

Assignment 9 — Continuation-passing style — assigned Monday 11 November — due Friday 13 December

This assignment is worth 520 points. Full credit is 200 points. The Friday 13 December deadline for this assignment is firm. No credit will be given for late submissions.

9.1 CPS language (40pts)

Define a CPS language suitable for translating your core lambda language into continuation-passing style. Define both an external (textual) form, and an internal form in the compiler (ML types or equivalent). You can use the CPS language of the SML/NJ compiler, described in Appel: *Compiling with Continuations* as your guide.

9.2 CPS transformation (40pts)

Define the rules for transforming arbitrary core lambda language expressions into your CPS language. (See Chapter 5 of Appel.) Implement these rules as a pass in the compiler.

9.3 Improving the CPS transformation (40pts)

A typical CPS transformation (as in Chapter 5 of Appel) introduces numerous administrative λ -abstractions. To remove them, write a simplification phase. (See Appel, pp. 68-69, under **Beta contraction**.)

9.4 Interpreter (40pts)

Write an interpreter for the CPS language.

9.5 Semantics

9.5.1 (40pts)

Give a semantic description of your core lambda language using the tools of structured operational semantics.

9.5.2 (40pts)

Render the operational description of your core lambda language in ML (or other programming language) to obtain an interpreter.

9.5.3 (40pts)

Give a denotational description of your core lambda language.

9.5.4 (40pts)

Render the denotational description of your core lambda language in ML (or other programming language) to obtain an executable semantic description, i.e., a semantics-based definitional interpreter.

9.5.5 (40pts)

Give a semantic description of your CPS language using the tools of structured operational semantics.

9.5.6 (40pts)

Render the operational description of your CPS language in ML (or other programming language) to obtain an interpreter.

9.5.7 (40pts)

Give a denotational description of your CPS language.

9.5.8 (40pts)

Render the denotational description of your CPS language in ML (or other programming language) to obtain an executable semantic description. See Chapter 3 of Appel for an example.

9.5.9 (40pts)

Prove that the rules you gave in 9.2 are sound.

How to turn in

Turn in your code by running `~darko/handin your-file` on a regular UNM CS machine or on *delta*.

You should use whatever filename is appropriate in place of `your-file`. You can put multiple files on the command line, or even directories. Directories will have their entire contents handed in, so please be sure to clean out any cruft.

Remember to submit extensive tests of your programs!

Homework must be accompanied by the following statement: *“I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents’ Policy Manual, including Section 4.8, Academic Dishonesty.”* The manual is available at <http://www.unm.edu/~brpm/index.htm>