

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Automata and Formal Languages

Exam: IN2041 / Endterm

Date: Thursday 12th February, 2026

Examiner: Prof. Dr. Javier Esparza

Time: 11:00 – 13:00

Working instructions

- This exam consists of **12 pages** with a total of **7 problems**.
Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 75 credits, including 5 bonus credits.
- To pass the exam, 35 credits are *sufficient*.
- Detaching pages from the exam is prohibited.
- Allowed resources:
 - one **analog dictionary** English ↔ native language
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- Physically turn off all electronic devices, put them into your bag and close the bag.

Left room from _____ to _____ / Early submission at _____

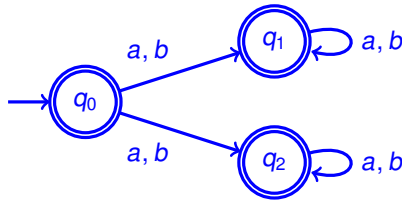
Problem 1 Quiz (15 credits)

For each of these statements, decide whether it is true or false. If it is true, give a proof; if it is false, give a counterexample. Otherwise no points will be awarded! We use $\Sigma := \{a, b\}$ as alphabet in this exercise.

0	
1	
2	

a)* Let A be an NFA such that all states of A are reachable. True or false: The output of $NFAtoDFA(A)$ has at least as many states as A .

False. Counterexample:



A has three states, the output of $NFAtoDFA(A)$ has only two: $\{q_0\}$ and $\{q_1, q_2\}$. (One can also remove q_0 and make both q_1 and q_2 initial.)

0	
1	
2	
3	

b)* Let A be an NFA with n states. True or false: There exists an NBA for $(L(A))^\omega$ with $n + 2$ states.

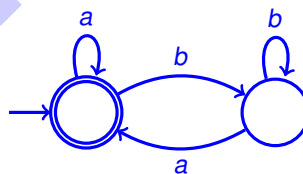
True. One can (like in the preprocessing step for $NFA_\epsilon to RE$) convert A into an equivalent $NFA_\epsilon B$ with $n + 2$ states such that B has only one initial state q_0 with no incoming transitions and only one final state q_f with no outgoing transitions. B can now be converted into an equivalent NFA C without changing the number of states (note that q_f still has no outgoing transitions). By adding transitions (q_f, a, q) for every transition (q_0, a, q) , we get an NBA for $(L(C))^\omega = (L(A))^\omega$.

Note: One does not need to add the state q_0 , it is only necessary for the final state(s) of B to have no outgoing transitions.

0	
1	
2	
3	

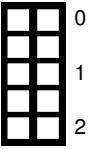
c)* Let A be a minimal NBA (that is, no NBA B with $L(B) = L(A)$ has less states than A). True or false: Every state of A recognizes a different ω -language.

False. Counterexample:



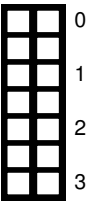
A is minimal, but both states recognize the language of all words with infinitely many a's.

d)* Let L be an ω -regular language. True or false: There exists an NRA with only one Rabin pair recognizing L .



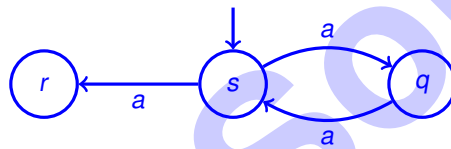
True. There exists an NBA recognizing L . It can be interpreted as an NRA with the Rabin pair $\langle F, \emptyset \rangle$ where F is the set of accepting states of the NBA.

e)* Let q, r be two reachable states of an NBA A such that r is reachable from q , but q is not reachable from r . Consider a run of *NestedDFS* on A . True or false: $d[r] < f[q]$, i.e. the algorithm discovers r before it backtracks from q . If the answer is "true", prove the statement using the parenthesis theorem given below; if the answer is "false", give a counterexample.



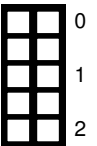
Parenthesis theorem. Let $I(q)$ denote the interval $[d[q], f[q]]$, and let $I(q) \prec I(r)$ denote that $f[q] < d[r]$ holds. In a DFS-tree, for any two states q and r , one of the following four conditions holds: (1) $I(q) \subseteq I(r)$ and q is a descendant of r ; (2) $I(r) \subseteq I(q)$ and r is a descendant of q ; (3) $I(q) \prec I(r)$ and neither is a descendant of the other; (4) $I(r) \prec I(q)$ and neither is a descendant of the other.

False. Counterexample:



Here, if the algorithm chooses q before r , it backtracks from q before discovering r .

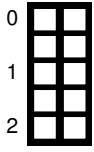
f)* Let $AP = \{p, q, r\}$ be a set of atomic propositions and let $\sigma \in (2^{AP})^\omega$ be a computation. True or false: If $\sigma \models p \mathbf{U} (q \mathbf{U} r)$, then $\sigma \models (p \mathbf{U} q) \mathbf{U} r$.



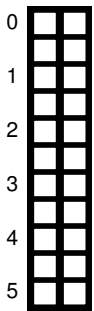
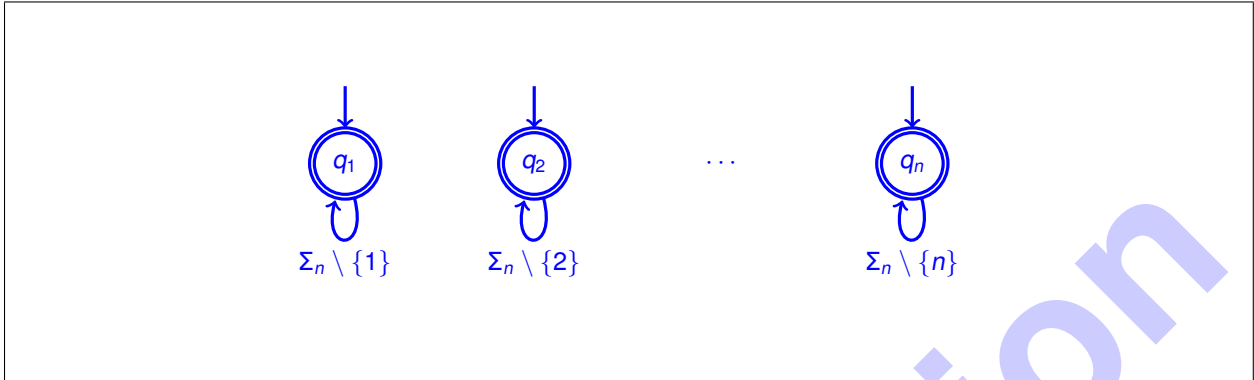
False. The computation $\{p\}\{r\}\emptyset^\omega$ satisfies $p \mathbf{U} (q \mathbf{U} r)$, but not $(p \mathbf{U} q) \mathbf{U} r$.

Problem 2 Regular languages (7 credits)

For $n \geq 1$, we define $\Sigma_n := \{1, \dots, n\}$ and $L_n := \{w \in \Sigma_n^* \mid \text{at least one letter of } \Sigma_n \text{ does not appear in } w\}$. For example, we have $\varepsilon, 1111, 23 \in L_3$ and $123, 2231 \notin L_3$.



a)* Sketch an NFA for L_n with $\mathcal{O}(n)$ states.

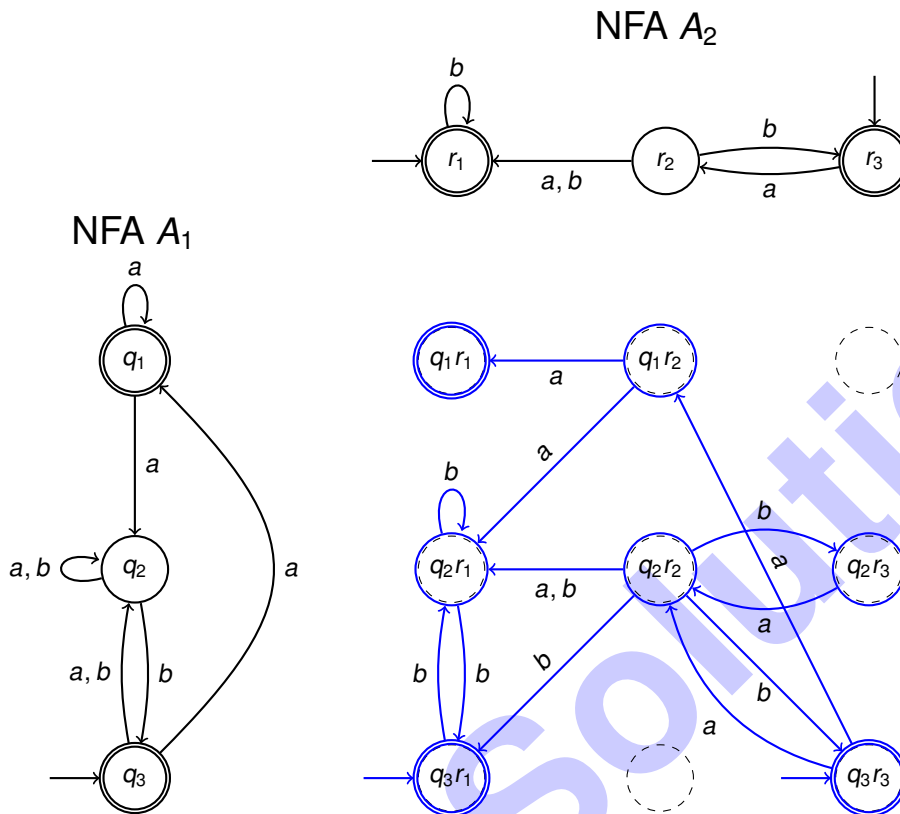


b)* **Bonus points:** Show using residuals that every DFA for L_n has at least 2^n states.

Let $u, v \in \Sigma_n^*$ be such that the set S_u of letters appearing in u is not equal to the set S_v of letters appearing in v . We show that $L_n^u \neq L_n^v$. W.l.o.g. let $S_u \not\subseteq S_v$. Then there exists a letter $a \in S_u \setminus S_v$. Let w be a word containing every letter in Σ_n except a . Then $w \in L_n^v$, but $w \notin L_n^u$, so $L_n^u \neq L_n^v$. Since u, v were arbitrary words such that $S_u \neq S_v$, L_n has at least as many residuals as there are possible sets S_u , i.e. at least one per subset of Σ_n , i.e. at least 2^n . Since the number of states in a DFA is at least the number of residuals, the statement follows.

Problem 3 Product construction (11 credits)

a)* Compute an NFA for $L(A_1) \cap L(A_2)$ using the product construction. You only need to construct reachable states and transitions.



0
1
2
3
4
5
6

Now consider the NGA (generalized Büchi automaton) B_1 and the NBA B_2 defined as follows:

- B_1 has the same states, transitions, and initial states as A_1 , and acceptance condition $\{\{q_1\}, \{q_3\}\}$ (that is, B_1 has two sets of accepting states, $\{q_1\}$ and $\{q_3\}$).
- B_2 is just A_2 , but interpreted as a Büchi automaton; in particular, the set of accepting states of B_2 is $\{r_1, r_3\}$.

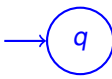
b)* Consider the NGA B with $L_\omega(B) = L_\omega(B_1) \cap L_\omega(B_2)$ which results from using the operation for intersection of NGAs described in the course. What is the acceptance condition of B ?

$\{\{q_1\} \times Q_2, \{q_3\} \times Q_2, Q_1 \times \{r_1, r_3\}\}$
 If we only consider reachable states: $\{\{q_1r_1, q_1r_2\}, \{q_3r_1, q_3r_3\}, \{q_1r_1, q_2r_1, q_3r_1, q_2r_3, q_3r_3\}\}$

0
1
2

c)* Give a minimal NGA C accepting the same language as the NGA B from (b). Justify your answer.

Since q_1 must be visited infinitely often, B_1 only accepts words which contain aa infinitely often. But no word in $L(B_2)$ contains aa more than once. Therefore $L_\omega(B_1) \cap L_\omega(B_2) = \emptyset$.
 Alternative solution: As we see by the automaton in a), no run of B can visit $\{q_1\} \times Q_2$ infinitely often. Therefore $L_\omega(B) = \emptyset$.

A minimal NGA C is e.g.  with acceptance condition $\{\emptyset\}$.

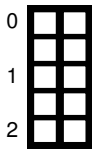
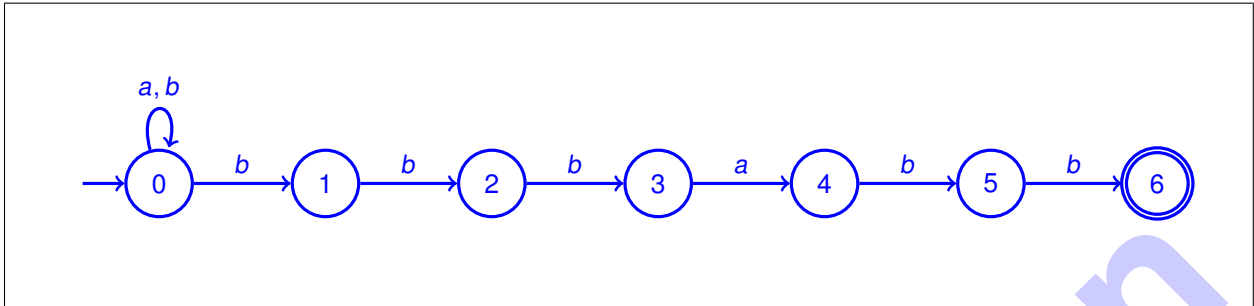
0
1
2
3

Problem 4 Pattern matching (6 credits)

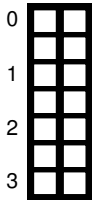
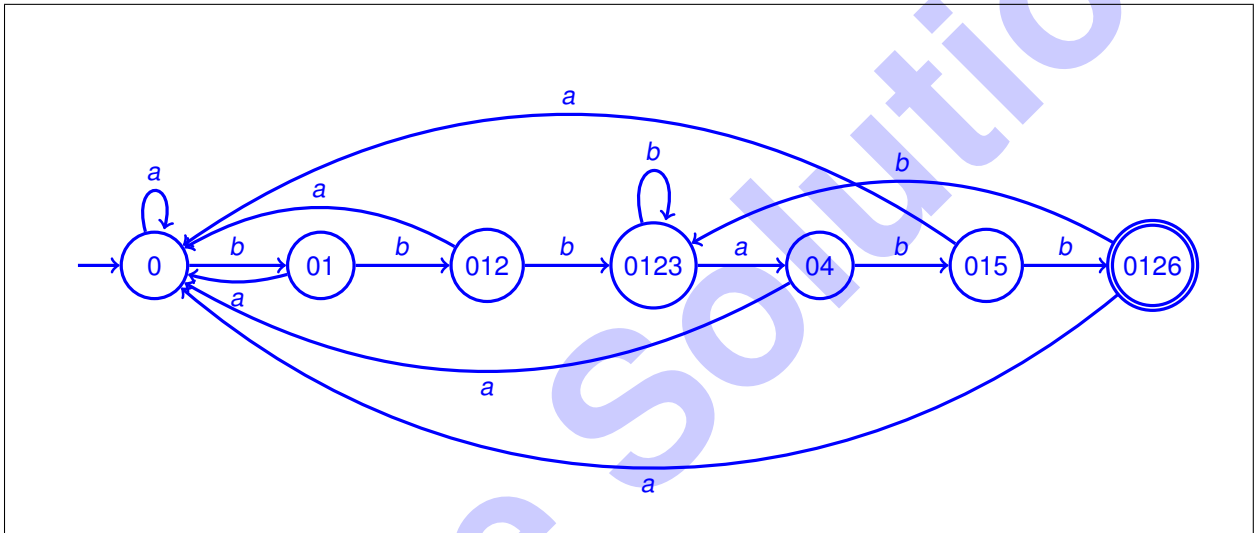
Consider the pattern $p = bbbabb$ over the alphabet $\Sigma = \{a, b\}$.



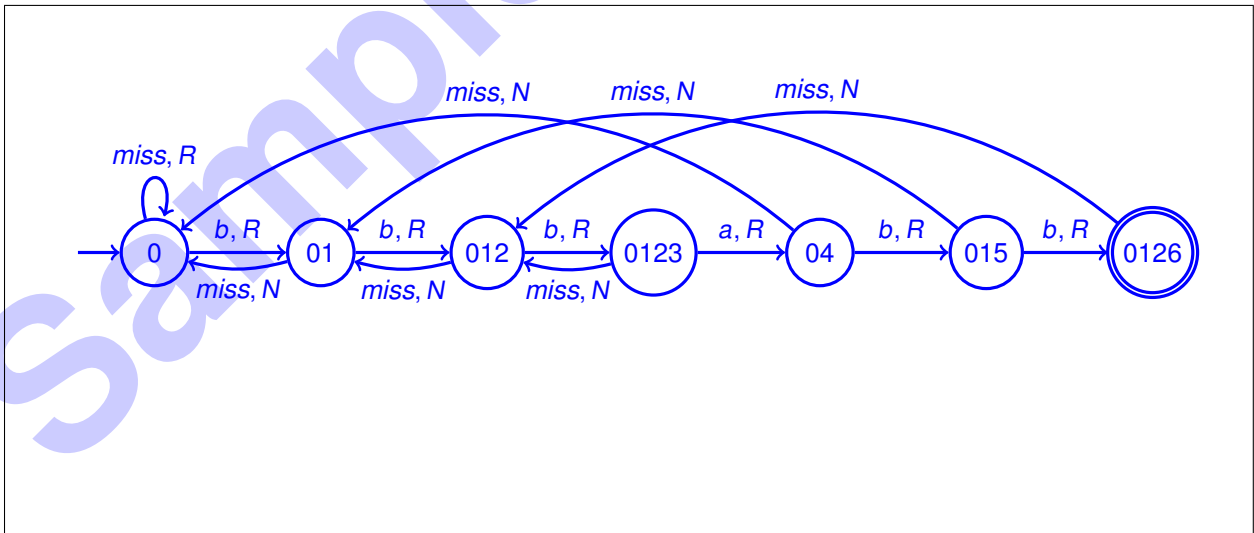
a)* Construct the NFA A_p recognizing Σ^*p according to the construction specified in the course.



b) Construct the DFA B_p by applying the powerset construction to A_p .



c) Construct the lazy DFA C_p for the pattern p by using B_p .



Problem 5 Transducers (8 credits)

For this exercise, let $\Sigma = \{a, b\}$.

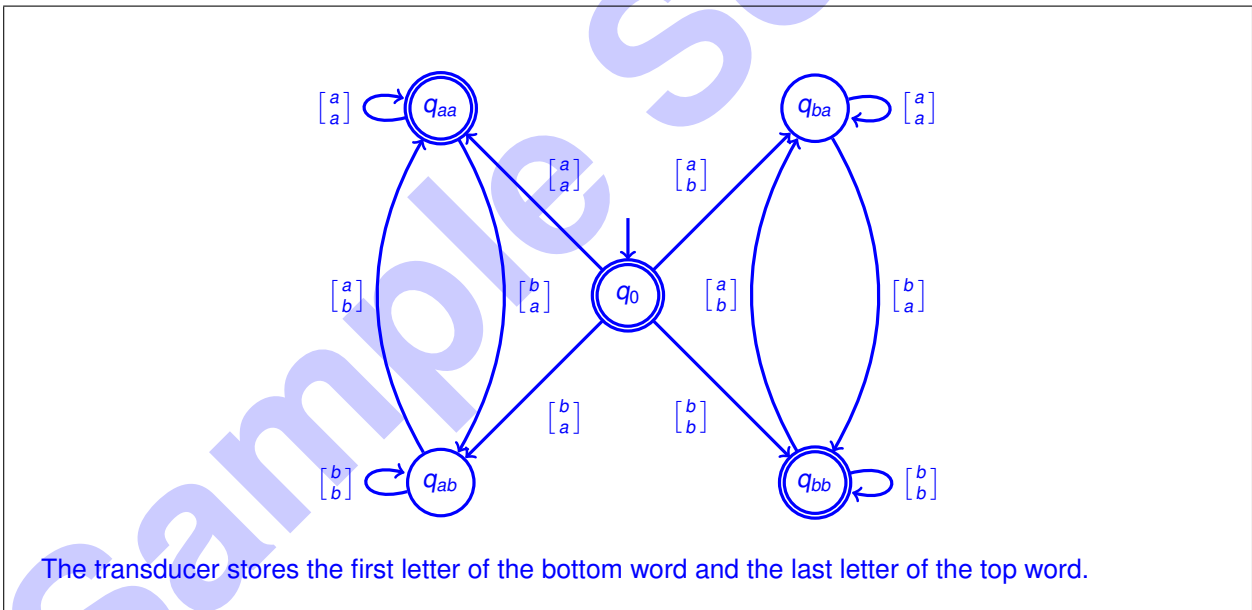
a)* Is there a transducer for the relation $R_1 := \{(a_1 \cdots a_n, a_n \cdots a_1) \mid n \in \mathbb{N}, a_1, \dots, a_n \in \Sigma\}$? If yes, draw one. If no, prove it using residuals.

0
1
2
3
4

No. For $n \in \mathbb{N}$, let $w_n = \begin{bmatrix} a^n \\ a^n \end{bmatrix}$. For all $i, j \in \mathbb{N}, i \neq j$, we have $\begin{bmatrix} ba^i \\ ba^i \end{bmatrix} \in R_1^{w_i}$, but $\begin{bmatrix} ba^i \\ ba^i \end{bmatrix} \notin R_1^{w_j}$. Hence $R_1^{w_i} \neq R_1^{w_j}$. This shows that R_1 has infinitely many residuals and therefore there is no transducer for R_1 .

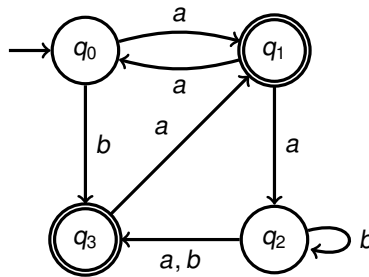
b)* Is there a transducer for the relation $R_2 := \{(a_1 \cdots a_n, a_n a_1 \cdots a_{n-1}) \mid n \in \mathbb{N}, a_1, \dots, a_n \in \Sigma\}$? If yes, draw one. If no, prove it using residuals.

0
1
2
3
4



Problem 6 Büchi Complementation (10 credits)

Consider the following Büchi automaton \mathcal{B} :



We denote by $\bar{\mathcal{B}}$ the complement of \mathcal{B} , defined using rankings and owing states as in the lecture. We write the states of $\bar{\mathcal{B}}$ as pairs $[r, O]$ where r is a ranking and O is a set of owing states. In r , we write the ranks of the nodes in the order q_0, q_1, q_2, q_3 from top to bottom.

a)* For each of the following pairs $[r, O]$, decide whether it is a state of $\bar{\mathcal{B}}$. If it is not, give a justification.

Note: The question is whether the pairs are states, not whether they are reachable from the initial states.

0	
1	
2	
3	
4	

1. $\left[\begin{array}{c} \perp \\ 2 \\ 5 \\ 0 \end{array}, \{q_0, q_1, q_3\} \right]$

2. $\left[\begin{array}{c} 6 \\ 0 \\ 7 \\ \perp \end{array}, \emptyset \right]$

3. $\left[\begin{array}{c} 5 \\ 5 \\ \perp \\ 2 \end{array}, \{q_3\} \right]$

4. $\left[\begin{array}{c} 1 \\ 2 \\ 2 \\ 0 \end{array}, \{q_0, q_1\} \right]$

1. This is not a state: q_0 does not have a rank and thus cannot be owing.
2. This is a state.
3. This is not a state: q_1 is accepting and thus cannot have an odd rank.
4. This is not a state: q_0 has an odd rank and thus cannot be owing.

b)* For each of the following transitions, decide whether it is a transition of \bar{B} . If it is not, give a justification.

Note: The question is whether these are transitions, not whether they are reachable transitions.

$$1. \begin{bmatrix} \perp \\ 0 \\ \perp \\ 4 \end{bmatrix}, \{q_1, q_3\} \xrightarrow{a} \begin{bmatrix} 0 \\ 2 \\ \perp \\ \perp \end{bmatrix}, \{q_0, q_1\}$$

$$2. \begin{bmatrix} 4 \\ 4 \\ \perp \\ \perp \end{bmatrix}, \{q_1\} \xrightarrow{a} \begin{bmatrix} 3 \\ 2 \\ 3 \\ \perp \end{bmatrix}, \{q_1\}$$

$$3. \begin{bmatrix} 1 \\ \perp \\ 6 \\ 4 \end{bmatrix}, \{q_3\} \xrightarrow{b} \begin{bmatrix} \perp \\ \perp \\ 6 \\ 4 \end{bmatrix}, \emptyset$$

$$4. \begin{bmatrix} 2 \\ 4 \\ \perp \\ \perp \end{bmatrix}, \{q_0, q_1\} \xrightarrow{a} \begin{bmatrix} 2 \\ 0 \\ 2 \\ \perp \end{bmatrix}, \{q_0, q_1\}$$

$$5. \begin{bmatrix} 3 \\ 4 \\ 5 \\ \perp \end{bmatrix}, \{q_1\} \xrightarrow{a} \begin{bmatrix} 4 \\ 2 \\ 1 \\ 0 \end{bmatrix}, \{q_0\}$$

$$6. \begin{bmatrix} 3 \\ \perp \\ 4 \\ 0 \end{bmatrix}, \emptyset \xrightarrow{b} \begin{bmatrix} 1 \\ \perp \\ 3 \\ 2 \end{bmatrix}, \{q_3\}$$

	0
	1
	2
	3
	4
	5
	6

1. This is not a transition: $q_1 \xrightarrow{a} q_2$, so q_2 should have a rank.
2. This is not a transition: $q_1 \notin \delta(q_1, a)$, so q_1 should not be owing.
3. This is not a transition: $q_0 \xrightarrow{b} q_3$, but the rank of q_0 in the source state is lower than the rank of q_3 in the target state.
4. This is not a transition: q_2 should be owing.
5. This is a transition.
6. This is not a transition: q_0 does not have an incoming b -transition and thus should not have a rank.

Problem 7 Construction (18 credits)

Answer the following questions. No justification is needed.

0 1 2 3 a)* Consider the regular expression $r = a^*b^*c^* + c^*a^*$ over the alphabet $\{a, b, c\}$. Give regular expressions for *all* residuals of $L(r)$.

$r, a^*b^*c^*, b^*c^*, c^*, \emptyset, c^*a^*, a^*$

0 1 2 b)* Let $L := \{aab, abb, bab, bbb\}$. Give the kernel $\langle L \rangle$ of L .

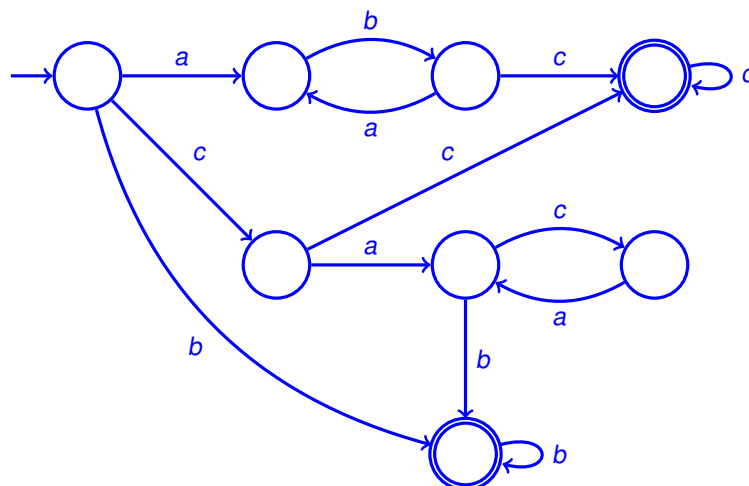
$\langle L \rangle = \{b\}$

0 1 2 3 4 c)* Give formulas of first-order logic on words for the following languages over the alphabet $\{a, b, c\}$. You may use the macros $first(x)$, $last(x)$ and $x = y + 1$, $x = y + 2$, ... without definition. If you use other macros, you must define them first.

1. $L(a^*ba^*c)$
2. $L((a + b)^*cc)$
3. $L((ab)^*)$

1. $\exists x \exists y (Q_b(x) \wedge Q_c(y) \wedge last(y) \wedge \forall z (z < x \vee (x < z \wedge z < y) \rightarrow Q_a(z)))$
2. $\exists x \exists y (last(y) \wedge y = x + 1 \wedge Q_c(x) \wedge Q_c(y) \wedge \forall z (z < x \rightarrow Q_a(z) \vee Q_b(z)))$
3. $\forall x (first(x) \rightarrow Q_a(x)) \wedge (last(x) \rightarrow Q_b(x)) \wedge \forall y (y = x + 1 \rightarrow (Q_a(x) \leftrightarrow Q_b(y)))$

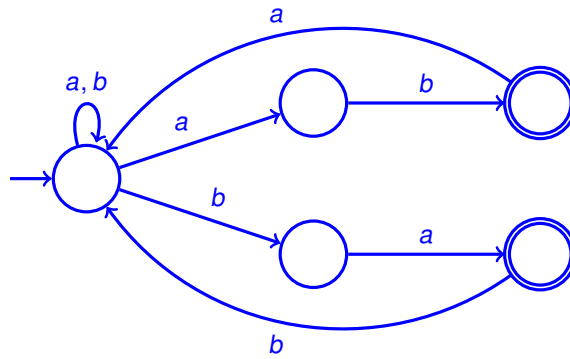
0 1 2 3 4 d)* Give a *deterministic* Büchi automaton over the alphabet $\{a, b, c\}$ for the language of the ω -regular expression $(ab)^*c^\omega + (ca)^*b^\omega$.



e)* Let $\Sigma = \{a, b\}$. We say that a letter is *isolated* if it is neither preceded nor followed by the same letter. For example, in the word $aaab**aa**bab^\omega$ the three highlighted letters are isolated. Let $L \subseteq \Sigma^\omega$ be the language of ω -words where infinitely many letters are isolated. Give an NBA for L .

	0
	1
	2
	3

L can be described as “ aba infinitely often or bab infinitely often”.



f)* Let $AP = \{p, q\}$. Give an ω -regular expression for all computations over $\Sigma = 2^{AP}$ satisfying $(Gp) U q$.

	0
	1
	2

$$(\{q\} + \{p, q\})\Sigma^\omega + \{p\}^*\{p, q\}(\{p\} + \{p, q\})^\omega$$

Additional space for solutions—clearly mark the (sub)problem your answers are related to and strike out invalid solutions.

Sample Solution

Sample Solution