$A \subset \Sigma^*$ heißt **entscheidbar**, wenn die charakteristische Funktion von A berechenbar ist.

$$\chi_A : \Sigma^* \to \{0,1\} : \quad \chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

$A \subset \Sigma^*$ heißt **entscheidbar**, wenn die charakteristische Funktion von A berechenbar ist.

$$\chi_A : \Sigma^* \to \{0,1\} : \quad \chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

Satz:

A ist enscheidbar $\Leftrightarrow$ A ist semi-entscheidbar und $\overline{A}$ ist semi-entscheidbar

---

Kodierung von Turing Maschinen als binäre Strings

$M = (\Sigma,\Gamma,\#,Q,s,F,\Delta)$

Σ	Eingabealphabet
$\Gamma \supset \Sigma$	Bandalphabet
$\# \in \Gamma \setminus \Sigma$	Leerzeichen (Inhalt von nicht initialisierten Zellen)
Q	Zustandsmenge (endlich)
$s \in Q$	Startzustand
$F \subset Q$	Endzustände
$\Delta \subset (Q\times\Gamma) \times (Q\times\Gamma\times\{L,B,R\})$	Rechenschrittregeln

derzeitiger Zustand, derzeitig gelesenes Symbol, neuer Zustand, geschriebenes Symbol, Kopfbewegung L = links R = rechts B = bleibe

---

Kodierung von Turing Maschinen als binäre Strings

$M = (\Sigma,\Gamma,\#,Q,s,F,\Delta)$ **o.B.d.A**

$\Sigma = \{ a_2, \cdots, a_t \}$
$\Gamma = \{ a_1, \cdots, a_t, \cdots, a_g \}$
$\# = a_1$
$Q = \{ q_1, \cdots, q_m \}$
$s = q_1$
$F = \{ q_2 \}$
$\Delta \subset (Q\times\Gamma) \times (Q\times\Gamma\times\{L,B,R\})$ $\quad L = \mu_1, B = \mu_2, R = \mu_3$
$\Delta = \{ \delta_1, \cdots, \delta_D \}$

Kodierung von Turing Maschinen als binäre Strings

$M = (\Sigma,\Gamma,\#,Q,s,F,\Delta)$ **o.B.d.A**

$\Sigma = \{ a_1, \cdots, a_t \}$
$\Gamma = \{ a_1, \cdots, a_t, \cdots, a_g \} \cup \{\#\}$
$Q = \{ q_1, \cdots, q_m \}$
$s = q_1$
$F = \{ q_2 \}$
$\Delta \subset (Q\times\Gamma) \times (Q\times\Gamma\times\{L,B,R\})$ $\quad L = \mu_1, B = \mu_2, R = \mu_3$
$\Delta = \{ \delta_1, \cdots, \delta_D \}$

$\mathrm{cod}(M) = \mathrm{cod}(\delta_1)\, \mathrm{cod}(\delta_2) \cdots \mathrm{cod}(\delta_D)$

Kodierung von Turing Maschinen als binäre Strings

$M = (\Sigma,\Gamma,\#,Q,s,F,\Delta)$ **o.B.d.A**

$\Sigma = \{ a_1, \cdots, a_t \}$
$\Gamma = \{ a_1, \cdots, a_t, \cdots, a_g \} \cup \{\#\}$
$Q = \{ q_1, \cdots, q_m \}$
$s = q_1$
$F = \{ q_2 \}$
$\Delta \subset (Q\times\Gamma) \times (Q\times\Gamma\times\{L,B,R\})$ $\quad L = \mu_1, B = \mu_2, R = \mu_3$
$\Delta = \{ \delta_1, \cdots, \delta_D \}$

$\mathrm{cod}(M) = \mathrm{cod}(\delta_1)\, \mathrm{cod}(\delta_2) \cdots \mathrm{cod}(\delta_D)$

$\mathrm{cod}(\, (q_h,a_i,q_j,a_k,\mu_p)\,) = 0\, 1^h\, 0\, 1^i\, 0\, 1^j\, 0\, 1^k\, 0\, 1^p\, 0$

Dekodieren: binäre Strings als Turing Maschinen interpretieren

$w \in \{0,1\}^*$

$M_w = \begin{cases} M \text{ mit } cod(M)=w & \text{falls } w \text{ eine legale TM Kodierung} \\ \text{TM die nichts akzeptiert} & \text{sonst} \end{cases}$

Dekodieren: binäre Strings als Turing Maschinen interpretieren

$w \in \{0,1\}^*$

$M_w = \begin{cases} M \text{ mit } cod(M)=w & \text{falls } w \text{ eine legale TM Kodierung} \\ \text{TM die nichts akzeptiert} & \text{sonst} \end{cases}$

Es gilt: $M_{cod(M)} \approx M$

Kodieren beliebiger Strings als binäre Strings

$\Sigma = \{a_1, \cdots, a_t\}$

$x \in \Sigma^*$ mit $x = a_{i_1} a_{i_2} a_{i_3} \cdots a_{i_n}$

$cod(x) = 0\ 1^{i_1}\ 0\ 1^{i_2}\ 0\ 1^{i_3}\ 0\ 1^{i_4} \cdots 0\ 1^{i_n}\ 0$

Dekodieren: Interpretation binärer Strings als allgemeine Strings

$decod(\ 0\ 1^{i_1}\ 0\ 1^{i_2}\ 0\ 1^{i_3}\ 0\ 1^{i_4} \cdots) = a_{i_1} a_{i_2} a_{i_3} a_{i_4} \cdots$

Es gilt: $decod(\ cod(x)\ ) = x$

Universelle Turingmaschine

Satz: Es gibt eine Turingmaschine

$U = (\ \{0,1,\$\}\ ,\ \{0,1,\#\}\ ,\ \#\ ,\ Q_U\ ,\ s\ ,\ F\ ,\ \Delta_U\ )$

die sich bei Eingabe $cod(M)\ \$\ cod(x) \in \{0,1,\$\}^*$,
mit M eine Turingmaschine und x ein String,
genau so verhält wie M mit Eingabe x.

D.h. U hält genau dann, wenn M auf Eingabe x
hält, und wenn in diesem Fall M den Bandinhalt
z hinterlässt, dann hinterlässt U den Bandinhalt
cod(z).

Anmerkung: Das bedeutet wirklich, U ist programmierbar.

UNIVerselle Sprache

$UNIV = \{\ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid decod(x) \in L(M_w)\ \}$

UNIVerselle Sprache

$\text{UNIV} = \{ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$

UNIV = L(U), also UNIV ist Turing-akzeptierbar

UNIVerselle Sprache

$\text{UNIV} = \{ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$

UNIV = L(U), also UNIV ist Turing-akzeptierbar

Ist UNIV entscheidbar??

UNIVerselle Sprache

$\text{UNIV} = \{ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$

UNIV = L(U), also UNIV ist Turing-akzeptierbar

Ist UNIV entscheidbar??

Ist $\overline{\text{UNIV}}$ auch Turing-akzeptierbar??

Selbst-Akzpetierende Maschinen

$\text{SAM} = \{ w \in \{0,1\}^* \mid w \in L(M_w) \}$

$\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:
Es gibt **keine** TM M mit $L(M) = \overline{\text{SAM}}$.

Satz: $\overline{\mathrm{SAM}} = \{\, w \in \{0,1\}^* \mid w \notin L(M_w) \,\}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:
Es gibt **keine** TM M mit $L(M) = \overline{\mathrm{SAM}}$.

Für **jede** TM M gilt $L(M) \neq \overline{\mathrm{SAM}}$.

Satz: $\overline{\mathrm{SAM}} = \{\, w \in \{0,1\}^* \mid w \notin L(M_w) \,\}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:
Es gibt **keine** TM M mit $L(M) = \overline{\mathrm{SAM}}$.

Für **jede** TM M gilt $L(M) \neq \overline{\mathrm{SAM}}$.

Für **jedes** $w \in \{0,1\}^*$ gilt $L(M_w) \neq \overline{\mathrm{SAM}}$.

Satz: $\overline{\mathrm{SAM}} = \{\, w \in \{0,1\}^* \mid w \notin L(M_w) \,\}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:
Es gibt **keine** TM M mit $L(M) = \overline{\mathrm{SAM}}$.

Für **jede** TM M gilt $L(M) \neq \overline{\mathrm{SAM}}$.

Für **jedes** $w \in \{0,1\}^*$ gilt $L(M_w) \neq \overline{\mathrm{SAM}}$.

$L(M_w)$ und $\overline{\mathrm{SAM}}$ unterscheiden sich in w !!

$w \notin L(M_w) \;\Leftrightarrow\; w \in \overline{\mathrm{SAM}}$

Satz: $\mathrm{SAM} = \{\, w \in \{0,1\}^* \mid w \in L(M_w) \,\}$
ist **nicht** entscheidbar.

Satz: $\mathrm{SAM} = \{\, w \in \{0,1\}^* \mid w \in L(M_w) \,\}$
ist **nicht** entscheidbar.

Beweis:

Wäre SAM entscheidbar, dann müsste

SAM TM akzeptierbar und auch

$\overline{\mathrm{SAM}}$ TM akzeptierbar sein.

Widerspruch.

Analog: „spezielles Halteproblem“

Ob eine SML Funktion mit
seinem eigenen Programmtext als Argument
terminiert oder nicht,
ist im Allgemeinen **nicht entscheidbar**.

UNIVerselle Sprache

UNIV = $\{ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid decod(x) \in L(M_w) \}$

Selbst-Akzeptierende Maschinen

SAM = $\{ w \in \{0,1\}^* \mid w \in L(M_w) \}$

$\overline{SAM}$ = $\{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$

SAM **nicht entscheidbar**, da $\overline{SAM}$ **nicht Turing akzeptierbar.**

Reduktionen

Def.: Man sagt
"Sprache $A \subset \Sigma^*$ reduziert sich auf Sprache $B \subset \Gamma^*$"
"A ist leichter als B"
und schreibt
$A \preceq B$,
wenn es eine überall berechenbare Funktion $F: \Sigma^* \to \Gamma^*$ gibt,
sodass für alle $x \in \Sigma^*$ gilt
$x \in A \Leftrightarrow F(x) \in B$

Satz: Wenn $A \preceq B$ dann gilt:
(i) B entscheidbar ⇒ A entscheidbar
(ii) A nicht entscheidbar ⇒ B nicht entscheidbar

Satz: Wenn $A \preceq B$ dann gilt:
(i) B entscheidbar ⇒ A entscheidbar
(ii) A nicht entscheidbar ⇒ B nicht entscheidbar

Anwendung: UNIV ist nicht entscheidbar, denn
$SAM \preceq UNIV$,und SAM nicht entscheidbar.

SAM = $\{ w \in \{0,1\}^* \mid w \in L(M_w) \}$
UNIV = $\{ w\$x \in \{0,1\}^*\$\{0,1\}^* \mid decod(x) \in L(M_w) \}$

$w \overset{F}{\mapsto} w\$cod(w)$ $\qquad w \in SAM \Leftrightarrow w\$cod(w) \in UNIV$

u beliebiger, fester String

$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$

Satz: Für jedes u ist ACC_u **nicht** entscheidbar.

u beliebiger, fester String

$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$

Satz: Für jedes u ist ACC_u **nicht** entscheidbar.

Beweis: $UNIV \preceq ACC_u$ (und UNIV ist nicht entscheidbar)

Reduktionsfunktion: w\$x wird umgewandelt in Beschreibung einer TM $Z_{w,x}$

$Z_{w,x}$: Eingabe y
1. Lasse M_w mit Eingabe decod(x) laufen.
2. If y=u then akzeptiere, sonst divergiere

u beliebiger, fester String

$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$

Satz: Für jedes u ist ACC_u **nicht** entscheidbar.

Beweis: $UNIV \preceq ACC_u$ (und UNIV ist nicht entscheidbar)

Reduktionsfunktion: w\$x wird umgewandelt in Beschreibung einer TM $Z_{w,x}$

$Z_{w,x}$: Eingabe y

1. Lasse M_w mit Eingabe decod(x) laufen.
2. If $y=u$ then akzeptiere, sonst divergiere

w\$x $\in$ UNIV $\Rightarrow$ M_w akzeptiert decod(x) $\Rightarrow$ $Z_{w,x}$ akzeptiert u $\Rightarrow$ cod($Z_{w,x}$) $\in ACC_u$

u beliebiger, fester String

$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$

Satz: Für jedes u ist ACC_u **nicht** entscheidbar.

Beweis: $UNIV \preceq ACC_u$ (und UNIV ist nicht entscheidbar)

Reduktionsfunktion: w\$x wird umgewandelt in Beschreibung einer TM $Z_{w,x}$

$Z_{w,x}$: Eingabe y

1. Lasse M_w mit Eingabe decod(x) laufen.
2. If $y=u$ then akzeptiere, sonst divergiere

w\$x $\in$ UNIV $\Rightarrow$ M_w akzeptiert decod(x) $\Rightarrow$ $Z_{w,x}$ akzeptiert u $\Rightarrow$ cod($Z_{w,x}$) $\in ACC_u$

w\$x $\notin$ UNIV $\Rightarrow$ M_w akzept. nicht decod(x) $\Rightarrow$ $Z_{w,x}$ akzept. u nicht $\Rightarrow$ cod($Z_{w,x}$) $\notin ACC_u$