Entscheidbarkeit und Semi-entscheidbarkeit – Eigenschaften

L_1, L_2 entscheidbar $\Rightarrow$ $L_1 \cap L_2$, $L_1 \cup L_2$, $L_1 \setminus L_2$ ebenfalls entscheidbar.

Beweis für $\cap$:

Für $i \in \{1,2\}$ sei M_i eine TM, die L_i entscheidet.

Sei M' die TM, die folgenden Algorithmus ausführt:

Eingabe: w

Führe M_1 auf w aus.

Falls M_1 akz., führe M_2 auf w aus.

Falls M_2 akz., akzeptiere.

Sonst lehne ab.

L_1, L_2 semi-entscheidbar $\Rightarrow$ $L_1 \cap L_2$, $L_1 \cup L_2$ semi-entscheidbar.

Beweis für $\cap$: wie bei Entscheidbarkeit.

Für $i \in \{1,2\}$ sei M_i eine TM, die L_i semi-entscheidet.

Folgender Algorithmus ist Semi-entscheidungs-Algorithmus für $L_1 \cup L_2$:

```
Eingabe: w
  i = 0
  while(true) {
      Simuliere M1 für i Schritte. Falls M1 akz., akzeptiere.
      Simuliere M2 für i Schritte. Falls M2 akz., akzeptiere.
      i++
  }
```

L entscheidbar $\iff$ L und $\bar{L} = \Sigma^* \setminus L$ semi-entscheidbar.

"$\Rightarrow$": L entscheidbar $\Rightarrow$ L semi-entscheidbar.

L entscheidbar $\Rightarrow$ $\bar{L}$ semi-entscheidbar
(vertausche acc und rej in Maschine, die L entscheidet)

"$\Leftarrow$": L und $\bar{L}$ semi-entscheidbar.
Sei M_1 (M_2) eine TM, die L ($\bar{L}$) semi-entscheidet.
Folgender Alg. entscheidet L:

```
Eingabe: w
  i=1
  while(true) {
      Simuliere M1 auf w für i Schritte. Falls M1 akz., akzeptiere.
      Simuliere M2 auf w für i Schritte. Falls M2 akz., lehne ab.
      i++
  }
```

Universelle Turingmaschine

Universell: "kann sich je nach Eingabe wie beliebige TM verhalten" bzw.
"kann beliebige TM simulieren"

Ziel: universelle TM U bekommt als Eingabe $\langle M, w\rangle$

U akz. $\langle M, w\rangle$ gdw. M die Eingabe w akz.

U lehnt $\langle M, w\rangle$ ab gdw. M die Eingabe w ablehnt.

Dafür müssen TMs als Wörter kodiert werden.

Man nennt so eine Kodierung auch Gödelisierung.

Alphabet: $\{0, 1, [,], \cdot, |\}$

$\Gamma = \{a_0, a_1, \ldots, a_k\}$

$Q = \{q_0, q_1, \ldots, q_n\}$

Kodiere Zustände und Symbole durch Index, also

$$\langle q_i \rangle = bin(i)$$

$$\langle a_i \rangle = bin(i)$$

Im Falle einer Eingabe $\langle M, w \rangle$ für U wird w wie folgt kodiert:

$$\langle w \rangle = [\langle w_1 \rangle \cdot \langle w_2 \rangle \cdot \ldots \cdot \langle w_m \rangle],$$ wenn

$w = w_1 w_2 \cdots w_m$ für $w_i \in \Gamma$.

Überführungsfkt. δ:

$\delta(q, a) = (q', b, \Delta)$ wird kodiert durch

$$[\langle q \rangle \cdot \langle a \rangle \,|\, \langle q' \rangle \cdot \langle b \rangle \cdot \langle \Delta \rangle]$$

$\langle -1 \rangle = 0$

$\langle +1 \rangle = 1$

Übergänge zu rej weglassen

$\rightsquigarrow$ Kodierung $\langle \delta \rangle$ von δ

$M = (Q, \Sigma, \Gamma, \delta, q_0, acc, rej, \square)$

Kodierung $\langle M \rangle$ von M:

$[\langle acc \rangle \bullet \langle \square \rangle][\langle \delta \rangle]$

$\langle M, w \rangle$

$$\langle M \rangle = [\langle acc \rangle \cdot \langle \square \rangle][\langle \delta \rangle]$$

$$\langle w \rangle = [\langle w_1 \rangle \cdot \langle w_2 \rangle \cdot \ldots \cdot \langle w_m \rangle]$$

Arbeitsweise von UTM U

Kodierung der Startkonfiguration

$$[[\langle start \rangle \cdot \langle w_1 \rangle][\langle rej \rangle \cdot \langle w_2 \rangle] \ldots [\langle rej \rangle \cdot \langle w_m \rangle]]$$

Ein Simulationsschritt von U:

- Suche Bandzelle an aktueller Kopfposition (Zelle mit Zustand $q \neq rej$ in erster Komponente)
- Suche nach der Kodierung dieser Zelle in Kodierung von δ
- Falls kein passender Übergang vorhanden, lehne ab.
- Verändere Konfiguration auf Arbeitsband gemäß δ
- Falls acc erreicht wird, akz.

Es gibt unentscheidbare Sprachen

$$\text{SELFREJECT} := \{ \langle M \rangle \mid M \text{ lehnt } \langle M \rangle \text{ ab} \}$$

SELFREJECT ist semi-entscheidbar mit folgendem Algorithmus:

Eingabe: $\langle M \rangle$

Simuliere M auf Eingabe $\langle M \rangle$.

Falls M ablehnt, akzeptiere.

Sonst: Endlosschleife.

$\text{SELFREJECT} := \{\langle M\rangle \mid M \text{ lehnt } \langle M\rangle \text{ ab}\}$

SELFREJECT ist unentscheidbar.

Angenommen SELFREJECT wäre entscheidbar. Dann gibt es eine Maschine SR, die SELFREJECT entscheidet.

Dann gilt f. a. TMs M:

$$SR \text{ akz. } \langle M\rangle \iff M \text{ lehnt } \langle M\rangle \text{ ab}$$

$$SR \text{ akz. } \langle SR\rangle \iff SR \text{ lehnt } \langle SR\rangle \text{ ab} \quad ↯$$

Also: Eine solche Maschine SR existiert nicht.

$\overline{\text{SELFREJECT}}$ ist nicht semi-entscheidbar.

Wissen, dass SELFREJECT semi-entscheidbar ist. Wäre $\overline{\text{SELFREJECT}}$ ebenfalls semi-entscheidbar, würde folgen, dass SELFREJECT entscheidbar ist. ↯

$\text{SELFHALT} = \{ \langle M \rangle \mid M \text{ hält auf Eingabe } \langle M \rangle \}$

Beweise für Unentscheidbarkeit mittels Reduktion

$L \leq_T L'$: es gibt einen Algorithmus, der Orakelfragen an L' stellen darf und L entscheidet.

Fragen der Form "$x \in L'$?", Antwort erfolgt in konst. Zeit

$L \leq_T L'$ und L' entscheidbar $\Rightarrow$ L entscheidbar.

$L \leq_T L'$ und L unentscheidbar $\Rightarrow$ L' unentscheidbar.

$\text{ACCEPT} := \{ \langle M, w \rangle \mid M \text{ akzeptiert bei Eingabe } w \}$

$\text{SELFREJECT} \leq_T \text{ACCEPT}$

$\text{SELFREJECT} := \{ \langle M \rangle \mid M \text{ lehnt } \langle M \rangle \text{ ab} \}$

Eingabe: $\langle M \rangle$

Modifiziere M so, dass acc und rej vertauscht sind

Sei M' die erhaltene Maschine

Orakelfrage: $\langle M', \langle M \rangle \rangle \in \text{ACCEPT}$?

Falls ja, akzeptiere.

Sonst lehne ab.

$\langle M \rangle \in \text{SELF REJECT}$

$\iff M$ lehnt $\langle M \rangle$ ab

$\iff M'$ akz. $\langle M \rangle$

$\iff \langle M', \langle M \rangle \rangle \in \text{ACCEPT}$

Es folgt: ACCEPT ist unentscheidbar.

$\text{NEVERACCEPT} := \{ \langle M \rangle \mid M \text{ akz. keine Eingabe} \}$

$\text{ACCEPT} \leq_T \text{NEVERACCEPT}$

Eingabe: $\langle M, w \rangle$

Konstruiere Maschine M', die wie folgt arbeitet:

Eingabe: z

Lösche die Eingabe

Schreibe w auf das Band

Führe M aus

Orakelfrage: $\langle M' \rangle \in \text{NEVERACCEPT}$?

Falls ja, lehne ab.

Sonst akzeptiere.

Es folgt: NEVERACCEPT ist unentscheidbar.

Der Satz von Rice

Sei S eine nicht-triviale Menge von berechenbaren Funktionen.

Dann ist $C(S) := \{ \langle M \rangle \mid M$ berechnet eine Funktion aus $S \}$

unentscheidbar.

(nicht-trivial: $S \neq \emptyset$ und S ist nicht Menge aller berechenbaren Funktionen)

Beweis: Sei Ω die überall undef. Fkt. Falls $\Omega \in S$:

Reduktion von SELFREJECT auf $\overline{C(S)}$.

Wähle $f_0 \notin S$. Folgender Algorithmus zeigt SELFREJECT $\leq_T \overline{C(S)}$.

Eingabe: $\langle M \rangle$

Konstruiere M', die wie folgt arbeitet:

Eingabe: z

Führe M auf Eingabe $\langle M \rangle$ aus.

Falls M ablehnt, berechne $f_0(z)$.

Sonst gehe in Endlosschleife.

Orakelfrage: $\langle M' \rangle \in \overline{C(S)}$? Falls ja: akz. Sonst lehne ab.

Haben gezeigt: SELFREJECT $\leq_T \overline{C(S)}$.

$\Rightarrow$ $\overline{C(S)}$ ist unentscheidbar. $\Longrightarrow$ $C(S)$ ist unentscheidbar.

Für $\Omega \notin S$ kann analog mit einem $f_0 \in S$ argumentiert werden.

(Dort direkt Reduktion auf $C(S)$ statt auf $\overline{C(S)}$.)

Sei S eine nicht-triviale Menge von berechenbaren Funktionen.

Dann ist $C(S) := \{ \langle M \rangle \mid M \text{ berechnet eine Funktion aus } S \}$ unentscheidbar.

Was sagt der Satz von Rice aus?

Es kann algorithmisch nicht beantwortet werden, welche Fkt. ein gegebenes Programm berechnet bzw. ob die berechnete Fkt. eine gewisse Eigenschaft hat.

$L_1 = \{ \langle M \rangle \mid M$ berechnet die Fkt. $x \mapsto x^2 \}$

Satz von Rice mit $S = \{ x \mapsto x^2 \}$ $\Rightarrow$ L_1 ist unentscheidbar.

$L_2 = \{ \langle M \rangle \mid M$ gibt immer gerade Zahlen aus $\}$

Satz von Rice mit $S = \{ f \mid f$ berechenbar und $\forall x: f(x)$ gerade $\}$

$\Rightarrow$ L_2 ist unentscheidbar.

$L_3 = \{ \langle M \rangle \mid M$ hält auf min. 1000 Eingaben $\}$

Satz von Rice mit $S = \{ f \mid f$ berechenbar und auf ≥ 1000 Eingaben def. $\}$

$\Rightarrow$ L_3 ist unentscheidbar.

Satz von Rice nur anwendbar, wenn es um Eigenschaften der von einer Maschine berechneten Fkt. geht.

Nicht anwendbar, wenn es um eine Eigenschaft der Maschine geht, die nicht nur vom Ausgabeverhalten der Maschine abhängt.

$$\{\langle M\rangle \mid M \text{ hat min. 100 Zustände und hält auf Eingabe } \langle M\rangle\}$$

Sei S eine nicht-triviale Menge von berechenbaren Funktionen.

Dann ist $C(S) := \{ \langle M \rangle \mid M$ berechnet eine Funktion aus $S \}$ unentscheidbar.

$L := \{ \langle M \rangle \mid M$ akzeptiert min. 1000 Eingaben $\}$

$L_3 = \{ \langle M \rangle \mid M$ hält auf min. 1000 Eingaben $\}$

bereits gesehen: L_3 ist unentscheidbar (Satz von Rice)

Zeige noch $L_3 \leq_T L$. Reduktion wie folgt:

Eingabe: $\langle M \rangle$

Konstruiere Maschine M', die sich wie M verhält, deren akz. Zustand der Halte-Zustand von M ist und deren abl. Zustand ein neuer Zustand ist

Orakelfrage: $\langle M' \rangle \in L$? Falls ja: akz. Sonst: lehne ab