浙江大学 2013-2014 学年 秋冬 学期

《计算理论》课程期末考试试卷答案

课程号：21120520　开课学院：计算机学院

考试试卷：☑ A卷　☐ B卷

考试形式：☑ 闭卷　☐ 开卷，允许带 ________入场

考试日期: 2014 年 1 月 15 日，考试时间：120 分钟

诚信考试，沉着应考，杜绝违纪

考生姓名________________ 学号________________ 所属院系________________

题序	1	2	3	4	5	6	总分
得分							
评卷人							

Zhejiang University
Theory of Computation, Fall-Winter 2013
Final Exam (Solution)

1. **(24%) Determine whether the following statements are true or false. If it is true fill a ◯ otherwise a × in the bracket before the statement.**

   (a) (◯) Language $\{a^m b^n c^j | m, n, j \in \mathbb{N} \text{ and } m + n + j \geq 2014\}$ is regular.

   (b) (×) Let L be a regular language, so is $\{ww^R | \ w \in \Sigma^* \text{ and } w \in L\}$.

   (c) (×) Let L_1 and L_2 be two languages. If L_1L_2 is regular, then either L_1 or L_2 is regular.

   (d) (◯) Let L be a context-free language, then L^* is also context-free.

   (e) (◯) Language $\{w_1\#w_2\#\cdots\#w_n | n \in \mathbb{N}, \text{ for each } i, w_i \in \{a,b\}^* \text{ and for some } i, w_i \text{ is a palindrome}\}$ is context-free.

   (f) (×) Let L be a context-free language, then so is $H(L) = \{x | \exists y \in \Sigma^*, |x| = |y| \text{ and } xy \in L\}$.

   (g) (×) Language $\{``M_1"``M_2" | M_1 \text{ and } M_2 \text{ are two finite automata, } L(M_1) \subseteq L(M_2)\}$ is undecidable.

   (h) (◯) There's a function φ such that φ can be computed by some Turing machines, yet φ is not a primitive recursive function.

   (i) (◯) If L_1, L_2, and L_3 are all recursively enumerable, then $L_1 \cap (L_2 \cup L_3)$ must be recursively enumerable.

   (j) (◯) Let L_1 and L_2 be two recursively enumerable languages. If $L_1 \cup L_2$ and $L_1 \cap L_2$ are recursive, then both L_1 and L_2 are recursive.

   (k) (◯) Let L be a recursively enumerable language and $L \leq_\tau \overline{H}$, then L is recursive, where $H = \{``M"``w" | \text{ Turing machine } M \text{ halts on } w\}$.

   (l) (◯) The set of undecidable languages is uncountable.

2. **(20%)** Decide whether the following languages are regular or not and provide a formal proof for your answer.
   (a) $L_1 = \{uvu^R | u, v \in \{a, b\}^+\}$
   (b) $L_2 = \{uvu | u, v \in \{a, b\}^+\}$

   where $L^+ = LL^*$.

   Solution:

   (a) L_1 **is regular.** $\cdots\cdots$ 5pt

   There is no reason to let u be more than one character. So all that is required is that the string have at least two characters and the first and last must be the same. $L = (a\{a \cup b\}\{a, b\}^* a) \cup (b\{a \cup b\}\{a, b\}^* b)$.

   $\cdots\cdots$ **5pt**

   (b) L_1 **is not regular.** $\cdots\cdots$ **5pt**

   Assume L_2 is regular, let n be the constant whose existence the pumping theorem guarantees.

   Let $w = a^n baa^n b$ that is $u = a^n b$ and $v = a$, so $w \in L_2$. So the pumping theorem must hold.

   – Let $w = xyz$ such that $|xy| \leq n$ and $y \neq e$, then $y = a^i$ where $i > 0$. But then $xy^2z = a^{n+i}baa^n b \notin L_2$.

   The theorem fails, therefore L_2 is not regular. $\cdots\cdots$ **5pt**

3. **(20%)** Let $L_3 = \{ab^m c^n a^{m+2n} c | m, n \in \mathbb{N}\}$.
   (a) Give a context-free grammar for the language L_3.
   (b) Design a PDA $M = (K, \Sigma, \Gamma, \Delta, s, F)$ accepting the language L_3.

   Solution: (a)

   (a) The CFG for L_3 is $G = (V, \Sigma, S, R)$, where $V = \{S, S_1, S_2, a, b, c\}$, $\Sigma = \{a, b, c\}$, and $\cdots\cdots$ **3pt**

   $$R = \{S \to aS_1c, S_1 \to bS_1a, S_1 \to S_2, S_2 \to cS_2a^2, S_2 \to e\}.$$

   $\cdots\cdots$ **7pt**

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

	(q, σ, β)	(p, γ)
$K = \{\underline{p, q}\}$	(p, e, e)	(q, S)
$\Sigma = \{a, b, c\}$	(q, e, S)	(q, aS_1c)
$\Gamma = \{\underline{S, S_1, S_2, a, b, c}\}$	(q, e, S_1)	(q, bS_1a)
$s = \underline{p}$	(q, e, S_1)	(q, S_2)
$F = \underline{\{q\}}$	(q, e, S_2)	(q, cS_2a^2)
	(q, e, S_2)	(q, e)
	(q, e, a)	(q, a)
	(q, e, b)	(q, b)
	(q, e, c)	(q, c)

   $\cdots\cdots$ **10pt**

4. **(12%)** Try to construct a Turing Machine to decide the following language

$$L = \{ww^R | w \in \{0,1\}^*\}.$$

Where w^R is the inverse of w. Always assume that the Turing machines start computation from the configuration $\triangleright\underline{\sqcup}w$. When describing the Turing machines, you can use the elementary Turing machines described in textbook.

Solution: We can design the following Turing Machine to decide L:

$$\begin{tikzcd}
{} \arrow[r] & R \arrow[r, "d\in\{0,1\}"'] \arrow[d, "\sqcup"] & \sqcup R_{\sqcup} L \arrow[r, "d\in\{0,1\}"'] \arrow[d, "\neq d"] & \sqcup L_{\sqcup} \arrow[ll, bend right] \\
& y & n &
\end{tikzcd}$$

$\cdots\cdots$ 12**pt**

5. **(12%)** Show that the function: $\varphi : \mathbb{N} \to \mathbb{N}$ given by

$$\varphi(x) = \begin{cases} x \mod 3, & \text{if } x \text{ is a composite number;} \\ x^2 + 1, & \text{otherwise.} \end{cases}$$

Solution: Since

$$\varphi(x) = \text{rem}(x, 3) \cdot (1 \sim \text{prime}(x)) + (x^2 + 1) \cdot \text{prime}(x)$$

and $\text{rem}(x, 3)$, $x^2 + 1$ are primitive recursive functions, $\text{prime}(x)$ is a primitive recursive predicate, hence $\varphi(x)$ is primitive recursive.

$\cdots\cdots$ 12**pt**

6. **(12%)** Consider the problem

$L_{even} = \{$"M"$|M$ is a TM and $L(M)$ contains at least one string of even number of b's$\}$

(a) Show that L_{even} is recursively enumerable.

(b) Show that L_{even} is non-recursive.

Solution:
(a) L_{even} is **recursively enumerable**.

We can use **the Universal Turing machine** U to simulate Turing M on string of even length. $\cdots\cdots$ 4**pt**

1. Do 1 step of M's computation on w_0
2. Do 2 steps of M's computation on w_0 and w_1
3. Do 3 steps of M's computation on w_0, w_1, w_2

$\cdots\cdots$
Here $w_0, w_1, \cdots$ is the lexicographic enumeration of $\sum^*$ and $w_0, w_1, \cdots$ are of even number of b's.

If the Universal Turing machine U discover the halting computation of both M on one input of even length then halts, otherwise U still simulate the computation of Turing machine M. $\cdots\cdots$ 4**pt**

(b) L_{even} is **non-recursive**. We will show this by reducing H to L_{even}. Since H is undecidable, it follows that L_{even} is undecidable. Assume there is a TM D that decides L_{even}. The Turing machine T_H deciding $H = \{``M" |$ Turing Machine halts on $e\}$.

Turing machine T_H as follows:

1. On input "M", We build the TM M_{even} as follows:
2. If $x \neq e$, reject; otherwise, Simulate M on e.
3. If M halts on e, then accept; if M does not halt on e , then reject.
4. Simulate D on "M_{even}".
5. If D accepts "M_{even}", accept; If D rejects "M_{even}", reject.

We know that if M halts on e, $L(M_{even}) = \{e\}$ and accepts at least one string of even length; Otherwise,if M halts on e, $L(M_{even}) = \emptyset$. Hence if M halts on e, D accepts "M_{even}"; Otherwise,if M halts on e, D accepts "M_{even}". Therefore, Turing machine T_H above decides H. But the halting language H is known to be undecidable, this is a contradiction. Thus our assumption that there was a machine D deciding M_{even} must have been incorrect. M_{even} is not recursive.

$\cdots\cdots$ 4**pt**