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

Registration number

1	☐	☐	☐	☐	☐	☐	☐	☐
2	☐	☐	☐	☐	☐	☐	☐	☐
3	☐	☐	☐	☐	☐	☐	☐	☐
4	☐	☐	☐	☐	☐	☐	☐	☐
5	☐	☐	☐	☐	☐	☐	☐	☐
6	☐	☐	☐	☐	☐	☐	☐	☐
7	☐	☐	☐	☐	☐	☐	☐	☐
8	☐	☐	☐	☐	☐	☐	☐	☐
9	☐	☐	☐	☐	☐	☐	☐	☐
0	☒	☐	☐	☐	☐	☐	☐	☐

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 / Endterm	**Date:**	Tuesday 14th February, 2023
Examiner:	Prof. Javier Esparza	**Time:**	14:15 – 15:45

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

Working instructions

- This exam consists of **16 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 Constructing NFAs (4 credits)

Give two **NFA**s, each of them with at most 25 states, that accept the following languages L_1 and L_2. Note that you do not need to explicitly draw all the states, rather describe the structure of the automata precisely.

There are solutions with 12 states for L_1 and 11 states for L_2. However, full points will be awarded for any correct solution with at most 25 states.

0
1
2

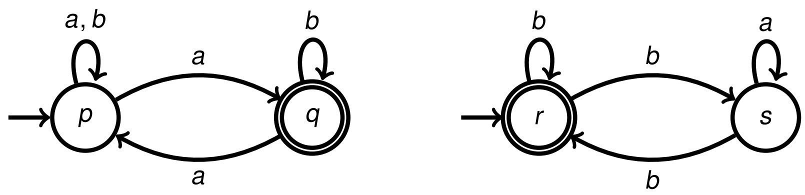
a) L_1 is the set of strings over the alphabet $\{0, 1, \dots, 9\}$ such that the final digit has appeared before. For example, $4610216, 33, 2869322, 19401, 12343234 \in L_1$ and $25692553, 2, 14653345609, \epsilon \notin L_1$.

0
1
2

b) L_2 is the set of strings over the alphabet $\{0, 1, \dots, 9\}$ such that the final digit has *not* appeared before. For example, $46653, 5489890, 2 \in L_2$ and $5788, 578045, \epsilon \notin L_2$.

Problem 2 Automata and regular expressions (6 credits)

Consider the following two NFAs A_1 and A_2.



a) Use the *IntersNFA* algorithm, as described in the lectures, to construct an NFA A which recognizes $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$. **Label the states of A as 1, 2, 3 and 4, in the order by which the algorithm *IntersNFA* constructs them.**

0
1
2

b) Use the *NFAtoRE* algorithm, as described in the lectures, to convert the NFA A constructed in the previous subproblem, into a regular expression. **The solution must contain the automaton after the preprocessing step and also the automata obtained after removing each state.** Further,

0
1
2
3
4

- **You must remove states in ascending order**, i.e., you must first remove the state 1, then state 2 and so on.

- While writing the solution, if you come across a long regular expression, you can abbreviate it by a variable and use this abbreviation. **For example**, you can let σ stand for the regular expression $(b^*a + a^*b)^*$ and then instead of writing $(b^*a + a^*b)^*$ throughout the solution, you can instead use σ.

Problem 3 Operations on languages (4 credits)

Let L be a language. The operation $T(L)$ is defined as follows:

- Remove every even-length word from L.
- For each odd-length word, remove the middle letter.

For example, if $L = \{aab, bbaa, babab\}$, then $T(L) = \{ab, baab\}$. The even-length word *bbaa* is deleted, the middle letter of *aab* is removed to make *ab*, and the middle letter of *babab* is removed to make *baab*.

For each of the following languages L_i, $i = 1, 2$, determine if $T(L_i)$ is regular or not. If it is regular give a regular expression representing it; if it is not regular prove this by analyzing the set of its residuals.

0 1 2

a) L_1 is the language of the regular expression $(bab)^*$.

0 1 2

b) L_2 is the language of the regular expression aa^*bb^*.

Problem 4 Fixed-length languages (7 credits)

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).

a) Give a fixed-length language L over $\Sigma = \{a, b\}$ of length 4, such that $M(L)$ has exactly 10 states. Enumerate the words of L and draw $M(L)$.

0
1
2

0
1
2
3

b) What is the maximal possible number of states in $M(L)$, if L is a fixed-length language over $\Sigma = \{a, b\}$ of length 4? Justify your answer.

c) What is the minimal possible number of states in $M(L)$, if L is a fixed-length language over $\Sigma = \{a, b\}$ of length 4? Justify your answer.

0
1
2

Problem 5 Automata and logic (5 credits)

0 1 2 3

a) Let $\phi(x, y) = (x + y \geq 2) \wedge \exists z(y + z \leq 1)$, where x, y, z range over natural numbers (including 0). Give an NFA A_φ over $\{0, 1\}^2$ which recognizes the set of all least significant bit first (lsbf) encodings of pairs which satisfy the formula ϕ. You can use any method to compute A_φ.

0 1 2

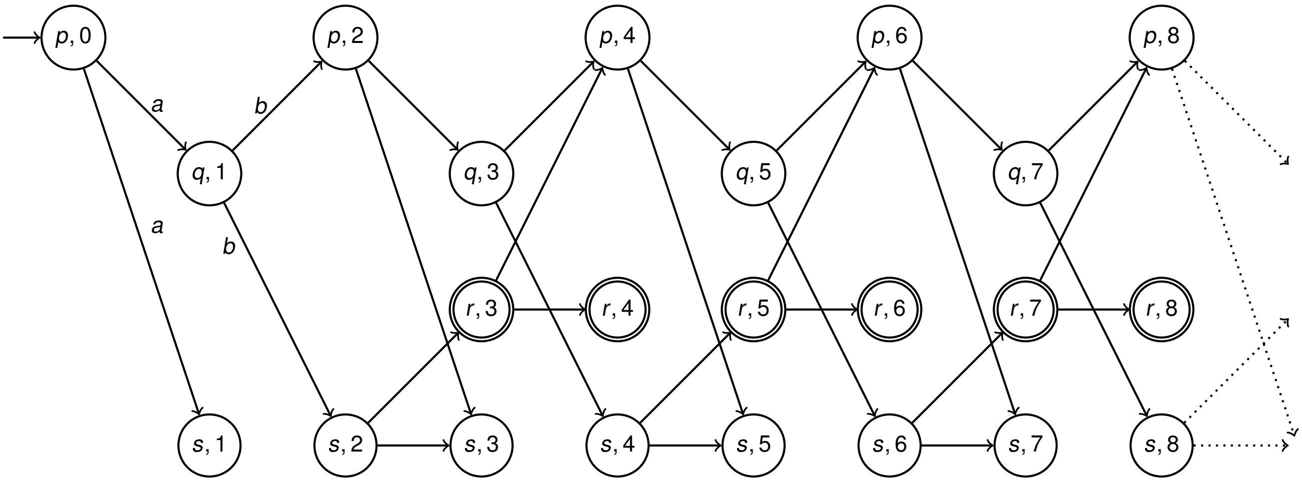
b) Give a regular expression and an NFA recognizing the language of the following MSO formula:

$$\exists X, y.\ Q_b(y) \wedge \forall x(x \in X \Rightarrow Q_a(x)) \wedge \forall x(x = y \vee x \in X) \wedge \text{evensize}(X)$$

Here evensize(X) is the formula which is true for a set X if and only if X contains an even number of elements.

Problem 6 DAGs and Büchi automata (7 credits)

The following is a fragment of dag($(ab)^\omega$) on a Büchi automaton with four states $\{p, q, r, s\}$. We omit most of the edge labels to increase readability.



a) Give a Büchi automaton B with four states $\{p, q, r, s\}$ over $\Sigma = \{a, b\}$ such that dag($(ab)^\omega$) on B matches the graph from above.

0
1
2

0
1
2
3
4

b) Prove that every infinite path of dag($(ab)^\omega$) has only even stable rankings.

0
1

c) Does dag($(ab)^\omega$) have an odd ranking? Justify your answer.

Problem 7 LTL tautologies (7 credits)

Recall: An LTL formula is a tautology if it is satisfied by all computations. For each of the following formulas over AP = $\{p, q, r\}$, check if it is a tautology. If it is a tautology, give a formal proof. Otherwise, give a computation that does not satisfy the formula.

a) $\mathbf{G}(p \mathbin{\mathbf{U}} \neg p) \Leftrightarrow \mathbf{G}(\neg p \mathbin{\mathbf{U}} p)$

0 1 2

b) $((\mathbf{G}\neg p) \mathbin{\mathbf{U}} p) \Leftrightarrow p$

0 1 2 3

c) $\big(((\mathbf{G}\neg p) \mathbin{\mathbf{U}} p) \land \neg p\big) \Rightarrow \big((\mathbf{G}q) \lor (\neg q \mathbin{\mathbf{U}} r)\big)$

0 1 2

Page empty

Problem 8 ω-regular languages (5 credits)

This is a bonus question.

0
1
2
3
4
5

Let $S = (Q, \Sigma, \delta, Q_0)$ be a semi-automaton. Let $\mathcal{F} = \{F_1, \ldots, F_k\}$ where each $F_i \subseteq Q$. Let $M_{S,\mathcal{F}}$ be the set of **infinite words** w such that there is a run ρ of w on S and a set $F_i \in \mathcal{F}$ with the property that the set of states appearing in the run ρ is **exactly** F_i. Prove that $M_{S,\mathcal{F}}$ is ω-regular.

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

☐ Page empty