"Theorems for free": a (calculational) introduction

J.N. Oliveira

Dept. Informática,
Universidade do Minho
Braga, Portugal

2003 (updated 2008)

Parametric polymorphism: why?

- Less code (**specific** solution = **generic** solution + **customization**)
- Intellectual reward
- Last but not
  least, quotation (from *Theorems for free!*, by Philip Wadler [4]):

  > *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:

$$f \quad : \quad t$$

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

$$\begin{array}{lcll} t & ::= & t' \leftarrow t'' & \\ t & ::= & \mathsf{F}(t_1, \ldots, t_n) & \\ t & ::= & v & \text{type } \textit{variables } v\text{, cf. } \textit{polymorphism} \end{array}$$

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:

$$R_{t:=\mathsf{F}(t_1,\ldots,t_n)} = \mathsf{F}(R_{t_1}, \ldots, R_{t_n}) \quad (1)$$

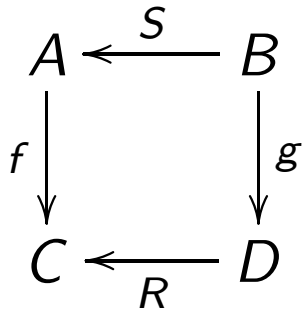
$$R_{t:=v} = R_v \quad (2)$$

$$R_{t:=t' \leftarrow t''} = R_{t'} \leftarrow R_{t''} \quad (3)$$

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

"Reynolds arrow" operator

$$f(R \leftarrow S)g \quad \equiv \quad f \cdot S \subseteq R \cdot g \qquad (4)$$



That is to say,

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

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

Free theorem (FT) of type t

The following (remarkable) theorem — due to J. Reynolds [3], advertised by P. Wadler [4] and re-written by Backhouse [1] in the pointfree style — 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 (id)

- The target function: $\theta = id : a \leftarrow a$.
- Calculation of $R_{t=a\leftarrow a}$:

$$
\begin{array}{ll}
 & R_{a\leftarrow a} \\
\equiv & \quad \{ \text{ rule } \ R_{t=t'\leftarrow t''} \ = \ R_{t'} \leftarrow R_{t''} \ \} \\
 & R_a \leftarrow R_a
\end{array}
$$

- Calculation of FT (R_a abbreviated to R):

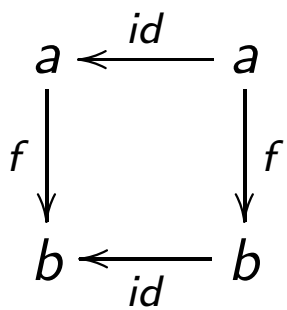
$$
\begin{array}{ll}
 & id(R \leftarrow R)id \\
\equiv & \quad \{ \ (4) \ \} \\
 & id \cdot R \subseteq R \cdot id
\end{array}
$$

First example (id)

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

$$id \cdot f \;=\; f \cdot id$$

cf.



which can be read alternatively as stating that id is the **unit** of composition.

Second example (*invl*)

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

$$
\begin{array}{ll}
 & R_{a^\star \leftarrow a^\star} \\
\equiv & \quad \{ \text{ rule } \quad R_{t=t' \leftarrow t''} \quad = \quad R_{t'} \leftarrow R_{t''} \quad \} \\
 & R_{a^\star} \leftarrow R_{a^\star} \\
\equiv & \quad \{ \text{ rule } \quad R_{t=\mathsf{F}(t_1,\ldots,t_n)} \quad = \quad \mathsf{F}(R_{t_1},\ldots,R_{t_n}) \quad \} \\
 & R_a{}^\star \leftarrow R_a{}^\star
\end{array}
$$

where

$$l \; R^\star l' \stackrel{\text{def}}{=} length \; l = length \; l' \wedge \langle \forall \; i \; : \; i \in inds \; l : \; (l \; i) R (l' \; i) \rangle$$

Calculation of FT follows.

Second example (*invl*)

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

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

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

$$invl \cdot r^\star \;=\; r^\star \cdot invl$$

that is,

$$invl[\; r\;\; a \mid a \leftarrow l\;] \;=\; [\; r\;\; b \mid b \leftarrow invl\;\; l\;]$$

Second example (*invl*)

Further calculation (back to R):

$$
\begin{array}{ll}
 & invl \cdot R^\star \subseteq R^\star \cdot invl \\
\equiv & \quad \{ \text{ shunting rule (14) } \} \\
 & R^\star \subseteq invl^\circ \cdot R^\star \cdot invl \\
\equiv & \quad \{ \text{ going pointwise (12, 13) } \} \\
 & \langle \forall\ l, r\ ::\ l\ R^\star r \Rightarrow (invl\ l) R^\star (invl\ r) \rangle
\end{array}
$$

An instance of this pointwise version of *invl*-FT will state that, for example, *invl* will respect element-wise orderings ($R := <$):

Second example (*invl*)

$$\mathit{length}\ l = \mathit{length}\ l' \wedge \langle \forall\ i\ :\ i \in \mathit{inds}\ l :\ (l\ i) < (r\ i) \rangle$$
$$\Downarrow$$
$$\mathit{length}(\mathit{invl}\ l) = \mathit{length}(\mathit{inv}\ l')$$
$$\wedge$$
$$\langle \forall\ j\ :\ j \in \mathit{inds}\ l :\ (\mathit{invl}\ l)j < (\mathit{invl}\ r)j \rangle$$

(Guess other instances.)

Third example: FT of *sort*

Our next example calculates the FT of

$$sort : a^\star \leftarrow a^\star \leftarrow (Bool \leftarrow (a \times a))$$

(the first parameter stands for the chosen ordering relation):

$$\begin{array}{ll} & sort(R_{(a^\star \leftarrow a^\star) \leftarrow (Bool \leftarrow (a \times a))})sort \\ \equiv & \quad \{ \ \ (1,\ 2,\ 3);\ \text{abbreviate } R_a := R \ \ \} \\ & sort((R^\star \leftarrow R^\star) \leftarrow (R_{Bool} \leftarrow (R \times R)))sort \\ \equiv & \quad \{ \ \ R_{t:=Bool} = id\ \text{(constant relator)} — \text{cf. exercise 7} \ \ \} \\ & sort((R^\star \leftarrow R^\star) \leftarrow (id \leftarrow (R \times R)))sort \end{array}$$

Third example: FT of *sort*

$$
\begin{array}{ll}
 & sort((R^\star \leftarrow R^\star) \leftarrow (id \leftarrow (R \times R)))sort \\
\equiv & \quad \{\ (4)\ \} \\
 & sort \cdot (id \leftarrow (R \times R)) \quad \subseteq \quad (R^\star \leftarrow R^\star) \cdot sort \\
\equiv & \quad \{\ \text{shunting (14)}\ \} \\
 & (id \leftarrow (R \times R)) \quad \subseteq \quad sort^\circ \cdot (R^\star \leftarrow R^\star) \cdot sort \\
\equiv & \quad \{\ \text{introduce variables } f \text{ and } g \text{ (12, 13)}\ \} \\
 & f(id \leftarrow (R \times R))g \quad \Rightarrow \quad (sort\ f)(R^\star \leftarrow R^\star)(sort\ g) \\
\equiv & \quad \{\ \text{(4) twice}\ \} \\
 & f \cdot (R \times R) \subseteq g \quad \Rightarrow \quad (sort\ f) \cdot R^\star \subseteq R^\star \cdot (sort\ g)
\end{array}
$$

Third example: FT of *sort*

Case $R := r$:

$$f \cdot (r \times r) = g \quad \Rightarrow \quad (sort\ f) \cdot r^\star = r^\star \cdot (sort\ g)$$

$$\equiv \qquad \{\ \text{introduce variables}\ \}$$

$$\left\langle \begin{array}{c} \forall\ a, b\ :: \\ f(r\ a, r\ b) = g(a, b) \end{array} \right\rangle \ \Rightarrow \ \left\langle \begin{array}{c} \forall\ l\ :: \\ (sort\ f)(r^\star\ l) = r^\star(sort\ g\ l) \end{array} \right\rangle$$

Denoting predicates f, g by infix orderings $\leq, \preceq$:

$$\left\langle \begin{array}{c} \forall\ a, b\ :: \\ r\ a \leq r\ b \equiv a \preceq b \end{array} \right\rangle \ \Rightarrow \ \left\langle \begin{array}{c} \forall\ l\ :: \\ sort\ (\leq)(r^\star\ l) = r^\star(sort\ (\preceq)\ l) \end{array} \right\rangle$$

That is, for r monotonic and injective,

$$sort\ (\leq)\ [\ r\ a \mid a \leftarrow l\ ]$$

is always the same list a

$$[\ r\ a \mid a \leftarrow sort\ (\preceq)\ l\ ]$$

Exercises

Exercise 1: Let C be a nonempty data domain and let and $c \in C$. Let $\underline{c}$ be the *"everywhere c"* function:

$$\begin{array}{lcl} \underline{c} & : & A \longrightarrow C \\ \underline{c}\, a & \triangleq & c \end{array} \tag{5}$$

Show that the free theorem of $\underline{c}$ reduces to

$$\langle \forall\ R\ ::\ R \subseteq \top \rangle \tag{6}$$

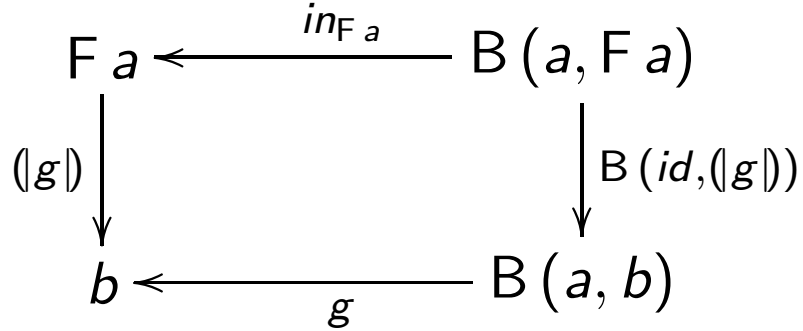
□

Exercise 2: Calculate the free theorem associated with the projections $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ and instantiate it to (a) functions; (b) coreflexives. Introduce variables and derive the corresponding pointwise expressions.

□

Fourth example: FT of $(\!|\_|\!)$

Recall the catamorphism (fold) combinator:



So $(\!|\_|\!)$ has generic type

$$(\!|\_|\!) : b \leftarrow \mathsf{F}\, a \leftarrow (b \leftarrow \mathsf{B}\,(a, b))$$

where $\mathsf{F}\, a \cong \mathsf{B}\,(a, \mathsf{F}\, a)$. Then $(\!|\_|\!)$-FT is

$$(\!|\_|\!) \cdot (R_b \leftarrow \mathsf{B}\,(R_a, R_b)) \quad \subseteq \quad (R_b \leftarrow \mathsf{F}\, R_a) \cdot (\!|\_|\!)$$

Fourth example: FT of $(\!|\_|\!)$

This unfolds into (R_a, R_b abbreviated to R, S):

$$(\!|\_|\!) \cdot (S \leftarrow \mathsf{B}\,(R,S)) \quad \subseteq \quad (S \leftarrow \mathsf{F}\,R) \cdot (\!|\_|\!)$$

$\equiv$ { shunting (14) }

$$(S \leftarrow \mathsf{B}\,(R,S)) \quad \subseteq \quad (\!|\_|\!)^{\circ}(S \leftarrow \mathsf{F}\,R) \cdot (\!|\_|\!)$$

$\equiv$ { introduce variables f and g (12, 13) }

$$f(S \leftarrow \mathsf{B}\,(R,S))g \quad \Rightarrow \quad (\!|f|\!)(S \leftarrow \mathsf{F}\,R)(\!|g|\!)$$

$\equiv$ { definition $f(R \leftarrow S)g \quad \equiv \quad f \cdot S \subseteq R \cdot g$ }

$$f \cdot \mathsf{B}\,(R,S) \subseteq S \cdot g \quad \Rightarrow \quad (\!|f|\!) \cdot \mathsf{F}\,R \subseteq S \cdot (\!|g|\!)$$

$(\!|\_|\!)$-FT corollaries

From

$$f \cdot \mathsf{B}\,(R, S) \subseteq S \cdot g \quad \Rightarrow \quad (\!|f|\!) \cdot \mathsf{F}\,R \subseteq S \cdot (\!|g|\!)$$

we can infer:

- $(\!|\_|\!)$-*fusion* $(R, S := id, s)$:

$$f \cdot \mathsf{B}\,(id, s) = s \cdot g \quad \Rightarrow \quad (\!|f|\!) = s \cdot (\!|g|\!)$$

- $(\!|\_|\!)$-*absorption* $(R, S := r, id)$:

$$f \cdot \mathsf{B}\,(r, id) = g \quad \Rightarrow \quad (\!|f|\!) \cdot \mathsf{F}\,r = (\!|g|\!)$$

  Substituting $g := f \cdot \mathsf{B}\,(r, id)$:

$$(\!|f|\!) \cdot \mathsf{F}\,r = (\!|f \cdot \mathsf{B}\,(r, id)|\!)$$

Exercises

Exercise 3: The following is a well-known Haskell function

```
filter ::  forall a.  (a -> Bool) -> [a] -> [a]
```

Calculate the free theorem associated with its type

$$filter : a^\star \leftarrow a^\star \leftarrow (Bool \leftarrow a))$$

and instantiate it to the case where all relations are functions.

□

Exercise 4: In many sorting problems, data are sorted according to a given *ranking* function which computes each datum's numeric rank (eg. students marks, credits, etc). In this context one may parameterize sorting with an extra parameter f ranking data into a fixed numeric datatype, eg. the integers: $serial : (a \to \mathbb{N}) \to a^\star \to a^\star$.
Calculate the FT of $serial$.

□

Exercises

Exercise 5: Choose arbitrary functions from Haskell's Prelude and calculate their FT.

□

Exercise 6: Let $\leq$ be a preorder and f be a function taking values on the carrier set of $\leq$.

1. Define the pointwise version of relation $\sqsubseteq \triangleq f^{\circ} \cdot \leq \cdot f$
2. Show that $\sqsubseteq$ is a preorder.
3. Show that $\sqsubseteq$ is not (in general) a total order even in the case $\leq$ is so.

□

Automatic generation of free theorems (Haskell)

See the interesting site in Janis Voigtlaender's home page:

http://linux.tcs.inf.tu-dresden.de/~voigt/ft

Relators in our calculational style are implemented in this automatic generator by structural *lifting*.

Background: relators

Relators [2] have to do with parametric datatyping: a parametric datatype G is said to be a relator wherever, given a relation from A to B, $\mathsf{G}\,R$ extends R to G-structures: it is a relation from $\mathsf{G}\,A$ to $\mathsf{G}\,B$

$$\begin{array}{ccc} A & \cdots\cdots & \mathsf{G}\,A \\ {\scriptstyle R}\downarrow & & \downarrow{\scriptstyle \mathsf{G}\,R} \\ B & \cdots\cdots & \mathsf{G}\,B \end{array} \tag{7}$$

which obeys the following properties:

$$\mathsf{G}\,id = id \tag{8}$$

$$\mathsf{G}\,(R \cdot S) = (\mathsf{G}\,R) \cdot (\mathsf{G}\,S) \tag{9}$$

$$\mathsf{G}\,(R^\circ) = (\mathsf{G}\,R)^\circ \tag{10}$$

and is monotonic:

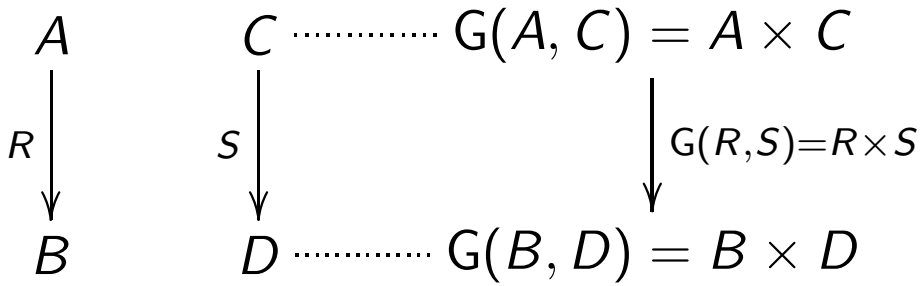
$$R \subseteq S \;\Rightarrow\; \mathsf{G}\,R \subseteq \mathsf{G}\,S \tag{11}$$

Exercises

Exercise 7: Show that the *identity* relator Id, which is such that $\mathsf{Id}\ R \quad = \quad R$ and the *constant* relator K (for a given data type K) which is such that $\mathsf{K}\ R \quad = \quad id_K$ are indeed relators.

□

Exercise 8: Show that product



is a (binary) relator.

□

Background

Going pointwise:

$$R \subseteq S \quad \equiv \quad \langle \forall\ b, a\ ::\ b\ R\ a \Rightarrow b\ S\ a \rangle \tag{12}$$

Function converses:

$$(f\ b) R (g\ a) \quad \equiv \quad b(f^{\circ} \cdot R \cdot g)a \tag{13}$$

Shunting rules:

$$f \cdot R \subseteq S \quad \equiv \quad R \subseteq f^{\circ} \cdot S \tag{14}$$

$$R \cdot f^{\circ} \subseteq S \quad \equiv \quad R \subseteq S \cdot f \tag{15}$$


K. Backhouse and R.C. Backhouse.
Safety of abstract interpretations for free, via logical relations and Galois connections.
SCP, 15(1–2):153–196, 2004.


R.C. Backhouse, P. de Bruin, P. Hoogendijk, G. Malcolm, T.S. Voermans, and J. van der Woude.
Polynomial relators.
In *AMAST'91*, Workshops in Computing, pages 303–362. Springer, 1992.


J.C. Reynolds.
Types, abstraction and parametric polymorphism.
Information Processing 83, pages 513–523, 1983.


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