

“Theorie der Informatik” (CS206)

Akzeptierbar, rekursiv aufzählbar, (Semi-) entscheidbare Menge/Sprache

3. April 2014

Proff Malte Helmert und Christian Tschudin

Departement Mathematik und Informatik, Universität Basel

Vorschau verwandter Konzepte mit unterschiedl. Stärken

Wort ableiten

Wort akzeptieren

semi-entscheidbar

rekursiv aufzählbar

charakteristische Funktion, Berechenbarkeit

entscheidbar

Vollständigkeit

Entscheidbare Menge – Definition

Eine Menge $A \subseteq \Sigma^*$ heisst entscheidbar, falls die charakteristische Funktion $\chi_A : \Sigma^* \rightarrow \{0, 1\}$ berechenbar ist, wobei

$$\chi_A(w) := 1 \quad \text{falls } w \in A$$

$$\chi_A(w) := 0 \quad \text{falls } w \notin A$$

Semi-Entscheidbare Menge – Definition

Eine Menge $A \subseteq \Sigma^*$ heisst **semi**-entscheidbar, falls die Funktion $\chi'_A : \Sigma^* \rightarrow \{0, 1\}$ berechenbar ist, wobei

$$\chi'_A(w) := 1 \quad \text{falls } w \in A$$

$$\chi'_A(w) := \text{undefiniert} \quad \text{falls } w \notin A$$

Das heisst:

Aus “berechenbare Menge” folgt nicht “entscheidbare Menge” (χ' ist nämlich berechenbar!).

Entscheidbar / semi-entscheidbar

Wie aus semi-entscheidbaren Funktionen eine entscheidbare Funktion machen?

Satz (Beweis folgt weiter unten):

Eine Sprache ist entscheidbar

$\Leftrightarrow$

A wie auch das Komplement $\bar{A}$ sind semi-entscheidbar

Semi-entscheidbar $\sim$ “rekursiv-aufzählbar” (Grundidee)

Falls alle Wörter einer Sprache aufgezählt (erzeugt) werden können, kann ein Bezug zu semi-entscheidbar hergestellt werden:

- Ich kann ein Wort nach dem anderen (auch mit Wiederholungen) generieren,
- wenn ich auf das vorgegebene Wort treffe, halte ich an
(ich habe die Ableitung für das vorgegebene Wort gefunden).

Anwendung auf Wort-Problem:

- Gegeben: Wort w . Gesucht: gilt $w \in L$?
- alle Wörter von L generieren (zuerst der Länge 0, der Länge 1 etc)
- wenn w gefunden wird halten wir an und geben 1 aus

Dies entspricht der Definition von semi-entscheidbar!

“Akzeptieren” $\sim$ semi-entscheidbar

Definition von “rekursiv-aufzählbar”

Eine Sprache $A \subseteq \Sigma^*$ ist “rekursiv-aufzählbar”, falls

- entweder A ist die leere Menge
- oder es gibt eine totale und berechenbare Funktion $f : \mathbb{N} \rightarrow \Sigma^*$ mit $A = \{f(0), f(1), \dots\}$
(f “zählt” A auf, ev. mit Wiederholungen)

Satz:

Sprache A ist rekursiv aufzählbar

$\Leftrightarrow$

Sprache ist semi-entscheidbar

Beweis, dass “rekursiv-aufz. $\Leftrightarrow$ semi-entscheidbar”

Richtung “ $\Rightarrow$ ”:

- Aufzählfunktion $f(x)$ für die Wörter einer Sprache A ist bekannt.
Zu zeigen: $\chi(w)$ ist berechenbar für alle $w \in A$
- Programm:

```
Wort w als Input;  
z = 1;  n = 0;  
WHILE  z <> 0  DO  
    IF f(n) == w THEN  z = 0;  FI  
    n = n + 1;  
DONE  
RETURN 1
```

Beweis, “rekursiv-aufz. $\Leftrightarrow$ semi-entscheidbar” (Forts)

Richtung “ $\Leftarrow$ ”:

- Sprache A ist semi-entscheidbar, d.h. es ex. Programm M für χ
Wie sieht eine Aufzählung f aus?
 $f(x)$ muss total und berechenbar sein!
- Vorgehen: Baue eine Funktion $f(n)$, die für jedes n
ein Wort aus A erzeugt. n wird als Kodierung eines Zahlenpaars
 $\langle x, y \rangle$ interpretiert, wobei:
 - x ist ein Wortkandidat
 - y ist eine Laufzeitbeschränkung.

Beweis, “rekursiv-aufz. $\Leftrightarrow$ semi-entscheidbar” (Forts 2)

- Wähle ein festes $a \in A$ beliebig.
Betrachte Hilfsprogramm $f(n)$:

Berechne x, y aus n ;
IF $M(x)$ stoppt in y Schritten und gibt 1 aus THEN
 OUTPUT(x);
ELSE
 OUTPUT(a);
FI
- Stoppt immer,
Gibt immer nur Wörter aus A aus.
D.h. $f()$ total und berechenbar, zudem: $f(\mathbb{N}) \subseteq A$

Beweis, “rekursiv-aufz. $\Leftrightarrow$ semi-entscheidbar” (Forts 3)

- Kann es Elemente $z \in A$ geben, so dass es kein n dazu gibt?
(d.h. $f(\mathbb{N}) \subset A$?)
- Nein, denn:
 - sei $z \in A$
 - dann hält M nach s Schritten an
 - daraus kann n berechnet werden: $n := c(z, s)$
- Mit anderen Worten: unser Programm zu $f()$
ist die gewünschte (vollständige) Aufzählung von A

“Akzeptieren” (wichtiger Ueberblick!)

Zu einer Sprache L

gibt es eine TM M , die ein Wort w **akzeptiert**
(d.h. M hält an falls $w \in L$, und gibt 1 aus)

$\Leftrightarrow$

Sprache L ist **rekursiv aufzählbar**

$\Leftrightarrow$

Menge L ist **semi-entscheidbar**

“Entscheiden” (volle Entscheidbarkeit)

Eine Sprache L ist entscheidbar, wenn es eine TM gibt, die für ein beliebiges Wort w entscheidet, ob $w \in L$ **oder ob** $w \notin L$.

(siehe auch Definition der charakteristischen Funktion)

- Andere Formulierung: falls eine Menge entscheidbar ist, gibt es eine TM, die in *jedem* Fall (beliebiges w) anhält und den Entscheid bekanntgibt.
- Entscheidbarkeit (einer Sprache oder Menge) **ist eine stärkere** d.h. einschränkendere Eigenschaft als Akzeptierbarkeit.

Semi-Entscheidbar i.A. zu schwach für Entscheidbarkeit

$A \subseteq \Sigma^*$ heisst entscheidbar, falls $\chi_A(w)$ berechenbar ist.

Satz: Eine Menge A ist entscheidbar $\Leftrightarrow w \in A$ wie auch $w \notin A$ sind semi-entscheidbar.

- “ $\Rightarrow$ ” klar
- “ $\Leftarrow$ ” Wir haben zwei χ -Funktionen:
 $\chi_A(w) = 1$ wenn $w \in A$, undefiniert sonst
 $\chi_{\bar{A}}(w) = 1$ wenn $w \notin A$, undefiniert sonst
Daraus eine (immer abbrechendes) Programm bauen
(siehe nächstes Slide)

A und $\bar{A}$ semi-entscheidbar $\Rightarrow A$ entscheidbar (Forts)

Iteration über Programmlaufzeit s :

```
Wort w als Input;  
s = 1;  z = 1;  
WHILE  z <> 0  DO  
  IF M_A(w) hält in s Schritten THEN  
    OUTPUT(1);  
    z = 0;  
  FI  
  IF M_NOTA(w) hält in s Schritten THEN  
    OUTPUT(0);  
    z = 0;  
  FI  
  s = s + 1;  
DONE
```

Ueberblick verwandter Konzepte mit unterschiedl. Stärken

Wort ableiten

Wort akzeptieren

semi-entscheidbar

rekursiv aufzählbar

charakteristische Funktion, Berechenbarkeit

entscheidbar

Vollständigkeit

Einschub: Postsches Korrespondenz Problem (PKP)



- Emil Leon Post 1897-1954
- In Polen geboren, in USA aufgewachsen und gearbeitet
- 1944 Veröffentlichung über “Recursively enumerable sets”
- 1946 PKP und dessen Unentscheidbarkeit, als erstes konkretes Beispiel einer nicht rekursiv lösbaren Aufgabe

Einschub: Postsches Korrespondenz Problem (Forts.)

Das PKP ist ein “string matching”-Problem.

Gegeben: eine endliche Menge von Dominosteintypen.

Gefragt: Dominos (mit Wiederholung) so plazieren, dass oben und unten das gleiche Wort entsteht.

Beispiel: siehe nächstes Slide.

Eigenschaft: PKP ist nicht entscheidbar – das PKP kann auf das sog. Akzeptanzproblem für Turingmaschinen zurückgeführt werden.

Postisches Korrespondenz Problem – Beispiel

Typen:

01	0	00
0	00	1

Mögliche Lösung des PCP:

01	00	0	0
0	1	00	00