Recursive Functions

Rongen Lin

Nanjing University

December 12, 2019

Why we study it?

- Computability theory
- Number theory
- Proof theory

Primitive Recursive Functions

Computable

Computable

A given function g is said to be computable if it "can be computed by some program".

Primitive Recursive Functions

Definition

Primitive Recursive Functions

A function is called primitive recursive if it can be obtained from the initial functions by a finite number of applications of composition and recursion.

Initial functions

Constant function: $n(x) = 0$
Successor function: $S(x) = x + 1$
Projection function: $P_i^n(x_1, \ldots, x_n) = x_i$.

Composition

Let f be a function of k variables and let $g_1, \ldots, g_k$ be functions of n variables. Let

$$h(x_1, \ldots, x_n) = f(g_1(x_1, \ldots, x_n), \ldots, g_k(x_1, \ldots, x_n)).$$

Then h is said to be obtained from f and $g_1, \ldots, g_k$ by composition.

Theorem I

If h is obtained from the computable functions $f, g_1, \ldots, g_k$ by composition, then h is computable.

Recursion I

Suppose k is some fixed number and

$$h(0) = k,$$
$$h(t+1) = g(t, h(t)),$$

then h is said to be obtained from g by recursion.

Recursion II

$$h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n),$$
$$h(x_1, \ldots, x_n, t+1) = g(t, h(x_1, \ldots, x_n, t), x_1, \ldots, x_n).$$

Theorem II

Let h be obtained from g (and f), and let g (and f) be computable. Then h is also computable.

Primitive Recursive Functions

Theorems

Theorem III

Every primitive recursive function is computable.

Theorem IV

Not all computable functions are primitive recursive.

Primitive Recursive Functions

Examples

$x + y$

The recursion equations for $f(x, y) = x + y$ are

$$f(x, 0) = x,$$
$$f(x, y + 1) = f(x, y) + 1.$$

We can rewrite these equations as

$$f(x, 0) = P_1^1(x),$$
$$f(x, y + 1) = S(P_2^3(y, f(x, y), x)).$$

Primitive Recursive Functions

Examples

$x \cdot y$

The recursion equations for $f(x, y) = x \cdot y$ are

$$f(x, 0) = 0,$$
$$f(x, y + 1) = f(x, y) + x.$$

We can rewrite these equations as

$$f(x, 0) = n(x),$$
$$f(x, y + 1) = g(P^3_2(y, f(x, y), x), P^3_3(y, h(x, y), x)).$$

Here $g(x_1, x_2)$ is $x_1 + x_2$.

Primitive Recursive Functions

Examples

$x!$

The recursion equations for $h(x) = x!$ are

$$0! = 1,$$
$$(x+1)! = x! \cdot S(x).$$

We can rewrite these equations as

$$h(0) = 1,$$
$$h(t+1) = g(t, h(t)),$$
$$g(x_1, x_2) = S(x_1) \cdot x_2.$$

Primitive Recursive Functions

Examples

$P(x)$

The predecessor function $P(x)$ is defined as follows:

$$P(x) = \begin{cases} x - 1 & x \neq 0 \\ 0 & x = 0. \end{cases}$$

The recursion equations for $P(x)$ are

$$P(0) = 0,$$
$$P(t+1) = t.$$

Primitive Recursive Functions

Examples

$x \dot{-} y$

The function $x \dot{-} y$ is defined as follows:

$$x \dot{-} y = \begin{cases} x - y & x \geq y \\ 0 & x < y. \end{cases}$$

The recursion equations for $x \dot{-} y$ are

$$x \dot{-} 0 = x,$$
$$x \dot{-} (t+1) = P(x \dot{-} t).$$

Primitive Recursive Functions

Examples

$|x - y|$

$|x - y|$ is primitive recursive since

$$|x - y| = (x \dot{-} y) + (y \dot{-} x).$$

$\alpha(x)$

The function $\alpha(x)$ is defined as follows:

$$\alpha(x) = \begin{cases} 1 & x = 0 \\ 0 & x \neq 0. \end{cases}$$

$\alpha(x)$ is primitive recursive since

$$\alpha(x) = 1 \dot{-} x.$$

Or we can simply write the recursion equations:

$$\alpha(0) = 1.$$
$$\alpha(t+1) = 0.$$

Primitive Recursive Functions

Examples

$\neg x$

$$\neg P = \alpha(P)$$

$x \wedge y$

$$x \wedge y = x \cdot y$$

$x \vee y$

$$x \vee y = \neg(\neg x \wedge \neg y)$$

Primitive Recursive Functions

Examples

$x = y$

$$d(x, y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

It is primitive recursive since

$$d(x, y) = \alpha(|x - y|).$$

$x \neq y$

$$e(x, y) = \alpha(d(x, y))$$

$x \leq y$

$$\alpha(x \dot{-} y)$$

$x < y$

$$\neg(y \leq x)$$

Primitive Recursive Functions

Examples

If-Else

$$f(x_1, \dots, x_n) = \begin{cases} g(x_1, \dots, x_n) & P(x1, \dots, x_n) \\ h(x_1, \dots, x_n) & \text{otherwise} \end{cases}$$

It is primitive recursive because

$$f(\dots) = g(x_1, \dots, x_n) \cdot P(x_1, \dots, x_n) + h(x_1, \dots, x_n) \cdot \alpha(P(x_1, \dots, x_n)).$$

Primitive Recursive Functions

Examples

$\sum$

The function $h(x_1, \ldots, x_n, m)$ is defined as follows:

$$h(x_1, \ldots, x_n, m) = \sum_{i=0}^{m} f(x_1, \ldots, x_n, i).$$

The recursion equations are

$$h(x_1, \ldots, x_n, 0) = f(x_1, \ldots, x_n, 0),$$
$$h(x_1, \ldots, x_n, y+1) = g(h(x_1, \ldots, x_n, y), f(x_1, \ldots, x_n, y+1)),$$
$$g(x, y) = x + y.$$

A complicated example

The function g is defined as follows:

$$g(x, k_1, \ldots, k_r, C_1, \ldots, C_r, f) = \begin{cases} f(x) & \nexists i : k_i = x \\ C_i & \exists i : k_i = x. \end{cases}$$

It is primitive recursive since

$$g(x, k_1, \ldots, k_r, C_1, \ldots, C_r, f) = f(x)\alpha(\sum_{i=1}^{r} d(x, k_i)) + \sum_{i=1}^{r} C_i d(x, k_i).$$

Primitive Recursive Functions

Examples

Forall

$$(\forall t)_{\leq y} P(t, x_1, \ldots, x_n) \Leftrightarrow \left[\prod_{t=0}^{y} P(t, x_1, \ldots, x_n)\right] = 1$$

Exist

$$(\exists t)_{\leq y} P(t, x_1, \ldots, x_n) \Leftrightarrow \left[\sum_{t=0}^{y} P(t, x_1, \ldots, x_n)\right] \neq 0$$

Minimalization

$$g(y, x_1, \ldots, x_n) = \sum_{u=0}^{y} \prod_{i=0}^{u} \alpha\left(P(t, x_1, \ldots, x_n)\right)$$

$$\min_{t \leq y} P(t, x_1, \ldots, x_n) = \begin{cases} g(y, x_1, \ldots, x_n) & \text{if} \quad (\exists t)_{\leq y} P(t, x_1, \ldots, x_n) \\ 0 & \text{otherwise} \end{cases}$$

$\mathrm{Prime}(x)$ p_n $\lfloor \frac{x}{y} \rfloor$ $x \bmod y$ $\phi(x)$ $\mu(x)$ ...

Primitive Recursive Functions

Theorem IV

Not all computable functions are primitive recursive.

The PR functions of one argument can be **computably enumerated**. This means that we can use some Gödel numbering to encode definitions of PR function as numbers. Let f_n denote the n-th unary PR function. Now define the function ev by $ev(i,j) = f_i(j)$. Suppose ev were PR, then the function g defined by $g(i) = S(ev(i,i))$ would also be PR. But there is some number n such that $g = f_n$, then

$$g(n) = S(ev(n,n)) = S(f_n(n)) = S(g(n))$$

gives a contradiction. So ev is not PR.

μ-recursive function
Definition

μ-recursive function

The μ-**recursive functions** (or **general recursive functions**) are partial functions that take finite tuples of natural numbers and return a single natural number. They includes the initial functions and is closed under composition, primitive recursion and the μ operator.

Minimization

$$\mu(f)\,(x_1, \ldots, x_k) = z \Longleftrightarrow \begin{array}{ll} f(i, x_1, \ldots, x_k) > 0 & \text{for } i = 0, \ldots, z-1 \\ f(z, x_1, \ldots, x_k) = 0 & \end{array}.$$

μ-recursive function

Comparison

- The μ-recursive functions are closely related to PR functions, and their inductive definition builds upon that of the PR functions.
- The PR functions are a subset of the μ-recursive functions.

μ-recursive function

Ackermann function

Definition

$$A(m, n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

Its value grows rapidly, even for small inputs. For example, $A(4, 2)$ is an integer of 19729 decimal digits (equivalent to $2^{65536}-3$).
Since the function $f(n) = A(n, n)$ grows very rapidly, its inverse function, f^{-1}, grows very slowly. This **inverse Ackermann function** f^{-1} is usually denoted by α. This inverse appears in the time complexity of some algorithms, such as the disjoint-set data structure and Chazelle's algorithm for minimum spanning trees.

Theorem V

the Ackermann function is not primitive recursive.

μ-recursive function

Ackermann function

Lemma I

$$A(m, n) \geq n + 1.$$

Definition of Ackermann functions

$$A(m, n) = \begin{cases} n + 1 & \text{if } m = 0 \\ A(m - 1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

Proof.

$$A(0, n) = n + 1 \geq n + 1$$
$$A(m + 1, 0) = A(m, 1) \geq 1 + 1 > 0 + 1$$
$$A(m + 1, n + 1) = A(m, A(m + 1, n)) \geq A(m + 1, n) + 1 \geq (n + 1) + 1$$

□

Lemma II

$$A(m, n+1) > A(m, n).$$

Definition of Ackermann functions

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

Proof.

$$A(0, n+1) = n+2 \geq n+1 = A(0, n)$$
$$A(m+1, n+1) = A(m, A(m+1, n)) > A(m+1, n)$$

□

μ-recursive function

Ackermann function

Lemma III

$$A(m+1, n) \geq A(m, n).$$

Definition of Ackermann functions

$$A(m, n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1, 1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m, n-1)) & \text{if } m > 0 \text{ and } n > 0 \end{cases}$$

Proof.

$$A(m+1, n) \geq n+2$$

$$A(m+1, n+1) = A(m, A(m+1, n)) > A(m, n+1)$$

□

Corollary

$$f(m_1, n_1) > f(m, n) \quad \text{when} \quad m_1 > m, n_1 \geq n.$$

μ-recursive function

Ackermann function

Lemma IV

For any PR function $g(x_1, \ldots, x_k)$, we have

$$\exists m : (\forall a_1 \ldots, a_k : g(a_1, \ldots, a_k) < A(m, u)),$$

where $u = \max\{a_1, \ldots, a_k\}$.

Proof.

$$n(n) = 0 < n + 1 = A(0, n).$$

$$S(n) = n + 1 < n + 2 = A(1, n).$$

$$P^k_m(x_1, \ldots, x_k) = x_m < u + 1 = A(0, u).$$

$$g(\ldots) < A(m_0, \max\{A(m_1, u), \ldots, A(m_r, u)\}) \leq A(\tilde{m}, A(\tilde{m}, u))$$

$$< A(\tilde{m}, A(\tilde{m} + 1, u)) = A(\tilde{m} + 1, u + 1) \leq A(\tilde{m} + 2, u).$$

$$h(t) < A(m, t), g(x) < A(m + 1, x),$$

$$g(x + 1) = h(g(x)) < A(m, g(x)) < A(m, A(m + 1, x)) = A(m + 1, x + 1),$$

$$A(m + 1, x) \leq A(m + 1, u).$$

□

μ-recursive function

Ackermann function

Theorem V

the Ackermann function is not primitive recursive.

Proof.

Suppose $A(m, n)$ is PR, there exists m_0 that

$$A(m, n) < A(m_0, \max\{m, n\}).$$

Let $m = n = m_0$, then $A(m_0, m_0) < A(m_0, m_0)$ gives a contradiction. □

The Founding(Primitive) Titan must brainwash people by Primitive Recursive functions.