CSCI-P 523 final project proposal

Caner Derici and Ryan Scott

April 11, 2016

1 Description

Our plan is to enrich our existing R5 compiler with tail call optimization, i.e., the ability to invoke certain function calls at the end of a function without the need to allocate an extra stack frame. This would permit the compiler to produce extremely space-efficient code for recursive functions (which are ubiquitous in functional programming languages like Racket and R5).

2 Milestones

We will work on the final project in the following order:

1. Apply tail call optimization only to first-order functions which use immediate tail recursion, e.g.

   ```
   (define (fib n acc)
     (if (< n 2) 1 (fib (- n 1) (* n acc))))
   ```

2. Apply tail cail optimization to first-order functions that are mutually recursive, e.g.,

   ```
   (define (odd?  n)
     (if (eq? n 0) #f (even? (- n 1))))
   (define (even? n)
     (if (eq? n 0) #t (odd?  (- n 1))))
   ```

3. Apply tail cail optimization to all tail cails, even those involving higher-order functions. For example:

   ```
   (define (foo fn)
     (fn 5 1))
   (foo fib)
   ```

   Implementing this correctly may involve other static analysis techniques (e.g., decomposition of the source code into basic blocks).

We are confident in our ability to complete the first two milestones before the final deadline. The last milestone is less certain, so we will shoot for it as an ideal goal once the other two milestones have been finished.