

MATHEMATICS FOR INFORMATION SCIENCE

NO.2 PRIMITIVE RECURSIVE FUNCTION

Tatsuya Hagino

hagino@sfc.keio.ac.jp

Slides URL

<https://vu5.sfc.keio.ac.jp/slide/>

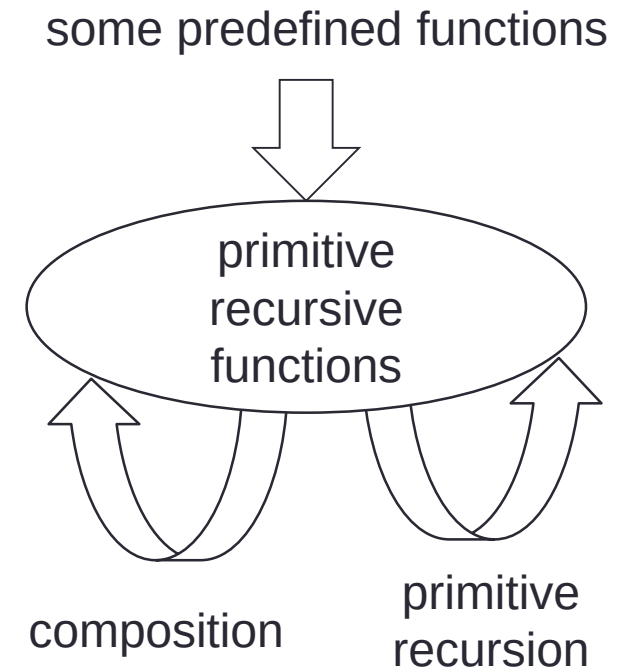
So far

- What is computation?
 - Computation = what computers can calculate
 - Computable functions = mathematical functions which computers can calculate
 - Computability = whether mathematical functions are computable or not
- While program and flow chart are equivalent.
- Any program can be converted into a while program with only one while.

Primitive Recursive Function

Definition :

- Primitive recursive functions consist of the followings:
 1. some predefined functions
 2. composition of primitive recursive functions
 3. functions defined by primitive recursion



Predefined Primitive Recursion Functions

- The following three functions are primitive recursive functions:

1. $zero : N^0 \rightarrow N$

- $zero() = 0$
- Always return 0

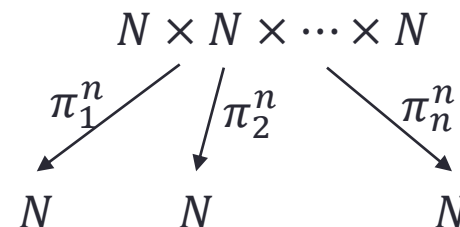
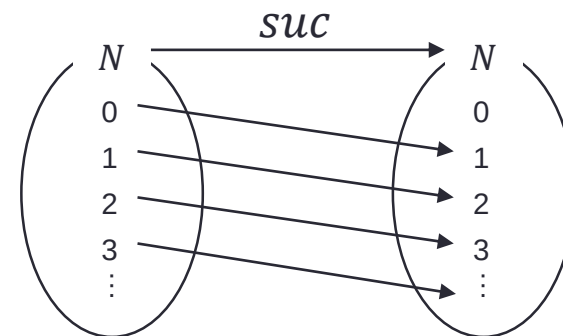
2. $suc : N \rightarrow N$

- $suc(x) = x + 1$
- next number of x
- $suc(2) = 3$
- $suc(100) = 101$

3. $\pi_i^n : N^n \rightarrow N$

- $\pi_i^n(x_1, \dots, x_n) = x_i$
- i th element of n given elements
- projection
- $\pi_1^2(x, y) = x$
- $\pi_2^2(x, y) = y$

- $N^0 = \{ \cdot \}$
- $N^1 = N$
- $N^2 = N \times N = \{(x, y) | x \in N, y \in N\}$
- $N^3 = N \times N \times N$
- $N^n = N \times N \times N \times \dots \times N$



- $$A \xrightarrow{h} B \xrightarrow{g} C$$
- $\curvearrowright g \circ h$

-
- $$\begin{array}{ccc}
 N^n & \xrightarrow{h_1} N \times N \times \cdots \times N & \xrightarrow{g} N \\
 & \xrightarrow{h_2} & \\
 & \vdots & \\
 & \xrightarrow{h_m} & \\
 N^n & \xrightarrow{f} N &
 \end{array}$$

Example 1. Primitive Recursive Function

- $id(x) = x$ $id : N \rightarrow N$
 - The identity function is primitive recursive.
 - $id(x) = \pi_1^1(x)$
- $one() = 1$ $one : N^0 \rightarrow N$
 - Function which always return 1 is primitive recursive.
 - $one() = suc(zero())$
- $two() = 2$ $two : N^0 \rightarrow N$
 - Function which always return a constant is primitive recursive.
- $add2(x) = x + 2$ $add2 : N \rightarrow N$
 - Function which adds 2 is primitive recursive.
 - $add2(x) = suc(suc(x))$

Primitive Recursion

- Given primitive recursive functions $g : N^n \rightarrow N$ and $h : N^{n+2} \rightarrow N$, $f : N^{n+1} \rightarrow N$ is also primitive recursion:
 - $f(x_1, \dots, x_n, 0) = g(x_1, \dots, x_n)$
 - $f(x_1, \dots, x_n, \text{succ}(y)) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y))$
- It is called that the function is defined by **primitive recursion**.
- If $n = 0$, when $g : N^0 \rightarrow N$ and $h : N \times N \rightarrow N$ are primitive recursive, $f : N \rightarrow N$ is primitive recursive:
 - $f(0) = g()$
 - $f(\text{succ}(y)) = h(y, f(y))$
- Define $f(x)$ when x is 0 and when x is not 0:
 - when x is 0, the value is constant $g()$
 - when x is not 0, x can be written as $x = \text{succ}(y)$, and $f(x)$ can be calculate from y and $f(y)$ by $h(y, f(y))$
 - If y is 0 then $f(y)$ is $g()$, but if not, repeat the case of $f(\text{succ}(y))$ again until y becomes 0
- $f(1) = f(\text{succ}(0)) = h(0, f(0)) = h(0, g())$
- $f(2) = f(\text{succ}(1)) = h(1, f(1)) = h(1, h(0, g()))$
- $f(3) = f(\text{succ}(2)) = h(2, f(2)) = h(2, h(1, h(0, g())))$

Example 2. Primitive Recursive Function

- $pred: N \rightarrow N$ $pred(x) = x - 1$
 - returns previous number (reverse of suc)

- $pred(0) =$

- $pred(suc(y)) =$

- $pred(1) =$

- $pred(5) =$

Primitive Recursion ($n = 0$)

- $f(0) = g()$
- $f(suc(y)) = h(y, f(y))$

Example 3. Primitive Recursive Function

- $add: N^2 \rightarrow N$

$$add(x, y) = x + y \quad \text{Addition}$$

- $add(x, 0) =$

- $add(x, suc(y)) =$

Primitive Recursion ($n = 1$)

- $f(x, 0) = g(x)$

- $f(x, suc(y)) = h(x, y, f(x, y))$

- $add(1, 1) =$

- $add(3, 2) =$

Example 4. Primitive Recursive Function

- $sub: N^2 \rightarrow N$

$$sub(x, y) = x - y \quad \text{Subtraction}$$

- $sub(x, 0) =$

- $sub(x, suc(y)) =$

Primitive Recursion ($n = 1$)

- $f(x, 0) = g(x)$

- $f(x, suc(y)) = h(x, y, f(x, y))$

- $mul: N^2 \rightarrow N$

$$mul(x, y) = x \times y \quad \text{Multiplication}$$

- $mul(x, 0) =$

- $mul(x, suc(y)) =$

Primitive Recursive Function

- **Definition:**

- **Primitive recursive functions** are:
 - The following three functions are primitive recursive functions:
 - $zero : N^0 \rightarrow N$ $zero() = 0$
 - $suc : N \rightarrow N$ $suc(x) = x + 1$ (next number of x)
 - $\pi_i^n : N^n \rightarrow N$ $\pi_i^n(x_1, \dots, x_n) = x_i$

For short, we use 0 for $zero()$

- Composition of primitive recursive functions are primitive recursive:
 - When $g : N^m \rightarrow N$ and $h_i : N^n \rightarrow N$ ($i = 1, 2, \dots, m$) are primitive recursive, $f : N^n \rightarrow N$ is also primitive recursive:

$$f(x_1, x_2, \dots, x_n) = g(h_1(x_1, x_2, \dots, x_n), \dots, h_m(x_1, x_2, \dots, x_n))$$
- Define a function using **primitive recursion**:
 - When $g : N^n \rightarrow N$ and $h : N^{n+2} \rightarrow N$ are primitive recursive, $f : N^{n+1} \rightarrow N$ is also primitive recursive:

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

$$f(x_1, \dots, x_n, suc(y)) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y))$$

Summation and Product

- **Lemma:**

- If $f(x_1, \dots, x_n, y)$ is primitive recursive, the following functions are also primitive recursive:

$$g(x_1, \dots, x_n, y) = \sum_{z=0}^{y-1} f(x_1, \dots, x_n, z)$$

$$h(x_1, \dots, x_n, y) = \prod_{z=0}^{y-1} f(x_1, \dots, x_n, z)$$

- If $n = 0$,
 - $g(y) = f(0) + f(1) + \dots + f(y-1)$
 - $h(y) = f(0) \times f(1) \times \dots \times f(y-1)$

- **Proof:**

- $g(x_1, \dots, x_n, 0) =$
- $g(x_1, \dots, x_n, \text{succ}(z)) =$
- $h(x_1, \dots, x_n, 0) =$
- $h(x_1, \dots, x_n, \text{succ}(z)) =$

Primitive Recursive Predicate

- **Definition:**

- A predicate $p(x_1, \dots, x_n): N^n \rightarrow \{T, F\}$ is **primitive recursive**, when its **characteristic function** $C_p(x_1, \dots, x_n): N^n \rightarrow N$ is primitive recursive.
 - $C_p(x_1, \dots, x_n) = 1$ when $p(x_1, \dots, x_n)$ is T
 - $C_p(x_1, \dots, x_n) = 0$ when $p(x_1, \dots, x_n)$ is F

- **Example:**

- Predicate $x = 0$ is primitive recursive:
 - $C_{=0}(x) =$
- Predicate $x \leq y$ is primitive recursive:
 - $C_{\leq}(x, y) =$
- Predicate $x = y$ is primitive recursive:
 - $C_{=}(x, y) =$

Logical Formula

• Lemma:

- When $p(x_1, \dots, x_n)$, $q(x_1, \dots, x_n)$, $r(x_1, \dots, x_n, z)$ are primitive recursive predicates, the following predicates are also primitive recursive:
 - $p(x_1, \dots, x_n) \wedge q(x_1, \dots, x_n)$
 - $p(x_1, \dots, x_n) \vee q(x_1, \dots, x_n)$
 - $\neg p(x_1, \dots, x_n)$
 - $\forall z < y (r(x_1, \dots, x_n, z)) \equiv \forall z (z < y \Rightarrow r(x_1, \dots, x_n, z))$
 - $\exists z < y (r(x_1, \dots, x_n, z)) \equiv \exists z (z < y \wedge r(x_1, \dots, x_n, z))$

Proof:

- $C_{P \wedge Q}(x_1, \dots, x_n) =$
- $C_{\neg P}(x_1, \dots, x_n) =$
- $C_{P \vee Q}(x_1, \dots, x_n) =$
- $C_{\forall R}(x_1, \dots, x_n, y) =$
- $C_{\exists R}(x_1, \dots, x_n, y) =$

Example

- $x > y$ is primitive recursive:
 - $(x > y) \equiv$
- $divisible(x, y) \equiv ``x$ is divisible by y'' is primitive recursive:
 - $divisible(x, y) \equiv$
- $prime(x) \equiv ``x$ is a prime number'' is primitive recursive:
 - $prime(x) \equiv$

Minimum

- Lemma:**

- If $p(x_1, \dots, x_n, z)$ is a primitive recursive predicate, the following function is primitive recursive:

$$\mu_{z < y} p(x_1, \dots, x_n, z) = \min(\{z \mid p(x_1, \dots, x_n, z)\} \cup \{y\})$$

the minimum number z which makes $p(x_1, \dots, x_n, z)$ hold and less than y .

- Proof:**

- Let $g(x_1, \dots, x_n, z)$ be the characteristic function of $\forall v < z (\neg p(x_1, \dots, x_n, v))$. Then,

$$\mu_{z < y} p(x_1, \dots, x_n, z) = \sum_{z=0}^{y-1} g(x_1, \dots, x_n, z + 1)$$

if $n = 1$
 $\mu_{z < y} P(x, z)$

z	0	1	2	3	4	5	6	...	$y - 1$	y
$P(x, z)$	F	F	F	T	F	T	T	...	F	T
$\neg P(x, z)$	T	T	T	F	T	F	F	...	T	F
$\forall v < z (\neg P(x, v))$	T	T	T	T	F	F	F		F	F
$g(x, z)$	1	1	1	1	0	0	0	...	0	0

$$\mu_{z < y} P(x, z) = \sum_{z=0}^{y-1} g(x, z + 1) = \sum_{z=1}^y g(x, z) = 3$$

Corollaries

Corollary: $div(x, y) = x \div y$ is primitive recursive.

- $div(x, y) = \mu_{z < x} (x < (y \times (z + 1)))$

- $div(5, 2) =$

Theorem: Arithmetic operations (addition, subtraction, multiplication and division) are primitive recursive.

Corollary: $pr(x) = \text{"}x\text{th prime number"}$ is primitive recursive.

- $pr(0) = 2$

- $pr(suc(x)) = \mu_{y < pr(x)!+2} (pr(x) < y \wedge prime(y))$

Summary

- Primitive recursive function
 - *zero*
 - *suc*
 - π_i^n
 - function composition
 - define by primitive recursion
- Primitive recursive predicate
- Arithmetic operations are primitive recursive.