ON THE INDISPENSABILITY OF BAR RECURSION

(ABSTRACT)

DANKO ILIK

Ever since Kreisel and Spector’s [1, 2] work on bar recursion, we know how to reduce the consistency of classical Analysis (PA+AC) to computation based on general recursion (the primitive recursive System T extended by a form of bar recursion). The method has been also applied to extracting bounds from actual analytic proofs, most extensively in the work on proof mining [3]. The approach of Krivine [4, 5], which relies on proof-theoretically-novel concepts from the theory of programming languages, also seems to take for granted that there is some kind of infinite search present in the interpretation of proofs of Analysis as programs.

Yet, as Schwichtenberg has shown in 1979 [6], when restricted to types 0 and 1, bar recursion cannot be used to define non-primitive recursive functions. Since a previous analysis of Kreisel [1, §12.2] shows that bar recursion at those low types is actually enough to give a computational interpretation of statements of the form $\exists\alpha^N \forall^Nx^N A_0(\alpha, x)$, $A_0$-quantifier-free, and this class of formulas appears to be the largest class we can possibly hope to interpret computationally, if we want to interpret all formulas of a class, it follows that one should not need more than System T, i.e. higher-type primitive recursion, for the computational interpretation of the $\Sigma_2$-fragment of Analysis.

It has however not be known how to dispense with bar recursion and obtain directly realizing programs in System T. In this talk, I will present recent work of mine on this topic [7] (under review). We will use a technique from the theory of programming languages, type-directed partial evaluation, which has links to Berger and Schwichtenberg’s normalization-by-evaluation program. The key observation will be that one should not try to realize the Axiom of Choice uniformly, i.e. once and for all, by a single program that will work regardless of the concrete proofs involved. We can rather always, but not in the same manner, partially evaluate every classical proof of a $\Sigma_2$-formula using the axiom in order to extract the computational core/witnesses involved. The technical difficulty of this approach is to give a normalization-by-evaluation program not only for lambda calculus, but also in presence of higher-type primitive recursion and control operators.

This work extends previous investigations by Nakata and me [8, 9] on using delimited control operators to interpret versions of the classical Axiom of Choice. Independently, Herbelin [10] has obtained similar results using delimited control operators himself.

REFERENCES


Date: June 14, 2015.


