

计算理论导论 第四次作业

周书予

2000013060@stu.pku.edu.cn

April 30, 2022

1

考虑建立 $\text{Vertex-Cover} = \{\langle G, k \rangle \mid G \text{ has a subset of } k \text{ vertices that covers all edges}\}$ 到 $\text{Dominating-Set} = \{\langle G, k \rangle \mid G \text{ has a subset of } k \text{ vertices that covers all vertices}\}$ 的多项式时间规约。

对于 $G = (V, E)$, 记 G 中孤立点 (度数为 0 的点) 的集合为 $\text{iso}(G)$, 考虑映射 $f(\langle G, k \rangle) = \langle G', k + |\text{iso}(G)| \rangle$, 其中 $G' = (V', E')$ 满足

$$\begin{aligned} V' &= V \cup E \\ E' &= E \cup \bigcup_{(u,v) \in E} \{((u,v), u), ((u,v), v)\} \end{aligned}$$

也即, G' 保留 G 中所有点与边, 并对原图中的每条边 (u, v) 建立一个新点, 并连接两点 u, v .

容易发现 f 是多项式时间可计算的. 考虑验证 $\langle G, k \rangle \in \text{Vertex-Cover} \Leftrightarrow f(\langle G, k \rangle) \in \text{Dominating-Set}$:

- 如果 $\langle G, k \rangle \in \text{Vertex-Cover}$, 则存在大小为 k 的覆盖集 $S \subseteq V$, 满足对任意 $(u, v) \in E$ 都有 $u \in S$ 或者 $v \in S$. 我们验证 $S \cup \text{iso}(G)$ 是 G' 的一个支配集:
 - 对于任意 $u \in V \subseteq V'$, 要么 $u \in \text{iso}(G)$, 要么存在一条边 $(u, v) \in E$, 说明 $u \in S$ 与 $v \in S$ 至少一者成立, 两种情况下 u 均被 $S \cup \text{iso}(G)$ 支配.
 - 对于任意 $(u, v) \in E \subseteq V'$, $u \in S$ 与 $v \in S$ 至少一者成立, 从而 (u, v) 被 $S \cup \text{iso}(G)$ 支配.
 - 从而 V' 被 $S \cup \text{iso}(G)$ 支配. 而 $|S \cup \text{iso}(G)| \leq |S| + |\text{iso}(G)| = k + |\text{iso}(G)|$, 随便补充一些点使支配集大小达到 $k + |\text{iso}(G)|$ 即可说明 $f(\langle G, k \rangle) \in \text{Dominating-Set}$.
- 如果 $f(\langle G, k \rangle) \in \text{Dominating-Set}$, 则存在大小为 $k + |\text{iso}(G)|$ 的支配集 $S' \subseteq V'$, 满足对于任意 $v' \in V'$, 都要么 $v' \in S'$, 要么存在 $(u', v') \in E'$ 使得 $u' \in S'$.
 - 如果 $S' \subseteq V$, 即 S' 中的点都是原 G 中的点, 则直接构造 $S = S' \setminus \text{iso}(G)$ 即为 G 的一个覆盖集 (因为每条边都被覆盖了). 注意到 S' 是支配集, 从而必然有 $\text{iso}(G) \subseteq S'$, 说明 $|S| = |S' \setminus \text{iso}(G)| = k$, 故 $\langle G, k \rangle \in \text{Vertex-Cover}$ 成立.
 - 如果 S' 中存在某个点 $(u, v) \in E$ 不是原 G 中的点, 注意到选择 (u, v) 时支配的点集只有 $\{u, v, (u, v)\}$, 而通过把 (u, v) 的选择改成 u 或者 v , 也可以支配这个点集. 换句话说, 如果 S' 包含 (u, v) 是支配集, 则 $S' \setminus \{(u, v)\} \cup \{u\}$ 也是支配集, 同时有 $|S' \setminus \{(u, v)\} \cup \{u\}| \leq |S'|$. 因此可以不断替换 S' 中不是原 G 中的点, 将 S' 转化为第一种情形. 替换过程中可能发生 $|S'|$ 变小的情况, 但这是无足轻重的, 因为存在较小的支配集/覆盖集总能推出存在更大的支配集/覆盖集.

故 $\text{Vertex-Cover} \leq_p \text{Dominating-Set}$, 从而证明了 $\text{Dominating-Set} \in \text{NP-hard}$. 而显然 $\text{Dominating-Set} \in \text{NP}$, 因为只需要将大小为 k 的支配集作为 certificate 即可, 故 $\text{Dominating-Set} \in \text{NP-complete}$.

2

假设存在 **NP**-complete 的 unary language L , 则存在多项式时间可计算函数 $f : \text{CNF} \rightarrow 1^*$ 满足 $\varphi \in \text{SAT} \Leftrightarrow f(\varphi) \in L$. 由于 f 多项式时间可计算, 故存在常数 c 使得 $\forall \varphi \in \text{CNF}, |f(\varphi)| \leq |\varphi|^c$.

我们通过设计一个判定 **SAT** 的多项式时间算法来证明 **P** = **NP**.

Algorithm 1 SAT in polynomial time

Require: a boolean formula φ on n variables x_1, x_2, \dots, x_n

```

1:  $S_0 \leftarrow \langle f(\varphi), \varphi \rangle$ 
2: for  $i = 1 \rightarrow n$  do
3:    $S_i \leftarrow \emptyset$ 
4:   for  $\langle f(\phi), \phi \rangle$  in  $S_{i-1}$  do  $\triangleright \phi$  is a boolean formula on variables  $x_i, x_{i+1}, \dots, x_n$ 
5:     substitute  $x_i = \text{True}$  and then  $x_i = \text{False}$  in  $\phi$ , obtaining boolean formulas  $\phi_T$  and  $\phi_F$ 
6:     insert  $\langle f(\phi_T), \phi_T \rangle$  and  $\langle f(\phi_F), \phi_F \rangle$  into  $S_i$ 
7:   for each  $n \in \mathbb{N}$ , retain only one element of form  $\langle 1^n, \psi \rangle$  in  $S_i$  and drop the others
8: if  $\langle f(\text{True}), \text{True} \rangle \in S_n$  then
9:   return  $\varphi$  is satisfiable
10: else
11:   return  $\varphi$  is not satisfiable
  
```

如果 φ 是可满足的, 则在任意 S_i 中, 都包含元素 $\langle f(\phi), \phi \rangle$, 其中 ϕ 是可满足的 (这是因为 $f(\phi) = f(\psi)$ 能说明 ϕ, ψ 可满足性相同, 故不会因缩减 S_i 的规模而导致可满足的 CNF 全部被丢弃), 从而 S_n 中包含可满足的 CNF **True**, 算法输出 φ 是可满足的.

如果 φ 是不可满足的, 则显然任意 S_i 中不会有可满足的 ϕ 形成 $\langle f(\phi), \phi \rangle$ 出现, 因此 S_n 中也不会包含 **True**, 算法输出 φ 是不可满足的.

注意到因为代入只会使表达式长度变短, 算法中涉及到的任意 ϕ 都有 $|\phi| \leq |\varphi|$, 故 $f(\phi) \leq |\phi|^c \leq |\varphi|^c$, 这说明任意 S_i 在缩减规模后大小都不超过 $|\varphi|^c$, 因此算法的运行时间是关于 $|\varphi|$ 的多项式.

3

$$\text{SPACE-TM} = \{ \langle M, w, 1^n \rangle \mid \text{DTM } M \text{ accepts } w \text{ in space } n \}$$

- **SPACE-TM** \in **PSPACE**: 构造判定 **SPACE-TM** 的图灵机 D 为, 使用通用图灵机 \mathcal{U} 模拟 M 在输入 w 上运行, 并时刻检查是否只使用了不超过 n 的运行空间. 由于通用图灵机模拟只需要常数的额外空间, 故 D 的运行空间关于输入规模呈线性. 显然 $L(D) = \text{SPACE-TM}$, 故 **SPACE-TM** \in **PSPACE**.
- **SPACE-TM** \in **PSPACE-hard**: 对于任意 **PSPACE** 语言 L , 都存在一台运行空间为 $O(n^c)$ 的图灵机 M 满足 $L(M) = L$. 取映射 f 满足 $f(x) = \langle M, x, 1^{|x|^c} \rangle$, 显然 f 是多项式时间可计算的, 且 $x \in L \Leftrightarrow f(x) \in \text{SPACE-TM}$, 于是 $L \leq_p \text{SPACE-TM}$, 故 **SPACE-TM** \in **PSPACE-hard**.

综上, **SPACE-TM** \in **PSPACE-complete**.

4

$$\text{EXACT-INDSET} = \{ \langle G, k \rangle \mid \text{the largest independent set of } G \text{ is of size } k \}$$

4.1

图 G 的最大独立集大小为 k , 当且仅当存在大小为 k 的独立集, 且不存在大小为 $k+1$ 的独立集.

$$\text{EXACT-INDSET} = \text{INDSET} \cap \overline{\text{INDSET}'}$$

其中

$$\begin{aligned} \text{INDSET} &= \{\langle G, k \rangle \mid G \text{ has an independent set of size } k\} \in \mathbf{NP} \\ \text{INDSET}' &= \overline{\{\langle G, k \rangle \mid G \text{ has an independent set of size } k+1\}} \\ &= \{\langle G, k \rangle \mid G \text{ has no independent set of size } k+1\} \in \mathbf{coNP} \end{aligned}$$

故 $\text{EXACT-INDSET} \in \mathbf{DP}$.

4.2

回顾 $3\text{SAT} \leq_p \text{INDSET}$ 的证明, 我们对 $3\text{CNF } \varphi$ 的每个大小为 $l \leq 3$ 的 clause 建立了不超过 2^l 个点表示能使这个 clause 为 True 的赋值方式, 并在所有会产生冲突的赋值方式之间连边. 记这张图为 G , φ 一共有 k 个 clause, 此时有 $\varphi \in 3\text{SAT} \Leftrightarrow \langle G, k \rangle \in \text{EXACT-INDSET}$.

可以考虑对构造的图进行一些修改: 建立 $k-1$ 个新点, 分别与原先的所有点连边, 这样一来 G 的最大独立集必然是 k 或者 $k-1$, 且有 $\varphi \in 3\text{SAT} \Leftrightarrow \langle G, k \rangle \in \text{EXACT-INDSET}, \varphi \notin 3\text{SAT} \Leftrightarrow \langle G, k-1 \rangle \in \text{EXACT-INDSET}$.

回到原问题. 对于任意 $L \in \mathbf{DP}$, 存在 $L_1, L_2 \in \mathbf{NP}$, 使得 $x \in L \Leftrightarrow x \in L_1, x \notin L_2$. 考虑多项式时间可计算函数 f, g 分别将 L_1, L_2 规约到 3SAT , 设 $f(x), g(x)$ 两个 3CNF 分别有 k_1, k_2 个 clause, 按照前述的方式对 $f(x), g(x)$ 建图得到 G_1, G_2 , 则有

$$x \in L \Leftrightarrow x \in L_1, x \notin L_2 \Leftrightarrow f(x) \in 3\text{SAT}, g(x) \notin 3\text{SAT} \Leftrightarrow \langle G_1, k_1 \rangle, \langle G_2, k_2 - 1 \rangle \in \text{EXACT-INDSET}$$

对于图 $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, 建立其“笛卡尔积” $G = G_1 \times G_2 = (V, E)$, 其中 $V = V_1 \times V_2$, $((u_1, u_2), (v_1, v_2)) \in E$ 当且仅当 $(u_1, v_1) \in E_1$ 或者 $(u_2, v_2) \in E_2$. 可以验证 $\langle G_1, k_1 \rangle, \langle G_2, k_2 \rangle \in \text{EXACT-INDSET} \Rightarrow \langle G, k_1 k_2 \rangle \in \text{EXACT-INDSET}$.

记 $\langle G = G_1 \times G_2, k_1(k_2 - 1) \rangle = h(x)$, 其中 G_1, G_2, k_1, k_2 如前述定义.

接下来将证明 $x \in L \Leftrightarrow h(x) \in \text{EXACT-INDSET}$.

- $x \in L \Rightarrow \langle G_1, k_1 \rangle, \langle G_2, k_2 - 1 \rangle \in \text{EXACT-INDSET} \Rightarrow \langle G, k_1(k_2 - 1) \rangle \in \text{EXACT-INDSET}$, 这个方向是容易的.
- 如果 $x \notin L$, 则要么 $x \notin L_1 \Rightarrow \langle G_1, k_1 - 1 \rangle \in \text{EXACT-INDSET}$, 要么 $x \in L_2 \Rightarrow \langle G_2, k_2 \rangle \in \text{EXACT-INDSET}$, 这将导致 (1) $\langle G, (k_1 - 1)(k_2 - 1) \rangle$, (2) $\langle G, k_1 k_2 \rangle$, (3) $\langle G, (k_1 - 1)k_2 \rangle$, 三者中存在某一者属于 EXACT-INDSET . 为了与 $\langle G, k_1(k_2 - 1) \rangle \in \text{EXACT-INDSET}$ 产生矛盾, 需要额外保证 $k_1 \neq 0, k_2 \neq 0, k_1 \neq k_2$, 注意到 k_1, k_2 分别是两个 3CNF 的 clause 数量, 只需要在构造 $f(x), g(x)$ 时重复部分 clause 即可实现这样的保证.

所以 h 是从 L 到 EXACT-INDSET 的多项式时间规约, 说明 $L \leq_p \text{EXACT-INDSET}$.

5

$$\text{STR-CON} = \{\langle G \rangle \mid G \text{ is a strongly connected directed graph}\}$$

欲证明 $\text{STR-CON} \in \mathbf{NL-complete}$, 只需证明:

- **STR-CON** \in **NL**: 按照 **NL** 的 certificate definition, 只需要证明存在 $O(\log n)$ 空间图灵机 (verifier) M , 对于任意有向图 $G = (V, E)$, G 强连通当且仅当存在多项式规模的 certificate u 满足 $M(G, u) = 1$.

按如下形式构造 M 和 u : 按照字典序升序, u 给出每对 $(i, j) \in V \times V$ “ G 中存在 i 到 j 的有向路径”的 certificate, 每段 certificate 包含一个 G 的顶点序列 $v_0 v_1 \cdots v_k$, 相邻 certificate 之间用特殊符号分隔. 显然这样的 u 是多项式规模的.

至于 verifier M , 它应该检查: (i) 是否按照升序完整给出了所有 (i, j) 对的 certificate, (ii) 每个 (i, j) 对的 certificate 是否合法, 即是否满足 $v_0 = i, v_k = j$, 且 $(v_t, v_{t+1}) \in E$ 对所有 $0 \leq t < k$ 成立. 显然 M 只需要 $O(\log n)$ 的工作空间.

- **STR-CON** \in **NL-hard**: 考虑将

$$\text{PATH} = \{\langle G, s, t \rangle \mid G = (V, E) \text{ is a directed graph in which there is a path from } s \text{ to } t\}$$

规约到 **STR-CON**. 构造映射 f , $f(\langle G, s, t \rangle) = G' = (V, E')$, 其中 G' 有与 G 相同的点集 V , 包含 G 中的所有边, 同时对于任意 $v \in V$ 都有 $(t, v), (v, s) \in E'$.

- G' 的构造是简单的, 可以在 $O(\log n)$ 的工作空间中计算出 G' 邻接矩阵的每一位, 故 f 是隐对数空间可计算函数.
- 当 $\langle G, s, t \rangle \in \text{PATH}$ 时, G' 中存在 s 到 t 的有向路径, 因而对于任意 $(i, j) \in V \times V$ 都有 $i \rightarrow s \rightarrow t \rightarrow j$ 的有向路径, 说明 G' 强连通.
- 当 $\langle G, s, t \rangle \notin \text{PATH}$ 时, G' 中不存在 s 到 t 的有向路径, 导致 G' 不强连通.

于是 $\langle G, s, t \rangle \in \text{PATH} \Leftrightarrow f(\langle G, s, t \rangle) \in \text{STR-CON}$, 说明 $\text{PATH} \leq_l \text{STR-CON}$, 说明 **STR-CON** \in **NL-hard**.

6

定义 1 (NL 的 certificate definition). 称语言 $L \in \text{NL}$, 如果存在一台 DTM M 和多项式 $p: \mathbb{N} \rightarrow \mathbb{N}$, 满足对于任意 $x \in \{0, 1\}^*$, $x \in L \Leftrightarrow \exists u \in \{0, 1\}^{p(|x|)}$, s.t. $M(x, u) = 1$, 其中 u 被放在 M 一条只能读一次的纸带上, 且 M 只能使用到 $O(\log |x|)$ 个工作纸带空间.

考虑去掉“只能读一次”的限制, 记新得到的语言类为 **NL'**. 以下证明 **NL'** = **NP**:

- **NL'** \subseteq **NP**: 显然, 因为只要在定义中去掉对 M 工作纸带空间的限制, 就变成了 **NP** 的定义.
- **NP** \subseteq **NL'**: 考虑证明任意 $L \in \text{NP}$ 都满足 $L \leq_l \text{SAT} \in \text{NL}'$, 从而说明 $L \in \text{NL}'$:
 - 任取 $L \in \text{NP}$, 在 Cook Levin Theorem 中证明了 $L \leq_p \text{SAT}$, 方式是考虑判定 L 的 oblivious TM M , 对 M 在输入 x 上运行过程的 snapshot 序列建立多项式规模的 CNF. 我们指出这个规约实际上还是隐对数空间可计算的, 因为构造出的这个 CNF 的长度以及任意一位比特都可以在对数空间内计算得到. 所以 $L \leq_l \text{SAT}$.
 - 对于 $\varphi \in \text{SAT}$, 将其一组可满足赋值作为其 certificate, 构造 DTM M , 其计算过程为顺序遍历输入 CNF φ 的每一个 clause, 在给出的 certificate 中查询每一个文字的真值, 判断是否每一个 clause 内都存在至少一个为 True 的文字. M 的计算只需要 $O(\log |\varphi|)$ 额外空间, 因此 M 符合 **NL** 的 certificate definition 中的要求, 而 $L(M) = \text{SAT}$, 说明 $\text{SAT} \in \text{NL}'$.