universität innsbruck

WS 2023 lecture 11

Computability Theory

Aart Middeldorp

Outline

Definitions

- Gödel number of CL–term is defined inductively:

$$\mathfrak{g}(\mathsf{I}) = \langle 0 \rangle \qquad \mathfrak{g}(\mathsf{K}) = \langle 1 \rangle \qquad \mathfrak{g}(\mathsf{S}) = \langle 2 \rangle \qquad \mathfrak{g}(t\,u) = \langle 3, \mathfrak{g}(t), \mathfrak{g}(u) \rangle \qquad \mathfrak{g}(x_i) = \langle 4, i \rangle$$

- $\mathsf{enc}(n) = \underline{\mathfrak{g}(\underline{n})}$
- sets T and U of CL–terms are recursively separable if $\{\mathfrak{g}(t) \mid t \in T\}$ and $\{\mathfrak{g}(u) \mid u \in U\}$ are recursively separable
- set T of CL–terms is conversion–closed if $u \in T$ whenever $t \in T$ and $t \leftrightarrow^* u$

Theorem

CL–representable functions are partial recursive

Notation

$\ulcorner t \urcorner = \underline{\mathfrak{g}(t)}$ is Church numeral of Gödel number of CL–term t

Theorem

$\forall$ CL–term F $\exists$ CL–term X such that $F \ulcorner X \urcorner \leftrightarrow^* X$

Theorem

- non-empty conversion–closed sets of CL–terms are recursively inseparable
- non-trivial conversion–closed sets of CL–terms are not recursive

Corollary

set of normalizing CL–terms is not recursive: decision problem

instance: CL–term t
question: is t normalizing ?

is undecidable

Definition (Types)

set $\mathbb{T}$ of types is defined inductively:

- $\mathbb{V} \subseteq \mathbb{T}$ infinite set of type variables
- $\mathbb{C} \subseteq \mathbb{T}$ type constants
- if $\sigma, \tau \in \mathbb{T}$ then $(\sigma \to \tau) \in \mathbb{T}$

Definition (Type Assignment, Curry-style)

- type assignment formula $t : \tau$ with CL–term t and type τ
- type assignment system TA

$$\frac{}{\mathsf{I} : \sigma \to \sigma} \qquad \frac{}{\mathsf{K} : \sigma \to \tau \to \sigma} \qquad \frac{}{\mathsf{S} : (\rho \to \sigma \to \tau) \to (\rho \to \sigma) \to \rho \to \tau}$$

$$\frac{t : \sigma \to \tau \qquad u : \sigma}{t\,u : \tau}$$

for all types σ, τ, ρ and CL–terms s and t

Notation

$\Gamma \vdash t : \tau$ if $t : \tau$ can be derived in TA from assumptions in Γ

Theorem

if $\Gamma, x : \sigma \vdash t : \tau$ and $x \notin \mathcal{V}\mathrm{ar}(\Gamma)$ then $\Gamma \vdash [x]t : \sigma \to \tau$

Theorem (Subject Reduction)

if $\Gamma \vdash t : \tau$ and $t \to u$ then $\Gamma \vdash u : \tau$

Definition

CL–term t with $\mathcal{V}\mathrm{ar}(t) = \{x_1, \dots, x_n\}$ is typable if

$$x_1 : \rho_1, \dots, x_n : \rho_n \vdash t : \tau$$

for some types $\rho_1, \dots, \rho_n, \tau$

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

Theorem (Strong Normalization)

typable CL–terms are terminating (SN)

Definition

typable CL–term t is strongly computable (SC) if

- t has atomic type $\tau \in \mathbb{V} \cup \mathbb{C}$ and is SN
- t has type $\sigma \to \tau$ and $t\,u$ is SC whenever $u : \sigma$ is SC

Lemma ❶

for any type τ

❶ every term $x\,u_1 \cdots u_n$ of type τ with variable x and SN terms $u_1, \dots, u_n$ is SC

❷ every SC term of type τ is SN

Proof

induction on τ

- base case: τ is atomic
  - ① $t = x\,u_1 \cdots u_n$ is SN $\implies$ t is SC
  - ② $t : \tau$ is SC $\implies$ t is SN
- step case: $\tau = \rho \to \sigma$
  - ① $v : \rho$ is SC $\implies$ v is SN $\implies$ $x\,u_1 \cdots u_n\,v : \sigma$ is SC $\implies$ $x\,u_1 \cdots u_n : \tau$ is SC
  - ② consider SC term $t : \tau$ and $x : \rho$ with $x \in \mathcal{V} \setminus \mathcal{V}\mathrm{ar}(t)$

    $\implies$ x is SC $\implies$ $t\,x$ is SC $\implies$ $t\,x$ is SN $\implies$ t is SN

Lemma ❷

S, K and I are SC

Proof

$\mathsf{K} : \sigma \to \tau \to \sigma$

- $\sigma = \sigma_1 \to \cdots \to \sigma_n \to \theta$ with atomic type θ and $n \geqslant 0$
- consider arbitrary SC terms $s : \sigma \quad t : \tau \quad u_1 : \sigma_1 \quad \dots \quad u_n : \sigma_n$
- s is SC $\implies$ $s\,u_1 \cdots u_n$ is SC $\implies$ $s\,u_1 \cdots u_n$ is SN
- t is SC $\implies$ t is SN
- $\mathsf{K}\,s\,t\,u_1 \cdots u_n$ is SN: any infinite reduction starts with

$$\begin{aligned} \mathsf{K}\,s\,t\,u_1 \dots u_n \to^* \mathsf{K}\,s'\,t'\,u_1' \dots u_n' &\to s'\,u_1' \dots u_n' \to^* \cdots \\ \implies \quad s\,u_1 \dots u_n &\to s'\,u_1' \dots u_n' \to^* \cdots \end{aligned} \qquad ↯$$

with $s \to^* s' \quad t \to^* t' \quad u_1 \to^* u_1' \quad u_n \to^* u_n'$

S and I: homework exercise

Lemma

every typable CL–term is SC

Proof

induction on term t

- t is variable $\implies$ t is SC by lemma (1)
- $t \in \{I, K, S\}$ $\implies$ t is SC by lemma (2)
- $t = t_1 t_2$ $\implies$ t_1 and t_2 are SC by induction hypothesis $\implies$ t is SC

Corollary

typable CL–terms are SN

Outline

Theorem

problem

instance: CL–term t

question: is t typable ?

is decidable

Definition

principal type of combinator t is any type σ such that

① $\vdash t : \sigma$

② if $\vdash t : \tau$ then τ is substitution instance of σ

Example

SKK has principal type $a \to a$ (where a is type variable)

Theorem

every typable combinator has principal type

Type Inference

principle types can be computed by typing rules of TA (with type variables σ, τ, ρ)

$$\frac{}{\mathsf{I} : \sigma \to \sigma} \qquad \frac{}{\mathsf{K} : \sigma \to \tau \to \sigma} \qquad \frac{}{\mathsf{S} : (\rho \to \sigma \to \tau) \to (\rho \to \sigma) \to \rho \to \tau}$$

$$\frac{t : \sigma \to \tau \qquad u : \sigma}{t\,u : \tau}$$

and unification algorithm

Example ❶

principle type of SKK : $\sigma_1 \to \sigma_1$

- SK : $\alpha \to \beta$ K : α
- S : $\gamma \to \delta$ K : γ
- unification problem

$$\begin{array}{llll} \alpha \approx \sigma_1 \to \tau_1 \to \sigma_1 & \delta \approx \alpha \to \beta & \gamma \to \delta \approx (\rho_2 \to \sigma_2 \to \tau_2) \to (\rho_2 \to \sigma_2) \to \rho_2 \to \tau_2 & \\ \gamma \approx \sigma_3 \to \tau_3 \to \sigma_3 & \gamma \approx \rho_2 \to \sigma_2 \to \tau_2 & \delta \approx (\rho_2 \to \sigma_2) \to \rho_2 \to \tau_2 & \\ \alpha \approx \rho_2 \to \sigma_2 & \beta \approx \rho_2 \to \tau_2 & \rho_2 \approx \sigma_1 & \sigma_2 \approx \tau_1 \to \sigma_1 \approx \tau_3 \\ \sigma_3 \approx \rho_2 \approx \sigma_1 \approx \tau_2 & \tau_3 \approx \sigma_2 & \sigma_3 \approx \tau_2 & \end{array}$$

- mgu

$$\left\{ \begin{array}{llll} \alpha \mapsto \sigma_1 \to \tau_1 \to \sigma_1 & \gamma \mapsto \sigma_1 \to (\tau_1 \to \sigma_1) \to \sigma_1 \\ \beta \mapsto \sigma_1 \to \sigma_1 & \delta \mapsto (\sigma_1 \to \tau_1 \to \sigma_1) \to \sigma_1 \to \sigma_1 \end{array} \right\}$$

Example ❷

SII cannot be typed

- $\mathsf{SI} : \alpha \to \beta \quad \mathsf{I} : \alpha$
- $\mathsf{S} : \gamma \to \delta \quad \mathsf{I} : \gamma$
- unification problem

$$\alpha \approx \sigma_1 \to \sigma_1 \qquad \delta \approx \alpha \to \beta \qquad \gamma \to \delta \approx (\rho_2 \to \sigma_2 \to \tau_2) \to (\rho_2 \to \sigma_2) \to \rho_2 \to \tau_2$$

$$\gamma \approx \sigma_3 \to \sigma_3 \qquad \gamma \approx \rho_2 \to \sigma_2 \to \tau_2 \qquad \delta \approx (\rho_2 \to \sigma_2) \to \rho_2 \to \tau_2$$

$$\sigma_3 \approx \rho_2 \approx \sigma_2 \to \tau_2 \approx \sigma_1 \approx \sigma_2$$

$$\alpha \approx \rho_2 \to \sigma_2 \qquad \beta \approx \rho_2 \to \tau_2 \qquad \sigma_1 \approx \rho_2 \approx \sigma_2$$

has no solution

Example ❸

principle type of B = S(KS)K : $(\sigma_4 \to \tau_4) \to (\rho_4 \to \sigma_4) \to \rho_4 \to \tau_4$

- S(KS) : $\alpha \to \beta$ K : α
- S : $\gamma \to \delta$ KS : γ
- K : $\epsilon \to \eta$ S : ϵ
- unification problem

$$\alpha \approx \sigma_1 \to \tau_1 \to \sigma_1 \qquad \delta \approx \alpha \to \beta \qquad \gamma \to \delta \approx (\rho_2 \to \sigma_2 \to \tau_2) \to (\rho_2 \to \sigma_2) \to \rho_2 \to \tau_2$$

$$\gamma \approx \eta \qquad \epsilon \to \eta \approx \sigma_3 \to \tau_3 \to \sigma_3 \qquad \epsilon \approx (\rho_4 \to \sigma_4 \to \tau_4) \to (\rho_4 \to \sigma_4) \to \rho_4 \to \tau_4$$

- mgu

$$\{ \ \beta \mapsto (\sigma_4 \to \tau_4) \to (\rho_4 \to \sigma_4) \to \rho_4 \to \tau_4 \quad \cdots \quad \}$$

Outline

Definition

type τ is inhabited if $\vdash t : \tau$ for some combinator t

Remark

not every type is inhabited

Theorem

problem

instance: type τ
question: is τ inhabited ?

is decidable

Remark

$$\mathsf{I} : \sigma \to \sigma \qquad \mathsf{K} : \sigma \to \tau \to \sigma \qquad \mathsf{S} : (\rho \to \sigma \to \tau) \to (\rho \to \sigma) \to \rho \to \tau$$

valid formulas in intuitionistic propositional logic

Theorem (Curry–Howard)

type τ (without type constants) is inhabited if and only if τ is valid formula in implication fragment of intuitionistic propositional logic

Outline

Syntax

- basic connectives $\to \quad \wedge \quad \vee \quad \bot$
- derived connectives
  - $\neg\varphi$ abbreviates $\varphi \to \bot$
  - $\top$ abbreviates $\bot \to \bot$
  - $\varphi \leftrightarrow \psi$ abbreviates $(\varphi \to \psi) \wedge (\psi \to \varphi)$
- implication fragment contains only $\to$

Formal Semantics

- Heyting algebras
- Kripke models

Definition

Kripke model is triple $\mathcal{C} = \langle C, \leqslant, \Vdash \rangle$ with

- non-empty set C of states
- partial order $\leqslant$ on C
- binary relation $\Vdash$ between elements of C and propositional atoms

such that $d \Vdash p$ whenever $c \Vdash p$ and $c \leqslant d$

Definition

Kripke model $\mathcal{C} = \langle C, \leqslant, \Vdash \rangle$, $c \in C$

- $c \Vdash \varphi \wedge \psi$ if and only if $c \Vdash \varphi$ and $c \Vdash \psi$
- $c \Vdash \varphi \vee \psi$ if and only if $c \Vdash \varphi$ or $c \Vdash \psi$
- $c \Vdash \varphi \rightarrow \psi$ if and only if $d \Vdash \psi$ for all $d \geqslant c$ with $d \Vdash \varphi$
- $c \nVdash \bot$

Terminology

c **forces** p if $c \Vdash p$

Example

Kripke model $\mathcal{C} = \langle C, \leqslant, \Vdash \rangle$ with $C = \{a, b, c\}$, $a \leqslant b$, $a \leqslant c$, $b \Vdash p$, $c \Vdash q$

- $a \Vdash (p \rightarrow q) \rightarrow q$
- $a \Vdash \neg\neg(p \vee q)$
- $a \nVdash p \vee \neg p$

Definition

Kripke model $\mathcal{C} = \langle C, \leqslant, \Vdash \rangle$, $c \in C$

- $c \Vdash \Gamma$ if $c \Vdash \varphi$ for all $\varphi \in \Gamma$
- $\mathcal{C} \Vdash \varphi$ if $c \Vdash \varphi$ for all $c \in C$

Definition

$\Gamma \Vdash \varphi$ if $c \Vdash \varphi$ whenever $c \Vdash \Gamma$ for all Kripke models $\mathcal{C} = \langle C, \leqslant, \Vdash \rangle$ and $c \in C$

Lemma (Monotonicity)

if $c \leqslant d$ and $c \Vdash \varphi$ then $d \Vdash \varphi$

Lemma

if $\Vdash \varphi \vee \psi$ then $\Vdash \varphi$ or $\Vdash \psi$

Outline

Important Concepts

- $\Vdash$
- implication fragment
- intuitionistic propositional logic
- Kripke model
- principal type
- SC
- type inference
- type inhabitation

homework for January 8

next lecture (January 8): online evaluation in presence $\Longrightarrow$ bring device

Outline

Test on January 29

- 15:15 – 18:00 in HS 10
- online registration required before 10 am on January 23
- closed book

Earlier Exams / Tests

- SS 2022 (test)
- WS 2017 – 2
- WS 2017 – 1
- WS 2014 – 2
- WS 2014 – 1
- SS 2012
- SS 2008 – 2
- SS 2008 – 1
- SS 2007
- SS 2006 – 2
- SS 2006 – 1
- WS 2004