Advanced Topics in Theoretical Computer Science

Part 3: Recursive functions

22.11.2012

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Until now

- Recapitulation: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Other computation models: e.g. Büchi Automata, λ -calculus

Until now

We showed that:

- LOOP \subseteq WHILE = GOTO \subseteq TM
- $\bullet \ \ \mathsf{WHILE} = \mathsf{GOTO} \subsetneq \mathsf{WHILE}^{\mathsf{part}} = \mathsf{GOTO}^{\mathsf{part}} \subseteq \mathsf{TM}^{\mathsf{part}}$
- LOOP \neq TM

Still to show:

- $\mathsf{TM} \subseteq \mathsf{WHILE}$
- $\mathsf{TM}^{\mathsf{part}} \subseteq \mathsf{WHILE}^{\mathsf{part}}$

For proving this, another model of computation will be used: recursive functions

Contents

- Recapitulation: Turing machines and Turing computability
- Register machines (LOOP, WHILE, GOTO)
- Recursive functions
- The Church-Turing Thesis
- Computability and (Un-)decidability
- Complexity
- Other computation models: e.g. Büchi Automata, λ -calculus

- Introduction/Motivation
- Primitive recursive functions
- $\mathcal{P} = \text{LOOP}$
- μ -recursive functions
- $F_{\mu} = WHILE$
- Summary

 $\mapsto \mathcal{P}$

Motivation

Functions as model of computation (without an underlying machine model)

Motivation

Functions as model of computation (without an underlying machine model)

Idea

- Simple ("atomic") functions are computable
- "Combinations" of computable functions are computable

(We consider functions $f : \mathbb{N}^k \to \mathbb{N}, \ k \ge 0$)

Motivation

Functions as model of computation (without an underlying machine model)

Idea

- Simple ("atomic") functions are computable
- "Combinations" of computable functions are computable

(We consider functions $f : \mathbb{N}^k \to \mathbb{N}, \ k \ge 0$)

Questions

- Which are the atomic functions?
- Which combinations are possible?

The following functions are primitive recursive and μ -recursive:

The following functions are primitive recursive and μ -recursive:

The constant null

 $0:\mathbb{N}^0 \to \mathbb{N}$ with 0() = 0

The following functions are primitive recursive and μ -recursive:

The constant null

$$0: \mathbb{N}^0 \to \mathbb{N}$$
 with $0() = 0$

Successor function

$$+1:\mathbb{N}^1 o\mathbb{N}$$
 with $+1(n)=n+1$ for all $n\in\mathbb{N}$

The following functions are primitive recursive and μ -recursive:

The constant null

$$0: \mathbb{N}^0 \to \mathbb{N}$$
 with $0() = 0$

Successor function

$$+1:\mathbb{N}^1 o\mathbb{N}$$
 with $+1(n)=n+1$ for all $n\in\mathbb{N}$

Projection function

$$\pi_i^k : \mathbb{N}^k \to \mathbb{N}$$
 with $\pi_i^k(n_1, \ldots, n_k) = n_i$

The following functions are primitive recursive and μ -recursive:

The constant null

$$0: \mathbb{N}^0 \to \mathbb{N}$$
 with $0() = 0$

Successor function

$$+1:\mathbb{N}^1 o\mathbb{N}$$
 with $+1(n)=n+1$ for all $n\in\mathbb{N}$

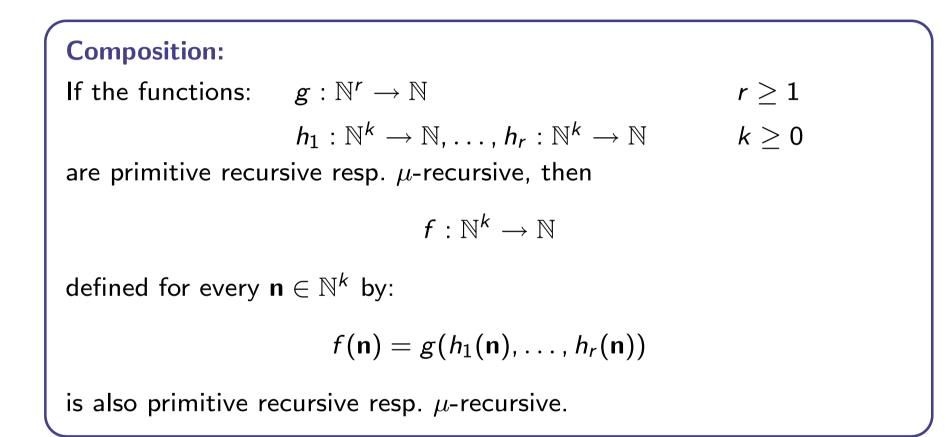
Projection function

$$\pi_i^k : \mathbb{N}^k \to \mathbb{N}$$
 with $\pi_i^k(n_1, \ldots, n_k) = n_i$

Notation:

We will write **n** for the tuple (n_1, \ldots, n_k) , $k \ge 0$.

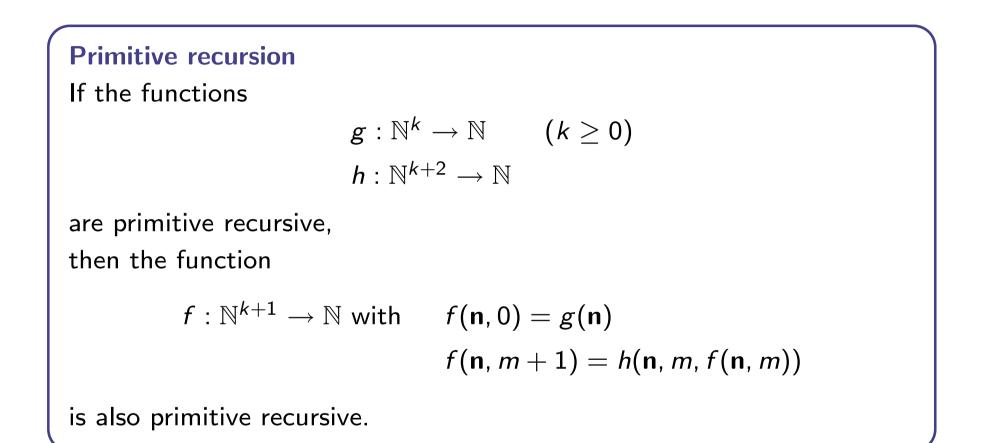
Recursive functions: Composition

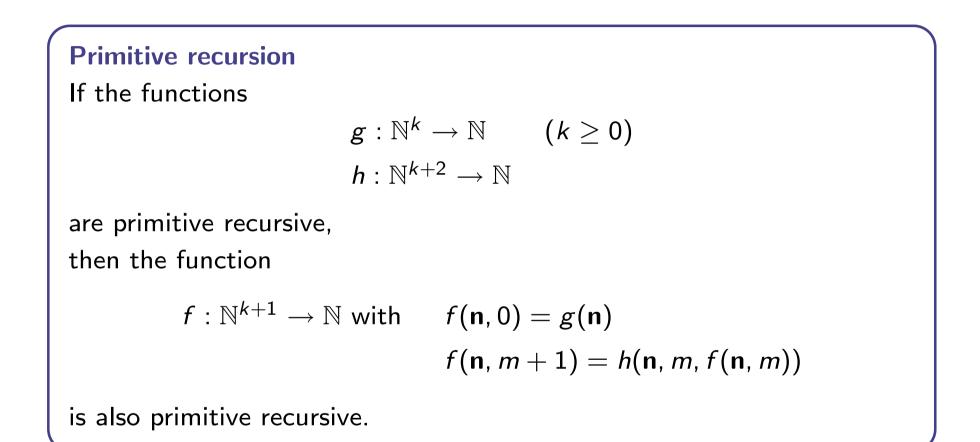


Notation without arguments: $f = g \circ (h_1, \ldots, h_r)$

- Introduction/Motivation
- Primitive recursive functions
- $\mathcal{P} = \text{LOOP}$
- μ -recursive functions
- $F_{\mu} = WHILE$
- Summary

 $\mapsto \mathcal{P}$





Notation without arguments: $f = \mathcal{PR}[g, h]$



- Atomic functions: The functions
 - Null O
 - Successor +1
 - Projection π_i^k $(1 \le i \le k)$
 - are primitive recursive.
- **Composition:** The functions obtained by composition from primitive recursive functions are primitive recursive.
- **Primitive recursion:** The functions obtained by primitive recursion from primitive recursive functions are primitive recursive.



- Atomic functions: The functions
 - Null 0
 - Successor +1
 - Projection π_i^k $(1 \le i \le k)$

are primitive recursive.

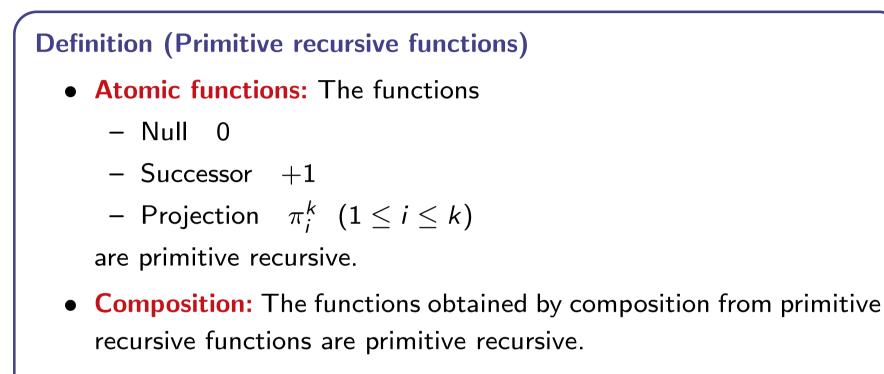
- **Composition:** The functions obtained by composition from primitive recursive functions are primitive recursive.
- **Primitive recursion:** The functions obtained by primitive recursion from primitive recursive functions are primitive recursive.



- Atomic functions: The functions
 - Null O
 - Successor +1
 - Projection π_i^k $(1 \le i \le k)$

are primitive recursive.

- **Composition:** The functions obtained by composition from primitive recursive functions are primitive recursive.
- **Primitive recursion:** The functions obtained by primitive recursion from primitive recursive functions are primitive recursive.



• **Primitive recursion:** The functions obtained by primitive recursion from primitive recursive functions are primitive recursive.

Notation: $\mathcal{P} =$ The set of all primitive recursive functions

$$f(n) = n + c$$

$$f(n) = n$$

$$f(n, m) = n + m$$

$$f(n, m) = n - 1$$

$$f(n, m) = n - m$$

$$f(n, m) = n * m$$

f(n) = n + c, for $c \in \mathbb{N}$, c > 0

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f(n) = \underbrace{(+1)(\dots((+1)(n)))}_{c \text{ times}}$

f(n) = n + c, for $c \in \mathbb{N}$, c > 0

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{i = 1}$$

c times

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f: \mathbb{N} \to \mathbb{N}$$
, with $f(n) = n$

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f = \pi_1^1$$

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f = \pi_1^1$$

f(n,m)=n+m

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f = \pi_1^1$$

f(n, m) = n + mf(n, 0) = nf(n, m + 1) = (+1)(f(n, m))

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f=\pi_1^1$$

 $\begin{aligned} f(n,m) &= n + m \\ f(n,0) &= n \\ f(n,m+1) &= (+1)(f(n,m)) \end{aligned} \begin{array}{l} g(n) &= n \\ h(n,m,k) &= +1(k) \end{array} \begin{array}{l} g &= \pi_1^1 \\ h &= (+1) \circ \pi_3^3 \end{aligned}$

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f = \pi_1^1$$

f(n,m)=n+m

f(n,0) = ng(n) = n $g = \pi_1^1$ f(n,m+1) = (+1)(f(n,m))h(n,m,k) = +1(k) $h = (+1) \circ \pi_3^3$

 $f=\mathcal{PR}[\pi_1^1$, $(+1)\circ\pi_3^3]$

f(n) = n + c, for $c \in \mathbb{N}, c > 0$ $f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$

Identity

$$f = \pi_1^1$$

f(n,m)=n+m

 $f=\mathcal{PR}[\pi_1^1$, $(+1)\circ\pi_3^3]$

f(n)=n-1

$$f(n) = n - 1$$
$$f(0) = 0$$
$$f(n+1) = n$$

$$f(n) = n - 1$$

$$f(0) = 0 g() = 0 g = 0$$

$$f(n+1) = n h(n,k) = n h = \pi_1^2$$

 $f = \mathcal{PR}[0, \pi_1^2]$

$$f(n) = n - 1$$

 $f = \mathcal{PR}[0, \pi_1^2]$

f(n,m)=n-m

$$egin{aligned} f(n) &= n-1 \ && f = \mathcal{PR}[0,\pi_1^2] \ f(n,m) &= n-m \ && f(n,0) &= n \ f(n,m+1) &= f(n,m)-1 \ && h(n,m,k) &= k-1 \ && h &= (-1) \circ \pi_3^3 \end{aligned}$$

 $f=\mathcal{PR}[\pi_1^1$, $(-1)\circ\pi_3^3]$

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, (-1) \circ \pi_3^3]$$

f(n,m) = n * m

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, (-1) \circ \pi_3^3]$$

$$f(n, m) = n * m$$

$$f(n, 0) = 0$$

$$g(n) = 0$$

$$g(n) = 0$$

$$g = 0$$

$$h(n, m, k) = k + n$$

$$h = + \circ (\pi_3^3, \pi_1^3)$$

$$f = \mathcal{PR}[0, + \circ (\pi_3^3, \pi_1^3)]$$

$$f(n) = n - 1$$

$$f = \mathcal{PR}[0, \pi_1^2]$$

$$f(n, m) = n - m$$

$$f = \mathcal{PR}[\pi_1^1, (-1) \circ \pi_3^3]$$

$$f(n, m) = n * m$$

$$f = \mathcal{PR}[0, + \circ (\pi_3^3, \pi_1^3)]$$

Re-ordering/Omitting/Repeating Arguments

Lemma The set of primitive recursive functions is closed under:

- Re-ordering
- Omitting
- Repeating

of arguments when composing functions.

Re-ordering/Omitting/Repeating Arguments

Lemma The set of primitive recursive functions is closed under:

- Re-ordering
- Omitting
- Repeating

of arguments when composing functions.

Proof: (Idea)

A tuple of arguments $\mathbf{n'} = (n_{i_1}, \ldots, n_{i_k})$ obtained from $\mathbf{n} = (n_1, \ldots, n_k)$ by re-ordering, omitting or repeating components can be represented as:

$$\mathbf{n'} = (\pi_{i_1}^k(\mathbf{n}), \dots, \pi_{i_k}^k(\mathbf{n}))$$

Lemma (Case distinction is primitive recursive)

- If g_i , h_i $(1 \le i \le r)$ are primitive recursive functions, and
 - for every *n* there exists a unique *i* with $h_i(n) = 0$

then the function f defined by:

$$f(n) = \begin{cases} g_1(n) & \text{if } h_1(n) = 0 \\ \dots & \\ g_r(n) & \text{if } h_r(n) = 0 \end{cases}$$

is primitive recursive.

Lemma (Case distinction is primitive recursive)

- If g_i , h_i $(1 \le i \le r)$ are primitive recursive functions, and
 - for every *n* there exists a unique *i* with $h_i(n) = 0$

then the function f defined by:

$$f(n) = \begin{cases} g_1(n) & \text{if } h_1(n) = 0 \\ \dots & \\ g_r(n) & \text{if } h_r(n) = 0 \end{cases}$$

is primitive recursive.

Proof: $f(n) = g_1(n) * (1 - h_1(n)) + \cdots + g_r(n) * (1 - h_r(n))$