10.4 图灵机

- 图灵机的基本模型
- 图灵机接受的语言

  ——递归可枚举语言
- 用图灵机计算函数

  ——部分可计算函数与可计算函数

问题的提出

1900年 D. Hilbert 在巴黎第二届数学家大会上提出著名的23个问题.

第10个问题:如何判定整系数多项式是否有整数根? 要求使用“有限次运算的过程”

1970 年证明不存在这样的判定算法, 即这个问题是不可判定的, 或不可计算的.

计算模型

从20世纪30年代先后提出

图灵机 **A.M.Turing, 1936年**

λ转换演算 **A.Church, 1935年**

递归函数 **K.Gödel, 1936年**

正规算法 **A.A.Markov, 1951年**

无限寄存器机器 **J.C.Shepherdson, 1963年**

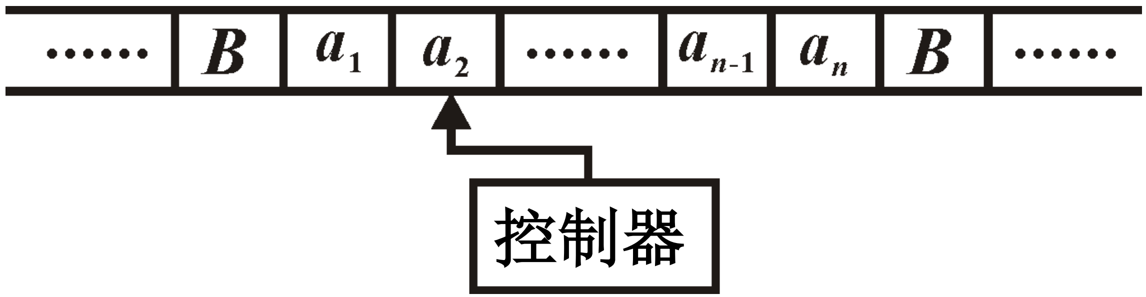
…

Church-Turing论题

已经证明这些模型都是等价的, 即它们计算的函数类 (识别的语言类) 是相同的.

Church-Turing论题: 直观可计算的函数类就是图灵机以及任何与图灵机等价的计算模型可计算 (可定义) 的函数类

图灵机的基本模型



定义 图灵机(TM) $M=\langle Q,\Sigma,\Gamma,\delta,q_0,B,A\rangle$, 其中

(1) 状态集合Q: 非空有穷集合;

(2) 输入字母表Σ: 非空有穷集合;

(3) 带字母表Γ: 非空有穷集合且$\Sigma\subset\Gamma$;

(4) 初始状态 $q_0\in Q$;

图灵机的基本模型(续)

(5) 空白符$B\in\Gamma-\Sigma$;
(6) 接受状态集$A\subseteq Q$;
(7) 动作函数δ是$Q\times\Gamma$到$\Gamma\times\{L,R\}\times Q$的部分函数,
即$\mathrm{dom}\delta\subseteq Q\times\Gamma$.

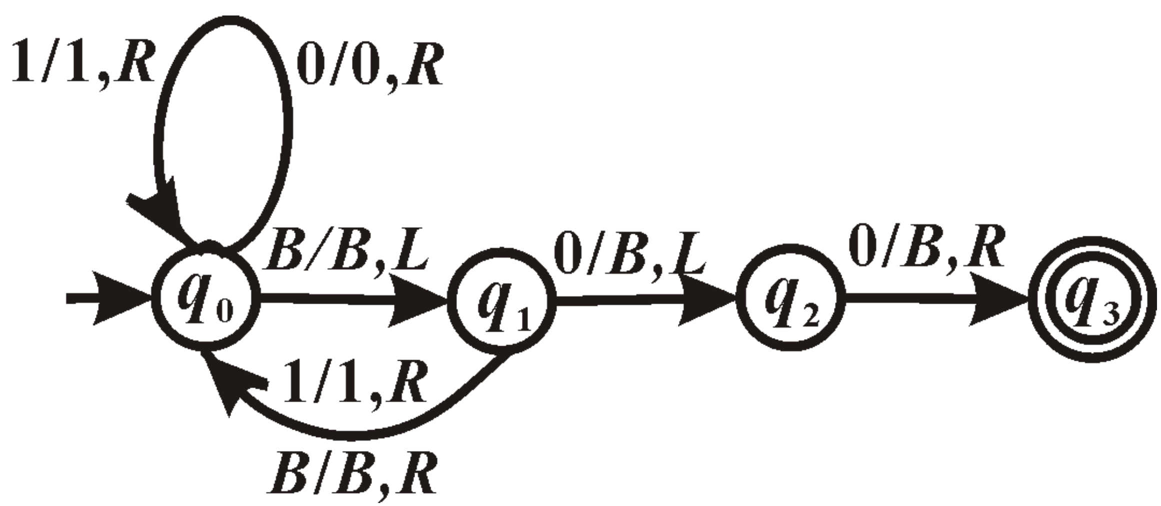
$\delta(q,s)=(s',R,q')$的含义: 当处于状态q, 读写头扫视符号s时, M的下一步把状态转移到q', 读写头把这个s改写成s', 并向右移一格;

$\delta(q,s)=(s',L,q')$的含义类似, 只是读写头向左移一格; 若$\delta(q,s)$没有定义, 则M停机.

一个TM M的实例

例1

δ	0	1	B
$\rightarrow q_0$	$(0,R,q_0)$	$(1,R,q_0)$	(B,L,q_1)
q_1	(B,L,q_2)	$(1,R,q_0)$	(B,R,q_0)
q_2	(B,L,q_3)	—	—
$*q_3$	—	—	—



图灵机的计算

格局: 带的内容, 当前的状态和读写头扫视的方格

$\sigma = \alpha q \beta$, 其中 $\alpha, \beta \in \Gamma^*$, $q \in Q$

初始格局 $\sigma_0 = q_0 w$, 其中 $w \in \Sigma^*$ 是输入字符串

接受格局 $\sigma = \alpha q \beta : q \in A$

停机格局 $\sigma = \alpha q s \beta : \delta(q,s)$ 没有定义

$\sigma_1 \vdash \sigma_2$: 从 σ_1 经过一步能够到达 σ_2, 称 σ_2 是 σ_1 的后继

$\sigma_1 \vdash^* \sigma_2$: 从 σ_1 经过若干步能够到达 σ_2

图灵机的计算(续)

计算: 一个有穷的或无穷的格局序列, 序列中的每一个格局都是前一个格局的后继.

$\forall w \in \Sigma^*$, M从$\sigma_0 = q_0 w$开始的计算有3种可能:

(1) 停机在接受格局, 即计算为$\sigma_0, \sigma_1, \ldots, \sigma_n$, 其中$\sigma_n$是接受的停机格局;

(2) 停机在非接受格局, 即计算为$\sigma_0, \sigma_1, \ldots, \sigma_n$, 其中$\sigma_n$是非接受的停机格局;

(3) 永不停机, 即计算为$\sigma_0, \sigma_1, \ldots, \sigma_n, \ldots$

图灵机接受的语言

定义 $\forall w \in \Sigma^*$, 如果M从$\sigma_0 = q_0 w$开始的计算停机在接受格局, 则称M接受输入串w. M接受的语言$L(M)$是M接受的所有输入串, 即$L(M)=\{w \in \Sigma^* \mid M\text{接受}w\}$.

例1 (续) M关于输入w=10100的计算:

$q_0 10100B \vdash 1q_0 0100B \vdash 10q_0 100B \vdash 101q_0 00B \vdash 1010q_0 0B$
$\vdash 10100q_0 B \vdash 1010q_1 0B \vdash 101q_2 0BB \vdash 101Bq_3 BB$

由于停机在接受格局, 故M接受10100.

$$L(M)=\{w00 \mid w \in \{0,1\}^*\}$$

图灵机接受的语言(续)

定义 能被图灵机接受的语言称作**递归可枚举的**, 记作**r.e.**

定理 语言L是r.e.当且仅当 L是 0 型语言.

图灵机与 0 型文法是等价的

用图灵机计算函数

Σ上的m元部分字函数: $(\Sigma^*)^m$的某个子集到Σ^*的部分函数

TM M计算的m元部分字函数f : 设输入字母表为Σ, $\forall x_1,\ldots,x_m \in \Sigma^*$, 如果$M$从初始格局$\sigma_0 = q_0 x_1 B \ldots x_m B$开始的计算停机(不管是否停机在接受状态), 从停机时带的内容中删去Σ以外的字符, 得到字符串y, 则 $f(x_1,x_2,\ldots,x_m)=y$; 如果M从初始格局σ_0开始的计算永不停机, 则$f(x_1,x_2,\ldots,x_m)$没有定义,记作 $f(x_1, x_2, \ldots , x_m)\uparrow$.

例1(续) M计算函数: $\forall x\in\{0,1\}^*$, $f(x) = \begin{cases} w, & \text{若} x = w00 \\ w1, & \text{若} x = w10 \\ \uparrow, & \text{否则} \end{cases}$

数论函数

数论函数: 自然数集合N上的函数

N上的m元部分函数

N上的m元全函数: 在N^m的每一点都有定义

例如 $x+y$是全函数, $x-y$是部分函数, 当$x<y$时, $x-y\uparrow$

一进制表示: 用1^x表示自然数x

例如 111表示3, 空串ε表示0

数论函数的一进制表示:字母表{1}上的字函数, 用一进制表示自然数

例如 $x+y$ 可表成 $f(1^x,1^y)=1^{x+y}$

递归函数

定义 设f是N上的m元部分函数, 如果图灵机M计算f的一进制表示, 即M的输入字母表为$\{1\}$, $\forall x_1,\ldots,x_m \in N$, 从初始格局 $\sigma_0 = q_0 1^{x_1} B 1^{x_2} B \cdots 1^{x_m} B$ 开始, 若$f(x_1,\ldots,x_m) = y$, 则M的计算停机, 且停机时带的内容(不计$\{1\}$以外的字符)为1^y; 若$f(x_1,\ldots, x_m)\uparrow$, 则$M$永不停机, 那么称$M$以一进制方式计算$f$.

定义 图灵机M以一进制方式计算的N上的m元部分函数称作部分递归函数, 或部分可计算函数; 部分递归的全函数称作递归函数, 或可计算函数.

递归函数(续)

例1(续) M以一进制方式计算

$$f(x)=\begin{cases} x/4, & \text{若}x\text{被4整除} \\ x/2, & \text{若}x\text{被2整除,但不被4整除} \\ \uparrow, & \text{否则} \end{cases}$$

这是一个部分递归函数.