

Advanced Topics in Theoretical Computer Science

Part 3: Recursive Functions (1)

30.11.2022

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

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

3. Recursive functions

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

Recursive functions

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 $f : \mathbb{N}^k \rightarrow \mathbb{N}$, $k \geq 0$)

Recursive functions

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 $f : \mathbb{N}^k \rightarrow \mathbb{N}$, $k \geq 0$)

Questions

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

Recursive functions: Atomic functions

The following functions are primitive recursive and μ -recursive:

Recursive functions: Atomic functions

The following functions are primitive recursive and μ -recursive:

The constant null

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

Recursive functions: Atomic functions

The following functions are primitive recursive and μ -recursive:

The constant null

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

Successor function

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

Recursive functions: Atomic functions

The following functions are primitive recursive and μ -recursive:

The constant null

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

Successor function

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

Projection function

$$\pi_i^k : \mathbb{N}^k \rightarrow \mathbb{N} \text{ with } \pi_i^k(n_1, \dots, n_k) = n_i$$

Recursive functions

Notation:

We will write \mathbf{n} for the tuple (n_1, \dots, n_k) , $k \geq 0$.

Recursive functions: Composition

Composition:

If the functions: $g : \mathbb{N}^r \rightarrow \mathbb{N}$ $r \geq 1$

$h_1 : \mathbb{N}^k \rightarrow \mathbb{N}, \dots, h_r : \mathbb{N}^k \rightarrow \mathbb{N}$ $k \geq 0$

are primitive recursive resp. μ -recursive, then

$$f : \mathbb{N}^k \rightarrow \mathbb{N}$$

defined for every $\mathbf{n} \in \mathbb{N}^k$ by:

$$f(\mathbf{n}) = g(h_1(\mathbf{n}), \dots, h_r(\mathbf{n}))$$

is also primitive recursive resp. μ -recursive.

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

Primitive recursive functions

Until now:

- **Atomic functions** (Null, Successor, Projections)
- **Composition**

Next:

- **Primitive recursion**

Definition of primitive recursive functions

Primitive recursive functions

Primitive recursion

If the functions

$$g : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

$$h : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$$

are primitive recursive,
then the function

$$f : \mathbb{N}^{k+1} \rightarrow \mathbb{N} \text{ with } f(\mathbf{n}, 0) = g(\mathbf{n})$$

$$f(\mathbf{n}, m + 1) = h(\mathbf{n}, m, f(\mathbf{n}, m))$$

is also primitive recursive.

Primitive recursive functions

Primitive recursion

If the functions

$$g : \mathbb{N}^k \rightarrow \mathbb{N} \quad (k \geq 0)$$

$$h : \mathbb{N}^{k+2} \rightarrow \mathbb{N}$$

are primitive recursive,
then the function

$$f : \mathbb{N}^{k+1} \rightarrow \mathbb{N} \text{ with } f(\mathbf{n}, 0) = g(\mathbf{n})$$

$$f(\mathbf{n}, m + 1) = h(\mathbf{n}, m, f(\mathbf{n}, m))$$

is also primitive recursive.

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

Primitive recursive functions

Definition (Primitive recursive functions)

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

Definition (Primitive recursive functions)

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

Definition (Primitive recursive functions)

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

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

Arithmetical functions: definitions

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

$$f(n) = n$$

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

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

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

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

Arithmetical functions: definitions

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

Arithmetical functions: definitions

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

$$f(n) = \underbrace{(+1)(\dots((+1)(n)))}_{c \text{ times}}$$

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \dots \circ (+1)}_{c \text{ times}}$$

Identity

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

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

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

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

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

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

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

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

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

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

$$g(n) = n$$

$$g = \pi_1^1$$

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

$$h(n, m, k) = +1(k)$$

$$h = (+1) \circ \pi_3^3$$

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

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

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

$$g(n) = n$$

$$g = \pi_1^1$$

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

$$h(n, m, k) = +1(k)$$

$$h = (+1) \circ \pi_3^3$$

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

Arithmetical functions: definitions

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

$$f = \underbrace{(+1) \circ \cdots \circ (+1)}_{c \text{ times}}$$

Identity

$$f = \pi_1^1$$

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

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

Arithmetical functions: definitions

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

Arithmetical functions: definitions

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

$$f(0) = 0$$

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

Arithmetical functions: definitions

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

$$f(0) = 0$$

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

$$g() = 0$$

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

$$g = 0$$

$$h = \pi_1^2$$

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

Arithmetical functions: definitions

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

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

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

Arithmetical functions: definitions

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

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

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

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

$$g(n) = n$$

$$g = \pi_1^1$$

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

$$h(n, m, k) = k - 1$$

$$h = (-1) \circ \pi_3^3$$

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

Arithmetical functions: definitions

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

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

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

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

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

Arithmetical functions: definitions

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

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

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

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

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

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

$$g(n) = 0$$

$$g = 0$$

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

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

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

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

Arithmetical functions: definitions

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

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

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

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

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

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

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

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

Additional Arguments

Lemma. Assume $f : \mathbb{N}^k \rightarrow \mathbb{N}$ is primitive recursive.

Then, for every $p \in \mathbb{N}$, the function $f' : \mathbb{N}^k \times \mathbb{N}^p \rightarrow \mathbb{N}$ defined for every $\mathbf{n} \in \mathbb{N}^k$ and every $\mathbf{m} \in \mathbb{N}^p$ by:

$$f'(\mathbf{n}, \mathbf{m}) = f(\mathbf{n})$$

is primitive recursive.

Additional Arguments

Lemma. Assume $f : \mathbb{N}^k \rightarrow \mathbb{N}$ is primitive recursive.

Then, for every $p \in \mathbb{N}$, the function $f' : \mathbb{N}^k \times \mathbb{N}^p \rightarrow \mathbb{N}$ defined for every $\mathbf{n} \in \mathbb{N}^k$ and every $\mathbf{m} \in \mathbb{N}^p$ by:

$$f'(\mathbf{n}, \mathbf{m}) = f(\mathbf{n})$$

is primitive recursive.

Proof:

Case 1: $k = 0$, i.e. f is a constant. Then $f^1 : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ with $f^1(\mathbf{n}, m) = f(\mathbf{n})$ for all $m \in \mathbb{N}$ can be expressed by primitive recursion as follows:

$$f^1(0) = f$$

$$f^1 = \mathcal{PR}[f, \pi_2^2]$$

$$f^1(n+1) = f^1(n) = \pi_2^2(n, f^1(n))$$

By iterating this construction p times we obtain extensions f^2, f^3, \dots, f^p with $2, 3, \dots, p$ additional arguments. The function f' is f^p .

Case 2: $k \neq 0$. Let $\mathbf{n} = (n_1, \dots, n_k, m_1, \dots, m_p)$

$$\text{Then } f'(\mathbf{n}) = f(\pi_1^{k+p}(\mathbf{n}), \dots, \pi_k^{k+p}(\mathbf{n})) = f \circ (\pi_1^{k+p}, \dots, \pi_k^{k+p}).$$

Case distinction

Lemma (Case distinction is primitive recursive)

If • g_i, h_i ($1 \leq i \leq r$) 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.

Case distinction

Lemma (Case distinction is primitive recursive)

If • g_i, h_i ($1 \leq i \leq r$) 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: $f(n) = g_1(n) * (1 - h_1(n)) + \dots + g_r(n) * (1 - h_r(n))$

Sums and products

Theorem

If $g : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ is a primitive recursive function then the following functions $f_1, f_2 : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ are also primitive recursive:

$$f_1(\mathbf{n}, m) = \begin{cases} 0 & \text{if } m = 0 \\ \sum_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$
$$f_2(\mathbf{n}, m) = \begin{cases} 1 & \text{if } m = 0 \\ \prod_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$

Sums and products

Theorem

If $g : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ is a primitive recursive function then the following functions $f_1, f_2 : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ are also primitive recursive:

$$f_1(\mathbf{n}, m) = \begin{cases} 0 & \text{if } m = 0 \\ \sum_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$
$$f_2(\mathbf{n}, m) = \begin{cases} 1 & \text{if } m = 0 \\ \prod_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$

Proof: f_1 and f_2 can be written using primitive recursion and case distinction:

$$f_1(\mathbf{n}, 0) = 0$$

$$f_1(\mathbf{n}, m + 1) = f_1(\mathbf{n}, m) + g(\mathbf{n}, m)$$

Sums and products

Theorem

If $g : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ is a primitive recursive function then the following functions $f_1, f_2 : \mathbb{N}^k \times \mathbb{N} \rightarrow \mathbb{N}$ are also primitive recursive:

$$f_1(\mathbf{n}, m) = \begin{cases} 0 & \text{if } m = 0 \\ \sum_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$
$$f_2(\mathbf{n}, m) = \begin{cases} 1 & \text{if } m = 0 \\ \prod_{i < m} g(\mathbf{n}, i) & \text{if } m > 0 \end{cases}$$

Proof: f_1 and f_2 can be written using primitive recursion and case distinction:

$$f_1(\mathbf{n}, 0) = 0$$

$$f_1(\mathbf{n}, m + 1) = f_1(\mathbf{n}, m) + g(\mathbf{n}, m)$$

$$f_2(\mathbf{n}, 0) = 1$$

$$f_2(\mathbf{n}, m + 1) = f_2(\mathbf{n}, m) * g(\mathbf{n}, m)$$