Lab 9 – Manipulating Grammars

Ryan Munger

Ryan.Munger1@marist.edu

April 5, 2025

1 Crafting a Compiler

1.1 Exercise 5.5 – LL(1) Grammar

Transform the following grammar into LL(1) form using the techniques presented in Section 5.5.

```
1 DeclList        → DeclList ; Decl
2                 | Decl
3 Decl            → IdList : Type
4 IdList          → IdList , id
5                 | id
6 Type            → ScalarType
7                 | array ( ScalarTypeList ) of Type
8 ScalarType      → id
9                 | Bound . . Bound
10 Bound          → Sign intconstant
11                 | id
12 Sign           → +
13                 | -
14                 | Epsilon
15 ScalarTypelist → ScalarTypeList , ScalarType
16                 | ScalarTyp
```

LL(1) requires a unique prediction for each combination of nonterminal and lookahead symbols. We must eliminate common prefixes and left recursion.

Updated grammar:

```
1 DeclList         → Decl DeclList'
2 DeclList'        → ; Decl DeclList'
3                  | Epsilon
4 Decl             → IdList : Type
5 IdList           → id IdList'
6 IdList'          → , id IdList'
7                  | Epsilon
8 Type             → ScalarType
9                  | array ( ScalarTypeList ) of Type
10 ScalarType      → id
12                 | Bound . . Bound
13 Bound           → Sign intconstant
14                 | id
15 Sign            → +
16                 | -
17                 | Epsilon
18 ScalarTypeList  → ScalarType ScalarTypeList'
19 ScalarTypeList' → , ScalarType ScalarTypeList'
20                 | Epsilon
```

2 Dragon Book

2.1 Exercise 4.5.3 – Bottom-Up Parsing

Give bottom-up parses for the following input strings and grammars.

Input: 000111
Grammar: $S \rightarrow 0S1|01$

Bottom-Up/Shift Reduce Parse:

```
Input:   | Operation  | Stack
000111   | Shift 0    | Eps
 00111   | Shift 0    | 0
  0111   | Shift 0    | 00
   111   | Shift 1    | 000
    11   | Reduce 01  | 0001
    11   | Shift 1    | 00S
     1   | Reduce 0S1 | 00S1
     1   | Shift 1    | 0S
   Eps | Reduce 0S1 | 0S1
   Eps | Accept S   | S
```

Input: aaa*a++
Grammar $S \rightarrow SS + |SS * |a$

Bottom-Up/Shift Reduce Parse:

```
Input:    | Operation  | Stack
aaa*a++   | Shift a    | Eps
 aa*a++   | Reduce a   | a
 aa*a++   | Shift a    | S
  a*a++   | Reduce a   | Sa
  a*a++   | Shift a    | SS
   *a++   | Reduce a   | SSa
   *a++   | Shift *    | SSS
    a++   | Reduce SS* | SSS*
     ++   | Shift a    | SS
     ++   | Reduce a   | SSa
     ++   | Shift +    | SSS
      +   | Reduce SS+ | SSS+
      +   | Shift +    | SS
    Eps | Reduce SS+ | SS+
    Eps | Accept S   | S
```