

Lab Three

Juan S. Vasquez

jvasquez1@marist.edu

February 14, 2018

1 EXERCISE 4.7

```
1 Start -> E $
2 E      -> T plus E
3         | T
4 T      -> T times F
5         | F
6 F      -> (E)
7         | num
```

1.1 LEFTMOST DERIVATION FOR NUM PLUS NUM TIMES NUM PLUS NUM \$

```
Start ->lm E $
->lm T plus E $
->lm F plus E $
->lm num plus E $
->lm num plus T $
->lm num plus T times F $
->lm num plus F times F $
->lm num plus num times F $
->lm num plus num times (E) $
->lm num plus num times (T plus E) $
->lm num plus num times (F plus E) $
->lm num plus num times (num plus E) $
->lm num plus num times (num plus T) $
->lm num plus num times (num plus F) $
->lm num plus num times (num plus num) $
```

1.2 RIGHTMOST DERIVATION NUM TIMES NUM PLUS NUM TIMES NUM \$

Start \rightarrow_{rm} E \$
 \rightarrow_{rm} T \$
 \rightarrow_{rm} T times F \$
 \rightarrow_{rm} T times num \$
 \rightarrow_{rm} F times num \$
 \rightarrow_{rm} (E) times num \$
 \rightarrow_{rm} (T plus E) times num \$
 \rightarrow_{rm} (T plus T) times num \$
 \rightarrow_{rm} (T plus F) times num \$
 \rightarrow_{rm} (T plus num) times num \$
 \rightarrow_{rm} (T times F plus num) times num \$
 \rightarrow_{rm} (T times num plus num) times num \$
 \rightarrow_{rm} (F times num plus num) times num \$
 \rightarrow_{rm} (num times num plus num) times num \$

1.3 GRAMMAR STRUCTURE

When we evaluate the CSTs created by the rightmost and leftmost derivations, we can see that both of the trees follow order of operations dictated by mathematics. Parentheses in both cases execute first, followed by multiplication and then addition (if applicable).

2 EXERCISE 5.2

```
1 Start -> Value $
2 Value -> num
3       | lparen Expr rparen
4 Expr -> plus Value Value
5       | prod Values
6 Values -> Value Values
7       |  $\lambda$ 
```

2.1 5.2C - RECURSIVE DESCENT PARSER

```
procedure Start()
.   switch(...)
.       case ts.peek() ∈ {num, lparen, rparen, plus, prod, $}
.           call Value()
.           call match($)
```

```
procedure Value()
.   switch(...)
.       case ts.peek() ∈ {num}
.           call match(num)
.       case ts.peek() ∈ {lparen}
.           call match(lparen)
.           call Expr()
.           call match(rparen)
```

```
procedure Expr()
.   switch(...)
.       case ts.peek() ∈ {plus}
.           call match(plus)
.           call Value()
.           call Value()
.       case ts.peek() ∈ {prod}
.           call match(prod)
.           call Values()
```

```
procedure Values()
.   switch(...)
.       case ts.peek() ∈ {num, lparen, rparen, plus, prod}
.           call Value()
.           call Values()
.       case ts.peek() ∈ {$}
.           return()
```

3 EXERCISE 4.2.1

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

3.1 4.2.1A - LEFTMOST DERIVATION

$S \xrightarrow{\text{lm}} SS^*$
 $\xrightarrow{\text{lm}} SS+S^*$
 $\xrightarrow{\text{lm}} aS+S^*$
 $\xrightarrow{\text{lm}} aa+S^*$
 $\xrightarrow{\text{lm}} aa+a^*$

3.2 4.2.1B - RIGHTMOST DERIVATION

$S \xrightarrow{\text{lm}} SS^*$
 $\xrightarrow{\text{lm}} Sa^*$
 $\xrightarrow{\text{lm}} SS+a^*$
 $\xrightarrow{\text{lm}} Sa+a^*$
 $\xrightarrow{\text{lm}} aa+a^*$

3.3 4.2.1C - PARSE TREE

