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

1								
2								
3								
4								
5								
6								
7								
8								
9								
0	☒							

Registration number

Signature

Note:
- Cross your Registration number(with leading zero). It will be evaluated automatically.
- Sign in the corresponding signature field.

Automaten und formale Sprachen

Exam:	IN2041 / Retake	**Date:**	Tuesday 4th April, 2023
Examiner:	Prof. Javier Esparza	**Time:**	17:00 – 19:00

	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8
I								

Working instructions

- This exam consists of **18 pages** with a total of **8 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 45 credits.
- Detaching pages from the exam is prohibited.
- **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.

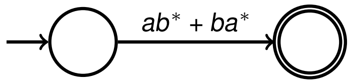
Left room from __________ to __________ / Early submission at __________

☐ Exam empty

Problem 1 NFAs and regular expressions (6 credits)

Let NFA-regtoNFA-ϵ be the algorithm given in the lectures, which given an NFA-reg M as input produces as output an NFA-ϵ which recognizes the same language as M.

a) Let A be the following NFA-reg over the alphabet $\Sigma = \{a, b\}$.



Apply the NFA-regtoNFA-ϵ algorithm on A to produce an NFA-ϵ B.

b) Consider the NFA-ϵ B from the previous subproblem. Apply any algorithm which converts an NFA-ϵ to an NFA recognizing the same language (for example, the NFA-ϵtoNFA algorithm given in the lectures), on the NFA-ϵ B, to produce an NFA C.

0
1
2

Page empty

0 1 2

c) Give a minimal NFA recognizing $\mathcal{L}(ab^* + ba^*)$. Note that you have to produce such an NFA and prove that any NFA which has strictly less states cannot recognize $\mathcal{L}(ab^* + ba^*)$.

Problem 2 Occurrences of subwords (5 credits)

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

a) Let L be the language of finite words over Σ defined as

$$L = \{w : w \text{ contains } \textbf{no} \text{ occurrence of } aba\}$$

Give the minimal DFA for the language L. **Hint:** It might help to think in terms of pattern matching.

b) Let L' be the language of finite words over Σ defined as

$$L' = \{w : w \text{ contains } \textbf{at least two distinct} \text{ (but possibly overlapping) occurrences of } aba\}$$

For example, the words $b\underbrace{aba}a\underbrace{aba}a\underbrace{aba}$, $\overbrace{ab}\underbrace{aba} \in L'$ but $ab, abaab \notin L'$.

Give a regular expression for the language L'.

0 1 2 3

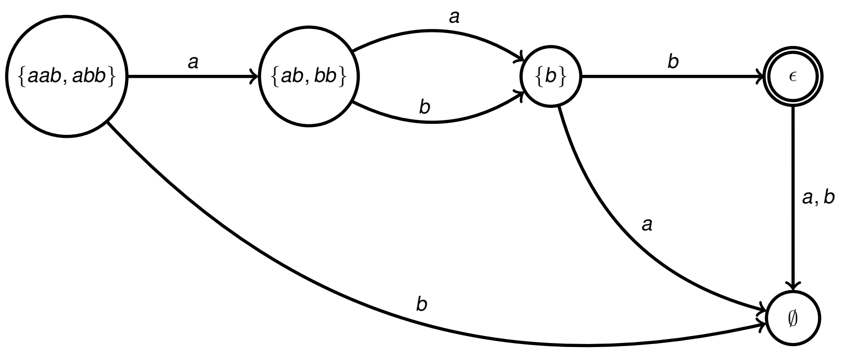
c) Give the minimal DFA for the language L' defined in the previous subproblem. **Hint:** It might help to think in terms of pattern matching. The final answer should have 7 states.

Problem 3 Fixed-length languages (4 credits)

0
1
2
3
4

For a fixed-length language L over $\Sigma = \{a, b\}$ we denote by q_L the state of the master automaton representing L. We also denote by $M(L)$ the fragment of the master automaton that contains q_L and all its residuals, that is, it contains all the states between q_L and $q_\emptyset$, **including q_L and $q_\emptyset$** (and no other states).

How many fixed-length languages L of length 3 exist such that $M(L)$ contains exactly 5 states? **For instance, here is an example of a language L of length 3 such that $M(L)$ contains exactly 5 states.**



Problem 4 **First-order logic on words (6 credits)**

Let $\Sigma = \{0, 1\}$ and let $n \geq 1$ be some natural number. Given a string $w \in \Sigma^*$, let msbf(w) denote the number represented by w in binary in the most significant bit first encoding. For example, if $w = 0011$, then msbf(w) = 3 and if $w = 1011$, then msbf(w) = 11.

For the purposes of this exercise, whenever you are asked to construct a formula over FO(Σ), in addition to the syntax of FO(Σ), you are only allowed to use the following macros: first(x), last(x), $x = y$, $y = x + k$, $y < x + k$ and $y < k$ for some number k. *If you use any other macros, you have to explicitly give the FO formulas that these macros stand for.*

0 ☐ 1 ☐

a) For $n \geq 1$, consider the language $L_n := \{w : w \in \Sigma^{2n}\}$. Give a formula ϕ_n over FO(Σ) which recognizes L_n. The formula ϕ_n must be of size polynomial in n, i.e., there must be a polynomial p such that the size of each ϕ_n is at most $p(n)$.

0 ☐ 1 ☐ 2 ☐

b) For $n \geq 1$, consider the language $L'_n := \{uu : u \in \Sigma^n\}$. Give a formula ϕ'_n over FO(Σ) which recognizes L'_n. The formula ϕ'_n must be of size polynomial in n, i.e., there must be a polynomial p' such that the size of each ϕ'_n is at most $p'(n)$.

c) For $n \geq 1$, consider the language $L''_n := \{uv : u, v \in \Sigma^n,\ \text{msbf}(u) \geq \text{msbf}(v)\}$. Give a formula ϕ''_n over FO(Σ) which recognizes L''_n. The formula ϕ''_n must be of size polynomial in n, i.e., there must be a polynomial p'' such that the size of each ϕ''_n is at most $p''(n)$.

0
1
2
3

Page empty

Problem 5 Operations on languages (6 credits)

Let $\Sigma = \{a, b\}$. Let $L \subseteq \Sigma^*$ be any language consisting of finite words over Σ. We define the ω-language $La^\omega \subseteq \Sigma^\omega$ as

$$La^\omega = \{wa^\omega : w \in L\}$$

Note that La^ω is a language of *infinite words* over Σ. Intuitively, each word in La^ω is obtained by first taking some finite word $w \in L$ and then adding the infinite suffix a^ω to it.

0 1

a) Prove or disprove: If $L \subseteq \Sigma^*$ is regular, then La^ω is ω-regular.

0 1 2

b) Prove or disprove: If La^ω is ω-regular for some $L \subseteq \Sigma^*$, then L is regular.

c) Prove or disprove: If $(L \cdot \{b\})a^\omega$ is ω-regular for some $L \subseteq \Sigma^*$, then L is regular.

0
1
2
3

☐ Page empty

Problem 6 Acceptance conditions (10 credits)

Throughout this exercise, we will only be considering languages of infinite words over $\Sigma = \{a, b, c\}$.

0 1 2

a) Consider the ω-regular language L_1 defined as

$$L_1 = \{w \in \Sigma^\omega : ab \text{ and } ac \text{ appear infinitely often in } w\}$$

Give a non-deterministic Büchi automaton $(A_1, \mathcal{F}_1)$ which accepts L_1 such that A_1 has **at most 5 states**.

0 1 2

b) Give a non-deterministic generalized Büchi automaton $(A_1', \mathcal{F}_1')$ which accepts L_1 such that A_1' has **at most 3 states**.

c) Consider the ω-regular language L_2 defined as

$$L_2 = \{w \in \Sigma^\omega : ab \text{ appears infinitely often in } w \text{ and } ac \text{ appears } \textbf{finitely} \text{ often in } w\}$$

Give a **deterministic** Rabin automaton $(A_2, \mathcal{F}_2)$ which accepts L_2 such that A_2 has **at most 4 states**.

0
1
2
3

d) **This is a bonus subproblem.**

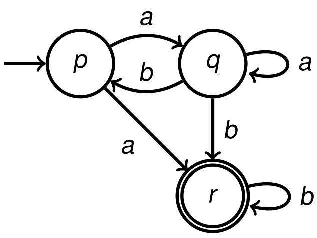
Give a non-deterministic Muller automaton $(A_2', \mathcal{F}_2')$ which accepts L_2 such that A_2' has **at most 3 states**.

0
1
2
3

Page empty

Problem 7 DAGs and Büchi automata (4 credits)

Consider the following Büchi automaton over $\Sigma = \{a, b\}$.



0 1 2

a) Draw $dag((aab)^\omega)$ and give an odd ranking for it.

b) Find an ω-word *w* such that *dag*(*w*) **does not** have an odd ranking. Draw *dag*(*w*) and prove that it does not have an odd ranking by analyzing the *dag*.

0

1

2

Page empty

Problem 8 Linear Temporal Logic (4 credits)

0
1
2

a) Let $AP = \{p, q, r\}$ and let $\Sigma = 2^{AP}$. Consider the formulas

$$\phi := (p \mathbf{U} q) \mathbf{U} r \quad \text{and} \quad \xi := p \mathbf{U} (q \mathbf{U} r)$$

Give four computations $\sigma_1, \sigma_2, \sigma_3, \sigma_4$, all of them over AP, such that

- $\sigma_1 \models \phi$ and $\sigma_1 \models \xi$
- $\sigma_2 \models \phi$ and $\sigma_2 \not\models \xi$
- $\sigma_3 \not\models \phi$ and $\sigma_3 \models \xi$
- $\sigma_4 \not\models \phi$ and $\sigma_4 \not\models \xi$

0
1

b) **This is a bonus subproblem.**

Let $AP = \{p, q\}$ and let $\Sigma = 2^{AP}$. Give an ω-regular expression over Σ for the set of all computations which satisfy the formula

$$\varphi := (p \mathbf{U} q) \mathbf{U} p$$

c) **This is a bonus subproblem.**

0

1

Consider the formula φ defined in the previous subproblem. Use the ω-regular expression you defined in the previous subproblem to derive a formula φ' such that φ' and φ are equivalent and φ' is of strictly smaller size than φ.

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