Talen en Automaten

Test 2, Wed 18$^{\text{th}}$ Jan, 2017
8h30 – 11h30

This test consists of **5** problems 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.

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

- $L_1 = \{wcvcz \mid w, v, z \in \{a, b\}^* \text{ and }, |w|_a = |v|_a = |z|_a\}$.
- $L_2 = \{w \mid w \text{ does not contain } bb \text{ as subword}\}$.
- $L_3 = \{wb^n \mid |w|_b = n, n \geq 0\}$

One of L_1, L_2, L_3 is regular, one is context-free but not regular and one is not context free.

a) Which of the languages is regular? Show this by giving a regular grammar for this language. **(8pt)**

b) Which of the languages is context free but not regular? Give a context free grammar for this language. **(8pt)**

c) Is $L_2 \cap L_3$ regular? If so, give a regular grammar for it. Otherwise argue that it is not regular. (You don't have to give a full proof.) **(8pt)**

Solution: ....

a) L_2 is regular and generated by

$$\begin{array}{lcl} S & \rightarrow & aS \mid cS \mid bB \mid \lambda \\ B & \rightarrow & aS \mid cS \mid \lambda \end{array}$$

Here, S generates any number of a's and c's. In case S generates a b, we move to B and in B we can generate anything but a b.

b) L_3 is context-free (and not regular). L_3 is generated by

$$\begin{array}{lcl} S & \rightarrow & ASb \mid aS \mid cS \mid \lambda \\ A & \rightarrow & aA \mid Aa \mid cA \mid Ac \mid b \end{array}$$

Here, A generates 0 or more a's or c's and one b, so a word over the alfabet with exactly one b. S generates a sequence of n b's at the right and at the left any word over $\{a, b, c\}$ with n times a b.

c) Yes, $L_2 \cap L_3$ is regular: $L_2 \cap L_3 = \mathcal{L}((a + c)^* b(a + c)(a + c)^* b + (a + c)^*)$, the language of words with no b's or with two b's where one b is at the end and the other b not immediately before it. A regular grammar for $L_2 \cap L_3$ is

$$\begin{array}{lcl} S & \rightarrow & aS \mid cS \mid bA \mid \lambda \\ A & \rightarrow & aA \mid cA \mid aB \\ B & \rightarrow & b \end{array}$$

□

Problem 2.

Consider the following context-free grammar G over $\{a, b, c\}$.

$$\begin{array}{rcl} S & \rightarrow & a\,S\,b \mid C\,X \mid \lambda \\ C & \rightarrow & c\,C \mid \lambda \\ X & \rightarrow & S\,c \end{array}$$

a) Indicate for the following words if they are generated by G: ab, $aabba$, $abab$. Explain your answer. (So give a derivation in case the word is in $\mathcal{L}(G)$ and otherwise give an argument why it is not.) **(6pt)**

b) Use the procedure from the lecture to construct a PDA (push-down automaton) that accepts the language generated by G. **(8pt)**

c) Can all words of the shape $(ac)^n ab(cb)^n$ (with $n \geq 0$) be produced by G_1? Prove your answer. **(7pt)**

Solution: ....

a) ab: $S \rightarrow aSb \rightarrow ab$, so $ab \in \mathcal{L}(G)$.

The production rule $S \rightarrow CX$ always results in a word that contains a c, because $X \rightarrow Sc$ is the only production for X. Therefore the words in $\mathcal{L}(G)$ that contain only letters from $\{a, b\}$ do not use that production rule, so they are generated by the grammar $S \rightarrow aSb \mid \lambda$. This language is $\{a^n b^n \mid n \geq 0\}$. $aabba$ and $abab$ don't contain a c and they are not of the form $a^n b^n$ for some n, so these two words are not in $\mathcal{L}(G)$.

Another way to show that these two words are not in the language is by giving all possible failing derivations.

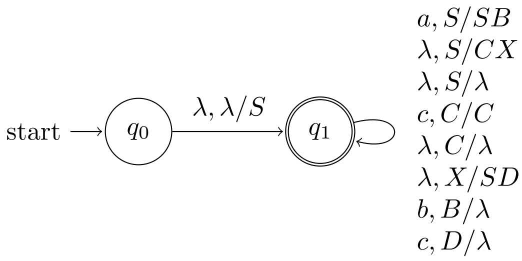
$aabba$:
$S \rightarrow \lambda$ ↯
$S \rightarrow CX \rightarrow CSc$ ↯
$S \rightarrow aSb \rightarrow ab$ ↯
$S \rightarrow aSb \rightarrow aCXb \rightarrow aCScb$ ↯
$S \rightarrow aSb \rightarrow aaSbb \rightarrow aabb$ ↯
$S \rightarrow aSb \rightarrow aaSbb \rightarrow aaaSbbb$ ↯
$S \rightarrow aSb \rightarrow aaSbb \rightarrow aaCXbb \rightarrow aaCScbb$ ↯
So $aabba \notin \mathcal{L}(G)$.

$abab$:
$S \rightarrow \lambda$ ↯
$S \rightarrow CX \rightarrow CSc$ ↯
$S \rightarrow aSb \rightarrow ab$ ↯
$S \rightarrow aSb \rightarrow aCXb \rightarrow aCScb$ ↯
$S \rightarrow aSb \rightarrow aaSbb$ ↯
So $abab \notin \mathcal{L}(G)$.

b) Using the procedure from the lectures, first transform the grammar to

$$\begin{array}{rcl} S & \rightarrow & a\,S\,B \mid C\,X \mid \lambda \\ C & \rightarrow & c\,C \mid \lambda \\ X & \rightarrow & S\,D \\ B & \rightarrow & b \\ D & \rightarrow & c \end{array}$$

Then construct the PDA



c) Yes. Proof by induction on n. First of all, for $n = 0$, $(ac)^0ab(cb)^0 = ab \in \mathcal{L}(G)$ as shown in part (a).
Now assume that for some n, $(ac)^nab(cb)^n \in \mathcal{L}(G)$. This means that $S \Rightarrow (ac)^nab(cb)^n$. Then we have
$S \to aSb \to aCXb \to acCXb \to acXb \to acScb \Rightarrow ac(ac)^nab(cb)^ncb = (ac)^{n+1}ab(cb)^{n+1}$, so, $(ac)^{n+1}ab(cb)^{n+1} \in \mathcal{L}(G)$.
Therefore we can conclude by induction that $(ac)^nab(cb)^n \in \mathcal{L}(G)$ for all $n \geq 0$.

Alternative: A proof with ellipsis/braces: We have

$$S \to aSb \to aCXb \to acCXb \to acXb \to acScb.$$

By repeating this n times we get

$$S \underbrace{\to \cdots \to}_{n} (ac)^nS(cb)^n \to (ac)^naSb(cb)^n \to (ac)^nab(cb)^n.$$

So $(ac)^nab(cb)^n \in \mathcal{L}(G)$ for all $n \geq 0$.

□

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

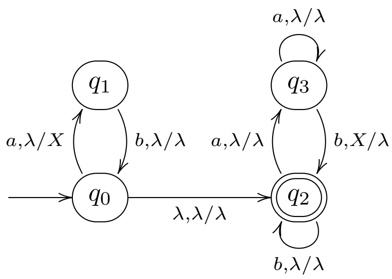
Problem 3.

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

$$L = \{(ab)^n w \mid w \text{ contains } n \text{ copies of } ab \text{ as subword, for some } n \geq 0\}$$

a) Give a PDA that accepts L. **(10pt)**

Solution: . . . .



□

b) Show that *abaabb* and *abbab* are accepted by your automaton, by giving the accepting computations. **(4pt)**

Solution: . . . .

$$\begin{aligned}(q_0, abaabb, \lambda) &\to (q_1, baabb, X)\\ &\to (q_0, baabb, X)\\ &\to (q_0, aabb, X)\\ &\to (q_2, aabb, X)\\ &\to (q_3, abb, \lambda)\\ &\to (q_3, bb, \lambda)\\ &\to (q_2, b, \lambda)\\ &\to (q_2, \lambda, \lambda)\end{aligned}$$

$$\begin{aligned}(q_0, abaabb, \lambda) &\to^* (q_0, bab, X)\\ &\to (q_2, bab, X)\\ &\to (q_2, ab, X)\\ &\to^* (q_2, \lambda, \lambda)\end{aligned}$$

□

c) Show that *ababab* is not accepted by your automaton. **(4pt)**

Solution: . . . .

From the configuration $(q_0, ababab, \lambda)$ there are initially three possibilities: read ab once and put X on the stack (returning to q_0), do this twice, or thrice. We check that neither of the three end up in an accepting state with an empty stack.

$$\begin{aligned}(q_0, ababab, \lambda) &\to^* (q_0, abab, X)\\ &\to (q_2, abab, X)\\ &\to^* (q_2, ab, lambda) \quad ↯\end{aligned}$$

where ↯ means no more transition is enabled.

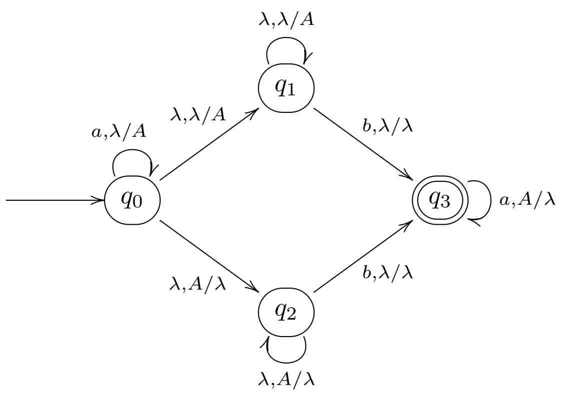
$$\begin{aligned}(q_0, ababab, \lambda) &\to^* (q_0, ab, XX)\\ &\to (q_2, ab, XX)\\ &\to^* (q_2, \lambda, X) \quad ↯\end{aligned}$$

$$\begin{aligned}(q_0, ababab, \lambda) &\to^* (q_0, \lambda, XXX)\\ &\to (q_2, \lambda, XXX) \quad ↯\end{aligned}$$

□

Problem 4.

We define the PDA M, with input alphabet $\Sigma = \{a, b\}$ and stack alphabet $\Gamma = \{A\}$, as follows.



a) Show that $aaba$ and $abaa$ are accepted by M. **(4pt)**

Solution: ....

$$\begin{aligned}(q_0, aaba, \lambda) &\to^* (q_0, ba, AA)\\ &\to (q_2, ba, A)\\ &\to (q_3, a, A)\\ &\to (q_3, \lambda, \lambda)\end{aligned}$$

$$\begin{aligned}(q_0, abaa, \lambda) &\to (q_0, baa, A)\\ &\to (q_1, baa, AA)\\ &\to (q_3, aa, AA)\\ &\to^* (q_3, \lambda, \lambda)\end{aligned}$$

□

b) Show that $aabaa$ is not accepted by M. **(4pt)**

Solution: ....

All a's must be read in q_0 to ever reach the accepting state. After reading these a's, we transition either to q_1 or to q_2. In the first case:

$$\begin{aligned}
(q_0, aabaa, \lambda) &\to^* (q_0, baa, AA) \\
&\to (q_1, baa, AAA) \\
&\to^* (q_1, baa, A^i) \qquad i \geq 3 \\
&\to (q_3, aa, A^i) \qquad i \geq 3
\end{aligned}$$

and the stack can never be emptied, so the word is not accepted. In the second case:

$$\begin{aligned}
(q_0, aabaa, \lambda) &\to^* (q_2, baa, A) \\
&\to (q_3, aa, A) \\
&\to (q_3, a, \lambda) \quad ↯
\end{aligned}$$

or

$$\begin{aligned}
(q_0, aabaa, \lambda) &\to^* (q_2, baa, A) \\
&\to (q_2, baa, \lambda) \\
&\to (q_3, aa, \lambda) \quad ↯
\end{aligned}$$

□

c) Is M deterministic? Explain your answer. **(4pt)**

Solution: ....

No, M is not deterministic. For instance, on reading an a in the initial configuration, there are two transitions enabled (either to stay in q_0 or to go to q_1). □

d) Give a precise description of $\mathcal{L}(M)$ using set notation. **(10pt)**

Solution: ....

$$\mathcal{L}(M) = \{a^i b a^j \mid i \neq j\}$$

□

Write your answers to Problem 5 on a separate folio (double page)

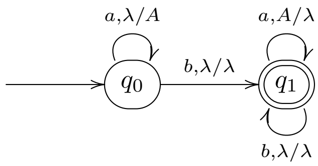
Problem 5.

Let M be the PDA with

$$\begin{aligned}
Q &= \{q_0, q_1\} & \delta(q_0, a, \lambda) &= \{\langle q_0, A\rangle\} \\
\Sigma &= \{a, b\} & \delta(q_0, b, \lambda) &= \{\langle q_1, \lambda\rangle\} \\
\Gamma &= \{A\} & \delta(q_1, a, A) &= \{\langle q_1, \lambda\rangle\} \\
F &= \{q_1\} & \delta(q_1, b, \lambda) &= \{\langle q_1, \lambda\rangle\}
\end{aligned}$$

a) Draw a state diagram for M. **(5pt)**

Solution: ....



□

b) Construct a CFG G such that $\mathcal{L}(M) = \mathcal{L}(G)$. Explain your answer. **(10pt)**

Solution: ....

$$
\begin{aligned}
S &\to (q_0, q_1) \\
(q_0, q_0) &\to \lambda \mid b(q_1, q_0) \mid a(q_0, q_1)a(q_1, q_0) \\
(q_0, q_1) &\to b(q_1, q_1) \mid a(q_0, q_1)a(q_1, q_1) \\
(q_1, q_0) &\to b(q_1, q_0) \\
(q_1, q_1) &\to \lambda \mid b(q_1, q_1)
\end{aligned}
$$

If one optimizes this grammar, one obtains

$$
\begin{aligned}
S &\to bB \mid aSaB \\
B &\to bB \mid \lambda
\end{aligned}
$$

□