Grundzüge der Theoretischen Informatik

Markus Bläser

Universität des Saarlandes

1.12.2021

Kapitel 13: Reduktionen

Anwendungen

→ V_0, V, T sind nicht entscheidbar

Lemma (13.7)

$H_0 \leq V_0$.

Lemma (13.8)

$V_0 \leq V$.

Lemma (13.9)

$V_0 \leq T$.

Lemma (13.10)

$\bar{H}_0 \leq V_0$.

Beweis 13.10

$i \longrightarrow F(i)$

- Wenn $göd^{-1}(i) =: P$ nicht auf i hält, dann soll $K_i := göd^{-1}(F(i))$ die Nullfunktion berechnen
- Wenn P auf i hält, dann berechnet K_i nicht die NullFkt.

$i \in \overline{H_0}$

$\Updownarrow$

$F(i) \in V_0$

K_i sieht wie folgt aus

1. Simuliere P auf i für x_0 viele Schritte
2. Falls P nicht nach x_0 Schritten hält, gib 0 aus
3. Sonst gib 1 aus

Wenn P auf i hält, dann sei t_0 die Schrittzahl nach der P auf i hält $i \in H_0$ ($i \notin \overline{H_0}$)

	0	1	2	....	t_0-1	t_0	t_0+1	......
φ_{k_i}	0	0	0		0	1	1	......

$\text{god}(k_i) \notin V_0$

$i \in \overline{H_0}$

Wenn P nicht auf i hält, dann ex. t_0 wie oben nicht

	0	1	2	....	42....	2^{100}	....
φ_{k_i}	0	0	0		0	0	...

$\text{god}(k_i) \in V_0$

$\overline{H_0} = \{ i \mid \text{göd}^{-1}(i) \text{ hält auf } i \text{ nicht} \}$

$V_0 = \{ j \mid \varphi_{\text{göd}^{-1}(j)}(x) = 0 \text{ für alle } x \in \mathbb{N} \}$

Anwendungen (2)

$\notin RE \Rightarrow V_0, V, T$ nicht in RE

$$H_0, \bar{H}_0 \leq V_0 \leq V, T$$

$$\bar{H}_0, H_0 \leq \bar{V}_0 \leq \bar{V}, \bar{T}$$

$\notin RE \Rightarrow \bar{V}_0, \bar{V}, \bar{T} \notin RE$

Theorem (13.11)

V, V_0 und T sind nicht rekursiv aufzählbar noch sind es ihre Komplemente.

$L \leq L' \rightarrow \bar{L} \leq \bar{L}'$

Kapitel 14: Mehr zu Reduktionen

$$\text{göd} : \mathcal{W} \to \mathbb{N}$$

Wir dürfen ab jetzt φ_i statt $\varphi_{\text{göd}^{-1}(i)}$ schreiben,
d.h. wir identifizieren Gödelnummer und Programm.

Das S-m-n-Theorem

P mit Gödelnummer g

$\varphi_P^s : \mathbb{N}^s \to \mathbb{N}$ $\quad \varphi_g^s$

Theorem (S-m-n-Theorem)

Für alle $m, n \geq 1$ *gibt es eine FOR-berechenbare Funktion* $S_n^m : \mathbb{N}^{m+1} \to \mathbb{N}$, *so dass für alle* $g \in \mathbb{N}$, $y \in \mathbb{N}^m$ *und* $z \in \mathbb{N}^n$

$$\varphi_g^{m+n}(y, z) = \varphi^n_{\underbrace{S_n^m(g,y)}_{g'(y)}}(z)$$

gilt.

Beweis

- Sei $P_g = \text{göd}^{-1}(g)$.
- Sei $y = (\eta_0, \ldots, \eta_{m-1})$.
- Konstruiere $Q_{g,y}$ mit

$$\varphi_{P_g}(y, z) = \varphi_{Q_{g,y}}(z) \quad \text{für alle } z \in \mathbb{N}^n :$$

z steht in $x_0, \ldots, x_{n-1}$

$x_{m+n-1} := x_{n-1};$
$\vdots$
$x_m := x_0;$
$x_{m-1} := \eta_{m-1};$
$\vdots$
$x_0 := \eta_0;$
P_g

$S^n_n : \quad (g, y) \mapsto \text{göd}(Q_{g,y})$

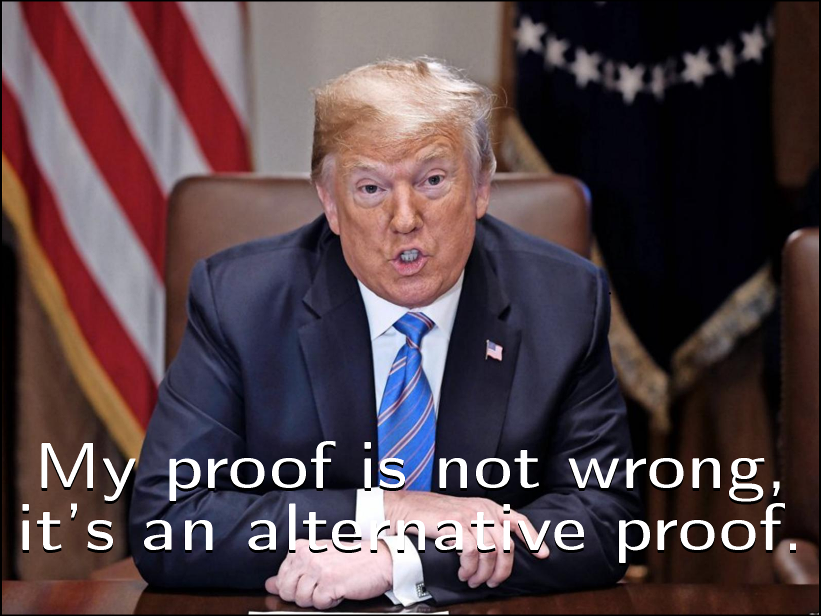
FOR-berechenbar

Das S-m-n-Theorem (2)

- Der Beweis des S-m-n-Theorems ist einfach.
  Dies liegt an der WHILE-Programmiersprache.
- Das S-m-n-Theorem gilt allgemein für akzeptable Programmiersysteme.
  Der Beweis ist dann deutlich abstrakter.

Anwendungen

Alternativer Beweis von Lemma 13.10: $\bar{\mathrm{H}}_0 \leq \mathrm{V}_0$.


My proof is not wrong,
it's an alternative proof.

Anwendungen

Alternativer Beweis von Lemma 13.10: $\bar{H}_0 \leq V_0$.

Definiere $f : \mathbb{N}^2 \to \mathbb{N}$ durch

$$f(g, m) = \begin{cases} 0 & \text{falls } g \text{ nicht auf } g \text{ nach } \leq m \text{ Schritten hält} \\ 1 & \text{sonst} \end{cases}$$

f ist WHILE-berechenbar, wir können das getaktete universelle Programm verwenden

Sei e eine Gödelnummer für ein Programm, das f berechnet

S-m-n: $F(g,m) = \varphi_e^2(g,m) = \varphi^1_{S^1_1(e,g)}(m)$ für alle m

$g \mapsto S^1_1(e,g)$ ist die gesuchte Reduktionsfkt

$g \in \overline{H}_0 \iff g$ hält nicht auf g

$\iff f(g,m) = 0$ für alle m

$\iff S^1_1(e,g) \in V_0$

Noch ein Problem . . .

Ref. von φ_g

$$D_c = \{g \mid g \in \mathbb{N} \text{ and } |\mathrm{dom}\, \varphi_g| \geq c\}$$

Theorem (14.3)

For every $c \geq 1$, $H_0 \leq D_c$. $\Rightarrow$ $D_c \notin REC$

Theorem (14.4)

For every c, $D_c \in$ RE.

$T = \{g \mid \varphi_g \text{ ist total}\}$

$= \{g \mid \mathrm{dom}\, \varphi_g = \mathbb{N}\}$

Beweis 14.3 $(H_0 \leq D_c)$

Def: $f: \mathbb{N}^2 \to \mathbb{N}$

$$f(c,x) = \begin{cases} 0 & \text{falls } \varphi_c(c) \text{ definiert ist} \\ \text{undef} & \text{sonst} \end{cases}$$

F ist WHILE-ber., da zuerst c auf c simuliert wird mit dem universellen Programm. Hält die Simulation, so gibt 0 aus.

Hält die Simulation nicht, dann ist $f(c,x)$ undef wie verlangt.

Sei e eine Gödelnummer, die F berechnet.

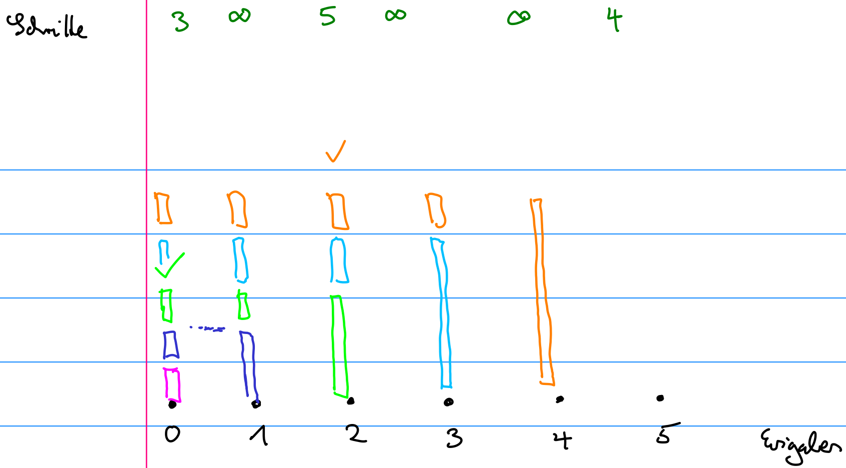
S-m-n-Theorem

$$F(i,x) = \varphi_e^2(i,x) = \varphi_{S_1^1(e,i)}(x) \quad \text{für alle } x \in \mathbb{N}$$

Beh.: $i \mapsto S_1^1(e,i)$ ist die gesuchte Reduktionsfkt.

- Sei $i \in H_0$, dann ist $f(i,x) = 0$ für alle x.
  Damit ist $\operatorname{dom} f(i,.) = \mathbb{N}$. Insbesondere ist $|\operatorname{dom}(\underbrace{f(i,.)}_{\varphi_{S_1^1(e,i)}})| \geq c$. $\Rightarrow S_1^1(e,i) \in D_c$

- Sei $i \notin H_0$, dann ist $f(i,x)$ immer undef.
  Damit ist $|\operatorname{dom} \varphi_{S_1^1(e,i)}| = 0$
  $\Rightarrow S_1^1(e,i) \notin D_c$. $\square$


Schritte
3 ∞ 5 ∞ ∞ 4
0 1 2 3 4 5
Eingaben

Beweis Theorem 14.4

P

Bijektion $\mathbb{N} \to \mathbb{N}^{c+1}$

$m \to (x_1, \ldots, x_c, i)$

Input: $g \in \mathbb{N}$

$m := 0$;
$t := 1$;
while $t \neq 0$ **do**
    Interpretiere m als Tupel $(x_1, \ldots, x_c, i)$
    if $x_1, \ldots, x_c$ sind paarweise verschieden **then**
        Simuliere g auf $x_1, \ldots, x_c$ für jeweils i Schritte
        if alle c Simulationen halten **then**
            $t := 0$
        fi
    fi
    $m := m + 1$;
od
return 1;

Wenn P hält und 1 ausgibt, dann ist $g \in D_c$, denn g hält auf $x_1, \dots, x_c$.

Wenn $g \in D_c$ dann gibt es $\xi_1, \dots, \xi_c$, so dass g auf c hält. Sei t die maximale Schrittanzahl dabei. Sobald $(\xi_1, \dots, \xi_c, t)$ in der While-Schleife betrachtet wird, wird P die Schleife verlassen und 1 ausgeben. Das Tupel kommt im n-ten Durchlauf der Schleife dran, wobei $n \mapsto (\xi_1, \dots, \xi_c, t)$ unter der Bijektion $\mathbb{N} \to \mathbb{N}^{c+1}$. $\square$

Die goldene Regel der Verwirrung

Folgende Aussagen sind äquivalent/meinen dasselbe:

- $L \in \mathrm{REC}$
- L ist entscheidbar
- L ist rekursiv
- χ_L ist WHILE-berechenbar

Ebenso:

- $L \in \mathrm{RE}$
- L ist semi-entscheidbar
- L ist rekursiv aufzählbar
- χ'_L ist WHILE-berechenbar