Polynomielle Reduzierbarkeit

DEF

Für $A \subseteq \Sigma^*$, $B \subseteq \Gamma^*$ schreiben wir

$A \leq_p B :\Leftrightarrow$ ($\leq_p$: „polynomiell reduzierbar", ($\leq_p$ ist wohldefiniert bis auf polynomiale Polynome)) $\begin{cases} \exists f: \Sigma^* \to \Gamma^* \text{ berechenbar durch eine (det.) TM } M_f \text{ mit } t_{M_f} \in \mathcal{O}(n^k) \text{ für ein } k \in \mathbb{N} \\ \text{sodass} \\ \quad x \in A \Leftrightarrow f(x) \in B \end{cases}$

(f hat polynomielle Komplexität)

LEMMA

Seien f, g berechenbar durch (det./nicht-det.) TM M_f, M_g und $u, v: \mathbb{N} \to \mathbb{N}$ monoton steigende Funktionen (z.B. Polynome, 2^n, ...)
(↑ streng genommen genügt es, wenn v monoton ist)

mit

$$time_{M_f}(x) \leq u(|x|) \quad \forall x \qquad time_{M_g}(x) \leq v(|x|) \quad \forall x$$

wobei $time \in \{dtime, ntime\}$. Dann kann $g \circ f$ (falls def.) berechnet werden durch $M_f; M_g$ (Hintereinanderausführen) mit

$$time_{M_f; M_g}(x) \leq u(|x|) + v(u(|x|)) \quad \forall x$$

BEWEIS

$$time_{M_f; M_g}(x) = \underbrace{time_{M_f}(x)}_{\leq u(|x|)} + \underbrace{time_{M_g}(f(x))}_{\leq v(|f(x)|) \leq v(u(|x|)) \text{ (Monotonie)}} \leq u(|x|) + v(u(|x|)) \quad \forall x$$

($|f(x)| \leq time_{M_f}(x) \leq u(|x|)$ ↑)

LEMMA

(i) $A \leq_p B \leq_p C \Rightarrow A \leq_p C$ ($\leq_p$ ist transitiv)

(ii) $A \leq_p B$, $B \in P \Rightarrow A \in P$

(iii) $A \leq_p B$, $B \in NP \Rightarrow A \in NP$

} analog zu Entscheidbarkeit

(iv) $(\exists B \in \underbrace{EXPTIME}_{= \bigcup_{p \text{ Polynom}} DTIME(2^{p(n)})} \forall A \in NP: A \leq_p B) \Rightarrow NP \subseteq EXPTIME$

Wir sehen bald, dass dies z.B. für $B = SAT$ gilt

BEWEIS

(i): $\exists f, g: x \in A \Leftrightarrow \underbrace{f(x) \in B \Leftrightarrow g(f(x)) \in C}_{x \in A \Leftrightarrow g \circ f(x) \in C}$

& $g \circ f$ berechenbar durch $M_f; M_g$ mit Komplexität gg. durch das Lemma:

u, v: Polynome $\Rightarrow u \circ v =: q$ Polynom $\Rightarrow time_{M_f; M_g}(x) \leq q(|x|)$ (*)

(ii), (iii): $\exists f: x \in A \Leftrightarrow f(x) \in B$ mit TM M_f polynomieller Komplexität & $\mathbb{1}_B$ wird berechnet durch eine (N)TM M_B mit polynomieller Komplexität

$\Rightarrow x \in A \Leftrightarrow \mathbb{1}_B \circ f(x) = 1 \Rightarrow \mathbb{1}_A = \mathbb{1}_B \circ f$

& genauso wie in (*): $M_f; M_B$ hat polynomielle Komplexität

(i⇒): Angenommen $\exists B$ sodass

- $\mathbb{1}_B$ berechnet wird durch eine det. TM M_B mit exp. Komplexität
- $\forall A \in NP: A \leq_p B$

Dann wollen wir zeigen, dass eine beliebiges $A \in NP$ durch eine deterministische TM mit exp. Komplexität berechnet werden kann:

Wie in (ii), (iii) gibt es ein f berechnet durch M_f mit polynomieller Komplexität, sodass

$$\mathbb{1}_A = \mathbb{1}_B \circ f \qquad (\text{ber. durch } M_f; M_B)$$

Nun folgt aus dem Lemma, dass es ein Polynom q gibt, sodass

$$\text{time}_{M_f; M_B}(x) \leq \underbrace{u(|x|)}_{\text{Polynom}} + \underbrace{v(u(|x|))}_{\text{Polynom}} \leq \underbrace{2^{q(|x|)}}_{2^{\text{Polynom}}}$$

(u = Polynom, $v = 2^{\text{Polynom}}$)

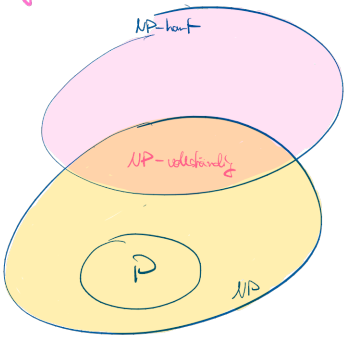
$\Rightarrow A \in EXPTIME$

$\Rightarrow NP \subseteq EXPTIME$

NP-Vollständigkeit

Def

- $B \subseteq \Sigma^*$ ist NP-hart $:\Leftrightarrow A \leq_p B \ \forall A \in NP$
- $B \subseteq \Sigma^*$ ist NP-vollständig $:\Leftrightarrow B$ NP-hart & $B \in NP$



Bem

Es genügt ein NP-hartes Problem B zu finden, das auf andere Probleme reduziert werden kann, um die NP-Härte dieser Probleme zu zeigen (B=SAT im Folgenden):

$$\left.\begin{array}{l} A \leq_p B \ \forall A \in NP \\ \& \ B \leq_p C \end{array}\right\} \xrightarrow{\text{Transitivität}} A \leq_p C \ \forall A \in NP$$

NP-vollst. Probleme gelten als die „schwierigsten" Probleme in NP. In der Tat, aus dem obigen Korollar folgt sofort:

Satz

Sei A NP-vollständig. Dann

$$A \in P \Leftrightarrow P = NP$$

SAT

DEF

$$SAT = \left\{ F \text{ Formel der Aussagenlogik} \;\middle|\; \exists \text{ Belegung } (a_1,\dots,a_k) \in \{0,1\}^k : F(a_1,\dots,a_k) = 1 \right\}$$

2^k mögliche Belegungen

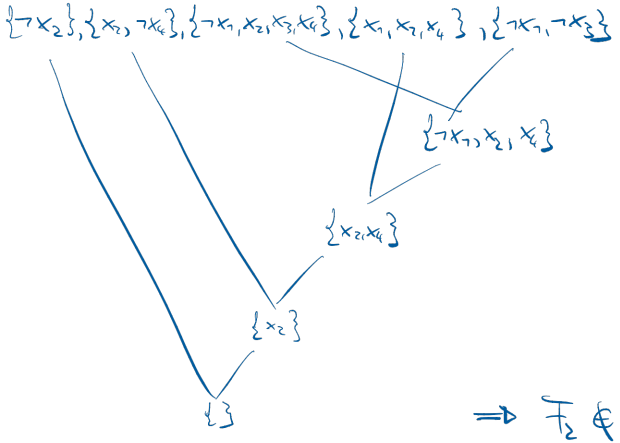
BEISPIEL

- $F_1(x_1,x_2,x_3) := (x_1 \vee (\neg x_2 \wedge \neg x_1)) \wedge x_3$

  z.B. $F_1(1,0,1) = 1 \Rightarrow F_1 \in SAT$

- $F_2(x_1,x_2,x_3,x_4) := \neg x_2 \wedge (x_2 \vee \neg x_4) \wedge (\neg x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_1 \vee x_2 \vee x_4) \wedge (\neg x_1 \vee \neg x_3)$

  Resolution:



$\Rightarrow F_2 \notin SAT$

BEM

SAT ist entscheidbar, z.B.

- Teste alle möglichen Belegungen (z.B. mit Wahrheitstafel)

oder

- Schließe $x \in \overline{SAT}$ aus, mithilfe von Resolution, indem alle Resolutionsmöglichkeiten durchprobiert werden

BEM

$SAT \in NP$ (Guess & Check - Argument):

$$SAT = \left\{ F \mid \exists\, y = (a_1,\dots,a_k) \text{ s.d. } (F,y) \in L' \right\}$$

wobei $L' := \left\{ (F,y) \mid F(y) = 1 \right\}$

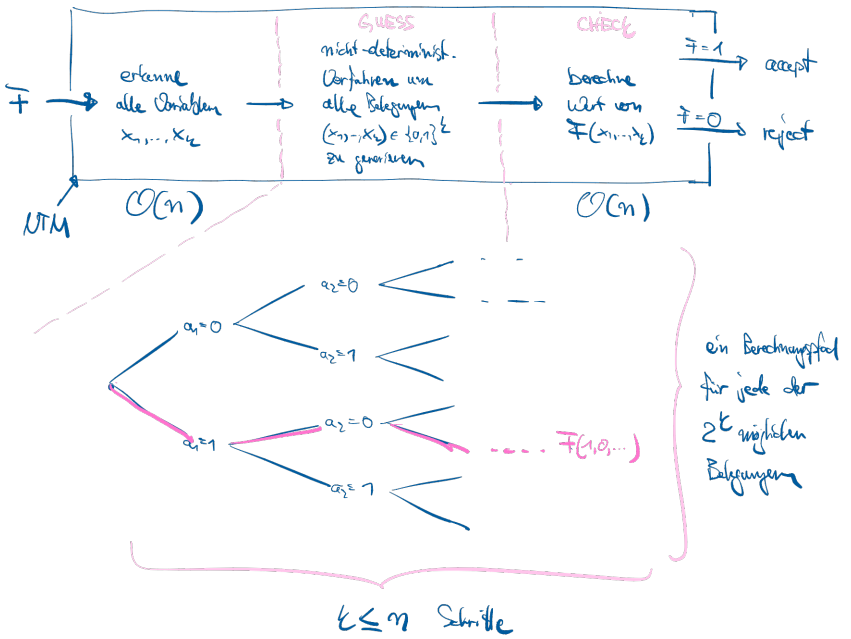
F enthält $a_1,\dots,a_n$ plus Verknüpfungen

- guess: $|y| = |(a_1,\dots,a_k)| = k \leq |F| = n$

  $\Rightarrow k \leq n$ nicht-deterministische Schritte um alle möglichen Belegungen zu erzeugen

- check: Die Auswertung einer Formel F der Länge $|F| = n$ hat Komplexität $O(n) \Rightarrow L' \in P$

$\Rightarrow SAT \in NP$



Satz: SAT ist NP-vollständig. Cook '71, Levin '73

zu zeigen: $L \leq_p SAT \quad \forall L \in NP$ (SAT NP-hart)

Sei $L \in NP$, d.h. $\exists N$ NTM, $\exists q$ Polynom:

$$L = T(N) \qquad t_N \in \mathcal{O}(q)$$

Wir konstruieren

$$f: \Sigma^* \longrightarrow \text{aussagenlogische Formeln}, \quad x \longmapsto f(x)$$

sodass

$$x \in L \iff f(x) \in SAT$$

Sei also $x \in \Sigma^*$ beliebig, $n := |x|$

Idee: Wir bilden die Eigenschaften von N ab auf die Komponenten der Formel F

① Konfigurationen von N (Zustände, Pos. des Schreib-Lesekopfs, Bandinhalt) ⓐ ⓑ ⓒ ⇝ (Boolsche) Variablen

② Funktionsweise von N (Übergänge von Konfigurationen) ⇝ Teilformeln von F

③ Erreichen eines akzeptierenden Endzustands ($z \in E$) ⇝ $F = 1$

(1) Seien Γ Arbeitsalphabet, Z Zustandsmenge, $T = \{0, \dots q(n)\}$ Zeitpunkte, $I = \{-q(n), \dots, q(n)\}$ Position relativ zum Start

↓

Variablen von $\mathcal{F}$: $(zust_{t,z})_{t\in T, z\in Z}$ ∪ $(pos_{t,i})_{t\in T, i\in I}$ ∪ $(band_{t,i,a})_{t\in T, i\in I, a\in\Gamma}$

(A) $=1$ ≙ „$Zustand(t) = z$" (B) $=1$ ≙ „$Pos(t) = i$" (C) $=1$ ≙ „$Band_i(t) = a$"

z.B.: $zust_{0,z_0} = 1$, $zust_{0,z} = 0 \ \forall z \neq z_0$

$pos_{0,1} = 1$, $pos_{0,j} = 0 \ \forall j \neq 1$

$band_{0,i,x_i} = 1$ (für Eingabe $x = x_1 \cdots x_n$)

$band_{0,i,a} = 0 \ \forall a \neq x_i$

nur bestimmte Belegungen der Variablen sind zugelassen

(2) Funktionsweise von $\mathcal{V}$:

- **Randbedingungen** $\mathcal{R}$: zu jedem Zeitpunkt t kann
  - (A) $\mathcal{N}$ nur in einem der Zustände $z \in Z$ sein
    ($\forall t\in T \ \exists! z\in Z : zust_{t,z} = 1$)
  - (B) der Schreib-Lese-Kopf nur in einer Position $i \in I$ sein
    ($\forall t\in T \ \exists! i\in Z : pos_{t,i} = 1$)
  - (C) jede Position $i \in I$ kann nur ein $a \in \Gamma$ enthalten
    ($\forall t\in T \ \forall i\in I \ \exists! a\in\Gamma : band_{t,i,a} = 1$)

Hierfür benötigen wir eine Teilformel G mit der Eigenschaft

$G(u_1, \dots, u_m) = 1 \iff$ für genau ein $i \in \{1, \dots, m\}$ ist $u_i = 1$

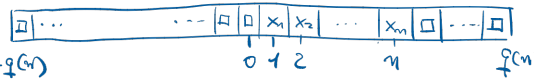
$= \left(\bigvee_{i=1}^{m} u_i\right) \wedge \left(\bigwedge_{i=1}^{m-1}\bigwedge_{j=i+1}^{m} \neg(u_i \wedge u_j)\right)$

[Mögliche Umsetzungsvariante von $XOR(u_1,u_2) = (u_1 \vee u_2) \wedge \neg(u_1 \wedge u_2)$ für $m > 2$ Variablen]

↓

$\mathcal{R} = \bigwedge_{t\in T}\left(G((zust_{t,z})_{z\in Z}) \wedge G((pos_{t,i})_{i\in I}) \wedge \bigwedge_{i\in I} G((band_{t,i,a})_{a\in\Gamma})\right)$ (A) (B) (C)

- **Anfangsbedingungen** $\mathcal{A}$: Konfiguration zum Zeitpunkt $t = 0$



$\mathcal{A} = zust_{0,z_0} \wedge pos_{0,1} \wedge \bigwedge_{i=1}^{n} band_{0,i,x_i} \wedge \bigwedge_{i\in I\setminus\{1,\dots,n\}} band_{0,i,\square}$

- **Übergangsregeln** $\ddot{U}_1 \wedge \ddot{U}_2$ = Übergänge durch Kopf ∧ Band bleibt überall sonst

$\ddot{U}_1 = \bigwedge_{t\in T, i, a}\Big(zust_{t,z} \wedge pos_{t,i} \wedge band_{t,i,a} \Rightarrow \bigvee_{z',a',j \in \delta(z,a)} (zust_{t+1,z'} \wedge pos_{t+1,i+j} \wedge band_{t+1,i,a'})\Big)$

$\ddot{U}_2 = \bigwedge_{t,i,a}(\neg pos_{t,i} \wedge band_{t,i,a} \Rightarrow band_{t+1,i,a})$

(3) $\mathcal{F}_N := \mathcal{R} \wedge \mathcal{A} \wedge \mathcal{U}_1 \wedge \mathcal{U}_2$ ist für jedes $x \in \Sigma^*$ erfüllbar

[z.B. durch die Belegung der Variablen $\{zust_{t,q}, pos_{t,i}, band_{t,i,a}\}_{t,i,a}$ die dem tatsächlichen Berechnungsbaum von N entsprechen.]

Um zu erreichen, dass $\mathcal{F}$ erfüllbar $\Leftrightarrow x \in L \ (= T(N))$ forcieren wir, dass immer ein akzeptierender Endzustand erreicht werden muss:

$$\mathcal{F} := \mathcal{F}_N \wedge \mathcal{F}_E$$

wobei

$$\mathcal{F}_E := \bigvee_{z \in E} zust_{f(n),z}$$

$\downarrow$

$f(x) := \mathcal{F} \in SAT \iff x \in L = T(N)$

Komplexität von f:

$\mathcal{F} = \mathcal{R} \wedge \mathcal{A} \wedge \mathcal{U}_1 \wedge \mathcal{U}_2 \wedge \mathcal{F}_E$ benötigt Berechnung von n $\overbrace{\ }^{\mathcal{O}(g(n))}$ $g(n)$, dann kann $\mathcal{F}$ von einer TM hingeschrieben werden.

$\Rightarrow$ Komplexität von f $\propto |\mathcal{F}|$

Wie wächst $|\mathcal{F}|$ mit n?

- $|G| = \left| \left(\bigvee_{i=1}^{m} u_i \right) \wedge \left(\bigwedge_{i=1}^{m-1} \bigwedge_{j=i+1}^{m} \neg (u_i \wedge u_j) \right) \right| \in \mathcal{O}(m^2)$
- $|\Gamma|, |\Sigma| \in \mathcal{O}(g(n))$
- $|\mathcal{R}| = \left| \bigwedge_{t \in T} \left(G((zust_{t,q})_{z \in Z}) \wedge G((pos_{t,i})_{i \in \Sigma}) \wedge \bigwedge_{i \in \Sigma} G((band_{t,i,a})_{a \in \Gamma}) \right) \right| \in \mathcal{O}(g(n)^3)$
  ($\mathcal{O}(g(n))$, $\mathcal{O}(1)$, $\mathcal{O}(g(n)^2)$, $\mathcal{O}(g(n))$)
- $|\mathcal{A}| = \left| zust_{0,q_0} \wedge pos_{0,1} \wedge \bigwedge_{i=1}^{n} band_{0,i,x_i} \wedge \bigwedge_{i \in \Sigma \setminus \{1,\dots,n\}} band_{0,i,\square} \right| \in \mathcal{O}(g(n))$
  ($\mathcal{O}(n)$, $\mathcal{O}(g(n))$; angenommen g hat Grad ≥ 1 (o.B.d.A.))
- $|\mathcal{U}_1| = \left| \bigwedge_{t \in [g(n)], i \in [g(n)], a \in \Gamma} \left(zust_{t,q} \wedge pos_{t,i} \wedge band_{t,i,a} \Rightarrow \bigvee_{z',a',j \in \delta(z,a)} (zust_{t+1,z'} \wedge pos_{t+1,i+j} \wedge band_{t+1,i,a'}) \right) \right|$
  ($\mathcal{O}(g(n))$, $\mathcal{O}(g(n))$, $\mathcal{O}(1)$)
- $|\mathcal{U}_2| = \left| \bigwedge_{t,i,a} \left(\neg pos_{t,i} \wedge band_{t,i,a} \Rightarrow band_{t+1,i,a} \right) \right| \Rightarrow |\mathcal{U}_1 \wedge \mathcal{U}_2| \in \mathcal{O}(g(n)^2)$
  ($\mathcal{O}(g(n))$, $\mathcal{O}(g(n))$, $\mathcal{O}(1)$)

$\downarrow$

$|\mathcal{F}| \in \mathcal{O}(g(n)^2) \Rightarrow f$ hat polynomielle Komplexität

WEB

① Zuvor:

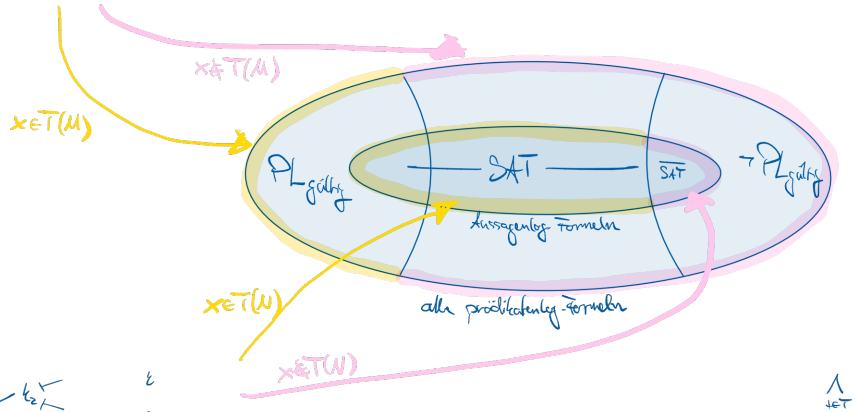
$z_0 x \vdash \alpha_1 z_1 \beta_1 \vdash \cdots \vdash \alpha_n z_n \beta_n \vdash \alpha z_e \beta$

$(za, z'c)$	falls $\delta(z,a) = (z', c, N)$
(za, cz')	falls $\delta(z,a) = (z', c, R)$
$(bza, z'bc)$	falls $\delta(z,a) = (z', c, L)$, für alle $b \in \Gamma$
$(\#za, \#z'\sqcup c)$	falls $\delta(z,a) = (z', c, L)$
$(z\#, z'c\#)$	falls $\delta(z,\sqcup) = (z', c, N)$
$(z\#, cz'\#)$	falls $\delta(z,\sqcup) = (z', c, R)$
$(bz\#, z'bc\#)$	falls $\delta(z,\sqcup) = (z', c, L)$, für alle $b \in \Gamma$

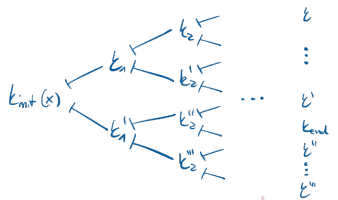
$\left(\begin{array}{l} P(f_{x_1}(a), f_{t_1}(a)) \wedge \cdots \wedge P(f_{x_\ell}(a), f_{t_\ell}(a)) \wedge \\ \forall u,v : \left(P(u,v) \Rightarrow \left(P(f_{x_1}(u), f_{t_1}(v)) \wedge \cdots \wedge P(f_{x_\ell}(u), f_{t_\ell}(v))\right)\right) \end{array} \right)$
$\Rightarrow \exists z\, P(z,z)$

Berechnungspfad $\xi_M(x)$ —$H \leq PCP$→ $\{(\vdots), \ldots, (\vdots)\}$ —$PCP \leq PL$→ Prädikatenlogische Formel F

$(x \in T(M) \Leftrightarrow \exists$ PCP Lösung$)$ $\quad$ $(\exists$ PCP Lösung $\Leftrightarrow F$ gültig$)$



② Hier:



$L \leq_p SAT$ HLEMP

$\bigwedge_{t \in T} \left(G((zust_{t,z})_{z \in Z}) \wedge G((pos_{t,i})_{i \in I}) \wedge \bigwedge_{i \in I} G((band_{t,i,a})_{a \in T}) \right)$
$\wedge\ zust_{0,z_0} \wedge pos_{0,1} \wedge \bigwedge_{i=1}^{n} band_{0,i,x_i} \wedge \bigwedge_{i \in I \setminus \{1,\ldots,n\}} band_{0,i,\sqcup}$
$\bigwedge_{t \in \{0,\ldots,q-1\}} \left(zust_{t,z} \wedge pos_{t,i} \wedge band_{t,i,a} \Rightarrow \bigvee_{z',a,d \in \delta(z,a)} \left(zust_{t+1,z'} \wedge pos_{t+1,i+d} \wedge band_{t+1,i,a'} \right) \right)$
$\bigwedge_{t,i,a} \left(\neg pos_{t,i} \wedge band_{t,i,a} \Rightarrow band_{t+1,i,a} \right) \wedge \bigvee_{z \in E} zust_{q(z),z}$

Berechnungspfad/baum $\xi_N(x)$ $\quad$ $(x \in T(N) \Leftrightarrow F$ erfüllbar$)$ $\quad$ Aussagenlogische Formel F

WEITERE BEISPIELE 1

SAT $\leq_p$ 3-SAT $\leq_p$ SUBSETSUM $\leq_p$ PARTITION $\leq_p$ BIN PACKING

① ② ③ ④

(hier nur Beweisideen, siehe Skript ab Seite 68 für Details)

NP

$3(\text{KNF})\text{-SAT} := \{ \varphi \text{ Aussagenlog. Formel in KNF} \mid \varphi \text{ ist erfüllbar \& hat höchstens 3 Literale pro Klausel} \} \in NP$

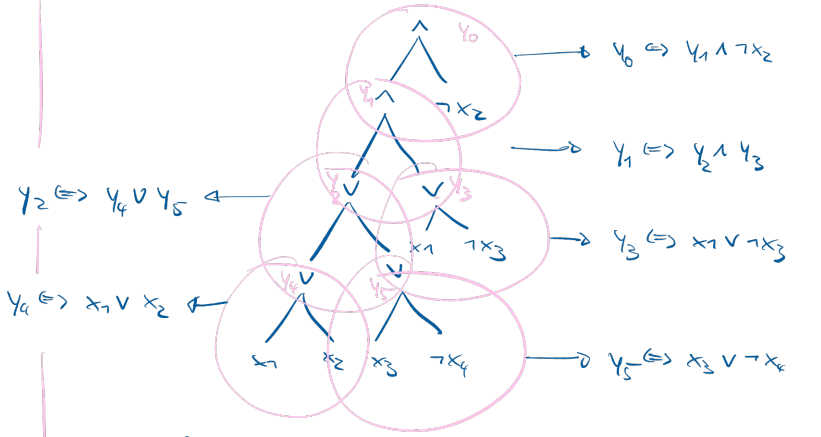
(raten & checken)

z.B. $x_1 \wedge (\neg x_2 \vee x_3 \vee x_4) \wedge \dots$

Beweisidee von ①: SAT $\leq_p$ 3-SAT

Wir konstruieren eine Abbildung von φ nach φ', wobei φ' nur 3 Literale pro Klausel hat & Erfüllbarkeitsäquivalenz zw. $\varphi \& \varphi'$ herrscht, anhand eines Beispiels:

$\varphi = (x_1 \vee \neg x_3) \wedge \neg x_2 \wedge (x_1 \vee x_2 \vee x_3 \vee \neg x_4)$



$\varphi' = (y_0 \Leftrightarrow y_1 \wedge \neg x_2) \wedge (y_1 \Leftrightarrow y_2 \wedge y_3) \wedge \dots \rightsquigarrow$ KNF (max 3 Lit.)

Beh.: 2-SAT ist in P

Resolution liefert einen polynomiellen Algorithmus:

$(x_1 \vee x_2) \wedge (\neg x_3 \vee x_4) \wedge (\neg x_2 \vee x_3) \wedge \dots$

$x_1 \vee x_3$ ← höchstens 2-elementig

&

Anzahl (Klauseln mit ≤ 2 Literalen aus k Variablen) $= k^2$

$n = |\varphi|$, $k \leq n$ $\Rightarrow O(n^2)$

Intuitiv:

2-SAT erlaubt nur Implikationen der Form $a \Rightarrow b$

(d.h. wenn man $a=1$ wählt folgt daraus $b=1$, für $b=0$ folgt $a=0$)

$\Rightarrow$ Um $\varphi \in$ 2-SAT zu verifizieren genügt es alle möglichen Implikationsstränge entlang zu laufen ($O(n^2)$-viele)

Bei 3-SAT gibt es Implikationen der Form $a \Rightarrow b \vee c$, wodurch es zur kombinatorischen Explosion kommen kann, denn es müssen dann beide Möglichkeiten untersucht werden

SUBSET SUM $:= \bigcup_k \left\{ (a_1,\dots,a_k,b) \in \mathbb{N}^{k+1} \mid \exists I \subseteq \{1,\dots,k\} : \sum_{i\in I} a_i = b \right\} \in NP$ (ganz schwer)

Beweisidee von (2): 3-SAT $\leq_p$ SUBSETSUM

$F = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (\neg x_2 \vee \neg x_5)$

$\Rightarrow$ 3 Klauseln, 5 Variablen

$\Rightarrow b := 444\ 11111$

Man konstruiert nun Zahlen $v_1,\dots,v_5, u_1,\dots,u_5, c_1,\dots,c_3, d_1,\dots,d_3$ (zu $a_1,\dots,a_8$) wie folgt:

	in welcher Klausel kommt x_i vor (Ziffern)	in welcher Klausel kommt $\neg x_i$ vor
v_1	100 10000	u_1 = 010 10000
v_2	000 01000	u_2 = 001 01000
v_3	000 00100	u_3 = 100 00100
v_4	000 00010	u_4 = 000 00010
v_5 = 110 00001		u_5 = 001 00001

(Existenz Variablen der Variablen)

(Hilfs-Zahlen)

c_1 = 100 000 00	d_1 = 200 000 00
c_2 = 010 000 00	d_2 = 020 000 00
c_3 = 001 000 00	d_3 = 002 000 00

$\Rightarrow$ Erfüllende Belegung von F, z.B. $(x_1,x_2,x_3,x_4,x_5) = (1,0,0,1,0)$ entspricht nun Auswahl von $v_1,\dots,v_5$, sowie $u_1,\dots,u_5$:

(wähle so, dass Summe $\geq 111\dots$ erfüllend)

$v_1 + v_2 + v_3 + v_4 + v_5 = \underbrace{212}_{\geq 111}\ 11111$ (gilt immer!)

(da alle Klauseln wahr sein müssen)

$\Rightarrow$ kann immer mit wahl von e_i, d_i zu 4444 ergänzt werden (b)

aber nur wenn ≥ 111 (wird für "$\in$" benötigt)

$\Rightarrow I = 1$

PARTITION $= \bigcup_k \left\{ (a_1,\dots,a_k) \in \mathbb{N}^k \mid \exists I \subseteq \{1,\dots,k\} : \sum_{i\in I} a_i = \sum_{i\in\{1,\dots,k\}\setminus I} a_i \right\}$

Beweis von (3): (SUBSET SUM $\leq_p$ PARTITION)

$(a_1,\dots,a_k,b) \mapsto (a_1,\dots,a_k, M-b+1, b+1)$ wobei $M = \sum_{i=1}^{k} a_i$ $\leftarrow \mathcal{O}(n)$

- angenommen $I \subseteq \{1,\dots,k\}$ erfüllt $\sum_{i\in I} a_i = b$, dann ist

$$\underbrace{\sum_{i\in I} a_i}_{=b} + M-b+1 = M+1 = \sum_{i=1}^{k} a_i + b+1 - \underbrace{\sum_{i\in I} a_i}_{b} = \sum_{i\in\{1,\dots,k,k+1,k+2\}\setminus(I\cup\{k+1\})} a_i$$

(Summe über $i \in I \cup \{k+1\}$)

$\Rightarrow I \cup \{k+1\}$ ist Lösung von PARTITION

- angenommen J löst PARTITION. Es gilt $\overbrace{M-b+1}^{k+1} + \overbrace{b+1}^{k+2} = M+2 > \sum_{i=1}^{k} a_i$

also muss $k+1 \in J$ & $k+2 \in \overline{J}$ (oder umgekehrt).

$\Rightarrow I := J \setminus \{k+1\}$ löst SUBSET SUM

$$\sum_{i\in I} a_i = \sum_{J\setminus\{k+1\}} \tilde{a}_i = \sum_{J} \tilde{a}_i - (M-b+1) = \sum_{\overline{J}} \tilde{a}_i - (M-b+1) = -\sum_{i\in I} a_i + 2b$$

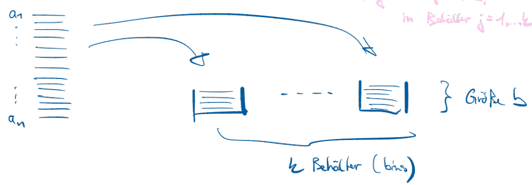
$\sum_{i\in I} a_i + b+1$; $\sum_{i=1}^{k} a_i - b + 1$

$\Leftrightarrow \sum_{i\in I} a_i = b$

$-\sum_{i\in I} a_i + b - 1$

$\text{BIN PACKING} := \left\{ (a_1,\dots,a_n, b, k) \in \mathbb{N}^{n+2} \;\middle|\; \exists f : \{1,\dots,n\} \to \{1,\dots,k\} : \begin{array}{l} \forall j \in \{1,\dots,k\}, \\ \sum_{i, f(i)=j} a_i \le b \end{array} \right\}$

↑ Zuordnung von Objekten a_i in Behälter $j = 1,\dots,k$



Beweis von ④: PARTITION $\le_p$ BIN PACKING

$(a_1,\dots,a_n) \longmapsto (a_1,\dots,a_n, b = \lfloor \frac{1}{2}\sum_{i=1}^{n} a_i \rfloor, k=2)$ stellt Reduktion her:

- angenommen $J \subset \{1,\dots,n\}$ löst PARTITION: $\sum_{i\in J} a_i = \sum_{i \in \bar{J}} a_i$ & $\sum_{J} a_i + \sum_{\bar{J}} a_i = \sum_{i=1}^{n} a_i$

$\Rightarrow$ | J | | $\bar{J}$ | $\} \le \frac{1}{2}\sum_{i=1}^{n} a_i$ $\quad \left(f(i) := \begin{cases} 1 & , i \in J \\ 2 & , i \in \bar{J} \end{cases} \right)$

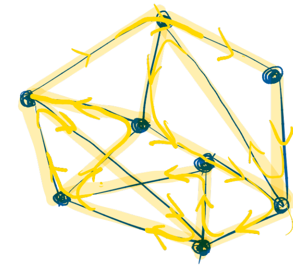
- falls $f : \{1,\dots,n\} \to \{1,2\}$ BIN PACKING löst, dann löst $J := \{ i \mid f(i) = 1 \} = f^{-1}(\{1\})$ $\quad (\Rightarrow \bar{J} = f^{-1}(\{2\}))$ PARTITION.

$$\left[\left. \begin{array}{l} \sum_{i\in J} a_i \le \frac{1}{2}\sum_i a_i \,, \quad \sum_{i\in\bar{J}} a_i \le \frac{1}{2}\sum_i a_i \\ \sum_{i\in J} a_i + \sum_{i\in\bar{J}} a_i = \sum a_i \end{array} \right\} \sum_{i\in J} a_i = \sum_{i\in\bar{J}} a_i \right.$$

$$\text{Da } \left\{ \begin{array}{l} s \le \frac{1}{2} \\ t \le \frac{1}{2} \end{array} \right\} \wedge s+t=1 \Rightarrow s = t \left(= \tfrac{1}{2}\right) \Big]$$

WEITERE BEISPIELE 2

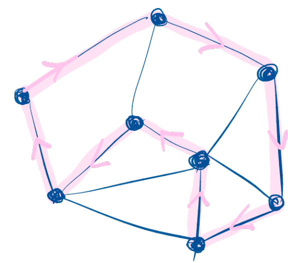
Graphenprobleme



EULER-KREIS Jede Kante muss genau einmal besucht werden

$\{ G = (V,E) \mid G \text{ besitzt einen Euler-Kreis} \} \in P$

↑ Satz (Euler 1736): G besitzt Euler-Kreis $\Leftrightarrow$ Jeder Knoten hat gerade Anzahl von Kanten



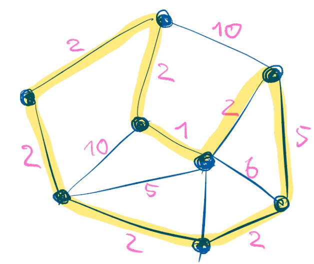
HAMILTON-KREIS Jeder Knoten muss genau einmal besucht werden

$\left\{ G = (V,E) \;\middle|\; \begin{array}{l} \exists \pi \text{ Permutation der Knotenindizes:} \\ \forall i = 1,\dots,n-1 \; (v_{\pi(i)}, v_{\pi(i+1)}) \in E, \\ \text{und } (v_{\pi(n)}, v_{\pi(1)}) \in E \end{array} \right\} \in NP$

Rundreise

gerichtete Variante

{ } in ungerichteter Variante



TRAVELING SALESMAN Jede Kante erhält zusätzlich eine Zahl ($\hat{=}$ Entfernung) und Ziel ist es, dass Gesamtlänge eine obere Schranke k nicht übertrifft.

$\left\{ ((M_{ij})_{i,j \in \{1,\dots,n\}}, k) \;\middle|\; \begin{array}{l} \exists \pi \text{ Permutation von } \{1,\dots,n\}: \\ \sum_{i=1}^{n-1} M_{\pi(i),\pi(i+1)} + M_{\pi(n),\pi(1)} \le k \end{array} \right\} \in NP$

↑ Entfernungsmatrix

($(i,j) \notin E$ kann dargestellt werden durch $M_{ij} > k$)

gerichteter
HAMILTON-KREIS ist NP-vollständig

Beweisidee von 3SAT $\leq_p$ gerichteter HAMILTON-KREIS

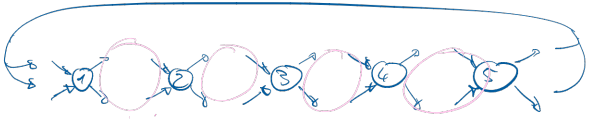
$F = (x_1 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee x_4 \vee x_5) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_5)$

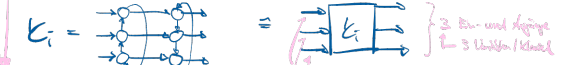
F : $n = 5$ (Variablen), $m = 3$ (Klauseln)

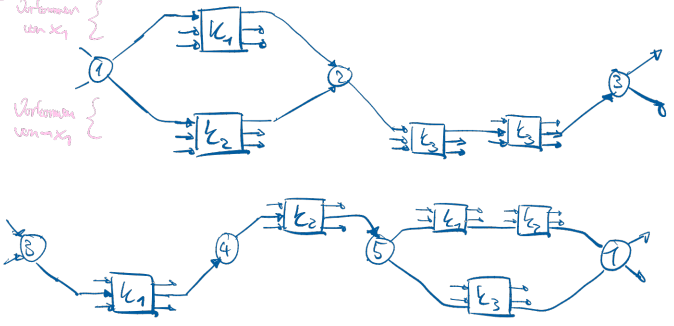
Konstruktion von G_F:

5 Knoten

teilgraphen k_1, k_2, k_3



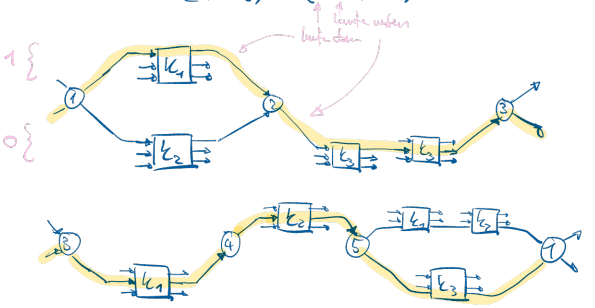




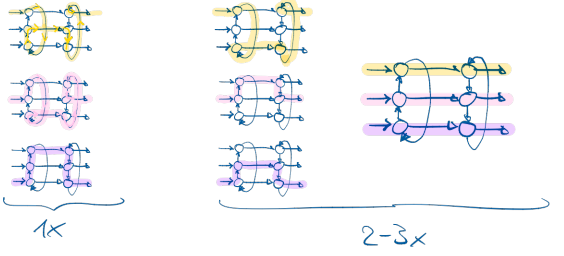
(NP)

Für F erfüllbar $\Rightarrow$ G_F hat Hamilton-Kreis

Betrachten wir z.B. $(x_1, \dots, x_5) = (1,0,0,1,0)$



Der Pfad durch k_i hängt davon ab wie oft k_i im (einfachen) Hamilton-Kreis vorkommt:



Für F erfüllbar $\Leftarrow$ G_F hat Hamilton-Kreis

benutzen wir die Eigenschaft von k_i (Beweis siehe Skript S80), dass

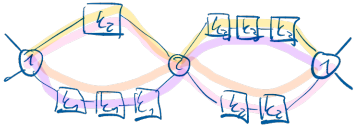
> Wenn G_F einen Hamilton-Kreis besitzt, so verläuft dieser, immer wenn er k_i passiert, so dass er k_i im Eingang auf der gleichen Höhe des Ausgangs verlässt.

Zusammen mit der Tatsache dass ein durchlaufener Klauselgraph k_i bedeutet dass Klausel i wahr ist & jedes k_i von einem H-K mind. einmal durchlaufen wird

$\mathcal{F}$ nicht erfüllbar $\Rightarrow$ $G_{\mathcal{F}}$ hat keinen Ham.-Kreis

z.B.: $\mathcal{F} = (\neg x_1 \vee \neg x_2 \vee \neg x_1) \wedge (x_1 \vee \neg x_2 \vee \neg x_2) \wedge (x_2 \vee x_2 \vee x_2)$

k_1 k_2 k_3



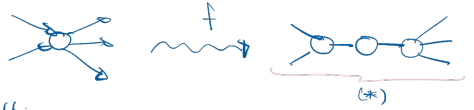
hat keinen Hamilton-Kreis

2 SAT

ungerichteter HAMILTON-KREIS ist NP-vollständig

Beweisskizze von gerichteter HAMILTON-KREIS $\leq_p$ ungerichteter HAMILTON-KREIS

Transformiere gerichteter Graphen $\longrightarrow$ ungerichtet Graphen:



$f \in \mathcal{O}(|G|)$ „Knoten $\to$ 6 Knoten"

Dann gilt:

G hat Hamilton-Kreis $\Leftrightarrow$ $f(G)$ hat Hamilton-Kreis

$\Rightarrow$: trivial

$\Leftarrow$: Wenn ein Knoten in (*) durchlaufen wird (links oder rechts), so muss der gesamte Teilgraph durchlaufen werden, da sonst eine Sackgasse entsteht. Zusätzlich gilt: entweder werden alle Knoten von links oder rechts durchlaufen (so falls Hamilton-Kreis gegen Pfeilrichtung läuft gibt es einen gegenläufigen)

2 SAT

TRAVELING SALESMAN ist NP-vollständig

Beweisskizze von ungerichteter HAMILTON-KREIS $\leq_p$ TRAVELING SALESMAN

intuitiv klar: keine Kante $\Rightarrow$ Entfernung > 1

$G = (\{1,\dots,n\}, E) \xrightarrow{f} (M, n)$ mit $M_{ij} = \begin{cases} 1, & \{i,j\} \in E \\ 2, & \{i,j\} \notin E \end{cases}$

ungerichteter Graph

Adjazenz-Matrix = Anzahl von Kanten

Wert beliebig > 1

Dann ist klar:

$\exists$ Hamilton-Kreis π in G $\iff$ $\sum_{i=1}^{n-1} M_{\pi(i)\pi(i+1)} + M_{\pi(n)\pi(0)} \leq n$

$\Rightarrow$: $\{\pi(i), \pi(i+1)\} \in E$, $\{\pi(n), \pi(0)\} \in E$ } wenn π Hamilton-Kreis $\Rightarrow$ $M_{\pi(i)\pi(i+1)} = 1$, $M_{\pi(n)\pi(0)} = 1$ } $\Big\} = n$

$\Leftarrow$: $\exists \pi$: $\sum M_{\pi(i)\pi(i+1)} + M_{\pi(n)\pi(0)} \leq n$ $\Rightarrow$ π Hamilton-Kreis, da $M_{\pi(i)\pi(i+1)} + 2$ ...