Chair for Foundations of Software Reliability and Theoretical Computer Science
Informatik
Technical University of Munich

Esolution
Place student sticker here

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.

Automaten und formale Sprachen

Exam:	IN2041 / Retake	**Date:**	Thursday 14th April, 2022
Examiner:	Prof. Javier Esparza	**Time:**	17:00 – 19:00

Working instructions

- This exam consists of **10 pages** with a total of **8 problems**.
- You can obtain a maximum of 40 points. There are 5 bonus points.
- Allowed resources:
  - Any electronic resources accessible using only the external mouse.
  - You can cite results from the lecture notes or slides, but results from the exercises must be rewritten in full. For example, you cannot write something like "this is true by exercise 3.1(a)."
- All answers have to be written on your own paper.
- Only write on one side of each sheet of paper.
- Write with black or blue pen on white DIN A4 paper.
- Write the page number, your name and immatriculation number on every sheet.

Problem 1 Statement (0 credits)

0 ☐ **It is MANDATORY that your answer sheet includes a signed copy of the following statement.**

"I did not communicate with anyone during this graded exercise and only used the allowed resources."

Problem 2 Omega-regular languages (6 credits)

a) Let L_1, L_2 be two ω-regular languages over the same alphabet. Show that if L_1, L_2 are distinct, then there are finite words u, v such that $u(v)^\omega$ is in the symmetric difference of L_1 and L_2. Recall that the symmetric difference is $(L_1 \cup L_2) \setminus (L_1 \cap L_2)$.

0 1 2 3

The symmetric difference of two ω-regular languages is still ω-regular, because ω-regular languages are closed under union, intersection and difference. Since L_1, L_2 are distinct, their symmetric difference is non-empty.
Let L be any non-empty ω-regular language, and let $B = (Q, \{0, 1\}, \delta, Q_0, F)$ be an NBA that recognizes L. Since Q is finite, there exist $u \in \Sigma^*, v \in \Sigma^+, q_0 \in Q_0$ and $q \in F$ such that

$$q_0 \xrightarrow{u} q \xrightarrow{v} q.$$

Consequently, we have $uv^\omega \in L$ by iterating v from state q. Applying this to $L = (L_1 \cup L_2) \setminus (L_1 \cap L_2)$, we conclude our proof.

b) Let R be a language and let L be an ω-language over the same alphabet. Consider the following statement:

0 1 2 3

> If R is a non-empty regular language and $L \cap R^\omega$ is a non-empty ω-regular language, then L is an ω-regular language.

If it is true, prove it. If it is false, give a counter-example.

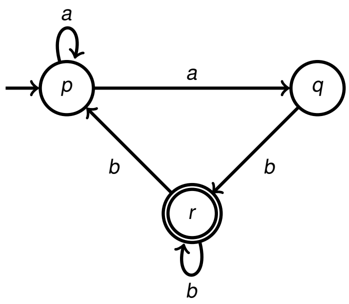
It is false. Let $\Sigma = \{0, 1, \dots, 9\}$ be an alphabet. Let R be the regular language $\{0\}$, and take $L = \{0^\omega, w\}$ where $w = w_1 w_2 w_3 \dots$ where w_i is the i-th decimal of square root of 2. Then $L \cap R^\omega$ equals $\{0^\omega\}$, which is an ω-regular language. But L is not ω-regular.
Suppose there exists a Büchi automaton $B = (Q, \{0, 1, \dots, 9\}, \delta, Q_0, F)$ such that $L_\omega(B) = \{0^\omega, w\}$. There exist $u \in \{0, 1, \dots, 9\}^*$, $v \in \{0, 1, \dots, 9\}^+$, $q_0 \in Q_0$ and $q \in F$ such that

$$q_0 \xrightarrow{u} q \xrightarrow{v} q \text{ and } uv \text{ is a prefix of } w.$$

Therefore, $uv^\omega \in L_\omega(B)$. But $uv^\omega \neq 0^\omega$ since uv is a prefix of w, and $uv^\omega \neq w$ since w is not periodic. This amounts to a contradiction.

Problem 3 Co-Büchi determinization (6 credits)

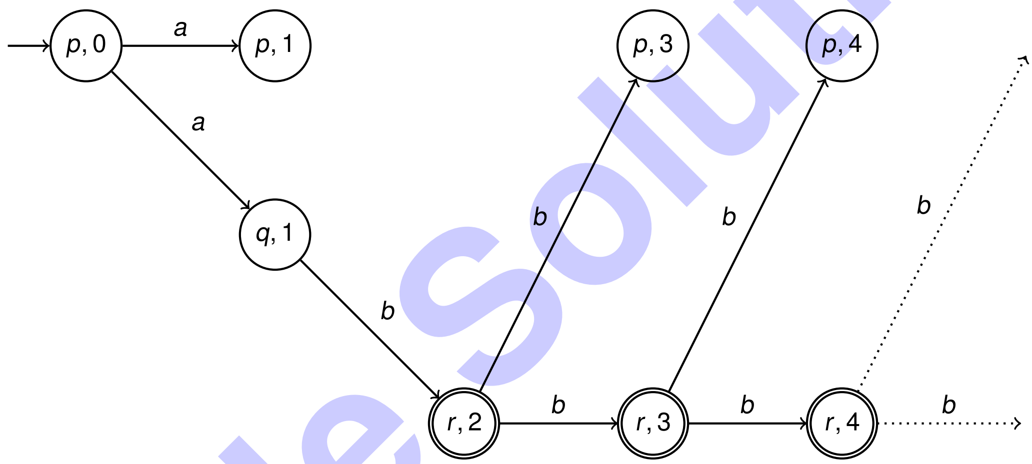
Consider the following co-Büchi automaton $\mathcal{A}$.



Recall that the procedure for determinizing a co-Büchi automaton is very similar to the procedure for complementing a Büchi automaton.

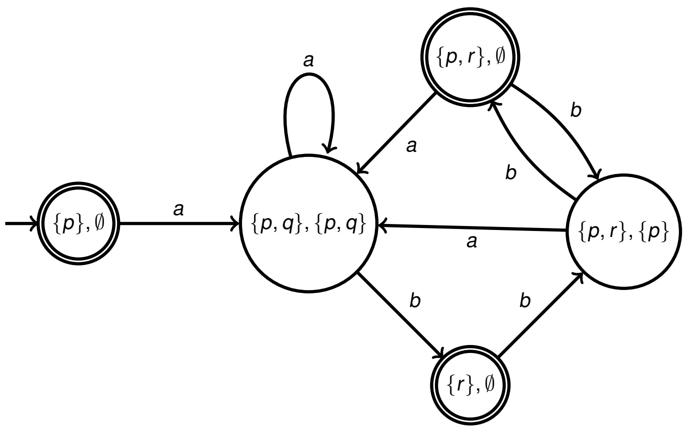
0 1 2

a) Sketch $dag(w)$ for $w = a(b)^\omega$ by drawing the first 5 levels (i.e. levels 0, 1, 2, 3 and 4).



0 1 2 3 4

b) Determinize $\mathcal{A}$: give a deterministic co-Büchi automaton recognizing the same language as $\mathcal{A}$. You must follow the construction given in class; in particular label the states with sets $[P, O]$ as in class. You may draw the trap state or omit it.



Problem 4 Double and half (9 credits)

Let $\Sigma = \{a, b\}$.

a) Prove or disprove: for every regular language $L \subseteq \Sigma^*$, the language $Double(L) = \{ww \in \Sigma^* : w \in L\}$ is regular.
To prove: show how to construct an automaton for $Double(L)$, given an automaton for L.
To disprove: give a regular language L and show that $Double(L)$ contains infinitely many residuals.

Take $L = \Sigma^*$. Let $w, w' \in \Sigma^*$ arbitrary, but distinct. We have $w \in Double(L)^w$ and $w \notin Double(L)^{w'}$. So $Double(L)^w \neq Double(L)^{w'}$. It follows that $Double(L)$ has infinitely many residuals.

b) Prove or disprove: for every regular language $L \subseteq \Sigma^*$ the language $Half(L) = \{w \in \Sigma^* : ww \in L\}$ is regular.
To prove: show how to construct an automaton for $Half(L)$, given an automaton for L.
To disprove: give a regular language L and show that $Half(L)$ contains infinitely many residuals

One can prove it, as in the exercise 5.2 (c).

c) Prove or disprove: for every regular language $L \subseteq \Sigma^*$, the language $Replicate(L) = \{w^{|w|} \in \Sigma^* : w \in L\}$ is regular.
For example, if $L = \{ab, abb\}$ then $Replicate(L) = \{abab, abbabbabb\}$.
To prove: describe how to construct an automaton for $Replicate(L)$, given an automaton for L.
To disprove: give a regular language L and show that $Replicate(L)$ contains infinitely many residuals.

to be written.

Problem 5 LTL formulas (6 credits)

In the following questions, each formula you give MUST contain at most six symbols, excluding parenthesis. The symbols allowed are $p, q, \neg, \vee, \wedge, X, U, F, G$.

0 1 2 3

a) Give LTL formulas φ_1 and φ_2 such that

- $\{q\}\{p, q\}^\omega \models \varphi_1 \wedge \varphi_2$
- $\{q\}^\omega \models \varphi_1 \wedge \neg\varphi_2$
- $\{p\}^\omega \models \neg\varphi_1 \wedge \varphi_2$
- $\emptyset^\omega \models \neg\varphi_1 \wedge \neg\varphi_2$

$\varphi_1 = GFq$, $\varphi_2 = GFp$.

0 1 2 3

b) **For 3 bonus points.** Give LTL formulas φ_1 and φ_2 such that:

- $(\{p, q\}\{p\})^\omega \models \varphi_1 \wedge \varphi_2$
- $\emptyset^\omega \models \varphi_1 \wedge \neg\varphi_2$
- $\emptyset\{q\}^\omega \models \neg\varphi_1 \wedge \varphi_2$
- $\{q\}^\omega \models \neg\varphi_1 \wedge \neg\varphi_2$

$\varphi_1 = GF(p \vee \neg q)$, $\varphi_2 = Fq \wedge \neg Gq$

Problem 6 MSO and regular languages (8 credits)

Throughout this exercise, you are only allowed to use the following standard expressions in specifying an MSO formula:

$$Q_a(x),\ Q_b(x),\ x < y,\ x \in X,\ \neg\varphi,\ \varphi_1 \vee \varphi_2,\ \exists x\varphi,\ \exists X\varphi$$

and the abbreviations

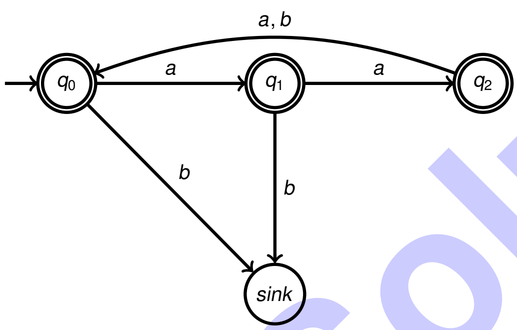
$$\forall x\varphi,\ \forall X\varphi,\ \varphi_1 \wedge \varphi_2,\ \varphi_1 \rightarrow \varphi_2,\ \varphi_1 \leftrightarrow \varphi_2,\ x = y,\ x \leq y,\ first(x),\ last(x),\ y = x + k$$

where k is a constant. If you want to use any other abbreviations, you must first define them.

Let $L = \{w \in \{a,b\}^* :$ for all positions i of the word w, if b appears at position i, then $i \equiv 0 \pmod 3\}$. (Recall that positions of a word start at 1).

a) Give a regular expression and a minimal DFA for the language L.

A regular expression for L is $(aa(a+b))^*(\epsilon + a + aa)$. The minimal DFA for L is the following:



b) Give an MSO formula $ZeroMod3(X)$ which satisfies the following: For any interpretation $(w, \mathcal{I})$ of the formula, $(w, \mathcal{I}) \models ZeroMod3(X)$ iff $\mathcal{I}(X) = \{i : i \in \{1, 2, ..., |w|\}$ and $i \equiv 0 \pmod 3\}$, i.e., $\mathcal{I}(X)$ is precisely the set of positions of w which are divisible by 3.

We first define

$$third(x) := \exists y(first(y) \wedge x = y + 2)$$

Then we set

$$ZeroMod3(X) := \forall x(x \in X \leftrightarrow (third(x) \vee \exists y(x = y + 3 \wedge y \in X)))$$

c) Give an MSO formula φ such that $L(\varphi) = L$. You can use the $ZeroMod3(X)$ formula for this subproblem.

We set φ to be

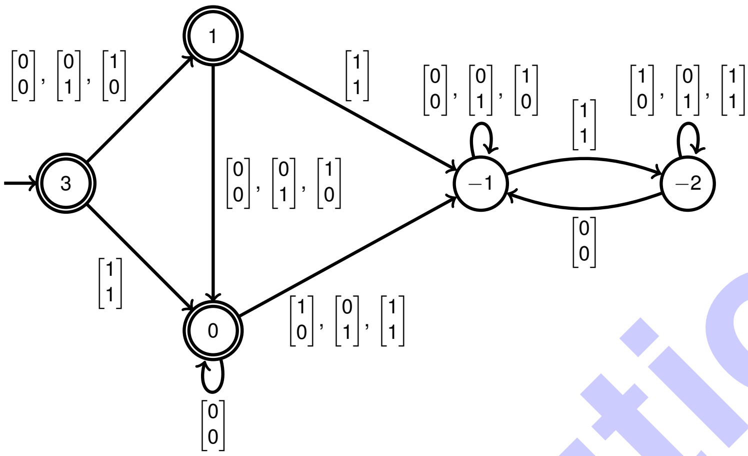
$$\exists X.ZeroMod3(X) \wedge \forall x.(Q_b(x) \rightarrow x \in X)$$

Problem 7 Presburger arithmetic and automata (6 credits)

Consider the inequality $\varphi = x + y \leq 3$.

0 1 2 3 4

a) Use the algorithm *AFtoDFA* to obtain a DFA recognizing the lsbf encoding of the solutions of φ over the naturals.



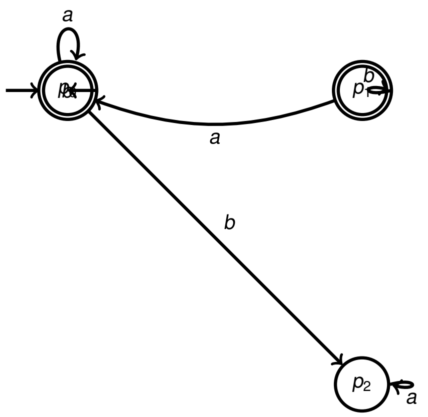
0 1 2

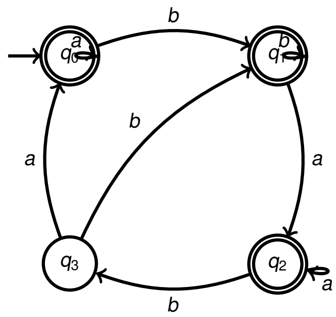
b) **For 2 bonus points.** Let ϕ be the inequality $x + y \leq 192$ and let A be the DFA obtained by applying the *AFtoDFA* algorithm on ϕ. Prove or disprove that A is a minimal DFA.
To prove: Show that all the states of A recognize different languages.
To disprove: Give two distinct states of A which recognize the same language.

On reading $\begin{bmatrix}0\\0\end{bmatrix}$, a state q will always go to $\lfloor \frac{q}{2} \rfloor$. This means that starting at 192, the algorithm will eventually construct the state 3. From state 3, we know by the previous subproblem that the algorithm will construct states -1 and -2. But states -1 and -2 accept the same language and so A is not a minimal DFA.

Problem 8 NFA inclusion (4 credits)

Consider the following NFAs A and B where A has the states $\{p_0, p_1, p_2\}$ and B has the states $\{q_0, q_1, q_2, q_3\}$.





Use the algorithm *IncNFA* to determine if $L(A) \subseteq L(B)$. You must give your answer in the following table format, where the i^{th} row contains the iteration number of the main **while** loop of the *IncNFA* algorithm, the contents of Q and W at the **beginning of that iteration** and the state that you pick from W during that iteration. The first entry of the table has been filled for you.

Iter.	Q	W	Chosen element
1	$\emptyset$	$\{[p_0, \{q_0\}]\}$	$[p_0, \{q_0\}]$
2	⋮	⋮	⋮
⋮	⋮	⋮	⋮

Iter.	Q	W	Chosen set
1	$\emptyset$	$\{[p_0, \{q_0\}]\}$	$\{[p_0, \{q_0\}]\}$
2	$\{[p_0, \{q_0\}]\}$	$\{[p_1, \{q_1\}], [p_2, \{q_1\}]\}$	$\{[p_1, \{q_1\}]\}$
3	$\{[p_0, \{q_0\}], [p_1, \{q_1\}]\}$	$\{[p_2, \{q_1\}], [p_0, \{q_2\}]\}$	$\{[p_0, \{q_2\}]\}$
4	$\{[p_0, \{q_0\}], [p_1, \{q_1\}], [p_0, \{q_2\}]\}$	$\{[p_2, \{q_1\}], [p_1, \{q_3\}], [p_2, \{q_3\}]\}$	$\{[p_2, \{q_3\}]\}$

After the fourth iteration, we have that p_2 is a final state, but no state in $\{q_3\}$ is a final state. Hence, the algorithm returns *false* in the fifth iteration and so $L(B) \not\subseteq L(C)$.

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