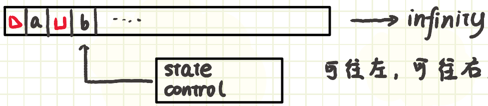
Title Lecture 6

Date 2023 · 11 · 8

Review

Turing Machine



可往左，可往右，可读可写

- definition: $M=(K,\Sigma,\delta,s,H)$

tape 开头　中间间隙

H: Halting states　Σ: tape alphabet（有 $\triangleright$ 和 $\sqcup$）

$\delta: (K-H)\times\Sigma \longrightarrow K\times(\{\leftarrow,\rightarrow\}\cup\{\Sigma-\{\triangleright\}\})$

$\delta(q,\triangleright)=(p,\rightarrow)$　moving　writing　开头不可写

configuration 需要得知 state 和 tape 位置

$\triangleright\sqcup a\sqcup ab\sqcup$ ⇒ $(p,\triangleright\sqcup a\sqcup a, b)$
(p) ↑ ⇒ $(p,\triangleright\sqcup a\sqcup ab\ )$

$\triangleright\sqcup a\sqcup a\sqcup\sqcup$ ⇒ $(p,\triangleright\sqcup a\sqcup a, e)$
(p) ↑ ⇒ $(p,\triangleright\sqcup a\sqcup a)$　非空格结尾

configuration: $K\times\triangleright(\Sigma-\{\triangleright\})^*\times(\{e\}\cup(\Sigma-\{\triangleright\})^*(\Sigma-\{\triangleright,\sqcup\}))$

w 箭头左侧　箭头右侧 u

① writing　$\delta(q,a_1)=(q,a_2)$　$u_1=u_2$　$w_2a_2=w_1a_1$

② 左移　$\delta(q,a)=(q,\leftarrow)$　$w_1=w_2a_2$　$u_2=a_1u_1$
$(a_1=\sqcup\ u_1=e\quad u_2=e)$

③ 右移　$\delta(q,a)=(q,\rightarrow)$　$w_2=w_1a_1$　$a_2u_2=u_1$
$(a_2=\sqcup\quad u_1=e\quad u_2=e)$

$(q_1,\triangleright w_1a_1u_1)\vdash_M^*(q_2,\triangleright w_2a_2u_2)$

$(q,\triangleright wau)$　$q\in H$ → Halting configuration.

① symbol writing Machine. M_a

$M=(K,\Sigma,\delta,s,\{h\})$　$\delta(s,\triangleright)=\delta(s,\rightarrow)$

$\delta(s,b)=(h,a)$　$b\in\{\Sigma-\{\triangleright\}\}$

② shifting Machine $M_\leftarrow$　$M_\rightarrow$

$M=(K,\Sigma,\delta,s,\{h\})$　$\delta(s,\triangleright)=\delta(s,\rightarrow)$

$\delta(s,b)=(h,\leftarrow)$　$b\in\{\Sigma-\{\triangleright\}\}$（右移类似）

- basic machine: M_a. $M_\leftarrow$　$M_\rightarrow$

$>M_1\xrightarrow{0}M_3$
$M_1\xrightarrow{1}M_2$

先跑 M_1 至 Halting
为 1 跑 M_2　为 0 跑 M_3 否则停机

$>R\xrightarrow{\Sigma}R$ (R^2)（右移2格）　$>R\xrightarrow{a\neq\sqcup}R_a$（右移1格写 a）

$>R\circlearrowleft\ \overline{\sqcup}$（右移直至右侧第一个空格）($R_\sqcup$)

$L_\sqcup$ 有可能无法停机，类似 $R_{\overline{\sqcup}}$　$L_{\overline{\sqcup}}$，$R_{\overline{\sqcup}}$ 可能不停机

构造 $S_\leftarrow$　$\triangleright\sqcup\sqcup w\sqcup\longrightarrow\triangleright\sqcup w\sqcup$

$>L_\sqcup\longrightarrow R\xrightarrow{a\neq\sqcup}\sqcup L a R$，$R\xrightarrow{\sqcup}L$

Title L 6

Recoganize language.

input alphabet: $\Sigma_0 \subseteq (\Sigma - \{\triangleright, \sqcup\})$

initial cofiguration: $(s, \triangleright\underline{\sqcup}w)$ → 指针指向 ⊔

$L(M) = \{w \in \Sigma_0^* : (s, \triangleright\underline{\sqcup}w) \vdash_M^* (h, \cdots), h \in H\}$ 半判定

↳ recursively enumerable 递归可枚举

- $M = (K, \Sigma_0, \Sigma, \delta, s, \{yes, no\})$ be a TM ↱ 接受

  M decides $L \in \Sigma_0^*$ if ① $w \in L$ $(s, \triangleright\underline{\sqcup}w) \vdash_M^* (yes, \cdots)$

  ② $w \in \Sigma_0^* - L$ $(s, \triangleright\underline{\sqcup}w) \vdash_M^* (no, \cdots)$ → 不接受

  一定可停机，而半判定的不接受为永不停机

  ↳ L : recursive language.

  可判定的L一定可半判定 → 将no设为非停机

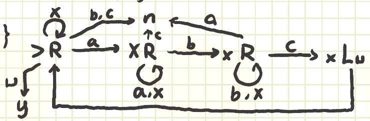
  recursive　　　recursive enumerable.

- compute function.

  $w \in \Sigma^*$ if $(s, \triangleright\underline{\sqcup}w) \vdash_M^* (h, \triangleright\underline{\sqcup}y)$ $h \in H$ $y \in \Sigma_0^*$

  y是M的输出　M可计算 $f: \Sigma_0^* \to \Sigma_0^*$

证明 $\{a^n b^n c^n\}$ 可判定



↳ 可接受 $(abc)^n$

须修正

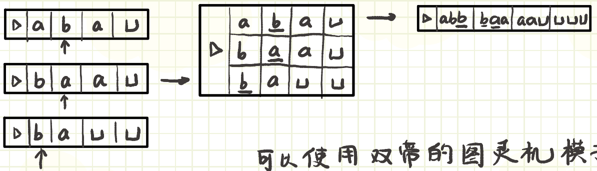
Title Lecture 7

Date 2023.11.15 Review

- multiple tape TM

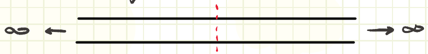
有 k 条 tape $\delta:(K-H)\times\Sigma^k \longrightarrow K\times(\Sigma\cup\{\leftarrow,\rightarrow\})^k$

TM 与 multitape TM 等价



可以使用双带的图灵机模拟

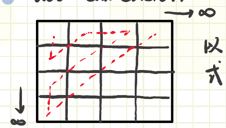
- Two-way infinite tape TM



- multiple head TM

每次扫描所有下划线作为 configuration

- two-dimension TM



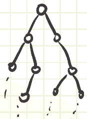
以图中的方式进行编号

- ramdon access TM

可能有跳步

- non-determic TM NTM 非确定性图灵机

与 TM 的不同仅为 Δ 不为函数 为一组关系



NTM 的半判定：若有一条计算分支可以停机。

NTM 的判定：$M=(K,\Sigma,\Delta,s,\{y,n\})$ with input Σ_0

M decides $L\subseteq\Sigma_0^*$ if

① con. N（与 M 无关）st. $\forall w\in\Sigma_0^*$，无 configuration C 有 $(s,\rhd\underline{\sqcup}w)\vdash_M^N C$ → 任何一条分支都在 N 步内停机

② $w\in L$ iff $(s,\rhd\underline{\sqcup}w)\vdash_M^*(y,\cdots)$ → 有分支被接受

$w\notin L$ iff 没有分支被接受

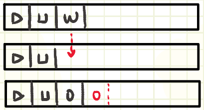
$C=\{100,110,1000,\cdots\}$ composite numbers. 非素数

Idea：查看 w 是否可被分解

∀ NTM 可被一台 DTM 模拟 每层的 node 有限

证明思路：NTM 半判定 L ⟶ DTM 半判定 L（使用 BFS）

使用 3-tape DTM 模拟



store the input

stimulate NTM

enumerate hint （0，表示向左走）

算法与每条分支都停机的 TM 等价

描述 TM
- $M=(\cdots)$
- diagram 
- high-level pseudo code

Title L7

- encoding

所有finite set可编码, finite collection of finite sets 可编码

- 伪代码描述图灵机

$L = \{G : G$ 是连通图$\}$ $M =$ on input "G"

default: 0.1 输入非法，拒绝 0.2 进行判定

1. select a node in G 2. mark its neighbor
3. repeat until no new marked node.
4. if all marked, accept, else reject.

$A_{DFA} = \{$"D" "w" : D is a DFA accepts w$\}$

M_{R_1} = on input D w 1. run D on w
2. if D accept w, accept. else, reject

$A_{NFA} = \{$"N", "w" : N is a NFA accepts w$\}$

M_{R_2} : on input N w 1. N ⟶ DFA D
2. run M_{R_1} on D w 3. output the result of M_{R_1}

上述例子中 "N" "w" $\in A_{NFA}$ iff "D" "w" $\in A_{DFA}$

$R_2(A_{NFA}) \longrightarrow R_1(A_{DFA})$

a reduction from R_2 to R_1

归约是一种由表象向与映射与向一致，需要保证结果一致。

$A_{REX} = \{$"R" "w" : R is a regular express with $w \in L(R)\}$

M_{R_3} : on input "R" "w" 1. R ⟶ an equivalent NFA N
2. run M_{R_2} on "N" "w" 3. output the result of M_{R_2}

$E_{DFA} = \{$"D" : D is a DFA and $L(D) = \phi\}$

M_{R_4} : on input D 1. if D has no final state, accept
2. else run DFS / BFS from s in the diagram
3. if $\exists$ path from s to final reject 4. else accept.

$EQ_{DFA} = \{$"D_1" "D_2" : $L(D_1) = L(D_2)\}$

- symmetric difference. $A \oplus B = \{x \in A \cup B \wedge x \notin A \cap B\}$

$A = B$ iff $A \oplus B = \phi$ $= (A \cup B) \cap (\bar{A} \cup \bar{B})$

由 $D_1 D_2$ 可构造出 D_3 st. $L(D_3) = (L(D_1) \cup L(D_2)) \cap (\overline{L(D_1)} \cup \overline{L(D_2)})$

M_{R_5} : on input $D_1 D_2$ 1. construct D_3 2. Run M_{R_4} on D_3
3. output the result of M_{R_4}