Chapter 3

Refinements and ornaments

"datatypes" for inductive families

3.1 Datatype descriptions

$\mathsf{Desc} : (I : \mathsf{Set}) \to \mathsf{Set}_1$

$\mu : \{I : \mathsf{Set}\} \to \mathsf{Desc}\ I \to (I \to \mathsf{Set})$

$\mathbf{data}\ \mathsf{RDesc}\ (I : \mathsf{Set}) : \mathsf{Set}_1\ \mathbf{where}$
$\quad \mathsf{v} : (is : \mathsf{List}\ I) \to \mathsf{RDesc}\ I$
$\quad \sigma : (S : \mathsf{Set})\ (D : S \to \mathsf{RDesc}\ I) \to \mathsf{RDesc}\ I$

$[\![\_]\!] : \{I : \mathsf{Set}\} \to \mathsf{RDesc}\ I \to (I \to \mathsf{Set}) \to \mathsf{Set}$
$[\![\ \mathsf{v}\ is\ ]\!]\ X = \mathbb{M}\ is\ X$
$[\![\ \sigma\ S\ D\ ]\!]\ X = \Sigma[\ s : S\ ]\ [\![\ D\ s\ ]\!]\ X$

$\mathbb{M} : \{I : \mathsf{Set}\} \to \mathsf{List}\ I \to (I \to \mathsf{Set}) \to \mathsf{Set}$
$\mathbb{M}\ [\,]\ X = \top$
$\mathbb{M}\ (i :: is)\ X = X\ i \times \mathbb{M}\ is\ X$

3.2 Ornaments

Evolutionary remark (*ornaments as relations*). We define ornaments as relations between descriptions (indexed by an erasure function), whereas McBride's original ornaments [2011] are rebranded as ornamental descriptions. One obvious advantage of relational ornaments is that they can arise between *existing* descriptions, whereas ornamental descriptions always produce (definitionally) new descriptions at the more informative end. This also means that there can be multiple ornaments between a pair of descriptions. For example, consider the datatype

$$\textbf{indexfirst data}\ \mathsf{Square}\ (A : \mathsf{Set}) : \mathsf{Set}\ \textbf{where}$$
$$\mathsf{Square}\ A \ni \_,\_\ (x : A)\ (y : A)$$

Between the description of Square A and itself, we have the identity ornament

$$\sigma[\,x : A\,]\ \sigma[\,y : A\,]\ \mathsf{v}\ \mathsf{tt}$$

and the ornament

$$\Delta[\,x : A\,]\ \Delta[\,y : A\,]\ \nabla[\,y\,]\ \nabla[\,x\,]\ \mathsf{v}\ \mathsf{tt}$$

whose forgetful function swaps the fields x and y. The other advantage of relational ornaments is that they allow new datatypes to arise at the less informative end. For example, *coproduct of signatures* as used in, e.g., data types à la carte [Swierstra, 2008], can be implemented naturally with relational ornaments but not with ornamental descriptions. In more detail: Consider (a simplistic variation of) *tagged descriptions* [Chapman et al., 2010], which are descriptions that, for any index request, always respond with a constructor field first. A tagged description with index set $I : \mathsf{Set}$ thus consists of a family of types $C : I \to \mathsf{Set}$, where each $C\ i$ is the set of constructor tags for the index request $i : I$, and a family of subsequent response descriptions for each constructor tag.

$$\mathsf{TDesc} : \mathsf{Set} \to \mathsf{Set}_1$$
$$\mathsf{TDesc}\ I = \Sigma[\,C : I \to \mathsf{Set}\,]\ ((i : I) \to C\ i \to \mathsf{RDesc}\ I)$$

Tagged descriptions are decoded to ordinary descriptions by

$$\lfloor\_\rfloor_T : \{I : \mathsf{Set}\} \to \mathsf{TDesc}\ I \to \mathsf{Desc}\ I$$
$$\lfloor C, D \rfloor_T\ i = \sigma\ (C\ i)\ (D\ i)$$

We can then define binary coproduct of tagged descriptions, which sums up the corresponding constructor fields, as follows:

$$\_\oplus\_ : \{I : \mathsf{Set}\} \to \mathsf{TDesc}\ I \to \mathsf{TDesc}\ I \to \mathsf{TDesc}\ I$$
$$(C, D) \oplus (C', D') = (\lambda\, i \mapsto C\ i + C'\ i), (\lambda\, i \mapsto D\ i \triangledown D'\ i)$$

Now given two tagged descriptions $tD = (C, D)$ and $tD' = (C', D')$ of type TDesc I, there are two ornaments from $\lfloor tD \oplus tD' \rfloor_T$ to $\lfloor tD \rfloor_T$ and $\lfloor tD' \rfloor_T$

$$\mathit{inlOrn} : \mathsf{Orn}\ \mathit{id}\ \lfloor tD \oplus tD' \rfloor_T\ \lfloor tD \rfloor_T$$
$$\mathit{inlOrn}\ i = \Delta[\,c : C\ i\,]\ \nabla[\,\mathsf{inl}\ c\,]\ \mathit{idOrn}\ (D\ i\ c)$$
$$\mathit{inrOrn} : \mathsf{Orn}\ \mathit{id}\ \lfloor tD \oplus tD' \rfloor_T\ \lfloor tD' \rfloor_T$$
$$\mathit{inrOrn}\ i = \Delta[\,c' : C'\ i\,]\ \nabla[\,\mathsf{inr}\ c'\,]\ \mathit{idOrn}\ (D'\ i\ c')$$

whose forgetful functions perform suitable injection of constructor tags. Note that the synthesised new description $\lfloor tD \oplus tD' \rfloor_T$ is at the less informative end of *inlOrn* and *inrOrn*. (*End of evolutionary remark.*)

Example?

Bibliography

James Chapman, Pierre-Évariste Dagand, Conor McBride, and Peter Morris [2010]. The gentle art of levitation. In *International Conference on Functional Programming*, ICFP'10, pages 3–14. ACM. doi:10.1145/1863543.1863547.

Conor McBride [2011]. Ornamental algebras, algebraic ornaments. To appear in *Journal of Functional Programming*.

Wouter Swierstra [2008]. Data types à la carte. *Journal of Functional Programming*, 18(4):423–436. doi:10.1017/S0956796808006758.