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

1. Let $PAL = \{ww^R : w \in \{a,b\}^*\}$, the language of even palindromes. Consider the argument: 1. Let $PAL \cap a^*bb^*a^* = L$. Then $L = \{a^n b^{2m} a^n, n \geq 0, m > 0\}$. 2. PAL is a CFL but not a DCFL. 3. Thus L is the intersection of CFL that is not a DCFL, with a regular language. 4. The intersection of a DCFL with a regular language is a DCFL. 5. Since PAL is not a DCFL, L is not a DCFL. $\boxed{2+2}$

   a) Which among the steps 1,2,3,4,5 are logically or factually flawed? What are the flaws? *Soln:* Steps 1, 2, 3, 4 are correct. But nothing rules out the possibility that the intersection of non-DCFL with a regular language could still be a DCFL. Hence step 5 does not logically follow from 1,2,3 and 4.

   b) Though the argument above is wrong, is the conclusion that L is not a DCFL actually true (provable in some other way)? Justify your answer. *Soln:* L is a DCFL. A DPDA can push first a^s in the stack, then pop out each a when a b is read, pushing remaining b^s in the stack and matching them with c^s.

2. Let $M_1 = (Q_1, \{a,b\}, \{A,B,\$\}, \delta_1, s_1, F_1)$. be a PDA and and $M_2 = (Q_2, \{a,b\}, \delta_2, s_2, F_2)$ be a finite automation accepting languages L_1 and L_2 respectively. Let $q_1, p_1 \in Q_1$ and $q_2, p_2 \in Q_2$. Let $\delta_1(q_1, \epsilon, A) = \{(p_1, AA)\}$, $\delta_1(q_1, a, A) = \delta_1(q_1, b, A) = \{(p_1, BA)\}$ and $\delta_2(q_2, a) = \delta_2(q_2, b) = p_2$. M_2 has no epsilon transitions. Consider the product automata $M = (Q_1 \times Q_2, \{a,b\}, \{A,B,\$\}, \delta, (s_1, s_2), F_1 \times F_2)$ accepting $L_1 \cap L_2$. $\boxed{1\text{x}4}$

   a) Define $\delta((q_1, q_2), \epsilon, A)$. Justify your choice. *Soln:* $\delta((q_1, q_2), \epsilon, A) = \{(p_1, q_2), AA\}$. M_1 moves according to its transition and M_2 does not change state as no input is read.

   b) Suppose M_2 is a DFA and M_1 has the property that $\delta_1(q_1, c, X)$ contains exactly one element for each $c \in \{a,b\}$, $X \in \{A,B,\$\}$, then will M be deterministic? Justify your answer. *Soln:* M may be non-deterministic because $\delta_1(q_1, \epsilon, X)$ may not be empty.

   c) M accepts input w if and only if $((s_1, s_2), w, \$) \vdash_M^* ((f_1, f_2), \epsilon, \gamma\$)$ for some $f_1 \in F_1$, $f_2 \in F_2$ and $\gamma \in \Gamma^*$

   d) Which (all) statements are true among the following? Indicate the statement numbers. 1) If M_1 and M_2 are deterministic then it could be true that M is non-deterministic. 2. M may be deterministic even though M_1 is non-deterministic. 3) M_1 is deterministic, M_2 is non-deterministic, but M is deterministic. 4) None of the (1),(2),(3) can happen. Justify your answer. *Soln:* None of them can happen. the product machine will be deterministic if and only if both M_1 and M_2 are deterministic.

3. Recall that we proved in the class that the language $L = \{ww : w \in \{a,b,c\}^*\}$ cannot be accepted by any single tape turing machine in $\Omega(n^2)$ (time) complexity in the worst case. Consider now the language $L' = \{ww^R : w \in \{a,b,c\}^*\}$, where w^R is the reverse of w. Define $L'_n = \{wc^{\frac{n}{2}}w^R : w \in \{a,b\}^{\frac{n}{4}}\}$ for n any positive multiple of 4. For $\frac{n}{4} < i \leq \frac{3n}{4}$ define the crossing sequence $c_i(z)$ for each string $z \in L'_n$ exactly the way we did in the class. $\boxed{2+1}$

   a) Suppose x, y are distinct strings in $\{a,b\}^{\frac{n}{4}}$ and $\frac{n}{4} < i, j \leq \frac{3n}{4}$. Is it true that $c_i(xc^{\frac{n}{2}}x^R) \neq c_j(yc^{\frac{n}{2}}y^R)$? Answer YES/NO and give brief proof/counter example. *Soln:* If $c_i(xc^{\frac{n}{2}}x^R) = c_j(yc^{\frac{n}{2}}y^R)$, if α is the sub-string consisting of symbols in positions 1 to i in the first string and β is the substring of symbols in positions $j+1$ to n of the second string, then $\alpha\beta$ will have to be accepted by the machine because the crossing sequences at i and j are the same. But $\alpha\beta$ does not belong to the language (why) and hence this is a contradiction. Hence, equality can't hold as was assumed.

   b) Will the same proof used in L work for proving an $\Omega(n^2)$ lower bound for the complexity of L'? Answer YES/NO. Justify your answer. *Soln:* The proof will go through without any modification.

4. Use Pumping Lemma (for context free languages) to prove that $L = \{a^p : p \text{ prime }\}$ is not context free. *Soln:* Suppose L is regular. Let k be the constant specified by the pumping lemma. Let p be a prime greater than $2k+2$. Since $a^p \in L$, pumping lemma asserts that we can find strings u, v, w, x, y such that $a^p = uvwxy$, $|vwx| \leq k$ and $|vx| \geq 1$ such that $uv^iwx^iy \in L$ for all $i \geq 0$. Let $|vx| = t$ and $|uwy| = s$. Note that $t \geq 1$ and $s > k > 1$ as $p > 2k+2$. By pumping lemma we have $a^{s+it} \in L$ for all values of L, or $s + it$ must be prime for all values of i. But this is impossible, for by setting $i = s$, we require $s(1+t)$ to be prime, which is a contradiction. $\boxed{4}$