

Theorems for Free: an Introduction

José N. Oliveira
Dept. Informatica
Universidade do Minho, 4700 Braga, Portugal
jno@di.uminho.pt
DI/UM, 2003-06
May 23, 2006

Parametric polymorphism: why?

- Less code (**specific** solution = **generic** solution + **customization**)
- Intellectual reward
- Last but not least, quotation (from **Theorems for free!**, by Philip Wadler [?]):

*From the **type** of a polymorphic function we can derive a **theorem** that is satisfies. (...) How useful are the theorems so generated? Only time and experience will tell (...)*

- No doubt: free theorems are **very** useful!

Polymorphic type signatures

Polymorphic function signature:

$$f : t$$

where t is a functional type, according to the following “grammar” of types:

$$t ::= t' \leftarrow t''$$

$$t ::= F(t_1, \dots, t_n)$$

$$t ::= v \quad \text{type variables } v, \text{ cf. polymorphism}$$

What does it mean that f is **parametrically** polymorphic?

Free theorem of type t

Let

- V be the set of type variables involved in type t
- $\{R_v\}_{v \in V}$ be a V -indexed family of relations (f_v in case all such R_v are functions).
- R_t be a relation defined inductively as follows:

$$R_{t=F(t_1, \dots, t_n)} = F(R_{t_1}, \dots, R_{t_n})$$

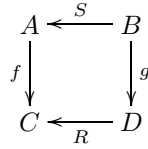
$$R_{t=v} = R_v$$

$$R_{t=t' \leftarrow t''} = R_{t'} \leftarrow R_{t''}$$

- What kind of relation is $R_{t'} \leftarrow R_{t''}$?

Reynolds arrow operator

$$f(R \leftarrow S)g \equiv f \cdot S \subseteq R \cdot g$$



That is to say,

$$\frac{\begin{array}{ccc} A & \xleftarrow{S} & B \\ & \xleftarrow{R} & \\ C & \xleftarrow{R} & D \end{array}}{C^A \xleftarrow{R \leftarrow S} D^B}$$

For instance, $f(id \leftarrow id)g \equiv f = g$ that is, $id \leftarrow id = id$

Free theorem (FT) of type t

The following (remarkable) theorem — due to J. Reynolds and advertised by P. Wadler — holds:

Given any function $\theta : t$, and V as above, then $\theta R_t \theta$ holds, for any relational instantiation of type variables in V .

Note that this theorem

- is a result about t
- holds **independently** of the actual definition of θ .
- holds about any function of type t

First example

- The target function: $\theta = \text{invl} : a^* \leftarrow a^*$.
- Calculation of $R_{t=a^* \leftarrow a^*}$:

$$\begin{aligned}
 & R_{a^* \leftarrow a^*} \\
 \equiv & \quad \{ \text{rule } R_{t=t' \leftarrow t''} = R_{t'} \leftarrow R_{t''} \} \\
 & R_{a^*} \leftarrow R_{a^*} \\
 \equiv & \quad \{ \text{rule } R_{t=F(t_1, \dots, t_n)} = F(R_{t_1}, \dots, R_{t_n}) \} \\
 & R_{a^*}^* \leftarrow R_{a^*}^*
 \end{aligned}$$

- Calculation of FT $\text{invl} (R_{a^* \leftarrow a^*}) \text{invl}$ follows

invl FT calculation

The FT itself will predict (R_a abbreviated to R):

$$\begin{aligned}
 & \text{invl}(R^* \leftarrow R^*) \text{invl} \\
 \equiv & \quad \{ \text{definition } f(R \leftarrow S)g \equiv f \cdot S \subseteq R \cdot g \} \\
 & \text{invl} \cdot R^* \subseteq (R^*) \cdot \text{invl}
 \end{aligned}$$

In case R is a function r , the FT theorem boils down to *invl*'s **natural** property:

$$\text{invl} \cdot r^* = r^* \cdot \text{invl}$$

Further calculation (back to R):

Pointwise version of FT

$$\begin{aligned}
 & \text{invl} \cdot R^* \subseteq R^* \cdot \text{invl} \\
 \equiv & \quad \{ \text{shunting rule} \} \\
 & R^* \subseteq \text{invl}^\circ \cdot R^* \cdot \text{invl} \\
 \equiv & \quad \{ \text{going pointwise} \} \\
 & l(R^*)r \Rightarrow (\text{invl } l)(R^*)(\text{invl } r) \\
 \equiv & \quad \{ \text{pointwise definition of } R^* \} \\
 & \forall i \in \text{inds } l.(l \ i)R(r \ i) \Rightarrow (\text{invl } l)(R^*)(\text{invl } r)
 \end{aligned}$$

Pointwise version of FT

For example, *invl* will respect orderings:

$$\begin{aligned}
 & \forall i \in \text{inds } l.(l \ i) < (r \ i) \\
 \Rightarrow & \quad \forall j \in \text{inds}(\text{invl } l).(\text{invl } l)j < (\text{invl } r)j
 \end{aligned}$$

Exercise: calculate the FT of

$$\text{sort} : a^* \leftarrow a^* \leftarrow (2 \leftarrow (a \times a))$$

(the first parameter stands for the ordering relation.)

Second example: FT of (\downarrow_-)

- (\downarrow_-) has generic type

$$(\downarrow_-) : b \leftarrow F a \leftarrow (b \leftarrow B(a, b))$$

where $F a \cong B(a, F a)$.

- FT- (\downarrow_-) :

$$(\downarrow_-) \cdot (R_b \leftarrow B(R_a, R_b)) \subseteq (R_b \leftarrow F R_a) \cdot (\downarrow_-)$$

- FT- (\downarrow_-) calculation follows $(R_a, R_b$ abbreviated to $R, S)$:

FT- $\langle _ \rangle$ corollaries

$$\begin{aligned}
 & \langle _ \rangle \cdot (S \leftarrow B(R, S)) \subseteq (S \leftarrow F R) \cdot \langle _ \rangle \\
 \equiv & \quad \{ \text{definition } f(R \leftarrow S)g \equiv f \cdot S \subseteq R \cdot g \} \\
 & f(S \leftarrow B(R, S))g \Rightarrow \langle f \rangle (S \leftarrow F R) \langle g \rangle \\
 \equiv & \quad \{ \text{idem} \} \\
 & f \cdot B(R, S) \subseteq S \cdot g \Rightarrow \langle f \rangle \cdot F R \subseteq S \cdot \langle g \rangle
 \end{aligned}$$

At this point, we can infer ...

FT- $\langle _ \rangle$ corollaries

From this, infer

- $\langle _ \rangle$ -fusion $(R, S := id, s)$:

$$(f \cdot B(id, s) = s \cdot g) \Rightarrow \langle f \rangle = s \cdot \langle g \rangle$$

- $\langle _ \rangle$ -absorption $(R, S := r, id)$:

$$\begin{aligned}
 & (f \cdot B(r, id) = g) \Rightarrow \langle f \rangle \cdot F r = \langle g \rangle \\
 \equiv & \quad \{ \text{replacement of } g \} \\
 & \langle f \rangle \cdot F r = \langle f \cdot B(r, id) \rangle
 \end{aligned}$$

Bibliography

[Wad89] P. Wadler. Theorems for free! In *4th International Symposium on Functional Programming Languages and Computer Architecture*, London, Sep. 1989. ACM.