universität innsbruck

WS 2023 lecture 5

Computability Theory

Aart Middeldorp

Outline

1. **Summary of Previous Lecture**
2. **Normal Form Theorem**
3. **Undecidability**
4. **s–m–n or Parameterization Theorem**
5. **Fixed Point Theorem**
6. **Summary**

Theorem

class R of recursive functions is smallest class of total functions that contains all projection functions, addition and multiplication, characteristic function $\chi_=$ of equality predicate, and is closed under composition and minimization

Definition

$\beta(a,i) = \pi_1(a) \bmod \big(1 + (i+1)\,\pi_2(a)\big)$ Gödel's β function

Definition

class PA of partial recursive functions is smallest class of partial functions that contains all initial functions and is closed under composition, primitive recursion and unbounded minimization:

$$(\mu\, i)\,(f(i,\vec{y}) = 0) = \min\{\, i \mid f(i,\vec{y}) = 0 \text{ and } f(j,\vec{y}) > 0 \text{ for all } j < i \,\}$$

belongs to PA whenever $f\colon \mathbb{N}^{n+1} \to \mathbb{N}$ belongs to PA

Definition

function $f\colon \mathbb{N}^n \to \mathbb{N}$ is WHILE computable if $\exists$ WHILE program $P(x_1, \dots, x_n; y)$ such that $y = f(x_1, \dots, x_n)$ after execution of P

Theorem

function φ is partial recursive $\iff$ φ is WHILE computable

Definition

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

- $\ulcorner \mathsf{z} \urcorner = \langle 0 \rangle$ $\quad \ulcorner \mathsf{s} \urcorner = \langle 1 \rangle$ $\quad \ulcorner \pi_i^n \urcorner = \langle 2, n, i \rangle$
- $\ulcorner f \urcorner = \langle 3, \ulcorner g \urcorner, \ulcorner h_1 \urcorner, \dots, \ulcorner h_m \urcorner \rangle$ if f is obtained by composing g and $h_1, \dots, h_m$
- $\ulcorner f \urcorner = \langle 4, \ulcorner g \urcorner, \ulcorner h \urcorner \rangle$ if f is obtained by primitive recursion from g and h
- $\ulcorner f \urcorner = \langle 5, \ulcorner g \urcorner \rangle$ if f is obtained by minimizing g

Kleene's Normal Form Theorem

$\exists$ primitive recursive function u $\forall n \geqslant 1$ $\exists$ primitive recursive predicate T_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1, \dots, x_n) \simeq u((\mu y)\, T_n(\ulcorner \varphi \urcorner, x_1, \dots, x_n, y))$

Corollary

partial recursiveness and recursiveness coincide for total functions

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, . . .

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

Example

$$0 + y = \pi^1_1(y)$$
$$(x+1) + y = \mathsf{s}(\pi^3_1(x+y, x, y))$$

$1 + 1 = 2$

Example

$$0 + y = \pi^1_1(y)$$
$$(x+1) + y = \mathsf{s}(\pi^3_1(x+y, x, y))$$

$1 + 1 = 2$:

- $\langle \ulcorner + \urcorner, \langle 1, 1 \rangle, 2 \rangle$
  - $\langle \ulcorner + \urcorner, \langle 0, 1 \rangle, 1 \rangle$
    - $\langle \ulcorner \pi^1_1 \urcorner, \langle 1 \rangle, 1 \rangle$
  - $\langle \ulcorner \mathsf{s} \circ \pi^3_1 \urcorner, \langle 1, 0, 1 \rangle, 2 \rangle$
    - $\langle \ulcorner \pi^3_1 \urcorner, \langle 1, 0, 1 \rangle, 1 \rangle$
    - $\langle \ulcorner \mathsf{s} \urcorner, \langle 1 \rangle, 2 \rangle$

Example

$$0 + y = \pi^1_1(y)$$
$$(x+1) + y = \mathsf{s}(\pi^3_1(x+y, x, y))$$

$1 + 1 = 2$:

$\langle \ulcorner + \urcorner, \langle 1, 1 \rangle, 2 \rangle$

- $\langle \ulcorner + \urcorner, \langle 0, 1 \rangle, 1 \rangle$
  - $\langle \ulcorner \pi^1_1 \urcorner, \langle 1 \rangle, 1 \rangle$
- $\langle \ulcorner \mathsf{s} \circ \pi^3_1 \urcorner, \langle 1, 0, 1 \rangle, 2 \rangle$
  - $\langle \ulcorner \pi^3_1 \urcorner, \langle 1, 0, 1 \rangle, 1 \rangle$
  - $\langle \ulcorner \mathsf{s} \urcorner, \langle 1 \rangle, 2 \rangle$

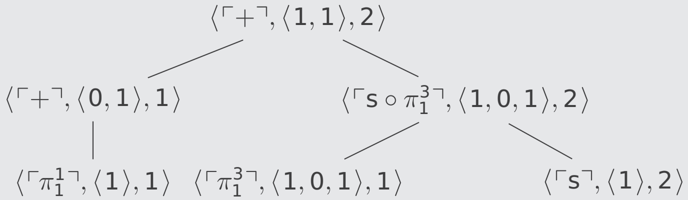
$\langle \ulcorner f \urcorner, \langle x_1, \ldots, x_n \rangle, z \rangle$ represents $f(x_1, \ldots, x_n) = z$

Example

$$0 + y = \pi_1^1(y)$$

$$(x+1) + y = \mathsf{s}(\pi_1^3(x+y, x, y))$$

$1 + 1 = 2$:



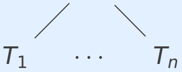
$\langle \ulcorner f \urcorner, \langle x_1, \dots, x_n \rangle, z \rangle$ represents $f(x_1, \dots, x_n) = z$

Notation

$\mathrm{triple}(x) = (\mathrm{seq}(x) \wedge \mathrm{len}(x) = 3)$

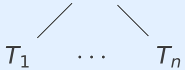
Key Idea

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



Key Idea

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



- dependencies between node $\langle x, y, z \rangle$ and root nodes of its children $T_1, \ldots, T_n$ can be checked by primitive recursive predicate

Key Idea

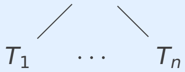
- encode computation tree $T = \langle x, y, z \rangle$ with children $T_1 \; \cdots \; T_n$ as number $\ulcorner T \urcorner = \langle \langle x, y, z \rangle, \ulcorner T_1 \urcorner, \dots, \ulcorner T_n \urcorner \rangle$
- dependencies between node $\langle x, y, z \rangle$ and root nodes of its children $T_1, \dots, T_n$ can be checked by primitive recursive predicate D

Definitions

- $\mathsf{T}(x) \iff \mathsf{D}(x) \wedge \left[\, \mathrm{len}(x) > 1 \iff (\forall\, i \leqslant \mathrm{len}(x) \dot{-} 2)\; \mathsf{T}((x)_{i+2}) \,\right]$

Key Idea

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



- dependencies between node $\langle x, y, z \rangle$ and root nodes of its children $T_1, \dots, T_n$ can be checked by primitive recursive predicate D

Definitions

- $\mathrm{T}(x) \iff \mathrm{D}(x) \wedge \left[\, \mathrm{len}(x) > 1 \iff (\forall\, i \leqslant \mathrm{len}(x) \dot{-} 2)\, \mathrm{T}((x)_{i+2}) \,\right]$
- $\mathrm{T}_n(e, \vec{x}, y) \iff \mathrm{T}(y) \wedge (y)_{1,1} = e \wedge (y)_{1,2} = \langle \vec{x} \rangle$

Key Idea

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

$$\begin{array}{ccc} & \langle x, y, z \rangle & \\ \swarrow & & \searrow \\ T_1 & \cdots & T_n \end{array}$$

- dependencies between node $\langle x, y, z \rangle$ and root nodes of its children $T_1, \dots, T_n$ can be checked by primitive recursive predicate D

Definitions

- $\mathrm{T}(x) \iff \mathrm{D}(x) \wedge \left[\, \mathrm{len}(x) > 1 \iff (\forall\, i \leqslant \mathrm{len}(x) \mathbin{\dot{-}} 2)\ \mathrm{T}((x)_{i+2}) \,\right]$
- $\mathrm{T}_n(e, \vec{x}, y) \iff \mathrm{T}(y) \wedge (y)_{1,1} = e \wedge (y)_{1,2} = \langle \vec{x} \rangle$
- $\mathrm{u}(x) = (x)_{1,3}$

Key Idea

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

  $T_1 \quad \cdots \quad T_n$

- dependencies between node $\langle x, y, z \rangle$ and root nodes of its children $T_1, \dots, T_n$ can be checked by primitive recursive predicate D

Definitions

- $\mathsf{T}(x) \iff \mathsf{D}(x) \wedge \big[\, \mathrm{len}(x) > 1 \iff (\forall\, i \leqslant \mathrm{len}(x) \mathbin{\dot-} 2)\, \mathsf{T}((x)_{i+2}) \,\big]$
- $\mathsf{T}_n(e, \vec{x}, y) \iff \mathsf{T}(y) \wedge (y)_{1,1} = e \wedge (y)_{1,2} = \langle \vec{x} \rangle$
- $\mathsf{u}(x) = (x)_{1,3}$

Theorem

$\varphi(x_1, \dots, x_n) \simeq \mathsf{u}((\mu\, y)\, \mathsf{T}_n(\ulcorner \varphi \urcorner, x_1, \dots, x_n, y))$

Definition of D (cont'd)

$$\text{initial}(x) \iff \text{seq}(x) \wedge \text{len}(x) = 1 \wedge \text{triple}((x)_1) \wedge \text{seq}((x)_{1,1}) \wedge \text{seq}((x)_{1,2}) \wedge \Big[\, \text{zero}((x)_1) \vee \text{successor}((x)_1) \vee \text{projection}((x)_1) \,\Big]$$

Definition of D (cont'd)

- zero $\langle\langle\langle 0\rangle, \langle x\rangle, 0\rangle\rangle$

$$\begin{aligned}\mathrm{initial}(x) \iff{} & \mathrm{seq}(x) \wedge \mathrm{len}(x) = 1 \wedge \mathrm{triple}((x)_1) \wedge \mathrm{seq}((x)_{1,1}) \wedge \mathrm{seq}((x)_{1,2}) \wedge \\ & \big[\, \mathrm{zero}((x)_1) \vee \mathrm{successor}((x)_1) \vee \mathrm{projection}((x)_1) \,\big]\end{aligned}$$

with

$$\mathrm{zero}(x) \iff (x)_1 = \langle 0\rangle \wedge \mathrm{len}((x)_2) = 1 \wedge (x)_3 = 0$$

Definition of D (cont'd)

- zero $\langle\langle\langle 0\rangle, \langle x\rangle, 0\rangle\rangle$
- successor $\langle\langle\langle 1\rangle, \langle x\rangle, x+1\rangle\rangle$

$$\text{initial}(x) \iff \text{seq}(x) \wedge \text{len}(x) = 1 \wedge \text{triple}((x)_1) \wedge \text{seq}((x)_{1,1}) \wedge \text{seq}((x)_{1,2}) \wedge \Big[\, \text{zero}((x)_1) \vee \text{successor}((x)_1) \vee \text{projection}((x)_1) \,\Big]$$

with

$$\text{zero}(x) \iff (x)_1 = \langle 0\rangle \wedge \text{len}((x)_2) = 1 \wedge (x)_3 = 0$$

$$\text{successor}(x) \iff (x)_1 = \langle 1\rangle \wedge \text{len}((x)_2) = 1 \wedge (x)_3 = \text{s}((x)_{2,1})$$

Definition of D (cont'd)

- zero $\langle\langle\langle 0\rangle, \langle x\rangle, 0\rangle\rangle$
- successor $\langle\langle\langle 1\rangle, \langle x\rangle, x+1\rangle\rangle$
- projection $\langle\langle\langle 2, n, i\rangle, \langle x_1, \ldots, x_n\rangle, x_i\rangle\rangle$

$$\begin{aligned} \text{initial}(x) \iff\ & \text{seq}(x) \wedge \text{len}(x) = 1 \wedge \text{triple}((x)_1) \wedge \text{seq}((x)_{1,1}) \wedge \text{seq}((x)_{1,2}) \wedge \\ & \big[\, \text{zero}((x)_1) \vee \text{successor}((x)_1) \vee \text{projection}((x)_1) \,\big] \end{aligned}$$

with

$$\begin{aligned} \text{zero}(x) &\iff (x)_1 = \langle 0\rangle \wedge \text{len}((x)_2) = 1 \wedge (x)_3 = 0 \\ \text{successor}(x) &\iff (x)_1 = \langle 1\rangle \wedge \text{len}((x)_2) = 1 \wedge (x)_3 = \text{s}((x)_{2,1}) \\ \text{projection}(x) &\iff \text{len}((x)_1) = 3 \wedge (x)_{1,1} = 2 \wedge (x)_{1,2} = \text{len}((x)_2) \wedge (x)_{1,3} \geqslant 1 \wedge \\ & \qquad (x)_{1,2} \geqslant (x)_{1,3} \wedge (x)_3 = (x)_{2,(x)_{1,3}} \end{aligned}$$

Definition of D (cont'd)

- composition $f(x_1, \ldots, x_n) = g(h_1(x_1, \ldots, x_n), \ldots, h_m(x_1, \ldots, x_n))$

$$\langle \langle \ulcorner f \urcorner, \langle x_1, \ldots, x_n \rangle, z \rangle, T_1, \ldots, T_m, T_{m+1} \rangle$$

  with

$$\ulcorner f \urcorner = \langle 3, \ulcorner g \urcorner, \ulcorner h_1 \urcorner, \ldots, \ulcorner h_m \urcorner \rangle$$

$$T_1 = \langle \langle \ulcorner h_1 \urcorner, \langle x_1, \ldots, x_n \rangle, z_1 \rangle, \ldots \rangle \quad \cdots \quad T_m = \langle \langle \ulcorner h_m \urcorner, \langle x_1, \ldots, x_n \rangle, z_m \rangle, \ldots \rangle$$

$$T_{m+1} = \langle \langle \ulcorner g \urcorner, \langle z_1, \ldots, z_m \rangle, z \rangle, \ldots \rangle$$

Definition of D (cont'd)

- composition $f(x_1,\dots,x_n) = g(h_1(x_1,\dots,x_n),\dots,h_m(x_1,\dots,x_n))$

$$\langle\langle\ulcorner f\urcorner,\langle x_1,\dots,x_n\rangle,z\rangle,T_1,\dots,T_m,T_{m+1}\rangle$$

with

$$\begin{aligned}\ulcorner f\urcorner &= \langle 3,\ulcorner g\urcorner,\ulcorner h_1\urcorner,\dots,\ulcorner h_m\urcorner\rangle\\ T_1 &= \langle\langle\ulcorner h_1\urcorner,\langle x_1,\dots,x_n\rangle,z_1\rangle,\dots\rangle \quad\cdots\quad T_m = \langle\langle\ulcorner h_m\urcorner,\langle x_1,\dots,x_n\rangle,z_m\rangle,\dots\rangle\\ T_{m+1} &= \langle\langle\ulcorner g\urcorner,\langle z_1,\dots,z_m\rangle,z\rangle,\dots\rangle\end{aligned}$$

$$\begin{aligned}\text{composition}(x) \iff{}& \text{seq}(x) \wedge \text{len}(x) \geqslant 3 \wedge \text{triple}((x)_1) \wedge \text{seq}((x)_{1,1}) \wedge (x)_{1,1,1} = 3 \wedge\\ & \text{len}((x)_{1,1}) = \text{len}(x) \wedge (\forall\, i \leqslant \text{len}(x) \dot{-} 3)\,\big[\,(x)_{i+2,1,1} = (x)_{1,1,i+3} \wedge\\ & (x)_{i+2,1,2} = (x)_{1,2} \wedge (x)_{i+2,1,3} = (x)_{\text{len}(x),1,2,i+1}\,\big] \wedge\\ & (x)_{\text{len}(x),1,1} = (x)_{1,1,2} \wedge (x)_{\text{len}(x),1,3} = (x)_{1,3}\end{aligned}$$

Definition of D (cont'd)

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

$$\langle\langle\ulcorner f\urcorner,\langle 0,y_1,\dots,y_n\rangle,z\rangle,\langle\langle\ulcorner g\urcorner,\langle y_1,\dots,y_n\rangle,z\rangle,\dots\rangle\rangle \quad \langle\langle\ulcorner f\urcorner,\langle x+1,y_1,\dots,y_n\rangle,z\rangle,T_1,T_2\rangle$$

with

$$\ulcorner f\urcorner = \langle 4,\ulcorner g\urcorner,\ulcorner h\urcorner\rangle$$

$$T_1 = \langle\langle\ulcorner f\urcorner,\langle x,y_1,\dots,y_n\rangle,z_1\rangle,\cdots\rangle \qquad T_2 = \langle\langle\ulcorner h\urcorner,\langle z_1,x,y_1,\dots,y_n\rangle,z\rangle,\cdots\rangle$$

Definition of D (cont'd)

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

$$\langle\langle\ulcorner f\urcorner,\langle 0,y_1,\ldots,y_n\rangle,z\rangle,\langle\langle\ulcorner g\urcorner,\langle y_1,\ldots,y_n\rangle,z\rangle,\ldots\rangle\rangle \quad \langle\langle\ulcorner f\urcorner,\langle x+1,y_1,\ldots,y_n\rangle,z\rangle,T_1,T_2\rangle$$

with

$$\ulcorner f\urcorner = \langle 4,\ulcorner g\urcorner,\ulcorner h\urcorner\rangle$$

$$T_1 = \langle\langle\ulcorner f\urcorner,\langle x,y_1,\ldots,y_n\rangle,z_1\rangle,\cdots\rangle \qquad T_2 = \langle\langle\ulcorner h\urcorner,\langle z_1,x,y_1,\ldots,y_n\rangle,z\rangle,\cdots\rangle$$

$$\begin{aligned}\text{recursion}(x) \iff{}& \text{seq}(x) \land \text{triple}((x)_1) \land \text{triple}((x)_{1,1}) \land (x)_{1,1,1} = 4 \land \text{seq}((x)_{1,2}) \land \\ & \big[(x)_{1,2,1} = 0 \implies \text{len}(x) = 2 \land (x)_{1,1,2} = (x)_{2,1,1} \land \\ & (x)_{1,2} = \langle 0\rangle ; (x)_{2,1,2} \land (x)_{1,3} = (x)_{2,1,3}\big] \land \big[(x)_{1,2,1} > 0 \implies \\ & \text{len}(x) = 3 \land (x)_{2,1,1} = (x)_{1,1} \land (x)_{2,1,2,1} = \text{p}((x)_{1,2,1}) \land \\ & (\forall\, i \leqslant \text{len}((x)_{1,2}) \dot{-} 2)\,\big[(x)_{1,2,i+2} = (x)_{2,1,2,i+2}\big] \land (x)_{3,1,1} = (x)_{1,1,3} \land \\ & (x)_{3,1,2} = \langle (x)_{2,1,3}\rangle ; (x)_{2,1,2} \land (x)_{3,1,3} = (x)_{1,3}\big]\end{aligned}$$

Definition of D (cont'd)

- minimization $f(y_1, \ldots, y_n) = (\mu\, x)\,(g(x, y_1, \ldots, y_n) = 0)$

  minimization(x) $\iff$ $\cdots$ (homework exercise)

Definition of D (cont'd)

- minimization $f(y_1, \ldots, y_n) = (\mu\, x)\, (g(x, y_1, \ldots, y_n) = 0)$

  minimization$(x) \iff \cdots$ (homework exercise)
- $\mathrm{D}(x) \iff$ initial(x) $\vee$ composition(x) $\vee$ recursion(x) $\vee$ minimization(x)

Definition of D (cont'd)

- minimization $\quad f(y_1, \ldots, y_n) = (\mu\, x)\,(g(x, y_1, \ldots, y_n) = 0)$

  minimization$(x) \iff \cdots$ (homework exercise)

- $\mathrm{D}(x) \iff$ initial(x) $\vee$ composition(x) $\vee$ recursion(x) $\vee$ minimization(x)

Kleene's Normal Form Theorem

$\exists$ primitive recursive function u $\;\forall\, n \geqslant 1\;$ $\exists$ primitive recursive predicate T_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$

$$\varphi(x_1, \ldots, x_n) \simeq \mathrm{u}((\mu\, y)\, \mathrm{T}_n(\ulcorner \varphi \urcorner, x_1, \ldots, x_n, y))$$

Corollary

$\exists$ primitive recursive function u $\forall n \geqslant 1$ $\exists$ primitive recursive function t_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1,\dots,x_n) \simeq \mathrm{u}((\mu\, y)\,(\mathrm{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Corollary

$\exists$ primitive recursive function u $\forall\, n \geqslant 1$ $\exists$ primitive recursive function t_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1,\dots,x_n) \simeq \mathsf{u}((\mu\, y)\ (\mathsf{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Proof

define $\mathsf{t}_n = \chi_{\neg \mathsf{T}_n}$

Corollary

$\exists$ primitive recursive function u $\forall\, n \geqslant 1$ $\exists$ primitive recursive function t_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1,\dots,x_n) \simeq \mathrm{u}((\mu\, y)\; (\mathrm{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Proof

define $\mathrm{t}_n = \chi_{\neg \mathrm{T}_n}$

Definition

$\varphi_e^n(x_1,\dots,x_n) = \mathrm{u}((\mu\, y)\; (\mathrm{t}_n(e, x_1,\dots,x_n,y) = 0))$

Corollary

$\exists$ primitive recursive function u $\forall\, n \geqslant 1$ $\exists$ primitive recursive function t_n
$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\varphi(x_1,\dots,x_n) \simeq \mathrm{u}((\mu\, y)\,(\mathrm{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Proof

define $\mathrm{t}_n = \chi_{\neg \mathrm{T}_n}$

Definition

$\varphi_e^n(x_1,\dots,x_n) = \mathrm{u}((\mu\, y)\,(\mathrm{t}_n(e, x_1,\dots,x_n,y) = 0))$

Lemma

- $\forall\, e \geqslant 0$ $\forall\, n \geqslant 1$ φ_e^n is partial recursive

Corollary

$\exists$ primitive recursive function u $\forall\, n \geqslant 1$ $\exists$ primitive recursive function t_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1,\dots,x_n) \simeq \mathrm{u}((\mu\, y)\,(\mathrm{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Proof

define $\mathrm{t}_n = \chi_{\neg \mathrm{T}_n}$

Definition

$$\varphi_e^n(x_1,\dots,x_n) = \mathrm{u}((\mu\, y)\,(\mathrm{t}_n(e,x_1,\dots,x_n,y) = 0))$$

Lemma

- $\forall\, e \geqslant 0$ $\forall\, n \geqslant 1$ φ_e^n is partial recursive
- e is no index $\implies$ φ_e^n is nowhere defined

Corollary

$\exists$ primitive recursive function u $\forall\, n \geqslant 1$ $\exists$ primitive recursive function t_n

$\forall$ partial recursive function $\varphi\colon \mathbb{N}^n \to \mathbb{N}$ $\quad \varphi(x_1,\dots,x_n) \simeq \mathsf{u}((\mu\, y)\ (\mathsf{t}_n(\ulcorner\varphi\urcorner, x_1,\dots,x_n,y) = 0))$

Proof

define $\mathsf{t}_n = \chi_{\neg\mathsf{T}_n}$

Definition

$$\varphi_e^n(x_1,\dots,x_n) = \mathsf{u}((\mu\, y)\ (\mathsf{t}_n(e, x_1,\dots,x_n,y) = 0))$$

Lemma

- $\forall\, e \geqslant 0 \ \ \forall\, n \geqslant 1 \ \ \varphi_e^n$ is partial recursive
- e is no index $\implies$ φ_e^n is nowhere defined
- $\varphi_0^n, \varphi_1^n, \varphi_2^n, \dots$ is enumeration of all n-ary partial recursive functions

Outline

Definition

predicate $P\colon \mathbb{N}^n \to \mathbb{B}$ is **decidable** if χ_P is recursive

Definition

predicate $P\colon \mathbb{N}^n \to \mathbb{B}$ is decidable if χ_P is recursive

Notation

φ_e denotes φ_e^1

Definition

predicate $P\colon \mathbb{N}^n \to \mathbb{B}$ is decidable if χ_P is recursive

Notation

φ_e denotes φ_e^1

Theorem

following problem is undecidable:

instance: natural number x

question: is $\varphi_x(x)$ defined ?

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$ is recursive

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$ is recursive
- function $g(x) = (\mu\, i)\,(f(x) = 0)$ is partial recursive

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x){\downarrow} \\ 0 & \text{if } \varphi_x(x){\uparrow} \end{cases}$ is recursive
- function $g(x) = (\mu\, i)\,(f(x) = 0)$ is partial recursive
- $\exists\, e$ such that $g = \varphi_e$

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$ is recursive
- function $g(x) = (\mu\, i)\,(f(x) = 0)$ is partial recursive
- $\exists\, e$ such that $g = \varphi_e$
- $g(e)\downarrow$

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$ is recursive
- function $g(x) = (\mu\, i)\,(f(x) = 0)$ is partial recursive
- $\exists\, e$ such that $g = \varphi_e$
- $g(e)\downarrow \iff f(e) = 0$

Proof (by contradiction)

- suppose $f(x) = \begin{cases} 1 & \text{if } \varphi_x(x)\downarrow \\ 0 & \text{if } \varphi_x(x)\uparrow \end{cases}$ is recursive
- function $g(x) = (\mu\, i)\,(f(x) = 0)$ is partial recursive
- $\exists\, e$ such that $g = \varphi_e$
- $g(e)\downarrow \iff f(e) = 0 \iff \varphi_e(e)\uparrow$ ↯

Outline

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$
- $2 + y$ is partial recursive $\implies$ $\exists\, e'$ such that $2 + y = \varphi_{e'}(y)$

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$
- $2 + y$ is partial recursive $\implies$ $\exists\, e'$ such that $2 + y = \varphi_{e'}(y)$
- claim: e' can be computed from e and 2 by primitive recursion

Example

- $x+y$ is partial recursive $\Longrightarrow$ $\exists\, e$ such that $x+y = \varphi_e^2(x,y)$
- $2+y$ is partial recursive $\Longrightarrow$ $\exists\, e'$ such that $2+y = \varphi_{e'}(y)$
- claim: e' can be computed from e and 2 by primitive recursion

Kleene's s–m–n or Parameterization Theorem

$\forall\, m,n \geqslant 1$ $\exists$ primitive recursive function $s_n^m \colon \mathbb{N}^{m+1} \to \mathbb{N}$ $\forall\, e \in \mathbb{N}$

$$\varphi_e^{m+n}(x_1,\ldots,x_m,y_1,\ldots,y_n) \simeq \varphi^n_{s_n^m(e,x_1,\ldots,x_m)}(y_1,\ldots,y_n)$$

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$
- $2 + y$ is partial recursive $\implies$ $\exists\, e'$ such that $2 + y = \varphi_{e'}(y)$
- claim: e' can be computed from e and 2 by primitive recursion

Kleene's s–m–n or Parameterization Theorem

$\forall\, m, n \geqslant 1$ $\exists$ primitive recursive function $s_n^m \colon \mathbb{N}^{m+1} \to \mathbb{N}$ $\forall\, e \in \mathbb{N}$

$$\varphi_e^{m+n}(x_1, \dots, x_m, y_1, \dots, y_n) \simeq \varphi^n_{s_n^m(e, x_1, \dots, x_m)}(y_1, \dots, y_n)$$

Proof

- primitive recursive function $f \colon \mathbb{N} \to \mathbb{N}$ computes index of n-ary constant function c_x

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$
- $2 + y$ is partial recursive $\implies$ $\exists\, e'$ such that $2 + y = \varphi_{e'}(y)$
- claim: e' can be computed from e and 2 by primitive recursion

Kleene's s–m–n or Parameterization Theorem

$\forall\, m, n \geqslant 1$ $\exists$ primitive recursive function $s_n^m \colon \mathbb{N}^{m+1} \to \mathbb{N}$ $\forall\, e \in \mathbb{N}$

$$\varphi_e^{m+n}(x_1, \dots, x_m, y_1, \dots, y_n) \simeq \varphi^n_{s_n^m(e, x_1, \dots, x_m)}(y_1, \dots, y_n)$$

Proof

- primitive recursive function $f \colon \mathbb{N} \to \mathbb{N}$ computes index of n-ary constant function c_x:

$$f(0) = \langle 3, \langle 0 \rangle, \langle 2, n, 1 \rangle \rangle \qquad f(x+1) = \ulcorner s \circ c_x \urcorner = \langle 3, \langle 1 \rangle, f(x) \rangle$$

Example

- $x + y$ is partial recursive $\implies$ $\exists\, e$ such that $x + y = \varphi_e^2(x, y)$
- $2 + y$ is partial recursive $\implies$ $\exists\, e'$ such that $2 + y = \varphi_{e'}(y)$
- claim: e' can be computed from e and 2 by primitive recursion

Kleene's s–m–n or Parameterization Theorem

$\forall\, m, n \geqslant 1$ $\exists$ primitive recursive function $\mathrm{s}_n^m \colon \mathbb{N}^{m+1} \to \mathbb{N}$ $\forall\, e \in \mathbb{N}$

$$\varphi_e^{m+n}(x_1, \dots, x_m, y_1, \dots, y_n) \simeq \varphi^n_{\mathrm{s}_n^m(e, x_1, \dots, x_m)}(y_1, \dots, y_n)$$

Proof

- primitive recursive function $f \colon \mathbb{N} \to \mathbb{N}$ computes index of n-ary constant function c_x:

$$f(0) = \langle 3, \langle 0 \rangle, \langle 2, n, 1 \rangle \rangle \qquad f(x + 1) = \ulcorner \mathrm{s} \circ \mathrm{c}_x \urcorner = \langle 3, \langle 1 \rangle, f(x) \rangle$$

- $\mathrm{s}_n^m(e, x_1, \dots, x_m) = \langle 3, e, f(x_1), \dots, f(x_m), \langle 2, n, 1 \rangle, \dots, \langle 2, n, n \rangle \rangle$

Outline

Kleene's Fixed Point Theorem

$\forall$ recursive function $f : \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f \colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x, y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = \mathrm{s}^1_1(\ulcorner \psi \urcorner, x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f \colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\implies$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\implies$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\implies$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\Longrightarrow$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\Longrightarrow$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner \psi \urcorner, x)$
- s–m–n theorem $\Longrightarrow$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\Longrightarrow$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$: $\varphi_e(x) = \varphi_{g(d)}(x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = \mathrm{s}^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\implies$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\implies$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$: $\varphi_e(x) = \varphi_{g(d)}(x) \simeq \psi(d,x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x)\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\implies$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\implies$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$: $\varphi_e(x) = \varphi_{g(d)}(x) \simeq \psi(d,x) \simeq \varphi_{\varphi_d(d)}(x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = s^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\Longrightarrow$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\Longrightarrow$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$: $\varphi_e(x) = \varphi_{g(d)}(x) \simeq \psi(d,x) \simeq \varphi_{\varphi_d(d)}(x) = \varphi_{f(g(d))}(x)$

Kleene's Fixed Point Theorem

$\forall$ recursive function $f\colon \mathbb{N} \to \mathbb{N}$ $\exists\, e \in \mathbb{N}$ such that $\varphi_e(x) \simeq \varphi_{f(e)}(x)$

Proof

- $\psi(x,y) = \begin{cases} \varphi_{\varphi_x(x)}(y) & \text{if } \varphi_x(x){\downarrow} \\ \text{undefined} & \text{otherwise} \end{cases}$ is partial recursive
- define $g(x) = \mathrm{s}^1_1(\ulcorner\psi\urcorner, x)$
- s–m–n theorem $\implies$ $\psi(x,y) \simeq \varphi_{g(x)}(y)$
- $f \circ g$ is recursive $\implies$ $\exists\, d$ such that $f(g(x)) = \varphi_d(x)$
- take $e = g(d)$: $\varphi_e(x) = \varphi_{g(d)}(x) \simeq \psi(d,x) \simeq \varphi_{\varphi_d(d)}(x) = \varphi_{f(g(d))}(x) = \varphi_{f(e)}(x)$

Example

construct partial recursive function that evaluates to its own index

Example

construct partial recursive function that evaluates to its own index

- π_1^2 is partial recursive $\implies$ $\exists\, n$ such that $\pi_1^2(x, y) = \varphi_n^2(x, y)$

Example

construct partial recursive function that evaluates to its own index

- π_1^2 is partial recursive $\implies$ $\exists\, n$ such that $\pi_1^2(x,y) = \varphi_n^2(x,y)$
- s–m–n theorem $\implies$ $\varphi_n^2(x,y) = \varphi_{g(x)}(y)$ for $g(x) = s_1^1(n,x)$

Example

construct partial recursive function that evaluates to its own index

- π^2_1 is partial recursive $\implies$ $\exists\, n$ such that $\pi^2_1(x,y) = \varphi^2_n(x,y)$
- s–m–n theorem $\implies$ $\varphi^2_n(x,y) = \varphi_{g(x)}(y)$ for $g(x) = s^1_1(n,x)$
- fixed point theorem $\implies$ $\exists\, m$ such that $\varphi_m(x) \simeq \varphi_{g(m)}(x)$

Example

construct partial recursive function that evaluates to its own index

- π_1^2 is partial recursive $\implies$ $\exists\, n$ such that $\pi_1^2(x,y) = \varphi_n^2(x,y)$
- s–m–n theorem $\implies$ $\varphi_n^2(x,y) = \varphi_{g(x)}(y)$ for $g(x) = \mathrm{s}_1^1(n,x)$
- fixed point theorem $\implies$ $\exists\, m$ such that $\varphi_m(x) \simeq \varphi_{g(m)}(x)$
- $\varphi_m(x) \simeq \varphi_{\mathrm{s}_1^1(n,m)}(x)$

Example

construct partial recursive function that evaluates to its own index

- π^2_1 is partial recursive $\implies$ $\exists\, n$ such that $\pi^2_1(x,y) = \varphi^2_n(x,y)$
- s–m–n theorem $\implies$ $\varphi^2_n(x,y) = \varphi_{g(x)}(y)$ for $g(x) = \mathrm{s}^1_1(n,x)$
- fixed point theorem $\implies$ $\exists\, m$ such that $\varphi_m(x) \simeq \varphi_{g(m)}(x)$
- $\varphi_m(x) \simeq \varphi_{\mathrm{s}^1_1(n,m)}(x) = \varphi^2_n(m,x)$

Example

construct partial recursive function that evaluates to its own index

- π^2_1 is partial recursive $\implies$ $\exists\, n$ such that $\pi^2_1(x,y) = \varphi^2_n(x,y)$
- s–m–n theorem $\implies$ $\varphi^2_n(x,y) = \varphi_{g(x)}(y)$ for $g(x) = \mathrm{s}^1_1(n,x)$
- fixed point theorem $\implies$ $\exists\, m$ such that $\varphi_m(x) \simeq \varphi_{g(m)}(x)$
- $\varphi_m(x) \simeq \varphi_{\mathrm{s}^1_1(n,m)}(x) = \varphi^2_n(m,x) = \pi^2_1(m,x) = m$

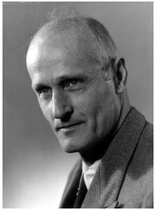
Outline

Important Concepts

- decidable predicate
- φ_e
- φ_e^n
- Kleene's fixed point theorem
- Kleene's normal form theorem
- Kleene's s–m–n theorem
- s_n^m
- T_n
- t_n
- u

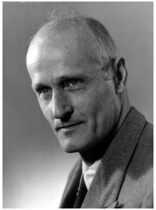
Important Concepts

- decidable predicate
- φ_e
- φ_e^n
- Kleene's fixed point theorem
- Kleene's normal form theorem
- Kleene's s–m–n theorem
- s_n^m
- T_n
- t_n
- u



Important Concepts

- decidable predicate
- φ_e
- φ_e^n
- Kleene's fixed point theorem
- Kleene's normal form theorem
- Kleene's s–m–n theorem
- s_n^m
- T_n
- t_n
- u



homework for November 6