1 Control Flow Analysis

Consider the following lambda calculus program.

\[(\lambda f. (f\,76) + (f\,77))\, (\lambda a. a)\]

(a) Add labels to the program. That is, make it an expression in the labeled lambda calculus of Lecture 24, where every label is unique.

(b) Let \(e\) be your labeled lambda calculus program. Write out \(C[e]_e\), i.e., the set of constraints for the program \(e\). (Hint, you should have 20 constraints in total. In particular, for each of the 3 applications, you should have 4 constraints, 2 for each of the lambda terms in the program.)

(c) Find \(C^*\) and \(r^*\), the smallest functions that satisfy the constraints you generated in the question above.

(d) Check that your functions \(C^*\) and \(r^*\) make sense. That is, if an expression labeled \(l\) can evaluate to an expression labeled \(l'\), do you have \(l' \in C^*(l)\)?

(e) Consider adding the expression \((\text{let } x = e_1 \text{ in } e_2)\) to the language. Define \(C[(\text{let } x = e_1 \text{ in } e_2)]_e\). Try rewriting the program above using one or more let expressions, and make sure that the constraints you generate for the modified program produce the same solution \(C^*\) and \(r^*\).