Generic Properties of Datatypes

Roland Backhouse and Paul Hoogendijk

Generic Programming Summer School

Oxford, August 2002

Outline

- Theorems For Free
- Commuting Datatypes ("Zips")
- Relators, Fans and Membership
- Properties of Zips
- Conclusion

Parametric Polymorphism

Summary: parametric polymorphism is a verifiable form of (type) genericity.

Common Type = Common Properties

$$\mathsf{length} : \langle \forall \alpha :: \mathbb{N} \leftarrow \mathsf{List}.\alpha \rangle$$

For all types A and B and all functions f of type $A \leftarrow B$,

$$\mathsf{length}_A \circ \mathsf{List}.f \;=\; \mathsf{length}_B \quad .$$

Let sq denote the function that squares a number.

$$\mathsf{sq} \circ \mathsf{length} : \langle \forall \alpha :: \mathbb{N} \leftarrow \mathsf{List}.\alpha \rangle$$

$$(\mathsf{sq} \circ \mathsf{length}_A) \circ \mathsf{List}.f \;=\; \mathsf{sq} \circ \mathsf{length}_B \quad .$$

Suppose $\mathsf{copycat}$ appends a copy of a list to itself.

$$\mathsf{length} \circ \mathsf{copycat} : \langle \forall \alpha :: \mathbb{N} \leftarrow \mathsf{List}.\alpha \rangle$$

$$(\mathsf{length}_A \circ \mathsf{copycat}_A) \circ \mathsf{List}.f \;=\; \mathsf{length}_B \circ \mathsf{copycat}_B \quad .$$

Polymorphism

Consider the type expressions defined by the following grammar:

$$\mathsf{Exp} \quad ::= \quad \mathsf{Exp} \times \mathsf{Exp} \;\mid\; \mathsf{Exp} \leftarrow \mathsf{Exp} \;\mid\; \mathsf{Const} \;\mid\; \mathsf{Var} \quad .$$

Here, Const denotes a set of constant types, like $\mathbb{N}$ (the natural numbers) and $\mathbb{Z}$ (the integers). Var denotes a set of type variables. We use Greek letters to denote type variables.

A term t is said to have *polymorphic* type $\langle \forall \alpha :: \mathsf{T}.\alpha \rangle$, where T is a type expression parameterised by type variables α, if t assigns to each type A a value t_A of type $\mathsf{T}.A$.

Mapping Relations to Relations

Type expressions are extended to denote functions from relations to relations.

$$R \times S : A \times B \sim C \times D \quad \Leftarrow \quad R : A \sim C \ \wedge \ S : B \sim D$$

$$((a, b) \ , \ (c, d)) \in R \times S \quad \equiv \quad (a, c) \in R \ \wedge \ (b, d) \in S \quad .$$

$$R \leftarrow S : (A \leftarrow B) \sim (C \leftarrow D) \quad \Leftarrow \quad R : A \sim C \ \wedge \ S : B \sim D$$

$$(f, g) \in R \leftarrow S \quad \equiv \quad \langle \forall\, b,d :: (f.b \ , \ g.d) \in R \ \Leftarrow \ (b, d) \in S \rangle \quad .$$

The constant type A is read as the identity relation id_A on A.

$$(x \ , \ y) \in A \quad \equiv \quad x = y \quad .$$

Example

$$R\times R\leftarrow R \;:\; (A\times A \leftarrow A) \sim (B\times B \leftarrow B) \quad \Leftarrow \quad R : A\sim B$$

$$\begin{array}{ll}
 & (f, g) \;\in\; R\times R\leftarrow R \\
= & \qquad \{ \qquad \text{definition of } \leftarrow \text{ on relations} \quad \} \\
 & \langle \forall\, a,b \;::\; (f.a\,,\; g.b) \in R\times R \;\Leftarrow\; (a, b)\in R\rangle \\
= & \qquad \{ \qquad \text{definition of } \times \text{ on relations} \quad \} \\
 & \langle \forall\, a,b \;:: \\
 & \qquad\qquad (\mathsf{fst}.(f.a)\,,\; \mathsf{fst}.(g.b))\in R \;\wedge\; (\mathsf{snd}.(f.a)\,,\; \mathsf{snd}.(g.b))\in R \\
 & \qquad\quad \Leftarrow (a, b)\in R \\
 & \rangle
\end{array}$$

Example

$$\mathsf{id}_{\mathrm{Bool}} \leftarrow R \times R \;:\; (\mathrm{Bool} \leftarrow A \times A) \sim (\mathrm{Bool} \leftarrow B \times B) \quad \Leftarrow \quad R : A \sim B$$

$$(f, g) \;\in\; \mathsf{id}_{\mathrm{Bool}} \leftarrow R \times R$$

$$= \quad \{ \quad \text{definition of } \leftarrow \text{ and } \times \text{ on relations} \quad \}$$

$$\langle \forall\, a,a',b,b' \;::\; (f.(a, a')\,,\; g.(b, b')) \in \mathsf{id}_{\mathrm{Bool}} \quad \Leftarrow \quad (a, b) \in R \wedge (a', b') \in R \rangle$$

$$= \quad \{ \quad \text{definition of } \mathsf{id}_{\mathrm{Bool}} \quad \}$$

$$\langle \forall\, a,a',b,b' \;::\; f.(a, a') = g.(b, b') \quad \Leftarrow \quad (a, b) \in R \wedge (a', b') \in R \rangle$$

Parametric

A term t of polymorphic type $\langle \forall \alpha :: T.\alpha \rangle$ is said to be *parametrically polymorphic* if, for each instantiation of relations R to type variables, $(t_A , t_B) \in T.R$, where R has type $A \sim B$.

$$\mathsf{fst} : \langle \forall\, \alpha,\beta :: \alpha \leftarrow \alpha \times \beta \rangle$$

Suppose $R : A \sim B$ and $S : C \sim D$.

$$\begin{array}{ll}
 & (\mathsf{fst}_{A,C} , \mathsf{fst}_{B,D}) \;\in\; R \leftarrow R \times S \\
= & \quad \{ \quad \text{definition of } \leftarrow \text{ and } \times \text{ on relations} \quad \} \\
 & \langle \forall\, a,b,c,d :: (\mathsf{fst}_{A,C}.(a,c) , \mathsf{fst}_{B,D}.(b,d)) \in R \;\Leftarrow\; (a,b) \in R \wedge (c,d) \in S \rangle \\
= & \quad \{ \quad \text{definition of } \mathsf{fst} \quad \} \\
 & \langle \forall\, a,b,c,d :: (a,b) \in R \;\Leftarrow\; (a,b) \in R \wedge (c,d) \in S \rangle \\
= & \quad \{ \quad \text{calculus} \quad \} \\
 & \mathsf{true}
\end{array}$$

Ad Hoc Polymorphism

Suppose "$==$" denotes a polymorphic "equality" operator. That is,

$$== \;:\; \langle \forall \alpha :: \mathsf{Bool} \leftarrow \alpha \times \alpha \rangle$$

$==$ is parametric

$=$ { definition (R ranges over relations of type $\mathsf{A} \leftarrow \mathsf{B}$) }

$$\langle \forall \mathsf{R} :: (==_{\mathsf{A}} , ==_{\mathsf{B}}) \;\in\; \mathsf{id}_{\mathsf{Bool}} \leftarrow \mathsf{R} \times \mathsf{R} \rangle$$

$=$ { definition of $\leftarrow$ and $\times$ on relations, and of $\mathsf{id}_{\mathsf{Bool}}$ }

$$\langle \forall \mathsf{R} :: \langle \forall \mathtt{u},\mathtt{v},\mathtt{x},\mathtt{y} :: (\mathtt{u} ==_{\mathsf{A}} \mathtt{v}) = (\mathtt{x} ==_{\mathsf{B}} \mathtt{y}) \;\Leftarrow\; (\mathtt{u},\mathtt{x}) \in \mathsf{R} \wedge (\mathtt{v},\mathtt{y}) \in \mathsf{R} \rangle \rangle$$

$\Rightarrow$ { take R to be an arbitrary function $\mathtt{f}$

(so $(\mathtt{u},\mathtt{x}) \in \mathsf{R} \;\equiv\; \mathtt{u} = \mathtt{f}.\mathtt{x}$ and $(\mathtt{v},\mathtt{y}) \in \mathsf{R} \;\equiv\; \mathtt{v} = \mathtt{f}.\mathtt{y}$ }

$$\langle \forall \mathtt{f} :: \langle \forall \mathtt{x},\mathtt{y} :: (\mathtt{f}.\mathtt{x} ==_{\mathsf{A}} \mathtt{f}.\mathtt{y}) = (\mathtt{x} == \mathtt{y}) \rangle \rangle$$

Conclusion: all functions in the language of terms are injective, or "equality" is not both real equality and parametric.