Amortised memory analysis using the depth of data structures

Brian Campbell

September 1, 2008

Abstract

The heap space analysis presented by Hofmann and Jost in their 2003 POPL paper [2] finds linear space bounds for many functional programs. It uses an amortized analysis: assigning hypothetical amounts of free space (called potential) to data structures in proportion to their sizes using type annotations. Constraints on these annotations in the type system ensure that the total potential assigned to the input is an upper bound on the total memory required to satisfy all allocations.

I will describe a related system for bounding the stack space requirements which uses the depth of data structures, by allowing us to express potential in terms of maxima as well as sums. This is achieved by adding extra structure to typing contexts (inspired by O'Hearn's bunched typing) to describe the form of the bounds. I will also describe the extra steps that must be taken to construct a typing during the analysis.

References

[1] Brian Campbell. *Type-based amortized stack memory prediction.* PhD thesis, University of Edinburgh, 2008.

[2] Martin Hofmann and Steffen Jost. Static prediction of heap space usage for first-order functional programs. In *POPL '03: Proceedings of the 30th ACM Symposium on Principles of Programming Languages*, New Orleans, 2003. ACM Press.