Quotienting the Delay Monad by Weak Bisimilarity

James Chapman, Tarmo Uustalu and Niccolò Veltri

Institute of Cybernetics at Tallinn University of Technology, Estonia
{james|tarmo|niccolo}@cs.ioc.ee

The delay datatype was introduced by Capretta [2] as a means to incorporate general recursion to Martin-Löf type theory and it is useful in this setting for modeling non-terminating behaviours. This datatype is a (strong) monad and constitutes a constructive alternative to the maybe monad. For a given set X, each element of $\mathsf{D}\,X$ is a possibly infinite computation that returns a value of X, if it terminates. We define $\mathsf{D}\,X$ as a coinductive type by the rules

$$\frac{}{\mathsf{now}\,x : \mathsf{D}\,X} \qquad \frac{c : \mathsf{D}\,X}{\mathsf{later}\;c : \mathsf{D}\,X}$$

Weak bisimilarity is defined in terms of convergence. This binary relation between $\mathsf{D}\,X$ and X relates a terminating computation to its value and is inductively defined by the rules

$$\frac{}{\mathsf{now}\;x \downarrow x} \qquad \frac{c \downarrow x}{\mathsf{later}\;c \downarrow x}$$

Two computations are weakly R-bisimilar if they differ by a finite number of application of the constructor later, i.e., they either converge to R-related values or diverge. Weak R-bisimilarity is defined coinductively by the rules

$$\frac{c \downarrow x \quad xRx' \quad c' \downarrow x'}{c \approx_R c'} \qquad \frac{c \approx_R c'}{\mathsf{later}\;c \approx_R \mathsf{later}\;c'}$$

Just as D, one would expect $\hat{\mathsf{D}}$ defined by $\hat{\mathsf{D}}\,X = \mathsf{D}\,X/{\approx_X}$ (by writing $\approx_X$ we mean $\approx_{\equiv_X}$) to also be a (strong) monad. Morally, this ought to be the case, but there are issues with defining the multiplication, having to do with quotients in type theory.

One possible approach to quotients is to avoid them altogether as first-class entities, by working with setoids. This is unproblematic, we can define $\hat{\mathsf{D}}$ as a functor on **Setoid** by $\hat{\mathsf{D}}\,(X, R) = (\mathsf{D}\,X, \approx_R)$ and it is then also a (strong) monad. This route was taken by Capretta [2] and the same was done also by Benton et al. [1].

If we want to use quotients as sets, then we need to add them to type theory. We follow the approach of Hofmann [3]. We postulate, for any set X and equivalence relation R on X the existence of the following data: a set X/R, a constructor $\mathsf{abs} : X \to X/R$ with a proof $\mathsf{sound} : \Pi x, x' : X.\, \Pi p : xRx'.\, \mathsf{abs}\,x \equiv \mathsf{abs}\;x'$, for any predicate P on X/R, a dependent eliminator $\mathsf{lift} : \Pi f : \Pi x : X.\, P\,(\mathsf{abs}\,x).\, \mathsf{compat}\,f \to \Pi q : X/R.\, P\,q$ where $\mathsf{compat}\,f = \Pi x, x' : X.$ $\Pi p : xRx'.\, \mathsf{subst}\,P\,(\mathsf{sound}\,p)(f\,x) \equiv f\,x'$ and a proof of the beta-rule $\mathsf{lift}_\beta : \mathsf{lift}\,f\,p\,(\mathsf{abs}\,x) \equiv f\,x$. A special case of quotients are so-called squash types, which are quotients by the total relation. We write $\|X\|$ for $X/\top$. We call a set X a proposition, if $\Pi x, x' : X.\, x \equiv x'$. Squash types are propositions.

For $\hat{\mathsf{D}}$ as a functor on **Set** to be a monad, we have to have a multiplication $\mu_X : \hat{\mathsf{D}}\,(\hat{\mathsf{D}}\,X) \to \hat{\mathsf{D}}\,X$. For this, a crucial element needed is a function $\psi : \mathsf{D}\,(\mathsf{D}\,X/{\approx_X}) \to \mathsf{D}\,(\mathsf{D}\,X)/\mathsf{D}\,{\approx_X}$. This seems difficult, but there is a canonical function $\theta = \mathsf{lift}\,(\mathsf{D}\,\mathsf{abs})\,(\ldots) : \mathsf{D}\,(\mathsf{D}\,X)/\mathsf{D}\,{\approx_X} \to$

$\mathsf{D}\,(\mathsf{D}\,X/{\approx_X})$ in the opposite direction. We say that a function $f : X \to Y$ between two sets X and Y is surjective, if $\Pi x : X.\, \|\Sigma y : Y.\, f\,x \equiv y\|$. The function θ is generally not surjective. But it can be proven surjective assuming countable choice: for all sets Y and relations P between $\mathbb{N}$ and Y, we assume $\Pi n : \mathbb{N}.\, \|\Sigma y : Y.\, P\,n\,y\| \to \|\Sigma f : \mathbb{N} \to Y.\, \Pi n : \mathbb{N}.\, P\,n\,(f\,n)\|$. To actually have an inverse ψ for θ we need more. In addition to countable choice, we need the quotient X/R to be weakly effective in the sense that $\Pi x, x' : X.\, \mathsf{abs}\,x \equiv \mathsf{abs}\,x' \to \|xRx'\|$. This can be proved by additionally assuming that equivalent propositions are equal: if two sets X, Y are propositions, then $X \leftrightarrow Y \to X \equiv Y$.

Assuming countable choice and equality of equivalent propositions (in addition to the assumptions of equality of extensionally equal functions and equality of (strongly) bisimilar coinductive data), multiplication $\mu_X : \hat{\mathsf{D}}\,(\hat{\mathsf{D}}\,X) \to \hat{\mathsf{D}}\,X$ can be defined and satisfies the monad laws.

One should be careful to not assume too much, as it is easy to arrive at provability of non-constructive principles such as excluded middle. For example, assuming effectiveness for all quotients together with equality of equivalent propositions gives excluded middle. Also, it may be tempting to postulate surjectivity of the canonical function $(X \to Y)/(X \to R) \to X \to Y/R$ which is there for all X, Y and R. But this is logically equivalent to weak existence of a section for abs for all quotients: for all X and R, we have $\|\Sigma f : X/R \to X.\, \Pi\, q : X/R.\, \mathsf{abs}\,(f\,q) \equiv q\|$. And that in turn is logically equivalent to the full axiom of choice: for all sets X, Y and relations P between X and Y, it holds that $\Pi x : X.\, \|\Sigma y : Y.\, P\,x\,y\| \to \|\Sigma f : X \to Y.\, \Pi x : X.\, P\,x\,(f\,x)\|$.

A yet different approach is not to quotient objects $\mathsf{D}\,X$ by the relations $\approx_X$ and aim at $\hat{\mathsf{D}}$ being a strong monad, but instead quotient homsets $\mathbf{Kl}(\mathsf{D})(X, Y)$, i.e., function spaces $X \to \mathsf{D}\,Y$, by the pointwise extension $X \to {\approx_Y}$. This gives an arrow on **Set** in the sense of Hughes.

We have formalized our development in Agda.

Acknowledgement This research was supported by the ERDF funded Estonian ICT national programme project “Coinduction”, the Estonian Science Foundation grants No. 9219 and 9475 and the Estonian Ministry of Education and Research institutional research grant IUT33-13.

References

[1] N. Benton, A. Kennedy, C. Varming. Some domain theory and denotational semantics in Coq. In S. Berghofer, T. Nipkow, C. Urban, M. Wenzel, eds., *Proc. of 22nd Int. Conf. on Theorem Proving in Higher Order Logics, TPHOLs 2009*, v. 5674 of *Lect. Notes in Comput. Sci.*, pp. 115–130. Springer, 2009.

[2] V. Capretta. General recursion via coinductive types. *Log. Methods in Comput. Sci.*, 1(2:1), 2005.

[3] M. Hofmann. *Extensional constructs in intensional type theory*, CPHS/BCS Distinguished Dissertations. Springer, 1997.