

Proving Program Termination and Liveness

Byron Cook

Microsoft Lab., Cambridge University, United Kingdom

These lectures will introduce students to automatic methods available for proving that programs eventually do something good (e.g. that programs don't hang, that programs don't forget to release locks, etc). Topics will include rank-function synthesis, variance analysis and refinement, fair termination, concurrency, and the support for programs with dynamically allocated data structures.

References

1. B. Cook, A. Podelski, and A. Rybalchenko. *Termination Proofs for Systems Code*. In “Programming Language Design and Implementation” (PLDI'06); 2006.
2. J. Berdine, A. Chawdhary, B. Cook, D. Distefano, and P. O'Hearn. *Variance Analyses from Invariance Analyses*. In “Principles of Programming Languages” (POPL'07); 2007.
3. B. Cook, A. Gotsman, A. Podelski, A. Rybalchenko, and M. Vardi. *Proving that Programs Eventually do Something Good*. In “Principles of Programming Languages” (POPL'07); 2007.
4. A. Chawdhary, B. Cook, S. Gulwani, M. Sagiv, and H. Yang. *Ranking Abstractions*. In “European Symposium on Programming” (ESOP'08); 2008.
5. B. Cook, A. Podelski, and A. Rybalchenko. *Proving Thread Termination*. In “Programming Language Design and Implementation” (PLDI'07); 2007.