Scott Hansen

Lab 9

Crafting a Compiler:
Exercise 5.5 and 6.51(challenge question)

5. 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                | λ
15 ScalarTypelist → ScalarTypeList , ScalarType
16                | ScalarType
```

Rule	Left	Transformation
1	DeclList	Decl DeclList’
	DeclList’	; DeclList
		ε
3	Decl	IdList : Type
4	IdList	id IdList’
	IdList’	, IdList
		ε
5		
6	Type	ScalarType
7		Array (ScalarTypeList) of Type

8	ScalarType	id
9		Bound .. Bound
10	Bound	Sign intconstant
11		id
12	Sign	+
13		-
14		ε
15	ScalarTypeList	ScalarType ScalarTypeList'
	ScalarTypeList'	, ScalarTypeList
		ε
16		

51. The bottom-up parsing techniques given in this chapter are more powerful than top-down techniques given in Chapter 5.

Using the alphabet $\{a, b\}$, devise a language that is not LL(k) for any k but is LR(k) for some k. What property of LR(k) parsing allows such a grammar to be constructed?

Dragon:
Do exercise 4.5.3 and 4.6.5

Exercise 4.5.1: For the grammar $S \rightarrow 0\ S\ 1 \mid 0\ 1$ of Exercise 4.2.2(a), indicate the handle in each of the following right-sentential forms:

Exercise 4.5.2: Repeat Exercise 4.5.1 for the grammar $S \rightarrow S\ S + \mid S\ S * \mid a$ of Exercise 4.2.1 and the following right-sentential forms:

Exercise 4.5.3: Give bottom-up parses for the following input strings and grammars:

a) The input 000111 according to the grammar of Exercise 4.5.1.

b) The input $aaa * a + +$ according to the grammar of Exercise 4.5.2.

a.

0

00
000
0001
00S
00S1
0S
0S1
S

b.

first

a
S
Sa
SS
SSa
SSS
SSS*
SS
SSa
SSS
SSS+
SS
SS+
S

last

Exercise 4.6.5: Show that the following grammar:

$$\begin{array}{lcl} S & \rightarrow & A\ a\ A\ b \mid B\ b\ B\ a \\ A & \rightarrow & \epsilon \\ B & \rightarrow & \epsilon \end{array}$$

is LL(1) but not SLR(1).

First	Symbols	a	b
{a,b}	S	1	2
{}	A		
{}	B		
a	a		
b	b		

Every row and column intersection only points to one production and is therefore atomic.