计算理论习题集

2022 年 12 月 3 日

以下习题主要来自于本校计算理论历年试卷，解答来自于标准答案，我收集到的答案以及我自己写的答案. 为保持一致，题目基本为英文. 如有错误，欢迎指正!

1 Finite Automata and Regular Language

1. Determine whether the following statements are true or false.

   (1) Infinite unions of regular sets ~~are~~ regular.

   (2) Language $\{a^{6n}b^{3m}c^{p+10} \mid n \geq 0, m \geq 0, p \geq 0\}$ is regular.

   (3) If L_1 and $L_1 \cup L_2$ are regular languages, then L_2 is a regular language.

   (4) Let A, B, C be three languages, and $A \subseteq B \subseteq C$. If both A and C are regular, then B is regular.

   (5) If A is regular and B is non-regular, then $A \circ B$ must be non-regular.

   (6) If A is non-regular and both B and $A \cap B$ are regular, then $A \cup B$ is non-regular.

   (7) Language $\{a^ib^jc^k \mid i, j, k \in \mathbb{N} \text{ and } i + j \not\equiv k \mod 3\}$ is not regular.

   (8) Let A and B be two regular languages, then $A \oplus B$ is also regular.

   (9) $\{w : w \text{ is a regular expression for } \{a^nb^m : n + \cancel{m} \leq 2007\}\}$ is a finite language.

   (10) If $L_1 \circ L_2$ is a regular language, then either L_1 or L_2 is regular.

解答:

(1) ✗. 注意到 $L = \{a^nb^n \mid n \geq 0\}$ 不是正则语言，但 $L = \{ab\} \cup \{aabb\} \cup \ldots$.

(2) ✓.

(3) ✗. 令 $L_1 = \sum^*$, 则 $L_1 \cup L_2 = \sum^*$.

(4) ✗. 令 $A = \sum^*, C = \emptyset$, 则 $A \subseteq B \subseteq C$ 恒成立.

(5) ×. 同上.

(6) ✓. 假设 $A \cup B$ 是正则语言, 那么由题设 $B, A \cap B, A \cup B$ 都是正则的.

由于 $A = (\overline{B} \cap (A \cup B) \cup (A \cap B))$, 而我们知道正则语言在交, 并, 补下都是封闭的, 说明 A 也是正则语言, 矛盾!

(7) ×. 在模运算下只有有限个情况.

(8) ✓. $A \oplus B = (A \cap \overline{B}) \cup (B \cap \overline{A})$.

(9) ×.

(10) ×. 我们只需要举出 L_1, L_2 都不正则, 但它们连接正则的例子, 这样的例子事实上是很多的. 令 L_1 为任一非正则语言, $L_2 = \overline{L_1}$, 显然 L_2 也不正则. 那么 $L_1 \cup \{e\}$ 和 $_2 \cup \{e\}$ 也不正则 (只改变有限元素). 然而 $(L_1 \cup \{e\}) \circ (L_2 \cup \{e\}) = \sum^*$, 是正则语言.

2. 写出以 ab 串结尾的语言 (字母表为 $\{a, b\}$) 的正则表达式, 画出 NFA, 转化成 DFA, 并得到最小化 DFA.

解答: 见讲义.

3. Say whether each of the following languages is regular or not (prove your answers):

(1) $L_1 = \{w \mid w \in \{a, b\}^* \text{ and } w \neq w^R\}$.

(2) $L_2 = \{wtw \mid w, t \in \{a, b\}^+\}$.

(3) $L_3 = \{wtw \mid w, t \in \{a, b\}^*\}$.

(4) $L_4 = \left\{uvu^R \mid u, v \in \{a, b\}^+\right\}$.

解答:

(1) 考虑 $L_1' = \{w \mid w = w^R\}$. Pumping Theorem. $w = a^n b a^n = xyz$, $xy^2z = a^{n+i} b a^n \notin L_1'$.

(2) Pumping Theorem. $w = a^n b a a^n b = xyz$, $xy^2z = a^{n+i} b a a^n b \notin L_2$.

(3) ✓. $w = e \implies \{a, b\}^* \subseteq L_3 \implies L_3 = \{a, b\}^*$.

(4) ✓. L_4 本质上识别的是该字符串首尾是不是相同的字符, 因为其他的多余字符都可以交给 v 来处理.

2 Context Free Language

1. Determine whether the following statements are true or false.

   (1) Suppose that L is context-free and R is regular, then $L - R$ is context-free language.

   (2) Every regular language can be generated by context-free grammar.

   (3) A and B are two context-free languages, so is $A \oplus B$, where $A \oplus B = (A - B) \cup (B - A)$.

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

   (5) Language$\{xcy \mid x, y \in \{a,b\}^*, |x| \le |y| \le 3|x|\}$ is context-free.

解答:

(1) ✓. $L - R = L \cap \overline{R}$.

(2) ✓.

(3) ✗.

(4) ✗.

(5) ✓.

2. Let $L = \{ab^m c^n a^{m+2n} c \mid m, n \in \mathbb{N}\}$.

   (1) Give a context-free grammar for the language L.

   (2) Design a PDA $M = (K, \sum, \Gamma, \Delta, s, F)$ accepts the language.

解答:

(1) $G = (V, \sum, S, R)$, $V = \{S, S_1, S_2, a, b, c\}$, $\sum = \{a, b, c\}$ and

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

(2)

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

3. 令 $L = \{w \in \{a,b\}^* \mid a \neq b\}$, 即那些 a, b 个数不相等的串构成的语言. 试用 CFG 写出能表示 L 的文法.

解答:

$$
\begin{aligned}
&S \to P \mid Q \\
&P \to XAX \mid PP \\
&Q \to XBX \mid QQ \\
&X \to aXb \mid bXa \mid XX \mid \varepsilon \\
&A \to aA \mid a \\
&B \to bB \mid b
\end{aligned}
$$

3 Turing Machine and Undecidability

1. (1) If A is recursive and $B \subseteq A$, Then B is recursive as well.

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

   (3) 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.

   (4) 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.

   (5) Let A and B be recursively enumerable languages and $A \cap B = \emptyset$. If $\overline{A \cup B}$ is also recursively enumerable, then both A and B is decidable.

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

   (7) The set of undecidable languages is uncountable.

解答:

(1) $\times$.

(2) ✓.

(3) ✓. 递归可枚举语言在交, 并下封闭.

(4) ✓.

(5) ✓.

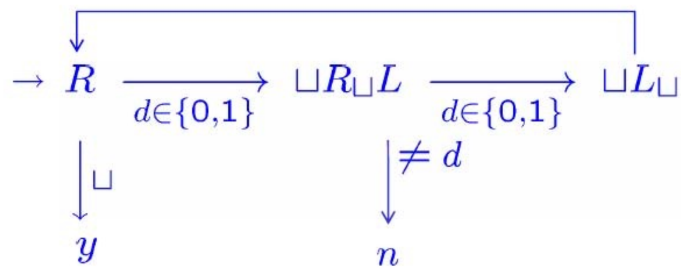
(6) ✓. $\bar{L} \leq_\tau H \implies \bar{L}$ is recursively enumerable.

(7) ✓. The set of Turing machines is countable(encoding TM), so the number of decidable language is countable.

2. Try to construct a Turing Machine to decide the following language.

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

You can assume the start configuration of the Turing machine is $\triangleright \underline{\sqcup} w$.



解答：

3. Show that the function: $\varphi : \mathbb{N} \to \mathbb{N}$ given by

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

解答： Since

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

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.

4. Show the following function $\varphi_k : \underbrace{\mathbb{N} \times \mathbb{N} \times \cdots \times \mathbb{N}}_{k} \mapsto \mathbb{N}$, and $k \in \mathbb{N}, k \geq 2$

$$\varphi_k \left(n_1, n_2, \cdots, n_k\right) = \max_k \left\{n_1, n_2, \cdots, n_k\right\}$$

is primitive recursive.

解答：

$$\varphi_k(n_1, n_2, \cdots, n_k) = \begin{cases} \max_2\{n_1, n_2\}, & \text{if } k = 2 \\ \max_2\{\max_{k-1}\{n_1, n_2, \cdots, n_{k-1}\}, n_k\}, & \text{if } k \geq 3 \end{cases}$$

$\max_2\{n_1, n_2\} = n_1 \cdot (n_1 \geq n_2) + n_2 \cdot (1 \sim (n_1 \geq n_2))$ is primitive recursive.

5. $L_{\text{even}} = \{``M'' \mid M$ is a TM and $L(M)$ contains at least one string of even number of b' s $\}$

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

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

解答：

(1) UTM.

(2) 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 $H = \{``M'' \mid$ 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 $e, L(M_{\text{even}}) = \{e\}$ and accepts at least one string of even length; Otherwise, if M halts on $e, L(M_{\text{even}}) = \emptyset$. 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.

6. Classify whether each of the following languages are recursive, recursively enumerable but not recursive, or non-recursively enumerable.

   1. The language $AL = \{``M'' \mid$ TM M accepts at least 2018 strings $\}$.
   2. The language $E = \{``M'' \mid$ TM M accepts exactly 2018 strings $\}$.
   3. The language $AM = \{``M'' \mid$ TM M accepts at most 2018 strings $\}$.

解答:

1. 递归可枚举但不递归. 利用 UTM 在一个串上一步模拟, 两个串上两步模拟,.... 如果 2018 个串接受, 就接受. 这说明了该语言是递归可枚举的.

   为了证明它不是递归的, 我们证明停机问题可以规约到它. 考虑 "M""w" 是停机问题下的一组实例, 而图灵机 T 可以判定语言 AL. 那么我们只需要构造新图灵机 N, 这个图灵机无论输入什么串, 都会先模拟 M 在 w 上运行, 如果这个模拟终止了, 就接受串. 在这种情况下, N 接受所有串, 自然也接受至少 2018 个串. 所以图灵机 T 如果接受 N, 说明 "M""w" 停机; 拒绝 N, 说明 "M""w" 不停机, 也就完成了规约.

2. 我们将停机问题的补规约到 E. 即考虑 "M""w" 是停机问题的补下的一组实例, 我们要构造图灵机在恰好 2018 个串下面停机, 当且仅当 M **不在** w 下停机.

   首先固定 2018 个串 $v_1, \ldots, v_{2018}$, 而图灵机 N 在输入 $n = v_i$ 时接受然后停机. 如果不是固定的任意 2018 个串中的一个, N 就模拟 M 在 w 上的运行. 如果模拟停机, 就接受然后停机.

   M 不在 w 上停机, N 就只接受 2018 个串. M 在 w 上停机, N 接受所有串. 这就完成了规约.

3. 非递归可枚举. 由 1. 我们也可以知道接受多于 2018 个串的语言同样是递归可枚举但不递归的. 注意接受多于 2018 个串的图灵机构成的语言正好是 AM 的补集. 假设 AM 递归可枚举, 说明接受多于 2018 个串的图灵机构成的语言的补是递归可枚举的, 加上自身是递归可枚举的, 就说明它是递归的, 矛盾!