Observational Equivalence for Behavioural Differential Equations

Henning Basold[1,2], Helle H. Hansen[1,2]

[1] Radboud University Nijmegen
[2] CWI Amsterdam

> *"We cannot create observers by saying 'observe', but by giving them the power and the means for this observation and these means are procured through education of the senses."*
>
> — Maria Montessori

Recently, Abel and others [2] have proposed a type system, we denote it by F_μ^{cop} here, in which coinductive types are programmed by means of observations. In a follow-up paper, Abel and Pientka [1] extended F_μ^{cop} with, amongst other things, dependent types to F_ω^{cop}, and showed how to check termination and productivity of programs in a modular way. Both systems allow mutually recursive terms, which we interpret as Behavioural Differential Equations (BDE) [6]. A natural question is whether two such BDEs give rise to the same behaviour. This question is related to differences between intensional and extensional Type Theory [3].

For example, we can define types representing natural numbers and streams by $\mathrm{Nat} := \mu X.(1 + X)$ and $\mathrm{Str}\, A := \nu X.(A \times X)$, respectively. On these types, we can define the stream over Nat that is constantly 1 in several ways, two of which are shown below.

$$\begin{aligned} &\mathit{ones} : \mathrm{Str}\,\mathrm{Nat} \\ &\mathrm{hd}\ \mathit{ones} = 1 \\ &\mathrm{tl}\ \mathit{ones}\ \ = \mathit{ones} \end{aligned} \qquad \begin{aligned} &\mathit{ones}_1, \mathit{ones}_2 : \mathrm{Str}\,\mathrm{Nat} \\ &\mathrm{hd}\ \mathit{ones}_1 = 1 \\ &\mathrm{tl}\ \mathit{ones}_1\ \ = \mathit{ones}_2 \\ &\mathrm{hd}\ \mathit{ones}_2 = 1 \\ &\mathrm{tl}\ \mathit{ones}_2\ \ = \mathit{ones}_1 \end{aligned}$$

These definitions are syntactically different, but if we take an experimental perspective, then they should coincide, since we cannot distinguish them by making *observations*. Here, an observation evaluates a term and makes further observations on the result. For example, for streams we define the set of observations to be $\mathrm{Obs}(\mathrm{Str}\, A) = \{\lambda s.O(\mathrm{hd}(\mathrm{tl}^n\ s)) \mid n \in \mathbb{N}, O \in \mathrm{Obs}(A)\}$. We say that two terms $t_1, t_2 : A$ are *observationally equivalent*, written $t_1 \equiv_{\mathrm{obs}} t_2$, if $O\, t_1 \equiv O\, t_2$ for all observations $O \in \mathrm{Obs}(A)$. Here, $\equiv$ is the equivalence induced by the reduction relation given in [2]. In the example we have $\mathit{ones} \equiv_{\mathrm{obs}} \mathit{ones}_1 \equiv_{\mathrm{obs}} \mathit{ones}_2$.

This notion of observational equivalence turns out to have some good properties. It subsumes extensionality in the form of η-equivalence. We can also use it as an equivalence relation to form a category that has closed types as objects and equivalence classes of terms with type $A \to B$ as arrows. If we additionally

restrict to *observationally normalising* terms, i.e. terms normalising under every observation, we find that this category has products and coproducts. Moreover, we can interpret types with free variables in positive position as functors on this category. For types in which variables occur only in *strictly* positive position, the induced functors have initial algebras and final coalgebras.

The observations on a type A induce a natural topology on the set of terms $\mathcal{T}(A)$ inhabiting A. Under this topology, terms $A \to B$ are continuous functions $\mathcal{T}(A) \to \mathcal{T}(B)$ by application. Moreover, this topology coincides with known ones. For example, the set $\mathcal{T}(\mathrm{Str\,Nat})$ carries the usual product topology, which comes from the approximation metric on streams. Note, that we consider continuity as the fundamental notion of computability in this setting, in the spirit of the Brouwerian school of intuitionism [5,4].

Clearly, the notion of observational equivalence is rather semantic in nature. So we would like to have means of formulating and proving, in a verifiable way, that two terms t_1, t_2 are observationally equivalent. To this end, we introduce an equivalence relation $t_1 \sim t_2$ between terms, which is supposed to represent observational equivalence (the symbol looks intentionally like bisimilarity), and a proof system for $\sim$. This is analogous to the development by Altenkirch et al. [3], where an equality is introduced as dependent type together with rules for constructing elements of that type. We expect our proof system to be sound and complete with respect to observational equivalence.

A pleasant fact about the proof system is that proofs are written in the same BDE syntax as the programs are. In fact, the programs can be seen as the non-dependent fragment of F_ω^{cop}, and $\sim$ becomes a dependent type. The techniques from [1] can now be applied to the proofs, as well. Thus we have that proof checking for $\sim$ is decidable. For the problem of finding proofs, we would like to investigate notions like "bisimulation up-to" techniques.

We hope, that this work can contribute to the development of Observational Type Theory [3], to solve problems related to extensionality in Type Theory.

References

1. A. Abel and B. Pientka. Wellfounded recursion with copatterns: a unified approach to termination and productivity. In *ICFP*, pages 185–196, 2013.
2. A. Abel, B. Pientka, D. Thibodeau, and A. Setzer. Copatterns: Programming Infinite Structures by Observations. In *Proc. of POPL '13*, page 27–38. ACM, 2013.
3. T. Altenkirch, C. McBride, and W. Swierstra. Observational Equality, Now! In *Proc. of PLPV '07*, page 57–68. ACM, 2007.
4. M. Escardó. Synthetic Topology: of Data Types and Classical Spaces. *ENTCS*, 87:21–156, Nov. 2004.
5. N. Ghani, P. Hancock, and D. Pattinson. Continuous Functions on Final Coalgebras. *ENTCS*, 249:3–18, 2009.
6. J. J. M. M. Rutten. Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *TCS*, 308(1-3):1–53, 2003.