Homework 1: Lectures 1 & 2

CS 440: Programming Languages and Translators, Fall 2019

Due Wed Sep 4, 11:59 pm

8/26: Q2; 8/28: Q4, 9/2: Q4

How to submit

See http://cs.iit.edu/~cs440 → Homework Policies for information on working with others, how to submit, etc. If you want to submit multiple files, zip them together and submit the zipped file.

Problems [50 points]

1. [15 = 5 * 3 pts] In ghci, what happens on the inputs below: Give results or briefly describe the error(s). ("Brief" = Don’t rewrite the entire error message; translate it into a few words in English.)
   a. \( \sin (\cos \pi) \)
   b. \( \cos -1 \)
   c. \( \sin \cos \pi \)
   d. \( (\sqrt \cdot \text{head} [\sqrt]) \) 16.0  (Recall: infix dot is function composition)

2. [9 = 3 * 3 pts]. What do you get if you delete all the extra (and maybe even bad) parentheses from
   a. \( (\cos(\sqrt{2.5})+((\sin)\pi))(\ast)2) \)
   b. \( (((:)(('a':("b")) + "cd")) (((['c']) + "(d)")) ) \) [8/26]
   c. \( ((([[17]])))::([^])) \)

3. [4 pts] Rewrite the following expression so that it uses prefix functions throughout.
   (Hint: think (+), for example.)
   \( (\ast) \) : (a + b) * c) / (d ^e) -- ^ is exponentiation

4. [4 pts]. Rewrite the following expression so that it uses infix [9/2] notation throughout.
   (Hint: what do backquotes do?)
   \( \cdot \) f (g x (h a b)) (c (d e f)) \) [added 8/28]

5. [5 pts] Complete the following function definition so that on any list, \( f \) returns True.
   \( f \) x = x == [x !! i] ?

6. [5 pts]. Complete the following function definition: \( \text{stutter} \) \( n \) \( x \) should return a list of length \( n \) where each element is \( x \). E.g., stutter 3 5 = [5,5,5]. Use a list comprehension to produce the result.
   Note \( \text{stutter} \) 0 \( x \) = [ ]. Don’t worry about \( n < 0 \); the canonical solution makes stutter return [ ].
   \( \cdot \) stutter n x = ???

7. [11 pts]. Calculate the first 5 elements of the list \( f \) defined below using successive approximations.
   Don’t count [ ] as one of the 5.) If you like, you can cut down on the writing by defining extra variables. E.g., maybe apx1 is the first approximation, apx2 is the second, and so on.
   \( f = (1,1) : [(b,a+b) | (a,b) <= f] \)