Exercise 1

Just to recall:

Name and define all grammars and automata in the Chomsky hierarchy:

FSA	RG
PDA	CFG
LBA	CSG
TM	G0

Explain the reasons for vertical (automata – grammars) and horizontal (FSA – TM) split.

Define formally the finite state automaton.

Practice:

Design a finite state automaton that counts a tennis score.

Design finite state automata working with the alphabet {a,b}, that accept the following words:

- the number of letters „a“ is divisible by 3,
- the number of letters „a“ is divisible by 2,
- the number of letters „a“ is divisible by 3 or by 2,
- the number of letters „a“ is divisible by 3 and by 2,
- the number of letters „a“ is divisible by 3 but not by 2,
- finishes by „baba“,
- contains „baba“,
- starts with „baba“,
- finishes by „b“ and its length is (3k+1),
- finishes by „b“ or its length is (3k+1),
- finishes by two or more letters „b“ that are immediately preceded by at least one letter „a“,
- its length is at least 2 and the first and the last letter in the word are identical,
- the first and the last letter in the word are identical,
- its length is at least 4 and the first two letters are identical to the last two letters,
- the first two letters are identical to the last two letters.

What are the following automata doing (describe the language)?

a)

	0	1
↔q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_0	q_2

b)

	0	1
→q_0	q_1	q_0
←q_1	q_2	q_1
←q_2	q_0	q_2

c)

	0	1
→q_0	q_0	q_1
q_1	q_0	q_2
←q_2	q_0	q_2

d)

	0	1
→q_0	q_0	q_1
←q_1	q_2	q_1
←q_2	q_0	q_1

Exercise 2

Just to recall:

Formulate the Myhill-Nerode Theorem and the Pumping Lemma.
Define the state equivalence.

Practice:

Decide and prove if the following languages are regular

- 0^k1^l, $k \leq l$
- 0^k1^k
- $1^k0^k1^l$
- $1^k0^l1^k$
- $1^k1^k0^l$
- $1^k1^l0^k$
- 0^n1^m for $n \leq m \leq 2n$
- $0^n1^m0^n$ for $m \leq n$
- ww, $w \in \{0,1\}^*$
- w, $w \in \{0,1\}^*$ and w contains identical counts of letters 0 and 1
- ww^R, $w \in \{a\}^*$
- a^{2n}
- a^{2^n}
- a^{n^2}
- a^p, p is a prime number
- $a^ib^ic^i$
- $a^ib^ic^j$ pro $i<j$

Find all equivalent states in the following automata:

a)

	a	b
↔0	0	5
1	1	3
2	2	7
3	3	2
←4	6	1
5	5	1
←6	4	2
7	7	0

b)

	b	a
A	F	A
B	A	B
C	D	C
D	B	D
E	C	E
↔F	E	F

c)

	a	b
→1	2	3
2	2	4
←3	3	5
4	2	7
←5	6	3
←6	6	6
7	7	4
8	2	3
9	9	4

d)

	b	a
A	G	H
B	A	B
C	D	E
D	B	D
E	D	C
F	E	F
↔G	F	G
H	G	A

e)

	a	b
↔0	1	2
1	3	0
2	4	5
3	0	2
4	2	5
5	0	3

f)

	a	b
→0	1	2
1	0	3
2	4	1
3	0	1
←4	2	2
5	4	3

g)

	a	b
↔0	0	1
←1	2	3
2	3	4
3	1	0
4	3	2

Exercise 3

Just to recall and to think about:

1) *The Pumping lemma as formulated in the lecture assumes that the part to iterate is located at the beginning of the world ($|uv| \leq n$). Think if it is possible to formulate the Pumping lemma, where:*
   - *the part to iterate is located anywhere in the world*
   - *the part to iterate is located at the end of the world*
   - *the part to iterate is located close to some given place in the world.*
2) *Does the Pumping lemma hold also for complements of regular languages? In particular "we can divide the word outside the regular language in such a way that we can iterate the "middle" part and the resulting word is still outside the language"*
3) *Define formally the equivalence of automata. Is there any relation to state equivalence?*
4) *What is a reduced finite state automaton? Define it.*

Practice:

Decide and prove if the following automata are equivalent:

a)

	a	b
↔0	0	5
1	1	3
2	2	7
3	3	2
←4	6	1
5	5	1
←6	4	2
7	7	0

b)

	b	a
A	F	A
B	A	B
C	D	C
D	B	D
E	C	E
↔F	E	F

c)

	b	a
A	G	H
B	A	B
C	D	E
D	B	D
E	D	C
F	E	F
↔G	F	G
H	G	A

What is the shortest word in automaton (a) that differentiates states 1 and 5? Are the more such words?

In the following automata find all the shortest words that differentiate a given pairs of states:

e) states 3 and 5

	a	b
↔0	1	2
1	3	0
2	4	5
3	0	2
4	2	5
5	0	3

f) states 0 and 1

	a	b
→0	1	2
1	0	3
2	4	1
3	0	1
←4	2	2
5	4	3

g) states 2 and 4

	a	b
↔0	0	1
←1	2	3
2	3	4
3	1	0
4	3	2

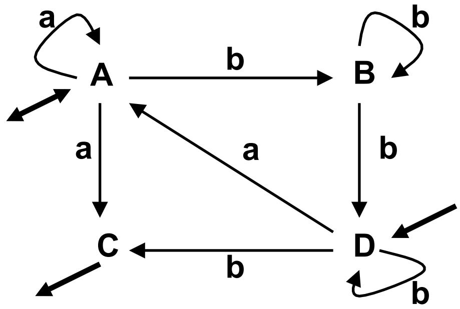
Exercise 4

Just to recall and to think about:

1) *Which set and string operations have the closure property for regular languages?*
2) *Do we obtain a complement of language if we swap the role of accepting and non-accepting states in a nondeterministic finite-state automaton?*

Practice:

Convert the following automaton to a reduced finite state automaton:



Propose algorithms that decide if the following propositions hold:

- L(A) = ∅
- L(A) = L(B)
- L(A) = X*
- L(A)⊆L(B)
- L(A) je nekonečný

Let L= {ab, c}. Describe the following languages: $L^+, L^*, (L^*)^*$

Propose a FSA accepting all words in the alphabet {a,b} that do not contain the word "baba".

Let X={0,1} be an alphabet. Design a FSA accepting words where the number if letters 0 is divisible by:

- *2*
- *3*
- *2 or 3*
- *2 and 3*
- *2 but not by 3.*

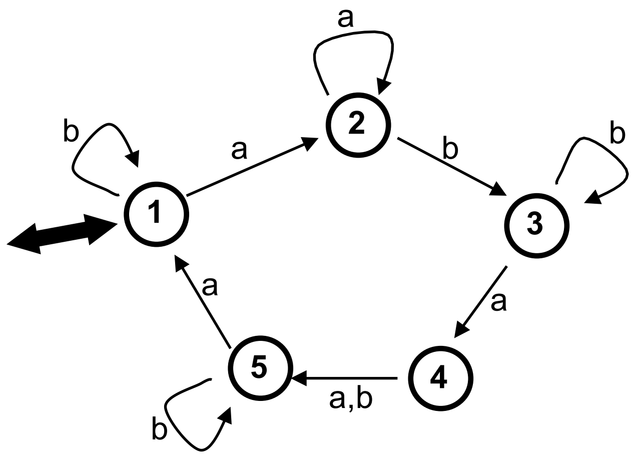
Let $L_1=\{u|u\in\{0,1\}^* \ \& \ |u|_0=2k\}$ *and* $L_2=\{u|u\in\{0,1\}^* \ \& \ |u|_0=3k\}$ *be languages. Propose the smallest FSA accepting the language* $L_2 \setminus L_1$.

What is the minimal number of states in a FSA that accepts the following language?

$$\mathbf{L_n = \{ w \mid w\in\{0,1\}^*, w=u1v, |v|=n-1 \}}.$$

What is the minimal number of states in a FSA that accepts the language $(\mathbf{L_n})^R$?

Let the following automat accepts some language L:



Propose (nondeterministic) FSAs accepting the following languages:

- $L_1 = \{uv \mid uav \in L \vee ubv \in L\}$
- $L_2 = \{uv \mid uav \in L\}$
- $L_3 = \{uav \mid uv \in L\}$
- $L \cup L_1$, $L \cup L_2$, $L \cup L_2$

Exercise 5

Just to recall and to think about:

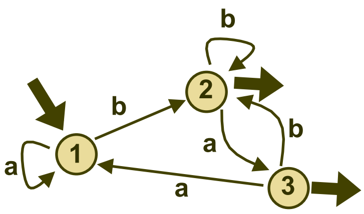
1) *What are the core algebraic operations to obtain all regular languages?*
2) *How is the language {λ} composed from the elementary languages?*
3) *Is there any relation between the proof of the Kleene's theorem and all-pairs-shortest-path algorithms?*

Practice:

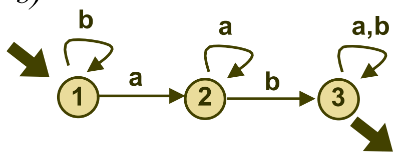
1) *Write a regular expression whose value is a language in the alphabet {a,b} consisting exactly from the words that start with "ba" and finish with "ab". Convert the regular expression to a corresponding FSA.*
2) *Write a regular expression whose value is a language containing words a, a*a, a*a*a,….*
3) *Convert the following regular expressions to FSAs accepting the languages that are values of these expressions:*
   - $ab+ba$
   - a^2+b^2+ab
   - $a+b^*$
   - $(ab+c)^*$
   - $((ab+c)^+a(bc)^*+b)^*$
   - $((ab+c)^*a(bc)^*+b)^*$
   - $(01^*+101)^*0^*1$
   - $(01)^*1111(01)^*+(0+1)^*000$

4) Convert the following automata to regular expressions such that the value of the expression is the language accepted by a given FSA:

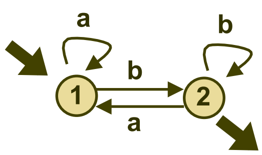
a)



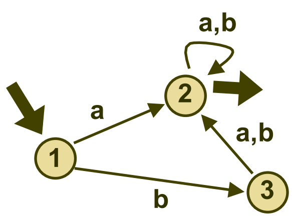
b)



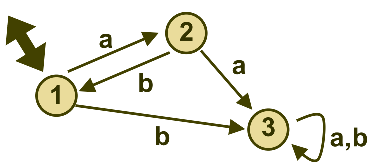
c)



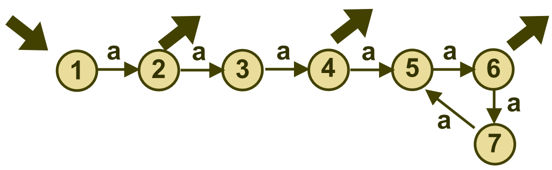
d)



e)



f)



Exercise 6

Just to recall and to think about:

1) *Regular expression is a word. Is the language consisting of all regular expressions acceptable by some finite state automaton?*
2) *How can a finite state automaton inform about its computation? What is the difference between Moore and Mealy machines?*
3) *What is the advantage of non-determinism and the possibility to move the reading head in both directions?*

Practice:

1) *Let L be a regular language. Is the language* $\{u \mid \#u\# \in L\}$ *also regular? Prove it! Note: The symbol "#" is a part of the alphabet.*
2) *Let L be a language accepted by a finite state automaton A. Construct a two-way (non-deterministic) finite state automaton accepting the following language:*
   - $\{\#u\# \mid uu^R \in L\}$
   - $\{\#u\# \mid uu \in L\}$
   - $\{\#u\# \mid uv \in L \ \& \ |u|=|v|\}$
   - $\{\#u\# \mid u=wv \ \& \ w^R wv \in L\}$

   Convert the obtained automata to a finite state automata.

3) *Design a Mealy machine working with the alphabet {0,1} that inverts the input word (0→1,1→0). Convert the machine to an equivalent Moore machine.*

4) *Design a Mealy Machine working with the alphabet {0,1} that implements the following output function:*

   output 1, if the input symbol is a part of sequence of 1s, which is directly preceded by symbols 00,
   output 0 in all other cases.

   Convert the Mealy machine to an equivalent Moore machine.

5) *Design a formal machine that sums two binary numbers. Think about defining the input for such a machine.*

Exercise 7

Just to recall and to think about:

1) *Explain the notion of a formal grammar.*
2) *What is the difference between generative and analytical formal grammar?*
3) *How are the types of grammars in the Chomsky hierarchy distinguished?*
4) *What is the reason for the names of context-sensitive and context-free grammars?*
5) *If we combine left-linear and right-linear rewriting rules do we obtain a grammar accepting a regular language?*

Practice:

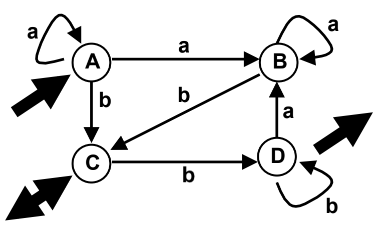
1) Design a formal grammar that generates the following language:

- $\{a^i b^i \mid i \geq 0\}$
- *well-formed expressions with left and right brackets*
- *well-formed arithmetic expressions with a single constant "c", operations + and *, and left and right brackets*
- $\{w \mid w \in \{a,b\}^* \ |w|_b = 3k\}$
- *binary numbers that are multiples of 3*
- $\{a^i b^i c^i \mid i \geq 0\}$
- $\{a^i b^j c^{i+j} \mid i,j \geq 0\}$
- $\{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$
- $\{a^i b^j c^k \mid i=j \vee j=k\}$
- $\{ww^R \mid w \in \{a,b\}^*\}$
- $\{ww \mid w \in \{a,b\}^*\}$
- $\{0^n 1^m 0^n \mid 0 \leq m \leq n\}$
- $\{0^n 1^m \mid 0 \leq n \leq m \leq 2n\}$
- a^{2^n}

2) Is the following grammar context-sensitive (the capital letters denote the non-terminal symbols)? Can the grammar be converted to a context-sensitive form?

$S \rightarrow aSbA \mid \lambda$
$A \rightarrow aBbA \mid bCB \mid CD$
$B \rightarrow bbBa \mid aS$
$C \rightarrow aAaA \mid \lambda$
$D \rightarrow SC \mid aABb$

3) Convert the following FSA to a grammar that generates the language accepted by the FSA. What type of grammar do we obtain?



4) *Convert the following grammar to a finite state automaton accepting the same language. Can any grammar be converted to an "equivalent" finite state automaton?*

$S \rightarrow abS \mid bbaA \mid \lambda$
$A \rightarrow abA \mid bB \mid C$
$B \rightarrow acS \mid bC \mid \lambda$
$C \rightarrow abb \mid bA \mid A$

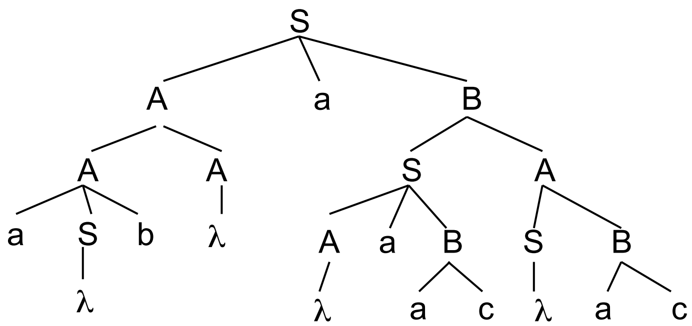
Exercise 8

To recall and to think about:

1) *Whatis the main difference between the reduced context-free grammar and reduced FSA?*
2) *Assume a word generated by a given context-free grammar. Is the derivation for this word unique?*
3) *Does the order of application of production rules influence the final generated word for a CFG? Explain.*
4) *Is there another way to describe how the word is derived (different from the derivation)?*

Practice:

1) *The figure shows a syntax (derivation) tree for some context-free grammar G.*



- *What is the generated word given by this tree?*
- *Write a left derivation for this word.*
- *Write all the rewriting rules used in this syntax tree.*
- *Can we say something about ambiguity of the grammar G?*

2) *Reduce the following CFGs:*

$S \rightarrow aSb \mid aAbb \mid \lambda$
$A \rightarrow aAB \mid bB$
$B \rightarrow aAb \mid BB$
$C \rightarrow CC \mid cS$

$S \rightarrow aA \mid bB \mid aSa \mid bSb \mid \lambda$
$A \rightarrow bCD \mid Dba$
$B \rightarrow Bb \mid AC$
$C \rightarrow aA \mid c$
$D \rightarrow DE$
$E \rightarrow \lambda$

3) *Decide (and prove) if the following grammar G satisfies L(G)=∅.*

$S \rightarrow aS \mid AB \mid CD$
$A \rightarrow aDb \mid AD \mid BC$
$B \rightarrow bSb \mid BB$
$C \rightarrow BA \mid ASb$
$D \rightarrow ABCD \mid \lambda$

Exercise 9

To recall and to think about:

1) *What is main difference between a pushdown automaton and a finite state automaton?*
2) *Describe the mechanism how the pushdown automata accept words.*
3) *Is there any difference if the pushdown automaton accepts the works using an acceptance state or using an empty stack?*
4) *Do deterministic pushdown automata accept the same class of languages as nondeterministic pushdown automata?*

Practice:

1) *Design pushdown automata accepting the following languages. For each automaton explore both types of accepting the words. If possible, try to design the automaton as a deterministic automaton. For the automata that use an empty stack to accept the words, try to design an automaton with a single state.*

   - $\{0^n 1^m \mid 0 \leq n \leq m\}$
   - $\{wcw^R \mid w \in \{a,b\}^*\}$
   - $\{ww^R \mid w \in \{a,b\}^*\}$
   - $\{w \mid w \in \{a,b,c\}^* \; w{\downarrow}_{a,b} = uu^R\}$, *where* $w{\downarrow}_{a,b}$ *is a word w, where all symbols different from "a" and "b" were removed*
   - $\{w \mid w \in \{a,b\}^* \; |w|_b = |w|_a\}$
   - $\{ucv \mid u,v \in \{a,b\}^* \; |u| \neq |v|\}$
   - $L_i = \{ucv \mid u,v \in \{a,b\}^*$ *u and v are different in the i-th symbol from the left* $\}$
   - $\{ucv \mid u,v \in \{a,b\}^* \; u \neq v\}$
   - *well-formed bracketed expression (such as "(())()")*
   - $\{a^i b^i \mid i \geq 0\}$
   - $\{a^i b^j c^{i+j} \mid i,j \geq 0\}$
   - $\{a^i b^j c^k \mid i=j \vee j=k\}$

2) *Convert the following grammar G to a pushdown automata accepting language L(G) using the acceptance states. Show, how the automaton accepts the word (a+a)*a.*

$E \rightarrow E+T \mid T$
$T \rightarrow T*F \mid F$
$F \rightarrow (E) \mid a$

3) *Take the context-free grammars from Exercise 7 and convert them to pushdown automata.*

Exercise 10

To recall and to think about:

1) *What is typical for derivations done with the grammar in the Greibach normal form is used?*
2) *What is typical for syntax trees for grammars in the Chomsky normal form is used?*
3) *What is the relation between the depth of the syntax tree for grammars in the Chomsky normal form and the length of the generated word?*
4) *Formulate and prove the pumping lemma for context-free languages.*
5) *Formulate and prove the pumping lemma for linear languages (use the ideas from CFL).*

Practice:

1) *Convert the following grammars to the Chomsky normal form:*

$S \to A \mid 0SA \mid \lambda$	$S \to 0A10B11$
$A \to 1A \mid 1 \mid B1$	$A \to 0A1 \mid \lambda$
$B \to 0B \mid 0 \mid \lambda$	$B \to 0B11 \mid \lambda$

2) *Convert the following grammar to the Greibach normal form:*

$S \to (E)$
$E \to F+F \mid F*F$
$F \to a \mid S$

3) *Decide and prove if the these languages are context-free:*

- $\{a^i b^i c^i \mid i \geq 0\}$
- $\{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$
- $\{a^i b^j c^{i+j} \mid i,j \geq 0\}$
- $\{ww^R \mid w \in \{a,b\}^*\}$
- $\{ww \mid w \in \{a,b\}^*\}$
- $\{0^n 1^m 0^n \mid 0 \leq m \leq n\}$
- $\{0^n 1^m \mid 0 \leq n \leq m \leq 2n\}$
- $\{0^n 1^n 0^n 1^n \mid 0 \leq n\}$
- $\{0^i 1^j 0^i 1^j \mid 0 \leq i \leq j\}$
- a^{2^n}
- a^{n^2}