COMP 520: Compilers
Sample Solutions – Written Assignment 2

(Note: the rules applied in these solutions can be found in lecture 4, slides 6 – 19)

1. For the following grammars, find the Nullable, Starters, and Followers sets for the nonterminals and justify whether or not the grammar meets the LL(1) condition.

Grammar (a)  
S ::= A B c  
A ::= a B | ε  
B ::= b | ε

First determine the grammar properties by finding the fixpoint of the relevant equations.

1. Nullable
   \[ N_0(S) = N_0(A) = N_0(B) = \text{false} \]
   \[ N_{i+1}(S) = N_i(A) \land N_i(B) \land N(c) = N_i(A) \land N_i(B) \land \text{false} = \text{false} \]
   \[ N_{i+1}(A) = (N(a) \land N_i(B)) \lor N(\varepsilon) = (\text{false} \land N_i(B)) \lor \text{true} = \text{true} \]
   \[ N_{i+1}(B) = N(b) \lor N(\varepsilon) = \text{false} \lor \text{true} = \text{true} \]

   No iteration is needed since the equations are at the fixpoint solution.

2. Starters
   \[ ST_0(S) = \{ \}, \quad ST_0(A) = ST_0(B) = \{ \varepsilon \} \]
   \[ ST_{i+1}(S) = ST_i(A) \oplus ST_i(B) \oplus ST(c) = (ST_i(A) - \{ \varepsilon \}) \cup (ST_i(B) - \{ \varepsilon \}) \cup \{ c \} \]
   \[ ST_{i+1}(A) = (ST(a) \oplus ST_i(B)) \cup ST(\varepsilon) = \{ a \} \cup \{ \varepsilon \} = \{ a, \varepsilon \} \text{ (fixpoint)} \]
   \[ ST_{i+1}(B) = ST(b) \cup ST(\varepsilon) = \{ b \} \cup \{ \varepsilon \} = \{ b, \varepsilon \} \text{ (fixpoint)} \]

   Iterate: \[ ST_1(S) = \{ c \}, \quad ST_2(S) = \{ a, b, c \} \text{ (fixpoint)} \]

3. Followers
   \[ FL_0(S) = \{ \} \quad \text{S not on RHS of any rule} \]
   \[ FL_0(A) = ST(Bc) - \{ \varepsilon \} = \{ b, \varepsilon \} \oplus ST(c) = \{ b, c \} \quad \text{A on RHS rule 1} \]
   \[ FL_0(B) = (ST(c) \cup ST(\varepsilon)) - \{ \varepsilon \} = \{ c \} \quad \text{B on RHS rule 1,2} \]

   Since A ::= a B in rule 2 is of the form A ::= a By, where γ is nullable, we must add the followers of A into the followers of B:
   \[ FL_{i+1}(B) = FL_i(B) \cup FL_i(A) = \{ c \} \cup \{ b, c \} = \{ b, c \} \]

   As FL_1(B) is fully defined, no further iteration is needed and \[ FL(B) = FL_1(B) = \{ b, c \}. \]

   Finally we need to add ε to FL(N) for any nonterminal N that can appear as the last symbol in a derivation starting from S. As this is an augmented grammar in which (a) the start symbol S does not appear on the RHS of any rule and (b) the rule for S ends in a terminal (c in this case), we can be sure only FL(S) includes \{ ε \}. So \[ FL(S) = \{ \varepsilon \}. \]
In summary we have

<table>
<thead>
<tr>
<th></th>
<th>Nullable</th>
<th>Starters</th>
<th>Followers</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>F</td>
<td>{a, b, c}</td>
<td>{ε}</td>
</tr>
<tr>
<td>A</td>
<td>T</td>
<td>{a, ε}</td>
<td>{b, c}</td>
</tr>
<tr>
<td>B</td>
<td>T</td>
<td>{b, ε}</td>
<td>{b, c}</td>
</tr>
</tbody>
</table>

To determine whether the LL(1) condition is met for this grammar, inspect all choice points to determine if they define the predict function unambiguously. There are only two choice points: the alternatives in the second and third rules.

- **A ::= aB | ε**
  - Predict(aB) = Starters(aB) ⊕ Followers(A) = \{a\}
  - Predict(ε) = Starters(ε) ⊕ Followers(A) = \{b, c\}
  - The predict sets are disjoint, so the LL(1) condition is met.

- **B ::= b | ε**
  - Predict(b) = Starters(b) ⊕ Followers(B) = \{b\}
  - Predict(ε) = Starters(ε) ⊕ Followers(B) = \{b, c\}
  - The predict sets are not disjoint, so the LL(1) condition is not met.

Since the LL(1) condition is not met by at least one choice point in the grammar, this grammar is not LL(1).

**Grammar (b)**

\[
\begin{align*}
S & ::= A \$
A & ::= a A x | B \\
B & ::= b B x | D \\
D & ::= d D x | f
\end{align*}
\]

First determine the grammar properties by finding the fixpoint of the relevant equations.

1. **Nullable**
   \[
   \begin{align*}
   N_0(S) &= N_0(A) = N_0(B) = N_0(D) = false \\
   N_{i+1}(S) &= N_i(A) \land N(\$) = false \\
   N_{i+1}(A) &= N(a A x) \lor N_i(B) = N_i(B) \\
   N_{i+1}(B) &= N(b B x) \lor N_i(D) = N_i(D) \\
   N_{i+1}(D) &= N(d D x) \lor N(f) = false
   \end{align*}
   \]
   After two iterations we have a fixpoint \(N(S) = N(A) = N(B) = N(D) = false\)

2. **Starters**
   \[
   \begin{align*}
   ST_0(S) &= ST_0(A) = ST_0(B) = ST_0(D) = \{\} \text{ since no nullable NTs in this grammar} \\
   ST_{i+1}(S) &= ST_i(A) \oplus ST(\$) = ST_i(A) \\
   ST_{i+1}(A) &= ST(a A x) \cup ST_i(B) = \{a\} \cup ST_i(B) \\
   ST_{i+1}(B) &= ST(b B x) \cup ST_i(D) = \{b\} \cup ST_i(D) \\
   ST_{i+1}(D) &= ST(d D x) \cup ST(f) = \{d, f\}
   \end{align*}
   \]
Iterate to fixpoint:
\[ ST(S) = \{a, b, d, f\}, \quad ST(A) = \{a, b, d, f\}, \quad ST(B) = \{b, d, f\}, \quad ST(D) = \{d, f\} \]

3. Followers

\[ FL_0(S) = \{\} \quad \text{S not on RHS of any rule} \]
\[ FL_0(A) = \{x, $\} \quad \text{A on RHS rule 2 and rule 1} \]
\[ FL_0(B) = \{x\} \quad \text{B on RHS rule 3, and at end of rule 2 RHS} \]
\[ FL_0(D) = \{x\} \quad \text{D on RHS rule 4 and at end of rule 3 RHS} \]

In this grammar B and D occur at the RHS of a rules 2 and 3, respectively. Thus we have to add the follow of their respective LHS, i.e.

\[ FL_{i+1}(B) = FL_i(B) \cup FL_i(A) \]
\[ FL_{i+1}(D) = FL_i(D) \cup FL_i(B) \]

and iterate to fixpoint. This yields \( FL(B) = FL(D) = \{x, $\} \).

Finally we need to add \( \varepsilon \) to \( FL(N) \) for any nonterminal \( N \) that can appear as the last symbol in a derivation starting from \( S \). As this is an augmented grammar, we can be sure only \( FL(S) \) includes \( \{\varepsilon\} \). So \( FL(S) = \{\varepsilon\} \).

In summary we have

<table>
<thead>
<tr>
<th></th>
<th>Nullable</th>
<th>Starters</th>
<th>Followers</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>F</td>
<td>{a, b, d, f}</td>
<td>{\varepsilon}</td>
</tr>
<tr>
<td>A</td>
<td>F</td>
<td>{a, b, d, f}</td>
<td>{x, $}</td>
</tr>
<tr>
<td>B</td>
<td>F</td>
<td>{b, d, f}</td>
<td>{x, $}</td>
</tr>
<tr>
<td>D</td>
<td>F</td>
<td>{d, f}</td>
<td>{x, $}</td>
</tr>
</tbody>
</table>

To determine whether the LL(1) condition is met for this grammar, inspect all choice points to determine if they define the predict function unambiguously. There are three choice points: the

- **A ::= a A x | B**
  
  \[ \text{Predict(a A x) = \{}a\} } \]
  
  \[ \text{Predict(B) = ST(B) = \{b, d, f\} } \]

  The predict sets are disjoint, so the LL(1) condition is met.

- **B ::= b B x | D**
  
  \[ \text{Predict(b B x) = \{}b\} } \]
  
  \[ \text{Predict(D) = Starters(D) = \{d, f\} } \]

  The predict sets are disjoint, so the LL(1) condition is met.

- **D ::= d D x | f**
  
  \[ \text{Predict(d D x) = \{}d\} } \]
  
  \[ \text{Predict(f) = \{}f\} } \]

  The predict sets are disjoint, so the LL(1) condition is met.

Since the LL(1) condition is met at all choice points in the grammar, this grammar is LL(1).
Grammar (c)  
\[ S ::= A \ c \]

\[ A ::= a \ A^* \ b \mid b \]

The Nullable and Starters functions are straightforward to compute. In this grammar \( A \) appears on the RHS in rule 1 where it is followed by \( c \), and it appears on the RHS in rule 2 where it is either followed by \( b \) or by \( ST(A) \) (since \( A^* \) can yield consecutive occurrences of \( A \)). So we have

\[ FL_0(S) = \{ \} \]

\[ FL_0(A) = (\{c\} \cup ST(A^* b)) - \{\varepsilon\} = \{c\} \cup \{a, b\} \cup \{b\} = \{a, b, c\} \]

Since every occurrence of \( A \) in a rule is followed by a terminal, there are no other followers to include. Since this grammar is in augmented form, \( FL(S) = \{\varepsilon\} \). In summary:

<table>
<thead>
<tr>
<th></th>
<th>Nullable</th>
<th>Starters</th>
<th>Followers</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S )</td>
<td>F</td>
<td>{a, b}</td>
<td>{\varepsilon}</td>
</tr>
<tr>
<td>( A )</td>
<td>F</td>
<td>{a, b}</td>
<td>{a, b, c}</td>
</tr>
</tbody>
</table>

LL(1) Condition: The first rule has no choice points. There are two choice points in the second rule:

- \( a \ A^* \ b \mid b \)
  
  Predict(\( a \ A^* \ b \)) = \{a\} and Predict(b) = \{b\}
  
  The predict sets are disjoint, so the LL(1) condition is met at this choice point.

- \( A^* \ b \)
  
  First we need to check that \( A \) is not nullable - this is the case from the table above. Second we need to be able to predict repetition or termination of the Kleene star. For repetition: Predict(\( A \)) = Starters(\( A \)) = \{a, b\}. To predict end of the repetition: Predict(b) = \{b\}. The predict sets are not disjoint, so the LL(1) condition is not met at this choice point.

Since the LL(1) condition is not met by at least one choice point in the grammar, this grammar is not LL(1).