Who Graded What

Hong Lu: #1, #2, #6
Amit Chakrabarti: #3, #4, #5
Vibhor Bhatt: #7, #8

The Solutions

1. Write a regular expression for the language generated by the following grammar:

\[
S \rightarrow AT \\
T \rightarrow ABT | TBA | AA \\
A \rightarrow 0 \\
B \rightarrow 1
\]

A single line answer will do; you don’t have to justify or show any steps. Your regular expression should be as simple as possible.

**Solution:** The grammar generates \(0(01)^*00(10)^*\).

2. Draw a DFA for the language

\[\{x \in \{0, 1\}^*: x \text{ contains an equal number of occurrences of the substrings 01 and 10}\}\]

For example, 101 and 0000 are in the language, but 1010 is not.

**Solution:** The idea is to handle strings beginning with a 1 and strings beginning with a 0 separately. The following DFA does the job:

![DFA Diagram]

3. Recall that \(x^R\) denotes the reverse of the string \(x\). For a language \(L\), let \(L^R = \{x^R : x \in L\}\). Give a complete formal proof that if \(L\) is regular, so is \(L^R\).

**Solution:** The idea is to start with a DFA for the regular language \(L\) and “reverse all the arrows” in this DFA, make its start state an accept state of the resulting machine, and make all its accept states start states (or rather, since only one start state is allowed, to introduce
4. Consider two languages \(A, B \subseteq \Sigma^*\). Prove that \((A^*B^*)^* = (A \cup B)^*\). Remember that to prove \(X = Y\) for sets \(X\) and \(Y\) you must separately prove \(X \subseteq Y\) and \(Y \subseteq X\).

**First Solution:** Suppose \(x \in (A^*B^*)^*\). By definition of Kleene star, \(x\) is a concatenation of zero or more strings, each in \(A^*B^*\), i.e. \(x = x_1x_2 \ldots x_n\) with each \(x_i \in A^*B^*\). Again, by definition, each \(x_i = y_{i1}y_{i2} \ldots y_{is_i}z_{i1}z_{i2} \ldots z_{it_i}\) with each \(y_{ij} \in A\) and each \(z_{ij} \in B\). Putting it all together:

\[
x = y_{11} \ldots y_{1s_1}z_{11} \ldots z_{1t_1}y_{21} \ldots y_{2s_2}z_{21} \ldots z_{2t_2} \ldots \ldots \ldots \ldots z_{nt_n}.
\]

Since each \(y_{ij}\) and each \(z_{ij}\) is in \(A \cup B\), it follows that \(x \in (A \cup B)^*\). We have shown that \((A^*B^*)^* \subseteq (A \cup B)^*\).
Now, suppose \( x \in (A \cup B)^* \). Then \( x = x_1x_2\ldots x_n \) where each \( x_i \in A \cup B \). If an \( x_i \in A \), we can write \( x_i = x_i \varepsilon \) which puts it in \( A^*B^* \). If an \( x_i \in B \), we can similarly write \( x_i = \varepsilon x_i \) which puts it in \( A^*B^* \). Therefore, each \( x_i \in A^*B^* \), whence \( x \in (A^*B^*)^* \). We have shown that \( (A \cup B)^* \subseteq (A^*B^*)^* \).

This completes the proof.

**Second Solution:** Since \( A^* \subseteq (A \cup B)^* \) and \( B^* \subseteq (A \cup B)^* \), we have

\[
A^*B^* \subseteq (A \cup B)^*(A \cup B)^* = (A \cup B)^*.
\]

Applying the Kleene star to both sides, we get \((A^*B^*)^* \subseteq ((A \cup B)^*)^* = (A \cup B)^*\).

Since \( A \subseteq A^* \subseteq A^*B^* \) and \( B \subseteq B^* \subseteq A^*B^* \), we have \( A \cup B \subseteq A^*B^* \). Applying the Kleene star to both sides, we get \((A \cup B)^* \subseteq (A^*B^*)^*\).

5. **Prove that there exist languages \( A, B, C \subseteq \{0, 1\}^* \) that satisfy all of the following properties:**

(a) \( A = B \cap C \).

(b) \( B \) and \( C \) are both non-regular.

(c) \( A \) is infinite and regular.

To get any credit, you **must** prove all three properties for whatever \( A, B, C \) you have decided to use.

**Solution:** Define the sets \( X = \{0^n : n \geq 0\} \), \( Y = 0^* - X \) and \( A = 11^* \). Note that, by construction, any two of \( X, Y \) and \( A \) are disjoint; so a Venn diagram of these three sets would look like this:

![Venn Diagram](image)

Define \( B = X \cup A \), \( C = Y \cup A \). By the disjointness observed above (see diagram), condition (a) clearly holds. Since \( A = 11^* \) is regular, condition (c) also holds. The only nontrivial thing is to establish condition (b). Let us use bars to denote complements of languages with respect to \( \{0, 1\}^* \). Then, by the disjointness conditions (see diagram), \( X = B - A = B \cap \overline{A} \). Similarly, \( Y = C \cap \overline{A} \). Now we shall establish condition (b) using a proof by contradiction.

First, suppose \( B \) is regular. Then, by closure of regular languages under complement and intersection, \( X = B \cap \overline{A} \) would also be regular. But we've prove in class that \( X \) is not regular, so we have a contradiction. Thus, \( B \) must be non-regular. Next, suppose \( C \) is regular. Again, by closure properties, \( Y = C \cap \overline{A} \) would be regular. The complement of \( Y \) with respect to \( 0^* \) is \( X \), so \( X \) must also be regular and again we have a contradiction. Thus, \( C \) must be non-regular.

6. **A permutation of a string \( x \) is any string that can be obtained by rearranging the characters of \( x \). Thus, for example, the string \( abc \) has exactly six permutations:**

\[
abc, \text{acb, bac, bca, cab, cba}.
\]
Clearly, if $y$ is a permutation of $x$, then $|y| = |x|$. For a language $L$ over alphabet $\Sigma$, define

$$\text{PERMUTE}(L) = \{ x \in \Sigma^* : x \text{ is a permutation of some string in } L \},$$

$$\text{SELECT}(L) = \{ x \in \Sigma^* : \text{every permutation of } x \text{ is in } L \}.$$ 

Classify each of the following statements as \text{TRUE} or \text{FALSE}, and give proofs justifying your classifications.

6.1. If $L_1 = 1^*0$, then $\text{PERMUTE}(L_1)$ is regular.
\textbf{Solution:} \text{TRUE}. In this case $\text{PERMUTE}(L_1) = \{ x \in \{0,1\}^* : x \text{ contains exactly one } 0 \} = 1^*01^*$, which is clearly regular.

6.2. If $L_2 = 0^*1^*$, then $\text{SELECT}(L_2)$ is regular.
\textbf{Solution:} \text{TRUE}. We claim that $\text{SELECT}(L_2) = 0^* \cup 1^*$, which is clearly regular. To prove this, we first note that any string in $0^* \cup 1^*$ has only one permutation (namely, itself) and that belongs to $L_2$; therefore $0^* \cup 1^* \subseteq \text{SELECT}(L_2)$. We then note that any string $x \in L_2 = (0^* \cup 1^*)$ contains at least one 0 and at least one 1, so one of its permutations must contain the substring “10.” This permutation clearly does not belong to $L_2$, so $x \notin \text{SELECT}(L_2)$. Therefore $\text{SELECT}(L_2) \subseteq 0^* \cup 1^*$.

6.3. Regular languages are closed under the operation \text{PERMUTE}.
\textbf{Solution:} \text{FALSE}. For a counterexample, consider $L = (01)^*$. Any string in $L$ contains an equal number of 0’s and 1’s, so, when we take all permutations of all strings in $L$, we get precisely the language $\{ x \in \{0,1\}^* : x \text{ contains as many } 0\text{'s as } 1\text{'s}\}$. We have proved in class, using the pumping lemma, that this language is not regular. Thus $\text{PERMUTE}(L)$ need not be regular even if $L$ is.

6.4. Regular languages are closed under the operation \text{SELECT}.
\textbf{Solution:} \text{FALSE}. For a counterexample, consider $L' = (01)^*$, with the complement being taken with respect to alphabet $\{0,1\}$. Clearly $L'$ is regular. If a string $x$ has an unequal number of 0’s and 1’s, then no matter how we permute it we will never land in the set $(01)^*$, i.e., every permutation of $x$ is in $L'$, i.e., $x \in \text{SELECT}(L')$. On the other hand, if a string $x$ has as many 0’s as 1’s, then it can be permuted into the form $(01)^*$, so $x \notin \text{SELECT}(L')$. In short, we have shown that $\text{SELECT}(L') = \{ x \in \{0,1\}^* : x \text{ contains an unequal number of } 0\text{'s and } 1\text{'s}\}$. Thus, $\text{SELECT}(L')$ is the complement of a non-regular language, whence it is itself non-regular.

7. Draw a PDA for the language $\{0^i1^j : i < j < 2i\}$. For clarity, keep your stack alphabet disjoint from $\{0,1\}$. Provide a brief justification (no need for a formal proof) that your PDA works correctly.
\textbf{Solution:} Let $L$ be the given language. First, let us try to build a PDA for a slightly easier language: $L' = \{0^i1^j : i \leq j \leq 2i\}$. The idea is to push either one or two a’s (choose nondeterministically) onto our stack as we read the 0’s in the input string, and then pop one symbol at a time as we read the 1’s. If we empty the stack at the same time as when we finish reading the input, we may accept.

But for the language $L$, we must reject strings of the form $0^i1^i$ and $0^i1^{2i}$. To do so, observe that the shortest string in $L$ is 00111 and that any string in $L$ is therefore of the form

$$0^i0^i1^j1^j111$$

for some $i' \geq 0$, $j' \geq 0$, and $(i' + 2) < (j' + 3) < 2(i' + 2)$. Since $i'$ and $j'$ are integers, the latter condition is equivalent to $i' \leq j' \leq 2i'$. Thus, strings in $L$ are simply strings in $L'$ with two 0’s prepended and three 1’s appended. With this in mind, we build our PDA:
8. Design a context-free grammar for the complement of the language \( \{a^n b^n : n \geq 0\} \) over the alphabet \( \{a, b\} \). Give brief explanations for the “meanings” of your variables (i.e. explain what strings are generated by each of your variable).

**Solution:** A string in the complement of \( \{a^n b^n : n \geq 0\} \) is either of the form \( \{a^i b^j : i > j \geq 0\} \) or of the form \( \{a^i b^j : j > i \geq 0\} \) or not of the form \( a^* b^* \), which means that it contains the substring \( ba \). In the following grammar, the variable \( U \) generates strings of the third type, while \( T \) generates strings of the form \( a^n b^n \) to which we either prepend a string of one or more \( a \)'s (generated by \( A \)) or append a string of one or more \( b \)'s (generated by \( B \)).

\[
\begin{align*}
S & \rightarrow \quad AT \mid TB \mid U \\
A & \rightarrow \quad Aa \mid a \\
B & \rightarrow \quad Bb \mid b \\
T & \rightarrow \quad aTb \mid \varepsilon \\
U & \rightarrow \quad VbaV \\
V & \rightarrow \quad Va \mid Vb \mid \varepsilon
\end{align*}
\]

**Another Cute Solution:** The following cute CFG, suggested by Jason Reeves ’07, also happens to work (proof left as an exercise to the reader):

\[
\begin{align*}
A & \rightarrow \quad B \mid aAb \\
B & \rightarrow \quad a \mid b \mid aCa \mid bCb \mid bCa \\
C & \rightarrow \quad \varepsilon \mid B \mid aCb
\end{align*}
\]