Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, P. Bamberger, H. Ghodselahi, Y. Maus, P. Schneider

Theoretical Computer Science - Bridging Course
Summer Term 2017
Exercise Sheet 3

Hand in (electronically or hard copy) by 12:15 pm, May 29th, 2017

Exercise 1: Regular Expressions (3 points)

Give a regular expression, which defines the language U of well-formed URL's (Uniform Resource Locators) or a meaningful, *non-finite* subset thereof. An example of a well-formed URL $u \in U$ is

$$u = \texttt{https://en.wikipedia.org/wiki/URL}$$

Assume that the alphabet is $\Sigma = A \cup B$ with $A = \{a, b, c, \dots, y, z\}$ and $B = \{:, /, .\}$ (feel free to extend the alphabet if necessary).

Hint: You can assume that any finite sets, which you require to construct your regular expression, like e.g. the set of valid top level domains $D := \{com, de, org, net, \dots\} \subseteq \Sigma^*$, are given (briefly describe any set you are using). Your solution does not have to cover all aspects of URL's. Restrict yourself to certain cases if you feel that otherwise your regular expression is becoming too complex.

Solution

We construct a regular expression according to the Wikipedia article (not considering every part described in there): A url has to start with a string from A (this is a restriction, in fact digits, plus (+), period (.), or hyphen (-) are also allowed) followed by a colon (:). So the regular expression starts with A^+ :. A is a regular expression if we think of it as the union of all characters from A. A^+ is regular because it is defined as AA^*. Some schemes, but not all, require two slashes after the colon, so in the regular expression we write $A^+ : (\epsilon \cup //)$. For the hostname we also use A^+. The path is a sequence of strings from A separated by slashes. The path must begin with a single slash. The regular expression for this is therefore $(/A^+)^*$. We use the Kleene Star because the path can be empty. So on the whole we have

$$A^+ : (\epsilon \cup //)A^+(/A^+)^*.$$

Exercise 2: Limits of the Pumping Lemma (2+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 can not 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.

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: Applications of the Pumping Lemma (3+3 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.

Solution

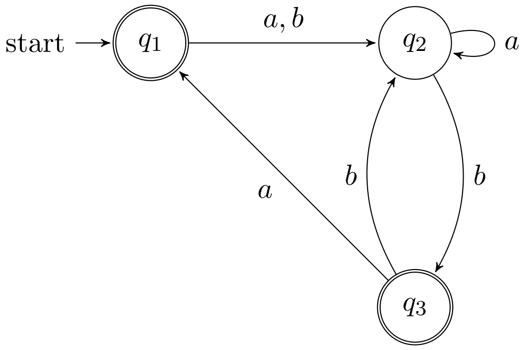
a) We show that the properties from the Pumping Lemma do not hold for L. This means that for all numbers p there is a word x such that for every composition of x, properties 1, 2, 3 do not hold all together.

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 now (from the lecture) that it is not regular, so L is not regular either.

Exercise 4: GNFA (5 points)

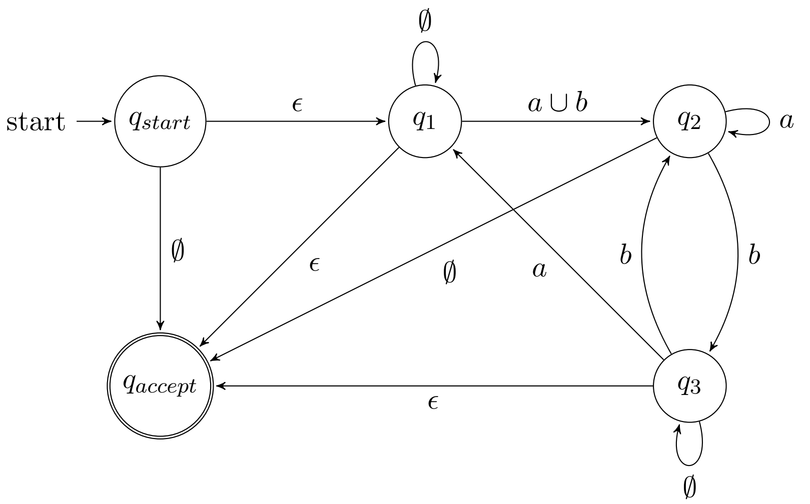
Consider the following NFA:



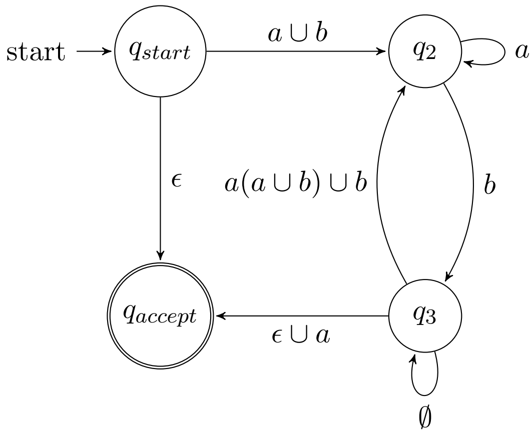
Give the regular expression defining the language recognized by this NFA by *stepwise* converting it into an equivalent GNFA with only two nodes.

Solution

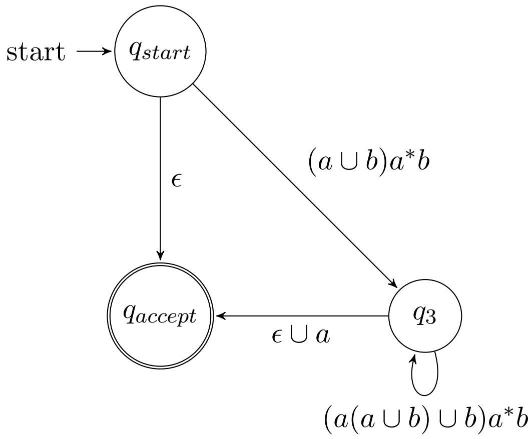
1) Add a new start and accepting state, connect them with ϵ transitions from/to the previous start/accept states, replace multiple labels with unions, add transitions with $\emptyset$ when not present in the original DFA (for a better readability, some edges with label $\emptyset$ are left out in the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



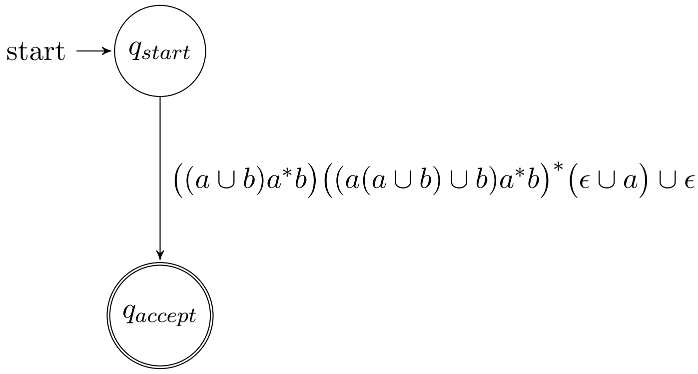
2) Rip off q_1:



3) Rip off q_2:


start
q_{start}
$(a \cup b)a^*b$
ϵ
q_{accept}
q_3
$\epsilon \cup a$
$(a(a \cup b) \cup b)a^*b$

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}