RESIT AUTOMATA THEORY
Wednesday 1 February 2023, 09.00 - 12.00

This exam consists of eight 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

$$L = \{x \in \{a,b\}^* \mid x \text{ contains (at least) a substring } abaabb\}$$

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

---

2. [11 pt] Formally, a finite automaton is a 5-tuple $(Q, \Sigma, q_0, A, \delta)$, where δ is the transition function.

The extended transition function $\delta^* : Q \times \Sigma^* \to Q$ is defined by

$$\begin{aligned} \delta^*(q, \Lambda) &= q && \text{for every } q \in Q \\ \delta^*(q, y\sigma) &= \delta(\,\delta^*(q,y), \sigma\,) && \text{for every } q \in Q,\ y \in \Sigma^*,\ \sigma \in \Sigma \end{aligned}$$

Show, using induction on the length of z, that

$$\delta^*(q, yz) = \delta^*(\,\delta^*(q,y), z\,) \qquad \text{for every } q \in Q,\ y, z \in \Sigma^*$$

(i.e., if we want to process the string yz starting in state q, we can first process the string y, and then the string z).

---

3. [16 pt]

(a) Let $L \subseteq \Sigma^*$ be a language, and let $x \in \Sigma^*$ be a string.
How is the set L/x defined (the 'future set' of x with respect to L)? That is, what strings are in L/x?
If you do not know the answer to this question, you can 'buy' it from the lecturer. Perhaps you can then solve (b) and (c).

Let $L_1 = \{(ab)^i(bc)^j \mid i \geq j\}$.

(b) For each of the following strings x, give or describe the elements of L_1/x:

i. $x = (ab)^4(bc)^4$
ii. $x = (ab)^4(bc)^2$
iii. $x = (ab)^4$

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

---

4. [11 pt] In a previous exam, there was an exercise on regular languages over the alphabet $\{a, b, c\}$. Let $L = \{a^i b^j c^k \mid i, j, k \geq 1\}$. A regular expression for L is, for example, $aa^*bb^*cc^*$. The previous exam also asked for a regular expression for L' (the complement of L).

   Consider the following three regular expressions r_1, r_2, r_3:

   (a) $r_1 = (a+b)^* + (b+c)^* + (a+b+c)^*(ac+ca+ba+cb)(a+b+c)^*$
   (b) $r_2 = (a+b)^* + (a+c)^* + (b+a^*c)(a+b+c)^*$
   (c) $r_3 = a^*c^*b^* + b^*a^*c^* + b^*c^*a^* + c^*a^*b^* + c^*b^*a^*$

   Let $L(r_i)$ be the language described by the expression r_i. For each of these three expressions r_i it holds that $L(r_i) \subseteq L'$. Now answer the following question for each of these three expressions r_i:

   Is $L' \subseteq L(r_i)$?
   If not, give a string x that is in L', but not in $L(r_i)$.
   If yes, then you don't have to explain that.

   Indeed, if the answer is 'yes', then r_i is a correct regular expression for language L'.

---

5. [10 pt]

   (a) Give a context-free grammar G_1, such that

   $$L(G_1) = AEqB = \{x \in \{a, b\}^* \mid n_a(x) = n_b(x)\}$$

   (b) Give a context-free grammar G_2, such that

   $$L(G_2) = \{x \in \{a, b\}^* \mid n_a(x) = n_b(x) + 1\}$$

---

6. [14 pt]

   (a) When do we say that a context-free grammar $G = (V, \Sigma, S, P)$ is in *Chomsky normal form*?

   If you do not know the answer to this question, you can 'buy' it from the lecturer so that you may be able to solve the next part.

   (b) Consider the context-free grammar G_1 with terminals $\{a, b\}$, start variable S and productions

   $$S \to Sa \mid bb \mid AB \qquad A \to aAbS \mid BBa \mid Ba \mid a \qquad B \to SB \mid a$$

   G_1 contains no Λ-productions and no unit productions.

   Construct a context-free grammar G_2 in Chomsky normal form, such that $L(G_2) = L(G_1) - \{\Lambda\}$. Clearly explain how you arrived at your answer, and provide intermediate results.

---

7. [18 pt] Let

$$\begin{aligned} L_1 &= \{x \in \{a,b,c\}^* \mid n_a(x) \neq n_c(x)\} \\ L_2 &= \{x \in \{a,b,c\}^* \mid n_a(x) + n_c(x) > n_b(x) + 1\} \end{aligned}$$

Draw a pushdown automaton M, such that $L(M) = L_1 \cup L_2$. [1] You do not lose points if M is non-deterministic and/or contains Λ-transitions.

Hint: draw first separate pushdown automata for L_1 and L_2 and combine those.

Explain how M uses its states and stack to accept precisely the right language.

---

8. [14 pt] Let G be the context-free grammar with start variable S and the following productions:

$$S \to aSb \mid bbT \qquad T \to Tc \mid \Lambda$$

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

(b) Perform a successful computation in $NB(G)$ for the input $x = abbcb$, i.e., a computation that results in the acceptance of x.

Show this computation with a table of the following form:

state q	reverse stack content	remaining input	move
q_0	Z_0	$abbcb$	...
...	...	...	...

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

You may perform a reduction in one step in the table, even if multiple transitions of $NB(G)$ are actually followed.

`end of exam`

[1] Indeed, the language $L_1 \cup L_2$ is the complement of the language $\{x \in \{a,b,c\}^* \mid n_a(x) = n_c(x) \text{ and } n_a(x) + n_c(x) \leq n_b(x) + 1\}$ that we know from an earlier exam.