§3 Advanced Computability

KAIST
CS422 M. Ziegler

- WHILE programs
- UTM Theorem
- Normalform Theorem
- SMN Theorem / Currying (Schönfinkeling)
- Fixedpoint Theorem, Quines
- Oracle WHILE programs
- Post's Problem / Friedberg&Muchnik
- Arithmetic Hierarchy

WHILE Programs

KAIST
CS422 M. Ziegler

Syntax in Backus—Naur Form:

$P := (x_j := 0 \mid x_j := x_i + 1 \mid P\,;P \mid$

$\text{LOOP } x_j \text{ DO } P \text{ END} \mid \text{WHILE } x_j \text{ DO } P \text{ END})$

body better modify x_j

Semantics: loop executed as long as $x_j \neq 0$

Observation: a) To every LOOP program P there is an equivalent WHILE program P' with*out* LOOPs.

b) As opposed to LOOP programs, WHILE programs have *un*decidable Halting Problem.

Rado's Corollary: WHILE programs do **not** admit a bound $t(P,n)$ such that P on input $\underline{x}\in\mathbb{N}^k$ either at most $t(P,\|\underline{x}\|_1)$ steps or runs indefinitely.

First UTM Theorem

UTM-Theorem: There exists a LOOP program U' that, given $\langle P\rangle\in\mathbb{N}$ and $\langle x_1,\ldots,x_k\rangle\in\mathbb{N}$ and $N\in\mathbb{N}$, simulates P on input $(x_1,\ldots,x_k)$ for N steps.

Proof (Sketch): Use one variable y for $\langle x_1,\ldots,x_k\rangle$, and z to store the current program counter of P:

Switch/case $\langle P\rangle[z]$ of:

„x_j:=0" :	$\langle x_1,\ldots x_j,\ldots,x_k\rangle := \langle x_1,\ldots 0,\ldots,x_k\rangle$; $z:=z+1$
„x_i:=x_j+1" :	$\langle x_1,\ldots x_i,\ldots x_k\rangle := \langle x_1,\ldots x_j+1\ldots x_k\rangle$; $z:=z+1$
„WHILE x_j DO" :	if $x_j=0$ then $z:=1+$#of corresponding END
„END" :	z := line# of corresponding WHILE

Definition: Let $\langle P\rangle\in\mathbb{N}$ denote the encoding of WHILE program P (e.g. as ascii sequence).

Normalform Theorem

UTM-Theorem: There exists a LOOP program U' that, given $\langle P\rangle\in\mathbb{N}$ and $\langle x_1,\ldots,x_k\rangle\in\mathbb{N}$ and $N\in\mathbb{N}$, simulates P on input $(x_1,\ldots,x_k)$ for N steps.

Normalform-Thm: To every WHILE program P there exists an equivalent one P' containing only one WHILE command (and several LOOPs).

Corollary: A WHILE program U can semi-decide the *Halting problem* for WHILE programs.

$$H = \{\ (\langle P\rangle,\langle x_1,\ldots x_k\rangle) : P \text{ terminates on input } (x_1\ldots x_k)\ \}$$

SMN Theorem: Currying

Definition: Let $C=\langle P\rangle\in\mathbb{N}$ denote the encoding of WHILE program P , $P=\rangle C\langle$ its inverse/decoding.

Type conversion **example**
$f(x,y)=\sin(x)\cdot e^y$

SMN-Theorem: There exists a WHILE program that, given $\langle P\rangle\in\mathbb{N}$ and $x\in\mathbb{N}$, returns $\langle P(x,\cdot\,)\rangle$, where $P(x,\cdot\,)(\underline{y}) :\equiv P(x,\underline{y})$

UTM-Theorem: There is a WHILE program that, given $\langle P\rangle\in\mathbb{N}$, returns $\langle Q\rangle\in\mathbb{N}$ with $Q(x,\underline{y}) = \rangle P(x)\langle\ (\underline{y})$

WHILE program that, given $\langle P\rangle,\langle Q\rangle$, returns $\langle Q\circ P\rangle$

Fixedpoint Theorem and Quines

Def: For partial functions $f,g:\subseteq\mathbb{N}\to\mathbb{N}$ write $f\equiv g$ to mean $\mathrm{dom}(f)=\mathrm{dom}(g)$ and $\forall x\in\mathrm{dom}: f(x)=g(x)$.

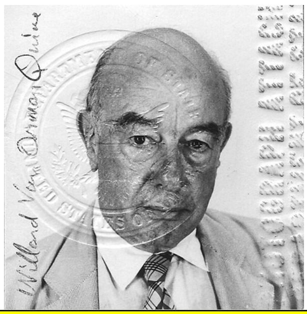
$x\equiv y \quad :\Leftrightarrow \quad \rangle x\langle \equiv \rangle y\langle$

Theorem: Every total computable function $\varphi:\mathbb{N}\to\mathbb{N}$ has a „semantic fixedpoint", i.e. $x\in\mathbb{N}$ s.t. $\varphi(x)\equiv x$.

Proof: Let $x := \psi'(\langle\varphi\circ\psi\rangle)$, where $\psi(y) := \rangle y\langle\ (y)$ partial! defined?

$\psi'(y) := \langle\, z\to\rangle\psi(y)\langle\ (z)\,\rangle$ <u>semantic extension</u> of $\psi(y)$

Application (Quines):
Let $\mathcal{A}=\mathcal{A}(p,y)$ be a program.
Consider "fixedpoint" P of $\varphi(p) := \langle\mathcal{A}(p,\cdot\,)\rangle$.



$\mathbb{N}\ni\langle P\rangle$ = code of program P, $\rangle C\langle$ = program with code $C\in\mathbb{N}$

Oracle WHILE programs

$P := (\, x_j := 0 \mid x_j := x_i + 1 \mid P\,;P \mid$ $x_j := \varphi(x_i) \mid$ LOOP x_j DO P END $\mid$ WHILE x_j DO P END $)$

Fix some *arbitrary* total $\varphi:\mathbb{N}\to\mathbb{N}$

Examples:

- $\varphi := \chi_{\mathbb{P}}$ characteristic function of Primality Probl.
- $\varphi := \chi_H$ characteristic function of Halting Problem
- $\varphi := \chi_T$ characteristic function of Totality Problem

$\chi_{\mathbb{P}} \lneq \chi_H \equiv \chi_{\bar{H}} \lneq \chi_T$

cmp. Cardinality…
(Cantor–Schröder–Bernstein)

For $\psi,\varphi \subseteq \mathbb{N}$ write $\psi \leqslant \varphi$ if there is a WHILE program with oracle φ computing ψ.

a) φ computable $\Rightarrow$ so ψ b) $\psi \leqslant \varphi \leqslant \chi \Rightarrow \psi \leqslant \chi$

Arithmetic Hierarchy

$P := (\, x_j := 0 \mid x_j := x_i + 1 \mid P\,;P \mid$ LOOP x_j DO P END $\mid$ WHILE x_j D

Fix some *arbitrary* t

Def: $\Delta_1 = \Sigma_0 = \Pi_0 =$ decidable

$\Delta_{k+1} = \text{decidable}^{\Sigma_k} = \text{decidable}^{\Pi_k}$

$\Sigma_{k+1} = \text{semi-decidable}^{\Sigma_k}$

$\Pi_{k+1} = \text{co-semi-decidable}^{\Sigma_k}$

Lemma: a) $\Delta_k = \text{co-}\Delta_k$

b) $\Delta_k = \Sigma_k \cap \Pi_k$

c) $\Sigma_k \cup \Pi_k \subseteq \Delta_{k+1}$