Aufwandsvergleich

n – Größe der Eingabe

$f(n)$ – Anzahl der Schritte bis zur Lösung – Aufwand

n / f	5	10	50	100
n^2	0,000025 s	0,0001 s	0,0025 s	0,01 s
n^5	0,003125 s	0,1 s	312,5 s	ca. 3 Std.
2^n	0,000032 s	0,001024 s	ca. 36 Jahre	ca. 10^{17} Jahre
n^n	0,003125 s	ca. 3Std.	$> 10^{71}$ Jahre	

Zeitkomplexität – Definition I

Definition:
Sei $M = (k, X, Z, z_0, Q, \delta, F)$ eine deterministische akzeptierende k-Band-TURING-Maschine, die bei jeder Eingabe einen Stopzustand erreicht. Ferner sei $r = \#(X)$.

i) Mit $t_M(w)$ bezeichnen wir die Anzahl der (direkten) Überführungsschritte, die M ausführt, um die Anfangskonfiguration $(z_0, \lambda, w, \lambda, *, \lambda, *, \ldots, \lambda, *)$ in die zugehörige Endkonfiguration zu transformieren, und nennen $t_M(w)$ die Zeitkomplexität von w bezüglich M.

Zeitkomplexität – Definition II

ii) Für eine natürliche Zahl n setzen wir

$$t_M(n) = max\{t_M(w) : |w| = n\}$$

und

$$\overline{t_M}(n) = \frac{\sum_{|w|=n} t_M(w)}{r^n}.$$

Die Funktionen t_M und $\overline{t_M}$ von $\mathbf{N}$ in $\mathbf{N}$ heißen Zeitkomplexität des ungünstigsten Falles (worst-case time complexity) und durchschnittliche Zeitkomplexität (average time complexity) von M.

Raumkomplexität – Definition I

Definition:
Sei $M = (k, X, Z, z_0, Q, \delta, F)$ eine deterministische akzeptierende k-Band-TURING-Maschine, die bei jeder Eingabe einen Stopzustand erreicht. Ferner sei $r = \#(X)$.

i) Mit $s_M(w)$ bezeichnen wir die Anzahl der Zellen auf den Arbeitsbändern, über denen während der Überführung der Anfangskonfiguration $(z_0, \lambda, w, \lambda, *, \lambda, *, \ldots, \lambda, *)$ in die zugehörige Endkonfiguration mindestens einmal der Lese-/Schreibkopf stand. $s_M(w)$ heißt die Raumkomplexität von w auf M.

Raumkomplexität – Definition II

ii) Für $n \in \mathbf{N}$ setzen wir

$$s_M(n) = max\{s_M(w) : |w| = n\}$$

und

$$\overline{s_M}(n) = \frac{\sum_{|w|=n} s_M(w)}{r^n}.$$

s_M und $\overline{s_M}$ heißen Raumkomplexität des ungünstigsten Falles bzw. durchschnittliche Raumkomplexität von M.

Zwei Sätze zur Zeitkomplexität

Satz: Zu jeder k-Band-$\textsc{Turing}$-Maschine M (die auf jeder Eingabe stoppt) gibt es eine $\textsc{Turing}$-Maschine M' (die auf jeder Eingabe stoppt) derart, dass

$$T(M') = T(M) \quad \text{und} \quad t_{M'}(n) = O((t_M(n))^2)$$

gelten.

Satz: Zu jeder Funktion g von **N** in **N** gibt es eine rekursive Sprache L derart, dass für jede $\textsc{Turing}$-Maschine M (die auf jeder Eingabe stoppt) mit $T(M) = L$

$$t_M(n) \geq g(n)$$

gilt.

Zeitschranken I

Definition: Es seien $t : \mathbf{N} \to \mathbf{N}$ eine Funktion, $f : X^* \to X^*$ eine $\textsc{Turing}$-berechenbare Funktion und $M = (X', Z, z_0, Q, \delta)$ eine deterministische $\textsc{Turing}$-Maschine mit $X \subseteq X'$ und $f_M = f$. Wir sagen, dass M die Funktion f in der Zeit t berechnet, wenn M für jedes Wort w aus dem Definitionsbereich von f nach höchstens $t(|w|)$ Überführungsschritten einen Stopzustand erreicht.

Definition: Es seien $t : \mathbf{N} \to \mathbf{N}$ eine Funktion und $L \subset X^*$ eine rekursiv-aufzählbare Sprache und $M = (X', Z, z_0, Q, \delta, F)$ eine akzeptierende (deterministische oder nichtdeterministische) $\textsc{Turing}$-Maschine mit $X \subset X'$ und $L = T(M)$. Wir sagen, dass M die Sprache L in der Zeit t akzeptiert, wenn M für jedes Wort $w \in L$ nach höchstens $t(|w|)$ Überführungsschritten einen akzeptierenden Stopzustand erreicht.

Zeitschranken II

Definition: Es seien $t : \mathbf{N} \to \mathbf{N}$ eine Funktion und $L \subset X^*$ eine rekursive Sprache und $M = (X', Z, z_0, Q, \delta, F)$ eine akzeptierende deterministische TURING-Maschine mit $X \subset X'$ und $L = T(M)$. Wir sagen, dass M die Sprache L in der Zeit t entscheidet, wenn M für jedes Wort $w \in X^*$ nach höchstens $t(|w|)$ Überführungsschritten einen Stopzustand erreicht.

Definition:
P sei die Menge aller Sprachen, die von deterministischen akzeptierenden TURING-Maschinen in polynomialer Zeit entschieden werden können.
NP sei die Menge aller Sprachen, die von nichtdeterministischen akzeptierenden TURING-Maschine in polynomialer Zeit akzeptiert werden können.

Das Erfüllbarkeitsproblem SAT

Alternative – aussagenlogischen Ausdruck in n Booleschen Variablen der Form
$A(x_1, x_2, ..., x_n) = x_{i_1}^{\sigma_{i_1}} \vee x_{i_2}^{\sigma_{i_2}} \vee ... \vee x_{i_r}^{\sigma_{i_r}}$,
wobei $i_j \in \{1, 2, \ldots, n\}$ und $\sigma_{i_j} \in \{0, 1\}$ für $1 \leq j \leq r$ gelten,
x^1 die Identität und x^0 die Negation sind.

Belegung – $\alpha : x_i \to a_i \in \{0, 1\}$

Wert – $w_\alpha(A(x_1, \ldots, x_n)) = 1$ genau dann, wenn $a_{i_j} = \sigma_{i_j}$ für ein j, $1 \leq j \leq r$

Problem:	SAT
Gegeben:	n Boolesche Variable $x_1, x_2, ..., x_n$ und m Alternativen $A_i(x_1, x_2, ..., x_n)$, $1 \leq i \leq m$
Frage:	Gibt es eine Belegung $\alpha : x_i \to a_i \in \{0, 1\}$ derart, dass $w_\alpha(A_j(x_1, x_2, ..., x_n)) = 1$ für $1 \leq j \leq m$ gilt.

Transformierbarkeit von Problemen

Definition: Seien L_1 und L_2 zwei Sprachen. Wir sagen, dass L_1 auf L_2 transformierbar ist, falls es eine Funktion τ gibt, die L_1 auf L_2 so abbildet, dass $a \in L_1$ genau dann gilt, wenn $\tau(a) \in L_2$ ist.

Definition: Wir sagen, dass die Sprache L_1 polynomial auf die Sprache L_2 transformierbar ist, wenn L_1 durch eine Funktion τ auf L_2 transformiert wird, die mit polynomialer Zeitkomplexität berechnet werden kann, d.h. τ wird von einer (deterministischen) TURING-MASCHINE M in der Zeit p berechnet, wobei p ein Polynom ist.

Bezeichnung: $L_1 \alpha L_2$ für polynomiale Transformierbarkeit von L_1 auf L_2

Lemma: i) α ist eine transitive Relation auf der Menge der Sprachen und damit eine (reflexive) Halbordnung.
ii) Aus $L_2 \in \mathbf{P}$ und $L_1 \alpha L_2$ folgt $L_1 \in \mathbf{P}$.
iii) Aus $L_2 \in \mathbf{NP}$ und $L_1 \alpha L_2$ folgt $L_1 \in \mathbf{NP}$.

SAT versus Cliquenproblem I

Es sei $G = (V, E)$ ein Graph.

Eine Teilmenge $V' \subseteq V$ heißt *Clique* in G, falls $(v, v') \in E$ für alle paarweise verschiedenen $v, v' \in V'$ gilt.

Cliquenproblem:

Gegeben: Graph $G = (V, E)$, natürliche Zahl $k \geq 1$,
Frage: Gibt es eine k-elementige Clique in G ?

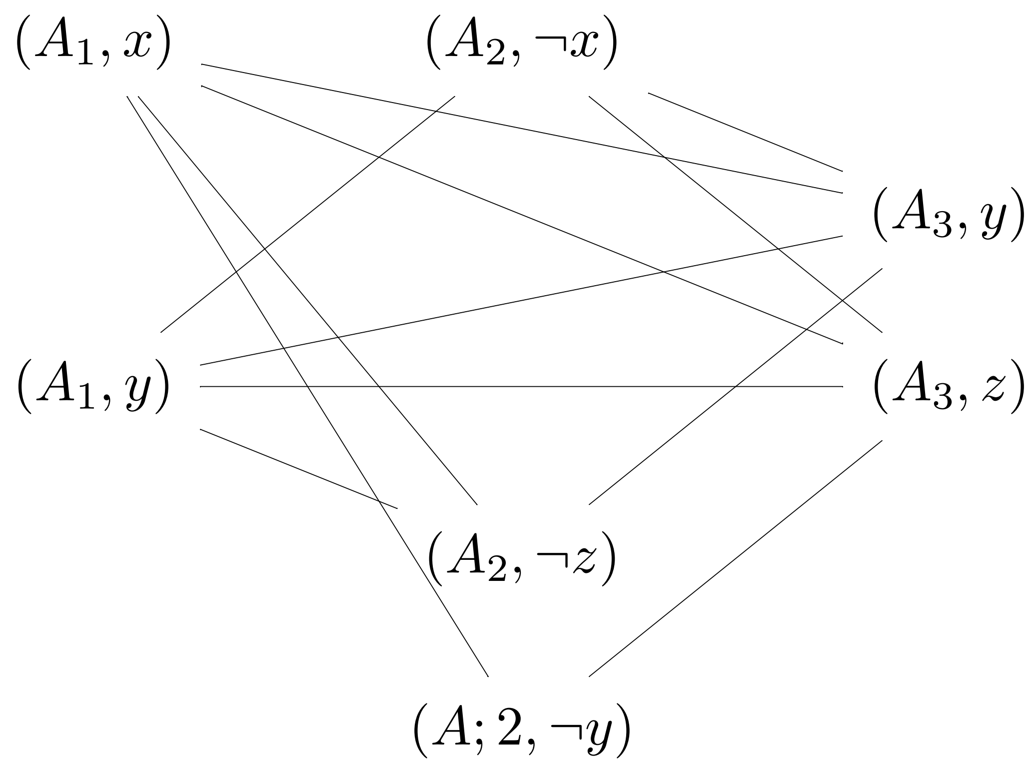
SAT gegeben durch
$A_i(x_1, x_2, \ldots, x_n) = x_{i,1}^{\sigma_{i,1}} \vee x_{i,2}^{\sigma_{i,2}} \vee \ldots x_{i,r_i}^{\sigma_{i,r_i}}, \quad 1 \leq i \leq m,$
Konstruktion eines Graphen $G = (V, E)$ durch
$V = \{(A_i, x_{i,j}^{\sigma_{i,j}}) : 1 \leq i \leq m, 1 \leq j \leq r_i\}$
(A, x^σ) und $(A', x'^{\sigma'})$ werden genau dann durch eine Kante verbunden,
wenn $A \neq A', x \neq x'$ oder $A \neq A', x = x', \sigma = \sigma'$
$k = m$.

SAT versus Cliquenproblem II

$A_1 = x \vee y,$

$A_2 = \neg x \vee \neg y \vee \neg z,$

$A_3 = y \vee z$



Geschäftsreisenden-Problem versus HAMILTON-Kreis-Problem

Problem des Geschäftsreisenden:

Gegeben: $n \geq 1$, n Städte $C_1, C_2, ..., C_n$,
die Entfernungen $d(C_i, C_j)$ zwischen den Städten C_i und C_j
für $1 \leq i, j \leq n$, $B \geq 0$

Frage: Gibt es eine Rundreise $C_{i_1}, C_{i_2}, ..., C_{i_n}$ durch alle Städte,
für die $(\sum_{j=1}^{n-1} d(C_{i_j}, C_{i_{j+1}}) + d(C_{i_n}, C_{i_1}) \leq B$ gilt?

Problem der Existenz von HAMILTON*-Kreisen*:

Gegeben: Graph $G = (V, E)$ mit $\#(V) = n$

Frage: Enthält G einen HAMILTON-Kreis,
d.h. gibt es eine Folge $v_1, v_2, ..., v_n$ von paarweise verschiedenen
Knoten des Graphen G so, dass $(v_i, v_{i+1}) \in E$ für $1 \leq i \leq n$
und $(v_n, v_1) \in E$ gelten?

NP-Vollständigkeit

Definition:
Eine Sprache L heißt **NP**-vollständig, wenn folgende Bedingungen erfüllt sind:
i) $L \in \mathbf{NP}$,
ii) $L' \alpha L$ gilt für jede Sprache $L' \in \mathbf{NP}$.

Satz:
Die folgenden Aussagen sind gleichwertig:
i) $\mathbf{P} = \mathbf{NP}$.
ii) $L \in \mathbf{P}$ gilt für jede **NP**-vollständige Sprache L.
iii) $L \in \mathbf{P}$ gilt für eine **NP**-vollständige Sprache L.

Satz:
SAT ist **NP**-vollständig.

Beweis der NP-Volständigkeit von SAT I

L akzeptiert von nichtdeterministischer TURING-Maschine
$M = (\{a_1, a_2, \ldots, a_r\}, \{z_0, z_1, \ldots, z_m\}, z_0, \{z_1\}, \delta, \{z_1\}),\ \ * = x_0,$
in polynomialer Zeitschranke p

$w = a_{i_1} a_{i_2} \ldots a_{i_n}$ zu Beginn in Zellen 1 bis n,
$t = p(n)$

Z_{ij}, $1 \leq i \leq t, 0 \leq j \leq m$,
Z_{ij} nimmt genau dann den Wert *wahr* an, wenn M zur Zeit i im Zustand z_j ist,
H_{ik}, $1 \leq i \leq t, -t \leq k \leq t$,
H_{ik} nimmt genau dann den Wert *wahr* an, wenn der Kopf von M zur Zeit i über der Zelle k steht,
S_{ikl}, $1 \leq i \leq t, -t \leq k \leq t, 0 \leq l \leq r$,
S_{ikl} nimmt genau dann den Wert *wahr* an, wenn zur Zeit i in der Zelle k auf dem Band von M der Buchstabe a_l steht

Beweis der NP-Volständigkeit von SAT II

(1) $Z_{i0} \vee Z_{i1} \vee \ldots \vee Z_{im}$ für $1 \leq i \leq t$,
M befindet sich zur Zeit i in mindestens einem Zustand z_j

(2) $\neg Z_{ij} \vee \neg Z_{ij'}$ für $1 \leq i \leq t, 0 \leq j < j' \leq m$,
M befindet sich zur Zeit i in höchstens einem Zustand z_j

(3) $H_{i,-t} \vee H_{i,-t+1} \vee \ldots \vee H_{it}$ für $1 \leq i \leq t$,
Kopf von M befindet sich zur Zeit i über mindestens einer Zelle

(4) $\neg H_{ik} \vee \neg H_{ik'}$ für $1 \leq i \leq t, -t \leq k < k' \leq t$,
Kopf von M befindet sich zur Zeit i über höchstens einer Zelle

(5) $S_{ik0} \vee S_{ik1} \vee \ldots \vee S_{ikr}$ für $1 \leq i \leq t, -t \leq k \leq t$,

(6) $\neg S_{ikl} \vee \neg S_{ikl'}$ für $1 \leq i \leq t, -t \leq k \leq t, 0 \leq l < l' \leq r$,
wegen (5) und (6) steht in der Zelle k zur Zeit i genau ein Buchstabe

Beweis der NP-Volständigkeit von SAT III

(7) Z_{10},

(8) H_{11},

(9) $S_{11i_1}, S_{12i_2}, \ldots, S_{1ni_n}$ und S_{1k0} für $-t \leq k \leq t, k \notin \{1, 2, ..., n\}$

Alternativen (7), (8) und (9) beschreiben die Anfangskonfiguration

(10) Z_{t1},

(10) sichert das Erreichen einer Endkonfiguration

(11) $(\neg Z_{ij} \vee \neg H_{ik} \vee \neg S_{ikl}$
$\vee(Z_{i+1,j_1} \wedge H_{i+1,k_1} \wedge S_{i+1,k,l_1}) \vee \ldots \vee (Z_{i+1,j_u} \wedge H_{i+1,k_u} \wedge S_{i+1,k,l_u}))$

für $1 \leq i \leq t-1$, $0 \leq j \neq 1 \leq m$, $-t \leq k \leq t$, $0 \leq l \leq r$,
und $\delta(z_j, a_l) = \{(z_{j_1}, a_{l_1}, d_1), (z_{j_2}, a_{l_2}, d_2), \ldots, (z_{j_u}, a_{l_u}, d_u)\}$

(11) beschreibt das Verhalten von M ohne Erreichen eines Endzustands

Beweis der NP-Volständigkeit von SAT IV

(12) $(\neg Z_{i1} \vee \neg H_{ik} \vee \neg S_{ikl} \vee (Z_{i+1,1} \wedge H_{i+1,k} \wedge S_{i+1,k,l}))$
für $1 \leq i \leq t-1$, $-t \leq k \leq t$, $0 \leq l \leq r$,
(12) sichert, dass die Konfiguration nicht mehr verändert wird, wenn der Endzustand erreicht wird

(13) $\neg S_{ikl} \vee \neg H_{ik'} \vee S_{ikl}$ für $1 \leq i \leq t, -t \leq k, k' \leq t, k \neq k', 0 \leq l \leq r$
wegen (17) erfolgt keine Änderung des Zelleninhalts, wenn sich der Kopf nicht über der Zelle befindet.

NP-vollständige Probleme I

Satz:
Ist die **NP**-vollständige Sprache L polynomial auf die Sprache L' aus **NP** transformierbar, so ist L' auch **NP**-vollständig.

Satz:
Das Cliquenproblem ist **NP**-vollständig.

Satz:
Das Problem der Existenz von HAMILTON-Kreisen ist **NP**-vollständig.

Satz:
Das Problem des Geschäftsreisenden ist **NP**-vollständig.

NP-vollständige Probleme II

Satz:

Das Problem der minimalen Rundreise

Gegeben: natürliche Zahl $n \geq 1$,
Städte $C_1, C_2, \ldots, C_n$ mit den Abständen $d(C_i, C_j)$, $1 \leq i, j \leq n$,

Frage: Wie groß ist der minimale Wert von

$d(C_{i_n}, C_{i_1}) + \Sigma_{j=1}^{n-1} d(C_{i_j}, C_{i_{j+1}})$,

wobei das Minimum über alle Permutation von $\{1, 2, \ldots n\}$ zu nehmen ist?

ist **NP**-vollständig.

NP-vollständige Probleme III

Satz:

Das Problem der (Knoten-)Färbbarkeit von Graphen

Gegeben:	Graph $G = (V, E)$ und natürliche Zahl $k \geq 3$
Frage:	Gibt es eine Färbung der Knoten von G mit k Farben, so daß durch eine Kante verbundene Knoten jeweils verschieden gefärbt sind?

ist **NP**-vollständig.

NP-vollständige Probleme IV

Satz:

Das Problem der Teilmengensumme

Gegeben: endliche Menge $A \subseteq \mathbf{N}$ und natürliche Zahl $b \in \mathbf{N}$

Frage: Gibt es eine Teilmenge $A' \subseteq A$ derart, dass $\sum_{a \in A'} a = b$ gilt?

ist **NP**-vollständig.

Satz:

Das Problem der Lösbarkeit diophantischer quadratischer Gleichungen

Gegeben: natürliche Zahlen a, b, c

Frage: Gibt es eine Lösung von $ax^2 + by = c$ in natürlichen Zahlen?

ist **NP**-vollständig.

Relatione Datenbanken I

Objekt	Name	Vorname	Immatrik.-nummer	Universität	Fakultät/ Fachbereich
1	Meyer	Heike	12345678	RWTH Aachen	Informatik
2	Schulz	Ulrike	21436587	TU München	Elektrotechn.
3	Müller	Heike	12348765	TU Dresden	Elektrotechn.
4	Muster	Fritz	56781234	TH Darmstadt	Mathematik.
5	Meyer	Ulrich	65874321	TU Berlin	Mathematik
6	Müller	Fritz	87654321	RWTH Aaachen	Informatik

$\{B_1, B_2, \ldots, B_r\} \succ A$ genau dann, wenn es eine Funktion f mit $f(B_{1,i}, B_{2,i}, \ldots B_{r,i}) = A_i$ für $1 \leq i \leq n$ gibt.

{Immatrik.-nummer} $\succ$ Name und {Name, Vorname} $\succ$ Immatrik.-nummer, aber nicht {Name} $\succ$ Vorname und {Vorname} $\succ$ Name.

Relatione Datenbanken II

Datenbank mit der Menge H von Attributen

Eine Teilmenge K von H heißt Schlüssel, falls $K \succ B$ für jedes $B \in H$ gilt.

Satz:

Das Problem der Existenz von Schlüsseln in einer Datenbank
Gegeben: Datenbank mit Menge H von Attributen, natürliche Zahl k
Frage: Gibt es einen Schlüssel K für F mit $\#(K) \leq k$?
ist **NP**-vollständig.