

Subset Sum

Definition Sprache SubsetSum

Sei $M = \{m_1, \dots, m_k\} \subset \mathbb{N}$ und $t \in \mathbb{N}$. Wir definieren die Sprache

$$\text{SUBSETSUM} := \{(M, t) \mid \exists S \subset M : \sum_{s \in S} s = t\}.$$

Satz

SUBSETSUM ist \mathcal{NP} -vollständig.

- 1 SUBSETSUM $\in \mathcal{NP}$ (Übung)
- 2 $3\text{SAT} \leq_p \text{SUBSETSUM}$

Idee der Reduktion $f(\phi(x_1, \dots, x_n)) = (S, t)$: Konstruieren

- Für jedes x_i Elemente $y_i, z_i \in S$ für $x_i = 1$ bzw. $x_i = 0$.
- Für jede Klausel K_j Variablen $g_j, h_j \in S$ für nicht erfüllte Literale.
- Definieren Tabelle T mit Zeilen y_i, z_i, g_j, h_j und Zeile t . Die Spalten bestehen aus x_i und K_j für $i \in [n], j \in [m]$.
- Einträge in einer Zeile werden als Dezimaldarstellung interpretiert.

Konstruktion der Reduktion f

Algorithmus M_f

EINGABE: $\phi(x_1, \dots, x_n) = K_1 \wedge \dots \wedge K_m$ mit $K_j = \ell_{j1} \vee \ell_{j2} \vee \ell_{j3}$

- 1 Erstelle Tabelle T mit
 - ▶ $2(n + m) + 1$ Zeilen y_i, z_i, g_j, h_j, t für alle i, j
 - ▶ $n + m$ Spalten x_i, K_j für alle i, j
- 2 Definiere mittels des Dezimalwerts der betreffenden Zeile:
 - ▶ y_i : Einsen in Spalte x_i und allen Spalten K_j , falls x_i Literal in K_j .
 - ▶ z_i : Einsen in Spalte x_i und allen Spalten K_j , falls $\neg x_i$ Literal in K_j .
 - ▶ g_j, h_j : Einsen jeweils in Spalte K_j .
 - ▶ t : Einsen in Spalten x_i , Dreien in Spalten K_j .
- 3 Fülle alle nicht belegten Tabelleneinträge mit Nullen.

AUSGABE: (M, t) mit $M = \{y_1, z_1, \dots, y_n, z_n, g_1, h_1, \dots, g_m, h_m\}$.

Laufzeit:

- Eingabelänge $|\phi| \geq \max\{m, n\} = \Omega(m + n)$
- $T(M_f) = \mathcal{O}((n + m)^2)$, d.h. polynomiell in der Eingabelänge.

Bsp für $\phi = (x_1 \vee x_2 \vee x_2) \wedge (\neg x_1 \vee x_2 \vee \neg x_2)$

- Definieren Tabelle T

	x_1	x_2	K_1	K_2
y_1	1	0	1	0
z_1	1	0	0	1
y_2	0	1	1	1
z_2	0	1	0	1
g_1	0	0	1	0
h_1	0	0	1	0
g_2	0	0	0	1
h_2	0	0	0	1
t	1	1	3	3

- Setze $y_1 = 1010, z_1 = 1001, \dots, t = 1133$.
- Belegung $x_1, x_2 = 1$ erfüllt Literale x_1, x_2 in K_1 und Literal x_2 in K_2 .
- Zahlen y_1, y_2 summieren sich mit g_1 für K_1 und g_2, h_2 für K_2 zu t .

Korrektheit: $\phi \in 3SAT \Rightarrow f(\phi) \in SUBSETSUM$

Sei $\phi \in 3SAT$

- Dann besitzt ϕ eine erfüllende Belegung B .
- Nimm y_i in S auf, falls $x_i = 1$ in B . Sonst nimm z_i in S auf.
- Betrachten $t' = \sum_{s \in S} s$:
 - ▶ B ist konsistente Belegung: Obere n Dezimalstellen von t' sind 1.
 - ▶ B ist erfüllend: Untere m Dezimalstellen t_1, \dots, t_m sind aus $\{1, 2, 3\}$.
- Falls $t_i = 1$, nimm g_i und h_i in S auf. Falls $t_i = 2$, nimm g_i in S auf.
- Damit gilt $\sum_{s \in S} s = t$.
- D.h. $f(\phi) = (M, t) \in SUBSETSUM$

Korrektheit $f(\phi) \in \text{SUBSETSUM} \Rightarrow \phi \in 3\text{SAT}$

Sei $f(\phi) \in \text{SUBSETSUM}$

- Dann gibt es $S \subseteq M$ mit $\sum_{s \in S} s = t$, wobei $t = 1 \dots 13 \dots 3$.
- Die oberen n Dezimalstellen von t sind 1.
 - ▶ Damit enthält S für jedes i genau eines der Elemente y_i, z_i .
 - ▶ Sei B die Belegung mit $x_i = 1$ für $y_i \in S$ und $x_i = 0$ sonst.
- Die unteren m Dezimalstellen t_1, \dots, t_m von t sind 3.
 - ▶ D.h. t_j kann nicht allein als Summe von g_j und h_j dargestellt werden.
 - ▶ Für jedes t_j kommt mindestens eine 1 aus einer Zeile y_i bzw. z_i .
 - ▶ D.h. das Literal x_i bzw. $\neg x_i$ erfüllt die Klausel K_j .
- Damit ist B eine erfüllende Belegung für ϕ .
- D.h. $\phi \in 3\text{SAT}$.

Das Rucksackproblem

Definition Sprache Rucksack

Gegeben sind n Gegenstände mit Gewichten $W = \{w_1, \dots, w_n\} \subset \mathbb{N}$ und Profiten $P = \{p_1, \dots, p_n\} \subset \mathbb{N}$. Seien ferner $B, k \in \mathbb{N}$.

$$\text{RUCKSACK} := \{(W, P, B, k) \mid \exists I \subseteq [n] : \sum_{i \in I} w_i \leq B \text{ und } \sum_{i \in I} p_i \geq k.\}$$

Satz

RUCKSACK ist \mathcal{NP} -vollständig.

Beweis: zu zeigen

- 1 RUCKSACK $\in \mathcal{NP}$ (bereits gezeigt)
- 2 SUBSETSUM \leq_p RUCKSACK

Reduktion $f(M, t) = (W, P, B, k)$

Algorithmus M_f

EINGABE: M, t

- 1 Setze $B \leftarrow t$ und $k \leftarrow t$.
- 2 For $i \leftarrow 1$ to n : Setze $w_i \leftarrow m_i$ und $p_i \leftarrow m_i$

AUSGABE: W, P, B, k

Laufzeit:

- Eingabelänge: $\log(t) + \sum_{i=1}^n \log(m_i)$
- Schritt 1: $\mathcal{O}(\log t)$, Schritt 2: $\mathcal{O}(\sum_{i=1}^n \log(m_i))$
- D.h. Gesamtlaufzeit ist polynomiell in der Eingabelänge.

$(M, t) \in \text{SUBSETSUM} \Leftrightarrow f(M, t) \in \text{RUCKSACK}$

Sei $(M, t) \in \text{SUBSETSUM}$

- Dann gibt es eine Menge $I \subseteq [n]$ mit $\sum_{i \in I} m_i = t$.
- Damit gilt $\sum_{i \in I} m_i \leq t$ und $\sum_{i \in I} m_i \geq t$.
- Es folgt $\sum_{i \in I} w_i \leq B$ und $\sum_{i \in I} p_i \geq t$.
- Damit gilt $f(M, t) = (W, P, B, k) \in \text{RUCKSACK}$

Sei $(W, P, B, k) = f(M, t) \in \text{RUCKSACK}$

- Dann gibt es eine Menge $I \subseteq [n]$ mit $\sum_{i \in I} w_i \leq B$ und $\sum_{i \in I} p_i \geq k$.
- D.h. es gibt eine Menge $I \subseteq [n]$ mit $\sum_{i \in I} s_i \leq t$ und $\sum_{i \in I} s_i \geq t$.
- Setze $S = \sum_{i \in I} m_i$. Dann gilt $S \subseteq M$ und $\sum_{s \in S} s = t$.
- Damit ist $(M, t) \in \text{SUBSETSUM}$

Exakte Überdeckung

Definition Exakte Überdeckung

Sei $U = \{u_1, \dots, u_n\}$ und $F = \{S_1, \dots, S_m\} \subseteq \mathcal{P}(U)$, d.h. $S_i \subseteq U$. Eine Menge $C \subseteq F$ heißt *exakte Überdeckung* von U falls

- 1 $\bigcup_{S_i \in C} S_i = U$
- 2 $S_i \cap S_j = \emptyset$ für alle $S_i, S_j \in C$ mit $i \neq j$.

COVER := $\{(U, F) \mid F \text{ enthält eine exakte Überdeckung von } U.\}$

Bsp:

- $U = \{1, 2, 3, 4, 5\}, F = \{\{2, 3\}, \{1, 3\}, \{4, 5\}, \{1\}\}$
- $C = \{\{2, 3\}, \{4, 5\}, \{1\}\}$ ist eine exakte Überdeckung von U .
- F ist *keine* exakte Überdeckung von U .

\mathcal{NP} -Vollständigkeit der exakten Überdeckung

Satz

COVER ist \mathcal{NP} -vollständig.

Zeigen

- 1 COVER $\in \mathcal{NP}$ (Übung)
- 2 $3\text{SAT} \leq_p \text{COVER}$

Idee der Reduktion

- U enthält alle Variablen x_i , Klauseln K_j und Literale ℓ_{jk} .
- F enthält geeignete Mengen für Variablen, Klauseln und Literale.

Reduktion $f(\phi) = (U, F)$

Algorithmus M_f

EINGABE: $\phi(x_1, \dots, x_n) = K_1 \wedge \dots \wedge K_m$ mit $K_j = l_{j1} \vee l_{j2} \vee l_{j3}$

- 1 Setze $U = \{x_1, \dots, x_n, K_1, \dots, K_m, l_{11}, l_{12}, l_{13}, \dots, l_{m1}, l_{m2}, l_{m3}\}$.
- 2 Definition von F als Vereinigung der Mengen
 - ▶ Variablen: $V_{0i} = \{x_i\} \cup \{l_{jk} \mid l_{jk} = x_i\}$ und
 $V_{1i} = \{x_i\} \cup \{l_{jk} \mid l_{jk} = \neg x_i\}$ für alle i, j, k .
 - ▶ Klauseln: $K_{jk} = \{K_j, l_{jk}\}$ für alle $j \in [m], k \in [3]$.
 - ▶ Literale: $L_{jk} = \{l_{jk}\}$ für alle $j \in [m], k \in [3]$.

AUSGABE: U, F

Laufzeit:

- Eingabelänge von ϕ ist $|\phi| = \Omega(m + n)$
- Schritt 1: $\mathcal{O}(n + m + |\phi|)$
- Schritt 2: Variablen $\mathcal{O}(n)$, Klauseln $\mathcal{O}(m)$, Literale $\mathcal{O}(|\phi|)$.
- D.h. die Laufzeit ist linear in der Eingabelänge.

Bsp.: $(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$

- $U = \{x_1, x_2, x_3, K_1, K_2, l_{11}, l_{12}, l_{13}, l_{21}, l_{22}, l_{23}\}$
- $V_{0i} : T_{01} = \{x_1, l_{11}\}, T_{02} = \{x_2, l_{12}, l_{22}\}, T_{30} = \{x_3, l_{33}\}$
- $V_{1i} : T_{11} = \{x_1, l_{21}\}, T_{21} = \{x_2\}, T_{31} = \{x_3, l_{13}\}$
- $K_{1k} : K_{11} = \{K_1, l_{11}\}, K_{12} = \{K_1, l_{12}\}, K_{13} = \{K_1, l_{13}\}$
- $K_{2k} : K_{21} = \{K_2, l_{21}\}, K_{22} = \{K_1, l_{22}\}, K_{23} = \{K_1, l_{23}\}$
- $L_{1k} : L_{11} = \{l_{11}\}, L_{12} = \{l_{12}\}, L_{13} = \{l_{13}\}$
- $L_{2k} : L_{21} = \{l_{21}\}, L_{22} = \{l_{22}\}, L_{23} = \{l_{23}\}$
- Erfüllende Belegung von ϕ : $x_1 = 0, x_2 = 1, x_3 = 1$.

Korrektheit: $\phi \in 3SAT \Rightarrow f(\phi) = (U, F) \in COVER$

Sei $\phi(x_1, \dots, x_n) \in 3SAT$

- Dann gibt es eine erfüllende Belegung B der Variablen x_1, \dots, x_n .
- B setzt in jeder Klausel K_j mindestens ein Literal ℓ_{jk} auf wahr.
- Definiere Menge $C \subseteq F$ mittels B :
 - ▶ Variablen: Falls $x_i = 0$, nimm V_{0i} in C auf. Sonst V_{1i} .
 - ▶ Klauseln: Nimm Menge K_{jk} , die ℓ_{jk} enthält, in C auf.
 - ▶ Literale: Für alle nicht von C abgedeckten ℓ'_{jk} , nimm L'_{jk} in C auf.
- C ist eine exakte Überdeckung, denn
 - ▶ Variablen x_i : Werden durch V_{0i}, V_{1i} abgedeckt.
 - ▶ Klauseln K_j : Werden durch K_{jk} abgedeckt.
Die paarweisen Schnitte der Mengen V_{0i}, V_{1i}, K_{jk} sind *leer*.
 - ▶ Literale ℓ'_{jk} : Werden durch L'_{jk} abgedeckt.
- Damit ist $(U, F) \in COVER$

Korrektheit: $f(\phi) = (U, F) \in \text{COVER} \Rightarrow \phi \in 3\text{SAT}$

Sei $f(\phi) = (U, F) \in \text{COVER}$

- Dann gibt es eine Menge $C \subseteq F$ mit
 - ▶ Die Vereinigung der Mengen in C deckt U ab.
 - ▶ Der paarweise Schnitt von Mengen in C ist leer.
- Damit gilt für C
 - ▶ Variablen x_i : Entweder ist V_{0i} oder V_{1i} in C .
 - ▶ Klauseln K_j : Genau eine Klauselmeng K_{jk} ist in C .
- Definieren Variablen in B : $x_i = 0$ falls $V_{0i} \in C$, sonst $x_i = 1$.
 - ▶ Die von den V_{0i}, V_{1i} abgedeckten Literale sind auf falsch gesetzt.
 - ▶ Jede Klauselmeng K_{jk} muss ein wahres Literal ℓ_{jk} enthalten.
- D.h. B ist eine erfüllende Belegung.
- Damit gilt $\phi \in 3\text{SAT}$.

Hamiltonscher Kreis

Definition Hamiltonscher Kreis

Sei G ein Graph. Ein Kreis in G , der jeden Knoten genau einmal enthält, heißt *Hamiltonscher Kreis*.

Für gerichtete Graphen definieren wir die Sprache

$\text{GH-KREIS} := \{G \mid G \text{ gerichtet, } G \text{ besitzt einen Hamiltonschen Kreis.}\}$

Für ungerichtete Graphen definieren wir analog

$\text{UH-KREIS} := \{G \mid G \text{ ungerichtet, } G \text{ besitzt Hamiltonschen Kreis.}\}$

Satz

GH-KREIS ist \mathcal{NP} -vollständig.

- Beweis kann mittels $\text{COVER} \leq_p \text{GH-KREIS}$ geführt werden.
- Wir verzichten hier auf den nicht-trivialen Beweis.

NP-Vollständigkeit von Hamiltonkreis

Satz

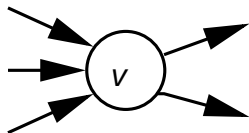
UH-KREIS ist \mathcal{NP} -vollständig.

Zeigen

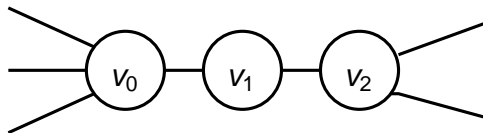
- 1 UH-KREIS $\in \mathcal{NP}$ (Übung)
- 2 GH-KREIS \leq_p UH-KREIS

Idee der Reduktion f:

Ersetze



durch



Reduktion $f(G) = G'$

Algorithmus M_f

EINGABE: $G = (V, E)$ gerichteter Graph mit $V = [n], E = [m]$

- 1 Konstruktion der Knotenmenge V' :
 - ▶ Ersetze alle $v \in V$ durch v_0, v_1, v_2
- 2 Konstruktion der Kantenmenge E' :
 - ▶ $E' = \{\{u_2, v_0\} \mid (u, v) \in E\} \cup \{\{v_0, v_1\}, \{v_1, v_2\} \mid v \in V\}$.

AUSGABE: $G' = (V', E')$ ungerichteter Graph

Laufzeit:

- Eingabelänge $|G| = \Omega(n + m)$
- Schritt 1: $\mathcal{O}(n)$, Schritt 2: $\mathcal{O}(n + m)$
- D.h. die Gesamtlaufzeit ist linear in der Eingabelänge.

Korrektheit: $G \in \text{GH-KREIS} \Leftrightarrow f(G) = G' \in \text{UH-KREIS}$

Sei $G \in \text{GH-KREIS}$

- Dann existiert eine Permutation $\pi : [n] \rightarrow [n]$, so dass G einen Hamiltonschen Kreis $H = (\pi(1), \pi(2), \dots, \pi(n), \pi(1))$ enthält.
- G' enthält den Hamiltonschen Kreis $H' = (\pi(1)_0, \pi(1)_1, \pi(1)_2, \dots, \pi(n)_0, \pi(n)_1, \pi(n)_2, \pi(1)_0, \pi(1)_1, \pi(1)_2)$.
- Damit ist $G' \in \text{UH-KREIS}$

Sei $G' \in \text{UH-KREIS}$

- G' enthält einen Hamiltonschen Kreis H' .
 - ▶ H' muss für alle $v \in V'$ die Kanten $\{v_0, v_1\}$ und $\{v_1, v_2\}$ enthalten, sonst könnte v_1 nicht in H' sein.
 - ▶ H' ist oBdA von der Form $(\pi(1)_0, \pi(1)_1, \pi(1)_2, \dots, \pi(n)_0, \pi(n)_1, \pi(n)_2, \pi(1)_0, \pi(1)_1, \pi(1)_2)$.
- G besitzt Hamiltonschen Kreis $H = (\pi(1), \pi(2), \dots, \pi(n), \pi(1))$.
- Damit ist $G \in \text{GH-KREIS}$

Übersicht unserer \mathcal{NP} -vollständigen Probleme

Vorlesung:

- SAT
- 3SAT
- CLIQUE
- KNOTENÜBERDECKUNG
- SUBSETSUM
- RUCKSACK
- COVER
- GH-KREIS
- UH-KREIS

Übung:

- TEILGRAPH
- INDEPENDENT SET
- 0,1-PROGRAMMIERUNG
- LÄNGSTER PFAD
- HALF-CLIQUE