Gianna Julio

CMPT432 - Design of Compilers
Lab 4

---

1. [***Crafting a Compiler* - Exercise 4.9**] Compute First and Follow sets for the nonterminals of the following grammar:

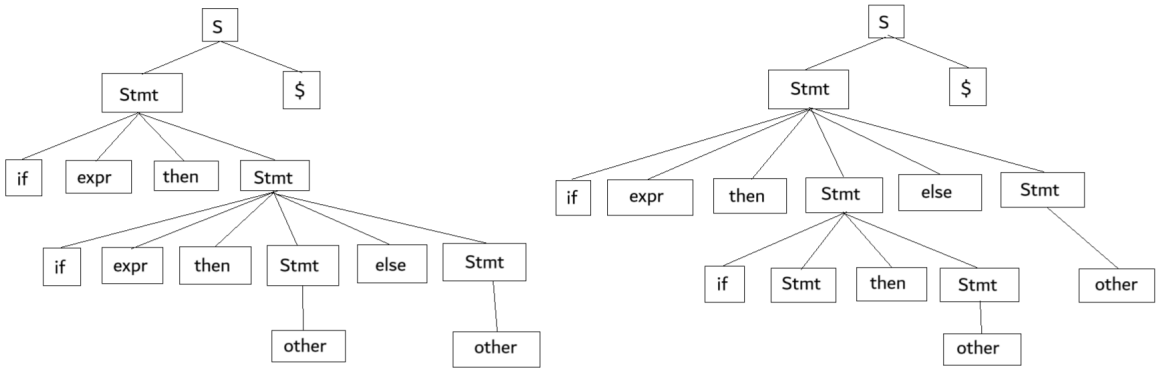
```
1 S → a S e
2   | B
3 B → b B e
4   | C
5 C → c C e
6   | d
```

```
First(S) = {a, b, c, d}
First(B) = {b, c, d}
First(C) = {c, d}

Follow(S) = {$, e}
Follow(B) = {e}
Follow(C) = {e}
```

2. [***Crafting a Compiler* - Exercise 5.10**] Show the two distinct parse trees that can be constructed for *if expr then if expr then other else other* using the grammar given below. For each parse tree, explain the correspondence of *then* and *else*.

```
1 S    → Stmt $
2 Stmt → if expr then Stmt else Stmt
3      | if expr then Stmt
4      | other
```



In the first parse tree, the first *then* corresponds to the first expansion of the initial Stmt. The second *then* and the *else* correspond to the second expansion of the second Stmt. In the second parse tree, the first *then* and the *else* correspond to the first expansion of the initial Stmt. The second *then* corresponds to the second expansion of the second Stmt.

3. [***"Dragon" Textbook* - Exercise 4.4.3**] Compute FIRST and FOLLOW for the grammar `S` $\rightarrow$ `SS +` | `SS*` | `a`

   `First(S) =` {`a`}
   `Follow(S) =` {`+, *`}