

选择题第二题未解决

1. Choose the correct answer and justify you answer.

- (1) (6%) Let $A = \{0^n 1^n 2^n | n \in \mathbb{N}\}$ and
- (a) $A \subseteq B$, then B is decidable.
 - (b) $B \subseteq A$, then B is decidable.
 - (c) $B \leq_\tau A$, where recursive function τ is a reduction from A to B , then B is decidable.

The correct answer is (). _____

选 (b) (a) 错在 B 是 A 的超集，不一定是可判定的。例如用 0、1、2 三个数字编码出停机问题。(c) 错在方向反了， A 归约到 B ，那么 B 可判定那么 A 可判定，可是 B 本身我们不知道。

- (2) (6%) Let A and B be any languages such that $A \leq_\tau B$. Under what conditions is it the case that $\overline{A} \leq \overline{B}$?
- (a) Only when both A and B are decidable.
 - (b) Only when both A and B are recursively enumerable.
 - (c) Always.

The correct answer is (). _____

选 (a) 递归语言对补封闭，因此 A 可以归约到 B ， A 的补当然可以归约到 B 的补，这个是显然的。如果 B 不是递归但是递归可枚举，而 A 是递归的，那么 (b) (c) 都不成立。

- (3) (6%) Just as we encoded Turing Machines as strings, we can also encode DFAs as strings. Let “ M ” be the encoding string of DFA M . Consider the following language $L_{DFA}^d = \{\text{“}M\text{”} | \text{“}M\text{”} \notin L(M)\}$. What can we say about L_{DFA}^d ?
- (a) L_{DFA}^d is regular.
 - (b) L_{DFA}^d is not regular but it is decidable.
 - (c) L_{DFA}^d is not recursively enumerable.

The correct answer is (). _____

选 (b) “ M ” $\notin L(M)$ ，说明 M 是一个自己不接受自己的编码的 DFA 的编码的集合。我们把那个 L_{DFA}^d 简写成 L ，否则我输入太麻烦了……构造图灵机 M^* 判定了 L ，首先 “ M^* ” $\notin L$ 。对于输入 “ M ”，我们只需要 M^* 去模拟 M 收到输入 “ M ” 的情况。因为是 DFA，所以 “ M ” 的字符总会读完，读完的时候是终结状态就接受，否则就拒绝。拒绝说明 “ M ” $\notin L(M)$ ， M^* 给出 yes，否则反之。因此就知道选 (b) 了。

至于为什么不是 regular 的，直观说，一个 DFA 给出的判定是这个 DFA 是否接受这个字符串，而不是收到的编码的原 DFA 是否能接受自己的编码，它本有能力模拟和判断别人，所以本有 DFA 来接受它。形式来说，就是对角化定理，他和每个 DFA 所能接受的语言都不一样，所以不是正则的。

(4) (6%) Just as we encoded Turing Machines as strings, we can also encode PDAs as strings. Let “ M ” be the string encoding of PDA M . Consider the following language $L_{PDA}^d = \{“M” \mid “M” \notin L(M)\}$. What can we say about L_{PDA}^d ?

- (a) L_{PDA}^d is decidable.
- (b) L_{PDA}^d is not decidable but it is recursively enumerable.
- (c) L_{PDA}^d is not recursively enumerable.

The correct answer is (). _____

只是 DFA 改成 PDA。不失一般性地我们假设这个 PDA 是非确定性的，然后只要一条路是通向终结状态并且栈最后变空了那么就算是 M 接受了 “ M ”。我们构造非确定图灵机 M^* 判定 L 。我们需要多条带，一条存着输入 “ M ”，一条模拟 PDA 的堆栈。然后给定输入 M^* 模拟 M 对输入 “ M ” 的操作。因为 M^* 也是非确定的，所以每次可以非确定使用 PDA 的规则。如果其中一次成功了那么 “ M ” $\in L(M)$ 然后拒绝这个输入，否则全部情况都不接受那么 M^* 就接受输入。但是，非确定的 PDA 的其中一些运算不一定可以停机，它完全可以在某条路上死循环给不出接受的答案。因此，这是个非递归的 co-r.e. 的问题，选择 (c)。

(5) (6%) Let A and B be disjoint, recursively enumerable languages. Further let $A \cup B$ also be recursively enumerable. What can you say about A and B ?

- (a) It is possible that neither A nor B is decidable.
- (b) At least one among A and B is decidable.
- (c) Both A and B are decidable.

The correct answer is (). _____

Disjoint 在这里应该是无不相交的意思。

(C) 设 M_1 半判定 A , M_2 半判定 B , M_3 半判定 $\neg (A \cup B)$, 然后这三个机器合起来就是判定 A 和 B 了。

2. (12%) Using the pumping theorem to show that

$$L_1 = \{w \in \{a, b\}^* \mid w \text{ has an equal number of } a's \text{ and } b's\}$$

is not regular.

解析：用泵定理证明。设 $L_2 = \{a^n b^n\}$, L_2 包含于 L_1 。给定整数 k , 考虑字符串 $w = a^k b^k$, $w \in L_2$, 我们可以把 w 重写为 $w = xyz$, 且 $|xy| \leq k$, 且 $y \neq \epsilon$, 即 $y = a^i$, $i > 0$. 但是 $xz = a^{k-i} b^k \notin L_2$, 与泵定理矛盾。所以 L_1 不是正则的。

3. (16%)

(a) Construct a context-free grammar that generates language

$$L_2 = \{a^m b^n c^k | m, n, k \in \mathbb{N}, \text{ and } m + n \leq k\}.$$

(b) Construct a pushdown automata that accepts language L_2 .

(a) $G = (V, \Sigma, R, S)$

$V = \{a, b, c, S, S1, S2, S3\}$

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

$R = \{$

$S \rightarrow S1$

$S1 \rightarrow aS1c$

$S1 \rightarrow S2$

$S2 \rightarrow bS2c$

$S2 \rightarrow S3$

$S3 \rightarrow S3c$

$S3 \rightarrow e$

$\}$

S 是起始符

(b) $M = (K, \Sigma, \Gamma, \Delta, p, F)$

$K = \{p, q\}$

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

$\Gamma = \{a, b, c, S1, S2, S3\}$

p 为初始状态

$F = \{q\}$

$\Delta = \{$

$(p, e, e) (q, s)$

$(q, e, s) (q, s_1)$

$(q, e, s_1) (q, a s_1 c)$

$(q, e, s_1) (q, s_2)$

$(q, e, s_2) (q, b s_2 c)$

$(q, e, s_2) (q, s_3)$

$(q, e, s_3) (q, s_3 c)$

$(q, e, s_3) (q, e).$

~~$(q, e, s) (q, e)$~~

$(q, a, a) (q, e)$

$(q, b, b) (q, e)$

$(q, c, c) (q, e).$

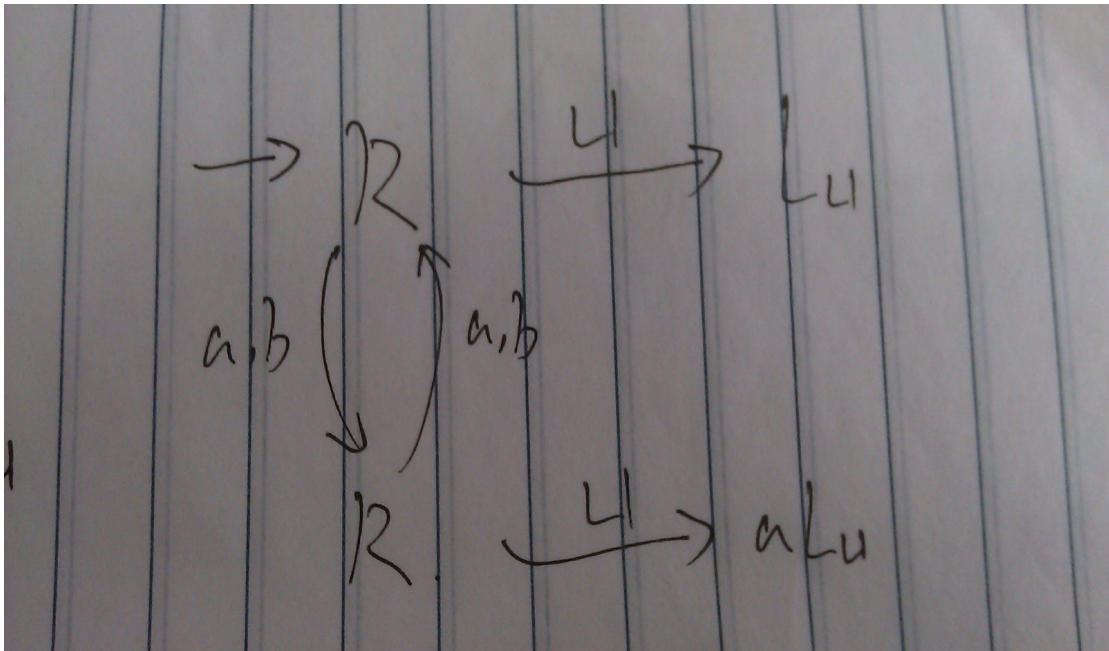
}

字迹潦草请见谅……

4. (12%) Construct a Turing machine that computes the function $f : \{a, b\}^* \rightarrow \{a, b\}^*$ given by

$$f(w) = \begin{cases} w, & \text{if the length of } w \text{ is even} \\ wa, & \text{if the length of } w \text{ is odd.} \end{cases}$$

When describing the Turing machines above, you can use the elementary Turing machines described in textbook. Always assume that the Turing machines start computation from the configuration $\triangleright \sqcup w$ where $w \in \{a, b\}^*$ is the input string.



5. (15%) Classify whether each of the following languages are recursive, recursively enumerable-but-not-recursive, or non-recursively enumerable. Prove your answers.

- (a) $L_4 = \{ "M" \mid \text{Turing machine } M \text{ halts on } "M" \};$
 (b) $L_5 = \{ "M" \mid \text{Turing machine } M \text{ does not halt on } "M" \}.$

(a) L_4 可用通用图灵机半判定，所以是递归可枚举的但不是递归的，详细说明见课本 164 页。

(b) L_5 不是递归可枚举的。可以用对角化原理证明，详细说明同见课本 164 页。

6. (15%)

- (a) Give the definition of \mathcal{NP} -Complete problem.
- (b) Describe carefully what the ingredients of an \mathcal{NP} -completeness proof are.
- (c) Consider the following problem, called the **MAX-SAT** problem: Given a set F of clauses, and an integer k , is there a truth assignment that satisfies at least k clauses? Show that **MAX SAT** problem is \mathcal{NP} -complete.

(a) 抄自定义 7.1.2, 设语言 L 是 Σ^* 的子集, 如果
(1) $L \in \text{NP}$, 并且
(2) 对每个语言 $L' \in \text{NP}$, 存在从 L' 到 L 的多项式归约,
那么 L 称为是 NP 完全的。

(b) 本人水平有限, 不知道 NPC 完全的证明需要什么 ingredient。个人认为方法有两种, 都是归约, 给定问题多项式时间归约到一个 NPC 问题, 那么这个问题也是 NPC , 或者一个 NPC 问题多项式时间归约到给定问题, 那么这个问题也是 NPC 。前者用定理, 后者用定义。

(c) 假设子句集 F 里面有个 n 个子句, $F = \{S_1, S_2, \dots, S_n\}$

我们任取其中 k 个子句来验证, 即验证 $\prod_{i=1}^k S_{ij}$, 我们设这样的合取的子句为 T_i , 那么总共有 C_n^k 个 T_i 。

所有这些 T_i 进行析取, 那么就成了满足题目所说的至少有 k 个子句被满足的布尔表达式, 那么, 要判定是否存在满足至少 k 个子句的真值复制, 就变成了一个 $\sum T_i$ (i 从 1 到 C_n^k) 的 SAT 问题, 而且归约的时间是多项式时间内的。因此, **MAX-SAT** 问题多项式时间归约到 SAT, 其是 NPC 的。 希望我表达清楚了喵 (>_<)