Talen en Automaten

Retake Exam, Tue 1st May, 2018
8:30 – 11:30

This test consists of **7** problems over **5 pages. Explain your approach.** You can score a maximum of 100 points, and each question indicates how many points it is worth. The test is closed book. You are NOT allowed to use a calculator, a computer or a mobile phone. You may answer in Dutch or in English. Please write clearly, and do not forget to put on each page: your name and your student number.

Notation Throughout the test, we denote for any alphabet A, $w \in A^*$ and $a \in A$ by $|w|_a$ the number of a's in w, and by $|w|$ the length of the word w.

Problem 1.

For each of the following claims, state whether it is true or false. Support your claim with either a counterexample or a (short) argument/proof. **(16pt)**

1. If a language L contains finitely many words, it is regular.
2. If a language is context-free, it is not regular.
3. If L is regular, then $K \cap L$ is also regular.
4. The language $L = \mathcal{L}\left(b(a+b)^*\right) \cup \{a^n b^m \mid n \neq m\}$ is regular.

Solution: ....

1. True; given a finite language $\{w_1, \ldots, w_n\}$, a corresponding expression is simply $w_1 + \ldots + w_n$.
2. False; every regular language is context-free.
3. False; take $L = A^*$ for $A = \{a, b\}$ and $K = \{a^n b^n \mid n \geq 0\}$; then $K \cap L = K$ is not regular.
4. False; if L is regular then $\overline{L} \cap \mathcal{L}(a^* b^*) = \{a^n b^n \mid n \geq 0\}$ is regular, contradiction.

□

Problem 2.

You are in charge of the security of a large web-based company, and you should formally describe what a good password looks like. Fortunately, you know regular expressions...

Th alphabet of our passwords will be $\Sigma = \{a, b, c, A, B, C, 0, 1, 2\}$. A *safe password* is a word $w \in \Sigma^*$ such that w begins and ends with a number, w contains at least one lower-case letter and w contains at least one upper-case letter.

Give a regular expression e such that $\mathcal{L}(e) = \{w \in \Sigma^* \mid w \text{ is a safe password}\}$. **(10pt)**

Solution: ....

First, let $e = (a+b+c+A+B+C+0+1+2)^*$. The desired expression is given by:

$$(0+1+2)e(a+b+c)e(A+B+C)e(0+1+2) + (0+1+2)e(A+B+C)e(a+b+c)e(0+1+2)$$

□

Problem 3.

Let $A = \{a, b\}$ be our finite alphabet and define $f : A^* \to A^*$ inductively as follows:

$$f(\lambda) = \lambda \qquad f(av) = f(v)aba \qquad f(bv) = f(v)bb$$

Prove that $f(wa) = abaf(w)$ for all $w \in A^*$, by induction on w. **(10pt)**

Solution: ....

Base case:

$$f(\lambda a) = f(a) = f(a\lambda) = f(\lambda)aba = aba = abaf(\lambda)$$

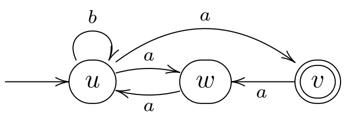
Induction step: suppose $f(wa) = abaf(w)$ for some $w \in A^*$. We need to prove the statement holds again for xw, for every $x \in A$, i.e., $f(awa) = abaf(aw)$ and $f(bwa) = abaf(bw)$:

- $f(awa) = f(wa)aba = abaf(w)aba = abaf(aw)$
- $f(bwa) = f(wa)bb = abaf(w)bb = abaf(bw)$

□

Problem 4.

Consider the following NFA M over the alphabet $A = \{a, b\}$:



a) Show that aaa is accepted by M, and aab is not. **(5pt)**

Solution: ....

For aaa, we have the computation

$$(u, aaa) \vdash (w, aa) \vdash (u, a) \vdash (v, \lambda)$$

and since v is accepting, the word aaa is accepted.

For aab, there are two possibilities: first,

$$(u, aab) \vdash (v, ab) \vdash (w, b)$$

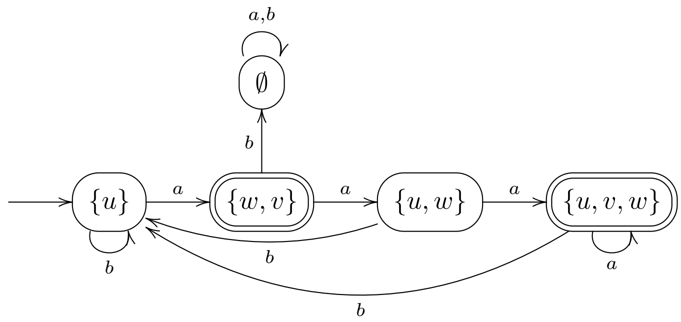
and here the computation is stuck; second,

$$(u, aab) \vdash (w, ab) \vdash (u, b) \vdash (u, \lambda)$$

and u is not accepting. Since there is no computation starting with (u, aab) and ending in an accepting state with the empty word as second component, aab is not accepted. □

b) Use the construction from the lecture to give a deterministic finite automaton N such that $\mathcal{L}(M) = \mathcal{L}(N)$. Clearly show how states in N relate to states in M. **(10pt)**

Solution: ....



□

Problem 5.

Consider the following three languages over the alphabet $A = \{a, b, \#\}$.

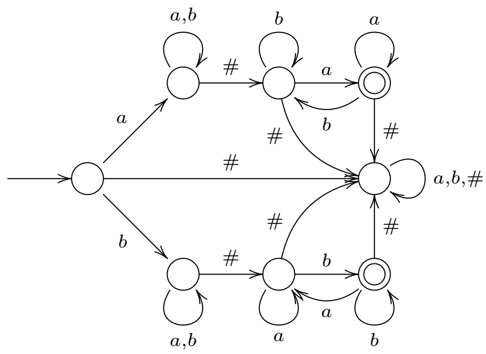
$$L_1 = \{w\#v \mid w, v \in \{a, b\}^* \text{ and the first letter of } w \text{ equals the last letter of } v\}$$
$$L_2 = \{w\#v\#z \mid w, v \in \{a, b\}^* \text{ and } |w|_a = |v|_a = |z|_a\}$$
$$L_3 = \{w\#v\#z\#u \mid w, v, z, u \in \{a, b\}^* \text{ and } |w| = |v| \text{ and } |z| = |u|\}$$

Two of these languages are context-free, and the third is not context-free. One of the three languages is regular.

a) Which of the languages L_1, L_2, L_3 is regular? Give an DFA which accepts that language. **(10pt)**

Solution: ....

L_1 is regular. DFA:



□

b) Which of the languages L_1, L_2, L_3 is context-free but not regular? Give a context-free grammar which generates that language. **(10pt)**

Solution: ....

L_3 is context-free but not regular. Grammar:

$$\begin{array}{lcl} S & \to & T\#T \\ T & \to & aTa \mid bTb \mid aTb \mid bTa \mid \# \end{array}$$

□

c) Choose a non-regular language from L_1, L_2, L_3, and use the pumping lemma to prove that it is indeed not regular. **(10pt)**

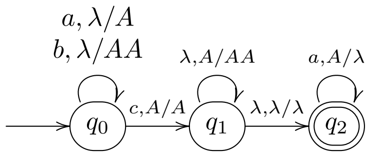
Solution: ....

- Suppose L_2 is regular, and let k be a constant given by the PL. Let $w = a^k\#a^k\#a^k$. By the pumping lemma, there are x, y, z such that $w = xyz$, $|xy| \leq k$, $|y| > 0$ and for all n, $xy^nz \in L_2$. It follows that $y = a^i$ for some $i \geq 0$, hence $xy^2z = a^{k+i}\#a^k\#a^k \notin L_2$, contradiction.
- L_3 is also not regular; the proof is similar to the above, for instance taking $w = a^k\#a^k\#a^k\#a^k$.

□

Problem 6.

Consider the following push-down automaton M over the alphabet $A = \{a, b, c\}$.



a) Is M deterministic? Explain your answer. **(3pt)**

Solution: ....

No; if A is on top of the stack in state q_1, there are two enabled transitions. □

b) Describe the language $\mathcal{L}(M)$ using set notation. **(8pt)**

Solution: ....

$$\mathcal{L}(M) = \{wca^n \mid w \in \{a,b\}^*, 0 < |w|_a + 2|w|_b \leq n\}$$

□

Problem 7.

Consider the language over the alphabet $A = \{(,), [,]\}$ given by the following grammar G:

$$S \to (S) \mid SS \mid [S] \mid \lambda$$

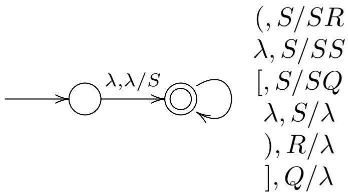
Give a PDA M such that $\mathcal{L}(G) = \mathcal{L}(M)$. **(8pt)**

Solution: ....

We can, for instance, use the construction from the lecture. To this end, first transform the grammar to the correct form:

$$\begin{array}{rcl} S & \to & (SR \mid SS \mid [SQ \mid \lambda \\ R & \to &) \\ Q & \to &] \end{array}$$

From this grammar, we construct a PDA in the standard way:



□