Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Kooperation der AE Klinische Psychologie
und Arbeits- und Organisationspsychologie
der UdS und dem DFKI
Kontaktpersonen:
Prof. Dr. Monika Equit
Dr. Nida Bajwa

UNIVERSITÄT DES SAARLANDES

Verdiene 40€

Oder 5 VPS (für Psychologiestudent*innen)

Versuchspersonen gesucht!

Die Klinische Psychologie, die Arbeits- und Organisationspsychologie und das DFKI suchen geeignete Probanden für eine interdisziplinäre Studie zur Erprobung eines digitalen Interventionsverfahrens. Mit deiner Teilnahme kannst Du Dir etwas dazuverdienen und gleichzeitig die Forschung unterstützen!

Wie kann ich teilnehmen?

1. Du nimmst an unserem Screening-Fragebogen teil
2. Wir benachrichtigen dich wenn Du zur Teilnahme geeignet bist

Was umfasst meine Teilnahme?

- Teilnahme an Screening
- Einführungstreffen bei dem Du in das Studienmaterial eingewiesen wirst
- Abendliches reflektieren deines Tages mit Hilfe des Studienmaterials
- Währenddessen Fragebögen
- Abschlusstreffen für Feedback über die Intervention

Chance auf 10€ Amazon-Gutschein durch Teilnahme am Screening

Du hast Interesse? Dann nimm mit diesem QR Code oder dem Link direkt am Screening teil!
https://www.soscisurvey.de/EScreen/?q=screen

Grundzüge der Theoretischen Informatik

Markus Bläser

Universität des Saarlandes

24.11.2021

Kapitel 12: Das Halteproblem

Das Halteproblem

Halteproblem:

$$H = \{\langle g, m\rangle \mid \text{göd}^{-1}(g) \text{ hält auf } m\}.$$

Spezielle Halteproblem:

$$H_0 = \{g \mid \text{göd}^{-1}(g) \text{ hält auf } g\}.$$

Theorem (12.1)

$H_0 \notin$ REC, *d.h. das spezielle Halteproblem ist unentscheidbar.*

Beweis

$\{ g \mid \text{göd}^{-1}(g) \text{ hält auf } g \}$

- Annahme: Es gibt ein P_0, das H_0 entscheidet.
- Sei P folgendes Program:

```
1: P_0;
2: if x_0 = 1 then
3:     x_1 := 1;
4:     while x_1 ≠ 0 do
5:         x_1 := 1
6:     od
7: fi
```

P_0 kann nicht existieren. $\square$

Was macht P auf Eingabe $g := \text{göd}(P)$?

1) P hält auf g. Dann gibt P_0 1 aus und P geht in Zeile 2 in eine Endlosschleife, also hält P nicht auf g ↯

2) P hält nicht auf g. Dann gibt P_0 0 aus und P hält auf g ↯

Rekursiv aufzählbare Sprachen

REC ≠ RE

REG

Achtung!

Definition (12.2)

1. $L \subseteq \mathbb{N}$ heißt *rekursiv aufzählbar*, falls es ein WHILE-Programm P gibt mit:
   1.1 für alle $x \in L$ gilt $\varphi_P(x) = 1$ und
   1.2 für alle $x \notin L$ gilt $\varphi_P(x) = 0$ oder $\varphi_P(x)$ ist undefiniert
2. Die Menge aller rekursiv aufzählbaren Sprachen wird mit RE bezeichnet.

Modifizierte charakteristische Funktion:

$$\chi'_L(x) = \begin{cases} 1 & \text{falls } x \in L, \\ \text{undefined} & \text{sonst.} \end{cases}$$

Es gilt: $L \in \mathrm{RE}$ genau dann wenn χ'_L WHILE-berechenbar ist.

Rekursiv aufzählbare Sprachen (2)

Theorem (12.4)

$\mathrm{H}, \mathrm{H}_0 \in \mathrm{RE}$. $\rightsquigarrow$ $\mathrm{REC} \subsetneq \mathrm{RE}$

Theorem (12.6)

Die folgenden Aussagen sind für $\mathrm{L} \subseteq \mathbb{N}$ *äquivalent:*

1. $\mathrm{L} \in \mathrm{REC}$.
2. $\mathrm{L}, \bar{\mathrm{L}} \in \mathrm{RE}$.

$\bar{L} = \mathbb{N} \setminus L = \{ x \in \mathbb{N} \mid x \notin L \}$

Corollary (12.7)

$\bar{\mathrm{H}}_0 \notin \mathrm{RE}$.

Wäre $\bar{H}_0 \in RE$, dann folgt aus 12.6, dass $H_0 \in REC$ ↯.

Beweis 12.4

Simuliere g auf n mit dem universellen WHILE-Prg.
Wenn die Simulation endet, gib 1 aus.

1) $\langle g,n \rangle \in H$, dann endet die Simulation und
es wird 1 ausgegeben

2) $\langle g,n \rangle \notin H$, dann endet die Simulation nicht
und die Ausgabe ist undef.

D.h. das Programm oben berechnet χ'_H.

H_0 geht genau so. $\square$

Wie sieht die char. Fkt von $\overline{L}$ aus? 12,5

$\chi_{\overline{L}}(x) = 1 - \chi_L(x)$

L ist entscheidbar, was ist mit $\overline{L}$?

$L \in REC \Rightarrow L, \overline{L} \in REC \subseteq RE$

$$\chi_L(x) = \begin{cases} 1 & \text{falls } x \in L \\ 0 & \text{falls } x \notin L \end{cases}$$

χ_L' ist WHILE-berechenbar. Was ist mit $\chi_{\overline{L}}'$?

$$\chi_{\overline{L}}'(x) = \begin{cases} \text{undef.} \\ 1 \end{cases}$$

$\chi_L'(x) = 1$

$\chi_L'(x)$ undef.

$L, \bar{L} \in RE \overset{!}{\Rightarrow} L \in REC$

χ_L^1 ist berechenbar $\qquad P$

$\chi_{\bar{L}}^1$ ist berechenbar. $\qquad \bar{P}$

Daraus kann man ein Programm für χ_L bauen.
Wir modifizieren das universelle Prg. so, dass es zwei Programme parallel ausführen kann, d.h. alle internen Variablen werden ogdoppelt und in der WHILE-Schleife werden jeweils ein Schritt von beiden Programmen ausgeführt. $\square$

$L, \bar{L} \in \mathrm{REC}$

$L \in \mathrm{RE}$,
$\bar{L} \notin \mathrm{RE}$
(oder umgekehrt)

H_1 H_0

$L, \bar{L} \notin \mathrm{RE}$ $\dots$

Namensgebung

Warum heißen rekursiv aufzählbare Sprachen rekursiv aufzählbar?

Aufgabe

Die folgenden Aussagen sind für $L \subseteq \mathbb{N}$ äquivalent:

1. $L \in$ RE. (im Sinne unserer Def., d.h. χ'_L ist while-ber.)
2. Es gibt ein WHILE-Programm P mit $L = \mathrm{im}\, \varphi_P$.
3. $L = \emptyset$ oder es gibt ein FOR-Prgramm P mit $L = \mathrm{im}\, \varphi_P$.

$\{ \varphi_P(0), \varphi_P(1), \varphi_P(2), \varphi_P(3) \dots \quad \} = L$

Q berechnet χ'_L

Kapitel 13: Reduktionen

Weitere Probleme

$\langle .,. \rangle : \mathbb{N}^2 \to \mathbb{N}$

$V \subseteq \mathbb{N}$ $\qquad \bar{V} = \mathbb{N} \setminus V$

$= \{ \langle i,j \rangle \mid \varphi_{\text{göd}^{-1}(i)} \neq \varphi_{\text{göd}^{-1}(j)} \}$

- Programmäquivalenz (allgemeine Verifikation)

$$V = \{\langle i, j\rangle \mid \varphi_{\text{göd}^{-1}(i)} = \varphi_{\text{göd}^{-1}(j)}\}.$$

- Spezielles Verfikationsproblem

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

- Terminierungsproblem

$$T = \{i \mid \varphi_{\text{göd}^{-1}(i)} \text{ is total}\}.$$

Reduktion

Idee:

Wir zeigen z.B.: $V \in \text{REC} \Longrightarrow H_0 \in \text{REC}$

P für χ_V $\rightsquigarrow$ Q für χ_{H_0}

Reduktionen

$x \in L \rightarrow f(x) \in L'$

$x \notin L \Rightarrow f(x) \notin L'$

Definition (13.1)

Seien $L, L' \subseteq \mathbb{N}$.

1. Eine WHILE-berechenbare totale Funktion $f : \mathbb{N} \to \mathbb{N}$ heißt *many-one-Reduktion* von L auf L', falls

$$\text{für alle } x \in \mathbb{N}: \quad x \in L \iff f(x) \in L'.$$

2. Falls solch ein f existiert, dann heißt L *rekursiv many-one-reduzierbar* auf L'. "$L \leq L'$"

Reductionen (2)

RE= recursive enumerable

REC = recursive

Lemma (13.3)

Seien $\mathrm{L}, \mathrm{L}' \subseteq \mathbb{N}$ *und* $\mathrm{L} \leq \mathrm{L}'$. *Dann gilt:*

1. *Falls* $\mathrm{L}' \in \mathrm{RE}$, *dann ist* $\mathrm{L} \in \mathrm{RE}$.
2. *Falls* $\mathrm{L}' \in \mathrm{REC}$, *dann ist* $\mathrm{L} \in \mathrm{REC}$.

Corollary (13.4)

Seien $\mathrm{L}, \mathrm{L}' \subseteq \mathbb{N}$ *und* $\mathrm{L} \leq \mathrm{L}'$. *Dann gilt:*

1. *Falls* $\mathrm{L} \notin \mathrm{RE}$, *dann ist* $\mathrm{L}' \notin \mathrm{RE}$.
2. *Falls* $\mathrm{L} \notin \mathrm{REC}$, *dann ist* $\mathrm{L}' \notin \mathrm{REC}$.

$H_0 \leq V \quad \Rightarrow \quad V \notin REC$

13.3.

$L \leq L'$ per F

! $L' \in$ ~~RE~~ REC, dann ist L in ~~RE~~ REC.

$\exists P'$ für ~~$\chi'_{L'}$~~ $\chi_{L'}$

Gesucht ist ein Programm P für ~~χ'_L~~ χ_L

$$\left.\begin{array}{l} 1: x_0 := F(x_0); \\ 2: P' \end{array}\right\} P$$

Wenn $x \in L$, dann ist $F(x) \in L'$

und damit hält P' und gibt 1 aus

Wenn $x \notin L$, dann ist $F(x) \notin L'$

und damit ~~hält P' nicht~~ hält P' und gibt 0 aus

$\rightarrow P$ berechnet die ~~modifizierte~~ char. ~~Fkt.~~ Fkt. ~~χ'_L~~ χ_L

χ_L $\chi_{L'}$ f

$x \in L \Rightarrow f(x) \in L'$

$x \notin L \Rightarrow f(x) \notin L'$

$\chi_L(x) = 0$ $\chi_{L'}(f(x)) = 0$

$\chi_L(x) = 1$ $\chi_{L'}(f(x)) = 1$

$\chi_L(x) = \chi_{L'}(f(x)) \rightarrow \chi_L = \chi_{L'} \circ f$

WHILE-ber $\Leftarrow$ WHILE-ber

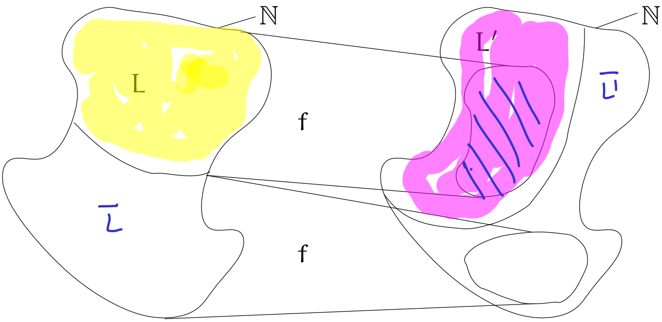
agnanso: $\chi'_L = \chi'_{L'} \circ f$

Reduktionen (3)

Lemma (13.6)

$\leq$ *ist eine transitive Relation.*

$x \in L \quad \Rightarrow \quad f(x) \in L'$



$x \notin L \quad \Rightarrow \quad f(x) \notin L'$