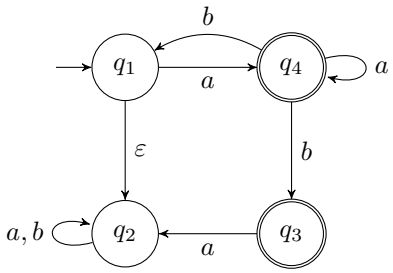
DELFT UNIVERSITY OF TECHNOLOGY
Faculty of Electrical Engineering, Mathematics and Computer Science

Midterm Exam TI2316
Automata, Languages & Computability

May 31, 2017, 13:30–15:30

- Total number of pages (without this cover page): 1.
- This exam consists of 4 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 a calculator, 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.

1. Suppose we have an NFA $N = (Q, \Sigma, \delta, q_1, F)$ with $\Sigma = \{a, b\}$ and the following transition graph:



   (a) (1 point) Give $Q - F$.

   (b) (1 point) Give $\delta^*(q_1, abaab)$.

   (c) (5 points) Use the method described by Sipser to find a regular expression R that is equivalent to N, i.e., such that $L(N) = L(R)$. Show every step in the process.

   (d) (3 points) Give the transition graph of an NFA H such that $L(H) = L(N) \circ L(ca^*c)$ and H contains no more than 7 states.

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

$$L = \{uv \mid u \in \Sigma^*, |u| = m \text{ and } v = a^n b^n c^n, \text{ with } m = 3n \text{ and } n \geq 0\}$$

   (a) (7 points) Is L a regular language? If so, give a regular expression R such that $L(R) = L$. If not, prove this using the pumping lemma.

   (b) (3 points) Give a brief description (in no more than 15 lines) of how one might construct a DFA D that recognizes the difference of the languages of two NFAs N_1 and N_2, i.e., $L(D) = L(N_1) - L(N_2)$.

3. Suppose we have a context-free grammar $G = (V, \Sigma, R, S)$, with R containing the following rules:

$$S \to ABC$$
$$A \to aC \mid D$$
$$B \to bB \mid A \mid \varepsilon$$
$$C \to Ac \mid Cc \mid \varepsilon$$
$$D \to aa.$$

   (a) (1 point) Give a parse tree for the word $acbbc$.

   (b) (2 points) Give two different *leftmost* derivations of the word $aaaac$.

   (c) (3 points) Give a grammar G' such that $L(G') = L(G)^*$.

   (d) (4 points) Prove that if L_1 and L_2 are context-free languages, then $L_1 L_2$ is also context free. (That is, context-free languages are closed under concatenation.)

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

$$L = \{a^m b^n \mid 2m = 3n + 1 \text{ and } m, n \geq 0\}$$

   (a) (6 points) Give a PDA P that recognizes L, i.e., such that $L(P) = L$. A transition graph suffices. Your PDA must have no more than 9 states.

   (b) (2 points) Explain in no more than 10 lines how P recognizes L.

   (c) (2 points) Show how P processes the word $aaaaabbb$ by giving a sequence of state descriptions.

End of the exam