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

Exam TI2316
Automata, Languages & Computability

July 4, 2017, 13:30–16:30

- If your grade on the midterm was ≥ 5, you **do not have to do questions 1 through 3**. That means you do have to score at least 10 points on questions 4 through 6 (corresponding to a grade of 4 over only those questions), with the resulting final grade being the average of the two parts. You can always do the questions of the first part as well if you are not satisfied with your result on the midterm; the best grade counts.
- **Use separate answer sheets for the 2 parts.**
- Total number of pages (without this cover page): 4.
- This exam consists of 6 open questions of equal weight.
- 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 will lead to deduction of points.

The following 3 questions are about the first half of the course.

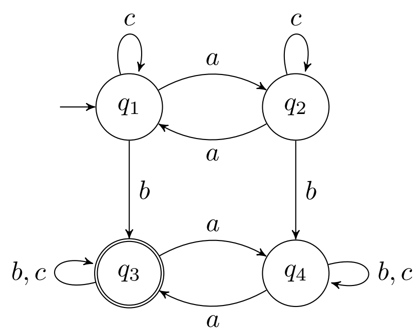
1. Suppose we have the following language L over the alphabet $\Sigma = \{a, b, c\}$:

   $L = \{w \in \Sigma^* \mid c_a(w) \text{ is even and } w \text{ contains a } b\}$,

   with $c_x(w)$ representing the number of occurrences of the symbol x in w.

   (a) (6 points) Give a deterministic finite automaton (DFA) D with *at most 7 states* such that $L(D) = L$. A transition diagram suffices. Also give a brief description of how D works. *(max. 10 lines)*

   Solution:

   The DFA keeps track of the even number of as horizontally (between q_1/q_2 and q_3/q_4) and the presence of a b vertically (between q_1/q_3 and q_2/q_4). Note that transitions also need to be defined for cs.

   (b) (2 points) Is L context free? Motivate your answer. *(max. 5 lines)*

   Solution: Yes, as every regular language is context free. L is a regular language as it can be recognized by a DFA (see answer to (a)).

   (c) (2 points) Suppose we append a string w' to every word of L. Is the resulting language regular? Motivate your answer. *(max. 5 lines)*

   Solution: Yes, this means concatenating L with another language L' which contains only a single word: $L' = \{w'\}$. As L and L' are both regular and regular languages are closed under concatenation, the resulting language is also regular.

2. Suppose we have the following regular expression:

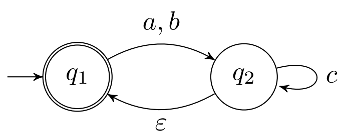
   $R = ((a \cup b)c^*)^*$

   (a) (2 points) Give two different words in $L(R)$.

   Solution: Examples: ε, a, b, ac, bc, $\ldots$

   (b) (5 points) Create an NFA N such that $L(N) = L(R)$. Use no more than 10 states.

   Solution:

(c) (3 points) Is it possible to create a finite automaton M with $L(M) = \overline{L(R)}$? Motivate your answer. *(max. 10 lines)*

> **Solution:** One could find an expression R' with $L(R') = \overline{L(R)}$, then make an automaton for that as above, or one could turn the above automaton into a DFA and invert its accept states. Note that inverting the accept states of an NFA doesn't always work.

3. Suppose we have languages L_1 and L_2, both defined over alphabet $\Sigma = \{a, b\}$:

$$L_1 = \{vw \mid v, w \in \Sigma^*,\ v = w^{\mathcal{R}}\}$$

$$L_2 = \{a^n \mid n \geq 0\}$$

(a) (2 points) Give a context-free grammar G_1 that generates L_1, i.e., such that $L(G_1) = L_1$.

> **Solution:** $G_1 = \langle\{S\}, \Sigma, R, S\rangle$, with R containing the following rules:
>
> $S \to aSa \mid bSb \mid \varepsilon$

(b) (3 points) Give a context-free grammar G_2 that generates $\overline{L_2}$, i.e., such that $L(G_2) = \overline{L_2}$.

> **Solution:** $G_2 = \langle\{T, A\}, \Sigma, M, T\rangle$, with M containing the following rules:
>
> $T \to AbA$
> $A \to aA \mid bA \mid \varepsilon$

(c) (2 points) Give a context-free grammar G_3 such that $L(G_3) = L_1 \cup \overline{L_2}$.

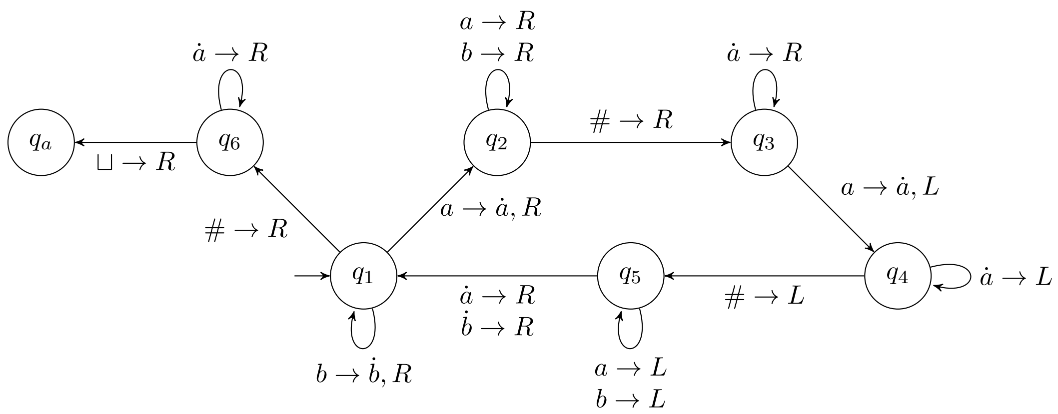
> **Solution:** $G_3 = \langle\{X, S, T, A\}, \Sigma, H, X\rangle$, with H containing the following rules:
>
> $X \to S \mid T$
> $S \to aSa \mid bSb \mid \varepsilon$
> $T \to AbA$
> $A \to aA \mid bA \mid \varepsilon$

(d) (3 points) Is G_3 in Chomsky normal form (CNF)? If so, indicate the properties it has that make it so; if not, indicate the properties it has that violate CNF. Mention 3 properties.

> **Solution:** G_3 as above is not in CNF: 1) it has unit rules (like $X \to T$), 2) a non-start variable (S) that can be rewritten to ε and 3) rules that consist of a mix of terminals and variables (like $S \to aSa$).

The following 3 questions are about the second half of the course.

4. Suppose we have the following transition diagram of Turing machine M with input alphabet $\Sigma = \{a, b, \#\}$ and tape alphabet $\Gamma = \{a, \dot{a}, b, \dot{b}, \#, \sqcup\}$. The tape is bounded on the left, q_a is the accepting state and any undefined transition implicitly leads to the reject state q_r, with the head moving right.



(a) (2 points) Give the series of configurations that M goes through when processing input $ba\#a$.

> **Solution:** $q_1ba\#a \vdash \dot{b}q_1a\#a \vdash \dot{b}\dot{a}q_2\#a \vdash \dot{b}\dot{a}\#q_3a \vdash \dot{b}\dot{a}q_4\#\dot{a} \vdash \dot{b}q_5\dot{a}\#\dot{a} \vdash \dot{b}\dot{a}q_1\#\dot{a} \vdash \dot{b}\dot{a}\#q_6\dot{a} \vdash \dot{b}\dot{a}\#\dot{a}q_6 \vdash \dot{b}\dot{a}\#\dot{a} \sqcup q_a$.

(b) (1 point) Does M accept input $ba\#a$? Motivate your answer. *(max. 3 lines)*

> **Solution:** M accepts $ba\#a$ as the processing ends in the accepting state q_a as shown above.

(c) (4 points) Give $L(M)$ as a set and explain how you reached your answer. *(max. 20 lines)*

> **Solution:** The main loop in M starts by marking an a following by going right until you have encountered the # symbol. Now the first unmarked a is found and marked, after which M goes left until it has read #. After reading # in q_4 M goes left through unmarked as and bs until reading the first marked symbol (which has to exists, since cannot enter the looping without marking an a), after which it goes right and ends up again in q_1. There are 3 options: 1) it reads an a and goes into the main loop again, 2) it reads a b which is marked or 3) it reads # and goes to q_6. The latter means that all symbols before # are marked and in q_6 M checks whether all as after # have been marked. This encodes that the number of as before and after # has to be equal. Note that when processing the part of the input after # (q_3 and q_4) no transitions for b are defined, so this part of the input can consist only of as. The bs before # are marked in q_1 but M does not check their number. Hence,
>
> $$L(M) = \{w\#a^n \mid w \in \{a,b\}^*, c_a(w) = n \geq 0\},$$
>
> with $c_x(w)$ representing the number of occurrences of the symbol x in w.

(d) (3 points) Is M a decider for $L(M)$? Indicate how you can see this. *(max. 10 lines)*

> **Solution:** M is a decider for $L(M)$, because all the letters eventually become marked and so it either reaches a point where it can only transition (implicitly) to the reject state or it reaches the accept state.

5. (a) (3 points) Suppose we have languages L_1, L_2 and L_3, with $L_1 \subseteq L_2 \subseteq L_3$. We also know that L_1 and L_3 are Turing-decidable. Does L_2 have to be decidable as well? Motivate your answer. *(max. 10 lines)*

> **Solution:** No, take for instance $L_1 = \emptyset$ and $L_3 = \Sigma^*$. Now $L_2 = A_{\mathsf{TM}}$ is possible, but this language is not decidable.

(b) (3 points) Suppose we have languages L_1, L_2 and L_3, with $L_1 \leq_m L_2$ and $L_2 \leq_m L_3$. We also know that L_1 is Turing-decidable and L_3 is Turing-recognizable. Is L_2 decidable and/or recognizable? Motivate your answer. *(max. 10 lines)*

Solution: We know that L_2 must be at least recognizable, by Theorem 5.28. We do not know whether L_2 is decidable, however.

(c) (4 points) Let E_{PDA} be defined like E_{TM}:

$$E_{\mathsf{PDA}} = \{\langle P \rangle \mid P \text{ is a PDA and } L(P) = \emptyset\}$$

Is E_{PDA} decidable? Motivate your answer. *(max. 20 lines)*

Solution: Yes, it is decidable. A PDA can be converted to a CFG (Lemma 2.27), and then we can use the TM R defined in Theorem 4.8 to decide the language.

6. Suppose we have the following language:

$$L = \{\langle M, v, w \rangle \mid M \text{ outputs } w \text{ on the tape when run on input } v\}$$

Use mapping reduction to prove that L is undecidable. Give:

(a) (1 point) A suitable problem to reduce from.

Solution: The problem used here is $HALT_{\mathsf{TM}}$.

(b) (3 points) A suitable reduction function $f : \Sigma^* \to \Sigma^*$.

Solution: The reduction f is computed by the following TM F:
$F =$"On input $\langle M, w \rangle$:

1. Construct the following TM M':
   $M' =$"On input x:
   1. Clear the input from the tape.
   2. Run M on w.
   3. Write an a to the tape and accept."
2. Write $\langle M, a, a \rangle$ to the output tape."

(c) (6 points) A proof showing f satisfies the requirements of a mapping reduction.

Solution:

Proof. We need to show that f is computable and reliable. F computes f, so f is computable. f is also reliable, because:

- $\langle M, w \rangle \in HALT_{\mathsf{TM}} \Rightarrow f(\langle M, w \rangle) \in L$ holds, since if M halts on w, M' leaves a on the tape on any input x and this is exactly what it is supposed to leave. So, $\langle M, a, a \rangle \in L$.
- $\langle M, w \rangle \notin HALT_{\mathsf{TM}} \Rightarrow f(\langle M, w \rangle) \notin L$ holds, since if M does not halt on w, M' will never write the correct output to the tape.

$\square$

End of the exam