NP-Vollständigkeit ausgewählter Zahlprobleme

Prof. Dr. Berthold Vöcking
Lehrstuhl Informatik 1
Algorithmen und Komplexität
RWTH Aachen

Dezember 2011

Problem (SUBSET-SUM)

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

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

Das SUBSET-SUM-Problem ist in NP enthalten, denn die Lösung K kann als Zertifikat verwendet werden, das in polynomieller Zeit verifiziert werden kann.

Satz

SUBSET-SUM ist NP*-vollständig.*

Beweis:

Um die NP-Härte des Problems nachzuweisen, beschreiben wir eine Polynomialzeitreduktion von 3SAT.

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

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

Aus der Formel ϕ in 3KNF erzeugen wir verschiedene Zahlen mit jeweils $N + M$ Dezimalziffern.

Die k-te Ziffer einer Zahl a bezeichnen wir dabei mit $a(k)$.

Für jede boolesche Variable x_i, $i \in \{1, \ldots, N\}$ erzeugen wir zwei Zahlen a_i und a'_i, deren Ziffern wie folgt definiert sind

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

Alle anderen Ziffern setzen wir auf den Wert 0.

Diese Zahlen bezeichnen wir als *a-Zahlen*.

Beispiel:

Gegeben sei die Formel

$$(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \bar{x_3} \vee \bar{x_4}) \quad .$$

Aus dieser Formel werden folgende a-Zahlen erzeugt:

$$\begin{aligned}
a_1 &= 100010 \\
a_1' &= 100000 \\
a_2 &= 010011 \\
a_2' &= 010000 \\
a_3 &= 001010 \\
a_3' &= 001001 \\
a_4 &= 000100 \\
a_4' &= 000101
\end{aligned}$$

Reduktion 3SAT $\leq_p$ SUBSET-SUM

Zusätzlich erzeugen wir zwei sogenannte *h-Zahlen* h_j und h'_j für jede Klausel j, die nur an Ziffernposition $N + j$ eine 1 haben, alle anderen Ziffern sind 0.

Den *Summenwert* b definieren wir folgendermaßen: Wir setzen $b(k) = 1$ für $1 \leq k \leq N$ und $b(k) = 3$ für $N + 1 \leq k \leq N + M$.

Fortsetzung des Beispiels:

Die h-Zahlen und der Summenwert lauten

$$\begin{aligned} h_1 &= 000010 \\ h'_1 &= 000010 \\ h_2 &= 000001 \\ h'_2 &= 000001 \\ b &= 111133 \end{aligned}$$

Reduktion 3SAT $\leq_p$ SUBSET-SUM

Für eine Formel aus N Variablen und M Klauseln ergeben sich beispielsweise die folgenden Zahlen:

	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	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
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

Beobachtung 1:

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

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 Variablen- und der Füllzahlen gibt es keinen Additionsübertrag von Ziffer zu Ziffer, weil höchstens fünf Ziffern pro Spalte den Wert 1 haben.

Reduktion 3SAT $\leq_p$ SUBSET-SUM

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 Variablen- und der Füllzahlen gibt es keinen Additionsübertrag von Ziffer zu Ziffer, weil höchstens fünf Ziffern pro Spalte den Wert 1 haben.

Anmerkung: Beobachtung 2 beruht darauf, dass wir mit Dezimalziffern rechnen, d.h. zur Basis 10 rechnen. De facto wäre es auch ausreichend, wenn wir zur Basis 6 rechnen würden.

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

zu zeigen: ϕ erfüllbar $\Rightarrow \exists$ Teilmenge der a- und h-Zahlen, deren Summe gleich b ist

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

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

zu zeigen: ϕ erfüllbar $\Rightarrow$ $\exists$ Teilmenge der a- und h-Zahlen, deren Summe gleich b ist

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

- Falls $x_i^* = 1$, so *wähle* a_i aus, ansonsten wähle a_i'.
- Sei A die Summe der ausgewählten a-Zahlen.
- Da für jedes $i \in \{1, \ldots, N\}$ entweder a_i oder a_i' ausgewählt wurde, gilt $A(i) = 1$.

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

zu zeigen: ϕ erfüllbar $\Rightarrow \exists$ Teilmenge der a- und h-Zahlen, deren Summe gleich b ist

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

- Falls $x_i^* = 1$, so *wähle* a_i aus, ansonsten wähle a_i'.
- Sei A die Summe der ausgewählten a-Zahlen.
- Da für jedes $i \in \{1, \ldots, N\}$ entweder a_i oder a_i' ausgewählt wurde, gilt $A(i) = 1$.
- Zudem gilt $A(N + j) \in \{1, 2, 3\}$ für $1 \leq j \leq M$, weil in jeder Klausel mindestens ein und höchstens drei Literale erfüllt werden.

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

zu zeigen: ϕ erfüllbar $\Rightarrow \exists$ Teilmenge der a- und h-Zahlen, deren Summe gleich b ist

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

- Falls $x_i^* = 1$, so *wähle* a_i aus, ansonsten wähle a_i'.
- Sei A die Summe der ausgewählten a-Zahlen.
- Da für jedes $i \in \{1, \ldots, N\}$ entweder a_i oder a_i' ausgewählt wurde, gilt $A(i) = 1$.
- Zudem gilt $A(N+j) \in \{1, 2, 3\}$ für $1 \leq j \leq M$, weil in jeder Klausel mindestens ein und höchstens drei Literale erfüllt werden.
- Falls $A(N+j) < 3$ so können wir zusätzlich h_j oder h_j und h_j' auswählen um exakt den geforderten Wert 3 an Ziffernposition $N+j$ der Summe zu erhalten.

Also gibt es eine Teilmenge mit Summenwert b.

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

Sei A die Summe einer Teilmenge der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In A wird für jedes $i \in \{1, \ldots, N\}$ genau eine der Variablenzahlen a_i oder a'_i aufsummiert, denn ansonsten wäre $A(i) \neq 1$.

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

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

Sei A die Summe einer Teilmenge der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In A wird für jedes $i \in \{1, \ldots, N\}$ genau eine der Variablenzahlen a_i oder a'_i aufsummiert, denn ansonsten wäre $A(i) \neq 1$.

Setze $x_i = 1$, falls a_i in A aufsummiert wird; und $x_i = 0$, sonst.

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

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

Sei A die Summe einer Teilmenge der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In A wird für jedes $i \in \{1, \ldots, N\}$ genau eine der Variablenzahlen a_i oder a'_i aufsummiert, denn ansonsten wäre $A(i) \neq 1$.

Setze $x_i = 1$, falls a_i in A aufsummiert wird; und $x_i = 0$, sonst.

zu zeigen: x ist eine erfüllende Belegung für ϕ

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

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

Sei A die Summe einer Teilmenge der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In A wird für jedes $i \in \{1, \ldots, N\}$ genau eine der Variablenzahlen a_i oder a'_i aufsummiert, denn ansonsten wäre $A(i) \neq 1$.

Setze $x_i = 1$, falls a_i in A aufsummiert wird; und $x_i = 0$, sonst.

zu zeigen: x ist eine erfüllende Belegung für ϕ

- Es gilt $A(N + j) \geq 1$ für $1 \leq j \leq M$, denn ansonsten wäre $A(N + j) + H(N + j) < 3$.

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

Sei A die Summe einer Teilmenge der a-Zahlen und H die Summe einer Teilmenge der h-Zahlen, so dass gilt $A + H = b$.

In A wird für jedes $i \in \{1, \ldots, N\}$ genau eine der Variablenzahlen a_i oder a'_i aufsummiert, denn ansonsten wäre $A(i) \neq 1$.

Setze $x_i = 1$, falls a_i in A aufsummiert wird; und $x_i = 0$, sonst.

zu zeigen: x ist eine erfüllende Belegung für ϕ

- Es gilt $A(N + j) \geq 1$ für $1 \leq j \leq M$, denn ansonsten wäre $A(N + j) + H(N + j) < 3$.
- Dadurch ist sichergestellt, dass in jeder Klausel mindestens eines der Literale den Wert 1 hat, 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.

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 $a_1, \dots, a_N \in \mathbb{N}$ 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 $a'_1, \dots, a'_{N+2}$ bestehe.

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
- $a'_{N+2} = A + b$.

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.

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 $a'_{N+1} + a'_{N+2} = 3A$.

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 $a'_1, \ldots, a'_N$, 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 $a_1, \dots, a_N$ mit Summenwert b gibt, so gibt es auch eine Teilmenge der Zahlen $a'_1, \dots, a'_N$ mit diesem Summenwert.

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 $2A$.

$\square$

Härte des Bin Packing Problem

Problem (Bin Packing Problem – BPP)

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

zulässige Lösungen: $k \in \mathbb{N}$ *und Fkt* $f : \{1, \ldots, N\} \to \{1, \ldots, k\}$,

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

Zielfunktion: *Minimiere* k *(= Anzahl Behälter)*

Härte des Bin Packing Problem

Problem (Bin Packing Problem – BPP)

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

zulässige Lösungen: $k \in \mathbb{N}$ *und Fkt* $f : \{1, \ldots, N\} \to \{1, \ldots, k\}$,

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

Zielfunktion: *Minimiere* k *(= Anzahl Behälter)*

Entscheidungsvariante (BPP-E): $k \in \mathbb{N}$ ist gegeben. Passen die Objekte in k Behälter?

Satz

BPP-E ist NP*-vollständig.*

Beweis:

BPP-E $\in$ NP haben wir bereits gezeigt.

Die NP-Härte ergibt sich durch eine triviale Reduktion von PARTITION:

$$\text{Setze } k = 2,\ w_i = a_i \text{ für } 1 \leq i \leq N \text{ und } b = \left\lfloor \tfrac{1}{2} \sum_{i=1}^{N} w_i \right\rfloor .$$

$\square$

Wir sagen ein Optimierungsproblem Π ist NP-*hart*, wenn ein effizienter Algorithmus für Π einen effizienten Algorithmus für ein NP-hartes Entscheidungsproblem liefert.

Härte des Bin Packing Problem

Wir sagen ein Optimierungsproblem Π ist NP-*hart*, wenn ein effizienter Algorithmus für Π einen effizienten Algorithmus für ein NP-hartes Entscheidungsproblem liefert.

Aus einem effizienten Algorithmus für BPP ergibt sich ein effizienter Algorithmus für BPP-E. Wir haben gezeigt, dass BPP-E NP-hart ist. Es folgt

Korollar

BPP ist NP-*hart.*

Härte des Rucksackproblems

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?)