Technische Universität München
Winter term 2016/17
Prof. J. Esparza / Dr. M. Blondin

Automata and Formal Languages — Endterm

- 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.

Question 1 (6 points)

(a) Give the minimal DFA for the language $\{w \in \{a,b\}^* \mid w \text{ does not contain } bba\}$.

(b) Prove or disprove: if L_1 and $L_1 \cdot L_2$ are regular languages, then L_2 is regular. (To prove it, show how to construct an NFA for L_2 from NFAs for L_1 and $L_1 \cdot L_2$. To disprove it, give languages L_1 and L_2 such that L_1 and $L_1 \cdot L_2$ are regular, and L_2 is not regular.)

(c) Give a transducer over alphabet $\{0,1\} \times \{0,1\}$ accepting least significant bit first (lsbf) encodings of pairs $(x,y) \in \mathbb{N} \times \mathbb{N}$ such that $x > 0$ and $y = x - 1$.

(d) Let $\varphi = \mathbf{GF}(\neg p) \ \vee \ \mathbf{FG}p \ \vee \ (\neg p \ \mathbf{U} \ (\mathbf{XX}p))$ be an LTL formula over the set of atomic propositions $AP = \{p\}$. Give an ω-regular expression s such that $L_\omega(\varphi) = L_\omega(s)$.

(e) Give an ω-word satisfying $((\mathbf{X}p) \ \mathbf{U} \ q) \ \wedge \ ((\mathbf{X}\neg p) \ \mathbf{U} \ (\neg q))$.

(f) Give a Muller automaton over the alphabet $\Sigma = \{a,b,c\}$, with at most 3 states, recognizing the language of all ω-words containing infinitely many a's, infinitely many b's and infinitely many c's.

Question 2 ($2+2+2 = 6$ points)

Decide whether the following languages over alphabet $\{a,b\}$ are regular or not. If a language is regular, give a corresponding automaton or regular expression. If a language is not regular, prove that it has infinitely many residuals.

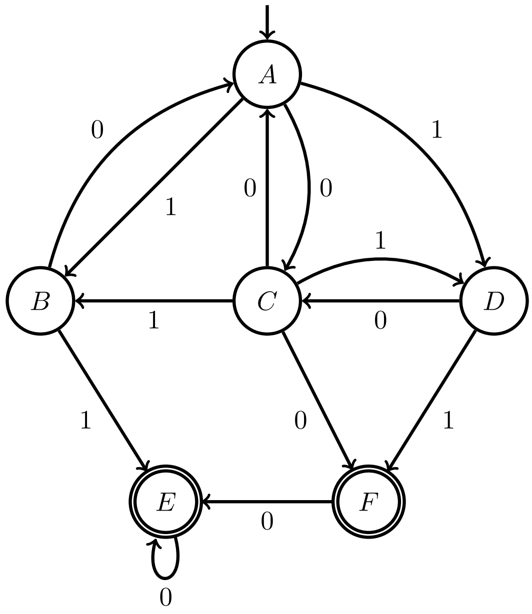
(a) $L_1 = \{w \in \{a,b\}^* \mid \ w \text{ contains as many } a\text{'s as } b\text{'s}\}$.

(b) $L_2 = \{a^m b^n \mid m \geq n\}$.

(c) $L_3 = \{a^m b^n \mid m < 1000 \wedge m \leq n\}$.

Question 3 ($1+2+2=5$ points)

Reduce the size of the following NFA by means of the partition refinement algorithm introduced in the course.



(a) Give the blocks of the initial partition.

(b) Describe for each step which block is being refined, which is the splitter, and which are the two blocks resulting from the split.

(c) Draw the quotient of the original automaton with respect to the partition computed in (b).

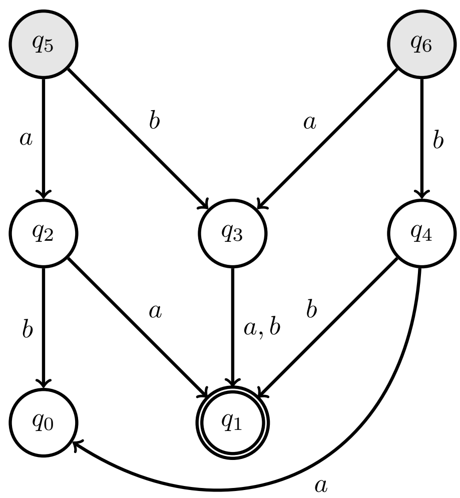
Question 4 (3 + 3 = 6 points)

Recall that, given a state q of the master automaton for fixed-length languages over an alphabet Σ, and given $a \in \Sigma$, we denote q^a the unique a-successor of q. Further, we denote by q_ϵ and $q_\emptyset$ the states of the master automaton recognizing $\{\epsilon\}$ and $\emptyset$, respectively.

(a) Fill the blanks in the algorithm for computing the state of the master automaton recognizing the intersection of the languages recognized by two given states. The alphabet is $\Sigma = \{a_1, a_2, \dots, a_m\}$.

Input: states q_1, q_2 with same length.
Output: state recognizing $L(q_1) \cap L(q_2)$.
`inter`(q_1, q_2) :
  if $G(q_1, q_2)$ is not empty **then return** $G(q_1, q_2)$
  else if [Blank 1] **then return** $q_\emptyset$
  else if [Blank 2] **then return** q_ε
  else
    for $i = 1, \dots, m$ **do**
      $r_i \leftarrow$ [Blank 3]
    $G(q_1, q_2) \leftarrow$ `make`$(r_1, \dots, r_m)$
    return $G(q_1, q_2)$

(b) Apply the algorithm to compute the state recognizing $L(q_5) \cap L(q_6)$ in the fragment of the master automaton shown below. Describe the tree of recursive calls, give the result of each call, and draw the resulting automaton. If you use optimizations described in the course, then describe them briefly.



Question 5 (2 + 2 + 2 = 6 points)

Let $L = \{w \in \{a, b\}^* \mid \text{ between any two } b\text{'s of } w \text{ there is at least one } a\}$.

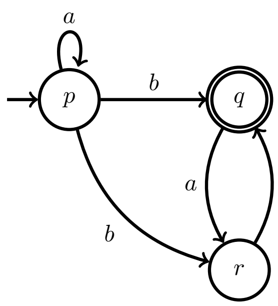
(a) Give an MSO formula φ such that $L(\varphi) = L$.

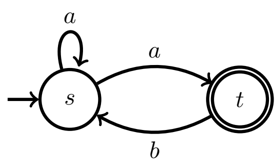
(b) Give the minimal DFA for L.

(c) Give a regular expression for L.

Question 6 (5 points)

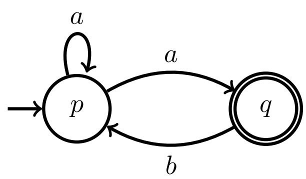
Let A, B be the following Büchi automata over $\Sigma = \{a, b\}$. Construct a Büchi automaton C such that $L_\omega(C) = L_\omega(A) \cap L_\omega(B)$ using the algorithm presented in the lecture notes.





Question 7 $(2+2+2 = 6$ points)

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



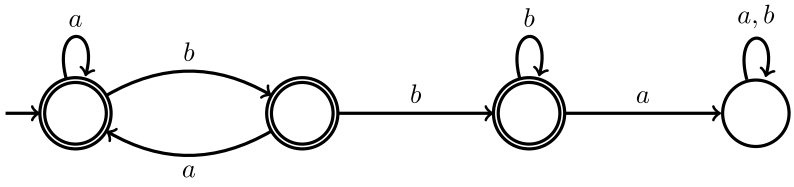
(a) Sketch $dag(aba^\omega)$ and $dag(a(ab)^\omega)$. You must build at least 5 levels.

(b) Explain the conditions that an odd ranking must satisfy, and give an odd ranking of $dag(aba^\omega)$.

(c) Using the definition of odd rankings, prove that $dag(a(ab)^\omega)$ does not have any odd ranking.

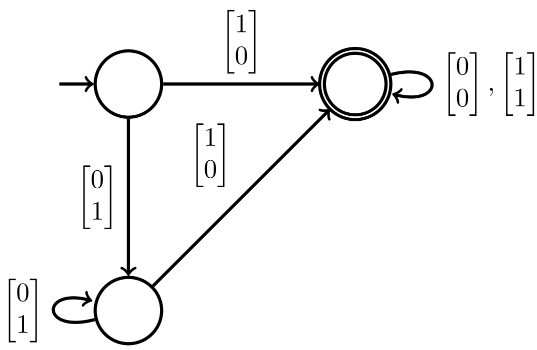
Solution 1 (6 points)

(a)



(b) False. Let $L_1 = \{a, b\}^*$ and $L_2 = \{a^n b^n : n \in \mathbb{N}\}$. Since $\varepsilon \in L_2$, we have $L_1 \cdot L_2 = \{a, b\}^*$. Thus, L_1 and $L_1 \cdot L_2$ are regular, but L_2 is not. $\square$

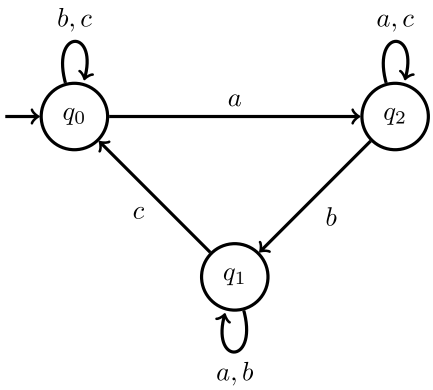
(c)



(d) $(\{p\}^*\emptyset\emptyset^*)^\omega + \Sigma^*\{p\}^\omega + \emptyset^*\Sigma\Sigma\{p\}\Sigma^\omega$ or simply Σ^* since the formula is a tautology.

(e) $\emptyset\{p, q\}\emptyset^\omega$

(f) The following Muller automaton with acceptance conditions $\{\{q_0, q_1, q_2\}\}$:



Solution 2 (2 + 2 + 2 = 6 points)

(a) L_1 is not regular. Let $i, j \in \mathbb{N}$ be such that $i \neq j$. We have $a^i b^i \in L_1$ and $a^j b^i \notin L_1$, which implies that the a^i-residual and a^j-residual of L_1 differ. Therefore, $(L_1)^\varepsilon, (L_1)^a, (L_1)^{aa}, \ldots$ are all distinct, which implies that L_1 has infinitely many residuals. $\square$

(b) L_2 is not regular. Let $i, j \in \mathbb{N}$ be such that $i \neq j$. We have $a^{\max(i,j)} b^{\max(i,j)} \in L_2$ and $a^{\min(i,j)} b^{\max(i,j)} \notin L_2$, which implies that the a^i-residual and a^j-residual of L_2 differ. Therefore, $(L_2)^\varepsilon, (L_2)^a, (L_2)^{aa}, \ldots$ are all distinct, which implies that L_2 has infinitely many residuals. $\square$

(c) L_3 is regular: $a^0 b^0 b^* + a^1 b^1 b^* + \cdots + a^{999} b^{999} b^*$.

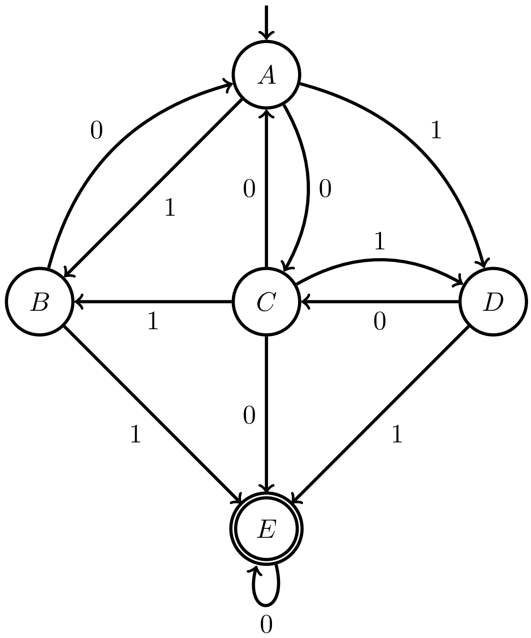
Solution 3 (1 + 2 + 2 = 5 points)

(a) $\{A, B, C, D\}$ and $\{E, F\}$

(b)

Iter.	Block to split	Splitter	New partition
0	—	—	$\{A, B, C, D\}, \{E, F\}$
1	$\{A, B, C, D\}$	$(1, \{E, F\})$	$\{A, C\}, \{B, D\}, \{E, F\}$
2	$\{A, C\}$	$(0, \{E, F\})$	$\{A\}, \{B, D\}, \{C\}, \{E, F\}$
3	$\{B, D\}$	$(0, \{C\})$	$\{A\}, \{B\}, \{C\}, \{D\}, \{E, F\}$
4	none, partition is stable	—	—

(c)

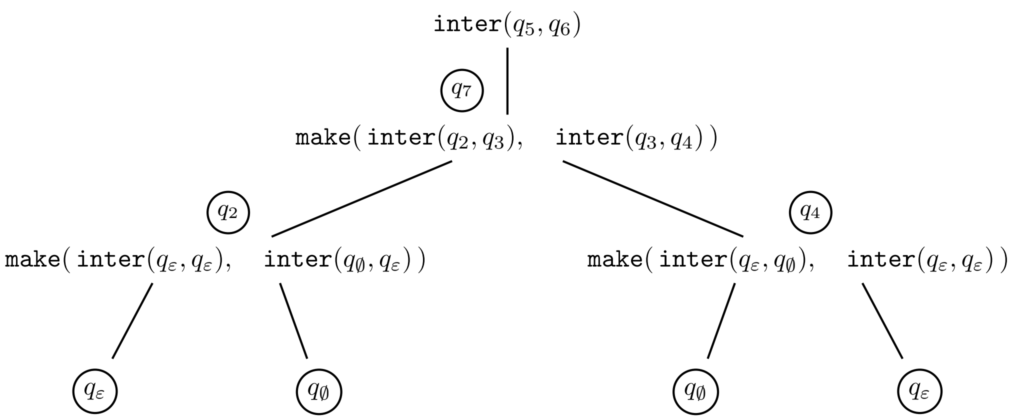


Solution 4 (3 + 3 = 6 points)

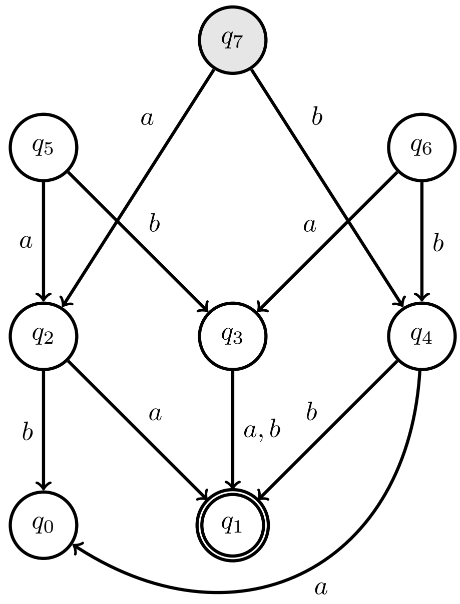
(a)

- Blank 1: $q_1 = q_\emptyset \vee q_2 = q_\emptyset$
- Blank 2: $q_1 = q_\varepsilon \wedge q_2 = q_\varepsilon$
- Blank 3: $\texttt{inter}(q_1^{a_i}, q_2^{a_i})$

(b) The tree of recursive calls is as follows:



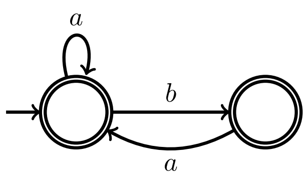
The resulting automaton is as follows:



Solution 5 $(2+2+2=6$ points)

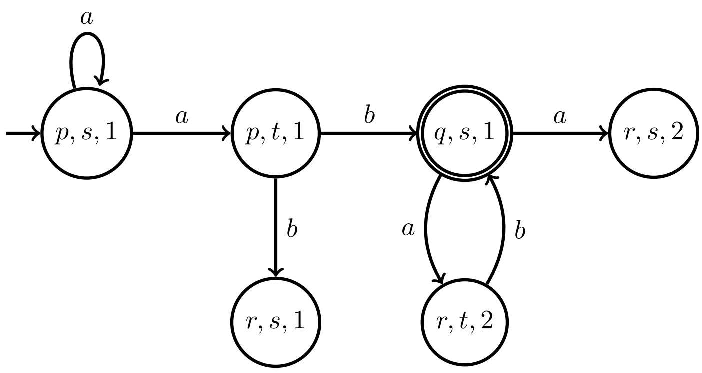
(a) $\forall x\ \forall y\ [((x<y)\wedge Q_b(x)\wedge Q_b(y))\rightarrow(\exists z\ ((x<z)\wedge(z<y)\wedge Q_a(z)))]$

(b)



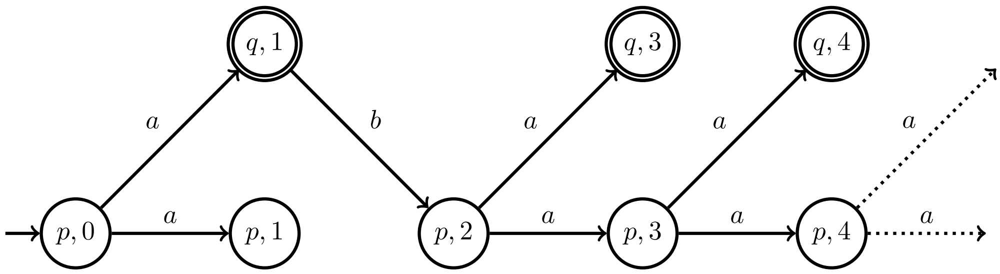
(c) $(a^*(ba)^*)^*(\varepsilon+b)$

Solution 6 (5 points)

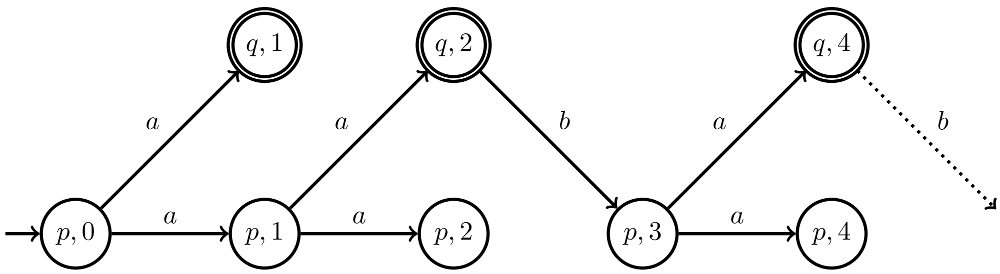


Solution 7 $(2 + 2 + 2 = 6$ points$)$

(a) $dag(aba^\omega)$:

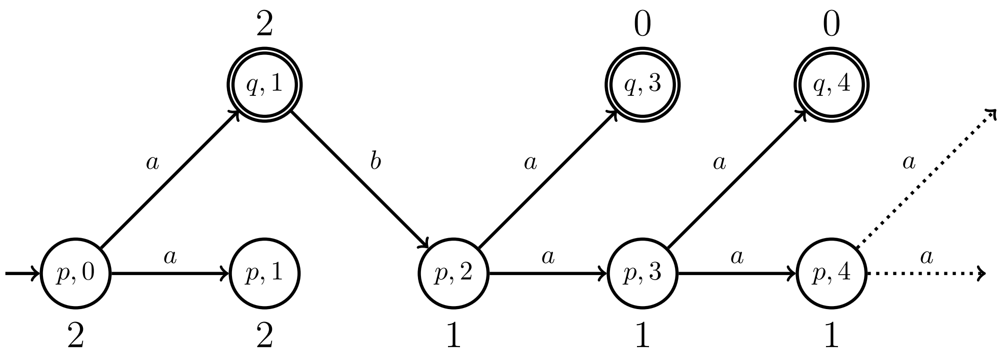


$dag(a(ab)^\omega)$:



(b) A ranking of $dag(w)$ is a function that maps each node of $dag(w)$ to a non negative number, called *rank*, and such that

- the rank of a node is greater than or equal to the rank of its children,
- the rank of an accepting node is even,
- there are infinitely many odd ranks along every infinite path of $dag(w)$.



(c) For the sake of contradiction, suppose $dag(a(ab)^\omega)$ has an odd ranking r. Let σ be the infinite path

$$(p,0) \xrightarrow{a} (p,1) \xrightarrow{a} (q,2) \xrightarrow{b} (p,3) \xrightarrow{a} (q,4) \xrightarrow{b} (p,5) \xrightarrow{a} \cdots$$

of $dag(a(ab)^\omega)$. Since r is non increasing, there exist $i, n \in \mathbb{N}$ such that r assigns rank n to every state of σ of level at least i. By assumption, r is odd and hence n must be odd. Since σ contains infinitely many accepting states, this implies that r assigns odd ranks to some accepting states, which is a contradiction. □