Talen en Automaten

Test 1, Tue 13th Dec, 2016
13h30 – 15h30

This test consists of **8** exercises over **7 pages**. Explain your approach, and **write your answers to the exercises on a separate folio (double pages) as indicated.**. 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, as it was introduced in the lecture. Moreover, recall that v is a *subword* of w if $w = xvy$ for some words x, y.

Write your answers to Problems 1 and 2 on a separate folio (double page)

Problem 1.

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

$$\begin{array}{rcl} f(\lambda) & = & c \\ f(a\,v) & = & a\,f(v) \\ f(b\,v) & = & f(v)\,b \\ f(c\,v) & = & b\,f(v)\,b \end{array}$$

a) Give two different words w and v for which $f(w) = f(v) = bacbb$. **(5pt)**

Solution: ....

cab and cba:

$f(cab) = b\,f(ab)\,b = ba\,f(b)\,b = ba\,f(\lambda)\,bb = bacbb$

$f(cba) = b\,f(ba)\,b = b\,f(a)\,bb = ba\,f(\lambda)\,bb = bacbb.$ □

b) Prove by induction the following property (for all $w \in A^*$) **(10pt)**

$$|f(w)|_b = |w|_b + 2|w|_c.$$

Solution: ....

We show by induction on w that $|f(w)|_b = |w|_b + 2|w|_c$.

- Induction Basis: $w = \lambda$. Here, we get that

$$|f(\lambda)|_b = |c|_b = 0 = |\lambda|_b + 2|\lambda|_c$$

- Induction Step: $w = xv$. For $v \in A^*$, assuming $|f(v)|_b = |v|_b + 2|v|_c$ (the Induction Hypothesis IH), we need to prove $|f(xv)|_b = |xv|_b + 2|xv|_c$ for all $x \in A$.

  Let $x \in A$. We need to distinguish three cases.

1. If $x = a$, then we have
$$\begin{aligned}|f(av)|_b &= |a\, f(v)|_b = |f(v)|_b \\ &\overset{\text{IH}}{=} |v|_b + 2|v|_c = |av|_b + 2|av|_c\end{aligned}$$
as required.
2. If $x = b$, then
$$\begin{aligned}|f(bv)|_b &= |f(v)\, b|_b = 1 + |f(v)|_b \\ &\overset{\text{IH}}{=} 1 + |v|_b + 2|v|_c = |bv|_b + 2|bv|_c.\end{aligned}$$
3. If $x = c$, then
$$\begin{aligned}|f(cv)|_b &= |b\, f(v)\, b|_b = 2 + |f(v)|_b \\ &\overset{\text{IH}}{=} 2 + |v|_b + 2|v|_c = |cv|_b + 2|cv|_c.\end{aligned}$$

So in all cases $|f(xv)|_b = |xv|_b + 2|xv|_c$.

This induction proves the desired equation. □

c) Give a regular language L such that $f(L) := \{f(w) \mid w \in L\}$ is not regular. **(5pt)** (Describe $f(L)$ and argue why $f(L)$ is not regular; a full proof is not required.)

Solution: ....

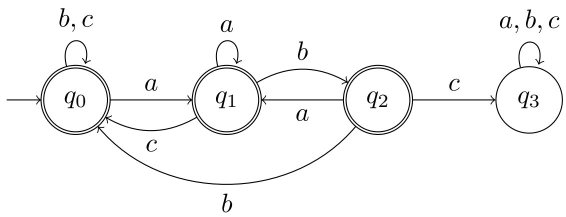
$L = \mathcal{L}((ab)^*)$ is regular. The function f moves all a's in the word to the front and all b's in teh word to the back, so $f(L) = \{a^n b^n \mid n \geq 0\}$, which is not regular as we know from the lectures. □

Problem 2.

Consider the following language L over the alphabet $A = \{a, b, c\}$.
$$L = \{w \mid w \text{ does not contain } abc \text{ as subword}\}$$

a) Give a DFA that accepts L. Explain your answer. **(10pt)**

Solution: ....



Explanation: the automaton keeps track of the last few letters read. Once abc occur as the last three letters read, we move to the trap state. In state q_0 the word read so far does not end in a or ab, in q_1 the last letter read is an a, in state q_2, the last letters are ab. After reading a c in q_2, the word contains abc, and we transition to q_3. □

b) Show that your DFA accepts *abab* and rejects *ababc*. **(5pt)**

Solution: ....

For *abab*:

$$\begin{aligned}(q_0, abab) &\to (q_1, bab)\\ &\to (q_2, ab)\\ &\to (q_1, b)\\ &\to (q_2, \lambda)\end{aligned}$$

and since q_2 is accepting, the word *abab* is accepted. For *ababc*, the beginning is the same:

$$\begin{aligned}(q_0, abab) &\to^* (q_2, c)\\ &\to (q_3, \lambda)\end{aligned}$$

but q_3 is not accepting, so *ababc* is not accepted. □

Write your answers to Problems 3,4 and 5 on a separate folio (double page)

Problem 3.

Give a regular expression for the following language over the alphabet $A = \{a, b\}$. **(10pt)**

$$L = \{w \mid |w|_a \text{ is not a multiple of 3 and } w \text{ does not contain } bb \text{ as a substring}\}$$

Explain your answer.

Solution: ....

$$((1+b)a(1+b)a(1+b)a)^*((1+b)a(1+b) + (1+b)a(1+b)a(1+b))$$

Explanation: $((1+b)a(1+b)a(1+b)a)^*$ represents the language of words w for which $|w|_a$ is a multiple of 3 that does not contain bb and that does not end with a b. To that we must concatenate at the end a word with one a or a word with two a's that does not contain bb. This is what $(1+b)a(1+b) + (1+b)a(1+b)a(1+b)$ repesents.

Alternatives: $((1+b)a(1+b) + (1+b)a(1+b)a(1+b))(a(1+b)a(1+b)a(1+b))^*$

□

Problem 4.

Consider the following language over the alphabet $A = \{a, b\}$.

$$L = \{(ab)^n w \mid w \text{ contains } n \text{ copies of } ab \text{ as subword}\}$$

a) Show that L is not regular. **(10pt)**

Solution: ....

Suppose that L is regular. Let $k \geq 1$ be the "pumping number" that exists due to the Pumping Lemma (PL). We take $w = (ab)^k a(ab)^k$. According to PL, there are x, y, z such that $w = xyz$ with $|xy| \leq k$ and $|y| \geq 1$ such that y can be "pumped". This means that y is a substring of the first $(ab)^k$ block.. Following the PL, $xy^0z \in L$. However, this is not the case because $xy^0z = va(ab)^k$, where $|v| < 2k$, so $v \neq (ab)^k$.

NB1. It is also possible to make a more fine analysis of the shape of y (starts with a or b, ends with a or b) and conclude from there that the word xy^iz (for some $i \neq 1$) is not in the language.

NB2. It is not possible to just take $w = (ab)^k(ab)^k$, because y may consist of an even number of ab's and then one *can* pump.

□

b) Now consider the language K over the alphabet $B = \{a, b, c\}$ given by **(5pt)**

$$K = \{w \in B^* \mid \text{ if } w \in A^* \text{ then } w \notin L\}$$

Is K regular? Explain your answer.

Solution: ....

If K were regular, then $\overline{K \cap A^*} = L$ as well. Contradiction with the previous exercise; hence K is not regular. □

Problem 5.

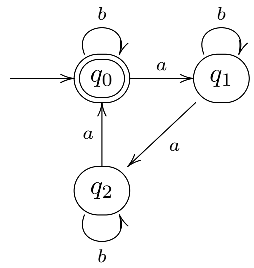
Use the product construction to give a DFA that accepts the following language over the alphabet $A = \{a, b\}$. **(10pt)**

$$L = \{w \mid |w|_a \text{ is a multiple of 3 and } |w|_b \text{ is odd}\}$$

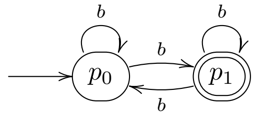
Show how your automaton arises as the product of two automata.

Solution: ....

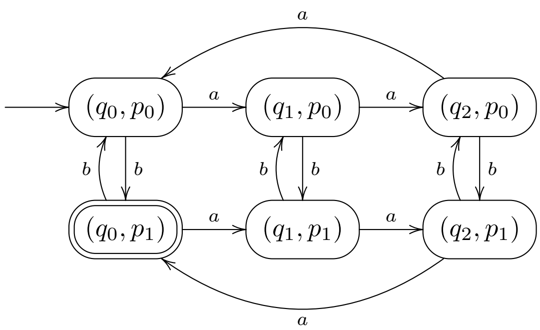
We construct an automaton for $L_1 = \{w \mid |w|_a \text{ is a multiple of } 3\}$:



and for $L_2 = \{w \mid |w|_b \text{ is odd}\}$:



and we construct the product:

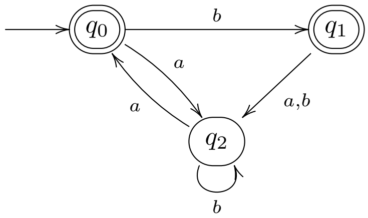


□

Write your answers to Problems 6,7 and 8 on a separate folio (double page)

Problem 6.

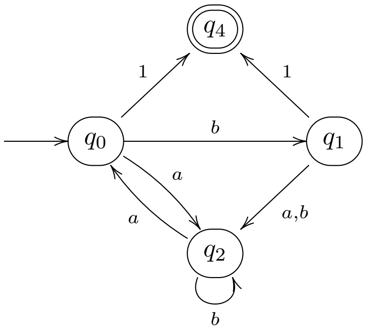
Let $A = \{a, b\}$ and the DFA M over A given by **(10pt)**



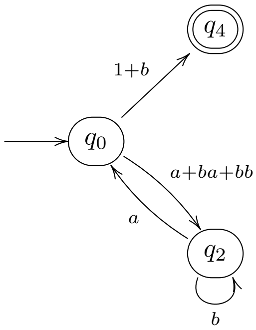
Use the procedure from the lecture to construct a regular expression e such that $\mathcal{L}(M) = \mathcal{L}(e)$. Show each intermediate step.

Solution: ....

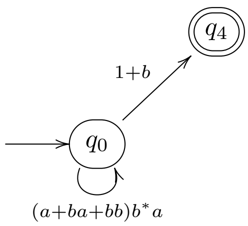
First, we construct an automaton with a single new final state.



Remove q_1:



Remove q_2:



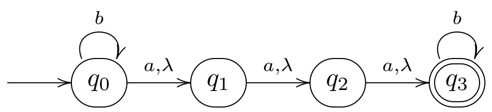
The resulting expression is $((a + ba + bb)b^*a)^*(1 + b)$. □

Problem 7.

Consider the regular expression $e = b^*(1 + a + aa + aaa)b^*$. Give an NFA-$\lambda$ M with at most four states such that $\mathcal{L}(M) = \mathcal{L}(e)$. Explain your answer. **(10pt)**

Solution: . . . .

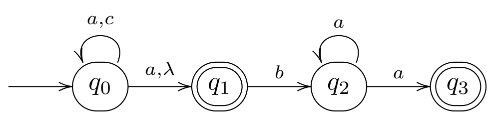
For instance:



Explanation: In q_3 we accept all words consisting of just b's, so starting from q_2 we accept all words of $\mathcal{L}((1 + a)b^*)$. So starting from q_1 we accept all words of $\mathcal{L}((1 + a + aa)b^*)$. So starting from q_0 we accept all words of $\mathcal{L}(b^*(1 + a + aa + aaa)b^*)$. □

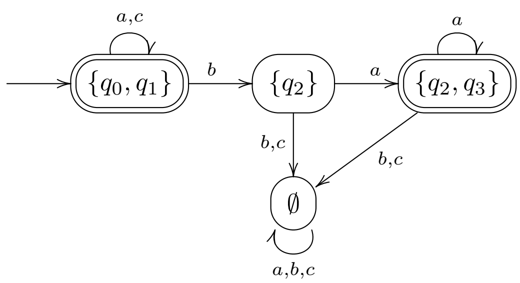
Problem 8.

Let $A = \{a, b, c\}$ and consider the following NFA-λ M over A. **(10pt)**



Use the powerset construction from the lecture to construct a DFA D with $\mathcal{L}(D) = \mathcal{L}(M)$. Indicate clearly from which subset of states in M a state in D originates.

Solution: . . . .



□