

第七章作业

袁晨圃

5.1

习题 1 (7.1): 判断真假

- a. $2n = O(n)$
- b. $n^2 = O(n)$
- c. $n^2 = O(n \log^2 n)$
- d. $n \log n = O(n^2)$
- e. $3^n = 2^{O(n)}$
- f. $2^{2^n} = O(2^{2^n})$

解:

- a. ✓
- b. ✗
- c. ✗
- d. ✓
- e. ✓
- f. ✓

习题 2 (7.2): 判断真假

- a. $n = o(2n)$
- b. $2n = o(n^2)$
- c. $2n = o(3^n)$
- d. $1 = o(n)$
- e. $n = o(\log n)$
- f. $1 = o(\frac{1}{n})$

解:

- a. ✗
- b. ✓
- c. ✓
- d. ✓
- e. ✗
- f. ✗

习题 3 (7.6): 证明 P 在并、连接、补运算下封闭

解:

- a. 并: 考虑 $L_1, L_2 \in P, L(M_1) = L_1, L(M_2) = L_2$

构造图灵机 $M =$ 对输入 w :

1. 在 w 上模拟 M_1 , 若接受则接受
2. 在 w 上模拟 M_2 , 若接受则接受
3. 拒绝

设 M_1, M_2 复杂度分别为 $O(n^{k_1}), O(n^{k_2})$, 那么 M 复杂度为 $O(n^{\max(k_1, k_2)})$

- b. 连接: 考虑 $L_1, L_2 \in P, L(M_1) = L_1, L(M_2) = L_2$

构造图灵机 $M =$ 对输入 w :

1. 枚举 w 的前缀 w_1, w_2 , 使得 $w = w_1 w_2$
 - i. 在 w_1 上模拟 M_1
 - ii. 在 w_2 上模拟 M_2
 - iii. 同时接受则接受
2. 拒绝

设 M_1, M_2 复杂度分别为 $O(n^{k_1}), O(n^{k_2})$, 那么 M 复杂度为 $O(n^{\max(k_1, k_2)+1})$

- c. 补: 考虑 $L_0 \in P, L(M_0) = L_0$

构造图灵机 $M =$ 对输入 w :

1. 在 w 上模拟 M_0 , 若接受则拒绝, 反之接受

时间复杂度与 M_0 相同

习题 4 (7.9): 无向图中的三角形是一个 3-团, 证明 $\text{TRIANGLE} \in \text{P}$, 其中 $\text{TRIANGLE} = \{\langle G \rangle \mid G \text{ 包含一个三角形}\}$

解: $M =$ 对于输入 $\langle G \rangle$:

1. 枚举所有的节点三元组 v_1, v_2, v_3
2. 判断 v_1, v_2, v_3 是否互相连接, 如果是, 则接受
3. 不存在这样的三元组, 则拒绝

习题 5 (7.12): 若图 G 节点重新排序后可以变得和 H 相同, 则称 G 和 H 是同构(isomorphic)的, 令 $\text{ISO} = \{\langle G, H \rangle \mid G \text{ 和 } H \text{ 是同构的图}\}$, 证明 $\text{ISO} \in \text{NP}$

解: 证据: 一个 permutation P

多项式时间验证器 $V =$ 对于输入 $\langle G, H, P \rangle$:

1. 枚举所有的节点二元组 v_1, v_2 , 比较 G 中 v_1, v_2 的邻接关系和 H 中 $P(v_1), P(v_2)$ 的邻接关系, 如果不同, 则拒绝
2. 如果对于所有的 (v_1, v_2) , 邻接关系都相同, 则接受

习题 6 (7.29): 令 $\text{SET-SPLITTING} = \{\langle S, C \rangle \mid S \text{ 是有穷集}, C = \{C_1, \dots, C_k\} \text{ 是由 } S \text{ 的某些子集组成的非空集合, 使得 } S \text{ 可以红蓝染色, } \forall C_i, C_i \text{ 中的元素不同色}\}$
证明 SET-SPLITTING 是 NP 完全的

解:

- a. 先证明 SET-SPLITTING 是 NP 的

证据: 一个染色方案 c

多项式时间验证器 $V =$ 对于输入 $\langle S, c \rangle$:

1. 枚举所有的 C_i , 判断 C_i 中的元素是否不同色, 如果不是, 则拒绝
2. 如果对于所有的 C_i , 元素都不同色, 则接受

- b. 证明 SET-SPLITTING 是 NP 完全的

将 3-SAT 多项式时间规约到 SET-SPLITTING :

每一个变元 x 对应 S 中两个节点 $x, \neg x$, $\{x, \neg x\} \in C$

构造两个特殊节点 T, F , $\{T, F\} \in C$

对于每一个字句 $l_1 \vee l_2 \vee l_3$, $\{F, l_1, l_2, l_3\} \in C$

若 3-SAT 可满足, 将为真的变量与 T 染成同色, 为假的变量与 F 染成同色

若 SET-SPLITTING 有解, 对应的将等价关系映射回逻辑变量的真值

5.2

习题 7 (7.16): 证明以下问题是 NP 完全的: 给定一个无向图 G 和 G 节点的一个子集 C , 是否可以通过给 G 的每条边赋予方向, 将 G 转换为一个有向图并且满足属于 C 的节点入度或出度为 0, 不属于 C 的节点入度至少为 1?

解:

- a. 首先这个问题肯定是 NP 的, 依次检验每个节点入度出度以及是否属于 C 即可
- b. 将 3-SAT p 规约到此问题

对于每一个变量 x 创建两个节点 $x, \neg x$, 添加边 $x \rightarrow \neg x$, 且有 $x, \neg x \in C$

考虑 3-SAT 中的每一个字句 $c = l_1 \vee l_2 \vee l_3$:

创建一个节点 c , 添加边 $l_1 \rightarrow c, l_2 \rightarrow c, l_3 \rightarrow c$, 且 $c \notin C$

若 3-SAT 可满足, 将为真的变量的方向定为 $x \rightarrow \neg x$ 且有 $\forall c, x \in c \rightarrow x \rightarrow c \wedge c \rightarrow \neg x$, 为假的定为 $\neg x \rightarrow x$, 类似

因为可满足, 所以赋值之后每个字句至少一个文字为真, 对应至少一个入边

若此问题有解, 映射回变量真假。

习题 8 (7.32): 如果图 G 中有某个节点子集, 其他节点至少与该子集中某一节点相邻, 则该子集称为支配集, 令

$$\text{DOMINATING-SET} = \{(G, k) \mid G \text{ 有一个包含 } k \text{ 个节点的支配集}\}$$

通过将 VERTEX-COVER 问题规约到此问题证明它是 NP 完全的

解:

- a. $\in \text{NP}$: 验证每个节点是否与给定的集合某一节点相邻
- b. 将 VERTEX-COVER 规约到 DOMINATING-SET

对于 VERTEX-COVER 输入 (G, k) , 构造 DOMINATING-SET (G', k)

将 G 改造成 G'

对于 $G = (V, E)$ 的每一条边 $e = (u, v)$ 添加一个中间节点 w_e

添加两条边 (u, w_e) 和 (v, w_e)

若存在顶点覆盖 S , 则每个边 $e = (u, v)$ 至少有一个端点在覆盖中, 对应 G' 中所有中间节点 w_e 被 S 支配

对于原图每一个点, 因为保留了原图的边, 所以肯定与 S 中一节点相邻。所以 S 是一个支配集

若存在支配集 S' , 对于每个新顶点, 若要支配, 则必须有 $u \in S' \vee v \in S'$, 所以每条边 $e = (u, v)$ 至少有端点在 S' 中, 所以 $S' \cap V$ 是顶点覆盖, 添加点使得集合大小重新为 k 即可

习题 9 (°7.39): 有一个盒子和一些卡片, 盒子里有栓塞, 卡片上有凹口, 每张卡片可以以两种方式放入盒子中。

每张卡片有两排孔, 有些孔没有打穿, 将所有卡片放进盒子, 使得盒子的底被完全覆盖, 那么谜题就破解了。

$\text{PUZZLE} = \{\langle c_1, \dots, c_k \rangle \mid c_i \text{ 代表一张卡片, 并且这个卡片集有解}\}$

证明 PUZZLE 是 NP 完全的

解:

a. 属于 NP: 验证证据, 每个卡片只使用一次, 并且每个位置都被盖住

b.

习题 10 (*7.45): 证明若 $P = NP$, 除了 $A = \emptyset$ 和 $A = \Sigma^*$ 以外, 所有语言 $A \in P$ 都是 NP 完全的。

解: 因为 $A \neq \emptyset, A \neq \Sigma^*$, 所以 $\exists w_1 \in A, \exists w_0 \notin A$

将任意属于 NP 的语言 L 规约到 A , 因为 $P=NP$, 所以 $L \in P$, 存在多项式时间判定器 M_L 判定 L , 构造一个多项式时间可计算函数 $f(x) = \begin{cases} w_1 & \text{if } x \in L \\ w_0 & \text{otherwise} \end{cases}$, 所以 $x \in L \iff f(x) \in A$

所以 A 是 NP 完全的