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
D4. Primitive Recursion and $\mu$-Recursion

D4.1 Introduction

Primitive Recursion: Idea

Primitive recursion and $\mu$-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"

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
- $\mu$-recursive functions
D4.2 Basic Functions and Composition

Basic Functions

Definition (Basic Functions)
The basic functions are the following functions in $\mathbb{N}_0^k \rightarrow \mathbb{N}_0$:
- constant zero function $null : \mathbb{N}_0 \rightarrow \mathbb{N}_0$:
  $null(x) = 0$ for all $x \in \mathbb{N}_0$
- successor function $succ : \mathbb{N}_0 \rightarrow \mathbb{N}_0$:
  $succ(x) = x + 1$ for all $x \in \mathbb{N}_0$
- projection functions $\pi^j_i : \mathbb{N}_0^i \rightarrow \mathbb{N}_0$ 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)

Composition

Definition (Composition)
Let $k \geq 1$ and $i \geq 1$. The function $f : \mathbb{N}_0^k \rightarrow \mathbb{N}_0$ created by composition from the functions $h : \mathbb{N}_0^i \rightarrow \mathbb{N}_0$, $g_1, \ldots, g_i : \mathbb{N}_0^k \rightarrow \mathbb{N}_0$ 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: Basisfunktionen, konstante Nullfunktion, Nachfolgerfunktion, Projektionsfunktionen, Identitätsfunktion

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 $one : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ 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 $x \in \mathbb{N}_0$.
   ⇔ composition rule with $k = 1$, $i = 1$, $h = succ$, $g_1 = null$. 

German: Einsetzungsschema, Einsetzung, Komposition
D4. Primitive Recursion and µ-Recursion

Basic Functions and Composition

Composition: Examples

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

Example (Composition)

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

\( f_1 \) is created by composition from \( \text{succ} \) and \( \pi_2^2 \),

since \( f_1(x, y) = \text{succ}(\pi_2^2(x, y)) \) for all \( x, y \in \mathbb{N}_0 \).

\( \rightsquigarrow \) composition rule with \( k = ?, i = ?, h = ?, g_1 = ? \)

Example (Composition)

3. Let \( r : \mathbb{N}_0^3 \rightarrow \mathbb{N}_0 \). Consider the function \( f_2 : \mathbb{N}_0^4 \rightarrow \mathbb{N}_0 \) with \( f_2(a, b, c, d) = r(c, c, b) \).

\( f_2 \) is created by composition from \( r \) and the projection functions,

since \( f_2(a, b, c, d) = r(\pi_3^4(a, b, c, d), \pi_2^4(a, b, c, d), \pi_2^2(a, b, c, d)) \).

\( \rightsquigarrow \) composition rule with \( k = ?, i = ?, h = ?, g_1 = ? \)

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

Example (Composition)

4. Let \( \text{add}(x, y) := x + y \).

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

Definition (Primitive Recursion)
Let \( k \geq 1 \). The function \( f : \mathbb{N}_0^{k+1} \to \mathbb{N}_0 \)
created by primitive recursion from functions \( g : \mathbb{N}_0 \to \mathbb{N}_0 \)
and \( h : \mathbb{N}_0^{k+2} \to \mathbb{N}_0 \) is defined as:

\[
\begin{align*}
f(0, x_1, \ldots, x_k) &= g(x_1, \ldots, x_k) \\
f(n + 1, x_1, \ldots, x_k) &= h(f(n, x_1, \ldots, x_k), n, x_1, \ldots, x_k)
\end{align*}
\]

for all \( n, x_1, \ldots, x_k \in \mathbb{N}_0 \).
\( f(n, x_1, \ldots, x_k) \) is undefined if any of the subexpressions is.

German: primitives Rekursionsschema, primitive Rekursion

Example \( k = 1 \):

\[
\begin{align*}
f(0, x) &= g(x) \\
f(n + 1, x) &= h(f(n, x), n, x)
\end{align*}
\]

Reminder (primitive recursion with \( k = 1 \)):
\( f(0, x) = g(x) \quad f(n + 1, x) = h(f(n, x), n, x) \)

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 \( k = 1 \)):
\( f(0, x) = g(x) \quad f(n + 1, x) = h(f(n, x), n, x) \)

Example (Primitive Recursion)
3. Let \( g(a) = 0 \) and \( h(a, b, c) = b \).
Which function is created by primitive recursion from \( g \) and \( h \)?
\( \rightsquigarrow \) with projection and composition: modified predecessor function

\[
\begin{align*}
f(0, x) &= g(x) = 0 \\
f(1, x) &= h(f(0, x), 0, x) = 0 \\
f(2, x) &= h(f(1, x), 1, x) = 1 \\
f(3, x) &= h(f(2, x), 2, x) = 2 \\
&\quad \ldots \\
&\rightsquigarrow f(a, b) = \max(a - 1, 0)
\end{align*}
\]
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?)

Example
The following functions are PRFs:

- \( \text{succ}(x) = x + 1 \) (⇝ basic function)
- \( \text{add}(x, y) = x + y \) (⇝ shown)
- \( \text{mul}(x, y) = x \cdot y \) (⇝ shown)
- \( \text{pred}(x) = \max(x - 1, 0) \) (⇝ shown)
- \( \text{sub}(x, y) = \max(x - y, 0) \) (⇝ shown)
- \( \text{binom}_2(x) = \binom{x}{2} \) (⇝ exercises)

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

And Now?
Does this have anything to do with the previous chapters?
⇝ Please be patient!
D4. Primitive Recursion and µ-Recursion

µ-Operator

Definition (µ-Operator)

Let \( k \geq 1 \), and let \( f : \mathbb{N}_0^{k+1} \to \mathbb{N}_0 \).

The function \( \mu f : \mathbb{N}_0^k \to \mathbb{N}_0 \) is defined by

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

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

\( \mu \) is called the µ-operator.

German: µ-Operator

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

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

Example (µ-Operator)

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

Which function is \( \mu f \)?

\[
\mu f = 0 \quad \text{if } x_1 = 0
\]

\[
\mu f \text{ undefined } \quad \text{if } x_1 \neq 0, x_2 = 0
\]

\[
\mu f = \left\lfloor \frac{x_2}{x_1} \right\rfloor \quad \text{otherwise}
\]

M. Hilmert (Univ. Basel) Theorie April 27, 2016 25 / 31
**D4. Primitive Recursion and \(\mu\)-Recursion**

### \(\mu\)-Recursive Functions

**Definition (\(\mu\)-Recursive Function)**

The set of \(\mu\)-recursive functions (\(\mu\)RFs) is defined inductively by finite application of the following rules:

1. Every basic function is a \(\mu\)RF.
2. Functions that can be created by composition from \(\mu\)RFs are \(\mu\)RFs.
3. Functions that can be created by primitive recursion from \(\mu\)RFs are \(\mu\)RFs.
4. Functions that can be created by the \(\mu\)-operator from \(\mu\)RFs are \(\mu\)RFs.

**German:** \(\mu\)-rekursive Funktion

---

**Summary: Primitive Recursion and \(\mu\)-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)
  - \(\mu\)-operator (\(\mu\))

- **primitive recursive functions (PRFs):**
  - built from (B) + (C) + (P)

- **\(\mu\)-recursive functions (\(\mu\)RFs):**
  - built from (B) + (C) + (P) + (\(\mu\))