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

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

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

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

Satz:

$$A \text{ ist enscheidbar} \Leftrightarrow \begin{cases} A \text{ ist semi-entscheidbar und} \\ \overline{A} \text{ ist semi-entscheidbar} \end{cases}$$

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**

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

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

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

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

Dekodieren: binäre Strings als Turing Maschinen interpretieren

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

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

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

Kodieren beliebiger Strings als binäre Strings

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

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

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

Dekodieren: Interpretation binärer Strings als allgemeine Strings

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

Es gilt: $\mathrm{decod}(\mathrm{cod}(x)) = x$

Universelle Turingmaschine

Satz: Es gibt eine Turingmaschine

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

die sich bei Eingabe $\mathrm{cod}(M)\ \$\ \mathrm{cod}(x) \in \{0,1,\$\}^*$,
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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Beweis:

Wäre SAM entscheidbar, dann müsste

SAM TM akzeptierbar und auch

$\overline{SAM}$ 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**.

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

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

Beweisidee: Nimm an, es gäbe TM G, die UNIV entscheidet.
Konstruiere daraus TM H, die SAM entscheidet. Widerspruch.

H: Eingabe w
1. ersetze Bandinhalt w durch w\$cod(w)
2. lasse G auf diesen Bandinhalt laufen

H akzeptiert (verwirft) w\$cod(w) g.d.w. G akzeptiert (verwirft) w

$w \in SAM \Leftrightarrow w\$cod(w) \in UNIV$

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

Analog: „allgemeines Halteproblem"

Ob eine gegebene SML Funktion für ein gegebenes Argument
terminiert oder nicht,
ist im Allgemeinen **nicht entscheidbar**.