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

Resit TI2316
Automata, Languages & Computability

August 14, 2018, 9:00–12:00

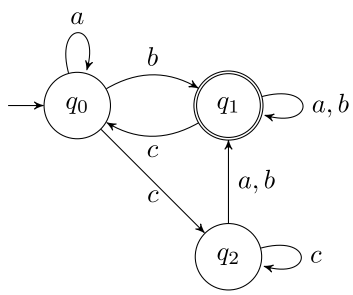
- Total number of pages (without this cover page): 8.
- This exam consists of 9 open questions, the weight of each subquestion is indicated on the exam.
- Consulting handouts, readers, notes, books or other sources during this exam is prohibited. The use of electronic devices such as calculators, mobile 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 or Dutch 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 page and indicate the number of pages handed in on (at least) the first page.
- **Note:** for some exercises a maximum is stated for the number of lines an answer can consist of! Exceeding this number may lead to deduction of points.

Question:	1	2	3	4	5	6	7	8	9	Total
Points:	9	12	6	6	11	7	7	6	6	70

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

Goal	MT 16-17	ET 16-17	MT 17-18	ET 17-18	R 17-18
Strings & Operations					
Inductive proof over strings					
Languages & Operations					
(Formal) def. of DFA	2b	1a	1a, 1b		1a, 5c
Accepting words & Recognising languages (DFA)				1a	1b
(Formal) def. of δ^*	1b		1a, 1b	1b	
Regular languages & operations					
Extension input alphabet of a DFA				2b	
Closure under union, complement, intersection	2b	2c		2a	1c
(Formal) def. of NFA	1a		1c	3c	5c
Equivalence of NFA & DFA		2c	3a,3c		2b
Accepting words & Recognising languages (NFA)			2a		2a
Closure under concat and star	1d	1c	2b		2c
(Formal) def. of regexp		2a		3a,3c	2d, 5c
(Formal) def. of GNFA			1c		
Equivalence of NFA & regexp	1c	2b	3b,3c	3d	
Accepting words & Recognising languages (GNFA)				3b	
Nonregular languages	2a		4	4	
Pumping lemma for reg. languages	2a		4	4	
Basic concepts of grammar	3a		5a		
(Formal) def. of CFGs	3c	1b,3a,3b	5b		3a,3c
Derivations & Ambiguity	3b				3b
Closure of context-free lang	3d	3c	5c		
Converting a CFG into CNF		3d		5a	
(Formal) def. of PDAs	4a, 4b, 4c				4a
Equivalence of CFGs & PDAs				5b	4b
(Formal) def of (multitape) DTMs		4a,4b,4c			
(Formal) def of (multitape) NTMs				8a,8b	5a,5b, 5c
Deciders		4d,5b,5c(?)		8c	
Equivalence of TMs				6b	
Comp. power of NTMs and DTMs					5d
Differences between NTMs and DTMs					
König's Lemma					
Enumerators				9a	6a
Recognisers		5b		9a	
Hilbert's Entscheidungsproblem					
Churing-Turing Thesis					
Encoding TMs/Problems					
Decidable languages				6a	6b
Countable vs Uncountable					8a
Hilbert Hotel					
(Un)Countability of $\mathbb{Q}$ and $\mathbb{R}$					8b
Halting problem				6c,6d	9a
Acceptance problem		5a(?)		9a	
Universal TMs					
co-Turing-recognizability				9b	
(Formal) def. of a reduction				7a,7b	
Direct reductions					
Computable functions				8a	9b
Mapping/Many-to-one reducability		5b,6a,6b,6c		10a,10b	9c
Rice's theorem					10
Reduction via computation histories					

1. Consider the following transition diagram of a DFA D over the alphabet $\Sigma = \{a, b, c\}$:



(a) (1 point) Give a word of length 6 that is in the language $L(D)$.

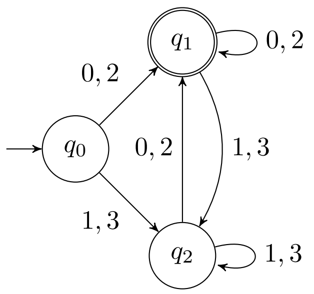
> **Solution:** $aaaaab$ is an example of a six-letter word that is in $L(D)$. Another example is $baaaaa$.

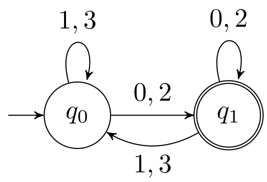
(b) (1 point) Give a word that is in the language $\{w | w = uv, u \in \Sigma, v \in \Sigma^* \text{ and } \delta(q_0, u) \in F\}$, where F is the set of accepting states of D.

> **Solution:** The second language concerns itself only with words of which the first letter would cause the word to be accepted. So for instance b is an example ($v = \varepsilon$).

(c) (4 points) Construct a DFA D' with at most 4 states over the alphabet $\Sigma = \{0, 1, 2, 3\}$ that accepts all strings that represent even numbers. E.g., $0 \in L(D')$, $00 \in L(D')$, $22 \in L(D')$, but $31 \notin L(D')$.

> **Solution:**

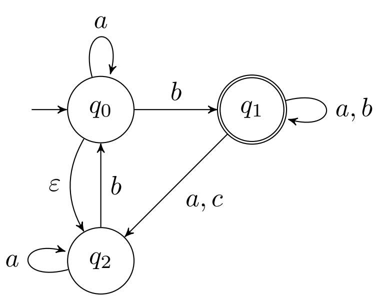




(d) (3 points) Describe how to create an NFA N over alphabet $\Sigma = \{a, b, c, 0, 1, 2, 3\}$ such that $L(N) = \overline{L(D)} \cup \overline{L(D')}$.

> **Solution:** Simply invert the accepting states of both machines and then put an epsilon transition from a new start state to both start states. Of course we should also rename the states to avoid conflicts.

2. Consider the following NFA N over the alphabet $\Sigma = \{a, b, c\}$:

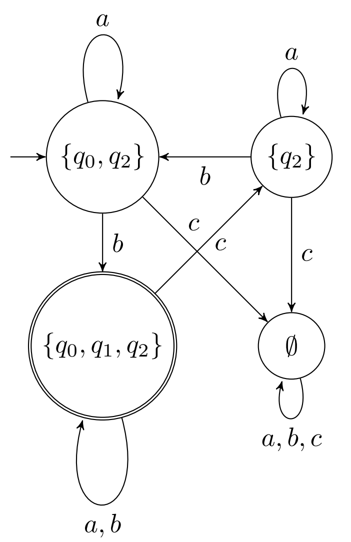


(a) (2 points) Take a new NFA N' that is identical to N except that $\delta(q_1, c) = \{q_0\}$. Give a word w, such that $w \notin L(N)$, but $w \in L(N')$.

Solution: Take for instance bcb. In N: $\{q_0, q_2\} \to \{q_0, q_1, q_2\} \to \{q_2\} \to \{q_0, q_2\}$ and thus reject. However, in N' we get: $\{q_0, q_2\} \to \{q_0, q_1, q_2\} \to \{q_0, q_2\} \to \{q_0, q_1, q_2\}$.

(b) (5 points) Using the procedure from Sipser, convert N to a DFA D, s.t. $L(D) = L(N)$. You may leave out unreachable states.

Solution:



(c) (2 points) Create a new NFA N'' such that $L(N'') = L(R)$ with $R = (a \cup b)^* c(b \cup a^*)$. Use at most 5 states in your NFA.

Solution: Start node reads a and b with self-loop, proceeds to next node with c. This node either goes to an accepting state q_b with a b where q_b has no transitions. Or it goes to an accepting state q_a with an ε-transitions, where q_a has a self-loop with a.

(d) (3 points) Describe how to create a new NFA M, such that

$$L(M) = \{w^2uv^3 \mid w \in L(D), u \in L(R), v \in L(N)\},$$

where D is your DFA from question b and R is the regular expression from question c. *(Answer in at most 8 lines.)*

Solution: Take two copies of D, the NFA from the previous question and three copies of N and chain them all together (making accepting states of the previous automaton no longer accepting and connecting them to the start state of the next automaton with an epsilon transition). Of course we should also rename the states to avoid conflicts.

3. Consider the grammar $G = \langle V, \Sigma, R, S\rangle$, with $\Sigma = \{a, b, c\}$ and R:

$$S \to aAa \mid bBb \mid cCc$$
$$A \to S \mid aBa$$
$$B \to S \mid bCb$$
$$C \to S \mid \varepsilon \mid a \mid b \mid c$$

(a) (1 point) Give a valid V for the grammar G.

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

(b) (3 points) Is G ambiguous? Motivate your answer. *(Answer in at most 10 lines.)*

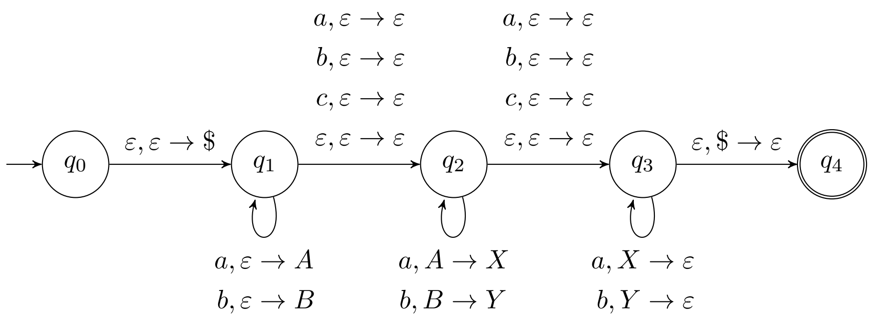
Solution: Yes. Take for instance the word: aaccaa. There are two left-most derivations for this word:
$S \to aAa \to aaBaa \to aaSaa \to aacCcaa \to aaccaa$, or alternatively:
$S \to aAa \to aSa \to aaAaa \to aaSaa \to aacCcaa \to aaccaa$.

(c) (2 points) Give a rule set R' of a grammar G' such that $L(G') = (L(G) \cup \overline{L(G)})^*$ and $|R'| \leq 6$. Explain how your rule set is constructed in at most 5 lines.

Solution: $L(G) \cup \overline{L(G)} = \Sigma^*$, so we can simply have the rules R':

$$S \to aS \mid bS \mid cS \mid \varepsilon$$

4. Consider the following PDA P over the alphabet $\Sigma = \{a, b, c\}$:



(a) (3 points) Describe the language of $L(P)$ in set notation. Explain your answer in at most 5 lines.

Solution:

$$L(P) = \{w \mid w = uvuwu, u \in \{a, b, \varepsilon\}, v, w \in \{a, b, c, \varepsilon\}\}$$

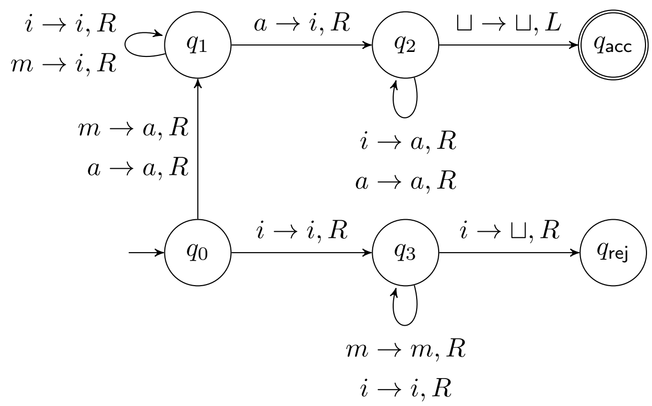
(b) (3 points) Consider now the grammar G, with the rule set:

$$S \to AXAXA \mid BXBXB \mid \varepsilon$$
$$A \to aA \mid \varepsilon$$
$$B \to bB \mid \varepsilon$$
$$X \to a \mid b \mid c \mid \varepsilon$$

Show that $L(G) \neq L(P)$.

Solution: Take for instance the word $aaacaca$ this is in $L(G)$ as it can be derived by: $S \overset{*}{\Rightarrow} AXAXA \overset{*}{\Rightarrow} aAcaAcaA \overset{*}{\Rightarrow} aaAcaca \overset{*}{\Rightarrow} aaaAcaca \overset{*}{\Rightarrow} aaacaca$, but it is not in $L(P)$ as we push 3 A's on the stack, but can only read one.

5. Consider the Turing Machine T:



(a) (2 points) The machine is currently in the configuration: $aiiq_1iaiia\sqcup$. What do we know about the initial word on the tape?*(Answer in at most 5 lines.)*

> **Solution:** We cannot determine what word it is, as the a and the i before the q_1 could have come from multiple letters. All we do know is that the word ends in $iaiia$.

(b) (2 points) Consider again the configuration from question a. Does the machine end up in an accepting state?

> **Solution:** We can say that the word will be accepted. The i will be read in q_1, then the a to move to q_2. It consumes the rest and then moves to q_{acc}.

(c) (2 points) Is T deterministic or non-deterministic? Explain your answer in at most 5 lines.

> **Solution:** T is non-deterministic, as q_2 has no transition for m and q_3 has two for i.

(d) (3 points) Is $L(T)$ regular? If so show it, if not explain why not. *(Answer in at most 5 lines.)*

> **Solution:** The language $L(T)$ can be expressed by the following regular expression: $(m \cup a)(m \cup i)^*a(i \cup a)^*$. Since the tape head only moves right on the accepting path, all the symbols written by the TM are irrelevant and we only need to consider the inputs read. This leads to the regular expression given above.

(e) (2 points) Given an arbitrary NTM N that decides $L(N)$, explain how a DTM D can decide $L(N)$. *(Answer in at most 10 lines.)*

> **Solution:** Something similar to the proof of theorem 3.16. Main notions are that we require 3 tapes (input, simulation, address) and that we consider all computations paths in order.

6. Consider the language:

$$L = \{w|w = f(x) \quad \forall x(\text{where } x \text{ is an integer larger than or equal to } 0)\}$$

where $f(x) = x + 8$.

(a) (5 points) Give a high-level description of a TM M that enumerates L.

Solution:

1. Clear the tape.
2. Put $8\sqcup$ on the tape.
3. Take the previous number of the tape and add 1 to it.
4. Write the result and a $\sqcup$ to the tape.
5. Go to step 3.

(b) (2 points) Consider a language L that can be enumerated. Now we also create another machine that recognises $\overline{L}$. What can we conclude about the decidability of L?

Solution: That means both $L(M)$ and $\overline{L(M)}$ are recognisable, thus $L(M)$ is decidable.

7. (a) (3 points) Consider the set A that contains all integers divisible by 42.[1] Show that A is countable.

> **Solution:** We need to find a bijection from $f : \mathbb{N} \to A$. Take $f(n) = 42n$. This is clearly a bijection as every n results in exactly one element of A and every element of A can only originate from exactly one element of $\mathbb{N}$.

(b) (4 points) Consider the following two subsets of $\mathbb{R}$ and explain whether they are countable or not. Answer in at most 5 lines per subset.

I. $\{x \in \mathbb{R} \mid x = \pi^k,$ with $k \in \mathbb{Q}\}$
II. $\{x \in \mathbb{R} \mid 0 \leq x \leq 0.1\}$.

> **Solution:**
>
> I. Since $\mathbb{Q}$ is countable, this set is also countable. Instead of using the bijection from $n \in \mathbb{N}$ to $k \in \mathbb{Q}$ we modify it to be from $n \in \mathbb{N}$ to π^k with $k \in \mathbb{Q}$.
>
> II. For this smaller range we can still use a diagonalisation method to prove the uncountability (not required to give one for the points).

8. (a) (2 points) Is A_{TM} Turing-recognisable and/or co-Turing-recognisable? Explain your answer. *(Answer in at most 5 lines.)*

> **Solution:** A_{TM} is only recognisable. The machine M when ran on input s when M accepts the word, will by definition not loop. Thus when $\langle M, w\rangle \in A_{TM}$, we will be able to observe this. However as it may loop for $\langle M, w\rangle$ we can not recognise $\overline{A_{TM}}$.

(b) (4 points) Consider now the following problem:

$\{M \mid M$ is a Turing Machine with 5 states$\}$

Show that this problem is co-Turing-recognisable.

> **Solution:** Consider the following TM that decides the language.
>
> 1. On input $\langle M\rangle$:
> 2. Count the number of states in $\langle M\rangle$.
> 3. If the number $\neq 5$: reject.
> 4. Otherwise: accept.
>
> Since it is decidable, it is also co-Turing-recognisable.

9. (6 points) Consider the following problem X:

$\{M \mid M$ is a Turing Machine, such that $P \subseteq L(M)\}$

where P is the set of all prime numbers. Use Rice's theorem to show that X is undecidable.

> **Solution:** We need to show two things:
>
> - $X \neq \emptyset$ and $X \neq \Sigma^*$: Take M with $L(M) = \emptyset$. Such a machine trivially exists and is not in X as $P \not\subseteq L(M)$. Also consider a machine N with $L(N) = \mathbb{N}$. Such a machine also trivially exists (it is even a regular language, as a simple regular expression can show) and is in X. So $X \neq \emptyset$ thanks to N and $X \neq \Sigma^*$ thanks to M.

[1] We say a number is divisible by 42 if after dividing by 42 the result is an integer. I.e., 84 is divisible by 42, but 21 is not.

- Take two machines M and N with $L(M) = L(N)$ then consider two cases. If $P \subseteq L(M)$ then also $P \subseteq L(N)$ so both are in X. Otherwise $P \not\subseteq L(M)$ so also $P \not\subseteq L(N)$ so both are not in X.

These are the two properties required by Rice's theorem, therefore X is undecidable.

End of the exam