Berechenbarkeit und Komplexität

NP-Vollständigkeit von DHC und einiger Zahlprobleme

Prof. Berthold Vöcking
präsentiert von Prof. Joost-Pieter Katoen

20. Januar 2009

Hamiltonkreisprobleme

Problem (Hamiltonkreis – Hamiltonian Circuit – HC)

Eingabe: Graph $G = (V, E)$

Frage: Gibt es einen Hamiltonkreis in G*?*

Problem (Gerichteter Hamiltonkreis – Directed HC – DHC)

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

Frage: Gibt es einen Hamiltonkreis in G*?*

Wiederholung der Ergebnisse

Lemma

$HC \leq_p DHC$.

Lemma

$DHC \leq_p HC$.

NP-Vollständigkeit von HC und DHC

Satz

HC und DHC sind NP*-vollständig.*

Beweis:

Beide Probleme sind offensichtlich in NP, da man die Kodierung eines Hamiltonkreises in polynomieller Zeit auf ihre Korrektheit überprüfen kann.

Da HC und DHC beidseitig aufeinander polynomiell reduzierbar sind, genügt es die NP-Härte eines der beiden Probleme nachzuweisen.

Wir zeigen: $\boxed{\text{SAT} \leq_p \text{DHC.}}$

NP-Vollständigkeit von HC und DHC – Fortsetzung Beweis

Reduktion:

Wir präsentieren eine polynomiell berechenbare Funktion f die eine KNF-Formel ϕ mit Variablen

$$x_1, \ldots, x_N$$

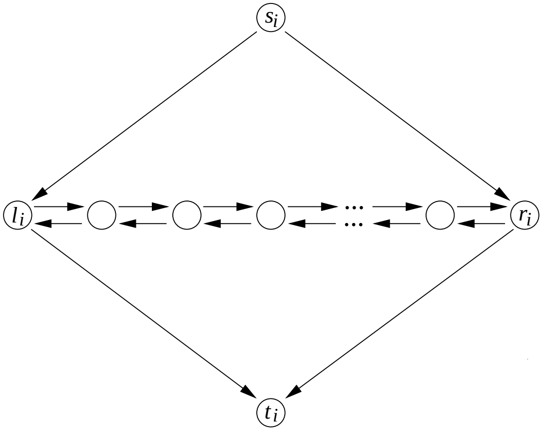
und Klauseln

$$c_1, \ldots, c_M$$

in einen gerichteten Graphen $G = (V, E)$ transformiert, sodaß:

$$\phi \text{ ist erfüllbar } \Leftrightarrow \; G \text{ hat einen Hamiltonkreis } .$$

Für jede Variable x_i enthalte der Graph G die folgende Struktur G_i.



Diese Struktur heißt *Diamantengadget*.

Diese N Gadgets werden miteinander verbunden, indem wir die Knoten t_i und s_{i+1} $(1 \leq i \leq N-1)$ sowie t_N und s_1 miteinander identifizieren. (Bild Tafel)

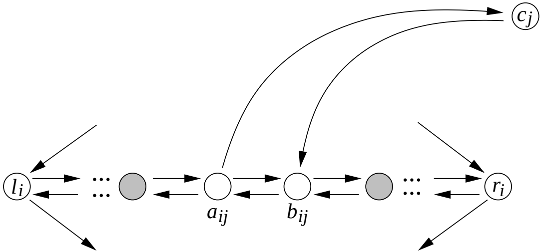
In dem so entstehenden Graphen besucht jede Rundreise, die beim Knoten s_1 startet, die Gadgets in der Reihenfolge $G_1, G_2, \ldots, G_N$.

Die Rundreise hat dabei für jedes Gadget G_i die Freiheit das Gadget *von links nach rechts*, also von l_i nach r_i, oder *von rechts nach links*, also von r_i nach l_i, zu durchlaufen.

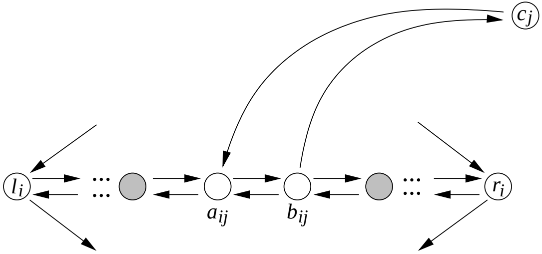
Die erste Variante interpretieren wir als Variablenbelegung $x_i = 1$, die zweite als Variablenbelegung $x_i = 0$.

Jetzt fügen wir einen weiteren Knoten für jede Klausel c_j ein.

Falls das Literal x_i in Klausel c_j enthalten ist, so verbinden wir das Gadget G_i wie folgt mit dem Klauselknoten c_j:



Falls das Literal $\bar{x}_i$ in Klausel c_j enthalten ist, so verbinden wir das Gadget G_i wie folgt mit dem Klauselknoten c_j:



Ist es nach Hinzunahme der Klauselknoten möglich, dass eine Rundreise zwischen den Gadgets hin- und herspringt statt sie in der vorgesehenen Reihenfolge zu besuchen? - Nein, weil ...

Korrektheit:

zu zeigen: G hat einen Hamiltonkreis $\Rightarrow \phi$ ist erfüllbar

- Wird ein Klauselknoten c_j aus einem Gadget G_i heraus von links nach rechts durchlaufen, so muss gemäß unserer Konstruktion, die Klausel c_j das Literal x_i enthalten.
- Also wird diese Klausel durch die mit der Laufrichtung von links nach rechts assoziierten Belegung $x_i = 1$ erfüllt.
- Bei einer Laufrichtung von rechts nach links, die mit der Belegung $x_i = 0$ assoziiert ist, wird die Klausel ebenso erfüllt, weil sie in diesem Fall das Literal $\bar{x}_i$ enthält.

Also erfüllt die mit der Rundreise assoziierte Belegung alle Klauseln.

zu zeigen: ϕ ist erfüllbar $\Rightarrow$ G hat einen Hamiltonkreis

- Eine Belegung beschreibt in welcher Richtung die Gadgets $G_1, \ldots, G_N$ jeweils durchlaufen werden.
- Klauselknoten c_j können wir in die Rundreise einbauen, indem wir eine der Variablen x_i auswählen, die c_j erfüllt, und c_j vom Gadget G_i aus besuchen.
- Sollte c_j für $x_i = 1$ erfüllt sein, so ist x_i unnegiert in c_j enthalten, und somit ist ein Besuch von c_j beim Durchlaufen des Gadgets G_i von links nach rechts möglich.
- Sollte c_j hingegen für $x_i = 0$ erfüllt sein, so ist x_i negiert in der Klausel enthalten, und der Besuch von c_j kann beim Durchlaufen des Gadgets G_i von rechts nach links erfolgen.

Also können alle Klauselknoten in die Rundreise eingebunden werden. $\square$

NP-Vollständigkeit von TSP

$\{1,2\}$-TSP ist eine eingeschränkte Variante des TSP-Problems, bei der wir nur die Gewichtswerte 1 und 2 erlauben.

Korollar

Die Entscheidungsvariante von $\{1,2\}$-TSP ist NP*-hart.*

Beweis: Zeige HC $\leq_p$ $\{1,2\}$-TSP. Wie? ...

Das SUBSET-SUM-Problem

Problem (SUBSET-SUM)

Eingabe: $a_1, \ldots, a_N \in \mathrm{N}$, $b \in \mathrm{N}$

Frage: Gibt es $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} a_i = b$*?*

Das SUBSET-SUM-Problem ist offensichtlich in NP enthalten, weil ...

NP-vollständigkeit des SUBSET-SUM-Problems

Satz

SUBSET-SUM ist NP*-vollständig.*

Beweis:

Um die NP-Härte des Problems nachzuweisen, beweisen wir:

$\boxed{\textit{3SAT} \leq_p \textit{SUBSET-SUM.}}$

Gegeben sei eine Formel ϕ in 3KNF. Diese Formel bestehe aus M Klauseln $c_1, \ldots, c_M$ über N Variablen $x_1, \ldots, x_N$.

Für $i \in \{1, \ldots, N\}$ sei

$$\begin{aligned} S(i) &= \{j \in \{1, \ldots, M\} \mid \text{Klausel } c_j \text{ enthält Literal } x_i\} \ , \\ S'(i) &= \{j \in \{1, \ldots, M\} \mid \text{Klausel } c_j \text{ enthält Literal } \bar{x}_i\} \ . \end{aligned}$$

Reduktion 3SAT $\leq_p$ SUBSET-SUM

Aus der 3KNF-Formel ϕ erzeugen wir eine SUBSET-SUM-Eingabe:

- Wir beschreiben die Eingabe von SUBSET-SUM in Form von Dezimalzahlen, die aus $N + M$ Ziffern bestehen.
- Die k-te Ziffer einer Zahl a bezeichnen wir dabei mit $a(k)$.
- Für jede boolesche Variable x_i, $i \in \{1, \ldots, N\}$, enthält die SUBSET-SUM-Eingabe zwei Zahlen a_i und a'_i, wobei

$$a_i(i) = 1 \quad \text{und} \quad \forall j \in S(i) : a_i(N + j) = 1 \ ,$$
$$a'_i(i) = 1 \quad \text{und} \quad \forall j \in S'(i) : a'_i(N + j) = 1 \ .$$

- Alle anderen Ziffern setzen wir auf den Wert 0.

Beispiel: Wie lauten die Zahlen für die Formel $(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3 \vee \bar{x}_4)$?

Reduktion 3SAT $\leq_p$ SUBSET-SUM

Aus der Formel ϕ in 3KNF erzeugen wir eine SUBSET-SUM-Eingabe:

- Zusätzlich erzeugen wir zwei sogenannte *Füllzahlen* h_j und h'_j für jede Klausel j, die nur an Ziffernposition $N + j$ eine 1 haben, alle anderen Ziffern sind 0.
- Außerdem definieren wir den Summenwert b folgendermaßen:

$$\begin{aligned} b(k) &= 1 \quad \text{für} 1 \leq k \leq N \\ b(k) &= 3 \quad \text{für} N + 1 \leq k \leq N + M. \end{aligned}$$

Reduktion 3SAT $\leq_p$ SUBSET-SUM: *Illustration*

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	1	0	0	$\cdots$	0	1	0	$\cdots$	$\cdots$
a'_1	1	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	1	0	$\cdots$	0	0	1	$\cdots$	$\cdots$
a'_2	0	1	0	$\cdots$	0	1	0	$\cdots$	$\cdots$
a_3	0	0	1	$\cdots$	0	1	1	$\cdots$	$\cdots$
a'_3	0	0	1	$\cdots$	0	0	0	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_N	0	0	0	$\cdots$	1	0	0	$\cdots$	$\cdots$
a'_N	0	0	0	$\cdots$	1	0	1	$\cdots$	$\cdots$
h_1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
h'_1	0	0	0	$\cdots$	0	1	0	$\cdots$	0
h_2	0	0	0	$\cdots$	0	0	1	$\cdots$	0
h'_2	0	0	0	$\cdots$	0	0	1	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
h_M	0	0	0	$\cdots$	0	0	0	$\cdots$	1
h'_M	0	0	0	$\cdots$	0	0	0	$\cdots$	1
b	1	1	1	$\cdots$	1	3	3	$\cdots$	3

Reduktion 3SAT $\leq_p$ SUBSET-SUM: *Beobachtungen*

Beobachtung 1:

Die Eingabezahlen zu SUBSET-SUM können in polynomieller Zeit erzeugt werden (obwohl die Zahlenwerte exponentiell groß sind).

Beobachtung 2:

Bei der Addition einer beliebigen Teilmenge der Zahlen a_i, a'_i, h_i, h'_i gibt es keinen Additionsübertrag von Ziffer zu Ziffer, weil ...

Reduktion 3SAT $\leq_p$ SUBSET-SUM: *Korrektheit*

zu zeigen: ϕ erfüllbar $\Rightarrow \exists$ Teilsumme mit Wert b

Angenommen es gibt eine erfüllende Belegung x^* für ϕ.

- Dann nehmen wir diejenigen Zahlen a_i in unsere Teilmenge K auf, für die gilt $x_i^* = 1$, ansonsten nehmen wir a_i' auf.
- Sei A die Summe der ausgewählten Zahlen a_i und a_i'.
- Da für jedes $i \in \{1, \ldots, N\}$ entweder a_i oder a_i' aufgenommen wird, gilt $A(i) = 1$.
- Zudem gilt $A(N + j) \in \{1, 2, 3\}$ für $1 \leq j \leq M$, weil ...
- Falls $A(N + j) < 3$ so können wir eine oder beide der Füllzahlen h_j und h_j' verwenden um exakt den geforderten Wert 3 an Ziffernposition $N + j$ der Summe zu erhalten.

Also gibt es eine Teilsumme mit Wert b.

Reduktion 3SAT $\leq_p$ SUBSET-SUM: *Korrektheit*

zu zeigen: $\exists$ Teilsumme mit Wert $b \Rightarrow \phi$ erfüllbar

Angenommen es gibt eine Teilsumme mit Wert b.

Dann enthält K für jedes $i \in \{1, \dots, N\}$ entweder die Zahl a_i oder die Zahl a'_i, denn sonst ...

Setze $x^*_i = 1$, falls $a_i \in K$, und $x^*_i = 0$, falls $a'_i \in K$.

zu zeigen: x^* ist eine erfüllende Belegung für ϕ

- Sei A die Summe der Zahlen a_i und a'_i aus K.
- Es gilt $A(N + j) \geq 1$ für $1 \leq j \leq M$, weil ...
- Dadurch ist sichergestellt, dass x^* für jede Klausel mindestens ein Literal mit Wert 1 enthält, so dass ϕ erfüllt ist.

Damit ist die Korrektheit der Reduktion nachgewiesen. $\square$

NP-Vollständigkeit von PARTITION

Problem (PARTITION)

Eingabe: $a_1, \ldots, a_N \in \mathbb{N}$

Frage: Gibt es $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} a_i = \sum_{i \in \{1,\ldots,N\} \setminus K} a_i$*?*

PARTITION ist ein Spezialfall von SUBSET-SUM, da die gestellte Frage äquivalent zur Frage ist, ob es eine Teilmenge K mit Summenwert $\frac{1}{2} \sum_{i=1}^{N} a_i$ gibt.

NP-Vollständigkeit von PARTITION

Satz

PARTITION ist NP*-vollständig.*

Beweis:

PARTITION ist offensichtlich $\in$ NP, weil es ein Spezialfall von SUBSET-SUM ist.

Um zu zeigen, dass PARTITION NP-hart ist, zeigen wir SUBSET-SUM $\leq_p$ PARTITION.

Reduktion von SUBSET-SUM auf PARTITION

Die Eingabe von SUBSET-SUM sei $a_1, \ldots, a_N \in \mathbb{N}$ und $b \in \mathbb{N}$.

Es sei $A = \sum_{i=1}^{N} a_i$.

Wir bilden diese Eingabe für SUBSET-SUM auf eine Eingabe für PARTITION ab, die aus den $N + 2$ Zahlen $a'_1, \ldots, a'_{N+2}$ bestehe.

Dazu setzen wir

- $a'_i = a_i$ für $1 \leq i \leq N$,
- $a'_{N+1} = 2A - b$, und
- $a'_{N+2} = A + b$.

In der Summe ergeben diese $N + 2$ Zahlen den Wert $4A$.

PARTITION fragt also danach, ob es eine Teilmenge der Zahlen $a'_1, \ldots, a'_{N+2}$ mit Summenwert $2A$ gibt.

Reduktion von SUBSET-SUM auf PARTITION

Die Reduktion ist in polynomieller Zeit berechenbar.

zeige: $\exists$ Lösung für PARTITION $\Rightarrow$ $\exists$ Lösung für SUBSET-SUM

- Wenn es eine geeignete Aufteilung der Eingabezahlen für PARTITION gibt, so können a'_{N+1} und a'_{N+2} dabei nicht in derselben Teilmenge sein, denn $a'_{N+1} + a'_{N+2} = 3A$.
- Deshalb ergibt sich auch eine Lösung für SUBSET-SUM, denn diejenigen Zahlen aus $a'_1, \dots, a'_N$, die sich in derselben Teilmenge wie a'_{N+1} befinden, summieren sich auf zu $2A - a'_{N+1} = b$.

zeige: $\exists$ Lösung für SUBSET-SUM $\Rightarrow$ $\exists$ Lösung für PARTITION

- Wenn es eine Teilmenge der Zahlen $a_1, \ldots, a_N$ mit Summenwert b gibt, so gibt es auch eine Teilmenge der Zahlen $a'_1, \ldots, a'_N$ mit diesem Summenwert.
- Wir können die Zahl $a'_{N+1} = 2A - b$ zu dieser Teilmenge hinzufügen, und erhalten dadurch eine Teilmenge mit Summenwert $2A$.

$\square$

Konsequenzen für KP und BPP

Problem (Entscheidungsvariante des Rucksackproblems – KP-E)

Eingabe: $B, P \in \mathbb{N}$, $w_1, \ldots, w_N \in \{1, \ldots, B\}$, $p_1, \ldots, p_N \in \mathbb{N}$

Frage: *Gibt es* $K \subseteq \{1, \ldots, N\}$ *mit* $\sum_{i \in K} w_i \leq B$ *und* $\sum_{i \in K} p_i \geq P$

Korollar

KP-E ist NP*-vollständig.*

Beweis durch einfache Reduktion von SUBSET-SUM (Wie?)

Konsequenzen für KP und BPP

Problem (Entscheidungsvariante von Bin Packing – BPP-E)

Eingabe: $B, k \in \mathbb{N}$, $w_1, \ldots, w_N \in \{1, \ldots, B\}$

zulässige Lösungen: *Gibt es eine Fkt* $f : \{1, \ldots, N\} \to \{1, \ldots, k\}$,

$$\textit{so dass } \forall i \in \{1, \ldots, k\} : \sum_{j \in f^{-1}(i)} w_j \leq B$$

Korollar

BPP-E ist NP*-vollständig.*

Beweis durch einfache Reduktion von PARTITION (Wie?)