Grins from my Ripley Cupboard

Conor McBride

May 24, 2009

This article is, I hope, unfinished. Mistakes: I've made many, but then again, a few are worth mentioning. This is one place where I propose to record them. It's a bit like that scene in *Alien: Resurrection* where Ripley finds the room where the unsuccessful attempts to reconstruct her are still alive and kept in tanks. Some of them are quite talkative.

1 Tit-for-tat Equality for Codata

I used to flog this one at Agda meetings. I believe Dr Abel may have implemented it at one stage. It's too good to be true. Here's a cutdown presentation of codata.

$$\begin{aligned}
\nu F &: \mathsf{Set} \\
\mathsf{out}_F &: \nu F \to F\,(\nu F) \\
\mathsf{in}_F &: F\,(\nu F) \to \nu F \\
\mathsf{coit}_F &: (S \to F\,S) \to S \to \nu F
\end{aligned}$$

$$\begin{aligned}
\mathsf{out}_F\,(\mathsf{in}_F\,t) &\rightsquigarrow t \\
\mathsf{out}_F\,(\mathsf{coit}_F\,f\,s) &\rightsquigarrow \mathsf{map}_F\,(\mathsf{coit}_F\,f)\,(f\,s)
\end{aligned}$$

Note that we don't dare add the computation rule

$$(\times) \qquad \mathsf{coit}_F\,f\,s \rightsquigarrow \mathsf{in}_F\,(\mathsf{map}_F\,(\mathsf{coit}_F\,f)\,(f\,s))$$

as it can take a while. But I thought I could at least make it hold definitionally, by unfolding one side of an equation when the other is already unfolded, adding this (and its mirror image)

$$\frac{\nu F \ni \mathsf{in}_F\,t \equiv \mathsf{in}_F\,(\mathsf{map}_F\,(\mathsf{coit}_F\,f)\,(f\,s))}{\nu F \ni \mathsf{in}_F\,t \equiv \mathsf{coit}_F\,f\,s}$$

to the equality testing algorithm. It should terminate — the amount of unfolding is constrained. There's no danger of restarting computation, because we're working with normal forms: if there's an out-in redex afterwards, there was an out-coit redex before.

And that, friends, is how I solved the Halting Problem. If $\equiv$ is transitive, as it should be, we're in trouble.[1] To see this, consider an automaton (a Turing machine perhaps) specified by a transition function of type $a : S \to 1 + S$ indicating whether the automaton halts or evolves to a new state. We may construct a coalgebra $\mathsf{tick}\,a$ for bitstreams in $\nu(2\times)$ with carrier $1 + S$ as follows

$$\begin{array}{lcllcl}
\mathsf{tick}\,a & : & & 1+S & \to & 2 \times (1+S) \\
\mathsf{tick}\,a & \mapsto & \lambda & (\mathsf{inl}\,()) & \mapsto & (\mathsf{true}, \mathsf{inl}\,()) \\
 & & \mid & (\mathsf{inr}\,s) & \mapsto & (\mathsf{true}, a\,s)
\end{array}$$

[1] This argument appears in *Let's see how things unfold*.

Now ask whether an automaton which has not yet halted ticks the same as one which has halted already:

$$\mathsf{coit}_{2\times}\ (\mathsf{tick}\ a)\ (\mathsf{inr}\ s)\ \equiv\ \mathsf{coit}_{2\times}\ (\mathsf{tick}\ a)\ (\mathsf{inl}\ ())\qquad ?$$

If the former ever halts, starting from s, then it is testably equal to a sufficiently large unfolding of form

$$\mathsf{in}_{2\times}\ (\mathsf{true}, \mathsf{in}_{2\times}\ (\mathsf{true}, \ldots \mathsf{in}_{2\times}\ (\mathsf{true}, \mathsf{coit}_{2\times}\ (\mathsf{tick}\ a)\ (\mathsf{inl}\ ()))\ldots))$$

which is clearly also an unfolding of the latter. By transitivity, a complete decision procedure for $\equiv$ must find the intermediate value for itself. So it's not just that my tit-for-tat trick doesn't decide this equality — it's undecidable by any trick. Hello, Ripley!

2 Functional Quotients with Aggressive Equality

Dr Oury and I sat in Cafézique, pondering Ripley 1, when we discovered another. Once upon a time, we rather liked the idea of implementing quotients by giving a function choosing a canonical representation.

$$\frac{\mathsf{Set} \ni X, Y \quad X \to Y \ni f}{\mathsf{Set} \ni X/_Y f} \qquad \frac{X \ni x}{X/_Y f \ni [x]}$$

The constructor 'class of', written $[-]$, packs up an element of the carrier set. Now, the eliminator allows unpacking, only if you promise to treat elements equally whenever f does.

$$\begin{array}{l}\mathsf{qunpack}(X, Y, f, c{:}X/_Y f, P{:}X/_Y f \to \mathsf{Set}, p{:}(x{:}X) \to P\ [x],\\ \qquad |\forall x, x'{:}X.\ f\ x = f\ x' \Rightarrow p\ x = p\ x'|) : P\ c\\ \mathsf{qunpack}\ X\ Y\ f\ [x]\ P\ p\ \_\ \mapsto\ p\ x\end{array}$$

One particular operation on the quotient is readily defined this way

$$\begin{array}{l}\hat{f}^{X,Y} : X/_Y f \to Y\\ \hat{f}^{X,Y}\ x\ \mapsto\ \mathsf{qunpack}\ X\ Y\ f\ x\ (\lambda\_ \mapsto Y)\ f\ (\lambda x, x', q \mapsto q)\end{array}$$

and the bit we really liked about functional quotients was the way we could use the function aggressively in the definitional equality.

$$\frac{Y \ni \hat{f}\ c \equiv \hat{f}\ c}{X/_Y f \ni c \equiv c'}$$

Note that by using $\hat{f}$, we don't even need to wait for the cs to be canonical — we can go right ahead and exploit any hand η-laws Y may possess, especially if Y is 1. Taking $!$ to be the function to 1 which always returns $\langle\rangle$, we get

$$\frac{\overline{1 \ni \langle\rangle \equiv \langle\rangle}}{2/_1 ! \ni [\mathsf{tt}] \equiv [\mathsf{ff}]}$$

We thought that was smashing! We hacked it up. We were very pleased, especially as a relation $R : X \to X \to \mathsf{Prop}$ can be seen as a function from X to an equivalence class.

Some time later, Dr O and I noticed that

$$(\times) \qquad \nu F/\mathsf{out}_F \ni [\mathsf{coit}_F\ f\ s] \equiv [\mathsf{in}_F\ (\mathsf{map}_F\ (\mathsf{coit}_F\ f)\ (f\ s))]$$

and became very worried. A slice of cake later, it was clear that it should have been much more obvious than that. There are such liars in the world, such cheats!

$$(\times) \qquad \mathsf{qunpack}\ 2\ 1\ !\ [\mathsf{tt}]\ (\lambda\_ \mapsto 2)\ \mathsf{id}\ \mathit{lie} \equiv \mathsf{tt} \equiv \mathsf{ff} \equiv \mathsf{qunpack}\ 2\ 1\ !\ [\mathsf{ff}]\ (\lambda\_ \mapsto 2)\ \mathsf{id}\ \mathit{lie}$$

Given a *lie* that all elements of 2 are equal propositionally, we can make them equal judgmentally. Hello, Ripley!

3 The Uniqueness of Magic

While the liars are about their business, let's have this Ripley. We love η-expansion because it gets rid of a lot of tedious bureaucratic nonsense. Guided by types, we can see where to apply it. But things can get out of hand. We already have

$$\frac{x{:}S \vdash T \ni f\ x \equiv g\ x}{(x{:}S) \to T \ni f \equiv g}$$

This rule tests whether functions coincide at an arbitrary and inert fresh variable, but one could try squeezing a bit more extensionality out of the system by comparing functions at any convenient finite bunch of patterns which happen to cover the domain. And what could be more convenient than the following?

$$\frac{}{(x{:}0) \to T \ni f \equiv g}$$

After all — no values, empty covering! But the liars know better: in particular, we discover that

$$(\times) \qquad \frac{0 \to 2 \ni \lambda\_ \mapsto \mathsf{tt} \equiv \lambda\_ \mapsto \mathsf{ff} \quad 0 \ni \mathit{lie} \equiv \mathit{lie}}{2 \ni (\lambda\_ \mapsto \mathsf{tt})\ \mathit{lie} \equiv \mathsf{tt} \equiv \mathsf{ff} \equiv (\lambda\_ \mapsto \mathsf{tt})\ \mathit{lie}}$$

Hello, Ripley!