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 个数放在 $\tilde{M}$ 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 $\hat{a} \in \tilde{\Gamma}$ 与之对应

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

1、在 $O(T(n))$ 时间内将 $\tilde{M}$ 输入带内容移入 $\tilde{M}$ 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 $\tilde{q}$，时间为 $O(T(n))$

2.2 在最右端 (第 $k \cdot T(n)$ 个位置) 得到了 M 的 Snapshot，用 $\tilde{\delta}$ 模拟 δ

2.3 回到 $\tilde{M}$ 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 会到最左端

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

所以存在 2 带 oblivious 图灵机，能在 $O(T^2(n))$ 时间内判定 $L \in DTIME(T(n))$

1.6 设有 ··· $L_2\, L_1\, L_0\, LR\, R_0\, R_1$ ··· 一些 Region

L_i / R_i 有 $2\cdot 2^i$ 个元素 LR有一个元素

原图灵机 $M(\Gamma, Q, \delta)$ k带

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

在$\tilde{M}$中, 允许 □ 空字符 空字符占 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 LR表示的是k个带的 head 位置, 一定不是 □, 且 ±k处有且仅有一个□

算法: 初始时 L_i / R_i 都半满

第 step 步模拟:

```
1   for i ← ⌈log_2 step⌉ to 1
2       if( step ≡ 0 (mod 2^(i-1)) ){
3           检查 Li···L0 是否至少 2^i -1 个非空位
4           如果不是, 从 Li+1 拿 2^i 个元素, 并对 Ri+1 对称操作
5           检查 R0···Ri 是否至少 2^i -1 个非空位
6           如果不是, 从 Ri+1 拿 2^(i+1) 个元素, 并对 Li+1 对称操作
7       }
8   }
9   此时 R0 R1 至少 1个非空位, L1 L0 至少 1个非空位
10  扫描 L1 L0 LR R0 R1, 完成模拟
```

第 4, 6行, 我们断言了 L_{i+1} 或 R_{i+1} 有元素

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

则下次对 L_i/R_i 操作时, $R_0 \cdots R_i$ / $L_0 \cdots L_i$ 都至少有 $2^i - 2^{i-2} - 1$ 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 i 对应数组长度是可以很大

所以我们考虑, 用A'模拟A, 记录 $\square i_0 W \sigma_0 \square i_1 W \sigma_1 \cdots$, 以空字符结束

A'的条目个数为 $O(T(n))$, A'长度 $O(T(n))$, head在末尾

进入 q_{access} 后, 对于 $\square i W \sigma$, 在A'末尾加一条记录

对于 $\square i R$, 从A'末尾往前找, 对比 i 和记录上 i_j 是否相同.

找到第一个相同的, 填在 R 后面 σ_j

q_{access} 代价 $O(T(n))$ 可以用A'模拟

所以若 f 在 $T(n)$ 被 RAM Turing Machine 计算, $f \in DTIME(T(n)^2)$

1.13: (a) $BIT(n,i) = Divide(p_i, n)$

$= \exists k: p_i k = n$

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

$= (p_i | n \leftrightarrow p_j | m)$ $a \leftrightarrow b = (a\wedge b) \vee (\neg a \wedge \neg b)$

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

There are at most $T\cdot k$ 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) $INIT_{M,x}(n) = \forall i (p_i | n \leftrightarrow M_i)$ $M_0 \cdots M_m$ 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) $Next(n,m) = (\forall i\, Bit(n,i) = Bit(m,i) \leftarrow i \neq Head(n)) \wedge (Bit(n,i) = calculate(n,i) \leftarrow i = Head(n))$

$\wedge\ NextState(n,m)$

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

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

(i) 假设 TRUE-EXP 由 M_1 计算，L表示一个数论结论的语言

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

$M_2(M, x) := M(x)$　　所以 $M_2 \in L$

$M_3(M) := \neg M_2(M, M)$　　所以 $M_3 \in L$

若 $M_3(M_3)=1$ 则 $M_3(M_3)=0$

若 $M_3(M_3)=0$ 则 $M_3(M_3)=1$　　出现矛盾

所以 TRUE-EXP 不可计算

1.14 由1.9知，RAM TURING MACHINE 由TM可多项式时间模拟

(a) E(i,j)表示i,j有边相连，不妨用邻接矩阵存入数组

```
for j ← 1 to n
    for i ← 1 to n
        for k ← 1 to n
            if E(i,j) ∧ E(j,k) : set E(i,k) ← 1
sum ← 0
for i ← 1 to n
    for j ← 1 to n
        if E(i,j) : sum ← sum+1
if sum = n² return ~~true~~ true
else return false
```

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

(b) TRIANGLE FREE

```
for j ← 1 to N
    for i ← 1 to N
        for k ← 1 to N
            if E(i,k) ∧ E(k,j) ∧ E(j,i) return False
return True
```

可以P内由 RAM TM判定

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

```
for i ← 1 to N
    if color[i] = NONE :
        color[i] = 0
        stack.push(i)
        while stack.top() ≠ NONE :
            k = stack.Top() , stack.pop()
            for j ← 1 to N :
                if E[k,j] :
                    if color[j] = color[k] : return false
                    if color[j] = NONE :
                        color[j] ← 1-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 ¬ CONNECTED  Return False
sum ← 0
for i ← 1 to N-1
    for j ← i+1 to N
        sum ← sum + E[i,j]
If sum = N-1  return True
else Return False.
```

可以 RAM TM P内判定

所以 TM P内判定

计算复杂性 10月8日 改

1.13.

(a) $BIT(n,i) = \exists p\ \forall tmp1\ (p > (i+c)^3) \land (p < (i+c+1)^3) \land [(tmp1 < p) \land (tmp1 > (i+c)^3 \rightarrow$
$\neg isPrime(tmp1))] \land Divides(p, n) \land isPrime(p)$

$isPrime(y) = \forall x\ (x=1) \lor (x=y) \lor \neg DIVIDES(x, y)$

$Divides(x, y) = \exists k: y = k \times x$

(b) $Compare(n, m, i, j) = BIT(n,i) \leftrightarrow Bit(m,j)$

$a \leftrightarrow b = (a \rightarrow b) \land (b \rightarrow a)$

$a \rightarrow b = b \lor \neg a$

(c) 假设 单带图灵机 在T时刻内必停机

$M(\Gamma, Q, \delta)$ 在某时刻，至多T个位置被访问

前 $T \cdot \lceil \log|\Gamma| \rceil$ 个 0/1表示的是 带上信息　　X初始　　除第一位外，全0结束

之后一个 $\lceil \log T \rceil$ bit 表示 head位置　　0初始　　0结束

之后 $\lceil \log|Q| \rceil$ bit 表示 状态　　0初始　　全1结束

$\delta: \{0,1\}^{\lceil \log|\Gamma| \rceil + \lceil \log|Q| \rceil} \rightarrow \{0,1\}^{\lceil \log|\Gamma| \rceil + \lceil \log|Q| \rceil + 2}$

$\tilde{\delta}: N \rightarrow N$ 表示编码后的函数

(d) 初始时 带上为输入

$Init_{M,x}(n) = [\forall_i (i < T\lceil \log|\Gamma| \rceil) \rightarrow (BIT(n,i) \leftrightarrow (x_i = 1))] \land [\forall_i (T\lceil \log\Gamma \rceil \le i < T\lceil \log\Gamma \rceil$
$+ \lceil \log Q \rceil + \lceil \log T \rceil) \rightarrow (BIT(n,i) \leftrightarrow 0)]$

(e) $HALT_M(n) = [\forall_i (1 \le i < T\lceil \log\Gamma \rceil + \lceil \log T \rceil) \rightarrow \neg BIT(n,i)]$
$\land [\forall i\ (T\lceil \log\Gamma \rceil + \lceil \log T \rceil \le i < T\lceil \log\Gamma \rceil + \lceil \log T \rceil + \lceil \log Q \rceil) \rightarrow BIT(n,i)]$

(f) $NEXT(n,m) = $ ~~$\exists head\ \forall_i (Bit(n, headstart + i) \leftrightarrow head_i = 1)$~~
$\land \exists snapshot\ \forall_j (j < \lceil \log\Gamma \rceil) \rightarrow (Bit(n, head + j) \leftrightarrow snapshot(j))$
$\land (\forall_j < \lceil \log Q \rceil) \rightarrow [Bit(n, statestart + j) \leftrightarrow snapshot(\lceil \log\Gamma \rceil + j)]$
$\land (\delta(snapshot)[motion] + head) = head[m] \land \delta(snapshot)[head\ symbol] \leftrightarrow headsymbol(n)$
$\land \delta(snapshot)[q: state\ end] =$ ... (f) 改在下一页

(f) 为了避免太长，用下列定义简化表达式

定义 $array[a:b] = number$

$\exists\, number\ \forall a < i < b\ \ number[i-a] = array[i]$

例 $arr1[3:5] = arr2[4:6]$

定义 $\mathrm{Equal}(arr1, a, b, arr2, c, d)$

$= \forall i\ \mathrm{Compare}(arr1, arr2, a+i, c+i) \leftarrow (i < b, i < d)$

$\mathrm{NextHead}(head1, head2, move)$

$\mathrm{Next}(n, m) = \exists\, head\ \mathrm{Equal}(head, 0, \lceil \log T \rceil, n, T\lceil \log \Gamma \rceil, \log T)$

$\wedge \exists\, snapshot\ \mathrm{Equal}(snapshot, 0, \lceil \log \Gamma \rceil, n, head \times \lceil \log \Gamma \rceil, \lceil \log \Gamma \rceil)$

$\wedge \mathrm{Equal}(snapshot, \lceil \log \Gamma \rceil, \lceil \log Q \rceil, n, T\lceil \log \Gamma \rceil + \lceil \log T \rceil, \lceil \log Q \rceil)$

$\wedge \exists\, result\ \delta(snapshot) = result\ \wedge \exists\, move\ \mathrm{Equal}(move, 0, 2, result, T\lceil \log \Gamma \rceil + \log T + \log Q, 2)$

$\wedge \mathrm{Equal}(result, 0, \lceil \log \Gamma \rceil, m, head \times \lceil \log \Gamma \rceil, \lceil \log \Gamma \rceil)$

$\wedge \mathrm{Equal}(result, \lceil \log \Gamma \rceil, \lceil \log Q \rceil, m, T\lceil \log \Gamma \rceil + \lceil \log T \rceil, \lceil \log Q \rceil)$

$\wedge \exists\, head2\ \ \mathrm{NextHead}(head, head2, move)$

$\wedge \mathrm{Equal}(head2, 0, \lceil \log T \rceil, m, T\lceil \log \Gamma \rceil, \log T)$

$\wedge \mathrm{Equal}(n, 0, (head+1) \times \lceil \log \Gamma \rceil, m, 0, (head+1) \times \lceil \log \Gamma \rceil)$

$\wedge \mathrm{Equal}(n, head \times \lceil \log \Gamma \rceil, (T-head)\lceil \log \Gamma \rceil, m, head \times \lceil \log \Gamma \rceil, (T-head)\lceil \log \Gamma \rceil)$

(g) $\mathrm{VALID}_M(m, t) = \mathrm{Next}(m_0, m_1) \wedge \cdots \wedge \mathrm{Next}(m_{t-1}, m_t)$

(h) $\mathrm{HALT}^e_{M,x}(t) = \overset{\exists n\ \exists m[0 \ldots t]}{\cancel{\forall}} \mathrm{HALT}_M(m_t) \wedge \mathrm{INIT}_{M,x}(n) \wedge \mathrm{VALID}_M(m, t)$

$\mathrm{HALT}_{M,x}(t) = \exists t'\ (t' \leq t) \wedge \mathrm{HALT}^e_{M,x}(t')$

(i) 由前面，可计算的图灵机为一个数论结论

设 数论结论 语言L

假设 True-Exp 由 M_1 计算

$M_1(x) = \text{True-Exp}(x) \quad M_1 \in L$

$M_2(M, x) := M(x) \quad M_2 \in L$

$M_3(M) := \neg M_2(M, M) \quad M_3 \in L$

分析 $M_3(M_3)$

$M_3(M_3) = 1 \Leftrightarrow M_3(M_3) = 0$ 矛盾