CS208 Mid-semester Exam (Spring 2012)

Time: 120 mins **Date: Feb 23, 2012**

- ***The exam is open-book, open-notes, open-material-brought-to-exam0hall.***
- ***Be brief, complete and stick to what has been asked. If needed, you may cite results/proofs covered in class without reproducing them.***
- ***Do not copy solutions from others***
- ***Penalty for copying: FR grade (and you know we mean it!)***

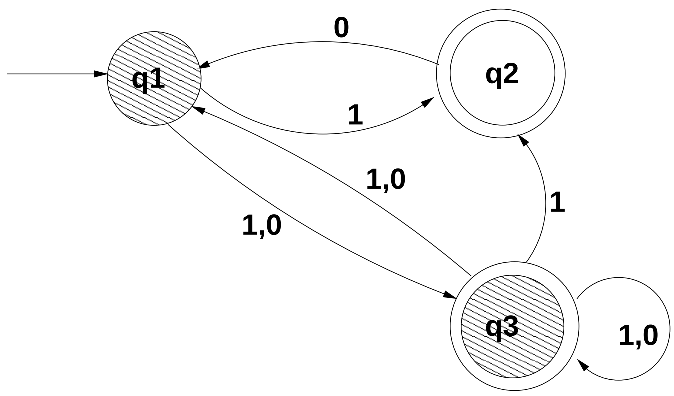

Figure 1: A special kind of non-deterministic automaton

1. Consider the NFA $A = (Q, \Sigma, \delta, Q_0, F)$ shown in Figure 1; here, $\Sigma = \{0, 1\}$. A *run* of A on a word $a_0 a_1 \ldots a_k \in \Sigma^*$ is a sequence of states $(q_0 q_1 \ldots q_{k+1})$ such that $q_0 \in Q_0$ and $q_{i+1} \in \delta(q_i, a_i)$ for $0 \leq i \leq k$. The length of the run is the length of the sequence $(q_0 q_1 \ldots q_{k+1})$, i.e, $k + 2$. Normally, we say that the word $w = a_0 a_1 \ldots a_k$ is accepted by A if there exists ***at least one*** run of A on w of length $k + 2$ such that $q_{k+1} \in F$.

   For the NFA in this question, however, we will define a new notion of acceptance. Note that in Figure 1, some states of the NFA have been shown shaded, while others have been left unshaded. Thus, the set Q of states of A can be thought of as partitioned into two subsets, Q_0 (representing shaded states) and Q_1 (representing unshaded states). We will say that a word $w = a_0 a_1 \ldots a_k \in \Sigma^*$ is accepted by A if and only if ***every*** run of A on w of length $k+2$ includes ***at least one*** state from Q_0 and ***at most one*** state from Q_1. Note that to decide whether a word w is accepted by A, we must now reason about every run of A on w, but we no longer require that a run must end in a final state. Let L' be the set of words accepted by A in the above manner.

Show that L' is regular by giving an NFA/DFA for L'. You must draw your NFA in the space provided below, and provide brief explanation of what each state in your NFA represents in the space separately provided for this.

[**Space for NFA/DFA: Please draw clearly. Untidy work will incur marks penalty**]

[Space for your justification]

2. Consider the following version of the pumping lemma for regular languages (let's call it PL for convenience) studied in class:
[PL]: "For every regular language L, there exists a constant $k > 0$ (pumping constant for L), such that for every word $w \in L$ with $|w| \geq k$, it is possible to decompose w as $w = x \circ y \circ z$ such that $|x \circ y| \leq k$, $|y| \geq 1$ and $x \circ y^i \circ z \in L$ for all $i \geq 0$."

For each of the following languages over $\Sigma = \{0, 1\}$, you are required to decide whether it satisfies PL or not, and justify your answer. Note that you are not being asked whether the language is regular or not. Simply saying whether the language is regular or not will, therefore, fetch no marks.

- If you think the language satisfies PL, you must provide the following in support of your answer:
  (a) The pumping constant k for L.
  (b) A decomposition $x \circ y \circ z$ for every word $w \in L$ with $|w| \geq k$, such that the conditions of PL are satisfied. Note that this may require you to specify separate decompositions for different words in L.
- If you think the language violates PL, you must provide the following in support of your answer:
  (a) A string w of length $\geq k$ for an arbitrary $k > 0$.
  (b) A value of i for each possible decomposition $x \circ y \circ z$ of the above w such that $|x \circ y| \leq k$, $|y| \geq 1$ and $x \circ y^i \circ z \notin L$. Note that this may require you to specify separate values of i for different decompositions of w.

In the following, we will use $\#0(w)$ and $\#1(w)$ to denote the count of 0's and 1's, respectively, in a word $w \in \Sigma^*$.

(a) $L = \{w \mid \#0(w) < \#1(w)\} \cup \{w \mid w \text{ has a subword } 00\}$, where a subword of w is a sequence of consecutive letters in w.

L satisfies PL	L violates PL
$k =$	w (as a function of k):
x, y, z for given w	i for each decomposition of w

(b) $L = \{0^i 1^j \mid i = 0 \text{ or } j \text{ is a prime no.}\}$. A prime no. is a natural no. > 1 that has only 1 and itself as divisors.

L satisfies PL	L violates PL
$k =$	w (as a function of k):
x, y, z for given w	i for each decomposition of w

(c) $l = \{w \mid \#0(w) = 0 \text{ or } \#1(w) = 2^i \text{ for some } i > 0\}$.

L satisfies PL	L violates PL
$k =$	w (as a function of k):
x, y, z for given w	i for each decomposition of w

3. Consider the NFA A over $\Sigma = \{a, b\}$, shown in Fig. 2. We wish to apply an adaptation of the DFA minimization algorithm studied in class to this NFA.

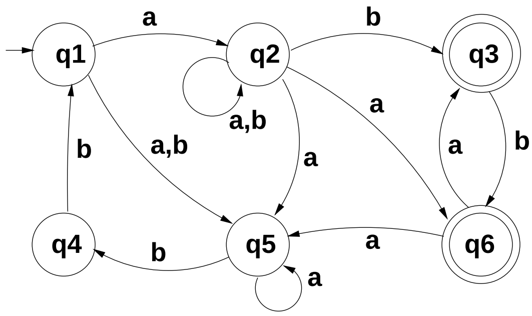

Figure 2: A non-deterministic automaton

We will first identify every pair of states in A that are *distinguishable*. We say that state q_i is distinguishable from q_j if and only if there exists a word $w \in \Sigma^*$ such that A started from q_i accepts w while A started from q_j doesn't accept w, or vice versa. Such a word w is said to distinguish q_i from q_j. When a pair of states is distinguishable, we also wish to record a string that distinguishes them. Note that there might be several strings that distinguish a pair of states; we wish to record only one of them.

(a) Fill in the following table as follows. The cell in row i and column j should contain $w \in \Sigma^*$ iff q_i is distinguishable from q_j, and w is a string distinguishing them. If q_i is not distinguishable from q_j, leave the corresponding cell blank. Note that filling in this table requires more work than in the corresponding problem for DFAs. You need only fill in the lower half triangle of the table.

	q_1	q_2	q_3	q_4	q_5	q_6
q_1						
q_2						
q_3						
q_4						
q_5						
q_6						

(b) Draw the NFA that results from merging all pairs of non-distinguishable states obtained from the table above. When you merge two states, all transitions coming into/going out of either state in the original NFA must come into/go out of the merged state in the merged automaton.

(c) Is the NFA obtained in the previous subquestion the smallest NFA for the corresponding language? If so, give brief justification. Otherwise, remove at least one state from the NFA obtained in the the previous subquestion, such that the resulting smaller NFA still accepts the same language.

4. The sub-questions in this question pertain to regular expressions over the alphabet $\Sigma = \{0, 1\}$.

   (a) Give a DFA (not an NFA) that recognizes the language represented by the regular expression $(\mathbf{0} + ((\mathbf{0} \circ \mathbf{1}) + ((\mathbf{0} \circ \mathbf{1} \circ \mathbf{1})^*)^*)^*$.

(b) The *star-height* of a regular expression $\mathbf{r}$, denoted $\mathsf{SH}(\mathbf{r})$, is a function from regular expressions to natural numbers. It is defined inductively as follows:

- $\mathsf{SH}(\mathbf{0}) = \mathsf{SH}(\mathbf{0}) = \mathsf{SH}(\varepsilon) = \mathsf{SH}(\mathbf{\Phi}) = 0.$
- $\mathsf{SH}(\mathbf{r_1} + \mathbf{r_2}) = \mathsf{SH}(\mathbf{r_1} \circ \mathbf{r_2}) = \max(\mathsf{SH}(\mathbf{r_1}), \mathsf{SH}(\mathbf{r_2}))$
- $\mathsf{SH}(\mathbf{r}^*) = \mathsf{SH}(\mathbf{r}) + 1$

Give a regular expression $\mathbf{r}$ over $\Sigma = \{0, 1\}$ such that the following hold:

- $\mathsf{SH}(\mathbf{r}) > 0$, and
- Every regular expression with star-height $< \mathsf{SH}(\mathbf{r})$ represents a language different from that represented by $\mathbf{r}$.

You must give brief justification why no regular expression with lesser star-height can represent the same language.