Homework 1 Solution

CS 440: Programming Languages and Translators, Fall 2019

Problems

1. (Various expressions in ghci)

1a. Calculates the sine of the (cosine of π).
   
   > sin (cos pi)
   -0.8414709848078965

1b. Error: We asked for cos of function unary minus. Need parens around – 1.
   
   > cos -1

1c. Error: Like (b): we asked for sin of function unary cosine. Need parens around cos pi.
   
   > sin cos pi

1d. Calculates 4th root of 16.0: [sqrt] is a list containing one element, the square root function. Head of that yields the function itself, and (sqrt . sqrt) 16.0 = sqrt(sqrt(16.0)).

2. (Fix parentheses)

2a. (cos(sqrt 2.5)+(sin pi))*2
   (The parens around * were bad, the others were redundant.)

2b. (: ('a' : "b" ++ "cd") ['c'] ++ "(d)"

2c. [[[[17]]],[[]]]
   (Trick: Just typing the original expression into ghci gives you this.)

3. (Use prefix) (/) (/) (\*) (\*) (+) a b c) (+) d e
   (Trick: substitute numbers for a, b, ..., e to double-check your computation)

4. (Use infix) (x `g` (a `h` b)) `f` (c (e `d` f))

5. (List comprehension) We want the list [x !! 0, x !! 1, x !! 2, etc]
   
   f x = x == [x !! i | i <- [0..length x - 1]]

6. (List comprehension) We want [x, x, x, .., x] where there are n x's.
   
   stutter n x = [x | i <- [1..n]]
7. (Use referential transparency to compute part of (infinite) list) The list \( f \) (which shows the Fibonacci numbers in overlapping pairs) was defined as

\[
f = (1,1) : [(b, a+b) | (a, b) <- f]
\]

The first five approximations to the infinite list (i.e., heads 1 - 5 of \( f \)) are:

\[
\begin{align*}
(1,1) : [(b,a+b) | (a,b) <- []] &= [(1,1)] \\
(1,1) : [(b,a+b) | (a,b) <- [(1,1)]] &= [(1,1), (1,2)] \\
(1,1) : [(b,a+b) | (a,b) <- [(1,1),(1,2)]] &= [(1,1),(1,2),(2,3)] \\
(1,1) : [(b,a+b) | (a,b) <- [(1,1),(1,2),(2,3)]] &= [(1,1),(1,2),(2,3),(3,5)] \\
(1,1) : [(b,a+b) | (a,b) <- [(1,1),(1,2),(2,3),(3,5)]] &= [(1,1),(1,2),(2,3),(3,5),(5,8)]
\end{align*}
\]

If you want more gory detail, giving names to the approximations might make things clearer:

\[
\begin{align*}
apx1 &= (1,1) : [(b,a+b) | (a,b) <- []] \\
&= (1,1) : [] = [(1,1)]
\end{align*}
\]

\[
\begin{align*}
apx2 &= (1,1) : [(b,a+b) | (a,b) <- apx1] \\
&= (1,1) : [(b,a+b) | (a,b) <- (1,1) : tail apx1] \\
&= (1,1) : (1,2) : [(b,a+b) | (a,b) <- []] \\
&= (1,1) : (1,2) : [] \\
&= [(1,1), (1,2)]
\end{align*}
\]

\[
\begin{align*}
apx3 &= (1,1) : [(b,a+b) | (a,b) <- apx2] \\
&= (1,1) : [(b,a+b) | (a,b) <- (1,1) : tail apx2] \\
&= (1,1) : (1,2) : [(b,a+b) | (a,b) <- [(1,2)]] \\
&= (1,1) : (1,2) : (2,3) : [(b,a+b) | (a,b) <- []] -- tail(tail apx2) \\
&= (1,1) : (1,2) : (2,3) : [] \\
&= [(1,1),(1,2),(2,3)]
\end{align*}
\]

etc.