Chair for Foundations of Software Reliability and Theoretical Computer Science
School of Computation, Information and Technology
Technical University of Munich

Eexam
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.

Automata and Formal Languages

Exam:	IN2041 / Retake	**Date:**	Wednesday 3rd April, 2024
Examiner:	Prof. Javier Esparza	**Time:**	17:00 – 19:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7
I							

Working instructions

- This exam consists of **12 pages** with a total of **7 problems** including two **bonus questions**. Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 80 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.
- The points of the bonus problems count for your grade, but we disregard them when calculating the grading scheme. In particular, to receive the best grade it suffices to achieve all non-bonus points.
- **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 (19 credits)**

Please answer the following questions. For true/false questions, provide a justification of your answer for "true" answers, and a counterexample for "false" answers. Otherwise **no points will be awarded!** We use $\Sigma := \{a, b\}$ as alphabet in this exercise.

0 1 2

a)* Let $A, B \subseteq \Sigma^*$. True or false: if $A \subseteq B$ and B is regular, then A is regular.

☐ True ☐ False

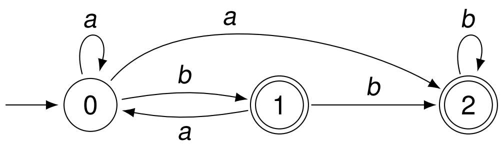
0 1 2

b)* Let $L \subseteq \Sigma^*$ denote the language of words that do not contain *bba*. True or false: every NFA N for L has at least 4 states.

☐ True ☐ False

0 1 2 3

c)* Give an ω-regular expression r for the language of this Büchi automaton over the alphabet $\{a, b\}$. Either use the algorithm from the lecture or justify your answer.



0 1 2

d)* Let AP = $\{p, q\}$ be a set of atomic propositions. Give an LTL formula φ over AP such that $L(\varphi)$ is the set of all computations where exactly one atomic proposition in AP occurs infinitely often.

e)* We say that a letter is *isolated*, if it is neither preceded nor followed by an occurrence of the same letter. For example, in the word $aaa\mathbf{b}aa\mathbf{ba}b^\omega$ the three highlighted letters are isolated.
Let $L \subseteq \Sigma^\omega$ denote the language of ω-words where only a finite number of letters are isolated. Give a Büchi-automaton for L with at most 5 states.

☐ 0 ☐ 1 ☐ 2

f)* Let $L \subseteq \Sigma^\omega$ denote an ω-regular (!) language and let $L' \subseteq \Sigma^*$ denote the *finite* prefixes of words in L, i.e. $L' := \{u \in \Sigma^* \mid \exists\, v \in \Sigma^\omega : uv \in L\}$. True or false: L' is regular.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

☐ True ☐ False

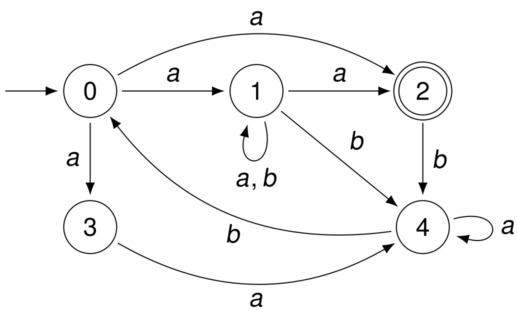
g)* **Bonus question:** Let $L \subseteq \Sigma^*$. True or false: if $L \cup L^R$ is regular, then so is L.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

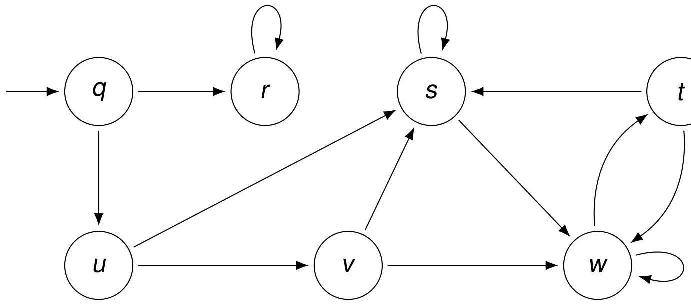
☐ True ☐ False

Problem 2 **Minimal Power (8 credits)**

Consider the following NFA N over the alphabet $\Sigma := \{a, b\}$.



a)* Convert N into an equivalent DFA M using the power set construction from the lecture. A template for M is given below. Complete it by annotating each transition with the corresponding letter(s), and by marking the final states.



b) Partition the set of states of M into equivalence classes.

Note: It suffices to provide the partition.

Problem 3 **Residuals (18 credits)**

Let L, L_1, L_2 be regular languages with n residuals. Prove or disprove:

a)* $\overline{L}$ has exactly n residuals.

☐ True ☐ False

0 1 2 3 4

b)* $L_1 \cup L_2$ has at most n^2 residuals.

☐ True ☐ False

0 1 2 3 4

c)* $L_1 L_2$ has at least n residuals.

☐ True ☐ False

0 1 2 3 4

0 1 2 3 4 5 6

d)* **Bonus question:** $L_1 L_2$ has at most $2n$ residuals.

☐ True ☐ False

Problem 4 Fixed-length Languages (10 credits)

Let $\Sigma := \{a_1, ..., a_m\}$ denote an alphabet. Consider the following algorithm:

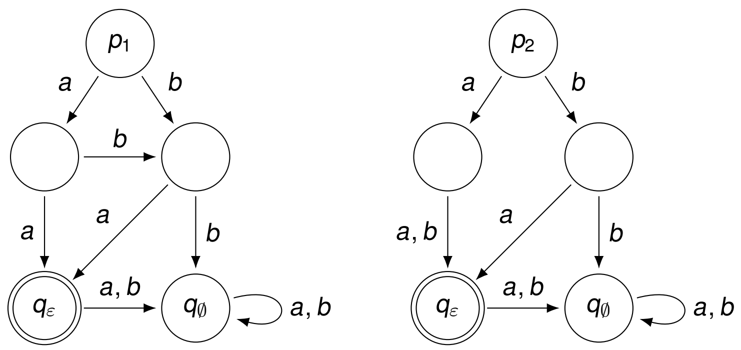
diff(q_1, q_2)
Input: states q_1, q_2 of the master automaton recognising languages of the same length
Output: state recognising $L(q_1) \setminus L(q_2)$

1 **if** $G(q_1, q_2)$ *is not empty* **then return** $G(q_1, q_2)$
2 **if** [1] **then return** $q_\emptyset$
3 **else if** [2] **then return** q_ε
4 **else**
5   **forall** $i = 1, ..., m$ **do** [3]
6     $G(q_1, q_2) \leftarrow \texttt{make}(r_1, ..., r_m)$
7   **return** $G(q_1, q_2)$

a)* Complete the algorithm by giving contents of the boxes 1, 2 and 3, such that it fulfils its specification.

1: 2: 3:

b)* Which of the following states p_i are possible outputs of the above algorithm? Justify your answer.



☐ p_1 ☐ p_2 ☐ p_3

c)* We now modify the above algorithm by deleting line 1. Let n denote the total size of the automata rooted at q_1 and q_2. Which of the following are true of the modified algorithm?

☐ The algorithm is incorrect: it may produce a wrong result.

☐ The algorithm is correct, but it may produce a different result than the unmodified algorithm.

☐ This modification does not change the output of the algorithm.

☐ The running time is always polynomial in n.

☐ The running time may be exponential in n.

Problem 5 **MSO (6 credits)**

These exercises ask you to give DFAs recognizing the languages of different formulas of monadic second-order logic. You do not need to use the algorithm of the course to construct them. Pay attention to using the correct alphabets. For example, a formula $\varphi \in \mathrm{MSO}(\Sigma)$ with one free variable x and $\Sigma := \{a, b\}$ will use the alphabet $\{a, b\} \times \{0, 1\}$. The word *abb* with $x = 2$ then corresponds to $[{a \atop 0}][{b \atop 1}][{b \atop 0}]$. Note that the lower part must contain exactly one 1, since x is a first-order variable.

0 1 2

a)* Let $\Sigma_1 = \{a, b\}$ and let $\varphi_1 = Q_a(x) \wedge Q_b(y)$ be a formula of $\mathrm{MSO}(\Sigma_1)$. Give a DFA recognizing $L(\varphi_1)$.

0 1 2

b)* Let $\Sigma_2 = \{a\}$ and let $\varphi_2 = \exists x\ x < y$ be a formula of $\mathrm{MSO}(\Sigma_2)$. Give a DFA recognizing $L(\varphi_2)$.

0 1 2

c)* Let $\Sigma_3 = \{a\}$ and let $\varphi_3 = \exists x\ x \in X$ be a formula of $\mathrm{MSO}(\Sigma_3)$. Give a DFA recognizing $L(\varphi_3)$.

Problem 6 LTL (9 credits)

In these exercises, we consider LTL formulas over the set $AP = \{p, q\}$ of atomic propositions. Let $\Sigma := 2^{AP} = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$.

a)* Give an ω-regular expression for the language of the formula $\mathrm{GF}(p \wedge \mathrm{X}q)$ over the alphabet Σ:

0 1 2

b)* Give a deterministic co-Büchi automaton recognizing the language of the formula $\mathrm{FG}(p \wedge \mathrm{X}q)$ over Σ.

0 1 2 3 4

c)* Give a Büchi automaton (deterministic or not) for the language of $\mathrm{FG}(p \,\mathrm{U}\, q)$ over Σ.

0 1 2 3

Problem 7 **Relations (10 credits)**

Let $R \subseteq \mathbb{N} \times \mathbb{N}$ be a relation on natural numbers. We say that a number $n \in \mathbb{N}$ *has a two-loop* w.r.t. R if there exists $m \neq n$ such that $(n, m) \in R$ and $(m, n) \in R$. (**Important**: $n \neq m$!) We let TL_R denote the set of all numbers that have a two-loop w.r.t. R.

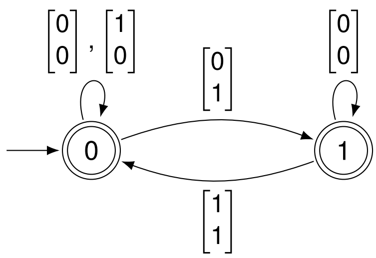
a) Give an algorithm satisfying the following specification:

- Input: a well-formed deterministic transducer recognizing a relation $R \subseteq \mathbb{N} \times \mathbb{N}$ in lsbf encoding.

  Note: Recall that a transducer recognizes R if for every pair $(n, m) \in R$ it accepts every encoding of (n, m), and for every pair $(n, m) \notin R$ it accepts no encoding of (n, m).

- Output: NFA N recognizing the set TL_R.

b) Apply the algorithm of part (a) to the transducer below, giving enough information about the intermediate steps. Interpret the result by describing TL_R not as a language, but as a set of numbers. (Solutions like “the set of numbers encoded by this language” get no points.)



Hint: In the model solution, no intermediate automaton has more than 3 states.

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