CSC 320 Summer 2017: Assignment #2
Due at beginning of class, Fri. June 2

Draw boxes for your marks. Place a 0 in the corresponding box for any questions you omit.

Question	1	2	3	4	5	6
Marks						

It is not the job of the TA to verify complicated regular expressions or automata and either find a counterexample for you or prove that your solution is correct. To get marks for the questions, you must justify the correctness of your solutions.

Questions #1 and #2 refer to the following languages:

(a) $\{w \in \{a,b\}^* : w \textit{ contains abaabb}\}$

(b) $\{w \in \{a,b\}^* : w \textit{ has babab as a prefix, and abab as a suffix}\}$

(c) $\{w \in \{0,1\}^* :$ *the number of 0's is even and the number of 1's is odd* $\}$

(d) $\{w \in \{a,b\}^* : w \textit{ has both bbabab and abababb as substrings}\}$

(e) $\{w \in \{0,1\}^* : w = 1^r\, 0\, 1^s\, 0 \textit{ for some } r \geq 2 \textit{ and } s \geq 3\}$

1. [20] Prove that any four of the above languages are regular by giving regular expressions for them. Clearly indicate the part you choose to omit.

2. [20] Prove that any four of the above languages are regular by designing DFA's for them. Clearly indicate the part you choose to omit.

3. [15] Let M be a NDFA, $M = (\{q_1, q_2, q_3, q_4, q_5\}, \{a, b\}, \Delta, \{q_1\}, \{q_5\})$, where $\Delta = \{(q_1, \varepsilon, q_2), (q_1, a, q_2), (q_2, \varepsilon, q_3), (q_2, a, q_3), (q_3, a, q_4), (q_4, b, q_5), (q_5, \varepsilon, q_1), (q_5, b, q_1), (q_5, b, q_5)\}$. Use the algorithm in the text (which is the same as the one shown in class) to convert this NDFA to an equivalent DFA. (It would probably help to draw the state diagram first).

4. Suppose you are given two DFA's $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ and $M_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Your aim is to construct a DFA $M = (K, \Sigma, \delta, s, F)$ which accepts the intersection of $L(M_1)$ and $L(M_2)$. Each state of M corresponds to a pair in $K_1 \times K_2$ and the significance of being in the state labelled (q, r) with q in K_1 and r in K_2 is that M_1 would be in state q and M_2 would be in state r after the same input.

(a) [2] Which state should be the start state for M?

(b) [2] Which states should be final states?

(c) [4] Define the transition function δ of M in terms of δ_1 and δ_2.

(d) [4] Apply your construction to the following two DFA's.
$M_1 = (\{q_0, q_1\}, \{a, b\}, \delta_1, q_0, \{q_1\})$, where
$\delta_1 = \{(q_0, a, q_0), (q_0, b, q_1), (q_1, a, q_0), (q_1, b, q_1)\}$.
$M_2 = (\{r_0, r_1, r_2\}, \{a, b\}, \delta_2, r_0, \{r_0, r_1\})$, where
$\delta_2 = \{(r_0, a, r_0), (r_0, b, r_1), (r_1, a, r_0), (r_1, b, r_2), (r_2, a, r_2), (r_2, b, r_0)\}$.

(e) [3] $L_1 \oplus L_2 = \{w\colon w \in L_1 \text{ or } w \in L_2 \text{ but } w \text{ is not in both } L_1 \text{ and } L_2\}$. How could you modify this construction to prove that regular languages are closed under EXCLUSIVE OR ($\oplus$)?

5. Let P be a finite set of ordered pairs of nonempty strings over some alphabet Σ. A *match* of P is any nonempty string w such that for some $n > 0$ and some (not necessarily distinct) pairs $(u_1, v_1)(u_2, v_2) \cdots (u_n, v_n)$ in P, $w = u_1 u_2 \cdots u_n = v_1 v_2 \cdots v_n$.
Example: If $P = \{(a, ab), (b, ca), (ca, a), (abc, c)\}$, then *abcaaabc* is a match of P since if the following five pairs are chosen, the concatenation of the first components and the concatenation of the second components are both equal to *abcaaabc*:
$(a, ab)(b, ca)(ca, a)(a, ab)(abc, c)$.

(a) [4] Prove that this system has a match:
$P = \{(b, aa), (bba, a), (b, bbbb), (aaa, b)\}$.

(b) [4] Prove that the following system has no match:
$P = \{(a, cab), (abc, a), (abc, ab), (ab, ba), (c, bcca)\}$

(c) [4] Suppose you are given an algorithm that decides whether or not a system P defined over $\Sigma_1 = \{0, 1\}$ has a match. Describe how you could use this algorithm to determine whether a system Q defined over $\Sigma_2 = \{a_1, a_2, \cdots, a_k\}$ has a match.

(d) [4] Show how you apply your tactic from part (c) to the system $P = \{(aabc, a), (fd, d), (f, abc), (d, ab)\}$ defined over $\Sigma = \{a, b, c, d, f\}$.

(e) [4] Define the *length* of a system P, denoted by $|P|$, where $P = \{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$ to be $\sum_{i=1}^{n} (\,|u_i| + |v_i|\,)$. Suppose the algorithm described in step (c) takes time $O(f(|P|)$. Give an upper bound on the time required to solve the arbitrary problem Q (based on your answer to part (c)).

Thought question: Design an algorithm to solve the problem described in 5(c) above.

6. The learning objective for this question is for students to understand reductions. In a reduction, you assume you have an algorithm for one problem, problem A. The goal is to use

this algorithm as a black box to solve another problem, problem B. For an arbitrary reduction, it suffices to prove that if there is an algorithm for A then there is one for B. For an NP-completeness reduction, you want to show that if there is a polynomial time algorithm for A, then there is a polynomial time algorithm for B.

Suppose you are given a boolean function *IsEmpty(M):*

Input: A DFA M

Returns: *true* if $L(M) = \phi$ and *false* otherwise.

(a) [4] Describe a construction which given a DFA $M_1 = (K_1, \Sigma, \delta_1, s_1, F_1)$ and a string $u = \sigma_1 \sigma_2 \cdots \sigma_k$ yields a DFA $M_2 = (K_2, \Sigma, \Delta_2, s_2, F_2)$ so that if you call isEmpty(M_2) it returns the answer to the question: ‘‘Does M_1 accept any strings which have u as a prefix?’’

(b) [2] Give a regular expression for the language that the DFA $M_1 = (\{s, p, q, r\}, \{a, b\}, \delta_1, s, \{s, r\})$ accepts where δ_1 is given in this table:

State	Symbol	Next State
s	a	p
s	b	q
p	a	r
p	b	p
q	a	q
q	b	r
r	a	s
r	b	s

(c) [4] Show how to apply your construction from part (a) to the DFA M_1 from part (b) and the string $u = a\ b\ a\ b\ a$ and draw a picure of the resulting DFA M_2.

Bonus: [10] Do questions #1 and #2 for all five languages.