Automata and Formal Languages — Endterm Exam

- You have 120 minutes to complete the exam.
- Answers must be written in a separate booklet. Do not answer on the exam.
- Please let us know if you need more paper.
- Write your name and Matrikelnummer on every sheet.
- Write with a non-erasable pen. Do not use red or green.
- You are not allowed to use auxiliary means other than pen and paper.
- You can obtain 40 points. You need 17 points to pass.
- The ⋆ symbol indicates a more challenging question.

Question 1 (11 points)

a. Let Σ be the alphabet \{a, b\}, and let \( p \) be the word pattern \( ababaa \). Build the DFA \( B_p \) (obtained by determinizing the naive NFA \( A_p \) for \( \Sigma^*p \)).

b. Give the fragment of the master automaton that contains the states of the language \( L = \{aab, bbb, bab\} \) and all its residuals (all the states between \( q_L \) and \( q_\emptyset \)).

c. Given a word \( w \) over the alphabet \( \Sigma = \{a, b\} \) we define \( \overline{w} \) to be the word obtained from \( w \) by replacing \( a \) by \( b \), and \( b \) by \( a \). For example, \( \overline{aababbb} = bbbabaab \) and \( \overline{bab} = abaa \). Decide whether the language \( L = \{w\overline{w} : w \in \Sigma^*\} \) is regular or irregular, and prove this by analyzing its residuals.

d. Give a regular expression recognizing the language of the following the MSO formula

\[
\varphi = \exists x \exists y. x \neq y \land Q_a(x) \land Q_a(y) \land [\forall z.(z \neq x \land z \neq y) \rightarrow (Q_b(z) \land x < z \land z < y)].
\]

e. Consider the following NBA.

\[
\text{Draw } \text{dag}(b(baa)^*). \text{ Does it admit an odd ranking? Give such a ranking if it exists, and provide a short justification if it does not.}
\]
Question 2  (3 points)
Let \( A \) be the following DFA:

```
(a) Compute the language partitions of \( A \).
(b) Draw the minimal automaton using the language partitions from (a).
```

---

Question 3  (6 points)
Given \( n \in \mathbb{N} \), let \( msbf(n) \) be the set of most significant bit first encodings of \( n \), i.e., the words that start with an arbitrary number of leading zeros, followed by \( n \) written in binary. For example, \( msbf(6) = 0*110 \) and \( msbf(3) = 0*11 \). Let \( \text{val} : \{0,1\}^* \rightarrow \mathbb{N} \) be the function that associates to every word \( w \in \{0,1\}^* \) the number \( \text{val}(w) \) represented by \( w \) in the most significant bit first encoding. For example, \( \text{val}(110) = 6 \) and \( \text{val}(011) = 3 \).

a. Let \( T \) be the following transducer over alphabet \( \Sigma = \{0,1\} \times \{0,1\} \).

```
\[
\begin{array}{c}
s0 \quad |0| \quad |1| \\
|0| \quad s0 \\
|1| \quad s1
\end{array}
\]
```

What is the relation between \( \text{val}(x) \) and \( \text{val}(y) \), for any \([x,y] \) accepted by \( T \)?

b. Draw a transducer \( T_{+1} \) recognizing the language

\[
\{[x,y] \in \Sigma^* | \text{val}(y) = \text{val}(x) + 1 \}.
\]
Question 4 (5 points)
Recall: A process can send a message \( m \) to the channel with the instruction \( c \! m \). A process can also consume the first message of the channel with the instruction \( c \? m \). If the channel is full when executing \( c \! m \), then the process blocks and waits until it can send \( m \). When a process executes \( c \? m \), it blocks and waits until the first message of the channel becomes \( m \).

Suppose there are two processes being executed concurrently that communicate through a channel \( c \). Channel \( c \) is a queue that can store up to 1 message. The two processes follow these two algorithms respectively:

```plaintext
process(1):
    while true do
        c ! m
        /* critical section */
        c ? m

process(2):
    while true do
        c ? m
        c ? m
        /* critical section */
        c ! m
```

a. Model the program by constructing a network of three automata:
   - One for process 1, using the alphabet \( \Sigma_1 = \{c?m, c!m, cs_1\} \),
   - One for process 2, using the alphabet \( \Sigma_2 = \{c?m, c!m, cs_2\} \),
   - One for the channel \( c \) of size 1, that is initially empty, using the alphabet \( \Sigma_c = \{c?m, c!m, c?m, c!m\} \).

b. Construct the asynchronous product \( P \) of the three automata obtained in (a). The alphabet of the automaton \( P \) should be \( \Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_c \).

c. Consider the state of the asynchronous product \( P \) where both processes are in the critical section. Is this state reachable? Give a short justification based on automaton \( P \).

Question 5 (3 points)
Consider the following DBAs \( B_1 \) and \( B_2 \):

```
\( B_1 \):

\begin{tikzpicture}
  \node[state, initial] (p0) at (0,0) {p_0};
  \node[state] (p1) at (1,0) {p_1};
  \draw (p0) edge[loop above] node {a} (p0)
        edge[below] node {b} (p1)
        (p1) edge[loop above] node {a} (p1)
        (p1) edge[below] node {b} (p0);
\end{tikzpicture}
```

```
\( B_2 \):

\begin{tikzpicture}
  \node[state, initial] (q0) at (0,0) {q_0};
  \node[state] (q1) at (1,0) {q_1};
  \draw (q0) edge[loop above] node {a} (q0)
        edge[below] node {b} (q1)
        (q1) edge[loop above] node {a} (q1)
        (q1) edge[below] node {b} (q0);
\end{tikzpicture}
```

a. Give \( \omega \)-regular expressions recognizing the languages of \( B_1 \) and \( B_2 \).

b. Give the DBA \( B_1 \cap B_2 \) using the algorithm seen in class. Give an \( \omega \)-regular expression for \( B_1 \cap B_2 \).
Question 6 (3 points)
Let $B$ be the following Büchi automaton.

![Automaton Diagram]

a. For every state of $B$, give the discovery time and finishing time assigned by a DFS on $B$ starting in $s_0$ (i.e. the moment they first become grey and the moment they become black). Visit successors $s_i$ of a given state in the ascending order of their indices $i$. For example, when visiting the successors of $s_2$, first visit $s_3$ and later $s_4$.

b. The language of $B$ is not empty. Give the witness lasso (as a sequence of states) found by applying TwoStack to $B$ following the same convention for the order of successors as above.

Question 7 (6 points)
Let $AP = \{p, q\}$ and let $\Sigma = 2^{AP}$. Recall: An LTL formula is a tautology if it is satisfied by all computations.

a. Is the following formula a tautology: $(\text{GF} p \land \text{GF} q) \Rightarrow \text{G}(p \text{ U } q)$? Provide a formal proof if it is and a counter-example if it is not.

b. Is the following formula a tautology: $\text{G}(p \text{ U } q) \Rightarrow (\text{GF} p \lor \text{GF} q)$? Provide a formal proof if it is and a counter-example if it is not.

c. Give a Büchi automaton for the $\omega$-language over $\Sigma$ defined by the following LTL formula: $\text{G}(p \text{ U } q)$.

Question 8 (3 points)
⋆ Given a language $L$ we define the language $\text{Cycle}(L) = \{vu \mid uv \in L\}$. For example, if $L = \{ab, abcd\}$ then $\text{Cycle}(L) = \{ab, ba, abcd, bcda, cdab, dabc\}$; in particular, $abcd$ is not in $\text{Cycle}(L)$ as it cannot be written as $vu$ such that $uv \in L$.

Find a language $L$ such that $L$ is not regular and $\text{Cycle}(L)$ is regular. Give proofs for both statements.
Solution 1  (11 points)

a. The automaton $B_p$ for $p = ababaa$ is given below:

![Automaton B_p](image)

b. The fragment of the master automaton for the language $L$ is given below:

![Fragment of master automaton](image)

c. The language $L$ is irregular as it has infinitely many residuals.

Let us prove that the set of residuals $\{L^a^i : i \in \mathbb{N}\}$ in infinite, that is, that any two residuals from this set are indeed different. If $i \neq j$, then we have that $b^i \in L^a^i$ and $b^j \notin L^a^j$. Therefore, it holds that $L^a^i \neq L^a^j$.

IMPORTANT! The fact that $b^i \in L^a^i$ and $b^j \notin L^a^j$ is correct, but it does not prove that $L^a^i \neq L^a^j$.

d. The regular expression is $ab^*a$ recognizes the MSO formula.

e. The $\text{dag}$ for $b(baa)^\omega$:

![Dag for b(baa)^\omega](image)

A possible odd ranking is annotated on the $\text{dag}$. In particular it exists because $b(baa)^\omega$ is not accepted by the NBA.

Solution 2  (3 points)

• The language partition is $P_\ell = \{\{A, D, G\}, \{B, E, H\}, \{C, F, I\}\}$.

• The minimal automaton is given below:

![Minimal automaton](image)
Solution 3  (6 points)

a. The value $\text{val}(x)$ is equal to twice $\text{val}(y)$, i.e. $L(T) = \{ [x, y] \in \Sigma^* \mid \text{val}(x) = 2 \times \text{val}(y) \}$.

b. Transducer $T_{+1}$:

\[
\begin{array}{c}
\text{val}(x) = 2 \times \text{val}(y) \\
[0, 0] \quad [0, 1] \\
\downarrow & \downarrow \\
\text{s0} \quad \text{s1} \\
\end{array}
\]

Solution 4  (5 points)

a. The automata for the channel, process(1) and process(2) are respectively:

\[
\begin{array}{c}
p_0 \quad p_1 \quad p_2 \\
\downarrow \quad \downarrow \quad \downarrow \\
q_0 \quad q_1 \quad q_2 \quad q_3 \\
\end{array}
\]

b. The asynchronous product $P$ is given below:

\[
\begin{array}{c}
p_0, q_0, \square \\
\downarrow \quad \downarrow \quad \downarrow \\
p_1, q_0, m \\
\downarrow \quad \downarrow \quad \downarrow \\
p_2, q_0, m \\
\end{array}
\]

b. The state where both processes are in the critical section is not reachable, since $P$ does not contain any of the states $(p_1, q_2, \square)$ and $(p_1, q_2, m)$.

6 of 8
Solution 5 (3 points)

a. \( L(B_1) = (b^*ab) \) and \( L(B_2) = (a^* + ba)^\omega \).

b. The intersection of \( B_1 \) and \( B_2 \) using \textit{IntersNBA} yields:

The language of \( B_1 \cap B_2 \) is \( a(ba)^\omega + b(ab)^\omega \).

Solution 6 (3 points)

a. We note "state[discovery time/finishing time]".

- If we start at 1: \( s_0[1/12], s_1[2/11], s_2[3/10], s_3[4/5], s_4[6/9], s_5[7/8] \).
- If we start at 0: \( s_0[0/11], s_1[1/10], s_2[2/9], s_3[3/4], s_4[5/8], s_5[6/7] \).

b. The lasso found by \textit{TwoStack} from \( s_0 \) is \( s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_0 \).

Solution 7 (6 points)

a. The formula is not a tautology. Here are two counterexamples: \( (\{p\} \emptyset \{q\})^\omega \) and \( \emptyset \{p,q\}^\omega \).

b. This formula is a tautology. We prove this by contradiction.

Suppose the formula is not a tautology. Then there exists an execution \( \sigma \) that does not satisfy it, that is, the following holds:

\[ \sigma \not\models G(p U q) \Rightarrow (GFp \lor GFq). \]

Therefore, we have the following:

\[ \sigma \models G(p U q), \quad (1) \]
\[ \sigma \not\models GFp \lor GFq. \quad (2) \]

First, note that from (1) we know the following:

\[ \sigma^k \models p U q, \quad \text{for every } k \geq 0. \quad (3) \]

Second, note that (2) is equivalent to \( \sigma \models FG \neg p \land FG \neg q \), that is \( \sigma \models FG \neg p \) and \( \sigma \models FG \neg q \). Since we have that \( \sigma \models FG \neg q \), by definition of the operator \( F \) we know the following:

\[ \sigma^i \models \neg q, \quad \text{for some } i \geq 0. \quad (4) \]

By definition of \( G \) this means the following:

\[ \sigma^j \models \neg q, \quad \text{for every } j \geq i. \quad (5) \]

Let us now focus again on (3) and on the index \( i \) defined in (4). From (3) we know that also for this particular index \( i \) it holds that \( \sigma^i \models p U q \). Therefore, by definition of \( U \), we know that there exists an index \( l \geq i \) with \( \sigma^l \models q \). This contradicts (5), and hence our assumption that the formula is not a tautology is wrong. This shows that the formula is indeed a tautology.
Solution 8  (3 points)

One possible solution language is $L = \{a^n ba^n \mid n \in \mathbb{N}\}$. For any $i \neq j$, $a^i b \in L^a$ but $a^j b \notin L^a$ so the $(L^a)_i$ are an infinite family of residuals, thus proving that $L$ is not a regular language. The language Cycle($L$) on the other hand is regular, as it is recognizable by the following DFA.

---

8 of 8