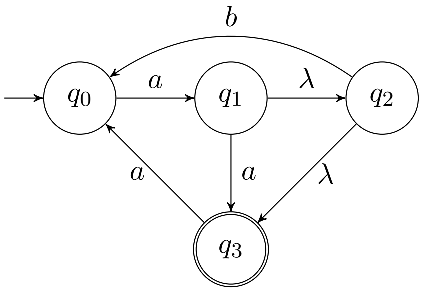
Name: Student number:

Exam 2, Module 7, Study Unit 202001361 Languages & Machines
Friday, April 1, 2022, 13:45 - 15:45

This second exam of Module 7 consists of the **Languages & Machines part** only, and is a **2h exam**. You are allowed to use a handwritten cheat sheet (A4, double sided) during the exam.

Exam sheet contains sample solution!

1. (8 points) Consider the following NFA-λ, M (only q_3 is accepting):



   (a) Does the automaton accept the word $abaa$?
   If yes, provide a sequence of states with which the automaton can accept the word.
   If no, provide a sequence of states with which the automaton rejects the word. If the automaton rejects because there is no transition to a successor state with the according input, add "deadend" to the end of the sequence.

   [ ] No.
   [✔] Yes.

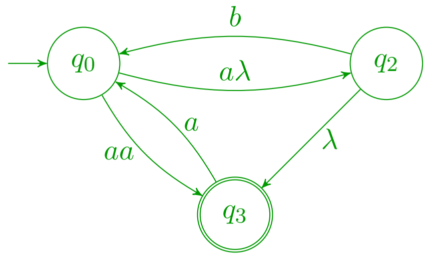
   The sequence of states is: $q_0, q_1, q_2, q_0, q_1, q_3$

   Does the automaton accept the word aab?
   If yes, provide a sequence of states with which the automaton can accept the word.
   If no, provide a sequence of states with which the automaton rejects the word. If the automaton rejects because there is no transition to a successor state with the according input, add "deadend" to the end of the sequence.

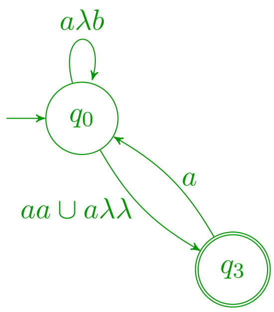
   [✔] No.
   [ ] Yes.

   The sequence of states is: q_0, q_1, q_3, deadend

(b) Transform the automaton M step by step to a regular expression.
States must be eliminated in the order q_1, q_2.
After elimination of q_1:



After elimination of q_2



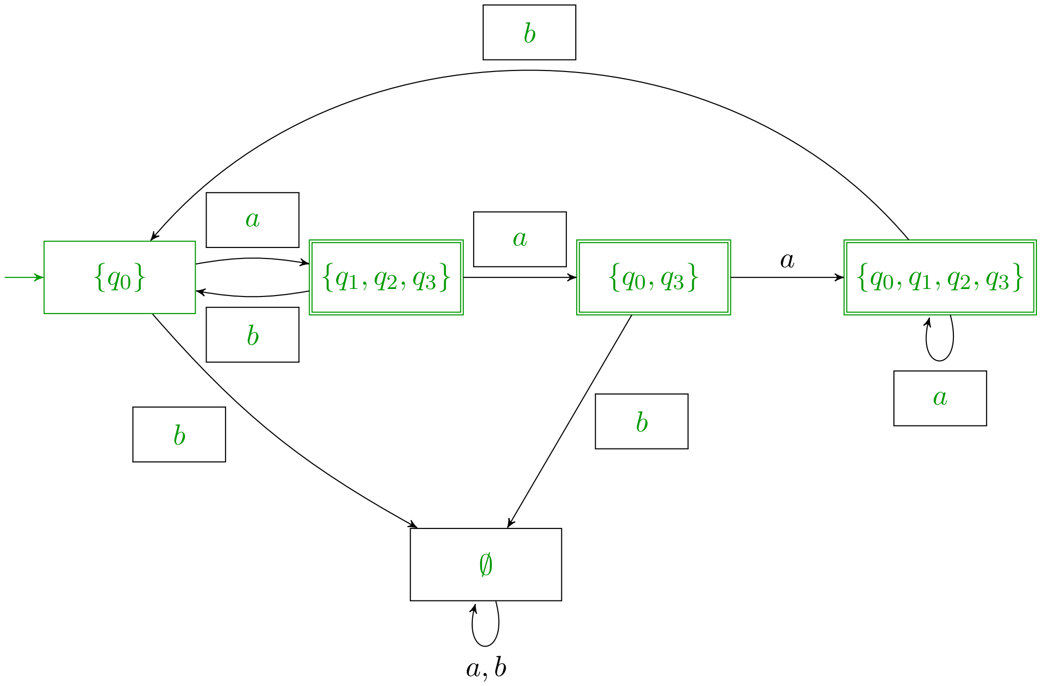
Thus, the regular expression obtained is:

$$(a\lambda b)^*(aa \cup a\lambda\lambda)\,(a(a\lambda b)^*(aa \cup a\lambda\lambda))^*$$

(c) Fill in the following table with the λ-closure and input-transition function of M.

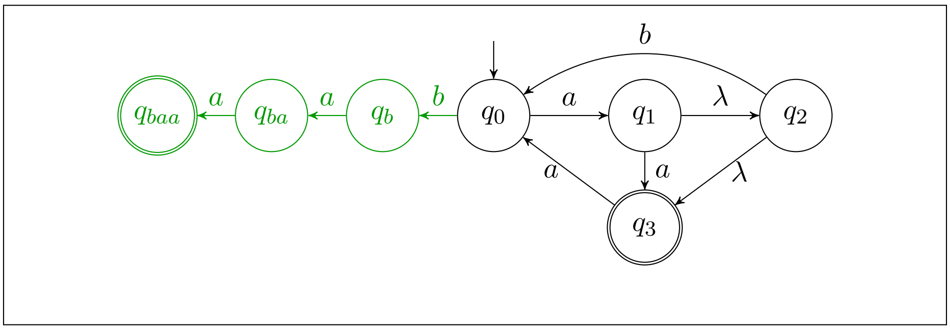
	λ-closure	a	b
q_0	$\{q_0\}$	$\{q_1, q_2, q_3\}$	$\emptyset$
q_1	$\{q_1, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0\}$
q_2	$\{q_2, q_3\}$	$\{q_0\}$	$\{q_0\}$
q_3	$\{q_3\}$	$\{q_0\}$	$\emptyset$

(d) The following (partially completed) DFA accepts the same language as M.



i. Mark the initial state
ii. Mark the accepting state/the accepting states
iii. Label the states with the set of λ-NFA states they represent.
iv. Complete the white boxes at the transitions.

(e) Modify the λ-NFA in the figure below such that the word "baa" is also in its language (but no further new words).


q_{baa}
a
q_{ba}
a
q_b
b
q_0
a
q_1
λ
q_2
b
a
a
λ
q_3

2. (8 points) For each of the following languages, indicate whether it is

- A: regular
- B: context-free but not regular
- C: recursive but not context-free
- D: recursively enumerable but not recursive
- E: not recursively enumerable

For each correct answer, you will get 1 point. For each incorrect answer, you will loose 1 point. For no answer, you get 0 points. The total points of this exercise will not become negative.

No further explanation is required.

[D] Language $L_1 := \{w \mid w$ encodes a multi-tape Turing machine which writes sequence $abba$ for input $w^R\}$

[A] Language $L_2 := \mathcal{L}(a^*b^*) \cup \mathcal{L}(a^*c^*)$

[E] Language $L_3 := \{w \mid w$ encodes a Turing machine which never writes the sequence def for any input$\}$

[A] Language $L_4 := \{b^j\, c^{3i}\, b^j\, (cb)^i \mid 0 \leq i \leq 62$ and $0 \leq j \leq 3i\}$

[C] Language $L_5 := \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$

[B] Language $L_6 := \{a^{i+2}\, c\, b^{i+4} \mid 0 \leq i\}$

[C] Language $L_7 := \{a^i b^j a^{4i} b^{7j} \mid i \geq 0, j \geq 0\}$

[A] Language $L_8 := \{a^i b^j c^k \mid i, j, k \geq 0$ and i even and j mod $5 = 0\}$

3. (5 points) Consider the following languages

- $L_1 = \mathcal{L}(a^* b^*)$
- $L_2 = \mathcal{L}(c^* d^*)$
- $L_3 = \{a^i b^j \mid i = n^2$ and $j = n^3$ for some $n \geq 0\}$
- $L_4 = \{w \in \{a, b\}^* \mid |w| \mod 3 = 0$ and $|w| \mod 2 \neq 0\}$
  Note that this is a regular language

Which of the following languages are regular? Mark languages for which this is the case with Y and mark those for which this does not hold with N. Leave out the answer if you do not know. Correct answers yield 1 point, for incorrect answers you loose 1 point. For no answer, you get 0 points. In total, you cannot get less than 0 points for this question.

[Y] $L_1 \cup L_3$
Note that $L_3 \subset L_1$

[Y] $L_2 \cap L_3$
Note that $L_2 \cap L_3 = \{\lambda\}$

[N] $L_3 \cup L_4$

[Y] $\overline{L_4}$

[N] $L_2 \cup L_3$

4. (10 points) Consider the following context-free grammar G:

$$G = \begin{cases} S & \rightarrow \; C\,B\,A \mid \lambda \mid A\,B \\ A & \rightarrow \; A\,A\,a \mid a \\ B & \rightarrow \; a\,b\,C \\ C & \rightarrow \; C\,a \mid c \end{cases}$$

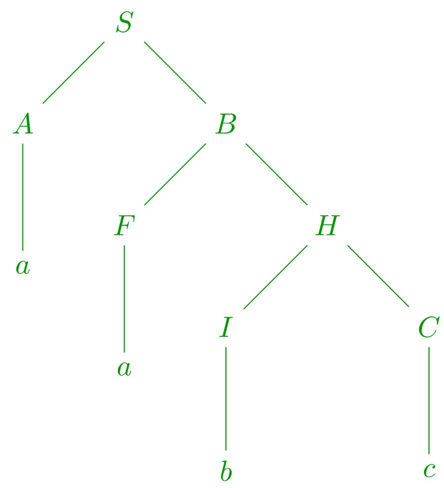
(a) Complete following grammar G' in Chomsky Normal Form (CNF) such that it is equivalent to G. It is only required that G' is in CNF, we do not require that you provide exactly what the transformation algorithm would result in.

$$G' = \begin{cases} S & \rightarrow \; C\,D \mid \boxed{\lambda} \mid A\,B \\ D & \rightarrow \; \boxed{B\,A} \\ \boxed{A} & \rightarrow \; \boxed{A\,E} \mid a \\ \boxed{E} & \rightarrow \; A\,F \\ F & \rightarrow \; a \\ B & \rightarrow \; \boxed{F\,H} \\ H & \rightarrow \; I\,C \\ I & \rightarrow \; \boxed{\text{b}} \\ C & \rightarrow \; \boxed{C\,F} \mid c \end{cases}$$

(b) Let $w = aabc$. Use the table below to apply the CYK-algorithm (after Cocke-Younger-Kasami) to decide whether $w \in \mathcal{L}(G')$. In case of empty sets, explicitly write "$\emptyset$".

	1	2	3	4
1	$\{A, F\}$	$\{E\}$	$\emptyset$	$\{S\}$
2		$\{A, F\}$	$\emptyset$	$\{B\}$
3			$\{I\}$	$\{H\}$
4				$\{C\}$

Use the results of the CYK-algorithm to provide a derivation tree of G' for w:

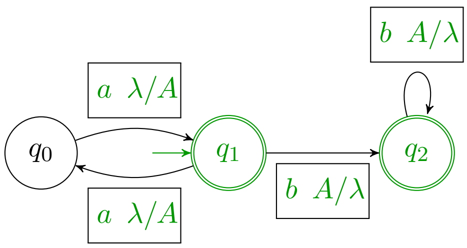


5. (8 points) Consider the context-free language $L = \{a^{2i}\, b^{2i} \mid i \geq 0\}$.

   (a) Give a context-free grammar G in Greibach Normal form for L.
   Note that you are not required to apply the algorithm to transform a general contex-free grammar to Greibach normal form.

$$G = \begin{cases} S & \rightarrow \quad \lambda \mid a\,A\,R\,B\,B \mid a\,A\,B\,B \\ R & \rightarrow \quad a\,A\,R\,B\,B \mid a\,A\,B\,B \\ A & \rightarrow \quad a \\ B & \rightarrow \quad b \end{cases}$$

   (b) Consider the following incomplete non-extended, deterministic PDA.



Complete the automaton such that it accepts L by

i. marking the initial state,
ii. ~~marking the final state or states, and~~
iii. by filling the four white boxes.

Note that this is not the automaton which would be obtain from the method from the lecture slides. We require this to be a deterministic, non-extended PDA. Only the symbol A may be pushed on the stack.

6. (5 points) Prove or disprove: the language $L := \{a^i b^i c^i \mid i = n^2 \text{ for some } n \geq 0\}$ is context-free.

L is not context-free. We can prove this in almost the same way as for the language

$$X := \{a^i b^i c^i \mid i \geq 0\}$$

seen in the lecture. The reason that this works is basically that n^2 is unbounded.

- Let k be arbitrarily given
- Choose $z = a^{k'} b^{k'} c^{k'}$ where k' is the smallest number larger or equal to k for which we have an $n \geq 0$ with $k' = n^2$
- Let $uvwxy = z$ such that $|vx| > 0$ and $|vwx| \leq k$
- We distinguish the following possible cases:
  (a) v and x both contain one (repeated) symbol from $\{a, b, c\}$
    - Then $v \in \{a^m, b^m, c^m\}$ and $x \in \{a^n, b^n, c^n\}$ with $m + n = |vx| > 0$
    - Let $\{\alpha, \beta, \gamma\} \subseteq \{a, b, c\}$ such that $v = \alpha^m$ and $x = \beta^n$ and $\gamma \notin \{\alpha, \beta\}$.
    - Then uv^2wx^2y contains the same number of γ's as z does, but more α's and/or β's
    - Choose $i = 2$; then $uv^iwx^iy \notin L$

  (b) v or x contains (at least) 2 different symbols from $\{a, b, c\}$
    - Then v^2 or x^2 contains $a\,b$ or c before an a or a c before $a\,b$;
    - Choose $i = 2$; then $uv^2wx^2y \notin L$

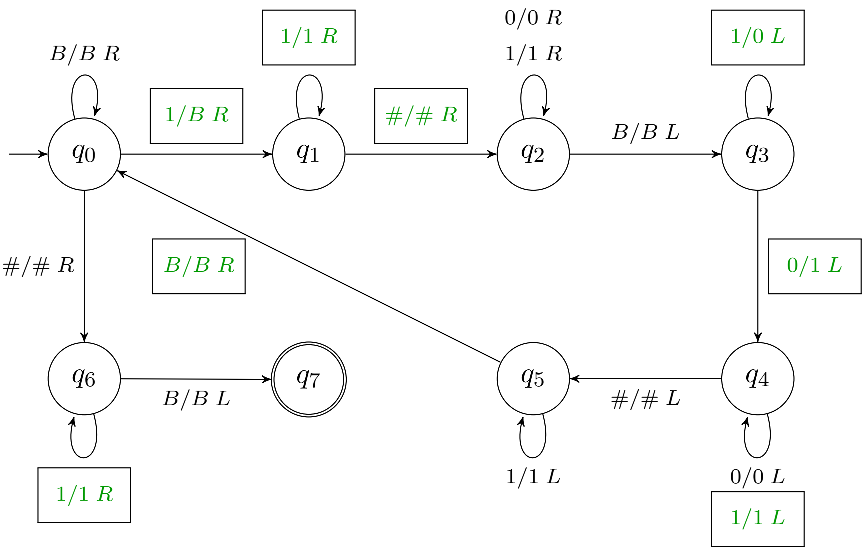
7. (6 points) Consider the language L over the alphabet $\Sigma = \{0, 1, \#\}$ with

$$L := \{1^i \# w \mid w \in \{0,1\}^* \text{ and } 2^{|w|} - \mathrm{val}(w) - 1 = i\}$$

where $\mathrm{val}(w)$ is the value of w interpreted as a binary number. Thus, for instance $\mathrm{val}(1101) = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$. We have for instance

- $11\#101 \in L$
- $11\#111 \notin L$
- $11111\#101\# \notin L$
- $11\#1101 \in L$
- $1111111\#000 \in L$
- $\# \in L$

Consider the following incomplete Turing machine to decide this language.



(a) Fill in the white boxes in the automaton figure.

(b) Decide what the following states do. Choose from the list of provided options

- state q_1: ii
- state q_3: v
- state q_4: i
- state q_6: iii

Possible options:

i Search $\#$ marking the left end of w.

ii Move from left to right over sequence of 1's. Once $\#$ is seen, move to next phase.

iii Check whether we only have only 1's in the w part. If yes, we can accept.

iv Accept.

v ~~Decrement~~Increment number w by repeatedly replacing 1's by 0's until the first 0 seen can be replaced by 1.

vi Read over blank character at the beginning of the tape. Erase the first 1 seen. If we see $\#$ instead, prepare to check whether we can accept.

vii Search blank character left of the sequence of 1's.

viii Move over 0's and 1's. Once a blank character is seen, move to next phase.

Notes:

Notes:

Notes:

Notes:

Notes: