Lightweight Languages for Interactive Graphics

Scott Draves

May 1, 1995

CMU–CS–95–148

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

also available on the web at
`http://hopeless.mess.cs.cmu.edu:8001/nitrous/top.html`

This research was partially supported by the National Science Foundation under grant number CCR-9057567. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation or the US Government.

Keywords: run-time code generation, rapid prototyping, compiler generators, picture/image generation, partial evaluation, nitrous

Abstract

Run time code generation (RTCG) and traditional compilation are two ends of the same underlying optimization: factoring computations out of repeated procedure calls. Self-applicable partial evaluation (PE) is a semantics-based program transformation traditionally used for automatic compiler generation (cogen). Recently, PE has also been applied to RTCG. My approach to RTCG is to implement a composable cogen for a compiler intermediate representation. My thesis is that this approach to optimizing programs can be particularly useful for interactive graphics toolkits. I intend to demonstrate this by implementing a compiler generator tuned for RTCG, and showing how it can be used to build software that is highly abstract and yet still as fast as less general programs.