Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Mohamad Ahmadi

Theoretical Computer Science - Bridging Course
Summer Term 2018
Exercise Sheet 3

for getting feedback submit electronically by 06:00 am, Friday, May 11th, 2018

Exercise 1: Regular Expresions *(6 Points)*

Give a regular expression for each of the following three languages.

(a) $L_1 = \{w_1w_2w_3 \mid w_1, w_2, w_3 \in \{a, b, c\}^*,\ w_1 \text{ contains no } a,\ w_2 \text{ contains no } b,\ w_3 \text{ contains no } c\}$

(b) $L_2 \subseteq \{a, b\}^*$ is the language of all words that do not have any of the words in $\{aaa, aaaa, \ldots\}$ as a consecutive substring.

(c) L_3 is the language, over alphabet $\{a, b\}$, of all strings not ending with aa.

Sample Solution

(a) $(b + c)^* \cdot (a + c)^* \cdot (a + b)^*$.

(b) $b^* + (b^*(a + aa)b^+)^*(a + aa)b^*$ or alternatively $(\epsilon + a + aa)(b + ba + baa)^*$.

(c) $\epsilon + a + b + (a + b)^*(ab + ba + bb)$

Exercise 2: The Pumping Lemma: Sufficiency or Necessity? *(4 Points)*

Consider the language $L = \{c^m a^n b^n \mid m, n \geq 0\} \cup \{a, b\}^*$ over the alphabet $\Sigma = \{a, b, c\}$.

(a) Describe in words (not using the pumping lemma), why L cannot be a regular language.

(b) Show that the property described in the Pumping Lemma is a necessary condition for regularity but not sufficient for regularity.

Hint: Use L as counter example, i.e., show that it can be 'pumped' (in the sense of the pumping lemma), but is still not regular.

Sample Solution

(a) For recognizing that a word has the same number of a's and b's, a DFA would have to count the number of appearances of these characters, requiring at least one state for each appearance. But as the number of appearances can be arbitrary large, the automaton would need an infinite number of states.

(b) We show that L has the properties described in the Pumping Lemma. Then we showed that for a language, having these properties do not imply regularity.

As the pumping length we choose an arbitrary $p \geq 1$. Let x be some word of length at least p. We must show that there is a composition $x = uvw$ having the three properties from the lemma:

(1) $|v| \geq 1$

(2) $|uv| \leq p$

(3) for all $i = 0, 1, 2, \ldots$ it holds: $uv^iw \in L$

This is clear if $x \in \{a, b\}^*$. So assume $x = c^m a^n b^n$ with $m \geq 1$. We can choose $u = \epsilon$, $v = c$ and $w = c^{m-1}a^n b^n$ as a composition of x having properties 1, 2, 3.

Exercise 3: Application of The Pumping Lemma *(6 Points)*

Show that the following languages over the alphabet $\Sigma = \{a, b\}$ are not regular:

(a) $L = \{a^m \mid \text{m is a square number}\}$ *(m being a square number means that $m = n^2$ for some non negative integer n)*

Hint: Use the Pumping Lemma.

(b) $L = \{a^m b^n \mid m \neq n\}$

Hint: Have a look at the languages $\{a^n b^n \mid n \in \mathbb{N}\}$ and a^*b^* and use the fact that the class of regular languages is closed under intersection, complement, concatenation and the Kleene star.

Sample Solution

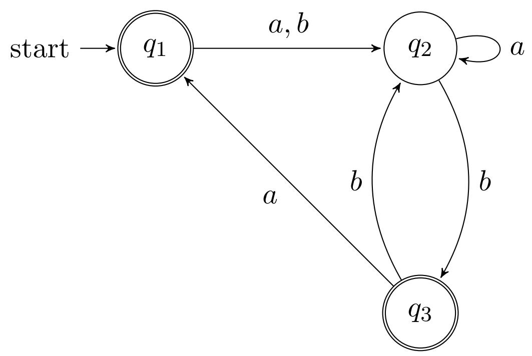
a) We show that the properties from the Pumping Lemma do not hold for L. This means that for all numbers p there is a word x such that for every composition of x, properties 1, 2, 3 do not hold all together.

Let p be some number and $x = a^{p^2}$. Clearly we have $x \in L$. Regard any composition $x = uvw$ with $|v| \geq 1$ and $|uv| \leq p$ (i.e., property 1 and 2). Especially we have $|v| \leq p$ and with $|uvw| = p^2$ we get $|uv^2w| = |uvw| + |v| \leq p^2 + p$. With $p^2 + p < p^2 + 2p + 1 = (p+1)^2$ it follows $|uv^2w| < (p+1)^2$. On the other hand, because of $|v| \geq 1$ we get $|uv^2w| > |uvw| = p^2$.
So $|uv^2w|$ lies strictly between p^2 and $(p+1)^2$, which makes it impossible to be a square number. Thus uv^2w does not lie in L in contrast to property 3 of the Pumping Lemma.

b) If L was regular, then also $\overline{L} \cap (a^*b^*) = \{a^n b^n \mid n \geq 0\}$. But from this set, we already know (from the lecture) that it is not regular, so L is not regular either.

Exercise 4: NFAs to Regular Expressions *(4 Points)*

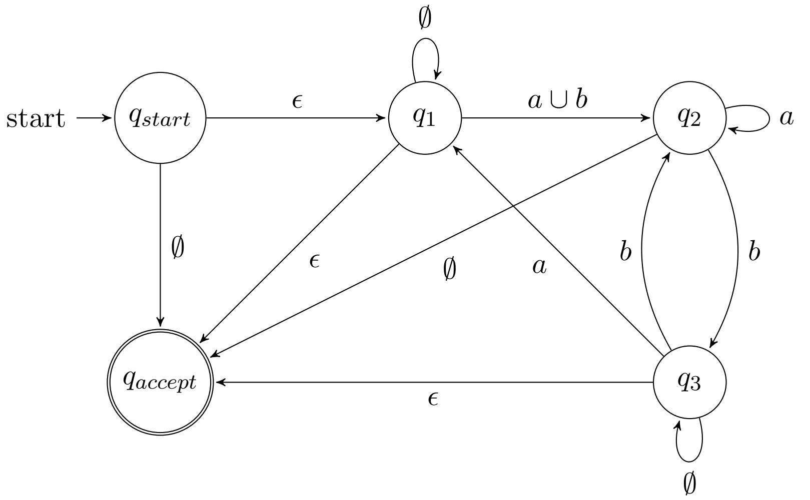
Consider the following NFA:



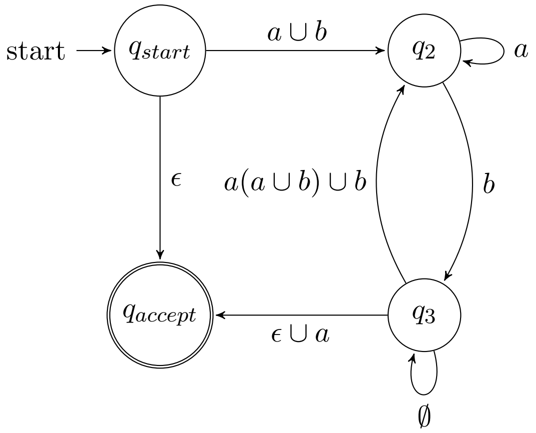
Give the regular expression defining the language recognized by this NFA by *stepwise* converting it into an equivalent GNFA with only two nodes.

Sample Solution

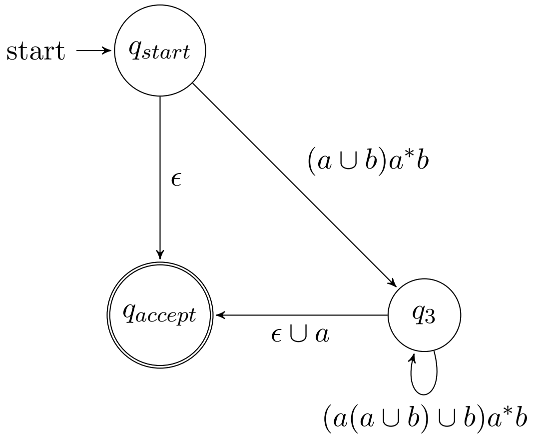
1) Add a new start and accepting state, connect them with ϵ transitions from/to the previous start/accept states, replace multiple labels with unions, add transitions with $\emptyset$ when not present in the original DFA (for a better readability, some edges with label $\emptyset$ are left out in the following diagram):



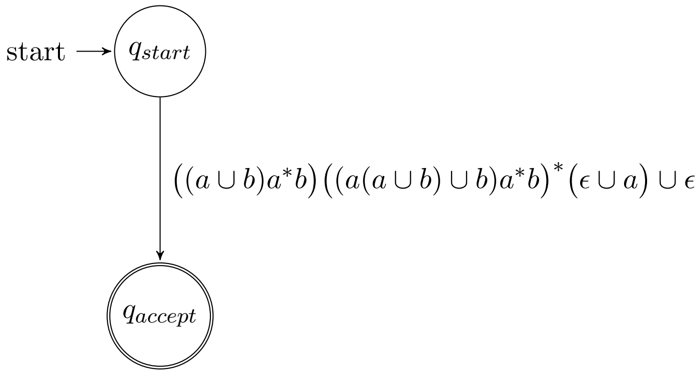
2) Rip off q_1:



3) Rip off q_2:



4) Rip off q_3:


start
q_{start}
$\big((a \cup b)a^*b\big)\big((a(a \cup b) \cup b)a^*b\big)^*\big(\epsilon \cup a\big) \cup \epsilon$
q_{accept}