理论计算机科学基础-学习笔记

唐嘉良 2020K8009907032

Lecture 25 NL = coNL，层次定理

1 上节回顾

Def 语言B被称为NL-complete（NL完全的），如果它满足以下两个条件：

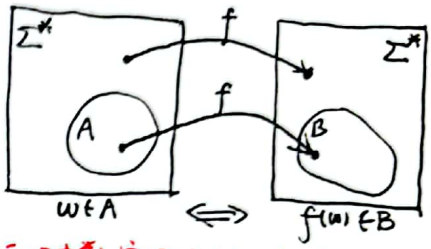
① $B \in NL$

② $\forall A \in NL$，有 $A \leq_L B$.

此定义类似NPC的定义，在于给出一个对数空间归约意义下最"强"的一类NL语言。于是类似NPC的相关性质，可以得到下面定理。

Then if $A \leq_L B$ and $B \in L$, then $A \in L$;

if $A \leq_L B$ and $B \in NL$, then $A \in NL$.



同多项式归约，对数归约也给出了类似结论，实质在于所定义的对数空间归约(给)诱导出了一种接受A的满足条件的Turing Machine. 书上对证明进行了改进，让任何时刻只存储f(w)的一个符号，结果是用时间换取空间。

Inf if B is NL-complete, $B \in L$, then $L = NL$.

此推论是明显的，其直接的直观意义也是显然：如果NL中最难的一类问题可以归约或等价于一个L类问题，那么NL就不再比L严格"强"，这也为证明P=NP提供了思路（为一个NPC问题寻找多项式确定性算法？）

Thm PATH is NP-complete.

证明要点：已经知道PATH是一个NL问题，我们仅需证明每个 $A \in NL$ 有 $A \leq_L PATH$.

这要求我们找寻NL语言的共性。因为不同的NL语言是千奇百怪的。于是自然地联想到从Definition出发，即A的非确定性对数空间Turing Machine，而PATH也似是图相关的，于是找到归约把字符串w映射为一个图。

为实现 $w \in A \Leftrightarrow f(w) \in PATH$，归约f的像结点对应非确定图灵机在输入w下面格局，两结点存在边等价于一个结点对应格局能在一步内(非确定图灵机上)产生第二个结点的格局。

于是，机器接受w $\Leftrightarrow$ 初始格局结点与接受格局间互存在路径。

Inf $NL \subseteq P$.

对数空间归约是多项式归约的特殊情况，因对数空间回归的时间复杂度是输入长度 n 的多项式 $p(n)$. 根据该事实，有下面的证明.

证明：由 $A \leq_L PATH$ $(\forall A \in NL)$，得 $A \leq_p PATH$

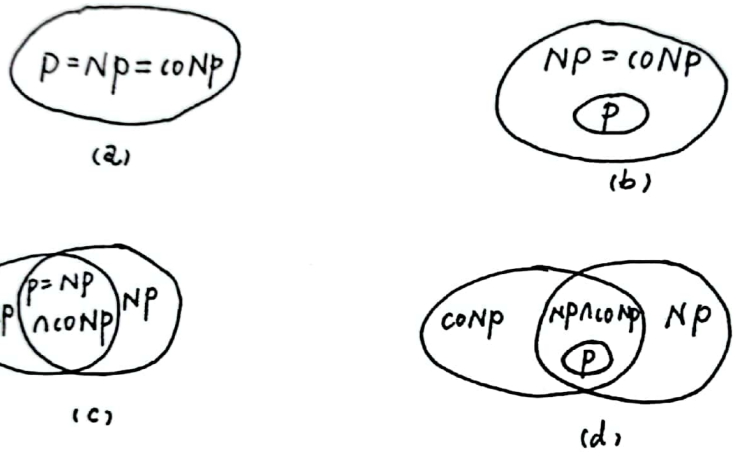
又由 $PATH \in P$，知 $A \in P$

于是 $NL \subseteq P$. □

注：消耗空间 $f(n)$ 的图灵机在 $n \cdot 2^{O(f(n))}$ 时间内运行.

2. $NL = coNL$

Rmk relations between P、NP and coNP.



(d) is the most impossible ralation.

直观上 $P = NP$ 的可能性不是很高，因为 Turing Machine 的确定性与否似乎是一个较本质的问题，(d) 在于相信 P、coNP 与 NP 之间存在一个"层次"关系.

Def $coNL = \{\overline{A} \mid A \in NL\}$. 该定义是用 NL 定义的.

目前的结论是 $L \subseteq NL = coNL \subseteq P$.

Thm (Immerman - Szelepcsényi Theorem) $NL = coNL$.

证明：考虑到 PATH 的 NL 完全性，从 PATH 着手，发现一个事实：$NL = coNL \Leftrightarrow \overline{PATH} \in NL$.

只需能证明这个结论，再给出对数空间非确定性图灵机判定 $\overline{PATH}$ 即可. ■

第一部分的证明是容易的，只是定义的运用.

一方面，若 $NL=coNL$，由 $PATH\in NL$ 知 $\overline{PATH}\in coNL=NL$ $\therefore \overline{PATH}\in NL$. ("⇒")

另一方面，若 $\overline{PATH}\in NL$，首先 $\forall A\in NL$，有 $A\le_L PATH$，有 $\overline{A}\le_L \overline{PATH}$. 由 $\overline{PATH}\in NL$ 知

$\overline{A}\in NL \Rightarrow A\in coNL \Rightarrow NL\subseteq coNL$

另外 $\forall A\in coNL$，有 $\overline{A}\in NL \Rightarrow \overline{A}\le_L PATH \Rightarrow A\le_L \overline{PATH}$. 由 $\overline{PATH}\in NL$ 知

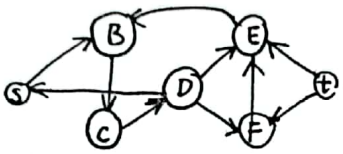
$A\in NL \Rightarrow coNL\subseteq NL$ $\therefore NL=coNL$. ("⇐")

第二部分的证明相对需要技巧.

设 $G=(V,E)$ 是有向图，$|G|=m$. 考虑 A_i 为 G 中与 s 距离不超过 i 的结点集合

则 $A_0=\{s\}$，$A_i\subseteq A_{i+1}$

example：



$A_0=\{s\}, A_1=\{s,B\}, A_2=\{s,B,C,D\},$
$A_3=A_4=A_5=A_6=A_7=\{s,B,C,D,E,F\}$.
令 $C_i=|A_i|$.

构造判定 $\overline{PATH}$ 的算法如下：

这样的计算一层一层"扩散"开，类似递归的思想，直至扩散至 C_m

"on input $<G,s,t>$

calculate C_m.

1. let $C_0=1$
2. for $i=0$ to $m-1$
3. let $C_{i+1}=1$
4. for each nodes ~~in~~ $v\neq s$ in G
5. let $d=0$
6. for each node u in G
7. Nondeterministically either perform or skip these steps:
8. Nondeterministically follow a path of length at most i from s and reject if it doesn't end at u.
9. Increment d.
10. If (u,v) is an edge in G, increment C_{i+1} and go to stage 5 with the next v.
11. if $d\neq C_i$, reject.

the nondeterministical calculating process based on given G,s,t,C and $c=C_m$, which judged $\overline{PATH}$.

12. let $d=0$
13. for each node u in G
14. Nondeterministically either perform or skip these steps:
15. Nondeterministically follow a path of length at most m from s and reject if it doesn't end at u.
16. if $u=t$, reject
17. Increment d
18. if $d\neq C_m$, reject. Otherwise accrept."

任何时刻仅需存 m, u, v, C_i, C_{i+1}, d, i 及末端 pointer，于是在对数空间内运行.

Then we get $\overline{PATH}\in NL$. □

Chapter 9 难解性.

像 NP-hard 这种需大量时间、空间的问题就是难解问题.

3. 空间层次定理

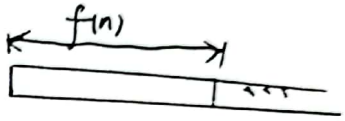
Def 对函数 $f: N \to N$，其中 $f(n)$ 至少为 $O(\log n)$，若 f 将 1^n 映成 $f(n)$ 二进制表示，且 f 在空间 $O(f(n))$ 内可计算，则称该函数是空间(世俗构造)的 (space constructiable).

Rmk 通常复杂度至少 $O(\log n)$ 的函数都是空间可构造的.

如 $O(\log n)$ 内计算 $\log n$，对亚线性空间界限 $\log n$，数 1 的数目计算 n 的二进制形式(在工作带上)位数，空间复杂度 $O(\log n)$.

Thm (空间层次定理) $\forall f: N \to N$ 空间可构造函数，$\exists$ 语言 A 在空间 $O(f(n))$ 可判定，但在空间 $o(f(n))$ 不可判定.

证明：构造 $O(f(n))$ 空间的算法 D 如下：



D = "对输入 w：

1. 令 $n = |w|$
2. 计算 $f(n)$，划分出 $f(n)$ 长带空间，若后续步骤试图调用更多空间，则拒绝.
3. 若 w 不是形如 $\langle M \rangle 10^*$ (M 是 TM)，则拒绝
4. w 上模拟 M，计算模拟步数，计数超 $2^{f(n)}$ 则拒绝 → $f(n)$ 的分配空间已经不够多
5. 若 M 接受则拒绝拒绝，否则接受. ". 反着取，以此构造一种不可判定性(对角线)

若 $\exists$ 图灵机 M 在 $o(f(n))$ 空间判定 A，$\exists$ 存在 n_0 s.t. $n > n_0$ 时 $d_0(f(n)) < f(n)$. 考虑 D 在 $w = \langle M \rangle 10^{n_0}$ 上运行，D 对 M 模拟仅需 $d_0(f(n))$ 空间，第4步略成. 则 D 与 M 在同一输入上结果相反，M 不判定 A. 矛盾! □

保证 A 与 $o(f(n))$ 内可判定的语言有不同之处

注：之所以选取 $2^{f(n)}$ 是因为要保证 D 不会一直循环，是一个判定器.

Inf $\forall f_1, f_2: N \to N$，$f_{1,2}$ 是函数，若 $f_1(n) = o(f_2(n))$ 且 f_2 空间可构造，则 $\mathrm{SPACE}(f_1(n)) \subsetneq \mathrm{SPACE}(f_2(n))$. □

Inf $\forall \varepsilon_1, \varepsilon_2 \in \mathbb{R}$，$0 < \varepsilon_1 < \varepsilon_2$，$\mathrm{SPACE}(n^{\varepsilon_1}) \subsetneq \mathrm{SPACE}(n^{\varepsilon_2})$

此定理看似显然，实则需讨论 ε_2 细节，因为空间可构造性不是常见的.

证明：① $\varepsilon_2 \in \mathbb{N}^+$ 时，常见 n^{ε_2} 空间可构造，结论成立

② $\varepsilon_2 \in \mathbb{Q}^+$ 时，同①，结论成立.

③ $\varepsilon_2 \in \mathbb{R}^+$ 时，$\exists r \in \mathbb{Q}^+$ s.t. $0 \le \varepsilon_1 < r < \varepsilon_2$. 于是 $\mathrm{SPACE}(n^{\varepsilon_1}) \subsetneq \mathrm{SPACE}(n^r)$.

又由 $\mathrm{SPACE}(n^r) \subseteq \mathrm{SPACE}(n^{\varepsilon_2})$，结论成立. □

已知 $L \subseteq NL = coNL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \subseteq EXPSPACE$.

Inf $NL \subsetneq PSPACE$

证明：由 Savitch Theorem，$NL \subseteq \mathrm{SPACE}((\log n)^2)$，由空间层次定理，$\mathrm{SPACE}((\log n)^2) \subsetneq \mathrm{SPACE}(n)$. □

Inf $PSPACE \subsetneq EXPSPACE$

证明：$n^{\log n} = o(2^n)$，2^n 空间可构造，于是 $\mathrm{SPACE}(n^{\log n}) \subsetneq \mathrm{SPACE}(2^n)$

构造中间函数，方便使用空间层次定理.

$\forall k \in \mathbb{N}^+$，$\mathrm{SPACE}(n^k) \subseteq \mathrm{SPACE}(n^{\log n})$. 于是 $\mathrm{SPACE}(n^k) \subsetneq \mathrm{SPACE}(2^n)$. □

4. 时间层次定理

Def 对函数 $f: N \to N$，其中 $t(n)$ 至少为 $O(n \log n)$，若 t 把 1^n 映射为 $t(n)$ 二进制表示，且该函数在时间 $O(t(n))$ 内可计算，则称该函数是时间可构造的 (time constructible).

Thm (时间层次定理) $\forall$ 时间可构造函数 $t: N \to N$，存在语言 A，在时间 $O(t(n))$ 内可判定，但不能在时间 $o\left(\frac{t(n)}{\log t(n)}\right)$ 内判定. → 增加了 $\log t(n)$ 倍的时间开销 (不同于空间层次增加和空间开销) 的原因为尚且不知如何实现更高效的模拟.

证明：与空间层次定理类似，采用对角线方法

构造算法 ($O(t(n))$ 时间的) D，判定的语言 A 不能在 $o\left(\frac{t(n)}{\log t(n)}\right)$ 内判定.

$D =$ "对输入 w，

1. 记录 $n = |w|$
2. 计算 $t(n)$，将 $\lceil \frac{t(n)}{\log t(n)} \rceil$ 存放在二进制计数器中，执行 4. 之前将值减 1，若减到 0 则拒绝.
3. 在 O 上模拟 M
4. M 接受则拒绝，M 拒绝则接受."

这样的 D 可以保证 $O(t(n))$ 运行，同时又规避了所有 $o\left(\frac{t(n)}{\log t(n)}\right)$ 的 (可判定) 语言

定理中出现 $\log t(n)$ 因子，予示了时间层次定理中 $\log t(n)$ 倍的层次关系

计数器修改最多 k 次，仅需有它即可

可以看出 3、4 总时间不超过 $d\lceil \frac{t(n)}{\log t(n)} \rceil$，最多采用修改计数器时间不超过 $d\lceil \frac{t(n)}{\log t(n)} \rceil \times O(\log t(n)) = O(t(n))$. 类似空间层次定理可证明 A 被 D 判定，A 无法在 $o\left(\frac{t(n)}{\log t(n)}\right)$ 时间内被判定. □

Inf $\forall$ 函数 $t_1, t_2: N \to N$，其中 $t_1(n) = o\left(\frac{t_2(n)}{\log t_2(n)}\right)$ 且 $t_2(n)$ 是时间可构造的，则 $\mathrm{TIME}(t_1(n)) \subsetneq \mathrm{TIME}(t_2(n))$

Inf $\forall \varepsilon_1, \varepsilon_2 \in \mathbb{R}$，$0 < \varepsilon_1 < \varepsilon_2$，有 $\mathrm{TIME}(n^{\varepsilon_1}) \subsetneq \mathrm{TIME}(n^{\varepsilon_2})$

Inf $P \subsetneq \mathrm{EXPTIME}$

时空层次定理，关系小结：

$$L \subseteq NL = coNL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE \subseteq EXPTIME \subseteq EXPSPACE$$

$NL \subsetneq PSPACE$

$P \subsetneq EXPTIME$

$PSPACE \subsetneq EXPSPACE$.