Komplexität von Entscheidungsproblemen

(bekannt aus INFO 4)

Entscheidungsproblem: Menge von Instanzen I
Teilmenge von positiven $Y \subseteq I$

($\hat{=}$ Erkennung einer Sprache)

traditionelle Komplexitätstheorie betrachtet meist nur Entsch.

$TIME(t(n))$

$SPACE(t(n))$ $\qquad TIME(t(n)) \subseteq SPACE(t(n))$

Nichtdeterminismus:

guess $b \in \{0,1\}$ allgemeiner guess $x \in S$ (Zeit $|x|$)

Berechnungsbaum. Erfolg wenn es gibt erfolgreichen Pfad.

$NTIME(t(n))$

$NSPACE(t(n))$ - $NTIME(t(n)) \subseteq SPACE(t(n))$

$NSPACE(t(n)) \subseteq SPACE(t^2(n))$ (SAVITCH)

$NSPACE(t(n)) \subseteq TIME(2^{O(t(n))})$

Komplexitätsklassen:

$P := \bigcup_k TIME(n^k)$ $\qquad NP := \bigcup_k NTIME(n^k)$

$PSPACE := \bigcup_k SPACE(n^k)$

$EXP := \bigcup_k TIME(2^{n^k})$

$$P \subseteq NP \subseteq PSPACE \subseteq EXP$$

NP $\hat{=}$ Verifikation von Lösungen:

Problem $Q \in NP$ gdw

es gibt R in P ~~mit~~ und Polynom $q()$

mit I_R = Paare (x,w) $x \in I_Q$, $|w| \leq q(|x|)$

$$\forall x \in I_Q \quad x \in Y_Q \text{ gdw } \exists w \in W_Q \ (x,w) \in Y_R$$

Bsp: typische NP-Probleme:

SAT, CLIQUE, Integer Linear Programming (ILP)

Beweis: einfach: $\rightarrow$ zu $x \in I_Q$ rate w sodass $(x,w) \in Y_P$

$\leftarrow$ Berechnungspfade der NTM als Zeugen (= Folge der geratenen Werte.)

Reduktion

f ist Reduktion von P auf Q:

- f berechenbar in polynomialer Zeit
- $f: I_P \rightarrow I_Q$ so dass

$$\forall x \in I_P : x \in Y_P \leftrightarrow f(x) \in Y_Q$$

$P \leq_m Q$ falls es eine Reduktion von P auf Q gibt.

reflexiv - transitiv (Prä-Ordnung)

$\rightarrow$ induziert Äquivalenz $\equiv_m$

Bsp: SAT $\leq_m$ ILP

Vollständigkeit

Problem P heisst hart für Klasse C, falls f. alle $Q \in C$ $Q \leq_m P$

P vollständig in C

$P \in C$ und P hart für C

NP-vollständige Probleme

interessant: Q NP-vollständig:

$$Q \in P \rightarrow P = NP$$

Bekannt: (Cook) SAT, 3SAT NP-vollständig

mit oben ILP ist NP-vollständig!

NP-vollständige Graphenprobleme:

Graph $G = (V, E)$ ungerichtet.

- Vertex Cover: $U \subseteq V$ so dass f. alle $e = \{x,y\} \in E$: $x \in U \vee y \in U$.

  Problem VC:

  Instanz: Graph G, nat. Zahl k

  Frage: gibt es ein Vertex Cover $U \subseteq V$ mit $|U| \leq k$

- Clique: $U \subseteq V$ mit $\{x,y\} \in E$ f. alle $x, y \in U$.

  Problem CLIQUE:

  Instanz: Graph G, nat. Zahl k

  Frage: gibt es eine Clique $U \subseteq V$ mit $|U| \geq k$.

- Hamilton-Zyklus: geschlossener Weg, der jeden Knoten genau einmal passiert.

  Problem HC:

  Instanz: Graph G

  Frage: Gibt es einen Hamilton-Zyklus.

Alle diese Probleme sind offenbar in NP.
Sind alle auch vollständig: Reduktionen

$$3SAT \leq_m VC \begin{array}{l} \leq_m HC \\ \leq_m CLIQUE \end{array}$$

Der komplementäre Graph $\overline{G}$ zu $G = (V, E)$ ist

$$\overline{G} = (V, \overline{E}) \text{ mit } \{x,y\} \in \overline{E} \text{ gdw } \{x,y\} \notin E.$$

Lemma: $U \subseteq V$ ist vertex cover
gdw $V \setminus U$ ist Clique in $\overline{G}$

Also: $(G, k) \in Y_{VC}$ gdw $(\overline{G}, n-k) \in Y_{CLIQUE}$

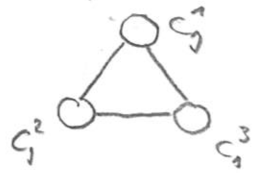
3SAT $\leq_m$ VC :

Instanz von 3SAT : Variablen $x_1, \dots, x_n$
Klauseln $C_1, \dots, C_m$

Definiere einen Graphen wie folgt.

Für jede Variable x_i :

Für jede Klausel $C_j = \{ l_{j1}, l_{j2}, l_{j3} \}$:



Jedes vertex cover U enthält mindestens
eines von x_i, $\overline{x_i}$
zwei aus c_j^1 c_j^2 c_j^3.
$\Rightarrow |U| \geq n + 2m$

Vertex cover U mit $|U| = 2m + n$ enthält genau diese !

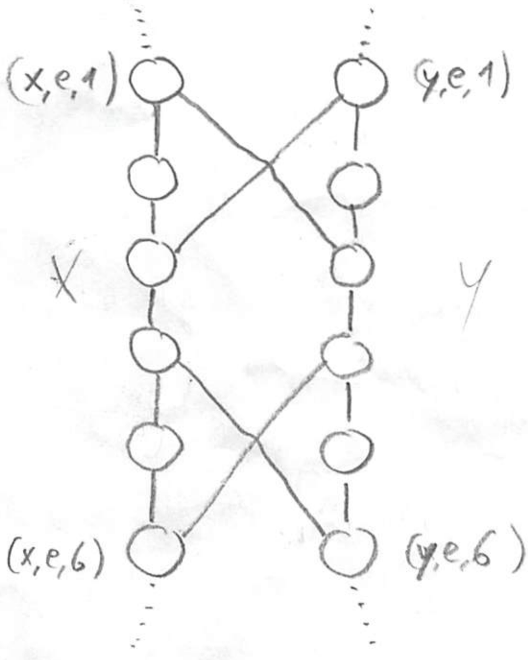
erfüllende Bewertung definiert vertex cover der grösse $2m+n$, und umgekehrt.

VC $\leq_m$ HC :

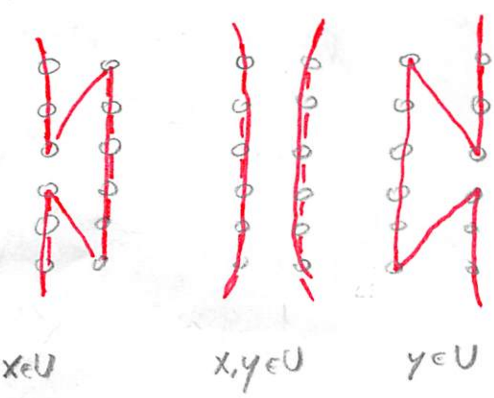
Instanz von VC : $G = (V, E)$, $k \in \mathbb{N}$.

Definiere neuen Graphen G' wie folgt:

Für $x \overset{e}{—} y$ in G :



3 Möglichkeiten wie ein Hamilton-Zyklus durchlaufen kann.



Komplexitätsklassen NPO PO (Folie)

- analog zu P, NP f. Entscheidungsprobleme

- Minimum Path, Minimum Vertex Cover, Maximum Clique $\in$ NPO.

$P \in NPO \longrightarrow P_D \in NP$:

input $x \in I_P$, $k \in \mathbb{N}$
guess y $|y| \le q(|x|)$ — $q(|x|)$ Schritte
if $y \in S(x)$ then — polynomial
  $m := m(x,y)$ — polynomial
if $m \ge k$ then accept
  else reject

$\square$

Minimum Path ist in PO:

Breitensuche findet minimalen Weg von s nach t in polynomialer Zeit. (erklären, falls unbekannt)

$P \in PO \rightarrow P_D \in P$

Reduktionsbegriff für allgemeine Probleme: Turing-Reduzierbarkeit

Orakel für ein Problem P:
Unterprogramm, das P löst, ohne Kosten ($\substack{\text{Zeit}\\\text{Speicher}}$)

$P \le_T Q$: P kann in polynomialer Zeit mit einem Orakel für Q gelöst werden.

Wie $\le_m$: reflexiv, transitiv. $\equiv_T$ induzierte Äquivalenz

$P \le_m Q \Rightarrow P \le_T Q$ f. Entscheidungsprobleme (nicht umgekehrt!)

Optimierungsprobleme

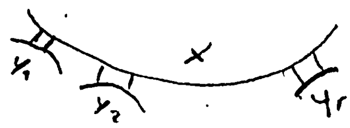
P ist NP-hart, falls f. alle $Q \in NP$ $Q \le_T P_E$

P_D NP-vollständig $\Longrightarrow$ P NP-hart

$\rightsquigarrow$ $P \neq NP \Rightarrow P_O \neq NPO$

II bezeichne dieses Gadget.

Seien $y_1, \ldots, y_r$ die Nachbarn von x.
Hänge die gadgets für die Kanten $\{x,y_1\} \ldots \{x,y_r\}$ zu einer Kette zusammen:
(f. jedes $x \in V$)



Ausserdem: k neue Knoten $a_1 \ldots a_k$

Endpunkte aller Ketten werden an $a_1, \ldots, a_k$ gehängt.

Vertex Cover $U = \{u_1, \ldots, u_k\}$ definiert Hamilton-Kreis in G'

a_1 – Kette(u_1) – a_2 Kette(u_2) – a_3 ... a_k Kette(u_k) – a_1

durchlaufe beide Seiten der Gadgets, wo Nachbar $\notin U$.

Umgekehrt: Hamilton-Kreis zerfällt in k teile

$$a_1 - \underset{x_1}{\text{Kette}} - a_{i_2} - \underset{x_2}{\text{Kette}} - a_{i_3} - \ldots \; a_{i_k} - \underset{x_k}{\text{Kette}} - a_1$$

Vertex-Cover: $\{x_1 \ldots x_k\}$.

Jeder Kante entspricht ein gadget, jeder wird durchlaufen □

Optimierungsprobleme

Definition (Folie)

Beispiele: Minimum Path
Folie Minimum Vertex Cover
Maximum Clique

} Probleme benannt wie im Buch!

Ähnlichkeit mit Entscheidungsproblemen (Erreichbarkeit, VC, Clique)
kein Zufall:

Problemvarianten (Folie!)

$$\max(m^*, k) = m^* \iff m^* \geq k$$

Erklärung.

$$\min(m^*, k) = m^* \iff m^* \leq k$$

Traveling Salesperson (Folie) auch als Graph repräsentierbar

ist NP-hart. Entscheidungsproblem TSP_D NP-vollständig

Reduktion $HC \leq_m TSP_D$:

gegeben Graph $G=(V,E)$ $V=\{v_1,\dots,v_n\}$.

Instanz von TSP_D :

- Städte $v_1,\dots,v_n$
- Distanz $D(i,j) = \begin{cases} 1 & \{v_i,v_j\} \in E \\ 2 & \text{sonst.} \end{cases}$
- Schranke $k=n$

Jede Tour hat die Kosten $\geq n$.
Hamilton-Kreis in G ist Tour der Länge n.
Umgekehrt: Tour der Länge n benutzt nur 1-Strecken,
diese sind Kanten in G $\rightarrow$ Hamilton-Kreis. $\square$

<u>Relation zw. Problemvarianten:</u>

Offensichtlich: $P_D \leq_T P_E \leq_T P_C$.

$P_E \leq_T P_D$: $|m(x,y)| \leq p(|x|)$ f. alle $y \in S(x)$.

Also: $m(x,y) < 2^{p(|x|)}$ $\therefore$ Binärsuche findet Wert in $O(p(|x|))$ Schritten.

$P_C \leq P_E$ im Fall MAXIMUM CLIQUE:

$G=(V,E)$, $v \in V$ $\quad G(v) :=$ induzierter Graph auf $\{v\} \cup \Gamma(v)$

$G^-(v) :=$ — " — $\Gamma(v)$ (ohne v)

MaxClique(G)
  $k :=$ MAXIMUM CLIQUE$_E$(G)
  if $k=1$ then return any $v \in V$
  Find node v with MAXIMUM CLIQUE$_E$($G(v)$) $= k$
  return $\{v\} \cup$ MaxClique($G^-(v)$)
end.

Allgemein: unbekannt.

Aber: $\boxed{\text{Ist } P_D \text{ NP-vollständig} \Rightarrow P_C \leq_T P_D}$

ObdA sei P maximierungsproblem.

Definiere neues Problem $P' \in NPO$ mit $P_C \leq_T P_E'$

Da P_D NP-vollständig ist: $P_E' \leq_T P_D' \leq_T P_D$ also $P_C \leq_T P_D$

$$I_{P'} = I_P \qquad S_{P'}(x) = S_P(x) \quad \text{f. alle } x \in I_P .$$

Sei $|y| \leq q(|x|)$ f. alle $y \in S_P(x)$.

$|S_P(x)| \leq 2^{q(|x|)}$

F. y sei $\lambda(y) :=$ Rang v. y in d. lexikographischen Ordnung

$\lambda(y) < 2^{q(|x|)}$ (= binäres Wert der Codierung)

Definiere nun:

$$m_{P'}(x,y) := 2^{q(|x|)+1} m_P(x,y) + \lambda(y)$$

F. jedes x gibt es genau eine optimale Lösung $y^* := y^*_{P'}(x)$ in P'

$m_{P'}(x,y) > m_{P'}(x,y') \to m_P(x,y) > m_P(x,y')$

Also. $y^* \in S_P^*(x)$. Optimal in P

Bei input $x \in I_P$ berechne $P_E'(x) = m^*_{P'}(x) = m_P(x,y^*)$

Nun ist $\lambda(y^*) = m^*_{P'}(x) \bmod 2^{p(|x|)+1}$,

daraus erhält man y^*, also $P_C \leq_T P_E'$

$\square$