Assignment 2 — Solutions
Total marks: 90.

Out: June 24
Due: July 8 by 18:45 in the dropbox, or 19:00 in class

Note that:

• The assignment can be handwritten or typed. It must be legible.
• You must do this assignment individually.
• Submit this assignment only if you have read and understood the policy on academic honesty on the course web page. If you have questions or concerns, please contact the instructor.
• Use the dropbox near the main office to submit your assignments, or hand them in at the beginning of class (please note the times and day above). No late submissions will be accepted.

1. [20 points] If \( L_1 \) and \( L_2 \) are two languages over the alphabet \( \Sigma \), then we define \( L_1 \circ L_2 \) to be the language \( \{ x_1y_1x_2y_2 \ldots x_ny_n \mid n \geq 0, \text{ each } x_i \text{ and } y_i \text{ are in } \Sigma^*, x_1x_2 \ldots x_n \in L_1 \text{ and } y_1y_2 \ldots y_n \in L_2 \} \). That is, each string in the language \( L_1 \circ L_2 \) is formed by interleaving one string from \( L_1 \) and one string from \( L_2 \).

a) If \( L_1 = \{ \text{blue, red} \} \) and \( L_2 = \{010\} \), write down three strings that are in \( L_1 \circ L_2 \).

Solution: \( b0l1u0e, bl0u1e0, blue010, 010red, \ldots \)

b) If \( L_1 = \{1\}^* \) and \( L_2 = \{010\} \), write down a regular expression that describes \( L_1 \circ L_2 \).

Solution: \( 1*011^*01^* \) (or alternatively \( 1^*01^*101^* \))

c) Prove that for all regular languages \( L_1 \) and \( L_2 \), \( L_1 \circ L_2 \) is also regular. If your proof involves constructing an automaton for \( L_1 \circ L_2 \), give a formal specification of this automaton. Also say which strings take your automaton to each one of its states (you don’t have to prove this).

Solution: Since \( L_1 \) and \( L_2 \) are regular, there are DFAs \( M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1) \) and \( M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2) \) that recognize them. We construct a NFA \( N \) that recognizes \( L_1 \circ L_2 \). Let \( N = (Q, \Sigma, \delta, q_0, F) \) where
\[
Q = Q_1 \times Q_2
\]
\[
F = F_1 \times F_2
\]
2. [10 points] Prove that for any regular expressions $\alpha$ and $\beta$, $(\alpha \cup \beta)^* = (\alpha^* \circ \beta^*)^*$.

**Solution:**

$\Rightarrow$ Suppose that $w \in L(\alpha \cup \beta)^*$. Then $w = w_1w_2\ldots w_n$ for some $n \geq 0$, where each $w_i$ is either in $L(\alpha)$ or in $L(\beta)$. If $w_i \in L(\alpha)$ then $w_i \in L(\alpha^* \circ \beta^*)$ since $\epsilon \in L(\beta^*)$. Similarly if $w_i \in L(\beta)$ then $w_i \in L(\alpha^* \circ \beta^*)$ since $\epsilon \in L(\alpha^*)$. Thus $w = w_1w_2\ldots w_n \in L((\alpha^* \circ \beta^*)^*)$.

$\Leftarrow$ Suppose that $w \in L((\alpha^* \circ \beta^*)^*)$. Then $w = w_1w_2\ldots w_n$ for some $n \geq 0$, where each $w_i$ is in $L(\alpha^* \circ \beta^*)$. The latter implies that $w_i = x_iy_i$ where $x_i \in L(\alpha^*)$ and $y_i \in L(\beta^*)$. Clearly if $s \in L(\alpha^*)$ or $s \in L(\beta^*)$, then $s \in L((\alpha \cup \beta)^*)$. Thus every $x_i$ and $y_i$ is in $L((\alpha \cup \beta)^*)$. The concatenation of strings in $L((\alpha \cup \beta)^*)$ is also in $L((\alpha \cup \beta)^*)$. Thus every $w_i$ is in $L((\alpha \cup \beta)^*)$ and $w$ is also in $L((\alpha \cup \beta)^*)$.

3. [10 points] Prove that $\{a^nb^{2n} \mid n \geq 0\}$ is not regular.

**Solution:** By contradiction. Assume that $L = \{a^nb^{2n} \mid n \geq 0\}$ is regular. By the pumping lemma for regular languages there exists $p$ such that for any $s \in L$ such that $|s| \geq p$, $s = xyz$ and for all $i \geq 0$, $xy^iz \in L$, $|y| > 0$, and $|xy| \leq p$. Consider $s = a^pb^{2p}$. By the pumping lemma, $s = xyz$ and for all $i \geq 0$, $xy^iz \in L$, $|y| > 0$, and $|xy| \leq p$. Since $|xy| \leq p$, we have that $y = a^k$ for some $k$, $0 < k \leq p$. Thus by the pumping lemma $xxyyz = a^{p+k}b^{2p}$ must be in $L$. But $a^{p+k}b^{2p}$ is clearly not in $L$ as it does not contain twice as many as $bs$ (since $k > 0$), so we have a contradiction. Therefore $L$ is not regular.

4. [10 points] Prove that $\{a^nbaba^n+m \mid n, m \geq 0\}$ is not regular.

**Solution:** By contradiction. Assume that $L = \{a^nbaba^n+m \mid n, m \geq 0\}$ is regular. By the pumping lemma for regular languages there exists $p$ such that for any $s \in L$ such that $|s| \geq p$, $s = xyz$ and for all $i \geq 0$, $xy^iz \in L$, $|y| > 0$, and $|xy| \leq p$. Consider $s = a^pbaba^{2p}$. By the pumping lemma, $s = xyz$ and for all $i \geq 0$, $xy^iz \in L$, $|y| > 0$, and $|xy| \leq p$. Since $|xy| \leq p$, we have that $y = a^k$ for some $k$, $0 < k \leq p$. Thus by the pumping lemma $xxyyz = a^{p+k}baba^{2p}$ must be in $L$. But $a^{p+k}baba^{2p}$ is clearly not in $L$ as $2p \neq p + k + p$ (since $k > 0$), so we have a contradiction. Therefore $L$ is not regular.
5. [20 points] Let \( L_1 = \{a^i b^j c^k \mid i, j, k \geq 0, i = j \text{ or } j = k\} \).

a) Give a context free grammar for \( L_1 \). You don’t have to prove that your answer is correct.

**Solution:**

\[
\begin{align*}
S & \rightarrow AB \mid CD \\
A & \rightarrow aAb \mid \epsilon \\
B & \rightarrow cB \mid \epsilon \\
C & \rightarrow aC \mid \epsilon \\
D & \rightarrow bDc \mid \epsilon
\end{align*}
\]

b) Show that your grammar is ambiguous.

**Solution:** The string \( abc \) has two different leftmost derivations in the grammar above:

\[
\begin{align*}
S \Rightarrow AB & \Rightarrow aAbB \Rightarrow abB \Rightarrow abc \Rightarrow abc \\
S \Rightarrow CD & \Rightarrow aCD \Rightarrow aD \Rightarrow abDc \Rightarrow abc
\end{align*}
\]

6. [20 points] Consider the following CFG \( G \) over the alphabet \( \{a, b\} \):

\[
\begin{align*}
S & \rightarrow aB \mid bA \\
A & \rightarrow a \mid aS \mid BAA \\
B & \rightarrow b \mid bS \mid ABB
\end{align*}
\]

a) Show that \( ababba \in L(G) \).

**Solution:** We have the following derivation:

\[
\begin{align*}
S \Rightarrow aB \Rightarrow abS \Rightarrow abaB \Rightarrow ababS \Rightarrow ababA \Rightarrow ababba
\end{align*}
\]

b) Prove that \( L(G) \) is the set of all non-empty strings over the alphabet \( \{a, b\} \) that have an equal number of \( a \)'s and \( b \)'s.

**Solution:**

We need to prove both directions.

First we show that every string \( s \) in \( L(G) \) has an equal number of \( a \)'s and \( b \)'s and is non-empty.

We begin by observing that any string produced by \( S, A, \) or \( B \) is non-empty. The reason is that in any derivation step \( l \Rightarrow r \) in a derivation from these symbols, the right hand side \( r \) is at least as long as the left hand side \( l \) (in any derivation step we can only replace one of these non-terminals by a non-empty string that only contains only the symbols \( S, A, B, a, \) and \( b \)). So the strings produced along any derivation have a non-decreasing length. It follows
that any derived string is non-empty. (It is straightforward to prove this formally by induction on the length of derivations.)

Now let’s show that every string \( s \) in \( L(G) \) has an equal number of \( a \)s and \( b \)s by strong induction on the length of \( s \). Since \( S \) produces intermediate strings involving \( A \) and \( B \), we need to prove assertions about the strings produced by those as well. We will do simultaneous induction on all these. The statement of the inductive hypothesis (IH) is: \( S \) produces strings that have an equal number of \( a \)s and \( b \)s, \( A \) produces strings that have one more \( a \) than \( b \)s (i.e. if the string contains \( k \) \( b \)s, then it contains \( k + 1 \) \( a \)s), and \( B \) produces strings that have one more \( b \) than \( a \)s.

Base case: Let \( |s| = 1 \). If \( s \) is generated by \( A \), it can only be \( a \), which has one more \( a \) than \( b \)s. Similarly, if \( s \) is generated by \( B \), it can only be \( b \), which has one more \( b \) than \( a \)s. No string of length 1 can be generated by \( S \).

Inductive step: Assume that the IH holds for all strings of length \( n \) or less. Consider a string \( s \) of length \( n + 1 \) produced by \( S \). If we use the rule \( S \to aB \), then \( s = aw \) where \( w \) is a string of length \( n \) generated by \( B \). By the IH, \( w \) is a string that has an equal number of \( a \)s and \( b \)s. Thus \( s \) is a string that has an equal number of \( a \)s and \( b \)s. Similarly, if we use the rule \( S \to bA \) to produce \( s \), then \( s = bw \) where \( w \) is a string of length \( n \) generated by \( A \). By the IH, \( w \) is a string that has one more \( a \) than \( b \)s. Thus \( s \) is a string that has one more \( a \) than \( b \)s. These are the only rules for \( S \). So the IH must hold for \( s \) of length \( n + 1 \) produced by \( S \).

Next, consider a string \( s \) of length \( n + 1 \) produced by \( A \). If \( s \) is produced by rule \( A \to a \), it is clear that \( s \) is a string that has one more \( a \) than \( b \)s. If \( s \) is produced by rule \( A \to aS \), then \( s = aw \) where \( w \) is a string of length \( n \) generated by \( S \). By the IH, \( w \) is a string that has an equal number of \( a \)s and \( b \)s. Thus \( s \) is a string that has one more \( a \) than \( b \)s. If \( s \) is produced by rule \( A \to BAA \), then \( s = xyz \) where \( x \) is generated by \( B \), and \( y \) and \( z \) are generated by \( A \). These must all be non-empty, so their lengths are at most \( n \). So by the IH, \( x \) is a string that has one more \( b \) than \( a \)s, and \( y \) and \( z \) are strings that have one more \( a \) than \( b \)s. Thus \( s \) is a string that has one more \( a \) than \( b \)s. These are the only rules for \( A \). So the IH must hold for \( s \) of length \( n + 1 \) produced by \( A \).

The proof for the case where \( s \) is generated by \( B \) is completely analogous. Therefore \( S \) produces strings that have an equal number of \( a \)s and \( b \)s, \( A \) produces strings that have one more \( a \) than \( b \)s, and \( B \) produces strings that have one more \( b \) than \( a \)s.

Now we show the other direction, i.e. that any non-empty string over the alphabet \( \{a, b\} \) that has an equal number of \( a \)s and \( b \)s is in \( L(G) \). We show this by strong induction on the length of the string. Again, we need assertions about
strings produced by $A$ and $B$ as well. The statement of the inductive hypothesis (IH) is: If a non-empty string $s$ has an equal number of $a$’s and $b$’s, it is produced by $S$, if it has one more $a$ than $b$s it is produced by $A$, and if it has one more $b$ than $as$ then it is produced by $B$.

Base case: $|s| = 1$. $A$ produces $a$, $B$ produces $b$, and there are no strings of length 1 with an equal number of $as$ and $bs$, so the claim holds.

Induction step: Assume the IH holds for all strings of length at most $n$. Consider a string $s$ of length $n + 1$ with an equal number of $a$'s and $b$'s. Either $s = aw$, where $w$ is a non-empty string that contains one more $b$ than $as$, or $s = bw$ where $w$ is a non-empty string that contains one more $a$ than $bs$. In both cases $w$ is of length $k$, so we can apply the IH. So if $s = aw$, by the IH $w$ is produced by $B$, and therefore $s$ is produced by $S$ using rule $A \to aB$. Similarly, if $s = bw$, by the IH $w$ is produced by $A$, and thus $s$ is produced by $S$ using rule $A \to bA$.

Consider a string $s$ of length $n + 1$ with one more $a$ than $bs$. Then either $s = a$, or $s = aw$ where $w$ is a non-empty string that has an equal number of $a$’s and $b$’s, or $s = xyz$ where $x$ is a non-empty string that contains one more $b$ than $as$ and $x$ and $z$ are non-empty strings that contain one more $a$ than $bs$. In the latter case, since $x$, $y$ and $z$ are non-empty, their length is less than $k$. So we can apply the IH and we have that $B$ produces $x$ and $A$ produces $y$ and $z$. Thus $A$ also produces $s$ using the rule $A \to BAA$. In the case where $s = a$, $s$ is produced by $A$ using the rule $A \to a$. In the case where $s = aw$ where $w$ is a non-empty string that has an equal number of $a$’s and $bs$, we have that $|w| = k$, so by the IH $S$ produces $w$. It follows that $A$ produces $s$ with the rule $A \to aS$. The proof that strings of length $n + 1$ with one more $b$ than $as$ are produced by $B$ is completely analogous.

Therefore every non-empty string that has an equal number of $as$ and $bs$ is produced by $S$, every non-empty string that has one more $a$ than $bs$ is produced by $A$, and every non-empty string that has one more $b$ than $as$ is produced by $B$. 