Juan Bancamper
Lab 3

Crafting a Compiler

4.7

a. E E E
b. T F T
c. This grammar allows for left to right associativity because of the recursive derivation of F. Whenever a token of E or T is created, F can return back to the beginning of the grammar, causing a non deterministic loop

5.2

```
c. parseStart(token)
        parseValue(token)
        match('$')

parseValue()
        switch()
                case "n"
                        match("num")
                case "("
                        match("(")
                        parseExpr()
                        match(")")

parseExpr()
        switch()
                case "+"
                        match('+')
                        parseValue()
                        parseValue()
                case 'p'
                        match("prod")
                        parseValues()

parseValues()
        switch()
                case ""
                        return
                default
                        parseValue()
                        parseValues()
```

Dragon

4.2.1

a. S
b. SS