CS 4114
Solutions to Final Exam
May 9, 2000

[80] 1. Let \( L_1 \) be the language
\[
L_1 = b^+ a^* \{ a^i b^i | i \geq 9 \} (ab)^+.
\]

Determine which of the following is true:

1. \( L_1 \) is regular;
2. \( L_1 \) is context-free, but not regular; or
3. \( L_1 \) is not context-free, but is r.e.

Prove your answer.

Choice 2 is correct.

To show that \( L_1 \) is context-free, we can give a context-free grammar to generate it or a PDA to accept it. Here is a CFG that generates \( L_1 \):

\[
S \rightarrow A B a a a a a a a a C b b b b b b b b D
\]

\[
A \rightarrow b B | b
\]

\[
B \rightarrow B a | \lambda
\]

\[
C \rightarrow a C b | \lambda
\]

\[
D \rightarrow a b D | a b.
\]

To show that \( L_1 \) is not regular, it suffices to show that
\[
L'_1 = b a^+ b^+ a b \cap L_1
\]

\[
= b a^* \{ a^i b^i | i \geq 9 \} a b
\]

is not regular; here we are using the fact that the class of regular languages is closed under intersection. To show that \( L'_1 \) is not regular, we can give a pumping lemma argument, as follows.

Let \( k > 0 \) be arbitrary. Let \( z = b a^k b^{k+9} a b \). Then \( z \in L'_1 \) and \( |z| \geq k \).

Let \( z = uvw \) be any decomposition of \( z \) satisfying \(|v| \geq 1 \) and \(|uv| \leq k \). There are two cases to consider, depending on whether \( u = \lambda \) or not.

If \( u = \lambda \), then \( v = b a^s \) where \( 0 \leq s \leq k - 1 \). As \( uv^0 w = a^{k+9-s} b^k a b \) does not begin with a \( b \), it fails to be in \( L'_1 \).

If \( u \neq \lambda \), then \( u = b a^r \) and \( v = a^s \), where \( 0 \leq r \leq k - 1 \) and \( 1 \leq s \). Then \( uv^0 w = b a^{k+9-s} b^{k+9} a b \notin L'_1 \), as there are \((k + 9)\) \( b \)'s in the second block of \( b \)'s, but only \((k + 9 - s)\) \( a \)'s in the first block of \( a \)'s.

As \( L'_1 \) does not satisfy the conclusion of the pumping lemma, it cannot be regular. We conclude that \( L_1 \) is not regular as well.
[90] 2. Let $G_2$ be the context-free grammar:

$$
S = ABA## | aaB## \\
A \rightarrow aB | bA \\
B \rightarrow bB | ba
$$

and let $L_2$ be the language generated by $G_2$.

1. Give a PDA that parses strings generated by $G_2$.

2. Prove that $G_2$ is not a strong LL$_1$ grammar.

3. Is $G_2$ strong LL$_2$? Justify your answer.

---

1. The PDA having the state diagram in Figure 1 is one that can parse according to $G_2$. State $q_1$ is drawn with an enormous circle to emphasize its central decision role in the parser. Note that we do not need to include the end makers (#) in the parser, though it is fine to do so, if you wish.

2. While we could compute all the lookahead sets ($LA_1$) for all the rules and show how the definition of a strong LL$_1$ grammar is not satisfied by $G_2$, it is easy to find my inspection that

$$
LA_1(B \rightarrow bB) = \{b\}
$$

and

$$
LA_1(B \rightarrow ba) = \{b\}.
$$

Since the $LA_1$ sets for the two $B$ rules are not disjoint, we know that $G_2$ cannot be strong LL$_1$.

3. To determine whether $G_2$ is strong LL$_2$, we need to determine the $LA_2$ sets for all the rules. As demonstrated in class and in the book, we first need to compute all the FIRST$_2$ sets for all the nonterminals by Algorithm 16.5.1. The iterations of Algorithm 16.4.1 yield these results:

<table>
<thead>
<tr>
<th></th>
<th>$F(S)$</th>
<th>$F(A)$</th>
<th>$F(B)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>1</td>
<td>${aa}$</td>
<td>$\emptyset$</td>
<td>${ba}$</td>
</tr>
<tr>
<td>2</td>
<td>${aa}$</td>
<td>${ab}$</td>
<td>${ba, bb}$</td>
</tr>
<tr>
<td>3</td>
<td>${aa, ab}$</td>
<td>${ab, ba}$</td>
<td>${ba, bb}$</td>
</tr>
<tr>
<td>4</td>
<td>${aa, ab, ba}$</td>
<td>${ab, ba, bb}$</td>
<td>${ba, bb}$</td>
</tr>
<tr>
<td>5</td>
<td>${aa, ab, ba, bb}$</td>
<td>${ab, ba, bb}$</td>
<td>${ba, bb}$</td>
</tr>
</tbody>
</table>
Figure 1: A PDA to parse according to $G_2$. 
Hence

\[
\text{FIRST}_2(S) = \{aa, ab, ba, bb\} \\
\text{FIRST}_2(A) = \{ab, ba, bb\} \\
\text{FIRST}_2(B) = \{ba, bb\}.
\]

Since all strings in all the \text{FIRST}_2 sets have length \(k = 2\), we can compute the \text{L.A}_2 sets for all the rules directly, without computing the \text{FOLLOW}_2 sets first. We obtain

\[
\text{FIRST}_2(S \rightarrow ABA##) = \text{FIRST}_2(A) = \{ab, ba, bb\} \\
\text{FIRST}_2(S \rightarrow aaB##) = \text{FIRST}_2(S \rightarrow ABA##) = \text{FIRST}_2(A) = \{ab\} \\
\text{FIRST}_2(A \rightarrow aB) = \text{FIRST}_2(S \rightarrow aaB##) = \text{FIRST}_2(S \rightarrow ABA##) = \text{FIRST}_2(A) = \{ab\} \\
\text{FIRST}_2(A \rightarrow bA) = \text{FIRST}_2(S \rightarrow AA##) = \text{FIRST}_2(S \rightarrow A\#\#) = \text{FIRST}_2(A) = \{ab\} \\
\text{FIRST}_2(B \rightarrow bB) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B) = \{bb\} \\
\text{FIRST}_2(B \rightarrow ba) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B) = \{bb\} \\
\text{FIRST}_2(B \rightarrow ba) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B \rightarrow b\#) = \text{FIRST}_2(B) = \{bb\}.
\]

Since the \text{FIRST}_2 sets for the two \(S\) rules are disjoint, the \text{FIRST}_2 sets for the two \(A\) rules are disjoint, and the \text{FIRST}_2 sets for the two \(B\) rules are disjoint, we know that \(G_2\) is strong LL_2.
Let $M_3$ be the Turing machine with input alphabet $\Sigma_3 = \{0, 1\}$ and state diagram

1. What language $L_h$ does $M_3$ accept by halting?
2. What language $L_f$ does $M_3$ accept by final state?
3. Determine whether $L_h$ is context-free.
4. Determine whether $L_f$ is context-free.

1. Clearly $M_3$ scans every input symbol at least once and halts (in $q_1$ or $q_2$) if a 1 is encountered. Hence $(0 \cup 1)^* 1 (0 \cup 1)^* \subseteq L_h$.

   If the input is $0^i$, then $M_3$ alternately erases a 0 on the left and a 0 on the right until no 0's remain. After all the input has been erased, $M_3$ will halt, either in $q_3$ or $q_4$. Hence $M_3$ accepts every input by halting. In other words,

   $$L_h = \Sigma_3^*.$$

2. From the previous analysis, we know that $M_3$ halts in state $q_4$ (the only final state) if and only if the input was a string of 0's of even length. Hence,

   $$L_f = \{0^{2i} \mid i \geq 0\}.$$

3. $L_h$ is context-free, indeed regular, as it can be represented by a regular expression:

   $$L_h = (0 \cup 1)^*.$$

4. $L_f$ is context-free, indeed regular, as it can be represented by a regular expression:

   $$L_f = (00)^*.$$