Algorithms and recursive functions

Teodor Rus

rus@cs.uiowa.edu

The University of Iowa, Department of Computer Science

Algorithms

- The Turing machine has been introduced as the formal concept representing the intuitive notion of an algorithm
  Turing thesis: all what can be computed can be computed by a Turing machine
- The normal algorithm has been introduced as a formal concept representing the intuitive notion of algorithm
  Markov thesis: all what can be computed can be computed by a normal algorithm
- Here we introduce recursive functions as yet another formal concept representing the intuitive notion of algorithm
  Church thesis: all what can be computed can be computed by recursive functions

Sets

- Sets: $\emptyset$, finite sets given by enumerating their elements, $\{e_1, \ldots, e_n\}$, and arbitrary sets given by properties of their elements, $\{e|P(e)\}$.
- If A is a set then $a \in A$ means “a is an element of A" and $a \notin A$ means “a is not an element of A"
- If A, B are sets then $A \subseteq B$ if $\forall a \in A[a \in B]$; $A = B$ iff $A \subseteq B$ and $B \subseteq A$.
- Set constructors: For A, B sets, $A \cup B$, $A \cap B$, $A \setminus B$, $\mathcal{C}(A)$, and $A \times B = \{\langle a, b\rangle | a \in A, B \in B\}$ are sets.
- If $A_1, A_2, \ldots, A_n$ are sets then $(((A_1 \times A_2) \times A_3) \times \ldots \times A_n) = A_1 \times A_2 \times A_3 \times \ldots \times A_n$ is a set.

Functions

- Let A, B be sets and $f : A \rightarrow B$ a function. Then we have $Domain(f) = \{a \in A | f(a) \in B\}$, $Range(f) = \{b \in B | \exists a \in A \wedge f(a) = b\}$
- f is partial if $Domain(f) \subset A$; f is total or everywhere if $Domain(f) = A$;
- If $f, g : A \rightarrow B$ are functions, then $f = g$ iff $Domain(f) = Domain(g)$ and $Range(f) = Range(g)$.
- n-place functions and operations: $A \times A \times \ldots \times A$, n times A, is denoted A^n; $f : A^n \rightarrow B$ is an n-place function and $f : A^n \rightarrow A$ is an n place operation.

Language of functions

- Alphabet: object symbols, functional symbols, separators
  - object symbols, such as $a, b, x, y, \ldots$, accompanied by indices
  - letters with upper, and possible lower, indices such as f^1, g^2, f^1_0, f^2_1, $\ldots$.
    f^n is called an n-place functional symbol
  - separators are parentheses, (,), and comma, “,".
- Terms: string of symbols constructed by the usual rules from object symbols, functional symbols, and separators.

Example terms

x, $f(x)$, $g(x, a)$, $g(f(x), g(a, x))$, etc.

Question: is there a CFG grammar describing this language?

Term value

- **Value of an object symbol:**
  - Assume that a function $\mathcal{V}$ from the set of object symbols to a non-empty set called the *fundamental set* is give
  - For x an object symbol $\mathcal{V}(x)$ is called the value of x
- **Value of an n-place functional symbol**:
  - Assume that a function $\mathcal{F}$ maps the function symbols into the set of operations define on the fundamental set
  - For $f(x_1, \ldots, x_n)$ a term and $a_i = \mathcal{V}(x_i)$, $1 \leq i \leq n$, $\mathcal{F}(f(a_1, \ldots, a_n))$ is the value of the term $f(x_1, \ldots, x_n)$ for a_1,. . .,a_n the values of its arguments.

Using induction

- The value of a term τ of length 1 is the value of the object symbol forming this term
- Let τ be a term of length $|\tau| > 1$. Then τ has the form $f(\tau_1, \ldots, \tau_n)$ for some functional symbol f^n and τ_i terms of length smaller than $|\tau|$
- By induction we can assume that term τ_i has value v_i, $1 \leq i \leq n$.
- Then $\mathcal{V}(f(\tau_1, \ldots, \tau_n)) = \mathcal{V}(f)(\mathcal{V}(\tau_1), \ldots, \mathcal{V}(\tau_n))$

Example: if $\mathcal{V}(x) = 2$, $\mathcal{V}(y) = 3$ and $\mathcal{F}(+) = addition$ then $\mathcal{V}(x + y) = 5$.

Example term values

Assume that the fundamental set is $\{0, 1, 2, \ldots\}$, $+$ is addition, $-$ is subtraction, $:$ is division, and ϵ is multiplication. Then we have:

- $\mathcal{V}((3x + y) : y + 4) = 7$ for $\mathcal{V}(x) = 2$, $\mathcal{V}(y) = 3$
- $\mathcal{V}(5 - (3 + 3))$, $\mathcal{V}(2 : 3)$, $\mathcal{V}(5 + (2 : 3))$ are undefined

Note: we may drop $\mathcal{V}$ in the expression $\mathcal{V}(\tau)$ where context decides that value of τ is required

Derived operations

Having at hand several operations on some set X, we can, with the aid of terms, write down an infinite set of new operations defined on X by:

- Let $f_1, \ldots, f_n$ be functional symbols denoting given operations on X. Moreover, let a_1,...,a_r be given elements in X.
- Construct arbitrary terms using symbols f_1,...,f_n, elements a_1,...,a_r, and auxiliary object symbols x_1,...,x_n.
- Given arbitrary values to x_1,...,x_n, each term $\tau(x_1, \ldots, x_n)$ becomes an n-place operations on X called a derived operation.

Abbreviations for terms

- If x is an object symbol and f an one-place functional symbol then the term $f(x)$ may be written fx, xf, or x^f
- For x, y object symbols, f a 2-place functional symbol, the term $f(x, y)$ may be written xfy, fxy, or xyf.
- If terms τ_1 and τ_2 are of length 1 then the parentheses are omitted when handling them, i.e., $\tau_1 + \tau_2$, $\tau_1\tau_2$, $\tau_1 - \tau_2$ rather than $(\tau_1) + (\tau_2)$, $(\tau_1)(\tau_2)$, $(\tau_1) - (\tau_2)$, will be used
- A number is further understood to be a natural number in $N = \{0, 1, 2, \ldots\}$, and a partial function $N \times \ldots \times N = N^k$ into N will be called a *numerical partial function*

Initial functions

The numerical functions s^1, o^n, I^n_m defined by:

- $s^1(x) = x + 1$
- $o^n(x_1, \ldots, x_n) = 0$
- $I^n_m(x_1, \ldots, x_n) = x_m$, $1 \leq m \leq n$, $n = 1, 2, \ldots$

Note: for s^1, o^1 we shall write s, o.

Order of object symbols

A term containing object symbols, x,y,z represents a uniquely defined derived operation only in the case when the object symbols are ordered in a defined way

Example: $2x + y$ represents a function that assign to the pair $\langle 1, 3\rangle$ the number 5 only if x is the first variable and y the second variable.

Constants and variables

Object and functional symbols encountered in terms may be divided in tow categories:

1. *Individual symbols:* are symbols whose values are fixed
2. *Variable symbols:* are symbols whose values are not fixed

Examples: If fundamental set is N, the given operation on N is $+$ and number 1 is fixed then all linear functions $b_0 + b_1x_1 + \ldots + b_nx_n$, $b_i \in N$, $b_1 > 0$, $1 \leq i \leq n$ are termal operations; if only $+$ is give and 1 is not fixed then only homogeneous linear functions $b_1x_1 + \ldots + b_nx_n$, $b_i \in N$, $b_i \in N$, $b_i > 0$, $1 \leq i \leq n$ are termal

Fundamental computable operators

- Operations on numerical functions are called operators
- **Property:** upon applying an operator to functions which are computable in the intuitive sense we obtain functions which are computable in the intuitive sense
- The two fundamental computable operators are: composition and recursion

Partial recursive functions

Definition: functions obtained from initial functions $s(x) = x + 1$, $o(x) = 0$, $I^n_m(x_1 \ldots, x_n) = x_m$ using composition and recursion operators are called partial recursive functions

Church thesis: class of partial recursive functions coincides with class of functions computable by algorithms or machines

Composition operator

- Let f_1^m,...,f_n^m by n arbitrary partial functions of m variables defined on some set A with values in a set B.
- Suppose that $f^n : B \to C$ is also a partial function of n variables.
- Then partial function $g : A \to C$ of m variables defined by $g(x_1, \ldots, x_m) = f(f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m))$ is the function obtained by the composition operator (also called substitution operator) from function $f, f_1, \ldots, f_m$.

Notation: substitution operator is denoted by S^{n+1}

Algebra of functions

- Let $\mathcal{F}^n$ be the collection of all partial numerical functions of n variables
- The operator $\mathbf{S}^{n+1}$ is an everywhere defined function $\mathbf{S}^{n+1} : \mathcal{F}^n \times \mathcal{F}^m \times \ldots \times \mathcal{F}^m \to \mathcal{F}^m$
- If $\mathcal{F}$ is the set of all partial numerical functions of arbitrary number of variables then $\mathbf{S}^{n+1}$ can be considered a partial function of $n+1$ variables defined on $\mathcal{F}$ with values in $\mathcal{F}$.
- The term $\mathbf{S}^{n+1}(f, f_1, \ldots, f_m)$ has as the value an m-place function iff f is an n-place function and the values of the functional symbols f_1,...,f_m are partial functions of m variables.

Examples

1. $\mathbf{S}^3(I_1^2, I_1^3, I_2^3) = I_1^2(I_1^3, I_2^3) = I_1^3$;
   $\mathbf{S}^3(I_1^2, I_1^3, I_2^2) = undefined$ (I_1^3, I_2^2 have different arities);
2. $\mathcal{B} = \langle \mathcal{F}, S^2, S^3, \ldots \rangle$ is a partial algebra of $\mathcal{F}$
3. Let $f_1^{n_1}$,. . .,$f_s^{n_s}$ be arbitrary numerical functions of n_1,. . .,n_s variables. The partial functions that can be obtained by substitution from $f_1^{n_1}$,. . .,$f_s^{n_s}$ and the function I_m^n, $(m, n = 1, 2, \ldots)$ are called *elementary functions* with respect to $f_1^{n_1}$,. . .,$f_s^{n_s}$.
4. Collection of elementary functions with respect to $f_1^{n_1}$,. . .,$f_s^{n_s}$ in the algebra $\mathcal{B}$ is the subalgebra of $\mathcal{B}$ generated by $f_1^{n_1}$,. . .,$f_s^{n_s}$, I_m^n.

Theorem

A necessary and sufficient condition for an n-place partial numerical function f to be elementary relative to $f_1,\ldots,f_s$ is to be possible to represent f as a term τ written with the aid of $f_1,\ldots,f_s$ and certain object variables $x_1,\ldots,x_n$, part of which could occur in τ factiously.

Proof idea

A relative elementary partial function f has two representations:

- **Operator representation**: the value of a term in algebra $\mathcal{B}$ for which $\mathbf{S}^i$ serve as functional symbol and f_1,. . .,f_s,I^n_m serve as object symbols
- **Termal representation:** the term (not the value of a term) composed of functional symbols f_1,. . .,f_s,I^n_m and certain object variables (perhaps fictitious)

Proof: by induction, showing that termal representation can be transformed

in operator representation and vice-versa.

Proof

Let f be the value of the operator τ.

1. If $|\tau| = 1$ then τ is either f_i and then termal representation of f is $f_i(x_{i_1}, \ldots, x_{i_n})$, or I^n_j and then termal representation of f is x_j.
2. If $|\tau| > 1$ then τ has the form $\mathbf{S}^{k+1}(\tau^1, \tau_1, \ldots, \tau_k)$ where τ^1,τ_1,...,τ_k are operator terms of length less than $|\tau|$
3. The induction hypothesis says that termal representations $c = c(x_1, \ldots, x_k)$, $c_i = c_i(x_1, \ldots, x_n)$, $1 \leq i \leq k$ of the terms τ^1,τ_1,...,τ_k are known.
4. Then $c(c_1(x_1, \ldots, x_n), \ldots, c_k(x_1, \ldots, x_n))$ is the termal representation of τ.

Proof, continuation

Suppose that n-place function f is defined in terms of its termal representation $\mathcal{T}(x_1, \ldots, x_n)$.

1. If $|\mathcal{T}(x_1, \ldots, x_n)| = 1$ then $\mathcal{T}$ has the form x_j and hence f is the value of the operator term I^n_j
2. If $|\mathcal{T}(x_1, \ldots, x_n)| > 1$ then $\mathcal{T}$ has the form $f_i(\mathcal{T}_1, \ldots, \mathcal{T}_{n_i})$ where $|\mathcal{T}_i| < |\mathcal{T}|$, $1 \leq i \leq n_i$.
3. The induction hypothesis is that every n-place partial function of x_1,. . .,x_n which is representable by a term $\mathcal{T}_j$, has an operator term form $\mathcal{A}_j$, $1 \leq j \leq n_i$. Then it is clear that the operator term representation of f is $\mathbf{S}^{n_i+1}(f_i, \mathcal{A}_1, \ldots, \mathcal{A}_{n_i})$

Example

Suppose that the usual functions $+$ and $\times$ are given.

Then the function of x_1,x_2,x_3 having termal representation $x_1 \times x_2 + x_3$ is the value of the operator term $S^3(+, S^3(\times, I^3_1, I^3_2), I^3_3)$.

Primitive recursion operator

1. Suppose that n-place partial numerical function g and $n+2$-place partial numerical function h are given
2. We say that $n+1$-place partial function f arises from the functions g and h by *primitive recursion* if for all natural number values of x_1,...,x_n we have:

$$\begin{aligned} f(x_1, \ldots, x_n, 0) &= g(x_1, \ldots, x_n) \\ f(x_1, \ldots, x_n, y+1) &= h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)) \end{aligned}$$

Notation: $f = \mathbf{R}(g, h)$

Example

If $n = 0$ then we have: a 1-place function f arises by primitive recursion from a constant a (a zero-place function) and the two-place partial function h if:

$$\begin{aligned} f(0) &= a \\ f(n+1) &= h(n, f(n)) \end{aligned}$$

Question: does there exist, for all partial functions g,h of n and $n+2$ variables, a partial function f of $n+1$ variables defined by the primitive recursion operator?

Answer

Since the domain of the function g,h,f is the set of natural numbers the answer is affirmative.

Note:

- f is uniquely defined by the equalities

$$
\begin{aligned}
f(x_1, \ldots, x_n, 0) &= g(x_1, \ldots, x_n) \\
f(x_1, \ldots, x_n, 1) &= h(x_1, \ldots, x_n, 0, f(x_1, \ldots, x_n, 0)) \\
\ldots & \quad \ldots \\
f(x_1, \ldots, x_n, m+1) &= h(x_1, \ldots, x_n, m, f(x_1, \ldots, x_n, m))
\end{aligned}
$$

- if for certain x_1,...,x_n,t, the value $f(x_1, \ldots, x_n, t)$ is undefined then values $f(x_1, \ldots, x_m, y)$, for $y \geq t$ are also undefined

Observation

If we “know" how to find the values of the functions g and h then the value of the function $f = \mathbf{R}(g, h)$ can be computed with a procedure of a completely “mechanical" nature.

Namely: to find the value of $f(a_1, \ldots, a_n, m+1)$ it suffices to successively find the numbers:

$b_0 = g(a_1, \ldots, a_n)$

$b_1 = h(a_1, \ldots, a_n, 0, b_0)$

$\ldots$

$b_{m+1} = h(a_1, \ldots, a_n, m, b_m)$

Primitive recursive function

Let Σ be a given set of arbitrary partial functions.

- A partial function f is called primitive recursive relative to Σ if f can be obtained from the functions in Σ and initial functions s, o, I^n_m by a finite number of operations of substitution and primitive recursion
- If functions in Σ are everywhere defined then primitives recursive functions relative to Σ are everywhere defined.
- Operations $\mathbf{S}^{n+1}$ and $\mathbf{R}(g, h)$ applied to partial functions which are primitive recursive relative to Σ yield as result primitive recursive functions relative to Σ

Examples

- Initial functions $s(x) = x + 1$, $o^1(x) = 0$, $I_1^1(x) = x$, and $I_m^n(x_1, \ldots, x_n) = x_m$, $m, n = 1, 2, \ldots$ are primitive recursive
- For the n-place function $o^n(x_1, \ldots, x_n)$ we have the representation $o^n = S^2(o^1, I_1^n)$ and hence o^n is primitive recursive
- An arbitrary constant a (seen as an n-place function $f^n = a$) admits a representation in the form of the term $s(s(\ldots s(o^n(x_1, \ldots, x_n) \ldots))$ and therefore is primitive recursive
- The two-place function $f(x, y) = x + y$ primitive recursive because it arises from primitive recursive function $I_1^1(x)$, $h(x, y, z) = z + 1$ by the recursion scheme:

$$\begin{aligned} x + 0 &= I_1^1(x) \\ x + (y + 1) &= (x + y) + 1 = s(x + y) \end{aligned}$$

More examples

- The two-place function $x \times y$ (written $x.y$) is primitive recursive defined by $g(x) = o(x)$, $h(x, y, z) = z + x$ by the scheme:

$$\begin{aligned} x.0 &= o(x) \\ x.(y+1) &= x.y + x \end{aligned}$$

- The two-place function x^y (where $x^0 = 1$) is primitive recursive defined by $g(x) = 1$, $h(x, y, z) = z.x$ by the scheme:

$$\begin{aligned} x^0 &= 1 \\ x^{y+1} &= x^y.x \end{aligned}$$

Sign functions

- One-place functions $sg(x)$ and $\overline{sg}(x)$ are primitive recursive defined by the scheme:

$$
\begin{aligned}
sg(0) &= 0 \\
sg(x+1) &= 1 \\
\overline{sg}(0) &= 1 \\
\overline{sg}(x+1) &= 0
\end{aligned}
$$

Difference functions

- The one-place function $x \dot{-} 1$ is primitive recursive defined by o^1 and I^2_1 by the scheme:

$$\begin{aligned} 0 \dot{-} 1 &= o^1 \\ (x+1) \dot{-} 1 &= x \end{aligned}$$

- The two-place function $x \dot{-} y$ defined by

$$x \dot{-} y = \begin{cases} x - y & \text{if } x \geq y. \\ 0 & \text{if } x < y \end{cases} \quad (1)$$

  is primitive recursive defined by I^1_1 and $h(x, y, z) = z \dot{-} 1$

- The two place function $|x - y| = (x \dot{-} y) + (y \dot{-} x)$ is primitive recursive

Algebraic view

The system $\mathcal{B} = \langle \mathcal{F}, \mathbf{R}, \mathbf{S}^2, \mathbf{S}^3 \ldots \rangle$ is a partial algebra

- The collection $\mathcal{F}_{pr}$ of all primitive recursive functions is a subalgebra of $\mathcal{B}$ generated by the functions o,s, I^n_m
- If Σ is a system of functions from $\mathcal{F}$ then the collection of functions primitive recursive relative to Σ is a subalgebra of $\mathcal{B}$ generated by functions of Σ and initial functions o,s,I^n_m.
- A function is primitive recursive iff it is the value of a term which is representable with individual object symbols o,s,I^n_m and the functional symbols $\mathbf{R}$, $\mathbf{S}^2$, $\mathbf{S}^3$,$\ldots$.

Operator and functional terms

- Operator terms are well-formed words in the alphabet $\{\mathbf{R}, \{\mathbf{S}^i | i > 1\}, o, s, I^n_m\}$ and symbols that denote functions from Σ.
- Representation of a primitive recursive function as an operator term is the standard (constructive) definition of this function
- The proof of primitive recursiveness of a function f consists of showing how to construct the operator term whose value is f.

Example proof

Proofs that truncated differences are primitive recursive:

$$
\begin{aligned}
x \dot{-} 1 &= R(o, I^2_1(x)) \\
x \dot{-} y &= R(I^1_1, I^3_3 \dot{-} 1) = R(I^1_1, S^2(R(o, I^2_1), I^3_3))
\end{aligned}
$$

Operation of minimalization

- Consider an n-place partial numerical function f and assume that its value, where defined, is computable.
- Fix the values of arguments x_1,...,x_{n-1} and consider the equation $f(x_1, \ldots, x_{n-1}, y) = x_n$
- To find solutions $y \in N$ we can compute the values $f(x_1, \ldots, x_{n-1}, y)$ for $y = 0, 1, 2, \ldots$
- The smallest value of y for which $f(x_1, \ldots, x_{n-1}, y) = x_n$ is denoted by $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$

Note

The process performing the operation of minimization will continue indefinitely if the following cases:

- the value $f(x_1, \ldots, x_{n-1}, 0)$ is not defined
- the values $f(x_1, \ldots, x_{n-1}, y)$ for $y = 0, 1, \ldots, a-1$ are defined but are different from x_n and the value $f(x_1, \ldots, x_{n-1}, a)$ is not defined
- the values $f(x_1, \ldots, x_{n-1}, y)$ for $y = 0, 1, 2 \ldots$ are defined and are different from x_n

In all these cases the operation of minimization is undefined.

Examples

1. $x - y = \mu_z(y + z = x)$
2. $\mu_x(sg\ x = 1) = 1$, $\mu_y(y \dot{-} x = 0) = 0$
3. $\mu_y(y(y - (x+1)) = 0)$ is undefined since $(0 - (x+1))$ is undefined. Note that the equation $y(y - (x+1)) = 0$ has the solution $y = x + 1$ but this does not coincide with the value of the expression $\mu_y(y(y - (x+1)) = 0)$.

Note

- The last example above shows that for a partial function $f(x_1, \ldots, x_{n-1}, y)$ the expression $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$ is not the smallest solution of the equation $f(x_1, \ldots, x_{n-1}, y) = x_n$
- If $f(x_1, \ldots, x_{n-1}, y)$ is defined everywhere and the equation $f(x_1, \ldots, x_{n-1}, y) = x_n$ has a solution then $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$ is the smallest solution for $f(x_1, \ldots, x_{n-1}, y) = x_n$.

Observation

The value of expression $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$ depends on the choice of the values for x_1,...,x_{n-1},x_n and therefore it is a partial function of x_1,...,x_n.

Function symbol denoting the operator that maps function f into the function $\mu_y(f(x_1, \ldots, x_{n-1}, y) = x_n)$ is $\mathbf{M}f$

Note: if f is 1-place then $\mathbf{M}f = f^{-1}$ and is called the *inverse* of f, i.e. $f^{-1}(x) = \mu_y(f(y) = x)$

Examples

$$sg^{-1}x = \begin{cases} x & \text{if } x = 0, 1 \\ undefined & \text{if } x > 1 \end{cases}$$

$$s^{-1}x = \begin{cases} x - 1 & \text{if } x > 0 \\ undefined & \text{if } x = 0 \end{cases}$$

The operator term $\mathrm{M}(+) = \mathbf{S}^3(-, I_2^2, I_1^2)$ represents $y - x = \mu_z(y + z = x)$

Note: notation f^{-1} cannot be used for many-placed function. In these cases notation used if $\mathrm{M}f$ and is called *minimalization operation*.

More notations

1. We may also encounter expressions of the form:
   $\mu_y(f(x_1, \ldots, x_n, y) = g(x_1, \ldots, x_n, y))$
   $\mu_y(f(x_1, \ldots, x_n, y) \neq g(x_1, \ldots, x_n, y))$
2. Values of expressions (1) are assumed to be those coinciding with the values of expressions:
   $\mu_y(|f(x_1, \ldots, x_n, y) - g(x_1, \ldots, x_n, y)| = 0)$
   $\mu_y(sg(|f(x_1, \ldots, x_n, y) - g(x_1, \ldots, x_n, y)|) = 1)$

Definitions

A partial function f is called *partial recursive relative to a system of partial functions* Σ if f can be obtained from functions of Σ and the initial functions o,s,I_m^n by a finite number of operations of substitution, primitive recursion, and minimalization

f is called *partial recursive* if it can be obtained from o,s,I_m^n by a finite number of operations of substitution, primitive recursion, and minimalization

Using language of terms

A partial function f is called partial recursive relative to a system Σ of partial functions if f is the value of an operator term represented with the aid of operator symbols $\mathbf{S}^i$, $\mathbf{R}$, $\mathbf{M}$, object symbols o,s,I^n_m and functions from Σ

f is called *partial recursive* if it is the value of an operator term represented with operator symbols $\mathbf{S}^i$, $\mathbf{R}$, $\mathbf{M}$ and the object symbols from o,s,I^n_m.

Properties

1. If f is primitive recursive relative to Σ then f is also partial recursive relative to Σ; in particular, all primitive recursive functions are partial recursive.
2. Class of partial recursive functions is wider than the class of primitive recursive functions because primitive recursive function are everywhere defined while partial recursive function may not be everywhere defined, such as sg^{-1}, and s^{-1}.
3. Operators $\mathbf{S}^i$, $\mathbf{R}$, $\mathbf{M}$ applied on partial recursive functions produce partial recursive functions

Characteristic functions

For A set:

- the characteristic function of A is defined by

$$\chi_A(x) = \begin{cases} 0 & \text{if } x \in A \\ 1 & \text{if } x \notin A \end{cases}$$

  is called the *characteristic function of* A.

- partial characteristic function of A is defined by:

$$\chi_A^{pc}(x) = \begin{cases} 0 & \text{if } x \in A \\ undefined & \text{if } x \notin A \end{cases}$$

Examples

1. Characteristic function of $\emptyset$, $\chi_{\emptyset} = 0$ for any value of the argument.
2. Partial characteristic function of $\emptyset$ is nowhere defined
3. Characteristic and partial characteristic functions of N coincide
4. A set $A \subseteq N$ is primitive recursive if χ_A is primitive recursive
5. $A \subseteq N$ is partial recursive if χ_A is partial recursive

Properties

1. Every (relative) primitive recursive set is (relative) partial recursive
2. χ_A and χ_A^{pc} are related by $\chi_A^{pc}(x) = 0 - \chi_A(x)$ for $x \in N$.
3. Since subtraction is partial recursive χ_A^{pc} is partial recursive

Theorem

Let $f(x)$ be an arbitrary primitive recursive function and let A be a primitive recursive set of natural numbers. Then the partial functions $f_p(x)$ defined by the scheme:

$$f_p(x) = \begin{cases} f(x) & \text{if } x \in A \\ undefined & \text{if } x \notin A \end{cases}$$

is partial recursive

Proof idea: χ_A^{pc} is partial recursive. From the definition, it follows that $f_p(x) = f(x) + \chi_A^{pc}(x)$

Church's thesis

The class of algorithmically (or machine) computable partial numerical functions coincides with the class of partial recursive functions

Note Turing has interpreted the concept of algorithmic computability by partial recursive functions relative to a given system of computable functions Σ.

Turing thesis

The class of functions which are algorithmically computable relative to any class of functions Σ coincides with the class of partial functions which are partial recursive relative to Σ

Note: Church thesis follows from Turing thesis

General recursive functions

- For every partial recursive function f there exists a mechanical process $\mathcal{P}(f)$ by means of which an $x \in N$ is transformed into the value $f(x)$ of function f
- $\mathcal{P}(f)$ continues indefinitely without yielding result if $f(x)$ is not defined
- Hence, everywhere defined partial recursive functions f are functions for which $\mathcal{P}(f)$ terminate after a finite number of steps.

Weak minimalization, M^l

$$\mathrm{M}^l f = \begin{cases} \mathrm{M}f & \text{if } \mathrm{M}f \text{ is everywhere defined} \\ undefined & \text{otherwise} \end{cases}$$

Definition: functions obtained from o,s,I^n_m by a finite number of operations of substitution, primitive recursion, and weak minimalization are called *general recursive* functions

Observations

- Since operations $\mathbf{S}^i$, $\mathbf{R}$, $\mathbf{M}^l$, performed on everywhere defined functions either yield nothing or everywhere defined functions all general recursive functions are everywhere defined
- If the result of operation $\mathbf{M}^l$ is defined then it coincides with the result of operation $\mathbf{M}$. Therefore, general recursive functions are everywhere defined partial recursive functions
- Converse, every everywhere defined partial recursive function is general recursive (true but difficult to prove).
- Every primitive recursive function is a general recursive function. However, not every general recursive function is primitive recursive.