\kappa \gamma.\ f\,\gamma\,\gamma) \to f\,\alpha\,\beta \\ refl_\kappa &= \Lambda\alpha.\ \Lambda\rho.\ \lambda f.\ f\,[\alpha] \\ elimEq_\kappa &= \Lambda\alpha\beta.\ \lambda e.\ \Lambda\rho.\ \lambda f.\ e\,[\rho]\,f \end{aligned}$$

use relational parametricity to prove the η-law

Existential Types

Specification

For $F : \kappa \to *$, $\exists_\kappa \alpha.\ F\alpha$, with

$$\begin{aligned}\mathrm{pack}_\kappa &: \forall_{\kappa\to *}\rho.\ \forall_\kappa \alpha.\ \rho\alpha \to (\exists_\kappa \alpha.\ \rho\alpha)\\ \mathrm{elimEx}_\kappa &: \forall_{\kappa\to *}\rho.\ \forall_* \beta.\ (\forall_\kappa \alpha.\ \rho\alpha \to \beta) \to (\exists_\kappa \alpha.\ \rho\alpha) \to \beta\end{aligned}$$

with β- and η-laws

Implementation

$$\begin{aligned}\exists_\kappa \alpha.\ F\alpha &= \forall_* \beta.(\forall_\kappa \alpha.F\alpha \to \beta) \to \beta\\ \mathit{pack}_\kappa &= \Lambda\rho\alpha.\lambda x.\Lambda\beta.\lambda f.f\,[\alpha]\ x\\ \mathit{elimEx}_\kappa &= \Lambda\rho\beta.\lambda fe.\ e\,[\beta]\ f\end{aligned}$$

use relational parametricity to prove the η-law

Higher-Kinded Initial Algebras

Specification

For *functors* $(F : (\kappa \to *) \to (\kappa \to *), \mathit{fmap}_F)$, $\mu F : \kappa \to *$, with

$$\begin{aligned} \mathrm{in}_F &: \forall_\kappa \alpha.\ F(\mu F)\alpha \to (\mu F)\alpha \\ \mathrm{fold}_F &: \forall_{\kappa \to *}\rho.(\forall_\kappa \alpha.\ F\rho\alpha \to \rho\alpha) \to (\forall_\kappa \alpha.\ (\mu F)\alpha \to \rho\alpha) \end{aligned}$$

with β- and η-laws

Implementation

$$\begin{aligned} \mu F &= \lambda\alpha.\forall_{\kappa \to *}\rho.(\forall_\kappa \beta.\ F\rho\beta \to \rho\beta) \to \rho\alpha \\ \mathit{fold}_F &= \Lambda\rho.\lambda f.\Lambda\alpha.\lambda x.\ x\,[\rho]\,f \\ \mathit{in}_F &= \Lambda\gamma.\lambda x.\Lambda\rho.\lambda f.\ f\,[\gamma]\,(\mathit{fmap}_F\,[\mu F]\,[\rho]\,(\mathit{fold}_F\,[\rho]\,f)\,[\gamma]\,x) \end{aligned}$$

use relational parametricity to prove the η-law

GADTs

Generalised Algebraic Datatypes

Example from Haskell:

```
data Z
data S a
data Vec :: * -> * -> * where
  VNil  :: Vec a Z
  VCons :: a -> Vec a n -> Vec a (S n)
```

Encoding using Initial Algebras and Equality Types

(Johann and Ghani, 2008)

$$Vec = \lambda\alpha.\ \mu(F\alpha) \qquad \text{where}$$

$$F\alpha\rho n = Eq_*\ n\ Z + (\exists_* n'.\alpha \times \rho\ n' \times Eq_*\ n\ (S\ n'))$$

Summary

Summary

Relationally parametric model of System $F\omega$

- Kinds as reflexive graphs
- Types as functors without composition
- Constructed within impredicative CIC
- Equality in parametric model implies observational equiv *(in the paper)*

Applications of Higher-kinded Parametricity

- Equality types
- Existentials
- Initial Algebras
- Generalised Algebraic Datatypes
- Natural number indexed types *(in the paper)*

Future work

- Extension to dependent types
- Final coalgebras