形式语言与自动机理论

图灵机与不可判定性

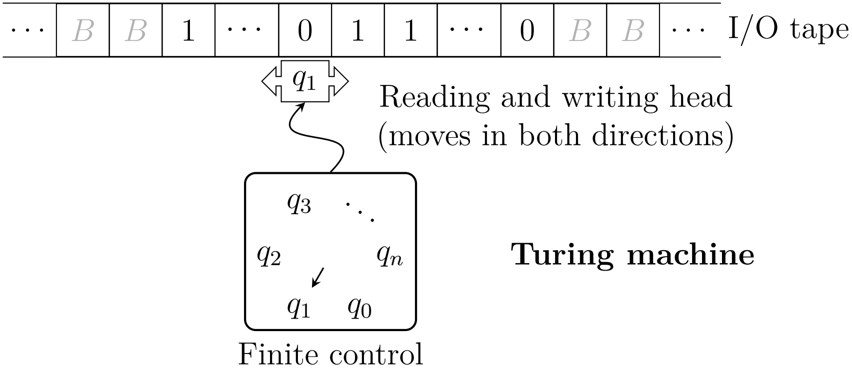
王春宇

计算机科学与技术学院
哈尔滨工业大学

图灵机与不可判定性

- 图灵机
  - 语言与停机
  - 整数函数计算器
  - 图灵机的变形
- 不可判定性

图灵机



⋯
B
B
1
⋯
0
1
1
⋯
0
B
B
⋯
I/O tape
q_1
Reading and writing head
(moves in both directions)
q_3
q_2
q_n
q_1
q_0
Finite control
Turing machine

图灵机的形式定义

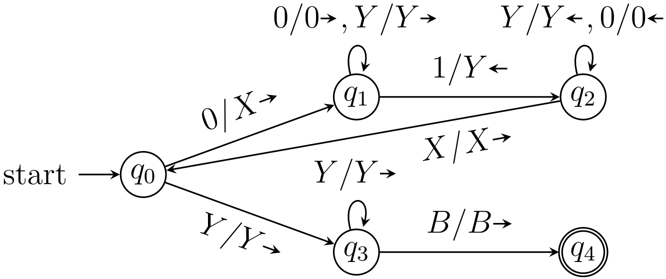
定义

图灵机(*TM, Turing Machine*) M 为七元组

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$$

1. Q: 有穷状态集;
2. Σ: 有穷输入符号集;
3. Γ: 有穷带符号集, 且总有 $\Sigma \subset \Gamma$;
4. δ: $Q \times \Gamma \to Q \times \Gamma \times \{L, R\}$ 转移函数;
5. $q_0 \in Q$: 初始状态;
6. $B \in \Gamma - \Sigma$: 空格符号;
7. $F \subseteq Q$: 终态集或接受状态集.

例 1. 设计识别 $\{0^n1^n \mid n \geq 1\}$ 的图灵机.



续例 1. 设计识别 $\{0^n1^n \mid n \geq 1\}$ 的图灵机.

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, X, Y, B\}, \delta, q_0, B, \{q_4\})$$

δ	0	1	X	Y	B
q_0	(q_1, X, R)	$-$	$-$	(q_3, Y, R)	$-$
q_1	$(q_1, 0, R)$	(q_2, Y, L)	$-$	(q_1, Y, R)	$-$
q_2	$(q_2, 0, L)$	$-$	(q_0, X, R)	(q_2, Y, L)	$-$
q_3	$-$	$-$	$-$	(q_3, Y, R)	(q_4, B, R)
q_4	$-$	$-$	$-$	$-$	$-$

瞬时描述

定义

图灵机虽有无穷长的带，但经过有限步，带上非空内容总是有限的．因此用全部非空符号、当前状态及带头位置，定义图灵机的瞬时描述*(ID)* 为

$$X_1X_2\cdots X_{i-1}qX_iX_{i+1}\cdots X_n$$

❶ 图灵机的当前状态 q;

❷ 带头在左起第 i 个非空格符 X_i 上;

❸ $X_1X_2\cdots X_n$ 是最左到最右非空格内容.

转移符号

定义

图灵机 M 中，如果 $\delta(q, X_i) = (p, Y, L)$，定义 *ID* **转移**为

$$X_1 \cdots X_{i-1} q X_i \cdots X_n \vdash_M X_1 \cdots X_{i-2} p X_{i-1} Y X_{i+1} \cdots X_n$$

如果 $\delta(q, X_i) = (p, Y, R)$ 那么

$$X_1 \cdots X_{i-1} q X_i \cdots X_n \vdash_M X_1 \cdots X_{i-1} Y p X_{i+1} \cdots X_n$$

若某 *ID* 是从另一个经有限步 *(*包括零步*)* 转移而得到的，记为 $\vdash_M^*$. 若 M 已知，简记为 $\vdash$ 和 $\vdash^*$.

续例 1. 设计识别 $\{0^n1^n \mid n \geq 1\}$ 的图灵机, 接受 0011 的 ID 序列.

$$
\begin{aligned}
q_0 0011 &\vdash Xq_1011 && \vdash X0q_111 && \vdash Xq_20Y1 \\
&\vdash q_2X0Y1 && \vdash Xq_00Y1 && \vdash XXq_1Y1 \\
&\vdash XXYq_11 && \vdash XXq_2YY && \vdash Xq_2XYY \\
&\vdash XXq_0YY && \vdash XXYq_3Y && \vdash XXYYq_3B \\
&\vdash XXYYBq_4B
\end{aligned}
$$

图灵机的语言

定义

如果 $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ 是一个图灵机，
则 M 接受的语言为

$$\mathbf{L}(M) = \{w \mid w \in \Sigma^*,\ q_0 w \vdash^* \alpha p \beta,\ p \in F,\ \alpha, \beta \in \Gamma^*\}.$$

定义

如果 L 是图灵机 M 的语言，即 $L = \mathbf{L}(M)$，
则称 L 是递归可枚举语言.

- 一般假定，当输入串被接受时，图灵机总会停机;
- 然而，对于不接受的输入，图灵机可能永远不停止.

图灵机的语言

定义

对接受和不接受的输入, 都保证停机的图灵机, 所接受的语言称为**递归语言**.

算法的形式化

保证停机的图灵机, 正是算法的好模型, 即算法概念的形式化.

整数函数计算器

- 例如, 将整数 $i \geq 0$ 表示为字符串 0^i;
- 若计算 k 个自变量 $i_1, i_2, \ldots, i_k$ 的函数 f, 用

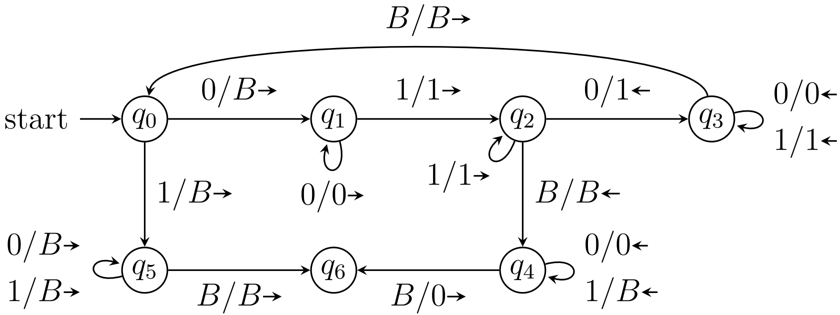
$$0^{i_1}10^{i_2}1\cdots10^{i_k}$$

  作为 TM M 的输入;
- 当 M 停机且输入带上为 0^m, 表示计算

$$f(i_1, i_2, \ldots, i_k) = m.$$

例 2. 设计整数真减法 ($\dot{-}$) 的图灵机

$$m \dot{-} n = \begin{cases} m-n & m \geq n \\ 0 & m < n \end{cases}.$$



图灵机的变形

状态中存储

有限控制器中可以存储有限个符号的图灵机:

$$M' = (Q', \Sigma, \Gamma, \delta, q_0', B, F')$$

其中 $Q' = Q \times \Gamma \times \cdots \times \Gamma$, $q_0' = [q_0, B, \cdots, B]$.

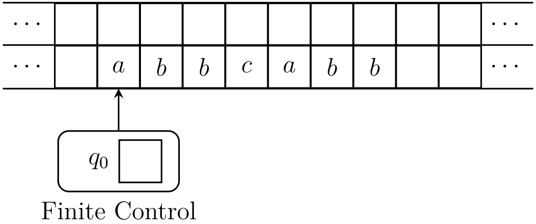
多道

多道图灵机:

$$M' = (Q, \Sigma, \Gamma', \delta, q_0, B', F)$$

其中 $\Gamma' = \Gamma \times \Gamma \times \cdots \times \Gamma$.

例 3. 利用状态中存储与多道设计 TM 识别 $L = \{wcw \mid w \in \{a,b\}^*\}$.



半无穷带图灵机

图灵机的输入输出带只有一侧是无穷的.

定理 39

半无穷带图灵机, 与图灵机等价.

多带图灵机

有穷控制器、k 个带头和 k 条带组成. 每个动作, 根据状态和每个带头符号:

❶ 改变控制器中的状态;

❷ 修改带头单元格中的符号;

❸ 每个带头独立的向左或右移动一个格, 或保持不动.

开始时, 输入在第 1 条带上, 其他都是空的.

定理 40

由多带图灵机接受的语言 L, 可被单带图灵机接受.

证明:

1. 用 $2k$ 道的单带图灵机 D 模拟 k 带图灵机 M;
2. D 用两道模拟 M 一带, 一道放置内容, 一道标记带头;
3. 为模拟 M 的一个动作, D 需要从左至右, 再从右至左, 各扫描一次;
4. 第一次扫描收集各带头处符号, 第二次更新带头符号和位置. □

图灵机的运行时间

定义

图灵机 M 在输入 w 上的**运行时间**, 为停机前移动的步数.

定义

图灵机 M 在所有长度为 n 的输入上, 运行时间关于 n 的最大值函数 $T(n)$, 称为 M 的**时间复杂度**.

- 只有保证停机的图灵机, 其时间复杂度 $T(n)$ 才有意义;
- 但是, 只有多项式时间的 $T(n)$, 才是计算机上实际可解的.

定理 41

单带图灵机 D 模拟 k 带图灵机 M 的 n 步移动, 需要使用 $O(n^2)$ 的时间.

证明:

1. M 移动 n 步, 带头相距不会超过 $2n$;
2. 而标记带头并调转方向至多需要 $2k$ 步;
3. 因此 D 模拟 M 的 1 步至多需要 $4n + 2k$ 步, 即 $O(n)$ 时间;
4. 因此模拟 n 步需要 $O(n^2)$ 时间. □

非确定图灵机 (NTM)

图灵机在每组状态 q 和带符号 X 的转移 $\delta(q, X)$, 可以有有限个选择:

$$\delta(q, X) = \{(q_1, Y_1, D_1), (q_2, Y_2, D_2), \cdots, (q_k, Y_k, D_k)\}.$$

- NTM 接受语言的方式, 与 NFA 和 PDA 是类似的
- 存在从初始 ID 到某接受状态 ID 的转移, 其他选择可以忽略

定理 42

如果 L 被非确定图灵机接受, 那么 L 被图灵机接受.

证明:

1. TM M 用控制器保存并用两条带模拟 NTM N 的动作:
   第 1 条带存储 N 未处理的 ID, 第 2 条带模拟 N 的带.
2. M 将第 1 条带最前端的 ID 复制到第 2 带, 若接受则停止;
3. 把当前 ID 可能的 k 个转移 ID 复制到第 1 条带的最末端;
4. 将第 1 带上最前端的 ID 抹掉, 从第 2 步重复.

证明 (续):

- 只有 N 进入接受的 ID 时, M 才会且一定会接受;
- 因为若 N 每步最多 m 个选择, 那么从初始 ID 经过 n 步最多可到的 ID 数量为

$$1 + m + m^2 + \cdots + m^n,$$

而 M 会 “先广搜索” 这些最多 nm^n 个的 ID. $\square$

TM 与 NTM

- NTM 的 n 步计算, TM 需要指数倍的时间 $O(m^n)$ 模拟, 但是否必然呢?
  — 仍是未知的
- NTM 以多项式时间解决的问题, TM 是否也能以多项式时间解决呢?

$$\mathrm{P} \stackrel{?}{=} \mathrm{NP}$$