	Graduate Course
THEORY OF COMPUTATION	College of Computer Science
Final Exam	Zhejiang University
Solution	Fall 2004

Class: __________ **ID:** __________ **Name:** __________ **Score:**__________

Time allowed: 2 Hours.

1. **(20%) Determine whether the following statements are true or false. If it is true write a ✓ otherwise a × in the bracket before the statement.**
   (a) (✓) Every context-free language is recursive.
   (b) (✓) Language $\{a^m b^n c^l \mid m, n, l \in \mathbb{N}, m + n > 3l\}$ is context free.
   (c) (✓) Every language in $\mathcal{NP}$ is recursive.
   (d) (×) All languages on an alphabet are recursively enumerable.
   (e) (×) There's a language L such that L is undecidable, yet L and its complement are both semi-decided by the some Turing machine.
   (f) (×) There's a function φ such that φ can be computed by some Turing machines, yet φ is not a primitive recursive function.
   (g) (×) Let $L_1, L_2 \subseteq \Sigma^*$ be languages, recursive function τ is a reduction from L_1 to L_2, if L_1 is decidable, then so is L_2.
   (h) (×) A language L is recursive if and only if it is Turing-enumerable.
   (i) (×) Suppose A, B are two languages and there is a polynomial-time reductions from A to B. If A is $\mathcal{NP}$-complete, then B is $\mathcal{NP}$-complete.
   (j) (✓) Every language in $\mathcal{NP}$-complete can be reducible to the 3-SAT problem in polynomial time.

2. **(20%) FA and regular languages:**
   (a) Decide whether the following language is regular or not and provide a formal proof for your answer.

$$L = \{a^m b^n \mid m, n \in \mathbb{N}, (m - n) \mod 3 \neq 0\}$$

(b) Let Σ be an alphabet and let $L_1, L_2 \subseteq \Sigma^*$ be languages so that L_1 is not regular but L_2 is regular. Assume $L_1 \cap L_2$ is finite. Prove that $L_1 \cup L_2$ is not regular.

Solution: (a) L **is not regular.**
(b) Assume $L_1 \cap L_2$ is finite. Since every finite set is regular, $L_1 \cap L_2$ is regular. Observe that

$$L_1 = ((L_1 \cup L_2) - L_2) \cup (L_1 \cap L_2).$$

If $L_1 \cup L_2$ were regular, since the regular languages are closed under the operations union, intersection and complement and since L_2 and $L_1 \cap L_2$ are regular, L_1 would be regular, a contradiction. Therefore, $L_1 \cup L_2$ is not regular.

3. **(20%) PDA and Context-free languages:**

(a) Give a context-free grammar for the language

$$L_3 = \{xy \mid x, y \in \{a,b\}^*, |x| = |y| \text{ and } x \text{ and } y^R \text{ differ in one positions } \}.$$

For example, $abbbbaba, abbbbbbb \in L_3$, but $aababb \notin L_3$.

(b) Design a PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ accepting the language L_3.

Solution: (a) We can construct the context-free grammar $G = (V, \Sigma, R, S)$ for language L_3, where

$V = \{a, b, S, A, B\}; \Sigma = \{a, b\}$; and

$$R = \{S \to aSa, S \to bSb, S \to aAb, A \to aAa, A \to bAb, A \to e,$$

$$S \to bBa, B \to aBa, B \to bBb, B \to e\}$$

(b) The PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ is defined below:

$K = \underline{\{p, q\}}$

$\Sigma = \{a, b\}$

$\Gamma = \underline{\{a, b, S, A, B\}}$

$s = \underline{p}$

$F = \underline{\{q\}}$

(q, σ, β)	(p, γ)
(p, e, e)	(q, S)
(q, e, S)	(q, aSa)
(q, e, S)	(q, bSb)
(q, e, S)	(q, aAb)
(q, e, A)	(q, aAa)
(q, e, A)	(q, bAb)
(q, e, A)	(q, e)
(q, e, S)	(q, bBa)
(q, e, B)	(q, aBa)
(q, e, B)	(q, bBb)
(q, e, B)	(q, e)
(q, a, a)	(q, e)
(q, b, b)	(q, e)

4. **(10%) Numerical Functions:**
Let $P(x, y)$ be primitive recursive predicates. Prove the following predicate

$$\forall y_{\leq u} P(x, y)$$

is also primitive recursive.

solution:

$$\forall y_{\leq u} P(x, y) \Leftrightarrow \Pi_{y=0}^{u} P(x, y) \neq 0$$

5. **(10%) Turing Machines**

Design a Turing machine for computing the following function.

$$f(x, y) = \begin{cases} 2x + 1, & \text{if } y \text{ is even} \\ 4x, & \text{if } y \text{ is odd} \end{cases}$$

where x and y are represented by binary strings respectively and separated with the symbol ";", i.e. the initial configuration in form of $\triangleright \underline{\sqcup} x; y$.
Solution: See figure.

6. **(10%) Decidability and Undecidability**

Show that the following language

$$\{\text{"}M\text{"}\,\text{"}w\text{"} \mid M \text{ is a TM and } M \text{ halts on } w\}$$

is recursively enumerable. An informal description suffices.
Solution: The universal Turing machine U can semidecides the language

$$\{\text{"}M\text{"}\,\text{"}w\text{"} \mid M \text{ is a TM and } M \text{ halts on } w\}$$

.

7. **(10%)$\mathcal{P}$ and $\mathcal{NP}$ Problems**

Given n natural numbers $x_1, x_2, \cdots, x_n$ to test whether there exist distinct $i_1, i_2, \cdots, i_k$ such that $x_{i_1} + x_{i_2} + \cdots + x_{i_k} = (x_1 + x_2 + \cdots + x_n)/2$, and $x_{i_1} + x_{i_2} + \cdots + x_{i_k}$ is not a prime number. Design a $\mathcal{NP}$ algorithm for it, and estimate its time complexity.
Solution: First compute

$$H = \frac{1}{2} \sum_{i=1}^{n} x_i$$

in $\mathcal{O}(n)$ steps.
Then guess $i_1, i_2, \cdots, i_k$ for some $k \leq n$, this takes $\mathcal{O}(n)$ steps, and compute $S = x_{i_1} + x_{i_2} + \cdots + x_{i_k}$ then check $S = H$.
If so, guess a factor x of S and then divide S by x and verify that the remainder is 0. If all the tests succeed print "yes" and halt else simply halt.
observe that if the answer is "yes", our algorithm can print "yes" and if the answer is "no", our algorithm cannot print "yes".