

# Lab Four

Emily Doran

Emily.Doran1@Marist.edu

March 15, 2021

## CRAFTING A COMPILER

### 4.9 (FIRST AND FOLLOW SET)

**Compute First and Follow Sets For the Nonterminals of the Following Grammar.**

```

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

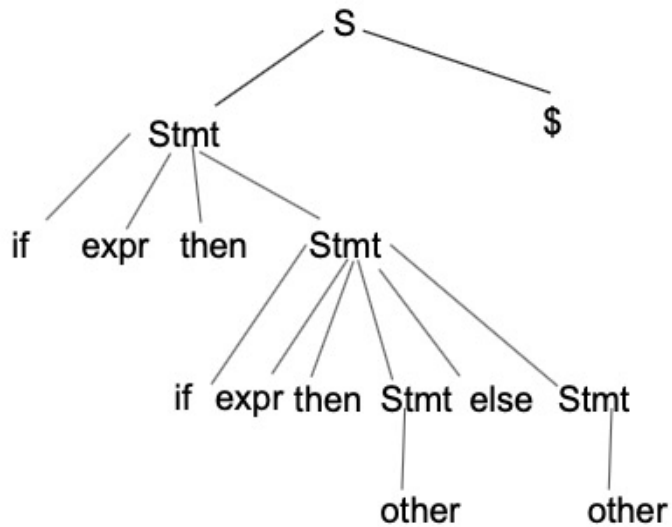
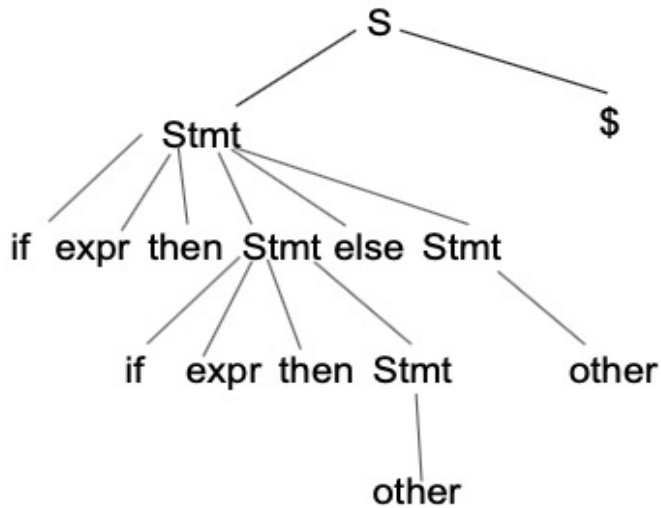
	Nullable	FIRST	FOLLOW
S	no	a, b, c, d	\$, e
B	no	b, c, d	e
C	no	c, d	e

### 5.10 (DANGLING ELSE PARSE TREES)

**Show the Two Distinct Parse Trees That Can be Constructed For**

`if expr then if expr then other else other`

**Using the Grammar Given in Figure 5.17 (Appendix) For Each Parse Tree, Explain the Correspondence of Then and Else.**



In the first tree, the then does not represent everything after it to be a singular statement. It just assumes that the if statement is what occurs, but the else is considered to be part of the first if. On the other hand, with the second parse tree, it does represent everything after the then to be a part of a single statement. Everything after that then will only be run if the first expr is true, there is no else statement if expr is false.

## DRAGON

### 4.4.3 (FIRST AND FOLLOW SET)

Compute **FIRST** and **FOLLOW** for the grammar of Exercise 4.2.1 (Appendix).

	Nullable	FIRST	FOLLOW
S	no	a	a+, a*, \$

## APPENDIX

FIGURE 5.17

### 5.6. A Non-LL(1) Language

```

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.

#### 4.2.1 GRAMMAR

**Exercise 4.2.1:** Consider the context-free grammar:

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