```
data Expr
                      type ESem b
= Add Expr Expr
                      = (b \rightarrow b \rightarrow b)
Mul Expr Expr
                        , b \rightarrow b \rightarrow b
I Con Int
                        , Int \rightarrow b
foldE :: ESem b \rightarrow Expr \rightarrow b
foldE(a,m,c) = f where
  f(Add e1 e2) = a (fe1) (fe2)
  f(Mul e1 e2) = m(fe1)(fe2)
  f (Con n)
              = c n
evalExpr :: Expr → Int
evalExpr = foldE evalSem
```

evalSem :: ESem Int evalSem = ( (+) , (\*) , id )