Title Lecture 10

Date 2023.12.6　　　　Review

## Numerical function

$f: N^k \to N\ (k \geq 0)$ . A TM computes $f: N^k \to N$ 　　= 进制

if $n_1 \cdots n_k \in N$ , $M(bin(n_1), \cdots, bin(n_k)) = f(bin(n_1 \cdots n_k))$

- basic function
  1. zero function　$zero(n_1, \cdots, n_k) = 0$
  2. identity function　$id_{k,j}(n_1 \cdots n_k) = n_j$
  3. successor function　$succ(n) = n+1$

- operations
  1. composition　$h: N \to N$　$g: N \to N$　$f(x) = g(h(x))$

     $h_i: N^e \to N$　　$g: N^k \to N$

     $f(n_1, \cdots n_e) = g(h_1(n_1, \cdots, n_e), \cdots h_k(n_1, \cdots n_e))$
  2. recursive definition　$f(0) = 1$　$f(n+1) = h(f(n), n)$

     $g: N^k \to N$　　$h: N^{k+2} \to N$　　$f: N^{k+1} \to N$

$$\begin{cases} f(n_1, \cdots, n_k, 0) = g(n_1, \cdots, n_k) \\ f(n_1, \cdots, n_k, t+1) = h(n_1, \cdots, n_k, t, f(n_1 \cdots, n_k, t)) \end{cases}$$

- basic function + $\begin{cases} \text{composition} \\ \text{recursive def.} \end{cases}$ = primitive recursive func.

  可计算.

1. $plus2(n) = n+2$ : $succ(succ(n))$
2. $plus(m,n) = m+n$ : $\begin{cases} plus(m,0) = m = id_{1,1}(m) \\ plus(m, n+1) = plus(m,n)+1 \end{cases}$

相比h少参数 ← $= succ(plus(m,n))$

$h(m,n,plus(m,n)) = succ(id_{3,3}(m,n,plus(m,n)))$

定义递归时不一定需要完整的参数列表

3. $mult(m,n) = m \cdot n$　　利用此前结果展开

$$\begin{cases} mult(m,0) = 0 = zero(m) \\ mult(m,n+1) = mult(m,n)+m = plus(m, mult(m,n)) \end{cases}$$

4. constant function　　$f(n_1, \cdots, n_k) = c$　　$\underbrace{succ(\cdots(succ}_{c\text{次}}(zero(n_1, \cdots, n_k))\cdots)$
5. sign function　$sgn(n) = \begin{cases} 1, & n>0 \\ 0, & n=0 \end{cases}$

   $sgn(0) = zero(0)$　　$sign(n+1) = 1$ ⟶ constant function.
6. predecessor function　$pred(n) = n-1$　$pred(0) = 0$

$$\begin{cases} pred(0) = 0 = zero(0) \\ pred(n+1) = id_{2,1}(n, pred(n)) = n \end{cases}$$

7. $m \dot{-} n = max(m-n, 0)$ 非负减法.

$$\begin{cases} m \dot{-} 0 = m \\ m \dot{-} (n+1) = m \dot{-} n \dot{-} 1 = pred(m \dot{-} n) \end{cases}$$

- primitive recursive func + $\begin{cases} \text{composition} \\ \text{recursive def.} \end{cases}$ ⟶ primitive recursive func.

8. $position(n) = sgn(n)$　9. $iszero(n) = 1 \dot{-} position$

若 predict P primitive recursive , $\neg P = 1 \dot{-} P$ 也是

10. $geq(m,n) = \begin{cases} 1, & m \geq n \\ 0, & m < n \end{cases} = iszero(m \dot{-} n)$
11. $eq(m,n) = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} = gep(m,n) \wedge gep(n,m)$

$P \vee q = 1 \dot{-} iszero(p+q)$

# Title L 10

- $f(n_1,\cdots,n_k) = \begin{cases} g(n_1,\cdots,n_k) & \text{if } p(n_1,\cdots,n_k) \\ h(n_1,\cdots,n_k) & \text{else} \end{cases}$ ← predicate

若 g·h·p primitive recursive，f 也是 $f = p\cdot g + (1 \dot{-} p)\cdot h$

⑫ $rem(m,n) = m \% n$ $\begin{cases} rem(0,n) = 0 \\ rem(m+1,n) = \begin{cases} 0 & \text{if 整除} \\ rem(m,n)+1 \end{cases} \end{cases}$

整除 ⟺ $eq(rem(m,n), pred(n))$

⑬ $div(m,n) = \lfloor m/n \rfloor \quad (n \neq 0)$

$\begin{cases} div(0,n) = 0 \\ div(m+1,n) = \begin{cases} 1+div(m,n) & \text{if } n \text{ 整除 } m+1 \\ div(m,n) & \text{else} \end{cases} \end{cases}$

⑭ $digit(m,n,p) = a_{m-1}$

把 n 转为 p 进制 $a_k p^k + \cdots + a_1 p_1 + a_0 = n$

$= div(rem(n,p^m), p^{m-1})$

⑮ $sum_f(m,n) = \sum_{k=0}^{n} f(m,k)$ f 原始递归

不可直接拆作 n个 f 相加，因为 n 也为 input.

$\begin{cases} sum_f(m,0) = f(m,0) \\ sum_f(m,n+1) = sum_f(m,n) + f(m, succ(n)) \end{cases}$

⑯ $g_p(n) = \begin{cases} 1 & \exists n' \leq n,\ p(n') = 1 \\ 0 & \text{otherwise} \end{cases}$ p 原始递归 predicate

$g_p(n) = p(0) \vee \cdots \vee p(n) = positive\left(\sum_{n'=0}^{n} p(n')\right) = positive(sum_p(n))$

⑰ $h_p(n) = \begin{cases} 1 & \forall n' \leq n\ p(n') = 1 \\ 0 & \text{otherwise} \end{cases}$ 与 sum 类似

$h_p(n) = h(0) \wedge \cdots \wedge h(n) = positive(mult_p(n))$

- PR = {f : f is primitive recursive func}

PR ≠ C = {f : f is computable}

PR → decidable  C → undecidable

证明：C undecidable. 反证

假定 decidable，C' = {一元可计算 numerical func}

C' 可判定 ⇒ C' 字典序图灵可遍历 $(g_1 \cdots g_n \cdots)$

M = on input n  1. enumerate C' to get $g_n$  2. compute $g_n(n)$

3. return $g_n(n)+1$

计算了 $g^*(n) = g_n(n)+1$  $g^*$ 一元可计算，但 $g^* \notin C'$

矛盾，故 C 不可判定. 判定可计算的过程不可计算

- minimalization

$g: N^{k+1} \to N$  $f(n_1,\ldots,n_k) = \begin{cases} \min n_{k+1} \text{ s.t. } g(n_1\cdots n_{k+1}) = 1 & \text{若 } n_{k+1} \text{ 存在} \\ 0 & \text{若不存在} \end{cases}$

找对应的 $n_{k+1}$

f is a minimalization of g, $\mu m[g(n_1,\ldots,n_k,m)=1]$

- A function g is minimalizable if g is computable.

and $\forall n_1,\ldots,n_k\ \exists m \geq 0,\ g(n_1,\ldots,n_k,m) = 1$

$log(m,n) = \lceil \log_{m+2}(n+1) \rceil$ ← 找 min p 使 $(m+2)^p \geq n+1$

$\mu p[geq((m+2)^p, n+1) = 1]$

若 $g(n_1 \cdots n_k, n_{k+1})$ minimalizable $\mu m[g(n_1\cdots n_k,m)=1]$ 可计算 但"g 是否 minimalizable"不可判定

- μ-recursive = basic functions + $\begin{cases} \text{composition} \\ \text{recursive def} \\ \text{minimalization of minimalizable func.} \end{cases}$