Partiality, Revisited

Thorsten Altenkirch[1*] and Nils Anders Danielsson[2†]

[1] School of Computer Science, University of Nottingham, UK
[2] University of Gothenburg and Chalmers University of Technology, Sweden

This note presents work in progress on representing partial computations in type theory. We define a monad $-_\perp : \mathbf{Set} \to \mathbf{Set}$.[1] This monad is a pointed ω-CPO, and can be used to solve recursive equations: given an ω-continuous function $f : (A \to B_\perp) \to (A \to B_\perp)$ we can construct a function $\mathit{fix}(f) : A \to B_\perp$ satisfying $\mathit{fix}(f) = f(\mathit{fix}(f))$.

Capretta (2005) gave a similar construction using setoids, but setoids are arguably awkward to work with, so our aim is to avoid them. A natural idea is to use quotient types, as suggested by Capretta et al. (2005) and worked out in more detail by Chapman et al. (2015): First define the delay monad $\mathit{Delay} : \mathbf{Set} \to \mathbf{Set}$ coinductively by the constructors $\eta : A \to \mathit{Delay}(A)$ and $\mathit{later} : \mathit{Delay}(A) \to \mathit{Delay}(A)$. Then one can, for instance, define the least value $\perp : \mathit{Delay}(A)$ as the solution to the guarded equation $\perp = \mathit{later}(\perp)$. However, the delay monad distinguishes computations that proceed with different speed (different number of *later* constructors). This can be remedied by quotienting the delay monad by an equivalence relation corresponding to weak bisimilarity. There are a number of ways to define this relation, for instance the following one (Capretta, 2005): First define a relation $\downarrow : A_\perp \to A \to \mathbf{Prop}$, where $p \downarrow a$ means that p terminates with the value a, inductively by $\eta(a) \downarrow a$ and $p \downarrow a \to \mathit{later}(p) \downarrow a$. Weak bisimilarity $\approx : \mathit{Delay}(A) \to \mathit{Delay}(A) \to \mathbf{Prop}$ can then be defined by $p \approx q := \Pi_{a:A}(p \downarrow a \leftrightarrow q \downarrow a)$, and $A_\perp$ as the quotient type $\mathit{Delay}(A)/{\approx}$. However, Chapman et al. (2015) noticed a potential problem with this construction: it seems to be hard or impossible to prove that $-_\perp$ is a monad in (some variant of) type theory. They also showed that $-_\perp$ really is a monad under the assumptions of countable choice and propositional extensionality.

Our aim is to show that a partiality monad can be constructed without having to introduce countable choice. We observe, as did Chapman et al. (2015), that the situation is reminiscent of the situation with the Cauchy reals. If the Cauchy reals are defined as a quotient, then it is for instance impossible to prove a specific form of the statement that every Cauchy sequence of Cauchy reals has a limit using $\mathsf{IZF}_{\mathrm{Ref}}$, a constructive set theory without countable choice (Lubarsky, 2007). The Univalent Foundations Program (2013, Section 11.3) circumvents this problem by defining the Cauchy reals as "the free complete metric space generated by $\mathbb{Q}$", using a higher inductive-inductive type that mutually defines the real numbers—including an inclusion of rational numbers and a limit construction—and a certain relation. We note that this definition of the real numbers forms a set and that the relation is propositional, so the higher inductive-inductive type used is a *quotient inductive-inductive type* (Altenkirch and Kaposi, 2016).

Using a similar approach we mutually define $A_\perp : \mathbf{Set}$ and $\sqsubseteq : A_\perp \to A_\perp \to \mathbf{Prop}$, where $\sqsubseteq$ represents information ordering, as a quotient inductive-inductive type:

$$\begin{array}{rcl} \perp & : & A_\perp \\ \eta & : & A \to A_\perp \\ \bigsqcup & : & \Pi_{f:\mathbb{N}\to A_\perp}(\Pi_{n:\mathbb{N}} f(n) \sqsubseteq f(n+1)) \to A_\perp \end{array}$$

*Supported by EPSRC grant EP/M016951/1 and USAF grant FA9550-16-1-0029.
†Supported by a grant from the Swedish Research Council (621-2013-4879).

[1] For the purpose of this note we define **Set** (**Prop**) as the type of types in the first universe that are sets (propositions) in the sense of the Univalent Foundations Program (2013).

To improve readability we present the constructors for $\sqsubseteq$ using inference rules:[2]

$$\frac{}{d \sqsubseteq d} \qquad \frac{}{\bot \sqsubseteq d} \qquad \frac{\bigsqcup(f,p) \sqsubseteq d}{\Pi_{n:\mathbb{N}} f(n) \sqsubseteq d} \qquad \frac{\Pi_{n:\mathbb{N}} f(n) \sqsubseteq d}{\bigsqcup(f,p) \sqsubseteq d}$$

These rules say that the relation is reflexive with $\bot$ as the least element, and that $\bigsqcup(f,p)$ has a given upper bound iff the sequence f has the same upper bound. We add two constructors for equality, one which turns $\sqsubseteq$ into a partial order (transitivity can be proved), and one which makes $\sqsubseteq$ propositional (one can then prove that $A_\bot$ is a set):

$$\frac{d \sqsubseteq d' \qquad d' \sqsubseteq d}{d = d'} \qquad\qquad \frac{p, q : d \sqsubseteq d'}{p = q}$$

We have showed—without assuming countable choice—that $-_\bot$ is a monad. The construction and this proof have been implemented in Agda,[3] using an experimental rewriting feature developed by Andreas Abel and Jesper Cockx to support higher inductive-inductive types.

However, we have not yet verified that this monad is correctly defined. Together with Paolo Capriotti and Nicolai Kraus we have ruled out the risk that the monad is trivial by proving that $\bot \neq \eta(x)$ (in Agda, using the univalence axiom), but we have not yet established a firm connection between the construction presented here and the quotiented delay monad mentioned above.

Further experiments might reveal whether our construction provides a good basis for the development of a theory of partial functions within type theory.

References

Thorsten Altenkirch and Ambrus Kaposi. Type theory in type theory using quotient inductive types. In *POPL'16, Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 18–29, 2016. doi:10.1145/2837614.2837638.

Venanzio Capretta. General recursion via coinductive types. *Logical Methods in Computer Science*, 1(2):1–28, 2005. doi:10.2168/LMCS-1(2:1)2005.

Venanzio Capretta, Thorsten Altenkirch, and Tarmo Uustalu. Partiality is an effect. Slides for a talk given by Uustalu at the 22nd meeting of IFIP Working Group 2.8, 2005.

James Chapman, Tarmo Uustalu, and Niccolò Veltri. Quotienting the delay monad by weak bisimilarity. In *Theoretical Aspects of Computing – ICTAC 2015*, volume 9399 of *LNCS*, pages 110–125. 2015. doi:10.1007/978-3-319-25150-9_8.

Robert S. Lubarsky. On the Cauchy completeness of the constructive Cauchy reals. *Mathematical Logic Quarterly*, 53(4–5):396–414, 2007. doi:10.1002/malq.200710007.

The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[2] Because $\sqsubseteq$ is propositional we omit the constructor names.

[3] With the K rule turned off, and with minor differences from this presentation. The source code can at the time of writing be found via Danielsson's personal web page (`http://www.cse.chalmers.se/~nad/`).