Mid I Part II **Theory of Computation** **Total: 15 Marks**

Instructor's Solution ______________________________

1. Is the language L consisting of all binary strings of length greater than or equal to 100 regular? Answer yes/no and then justify. If L is regular, how many states are required for a minimal DFA to accept L? Justify your answer. $\boxed{3}$

   Soln: $\overline{L}$ consists of all (binary) strings of length at most 100. This is a finite set and can be accepted by a DFA with 101 states (how?). Since regular languages are closed under complementation, L must be regular. It is easy to see that L can be accepted by a DFA with the same number of states (why?). Finally, if x and y and strings of length $0 \leq i < j \leq 100$ respectively, then $(x, y) \notin R_L$. (why?). Thus by Myhill Nerode theorem, at least 101 states are necessary.

2. Let M_1 and M_2 be finite state machines over $\{0, 1\}$ with state sets Q_1, Q_2 and F_1, F_2 as the set of final states. Suppose you want the product automation $M_1 \times M_2$ to accept $L(M_1) \cup L(M_2)$, what must be the final states? Justify your answer in one sentence. $\boxed{3}$

   Soln: The product machine must accept a string if one of the machines reach a final state. Hence $F = \{(q, q')|q \in F_1 \vee q' \in F_2\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

3. Consider the language L of all binary strings which has no "zeroes" before the occurance of the first "one". How many Myhill Nerode equivalance classes are there for the language? Write regular expressions for specifiying strings in each equivalance class. $\boxed{3}$

   Soln: It is easy to see that ϵ, 0 and 1 must belong to different Myhill-Nerode classes. All strings starting with a 0 are not in the language and are Myhill Nerode equivalent to 0 (why?). Similarly, all strings starting with a 1 are equivalent to 1 (why?). Thus $[0] = 0(1 + 0)^*$. $[1] = 1(1 + 0)^*$. and $[\epsilon] = \{\epsilon\}$.

4. Why is it impossible to prove that the language $L = \{a^i b^j c^k : j = k \text{ if } i = 1\}$ non-regular using the pumping lemma? Is the language actually non-regular? Justify your answer. $\boxed{3+3}$

   Soln: Given any number $n > 0$ and any string w in the language with $|w| \geq n$. We will show that we can find x, y, z such that $w = xyz$, $|xy| \leq n$, $|y| \geq 1$ and $xy^i z \in L$ for all $i \geq 0$. There are several possibilities for w to be analyzed:

   1. $w = ab^t c^t$ for some t. Set $x = \epsilon$, $y = a$ and $z = b^t c^t$.
   2. $w = a^k b^l c^m$ for some l, m, n with $k \geq 2$. Here pick $x = a^{k-2}$, $y = a^2$ and $z = b^l c^m$
   3. $w = b^l c^m$ for some l, m. set $x = \epsilon$, set y to be the first symbol in w and z to be the rest of w.

   (You must convince yourself that these choices of x, y, z suffices.)

   Thus, the language satisfies all conditions of the pumping lemma. This means, we can never derive a contradiction to pumping lemma starting with any n and $w \in L$ with $|w| \geq n$.

   Yet, the language is not regular. This can be proved as follows: Consider the string $w_i = ab^i$. If $i \neq j$, $(w_i, w_j) \notin R_L$ (why?). Thus there are infinitely many Myhill Nerode equivalance classes.