Talen en Automaten, NWI-IPC002

Retake exam, Wed 13th Apr, 2016
18:00 – 21:00

This test consists of **six** exercises over **7 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 and $a \in A$ by $|w|_a$ the number of a's in the word $w \in A^*$, as it was introduced in the lecture.

1. Consider the regular expression $e_1 = (\mathtt{a}+\mathtt{b})^*(\mathtt{b}+\mathtt{c})^*(\mathtt{c}+\mathtt{a})^*$ over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

   a) Give the shortest word over Σ that is *not* in $\mathcal{L}(e_1)$ and explain your answer. **(5pt)**

   Solution: ....

   Any one letter word is obviously in the language. For a two-letter word: Both $(\mathtt{a}+\mathtt{b})^*(\mathtt{b}+\mathtt{c})^*$ and $(\mathtt{b}+\mathtt{c})^*(\mathtt{c}+\mathtt{a})^*$ generate any single letter, and e_1 unfolds to $(\mathtt{a}+\mathtt{b})^*(\mathtt{b}+\mathtt{c})^*(\mathtt{b}+\mathtt{c})^*(\mathtt{c}+\mathtt{a})^*$. Thus, any two-letter word is generated.

   Now, consider the word `cab`. The `c` can only be generated by $(\mathtt{b}+\mathtt{c})^*$ and the `a` then only by $(\mathtt{c}+\mathtt{a})^*$. But then the `b` cannot be generated any more, hence $\mathtt{cab} \notin \mathcal{L}(e_1)$.

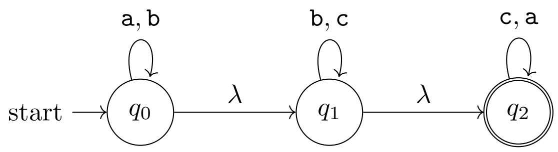
   By this reasoning, `cab` is the shortest word not in $\mathcal{L}(e_1)$.

   Remark: There is the subtle point that the exercise looks for "the" shortest word. It would need to justified that there is actually no other word of length 3 that is not in the language, but we do not expect that here.

   □

   b) Construct a nondeterministic automaton with λ-transitions, i.e. an NFA$_\lambda$, that accepts $\mathcal{L}(e_1)$. Explain your answer. **(7pt)**

   Solution: ....



   □

2. Consider the language $L_2 = \{w \mid w \text{ contains the substring } \mathtt{bbb}\}$ over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$. Here, u is a substring of w, if $w = xuy$ for some words x, y.

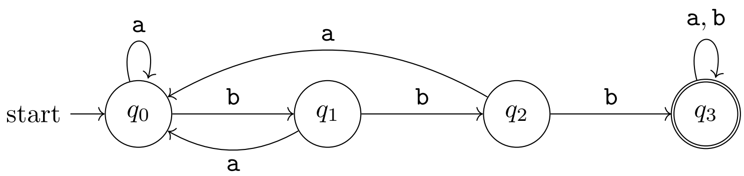
   a) Give a regular expression for L_2 and explain your answer. **(6pt)**

   Solution: ....

We can use $e_2 = (\texttt{a}+\texttt{b})^* bbb(\texttt{a}+\texttt{b})^*$, since clearly $w = xbbby$ for $x, y \in \Sigma^*$ iff $\mathcal{L}(e_2)$. □

b) Give a deterministic finite automaton (DFA) that accepts L_2. **(6pt)**

Solution: . . . .



□

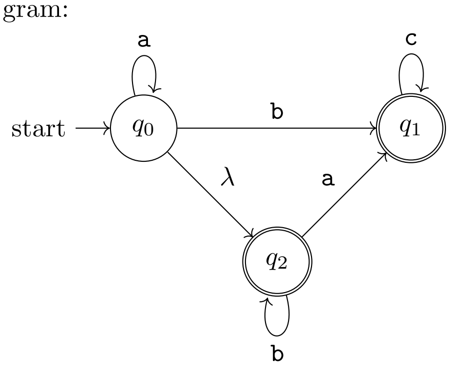
c) Give a regular grammar that generates L_2. **(3pt)**

Solution: . . . .

$$\begin{aligned} S &\to \texttt{a}S \mid \texttt{b}X \\ X_1 &\to \texttt{a}S \mid \texttt{b}X_2 \\ X_2 &\to \texttt{a}S \mid \texttt{b}X_3 \\ X_3 &\to \lambda \mid \texttt{a}X_3 \mid \texttt{b}X_3 \end{aligned}$$

□

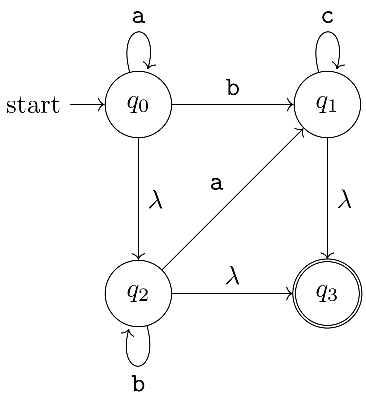
3. Let M_3 be the NFA$_\lambda$ over the alphabet $\{\texttt{a}, \texttt{b}, \texttt{c}\}$ and the following state diagram:



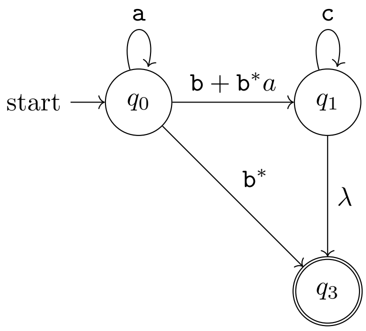
a) Compute a regular expression e_3 such that $\mathcal{L}(e_3) = \mathcal{L}(M_3)$ using the elimination-of-states algorithm. **(5pt)**

Solution: . . . .

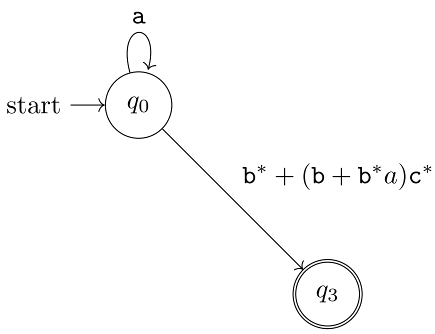
First, we need to add an auxiliary final state:



Next, we eliminate q_2:



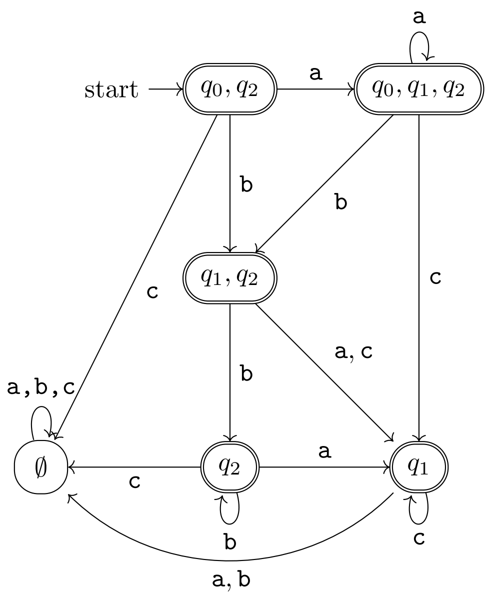
Finally, we eliminate q_1:



This results into the expression $\mathtt{a}^*(\mathtt{b}^* + (\mathtt{b} + \mathtt{b}^* a)\mathtt{c}^*)$. □

b) Construct a DFA (deterministic finite automaton) M_3' using the subset construction that accepts the same language as M_3. Label the states in M_3' so that it is clear which states in M_3 they correspond to. **(8pt)**

Solution: . . . .



□

4. Let G_4 be the following context free grammar over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

$$\begin{array}{lcl} S & \to & \mathtt{a}S \mid ASA \mid SC \mid \lambda \\ A & \to & \mathtt{ab}A \mid \lambda \\ C & \to & \mathtt{c}C \mid \lambda \end{array}$$

a) Give a leftmost derivation in G_4 of $\mathtt{abacab}$. **(3pt)**

Solution: . . . .

$S \to ASA \to \mathtt{ab}ASA \to \mathtt{abSA} \to \mathtt{aba}SA \to \mathtt{aba}SCA \to \mathtt{aba}CA \to \mathtt{abac}CA$
$\to \mathtt{abac}A \to \mathtt{abacab}A \to \mathtt{abacab}$

□

b) Is this grammar ambiguous? Explain your answer. **(3pt)**

Solution: . . . .

It is ambiguous, as we have the derivation

$$S \to ASA \to SA \to S,$$

which can be used in any valid derivation. □

c) Is $\{w \in \Sigma^* \,:\, |w|_{\mathtt{b}} \leq |w|_{\mathtt{a}}\} \subseteq \mathcal{L}(G_4)$? Explain your answer. **(4pt)**

Solution: . . . .

No, because $\mathtt{ba}$ is in that language but not generated by the grammar.

□

d) For which $m, n, k \geq 0$ do we have $\mathtt{a}^m\mathtt{b}^n\mathtt{c}^k \in \mathcal{L}(G_4)$? Explain your answer. **(4pt)**

Solution: ....

Since we have $S \to SC$, and we can produce any sequence of $\mathtt{a}$ from S and of $\mathtt{c}$ from S, have that $a^m c^k \in \mathcal{L}(G_4)$ for all $m, k \in \mathbb{N}$. Thus, $n = 0$ in this case.

If $n > 0$, we must have $S \to SC \to^* a^m SC \to a^m ASAC \to^* a^m abSC$. Thus, $a^{m+1}bc^k$ is also in the generated language for $m, k \in \mathbb{N}$, by a similar argument as above. □

5. Let L_5 be the language

$$L_5 = \{w\mathtt{a}v \mid w, v \in \{\mathtt{a}, \mathtt{b}\}^* \text{ and } |w| \leq |v|\}.$$

a) Give a context-free grammar G_5 that generates L_5. **(7pt)**

Solution: ....

$$S \to \mathtt{a}B \mid XSX$$
$$B \to XB \mid \lambda$$
$$X \to \mathtt{a} \mid \mathtt{b}$$

□

b) Provide derivations in G_5 of the words: $\mathtt{abbb}$, $\mathtt{bbaaaa}$, $\mathtt{baaaab}$. **(3pt)**

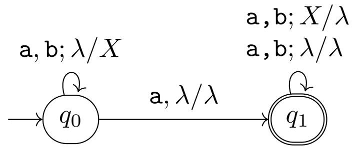
Solution: ....

- $$S \to \mathtt{a}B \to^* \mathtt{a}XXX \to^* \mathtt{abbb}$$
- $$S \to XSX \to X^2SX^2 \to X^2\mathtt{a}BX^2 \to X^2\mathtt{a}BX^2$$
  $$\to X^2\mathtt{a}XBX^2 \to X^2\mathtt{a}X^3 \to^* \mathtt{bbaaaa}$$
- $$S \to^* X^2\mathtt{a}X^3 \to^* \mathtt{baaaab}$$

□

c) Give a pushdown automaton M_5 that accepts L_5. **(8pt)**

Solution: ....



□

d) Provide accepting computations of M_5 of the words: `aab`, `aba`, `babaaa`. **(3pt)**

Solution: ....

- $$\texttt{aab}, q_0, \lambda \to \texttt{ab}, q_0, X \to \texttt{b}, q_1, X \to \lambda, q_1, \lambda$$
- $$\texttt{aba}, q_0, \lambda \to \texttt{ba}, q_1, \lambda \to^* \lambda, q_1, \lambda$$
- $$\texttt{babaaa}, q_0, \lambda \to \texttt{abaaa}, q_0, X \to \texttt{baaa}, q_1, X \to \texttt{aaa}, q_1, \lambda \to^* \lambda, q_1, \lambda$$

□

e) Prove that L_5 is not regular. **(10pt)**

Solution: ....

We prove this by appealing to the pumping lemma. So suppose that L_5 is regular and let p be its pumping length. Then we put $w = b^p a b^p$, which is in L_5. Thus, we get from the PL a splitting $w = uvx$ with $|uv| \leq p$ and $|v| \geq 1$, s.t. $uv^i x \in L_5$ for every $i \in \mathbb{N}$.
From this we get that $u = b^r$ and $v = b^s$ for some $r, s \in \mathbb{N}$ and $s \geq 1$. Thus,

$$uv^2x = b^r b^{2s} b^{p-r+s} a b^p = b^{p+s} a b^p$$

is in L_5. This, however, contradicts the definition of the language, as $p + s > p$. Hence, L_5 cannot be regular. □

6. Let $M_6 = (Q, \Sigma, \Gamma, \delta, q_0, F)$ be the PDA with

$Q = \{q_0, q_1, q_2\}$	$\delta(q_0, \texttt{a}, \lambda) = \{(q_0, A)\}$
$\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}\}$	$\delta(q_0, \texttt{a}, A) = \{(q_0, \lambda)\}$
$\Gamma = \{A, B, C\}$	$\delta(q_0, \lambda, \lambda) = \{(q_1, \lambda)\}$
$F = \{q_0\}$	$\delta(q_1, \texttt{b}, \lambda) = \{(q_1, B)\}$
	$\delta(q_1, \texttt{c}, B) = \{(q_2, C)\}$
	$\delta(q_2, \lambda, C) = \{(q_0, \lambda)\}$

a) Draw the state diagram of M_6. **(6pt)**

Solution: ....

... □

b) Check which of the following words are in $\mathcal{L}(M_6)$ and explain your answer by giving a computation or showing that there is none: `aaaa`, `abc`, `abbcca`. **(3pt)**

Solution: ....

- `aaaa` is accepted with the following computation

$$\texttt{aaaa}, q_0, \lambda \to \texttt{aaa}, q_0, A \to \texttt{aa}, q_0, AA \to \texttt{a}, q_0, A \to \lambda, q_0, \lambda$$

- A computation for `abc` has to start with

$$\texttt{abc}, q_0, \lambda \to \texttt{bc}, q_0, A.$$

  But the A can only be removed from the stack if another `a` is read. Thus, `abc` is not accepted.
- `abbcca` is accepted by the following computation.

$$\texttt{abbcca}, q_0, \lambda \to \texttt{bbcca}, q_0, A \to \texttt{bbcca}, q_1, A \to^* \texttt{cca}, q_1, BBA \to \texttt{ca}, q_2, CBA$$
$$\to \texttt{ca}, q_0, BA \to \texttt{ca}, q_1, BA \to \texttt{a}, q_2, CA \to \texttt{a}, q_0, A \to \lambda, q_0, \lambda$$

□

c) Describe $\mathcal{L}(M_6)$ using set notation. That is, write **(6pt)**

$$\mathcal{L}(M_6) = \{\dots\},$$

with ... a precise mathematical description of the words accepted by M_6.

Solution: ....

$$\mathcal{L}(M_6) = \{\texttt{a}^n\texttt{b}^m\texttt{c}^m\texttt{a}^n \mid m, n \geq 0\}.$$

□