



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

```

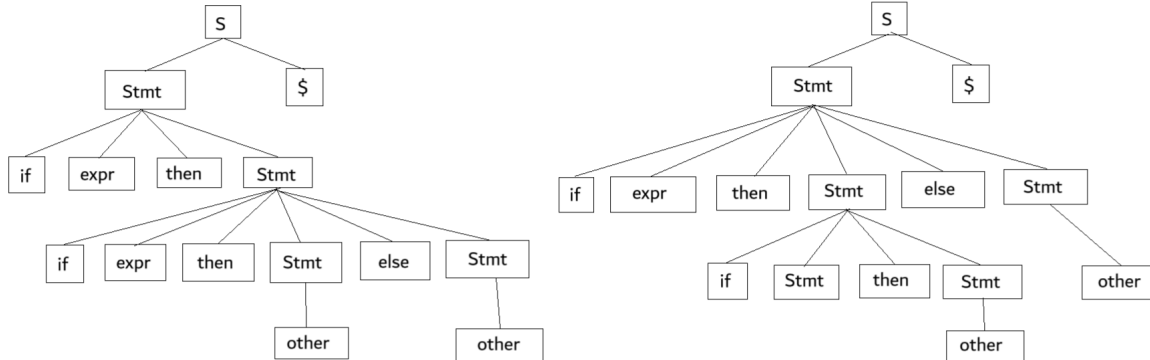
	nullable?	first	follow
S	no	{a, b, c, d}	{\$, e}
B	no	{b, c, d}	{e}
C	no	{c, d}	{e}

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 $S \rightarrow SS + \mid SS* \mid a$

	nullable?	first	follow
S	no	{a}	{+, *}