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 / Endterm	**Date:**	Wednesday 28th February, 2024
Examiner:	Prof. Javier Esparza	**Time:**	10:30 – 12:30

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

Working instructions

- This exam consists of **16 pages** with a total of **7 problems** including one **bonus problem**.
  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 problem 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 (22 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 $L \subseteq \Sigma^*$. True or false: if $L = L^R$, then L is regular.

Note: For a word $w = a_1...a_n$, where $a_1, ..., a_n \in \Sigma$, we write $w^R := a_n...a_1$ for the *reverse* of w, and define $L^R := \{w^R : w \in L\}$.

☐ True ☐ False

0 1 2 3 4

b)* True or false: every NFA M accepting the language $L := \{a, ab\}^{2024}$ has at least 4048 states.

☐ True ☐ False

0 1 2

c)* Construct a DFA with at most 5 states recognising the language of $(a + bab)^*$.

0 1 2

d)* Let $L \subseteq \Sigma^*$ denote the language of words where no b is preceded by an odd-length sequence of a, i.e. L is generated by the RE $((aa)^*b)^*a^*$. For example, $\varepsilon, aabbba, baab \in L$, and $aabab \notin L$.
Construct an NFA N with $L(N) = L$ and at most 5 states, such that all states of N are accepting.

e)* 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 every occurrence of $\{p\}\{p\}$ is followed by some later occurrence of $\emptyset\ \emptyset$.

0
1
2

f)* Let L denote an ω-regular (!) language. True or false: if $L = \Sigma L$, then $L = \emptyset$ or $L = \Sigma^\omega$.

0
1
2
3
4

☐ True ☐ False

g)* Let $L_1 \subseteq \Sigma^\omega$ denote the language of ω-words w such that the first letter of w appears infinitely often in w. Give a Büchi automaton accepting L_1 with at most 5 states.

0
1
2

h)* Consider the language L_1 defined in exercise g). True or false: every Büchi automaton A for L_1 has at least two **accepting** states (and any number of non-accepting states).

Note: If your automaton in exercise g) already accepts L_1 with only one accepting state, you are automatically awarded the points for this exercise.

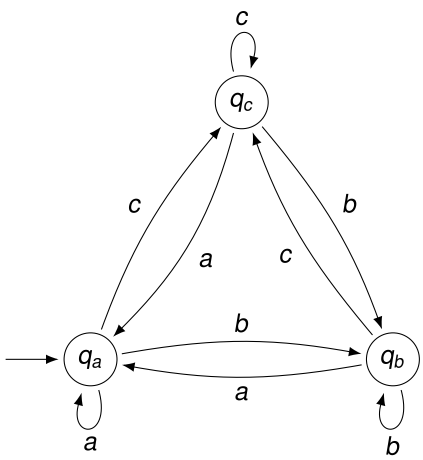
0
1
2
3
4

☐ True ☐ False

Problem 2 **Omega-Automata (8 credits)**

0
1
2
3
4

a)* Consider the following semi-automaton with $\Sigma = \{a, b, c\}$:



For each one of the languages L_1, L_2, and L_3 below, decide whether there exists a Rabin condition with only one Rabin pair such that the above semi-automaton recognizes the language. If it does, give such a condition in the format $\langle F, G\rangle$. If it does not, justify why no such condition exists.
Reminder: A run satisfies the Rabin condition $\langle F, G\rangle$ if it visits the set F of states infinitely often and the set of states G finitely often.

$L_1 = \{w \in \Sigma^\omega \mid w \text{ contains infinitely many } a \text{ and finitely many } b\}$

☐ Exists ☐ Does not exist

$L_2 = \{w \in \Sigma^\omega \mid w \text{ contains finitely many } b \text{ and finitely many } c\}$

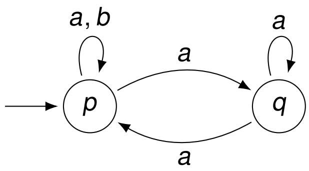
☐ Exists ☐ Does not exist

$L_3 = \{w \in \Sigma^\omega \mid w \text{ contains infinitely many } a, \text{ infinitely many } b \text{ and infinitely many } c\}$

☐ Exists ☐ Does not exist

b)* Consider the following non-deterministic semi-automaton with $\Sigma = \{a, b\}$:

0
1
2
3
4



Which of the following languages can be recognized by this semi-automaton by using an appropriate Büchi-condition? If such a condition exists, give it. If it does not, justify why no such condition exists.

$L_1 = \{w \in \Sigma^\omega \mid w \text{ contains infinitely many } a\}$

☐ Exists ☐ Does not exist

$L_2 = \{w \in \Sigma^\omega \mid w \text{ contains finitely many } b\}$

☐ Exists ☐ Does not exist

$L_3 = \{w \in \Sigma^\omega \mid w \text{ contains } aa \text{ infinitely often}\}$

☐ Exists ☐ Does not exist

Problem 3 FO/MSO on words (10 credits)

In the following questions we consider MSO(Σ), monadic second order logic on words, over the alphabet $\Sigma = \{a, b\}$. The first position in a word w has index 1. You are allowed to use the macros *first* and $x = y + 1, x = y + 2, ...$ from the lecture. You are not allowed to use any other macros.

0 1 2 3

a)* Give a formula *EveryThirdPosition*(X) which is true if and only if X contains exactly every third position, starting from the first position. For example, if the word has length 5, then $X = \{1, 4\}$ is the only model.

0 1 2 3

b)* Using the formula *EveryThirdPosition*(X) as a macro, give a formula *RepeatingABB*, which is true for w if and only if w is a finite prefix of $(abb)^\omega$, i.e. if and only if $w \in (abb)^*(\varepsilon \mid a \mid ab)$.

Define for every $k \in \mathbb{N}_{>0}$ the macros

$$PositionsGreaterK(X) \;:=\; \forall x\Big((x \in X) \leftrightarrow \big(\exists z_1 \exists z_2 ... \exists z_k \bigwedge_{i=1}^{k-1} (z_i < z_{i+1}) \wedge z_k < x\big)\Big)$$

$$\varphi_k(x) \;:=\; \exists X(PositionsGreaterK(X) \wedge x \in X)$$

0 1 2

c)* True or false: For every $k \in \mathbb{N}_{>0}$ there exists a formula in FO(Σ) (first-order logic on words) equivalent to the formula φ_k.

If yes, give a family of such formulas. If not, justify your answer.

☐ Yes ☐ No

0 1 2

d)* Given $k \in \mathbb{N}_{>0}$, define $\psi_k := \forall x(\varphi_k(x) \to Q_b(x))$. True or false: there is a formula of MSO(Σ) such that for every word $w \in \Sigma^*$, w satisfies the formula iff w satisfies ψ_k for at least one $k \in \mathbb{N}_{>0}$?

If yes, give an example of such a formula. If not, justify your answer.

☐ Yes ☐ No

Problem 4 LTL (10 credits)

a)* Consider the set of atomic propositions AP = $\{p, q\}$ and the LTL formula $\varphi := p\ \mathrm{U}(q\ \mathrm{U}\ \mathrm{G}(q \wedge p))$. Give an ω-regular expression s over the alphabet $\{\emptyset, \{p\}, \{q\}, \{p, q\}\}$ such that the languages of s and φ are equal. You do not need to justify your answer.

0 1 2 3

b)* Which of the following formulas are equivalent to the formula φ from the previous question?

- [ ] $(p \wedge q)\ \mathrm{U}\ \mathrm{G}(q \wedge p)$
- [ ] $p\ \mathrm{U}(\mathrm{G}q \wedge \mathrm{FG}p)$
- [ ] $p\ \mathrm{U}(\mathrm{FG}(q \wedge p))$

Give a counter-example for the ones that are not equivalent. You do not need to provide an explanation for the ones that are equivalent.

0 1 2 3 4

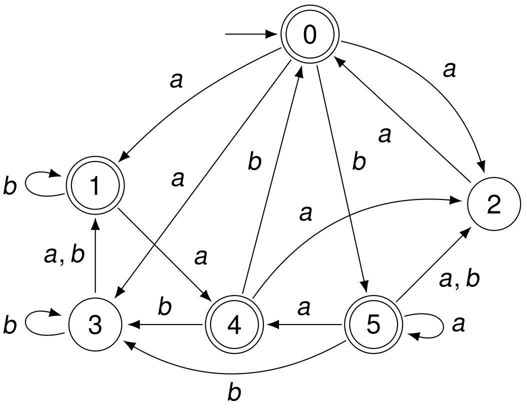
c)* Are the formulas $\psi_1 := p\ \mathrm{U}\ (\mathrm{F}q)$ and $\psi_2 :=$ **true** $\mathrm{U}\ (\mathrm{F}q)$ equivalent?
If yes, prove that they are equivalent. If no, provide a suitable computation as counterexample.

- [ ] Equivalent
- [ ] Not equivalent

0 1 2 3

Problem 5 NFA Universality (8 credits)

a)* Consider the following NFA N:



Decide whether N is universal, i.e. whether $L(N) = \{a, b\}^*$, using the algorithm UnivNFA from the lecture with the subsumption check. (If you execute the algorithm without the subsumption check, you can still receive up to 3 points.)

Whenever choosing which item to remove from the workset $\mathcal{W}$, choose the item that has been in the workset the longest. Consider always letter a before b.

While executing the algorithm, fill out the table below. Enter each item removed from the workset in the first column. Use the other two columns to enter the items (if any) that are added to the workset in this iteration. (If you wish, you can also note items that you considered adding to the workset, but did not add, e.g. because they failed the subsumption test. Mark these items with $\times$.)

☐ Universal ☐ Not universal

removed from $\mathcal{W}$	added to $\mathcal{W}$ (a)	added to $\mathcal{W}$ (b)

b) If N is universal, give a transition such that N is no longer universal when that transition is deleted. If N is not universal, give a word $w \in \{a, b\}^* \setminus L(N)$.

Problem 6 **The infinite master automaton (12 credits)**

We consider the master automaton M for fixed-length languages, as defined in the lecture: $M := (Q, \Sigma, \delta, F)$, where Q is the set of all fixed-length languages, $\Sigma := \{a, b\}$, $\delta(L, x) := L^x$ for $L \in Q, x \in \Sigma$ and $F := \{\{\varepsilon\}\}$. (Recall that M has infinitely many states.)

a)* Prove that there is no infinite, simple, directed path in M, i.e. an infinite sequence of pairwise distinct $L_1, L_2, ... \in Q$ such that for each i there is an $x \in \Sigma$ with $L_{i+1} = \delta(L_i, x)$.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

b)* Prove or disprove: the nonempty languages of M are connected when viewed as undirected graph. More precisely, show that for every pair of nonempty, fixed-length languages $R, S \in Q \setminus \{\emptyset\}$ there is a finite sequence $L_1, ..., L_k \in Q \setminus \{\emptyset\}$ with $L_1 = R$, $L_k = S$ and for each $i \in \{1, ..., k-1\}$ there is an $x \in \Sigma$ such that either $L_{i+1} = \delta(L_i, x)$ or $L_i = \delta(L_{i+1}, x)$.
Note: Such a sequence $L_1, ..., L_k$ is a *path connecting R and S*.

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4

☐ True ☐ False

0 1 2 3 4

c)* We now consider a variant of M, which we refer to as M'. In M' we set $Q := 2^{\Sigma^*}$ (that is, to the set of all languages over Σ, fixed-length or not and regular or not), define δ as for M, and define F as the set of all languages containing ε.
Is the statement of part a) still true in M'? More precisely, prove or disprove: there is an infinite, simple, directed path in M'.

☐ True ☐ False

Problem 7 **Bonus question: Fully connected (10 credits)**

Let $R \subseteq \mathbb{N} \times \mathbb{N}$ be a relation on natural numbers. We say that a number n is *fully connected* w.r.t. R if $(n, m) \in R$ for every $m \in \mathbb{N}$. We let FC_R denote the set of all fully connected numbers w.r.t. R.

a)* Give an algorithm satisfying the following specification:

0
1
2
3
4
5
6

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

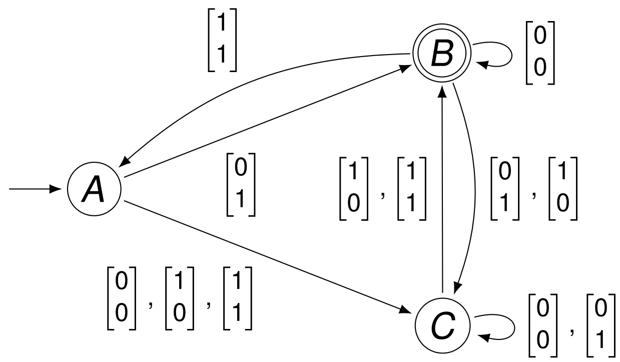
  Note: Recall that a transducer recognizes R if for every pair $(n, m) \in R$ it accepts every encoding of (n, m), and for every pair $(n, m) \notin R$ it accepts no encoding of (n, m).

- Output: a DFA recognizing the set FC_R.

Hint: $n \in FC_R$ iff $\forall m \in \mathbb{N} \colon (n, m) \in R$ iff $\neg \exists m \in \mathbb{N} \colon (n, m) \notin R$.

0
1
2
3
4

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



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