Review Sheet for Midterm of CS375

(Midterm - 3/9/2016 (Wednesday 10:00 am))

Spring, 2016
1. Suppose we are given the following NFA over the alphabet \{a, b\} (5 points):

(a) Find a regular expression for the language accepted by the NFA.

\[ a^* + b^* \quad \text{or} \quad a^*b^* \quad \text{or} \quad (a+b)^* \]

(b) Write down the transition table for the NFA.
Suppose we are given the following NFA over the alphabet \{a, b\} (5 points):

\(\Lambda(0) = \{0,1,2,3,4\}, \quad \Lambda(1) = \{1,2,3,4\}, \quad \Lambda(2) = \{2,3,4\}, \ldots\)

(c) Use (11.8) (slide 7 of my notes: Regular Language & Finite Automata III) to transform the NFA into a DFA.
2. Transform each of the following NFAs into a DFA (5 points).

(a) \( T(0, a) = \{0, 1\} \), where 0 is start and 1 is final.

(b) \( T(0, a) = \{1, 2\}, T(1, b) = \{1, 2\} \), where 0 is start and 2 is final.

In (a), assume alphabet = \{a\}

In (b), alphabet = \{a, b\}
3. Let the set of states for a DFA be $S = \{0, 1, 2, 3, 4, 5\}$, where the start state is 0 and the final states are 2 and 5. Let the equivalence relation on $S$ for a minimum-state DFA be generated by the following set of equivalent pairs of states: $\{\{0, 1\}, \{0, 4\}, \{1, 4\}, \{2, 5\}\}$. Write down the states of the minimum-state DFA (5 points).
4. For given regular expression, start by writing down the NFA obtained by algorithm (11.7) (slide 23 of my notes: Regular Language & Finite Automata II). Then use (11.8) to transform NFA into a DFA. Then use (11.10) (slides 21-28 of my notes: Regular Language & Finite Automata III) to find the minimum-state DFA: \((a + b)^*a\) (5 points).
Regular Grammar

5. Find a regular grammar to describe the following languages (5 points):
   (a) \{a, aaa, aaaaa, ..., a**(2n+1), ... \}.
   (b) \{Λ, a, b, c, aa, bb, cc, ..., a**n, b**n, c**n, ...\}.

6. Any regular language can also be defined by a grammar with productions of the following form, where \(w\) is a nonempty string of terminals: \(S \rightarrow \Lambda\), \(S \rightarrow w\), \(S \rightarrow T\) or \(S \rightarrow wT\). Find a grammar of this form for the language of the given regular expression: \((ab)^*a\) (5 points).
7. Show that the given language is not regular by using the Pumping Lemma (11.13) (slide 20 of my notes: Regular Language & Finite Automata IV):
\[ \{ a^n b^k \mid n, k \in \mathbb{N} \land n \geq k \}. \]

Why the language has to be infinite for us to use the Pumping Lemma?
Use (11.12) (slide 17 of my notes: Regular Language & Finite Automata IV) to construct an NFA to recognize the language of the given regular grammar: \( S \rightarrow aI|bJ, \ I \rightarrow bI|\Lambda, \ J \rightarrow aJ|\Lambda \)
1. What is a regular expression? Know how to find a regular expression for a given language. Know how to do 4(a),(b) on page 703.
(Find a regular expression for each of the following languages over the alphabet \{a, b\}:
(a) strings with even length,
(b) strings whose length is a multiple of 3)

2. Know how to simplify a regular expression. Know how to do 5(a), (b) on page 703.
3. Know how to prove some important properties of regular expressions such as

\[
\begin{align*}
(a) \quad R^* &= \Lambda + R^* = (\Lambda + R)^* \\
(b) \quad R^* &= \Lambda + R + \ldots + R^{k-1} + R^k R^*, \quad k \geq 1
\end{align*}
\]

on page 703.

4. Know how to transform a regular expression to a DFA or an NFA. Know how to do 2(d), (e) on page 726 and 6(b) on page 727.

5. Know how to do 8(a),(b) on page 727.
7. Know how to **transform** an NFA to a DFA. Know how to do 3(c) on page 743 and 4(b) on page 744.

8. Know how to **transform** a given DFA to a minimum-state DFA. But most importantly, know how to compute the states of the minimum-state DFA.
End