Theory of Computer Science

G. Röger
Spring Term 2020

University of Basel
Computer Science

Exercise Sheet 6 — Solutions

Exercise 6.1 (Regular Expressions and Pumping Lemma for Regular Languages; 3 marks)

Are the following languages over $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ regular? If so, prove it by specifying a regular expression which describes the language. If not, prove it with help of the Pumping-Lemma.

(a) $L_1 = \{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^{n+m} \mid n, m \in \mathbb{N}_0\}$

Solution:

Assume L_1 is regular. Let p be a pumping number of L_1. The word $w = \mathtt{a}^p\mathtt{b}\mathtt{c}^{p+1}$ is in L_1 and satisfies $|w| \geq p$. We know from the pumping lemma that there are words x, y and z with $w = xyz$, $|xy| \leq p$, $|y| \geq 1$ and $xy^iz \in L_1$ for all $i \geq 0$.

From $|xy| \leq p$ we know that xy can only consist of as. If we pump w smaller, i.e. we choose $i = 0$, we get the word $w_0 := xy^0z = \mathtt{a}^{p-|y|}\mathtt{b}\mathtt{c}^{p+1}$. As $|y| \geq 1$ we know that $p - |y| + 1 < p + 1$ and see that $w_0 \notin L_1$ (since the number of as added to the number of bs is not the number of cs, and thus the properties of the language are not satisfied). This is a contradiction to the pumping lemma and thus L_1 cannot be regular.

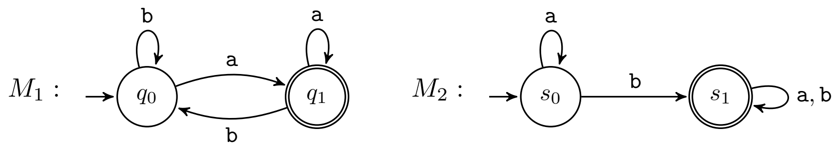
(b) $L_2 = \{\mathtt{a}^2\mathtt{b}^n\mathtt{a}^2\mathtt{c}^m \mid n, m \in \mathbb{N}_0\}$

Solution:

L_2 is regular because it is described by the regular expression $\mathtt{aab}^*\mathtt{aac}^*$.

Exercise 6.2 (Product Automaton; 2 marks)

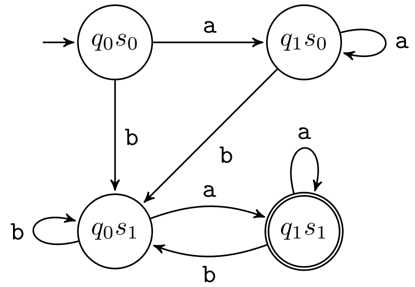
Given the following DFAs M_1 and M_2.



Specify the product automaton that accepts $\mathcal{L}(M_1) \cap \mathcal{L}(M_2)$.

Solution:

$\mathcal{L}(M_1) = \{w \mid w \text{ ends with } \mathtt{a}\}$, $\mathcal{L}(M_2) = \{w \mid w \text{ contains at least one } \mathtt{b}\}$
$\mathcal{L}(M_1) \cap \mathcal{L}(M_2) = \{w \mid w \text{ ends with } \mathtt{a} \text{ and contains at least one } \mathtt{b}\}$
The product automaton looks as follows:



Exercise 6.3 (Chomsky Normal Form, 3 marks)

Specify a grammar G' in Chomsky normal form that generates the same language as the context-free grammar $G = \langle \Sigma, V, P, \mathrm{S} \rangle$ with $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, $V = \{\mathrm{S}, \mathrm{W}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ and the following rules in P:

$\mathrm{S} \to \varepsilon$	$\mathrm{S} \to \mathrm{XW}$	$\mathrm{S} \to \mathrm{Z}$	$\mathrm{W} \to \mathrm{X}$	$\mathrm{X} \to \mathtt{a}\mathrm{Z}\mathtt{b}$
$\mathrm{Y} \to \mathrm{W}$	$\mathrm{Y} \to \mathtt{b}\mathrm{Y}$	$\mathrm{Z} \to \mathtt{bb}$	$\mathrm{Z} \to \mathrm{Z}\mathtt{a}$	$\mathrm{X} \to \mathrm{Y}$

Give sufficient intermediate steps.

Solution:

Step 1: eliminate rules of type $A \to B$ (with $A, B \in V$)
First, we eliminate the cycle consisting of W, X and Y.

$\mathrm{S} \to \varepsilon$	$\mathrm{S} \to \mathbf{RR}$	$\mathrm{S} \to \mathrm{Z}$		$\mathbf{R} \to \mathtt{a}\mathrm{Z}\mathtt{b}$
	$\mathbf{R} \to \mathtt{b}\mathbf{R}$	$\mathrm{Z} \to \mathtt{bb}$	$\mathrm{Z} \to \mathrm{Z}\mathtt{a}$	

Next, we rename variables in rules $A \to B$ in a way that for all rules $X_i \to X_j$, $i < j$. We only have one such rule: $\mathrm{S} \to \mathrm{Z}$.

$\mathbf{X}_1 \to \varepsilon$	$\mathbf{X}_1 \to \mathrm{RR}$	$\mathbf{X}_1 \to \mathbf{X}_2$		$\mathrm{R} \to \mathtt{a}\mathbf{X}_2\mathtt{b}$
	$\mathrm{R} \to \mathtt{b}\mathrm{R}$	$\mathbf{X}_2 \to \mathtt{bb}$	$\mathbf{X}_2 \to \mathbf{X}_2\mathtt{a}$	

Now we eliminate all rules $X_k \to X_{k'}$. In this case the only rule like this is $\mathrm{X}_1 \to \mathrm{X}_2$.

$\mathrm{X}_1 \to \varepsilon$	$\mathrm{X}_1 \to \mathrm{RR}$	$\mathbf{X}_1 \to \mathtt{bb}$	$\mathbf{X}_1 \to \mathbf{X}_2\mathtt{a}$	$\mathrm{R} \to \mathtt{a}\mathrm{X}_2\mathtt{b}$
	$\mathrm{R} \to \mathtt{b}\mathrm{R}$	$\mathrm{X}_2 \to \mathtt{bb}$	$\mathrm{X}_2 \to \mathrm{X}_2\mathtt{a}$	

Step 2: eliminate rules with a terminal symbol which are not of type $A \to a$.

$\mathrm{X}_1 \to \varepsilon$	$\mathrm{X}_1 \to \mathrm{RR}$	$\mathrm{X}_1 \to \mathbf{BB}$	$\mathrm{X}_1 \to \mathrm{X}_2\mathbf{A}$	$\mathrm{R} \to \mathbf{A}\mathrm{X}_2\mathbf{B}$
	$\mathrm{R} \to \mathbf{B}\mathrm{R}$	$\mathrm{X}_2 \to \mathbf{BB}$	$\mathrm{X}_2 \to \mathrm{X}_2\mathbf{A}$	
$\mathbf{A} \to \mathtt{a}$	$\mathbf{B} \to \mathtt{b}$			

Step 3: eliminate rules of type $A \to B_1B_2 \ldots B_k$ with $k > 2$. We only have one such rule: $\mathrm{R} \to \mathrm{AX}_2\mathrm{B}$.
The final result is the grammar $G' = \langle \Sigma, V, P, \mathrm{X}_1 \rangle$ with terminal alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, variables $V = \{\mathrm{R}, \mathrm{X}_1, \mathrm{X}_2, \mathrm{A}, \mathrm{B}, \mathrm{K}_1\}$ and the following rules in P:

$\mathrm{X}_1 \to \varepsilon$	$\mathrm{X}_1 \to \mathrm{RR}$	$\mathrm{X}_1 \to \mathrm{BB}$	$\mathrm{X}_1 \to \mathrm{X}_2\mathrm{A}$	$\mathbf{R} \to \mathbf{AK}_1$
	$\mathrm{R} \to \mathrm{BR}$	$\mathrm{X}_2 \to \mathrm{BB}$	$\mathrm{X}_2 \to \mathrm{X}_2\mathrm{A}$	$\mathbf{K}_1 \to \mathbf{X}_2\mathbf{B}$
$\mathrm{A} \to \mathtt{a}$	$\mathrm{B} \to \mathtt{b}$			

Exercise 6.4 (Length of Derivations in Chomsky Normal Form; 2 marks)

Let G be a grammar in Chomsky normal form and $w \in \mathcal{L}(G)$ a non-empty word ($w \neq \varepsilon$), which is generated by G. Show that every derivation of w from the start variable of G consists of exactly $2|w| - 1$ steps.

Solution:

A grammar is in Chomsky Normal Form if each rule has one of the following three forms: (a) $A \to BC$, (b) $A \to a$ or (c) $S \to \varepsilon$, where A, B, C are variables, S is the start variable (and B and C are not the start variable) and a is a terminal symbol.
Let w be a word of the language with length $n > 0$ which is generated by the grammar. Since rules of type (c) are only utilized to derive the empty word, they are not relevant here. In order to introduce the n terminal symbols of w, any derivation of the word needs to apply rules of

type (b) exactly n times. Each of these rule applications remove one nonterminal symbol which means that for each of these rule applications there must have been one nonterminal symbol before. These nonterminal symbols must have been introduced by applying rules of type (a). Each such application increases the number of nonterminal symbols by 1. Since the start variable is already such a nonterminal symbol, we thus need to apply *at least* $n-1$ times a rule of type (a). Nonterminal symbols are only eliminated by rules of type (b), which means that we can apply rules of type (a) *at most* $n-1$ times. Thus each derivation of w consists of exactly $n+n-1=2n-1$ derivation steps.