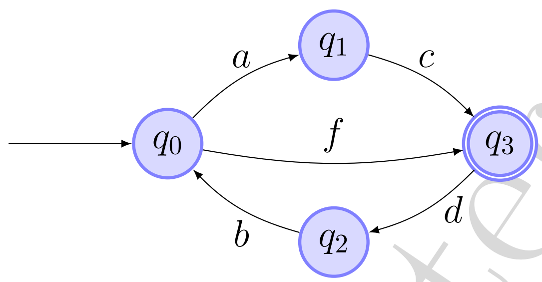
CPSC 313 (Winter 2016) L01

Midterm solution sketches

A 150 minutes exam held on March 3, 2016

1. Short answers

1. Give a regular expression for the language accepted by the following NFA.



There were four common mistakes:

(a) introduce an extra $+$ symbol, e.g. as in the following

$$\boldsymbol{r} = \Big(\boldsymbol{(ac+f)((db(ac+f))^*)}\Big)^+,$$

(b) omit a transition from q_0 to q_3 as e.g. in

$$\boldsymbol{r} = \boldsymbol{((ac+f)db)^+},$$

(c) give the loops as an alternative to the transition from q_0 to q_3 as e.g. in

$$\boldsymbol{r} = \boldsymbol{(ac+f+((ac+f)db)^*)},$$

(d) not permitting the two loops to intermix as in

$$\boldsymbol{r} = \boldsymbol{ac(dbac)^* + f(dbf)^*}.$$

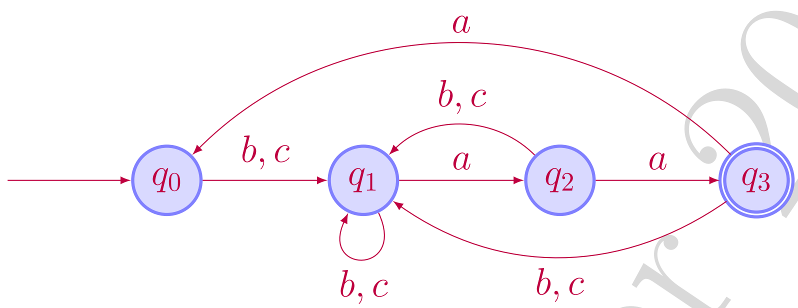
2. We say a context-free grammar is *symmetric* if whenever $A \to z_1 z_2 \cdots z_m$ is a rule in the grammar, then so is $A \to z_m \cdots z_2 z_1$. Here $z_i \in (V \cup T)$ for all i. Give an example of a language that can **not** be generated by a symmetric grammar.

$$L = \{ab\}$$

Assume $L = \{ab\}$ can be generated by a symmetric grammar G. This means that the string ab can be derived by successively applying rules in G. Then the reversed string ba can also be derived by instead successively applying the corresponding reversed rules in G. Since ba is not in L, this is a contradiction.

3. Give a DFA M accepting the language $L = \{w \in \{a, b, c\}^* \mid w \text{ ends in } baa \text{ or } caa\}$.

   We build a DFA around an NFA that accepts the regular expression $\boldsymbol{(b + c)aa}$, which is the path from q_0 to q_3 in the DFA below.



4. Give a right-linear grammar G for the language $L = L(\boldsymbol{r})$ where $\boldsymbol{r} = \boldsymbol{(a(b^*)c)^*}$.

$$S \to aA \ \mid \ \lambda; \qquad A \to bA \ \mid \ cS;$$

   The key observation here is the rule $A \to cS$, which expresses that once we are done with one instance of $\boldsymbol{ab^*c}$, we can restart and proceed to the next instance.

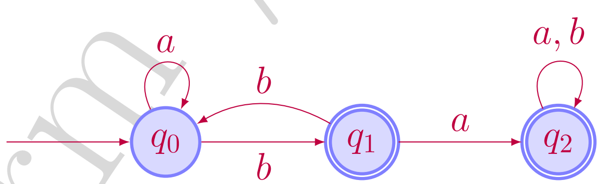
5. Give a regular expression $\boldsymbol{r}$ for the language $L = \{w \in \{a, b\}^* \mid |w|_a \le 2 \text{ or } |w|_b \le 2\}$.

$$\boldsymbol{r = b^*(a + \lambda)b^*(a + \lambda)b^* \ + \ a^*(b + \lambda)a^*(b + \lambda)a^*}$$

   The regular expression $(a + \lambda)$ captures that there may be an a symbol or not.

6. Give a DFA for the complement of the language $L(\boldsymbol{r})$ where $\boldsymbol{r} = \boldsymbol{(a + bb)^*}$.

   We build a DFA around an NFA that has two loops on q_0. One loop labeled a and another labeled bb which uses the intermediate state q_1.



7. Give a context-free grammar for the language $L = \{w \in \{a, b\}^* \mid |w|_a = 2|w|_b\}$. That is, a string w is in L if it contains twice as many a symbols as b symbols.

$$S \to aSaSbS \mid aSbSaS \mid bSaSaS \mid \lambda$$

   The most common mistake is to give restricted rules such as $S \to aaSb \mid abSa \mid baSa \mid \lambda$. Such constructions will not permit strings like $aaabbbaaa$, where the b symbols are nested deeply inside many a symbols. The split-recursion $S \to aSbSa$ can generate the outer-most recursive level of the string $aaabbbaaa$.

8. Give a language L_1 such that $L = L_1 L_2$ is regular, where $L_2 = \{a^i b^j \mid i \le j\}$.

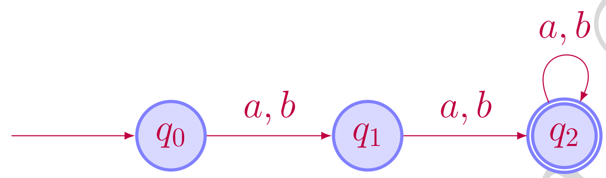
$$L_1 = L(\boldsymbol{a^*})$$

9. Give a context-free grammar for the language $L = \{a^i b^j c^k \mid i + k = j \text{ and } i, j, k \geq 0\}$.

$$S \to AC; \qquad A \to aAb \mid \lambda \qquad C \to bCc \mid \lambda$$

The language L is an iteration of two recursions, $L = \{a^i b^i b^k c^k \mid i, k \geq 0\}$. The two symbols A and C generate each of these two recursions.

10. Give a minimal DFA for the language $L = L(\boldsymbol{r})$ where $\boldsymbol{r} = ((\boldsymbol{a}^*\boldsymbol{b} + \boldsymbol{a}(\boldsymbol{b}^*))^+)(\boldsymbol{a} + \boldsymbol{b})$.



This question is implicitly asking to recognize that $(\boldsymbol{a}^*\boldsymbol{b}+\boldsymbol{a}(\boldsymbol{b}^*))^+$ is equivalent to $(\boldsymbol{a}+\boldsymbol{b})^+$.

2. Non-regular languages

Prove that the following language is non-regular

$$L = \left\{a^\alpha b^\beta c^\gamma d^\delta e^\epsilon f^\phi \mid \alpha \neq \beta \text{ and } \gamma \neq 2\epsilon \text{ and } \delta = \phi, \text{ and } \alpha, \beta, \gamma, \delta, \epsilon, \phi \geq 0\right\}.$$

You may use the theorems at the back of this exam without proofs.

1. By contradiction. Assume L is regular.
2. Let $L_2 = L \cap L(\boldsymbol{acd}^*\boldsymbol{f}^*)$. Then $L_2 = \{acd^i f^i \mid i \geq 0\}$.
3. Define the homomorphism $h(a) = \lambda$, $h(c) = \lambda$, $h(d) = a$, and $h(f) = b$.
4. Let $L_3 = h(L_2)$. Then $L_3 = L_{\text{cnt}} = \{a^i b^i \mid i \geq 0\}$, which is non-regular.

There are at least three alternative proofs: (1) Initially, apply the homomorphism $h(a) = h(b) = h(c) = h(e) = \lambda$ and $h(d) = a$ and $h(f) = b$, to obtain $h(L) = L_{\text{cnt}}$. (2) Apply the pumping lemma on the reversed language L^R. (3) Compute the complement language $\overline{L} \cap L(\boldsymbol{a}^*\boldsymbol{b}^*\boldsymbol{c}) = \{a^i b^i c \mid i \geq 0\}$, and then use either the pumping lemma, or a homomorphism that sends c to λ.

3. DFA with a reset button

We define a new computational model by equipping a DFA M with a **reset button**. A DFA M_{reset} with a **reset button** works as a DFA, except for the reset button. Consider we are running M_{reset} on some input string $w \in \Sigma^*$. If we press the reset button while being in some state $q_i \in Q$, we are instantaneously being transferred to the initial state q_0. We can press the reset button at most **once** during the entire computation of M_{reset} on w. We can choose when to press the reset button, or not press the button at all, but we can only press it once. Machine M_{reset} accepts a string w if we can end in a final state.

1. Let M be a DFA and $L = L(M)$ the language accepted by M. Let M_{reset} be M equipped with a reset button. Characterize the language L_{reset} accepted by M_{reset}. Explain and justify your answer.

$$L_{\text{reset}} = \Sigma^* \cdot L$$

This can be illustrated by drawing two copies of the same DFA, with λ-edges from every state in the first DFA to the initial state in the second DFA. The first DFA accepts Σ^* and the second L.

4. Ambiguities in regular languages •

Consider a right-linear grammar G. We say that two rules ρ_1 and ρ_2 in G are *ambiguous* if they are on the form

$$\rho_1 = A \to bu \qquad \text{and} \qquad \rho_2 = A \to bv$$

where $A \in V$, $b \in \Sigma$ and $u, v \in T^*(V \cup \{\lambda\})$. That is, the two rules can both be applied on the variable A, and they both start with the same terminal symbol b on the right-hand side.

We say that a grammar G is ambiguous if G contains a pair of rules that are ambiguous, or if G contains a rule of the form $A \to B$ where $A, B \in V$. We say that a grammar is *unambiguous* otherwise.

1. Show that for any right-linear grammar G, there exists another right-linear grammar G' that is unambiguous and that generates the same language.
(Full bonus points to complete solutions. No points to partial solutions.)

1. Let M be a DFA accepting $L(G)$.
2. For every edge $\delta(q_i, a) = q_j$ in the transition function δ for M, define the rule $Q_i \to aQ_j$.
3. For every final state $q_f \in F$ in M, define the rule $Q_f \to \lambda$.
4. Let Q_0 be the starting variable, for the initial state q_0 in M.
5. The resulting grammar G' is unambiguous and right-linear by construction, and it satisfies that $L(G') = L(M) = L(G)$.

5. True or False

Answer True or False. No justification required.
The languages are over the alphabet $\Sigma = \{a, b\}$, except in questions 1 and 8.

	Question	True	False
1	If $L = \left\{w \in \{a, b, c\}^* \mid \lvert w\rvert_a = \lvert w\rvert_b \text{ or } \lvert w\rvert_a = \lvert w\rvert_c\right\}$, then L is context-free	✓	
2	$L^* = (L^+)^*$	✓	
3	The grammar $S \to bSa \mid aSb \mid ab \mid ba$ generates the complement of the language generated by the grammar $S \to aSa \mid bSb \mid a \mid b \mid \lambda$		✗
4	$(L^+)^+ = L^+ \cdot L^*$	✓	
5	The grammar $S \to aSb \mid aS \mid Sb \mid \lambda$ generates a regular language	✓	
6	If $\lambda \in L$, then $\lambda \in L^+$	✓	
7	$\lambda^2 = \lambda$	✓	
8	$\{a^i b^i c^i \mid i \geq 0\} \cap \{b^i c^i d^i \mid i \geq 0\} = \{b^i c^i \mid i \geq 0\}$		✗
9	The grammar $S \to aSbSaSb \mid \lambda$ generates the language $\{a^i b^i a^i b^i \mid i \geq 0\}$		✗
10	The language $L(\boldsymbol{a}^*\boldsymbol{b}^*)$ is context-free	✓	
11	If both L_1 and L_2 are non-regular, then so is $L_1 \cap L_2$		✗
12	If $\{a^i b^i \mid i \geq 0\} \subseteq L$, then L is non-regular		✗
13	If $\lambda \in L$ then $L^{13} = L^{17}$		✗
14	If L is context-free, then so is L^*	✓	