1. (20 pts) In your own words, describe the differences in these parsing methods: \(\text{LL}(1)\), \(\text{LR}(0)\), \(\text{SLR}(1)\), and \(\text{LR}(1)\). Describe how they fail.

**Diffs**:
- \(\text{LL}(1)\): top down, left-most derivation
- \(\text{LR}(0)\): works using firsts + one lookahead
- \(\text{SLR}(1)\): all bottom up, right-most derivation, works on handles + shifting/reducing
- \(\text{LR}(1)\): not predictive (two prodps in one cell) / Needs to be left factored

**Fail**: \(\text{LL}(1)\) - not predictive (two prodps in one cell) / Needs to be left factored

LR(0), SLR(1), and LR(1) fail on shift/reduce + reduce/reduce errors. LR(0) can’t have anything else in a state w/ a reduce. SLR(1) fails if two reduces in state don’t have disjoint follow sets or shift is found in the follow set/same applies w/ LR(1) but using carry not follow set.

2. (20 pts) Using the grammar below, what is the handle for this right sentential form:

\((S + a)^* a\)

Why is it this?

- \(S \rightarrow S + S\)
- \(S \rightarrow S S\)
- \(S \rightarrow (S)\)
- \(S \rightarrow S^*\)
- \(S \rightarrow a\)

\(s \Rightarrow SS \Rightarrow Sa \Rightarrow S \times a \Rightarrow (S) \times a \Rightarrow (S + a)^* a \Rightarrow (S + a) \times a \Rightarrow (S + a)^* a\)

This \(a\) reduces to \(S\) to create a valid right sentential form.
3. (20 pts) Determine if the following grammar is LL(1) by using the First and Follow to build a parse table. State the reason for your conclusion.

\[ E \rightarrow [L] \quad \text{First}(E) = \{ \varepsilon, a \} \quad \text{Follow}(E) = \{ \varepsilon, \$, \} \]

\[ E \rightarrow a \quad \text{First}(L) = \{ L \} \quad \text{Follow}(L) = \{ L \} \]

\[ L \rightarrow ET \quad \text{First}(T) = \{ \varepsilon \} \quad \text{Follow}(T) = \{ \varepsilon \} \]

\[ T \rightarrow , L \]

\[ T \rightarrow \varepsilon \]

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>J</td>
<td>a</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Yes, LL(1) - each cell has only one production.

+4 for correct contents.
4. (20 points) Is the following grammar LR(0) or SLR(1)? Why or why not?

1. \( S \rightarrow bA \)
2. \( S \rightarrow b \)
3. \( A \rightarrow bb \)
4. \( A \rightarrow bc \)

\[
\begin{align*}
\text{I}_0 & \quad S \rightarrow S. \\
\text{I}_1 & \quad S \rightarrow bA. \\
\text{I}_2 & \quad S \rightarrow b. \\
\text{I}_3 & \quad A \rightarrow bb. \\
\text{I}_4 & \quad A \rightarrow bc.
\end{align*}
\]

Correct states with closures 5 pts.

\( \text{Follow}(S) = \{ \text{\$} \} \)

Shift on \( b \) or \( R2 \) on $.

Yes, SLR(1) \( \square \)

Not LR(0) because a reduce isn't in a state by itself. Shift/reduce conflict 5 pts.

States without closures 5 pts.

No yes 5 pts.
5. (20 pts) Is the following grammar LR(1)? Why or why not?

0. S → S + S
1. S → S * S
2. S → a

In 5, we will have a shift/reduce conflict because we will R11 in the cell with the shift on `+` and shift on `*`.

Carry symbols +5 pts
Correct states w/ closures +5 pts
Not LR(1) +5 pts