Komplexitätstheorie

Maike Buchin (RUB)

basierend auf dem Skript von

Hans Simon (RUB)

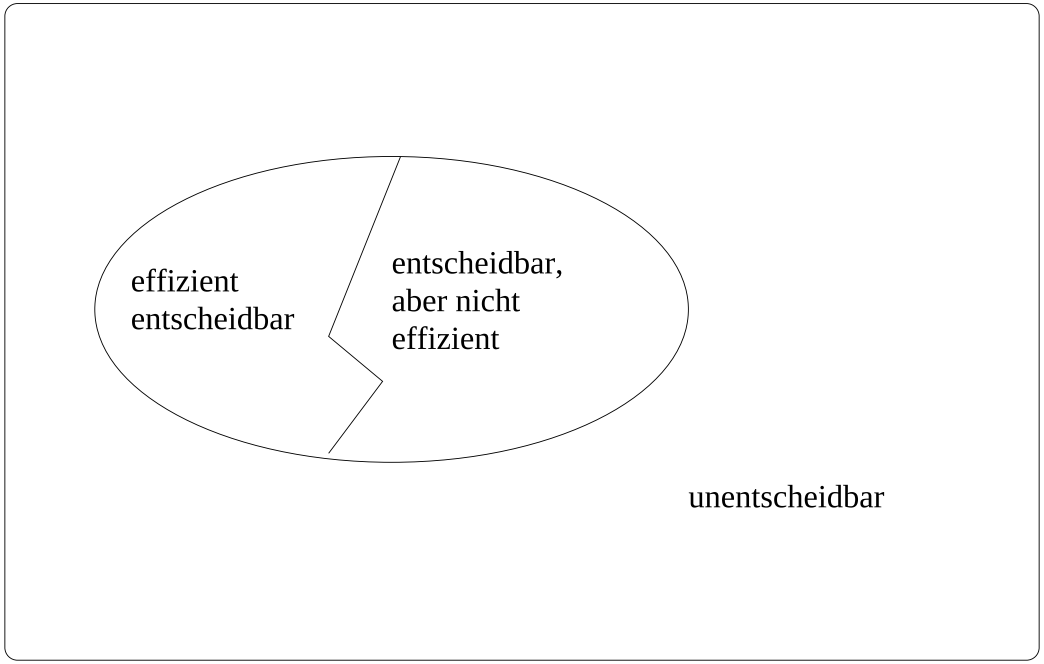
Lehrstuhl Mathematik und Informatik

Homepage: http://www.ruhr–uni–bochum.de/lmi

Zentrale Frage

Welche Funktionen sind effizient berechenbar ?

- In der Berechenbarkeitstheorie hatten wir die Grenze zwischen entscheidbaren und unentscheidbaren Problemen exploriert.
- Jetzt fragen wir uns ob ein Problem effizient lösbar ist, oder ob es inhärent einen hohen Ressourcenverbrauch hat.



Ressourcenverbrauch

Wir interessieren uns für den Ressourcenverbrauch in Form von

- Zeit (Rechenzeit) und
- Platz (Speicherplatz).

Um diesen zu beschreiben, machen wir Gebrauch von der (aus DiMa) bekannten Landau'schen O-Notation

$$O(f) = \{g : \mathbb{N}_0 \to \mathbb{N}_0 | \ \exists c, n_0 \in \mathbb{N}_0, \forall n \geq n_0 : \ g(n) \leq cf(n)\}.$$

Funktionen g aus $O(f)$ heißen *größenordnungsmäßig durch f beschränkt.*

Aus historischen Gründen schreibt man $g = O(f)$ anstelle von $g \in O(f)$.

Salopp gesprochen kann man sagen, dass das große O Konstanten und Terme nichtdominanter Größenordnung „schluckt“.

Zeit- und Platzschranken

Definition: Es seien S und T Funktionen von $\mathbb{N}_0$ nach $\mathbb{N}_0$ und M eine Mehrband-DTM.

1. M heißt $S(n)$-*platzbeschränkt*, wenn eine Rechnung von M auf einer Eingabe der Maximallänge n maximal $O(S(n))$ Bandzellen verbraucht.
2. M heißt $T(n)$-*zeitbeschränkt*, wenn eine Rechnung von M auf einer Eingabe der Maximallänge n maximal $O(T(n))$ Schritte dauert.

Diese Definition lässt sich analog auf Mehrband-NTM's übertragen.

Notation: Ab jetzt notieren wir die von einer TM M akzeptierte Sprache als $L(M)$ (statt, wie früher, als $T(M)$), weil wir das Symbol T für Zeitschranken reservieren möchten.

Effiziente Berechenbarkeit

- Obwohl auch Rechenzeiten der Form $T(n) = 10^8 n$ oder $T(n) = n^{1000}$ im Grunde inakzeptabel sind, läßt sich dennoch sagen, dass exponentielles Wachstum $T(n) = c \cdot 2^n$ schon für moderate Werte von n auch schnellste Rechenanlagen für Milliarden von Jahren beschäftigt.
- Daher hat sich die Sichtweise durchgesetzt, polynomielles Wachstum noch als „effizient“ zu betrachten, und superpolynomielles (oder gar exponentielles) Wachstum für „ineffizient“.

Deterministische Komplexitätsklassen

Definition:

- $DSpace(S)$ ist die Klasse aller Sprachen, die von einer $S(n)$-platzbeschränkten DTM akzeptiert werden können.
- $DTime(T)$ ist die Klasse aller Sprachen, die von einer $T(n)$-zeitbeschränkten DTM akzeptiert werden können.
- $P = \cup_{k\geq 1} DTime(n^k)$ ist die Klasse aller deterministisch in Polynomialzeit akzeptierbaren Sprachen.
- $PSpace = \cup_{k\geq 1} DSpace(n^k)$ ist die Klasse aller deterministisch in polynomiellem Platz akzeptierbaren Sprachen.

Nicht-Deterministische Komplexitätsklassen

Definition:

- $NSpace(S)$ ist die Klasse aller Sprachen, die von einer $S(n)$-platzbeschränkten NTM akzeptiert werden können.
- $NTime(T)$ ist die Klasse aller Sprachen, die von einer $T(n)$-zeitbeschränkten NTM akzeptiert werden können.
- $NP = \cup_{k\geq 1} NTime(n^k)$ ist die Klasse aller nicht-deterministisch in Polynomialzeit akzeptierbaren Sprachen.
- $NSpace = \cup_{k\geq 1} NSpace(n^k)$ ist die Klasse aller nicht-deterministisch in polynomiellem Platz akzeptierbaren Sprachen.

Einband vs. Mehrband TM

- Wie wir wissen, lassen sich Mehrband-TMs ohne Platzverlust und mit nur quadratischem „Blow-up“ bei der Zeitschranke durch Einband-TMs simulieren.
- Daher ändern sich die Komplexitätsklassen wie P, NP oder PSpace nicht, wenn wir in der Definition Einband- statt Mehrband-TMs gefordert hätten.
- Wir dürfen daher bei diesen Klassen mit unserem Standard Modell (mit nur einem Band) arbeiten, wann immer wir das für zweckmäßig halten.

DSpace = NSpace

Es hat sich gezeigt, dass

$$NSpace(S) \subseteq DSpace(S^2).$$

Für jede NTM existiert eine deterministische Simulation mit höchstens quadratischem „blow-up“ des Speicherplatzes (*Satz von Savitch*).

Daher erübrigt sich eine Unterscheidung von $DSpace$ und $NSpace$.

Zentrale Fragen der Komplexitätstheorie

1. Welche Probleme erfordern (im Wesentlichen) die gleichen Ressourcen an Zeit bzw. Platz und gehören daher in dieselbe Komplexitätsklasse ?
2. Wie ist das Verhältnis von Platz und Zeit?
3. Wie ist das Verhältnis von Determinismus und Nichtdeterminismus?

Wir wollen der dritten Frage anhand des berühmten ($\mathrm{P} \overset{?}{\neq} \mathrm{NP}$)-Problems nachgehen.

P vs. NP

Viele fundamentale Anwendungsprobleme gehören zur Klasse NP. Diese Probleme können bis heute nicht (deterministisch) in Polynomialzeit gelöst werden. Daher wird allgemein vermutet

$$P \neq NP \quad .$$

Man kann bis heute aber diese Vermutung nicht beweisen.

Immerhin ist es aber im Rahmen der NP-Vollständigkeitstheorie gelungen, „härteste Vertreter“ der Problemklasse NP dingfest zu machen. Diese werden NP-vollständige Probleme genannt.

Im Folgenden sehen wir einige Vertreter dieser Klasse.

Polynomielle Reduktion

Ein wichtiges Werkzeug zur Entwicklung der NP-Vollständigkeitstheorie sind:

Definition: Seien $L_1, L_2 \subseteq \Sigma^*$.

1. Wir sagen, L_1 ist *polynomiell reduzierbar* auf L_2 (in Zeichen: $L_1 \leq_{\text{pol}} L_2$), wenn eine Abbildung $f : \Sigma^* \to \Sigma^*$ existiert, so dass folgendes gilt:
   (1) $\forall w \in \Sigma^* :\ w \in L_1 \Leftrightarrow f(w) \in L_2$.
   (2) f ist von einer polynomiell zeitbeschränkten DTM berechenbar.
2. Eine Sprache $L_0 \subseteq \Sigma^*$ heißt *NP-vollständig*, falls
   (1) $L_0 \in \text{NP}$.
   (2) $\forall L \in NP :\ L \leq_{\text{pol}} L_0$.

   Falls die zweite Bedingung gilt, aber L_0 nicht notwendig zur Klasse NP gehört, dann heißt L_0 *NP-hart*.

Eigenschaften

Lemma: Falls $L_1 \leq_{pol} L_2$ und $L_2 \in P$, dann ist auch $L_1 \in P$.

Beweis: Die Frage, ob $w \in L_1$, kann wie folgt entschieden werden:

1. Berechne $f(w)$ mit der TM M.
2. Entscheide, ob $f(w) \in L_1$ mit der TM M_1.

Die Hintereinanderschaltung der Maschinen $M; M_1$ ist dabei wieder polynomiell zeitbeschränkt.

Anmerkung: Analog gilt: Falls $L_1 \leq_{pol} L_2$ und $L_2 \in NP$, dann ist auch $L_1 \in NP$.

Eigenschaften fortgesetzt

Lemma: Die Relation „$\leq_{pol}$“ ist transitiv. Ketten von Reduktionen ergeben also wieder eine Reduktion.

Beweis: analog zum vorherigen Beweis.

Folgerung: Aus $L_1 \leq_{pol} L_2$ und L_1 ist NP-hart folgt, dass L_2 NP-hart ist.

Ebenfalls folgt:

Lemma: Wenn ein NP-hartes Problem deterministisch in Polynomialzeit gelöst werden kann, dann folgt $P = NP$.

Satz: Sei A NP-vollständig. Dann gilt $A \in P \ \Leftrightarrow P = \mathrm{NP}$.

Folgerung: Zum Nachweis von $P = NP$ oder $P \neq NP$ würde es genügen, $A \in P$ oder $A \notin P$ für ein beliebiges NP-vollständiges Problem A zu zeigen.

Boolesche Formeln

Das erste NP-vollständige Problem beschäftigt sich mit Booleschen Formeln.

- Ein *Literal* ist eine negierte oder nicht negierte Boolesche Variable.
- Eine *Klausel* ist eine Disjunktion (Logisches Oder) von Literalen.
- eine *Formel in konjunktiver Normalform* (kurz: *CNF-Formel*) ist eine Konjunktion (Logisches Und) von Klauseln.

Beachte: eine Klausel kann auch aus einem einzigen Literal und eine CNF-Formel auch aus einer einzigen Klausel bestehen.

Beispiele für CNF-Formeln:

$$
\begin{aligned}
F_0 &= (x_1 \vee x_3) \wedge (x_1 \vee \overline{x_3}) \wedge \overline{x_1} \\
F_1 &= (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge x_2
\end{aligned}
$$

Erfüllbarkeitsproblem der Aussagenlogik

SAT ist folgendes Problem:

Eingabe eine CNF-Formel F über Variablen $x_1, \ldots, x_n$.

Frage Ist die Formel *erfüllbar*, d.h., existiert eine Belegung von $x_1, \ldots, x_n$ mit 0 oder 1, so dass F zu 1 ausgewertet wird?

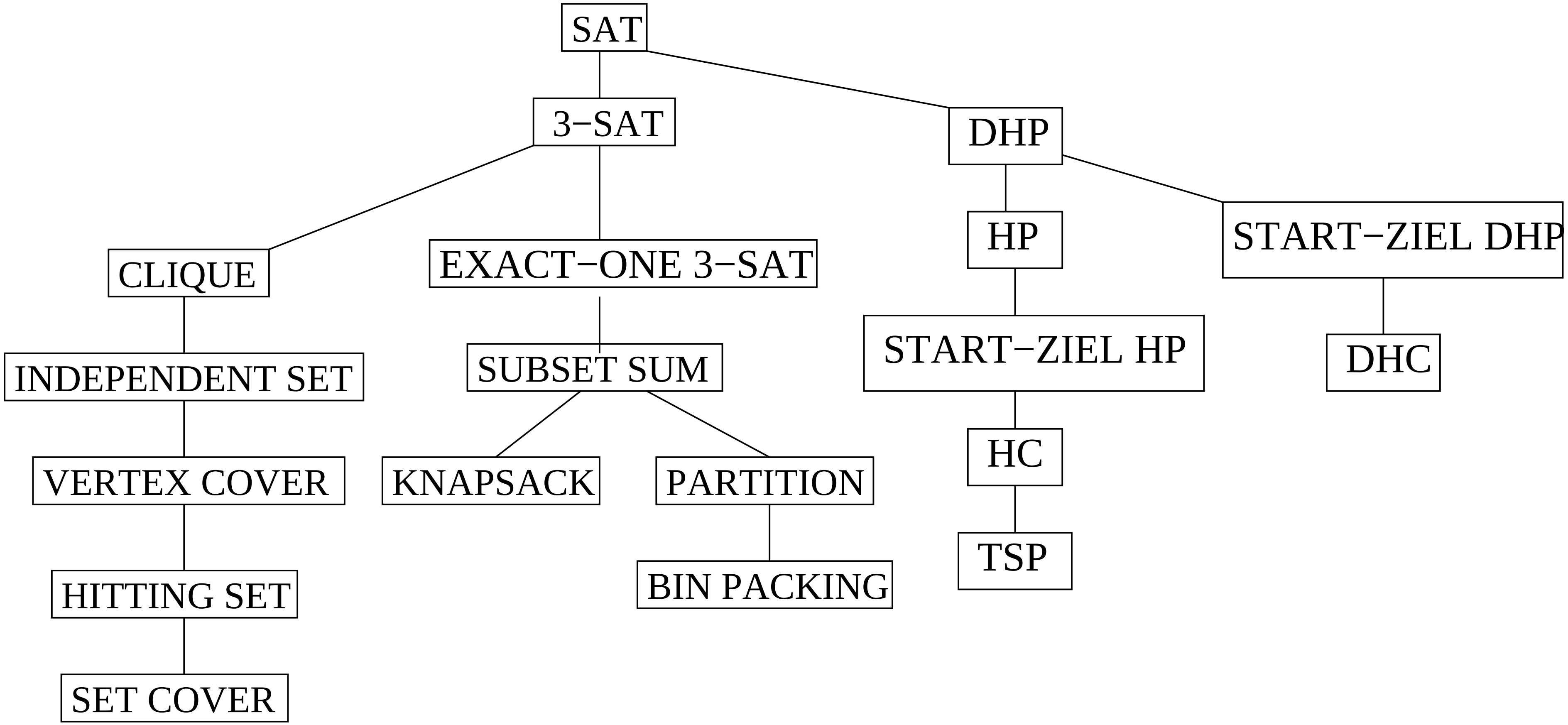
3SAT ist das eingeschränkte Problem, bei dem F nur aus Klauseln mit jeweils drei Literalen besteht.

Beispiel (fortgesetzt) $x_1 = 1, x_2 = 1, x_3 = 0$ erfüllt die obige Formel F_1. Die Formel F_0 hingegen ist nicht erfüllbar. Wieso ?

Satz von Cook (1971): SAT ist NP-vollständig. *(Beweis später)*

Weitere NP-vollständige Probleme

Ausgehend vom Satz von Cook lässt sich die NP-Vollständigkeit vieler Probleme zeigen.



Rate-Verifikationsprinzip

Vorgehen einer polynomiell zeitbeschränkten NTM M auf Eingabe w:

- Rate eine Art „Zertifikat“ $r \in \{0,1\}^*$ einer polynomiell in $n = |w|$ beschränkten Länge.
- Verifiziere mit Hilfe von r deterministisch, dass w zur Zielsprache $L(M)$ gehört.

Das Zertfikat r war intuitiv gesehen der Binärstring der Entscheidungen, die M trifft. Für konkrete formale Sprachen (=Probleme) aus NP hat das Zertifikat aber in der Regel eine sehr konkrete und plausible Form. Dies werden wir in den folgenden Beispielen sehen.

CLIQUE (vollständige Teilgraphen)

Eingabe: Ein ungerichteter Graph $G = (V, E)$ und ein $k \in \mathbb{N}_0$ mit $k \leq |V|$.

Frage: Existiert in G eine Clique der Größe mindestens k, d.h., eine Menge $C \subseteq V$ mit $|C| \geq k$, deren Knoten paarweise in G benachbart sind ?

Clique ist in NP:

Zertifikat: eine Menge $C \subseteq V$ der Größe k

Verifikation: Überprüfung, dass alle Knoten in C paarweise benachbart sind

CLIQUE ist NP-hart

Satz: 3-SAT $\leq_{pol}$CLIQUE.

Beweis: Sei $F = C_1 \wedge \ldots \wedge C_m$ mit $C_i = z_{i1} \vee z_{i2} \vee z_{i3}$ und $z_{ij} \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ eine Eingabe für 3-SAT. Wir ordnen der Formel F einen Graph $G = (V, E)$ und eine Zahl k zu, so dass gilt

$$C \text{ ist erfüllbar} \Leftrightarrow G \text{ besitzt einen Clique}$$

Schlüssel zur Reduktion:

- Zwei Literale z, z' heißen *kompatibel*, falls $z' \neq \bar{z}$.
- Mehrere Literale $z_1, \ldots, z_r$, $r \geq 2$, heißen *kompatibel*, wenn sie paarweise kompatibel sind.

Beobachtung: Es gibt genau dann eine Belegung, die $z_1, \ldots, z_r$ erfüllt, wenn $z_1, \ldots, z_r$ kompatibel sind.

CLIQUE ist NP-hart

Wir wählen $G = (V, E)$ und k als

$$\begin{aligned} V &= \{(1,1),(1,2),(1,3),(2,1),(2,2),(2,3),\ldots,(m,1),(m,2),(m,3)\} \\ E &= \{\{(i,j),(p,q)\} \mid i \neq p \text{ und } z_{ij} \neq \bar{z}_{pq}\} \\ k &= m \end{aligned}$$

Dann gilt:

F ist erfüllbar durch Belegung B

$\Leftrightarrow$ es gibt in jeder Klausel ein wahres Literal

$\Leftrightarrow$ es gibt kompatible Literale $z_{1,j_1}, \ldots, z_{m,j_m}$

$\Leftrightarrow$ es gibt Knoten $z_{1,j_1}, \ldots, z_{m,j_m}$ in G, die paarweise verbunden sind

$\Leftrightarrow$ G hat einen Clique der Größe k

VERTEXCOVER (Knotenüberdeckung)

Eingabe: Ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl $k \leq |V|$.

Frage: Existiert in G ein „Vertex Cover (Knotenüberdeckungsmenge)“ der Größe höchstens k, d.h., eine Menge $C \subseteq V$ mit $|C| \leq k$, die von jeder Kante aus E mindestens einen Randknoten enthält ?

VERTEXCOVER ist in NP.

Zertifikat: eine Menge $C \subseteq V$ der Größe k (oder kleiner)

Verifikation: Überprüfung, dass jede Kante in E mindestens einen Randknoten in C besitzt

VERTEXCOVER ist NP-hart

Satz: CLIQUE$\leq_{pol}$VERTEXCOVER.

Beweis: Ein Graph $G = (V, E)$ und eine Zahl k werden abgebildet auf

$\bar{G} = (V, \bar{E})$ mit $\bar{E} = \{\{u, v\} | u, v \in V, (u, v) \notin E\}$ und $m = |V| - k$

Dann gilt:

$$G \text{ besitzt einen k-Clique} \Leftrightarrow \bar{G} \text{ besitzt ein Vertexcover der Größe } m$$

Wähle dabei jeweils das Komplement des Clique bzw. Vertexcover als das jeweils andere.

SUBSETSUM (Teilsummen)

Eingabe: n Zahlen $S = a_1, \ldots, a_n \in \mathbb{N}_0$ und eine „Teilsummenzahl“ $b \in \mathbb{N}_0$.

Frage: Gibt es eine Menge $I \subseteq \{1, \ldots, n\}$, so dass $\sum_{i \in I} a_i = b$?

SUBSETSUM ist in NP.

Zertifikat: eine Menge $I \subseteq \{1, \ldots, n\}$

Verifikation: Überprüfung, dass die Zahlen a_i mit $i \in I$ sich genau zum Wert b aufaddieren

SUBSETSUM ist NP-hart

Satz: 3-SAT $\leq_{pol}$SUBSETSUM.

Beweis:

Sei $F = (z_{11} \vee z_{12} \vee z_{13}) \wedge \ldots \wedge (z_{m1} \vee z_{m2} \vee z_{m3})$ mit $z_{ij} \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$ eine 3-SAT Formel mit m Klauseln über n Variablen.

Wir wählen $b = \underbrace{4\ldots4}_{m}\ \underbrace{1\ldots1}_{n}$.

Zusätzlich wählen wir Zahlen $S = \{v_1, \ldots, v_n,\ v'_1, \ldots, v'_n,\ c_1 \ldots c_m,\ d_1, \ldots, d_m\}$

Alle diese Zahlen haben $m + n$ Ziffern, wir nennen die ersten m Ziffern den vorderen Block und die hinteren n Ziffern den hinteren Block.

Der k-ten Ziffer im vorderen Block ist die k-te Klausel zugeordnet, während der i-ten Ziffer im hinteren Block die i-te Variable zugeordnet ist.

SUBSETSUM ist NP-hart (fortgesetzt)

Bei den v_i steht im vorderen Block an der k-ten Stelle eine 1/2/3 wenn die Variable x_i 1/2/3-mal in der k-ten Klausel vorkommt.

Im hinteren Block der v_i steht eine 1 an Position i und 0 sonst.

Analog sind die v_i' bezüglich negativer Vorkommen von x_i gewählt.

Bei den c_j (bzw. d_j) steht an j-ter Stelle im vorderen Block eine 1 (bzw. 2). Alle übrigen Stellen sind 0.

Beobachtungen:

- Jede Lösung des SUBSETSUM Problems muss für $i = 1, \ldots, n$ entweder genau v_i oder v_i' enthalten.
- Eine Lösung des SUBSETSUM Problems muss für jedes $j = i, \ldots, m$ im vorderen Block einen positiven Beitrag eines v_i oder v_i' erhalten.

SUBSETSUM ist NP-hart (fortgesetzt)

Es gilt:

$$F \text{ hat erfüllende Belegung} \Leftrightarrow S \text{ hat eine Teilsumme mit Summe } b$$

Denn

⇒: Wähle v_i bzw. v_i' falls x_i auf 1 bzw. 0 gesetzt wird. Fülle mit c_j, d_j auf, um eine Lösung für S zu erhalten.

⇐: Eine korrekte Teilsumme muss für $i = 1, \ldots, n$ entweder v_i oder v_i' enthalten. Ausserdem muss sie für $j = 1, \ldots, m$ im vorderen Block mindestens einen positiven Beitrag durch ein v_i oder v_i' erhalten. Daher erfüllt eine Belegung, die für jedes $j = 1, \ldots, m$ das zu solch einem v_i oder v_i' entsprechende Literal wahr macht, die Formel F.

PARTITION (Zerlegung)

Eingabe: n Zahlen $a_1, \ldots, a_n \in \mathbb{N}_0$.

Frage: Kann man diese Zahlen in zwei gleichgroße Teilsummen zerlegen, d.h., existiert eine Teilmenge $I \subseteq \{1, \ldots, n\}$, so dass $\sum_{i \in I} a_i = \sum_{j \notin I} a_j$?

PARTITION ist in NP.

Zertifikat: eine Menge $I \subseteq \{1, \ldots, n\}$

Verifikation: Überprüfung, dass die Summe der Zahlen a_i mit $i \in I$ denselben Wert liefert wie die Summe der restlichen Zahlen

PARTITION ist NP-hart

Satz: SUBSETSUM$\leq_{pol}$PARTITION.

Beweis: Sei $(a_1, \dots, a_n, b)$ ein SUBSETSUM Problem. Wir setzen $M = \sum_{i=1}^{n} a_i$ und wählen $(a_1, \dots, a_n, M - b + 1, b + 1)$ als Eingabe für PARTITION. Beobachtung: Eine Lösung für Partition hat die Teilsumme $M + 1$.

Dann gilt die Äquivalenz der Lösungen, denn

$\Rightarrow$: Ist $I \subset \{1, \dots, n\}$ eine Lösung des SUBSETSUM Problem, so ist $I \cup \{n+1\}$ eine Lösung fuer PARTITION.

$\Leftarrow$: Ist $J \subset \{1, \dots, n+2\}$ eine Lösung des PARTITION Problem, dann enthält J entweder $n+1$ oder $n+2$, aber nicht beide. OBdA $n+1 \in J$. Dann lässt sich leicht nachrechnen, dass J ohne $n+1$ eine Lösung für SUBSETSUM ist.

Bin Packing (Behälterpackungsproblem)

Eingabe: n Objekte der Größen $a_1, \ldots, a_n \in \mathbb{N}_0$, m Behälter (=bins) der „Bingröße“ $b \in \mathbb{N}_0$.

Frage: Kann man die n Objekte so in die m Behälter verpacken, dass in jedem Behälter die Größen der in ihm enthaltenen Objekte sich zu höchstens b addieren, d.h., existiert eine Zerlegung von $\{1, \ldots, n\}$ in m disjunkte Teilmengen $I_1, \ldots, I_m$, so dass $\sum_{i \in I_j} a_i \leq b$ für alle $1 \leq j \leq m$ erfüllt ist ?

Bin Packing ist in NP.

Zertifikat: m Mengen $I_1, \ldots, I_m \subseteq \{1, \ldots, n\}$

Verifikation: Überprüfung, dass $I_1, \ldots, I_m$ eine Zerlegung von $\{1, \ldots, n\}$ bilden und dass $\sum_{i \in I_j} a_i \leq b$ für alle $1 \leq j \leq m$.

Bin Packing ist NP-hart

Satz: PARTITION $\leq_{pol}$ Bin Packing.

Beweis: Die Reduktion folgt leicht, indem $(a_1, \ldots, a_n)$ abgebildet wird auf $m = 2$ Behälter der Größe $b = \sum_{i=1}^{n} a_i/2$ und Objekte $(a_1, \ldots, a_n)$.

Hamiltonian Path (Hamiltonscher Pfad)

Directed Hamiltonian Path (DHP)

Eingabe: Ein gerichteter Graph $G = (V, E)$.

Frage: Gibt es in G einen gerichteten Hamiltonschen Pfad, d.h., können wir mit (gerichteten) Kanten aus E einen Pfad formen, der jeden Knoten aus V genau einmal durchläuft ?

Hamiltonian Path (HP)

das entsprechende Problem für ungerichtete Graphen

START-ZIEL-(D)HP

Variante von (D)HP: die Eingabe enthält neben dem Graphen $G = (V, E)$ zwei ausgezeichnete Knoten $s, t \in V$ und vom (gerichteten) Hamiltonschen Pfad wird gefordert, dass er in s beginnt und in t endet

Hamiltonian Circuit (Hamiltonscher Kreis)

Directed Hamiltonian Circuit (DHC)

Eingabe: Ein gerichteter Graph $G = (V, E)$.

Frage: Gibt es in G einen gerichteten Hamiltonschen Kreis, d.h., können wir mit (gerichteten) Kanten aus E einen Kreis formen, der (abgesehen von dem identischen Start- und Zielknoten) jeden Knoten aus V genau einmal durchläuft ?

Hamiltonian Circuit (HC)

das entsprechende Problem für ungerichtete Graphen

Hamiltonian Path/Circuit in NP

Alle diese Probleme, also DHP, HP, Start-Ziel-HP, DHC, und HC, sind in NP.

Z.B. für DHP:

Zertifikat: eine Folge $v_1, \ldots, v_n \in V$ der Länge $n = |V|$

Verifikation: Überprüfung, dass $V = \{v_1, \ldots, v_n\}$ und dass $(v_i, v_{i+1}) \in E$ für $i = 1, \ldots, n-1$.

Reduktionen

Wir zeigen nun:

$$\text{SAT} \leq_{pol} \text{DHP} \leq_{pol} \text{HP} \leq_{pol} \text{START-ZIEL HP} \leq_{pol} \text{HC}$$

Die Kette

$$\text{DHP} \leq_{pol} \text{START-ZIEL DHP} \leq_{pol} \text{DHC}$$

ist ähnlich zur Kette

$$\text{HP} \leq_{pol} \text{START-ZIEL HP} \leq_{pol} \text{HC}.$$

DHP ist NP-hart

Satz: SAT$\leq_{pol}$DHP.

Beweis: Sei $F = C_1 \wedge \ldots \wedge C_m$ eine SAT Formel mit m Klauseln über den n Variablen $x_1, \ldots, x_n$.

Wir konstruieren einen gerichteten Graphen G durch

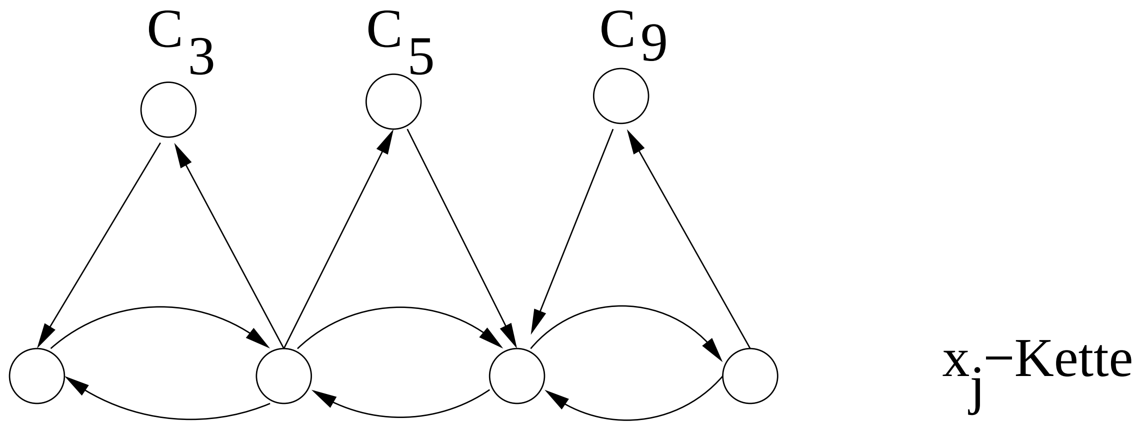
- für jede Variable x_i gibt es eine Komponente in G, die sogenannte x_i-Kette
- für jede Klausel C_j gibt es einen Knoten C_j in G
- zusätzlich enthält G einen ausgezeichneten Startknoten s und Zielknoten t

DHP ist NP-hart (fortgesetzt)

Tritt die Variable x_i (negiert oder unnegiert) in m_i Klauseln auf, so ist die x_i-Kette eine doppelt verkette Liste der Länge m_i.

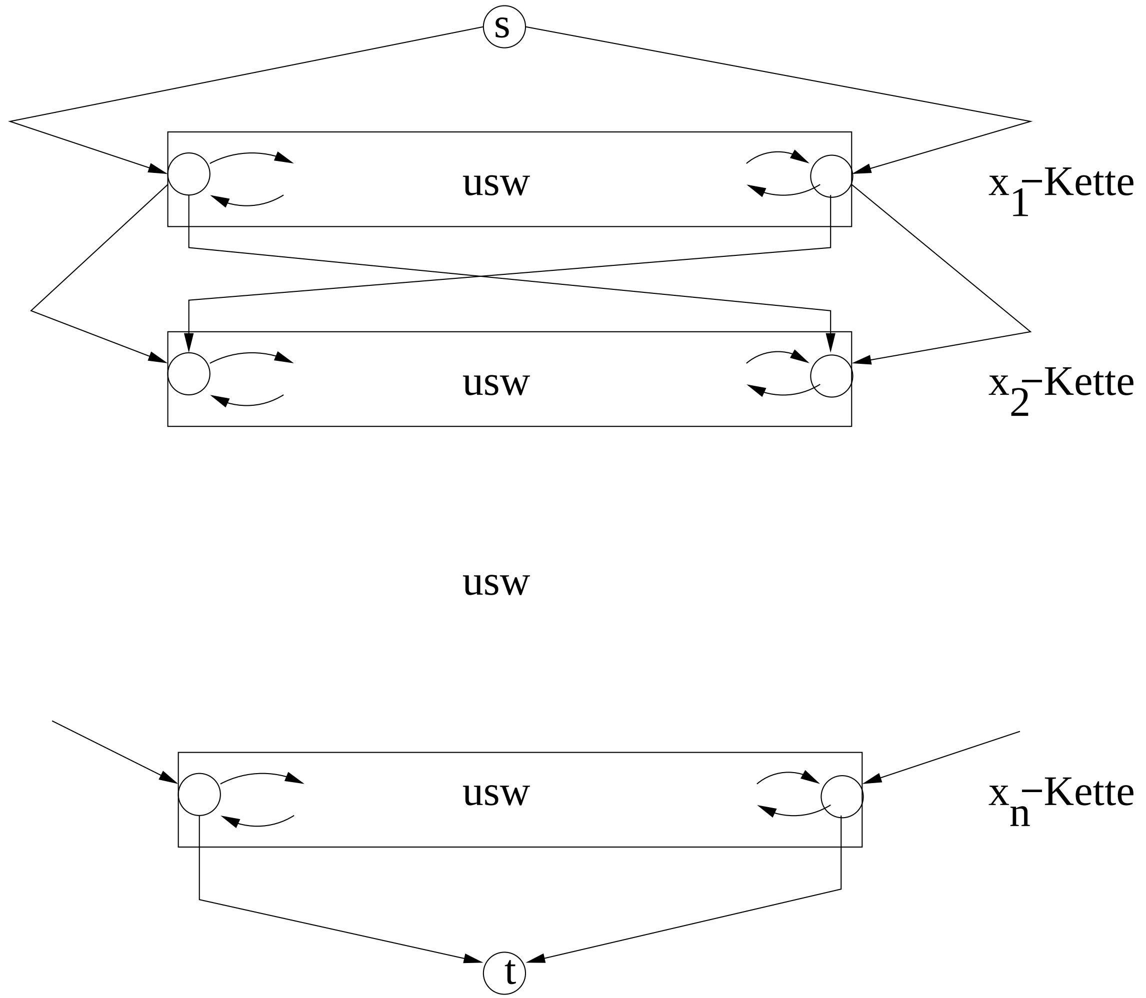
Für jedes negierte Vorkommen von x_i in Klausel C_j gibt es eine von rechts nach links führende Abzweigung aus der x_i-Kette nach C_j und wieder zurück.

Für jedes unnegierte Vorkommen von x_i in einer Klausel gibt es eine entsprechende von links nach rechts führende Abzweigung.



DHP ist NP-hart (fortgesetzt)

Die x_i-Ketten und Knoten s, t sind wie folgt verbunden:



DHP ist NP-hart (fortgesetzt)

Die einzige Chance, G mit einem Hamiltonschen Pfad zu durchlaufen, besteht offensichtlich darin

- in s zu starten,
- die x_i-Ketten in der Reihenfolge $i = 1, \ldots, n$ zu durchlaufen, wobei man jedesmal die freie Wahl zwischen der Durchlaufrichtung „links-rechts“ und „rechts-links“ hat,
- unterwegs jeden Klauselknoten C_j genau einmal zu besuchen (durch Wahl einer geeigneten Abzweigung aus einer Kette),
- und schließlich in t zu enden.

DHP ist NP-hart (fortgesetzt)

Wenn wir die x_i-Kette von rechts nach links (bzw. von links nach rechts) durchlaufen, können wir zu einer beliebigen Teilmenge der Klauseln abzweigen, in denen x_i negiert (bzw. unnegiert) vorkommt.

Wenn wir eine Belegung $x_i = 0$ (bzw. $x_i = 1$) mit der rechts-links Durchlaufrichtung (bzw. mit der links-rechts Durchlaufrichtung) der x_j-Kette identifizieren, dann wird offensichtlich:

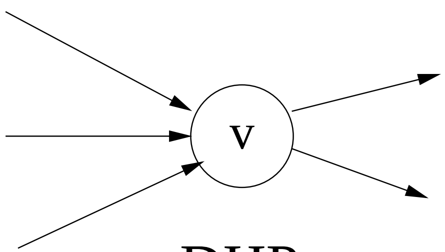
eine erfüllende Belegung von $F \Leftrightarrow$ ein Hamiltonscher Pfad in G

HP ist NP-hart

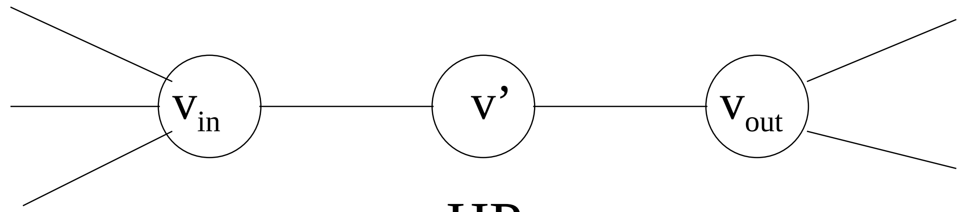
Satz: DHP$\leq_{pol}$HP.

Beweis: Wir verwandeln einen gerichteten Graphen $G = (V, E)$ in einen ungerichteten Graphen $G' = (V', E')$ durch

- jeder Knoten in V wird zu drei Knoten in $V' = \{v_{in}, v', v_{out} | \ v \in V\}$.
- E' besteht
  - aus den (ungerichteten) Verbindungskanten zwischen v_{in} und v' bzw. zwischen v' und v_{out} für alle $v \in V$
  - sowie den Kanten e' für alle $e \in E$. Wenn e eine (gerichtete) Kante von u nach v ist, so ist e' die (ungerichtete) Kante zwischen u_{out} und v_{in}.



DHP



HP

HP ist NP-hart (fortgesetzt)

Beobachtung: ein DHP existiert in $G \Leftrightarrow$ ein HP existiert in G'

$\Rightarrow$: Ein DHP in G lässt sich leicht umformen in einen HP in G'.

$\Leftarrow$: Wir stellen uns die Kanten in G' als gerichtete Kanten der Form (u_{out}, v_{in}), (v_{in}, v'), (v', v_{out}) vor. Wir beobachten, dass die Kanten auf dem HP entweder **alle** entlang dieser Orientierung oder **alle** entlang der umgekehrten Orientierung durchlaufen werden. Denn würden wir etwa bei v_{in} die Orientierung „umpolen“, also v_{in} über e'_1 betreten und über e'_2 gleich wieder verlassen, dann könnte HP den Knoten v' nicht durchlaufen, ohne v_{in} oder v_{out} mehrmals zu durchlaufen.

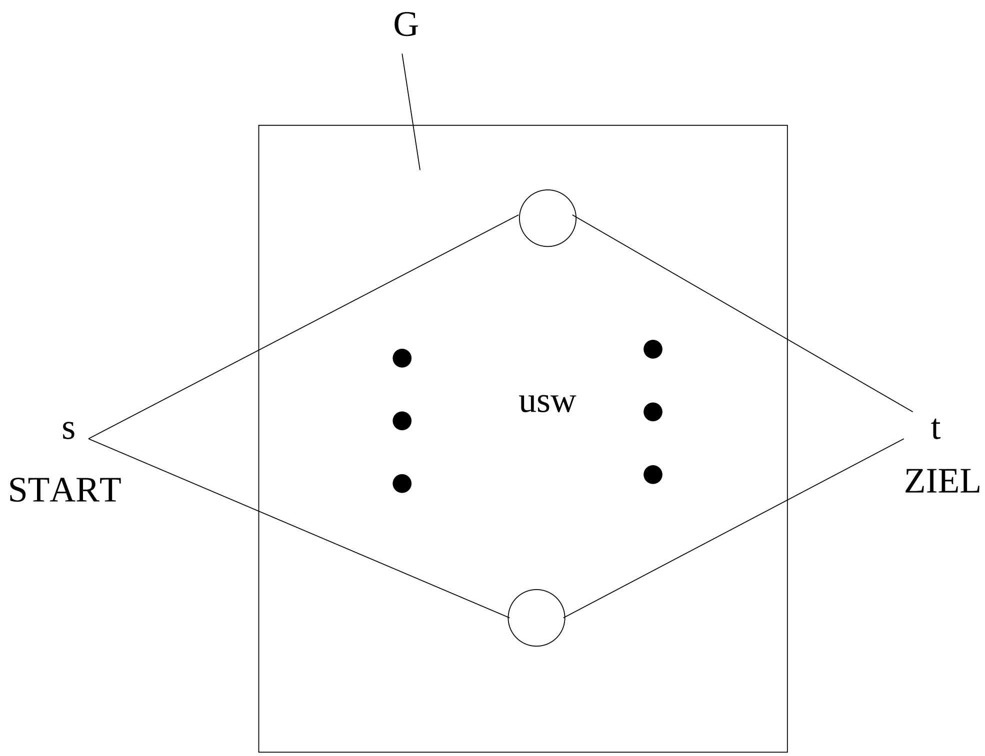
Wir können daher oBdA annehmen, dass alle Kanten von HP gemäß unserer vorgestellten Orientierung durchlaufen werden.

Die gleiche Durchlaufstrategie können wir dann aber auch in G anwenden. G besitzt folglich einen DHP.

START-ZIEL HP ist NP-hart

Satz: HP$\leq_{pol}$START-ZIEL HP.

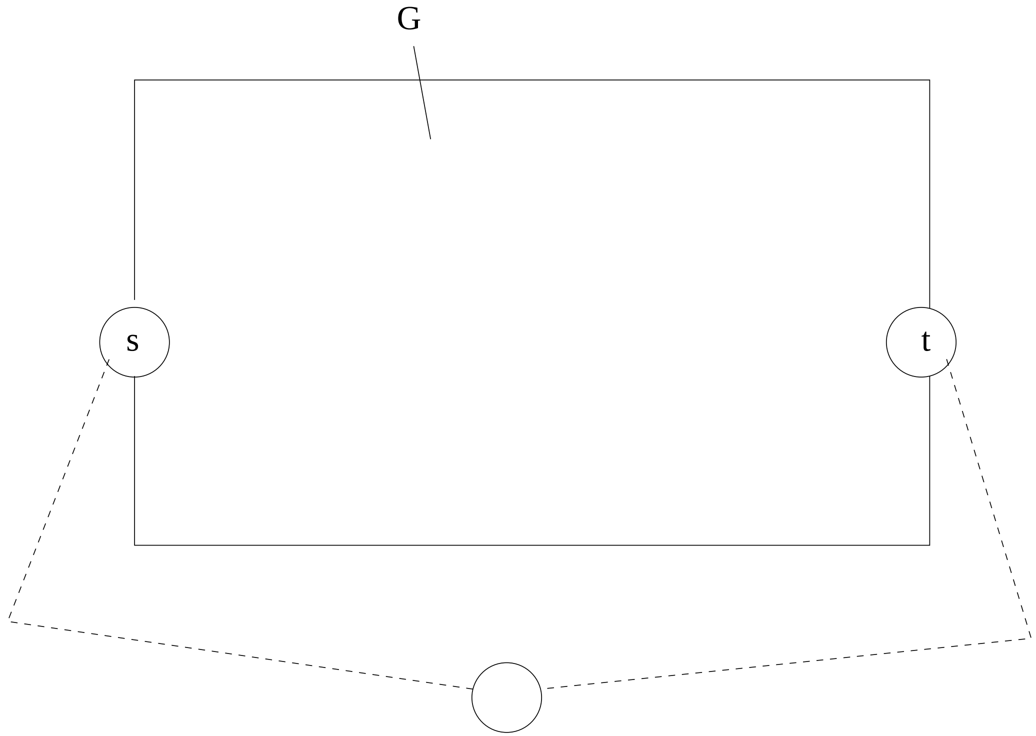
Beweis: Einem Graphen G fügen wir Knoten s, t hinzu, die mit allen Knoten von G verbunden sind. Dann überlegt man sich leicht, dass in diesem Graph ein START-ZIEL HP existiert, genau dann wenn in G ein HP existiert.



HC ist NP-hart

Satz: START-ZIEL HP$\leq_{pol}$HC.

Beweis: Dem Graphen G fügen wir einen Knoten und zwei Kanten hinzu, welche die Knoten s und t verbinden. Man überlegt sich leicht, dass in diesem Graph ein HC existiert, genau dann wenn in G ein START-ZIEL HP existiert.



Beweis des Satzes von Cook

SAT in NP

Denn:

Zertifikat: eine Belegung $a \in \{0,1\}^n$ der n Booleschen Variablen

Verifikation: Überprüfung, dass alle Klauseln in der Formel F durch a erfüllt werden

Als nächstes müssen wir zeigen, dass sich jede Sprache $L \in NP$ polynomiell auf SAT reduzieren lässt.

Idee des Beweises

Sei $L \in NP$ beliebig. Dann ist $L = T(M)$ für eine NTM M mit durch Polynom p beschränkter Laufzeit.

Sei $x \in \Sigma^*$ eine Eingabe für M. Wir geben eine boolesche Formel $F = F_x$ an, so dass gilt

$$x \in L \Leftrightarrow F \text{ ist erfüllbar.}$$

Idee: F beschreibt die Rechnung von M auf x.

Notation

Sei $x = x_1 \ldots x_n$ die Eingabe.

Sei $\Gamma = \{a_1, \ldots, a_\ell\}$ das Arbeitsalphabet von M.

Sei $Z = \{z_1, \ldots, z_k\}$ die Zustandsmenge von M.

OBdA können wir annehmen, dass M in einem einmal erreichten Endzustand stehen bleibt, also $(z_e, a, N) \in \delta(z_e, a)$ für einen Endzustand z_e.

Die Anzahl von Rechenschritten einer akzeptierenden Rechnung von M auf x ist durch $p(n)$ beschränkt. In dieser Zeit können auch nur maximal $p(n)$ Bandzellen rechts und links der Startposition erreicht werden. Wir nummerieren diese entsprechend durch, wobei der Kopf anfangs auf Position 1 steht.

Variablen von F

Variable	Indizes	Intendierte Bedeutung
$zust_{t,z}$	$t = 0, \ldots, p(n)$ $z \in Z$	$zust_{t,z} = 1 \;\Leftrightarrow$ nach t Schritten befindet sich M in Zustand x
$pos_{t,i}$	$t = 0, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$	$pos_{t,i} = 1 \;\Leftrightarrow$ nach t Schritten steht der der Schreib-Lese-Kopf von M an Position i
$band_{t,i,a}$	$t = 0, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$ $a \in \Gamma$	$band_{t,i,a} = 1 \;\Leftrightarrow$ nach t Schritten steht an Position i des Bandes von M das Zeichen a

Hilfsformel G

In F benutzen wir mehrfach die Hilfsformel G, mit folgender Eigenschaft:

$$G(x_1, \ldots, x_m) = 1 \Leftrightarrow \text{für genau ein } x_i \text{ ist } x_i = 1.$$

G lässt sich z.B. realisieren durch

$$(x_1 \vee \ldots \vee x_m) \wedge \bigwedge_{1 \leq i < j \leq m} (\overline{x_i} \vee \overline{x_j}).$$

Die Größe von G ist $1 + \binom{m}{2} = O(m^2)$.

Teilformeln von F

F hat die Form

$$F = R \wedge A \wedge U_1 \wedge U_2 \wedge E$$

wobei

R Randbedingungen

A Anfangsbedingungen

U_1, U_2 Übergangsbedingungen

E Endbedingungen

Randbedingungen

Die Randbedingungen besagen, dass zu jedem Zeitpunkt t die Variablen $zust_{t,z}, pos_{t,i}$ und $band_{t,i,a}$ eine gültige Konfiguration beschreiben.

$$R = \bigwedge_{t=0}^{p(n)} \Big(G(zust_{t,z_1}, \ldots, zust_{t,z_k}) \wedge G(pos_{t,-p(n)}, \ldots, pos_{t,p(n)})$$

$$\wedge \bigwedge_{i=-p(n)}^{p(n)} G(band_{t,i,a_1}, \ldots, band_{t,i,a_\ell}) \Big)$$

Anfangs- und Endbedingungen

Die Anfangsbedingungen besagen, dass zu Beginn (Zeitpunkt $t = 0$) die Variablen $zust_{0,z}, pos_{0,i}$ und $band_{0,i,a}$ die Startkonfiguration beschreiben.

$$A = zust_{0,z_0} \wedge pos_{0,1} \wedge \bigwedge_{j=1}^{n} band_{0,j,x_j} \wedge \bigwedge_{j=-p(n)}^{0} band_{0,j,\Box} \wedge \bigwedge_{j=n+1}^{p(n)} band_{0,j,\Box}$$

Die Endbedingungen besagen, dass zu Ende (Zeitpunkt $t = p(n)$) die Variablen $zust_{p(n),z}$ von einem Endzustand angenommen wird.

$$E = \bigvee_{z \in E} zust_{p(n),z}$$

Übergangsbedingungen

Die Übergangsbedingungen U_1 beschreiben die Variablen, die sich beim Übergang zum Zeitpunkt t zu einer Folgekonfiguration zum Zeitpunkt $t+1$ verändern. Die Übergangsbedingungen U_2 beschreiben die Variablen, die sich dabei nicht verändern.

$$U_1 = \bigwedge_{t,z,i,a} \Big((zust_{t,z} \wedge pos_{t,i} \wedge band_{t,i,a}) \rightarrow \bigvee_{(z',a',y) \in \delta(z,a)} (zust_{t+1,z'} \wedge pos_{t+1,i+y} \wedge band_{t+1,i,a'}) \Big)$$

$$U_2 = \bigwedge_{t,i,a} \Big((\overline{pos_{t,i}} \wedge band_{t,i,a}) \rightarrow band_{t+1,i,a} \Big)$$

Korrektheit

$\Rightarrow$: Nehmen wir an $x \in L$. Dann gibt es eine akzeptierende Rechnung von M auf x. Diese "erfüllt" alle Teilformeln von F entsprechend ihrer Intention, und damit auch F.

$\Leftarrow$: Nehmen wir an, F ist erfüllbar durch eine Belegung der Variablen. Dies entspricht nach Konstruktion von F einer akzeptierenden Rechnung, denn

- da R erfüllt ist, können für jedes t die Variablen sinnvoll als Konfiguration aufgefasst werden
- da A erfüllt ist, ist die Konfiguration für $t = 0$ die Startkonfiguration
- da U_1, U_2 erfüllt sind, ist die Konfiguration für $t+1$ eine Folgekonfiguration der Konfiguration für t
- da E erfüllt ist, endet die Rechnung in einer Endkonfiguration

Größe der Konstruktion

Bleibt zu zeigen, dass F in polynomieller Zeit berechnet werden kann. Dies folgt, da die Größe von F polynomiell in n ist. Unter Annahme, dass k, ℓ konstant sind, ergibt sich:

$$\begin{aligned}
|R| &= O(p(n)^2) \\
|A| &= O(p(n)) \\
|U_1| &= O(p(n)^2) \\
|U_2| &= O(p(n)^2) \\
|E| &= O(1)
\end{aligned}$$

Auch nach Umsetzung der Teilformeln G bleibt dies polynomiell.

3SAT ist NP-hart

Beweis: Sei F eine CNF Formel mit Klauseln von beliebig vielen Literalen. Wir ersetzen jede Klausel C in F durch eine Menge von Klauseln K_C aus jeweils genau drei (unterschiedlichen) Literalen.

Dabei gelte immer, dass eine erfüllende Belegung für C sich erweitern lässt zu einer erüllenden Belegung von K_C, und andersherum eine erfüllende Belegung von K_C sich einschränken lässt zu einer erüllenden Belegung von C.

Fallunterscheidung nach der Anzahl k von Literalen in C:

$k = 3$**:** in diesem Fall ist einfach $K_C = \{C\}$

$k < 3$**:** Sei $C = z_1$. Wir benutzen Hilfsvariablen a, b und wählen als K_C:

$$(z_1 \vee a \vee b), (z_1 \vee \overline{a} \vee b), (z_1 \vee a \vee \overline{b}), (z_1 \vee \overline{a} \vee \overline{b})$$

Sei $C = z_1 \vee z_2$. Dann benutzen wir Hilfsvariable a und wählen als K_C:

$$(z_1 \vee z_2 \vee a), (z_1 \vee z_2 \vee \overline{a})$$

3SAT ist NP-hart (fortgesetzt)

$k > 3$: Sei $C = z_1 \vee \ldots \vee z_k$. Wir splitten C auf unter Verwendung von Hilfsvariablen $h_2, \ldots, h_{k-2}$. Als K_C wählen wir:

$$(z_1 \vee z_2 \vee h_2), (\overline{h_2} \vee z_3 \vee h_3), \ldots, (\overline{h_{j-1}} \vee z_j \vee h_j), \ldots, (\overline{h_{k-2}} \vee z_{k-1} \vee z_k)$$

Wir überlegen uns, dass eine Belegung von C erfüllend ist, genau dann wenn eine entsprechende Belegung von K_C erfüllend ist.

$\Rightarrow$: Eine erfüllende Belegung von C erfüllt eines der Literale z_j. Wir erhalten eine erfüllende Belegung von K_C indem wir $h_2, \ldots, h_{j-1}$ auf wahr, und $h_j, \ldots, h_{k-2}$ auf falsch setzen.

$\Leftarrow$: Wir zeigen mit Hilfe der Logik des Zugzwanges: eine nicht erfüllende Belegung von C kann zu keiner erfüllenden Belegung von K_C führen.

Eine Belegung die C nicht erfüllt, erfüllt also keines der z_j. Um die erste Klausel $(z_1 \vee z_2 \vee h_2)$ aus K_C zu erfüllen, muss daher h_2 wahr sein. Um die zweite Klausel $(\overline{h_2} \vee z_3 \vee h_3)$ zu erfüllen, muss dann h_3 wahr sein. Iterativ folgt: alle h_j müssen wahr sein. Dann ist aber die letzte Klausel $(\overline{h_{k-2}} \vee z_{k-1} \vee z_k)$ nicht erfüllt.

Bemerkungen

Bemerkung: SAT kann deterministisch in $2^{O(n)}$ Zeit berechnet werden, z.B. durch systematisches Durchprobieren aller Lösungen.

Folgerung:

$$NP \subseteq \underbrace{\bigcup_{\text{Polynom } p} Time(2^{p(n)})}_{=: EXPTIME}$$

Bemerkung: 2SAT $\in P$.

Travelling Salesman (TSP, Handelsreisender)

Eingabe: Eine Kostenschranke K, n Städte $C_0, \ldots, C_{n-1}$ und eine Distanzmatrix $D = (d_{i,j})_{0 \le i,j \le n-1}$, wobei $d_{i,j} \in \mathbb{N}_0$ die Distanz zwischen C_i und C_j angibt.

Frage: Existiert eine Rundreise durch $C_0, \ldots, C_{n-1}$, deren Gesamtlänge K nicht überschreitet, d.h., existiert eine Permutation σ von $0, \ldots, n-1$ mit

$$\sum_{i=0}^{n-1} d_{\sigma(i)\sigma(i+1 \bmod n)} \le K \ ?$$

TSP ist in NP:

Zertifikat: eine Permutation σ von $0, \ldots, n-1$

Verifikation: Überprüfung, dass die durch σ gegebene Rundreise die Kostenschranke K nicht überschreitet

TSP ist NP-hart

Satz: HC$\leq_{pol}$TSP

Beweis: Sei $G = (V, E)$ mit $V = \{1, \ldots, n\}$ eine Eingabe von HC. Wir assoziieren zu G die Kostenschranke $K_G = n$ und die Distanzmatrix $D_G = D = (d_{ij})_{1 \leq i,j \leq n}$, wobei

$$d_{i,j} = \begin{cases} 0 & \text{falls } i = j, \\ 1 & \text{falls } i \neq j \text{ und } \{i,j\} \in E, \\ 2 & \text{falls } i \neq j \text{ und } \{i,j\} \notin E. \end{cases}$$

Offensichtlich kann man bezüglich D die Kostenschranke n genau dann einhalten, wenn die Rundreise nur durch Kanten aus E führt, also genau dann, wenn ein Hamiltonscher Kreis in G existiert.

Anmerkung: Der Beweis zeigt auch, dass metrisches TSP NP-vollständig ist, d.h. TSP mit der Einschränkung, dass Distanzmatrix symmetrisch ist und die Dreiecksungleichung erfüllt.

Entscheidungs- vs. Optimierungsproblem

In der Praxis interessiert man sich häufig für sogenannte Optimierungsprobleme, z.B. für TSP: finde eine *kürzeste* Route durch alle Städte.

Anmerkung: Optimierungsprobleme lassen sich in der Regel auf das Lösen von Entscheidungsproblemen reduzieren.

Details siehe Skript.

Umgang mit NP-harten Problemen

NP-harte Probleme treten häufig in der Praxis auf, z.B. TSP.

Wie kann man diese dennoch lösen?

- Heuristiken
- Approximation
- Spezialisierung
- Parametrisierte Komplexität
- ...

Mehr zu Approximationsalgorithmen im Skript.