Finitary-based Domain Theory in Coq: An Early Report

Moez A. AbdelGawad
moez@cs.rice.edu
SRTA-City, Alexandria, Egypt

April 27, 2015

In his "Lectures on a Mathematical Theory of Computation" [5], Dana Scott formulated domains in terms of neighborhood systems. Later, Scott favored a formulation in terms of information systems [6] but has not rewritten his lectures notes. Cartwright and Parsons later revised Scott's lecture notes to reflect a formulation of domains in terms of 'finitary basis' [3], where a finitary basis is an information system that is closed under least upper bounds on finite consistent subsets. Finitary basis have the desirable property that every finite computable object is represented by a single basis element instead of a set of elements.

We recently started an effort to formalize Cartwright and Parsons' eight-chapters monograph in Coq [2]. We finished formalizing the first chapter of the monograph. Our goal is to publish the full monograph formalization online when the monograph itself gets published as a primer on domain theory in the near future, as we plan for. Here we present an early brief report on the progress of our effort, where we focus on reporting the initial ease we found when starting the formalization but that was followed by a difficulty of proceeding at the same initial pace. We believe the initial ease is due to Coq's "built-in" support for set theory and partial orders (posets). The "formalization resistance" we later found as we dug into domain theory is due to the "thickness" of layers upon layers of definitions of domain theory (which is typical of any mathematical discipline), combined with Coq's unstructured proof syntax and with proof states being implicit and not explicitly stated in Coq proofs. In our experience, these factors have made proofs in Cartwright and Parsons' monograph that are relatively simple become lengthy and harder to grasp when formalized in Coq.

In particular, the initial ease we found in writing domain theory definitions and constructing proofs in Coq was due to the support for set theory in Coq (via type `Ensemble`) and for order theory/posets (via type `PO`), which made the first few initial definitions and proofs in our formalization straightforward.

However, later we faced hardship. Examples of proofs we found unnecessarily hard in Coq are the proofs that 'the singleton set containing bottom ($\{\bot\}$) is the bottom (i.e., smallest) ideal', that 'a finite subset of a union set has a finite

covering set', and that 'the union of a directed set of ideals is an ideal'. The three proofs were over fifty lines of Coq code, which we believe to be unnecessarily long since we used as much intermediate lemmas as possible as a means for shortening and structuring these proofs. Additionally, the syntax of Coq proofs (as an almost linear sequence of commands, each of whose effects is to change an implicit proof state) did not make the three Coq proofs of these theorems immediately reveal the main ideas of the proofs.

A consequence of our experience with Coq is that our domain theory formalization effort has unfortunately slowed down. Even though there is a chance we may keep using Coq as the tool of choice to express our formalization (given the time and effort we already invested in it), but we are also considering switching over to using other proof assistants such as Isabelle [4] (thus restarting our formalization effort almost from scratch), or even giving up our formalization effort altogether. As a consequence of our experience with Coq we are also considering the possibility of developing our own (vastly less powerful, yet vastly more user-friendly) Proof Designer-based proof assistant [1].

References

[1] Moez A. AbdelGawad. Set theory for the (smart) masses. Technical report, (Submitted to SETS 2015), 2015.

[2] Yves Bertot and Pierre Casteran. *Interactive Theorem Proving and Program Development Coq'Art: The Calculus of Inductive Constructions*. Springer, 2004.

[3] Robert Cartwright and Rebecca Parsons. Domain theory: An introduction. http://www.cs.rice.edu/~javaplt/411/15-spring/Readings/domains.pdf (Soon to be available as an arXiv preprint), 1988. Monograph (based on earlier notes by Dana Scott).

[4] Larry Paulson, Tobias Nipkow, and Makarius Wenzel. Isabelle. http://isabelle.in.tum.de, 2015.

[5] Dana S. Scott. Lectures on a mathematical theory of computation. Technical Monograph PRG-19, Oxford University Computing Laboratory, May 1981.

[6] Dana S. Scott. Domains for denotational semantics. Technical report, Computer Science Department, Carnegie Mellon University, 1983.