

计算理论导论 第三次作业

周书予

2000013060@stu.pku.edu.cn

April 21, 2022

1

考虑构造一台多项式时间图灵机 M 识别语言 $2COL$.

对于输入 $G = (V, E)$, M 的工作流程是:

1. 初始化队列 Q 为空, 同时将 V 每个点的染色标记为 0.
2. 如果 Q 为空, 则跳转到第 6 步. 否则从 Q 中取出 (v, c) .
3. 如果此时 v 的染色为 $3 - c$, 则拒绝输入 G .
4. 如果 v 的染色为 0, 则将 v 的染色标记为 c , 并遍历 v 在 G 中的所有相邻点 w , 将 $(w, 3 - c)$ 加入 Q .
5. 返回第 2 步.
6. 找到 $v \in V$ 满足 v 的染色为 0, 将 $(v, 1)$ 加入 Q . 如果这样的 v 不存在, 则接受输入 G .

对于任意的输入 G , M 都只会执行上述流程 $O(|G|)$ 步并停机, 流程中的每一步也只需要图灵机 M 的 $O(|G|)$ 步计算, 因此可以说明 $L(M) \in \mathbf{P}$. 接下来证明 $L(M) = 2COL$.

- $\forall G \in L(M)$, M 在停机前其已经构造出了一种合法的染色方案, 因此说明 $G \in 2COL$.
- $\forall G \notin L(M)$, 根据 M 的工作流程可知在对 G 进行染色的过程中出现了颜色冲突, 由归纳可知此时 G 中存在奇环, 而存在奇环的图是不存在 2 染色的, 因此 $G \notin 2COL$.

综上, $2COL \in \mathbf{P}$.

2

由 $L_1, L_2 \in \mathbf{NP}$ 知, 存在 DTM M_1, M_2 以及多项式函数 $p_1, p_2 : \mathbb{N} \rightarrow \mathbb{N}$, 满足对于任意 $x \in \{0, 1\}^*$, 都有 $x \in L_1 \Leftrightarrow \exists u_1 \in \{0, 1\}^{p_1(|x|)}, M_1(x, u_1) = 1, x \in L_2 \Leftrightarrow \exists u_2 \in \{0, 1\}^{p_2(|x|)}, M_2(x, u_2) = 1$.

构造 DTM M_3 满足 $M_3(x, u_1 \circ u_2) = 1$ 当且仅当 $M_1(x, u_1) = 1$ 或 $M_2(x, u_2) = 1$, 从而

$$\begin{aligned} & \exists u_1 \in \{0, 1\}^{p_1(|x|)}, M_1(x, u_1) = 1 \\ x \in L_1 \cup L_2 \Leftrightarrow & \quad \text{or} \quad \Leftrightarrow \exists u_1 \circ u_2 \in \{0, 1\}^{p_1(|x|)+p_2(|x|)}, M_3(x, u_1 \circ u_2) = 1 \\ & \exists u_2 \in \{0, 1\}^{p_2(|x|)}, M_2(x, u_2) = 1 \end{aligned}$$

故构造证明了 $L_1 \cup L_2 \in \mathbf{NP}$.

同理, 构造 DTM M_4 满足 $M_4(x, u_1 \circ u_2) = 1$ 当且仅当 $M_1(x, u_1) = 1$ 且 $M_2(x, u_2) = 1$, 从而

$$\begin{aligned} & \exists u_1 \in \{0, 1\}^{p_1(|x|)}, M_1(x, u_1) = 1 \\ x \in L_1 \cap L_2 \Leftrightarrow & \quad \text{and} \quad \Leftrightarrow \exists u_1 \circ u_2 \in \{0, 1\}^{p_1(|x|)+p_2(|x|)}, M_4(x, u_1 \circ u_2) = 1 \\ & \exists u_2 \in \{0, 1\}^{p_2(|x|)}, M_2(x, u_2) = 1 \end{aligned}$$

证明了 $L_1 \cap L_2 \in \mathbf{NP}$.

3

称 $G = (V_1, E_1)$ 与 $H = (V_2, E_2)$ 同构, 如果 G 可以被“重标号”使得与 H 相同, 亦即存在一个 $\sigma: V_1 \rightarrow V_2$ 的双射, 满足 $(u, v) \in E_1 \Leftrightarrow (\sigma(u), \sigma(v)) \in E_2$. 而这样的双射 σ 很容易用长度为 $|V_1| \log |V_1|$ 的 01 串来编码.

构造一台图灵机 M , 其工作流程是对于输入的 $G = (V_1, E_1), H = (V_2, E_2)$ 以及 $\sigma: V_1 \rightarrow V_2$, 检查 (1) σ 是不是双射, (2) $(u, v) \in E_1 \Leftrightarrow (\sigma(u), \sigma(v)) \in E_2$ 是否成立, 若两项均成立则接受, 否则拒绝. 显然 M 是多项式时间图灵机, 且根据定义, $\langle G, H, \sigma \rangle \in L(M)$ 当且仅当 G 与 H 同构, 即 $\langle G, H \rangle \in \text{ISO}$.

取 $p(n) = n^2$. 此时对于任意 $x \in \{0, 1\}^*$, $x \in \text{ISO}$ 当且仅当存在 $\sigma \in \{0, 1\}^{p(|x|)}$, $M(x, \sigma) = 1$, 故根据定义, $\text{ISO} \in \text{NP}$.

4

对于任意 $L \in \text{NP}$, 考虑构造多项式时间规约 f 满足 $x \in L \Leftrightarrow f(x) \in \text{HALT}$.

假设图灵机 M 与多项式 p 满足 $\forall x \in \{0, 1\}^*, x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}, M(x, u) = 1$, 考虑构造 $f(x) = \langle M', x \rangle$, 其中 M' 的工作流程是: 循环枚举所有长度为 $p(|x|)$ 的 01 串 u 并运行 $M(x, u)$, 一旦 $M(x, u)$ 返回 1 则停机, 否则永不停机.

- 虽然 M' 的运行时间可能达到 $|x|$ 的指数级别甚至更大, 但构造 M' 的描述是只需要多项式时间的, 因此 f 是多项式时间可计算的.
- 同时也不难验证 $\langle M', x \rangle \in \text{HALT}$ 当且仅当存在 $u \in \{0, 1\}^{p(|x|)}, M(x, u) = 1$, 即 $x \in L$.

因此我们证明了 $L \leq_p \text{HALT}$. 由 L 的任意性知 $\text{HALT} \in \text{NP-hard}$.

$\text{HALT} \notin \text{NP-complete}$, 因为否则会导致 $\text{HALT} \in \text{NP} \subseteq \text{EXP}$, 即存在图灵机 M 可以在 $O(2^{n^c})$ 的时间内判定停机问题, 而我们知道停机问题是不可判定的, 因此矛盾.

5

对于任意 $B \in \mathbf{P} \setminus \{\emptyset, \Sigma^*\}$, 都存在 $x, y \in \Sigma^*$ 满足 $x \in B$ 而 $y \notin B$.

显然 $B \in \text{NP}$. 欲证明 $B \in \text{NP-complete}$, 还需要考虑将任意 NP 语言 A 多项式时间规约到 B . 考虑映射 f 满足 $f(w) = \begin{cases} x, & w \in A \\ y, & w \notin A \end{cases} (\forall w \in \Sigma^*)$, 由 $\mathbf{P} = \text{NP}$ 假设可知 $w \in A$ 可在多项式时间内判定, 因此 f 是多项式时间可计算的, 而 $w \in A \Leftrightarrow f(w) \in B (\forall w \in \Sigma^*)$, 说明 $A \leq_p B$, 因此 $B \in \text{NP-complete}$.

语言 B 对应的 x, y 可能并不是多项式时间可计算的, 但只需要指出其“存在性”即可, 因为在“多项式时间规约”的定义中, 只要求了映射 f “存在”, 而不是某种意义上的“多项式时间可构造”. 满足条件的 x, y 完全可以视作映射的 f 的“硬编码”.

6

1. 考虑 $\phi = \bigwedge_i \left(\bigvee_j v_{ij} \right)$ 的一个 \neq -assignment x , 对于 ϕ 的每个从句 $c_i = \bigvee_j v_{ij}$, 都存在 k, k' 使得 $v_{ik}(x) = \text{True}, v_{ik'}(x) = \text{False}$, 这说明 $v_{ik}(\neg x) = \text{False}, v_{ik'}(\neg x) = \text{True}$, 从而 $\neg x$ 也使得 ϕ 的每个从句中包含两种不同的真值, 这说明 \neq -assignment 的否定仍是 \neq -assignment.
2. 考虑建立 3SAT 到 \neq SAT 的多项式时间规约: 考虑映射 f 满足

$$f \left(\bigwedge_i v_{i1} \vee v_{i2} \vee v_{i3} \right) = \bigwedge_i (v_{i1} \vee v_{i2} \vee z_i) \wedge (\overline{z_i} \vee v_{i3} \vee b)$$

我们不妨假设所有 $3CNF$ 都形如上式左边括号里的形式, 因为如若不然, 则可以通过重复一些文字来得到一个如上格式的等价的 CNF .

考虑证明 $w \in 3SAT \Leftrightarrow f(w) \in \neq SAT$:

- (a) 如果 $w = (\bigwedge_i v_{i1} \vee v_{i2} \vee v_{i3}) \in 3SAT$, 说明存在 assignment x 使得 $w(x) = \text{True}$. 在 x 的基础上, 将每个 z_i 赋值为 $\neg(v_{i1} \vee v_{i2})$, 将 b 赋值为 False , 可以得到对 $f(w) = (\bigwedge_i (v_{i1} \vee v_{i2} \vee z_i) \wedge (\bar{z}_i \vee v_{i3} \vee b))$ 的赋值 x' .

只要验证 x' 是 $f(w)$ 的一个 \neq -assignment, 就能说明 $f(w) \in \neq SAT$: 首先由 z_i 的赋值知从句 $v_{i1} \vee v_{i2} \vee z_i$ 中一定包含两种真值, 而要使 $\bar{z}_i \vee v_{i3} \vee b$ 只包含一种真值, 则必须有 $\bar{z}_i(x') = v_{i3}(x') = \text{False}$, 这导致了 $v_{i1}(x) = v_{i2}(x) = v_{i3}(x) = \text{False}$, 与 $w(x) = \text{True}$ 矛盾.

- (b) 如果 $f(w) \in \neq SAT$, 说明存在一个 \neq -assignment y . 讨论 $b(y)$ 的真值:

- 如果 $b(y) = \text{False}$, 则截取 y 中关于 v_{ij} 的部分得到的赋值 y' 可以满足 $w(y') = \text{True}$, 这是因为对于任意一组 $v_{i1}(y'), v_{i2}(y'), v_{i3}(y')$, 三者不可能同时为 False , 否则在 y 赋值下 $v_{i1} \vee v_{i2} \vee z_i, \bar{z}_i \vee v_{i3} \vee b$ 这两个从句中会有恰好一个从句包含三个 False , 与 \neq -assignment 矛盾.
- 如果 $b(y) = \text{True}$, 则选取前述 y' 的否定 $\neg y'$ 可以满足 $w(\neg y') = \text{True}$, 这是因为对于任意一组 $v_{i1}(\neg y'), v_{i2}(\neg y'), v_{i3}(\neg y')$, 三者不可能同时为 False , 否则在 y 赋值下 $v_{i1} \vee v_{i2} \vee z_i, \bar{z}_i \vee v_{i3} \vee b$ 这两个从句中会有恰好一个从句包含三个 True , 与 \neq -assignment 矛盾.

故 $w \in 3SAT$.

f 显然是多项式时间可计算的, 因此说明了 $3SAT \leq_p \neq SAT$.

3. 由 Cook-Levin Theorem 知 $3SAT \in \mathbf{NP}$ -complete, 即 $\forall L \in \mathbf{NP}, L \leq_p 3SAT$. 而前面我们证明了 $3SAT \leq_p \neq SAT$, 因此由 \leq_p 的传递性可知 $\forall L \in \mathbf{NP}, L \leq_p \neq SAT$, 即 $\neq SAT \in \mathbf{NP}$ -complete.