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

\[
\begin{align*}
S & \rightarrow A$ \\
A & \rightarrow xAx \\
A & \rightarrow C \\
B & \rightarrow yBy \\
B & \rightarrow C \\
C & \rightarrow zBz \\
C & \rightarrow wAw \\
C & \rightarrow \lambda
\end{align*}
\]

(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:**

\[
\begin{align*}
S & \rightarrow A$ \\
& \rightarrow xAx$ \\
& \rightarrow xCz$ \\
& \rightarrow xzBzz$ \\
& \rightarrow xzCzz$ \\
& \rightarrow zzzx$
\end{align*}
\]

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:**

\[
\begin{align*}
First(S) &= \{x, z, w, \$\} \\
First(A) &= \{x, z, w, \lambda\} \\
First(B) &= \{y, z, w, \lambda\} \\
First(C) &= \{z, w, \lambda\}
\end{align*}
\]
Follow($S$) = $\{\}$
Follow($A$) = $\{\$, $x, w\}$
Follow($B$) = $\{y, z\}$
Follow($C$) = $\{\$, $x, y, z, w\}$

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

Answer:

$\text{Predict}(1)$ = $\{x, z, w, \$\}$
$\text{Predict}(2)$ = $\{x\}$
$\text{Predict}(3)$ = $\{z, w, x, \$\}$
$\text{Predict}(4)$ = $\{y\}$
$\text{Predict}(5)$ = $\{z, w, y\}$
$\text{Predict}(6)$ = $\{z\}$
$\text{Predict}(7)$ = $\{w\}$
$\text{Predict}(8)$ = $\{\$, $x, y, z, w\}$

(e) Is this an LL(1) grammar? Why or why not?

Answer:

This is not an LL(1) grammar, because there are conflicts between the predict sets. For example, there are two productions for $C$ that both have $w$ in their predict set: $7$ & $8$.

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

\[
\begin{align*}
S & \rightarrow \ AB\$ \\
A & \rightarrow \ xA \\
A & \rightarrow \ B \\
B & \rightarrow \ yzB \\
B & \rightarrow \ z
\end{align*}
\]

(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:**

```
  S
 / \   \
/     \ /   \
A     B $ \\
 \   /   \  \\
 \ /     \ \\
/  \     \ \\
A x A z B \\
/   /   / \
B y z B \\
/   /   \\
  y z
```

(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:**

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

3
(e) Is this an LL(1) grammar?

**Answer:**
This is an LL(1) grammar, as there are no conflicts between predict sets.

(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:

\[
\begin{align*}
\text{First}(S) &= \{x, y, z\} \\
\text{First}(A) &= \{x, y, z, \lambda\} \\
\text{First}(B) &= \{y, z\}
\end{align*}
\]

Note that the First set of $A$ changed.

\[
\begin{align*}
\text{Follow}(S) &= \{} \\
\text{Follow}(A) &= \{y, z\} \\
\text{Follow}(B) &= \{y, z, \$\}
\end{align*}
\]

Note that none of the follow sets changed!

\[
\begin{align*}
\text{Predict}(1) &= \{x, y, z\} \\
\text{Predict}(2) &= \{x\} \\
\text{Predict}(3) &= \{y, z\} \\
\text{Predict}(4) &= \{y\} \\
\text{Predict}(5) &= \{z\} \\
\text{Predict}(6) &= \{y, z\}
\end{align*}
\]

But the predict set for rule 6 is $\text{Follow}(A)$. This means there is now a conflict: when expanding an $A$, both rule 3 and rule 6 have $y$ and $z$ in their predict sets. 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 $\lambda$) using rule 6. Thus, the grammar is not LL(1).