27.10.2004

Komplexitätstheorie

<u>Theorem 6</u> : Falls P = NP ist , dann ist auch E = NE .

<u>Padding</u> : Technik zum übertragen von Kollapsresultaten nach oben

Sei $\# \in \Sigma$ ein neues Symbol .
Für $w \in \Sigma^*$ ist pad (w) : = $w \underbrace{\# \ldots \#}_{2^{|w|} - |w|}$, also $|pad(w)| = 2^{|w|}$

$pad(L) = \{ pad(w) \,;\, w \in L \}$

Sei nun $L \in NE$, akzeptiert von NTM T in Zeit $O(2^{k*n})$
Ł Maschine T´ : bei input pad(w) : testet ob input von der Form pad(w) ist
Falls ja : verhält sich genau wie T bei input w .
T´ erkennt pad (L) in Zeit $O(2^{k*|w|}) = O(|pad(w)|^k)$
Also ist $pad(L) \in NP$

Da nach Annahme P = NP , gibt es DTM T´´ , die pad(L) in Zeit $O(|pad(w)|^{k'})$ akzeptiert .

Maschine T* : input w wird ergänzt zu pad(w) $O(2^{|w|})$
durch wie T´´ $O(|pad(w)|^{k'})$

T* akzeptiert L ! Laufzeit : $O(2^{|w|} + 2^{k'*|w|})$

Damit ist $L \in E$!

□

2. Die Klassen P und NP

Übersicht:

- Probleme in NP und P
- NP-vollständige Probleme
- Struktur von NP
- Relativierung

Probleme in NP und P

a) klassisches Beispiel : SAT

gegeben : aussagenlogische Formel F in KNF
Frage : ist F erfüllbar ?

KNF heisst : $F = C_1 \wedge \ldots \wedge C_m$ C_i : Klauseln
$C = a_1 \vee \ldots \vee a_k$
a_j : Literale $a = x$ oder $a = \text{not}(x)$

F ist erfüllbar , wenn folgendes existiert α : Var -> {0 , 1} , so dass α (F) = 1

SAT ϵ NP : Nichtdeterministischer Algorithmus „rät" Variablenbelegung

□

Spezialfälle von SAT :

k–SAT : jede Klausel enthält genau k Literale .
Horn-SAT : jede Klausel enthält höchstens ein positives Literal .

$C = x \vee \text{not}(y_1) \vee \ldots \vee \text{not}(y_k)$

Es gilt : Horn-SAT ϵ P , 2-SAT ϵ P

b) Graphen

Graphenprobleme :

k-Färbbarkeit ($\in$ NP)

gegeben : Graph G (V , E)

Frage : gibt es eine Färbung c : V -> { 1,...,k}

d.h. : { u,v } $\in$ E -> c(u) $\neq$ c(v)

Spezialfall : 2-Färbbarkeit ($\in$ P)

Ein Graph ist 2-färbbar , wenn er bipartit ist.

Erläuterung : ein Graph G ist genau dann **bipartit** , wenn er keinen Kreis ungerader Länge als Teilgraphen enthält.

Vertex Cover („Knotenüberdeckung") ($\in$ NP)

gegeben : Graph G = (V,E) , k $\in$ Nat

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

d.h.: für alle e $\in$ E ist U $\cap$ e $\neq$ {} .

ähnliches Problem : Matching ($\in$ P)

gegeben : Graph G = (G,V) , k $\in$ Nat

Frage : gibt es eine Matching M $\subseteq$ E mit |M| $\geq$ k ?

d.h.: für alle e, e´ $\in$ M ist e $\cap$ e´ = {} .

Proposition 1 : $L \in NP$ gdw. es gibt $R_L(\ ,\ ) \in P$ und $k \in Nat$ mit
für alle x $x \in L$ <-> existiert ein w mit $|w| \leq |x|^k$ und $R_L(w,x)$

Beweis : „ "

Sei T DTM für R_L mit Laufzeit $O(|w| + |x|^{k'})$

NTM für L : bei Eingabe von x rate w mit $|w| \leq |x|^k$
und überprüfe , dass $R_L(w,x)$ gilt

Laufzeit : $|x|^k + O(|w| + |x|)^{k'} = |x|^k + O(|x| + |x|^k)^{k'} = O(|x|^{k*k'})$

„ "

Zeuge für $w \in L$: Berechnung einer NTM T , die L akzeptiert .

$x \in L$ gdw. Es gibt $S_0, \ldots, S_t$ wobei S_t akzeptiert.
$t = |x|^k$, jedes S_i ist $|S_i| \leq |x|^{k'}$
$|S_0, \ldots, S_t|$ ist $O(|x|^{k*k'})$.

Es ist leicht ($\in P$) zu entscheiden , ob $S_0, \ldots, S_t$ akzeptierende Berechnung von T bei x ist .

□

z.B. : CO-PRIM : gegeben : $n \in Nat$
Frage : ist n in prim ? ($\in NP$, wegen Proposition1)

Graphenisomorphie ($\in NP$, wegen Proposition1) :

gegeben : G_1, G_2
Frage : ist G_1 isomorph zu G_2 ?

Tausende Probleme aus allen Bereichen der Informatik sind in NP,
aber wahrscheinlich nicht in P.

NP-Vollständigkeit

Ziel: die „schwierigsten“ Probleme in NP zu finden!

Hilfsmittel: Reduktion , dazu brauchen wir:

Übersetzer (transducer)

DTM T mit einem Endzustand $F = \{q_f\}$.
ObdA endet jede Berechnung in q_f so , dass alleSchreib-/Leseköpfe links stehen .

berechnet $f : \Sigma^* \quad \Sigma^*$: output steht am Ende auf Band 1

$FP = \{ f$; es existiert ein DTM T mit T berechnet f und $TIME_T(x) = O(|x|^k) \}$

Definition: L ist p-reduzierbar auf L´ ($L \leq_p L'$)
falls es gibt $f \in FP$ mit :
für alle $x \in L$ gdw. $f(x) \in L'$

anschaulich : L ist höchstens so schwierig wie L´ , denn ich kann L durch Verwendung von L´ lösen !

L heißt **NP-schwer** , falls für alle $L' \in NP$ $L' \leq L$.
L heißt **NP- vollständig** , falls $L \in NP$ und NP-schwer .

Eigenschaften:

- Reduktion ist transitiv :
  L NP-schwer und $L \leq_p L'$, dann ist L´ NP-schwer
- falls L NP-vollständig ist , dann ist $L \in P$ gdw. P =NP .
- SAT, 3-SAT, 3-Färbbarkeit, Vertex Cover sind NP-vollständig.