L Systems

```
41 expandOne :: Rules Char -> [Char] -> [Char]
42 expandOne _ [] = []
43 expandOne r@((k, v):xs) s@(y:ys) = lookupChar r y ++ expandOne r ys
```

K, v and xs is never used in this function. Might as well make it an underscore "_".

```
57 move :: Command -> Float -> TurtleState -> TurtleState
58 move (F) dist ((x, y), angle) = let rads = pi * (angle / 180)
59                                     deltaX = (cos rads)
60                                     deltaY = (sin rads)
61                                  in ((x + deltaX, y + deltaY), angle)
```

Could make dist an underscore as well since its never used

```
5 parse :: Rules Command -> [Char] -> [Command]
6 parse _ "" = []
7 parse a@((k,v):ys) s@('[':x:xs) = B comm : parse a xs
8 parse a@((k,v):ys) s@(']':xs) = [] : parse a xs
9 parse a@((k,v):ys) s@(x:xs)    = comm : parse a xs
0     where comm  = lookupChar commandMap x
1
```

Where is commandMap? Also comm is used in the second pattern match, which wouldn't work…, Even if it did x is not exactly what u want to work on here.

For trace1/2, ive attached the model answers, so feel free to have a look and figure it out. If u have any questions lmk and ill answer whenever I can.

Calculus

```
11 -------------------------------------------------------------------------
12 -- Type classes and class instances
13
14 -- Comment this out if you want to implement your own instance in terms
15 -- of `showExpr`
16 deriving instance Show Expr
17
18 instance Num Expr where
19   fromInteger = undefined
20   negate      = undefined
21   (+)         = undefined
22   (*)         = undefined
23
24 instance Fractional Expr where
25   fromRational = undefined
26   (/)          = undefined
27
28 instance Floating Expr where
29   sin = undefined
30   cos = undefined
31   log = undefined
```

Use these to simplify ur diff function. So for instance, set a + b = Add a b, then u can write a + b in diff, and the compiler will substitute it for Add a b.

Nothing else that stood out to me. Ur choice of variable names is ok, but being more descriptive wouldn't hurt.