Lab Nine

Blake Mackey

blake.mackey1@Marist.edu

May 16, 2021

1 Crafting a Compiler Exercises

1.1 Exercise 5.5

The grammar:

```
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        | \lambda
15 ScalarTypeList -> ScalarTypeList , TypeList
16                 | ScalarType
```

Can be modified to become LL(1) by removing left recursion:

```
1 DeclList -> Var1 Var2
2 Var1 -> Decl
3 Var2 -> ; Decl Var1
4 Decl -> IdList : Type
5 IdList -> Id1 Id2
6 Id1 -> id
7 Id2 -> , id Id2
8 Type -> ScalarType
```

```
9      | array ( ScalarTypeList ) of Type
10 ScalarType -> id
11             | Bound .. Bound
12 Bound -> Sign intconstant
13        | id
14 Sign -> +
15        | -
16        | \lambda
17 ScalarTypeList -> Scalar1 Scalar2
18 Scalar1 -> ScalarType
19 Scalar2 -> , TypeList Scalar1
```

2 Dragon Book Exercises

2.1 Exercise 4.5.3

2.1.1 A

For the grammar:

S –> 0 S 1
| 01

The following bottom-up parse can be produced for the string 000111.

Stack	Token Stream	Handle	Action
0	00111		Shift
00	0111		Shift
000	111		Shift
0001	11	01	Reduce S -> 01
00S	11		Shift
00S1	1	0S1	Reduce S -> 0S1
0S	1		Shift
0S1		0S1	Reduce S -> 0S1
S			Accept

2.1.2 B

For the grammar:

S –> S S +
| S S *
| a

The following bottom up parse can be produced for the string aaa * a + +.

Stack	Token Stream	Handle	Action
a	aa*a++	a	Reduce S -> a
S	aa*a++		Shift
Sa	a*a++	a	Reduce S -> a
SS	a*a++		Shift
SSa	*a++	a	Reduce S -> a
SSS	*a++		Shift
SSS*	a++	SS*	Reduce S -> SS*
SS	a++		Shift
SSa	++	a	Reduce S -> a
SSS	++		Shift
SSS+	+	SS+	Reduce S -> SS+
SS	+		Shift
SS+		SS+	Reduce S -> SS+
S			Accept