Talen en Automaten

Test 2, Wed 28$^{\text{th}}$ Jan, 2015

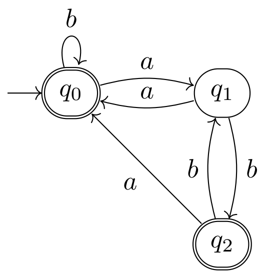
13h45 – 15h45

This test consists of **four** exercises over **2 pages**. It is advised to explain your approach and to check your answers carefully. You can score a maximum of 100 points. 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, your student number, and your werkcollege group. Put your student-card clearly visible at the corner of your table for inspection.

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 exercises.

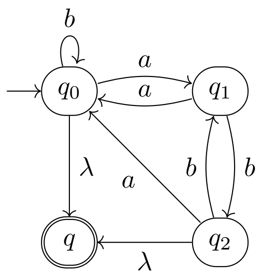
1 Non-deterministic Finite Automata

a) Let $\mathcal{N}$ be the NFA given by the following diagram



i) Give a λ-NFA $\mathcal{N}'$ with one final state that accepts the same language as $\mathcal{N}$. **(5pt)**

Solution: ....



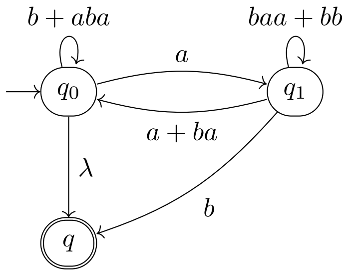
□

ii) Construct from $\mathcal{N}'$ a regular expression that generates the language accepted by $\mathcal{N}'$, using the procedure from the lecture. All intermediate steps belong to your answer. **(12pt)**

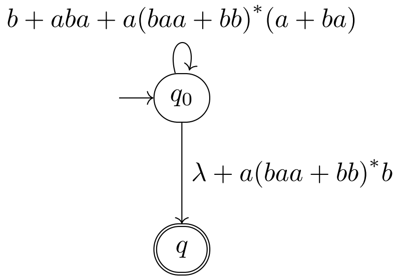
Solution: ....

There are two possible ways of collapsing states.

1. First collapse to



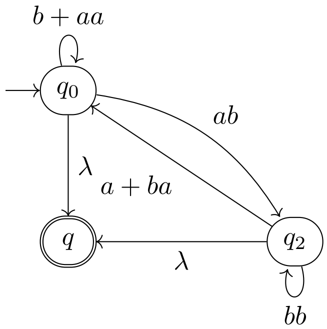
and then to



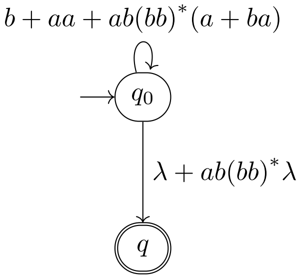
This results into the regular expression

$$(b + aba + a(baa + bb)^*(a + ba))^*(\lambda + a(baa + bb)^*b).$$

2. The alternative is to first collapse q_1. This gives



and then

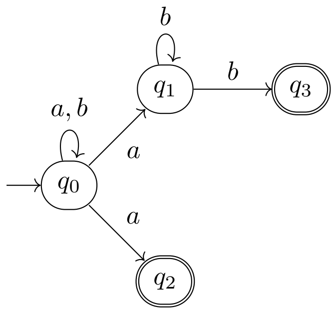


and this results into

$$(b + aa + ab(bb)^*(a + ba))^*(\lambda + ab(bb)^*).$$

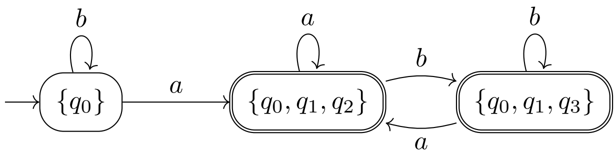
□

b) Let $\mathcal{N}$ be the NFA over the alphabet $\{a, b\}$ be given by the following diagram.



Use the subset construction to obtain a DFA that accepts the same language as $\mathcal{N}$. Leave out unreachable states and clearly mark the states by the set of states they are generated from. **(8pt)**

Solution: . . . .



□

2 Pumping Lemma for Regular Languages

Let A be the alphabet $\{a, b\}$ and L the language

$$L = \{vv^R \mid v \in A^*, |v|_a + |v|_b = 2k + 1, k \in \mathbb{N}\}.$$

Use the pumping lemma to show that L is not regular. **(15pt)**
Take care that the word, you choose in the contradiction, is indeed in L.
Solution: . . . .

We assume that L is regular and has the pumping length $p > 0$. The word we use in the contradiction is $w = vv^R = a^n bba^n$ with

$$n = \begin{cases} p, & p \text{ even} \\ p+1, & p \text{ odd} \end{cases}.$$

Clearly, $w \in L$, as $|v|_a + |v|_b$ is odd. By the pumping lemma, there are x, u, y with $w = xuy$, $|xu| \leq p$ and $|x| = s > 0$, such that $xu^i y \in L$ for all $i \in \mathbb{N}$. Observe that we have $xu = a^k$ for some $k > 0$, as $|xu| \leq p$. Thus $xu^2y \neq v'v'^R$ for any $v' \in A^*$, since $xu^2y = a^{k-s}a^{2s}a^{n-k}bba^n = a^{n+s}bba^n$ and $s > 0$, and xu^2y cannot be in L, which is a contradiction. Hence L is not regular. □

3 Context Free Grammars

Let A be the alphabet $\{a, b\}$ and L again be the language

$$L = \{vv^R \mid v \in A^*, |v|_a + |v|_b = 2k + 1, k \in \mathbb{N}\}.$$

a) Give a grammar G that generates the language L. **(15pt)**

Solution: ....

A possible grammar G is given by

$$S \to aXa \mid bXb$$
$$X \to aSa \mid bSb \mid \lambda$$

□

b) Show that the word *abbbba* is generated by G. **(5pt)**

Solution: ....

$$S \to aXa \to abSba \to abbXbba \to abbbba$$

□

c) Show that the words *aba* and *abba* are *not* generated. **(15pt)**

Solution: ....

- To generate *aba*, we need to use $S \to aXa$ to generate the a at the beginning. But then cannot use any other production, since all productions of X generate either zero or two b, contradicting the shape of *aba*. Hence *aba* is not generated by G.
- Here, we need to use again $S \to aXa$ to generate the a at the beginning. Next, we need to use $X \to bSb$, to generate the b after the first a. But now we are stuck with *abSba*, since there is no production $S \to \lambda$. Thus *abba* is not generated.

□

4 Push Down Automata

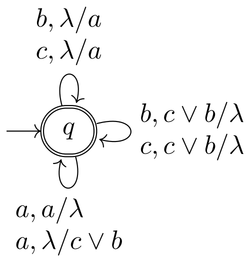
a) Let $\Sigma = \{a, b, c\}$ and the language L be given by

$$L = \{w \in \Sigma^* \mid |w|_a = |w|_b + |w|_c\}.$$

Give a PDA with one state that accepts exactly the language L. Clearly indicate the stack alphabet you are using. Moreover, give the accepting computation for the word *abca*. **(13pt)**

Solution: ....

Since each a can be accounted for by b or c in words of L, we use the stack alphabet $\Gamma = \{a, b \vee c\}$ (or any other two-symbol alphabet). A possible PDA accepting L is then given by



The word *abca* is accepted by

$$(q, abca, \lambda) \to (q, bca, c \vee b) \to (q, ca, \lambda) \to (q, a, a) \to (q, \lambda, \lambda).$$

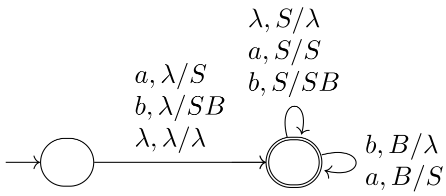
□

b) Let G be the grammar on the alphabet $\{a, b\}$ given by

$$S \to \lambda \mid aS \mid bSB$$
$$B \to b \mid aS$$

Construct a two state PDA that accepts the language generated by G. **(12pt)**

Solution: ....



□