Homework 5: Lambda Calculus

Due Thursday, March 12 at 9:55 am

I find the best way to learn the Church encodings and the Y combinator is to reproduce and explain them yourself. This assignment asks you to do just that. You may use any resources that you like, including the textbook and the Internet, provided that you cite them. The one exception is the code presented during lecture—please do not copy it directly (although you may look at it before you write your solutions).

If you have time, I recommend going through the challenge exercise even if you don't submit it.

1. Warmup

(25 points)
Reduce the following expressions to values. These are written in the λ calculus extended with basic infix arithmetic. After you have written them out, you may check and revise your work by porting them to Scheme.

a) $(\lambda x . (\lambda y . (x + y)))\ \ 10\ \ 5$

b) $(\lambda f . (\lambda x . (f\ x)))\ \ (\lambda y .\ y^2)\ \ 12$

c) $$\begin{array}{l} ((((\lambda f . (\lambda x . ((f\ x)\ f))) \\ \qquad (\lambda y .\ (\lambda g. (g\ y^2)))) \\ \qquad 2) \\ \quad (\lambda a . a)) \end{array}$$

The last is a single expression, indented to make it easier to align the functions with their arguments. It may help to write out the intermediate results when reducing these.

2. Lists

(25 points)
Explain, in your own words and using code examples in Scheme how to implement Church Lists using only single-argument procedure definition, variables, and procedure application. Recall that Church Lists require the following procedures: pair, first, rest.

Challenge

(no credit)
3. Derive the Y combinator, in your own words and using code examples in Scheme.