A Debugger for Lazy Functional Logic Languages[*]

Bernd Braßel

Last year we have developed a technique to replay the execution of lazy functional programs with a strict semantics by recording information of unevaluated expressions [BFH$^+$07]. The recorded information is called an *oracle* and is very compact. Oracles contain the number of strict steps between discarding unevaluated expressions. The technique of recording oracles and replaying program executions was employed in a debugger [BS08].

In meantime the work was further developed to include also lazy functional *logic* languages. There are many problems involved in generalizing the technique to the broader setting. The talk will give an idea of those problems and present how they could be solved. In addition, a demonstration of the developed debugging tool will be given.

A question far from being solved is what kind of debugging techniques are actually adequate for functional logic languages. The talk will present some ideas and the author is already looking forward to interesting discussions on the topic.

References

[BFH$^+$07] B. Braßel, S. Fischer, M. Hanus, F. Huch, and G. Vidal. Lazy call-by-value evaluation. In *Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming (ICFP'07)*, pages 265 – 276, 2007.

[BS08] Bernd Braßel and Holger Siegel. Debugging Lazy Functional Programs by Asking the Oracle. In Olaf Chitil, editor, *Proc. Implementation of Functional Languages (IFL 2007)*, Lecture Notes in Computer Science. Springer, 2008. To appear.

[*] This work was partially supported by the German Research Council (DFG) under Grant Ha 2457/5-2