Computability Theory

Aart Middeldorp

Outline

Definition

class PR of primitive recursive functions is smallest class of total functions $f \colon \mathbb{N}^n \to \mathbb{N}$ that contains all initial functions

- zero $\quad\quad\quad$ $\mathsf{z}(x) = 0$
- successor $\quad\quad$ $\mathsf{s}(x) = x + 1$
- projection $\quad\quad$ $\pi_i^n(x_1, \ldots, x_n) = x_i$ $\quad$ for all $n \geqslant 1$ and $1 \leqslant i \leqslant n$

and is closed under composition

- $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x})) \in \mathsf{PR}$ $\quad$ for all $g \colon \mathbb{N}^m \to \mathbb{N} \in \mathsf{PR}$ and $h_1, \ldots, h_m \colon \mathbb{N}^n \to \mathbb{N} \in \mathsf{PR}$

and primitive recursion

- $f(x, \vec{y}) \colon \mathbb{N}^{n+1} \to \mathbb{N}$ defined by

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

belongs to PR $\quad$ for all $g \colon \mathbb{N}^n \to \mathbb{N} \in \mathsf{PR}$ and $h \colon \mathbb{N}^{n+2} \to \mathbb{N} \in \mathsf{PR}$

Definition

predicate $P \colon \mathbb{N}^n \to \mathbb{B}$ is primitive recursive if its characteristic function $\chi_P \colon \mathbb{N}^n \to \mathbb{N}$

$$\chi_P(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } P(x_1, \ldots, x_n) \\ 0 & \text{otherwise} \end{cases}$$

is primitive recursive

Lemma

PR is closed under iteration, case analysis and bounded minimization

Definitions

▶ pairing function $\pi(x, y) = 2^x(2y + 1) \dot{-} 1$

▶ extraction functions
$$\pi_1(z) = (\mu\, x \leqslant z)\, (\exists\, y \leqslant z)\, \big[\, z = \pi(x, y)\, \big] \qquad \pi_2(z) = (\mu\, y \leqslant z)\, (\exists\, x \leqslant z)\, \big[\, z = \pi(x, y)\, \big]$$

Part I: Recursive Function Theory

Ackermann function, bounded minimization, bounded recursion, course−of−values recursion, diagonalization, diophantine sets, elementary functions, fixed point theorem, Fibonacci numbers, Gödel numbering, Gödel's β function, Grzegorczyk hierarchy, loop programs, minimization, normal form theorem, partial recursive functions, primitive recursion, recursive enumerability, recursive inseparability, s−m−n theorem, total recursive functions, undecidability, while programs, . . .

Part II: Combinatory Logic and Lambda Calculus

α−equivalence, abstraction, arithmetization, β−reduction, CL−representability, combinators, combinatorial completeness, Church numerals, Church−Rosser theorem, Curry−Howard isomorphism, de Bruijn notation, η−reduction, fixed point theorem, intuitionistic propositional logic, λ−definability, normalization theorem, termination, typing, undecidability, Z property, . . .

Outline

Definitions

- **Gödel number** of sequence $x_1, \ldots, x_n$: $\quad \langle x_1, \ldots, x_n \rangle = p_0^n \times p_1^{x_1} \times \cdots \times p_n^{x_n}$

- $(x)_i = (\mu\, j < x)\, \neg(p_i^{j+1} \mid x)$

- $\operatorname{len}(x) = (x)_0$

- $\operatorname{seq}(x) \iff x > 0 \wedge (\forall\, i \leqslant x)\,\big[\, (x)_i \neq 0 \implies i \leqslant \operatorname{len}(x) \,\big]$

- $x \,;\, y = p_0^{\operatorname{len}(x) + \operatorname{len}(y)} \times \displaystyle\prod_{i\,<\,\operatorname{len}(x)} p_{i+1}^{(x)_{i+1}} \times \prod_{i\,<\,\operatorname{len}(y)} p_{\operatorname{len}(x)+i+1}^{(y)_{i+1}}$

Lemma

- $x = \langle x_1, \ldots, x_n \rangle$ $\qquad\qquad\qquad\quad \implies \operatorname{len}(x) = n \wedge (x)_i = x_i \quad$ for all $1 \leqslant i \leqslant n$
- $\operatorname{seq}(x) \wedge \operatorname{len}(x) = n$ $\qquad\qquad\quad\;\, \implies x = \langle (x)_1, \ldots, (x)_n \rangle$
- $x = \langle x_1, \ldots, x_n \rangle \wedge y = \langle y_1, \ldots, y_m \rangle \implies x \,;\, y = \langle x_1, \ldots, x_n, y_1, \ldots, y_m \rangle$

Outline

Definition

if $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ then $\tilde{f}(x,\vec{y}) = \langle f(0,\vec{y}), \ldots, f(x,\vec{y}) \rangle$

Lemma

if $g\colon \mathbb{N}^n \to \mathbb{N}$ and $h\colon \mathbb{N}^{n+2} \to \mathbb{N}$ are primitive recursive then so is $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ defined as

$$f(0,\vec{y}) = g(\vec{y}) \qquad\qquad f(x+1,\vec{y}) = h(\tilde{f}(x,\vec{y}),x,\vec{y})$$

Proof

$f(x,\vec{y}) = (\tilde{f}(x,\vec{y}))_{x+1}$ and $\tilde{f}$ is primitive recursive:

$$\tilde{f}(0,\vec{y}) = \langle f(0,\vec{y}) \rangle = \langle g(\vec{y}) \rangle$$

$$\tilde{f}(x+1,\vec{y}) = \tilde{f}(x,\vec{y}) \,;\, \langle f(x+1,\vec{y}) \rangle = \tilde{f}(x,\vec{y}) \,;\, \langle h(\tilde{f}(x,\vec{y}),x,\vec{y}) \rangle = h'(\tilde{f}(x,\vec{y}),x,\vec{y})$$

with $h'(x,y,\vec{z}) = x \,;\, \langle h(x,y,\vec{z}) \rangle$

Outline

Definition

$$\text{ack}(0, y) = y + 1$$
$$\text{ack}(x + 1, 0) = \text{ack}(x, 1)$$
$$\text{ack}(x + 1, y + 1) = \text{ack}(x, \text{ack}(x + 1, y))$$

Example

- $\text{ack}(1, 3) = 5$
- $\text{ack}(2, 2) = 7$
- $\text{ack}(3, 4) = 125$
- $\text{ack}(4, 1) = 65533$

Lemma

Ackermann function is well-defined

Example (cont'd)

- $\text{ack}(4, 3) = 2^{2^{65533}} - 3$

Remarks

- Ackermann function grows very fast
- Ackermann function is used as compiler benchmark
- inverse of Ackermann function appears in complexity analysis of certain algorithms

Lemma

❶ $\text{ack}(x, y) > y$

❷ $\text{ack}(x, y + 1) > \text{ack}(x, y)$

❸ $\text{ack}(x + 1, y) \geqslant \text{ack}(x, y + 1)$

❹ $\text{ack}(x + 2, y) > \text{ack}(x, 2y)$

▸ skip proofs

Lemma ❶

- $\mathsf{ack}(x, y) > y$

Proof

induction on x

- $\mathsf{ack}(0, y) = y + 1 > y$
- induction on y

$$\mathsf{ack}(x + 1, 0) = \mathsf{ack}(x, 1) > 1 > 0$$
$$\mathsf{ack}(x + 1, y + 1) = \mathsf{ack}(x, \mathsf{ack}(x + 1, y)) > \mathsf{ack}(x + 1, y) \geqslant y + 1$$

Lemma ❷

- $\mathsf{ack}(x, y+1) > \mathsf{ack}(x, y)$

Proof

induction on x

$$\mathsf{ack}(0, y+1) = y+2 > y+1 = \mathsf{ack}(0, y)$$
$$\mathsf{ack}(x+1, y+1) = \mathsf{ack}(x, \mathsf{ack}(x+1, y)) > \mathsf{ack}(x+1, y)$$

Lemma ❸

- $\mathrm{ack}(x + 1, y) \geqslant \mathrm{ack}(x, y + 1) > \mathrm{ack}(x, y)$

Proof

induction on x

- $\mathrm{ack}(1, y) \geqslant y + 2 = \mathrm{ack}(0, y + 1)$ by induction on y

$$\mathrm{ack}(1, 0) = \mathrm{ack}(0, 1) = 2$$
$$\mathrm{ack}(1, y + 1) = \mathrm{ack}(0, \mathrm{ack}(1, y)) \geqslant \mathrm{ack}(0, y + 2)$$

- induction on y

$$\mathrm{ack}(x + 2, 0) = \mathrm{ack}(x + 1, 1)$$
$$\mathrm{ack}(x + 2, y + 1) = \mathrm{ack}(x + 1, \mathrm{ack}(x + 2, y)) \geqslant \mathrm{ack}(x + 1, \mathrm{ack}(x + 1, y + 1))$$
$$\geqslant \mathrm{ack}(x + 1, \mathrm{ack}(x, y + 2)) > \mathrm{ack}(x + 1, y + 2)$$

Lemma ❹

▶ $\mathrm{ack}(x+2, y) > \mathrm{ack}(x, 2y)$

Proof

induction on x

▶ $\mathrm{ack}(2, y) > \mathrm{ack}(0, 2y)$ by induction on y

$$\mathrm{ack}(2, 0) \geqslant \mathrm{ack}(1, 1) \geqslant \mathrm{ack}(0, 2) > \mathrm{ack}(0, 1) > \mathrm{ack}(0, 0)$$
$$\mathrm{ack}(2, y+1) = \mathrm{ack}(1, \mathrm{ack}(2, y)) > \mathrm{ack}(1, 2y+1) \geqslant \mathrm{ack}(0, 2y+2)$$

▶ induction on y

$$\mathrm{ack}(x+3, 0) \geqslant \mathrm{ack}(x+2, 1) \geqslant \mathrm{ack}(x+1, 2) > \mathrm{ack}(x+1, 0)$$
$$\mathrm{ack}(x+3, y+1) = \mathrm{ack}(x+2, \mathrm{ack}(x+3, y)) > \mathrm{ack}(x+2, \mathrm{ack}(x+1, 2y))$$
$$\geqslant \mathrm{ack}(x+2, \mathrm{ack}(x, 2y+1)) > \mathrm{ack}(x+2, 2y+1)$$
$$\geqslant \mathrm{ack}(x+1, 2y+2)$$

Lemma

$\forall$ primitive recursive function $f \colon \mathbb{N}^n \to \mathbb{N}$ $\exists$ constant $c \in \mathbb{N}$ such that

$$f(x_1, \ldots, x_n) < \mathrm{ack}(c, \max\{x_1, \ldots, x_n\})$$

Proof

induction on definition of primitive recursive functions

► initial functions

$z(x) = 0 < x + 1 = \mathrm{ack}(0, x)$

$s(x) = x + 1 = \mathrm{ack}(0, x) < \mathrm{ack}(1, x)$

$\pi_i^n(x_1, \ldots, x_n) = x_i \leqslant \max\{x_1, \ldots, x_n\} < \mathrm{ack}(0, \max\{x_1, \ldots, x_n\})$

Lemma

$\forall$ primitive recursive function $f \colon \mathbb{N}^n \to \mathbb{N}$ $\exists$ constant $c \in \mathbb{N}$ such that

$$f(x_1, \ldots, x_n) < \mathrm{ack}(c, \max\{x_1, \ldots, x_n\})$$

Proof (cont'd)

induction on definition of primitive recursive functions

- composition $f(\vec{x}) = g(h_1(\vec{x}), \ldots, h_m(\vec{x}))$

  $g(\vec{y}) < \mathrm{ack}(c_0, \max\{\vec{y}\})$ and $h_i(\vec{x}) < \mathrm{ack}(c_i, \max\{\vec{x}\})$ for all $1 \leqslant i \leqslant m$

  $c' = \max\{c_0 + 1, c_1, \ldots, c_m\}$

  $$
  \begin{aligned}
  f(\vec{x}) &< \mathrm{ack}(c_0, \max\{h_1(\vec{x}), \ldots, h_m(\vec{x})\}) \\
  &< \mathrm{ack}(c_0, \max\{\mathrm{ack}(c_1, \max\{\vec{x}\}), \ldots, \mathrm{ack}(c_m, \max\{\vec{x}\})\}) \\
  &= \mathrm{ack}(c_0, \mathrm{ack}(\max\{c_1, \ldots, c_m\}, \max\{\vec{x}\})) \\
  &\leqslant \mathrm{ack}(c' - 1, \mathrm{ack}(c', \max\{\vec{x}\})) = \mathrm{ack}(c', \max\{\vec{x}\} + 1) \leqslant \mathrm{ack}(c' + 1, \max\{\vec{x}\})
  \end{aligned}
  $$

Lemma

$\forall$ primitive recursive function $f \colon \mathbb{N}^n \to \mathbb{N}$ $\exists$ constant $c \in \mathbb{N}$ such that

$$f(x_1, \ldots, x_n) < \mathrm{ack}(c, \max\{x_1, \ldots, x_n\})$$

Proof (cont'd)

induction on definition of primitive recursive functions

- primitive recursion $\quad f(0, \vec{y}) = g(\vec{y})$
  $$f(x+1, \vec{y}) = h(f(x, \vec{y}), x, \vec{y})$$

  $g(\vec{y}) < \mathrm{ack}(c_1, \max\{\vec{y}\})$ and $h(a, b, \vec{y}) < \mathrm{ack}(c_2, \max\{a, b, \vec{y}\})$

  $c' = \max\{c_1, c_2 + 1\}$

  $f(x, \vec{y}) < \mathrm{ack}(c', x + \max\{\vec{y}\})$ by induction on x

  $$f(x, \vec{y}) < \mathrm{ack}(c', 2 \cdot \max\{x, \vec{y}\}) < \mathrm{ack}(c' + 2, \max\{x, \vec{y}\})$$

Theorem

Ackermann function is not primitive recursive

Proof

if Ackermann function is primitive recursive then

$$\mathrm{ack}(x, y) < \mathrm{ack}(c, \max\{x, y\})$$

for some constant c and thus

$$\mathrm{ack}(c, c) < \mathrm{ack}(c, \max\{c, c\}) = \mathrm{ack}(c, c) \qquad \lightning$$

Outline

Definition

index $\ulcorner f \urcorner \in \mathbb{N}$ of derivation of primitive recursive function f is defined inductively:

- $\ulcorner z \urcorner = \langle 0 \rangle$
- $\ulcorner s \urcorner = \langle 1 \rangle$
- $\ulcorner \pi_i^n \urcorner = \langle 2, n, i \rangle$
- $\ulcorner f \urcorner = \langle 3, \ulcorner g \urcorner, \ulcorner h_1 \urcorner, \ldots, \ulcorner h_m \urcorner \rangle$ if f is obtained by composing g and $h_1, \ldots, h_m$
- $\ulcorner f \urcorner = \langle 4, \ulcorner g \urcorner, \ulcorner h \urcorner \rangle$ if f is obtained by primitive recursion from g and h

Examples

- $\ulcorner + \urcorner = \langle 4, \langle 2, 1, 1 \rangle, \langle 3, \langle 1 \rangle, \langle 2, 3, 1 \rangle \rangle \rangle$
- $397065375000 = 2^3 \times 3^3 \times 5^6 \times 7^6 = \langle 3, \langle 1 \rangle, \langle 1 \rangle \rangle = \ulcorner s \circ s \urcorner$

Remark (key insight)

from index we can reconstruct function   (details in later lecture)

Corollary

$\exists$ computable list $f_0, f_1, f_2, f_3, \ldots$ of unary primitive recursive functions

Diagonalization

$$
\begin{array}{cccccc}
f_0(0) & f_0(1) & f_0(2) & f_0(3) & \cdots \\
f_1(0) & f_1(1) & f_1(2) & f_1(3) & \cdots \\
f_2(0) & f_2(1) & f_2(2) & f_2(3) & \cdots \\
f_3(0) & f_3(1) & f_3(2) & f_3(3) & \cdots \\
\vdots & \vdots & \vdots & \vdots & \ddots
\end{array}
\qquad g(x) = f_x(x) + 1
$$

Corollary

function g is computable but not primitive recursive

Outline

Definition

class R of recursive functions is smallest class of total functions that contains all initial functions and is closed under composition, primitive recursion, and (unbounded) minimization:

$$(\mu\, i)\, P(i, \vec{y}) = \min \{ i \mid P(i, \vec{y}) \} \in \mathsf{R}$$

for all $P \colon \mathbb{N}^{n+1} \to \mathbb{B}$ such that $\chi_P \in \mathsf{R}$ and $\forall \vec{y} \, \exists x \, P(x, \vec{y})$
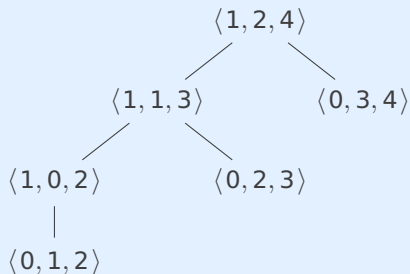
Theorem

Ackermann function is recursive

Representing Computations

$$\mathsf{ack}(0, y) = y + 1$$
$$\mathsf{ack}(x + 1, 0) = \mathsf{ack}(x, 1)$$
$$\mathsf{ack}(x + 1, y + 1) = \mathsf{ack}(x, \mathsf{ack}(x + 1, y))$$

$\mathsf{ack}(1, 2) = 4$:
$$\langle 1, 2, 4 \rangle$$

$$\langle 1, 1, 3 \rangle \qquad\qquad \langle 0, 3, 4 \rangle$$

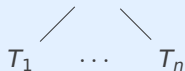$$\langle 1, 0, 2 \rangle \qquad\qquad \langle 0, 2, 3 \rangle$$

$$\langle 0, 1, 2 \rangle$$

$\langle x, y, z \rangle$ denotes $\mathsf{ack}(x, y) = z$

Definition

encode computation tree $T = \langle x, y, z \rangle$ as number $\ulcorner T \urcorner = \langle \langle x, y, z \rangle, \ulcorner T_1 \urcorner, \dots, \ulcorner T_n \urcorner \rangle$

Lemma

$\exists$ primitive recursive predicate $P \colon \mathbb{N} \to \mathbb{B}$ such that

$$P(x) \iff x \text{ encodes correct computation tree}$$

Corollary

- $\mathrm{ack}(x, y) = z \iff \exists\, t$ such that $P(t)$ and $(t)_1 = \langle x, y, z \rangle$
- $\mathrm{ack}(x, y) = \big((\mu\, t)\, (P(t) \wedge (t)_{1,1} = x \wedge (t)_{1,2} = y)\big)_{1,3}$

Proof

- $P(x) \iff \text{seq}(x) \land \text{seq}((x)_1) \land \text{len}((x)_1) = 3 \land (A(x) \lor B(x) \lor C(x))$

- $A(x) \iff \text{len}(x) = 1 \land (x)_{1,1} = 0 \land (x)_{1,3} = \text{s}((x)_{1,2})$

- $B(x) \iff \text{len}(x) = 2 \land (x)_{1,1} > 0 \land (x)_{1,2} = 0 \land (x)_{2,1,1} = \text{p}((x)_{1,1}) \land (x)_{2,1,2} = 1$
  $\land (x)_{2,1,3} = (x)_{1,3} \land P((x)_2)$

- $C(x) \iff \text{len}(x) = 3 \land (x)_{1,1} > 0 \land (x)_{1,2} > 0 \land (x)_{2,1,1} = (x)_{1,1} \land (x)_{2,1,2} = \text{p}((x)_{1,2})$
  $\land (x)_{3,1,1} = \text{p}((x)_{1,1}) \land (x)_{3,1,2} = (x)_{2,1,3} \land (x)_{3,1,3} = (x)_{1,3} \land P((x)_2) \land P((x)_3)$

$$T = \langle x, y, z \rangle \qquad \ulcorner T \urcorner = \langle \langle x, y, z \rangle, \ulcorner T_1 \urcorner, \ldots, \ulcorner T_n \urcorner \rangle$$

$$T_1 \quad \cdots \quad T_n$$

Outline

Important Concepts

- Ackermann function
- course−of−values recursion
- diagonalization

- Gödel numbering
- index
- R

- recursive function
- (unbounded) minimization

homework for October 16