DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering, Mathematics and Computer Science

Midterm TI2316
Automata, Languages & Computability

May 22, 2019, 13:30–15:30

- Total number of pages (without this cover page): 8.
- This exam consists of 7 open questions, the weight of each subquestion is indicated on the exam.
- Consulting handouts, readers, notes, books or other sources during this exam is prohibited. The use of electronic devices such as calculators, mobile phones etc is also prohibited.
- A single exam cannot cover all topics, so do not draw conclusions based on this exam about topics that are never tested.
- Formulate your answers in correct English and write legibly (use scrap paper first). Do not give irrelevant information, this could lead to a deduction of points.
- Before handing in your answers, ensure that your name and student number is on every sheet of paper.

Question:	1	2	3	4	5	6	7	Total
Points:	6	6	5	2	4	3	6	32

Learning goals coverage, based on the topics of the study assignments:

Goal	MT 16-17	ET 16-17	MT 17-18	ET 17-18	MT18-19
Strings & Operations					
Inductive proof over strings					
Languages & Operations					
(Formal) def. of DFA	2b	1a	1a, 1b		1a
Accepting words & Recognising languages (DFA)				1a	
(Formal) def. of δ^*	1b		1a, 1b	1b	1b
Regular languages & operations					1c
Extension input alphabet of a DFA				2b	
Closure under union, complement, intersection	2b	2c		2a	2a, 2b
(Formal) def. of NFA	1a		1c	3c	
Equivalence of NFA & DFA		2c	3a,3c		
Accepting words & Recognising languages (NFA)			2a		3a
Closure under concat and star	1d	1c	2b		3b
(Formal) def. of regexp		2a		3a,3c	3b
(Formal) def. of GNFA			1c		4
Equivalence of NFA & regexp	1c	2b	3b,3c	3d	
Accepting words & Recognising languages (GNFA)				3b	
Nonregular languages	2a		4	4	5
Pumping lemma for reg. languages	2a		4	4	7
Basic concepts of grammar	3a		5a		
(Formal) def. of CFGs	3c	1b,3a,3b	5b		5a
Derivations & Ambiguity	3b				5b
Closure of context-free lang	3d	3c	5c		
Converting a CFG into CNF		3d		5a	
(Formal) def. of PDAs	4a, 4b, 4c				
Equivalence of CFGs & PDAs				5b	6

1. Consider a DFA $D = (Q, \Sigma, \delta, q_0, F)$.
   (a) What, if anything, can we conclude about the language $L(D)$ when
      i. (1 point) $|F| = |Q|$? *(Explain your answer in at most 3 lines.)*

      Solution: $L(D) = \Sigma^*$, as all states accept.

      ii. (1 point) $0 < |F| < |Q|$? *(Explain your answer in at most 3 lines.)*

      Solution: We cannot conclude anything specifically. There are some accepting and some rejecting state(s), but we do not know anything about the specifics of the language. Under the assumption that all states are reachable, we can conclude: $L(D) \neq \emptyset$ and $L(D) \neq \Sigma^*$.

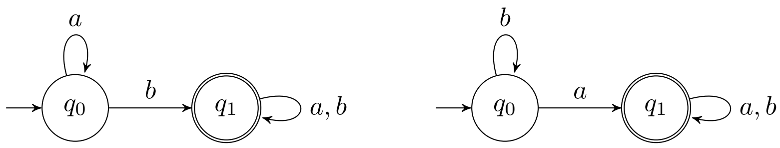
   (b) (2 points) A student claims that if $\delta^*(q_0, \text{pho}) = q_1$, $\delta(q_1, \text{e}) = q_0$, and $\delta^*(q_0, \text{phoenix}) \notin F$, it must also hold that nix $\notin L(D)$. You may assume {p,h,o,e,n,i,x} $\subseteq \Sigma$. Is that reasoning correct? *(Explain your answer in at most 5 lines.)*

   Solution: Yes, as we can chain the first two items of information to conclude that $\delta^*(q_0, \text{phoe}) = q_0$, so $\delta^*(q_0, \text{nix}) = \delta^*(q_0, \text{phoenix}) \notin F$. With all the steps:

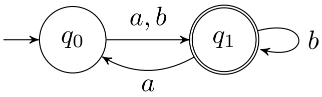
$$\begin{aligned}\delta^*(q_0, \text{phoenix}) &= \delta^*(\delta^*(q_0, \text{pho}), \text{enix}) \\ &= \delta^*(q_1, \text{enix}) \\ &= \delta^*(\delta^*(q_1, \text{e}), \text{nix}) \\ &= \delta^*(q_0, \text{nix})\end{aligned}$$

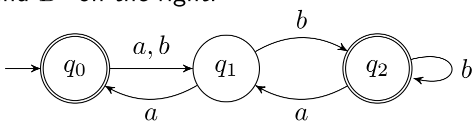
   (c) (2 points) Consider now a DFA $D' = (Q, \Sigma, \delta', q_0, F)$. A student claims that since the set of accepting states of D' and D are the same, it must hold that $L(D') = L(D)$. Is the reasoning correct? If so, prove it. If not, provide a counterexample in the form of two DFAs of at most 3 states each.

   Solution:



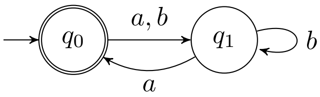
2. Consider the following two DFAs, D on the left and D' on the right:



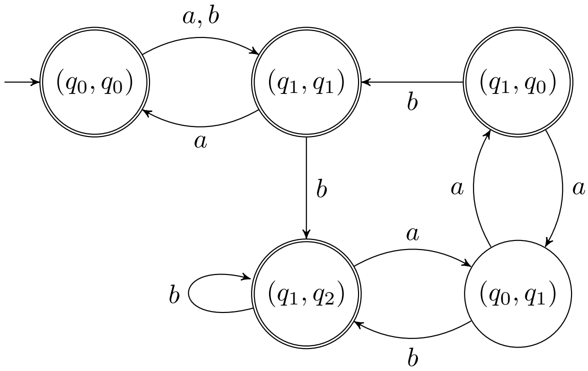


(a) (1 point) Construct a new DFA D'' of at most 2 states, such that $L(D'') = \overline{L(D)}$. A transition diagram and a short explanation suffice.

Solution: We should simply invert the accepting states.

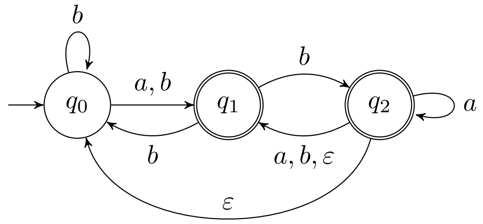


(b) (5 points) Use the procedure from Sipser to construct a DFA D''', such that $L(D''') = L(D) \cup L(D')$. You should leave out unreachable states.



Solution:

3. Consider the NFA N depicted as:

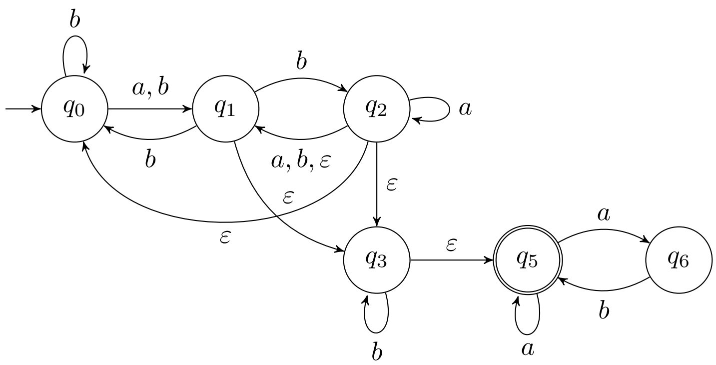


(a) (1 point) Give a word of length 5 that is in the language $L(N)$.

Solution: For example: abaaa

(b) (4 points) Add at most 5 states to N and modify the accepting states to create a new NFA N' such that $L(N') = L(N) \circ L(b^*(a \cup ab)^*)$.

Solution:



4. (2 points) State three properties a GNFA should adhere to in one line each.

Solution:

- Only one accepting state.
- No incoming transitions into the start state.
- No outgoing transitions from the accepting state.
- Every state is connected to every state, with the exception of the start and accepting state (which have no incoming and outgoing transitions respectively).
- Regular Expression on all transitions.

5. Consider the following rules of a CFG $G = (V, \Sigma, R, S)$, over $\Sigma = \{a, b, c\}$:

$$S \to bAb \mid aBa \mid A \mid B$$
$$A \to a \mid b$$
$$B \to c \mid AbA \mid \varepsilon$$

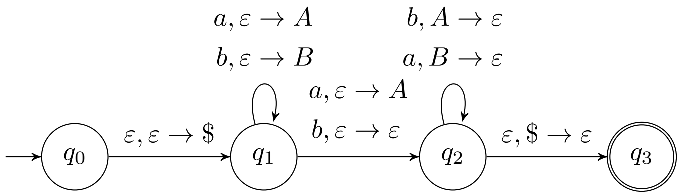
(a) (1 point) Give a valid set V.

Solution: $V = \{S, A, B\}$

(b) (3 points) Show that this grammar is ambiguous. *(Explain your answer in at most 6 lines.)*

Solution: Take for example the word bbb, which can be derived: $S \to bAb \to bbb$ or $S \to B \to AbA \to bbA \to bbb$.

6. (3 points) Consider the following PDA P; give a grammar G with at most 6 rules such that $L(G) = L(P)$.



Solution:

$$S \to aSb \mid bSa \mid ab \mid b$$

7. (6 points) Consider the following language, where the alphabet $\Sigma = \{a, b\}$:

$$L = \{b^n abv \mid v \in \Sigma^*, n \geq 1, \text{ and } |v| = 8n + 4\}$$

This language is **not** regular. Prove this using the pumping lemma in at most 20 lines.

Solution:

Proof. Proof by contradiction:

- Suppose L is regular.
- This means there must exist some pumping length $p > 0$ for L such that all words w longer than p can be split up into three parts x, y and z, with $|y| > 0$ and $|xy| \leq p$.
- For this division of w, and any $i \geq 0$, $xy^iz \in L$.
- Let's take the word $w = b^p aba^{8p+4}$ which is in L.
- This word is longer than p, so the above holds for this word.
- Given the requirements, we know that there is only one possible split
  - $x = b^\alpha, y = b^\beta$ and $z = b^{p-\alpha-\beta}aba^{8p+4}$, with $0 \leq \alpha < p$, $0 < \beta \leq p$ and $\alpha + \beta \leq p$.
    - Now, taking $i = 0$, we get $b^\alpha b^{p-\alpha-\beta}aba^{8p+4} = b^{p-\beta}aba^{8p+4}$, which is clearly not in L since $\beta > 0$.
- We have obtained a contradiction, so L must not be regular.

□