Fachbereich Mathematik
Kord Eickmeyer

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Winter Term 2015/16

Mock Exam – Formal Foundations of Computer Science

(EM.1) [multiple choice, 8 credits]

Decide whether the following statements are true or false in general. No justification is needed.

(i) For regular languages $L_1, L_2, L_3 \subseteq \{0,1\}^*$ one can algorithmically check if $L_1 = L_2 \cap L_3$.

(ii) If L is a regular language and $M \subseteq L$, then M is regular.

(iii) If L is a regular language and $M \subseteq L$, then M is context-free.

(iv) If $L_1, L_2 \subseteq \Sigma^*$ are context-free, then $L_1 \cap L_2$ is decidable (i.e. one can algorithmically check for $w \in \Sigma^*$ whether $w \in L_1 \cap L_2$).

(v) If $L_1, L_2 \subseteq \Sigma^*$ are context-free, then $L_1 \cap L_2$ is also context-free.

(vi) Every context-free grammar is context-sensitive.

(vii) Every context-free language is context-sensitive.

(viii) If L is context-sensitive, then L^* is context-sensitive.

Solution.

(i) yes (generate DFA for $L_2 \cap L_3$ and check equivalence with the DFA for L_1)

(ii) no, otherwise every language would be regular (since Σ^* is regular and $L = L \cap \Sigma^*$)

(iii) no, otherwise every language would be context-free

(iv) yes (to check whether $w \in L_1 \cap L_2$ for a given w, check whether $w \in L_1$ and whether $w \in L_2$ and answer yes if both are true)

(v) no (we saw $\{a^n b^n c^n \mid n \geq 0\}$ is not context-free but $\{a^n b^n c^m \mid m, n \geq 0\}$ and $\{a^n b^m c^m \mid m, n \geq 0\}$ are.

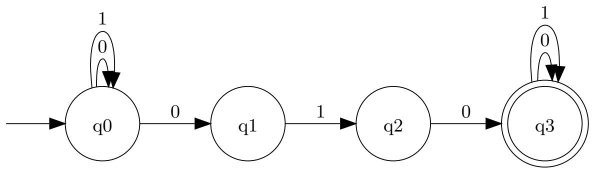
(vi) no, since context-free grammars may have arbitrary ϵ-productions.

(vii) yes, since for every context-free grammar we may find an equivalent one with only harmless ϵ-productions.

(viii) yes

(EM.2) [minimisation/conversion, 12 credits]

Consider the following NFA with four states:



(i) Convert this NFA into a DFA.

(ii) Minimise the resulting DFA.

Solution. We label states of the DFA in short-hand notation, i.e. 01 for $\{q0, q1\}$. With this we get

	0	1
$\rightarrow$ 0	01	0
01	01	02
02	013	0
013	013	023
023	013	03
03	013	03

The accepting states (shown in boldface) are those which contain 3, the initial state is 0. Minimisation shows that all three accepting states are equivalent.

(EM.3) [regular languages, 12 credits]

For a word $a_1a_2\dots a_\ell \in \{0,1\}^*$ let

$$f(a_1a_2\dots a_\ell) := \sum_{i=1}^{\ell} a_i 2^{i-1},$$

and $f(\epsilon) := 0$. In other words, $f(w)$ is the numerical value of w if it is interpreted as a natural number in binary representation, with the least significant bit first.

(i) Show that the language

$$L_1 := \{a_1b_1c_1a_2b_2c_2\dots a_\ell b_\ell c_\ell \mid f(a_1\dots a_\ell) + f(b_1\dots b_\ell) = f(c_1\dots c_\ell)\} \subseteq \{0,1\}^*$$

is regular by giving a DFA that accepts L_1.

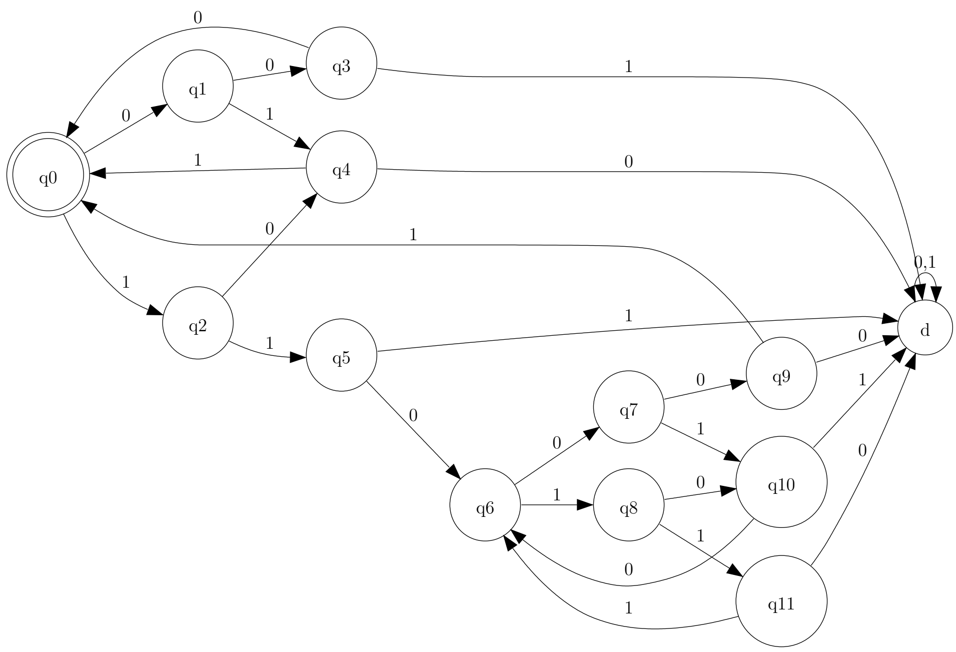
(ii) Show that the language

$$L_2 := \{uvw \mid u, v, w \in \{0,1\}^n \text{ for some } n \geq 0 \text{ and } f(u) + f(v) = f(w)\} \subseteq \{0,1\}^*$$

is *not* regular.

Solution.

(i) Note that in every block of three consecutive bits, the third must be the sum of the first two plus a carry bit. A word is accepted if after the last block the carry is zero. Thus we get (with initial state $q0$):



(ii) We show that L_2 is not regular: Let $n \geq 1$ and consider the word $0^n1^n1^n \in L_2$. Then $|w| \geq n$. Assume $w = xyz$ for words $x, y, z \in \{0,1\}^*$ with $|y| > 1$ and $|xy| \leq n$. Then $y = 0^k$ for some $1 < k \leq n$. If k is not divisible by 3 then $|xz|$ is not divisible by three and $xz \notin L_2$. Otherwise $xz = u1^m1^m$ for some word $u \in \{0,1\}^m$ which is *not* 0^m. But then $xz \notin L_2$. By the Pumping Lemma, L_2 is not regular.

(EM.4) [rotated regular language, 10 credits]

For a language $L \subseteq \Sigma^*$ we define $\mathrm{Rot}(L)$ as

$$\mathrm{Rot}(L) := \{a_2 a_3 \dots a_\ell a_1 \mid a_1 a_2 \dots a_\ell \in L\}.$$

Show that if L is regular, then $\mathrm{Rot}(L)$ is regular as well.

Solution. Let $A = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L. Assume the alphabet has size k and is given by $\Sigma = \{\sigma_1, \dots, \sigma_k\}$. We define an NFA $A' = (Q', \Sigma, \Delta, q_0', F')$ by

$$\begin{aligned}
Q' &:= Q^k \cup \{\tilde{q}\} \\
q_0' &:= (\delta(q_0, \sigma_1), \dots, \delta(q_0, \sigma_k)) \\
\Delta &:= \big\{\big((p^{(1)}, \dots, p^{(k)}), a, (q^{(1)}, \dots, q^{(k)})\big) \mid q^{(i)} = \delta(p^{(i)}, a)\big\} \cup \\
&\qquad \big\{\big((q^{(1)}, \dots, q^{(k)}), \sigma_i, \tilde{q}\big) \mid 1 \le i \le k, q^{(i)} \in F\big\}, \\
F' &:= \{\tilde{q}\}.
\end{aligned}$$

Then $L(A') = \mathrm{Rot}(L)$. Intuitively, the NFA simulates k copies of A in parallel, each starting in a possible successor state of q_0 after reading a single letter. If a letter σ_i is read, the automaton may guess that it is the last letter of the word to be processed, and go into the unique accepting state $\tilde{q}$ if the i-th copy is in an accepting state.

(EM.5) [context-free grammar, 12 credits]

Let $G = (\{f, g, x\}, \{T\}, P, T)$ be the context-free grammar with production rules

$$T \to x \mid fT \mid gTT$$

(i) Convert this grammar into Chomsky normal form.

(ii) Use the CYK algorithm to check that $fgfxx \in L(G)$.

Solution.

(i) We may choose $G' = (\{f, g, x\}, \{T, S, F, G\}, P', T)$ with production rules

$$\begin{aligned}
P' : T &\to x \mid FT \mid GS, \\
S &\to TT, \\
F &\to f, \\
G &\to G.
\end{aligned}$$

(ii) The CYK algorithm yields the following table, and we see that $fgfxx \in L(G)$.

	T	S	F	G
f			x	
g				x
x	x			
fg				
gf				
fx	x			
xx		x		
fgf				
gfx				
fxx		x		
fgfx				
gfxx	x			
fgfxx	x			