Midterm Exam

Instructions. This midterm is meant to be solved in 75 minutes. The start time is 12:30pm, and you are required to stop writing at 1:45pm (or earlier). When done, wait for a proctor to collect your solution, or follow any other instructions by the proctors.

- This exam consists of four tasks, overall with 60 points. The score will be normalized to be out of 30 points to have the same weight as a homework assignment.

- Write your name on every sheet of the exam.

- You are only allowed to use two letter-size sheets of hand written notes (4 pages). No other written document is allowed. In particular, you are not allowed to use any textbooks, nor additional personal notes, homework assignments or solutions, etc.

- No electronics are allowed during the exam (no smart phones, no laptops, no smart watches, no pocket calculators, etc). Keep them stored in your bag.

- Write your solutions in the appropriate spaces (additional empty space is available at the end of the exam). If you run out of paper, additional paper is available from the proctors.

- You are allowed to hand in before 1:45pm if you finish early but you are required to leave the room quietly in order not to disturb your classmates.
Task 1 – Short Questions

(15 points)

a) Indicate whether the following statements are true or false. No justification is necessary for your answer.

Every regular language is finite.  

true  false

If $L_1$ and $L_2$ are both regular, then $(L_1 \cup L_2) - (L_1 \cap L_2)$ is regular.

false  true

Every regular language can be generated by both a left- and a right-regular grammar.

true  false

Every regular language is context free.

false  true

There exist functions a Mealy Machine can compute, but a Moore Machine cannot compute.

false  true

Every subset of a regular language is regular.

true  false

For every language $L$, the concatenation $L\emptyset$ of $L$ and the empty language $\emptyset$ equals $\{\lambda\}$.

true  false

There exists an NFA which accepts the language $L = \{ww^R : w \in \{a, b\}^*\}$.

false  true

The union of two finite languages is regular.

true  false

The language $L = L(a^*b^*)$ with alphabet $\Sigma = \{a, b\}$ is context free.

true  false
b) Let \( L_1 = \{aab, aa\} \) and \( L_2 = \{a, b, ba, abb\} \) be languages with alphabet \( \Sigma = \{a, b\} \). Compute:

- \( L_1 L_2 \)

\[
L_1 L_2 = \{aaba, aabb, aabba, aababb, aab, aaabb\}.
\]

- \( L_1^R \)

\[
L_1^R = \{baa, aa\}.
\]

- \( L_2^2 L_1 \)

\[
L_2^2 L_1 = \{aa, ab, ba, bb, aba, baa, bab, bba, aabb, abba, abb, baba, \}
\]

\[
\quad \quad \quad \quad \quad babb, abbbba, baaabb, abbb\} \{aa, aab\}
\]

\[
\quad \quad = \{aaaa, abaa, baaa, bbaa, abaa, aaaa, \}
\]

\[
\quad \quad \quad \quad babaa, bbaaa, aabbaa, abbaaa, abbbbaa, babbaa, aabbaa, abbbbaa, \}
\]

\[
\quad \quad \quad \quad baabbaa, abbbbaaa, aabaa, abbaa, \}
\]

\[
\quad \quad \quad bab, bbaaabb, bbaaab, bbaaaba, bab, bab, abbbaa, \}
\]

\[
\quad \quad \quad \quad abbbbaa, babaaab, babbaab, abbbbaab, babbaaabb, ababbaaabb, \}
\]

\[
\quad \quad \quad \quad ababbaaab, babbaaab, babbaab, ababbaab, ababbaaab, \}
\]

\[
L_2^* \}
\]

First note that \( L_2^* \subseteq \{a, b\}^* \) by definition. Moreover, since \( \{a, b\} \subseteq L_2 \), then \( \{a, b\}^* \subseteq L_2^* \). But then, this means that \( L_2 = \{a, b\}^* \).
Consider the language

\[ L = \{abb^i : i \geq 0\} \cup \{aba^i : i \geq 0\} \]

a) Give the transition graph of a DFA \( M \) such that \( L(M) = L \).

**Hint:** It is helpful to make sure that your DFA accepts some string in the language, and does not accept strings which are not in the language.
b) Describe the language $L$ above using a regular expression.

The language $L$ equals $L(r)$ for the regular expression

$$r = (ab)(a^* + b^*) .$$
Consider the NFA $M = (Q, \Sigma, \delta, q_0, F)$ given by the following transition graph:

![Transition Graph](image)

a) Let $\delta$ be the transition function of $M$ defined via the above transition graph, and let $\delta^*$ be the corresponding extended transition function. Compute the following sets:

- $\delta^*(q_0, \lambda) = \{q_0, q_1\}$

- $\delta^*(q_0, aa) = \{q_2, q_3\}$

- $\delta^*(q_0, aba) = \emptyset$

- $\delta^*(q_0, abb) = \{q_1\}$
b) Give both a right- and a left-regular grammar generating the language \( L(M) \).

******

**Right-regular:** We can read the grammar directly from the NFA, and use variables \( V = \{Q_0, Q_1, Q_2, Q_3\} \), and terminals \( T = \{a, b\} \). The start variable is \( Q_0 \).

\[
\begin{align*}
Q_0 & \rightarrow Q_1 \mid aQ_2 \mid aQ_3 \\
Q_1 & \rightarrow bQ_1 \mid \lambda \\
Q_2 & \rightarrow bQ_1 \mid aQ_3 \\
Q_3 & \rightarrow aQ_2 \mid aQ_3 \mid \lambda .
\end{align*}
\]

**Left-regular.** This was a bit more complicated. Equivalently, we can think of the above NFA as having an additional state \( q_4 \) which is the only final state, and \( \lambda \) transitions go from \( q_1 \) and \( q_3 \) which are not final states any more. We saw in the homework that this does not change the language accepted by the NFA. Now, one can easily read the left-regular grammar from this NFA, by starting from the **final state**, and “going backwards”.

The resulting left-regular grammar has variables \( V = \{Q_0, Q_1, \ldots, Q_4\} \), terminals \( T = \{a, b\} \), start variable \( Q_4 \), and the following productions:

\[
\begin{align*}
Q_4 & \rightarrow Q_1 \mid Q_3 \\
Q_1 & \rightarrow Q_1 b \mid Q_0 \\
Q_3 & \rightarrow Q_3 a \mid Q_2 a \mid Q_0 a \\
Q_2 & \rightarrow Q_0 a \mid Q_3 a \mid Q_1 b \\
Q_0 & \rightarrow \lambda .
\end{align*}
\]

******
c) Give the transition graph of a DFA $N$ accepting the same language $L(M)$ as the NFA $M$.

*****

We apply the transformation presented in class. The resulting states will correspond to subsets of \{q_0, q_1, q_2, q_3\}. Let $\delta$ and $\delta^*$ be as above, the transition function (and its extended form) for $M$. We define by $\delta_N$ the transition function for $N$.

- The initial state is $\delta^*(q_0, \lambda) = \{q_0, q_1\}$.
- From \{q_0, q_1\}, we need to define an $a$- and a $b$-transition. For $a$, note that from \{q_0, q_1\}, in $M$, we can reach both $q_2$ and $q_3$ (but not $q_0$ itself, nor $q_1$), and thus $\delta_N(\{q_0, q_1\}) = \{q_2, q_3\}$. Moreover, for $b$, in $M$ we can reach only $q_1$ from \{q_0, q_1\}, and thus $\delta_N(\{q_0, q_1\}, b) = \{q_1\}$.
- From \{q_1\}, we need to define transitions for $a$ and $b$. We see that in $M$, from \{q_1\}, we can reach only \{q_1\} following $b$-paths, and nowhere following $a$-transitions. Thus, $\delta_N(\{q_1\}, a) = \emptyset$ and $\delta_N(\{q_1\}, b) = \{q_1\}$.
- Finally, from \{q_2, q_3\}, following $a$-transitions, we can reach both $q_2$ and $q_3$, whereas following $b$-transitions we only get to \{q_1\}, and thus $\delta_N(\{q_2, q_3\}, a) = \{q_2, q_3\}$, and $\delta_N(\{q_2, q_3\}, b) = \{q_1\}$.
- We still need to set $\delta_N(\emptyset, a) = \delta_N(\emptyset, b) = \emptyset$.

It is clear that the accepting states are all of those containing either $q_1$ or $q_3$, and thus all states \{q_0, q_1\}, \{q_1\}, and \{q_2, q_3\} are accepting in the DFA $N$.

The transition graph of the DFA $N$ is therefore as follows (drawn according to the above description of $\delta_N$):

*****
We consider the language

\[ L = \{ a^{2n}b^n : n \geq 0 \} \]

over the alphabet \( \Sigma = \{a, b\} \).

a) Show that the language \( L \) is not regular by using the Pumping Lemma.

Fix an arbitrary integer \( m \). Then, consider the string \( w = a^{2m}b^m \in L \). For every splitting \( w = xyz \) where \( |xy| \leq m, |y| \geq 1 \), we have that \( xy \) must be part of the first \( m \) \( a \)'s, and thus \( x = a^i, y = a^j \), and \( z = a^{2m-i-j}b^m \) for some \( i \) and \( j \) such that \( i + j \leq m, j \geq 1 \).

Now, let us look at

\[ xy^2z = a^{2m+2j}b^m. \] (1)

Since \( j \geq 1 \), we must have that \( 2m + j \neq 2m \), and thus \( xy^2z \notin L \).

Thus, by the Pumping Lemma, \( L \) is not regular.
b) Show that the language $L$ is context free by giving an appropriate context-free grammar $G$ generating it, i.e., such that $L = L(G)$.

The following grammar generates the language $L$ (with start variable $S$):

- $S \rightarrow aaSb$
- $S \rightarrow \lambda$
