第二章 可计算函数

1.源语言

指令	说明
x=x+1	变量加 1
x=x-1	变量减 1
TO A IF x≠0	若 x 不等于 0,跳转到 A;如果 x 等于 0,继续执行.IF 后边只能是不等于 0
TO A	无条件跳转到 A(可消去的)
y=x	赋值(可消去的)

约定:

① $x_1,x_2,\cdots,x_n$ 表示输入,$z_1,z_2,\cdots,z_n$ 表示临时变量,y 表示输出.

② 除了输入变量,其他变量的默认值是 0

③ 转向未定义标号或者没有下一条语句可执行,程序停止.

2.相关定理和推论

- 程序 p 对应的函数为φ_p,若有一函数 F=φ_p,则称 F 半可计算.
- 定义域为全域的函数称为全函数
- 部分可计算的全函数是可计算函数
- 非形式化描述:若有一个源语言程序能计算某个函数,即:
  - 当这个函数是全函数时,若有一个源语言程序在相同的输入下与这个函数有的相同的输出,则称这个函数是可计算函数;
  - 当这个函数不是全函数,若有一个源语言程序程序在函数有定义的时候与函数拥有相同的输出,在函数没有定义的时候不停机或者无定义,则称这个函数是部分可计算函数.

第三章 递归函数

1.基本算子

复合:y=f(g(x))是复合算子作用于 y=f(x)和 z=g(x)的结果.

递归:h(0)=m,h(t+1)= φ(h(t),t),h 函数是递归算子作用于φ和 m 的结果.

取极小:f(x1,x2,⋯,xn,z)为全函数,h(x1,x2,⋯,xn)=$\min_{z}\{f\ (\mathrm{x1,x2,\ldots xn,z})=0\}$,h 是取极小算子作用于函数 f 的结果.(只有当 f 是正则函数的时候,h 才是全函数)

- 函数值能取到 0 的函数称为正则函数

2.初始函数

后继函数:S(x)=x+1

零函数:n(x)=0

投影函数:$\mathrm{U}_{i}^{n}(x1,x2,\ldots,xn)=xi$(1≤i≤n)

- 由初始函数出发,使用**复合**算子和**递归**算子得到的函数称为原始递归函数,原始递归函数都是全函数
- 由初始函数出发,使用**复合**算子,**递归**算子和**对正则函数取极小**算子得到的函数称为递归函数,递归函数都是全函数
- 由初始函数出发,使用**复合**算子,**递归**算子和**取极小**算子得到的函数称为部分递归函数
- 原始递归∈递归∈部分递归

3.原始递归谓词和原始递归集合

- 谓词的特征函数在谓词为真的时候取 0,谓词为假的时候取 1.
- 给出集合 S,和自变量 x1⋯xn,若自变量属于 S,则此时集合 S 的特征函数为 0,若自变量不属于集合 S,则此时 S 的特征函数为 1.
- 谓词和集合的原始递归性等价于他们的特征函数是否原始递归.
- 谓词的原始递归性对于取反,合取,析取,特称量词,全称量词封闭.
- 集合的原始递归性对于取补,并集,交集封闭
- 分支定理:g 和 h 函数原始递归,p 为原始递归谓词,则函数 f 原始递归:f=$\begin{cases} g,p\text{为真} \\ \hbar.p\text{为假} \end{cases}$
- 受囿取极小:$\min_{t\le y}\{P(t,\ldots)\}$=$\begin{cases}\text{在 }0\sim y\text{范围内使谓词}P\text{为真的最小的}t\text{值} \\ 0,\qquad\qquad t\in(0,y]\text{时不能使谓词}P\text{为真}\end{cases}$

4.递归与可计算性

- 递归函数可计算
- 部分递归函数部分可计算

5. 可用的原始递归函数和原始递归谓词表

<table>
<tr><th>原始递归函数</th><th>原始递归谓词</th></tr>
<tr><td>
① 加法:x+y
② 点乘:x·y
③ 阶乘:x!
④ 指数:x^y
⑤ 前驱:P(x)=$\begin{cases} x-1, x\neq 0 \\ 0 \quad\ \ , x=0 \end{cases}$
⑥ 点减:x∸ y=$\begin{cases} x-y, x\geq y \\ 0 \quad\ \ , x<y \end{cases}$
⑦ 绝对值减:|x-y|=(x∸ y)+(y∸ x)
⑧ α函数:α(x)=$\begin{cases} 1, x=0 \\ 0, x\neq 0 \end{cases}$
⑨ n+1 元求和:$\sum_{t=0}^{y} f(x1,\ldots,xn,t)$
⑩ n+1 元求积:$\prod_{t=0}^{y} f(x1,\ldots xn,t)$
⑪ 判等函数:d(x,y)=$\begin{cases} 0, x=y \\ 1, x\neq y \end{cases}$
⑫ 除法取整:[y/x]=$\min_{t\leq y}\{x\cdot(t+1)>y\}$
⑬ 第 i 个质数:P_i
⑭ x/y 的余数:R(x,y)
⑮ x 的质因数分解中非零指数的个数:t(x)
⑯ x 的质因数分解中 P_i 的指数:$(x)_i$
⑰ 在 x 的质因数分解中最大的一个非零指数是第几个质数的指数:Lt(x)
⑱ 德哥尔函数:$[a_1,a_2,\cdots,a_n]=\prod_{i=1}^{n} P_i^{a_i}$
⑲ 哥 德 尔 数 乘 法 :$x=[a_1,a_2,\cdots,a_n]$,$y=[b_1,b_2,\cdots,b_m]$, 则 $x*y=[a_1,a_2,\cdots,a_n,b_1,b_2,\cdots,b_m]=\prod_{i=1}^{n} P_i^{a_i}\cdot\prod_{j=n+1}^{n+m} P_j^{b_{j-n}}$
⑳ 在 x 的质因数分解中指数为 a 的有几个:#(a,x)
21 配对函数;$<x,y>=\frac{1}{2}(x+y)(x+y+1)+y$
22 取左函数:l(z)=$\min_{x\leq z}\{(\exists y)_{\leq z}, z=<x,y>\}$
23 取右函数:r(z)= $\min_{y\leq z}\{(\exists x)_{\leq z}, z=<x,y>\}$
</td><td>
① X=y
② x > y
③ x≤y
④ y|x: y 整除 x 时为真
⑤ Prim(x):x 为质数时为真
⑥ GN(x):x 的质因数分解中没有零指数时为真
</td></tr>
</table>

第四章 PT 程序和图灵机

1.PT 机

① 双向带,带上符号为 1 和 B,初始状态下,指针指向左端第一个 1;

② $\bar{x}$表示 x 的带上表示,x 的带上表示就是在带上连续写 x+1 个 1;

③ 指令集:

原始指令	宏指令
RIGHT //指针右移一格 **LEFT** //指针左移一格 **WRITE 1** //在当前格中写 1 **WRITE B** //在当前格中写 B **TO A IF B** **TO A IF 1**	**TO A** **RIGHT TO NEXT B** **LEFT TO NEXT B** **WRITE B1** //写 B 右移写 1 右移 **指令[k]** //执行该指令 K 次

2.广义 PT 机

① 双向带,带上符号取自字母表:{S_0,S_1,⋯,S_n},其中 S_0=B,S_1=1;

② 带上表示和指令集与狭义 PT 机类似;

③ 哥德尔编码

a) 哥德尔编码用一个数描述一条广义 PT 机的带;

b) 第一步:字母表对应到指数上:{S_0,S_1,⋯,S_n}->{1,2,⋯,n+1}

c) 一条带的哥德尔编码 TAPE(x)=$\prod_{i=1}^{x+1} P_i^{第i个符号对应的指数}$

3.图灵机

① 双向带,带上符号取自字母表,并且每个图灵机有自己的状态集;

② 如果当前状态和当前注视符没有对应的指令,图灵机停机;

③ 四元组图灵机:[当前状态] [当前注视符] [左移/右移/输出符] [下一个状态];

④ 五元组图灵机:[当前状态] [当前注视符] [输出符] [左移或右移] [下一个状态].

4.PT 程序的编码:哥德尔数

① 指令编码:

指令	编码
RIGHT	1
LEFT	2
WRITE 1	3
WRITE B	4
TO A_i IF 1	2i+4
TO A_i IF B	2i+3

② 转换:

a) <标号代码,无标号指令代码>=x_j;

b) 整个程序的哥德尔数=$\prod_{j=1}^{n} P_j^{x_j}$.

5.图灵机和可计算性

- 函数的(部分)可计算性等价于他的 PT 程序(部分)可计算性
- 广义 PT 机等价于 PT 机
- PT 可计算函数可以被相同字母表的图灵机计算
- 四元组和五元组可以互相模拟
- 五元组可以被相同字母表的广义 PT 机程序模拟

第五章 半可计算性

1.半可计算谓词和半可计算集合

- **半可计算性针对的是谓词和集合,部分可计算性针对的是函数;**
- 如果能在有限步骤内判定一个元素是否属于某集合,则该集合是**递归**的;
- 若对于一个属于某集合的元素,存在算法能在有限步骤内判定为"是",而对于一个不属于该集合的元素,算法不能在有限步骤中给出回答,则称该集合是**递归可枚举**的;
- 若存在算法能在有限步骤内判定一个谓词的真假,则称这个谓词是**可判定**的;
- 若存在算法,当某谓词为真的时候,能在有限步骤中给出判定"是",而当谓词为假时,算法不能在有限步骤内终止,则称该谓词是**半可判定**的;
- $F(x)\downarrow$表示 F 在输入为 x 时有定义,$F(x)\uparrow$表示 F 在输入为 x 时无定义,当把这个符号视为谓词的时候,$F(x)\downarrow$为真当且仅当 F 在输入为 x 时有定义;
- 一个谓词是半可计算的当且仅当他取真值的定义域等于某部分可计算函数的定义域;
- 一个集合是半可计算的当且仅当他的特征谓词是半可计算的;

2.半可计算性的封闭特征

- 谓词的半可计算性对于析取,合取封闭;
- 集合的半可计算性对于并集, 交集封闭;
- 谓词$H(\bar{x})$是半可计算的,当且仅当存在一个可计算谓词 $C(\bar{x},y)$使得 $H(\bar{\boldsymbol{x}})\Leftrightarrow(\exists y)c(\bar{\boldsymbol{x}},y)$;
- $H(v,\bar{\boldsymbol{x}})$是半可计算谓词,则$(\exists v)H(v,\bar{\boldsymbol{x}})$也是半可计算谓词;
- $H(v,\bar{\boldsymbol{x}})$是半可计算谓词,则$(\forall v)_{\leq z}H(v,\bar{\boldsymbol{x}})$是半可计算谓词;

3.半可计算性与可计算性

- 可计算的谓词都是半可计算的;
- 可计算谓词的取反是半可计算的;
- $P(\bar{\boldsymbol{x}})$是可计算的当且仅当 $P(\bar{\boldsymbol{x}})$和$\sim P(\bar{\boldsymbol{x}})$都是半可计算的;
- 定义谓词 K(x),当且仅哥德尔数为 x 的 PT 机程序能停机时为真
  - K(x)是半可计算的,但不是可计算的;
  - ~K(x)不是半可计算的.
- 函数 F(x)部分可计算 等价于 谓词 V=F(x)是半可计算的
- 集合 W_z={x|使编码为 Z 的 PT 程序停机的输入 x}
- 半可计算集之集是可枚举的,枚举出来就是 $W_0,W_1,W_2,\cdots\cdots$
- 整数集之集是不可枚举的;

第六章 半图厄系统

1.图厄处理和图厄系统

- 字母表:D={a1,a2,⋯,an};
- 字:字母表中元素的有穷序列;D 中所有字的集合为 D*
- D 上的半图厄处理是产生式集合βi->β'i, βi 和β'i 是 D 上的字
- 规定∅->∅存在于所有的半图厄系统中且为第 0 条产生式,所以 $A\overset{0}{\Rightarrow}A$ 总是成立的;
- 设σ为半图厄处理,若 A->B 在其中,则 B->A 也在其中,则称σ为图厄处理;
- π=(P,A)称为(半)图厄系统,其中 P 为(半)图厄处理,A 称为公理或初始字,公理能推出的符号称为定理;
- 若半图厄系统中,对于字 W,最多只有一个字 U 使得 ($w \Rightarrow u$),(每个字最多只能推出一个其他的字)则该系统单演;

2.半图厄系统和半可计算集合

- 半图厄系统可以模拟图灵机
- M 为图灵机,$\emptyset(\mathrm{M})$表示让 M 停机的所有输入 X 的集合,π(P,A)为半图厄系统,T(π)表示π的所有定理的集合,则对于任意图灵机 M 都存在半图厄系统π使得:

$$\mathrm{X} \in \emptyset(\mathrm{M}) \Leftrightarrow \overline{X} \in T(\pi)$$

- 任意的半可计算集都可以由一个半图厄系统恰好生成(通过模拟图灵机生成集合中的数字的带上表示)

3.判定问题

- 特征谓词在其对应的判定问题的答案是"是"的时候取真
- 一个判定问题是(半)可判定的等价于其特征谓词是(半)可计算的;
- 图灵机停机问题(判定某一图灵机对于输入 X 是否能停机)是不可判定的;
- 存在图灵机,使得他对任意输入 X 的停机问题都是不可判定的
- 存在半图厄系统,使得判定任给的字是否是其定理这个问题是不可判定的;

第七章 图灵机

1.图灵机模型

- 图灵机 M 记作$M=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$
  - Q:状态有穷集;
  - Γ:带上可用符号的有穷集;
  - B: Γ中的空白符;
  - Σ: Γ中的非空白符组成的集合;
  - δ:次动作函数: δ(当前状态,当前注视符)=(下一个状态,输出符,左移/右移)
  - q_0:Q 中的初始状态;
  - F:Q 中的终止状态组成的集合.
- **使用图灵机计算某个函数:**初始状态下带上是输入值,当输入有意义的时候,图灵机最终停机到接受状态上,然后在带上留下函数的输出,当输入没有意义的时候,图灵机不停机或者停机到非接收状态上;
- **设计接受某个语言的图灵机:**初始状态下带上是一个符号串,假如这个符号串符合规则,那么图灵机在这上运行到最后应当停机到接受状态,如果不符合规则,图灵机应当不停机或者停机到非接收状态上;

2.各种图灵机

- 双向无限带
- 多带图灵机:多带,每条带上的指针可以独立移动(R/L)或者不动(D)
- 非确定图灵机
- 多维图灵机
- 多头图灵机
- 离线图灵机
  - 输入带是有限且只读的,以Ø开头,以$结尾;
  - 工作带是多带的且是单向的.

第八章 计算复杂性理论

1.复杂度定义

- 空间复杂度:离线图灵机 M 计算一个输入长度为 n 的问题是,在任意一条存储带上都之多扫视 S(n)个单元,称 M 是 S(n)空间有界的且空间复杂度为 S(n);
- 时间复杂度:双向多带图灵机 M 计算某个输入长度为 n 的问题时,在停机之前最多做 T(n)个动作,称 M 是 T(n)时间有界的且时间复杂度为 T(n).
- 空间复杂度 S(n)= $max(1,\lceil S(n)\rceil)$
- 时间复杂度 T(n)=$max(n+1,\lceil T(n)\rceil)$

2.空间可构造

- 称一个函数 S(n)是空间可构造的,如果有某个图灵机 M,M 是 S(n)空间有界的,且对于每个 n,都存在某个长度为 n 的输入,
- 空间可构造函数集包括 logn,n,n^K,2^n,n!
- 如果 $S_1(n)$和 $S_2(n)$是空间可构造的,则 $2^{S1(n)}$和 $S_1(n)^{S2(n)}$也是空间可构造的.
- 上述 M 至少对于一个长度为 n 的输入使用 S(n)个空间,如果存在 M'对于一切长度为 n 的输入,M'都使用 S(n)个空间,则称 S(n)为完全空间可构造的函数.

3.时间可构造

- 称函数 T(n)为时间可构造的,如果有某个多带图灵机 M,M 是 T(n)时间有界的,使得对于每个 n 都存在长度 n 的输入
- 上述 M 至少对于一个长度为 n 的输入使用 T(n)个动作,如果存在 M'对于一切长度为 n 的输入,M'都使用 T(n)个动作,则称 T(n)为完全时间可构造的函数.