# Partial recursive functions

Peter Mayr

Computability Theory, September 25, 2023

# Primitive recursive = inductively defined

For proving his incompleteness theorems, Gödel introduced a computational model based on functions on $\mathbb{N} = \{0, 1, 2, \dots\}$.

## Example

addition: $x + 0 := x$ projection

$x + (y + 1) := (x + y) + 1$ successor of $x+y$

factorial: $0! := 1$

$(n + 1)! := (n!) \cdot (n + 1)$

## Definition

The class of **primitive recursive** functions is the least class of finitary total functions on $\mathbb{N}$ that contains

1. constant 0
2. successor $s\colon \mathbb{N} \to \mathbb{N},\ x \mapsto x+1$
3. projections $p_i^n\colon \mathbb{N}^n \to \mathbb{N},\ \underbrace{(x_1, \dots, x_n)}_{\bar{x}} \mapsto x_i$
4. compositions $h\colon \mathbb{N}^n \to \mathbb{N}, \bar{x} \mapsto f(g_1(\bar{x}), \dots, g_k(\bar{x}))$
   for $f\colon \mathbb{N}^k \to \mathbb{N}, g_1, \dots, g_k\colon \mathbb{N}^n \to \mathbb{N}$ primitive recursive
5. every function $h\colon \mathbb{N}^{n+1} \to \mathbb{N}$ defined by a primitive recursive scheme

$$h(\bar{x}, 0) := f(\bar{x})$$
$$h(\bar{x}, y+1) := g(\bar{x}, y, h(\bar{x}, y))$$

   for $f, g$ primitive recursive.

Most of the usual functions in number theory $xy, x^y, \ldots$ are primitive recursive.

## Example

Monus $x \dot{-} y := \begin{cases} x - y & \text{if } x \geq y \\ 0 & \text{else} \end{cases}$ is primitive recursive.

$\prod_{y \leq z} f(\bar{x}, y)$ is primitive recursive if $f$ is. (HW)

# Predicates

## Definition

A predicate (relation) on $\mathbb{N}$ is **primitive recursive** if its characteristic function is.

## Example

$x \leq y$

$\leq \; = \{ (x,y) \in \mathbb{N}^2 \mid x \leq y \}$

$c_{\leq}(x,y) = \begin{cases} 1 & \text{if } x \dot{-} y = 0 \\ 0 & \text{else} \end{cases}$ is primitive recursive (HW)

"$x$ is prime"

## Lemma

Primitive recursive predicates are closed under $\wedge, \vee, \neg$ and bounded quantifiers; also $=$ is primitive recursive.

## Proof.

Let $P, Q$ be primitive recursive.

$P(x) \wedge Q(x)$ has char function $c_P(x) \cdot c_Q(x)$

$\neg P$

$R(y) := \forall x \le y \neg P(x)$

$x = 0$

$x = y \Leftrightarrow x \dot{-} y = 0 \wedge y \dot{-} x = 0$

$1 \dot{-} c_P(x)$

$\prod_{x \le y} c_P(x)$

$c_0(0) := 1 \ , \ c_0(k+1) := 0$

## Lemma (Definition by cases)

Let $f_1, f_2 \colon \mathbb{N}^k \to \mathbb{N}$ and a $k$-ary predicate $P$ on $\mathbb{N}$ be primitive recursive. Then

$$f(x) := \begin{cases} f_1(x) & \text{if } P(x) \\ f_2(x) & \text{else} \end{cases}$$

is primitive recursive.

Proof. $f(x) = f_1(x) \cdot c_P(x) + f_2(x) \cdot (1 \dot{-} c_P(x))$

$\square$

# The bounded search operator (minimization) $\mu$

## Definition

For a $(k+1)$-ary predicate $P$ let

$$\mu(t < y)\, P(x,t) := \begin{cases} \text{the least } t < y \text{ such that } P(x,t) \text{ if it exists,} \\ y \text{ else.} \end{cases}$$

function in $x, y$.

## Example

$\mu(t < 10)$ [$t$ is prime and $t > 7$] $=$ 10

20 11

## Lemma

Let $P$ be a primitive recursive predicate. Then

$$f(x,y) := \mu(t < y) P(x,t)$$

is primitive recursive.

## Proof.

HW

## Example

$n \mapsto p_n$ ($n$-th prime number) is primitive recursive.

$p_0 := 2$

$p_{n+1} :=$ smallest prime $> p_n$

$= \mu(t < p_n! + 1)\ [\,t \text{ is prime and } t > p_n\,]$