Mechanizing a Decision Procedure for Coproduct Equality

Arbob Ahmad Daniel R. Licata Robert Harper
Carnegie Mellon University

In many applications of λ-calculi, it is essential to compare λ-terms for equality. For example, type checking a dependently typed programming language requires deciding equality of its terms and types. Whereas it is relatively simple to decide $\beta\eta$-equality for function and product types, deciding the full $\beta\eta$-equational theory for coproduct (disjoint sum) types is less straightforward. The coarsest η-rule for sum types, which is the uniqueness condition of the categorical universal mapping property for coproducts, is the following:

$$[E/x]E' \equiv \texttt{case}(E, x_1.[\texttt{inl}\ x_1/x]E', x_2.[\texttt{inr}\ x_2/x]E') \qquad \text{if } E : A_1 + A_2$$

The non-local nature of this rule makes it difficult to give algorithmic formulations of this equational theory. Several decision procedures for coproduct equality have been studied, using techniques such as rewriting [Ghani, 1995, Lindley, 2007] and normalization by evaluation [Altenkirch et al., 2001]. However, these decision procedures and their correctness proofs are quite complex and subtle. Because coproduct equality is both theoretically and practically interesting, and because it is a difficult problem, we believe that it is an excellent candidate for a mechanization.

In this work, we are giving a new decision procedure for coproduct equality and mechanizing this algorithm and its proof of correctness in Twelf. Our algorithm is an application of the canonical-forms methodology invented by Watkins et al. [2002]. Specifically, traditional simply typed λ-calculus (STLC) terms are translated into an internal canonical forms language, with the property that many equal STLC terms are mapped to the same internal language term. The remaining equalities are implemented by a congruence relation on canonical forms, similar to the permuting conversions in CLF [Watkins et al., 2002]. We have chosen to develop this new algorithm, rather than to start from existing work, both because the canonical forms methodology extends well to dependent types, which is crucial to many of our applications, and because this style of algorithm is amenable to mechanization in Twelf.

Both our algorithm and its Twelf formalization are works in progress. The above coproduct η-rule can be decomposed into individual rules stating that `case` commutes with each syntactic form of the language. On paper, we have validated most of these equalities—we have shown that case commutes with the principal premise of each elimination form and with the premises of each introduction form. We are currently investigating the remaining equalities. In Twelf, we have represented the STLC and the canonical forms language, mechanized the translation of terms, proved that every STLC term has a unique canonical form, and formalized some of the aforementioned equalities. Our mechanization is currently a medium-size development, consisting of approximately 10,000 lines of Twelf code. It was written primarily by the first author, a CMU undergraduate student who had given paper proofs of the mechanized theorems but who had no previous Twelf experience. At the workshop, we will report on both our algorithm for coproduct equality and the process of mechanizing it in Twelf.

References

T. Altenkirch, P. Dybjer, M. Hofmann, and P. Scott. Normalization by evaluation for typed λ-calculus with coproducts. In *IEEE Symposium on Logic in Computer Science*, pages 203–210. IEEE Press, 2001.

N. Ghani. $\beta\eta$-equality for coproducts. In *International Conference on Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 171–185. Springer-Verlag, 1995.

S. Lindley. Extension rewriting with sums. In *International Conference on Typed Lambda Calculi and Applications*, 2007.

K. Watkins, I. Cervesato, F. Pfenning, and D. Walker. A concurrent logical framework I: Judgments and properties. Technical Report CMU-CS-02-101, Department of Computer Science, Carnegie Mellon University, 2002. Revised May 2003.