

Lab 4

Crafting a Compiler

4.9 (First and Follow Set)

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	a	b	c	d	e
S	no	a,b,c,d	E, \$	1	2	2	2	~
B	no	b,c,d	e,b	~	3	4	4	~
C	no	c,d	e	~	~	5	6	~

5.10 (dangling else parse trees)

Show the two distinct parse trees that can be constructed for

Input: if expr then if expr then other else other

Using grammar given in Figure 5.17. For each parse tree, explain the correspondence of then and else.

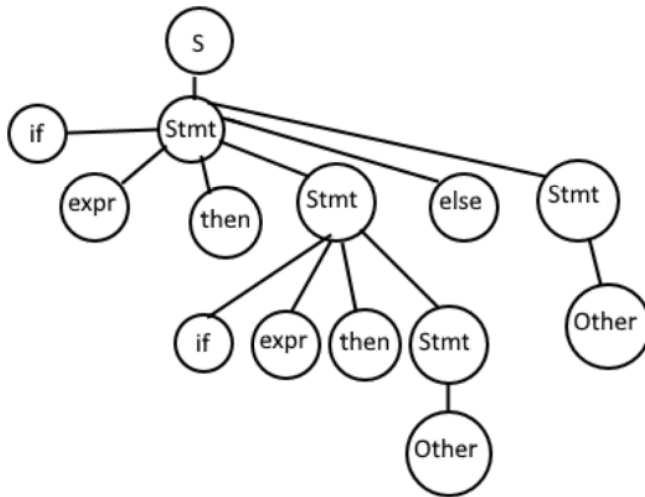
```

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

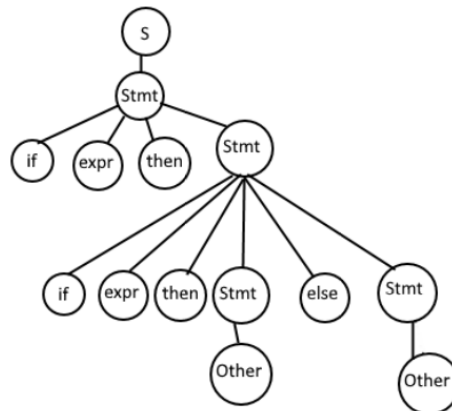
```

Figure 5.17: Grammar for if-then-else.

First Parse Tree: if expr then (if expr then other) else other



Second Parse Tree: if expr then (if expr then other else other)



Because the language is ambiguous there are multiple ways as to which this can be parsed. The first section has the if-else as the first condition that is read and the single if condition inside the first statement of the if-else. The second version can have a plain if statement be the first thing read with the if-else statement inside of it.

Dragon 4.4.3 (First and Follow set)

Compute FIRST and FOLLOW for the grammar of Exercise 4.2.1.

Exercise 4.2.1: Consider the context-free grammar:

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

	Nullable	first	follow
S	no	a	a,+,*, \$

