Mid-term Exam
COSE215, Spring 2015

Instructor: Hakjoo Oh

Problem 1. (25pts) Recall that a DFA is defined by a tuple of five components:

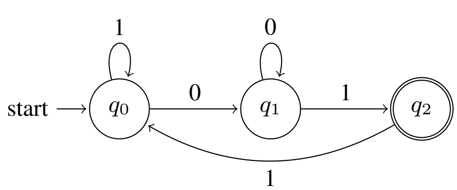
$$M = (Q, \Sigma, \delta, q_0, F)$$

where

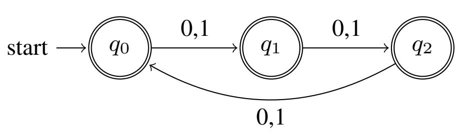
- Q: a finite set of states
- Σ: a finite set of input symbols (or input alphabet)
- $\delta \in Q \times \Sigma \to Q$: a total function called transition function
- $q_0 \in Q$: the initial state
- $F \subseteq Q$: a set of final states

1. (10pts) According to the definition, are the following automata (represented by transition graphs) valid DFAs? For each automaton, answer with yes or no, and justify your answer.

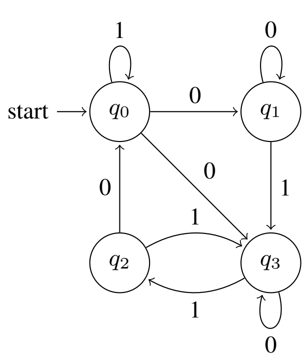
(a)



(b)



(c)



2. (5pts) We can extend the transition function $\delta : Q \times \Sigma \to Q$ to the extedned transition function $\delta^* : Q \times \Sigma^* \to Q$. Given a state $q \in Q$ and a string $w \in \Sigma^*$, $\delta^*(q, w)$ computes the state that the automaton reaches when starting from state q and consuming the string w. Fill in the definition of δ^*:

$$\begin{aligned} \delta^*(q, \lambda) &= \\ \delta^*(q, wa) &= \end{aligned}$$

3. (5pts) Consider the following transition table for a DFA:

	0	1
q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

where q_0 is the initial state. Compute $\delta^*(q_0, 10101)$ *according to its definition*. Show its full derivation steps.

4. (5pts) The language of DFA M, denoted $L(M)$, is defined as the set of all accepted strings. Fill in the definition of $L(M)$:

$$L(M) = \{w \in \Sigma^* \mid \qquad\qquad\qquad\qquad \}$$

Problem 2. (10pts) Given an NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ (without λ-transitions), we can construct an equivalent DFA D as follows:

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

where Q_D, δ_D, and F_D are defined below. Complete the definitions of them:

1. (2pts) $Q_D =$
2. (3pts) $F_D =$
3. (5pts) For each $S \in Q_D$ and input symbol $a \in \Sigma$:

   $$\delta_D(S, a) =$$

Problem 3. (10pts) Convert the following NFA to an equivalent DFA:

	0	1
p	$\{p, q\}$	$\{p\}$
q	$\{r, s\}$	$\{t\}$
r	$\{p, r\}$	$\{t\}$
s	$\emptyset$	$\emptyset$
t	$\emptyset$	$\emptyset$

where p is the initial state and s is the final state. What strings does the automaton accept?

Problem 4. (5pts) Consider the syntax of regular expressions:

$$R \to \emptyset \mid \lambda \mid a \mid R_1 + R_2 \mid R_1 R_2 \mid R^* \mid R^+ \mid R? \mid (R)$$

Complete the following semantic definition of the regular expressions:

$$\begin{array}{rcl} L(\emptyset) & = & \emptyset \\ L(\lambda) & = & \\ L(a) & = & \{a\} \\ L(R_1 + R_2) & = & \\ L(R_1 R_2) & = & \\ L(R^*) & = & (L(R))^* \\ L(R^+) & = & \\ L(R?) & = & \\ L((R)) & = & L(R) \end{array}$$

where the operator $+$ means "one or more of" and the operator ? means "zero or one of".

Problem 5. (20pts) Consider the following language:

$$L = \{a^n b^m \mid n \text{ is even, } m \text{ is odd, and } n, m \geq 0\}$$

1. (10pts) Is it a context-free language? If so, give a context-free grammar that describes the language. Otherwise, explain why not.
2. (10pts) Is it a regular language? If so, provide a finite automata or regular expression for the language. Otherwise, prove it not to be regular using the following pumping lemma:

> For any regular language L there exists an integer n, such that for all $x \in L$ with $|x| \geq n$, there exist $u, v, w \in \Sigma^*$, such that
>
> (a) $x = uvw$
> (b) $|uv| \leq n$
> (c) $|v| \geq 1$
> (d) for all $i \geq 0$, $uv^i w \in L$.

Problem 6. (20pts) Consider the following language:

$$L = \{w \in \{0, 1\}^* \mid w = w^R\}$$

1. (10pts) Is it a context-free language? If so, give a context-free grammar that describes the language. Otherwise, explain why not.
2. (10pts) Is it a regular language? If so, provide a finite automata or regular expression for the language. Otherwise, prove it not to be regular using the pumping lemma.

Problem 7. (10pts) True/false questions:

1. Every context-free language is regular.
2. Every regular language is context-free.
3. Every NFA can be transformed into an equivalent DFA.
4. Every DFA can be transformed into an equivalent NFA.
5. There is some language that can be specified by a regular expression but not by a finite automaton.
6. If a language L satisfies the pumping lemma, then L is regular.
7. The syntax of modern programming languages (e.g., C, Java, ML) can be expressed by a regular expression.
8. A language L is accepted by some λ-NFA if and only if L is accepted by some DFA.
9. There exists a language that is not regular but satisfies the pumping lemma.
10. The concatenation of two regular languages is regular.