计算模型①

Models of Computation

刘显敏
liuxianmin@hit.edu.cn

海量数据计算研究中心
HARBIN INSTITUTE OF TECHNOLOGY

计算理论—2023年春

什么是计算模型

什么是计算的理论模型

- 有效计算的严格数学定义
- 模型独立于具体的机器

为什么需要计算模型

- 算术公理化⇨希尔伯特第二问题⇨证明论、元数学⇨哥德尔不完备定理（1931）
- 丢番图方程的可解问题⇨希尔伯特第十问题⇨算法⇨不可判定性⇨马蒂亚塞维奇（1970）
- 什么是可以计算的、多久可以算完等
- 随机性、量子计算、并行计算等

图灵机的直观想法

回想一下如何计算加法和乘法

确定图灵机

阿兰·图灵于1936年提出**图灵机**，实现如下直观想法

- 有一个双向无穷带（infinite tape）存储信息
- 有一个带头（tape head）可以在带上移动、读写
- 无穷带用输入初始化，带头上有状态标记
- 如果需要存储信息，可以在带上写
- 如果要读取某位置的信息，带头需先移动到该位置
- 计算过程由一系列的移动和读写操作构成
- 计算过程中带头的状态不断改变
- 接受（accept）或拒绝（reject）状态即停止计算
- 计算可以一直进行下去，不停机（死机）

确定图灵机

定义 确定图灵机(Deterministic Turing Machine, DTM)

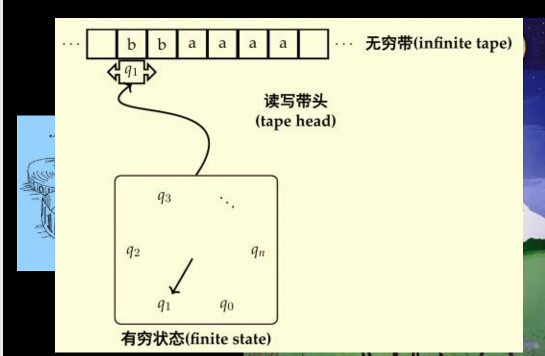
M由7元组构成，记作$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$，其中

- Q是有穷状态集合
- Σ是有穷输入符合集合
- Γ是有穷的带上符号集合，显然$\Sigma\subseteq\Gamma$
- q_0是图灵机的起始状态
- $F\subseteq Q$是停机(接受)状态集合
- $B\in\Gamma\setminus\Sigma$是空字符
- δ是形如$(Q\setminus F)\times\Gamma\rightarrow Q\times\Gamma\times\{R,L\}$的转移函数

转移函数是核心

- $\delta(p,a)=(q,b,L)$定义了一个动作
- 当机器处于状态p，并且带头所处的格子(cell)内容为a时，机器将a改成b，进入状态q，带头左移(L)

确定图灵机



确定图灵机

输入$w = w_1 w_2 \cdots w_n \in \Sigma^*$，图灵机M的计算方式如下

❶初始化时，w存储在M的连续n个格子里，其余部分是空白符，带头处于w_1格子处，状态为q_0

❷M开始运行后，根据δ的规则执行计算动作，计算一直持续，直到M进入接受状态，停机

❸如果M无法进入停机状态，M将永远一直运行下去

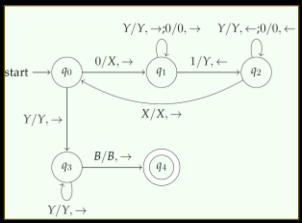
❹如果M无法找到可使用的转移规则，M拒绝

一个例子

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

$\delta(q_0, 0) = (q_1, X, \rightarrow)$
$\delta(q_0, Y) = (q_3, Y, \rightarrow)$
$\delta(q_1, 0) = (q_1, 0, \rightarrow)$
$\delta(q_1, 1) = (q_2, Y, \leftarrow)$
$\delta(q_1, Y) = (q_1, Y, \rightarrow)$
$\delta(q_2, 0) = (q_2, 0, \leftarrow)$
$\delta(q_2, Y) = (q_2, Y, \leftarrow)$
$\delta(q_2, X) = (q_0, X, \rightarrow)$
$\delta(q_3, Y) = (q_3, Y, \rightarrow)$
$\delta(q_3, B) = (q_4, B, \rightarrow)$

状态转移图



图灵机的计算描述

图灵机计算过程的瞬时描述(ID)

$$X_1 X_2 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n$$

- 图灵机的当前状态是q
- 图灵机的带头正在扫描X_i
- 图灵机带上的内容为$X_1 X_2 \cdots X_{i-1} X_i X_{i+1} \cdots X_n$

瞬时描述的移动，即图灵机的一步计算，表示为$\vdash$

- 如果有$\delta(q, X_i) = (p, Y, \leftarrow)$，那么有如下移动
  $X_1 \cdots X_{i-1} q X_i X_{i+1} \cdots X_n \vdash X_1 \cdots X_{i-2} p X_{i-1} Y X_{i+1} \cdots X_n$
- 特殊情况，瞬时描述的长度发生变化
  - 当$i = 1$时，$q X_1 X_2 \cdots X_n \vdash p B Y X_2 \cdots X_n$
  - 当$i = n$且$Y = B$时，$X_1 X_2 \cdots X_{n-1} q X_n \vdash X_1 X_2 \cdots X_{n-2} p X_{n-1}$

一个例子

多次移动（多步计算）可以表示为
$\alpha_1 q \beta_1 \vdash^* \alpha_2 p \beta_2$**或**$\alpha_1 q \beta_1 \vdash_M^* \alpha_2 p \beta_2$

图灵机的语言

给定图灵机M和字符串$w \in \Sigma^*$，如果存在M的瞬时描述序列$ID_1, ID_2, \ldots, ID_k$满足下列条件

- ID_1是图灵机M初始状态对应的瞬时描述
- ID_k对应图灵机M的某个接受状态
- 对任意$i \in [1, k-1]$，$ID_i \vdash ID_{i+1}$

则称M接受w

定义 图灵机M的语言(M识别的语言)

M接受的字符串集合称为M的语言，或M识别的语言，记作$L(M)$。

- 例子

$\{0110\}$　$\{0^n 1^n | n > 0\}$　$\{w\#w | w \in \{0,1\}^*\}$　$\{0^{2^n} | n \geq 0\}$

图灵机形式化定义的讨论

- 是否可以引入“拒绝”状态?
- 是否可以只有一个停机状态?
- 是否可以在移动动作中加入“停在原地”？
- 是否可以删除L(←)或者R(→)移动?
- 是否可以“无穷化”？
- 单向无穷带可以吗?
- 有穷带可以吗?
- ……

计算模型②

Models of Computation

刘显敏
liuxianmin@hit.edu.cn

海量数据计算研究中心
HARBIN INSTITUTE OF TECHNOLOGY

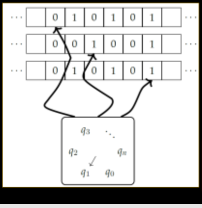
计算理论—2023年春

图灵机的变形

多带图灵机

- 图灵机有多条带，都有各自的带头
- 初始化时，输入在第一个带上，其它空白
- 转移函数

$$\delta: Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R, S\}^k$$



定理 多带与单带图灵机等价

令N是k-带图灵机，L(N)是N所识别的语言，则存在单带图灵机M满足L(M)=L(N)。

单带图灵机模拟多带图灵机
时间代价：平方级别

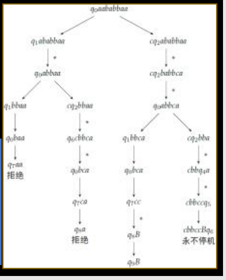
确定图灵机模拟非确定图灵机
时间代价：指数级别

非确定图灵机(不确定图灵机)

- 转移函数

$$\delta: Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$$

- 非确定图灵机有多种可能计算过程，对应多个瞬时描述转移
- w被接受，当且仅当存在转移序列使得$q_0 w \vdash^* \alpha p \beta$，$p \in F$

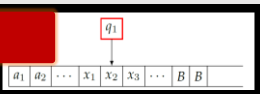


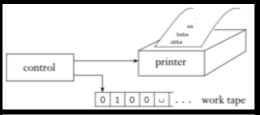
定理 确定与非确定图灵机等价

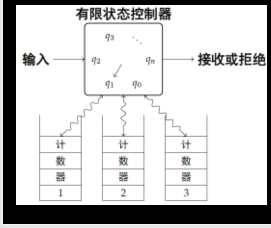
令N是非确定图灵机，L(N)是N所识别的语言，则存在确定图灵机M满足L(M)=L(N)。

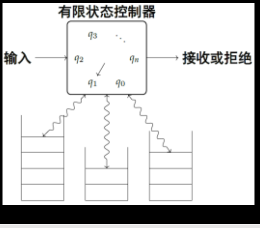
与标准图灵机等价

- 单向无穷带图灵机
- 多栈图灵机
- 计数图灵机
- 枚举图灵机









其它计算模型-随机访存图灵机

instruction	operand	semantics
load	j	$R_0 := R_j$
load	$= c$	$R_0 := c$
add	j	$R_0 := R_0 + R_j$
add	$= c$	$R_0 := R_0 + c$
sub	j	$R_0 := \max\{R_0 - R_j, 0\}$
sub	$= c$	$R_0 := \max\{R_0 - c, 0\}$
half		$R_0 := \lfloor \frac{R_0}{2} \rfloor$
jump	s	$\kappa := s$
jpos	s	if $R_0 > 0$ then $\kappa := s$
jzero	s	if $R_0 = 0$ then $\kappa := s$
halt		$\kappa := 0$

与标准图灵机等价

与标准图灵机等价-文法

- 一个文法由4元组构成$G = (V, \Sigma, R, S)$，V是字母集，Σ是终结符集，$S \in V \setminus \Sigma$是起始符，R是生成规则集合。

$$R \subseteq (V^*(V \setminus \Sigma)V^*) \times V^*$$

$$(u, v) \in R \text{ 记作 } u \to v$$

- 从符号S开始生成的字符串集合就是文法生成的语言
- 一个例子

生成语言$\{a^n b^n c^n | n \geq 1\}$的文法

$G = (V, \Sigma, R, S)$，$V = \{S, a, b, c, A, B, C, T_a, T_b, T_c\}$，$\Sigma = \{a, b, c\}$，$R$:

- $S \to ABCS, S \to T_c$
- $CA \to AC, BA \to AB, CB \to BC$
- $CT_c \to T_c c, CT_c \to T_b c, BT_b \to T_b b, BT_b \to T_a b, AT_a \to T_a a$
- $T_a \to \epsilon$

其它计算模型-函数

- 基本函数(ℕ)
  - 零函数：$zero_k(x_1, \dots, x_k) = 0$
  - 取值函数：$id_{k,j}(x_1, \dots, x_k) = x_j$
  - 后继函数：$succ(x) = x + 1$
- 函数的合成(composition)
  - 令函数$g: \mathbb{N}^k \to \mathbb{N}$，函数$h_1, \dots, h_k$形如$\mathbb{N}^l \to \mathbb{N}$，两者的合成

$$f(x_1, \dots, x_l) = g\big(h_1(x_1, \dots, x_l), \dots, h_k(x_1, \dots, x_l)\big)$$

- 函数的递归(recursion)
  - 令函数$g: \mathbb{N}^k \to \mathbb{N}$，函数$h: \mathbb{N}^{k+2} \to \mathbb{N}$，两者递归定义的函数

$$f(x_1, \dots, x_k, 0) = g(x_1, \dots, x_k)$$

$$f(x_1, \dots, x_k, m+1) = h\big(x_1, \dots, x_k, m, f(x_1, \dots, x_k, m)\big)$$

- 上述为基础递归函数

其它计算模型-函数

- 最小化函数
  - 令函数$g: \mathbb{N}^{k+1} \to \mathbb{N}$，$g$的最小化函数如下

$$f(x_1, \dots, x_k) = \begin{cases} \min\{m \mid g(x_1, \dots, x_k, m) = 1\} \\ 0 \text{ 如果}m\text{不存在} \end{cases}$$

- 上述为μ-递归函数

与标准图灵机等价

Church-Turing Thesis

- 1936-图灵

图灵机是“任意可能计算”的模型

- 合理的计算模型都是等价的，即与图灵机等价
- 图灵机与现代计算机能力相同，等价

算法等同于图灵机算法

- 丘奇-图灵论题并不是严格的数学表达，无法证明
- 丘奇-图灵论题的正确性来自于科学界的广泛认同

☛ **可计算理论**