



A Space Semantics for Core Haskell

BakewellAdam and RuncimanColin

Abstract

Haskell currently lacks a standard operational semantics. We argue that such a semantics should be provided to enable reasoning about operational properties of programs, to ensure that implementations guarantee certain space and time behaviour and to help determine the source of space faults. We present a small-step deterministic semantics for the sequential evaluation of Core Haskell programs and show that it is an accurate model of asymptotic space and time usage. The semantics is a formalisation of a graphical notation so it provides a useful mental model as well as a precise mathematical notation. We discuss its implications for education, programming and implementation. The basic semantics is extended with a monadic IO mechanism so that all the space under the control of an implementation is included.
