17. 11. 2010

Satz von Cook: SAT ist NP-vollständig.

$\{\varphi \mid \varphi$ aussagenlogische Formel, φ erfüllbar$\}$

$\models \varphi$

φ ist erfüllbar $\Leftrightarrow \neg\varphi$ nicht allgemeingültig

$\not\models \neg\varphi$

Das Erfüllbarkeitsproblem (SAT) ist mit dem Allgemeingültigkeitsproblem gleich komplex.

$SAT \in NP$ Zeuge v. Prüfe in polynomieller Zeit $\bar{v}(\varphi) = W$?

$|v| \leq 2|\varphi|$.

$\underline{A}$ $A \in NTIME(n^k)$ für ein k. $f: \Sigma^* \to \Sigma^*$

$A = A(N)$ $w \in A \leftrightarrow f(w) \in SAT$

$$f(w) = \varphi_w \in \mathcal{L}(\{x_{i,j,s} \mid i,j \in \mathbb{N}, s \in \Gamma \cup Q \cup \{\#\}\})$$

$$i,j \in \{1, \dots, n^k\}, \quad n = |w|$$

$$f : \Sigma^* \to \mathcal{L}, \quad w \mapsto \varphi_w$$

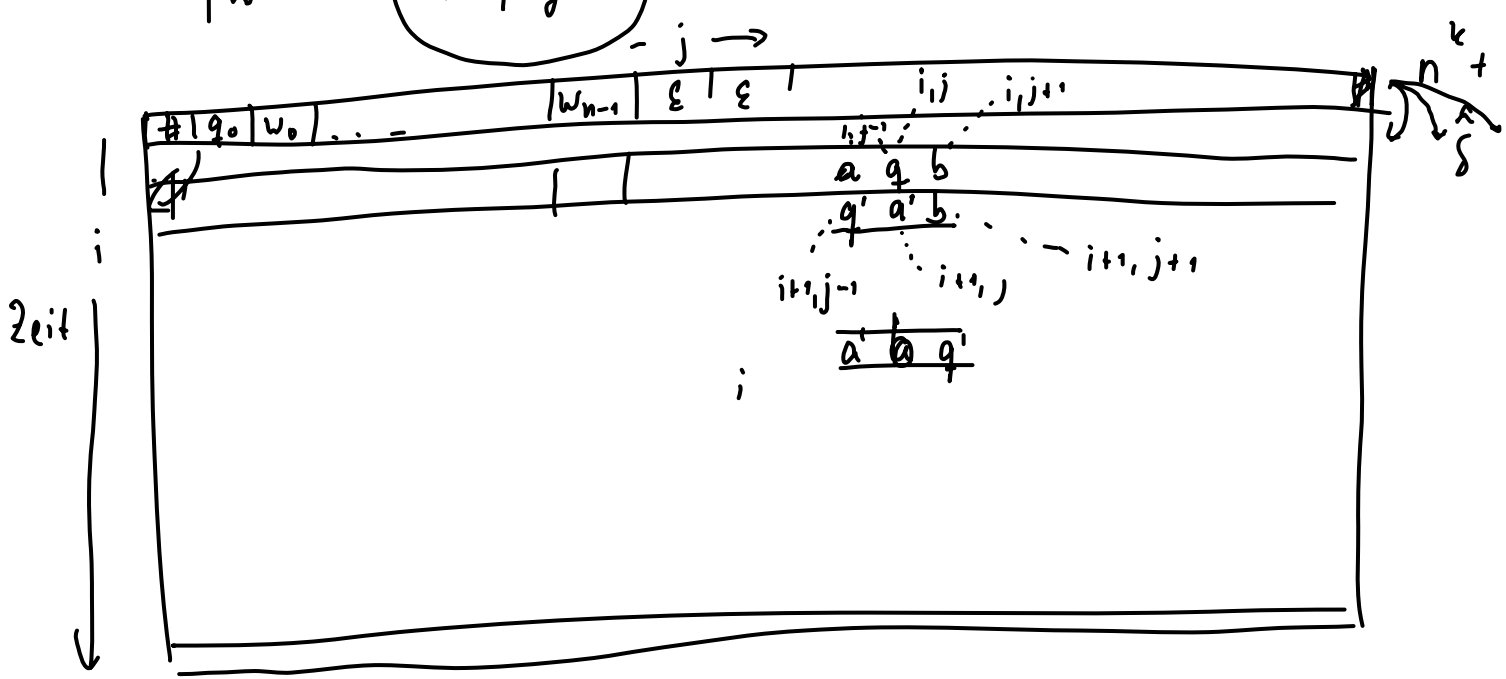
$$\mathbb{N}^3 \xleftrightarrow{\text{bij}} \mathbb{N}$$

$$\mathbb{N}^2 \times (\Gamma \cup Q \cup \{\#\}) \leftrightarrow \mathbb{N}$$

$$\varphi_w = \varphi_{\text{Anfang},w} \wedge \varphi_{\text{Zelle}} \wedge \varphi_{\text{Akzeptierung}} \wedge \varphi_{\text{Bewegung}}$$

(... automatisch erfüllt ...)

$\in$? KNF



Konvention: N muss auf Eingabe von w der Länge n, n^k Schritte laufen.

$$\varphi_{Akz.} = \bigvee_{1 \le j \le n^k} x_{n^k, j, q_{ak}}$$

Luxus: φ_w hat nun fast KNF

$$\varphi_{Bewegung} = \bigwedge_{1 \le i < n^k,\ 1 < j < n^k} \varphi_{zul(i,j)}$$

$$\varphi_{zul(i,j)} = \bigvee_{\begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix} \text{ ist zulässig}} \left(x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6} \right)$$

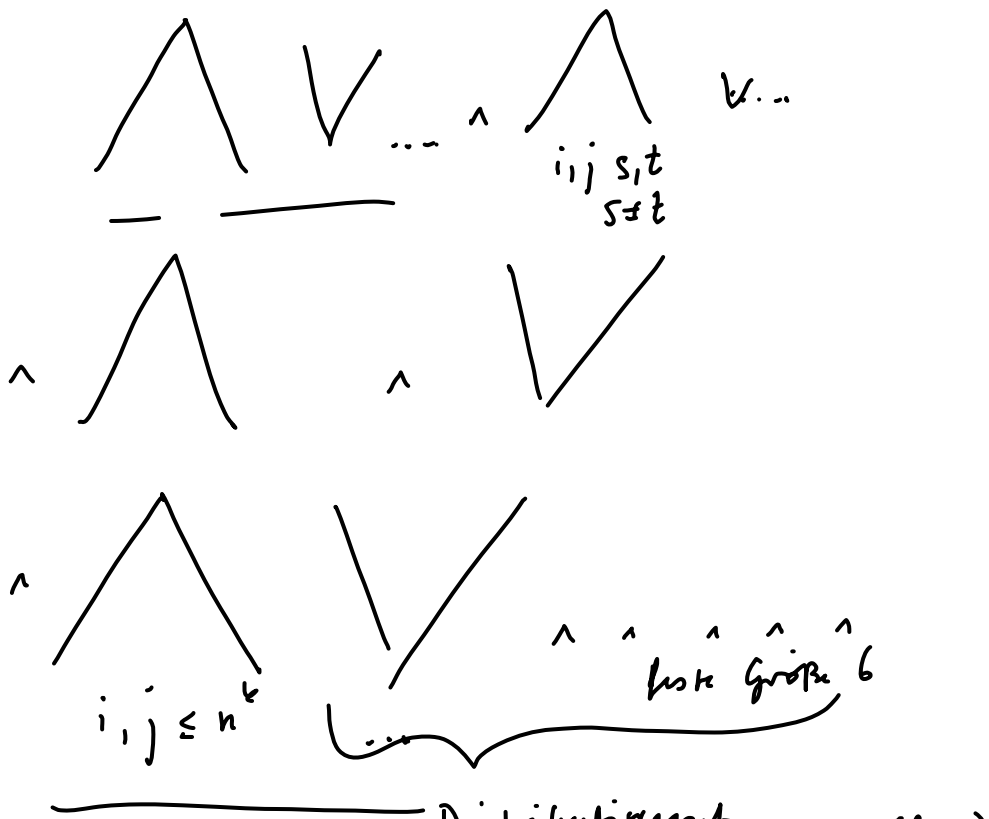
(kommt in $\hat{\delta}$ vor)

$a_i \in Q \cup \Gamma \cup \{\#\}$

$\begin{pmatrix} a & q & b \\ q' & a' & b \end{pmatrix}$ zul. $\leftrightarrow$ $\hat{\delta}(q,a) \ni (q', a', L)$

$\begin{pmatrix} a & q & b \\ a' & b & q' \end{pmatrix}$ zul. $\leftrightarrow$ $\hat{\delta}(q,a) = (q', a', R)$

φ_w ist vom Typ



$\approx O(|n^k|^{26})$

$$\hat{\delta} \subseteq \underline{(Q \times \Gamma) \times (Q \times \Gamma \times \{L, R\})}$$

N fest.

$\hat{\delta}$ konstant groß

$(N, w) \longmapsto \varphi_{N,w}$ polynomial

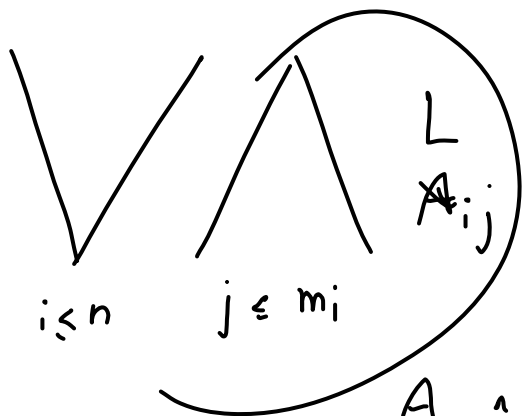
↑ Eingaben

$w \in A^{|N|}) \longleftrightarrow \varphi_w$ ~~Fallback~~ "..."

$NP = P$?

N akzeptiert w. Dann liefert ein akzeptierender Berechnungspfad eine Belegung von $x_{i,j,s}$, die φ_w wahr macht.

Soga SAT für KNF ist NP-vollständig
SAT für allg. Formeln ist " "
SAT für DNF ist in P



$A_5 \wedge \neg A_5$

$L_{i,j}$ Literal

SAT $\Leftarrow$ 3-SAT

polynomial reduzierbar

$\{\varphi \mid \varphi$ 3-KNF, Erf$(\varphi)\}$

konjunktive NF, da Form $\bigwedge_{i \le n} \bigvee_{1 \le j \le 3} L_{i,j}$

$B_1 \vee \dots \vee B_m$ erfüllbar gdw Baum

$(B_1 \vee B_2 \vee C_1) \wedge (\neg C_1 \vee B_3 \vee \overline{B} C_2) \wedge$

$(\neg C_2 \vee B_4 \vee C_3) \wedge \dots (\neg C_{m-3} \vee B_{m-1} \vee B_m) = \varphi_{\text{Baum}}$

ex v $\bar{v}(B_1 \dots \vee B_m) = W \longleftrightarrow$ ex v' $\overline{v'}(\varphi_{\text{Baum}}) = W$

2-SAT $\in P$

Krom-Formel

2-KNF ~~test~~

KNF und alle Disj. haben Größe ≤ 2

Beispiele für P:

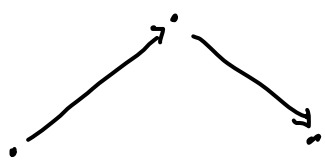
Graph: (V, E)

Punkte, Vertizes

Kanten
edges

$V \neq \emptyset$

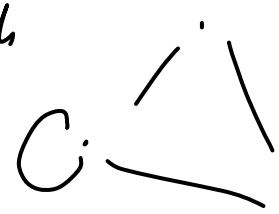
gerichteter Graph:



$E \subseteq V \times V$

$(a,b) \in E$

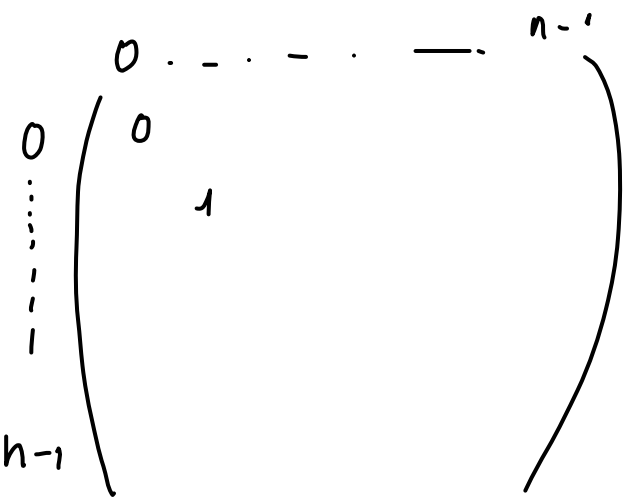
ungerichteter Graph (V, E)



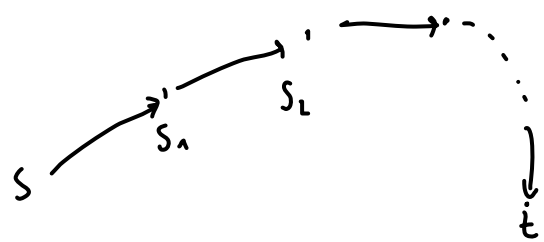
$E \subseteq [V]^2$

$= \{ \{x, y\} \mid x, y \in V, x \neq y \}$

Sei $V = \{0, \ldots, n-1\}$



$a_{i,j} = 1 \iff (i,j) \in E$



$$PFAD = \{ (V, E, s, t) \mid (V, E) \text{ ist ein ger. Graph, } |V| \text{ s, t} \in V, \text{ es gibt einen Pfad von s nach t} \}$$

Satz: PFAD $\in P$.

Bew: $|V| = n$, (V, E, s, t) Eingabe n^2

1. Setze eine Marke auf s

2. Für alle Vertices.

   a, b wenn a markiert ist $\leq n$

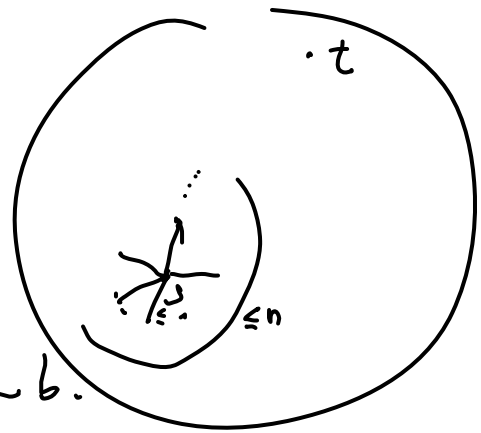
   und $(a, b) \in E$, dann markiere b.

   Rechenzeit $\leq n^2$

3. Wiederhole Punkt 2, bis in einer Runde von Punkt 2 keine Marke mehr hinzukommt. Dann gehe zu 4

   $\leq n$ mal mal.

4. Prüfe, ob t markiert. n Schritte



2004: PRIM = $\{ n \in \mathbb{N} \mid n \notin \text{prim} \} \in P$

$\text{poly}(\log(n))$

$\exists \; r, s \quad r \cdot s = n, \quad r \neq 1, s \neq 1.$

Def RELPRIM = $\{ (n, m) \mid$ n und m haben keinen gemeinsamen Teil $> 1 \}$

n, m sind relativ prim zueinander.

Satz RELPRIM $\in P$. (in Abh von $\lg(n), \lg(m)$)

$\longrightarrow$ $\frac{\text{polynom}(\log(x))\text{-Maschine}}{\text{Polyn}(n)\text{-Maschine}}$ $\quad n = \log x \quad x \leq 2^n$

Eingabe (x, y) der Größe $n = \lg(x) + \lg(y)$

Algorithmus: $x > y$

1. Ersetze x durch $x \bmod y$ $\quad x_1$

$$\vdots$$

$$x = y \cdot k + \underbrace{x \bmod y}_{\text{Rest} < y}$$

2. Vertausche x_1 und y $\quad y_1 = y - k' x_1$

3. Gehe zu Schritt 1 $\quad y = k' \cdot x_1 + \text{Rest}$

y_1

$< y_1 < x_1 < y < x$

4. Wenn nach n Schritten das Ergebnis $y = 0$ sind, dann schaue, ob $x = 1$

akzeptiere. Sonst lehne ab.

$x = p \cdot y$. $\quad p = p$

Durch Ind. alle x_i, y_i sind durch p teilbar.

$\frac{x}{2} \geq y$

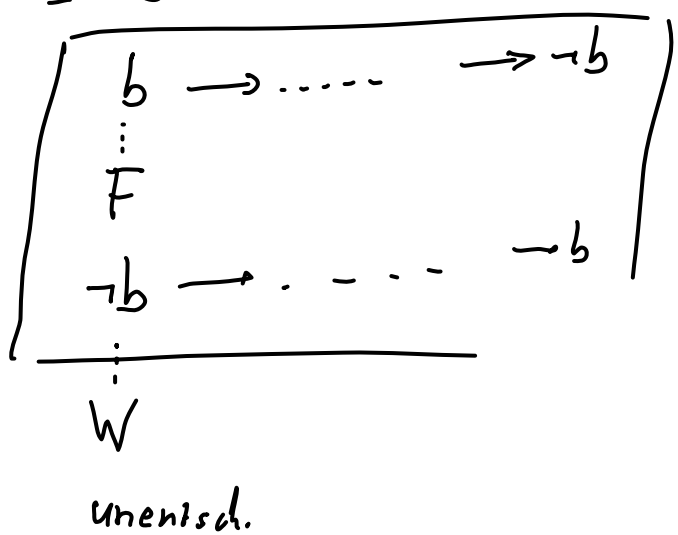
$x \geq 2y$ $\qquad x \bmod y < y \leq \frac{2y}{2}$.

$\leq \log(x) + \lg(y)$ Sd oft wird die Schleife wiederholt.

Satz 2-SAT $\in P$

Beweis:

$$\bigwedge_{i<I} (a_{i,1} \vee a_{i,2}) \quad (a \vee \neg b) \iff \equiv b \longrightarrow a$$



unentsch.

$(a \to b)$

$(b \to c)$ $\quad$ $(a \to c)$

$\bigwedge_{i < \overline{I}} (a_{i,1} \vee a_{i,2})$

$i < \overline{I}$ $\quad$ Literale

In φ sollen nur die Variablen $A_0, \ldots, A_n$ vorkommen.

$V = (\{A_i \mid i \leq n\} \cup \{\neg A_i \mid i \leq n\}, E)$

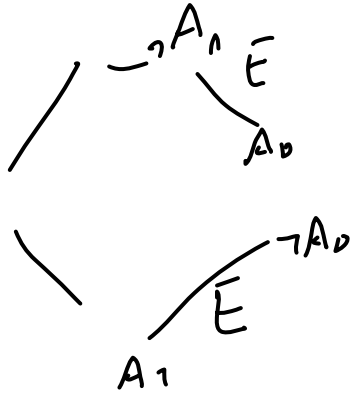
$\hat{=}$ F_0

$E = \{(a, b) \mid \neg a \vee b$ kommt in φ vor, oder $b \vee \neg a$ kommt in φ vor, $a, b \in V\}$.

$\neg\neg A_i \hat{=} A_i$

(V, E) gerichteter Graph $\quad (a,b) \in E \Leftrightarrow a \to b \models \varphi$

$\models \varphi \to (a \to b)$



$(n+2)^2$ potentielle Kanten

$|V| = 2n+2$

$E = \bar{F}_0$

E wählen Im Schritt s die Kante mit F_s gig.

Algorithmus: $a, b, c \in V$: ~~Im Schritt s~~ on

Wenn: $(a,b) \in F_s$, $(b,c) \in F_s$ $(a,c) \in \bar{F}_{s+1}$

falls $(a,c) \notin F_s$. $E = F_0 \subsetneq F_1 \subseteq F$

Falls $F_{s+1} = F_s$, [illegible] gilt wie-