Lab 4 – Analyzing Grammars

Ryan Munger
Ryan.Munger1@marist.edu

February 27, 2025

1 Crafting a Compiler

1.1 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
```

Figure 1: Grammar for 4.9

First Set: Set of all the possible terminals that can appear at the beginning of any string derived from a specific non-terminal.
Follow Set: Set of all the possible terminals that can immediately follow a specific non-terminal.

Production	Nullable	FIRST	FOLLOW
S	No	{a, b, c, d}	{e, d}
B	No	{b, c, d}	{e, d}
C	No	{c, d}	{e, d}

1.2 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 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
```

Figure 2: Grammar for 5.10

Tree 1:

```
S
-<Stmt>
--[if expr then]
--<Stmt>
---[if expr then]
---<Stmt>
----[Other]
--[else]
--<Stmt>
---[Other]
$
```

Tree 2:

```
S
-<Stmt>
--[if expr then]
--<Stmt>
---[if expr then]
---<Stmt>
----[Other]
---[else]
---<Stmt>
----[Other]
$
```

Correspondence of *then* & *else*: In tree #1, the *else* corresponds with the outer *if expr then*. In tree #2, the *else* corresponds to the inner *if expr then*. This is important as we need to know which *if expr then* the *else* is actually for.

2 Dragon Book

2.1 4.4.3 – First and Follow set

Compute FIRST and FOLLOW for the grammar below.

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

Production	**Nullable**	**FIRST**	**FOLLOW**
S	No	{a}	{+, *, a}