

Entscheidungsproblem: $f: \Sigma^* \rightarrow \{0,1\}$, charakteristische Fkt von $L: \{w \in \Sigma^* | f(w)=1\}$

Relation $R =$ alle Wörter aus Wertebereich, die als Ausgabe 1 haben

$$\gamma_4(\underline{224}) = 4 \cdot 4^0 + 2 \cdot 4^1 + 2 \cdot 4^2 = 44 \quad \text{Position des Wortes } \underline{224}$$
$$\delta_4(44) = [10 \cdot 4 + 4, 2 \cdot 4 + 2, 0 \cdot 4 + 2] = \underline{224} \quad \text{Wort an Position } 44$$

Lsg eines alg. Problems mit Alg, falls er, gestartet mit w nach endl. Schritten mit $f(w)$ terminiert

Alg = verarbeitet schrittweise Wörter, eindeutiger & endlicher Text, Ausgabe oder nicht terminiert

Fkt berechenbar = Alg existiert: F total = immer eine Ausgabe

Eingabe \in DEF \rightarrow Alg macht Ausgabe

Eingabe \in DEF \rightarrow Alg terminiert nicht

f berechenbar \Leftrightarrow Turing-Berechenbar

TM hat die Form $M = (Q, \Sigma, \Gamma, \delta, q_0, q_s)$ $\delta(p,a) = (a', R/N/L, q)$

Konvention: Fehlt für ein Paar (q,a) die entspr. Turingzeile: $q a a N q_s$

Konvention: $f(w_1, \dots, w_n)$ wird TM mit $q_0 w_1 \sqcup w_2 \sqcup \dots \sqcup w_n$ gestartet

Relation entscheidbar = Alg existiert: Alg terminiert immer, und zwar mit 0 oder 1

Relation semientscheidbar = (wie entscheidbar, nur) Alg terminiert nicht (bei Ausgabe 0)

W entscheidbar \leftrightarrow W semientscheidbar

L entscheidbar (semientscheidbar) gdw: es ex. Ein Alg., der L (semi-)entscheidet

$L \subseteq \Sigma^*$ entscheidbar \Leftrightarrow L semientscheidbar und $\Sigma^* \setminus L$ ist auch semientscheidbar

nicht totale Fkt berechenbar \leftrightarrow Graph R_f semientscheidbar

Aufzählungsalg. zählt Elemente irgendeiner Relation auf, ohne Eingabe (evtl mit Wdh)

Diagramm der Java-Programme in kanonischer Reihenfolge (jedes 1 Schritt laufen lassen, dann nächstes, bis eins terminiert)

„berechenbar“ verlangt nur Existenz eines Algorithmus

f Turing-berechenbar(-entscheidbar): es gibt eine TM, die f berechnet (entscheidet)

Alg. (nicht) berechenbar \leftrightarrow Alg. (nicht) Turing-berechenbar

$[[P2]]([[P1]](\dots))$ erst $P1$, dann $P2$ auf Werte angewandt

$[[P3]]^{k2} = P3$ $k2$ -mal ausgeführt

goto: (j, r_1, \dots, r_m) mit $j =$ Zeilennummer, $r_i =$ Werte für X_i

goto_m-berechenbar = es ex. Ein goto-Programm P mit $f_P^n = f$ goto-berechenbar

while: komfortabler, Rechnermodell, wie bei goto

Syntax: Gestalt der Programme als Texte (Wörter)

Semantik: Definition der Wirkung der Programme auf vorgelegte Daten

$X := 0, X_i = X_j, X := X + 1, X := X - 1, X_j := X_i \text{ op } X_k \text{ op} \in \{+, -, \cdot, \div, \text{mod}\}, X > 0, X = 0$

$\text{while}^0 = \text{while-Prog. ohne direkte Wertzuweisungen, nur mit } X := X \pm 1 \text{ und Bed. } X = 0$

while berechenbar \rightarrow while^0 -berechenbar, zu jedem while_m -Prog ex. Ein while_m^0 -Prog. ($m' \geq m$)

loop-Schleifen terminieren, loop durch while ersetzbar, umgekehrt nicht immer, für loop muss Schrittzahl vorher feststehen

WHILE-WHILE⁰-GOTO-TM:

$\text{while}^0 \rightarrow \text{goto}$: Hintereinanderausführen von Programmen, while mit if then else simulieren

goto \rightarrow TM: $q^0 |^{x_1} \sqcup |^{x_2} \sqcup \dots |^{x_m} \sqcup \dots$ dann mit Verschiebung der | und \sqcup

TM \rightarrow while: Zahlen mit | und \sqcup als Dualzahl schreiben, dann mit mod und div rechnen

Äquivalent: f ist Turing-, goto-, while-berechenbar

Turingzeile codiert: $(i, j, j', R/N/L, i') = 0^i 1 0^j 1 0^{j'} 1 0 0/00/000 1 0^{i'} 11$

WP: Stoppt TM bei einem bestimmten Wort? $M:w \rightarrow \text{STOP?}$ Das WP ist unentscheidbar

Cantor: Menge der unendlichen 0-1-Folgen ist nicht abzählbar (Diagonale negieren)

Problem $P = (\text{Inst}_P, \text{Pos}_P)$ und Q gilt: $P \leq Q$ (P auf Q reduzierbar), falls $f: \text{Inst}_P, \text{Pos}_P \text{ ex für } x \in \text{Inst}_P: x \in \text{Pos}_P \Leftrightarrow f(x) \in \text{Pos}_Q$

Wenn $P \leq Q$ und P unentscheidbar, dann auch Q unentscheidbar, Q mindestens so schwer wie P

HP: $M:\varepsilon \rightarrow \text{STOP?}$ M' schreibt w aufs Band, dann zum Anfang und M auf w arbeiten lassen

ÄP: Berechnen M^1 und M^2 dieselbe Funktion?

TM-Problem ist nichttrivial, wenn weder $\text{Inst}_P = \text{Pos}_P$ noch $\text{Pos}_P = \emptyset$

Rice: jedes semantische, nicht-triviale Entscheidungsproblem über TMs ist unentscheidbar

Domino, MPCP(1), PCP unentscheidbar Beweis über $\text{WP} \leq \text{MPCP}$ mit Eindeutigkeit des Aufbau

L_{WP} ist (Turing-)semientscheidbar, L aufzählbar $\Rightarrow L \leq L_{\text{WP}}$

Klasse P : Probleme, die man in polynomialer Zeit lösen kann, Sprache nennt man „effizient entscheidbar“ $M O(p(n))$ -zeitbeschränkt

These: P enthält genau die alg. Probleme, die praktisch lösbar sind

NTM ist $t(n)$ -zeitbeschränkt, wenn sie nach weniger als $t(|w|)$ Schritten stoppt

Jede NTM lässt sich durch DTM simulieren. Aufwand $2^{O(t(n))}$

NP-Sprachen werden durch eine $t(n)$ -NTM entschieden. Zeitaufwand ist polynomial in w

$L \in \text{NP}$ gdw es ex eine polynomial zeitbeschränkte NTM, die w akz. Gdw $w \in L$