

# Peter Chinetti

CS 440

Assignment LL Parsing

1. There are token types for each terminal of the grammar. There are also expression types for each production of the grammar. When parse is called, it transforms a token list into an expression tree. The method of transformation is by matching the token list input as the parameter to the parse function to one of the cases that matches a production for the grammar. These cases can recursively call other functions as they transform the token stream into an expression.
2. Adding if:

```
data Tokens = TokInt Int
             | TokPlus
             | TokGt
             | TokIf

data Exp = IntExp Int
         | PlusExp Exp Exp
         | GtExp Exp Exp
         | IfExp Exp Exp Exp

parse (TokIf: ts) =
  let (e1, r1) = parse ts
      (e2, r2) = parse r1
      (e3, r3) = parse r2
  in (IfExp e1 e2 e3, r3)
```