The Recursive Functions

early 20th century
mankind trying to grasp what is computability

1 What functions are computable in the intuitive sense?

attempts at definitions

- primitive recursive functions, didn't work; extend to μ-recursive functions
- Turing machines (*extremely* simple) you have not seen them yet
- MIPS with infinite registers and memory (from I2EA)
- C with infinite data types and memory (from I2EA)
- more .....

1 What functions are computable in the intuitive sense?

attempts at definitions

- primitive recursive functions, didn't work; extend to μ-recursive functions
- Turing machines (*extremely* simple) you have not seen them yet
- MIPS with infinite registers and memory (from I2EA)
- C with infinite data types and memory (from I2EA)
- more .....

all were shown equivalent

- the interesting case shown by Church: Turing machines are simulated by μ-recursive functions
- *Church's thesis*: the above definition(s) capture the intuitive concept of computability
- all sufficiently powerful programming languages simulate each other
- if you know one (sufficiently powerful) programming language: you know them all. Just find the constructs you are used to in the new language.

or implement them

2 prelude: countability

def: countable set A set A is countable if A is finite or if there is a bijection

$$b : \mathbb{N}_0 \to A$$

Note that the infinite sequence

$$(b(n)) = (b(0), b(1), b(2), \ldots)$$

enumerates A.

example:

$$A = \{n \in \mathbb{N}_0 \ : \ n \text{ even}\} \quad , \quad b(n) = 2n$$

2 prelude: countability

def: countable set A set A is countable if A is finite or if there is a bijection

$$b : \mathbb{N}_0 \to A$$

Note that the infinite sequence

$$(b(n)) = (b(0), b(1), b(2), \dots)$$

enumerates A.

example:

$$A = \{n \in \mathbb{N}_0 \ : \ n \text{ even}\} \quad , \quad b(n) = 2n$$

Lemma 1. *If A is finite, then $A*$ is countable*

enumerate

- $A^0, A^1, A^2, \dots$
- each A^n in lexicographic order

$$\mathbb{B}^* = (\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$$

2 prelude: countability

def: countable set A set A is countable if A is finite or if there is a bijection

$$b : \mathbb{N}_0 \rightarrow A$$

Note that the infinite sequence

$$(b(n)) = (b(0), b(1), b(2), \ldots)$$

enumerates A.

example:

$$A = \{n \in \mathbb{N}_0 \ : \ n \text{ even}\} \quad , \quad b(n) = 2n$$

Lemma 1. *If A is finite, then $A*$ is countable*

enumerate

- $A^0, A^1, A^2, \ldots$
- each A^n in lexicographic order

$$\mathbb{B}^* = (\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$$

Lemma 2. $\mathbb{N}_0 \times \mathbb{N}_0$ *is countable*

enumerate the sets

- $S_i = \{(a,b) \ : \ a+b\}$ for $i = 0, 1, 2, \ldots\}$
- each S_i in lexicographic order

$$\mathbb{N}_0 \times \mathbb{N}_0 = \{(0,0),\ (0,1),(1,0),\ (0,2),(1,1),(2,0),\ \ldots\}$$

2 prelude: countability

def: countable set A set A is countable if A is finite or if there is a bijection

$$b : \mathbb{N}_0 \to A$$

Note that the infinite sequence

$$(b(n)) = (b(0), b(1), b(2), \ldots)$$

enumerates A.

example:

$$A = \{n \in \mathbb{N}_0 \ : \ n \text{ even}\} \quad , \quad b(n) = 2n$$

Lemma 1. *If A is finite, then $A*$ is countable*

enumerate

- $A^0, A^1, A^2, \ldots$
- each A^n in lexicographic order

$$\mathbb{B}^* = (\{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$$

Lemma 2. *$\mathbb{N}_0 \times \mathbb{N}_0$ is countable*

enumerate the sets

- $S_i = \{(a,b) \ : \ a+b\}$ for $i = 0, 1, 2, \ldots\}$
- each S_i in lexicographic order

$$\mathbb{N}_0 \times \mathbb{N}_0 = \{(0,0),\ (0,1),(1,0),\ (0,2),(1,1),(2,0),\ \ldots\}$$

notation: for pairs $p = (x,y)$ we denote - as for vectors - its components as

$$p_1 = x\ ,\ p_2 = y$$ maybe helpful for exercises

Lemma 3. *If A and B are countable, then $A \times B$ is countable*

Proof. exercise □

Lemma 4. *For all k holds: N^k is countable.*

proof: exercise

3 a first glance (in this lecture) at diagonalisation

Lemma 5. *Let*

$$F = \{f \mid f : \mathbb{N}_0 \to \mathbb{B}\}$$

then F *is not countable.*

3 a first glance (in this lecture) at diagonalisation

Lemma 5. *Let*

$$F = \{f \mid f : \mathbb{N}_0 \to \mathbb{B}\}$$

then F is not countable.

- assume F can be enumerated as

$$F = \{f_0, f_1, f_2, \ldots\}$$

- consider the infinite matrix of table 1

$f_0(0)$	$f_0(1)$	$f_0(2)$	$\ldots$	$f_0(x)$	$\ldots$
$f_1(0)$	$f_1(1)$	$f_1(2)$	$\ldots$	$f_1(x)$	$\ldots$
$f_2(0)$	$f_2(1)$	$f_2(2)$	$\ldots$	$f_2(x)$	$\ldots$
			$\ldots$		
$f_x(0)$	$f_x(1)$	$f_x(2)$	$\ldots$	$f_x(x)$	$\ldots$
			$\ldots$		

Table 1: row x of this matrix is the function table of function f_x. Function g is defined such that it differs from f_x at argument x, i.e. on the diagonal of the matrix.

3 a first glance (in this lecture) at diagonalisation

Lemma 5. *Let*

$$F = \{f \mid f : \mathbb{N}_0 \to \mathbb{B}\}$$

then F is not countable.

- assume F can be enumerated as

$$F = \{f_0, f_1, f_2, \ldots\}$$

- consider the infinite matrix of table 1

$\boxed{f_0(0)}$	$f_0(1)$	$f_0(2)$	$\ldots$	$f_0(x)$	$\ldots$
$f_1(0)$	$\boxed{f_1(1)}$	$f_1(2)$	$\ldots$	$f_1(x)$	$\ldots$
$f_2(0)$	$f_2(1)$	$\boxed{f_2(2)}$	$\ldots$	$f_2(x)$	$\ldots$
			$\ldots$		
$f_x(0)$	$f_x(1)$	$f_x(2)$	$\ldots$	$\boxed{f_x(x)}$	$\ldots$
			$\ldots$		

Table 1: row x of this matrix is the function table of function f_x. Function g is defined such that it differs from f_x at argument x, i.e. on the diagonal of the matrix.

- define $g : \mathbb{N}_0 \to \mathbb{B}$ such that it differs from f_x at argument x

$$\forall x. \quad g(x) = f_x(x) \oplus 1$$

- then $g \notin \{f_0, f_1, f_2, \ldots\}$: otherwise $g = f_x$ for some x and we get contradiction

$$f_x(x) = g(x) = f_x(x) \oplus 1$$

4 definition of primitive recursive functions

you have seen it where?

Inductive definition of a set *PR* of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

4 definition of primitive recursive functions

you have seen it where?

Inductive definition of a set PR of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

base cases:

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$c_s^r(x) = s \;,\; s \in \mathbb{N}_0$$

2. projections $p_i^r \in PR$ where

$$p_i^r(x) : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$p_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \to \mathbb{N}_0$$

$$S(x) = x + 1$$

where have you seen it first?

4 definition of primitive recursive functions

you have seen it where?

Inductive definition of a set PR of computable functions:

$$f : \mathbb{N}_0^r \rightarrow \mathbb{N}_0$$

base cases:

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \rightarrow \mathbb{N}_0$$

$$c_s^r(x) = s \ , \ s \in \mathbb{N}_0$$

2. projections $p_i^r \in PR$ where

$$p_i^r(x) : \mathbb{N}_0^r \rightarrow \mathbb{N}_0$$

$$p_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \rightarrow \mathbb{N}_0$$

$$S(x) = x + 1$$

where have you seen it first?

induction steps of definition:

4. function composition. If the following function are all in PR

$$f : \mathbb{N}_0^r \rightarrow \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \rightarrow \mathbb{N}_0$$

then also $h \in PR$ where

$$h : \mathbb{N}_0^m \rightarrow \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in PR

$$g : \mathbb{N}_0^r \rightarrow \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \rightarrow \mathbb{N}_0$$

then also $f \in PR$ where

$$f : \mathbb{N}_0^{r+1} \rightarrow \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

6. these are all

excluding everything else

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$c_s^r(x) = s \ , \ s \in \mathbb{N}_0$$

2. projections $p_i^r \in PR$ where

$$p_i^r(x) : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$p_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \to \mathbb{N}_0$$

4. function composition. If the following function are all in PR

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

   then also $h \in PR$ where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in PR

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

   then also $f \in PR$ where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

examples you know as KIU CS students :)

- addition

$$\begin{aligned} f(0,x) &= x = p_1^1(x) \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 = c_0^1(x) \\ g(n+1,x) &= f(x, g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n, h(n,x)) \end{aligned}$$

recursion theory was for you like mother's milk

1. constant functions $c_s^r \in PR$ where

$$c_s^r : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$c_s^r(x) = s \ , \ s \in \mathbb{N}_0$$

2. projections $p_i^r \in PR$ where

$$p_i^r(x) : \mathbb{N}_0^r \to \mathbb{N}_0$$

$$p_i^r(x) = x_i$$

3. successor function $S \in PR$

$$s : \mathbb{N}_0 \to \mathbb{N}_0$$

4. function composition. If the following function are all in PR

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

then also $h \in PR$ where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$

$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in PR

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

then also $f \in PR$ where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

examples you know

- addition

$$\begin{aligned} f(0,x) &= x = p_1^1(x) \\ f(n+1,x) &= S(f(n,x)) \end{aligned}$$

- multiplication

$$\begin{aligned} g(0,x) &= 0 = c_0^1(x) \\ g(n+1,x) &= f(x, g(n,x)) \end{aligned}$$

- exponentiation

$$\begin{aligned} h(0,x) &= 1 \\ h(n+1,x) &= g(n, h(n,x)) \end{aligned}$$

explanation for math students, why…

- in computer architecture you should be able to explain/prove that your adders work
- so why does 1 + 1 = 10 make sense?
- decimal counter part: 9 + 1 = 10 theorem!
- using
  - Z = number of fingers/toes
  - 9+1 = Z definition
  - $10 = 1 \cdot Z^1 + 0 \cdot Z^0$
- thus need to define exponentiation without decimal numbers

5 the pr functions cannot be all computable functions

a modern view

example for defining +

$$
\begin{aligned}
f_1(x) &= p_1^1(x) \\
f_2(x) &= S(x) \\
f_3(0,x) &= f_1(x) \\
f_3(n+1,y) &= f_2(f_3(n,x)
\end{aligned}
$$

5 the pr functions cannot be all computable functions

a modern view

example for defining +

$$\begin{aligned} f_1(x) &= p_1^1(x) \\ f_2(x) &= S(x) \\ f_3(0,x) &= f_1(x) \\ f_3(n+1,y) &= f_2(f_3(n,x)) \end{aligned}$$

def: derivation A *derivation* of a pr function f is a finite sequence of definitions of functions

$$(f_1, f_2, \dots, f_s)$$

such that for each i

- either f_i is pr by rules 1 to 3
- or f_i is defined by rule 4 or 5 using only previously listed functions f_j with $j < i$.
- $f = f_s$ is the last function defined in the sequence

5 the pr functions cannot be all computable functions

a modern view

example for defining +

$$\begin{aligned} f_1(x) &= p_1^1(x) \\ f_2(x) &= S(x) \\ f_3(0,x) &= f_1(x) \\ f_3(n+1,y) &= f_2(f_3(n,x)) \end{aligned}$$

def: derivation A *derivation* of a pr function f is a finite sequence of definitions of functions

$$(f_1, f_2, \ldots, f_s)$$

such that for each i

- either f_i is pr by rules 1 to 3
- or f_i is defined by rule 4 or 5 using only previously listed functions f_j with $j < i$.
- $f = f_s$ is the last function defined in the sequence

observe:

- derivations d are formed by symbols of a finite alphabet A, thus $d \in A^*$
- one can write a parser P, which decides if an input d is a derivation of a function $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$
- enumerate all elements $w \in A^*$ and test each w by the parser P.
- for all x define d_x as the derivation of the x'th string which passes the test and define

  $$f_x : \mathbb{N}_0 \rightarrow \mathbb{N}_0$$

  as the function defined by derivation d_x.
- this enumerates the pr functions $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$

$$\{f_0, f_1, \ldots\} = \{f \mid f : \mathbb{N}_0 \rightarrow \mathbb{N}_0 \ , \ f \text{ is pr}\}$$

5 the pr functions cannot be all computable functions

a modern view

example for defining +

$$\begin{aligned} f_1(x) &= p_1^1(x) \\ f_2(x) &= S(x) \\ f_3(0,x) &= f_1(x) \\ f_3(n+1,y) &= f_2(f_3(n,x)) \end{aligned}$$

def: derivation A *derivation* of a pr function f is a finite sequence of definitions of functions

$$(f_1, f_2, \ldots, f_s)$$

such that for each i

- either f_i is pr by rules 1 to 3
- or f_i is defined by rule 4 or 5 using only previously listed functions f_j with $j < i$.
- $f = f_s$ is the last function defined in the sequence

observe:

- derivations d are formed by symbols of a finite alphabet A, thus $d \in A^*$
- one can write a parser P, which decides if an input d is a derivation of a function $f : \mathbb{N}_0 \to \mathbb{N}_0$
- enumerate all elements $w \in A^*$ and test each w by the parser P.
- for all x define d_x as the derivation of the x'th string which passes the test and define

  $$f_x : \mathbb{N}_0 \to \mathbb{N}_0$$

  as the function defined by derivation d_x.
- this enumerates the pr functions $f : \mathbb{N}_0 \to \mathbb{N}_0$

$$\{f_0, f_1, \ldots\} = \{f \mid f : \mathbb{N}_0 \to \mathbb{N}_0 \text{ , } f \text{ is pr}\}$$

it's obviously over:

5 the pr functions cannot be all computable functions

observe:

- derivations d are formed by symbols of a finite alphabet A, thus $d \in A^*$
- one can write a parser P, which decides if an input d is a derivation of a function $f : \mathbb{N}_0 \to \mathbb{N}_0$
- enumerate all elements $w \in A^*$ and test each w by the parser P.
- for all x define d_x as the derivation of the x'th string which passes the test and define
$$f_x : \mathbb{N}_0 \to \mathbb{N}_0$$
as the function defined by derivation d_x.
- this enumerates the pr functions $f : \mathbb{N}_0 \to \mathbb{N}_0$
$$\{f_0, f_1, \dots\} = \{f \mid f : \mathbb{N}_0 \to \mathbb{N}_0 \ , \ f \text{ is pr}\}$$

it's obviously over:

- diagonalisation: the function
$$g : \mathbb{N}_0 \to \mathbb{N}_0$$
defined by
$$\forall x.\ g(x) = f_x(x) + 1$$
is not pr. diagonalisation as before
- we can write an interpreter I which given a derivation d_x and an input x evaluates $f_x(x)$
- now compute $g(x)$ as i) enumerate words $w \in A^*$ and test each w by parser P until d_x is found. ii) using the interpreter compute $f_x(x)$ iii) then add 1

5 the pr functions cannot be all computable functions

observe:

- derivations d are formed by symbols of a finite alphabet A, thus $d \in A^*$
- one can write a parser P, which decides if an input d is a derivation of a function $f : \mathbb{N}_0 \to \mathbb{N}_0$
- enumerate all elements $w \in A^*$ and test each w by the parser P.
- for all x define d_x as the derivation of the x'th string which passes the test and define
$$f_x : \mathbb{N}_0 \to \mathbb{N}_0$$
as the function defined by derivation d_x.
- this enumerates the pr functions $f : \mathbb{N}_0 \to \mathbb{N}_0$
$$\{f_0, f_1, \ldots\} = \{f \mid f : \mathbb{N}_0 \to \mathbb{N}_0 \ , \ f \text{ is pr}\}$$

it's obviously over:

- diagonalisation: the function
$$g : \mathbb{N}_0 \to \mathbb{N}_0$$
defined by
$$\forall x.\ g(x) = f_x(x) + 1$$
is not pr. diagonalisation as before
- we can write an interpreter I which given a derivation d_x and an input x evaluates $f_x(x)$
- now compute $g(x)$ as i) enumerate words $w \in A^*$ and test each w by parser P until d_x is found. ii) using the interpreter compute $f_x(x)$ iii) then add 1

so it's computable

Lemma 6. *There exists a total computable function, which is not promitive recursive*

6 more pr functions

- modified predecessor

$$p(x) = \begin{cases} x-1 & x>0 \\ 0 & x=0 \end{cases}$$

$$\begin{aligned} u(0) &= 0 \\ u(x+1) &= x \end{aligned}$$

6 more pr functions

- modified predecessor

$$p(x) = \begin{cases} x-1 & x>0 \\ 0 & x=0 \end{cases}$$

$$\begin{aligned} u(0) &= 0 \\ u(x+1) &= x \end{aligned}$$

- modified difference

$$x \dot{-} y = \begin{cases} x-y & x \geq y \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned} x \dot{-} 0 &= x \\ x \dot{-} (y+1) &= u(x \dot{-} y) \end{aligned}$$

- sign

$$sg(x) = \begin{cases} 1 & x>0 \\ 0 & x=0 \end{cases}$$

$$\begin{aligned} sg(0) &= 0 \\ sg(x+1) &= 1 \end{aligned}$$

alternatively

$$\begin{aligned} \overline{sg} &= 1 \dot{-} x \\ sg(x) &= \overline{sg}(\overline{sg}(x)) \end{aligned}$$

7 pr predicates

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \rightarrow \{true, false\}$$

examples: with $r = 2$

$$x \leq y \;,\; x|y \;,\; 2x < y$$

7 pr predicates

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \rightarrow \{true, false\}$$

examples: with $r = 2$

$$x \leq y \ , \ x|y \ , \ 2x < y$$

def: characteristic function of predicate P:

$$C_P : \mathbb{N}_0^r \rightarrow \mathbb{B}$$

$$C_P(x) = \begin{cases} 1 & P(x) = true \\ 0 & P(x) = false \end{cases}$$

def: pr predicates Predicate P is pr iff C_P is pr.

7 pr predicates

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \to \{true, false\}$$

examples: with $r = 2$

$$x \leq y \ , \ x|y \ , \ 2x < y$$

def: characteristic function of predicate P:

$$C_P : \mathbb{N}_0^r \to \mathbb{B}$$

$$C_P(x) = \begin{cases} 1 & P(x) = true \\ 0 & P(x) = false \end{cases}$$

def: pr predicates Predicate P is pr iff C_P is pr.

examples

- $x < y$

$$C_{x<y} = sg(x \dot{-} y)$$

7 pr predicates

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \to \{true, false\}$$

examples: with $r = 2$

$$x \leq y \ , \ x|y \ , \ 2x < y$$

def: characteristic function of predicate P:

$$C_P : \mathbb{N}_0^r \to \mathbb{B}$$

$$C_P(x) = \begin{cases} 1 & P(x) = true \\ 0 & P(x) = false \end{cases}$$

def: pr predicates Predicate P is pr iff C_P is pr.

examples

- $x < y$

$$C_{x<y} = sg(x \dot{-} y)$$

- if P is pr predicate on $\mathbb{N}_0^r$ and

$$g_1, \ldots, g_r : \mathbb{N}^m \to \mathbb{N}_0$$

  are pr, then

$$Q(x) = P(g_1(x), \ldots, g_r(x))$$

  is pr predicate on $\mathbb{N}_0^m$

- $2x < y$

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \to \{true, false\}$$

examples: with $r = 2$

$$x \leq y \;,\; x|y \;,\; 2x < y$$

def: characteristic function of predicate P:

$$C_P : \mathbb{N}_0^r \to \mathbb{B}$$

$$C_P(x) = \begin{cases} 1 & P(x) = true \\ 0 & P(x) = false \end{cases}$$

def: pr predicates Predicate P is pr iff C_P is pr.

examples

- $x < y$

$$C_{x<y} = sg(x \dot{-} y)$$

- if P is pr predicate on $\mathbb{N}_0^r$ and

$$g_1, \ldots, g_r : \mathbb{N}^m \to \mathbb{N}_0$$

  are pr, then

$$Q(x) = P(g_1(x), \ldots, g_r(x))$$

  is pr predicate on $\mathbb{N}_0^m$

- $2x < y$

Lemma 7. *If P and Q are pr predicates, then so are*

$$P \wedge Q \;,\; P \vee Q \;,\; \sim P$$

def: predicate on N_0^r for the time being just a function

$$P : \mathbb{N}_0^r \to \{true, false\}$$

examples: with $r = 2$

$$x \leq y \text{ , } x|y \text{ , } 2x < y$$

def: characteristic function of predicate P:

$$C_P : \mathbb{N}_0^r \to \mathbb{B}$$

$$C_P(x) = \begin{cases} 1 & P(x) = true \\ 0 & P(x) = false \end{cases}$$

def: pr predicates Predicate P is pr iff C_P is pr.

examples

- $x < y$

$$C_{x<y} = sg(x \dot{-} y)$$

- if P is pr predicate on $\mathbb{N}_0^r$ and

$$g_1, \ldots, g_r : \mathbb{N}^m \to \mathbb{N}_0$$

are pr, then

$$Q(x) = P(g_1(x), \ldots, g_r(x))$$

is pr predicate on $\mathbb{N}_0^m$

- $2x < y$

Lemma 7. *If P and Q are pr predicates, then so are*

$$P \wedge Q \text{ , } P \vee Q \text{ , } \sim P$$

$$\begin{aligned} C_{P \wedge Q}(x) &= C_P(x) \cdot C_Q(x) \\ C_{P \vee Q}(x) &= sg(C_P(x) + C_Q(x)) \\ C_{\sim P}(x) &= \overline{sg}(P(x)) \end{aligned}$$

- $x = y$

$$x = y \leftrightarrow \sim ((x < y) \vee (y < x))$$

- $x \leq y$

$$x \leq y \leftrightarrow x(x < y) \vee (x = y)$$

definition by case split

Lemma 8. *If*

$$P_1, \ldots, P_k$$

are pr predicates on $\mathbb{N}_0^r$ *and*

$$f_1, \ldots, f_r : \mathbb{N}_0^r \to \mathbb{N}_0$$

are pr functions and for each $x \in \mathbb{N}_0^r$ *there is at most one* i *such that* $P_i(x)$ *is true. Then*

$$g(x) = \begin{cases} f_1(x) & P_1(x) \\ \ldots & \\ f_k(x) & P_k(x) \\ 0 & \textit{otherwise} \end{cases}$$

is pr.

$$g(x) = C_{P_1}(x) \cdot f_1(x) + \ldots + C_{P_k}(x) \cdot f_k(x)$$

definition by case split

Lemma 8. *If*

$$P_1, \ldots, P_k$$

are pr predicates on $\mathbb{N}_0^r$ *and*

$$f_1, \ldots, f_r : \mathbb{N}_0^r \to \mathbb{N}_0$$

are pr functions and for each $x \in \mathbb{N}_0^r$ *there is at most one* i *such that* $P_i(x)$ *is true. Then*

$$g(x) = \begin{cases} f_1(x) & P_1(x) \\ \ldots & \\ f_k(x) & P_k(x) \\ 0 & \textit{otherwise} \end{cases}$$

is pr.

$$g(x) = C_{P_1}(x) \cdot f_1(x) + \ldots + C_{P_k}(x) \cdot f_k(x)$$

finite functions

Lemma 9. *Let* $f : \mathbb{N}_0^r \to \mathbb{N}_0$. *If* $f(x) \neq 0$ *for finitely many* x, *then* f *is pr.*

8 bounded μ-operator

def: bounded μ-operator For

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

we define

$$\mu_b f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

by

$$\boxed{\mu_b f(n,x) = \begin{cases} \min\{m \;:\; f(m,x) = 0,\; m \leq n\} & \text{if it exists} \\ 0 & \text{otherwise} \end{cases}}$$

$\mu_b f(n,x)$ returns smallest solution of equation $f(m,x) = 0$ in the bounded range $[0:n]$ if it exists...

8 bounded μ-operator

def: bounded μ-operator For

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

we define

$$\mu_b f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

by

$$\boxed{\mu_b f(n,x) = \begin{cases} \min\{m \,:\, f(m,x) = 0,\ m \le n\} & \text{if it exists} \\ 0 & \text{otherwise} \end{cases}}$$

$\mu_b f(n,x)$ returns smallest solution of equation $f(m,x) = 0$ in the bounded range $[0 : n]$ if it exists...

Lemma 10. *If f is pr, then $\mu_b f$ is pr*

$$\begin{aligned} \mu_b f(n,0) &= 0 \\ \mu_b f(n+1,x) &= \begin{cases} \mu_b f(n,x) & \mu_b f(n,x) \neq 0 \\ n+1 & f(n+1,x) = 0 \wedge \mu_b f(n,x) = 0 \wedge f(0,x) \neq 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

8 bounded μ-operator

def: bounded μ-operator For

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

we define

$$\mu_b f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

by

$$\boxed{\mu_b f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0, \ m \le n\} & \text{if it exists} \\ 0 & \text{otherwise} \end{cases}}$$

$\mu_b f(n,x)$ returns smallest solution of equation $f(m,x) = 0$ in the bounded range $[0:n]$ if it exists...

Lemma 10. *If f is pr, then $\mu_b f$ is pr*

$$\begin{aligned} \mu_b f(n,0) &= 0 \\ \mu_b f(n+1,x) &= \begin{cases} \mu_b f(n,x) & \mu_b f(n,x) \neq 0 \\ n+1 & f(n+1,x) = 0 \wedge \mu_b f(n,x) = 0 \wedge f(0,x) \neq 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

examples:

- $\lfloor x/y \rfloor$

$$\lfloor x/y \rfloor = \min\{m \ : \ m \le x \ \wedge \ (m+1)y > x\}$$

- divides: $y|x$

$$\begin{aligned} rem(x,y) &= x \dot{-} y \cdot \lfloor x/y \rfloor \\ C_|(x,y) &= \overline{sg}(rem(x,y)) \end{aligned}$$

Lemma 11. *For all* $k \leq 2$ *and* $i \in [1:k]$ *there are pr functions*

$$\begin{aligned} b^{(k)} &: \mathbb{N}_0^k \to \mathbb{N}_0 \quad \textit{bijective} \\ b_i^{(k)} &: \mathbb{N}_0 \to \mathbb{N}_0 \end{aligned}$$

such that for all $x \in \mathbb{N}_0^k$ *and all* i

$$b_i^{(k)}(b^{(k)}(x_1,\ldots,x_k)) = x_i$$

Lemma 11. *For all $k \leq 2$ and $i \in [1:k]$ there are pr functions*

$$\begin{aligned} b^{(k)} &: \mathbb{N}_0^k \to \mathbb{N}_0 \quad \textit{bijective} \\ b_i^{(k)} &: \mathbb{N}_0 \to \mathbb{N}_0 \end{aligned}$$

such that for all $x \in \mathbb{N}_0^k$ and all i

$$b_i^{(k)}(b^{(k)}(x_1,\ldots,x_k)) = x_i$$

- $k = 2$: enumerate as in figure 1

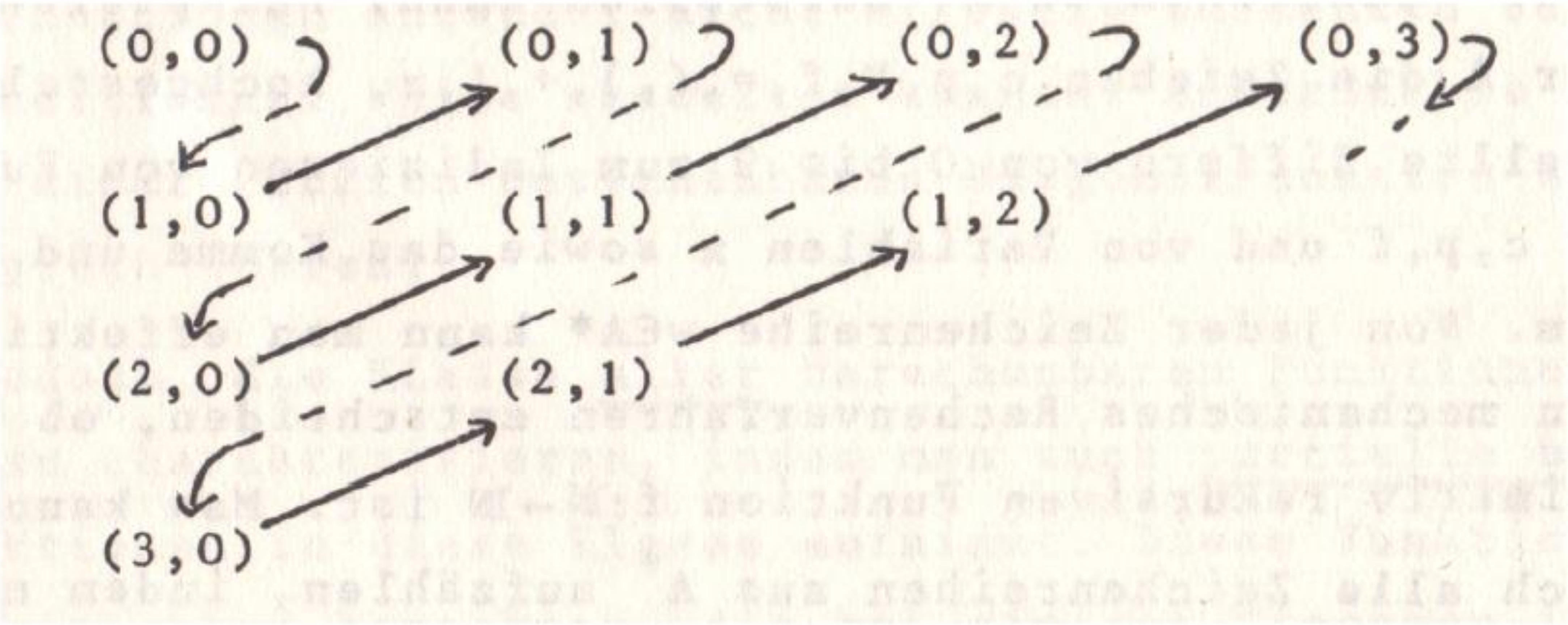

Figure 1: map $(0,0) \to 0$, $(1,0) \to 1$, $(0,1) \to 2$ etc

- diagonals $1,2,\ldots$ each with positions $0,1,\ldots$

primitive recursive bijections

Lemma 11. *For all $k \le 2$ and $i \in [1:k]$ there are pr functions*

$$\begin{aligned} b^{(k)} &: \mathbb{N}_0^k \to \mathbb{N}_0 \quad \textit{bijective} \\ b_i^{(k)} &: \mathbb{N}_0 \to \mathbb{N}_0 \end{aligned}$$

such that for all $x \in \mathbb{N}_0^k$ and all i

$$b_i^{(k)}(b^{(k)}(x_1,\dots,x_k)) = x_i$$

- $k = 2$: enumerate as in figure 1

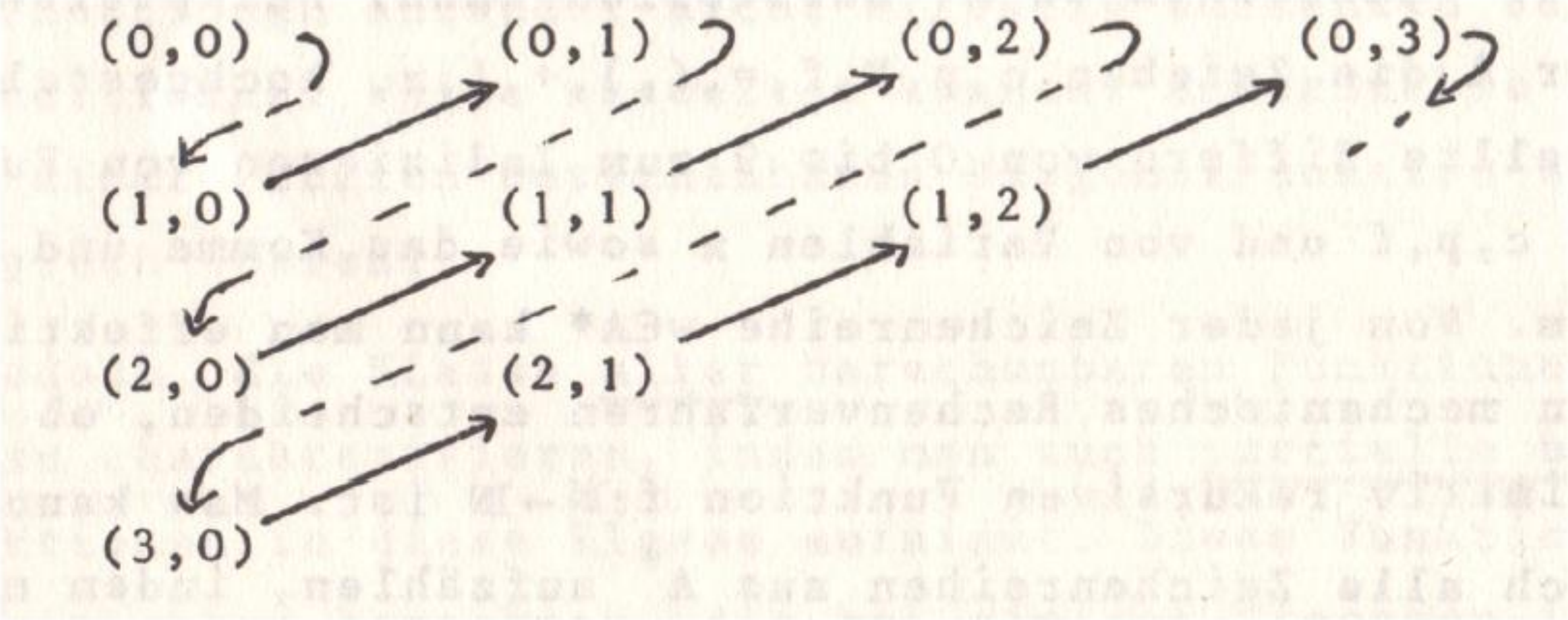

Figure 1: map $(0,0) \to 0$, $(1,0) \to 1$, $(0,1) \to 2$ etc

- diagonals $1,2,\dots$ each with positions $0,1,\dots$

- then

$$(n,m) \quad \text{stands on diagonal } n+m+1 \text{ at position } m$$

-

$$\begin{aligned} b^{(2)}(n,m) &= \sum_{i=1}^{n+m} i + m \\ &= \frac{(n+m)(n+m+1)}{2} + m \quad \text{this is pr} \end{aligned}$$

primitive recursive bijections

Lemma 11. *For all $k \leq 2$ and $i \in [1:k]$ there are pr functions*

$$\begin{aligned} b^{(k)} &: \mathbb{N}_0^k \to \mathbb{N}_0 \quad \textit{bijective} \\ b_i^{(k)} &: \mathbb{N}_0 \to \mathbb{N}_0 \end{aligned}$$

such that for all $x \in \mathbb{N}_0^k$ and all i

$$b_i^{(k)}(b^{(k)}(x_1,\ldots,x_k)) = x_i$$

- $k = 2$: enumerate as in figure 1

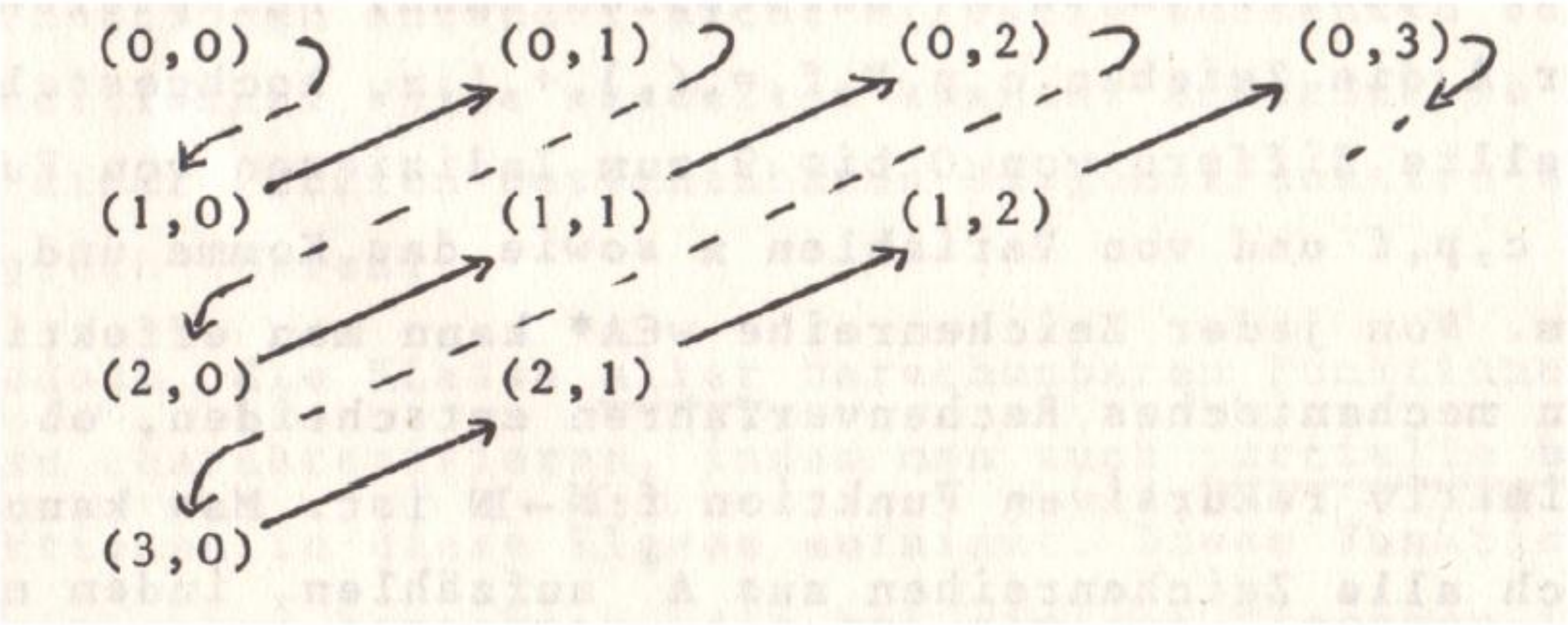

Figure 1: map $(0,0) \to 0$, $(1,0) \to 1$, $(0,1) \to 2$ etc

- diagonals $1,2,\ldots$ each with positions $0,1,\ldots$
- then

  (n,m) stands on diagonal $n+m+1$ at position m

-

$$\begin{aligned} b^{(2)}(n,m) &= \sum_{i=1}^{n+m} i + m \\ &= \frac{(n+m)(n+m+1)}{2} + m \quad \text{this is pr} \end{aligned}$$

- inverse mappings: given

$$b^{(2)}(n,m) = \sum_{i=1}^{s} i + m = x \quad \text{solve } s = ?$$

$$\begin{aligned} s(x) &= \max\{j : \frac{j(j+1)}{2} \leq x \,,\; j \leq x\} \\ &= \min\{j : \frac{j(j+1)}{2} > x \,,\; j \leq x\} \quad \text{this is pr} \\ b_2^{(2)}(x) &= x \dot{-} \frac{s(x) \cdot (s(x)+1)}{2} \quad (= m) \\ b_1^{(2)}(x) &= s(x) \dot{-} b_2^{(2)}(x) \quad (= n) \end{aligned}$$

Lemma 11. *For all $k \leq 2$ and $i \in [1:k]$ there are pr functions*

$$\begin{aligned} b^{(k)} &: \mathbb{N}_0^k \to \mathbb{N}_0 \quad \textit{bijective} \\ b_i^{(k)} &: \mathbb{N}_0 \to \mathbb{N}_0 \end{aligned}$$

such that for all $x \in \mathbb{N}_0^k$ and all i

$$b_i^{(k)}(b^{(k)}(x_1,\ldots,x_k)) = x_i$$

- $k > 2$: the fun part:

$$\begin{aligned} b^{(k)}(x_1,\ldots,b_k) &= b^{(2)}(x_1,b^{(k-1)}(x_2,\ldots,x_k)) \\ b_1^{(k)}(x) &= b_1^{(2)}(x) \\ b_i^{(k)}(x) &= b_{i-1}^{(k-1)}(b_2^{(2)}(x))) \quad \text{for } i > 1 \end{aligned}$$

9 μ-operator and μ-recursive functions

def: (unbounded) μ-operator For

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

we define

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

by

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \text{if it exists} \\ \Omega & \text{(undefined) otherwise} \end{cases}$$

$\mu f(n,x)$ returns smallest solution of equation $f(m,x) = 0$ if it exists. Could be implemented as unbounded while loop:

```
m=0;
while f(m,x) != 0 {m = m+1}; return m
```

9 μ-operator and μ-recursive functions

def: (unbounded) μ-operator For

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

we define

$$\mu f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

by

$$\mu f(n,x) = \begin{cases} \min\{m \ : \ f(m,x) = 0\} & \text{if it exists} \\ \Omega & \text{(undefined) otherwise} \end{cases}$$

$\mu f(n,x)$ returns smallest solution of equation $f(m,x) = 0$ if it exists. Could be implemented as unbounded while loop:

```
m=0;
while f(m,x) != 0 {m = m+1}; return m
```

Inductive definition of a set R of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

1. constant functions $c_s^r \in R$
2. projections $p_i^r \in R$
3. successor function $S \in R$
4. substitution. If the following function are all in R

   $$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

   then also $h \in R$ where

   $$h : \mathbb{N}_0^m \to \mathbb{N}_0$$
   $$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in R

   $$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

   then also $f \in R$ where

   $$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

   $$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

6. if $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ is in R, then μf is in R
7. these are all

9 μ-operator and μ-recursive functions

Inductive definition of a set R of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

1. constant functions $c_s^r \in R$
2. projections $p_i^r \in R$
3. successor function $S \in R$
4. substitution. If the following function are all in R

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

   then also $h \in R$ where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$
$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in R

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \text{ , } h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

   then also $f \in R$ where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

6. if $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ is in R, then μf is in R
7. these are all

what can be computed with functions in R?

- Answer: everything that can be computed at all
- this is known as *Church's thesis*.
- We cannot prove it (based on what definition or axiom could we do that?)
- we *can* supply evidence, and we *will* do it. Of course *here*

9 μ-operator and μ-recursive functions

Inductive definition of a set R of computable functions:

$$f : \mathbb{N}_0^r \to \mathbb{N}_0$$

1. constant functions $c_s^r \in R$
2. projections $p_i^r \in R$
3. successor function $S \in R$
4. substitution. If the following function are all in R

$$f : \mathbb{N}_0^r \to \mathbb{N} \text{ and } g_1, \ldots, g_r : \mathbb{N}_0^m \to \mathbb{N}_0$$

   then also $h \in R$ where

$$h : \mathbb{N}_0^m \to \mathbb{N}_0$$
$$h(x) = f(g_1(x), \ldots, g_r(x))$$

5. primitive recursion. If the following functions are in R

$$g : \mathbb{N}_0^r \to \mathbb{N}_0 \ , \ h : \mathbb{N}_0^{r+2} \to \mathbb{N}_0$$

   then also $f \in R$ where

$$f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$$

$$\begin{aligned} f(0,x) &= g(x) \\ f(n+1,x) &= h(n, f(n,x), x) \end{aligned}$$

6. if $f : \mathbb{N}_0^{r+1} \to \mathbb{N}_0$ is in R, then μf is in R
7. these are all

what can be computed with functions in R?

- Answer: everything that can be computed at all
- this is known as *Church's thesis*.
- We cannot prove it (based on what definition or axiom could we do that?)
- we *can* supply evidence, and we *will* do it. Of course *here*

hope in favor of Church's thesis proof that pr functions do not compute all computable functions relied on fact, that all pr functions are total. That proof collapses for μ-recursive functions. Exercise: in which line?