ECE 468

Problem Set 2 **Solutions**: Context-free Grammars, LL(1) Parsers

1. For the following sub-problems, consider the following context-free grammar:

$$\begin{aligned} S &\rightarrow A\$ && (1)\\ A &\rightarrow xAx && (2)\\ A &\rightarrow C && (3)\\ B &\rightarrow yBy && (4)\\ B &\rightarrow C && (5)\\ C &\rightarrow zBz && (6)\\ C &\rightarrow wAw && (7)\\ C &\rightarrow \lambda && (8) \end{aligned}$$

   (a) What are the terminals and non-terminals of this grammar?

   Answer: Terminals: $\{x, y, z, w, \$\}$, non-terminals: $\{S, A, B, C\}$

   (b) Show the derivation of the string $xzzx\$$ starting from S (specify which production you used at each step), and give the parse tree according to that derivation.

   Answer:

$$\begin{aligned} S &\rightarrow A\$ && \text{Rule 1}\\ &\rightarrow xAx\$ && \text{Rule 2}\\ &\rightarrow xCx\$ && \text{Rule 3}\\ &\rightarrow xzBzx\$ && \text{Rule 6}\\ &\rightarrow xzCzx\$ && \text{Rule 5}\\ &\rightarrow xzzx\$ && \text{Rule 8} \end{aligned}$$

   The parse tree follows directly from this derivation.

   (c) Give the first and follow sets for each of the non-terminals of the grammar.

   Answer:

$$\begin{aligned} First(S) &= \{x, z, w, \$\}\\ First(A) &= \{x, z, w, \lambda\}\\ First(B) &= \{y, z, w, \lambda\}\\ First(C) &= \{z, w, \lambda\} \end{aligned}$$

$$
\begin{aligned}
\mathit{Follow}(S) &= \{\} \\
\mathit{Follow}(A) &= \{\$, x, w\} \\
\mathit{Follow}(B) &= \{y, z\} \\
\mathit{Follow}(C) &= \{\$, x, y, z, w\}
\end{aligned}
$$

(d) What are the predict sets for each production?

Answer:

$$
\begin{array}{ll}
\mathit{Predict}(1) & \{x, z, w, \$\} \\
\mathit{Predict}(2) & \{x\} \\
\mathit{Predict}(3) & \{z, w, x, \$\} \\
\mathit{Predict}(4) & \{y\} \mathit{Predict}(5) \quad \{z, w, y\} \\
\mathit{Predict}(6) & \{z\} \\
\mathit{Predict}(7) & \{w\} \\
\mathit{Predict}(8) & \{\$, x, y, z, w\}
\end{array}
$$

(e) Give the parse table for the grammar. Is this an LL(1) grammar? Why or why not?

Answer:

	x	y	z	w	\$
S	1		1	1	1
A	2, 3		3	3	3
B		4, 5	5	5	
C	8	8	6, 8	7, 8	8

This is not an LL(1) grammar, because there are conflicts in the parse table.

(f) Show the steps your parser would take to parse "xzyyzx\$".

Answer: The parser would not be able to parse this string, because there are conflicts in the parse table.

2. for the following sub-problems, consider the following grammar:

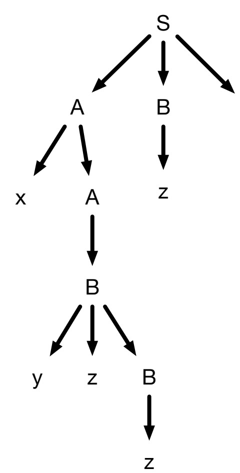
$$S \rightarrow AB\$ \tag{1}$$
$$A \rightarrow xA \tag{2}$$
$$A \rightarrow B \tag{3}$$
$$B \rightarrow yzB \tag{4}$$
$$B \rightarrow z \tag{5}$$

(a) What are the terminals and non-terminals of this grammar?

Answer: Terminals: $\{x, y, z, \$\}$; non-terminals: $\{S, A, B\}$

(b) Show the parse tree for $xyzzz\$$.

Answer:



(c) What are the first and follow sets for each of the non-terminals of the grammar?

Answer:

$$First(S) = \{x, y, z\}$$
$$First(A) = \{x, y, z\}$$
$$First(B) = \{y, z\}$$

$$Follow(S) = \{\}$$
$$Follow(A) = \{y, z\}$$
$$Follow(B) = \{y, z, \$\}$$

(d) What are the predict sets for each production?

Answer:

$$
\begin{aligned}
Predict(1) &= \{x, y, z\} \\
Predict(2) &= \{x\} \\
Predict(3) &= \{y, z\} \\
Predict(4) &= \{y\} \\
Predict(5) &= \{z\}
\end{aligned}
$$

(e) Give the parse table for this grammar. Is this an LL(1) grammar?

Answer:

	x	y	z	\$
S	1	1	1	
A	2	3	3	
B		4	5	

This is an LL(1) grammar, as there are no conflicts in the parse table.

(f) If we add the rule $A \rightarrow \lambda$, is the grammar still LL(1)? Why or why not?

Answer: Let us call the new rule rule 6. We can rebuild the first, follow, and predict sets:

$$
\begin{aligned}
First(S) &= \{x, y, z\} \\
First(A) &= \{x, y, z, \lambda\} \\
First(B) &= \{y, z\}
\end{aligned}
$$

Note that the First set of A changed.

$$
\begin{aligned}
Follow(S) &= \{\} \\
Follow(A) &= \{y, z\} \\
Follow(B) &= \{y, z, \$\}
\end{aligned}
$$

Note that none of the follow sets changed!

$$Predict(1) = \{x, y, z\}$$
$$Predict(2) = \{x\}$$
$$Predict(3) = \{y, z\}$$
$$Predict(4) = \{y\}$$
$$Predict(5) = \{z\}$$
$$Predict(6) = \{y, z\}$$

But the predict set for rule 6 is $Follow(A)$. If we build the parse table for this new grammar, we get:

	x	y	z	\$
S	1	1	1	
A	2	3, 6	3, 6	
B		4	5	

Which means we have a conflict: if we're expanding an A, and we see a y or z, don't know whether to turn it into a B using rule 3 or to remove it (turn it into λ) using rule 6. Thus, the grammar is not LL(1).