

Nichtdef. TM : δ Relation

Komplexitätsklasse NP: Sprachen,
die von NTM in polynomieller Zeit
entschieden werden können.

Def: polyn. Reduzierbarkeit: $L_1 \leq_p L_2$
 $:\Leftrightarrow \exists f: \Sigma_1^* \rightarrow \Sigma_2^*$ polyn. berechenbar
 $\forall w \in \Sigma_1^* : w \in L_1 \Leftrightarrow f(w) \in L_2$

Satz: $L_1 \leq_p L_2, L_2 \in P \Rightarrow L_1 \in P$

Lemma: $L_1 \leq_p L_2, L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$

Def: Eine Sprache L heißt NP-schwer,
falls $L' \leq_p L \forall L' \in NP$.

Eine NP-schwere Sprache L heißt
NP-vollständig, falls $L \in NP$.

Satz: Es sei L NP-vollständig

i) Falls $L \in P$, so ist $P = NP$.

ii) Falls $L \notin P$, so ist $L' \notin P$ für alle
NP-vollst. Sprachen L' . \square

Satisfiability Problem (SAT)

Gegeben: Boolesche Variablen x_1, \dots, x_n und
 m Klauseln, d.h. Disjunktionen von
Literalen x_i und $\bar{x}_i := \neg x_i, i = 1, \dots, n$

Frage: Gibt es eine Belegung der Variablen,
so dass alle m Klauseln erfüllt sind.

Beispiel: $(x_1 \vee \bar{x}_2 \vee x_3)$

$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee x_2) \wedge (x_2 \vee \bar{x}_4 \vee x_5)$

$$\wedge (x_3 \vee x_4 \vee \bar{x}_5) \quad (u=5, m=4)$$

Satz von Cook: SAT ist NP-vollständig.

Beweis: SAT \in NP: NTM „rät“ erfüllende Belegung von x_1, \dots, x_n und überprüft diese.

Es sei nun $L \in$ NP, z.z.zg: $L \leq_p$ SAT!

Zu L gibt es eine NTM N , die L mit worst-case Rechenzeit $p(n)$ entscheidet (p Polynomfkt.).

$$Q = \{q_0, q_1, \dots, q_e\}, \quad F = \{q_0\}, \quad q_1 \text{ Anfangszust.}$$

$$\Sigma = \{\sigma_1, \dots, \sigma_s\}$$

Zu z.zg. $x \in \Sigma^*$ konstruiere Instanz von SAT $f(x)$, so dass

$$N \text{ akz. } x : \Leftrightarrow x \in L \Leftrightarrow f(x) \in \text{SAT}$$

Zeige: f kann so konstruiert werden und zwar in polyn. Zeit.

$f(x)$ enthält 8 Arten von Klauseln über die folgenden Booleschen Variablen:

$\text{cursor}_{i,t}$ bedeutet „der Cursor zeigt zur Zeit t auf die i -te Stelle des Bands“,

$\text{zust}_{k,t}$ bedeutet „ N ist zur Zeit t im Zustand q_k “,

$\text{wahl}_{r,t}$ bedeutet „ N wählt zur Zeit t die r -te Alternative aus Δ für den nächsten Schritt aus“,

mit $0 \leq i \leq p(|X|)$, $1 \leq j \leq s$, $0 \leq k \leq l$,
 $0 \leq t \leq p(|X|)$, $1 \leq r \leq |Q| \cdot |Z| \cdot 3$

Verwende die folgende Schreibweise
für Boolesche Variablen P_1, \dots, P_a :

$\exists! i : P_i$ bedeutet

$$\left(\bigvee_i P_i \right) \wedge \bigwedge_{i < j} (\bar{P}_i \vee \bar{P}_j)$$

Θ_1 : Zu jeder Zeit t zeigt der Cursor von N auf genau eine Stelle des Bands:

$$\bigwedge_t \exists! i : \text{cursor}_{i,t}$$

Θ_2 : Zu jeder Zeit steht an jeder Stelle des Bands genau ein Zeichen:

$$\bigwedge_{i,t} \exists! j : \text{band}_{i,j,t}$$

Θ_3 : Zu jeder Zeit ist N in genau einem Zustand:

$$\bigwedge_t \exists! k : \text{zust}_{k,t}$$

Θ_4 : Zu jeder Zeit wählt N genau eine der sich bietenden Möglichkeiten für den nächsten Schritt aus:

$$\bigwedge_t \exists! r : \text{wahl}_{r,t}$$

Θ_5 : Zur Zeit 0 steht auf dem Band $\triangleright x \epsilon = \sigma_{j_1} \sigma_{j_2} \dots \sigma_{j_{n+1}} \sigma_{j_{n+2}} \dots \sigma_{j_{p(n)}}$. Der Anfangszustand ist $s = q_1$:

$$\text{zust}_{1,0} \wedge \text{cursor}_{0,0} \wedge \bigwedge_i \text{band}_{i,j_i,0}$$

Θ_6 : Spätestens zur Zeit $p(n)$ ist N im Zustand 'yes' = q_0 :

$$\bigvee_t \text{zust}_{0,t}$$

Θ_7 : Zu jeder Zeit kann nur das Symbol geändert werden, das an der Stelle steht, auf die der Cursor gerade zeigt:

$$\bigwedge_{i,j,t} \left((\text{cursor}_{i,t} \vee \text{band}_{i,j,t} \vee \neg \text{band}_{i,j,t+1}) \wedge (\text{cursor}_{i,t} \vee \neg \text{band}_{i,j,t} \vee \text{band}_{i,j,t+1}) \right)$$

(Beachte, dass $\neg X \Rightarrow (Y \Leftrightarrow Z)$ äquivalent ist zu $(X \vee Y \vee \neg Z) \wedge (X \vee \neg Y \vee Z)$.)

Θ_8 : Zu jeder Zeit t folgt der nächste Schritt der Maschine einer Regel aus der Relation \mathcal{S} (ein gegebenes Paar $(q_k, \sigma_j) \in \mathcal{Q} \times \Sigma$ eröffne die Möglichkeiten $((q_k, \sigma_j), (q_{k_r}, \sigma_{j_r}, D_r)) \in \mathcal{S}, r = 1, \dots, R$):

$$\bigwedge_{i,j,k,r,t} \left(\begin{aligned} &(\neg \text{zust}_{k,t} \vee \neg \text{cursor}_{i,t} \vee \neg \text{band}_{i,j,t} \vee \neg \text{wahl}_{r,t} \vee \text{zust}_{k_r,t+1}) \wedge \\ &(\neg \text{zust}_{k,t} \vee \neg \text{cursor}_{i,t} \vee \neg \text{band}_{i,j,t} \vee \neg \text{wahl}_{r,t} \vee \text{band}_{i,j_r,t+1}) \wedge \\ &(\neg \text{zust}_{k,t} \vee \neg \text{cursor}_{i,t} \vee \neg \text{band}_{i,j,t} \vee \neg \text{wahl}_{r,t} \vee \text{cursor}_{i+\delta_r,t+1}) \end{aligned} \right)$$

wobei

$$\delta_r := \begin{cases} 1 & \text{falls } D_r = \rightarrow, \\ 0 & \text{falls } D_r = -, \\ -1 & \text{falls } D_r = \leftarrow. \end{cases}$$

(Beachte, dass $A_1 \wedge A_2 \wedge A_3 \wedge A_4 \Rightarrow B$ äquivalent ist zu $\neg A_1 \vee \neg A_2 \vee \neg A_3 \vee \neg A_4 \vee B$.)

Man kann sich davon überzeugen, dass es genau dann eine Belegung der Variablen gibt, die alle Klauseln erfüllt, wenn N mit Eingabe x in einem akzept. Endzustand gelangen kann. \square

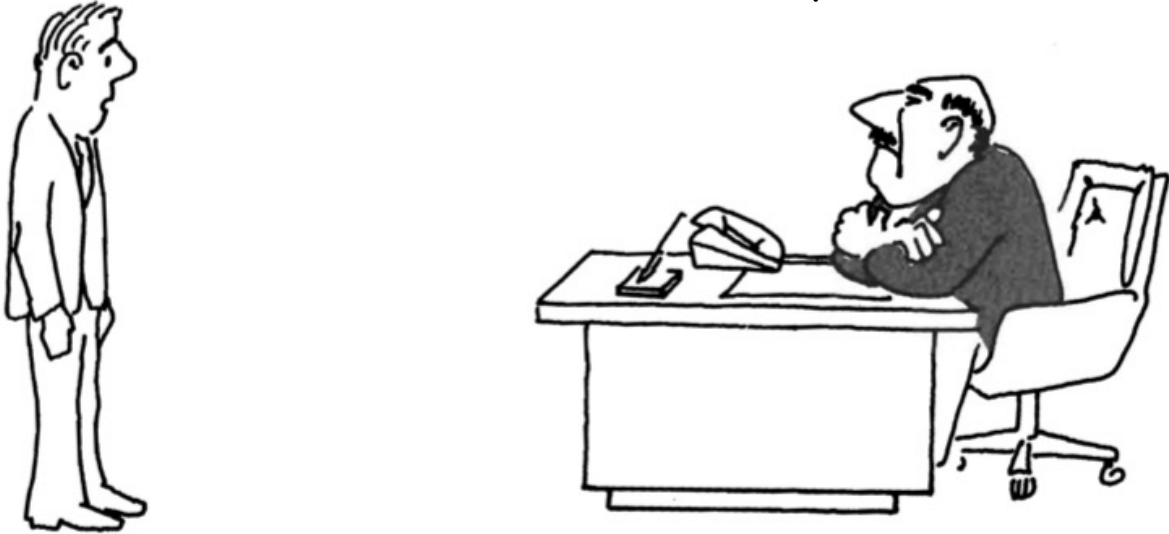
Lemma: Ist $L_2 \in \text{NP}$ und L_1 NP-vollst. und $L_1 \leq_p L_2$, so ist L_2 NP-vollst.

Beweis: Transitivität von " \leq_p ". \square

Satz: $SAT \leq_p CLIQUE$

Korollar: $CLIQUE$ ist NP-vollst. \square

(Bilder aus Buch: Garey & Johnson)



"I can't find an efficient algorithm, I guess I'm just too dumb."



"I can't find an efficient algorithm, because no such algorithm is possible!"



“I can’t find an efficient algorithm, but neither can all these famous people.”