

1.5 oblivious: 散漫的

设 M 在 $T(n)$ 时间内计算出判定语言 L

我们设计 \tilde{M} 来模拟 M M 有 k 带

M 工作带第 i 带第 j 单元放在 \tilde{M} 第 $i+jk$ 个位置 (工作带)

$M(\Gamma, Q, \delta)$ $\tilde{M}(\tilde{\Gamma}, \tilde{Q}, \tilde{\delta})$

$|\tilde{\Gamma}| = 2|\Gamma|$

$\forall a \in \Gamma$, 有 a 和 $\hat{a} \in \tilde{\Gamma}$ 与 a 对应

\hat{a} 表示 head 在此处 a 表示 head 不在此处 $\tilde{Q} = \Gamma^k \times \{L, R, S\}^k$

1. 在 $O(T(n))$ 时间内将 \tilde{M} 输入带内容移入 M 工作带模拟的 M 的输入带

2. \tilde{M} 读写头在原点位置, 对于每个对 M 的单步模拟:

2.1 从左往右扫, 读到 \hat{a} , 将其记入 \tilde{a} , 时间为 $O(T(n))$

2.2 在最右端 (第 $k \cdot T(n) + 1$ 位置) 得到了 M 的 Snapshot, 用了模拟 δ

2.3 回到最左端, 向右扫, 对于 M 中向右移的 Head, 可以在 $O(T(n))$ 全部右移

2.4 : 与 2.3 相反, 左移 \tilde{M} , 模拟 M 中 Head 左移, \tilde{M} Head 移至最左端

2.5 以上 2.1~2.4, 代价 $O(T(n))$, 最多 $T(n)$ 次 总代价 $O(T^2(n))$

所以有 2 带 oblivious 图灵机 在条件 $O(T^2(n))$ 时间内判定 $L \in \text{DTIME}(T(n))$

1.6 设有 $L_1, L_2, L_3, \dots, L_k, R_1, R_2, \dots, R_k$ 一些 Region

L_i / R_i 有 $2 \cdot 2^i$ 个元素 L_k 有一个元素

原图尺规 $M(\Gamma, Q, \delta)$ k 带

新图尺规 $\tilde{M}(\tilde{\Gamma}, \tilde{Q}, \tilde{\delta})$ $|\tilde{\Gamma}| = |\Gamma|^k$ $\tilde{\Gamma} = \Gamma^k$

在 \tilde{M} 中, 允许因空带 空带每占 0 (全满), 2^i (半满) 或者 $2 \cdot 2^i$ (全空) 个位置 (在 L_i 或 R_i 中)

其中 L_k 表示的是 k 个带的 head 位置, 一定不是因, 且 k 处有且仅有一个因

算法: 初始时 L_i / R_i 都半满

第 $step$ 步模拟:

1 for $i \leftarrow \lceil \log step \rceil$ to 1

2 if $(step \equiv 0 \pmod{2^{i-1}})$ {

3 检查 $L_i \dots L_0$ 是否至少 2^{i-1} 个非空位

4 如果不是, 从 L_{i+1} 拉 2^i 个元素, 并对 R_{i+1} 对称操作

5 检查 $R_0 \dots R_i$ 是否至少 2^{i-1} 个非空位

6 如果不是, 从 R_{i+1} 拉 2^{i+1} 个元素, 并对 L_{i+1} 对称操作

7 }

8 }

9 此时 $R_0 \dots R_i$ 至少 1 个非空位, $L_0 \dots L_i$ 至少 1 个非空位

10 扫描 $L_1 L_0 L_k R_0 R_i$, 完成模拟

$$\begin{array}{r} 8421 \\ \hline 01234 \\ 1248 \\ \hline 306 \end{array}$$

第 4, 6 行, 我们断言 L_{i+1} 或 R_{i+1} 有元素

分析: 对 L_i / R_i 进行一次操作, 则 $R_0 \dots R_i$ 至少有 2^{i-1} 个元素

对 L_i / R_i 操作时, $R_0 \dots R_i / L_0 \dots L_i$ 至少有 $2^{i-2} - 1$ 个元素 $> 2^{i-1} - 1$ 个元素

所以 4, 6 行 操作可行

3-6 行 $O(2^i)$ 总复杂度 $O(i \cdot 2^i) = O(T \log T)$ 且以上操作皆 oblivious

1.9: 分析: 数组 A 不可以直接模拟, 因为不知道数组长度是否可能很大

所以我们考虑, 用 A' 模拟 A , 记录 $a_1 w_0, a_1 w_1, \dots$, 以空带结束

A' 的带总数为 $O(T(n))$, A' 长度 $O(T(n))$, head 在根

进入 $qaccess$ 后, 对于 $a_1 w_0$, 在 A' 末尾加一条记录

对于 $a_1 R$, 从 A' 根往前找, 对比 L_i 和记录上 L_i 是否相同,

找到第几个配对的, 填在 R 前面 q

$qaccess$ 代价 $O(T(n))$ 可用 A' 模拟

所以若 f 在 $T(n)$ 被 RAM Turing Machine 计算, $f \in DTIME(T(n))$

1.13: $compBIT(n, i) = Divide(p_i, n)$

$= \exists k: p_k = n$

(b) $compare(n, m, i, j) = (n_i = m_j)$

$n \leftrightarrow b = (a \wedge b) \vee (\neg a \wedge \neg b)$

$= (p_i | n \leftrightarrow p_j | m)$

(c) Each Symbol can be encode as $\lceil \log |T| \rceil$ long binary string

There are at most T k symbols on the k tapes

The head location can be encode as a $\lceil \log T \rceil$ long binary string

The state q can also be encode with length $\lceil \log |Q| \rceil$

So such a configuration can be represented by a binary string

(d) $INIT_{M, X}(n) = \forall i (p_i | n \leftrightarrow M_i)$

$M_0 \dots M_m$ 是一个 configuration

$= \forall i (\exists x p_i x = n) \leftrightarrow M_i$

$= \forall i (\exists x p_i x = n \wedge M_i) \vee (\neg \exists x p_i x = n \wedge \neg M_i)$

(e) $HALT_M(n) = \forall i (Bit(n, i) = Bit(halt, i) \leftarrow i \in state_digits)$

(f) $Next(n, m) = (\forall i (Bit(n, i) = Bit(m, i) \leftarrow i \neq Head(n)) \wedge (Bit(n, i) = calculate(n, i) \leftarrow i = Head(n)) \wedge Next_state(n, m))$

$$(g) \text{VALID}_M(m, t) = \bigwedge_{i=1}^{t-1} \text{Next}_M(m_i, m_{i+1})$$

$$(h) \text{HALT}_{M_X}(t) = \exists m, \text{VALID}_M(m, t) \wedge \text{INIT}_{M_X}(m_0) \wedge \text{HALT}_M(m_t)$$

(i) 假设 TRUE-EXP 由 M_1 计算, L 表示一个数论结论的语言

$$M_1(x) = \text{TRUE-EXP}(x) \quad \forall x \in L \quad \text{由(h)知 } M_1(x) \in L$$

$$M_2(M, x) := M(x) \quad \text{所以 } M_2 \in L$$

$$M_3(M) = \neg M_2(M, M) \quad \text{所以 } M_3 \in L$$

$$\text{若 } M_3(M_3) = 1 \text{ 则 } M_3(M_3) = 0$$

$$\text{若 } M_3(M_3) = 0 \text{ 则 } M_3(M_3) = 1 \quad \text{出现矛盾}$$

所以 TRUE-EXP 不可计算

1.14 由 1.9 知, RAM TURING MACHINE 由 TM 可经式时间模拟

(a) $E(i, j)$ 表示 i, j 有力相连, 不用用邻接矩阵存入数组

for $j \leftarrow 1$ to n

for $i \leftarrow 1$ to n

for $k \leftarrow 1$ to n

if $E(i, j) \wedge E(j, k) : \text{set } E(i, k) \leftarrow 1$

sum $\leftarrow 0$

for $i \leftarrow 1$ to n

for $j \leftarrow 1$ to n

if $E(i, j) : \text{sum} \leftarrow \text{sum} + 1$

if sum = 0? return false

else return true

以上可由 RAM TM 在 P 内模拟, 所以 TM 在 P 内可判定 CONNECTED

(b) TRIANGLE FREE

for $j \leftarrow 1$ to N

for $i \leftarrow 1$ to N

for $k \leftarrow 1$ to N

if $E(i, k) \wedge E(k, j) \wedge E(j, i) : \text{return False}$

return True

所以 P 内由 RAM TM 判定

(c) BIPARTITE 用另一个工作带模拟栈: pop, push, top
(stack)

for $i \leftarrow 1$ to N

if $color[i] = \text{NONE}$:

$color[i] = 0$

$stack.push(i)$

while $stack.top() \neq \text{NONE}$:

$k = stack.top(), stack.pop()$

for $j \leftarrow 1$ to N :

if $E[k, j]$:

if $color[j] = color[k]$: return false

if $color[j] = \text{NONE}$:

$color[j] \leftarrow \neg color[k]$

$stack.push(j)$

return true

至多有 N 个元素入过 stack,

所以复杂度为多项式

用 RAM TM P 模拟

所以 TM P 可判定

(d)

if $\neg \text{CONNECTED}$ Return False

$sum \leftarrow 0$

for $i \leftarrow 1$ to $N-1$

for $j \leftarrow i+1$ to N

$sum \leftarrow sum + E[i, j]$

if $sum = N-1$ return True

else Return False

所以 RAM TM P 可判定

所以 TM P 可判定