

**Entscheidbarkeit****Nichtentscheidbare Probleme**

*Welche von denen gehören zu den Semientscheidbaren?*

- Diagonalsprache  $L_d := \{\omega_i : M_i \text{ akzeptiert } \omega_i \text{ nicht}\}$
- Kontextfreie Sprachen:  $L(G_1) \subseteq L(G_2)?$ , Mehrdeutigkeit (mehrere Ableitungen zum gleichen Wort)?,  $L(G)$  kontextfrei?,  $L(G)$  regulär?,  $L(G)$  det. kontextfrei?
- Diophantische Gleichungen: multivariates Polynome  $p$ , Koeffizienten ganzzahlig:  $\exists x_1, \dots, x_n \in \mathbb{Z} : p(x_1, \dots, x_n) = 0?$

**Semientscheidbare Probleme**

*Definition:* Es ex. eine TM, die genau die Wörter aus  $L$  akzeptiert, sonst aber nicht halten muss.

*Beispiele:*

- Halteproblem  $H := \{wv | T_w \text{ hält auf der Eingabe } v\}$
- Universelle Sprache  $L_u := \{wv | v \in L(T_w)\}$
- Postisches Korrespondenzproblem Geg: Menge von Wortpaaren  $(x_i, y_i) \in (\Sigma^+ \times \Sigma^+)^*$ . Gibt es eine endl. Folge von Indizes:  $x_{i_1} \dots x_{i_n} = y_{i_1} \dots y_{i_n}?$
- Komplement der Diagonalsprache

**Entscheidbare Probleme**

*Definition:* Es ex. eine TM, die genau die Wörter aus  $L$  akzeptiert und bei jeder Eingabe hält.

*Beispiele:*

*Presburger Arithmetik:* eingeschränkte prädikatenlogische Formeln.

Typ	$\in$	$\emptyset$	$=$	$\cap = \emptyset$
CH-3	J	J	J	J
Det. KF	J	J	J	N
CH-2	J	J	N	N
CH-1	J	N	N	N
CH-0	N	N	N	N

$$\mathcal{NP} \stackrel{?}{=} \mathcal{P}$$

**Definitionen  $\mathcal{NP}$ ,  $\mathcal{P}$** 

- $\text{time}_M(w) :=$  Anzahl Rechenschritte einer TM  $M$  bei Eingabe  $w$
- $\text{TIME}(f(n)) := \{L \in \Sigma^* : \exists \text{ TM } M : L(M) = L \text{ und } \forall w \in L(M) : \text{time}_M(w) \leq f(|w|)\}$
- $\mathcal{P} := \cup_{\text{Polynom } p} \text{TIME}(p(n))$
- $\text{ntime}_M(w) = \begin{cases} \min\{n : P = (s)w \Rightarrow^u n(f)v, \\ f \in F\} \text{ falls } w \in L(M) \\ 0, \text{ sonst} \end{cases}$
- $\text{NTIME}(f(n)) = \{L \in \Sigma^* : \exists \text{ NTM } M : L(M) = L \text{ und } \forall w \in L(M) : \text{ntime}_M(w) \leq f(|w|)\}$
- $\mathcal{NP} = \cup_{\text{Polynom } p} \text{NTIME}(p(n))$
- $V \in \mathcal{NP}$ -hart  $\Leftrightarrow \forall V' \in \mathcal{NP} : V' \leq_p V$
- $V \in \mathcal{NP}$ -vollständig  $\Leftrightarrow V \in \mathcal{NP} \cap \mathcal{NP}$ -hart

*Probleme siehe Tabelle Achtung:* „Rucksack“ ist Knapsack bei Sanders, aber Subsetsum bei Schöning.

**Grammatiken**

Sei  $G = (V, \Sigma, P, S)$ ,  $\forall l \rightarrow r \in P :$

**Def. CH-0 (rekursiv aufzählbar)** beliebig

*Wortproblemkomplexität:* semientscheidbar

**Definition CH-1 (längenbeschr.)**

$|l| \leq |r|$ . Sonderregel für  $\varepsilon$ -Produktion nur bei  $S$

*Beispiel:*  $a^n b^n c^n$

*Wortproblemkomplexität:*

$|\Sigma|^{|Q(n)|}$ , NP-hart **Entscheidbare**

*Probleme:*  $L(G) = \emptyset$ ,  $|L(G)| \neq \infty$ ,  $L(G) = \Sigma^*$

**Definition CH-2 (kontextfrei)**

CH-1 und  $l \in V$

*Beispiel:*  $a^n b^n$

*Wortproblemkomplexität:*  $O(n^3)$

*Pumpinglemma:*  $L$  kontextfrei

$\Rightarrow \exists n \in \mathbb{N} : \forall z \in L, |z| > n :$

$\exists u, v, w, x, y :$

$z = uvwx \wedge |vx| \geq 1 \wedge |vwx| \leq$

$n \wedge \forall i \in \mathbb{N}_0 : uv^i wx^i y \in L$

*Odgens Lemma:*  $L$  kontextfrei

$\Rightarrow \exists n \in \mathbb{N} : \forall z \in L, |z| \geq n :$  Wenn

wir in  $z$  mindestens  $n$  Buchstaben

markieren

$\exists u, v, w, x, y : z = uvwx$ , dass von

den mindestens  $n$  markierten

Buchstaben mindestens einer zu  $vx$

gehört und höchstens  $n$  zu  $vwx$

gehören und  $\forall i \geq 0 : uv^i wx^i y \in L$ .

*Chomsky-Normalform:* falls gilt:

$P \subseteq (V \times \Sigma) \cup (V \times VV)$

1. Terminale in eigene Regeln.
2. Regeln mit rechts  $> 2$  Nicht-Terminale aufsplitten
3.  $\varepsilon$ -Produktionen entfernen
4. Kettenproduktionen entfernen

**Definition Det. KF**

*Bitte noch eintragen*

*Wortproblemkomplexität:*  $O(n)$

**Definition CH-3 (regulär)**

CH-2 und  $r \in \Sigma \cup \Sigma V$

*Beispiel:*  $a^* b^*$

*Wortproblemkomplexität:*  $O(n)$

*Pumpinglemma:*  $L$  regulär

$\Rightarrow \exists n \in \mathbb{N} : \forall w \in L, |w| > n :$

$\exists u, v, x : w = uvx \wedge |v| \geq 1 \wedge |uv| \leq$

$n \wedge \forall i \in \mathbb{N}_0 : uv^i x \in L$

*Reguläre Ausdrücke:*

Beispiel:  $(\emptyset \cup \varepsilon)^* abc^+$

**Nerode-Relation**

*Für Sprache  $L$ :*

$R_L := \{(x, y) \in \Sigma^* \times \Sigma^* : \forall z \in \Sigma^* :$

$xz \in L \Leftrightarrow yz \in L\}$

*Für Automat  $M$ :*  $R_M := \{(x, y) \in$

$\Sigma^* \times \Sigma^* : \delta^*(s, x) = \delta^*(s, y)\}$

*Verfeinerung:*  $R$  verfeinert

$R' \Leftrightarrow R \subseteq R'$

*Satz:*  $L$  regulär  $\Leftrightarrow \text{index}(R_L) \neq \infty$

*Satz:*  $q \not\equiv r \Leftrightarrow \exists z \in \Sigma^* : \delta(q, z) \in$

$F \not\equiv \delta(r, z) \in F$

*Beispielanwendung:*  $a^n b^n$  ist nicht

regulär, denn  $\{a^n\}, n \in \mathbb{N}$  sind

unendlich verschiedene

Äquivalenzklassen, denn für  $i \neq j$

ist  $a^i b^i \in L$ , aber  $a^j b^i \notin L$ , also

$\{a^i\} \neq \{a^j\}$ .

**Abschlusseigenschaften**

Typ	$\cap$	$\cup$	$\cdot$	$*$
CH-3	J	J	J	J
Det. KF	N	N	J	N
CH-2	N	J	N	J
CH-1	J	J	J	J
CH-0	J	J	N	J
semient.	J	J	N	J
entsch.	J	J	J	J

**Automaten-Zuordnung**

Typ	Automat
CH-3	Endlicher Automat (NEA, DEA)
Det. KF	det. Kellerautomat (DKellerA)
CH-2	Kellerautomat (NKellerA)
CH-1	linear beschr. Automat (NLBTM)
CH-0	Turingmaschine (TM)

**Automatenäquivalenz**

$\varepsilon$ NEA ist zu  $\bar{\varepsilon}$ NEA ist zu DEA und

NTM ist zur DTM äquivalent.

NKellerA ist zu DKellerA nicht

äquivalent. Äquivalenz von NLBTM

und DLBTM ist noch nicht

bewiesen.

**Automaten**

Mealy-Automat: Ausgabe beim

Übergang, Moore-Automat:

Ausgabe beim Zustand.

**DTM**

$T = (Q, \Sigma, \Gamma, \delta, s, F), \delta : Q \times \Gamma \rightarrow$

$Q \times \Gamma \times \{L, R, N\}$

*sie hält in  $q(q)av$ :*

$\Leftrightarrow \delta(q, a) = (q, a, N)$ .

*Konvention:*

$\forall q \in F : \forall a \in \Gamma : \delta(q, a) = (q, a, N)$

*sie akzeptiert  $w$ :*  $\Leftrightarrow (s)w$  hält nach

endlich vielen Übergängen in

$x(f)y, f \in F$ .  $y$  ist die Ausgabe.

*rekursiv aufzählbar*

*(semientscheidbar):*  $\exists T : T$

akzeptiert  $L$

*rekursiv (entscheidbar):*  $\exists T : T$

akzeptiert  $L \wedge \forall w \in \Sigma^* : T$  hält.

**DTM-Varianten**

Mehrere Bänder, mehrere Köpfe,

mehrere Dimensionen – alles gleich

mächtig wie DTM.

**NTM**

$T = (Q, \Sigma, \Gamma, \delta, s, F), \delta : Q \times \Gamma \rightarrow$

$2^{Q \times \Gamma \times \{L, R, N\}}$

*sie hält wie:* DTM

*sie akzeptiert  $w$ :*  $\Leftrightarrow \exists$  Folge von

Konfigurationen

$s(w) \rightarrow \dots \rightarrow x(f)y, f \in F$

**Gödelnummer-Code**

1. Kodiere  $\delta$ :

$\delta(q_i, a_j) = (q_r, a_s, d_t) \rightarrow$

$0^i 10^j 10^r 10^s 10^t$

wobei

$d_t \in \{d_1 = L, d_2 = R, d_3 = N\}$

2. Die TM wird dann kodiert durch:

$111u_1 11u_2 11\dots 11u_z 111$

mit  $u_i$  die möglichen Übergänge in

bel. Reihenfolge.

**NLBTM**

NTM  $T = (Q, \Sigma, \Gamma, \delta, s, F) : \forall a =$

$a_1, \dots, a_n \in \Sigma^+ : a \not\stackrel{*}{\rightarrow} \alpha(q)\beta$  mit

$|\alpha\beta| < n$

**Nichtdet. Kellerautomat**

$K = (Q, \Sigma, \Gamma, \delta, s, \#), \delta :$

$Q \times (\Sigma \cup \varepsilon) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

*er akzeptiert  $w$ :*  $\Leftrightarrow \exists$  Folge von

Konfigurationen

$(s, w, \#) \rightarrow \dots \rightarrow (q, \varepsilon, \varepsilon), q \in Q$

beliebig.

**Det. Kellerautomat**

$K = (Q, \Sigma, \Gamma, \delta, s, \#), \delta :$

$Q \times (\Sigma \cup \varepsilon) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$  mit

$\forall q \in Q, a \in \Sigma, A \in \Gamma :$

$|\delta(z, a, A)| + |\delta(z, \varepsilon, A)| \leq 1$

*er akzeptiert  $w$ :*  $\Leftrightarrow \exists$  Folge von

Konfigurationen

$(s, w, \#) \rightarrow \dots \rightarrow (f, \varepsilon, \varepsilon), f \in F$

**Automatenminimierung**

(Für endliche Automaten)

1. nicht erreichbare Zustände weg
2. Tabelle aller Zustandspaare  $\{z, z'\}$  mit  $z \neq z'$  ( $z_1$  bis  $z_k$  links,  $z_0$  bis  $z_{k-1}$  unten)
3. Markieren der Zustandspaare mit  $z \in F$  und  $z \notin F$  oder umgekehrt.
4. Betrachte unmarkierte Paare  $\{z, z'\}$ . Wenn  $\{\delta(z, a), \delta(z', a)\}$  für mind. ein  $a \in \Sigma$  bereits markiert, markiere  $\{z, z'\}$ .
5. Wiederhole 4. bis keine Änderung mehr.
6. Unmarkierte Paare können verschmolzen werden.

**DEA  $\rightarrow$  reg. ex.**

Betrachte  $L_{ij}^m := \{w : \Sigma^* :$

Beim Verarbeiten von  $w$  geht  $A$

vom Zustand  $i$  nach  $j$  und dabei

höchstens durch  $m\}$ .

Es gilt  $L_{ij}^{m+1} = L_{ij} \cup$

$(L_{i, m+1}^m (L_{m+1, m+1}^m)^* L_{m+1, j}^m)$

So weitermachen, bis man  $L_{sf}^n$  hat

( $s$  Startzustand,  $f$  Endzustand,  $n$

Zahl der Zustände).

**NEA  $\rightarrow$  DEA**

Potenzmengenkonstruktion.

Knotenmengen sind Endzustände,

wenn einer ihrer enthaltenen

Zustände ein Endzustand ist.

**Whileprogramm**

$\mathbb{N}$  main( $\mathbb{N}x_1, \dots, x_k$ ) $\{$

$\mathbb{N}x_0 = 0; \mathbb{N}x_{k+1} = 0; \dots$

body;

return  $x_0$ ;

$\}$

body  $\in \{ \text{Sequenz } '; \text{while}(x_i \neq 0) :$

Schleife,  $x_i := x_i + c$  wobei

$c \in \{-1, 0, 1\}$  und  $0 - 1 := 0\}$

„loop“-Konstrukte im body erlaubt,

aber redundant.

**Loopprogramm**

$\mathbb{N}$  main( $\mathbb{N}x_1, \dots, x_k$ ) $\{$

$\mathbb{N}x_0 = 0; \mathbb{N}x_{k+1} = 0; \dots$

body;

return  $x_0$ ;

$\}$

body  $\in \{ \text{Sequenz } '; \text{loop}(x_i) :$

Schleife, wobei schon vor dem

Durchlauf bekannt ist wie oft die

Schleife wiederholt wird,  $x_i := x_i + c$

wobei  $c \in \{-1, 0, 1\}$  und  $0 - 1 := 0\}$

**Ackermannfunktion****Definition**

Function  $a(x, y)$

if  $x = 0$  then return  $y + 1$

if  $y = 0$  then return  $a(x - 1, 1)$

return  $a(x - 1, a(x, y - 1))$

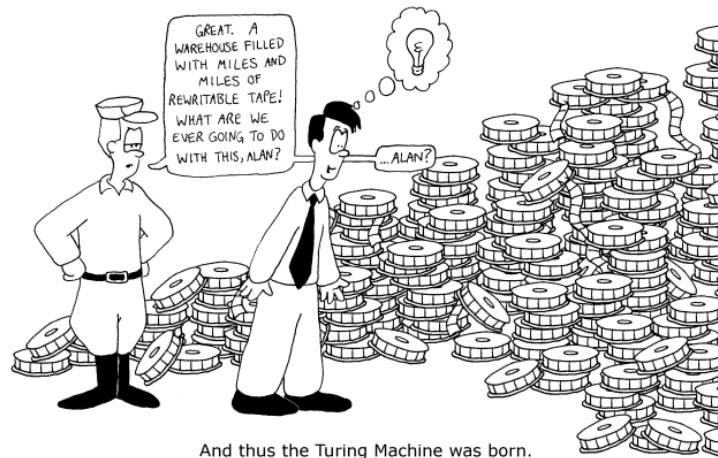
**Eigenschaften**

- $\forall$  Loopprogramm  $P : \exists k : \forall n \in \mathbb{N} : f_P(n) < a(k, n)$
- $y < a(x, y)$
- $a(x, y) < a(x, y + 1)$
- $a(x, y + 1) < a(x + 1, y)$
- $a(x, y) < a(x + 1, y)$
- $a(x, y) \leq a(x', y')$  falls  $x \leq x'$  und  $y \leq y'$

**Pseudopolynomialität**

## NP-Vollständige Probleme

Problem	Gegeben	Gesucht	polyn. red. von
SAT	aussagenlog. Formel	Erfüllbarkeit	TM
3SAT	boolesche Formel in KNF mit 3 Lit. pro Klausel	Erfüllbarkeit	SAT
Set Cover	endl. Menge $M$ und $T_1, \dots, T_k \subseteq M$ , Zahl $n \leq k$	$n$ Mengen $T_{i_1}, \dots, T_{i_n}$ mit $M = \bigcup_{j=1 \dots n} T_{i_j}$	3SAT
Steiner-Tree	Unger. Graph $G = (V, E)$ mit Gewichten $c : E \rightarrow \mathbb{R}^+$ , $V = R$ (Pflicht-) $\cup F$ (Steinerknoten)	Baum $T \subseteq E$ der mit minimalen Kosten alle Pflichtknoten verbindet	3SAT
Clique	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$	Clique $V' \subseteq V$ mit $ V'  \geq k$ , also $\forall i, j \in V', i \neq j$ , gilt: $\{i, j\} \in E$	3SAT
Vertex Cover	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$	überdeckende Knotenmenge $V' \subseteq V$ mit $ V'  \geq k$ , sodass $\forall \{u, v\} \in E: u \in V'$ oder $v \in V'$	Clique
Subset Sum	Zahlen $a_1, \dots, a_k \in \mathbb{N}$ und $W \in \mathbb{N}$	Teilmenge $J \subseteq \{1, \dots, k\}$ mit $\sum_{i \in J} a_i = W$	3SAT
Partition	Zahlen $a_1, \dots, a_k \in \mathbb{N}$	Teilmenge $J \subseteq \{1, \dots, k\}$ mit $\sum_{i \in J} a_i = \sum_{i \notin J} a_i$	Subset Sum
Bin Packing	Behältergröße $b \in \mathbb{N}$ , Behälteranzahl $k \in \mathbb{N}$ , Objekte $a_1, \dots, a_k \leq b$	Abb. $f : \{1, \dots, n\} \rightarrow \{1, \dots, k\}$ , sodass $\forall j = 1, \dots, k: \sum_{f(i)=j} a_i \leq b$	Partition
Knapsack	endl. Menge $M$ , Gewichsfkt. $w : M \rightarrow \mathbb{N}_0$ , Kostenfkt. (Profifkt.) $c : M \rightarrow \mathbb{N}_0$ , $W, C \in \mathbb{N}_0$	$M' \subseteq M$ mit $\sum_{a \in M'} w(a) \leq W$ und $\sum_{a \in M'} c(a) \geq C$	Subset Sum
ILP	Vektor $x = (x_1, \dots, x_n)$ und Bedingungen $a \cdot x R b$ mit $R \in \{\leq, \geq, =\}$ , $a \in \mathbb{Z}^n$ , $b \in \mathbb{Z}$	Gibt es eine Belegung von $x$ , so dass alle Bedingungen erfüllt sind?	Subset Sum
Gericht. Hamiltonkreis	gerichteter Graph $G = (V, E)$	Hamiltonkreis: einfacher Kreis der jeden Knoten genau einmal enthält	3SAT
Hamiltonkreis	ungerichteter Graph $G = (V, E)$	Hamiltonkreis	Gericht. Hamiltonkreis
TSP	Vollständiger Graph $G = (V, V \times V)$ mit Abstands fkt. $d : V \times V \rightarrow \mathbb{R}^+$ und Zahl $k$	Hamiltonkreis $C$ mit Länge $\sum_{(u,v) \in C} d(u, v) \leq k$	Hamiltonkreis
Coloring	ungerichteter Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$	$c : V \rightarrow 1, \dots, k$ mit $\forall \{u, v\} \in E : c(u) \neq c(v)$	3SAT



And thus the Turing Machine was born.

Und für die, die den Taschenrechner vergessen haben:



**Jetzt sogar mit Glückspfennig!**

Dies ist ein <http://mitschriebwiki.nomeata.de/>-Projekt. Mitgewirkt haben: Joachim Breitner, Wenzel Jakob, Martin Kiefel, Lucas Lürich und Jennifer Tesch.