RETAKE AUTOMATA THEORY
Wednesday 27 March 2024, 13:30 - 16:30

This exam consists of nine exercises, where [x pt] indicates how many points can be earned per exercise. A total of 100 points can be earned.
It is important to provide an explanation or motivation when a question asks for it.
A finite automaton in this exam (without further addition), refers to a deterministic finite automaton without Λ-transitions (which is elsewhere called *DFA*).

---

1. [6 pt] Let

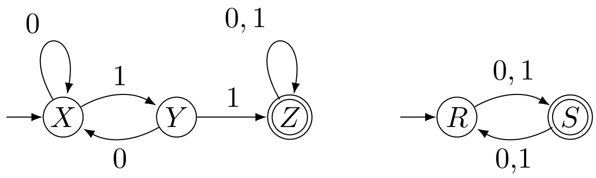
$$L = \{\sigma^2 w \sigma^2 \ \mid \ \sigma \in \{a,b\}, w \in \{a,b\}^* \ \}$$

For example, $aaaa \in L$, but $bb \notin L$.

Draw a finite automaton M, such that $L(M) = L$ with at most ten states. If your automaton has more than ten states, use the minimization algorithm – *on scratch paper*, to reduce the number of states.

---

2. [13 pt] Let M_1 and M_2 be the following two finite automata with the same input alphabet $\Sigma = \{0, 1\}$. Let $L_1 = L(M_1)$ and $L_2 = L(M_2)$.



M_1 M_2

(a) Using the product construction, construct a finite automaton M from M_1 and M_2, such that $L(M) = L_2 \setminus L_1$. Indicate M's initial state, set of accepting states and fill-in the following transition table.

δ	0	1
RX		
SX		
RY		
SY		
RZ		
SZ		

(b) Apply the minimization algorithm on the resulting automaton M. Proceed one column at a time (from left to right), and consider the pairs in each column from top to bottom. Fill-in the table below with the iteration number in which you establish that states i and j cannot be merged and identify which states can be merged, if any.

$i = SX$	·				
RY	·	·			
SY	·	·	·		
RZ	·	·	·	·	
SZ	·	·	·	·	·
$j =$	RX	SX	RY	SY	RZ

---

3. [12 pt]

   Consider the following language, L_1 over $\{a, b, c\}$:

   $$L_1 = \{\ a^i b^j c^j \mid i \geq 1, j \geq 0\} \cup \{\ b^j c^k \mid j \geq 0, k \geq 0\}$$

   (a) For each of the following strings x_i, give or describe the (concrete) elements of L_1/x_i:

      i. $x_1 = a^2b^4$

      ii. $x_2 = c^2b^2$

      iii. $x_3 = b^4$

   (b) Give or describe the (concrete) elements of (the equivalence class) $[a^2b^4]$, i.e., all elements of $\{a, b, c\}^*$ that are *indistinguishable* from a^2b^4 with respect to L_1.

---

4. [11 pt] Let

   $$L = \{x \in \{a, b\}^* \ \mid \ x \text{ ends with } aa \text{ or with } b\}$$

   Some elements of L are, for example, $abaa$ and $abab$. For each of the following regular expressions r_1, r_2, r_3, r_4 indicate whether it is a correct expression for L or not. If an r_i is not a correct expression, explain this, by giving a minimum-length string x

   (a) that is in L, but does not satisfy r_i,

   (b) or which is not in L, but satisfies r_i.

   If strings x can be found for both situations, you only need to provide one string x. Clearly indicate whether situation (a) or (b) applies for your provided string x.

   $$\begin{aligned} r_1 &: \quad (aa + b + a^*b)^*(aa + b) \\ r_2 &: \quad (a^*b^*b)^* + (a^* + b)^*aa \\ r_3 &: \quad (a + b)^*(b + ab + aa^*) \\ r_4 &: \quad (b + ab + aaa^*b)(b + ab + aaa^*b)^*(\lambda + aaa^*) + aaa^* \end{aligned}$$

---

5. [10 pt]

   (a) For each of the following statements, indicate whether or not it is true in general. You do not have to motivate your answer.

      i. If L_1 and L_2 are regular languages, then also $L_1 \cap L_2$ is a regular language.

      ii. If L_1 and L_2 are context-free languages, then also $L_1 \cap L_2$ is a context-free language.

      iii. If L_1 is a regular language, then also L_1^* is a regular language.

      iv. If L_1 is a context-free language, then also L_1^* is a context-free language.

   (b) Give an example of a context-free language L_1, such that its complement L_1' is not a context-free language.

---

6. [9 pt] Let $L = \{a^i b^j c^k \ \mid \ j > i + k\}$. For example, b, abb and $abbbbc$ are elements of L.

   Give a context-free grammar G, such that $L(G) = L$.

---

7. [10 pt] Let G be the context-free grammar with start variable (and only variable) S, and the following productions:

$$S \to SaS \mid b \mid \Lambda$$

In homework 3, you were asked to prove that any string $x \in \{a,b\}^*$ that has no occurrence of the substring bb can be generated by G. Now, you are asked to prove the converse, i.e., that a string generated by G has no occurrence of the substring bb.

To be concrete, let $x \in L(G)$, i.e., $x \in \{a,b\}^*$ and $S \Rightarrow^* x$. Use induction on the length (the number of steps) of the derivation of x to prove that x has no occurrence of the substring bb.

---

8. [15 pt] Let $L = \{x \in \{a,b\}^* \mid n_b(x) = 2 \cdot n_a(x)\}$

   (a) Give the first five elements in the canonical (shortlex) order of L.

   (b) Draw a pushdown automaton M, such that $L(M) = L$.

   This pushdown automaton must be based directly on the properties of the language. It should, therefore, not be the result of a standard construction for, for example, converting a context-free grammar into a pushdown automaton.

   Try to ensure that M is deterministic. If you do not succeed in this, you can still earn most of the points. You do not lose points if M has Λ-transitions.

   Also explain how M uses its states and stack (symbols) to accept precisely the right language.

---

9. [14 pt] Let G be the context-free grammar with start variable S and the following productions:

$$S \to aSb \mid A \qquad A \to aA \mid \Lambda$$

   (a) Draw the non-deterministic bottom-up pushdown automaton $NB(G)$. Do not forget to draw the auxiliary states (necessary for reductions by productions $X \to \alpha$ with $|\alpha| \geq 2$) with their transitions.

   (b) Carry out a successful computation in $NB(G)$ for input $x = aab$, i.e., a computation resulting in acceptance of x. Present this computation in a table of the following form:

state	stack (reversed)	remaining input	action
q_0	Z_0	aab	...
...	...	...	...

   Here, as usual, q_0 is the start state and Z_0 is the initial stack symbol of $NB(G)$.

   In the table, you may perform a reduction in one step, even if it actually requires a sequence of transitions of $NB(G)$.

   Hint: It may be helpful for choosing the right action, to first draw a derivation tree for x in G (on scratch paper).

`end of exam`