

Assignment 3

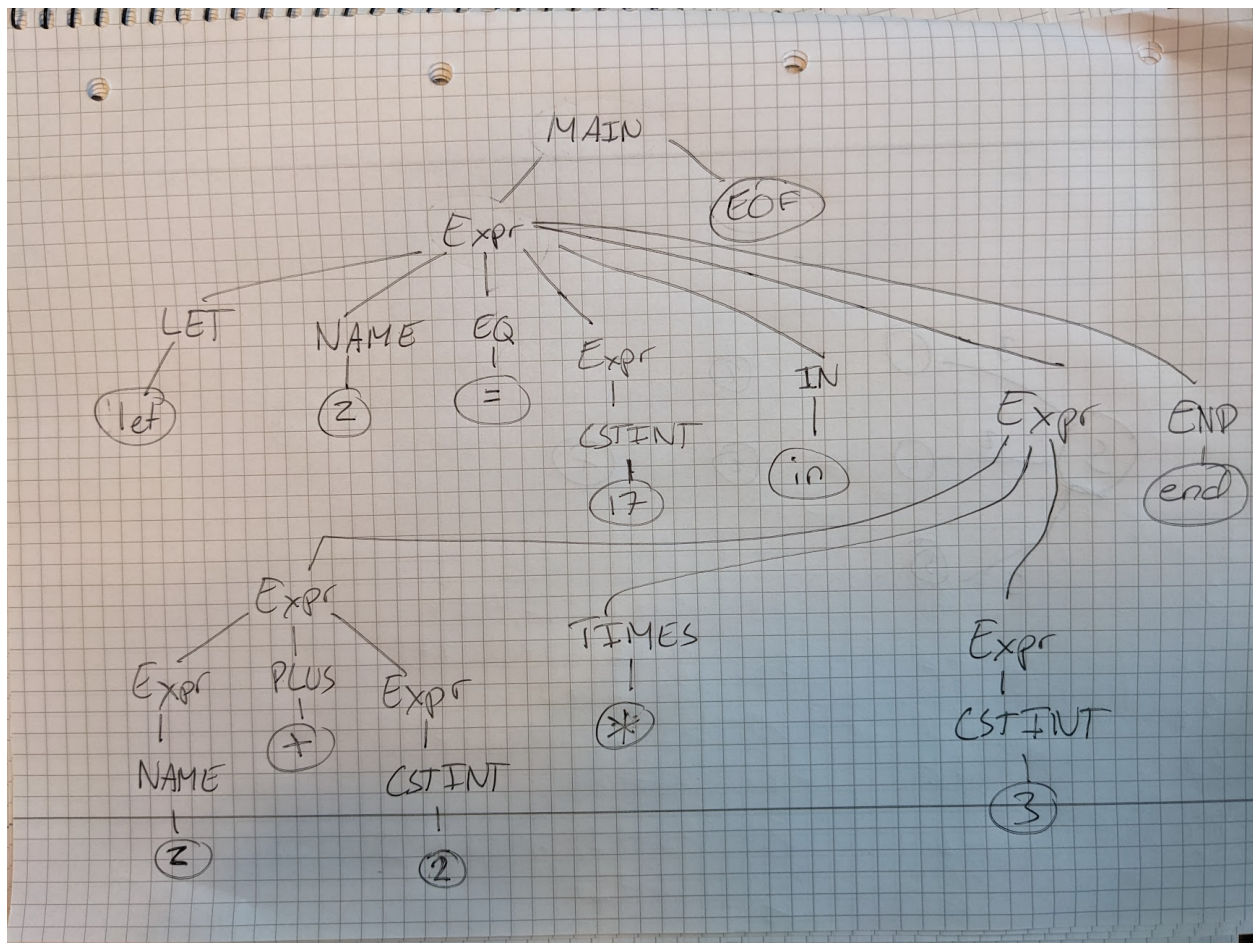
atro, aguh, frepe

Exercise 3.3

```
Grammar:
Main ::= Expr EOF           rule A
Expr ::= NAME               rule B
      | CSTINT              rule C
      | MINUS CSTINT        rule D
      | LPAR Expr RPAR      rule E
      | LET NAME EQ Expr IN Expr END rule F
      | Expr TIMES Expr     rule G
      | Expr PLUS Expr      rule H
      | Expr MINUS Expr     rule I

Right-most derivation:
Main
-> Expr EOF (rule A)
-> Let z = Expr in Expr end EOF (rule F)
-> Let z = Expr in Expr * Expr end EOF (rule G)
-> Let z = Expr in Expr * 3 end EOF (rule C)
-> Let z = Expr in Expr + Expr * 3 end EOF (rule H)
-> Let z = Expr in Expr + 2 * 3 end EOF (rule C)
-> Let z = Expr in z + 2 * 3 end EOF (rule B)
-> Let z = 17 in z + 2 * 3 end EOF (rule C)
```

Exercise 3.4



Exercise 3.5

```
fromString "(1 + 2) * 3" = Prim (" ", Prim("+", CstI 1, CstI 2), CstI 3)
fromString "let abc = (12 * 13) in abc - 12 end" = Let ("abc", Prim (" ", CstI 12, CstI 13), Prim ("-", Var "abc", CstI 12))
fromString "let abc = x in abc+3" = Fails due to missing "end" keyword.
```

See code for exercise 3.6, 3.7