Theory of Computer Science

D4. Primitive Recursion and μ-Recursion

Malte Helmert

University of Basel

April 27, 2016

Overview: Computability Theory

Computability Theory

- imperative models of computation:
  - D1. Turing-Computability
  - D2. LOOP- and WHILE-Computability
  - D3. GOTO-Computability
- functional models of computation:
  - D4. Primitive Recursion and μ-Recursion
  - D5. Primitive/μ-Recursion vs. LOOP-/WHILE-Computability
- undecidable problems:
  - D6. Decidability and Semi-Decidability
  - D7. Halting Problem and Reductions
  - D8. Rice's Theorem and Other Undecidable Problems
    ~~Post's Correspondence Problem~~
    ~~Undecidable Grammar Problems~~
    ~~Gödel's Theorem and Diophantine Equations~~

Further Reading (German)

Literature for this Chapter (German)

Theoretische Informatik – kurz gefasst
by Uwe Schöning (5th edition)

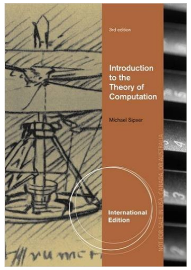
- Chapter 2.4

Further Reading (English)

Literature for this Chapter (English)

~~Introduction to the Theory of Computation~~
~~by Michael Sipser (3rd edition)~~

- This topic is not discussed by Sipser!



D4.1 Introduction

Formal Models of Computation: Primitive and μ-Recursion

Formal Models of Computation

- Turing machines
- LOOP, WHILE and GOTO programs
- primitive recursive and μ-recursive functions

In this chapter we get to know two models of computation with a very different flavor than Turing machines and imperative programming languages because they do not know “assignments” or “value changes”:

- primitive recursive functions
- μ-recursive functions

Primitive Recursion: Idea

Primitive recursion and μ-recursion are models of computation for functions over (one or more) natural numbers based on the following ideas:

- some simple basic functions are assumed to be computable (are computable “by definition”)
- from these functions new functions can be built according to certain “construction rules”

D4.2 Basic Functions and Composition

Basic Functions

Definition (Basic Functions)

The basic functions are the following functions in $\mathbb{N}_0^k \to \mathbb{N}_0$:

- constant zero function $null : \mathbb{N}_0 \to \mathbb{N}_0$:
  $null(x) = 0$ for all $x \in \mathbb{N}_0$
- successor function $succ : \mathbb{N}_0 \to \mathbb{N}_0$:
  $succ(x) = x + 1$ for all $x \in \mathbb{N}_0$
- projection functions $\pi_j^i : \mathbb{N}_0^i \to \mathbb{N}_0$ for all $1 \leq j \leq i$:
  $\pi_j^i(x_1, \ldots, x_i) = x_j$ for all $x_1, \ldots, x_i \in \mathbb{N}_0$
  (in particular this includes the identity function)

German: Basisfunktionen, konstante Nullfunktion, Nachfolgerfunktion, Projektionsfunktionen, Identitätsfunktion

Composition

Definition (Composition)

Let $k \geq 1$ and $i \geq 1$. The function $f : \mathbb{N}_0^k \to_p \mathbb{N}_0$ created by composition from the functions $h : \mathbb{N}_0^i \to_p \mathbb{N}_0$, $g_1, \ldots, g_i : \mathbb{N}_0^k \to_p \mathbb{N}_0$ is defined as:

$$f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_i(x_1, \ldots, x_k))$$

for all $x_1, \ldots, x_k \in \mathbb{N}_0$.

$f(x_1, \ldots, x_k)$ is undefined if any of the subexpressions is.

German: Einsetzungsschema, Einsetzung, Komposition

Composition: Examples

Reminder: $f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_i(x_1, \ldots, x_k))$

Example (Composition)

1. Consider $one : \mathbb{N}_0 \to \mathbb{N}_0$ with $one(x) = 1$ for all $x \in \mathbb{N}_0$.

one is created by composition from *succ* and *null*, since $one(x) = succ(null(x))$ for all $x \in \mathbb{N}_0$.

$\rightsquigarrow$ composition rule with $k = 1$, $i = 1$, $h = succ$, $g_1 = null$.

Composition: Examples

Reminder: $f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_i(x_1, \ldots, x_k))$

Example (Composition)

2. Consider $f_1 : \mathbb{N}_0^2 \to \mathbb{N}_0$ with $f_1(x, y) = y + 1$ for all $n \in \mathbb{N}_0$.

f_1 is created by composition from *succ* and π_2^2, since $f_1(x, y) = succ(\pi_2^2(x, y))$ for all $x, y \in \mathbb{N}_0$.

$\rightsquigarrow$ composition rule with $k = ?$, $i = ?$, $h = ?$, $g_{\ldots} = ?$

Composition: Examples

Reminder: $f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_i(x_1, \ldots, x_k))$

Example (Composition)

3. Let $r : \mathbb{N}_0^3 \to \mathbb{N}_0$.

Consider the function $f_2 : \mathbb{N}_0^4 \to \mathbb{N}_0$ with $f_2(a, b, c, d) = r(c, c, b)$.

f_2 is created by composition from r and the projection functions, since $f_2(a, b, c, d) = r(\pi_3^4(a, b, c, d), \pi_3^4(a, b, c, d), \pi_2^4(a, b, c, d))$.

$\rightsquigarrow$ composition rule with $k = ?$, $i = ?$, $h = ?$, $g_{\ldots} = ?$

$\rightsquigarrow$ Composition and projection in general allow us to reorder, ignore and repeat arguments.

Composition: Examples

Reminder: $f(x_1, \ldots, x_k) = h(g_1(x_1, \ldots, x_k), \ldots, g_i(x_1, \ldots, x_k))$

Example (Composition)

4. Let $add(x, y) := x + y$.

How can we use *add* and the basic functions with composition to obtain the function $double(x) : \mathbb{N}_0 \to \mathbb{N}_0$ with $double(x) = 2x$?

D4.3 Primitive Recursion

Primitive Recursion

Definition (Primitive Recursion)

Let $k \geq 1$. The function $f : \mathbb{N}_0^{k+1} \to_p \mathbb{N}_0$ created by primitive recursion from functions $g : \mathbb{N}_0^k \to_p \mathbb{N}_0$ and $h : \mathbb{N}_0^{k+2} \to_p \mathbb{N}_0$ is defined as:

$$f(0, x_1, \dots, x_k) = g(x_1, \dots, x_k)$$
$$f(n+1, x_1, \dots, x_k) = h(f(n, x_1, \dots, x_k), n, x_1, \dots, x_k)$$

for all $n, x_1, \dots, x_k \in \mathbb{N}_0$.

$f(n, x_1, \dots, x_k)$ is undefined if any of the subexpressions is.

German: primitives Rekursionsschema, primitive Rekursion

Example $k = 1$:

$$f(0, x) = g(x)$$
$$f(n+1, x) = h(f(n, x), n, x)$$

Primitive Recursion: Examples

Reminder (primitive recursion with $k = 1$):

$f(0, x) = g(x)$ $f(n+1, x) = h(f(n, x), n, x)$

Example (Primitive Recursion)

1. Let $g(a) = a$ and $h(a, b, c) = a + 1$.

Which function is created by primitive recursion from g and h?

$f(0, x) = g(x) = x$
$f(1, x) = h(f(0, x), 0, x) = h(x, 0, x) = x + 1$
$f(2, x) = h(f(1, x), 1, x) = h(x + 1, 1, x) = (x + 1) + 1 = x + 2$
$f(3, x) = h(f(2, x), 2, x) = h(x + 2, 2, x) = (x + 2) + 1 = x + 3$
...
$\rightsquigarrow f(a, b) = a + b$

Primitive Recursion: Examples

Reminder (primitive recursion with $k = 1$):

$f(0, x) = g(x)$ $f(n+1, x) = h(f(n, x), n, x)$

Example (Primitive Recursion)

2. Let $g(a) = 0$ and $h(a, b, c) = a + c$.

Which function is created by primitive recursion from g and h?

$\rightsquigarrow$ blackboard

Primitive Recursion: Examples

Reminder (primitive recursion with $k = 1$):

$f(0, x) = g(x)$ $f(n+1, x) = h(f(n, x), n, x)$

Example (Primitive Recursion)

3. Let $g(a) = 0$ and $h(a, b, c) = b$.

Which function is created by primitive recursion from g and h?

$f(0, x) = g(x) = 0$
$f(1, x) = h(f(0, x), 0, x) = 0$
$f(2, x) = h(f(1, x), 1, x) = 1$
$f(3, x) = h(f(2, x), 2, x) = 2$
...
$\rightsquigarrow f(a, b) = \max(a - 1, 0)$

$\rightsquigarrow$ with projection and composition: modified predecessor function

Primitive Recursive Functions

Definition (Primitive Recursive Function)

The set of primitive recursive functions (PRFs) is defined inductively by finite application of the following rules:

1. Every basic function is a PRF.
2. Functions that can be created by composition from PRFs are PRFs.
3. Functions that can be created by primitive recursion from PRFs are PRFs.

German: primitiv rekursive Funktion

Note: primitive recursive functions are always total. (Why?)

Primitive Recursive Functions: Examples

Example

The following functions are PRFs:

- $succ(x) = x + 1$ ($\rightsquigarrow$ basic function)
- $add(x, y) = x + y$ ($\rightsquigarrow$ shown)
- $mul(x, y) = x \cdot y$ ($\rightsquigarrow$ shown)
- $pred(x) = \max(x - 1, 0)$ ($\rightsquigarrow$ shown)
- $sub(x, y) = \max(x - y, 0)$ ($\rightsquigarrow$ exercises)
- $binom_2(x) = \binom{x}{2}$ ($\rightsquigarrow$ exercises)

Notation: in the following we write $x \ominus y$ for the modified subtraction $sub(x, y)$ (e. g., $pred(x) = x \ominus 1$).

And Now?

Does this have anything to do with the previous chapters?

$\rightsquigarrow$ Please be patient!

D4.4 μ-Recursion

μ-Operator

Definition (μ-Operator)

Let $k \geq 1$, and let $f : \mathbb{N}_0^{k+1} \rightarrow_p \mathbb{N}_0$.

The function $\mu f : \mathbb{N}_0^k \rightarrow_p \mathbb{N}_0$ is defined by

$$(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0 \text{ and } f(m, x_1, \dots, x_k) \text{ is defined for all } m < n\}$$

If the set to minimize is empty, then $(\mu f)(x_1, \dots, x_k)$ is undefined.

μ is called the μ-operator.

German: μ-Operator

μ-Operator: Examples

Reminder μf: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0 \text{ and } f(m, x_1, \dots, x_k) \text{ is defined for all } m < n\}$

if f total: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0\}$

Example (μ-Operator)

1. Let $f(a, b, c) = b \ominus (a \cdot c)$.

Which function is μf?

$$\begin{aligned}(\mu f)(x_1, x_2) &= \min\{n \in \mathbb{N}_0 \mid f(n, x_1, x_2) = 0\} \\ &= \min\{n \in \mathbb{N}_0 \mid x_1 \ominus (n \cdot x_2) = 0\} \\ &= \begin{cases} 0 & \text{if } x_1 = 0 \\ \text{undefined} & \text{if } x_1 \neq 0, x_2 = 0 \\ \lceil \frac{x_1}{x_2} \rceil & \text{otherwise} \end{cases}\end{aligned}$$

μ-Operator: Examples

Reminder μf: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0 \text{ and } f(m, x_1, \dots, x_k) \text{ is defined for all } m < n\}$

if f total: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0\}$

Example (μ-Operator)

2. Let $f(a, b) = b \ominus (a \cdot a)$.

Which function is μf?

$\rightsquigarrow$ blackboard

μ-Operator: Examples

Reminder μf: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0 \text{ and } f(m, x_1, \dots, x_k) \text{ is defined for all } m < n\}$

if f total: $(\mu f)(x_1, \dots, x_k) = \min\{n \in \mathbb{N}_0 \mid f(n, x_1, \dots, x_k) = 0\}$

Example (μ-Operator)

3. Let $f(a, b) = (b \ominus (a \cdot a)) + ((a \cdot a) \ominus b)$.

Which function is μf?

μ-Recursive Functions

Definition (μ-Recursive Function)

The set of μ-recursive functions (μRFs) is defined inductively by finite application of the following rules:

1. Every basic function is a μRF.
2. Functions that can be created by composition from μRFs are μRFs.
3. Functions that can be created by primitive recursion from μRFs are μRFs.
4. Functions that can be created by the μ-operator from μRFs are μRFs.

German: μ-rekursive Funktion

D4.5 Summary

Summary: Primitive Recursion and μ-Recursion

Idea: build complex functions from basic functions and construction rules.

- basic functions (B):
  - constant zero function
  - successor function
  - projection functions
- construction rules:
  - composition (C)
  - primitive recursion (P)
  - μ-operator (μ)
- primitive recursive functions (PRFs): built from (B) + (C) + (P)
- μ-recursive functions (μRFs): built from (B) + (C) + (P) + (μ)