$\mu-$*recursive functions*

Philip Geißler

17th of November

Abstract I

Figure 1: Alice and Bob `https://goo.gl/gRrDGH`

Abstract II

A function $f(x_1, x_2, \cdots, x_n)$ is computable if there exists an algoritm $A(x_1, x_2, \cdots, x_n)$ following given rules, that solves f in its domain and gives no output otherwise.

$$(x_1, x_2, \cdots, x_n) \in D_f \Longrightarrow A(x_1, x_2, \cdots, x_n) = f(x_1, x_2, \cdots, x_n)$$
$$(x_1, x_2, \cdots, x_n) \notin D_f \Longrightarrow A(x_1, x_2, \cdots, x_n) = \textbf{ENDLESS LOOP}$$

Table of contents

Abstract: computability

μ - recursive functions

- Definition and Disambiguation
- Basic mathematical understanding
- Initial functions
- Composition function
- Primitive recursion function
- μ minimisation function

Appendix: computability

Abstract III

well known computability models:

- turing machines

Abstract III

well known computability models:

- turing machines
- GOTO-programs

Abstract III

well known computability models:

- turing machines
- GOTO-programs
- WHILE-programs $\supset$ LOOP-programs

Abstract III

well known computability models:

- turing machines
- GOTO-programs
- WHILE-programs $\supset$ LOOP-programs
- λ-calculus

Abstract III

well known computability models:

- turing machines
- GOTO-programs
- WHILE-programs $\supset$ LOOP-programs
- λ-calculus
- μ-recursive functions $\supset$ primitive recursive functions

μ - recursive functions

Definition

Definition:

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

Definition

Definition:

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

Definition:

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

Definition:

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

Definition

Definition:

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

Definition

Definition:

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

Definition:

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

Definition:

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

Disambiguation

- single and m-tupels of natural numbers
- total and partial functions
- initial functions
    - constant function
    - successor function
    - projection function
- composition
- primitive recursion
- μ operator

Single and m-tupels of natural numbers

$$n \in \mathbb{N} \qquad \mathbb{N} = \{0, 1, 2, 3, \cdots\}$$

Single and m-tupels of natural numbers

$$n \in \mathbb{N} \qquad \mathbb{N} = \{0, 1, 2, 3, \cdots\}$$

$$t \in \mathbb{N}^m \qquad t = (n_1, n_2, n_3, \cdots, n_m)$$

Single and m-tupels of natural numbers

$$n \in \mathbb{N} \qquad \mathbb{N} = \{0, 1, 2, 3, \cdots\}$$

$$t \in \mathbb{N}^m \qquad t = (n_1, n_2, n_3, \cdots, n_m)$$

tupels are not sets!

Total and partial functions

$$L1 := \forall f_1, f_2 \in f : f(x) = f_1 \land f(x) = f_2 \implies f_1 = f_2$$

$\rightarrow$ each x has at most one corresponding f(x)

$$L2 := \forall x \; \exists f_1 : f(x) = f_1$$

$\rightarrow$ each x has at least one corresponding f(x)

Total and partial functions

$$L1 := \forall f_1, f_2 \in f : f(x) = f_1 \land f(x) = f_2 \Longrightarrow f_1 = f_2$$

$\rightarrow$ each x has at most one corresponding f(x)

$$L2 := \forall x \; \exists f_1 : f(x) = f_1$$

$\rightarrow$ each x has at least one corresponding f(x)

$$\begin{aligned} f &\mathrel{\hat{=}} \text{total function} && \Longrightarrow L1 = \text{true} \land L2 = \text{true} \\ f &\mathrel{\hat{=}} \text{partial function} && \Longrightarrow L1 = \text{true} \end{aligned}$$

Constant function

A function $N(x_1, x_2, \cdots, x_n)$ is a constant function if:

$$N(x_1, x_2, \cdots, x_n) = c \qquad \begin{array}{r} n, c \in \mathbb{N} \\ x_1, x_2, \cdots, x_n \in \mathbb{N} \end{array}$$

Successor function

A function $S(x_1)$ is a successor function if:

$$S(x_1) = x_1 + 1 \qquad\qquad x_1 \in \mathbb{N}$$

Projection function

A function $P_m^n(x_1, x_2, \cdots, x_n)$ is a projection function if:

$$P_m^n(x_1, x_2, \cdots, x_m, \cdots, x_n) = x_m \qquad \begin{aligned} n, m &\in \mathbb{N} \\ x_1, x_2, \cdots, x_n &\in \mathbb{N} \\ 1 \leqslant m &\leqslant n \end{aligned}$$

Composition function

A function $C(x_1, x_2, \cdots, x_n)$ is a composition function if:

$$
\begin{aligned}
f_n &= f_n(x_1, x_2, \cdots, x_n) \\
f_{m,k} &= f_{m,k}(x_1, x_2, \cdots, x_m) \\
C &= f_n \circ (f_{m,1}, f_{m,2}, \cdots, f_{m,n}) \\
&= f_n(f_{m,1}(x_{1,1}, x_{1,2}, \cdots, x_{1,m}), \cdots, f_{m,n}(x_{n,1}, x_{n,2}, \cdots, x_{n,m}))
\end{aligned}
$$

$$n, m, x_{k,l} \in \mathbb{N}$$

Primitive recursion function

A function $R(X_0, x_1, x_2, \cdots, x_n)$ is primitive recursive if:

$$X_0 = 0 \implies R(X_0, x_1, x_2, \cdots, x_n) = f(y_1, \cdots, y_m)$$
$$X_0 \neq 0 \implies R(X_0, x_1, x_2, \cdots, x_n) = g(R(X_0 - 1, x_1, \cdots, x_n), z_1, \cdots, z_l)$$

$$x_k, y_k, z_k, n, m, l \in \mathbb{N}$$

Digression: primitive recursive functions

Example:

$$f(x, y, z) = (x \cdot y)^2 - z \qquad\qquad x, y, z \in \mathbb{N}$$

Digression: primitive recursive functions

$$f(x, y, z) = (x \cdot y)^2 - z \qquad\qquad x, y, z \in \mathbb{N}$$

Basic functions we need for f(x,y,z):

$$add(x, y) = x + y \ , \ \ mult(x, y) = x * y$$

$$add(x, y) = \begin{cases} add(0, y) & = y \\ add(x + 1, y) & = S(add(x, y)) \end{cases}$$

$$mult(x, y) = \begin{cases} mult(0, y) & = 0 \\ mult(x + 1, y) & = add(mult(x, y), y) \end{cases}$$

Digression: primitive recursive functions

$$f(x, y, z) = (x \cdot y)^2 - z \qquad\qquad x, y, z \in \mathbb{N}$$

Basic functions we need for f(x,y,z):

$$decr(x, y) = x - 1 \ , \ \ sub(x, y) = x - y$$

$$decr(x) = \begin{cases} decr(0) & = 0 \\ decr(x + 1) & = x \end{cases}$$

$$sub(x, y) = \begin{cases} sub(x, 0) & = x \\ sub(x, y + 1) & = decr(sub(x, y)) \end{cases}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$f(x, y, z) = (x \cdot y)^2 - z \qquad\qquad x, y, z \in \mathbb{N}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$\begin{aligned} f(x, y, z) &= (x \cdot y)^2 - z & \qquad x, y, z \in \mathbb{N} \\ &= mult(x, y)^2 - z \end{aligned}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$\begin{aligned} f(x, y, z) &= (x \cdot y)^2 - z & \qquad x, y, z \in \mathbb{N} \\ &= mult(x, y)^2 - z \\ &= mult(x, y) \cdot mult(x, y) - z \end{aligned}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$\begin{aligned} f(x, y, z) &= (x \cdot y)^2 - z & x, y, z \in \mathbb{N} \\ &= mult(x, y)^2 - z \\ &= mult(x, y) \cdot mult(x, y) - z \\ &= mult(mult(x, y), mult(x, y)) - z \end{aligned}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$\begin{aligned} f(x, y, z) &= (x \cdot y)^2 - z && x, y, z \in \mathbb{N} \\ &= mult(x, y)^2 - z \\ &= mult(x, y) \cdot mult(x, y) - z \\ &= mult(mult(x, y), mult(x, y)) - z \\ &= sub(mult(mult(x, y), mult(x, y)), z) \end{aligned}$$

Digression: primitive recursive functions

$$mult(x, y) = x * y \quad sub(x, y) = x + y \quad mult, sub \in Primitive\ RF$$

$$\begin{aligned}
f(x, y, z) &= (x \cdot y)^2 - z & x, y, z \in \mathbb{N} \\
&= mult(x, y)^2 - z \\
&= mult(x, y) \cdot mult(x, y) - z \\
&= mult(mult(x, y), mult(x, y)) - z \\
&= sub(mult(mult(x, y), mult(x, y)), z) \\
\\
&= sub(mult(mult(P_1^3(x, y, z), P_2^3(x, y, z)), \\
&\quad mult(P_1^3(x, y, z), P_2^3(x, y, z))), P_3^3(x, y, z))
\end{aligned}$$

μ minimisation function

A function $\mu(f(Y_0, y_1, y_2, \cdots, y_n), x_1, x_2, \cdots, x_n)$ is μ minimizing if:

$$\begin{aligned} & 0 = f(z, y_1, y_2, \cdots, y_n) & \\ \wedge\ & 0 < f(i, y_1, y_2, \cdots, y_n) & 0 \leqslant i < z \\ \iff & n = \mu(f(Y_0, y_1, y_2, \cdots, y_n), x_1, x_2, \cdots, x_n) & \end{aligned}$$

$$n, z, i_k, y_k \in \mathbb{N}$$

Pseudoxample μ recursive function

Example:

$$Z_f = undef, undef, undef, 0, 1, 2, 3, \cdots$$

$$\implies f(x) = \begin{cases} undef & |x \leqslant 2 \\ x - 3 & |x \geqslant 3 \end{cases}$$

Pseudoxample μ recursive function

$$f(x) = \begin{cases} undef & |x \leqslant 2 \\ n - 3 & |x \geqslant 3 \end{cases}$$

Basic functions we need for f(x,y,z):

$$decr(x) = \begin{cases} decr(0) & = 0 \\ decr(x+1) & = x \end{cases}$$

$$sub(x, y) = \begin{cases} sub(x, 0) & = x \\ sub(x, y+1) & = decr(sub(x, y)) \end{cases}$$

Pseudoxample μ recursive function

$$f(x) = \begin{cases} undef & | x \leqslant 2 \\ n - 3 & | x \geqslant 3 \end{cases}$$

Basic functions we need for f(x,y,z):

$$\begin{aligned} 3sub(z, x) &= sub(sub(z, 3), x) \\ &= z - 3 - x \\ \implies \mu(3sub(z, x)) &= \text{smallest natural x with given z,} \\ &\quad \text{for that } 3sub(z, x) = 0 \Rightarrow \\ z = 0 &\leftrightarrow undef, z = 1 \leftrightarrow undef, \\ z = 2 &\leftrightarrow undef, z = 3 \leftrightarrow 0, z = 4 \leftrightarrow 1, \cdots \\ \implies f(x) & \end{aligned}$$

Appendix: computability

Abstract IV

μ - recursive functions $\longleftarrow$ primitive recursive functions

Abstract IV

$$\begin{tikzcd}
\mu \text{ - recursive functions} \arrow[d, leftrightarrow] & \text{primitive recursive functions} \arrow[l] \arrow[d, leftrightarrow] \\
\text{WHILE-programs} & \text{LOOP-programs} \arrow[l]
\end{tikzcd}$$

Abstract IV

class relations between the different complete computability models:

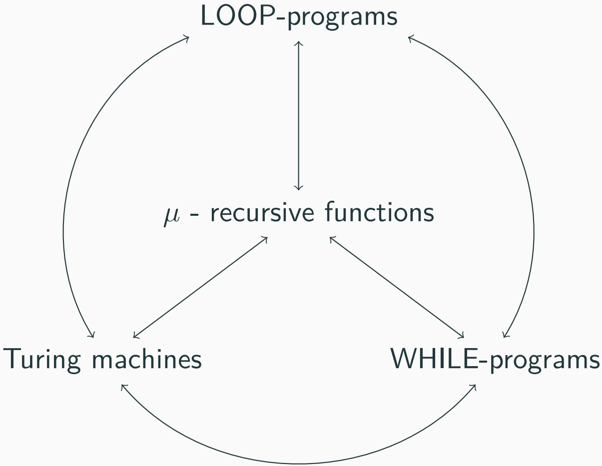
$$
\begin{tikzcd}
 & \mu\text{ - recursive functions} \arrow[d, leftrightarrow] & \text{primitive recursive functions} \arrow[l] \arrow[d, leftrightarrow] \\
 & \text{WHILE-programs} \arrow[dl, leftrightarrow, bend right] \arrow[d, leftrightarrow, bend left] & \text{LOOP-programs} \arrow[l] \\
\text{GOTO-programs} \arrow[r, leftrightarrow, bend right] & \text{Turing machines} &
\end{tikzcd}
$$

Abstract V

μ - recursive functions

Abstract V

class relations between the different complete computability models:



Sources

Access Date: December 8, 2017

- The picture in the introduction
  URL: `https://goo.gl/gRrDGH`
- Explanations of subtopics of μ-recursion
  URL: `https://goo.gl/yWvinS`
  URL: `https://goo.gl/xRPCiU`
  URL: `https://goo.gl/K4LKQB`
  URL: `https://goo.gl/XudPdL`
- U.Schöning, Theoretische Informatik kurz gefasst, 2008, Spektrum

$\mu -$ *recursive functions*

Philip Geißler

17th of November