Technische Universität München
Winter term 2019/20, 26.02.2020
Prof. J. Křetínský / M. Lazić / S. Sickert-Zehnter / C. Weil-Kennedy

Automata and Formal Languages — Endterm Exam

- You have 120 minutes to complete the exam.
- Answers must be written in a separate booklet. Do not answer on the exam.
- Please let us know if you need more paper.
- Write your name and Matrikelnummer on every sheet.
- Write with a non-erasable pen. Do not use red or green.
- You are not allowed to use auxiliary means other than pen and paper.
- You can obtain 40 points. You need 17 points to pass.
- The ★ symbol indicates a more challenging question.

Question 1 (11 points)

a. Let Σ be the alphabet $\{a, b\}$, and let p be the word pattern $ababaa$. Build the DFA B_p (obtained by determinizing the naive NFA A_p for $\Sigma^* p$).

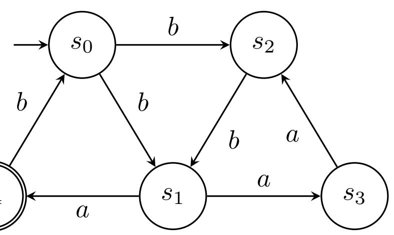
b. Give the fragment of the master automaton that contains the states of the language $L = \{aab, bbb, bab\}$ and all its residuals (all the states between q_L and $q_\emptyset$).

c. Given a word w over the alphabet $\Sigma = \{a, b\}$ we define $\overline{w}$ to be the word obtained from w by replacing a by b, and b by a. For example, $\overline{aababba} = bbabaab$ and $\overline{babb} = abaa$. Decide whether the language $L = \{w\overline{w} : w \in \Sigma^*\}$ is regular or irregular, and prove this by analyzing its residuals.

d. Give a regular expression recognizing the language of the following the MSO formula

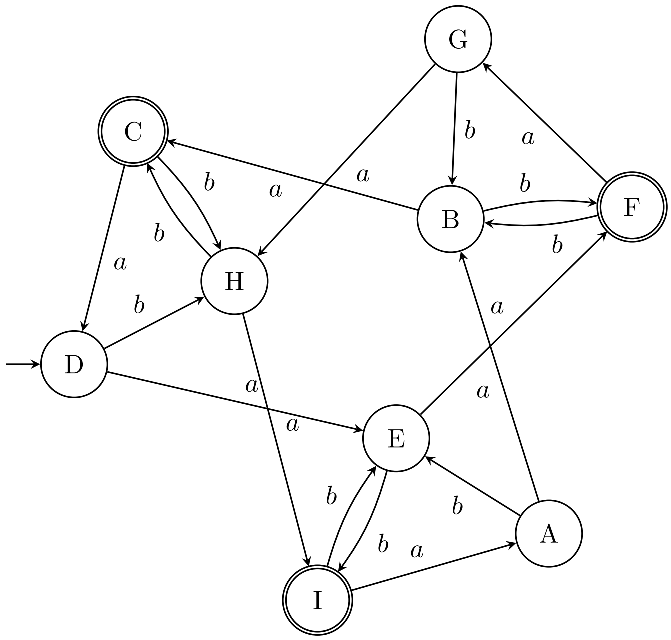
$$\varphi = \exists x \exists y.\, x \neq y \wedge Q_a(x) \wedge Q_a(y) \wedge [\forall z.(z \neq x \wedge z \neq y) \rightarrow (Q_b(z) \wedge x < z \wedge z < y)].$$

e. Consider the following NBA.



Draw $dag(b(baa)^\omega)$. Does it admit an odd ranking? Give such a ranking if it exists, and provide a short justification if it does not.

Question 2 (3 points)
Let $\mathcal{A}$ be the following DFA:

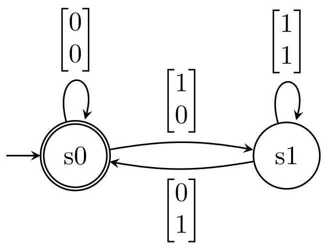


(a) Compute the language partitions of $\mathcal{A}$.

(b) Draw the minimal automaton using the language partitions from (a).

Question 3 (6 points)
Given $n \in \mathbb{N}$, let $msbf(n)$ be the set of *most significant bit first* encodings of n, i.e., the words that start with an arbitrary number of leading zeros, followed by n written in binary. For example, $msbf(6) = 0^*110$ and $msbf(3) = 0^*11$. Let $\text{val} : \{0,1\}^* \to \mathbb{N}$ be the function that associates to every word $w \in \{0,1\}^*$ the number $\text{val}(w)$ represented by w in the *most significant bit first* encoding. For example, $\text{val}(110) = 6$ and $\text{val}(011) = 3$.

a. Let T be the following transducer over alphabet $\Sigma = \{0,1\} \times \{0,1\}$.



What is the relation between $\text{val}(x)$ and $\text{val}(y)$, for any $[x, y]$ accepted by T?

b. Draw a transducer T_{+1} recognizing the language

$$\{[x, y] \in \Sigma^* \mid \text{val}(y) = \text{val}(x) + 1\}.$$

Question 4 (5 points)

Recall: A process can send a message m to the channel with the instruction $c\,!\,m$. A process can also consume the first message of the channel with the instruction $c\,?\,m$. If the channel is full when executing $c\,!\,m$, then the process blocks and waits until it can send m. When a process executes $c\,?\,m$, it blocks and waits until the first message of the channel becomes m.

Suppose there are two processes being executed concurrently that communicate through a channel c. Channel c is a queue that can store up to 1 message. The two processes follow these two algorithms respectively:

```
process(1):
    while true do
        c ! m
        /* critical section */
        c ? m
```

```
process(2):
    while true do
        c ? m
        c ? m
        /* critical section */
        c ! m
```

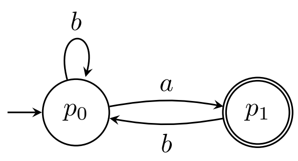
a. Model the program by constructing a network of three automata:
   - One for process 1, using the alphabet $\Sigma_1 = \{c?m, c!m, cs_1\}$,
   - One for process 2, using the alphabet $\Sigma_2 = \{\overline{c?m}, \overline{c!m}, cs_2\}$,
   - One for the channel c of size 1, that is initially empty, using the alphabet $\Sigma_c = \{c?m, c!m, \overline{c?m}, \overline{c!m}\}$.

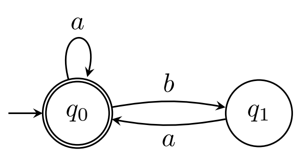
b. Construct the asynchronous product $\mathcal{P}$ of the three automata obtained in (a). The alphabet of the automaton $\mathcal{P}$ should be $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_c$.

c. Consider the state of the asynchronous product $\mathcal{P}$ where both processes are in the critical section. Is this state reachable? Give a short justification based on automaton $\mathcal{P}$.

Question 5 (3 points)

Consider the following DBAs B_1 and B_2:

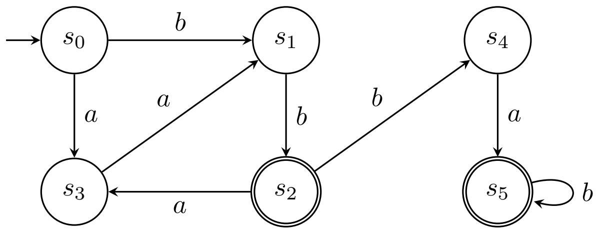




a. Give ω-regular expressions recognizing the languages of B_1 and B_2.

b. Give the DBA $B_1 \cap B_2$ using the algorithm seen in class. Give an ω-regular expression for $B_1 \cap B_2$.

Question 6 (3 points)
Let B be the following Büchi automaton.



a. For every state of B, give the discovery time and finishing time assigned by a DFS on B starting in s_0 (i.e. the moment they first become grey and the moment they become black). Visit successors s_i of a given state in the ascending order of their indices i. For example, when visiting the successors of s_2, first visit s_3 and later s_4.

b. The language of B is not empty. Give the witness lasso (as a sequence of states) found by applying *TwoStack* to B following the same convention for the order of successors as above.

Question 7 (6 points)
Let $\mathrm{AP} = \{p, q\}$ and let $\Sigma = 2^{\mathrm{AP}}$. Recall: An LTL formula is a tautology if it is satisfied by all computations.

a. Is the following formula a tautology: $(\mathbf{GF}p \wedge \mathbf{GF}q) \Rightarrow \mathbf{G}(p \;\mathbf{U}\; q)$? Provide a formal proof if it is and a counter-example if it is not.

b. Is the following formula a tautology: $\mathbf{G}(p \;\mathbf{U}\; q) \Rightarrow (\mathbf{GF}p \vee \mathbf{GF}q)$? Provide a formal proof if it is and a counter-example if it is not.

c. Give a Büchi automaton for the ω-language over Σ defined by the following LTL formula: $\mathbf{G}(p \;\mathbf{U}\; q)$.

Question 8 (3 points)
★ Given a language L we define the language $\mathsf{Cycle}(L) = \{vu \mid uv \in L\}$. For example, if $L = \{ab, abcd\}$ then $\mathsf{Cycle}(L) = \{ab, ba, abcd, bcda, cdab, dabc\}$; in particular, $acbd$ is not in $\mathsf{Cycle}(L)$ as it cannot be written as vu such that $uv \in L$.

Find a language L such that L is not regular and $\mathsf{Cycle}(L)$ is regular. Give proofs for both statements.