C00 Exercise 4: write

Ramana Kumar (rk436)

November 8, 2010

PhD Proposal: Types Above Set Theory

Most interactive theorem provers produce theorems in some type theory. The generally accepted foundation for mathematics, however, is set theory. Advantages of type theories include: type checking and inference, functions as a primitive notion, and shorter statements when set memberships are implied by the types. But there are many type theories, ranging from higher-order logic to the calculus of constructions, which vary both philosophically and practically. Furthermore, type systems make certain constructions difficult, especially in abstract algebra and category theory. Gordon [4] suggested approaches for getting the best of both worlds, the standardness and expressivity of set theory alongside the usability afforded by types.

Gordon identified two ways to combine type theory and set theory: (A) starting from type theory, axiomatically define a type of sets; or, (B) starting from set theory, add a layer of types as a derived language. He sketched both approaches [3], and suggested that (B), though harder, would yield more benefits. Krauss and Schropp [5] pursued (A) further, developing an automatic proof-preserving translation from Isabelle/HOL to Isabelle/ZF. My proposal is to develop (B), to add a layer of types for easier proving in an untyped set theory.

My research question, then, is whether any salient points in the design space of theorem proving tools that implement type theory above set theory are workable in practice. Gordon's shallow embedding of HOL in ZF [3] is a good starting point. Workability can follow from a solid theoretical foundation, but is ultimately decided by case studies. Agerholm [1] formalised a domain theory construction that would be a good case study since it cannot be done in type theory alone. Rethinking foundations, Ganesalingam [2] provides seeds for a third approach (C): an underlying language so bare that both set and type theory can be implemented above it.

References

[1] Sten Agerholm. Formalising a model of the λ-calculus in HOL-ST. Technical Report 354, University of Cambridge Computer Laboratory, 1994.

[2] Mohan Ganesalingam. A language for mathematics. `http://people.pwf.cam.ac.uk/mg262/`, 2009.

[3] Michael J. C. Gordon. Merging HOL with set theory: preliminary experiments. Technical Report 353, University of Cambridge Computer Laboratory, 1994.

[4] Michael J. C. Gordon. Set theory, higher order logic or both? In Joakim von Wright, Jim Grundy, and John Harrison, editors, *TPHOLs*, volume 1125 of *Lecture Notes in Computer Science*, pages 191–201. Springer, 1996.

[5] Alexander Krauss and Andreas Schropp. A mechanized translation from higher-order logic to set theory. In Matt Kaufmann and Lawrence C. Paulson, editors, *ITP*, volume 6172 of *Lecture Notes in Computer Science*, pages 323–338. Springer, 2010.