EXAM AUTOMATA THEORY

Thursday 23 December 2021, 10:15 - 13:15

This exam consists of eight questions, where the approximate number of points that can be earned is written between [and]. There are 100 points to be earned in total. Whenever a question asks for an explanation, motivation or clarification, it is important to give it.

Whenever we talk about a finite automaton in this exam (without further specification), we mean a deterministic finite automaton without Λ-transitions (which is called a DFA elsewhere).

---

1. [10 pts]

   a. Draw a finite automaton M such that

   $$L(M) = \{a^i(ba)^j a^k \mid i, j, k \geq 1\}$$

   b. Formally, a finite automaton is a 5-tuple $(Q, \Sigma, q_0, A, \delta)$.
   What are, for your finite automaton M from part (a): Q, Σ, q_0, A and $\delta^*(q_0, aaba)$?
   Give concrete answers for M, so no general explanation.

---

2. [9 pts] Let $L = \{x \in \{a, b\}^* \mid x = a^i(ba)^j a^k$ with $i, j, k \geq 0$ and x contains as many occurrences of the substring ba as of the substring $aa\}$.

   So a string $x \in L$ has to meet all the given requirements. Note that occurrences may overlap. For example, $x = aaa$ contains two occurrences of the substring aa.

   Indeed, we know this language L from the homework assigments. The first eight elements in the canonical order of L are $\Lambda, a, baa, aaba, abaa, babaaa, aaababa, aababaa$.

   Show that the elements of the infinte set $\{a^n \mid n \geq 0\}$ are pairwise L-distinguishable. In other words, let m, n be arbitrary numbers ≥ 0 with $m \neq n$; use the definition of distinguishability to show that a^m and a^n are L-distinguishable.

---

3. [8 pts]

   Give a regular expression for the language

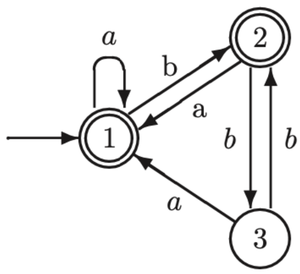
   $$L = \{x \in \{a, b\}^* \mid |x| \geq 2 \text{ and } x \text{ starts and ends with } aa \text{ or } ba\}$$

   Examples of elements of L are $aababa$ and $aaaaaaa$.

---

4. [15 pts] During the lectures we discussed two (related) constructions to systematically construct a regular expression that corresponds to a finite automaton. One method was the state-elimination method by Brzozowski and McCluskey. The other method builds up regular expressions $r^k(i,j)$,[1] corresponding to the strings that take you from state i to state j in the automaton, while on the way only states 1 to k (k included) may be passed. Here, we assume that the states are numbered as $1, 2, \ldots$. First all expressions $r^0(i,j)$ will be determined, then the expressions $r^1(i,j)$, then the expressions $r^2(i,j)$, etc.

   You are supposed to **partially** implement this second construction for the finite automaton M below:

   Give the complete resulting matrices r^0 and r^1, so with the values $r^0(i,j)$ and $r^1(i,j)$ for all i and j. In addition, give element $r^2(3,1)$. Although with this we do not yet have a complete regular expression for $L(M)$, this is enough for this question.

   You may simplify the regular expressions $r^1(i,j)$ that you get to equivalent expressions. You may **not** simplify the expression $r^2(3,1)$ (based on your regular expressions $r^1(i,j)$).

---

[1] In the book this expression is written as $r(i,j,k)$.

5. [16 pts] Again, let $L = \{x \in \{a,b\}^* \mid x = a^i(ba)^j a^k$ with $i,j,k \geq 0$ and x contains as many occurrences of the substring ba as of the substring $aa\}$.

   So a string $x \in L$ has to meet all the given requirements. Note that occurrences may overlap. For example, $x = aaa$ contains two occurrences of the substring aa. Homework assignment 3 asked for a context-free grammar for L.

   a. Let G_1 be the context-free grammar with start variable S and the following productions:

   $$\begin{aligned} S &\rightarrow AB \mid aAB \mid \Lambda \\ A &\rightarrow aAba \mid \Lambda \\ B &\rightarrow baBa \mid \Lambda \end{aligned}$$

      i. Is $L(G_1) \subseteq L$? If not, give an element of $L(G_1)$ that is not in L.

      ii. Is $L \subseteq L(G_1)$? If not, give an element of L that is not in $L(G_1)$.

   b. Let G_2 be the context-free grammar with start variable S and the following productions:

   $$\begin{aligned} S &\rightarrow aA \mid B \mid aaAbabaBa \\ A &\rightarrow aAba \mid \Lambda \\ B &\rightarrow baBa \mid \Lambda \end{aligned}$$

      i. Is $L(G_2) \subseteq L$? If not, give an element of $L(G_2)$ that is not in L.

      ii. Is $L \subseteq L(G_2)$? If not, give an element of L that is not in $L(G_2)$.

---

6. [17 pts] Let

   $$L = \{a^i b^j a^k \mid i,j,k \geq 0 \text{ and } i + k \leq 2j\}$$

   a. Give the first six elements in the canonical order of L.

   b. Draw a pushdown automaton M, such that $L(M) = M$.
   This pushdown automaton must be directly based on the properties of the language. So it must not be the result of a standard construction to, e.g., transform a context-free grammar into a pushdown automaton.
   Try to make M deterministic, and not containing any Λ-transitions. If you do not succeed in this, you can still get part of the points.
   Also, explain how M uses its states and stack to accept exactly the right language.

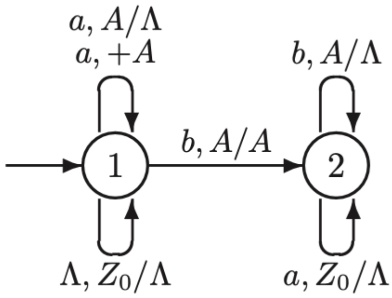
---

7. [12 pts]

   a. Let $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ be an arbitrary pushdown automaton.
   What does it mean when we say that a string $x \in \Sigma^*$ is accepted by M *by empty stack*. You may formulate your answer with words and/or configurations, as long as it is clear and complete.
   You will not earn points by answering that x is in the empty stack language $L_e(M)$ of M, because $L_e(M)$ is by definition the set of strings that are accepted by M by empty stack.
   If you do not know the answer to this question, you can 'buy' it from the teacher. Maybe then you can still make question (b).

   b. i. Explain how you can transform an arbitrary pushdown automaton $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ into another pushdown automaton M_1, such that $L(M_1) = L_e(M)$. In other words: M_1 accepts 'in the normal way' (with accepting states) the language that M accepts by empty stack.
   N.B.: You don't have to show that the construction you describe is correct. (But of course it does have to be correct.)

   ii. Now let M be the following PDA (with initial stack symbol Z_0):



   Draw the pushdown automaton M_1 that is the result of the construction from the previous question, such that $L(M_1) = L_e(M)$.

8. [13 pts] The pumping lemma for context-free languages is as follows:

   Suppose L is a context-free language.
   Then there exists an integer n, such that
   for every $u \in L$ with $|u| \geq n$, u can be written as $u = vwxyz$ for certain strings v, w, x, y, z for which:
   1. $|wy| \geq 1$ (i.e. $wy \neq \Lambda$).
   2. $|wxy| \leq n$.
   3. For every $m \geq 0$ the string $vw^m xy^m z$ is also in L.

   Now let

   $$L_1 = \{a^i b^{i+k} a^k \mid 0 \leq i < k\}$$

   and let n be the the number from the pumping lemma for this language.

   For each of the following four strings u_1, u_2, u_3, u_4, say whether it is suitable for finding a contradiction with the pumping lemma. In addition, for every string u_i that is **not** suitable, say why it is not suitable, for example by giving a concrete decomposition $vwxyz$ of u_i that perfectly satisfies the pumping lemma.

   $$u_1 = aabbbbbaaa$$
   $$u_2 = a^{n-1}b^{2n-1}a^n$$
   $$u_3 = a^n b^{2n+n} a^{2n}$$
   $$u_4 = a^n b^{2n+1} a^{n+1}$$

End of exam.