

APL: A Takehome Midterm

Mayer Goldberg, 2014

April 12, 2014

Contents

1	Instructions	1
2	The λ -Calculus	2
3	CPS	2

1 Instructions

- You should take about one week to do this exam. If you take two, that's not a problem. Please don't take longer. Of course, we shall begin "counting" these week or two once classes resume after the Passover break. If you have any questions, are stuck and need help, please ask in class, after class, during office hours, or call me at 055-6603392. It will be difficult for me to write you long, technical replies by email, so actually, email is not a good choice for seeking help, other than to set an appointment.
- We do not have a grader. Therefore, I would like your code to be as self-explanatory and easy to grade as possible:
 - Please submit your answers *typeset* i.e., in text files. Please do not send me handwritten documents or their scans -- I don't want to have to struggle with reading other peoples' handwriting.
 - Please submit a *zip* file with all the relevant files (no *rar*, *tgz* or other archival formats!).
 - I don't use Windows -- Please don't send me *doc* files. PDF or source files are the preferred format.
 - Please include a `readme.txt` with your full name, ID, and a statement to the effect that this work is your own, and that you have neither solicited nor obtained help from others, and that your answers have not been "found" on the internet. I personally detest academic dishonesty, find it intolerable and inexcusable -- PLEASE let's not go there!
 - Your code should include a sample run for interesting and important test cases. You can include this sample run as a text file, using the `script` utility on linux or Cygwin. Just type `script foo`, and you will get a new shell prompt. Run your code from the shell, then exit the shell either with `^D` or `exit`. You will return to your previous shell, and the file `foo` will contain the entire session from the new shell.

2 The λ -Calculus

Pick 3 areas, and do 1 problem of each.

1. Syntax

- (a) Exercises 57, 59

2. Definability

- (a) Exercises 27, 40

3. Fixed points

- (a) Write a λ -term that takes a Church numeral n and returns Smullyan's construction (Item #214 in the tutorial)
- (b) Implement in Java Φ_1^2, Φ_2^2 , the multiple fixed-point combinators that extend *Turing's fixed-point combinator*, for defining two, mutually-recursive functions. Demonstrate your code using your favourite two mutually-recursive procedures.

4. Bases

- (a) Exercises 80

- (b) Show that $\mathfrak{W} = \{I, J, K\}$ is a basis for the λ -calculus, where

- i. $I = \lambda x.x$
ii. $J = \lambda xyz.t.xy(xtz)$
iii. $K = \lambda xy.x$

3 CPS

Write in Scheme the procedure `find-match`, which takes a predicate, and a nested ordered-pair, and returns the *first* leaf that matches the predicate, *either from the left or from the right*, by recursively traversing the list only once. This problem was explained in class, and as discussed, you will have to decide whether to start searching from the left or from the right, and based on your choice, your answer may be off by one element.