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

TUDelft

Exam TI2316
Automata, Languages & Computability

July 1, 2019, 9:00–11:00

- Total number of pages (without this cover page): 7.
- 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 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	Total
Points:	7	6	6	4	4	5	5	37

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

Goal	ET 16-17	ET 17-18	R 17-18	ET 18-19
(Formal) def of (multitape) DTMs	4a,4b,4c			1a, 1c
(Formal) def of (multitape) NTMs		8a,8b	5a,5b, 5c	2a,2b,2c
Deciders	4d,5b,5c(?)	8c		3a
Equivalence of TMs		6b		1b
Comp. power of NTMs and DTMs			5d	
Differences between NTMs and DTMs				
König's Lemma				2
Enumerators		9a	6a	
Recognisers	5b	9a		3b
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		4a
Universal TMs				
co-Turing-recognizability		9b		4b
(Formal) def. of a reduction		7a,7b		5a,5b
Direct reductions				7
Computable functions		8a	9b	
Mapping/Many-to-one reducability	5b,6a,6b,6c	10a,10b	9c	
Rice's theorem			10	6
Reduction via computation histories				

1. Consider a Deterministic Turing Machine (DTM) $D = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$. You should consider each of the following questions individually (unless otherwise specified), i.e., if changes are made to the machine they are only for that one subquestion.

   (a) (1 point) Someone argues that if the only incoming transition to q_{accept} is $\delta(q_2, a) = (q_{\text{accept}}, c, R)$ then all words that are accepted by the machine must end in an a. Do you agree with this argument? If you agree, explain why. If not, describe a machine D that forms a counterexample.

   Solution: The statement is incorrect. Take a machine that modifies the tape. For example it could take as input e, transition to q_1 replacing this e with an a. The machine could modify the tape so that the last character becomes an a, this does not require an a to be the last letter of the input. Furthermore if more characters follow the transition, this word would still be accepted, thus it need not be the *last* letter either.

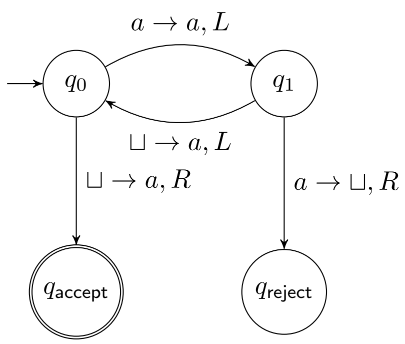
   (b) (2 points) Consider now an extension of the DTM model in which we change q_{accept} and q_{reject} by a set of accepting states A and a set of rejecting states R respectively. All other properties of the TM remain the same. We call such a deterministic TM an EDTM. Does this change expressivity of the model? That is, are there languages that cannot be recognised by a DTM, that can be recognised by an EDTM? If so, give an example of such a language or a set of requirements it should fulfil and explain them. If not, show how we can construct a classical DTM from such an EDTM.

   Solution: They are equivalent. As the DTM halts on an accepting state, we can simply merge all the accepting and rejecting states into two states q_{accept} and q_{reject} (with all incoming transitions going into that one state). Alternatively we can also add one new accept state again, and transition there for every symbol from the set of accept states (idem for the reject states).

(c) Consider now that $\Sigma = \{a\}$, $\Gamma = \Sigma \cup \{\sqcup\}$, $Q = \{q_0, q_1, q_{\text{accept}}, q_{\text{reject}}\}$ and δ specified as follows:

- $\delta(q_0, a) = (q_1, a, L)$
- $\delta(q_0, \sqcup) = (q_{\text{accept}}, a, R)$
- $\delta(q_1, a) = (q_{\text{reject}}, \sqcup, R)$
- $\delta(q_1, \sqcup) = (q_0, a, L)$

i. (2 points) Draw the machine D.



Solution:
NB: the q_0 to q_1 transition can also be correctly labelled with $a \to L$.

ii. (1 point) Give the language $L(D)$ of this machine.

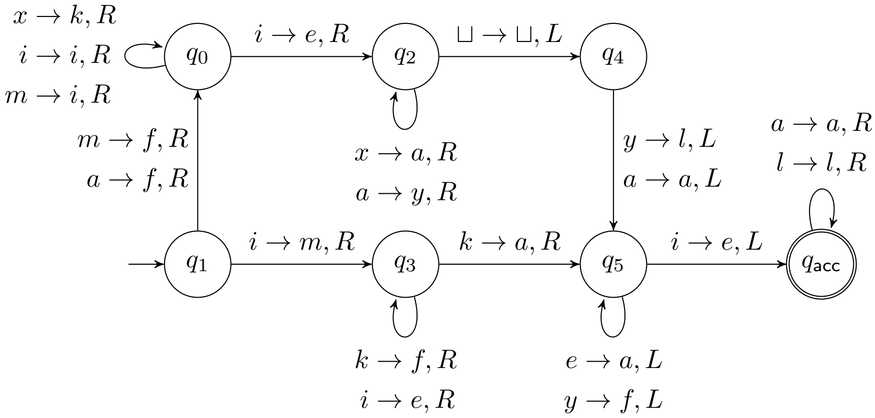
Solution: See below.

iii. (1 point) Assume the tape is bound on the left side, that is there is no empty space on the tape preceding the input. Is this DTM a decider? If so, explain why. If not, give a word. that shows it is not.

Solution: Yes. The empty string is accepted immediately, and any word that has at least one letter (this must be an a), will transition to q_1 without moving the head and then reject. So $L(D) = \{\epsilon\}$ and this is decided.

NB: Even assuming an infinite tape to the left, we would end up in the accept state so still decide the language.

2. Consider the following Turing machine T. You may assume all transitions that have not been drawn lead to a rejecting state.



(a) (3 points) Give a word $w \in L(T)$ of length 4 that starts with m and ends with a. Give a sequence of state configurations to show your word is in the language.

> **Solution:** For example: Read from left to right:
> $q_1 mmia\sqcup$ $\quad f q_0 mia\sqcup$ $\quad fi q_0 ia\sqcup$
> $fie q_2 a\sqcup$ $\quad fiey q_2 \sqcup$ $\quad fie q_4 y\sqcup$
> $fi q_5 el\sqcup$ $\quad f q_5 ial\sqcup$ $\quad q_{acc} feal\sqcup$

(b) (1 point) Is T deterministic or not? Explain your answer in at most 3 lines.

> **Solution:** No, as for example q_3 has two transitions for the symbol k.

(c) (2 points) Change at most 1 label of an edge so that the word *aix* is accepted. For your answer, you only need to give what edge needs to change and what the new label should be.

> **Solution:** For example: Change the edge from q_5 to q_{acc} so that it reads $f \to e, L$.

3. (6 points) Consider the following two languages. Both languages are recognisable. Explaining why is worth 2 points for each. Furthermore one of the languages is decidable. Picking the right one is worth 1 point and showing it is decidable is worth 1 point.

   (a) $MIN_{CFG} = \{\langle G\rangle \mid \text{G is a minimal CFG}\}$. We say a grammar is minimal if no rule can be removed without changing the language of G.

   > **Solution:** This is recognisable, we can simply enumerate strings from Σ^* (with the Σ of G) and look for strings that are in G but not in G with one rule removed. If we can find such a string for every rule R, then the grammar G is minimal and thus in the language.

   (b) $ALL_{DFA} = \{\langle A\rangle \mid \text{A is a DFA and } L(A) = \Sigma^*\}$

   > **Solution:**
   > The language is recognisable and decidable. We simply need to check if all reachable states are accepting. A simple BFS will do. If so the DFA is in the language, otherwise it is not.

4. For each of the following claims, state whether or not the claim is true and explain your answer in at most 5 lines per claim. There is no need for full proofs.

   (a) (2 points) The alternative to the word-acceptance problem A_{TM} called $\mathsf{PUZZLE}_{\mathsf{TM}} = \{\langle M\rangle \mid \text{M is a TM and M accepts the word } \textit{puzzle}\}$ is decidable.

   > **Solution:** Incorrect and we can use Rice's theorem to show this most easily. The language is clearly not empty, nor is it Σ^*. And if two machines share the same language then they also are both in the language A or both not in the language A.

   (b) (2 points) The problem $W_{\mathsf{TM}} = \{\langle M\rangle \mid \text{M is a TM and M is the only TM that accepts } L(M)\}$ is co-Turing-recognisable.

   > **Solution:** Correct. The language is empty, as every language has an infinite number of TMs that accept it. The empty language is decidable, thus also co-Turing-recognisable.

5. Consider three languages A, B and C.

   (a) (2 points) If A is decidable and $B \subseteq A$ and $A \subseteq C$ hold, then B and C are both decidable. Is this true? If so, prove it. If not, give a counterexample.

   > **Solution:** Take $A = \{\langle M, w\rangle\}$ for a given DTM M and some word w, $B = \emptyset$ (both decidable), now take any undecidable problem as C (for example A_{TM}) and both premises hold, but the conclusion is false as only one of B and C is decidable.
   > NB: You can also just take $A = B = \emptyset$ to make this work, or $A = C = \Sigma^*$ and for B take some undecidable problem.

   (b) (2 points) If $A \leq_{\mathsf{m}} B$ and $B \leq_{\mathsf{m}} C$ and B is undecidable, what, if anything, can we conclude about A and C?

   > **Solution:** We can only conclude that C is undecidable, but we cannot conclude anything about A.

6. (5 points) Use Rice's theorem to show the following language is undecidable. NB: Stating the two requirements of Rice's theorem is worth 2 points, applying the theorem correctly is worth the other 3.

$$L = \{\langle M\rangle \mid \text{M accepts at least two different strings of two different lengths}\}$$

Solution: We need to show $L \neq \emptyset$, which is true, consider a machine M with $L(M) = \{a, aa\}$, so $M \in L$. We need to show $L \neq \Sigma^*$, which is also true, consider a machine M with $L(M) = \emptyset$, so $M \notin L$. Together this shows the language is non-trivial (requirement 1).

The second property states that if $L(M_1) = L(M_2)$ then $M_1 \in L$ iff $M_2 \in L$. Take two machines M and M' with the same language. Now say $M_1 \in L$, this means that $L(M_1)$ has at least two different strings of two differnt lengths. Since $L(M_2) = L(M_1)$ this means that also $M_2 \in L$. The same argument can be made for if $M_1 \notin L$, meaning $L(M_1)$ does not have etc.

7. (5 points) Suppose we have the following language:

$$L = \{\langle M, w\rangle \mid M \text{ writes an } a \text{ on the tape when it is run on } w\}$$

We intend to prove that this language is undecidable. To that end we use a direct reduction from A_{TM} to L. Assume for the sake of contradiction that there is a TM R that decides L.

You should now construct a TM F that uses R to decide A_{TM}.

Solution: $F =$"On input $\langle M, w\rangle$:

1. Construct a Turing Machine T:
   $T =$ "On input x:
   (a) Simulate M on w.
   (b) If M accepts, write an a on the tape."
2. Run R on $\langle T, w\rangle$ (though any word will do, as T ignores the input anyway) to see if T writes an a on the tape.
3. If R accepts, M accepts w, so accept. Otherwise reject.

End of the exam