4.2 NP-Vollständigkeit

Definition 4.14

Eine Sprache L heißt **NP-schwer**, wenn $L' \leq_p L$ für jede Sprache $L' \in \mathrm{NP}$ gilt. Ist eine Sprache L NP-schwer und gilt zusätzlich $L \in \mathrm{NP}$, so heißt L **NP-vollständig**.

Theorem 4.15

Gibt es eine NP-schwere Sprache $L \in \mathrm{P}$, so gilt $\mathrm{P} = \mathrm{NP}$.

Korollar 4.16

Es sei L eine NP-vollständige Sprache. Dann gilt $L \in \mathrm{P}$ genau dann, wenn $\mathrm{P} = \mathrm{NP}$ gilt.

Theorem 4.18 (Satz von Cook und Levin)

SAT ist NP-vollständig.

Lemma 4.19

Sei L' eine beliebige Sprache und sei L eine beliebige NP-schwere Sprache.

Gilt $L \leq_p L'$, so ist auch L' NP-schwer.

Lemma 4.19

Sei L' eine beliebige Sprache und sei L eine beliebige NP-schwere Sprache.
Gilt $L \leq_p L'$, so ist auch L' NP-schwer.

3-SAT: Einschränkung von SAT auf Eingaben, in denen **jede Klausel aus drei Literalen** besteht.

Lemma 4.19

Sei L' eine beliebige Sprache und sei L eine beliebige NP-schwere Sprache.

Gilt $L \leq_p L'$, so ist auch L' NP-schwer.

3-SAT: Einschränkung von SAT auf Eingaben, in denen **jede Klausel aus drei Literalen** besteht.

Aus SAT $\in$ NP folgt 3-SAT $\in$ NP.

Lemma 4.19

Sei L' eine beliebige Sprache und sei L eine beliebige NP-schwere Sprache.
Gilt $L \leq_p L'$, so ist auch L' NP-schwer.

3-SAT: Einschränkung von SAT auf Eingaben, in denen **jede Klausel aus drei Literalen** besteht.

Aus SAT $\in$ NP folgt 3-SAT $\in$ NP.

Theorem 4.20

Es gilt SAT $\leq_p$ 3-SAT.

$\Rightarrow$ 3-SAT ist NP-vollständig.

Beweis: Sei φ Eingabe für SAT (Formel in KNF). Wir konstruieren daraus eine Eingabe $f(\varphi) = \varphi'$ für 3-SAT (Formel in KNF mit Klauseln der Länge 3), sodass

$$\varphi \text{ ist erfüllbar} \quad \Longleftrightarrow \quad \varphi' \text{ ist erfüllbar.}$$

Beweis: Sei φ Eingabe für SAT (Formel in KNF). Wir konstruieren daraus eine Eingabe $f(\varphi) = \varphi'$ für 3-SAT (Formel in KNF mit Klauseln der Länge 3), sodass

$$\varphi \text{ ist erfüllbar} \quad \Longleftrightarrow \quad \varphi' \text{ ist erfüllbar.}$$

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ Klausel in φ mit Länge $k \neq 3$.

Beweis: Sei φ Eingabe für SAT (Formel in KNF). Wir konstruieren daraus eine Eingabe $f(\varphi) = \varphi'$ für 3-SAT (Formel in KNF mit Klauseln der Länge 3), sodass

$$\varphi \text{ ist erfüllbar} \quad \Longleftrightarrow \quad \varphi' \text{ ist erfüllbar.}$$

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ Klausel in φ mit Länge $k \neq 3$.

k = 1: Ersetze $C = \ell_1$ in φ' durch $\ell_1 \vee \ell_1 \vee \ell_1$.

Beweis: Sei φ Eingabe für SAT (Formel in KNF). Wir konstruieren daraus eine Eingabe $f(\varphi) = \varphi'$ für 3-SAT (Formel in KNF mit Klauseln der Länge 3), sodass

$$\varphi \text{ ist erfüllbar} \quad \Longleftrightarrow \quad \varphi' \text{ ist erfüllbar.}$$

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ Klausel in φ mit Länge $k \neq 3$.

$\mathbf{k = 1}$: Ersetze $C = \ell_1$ in φ' durch $\ell_1 \vee \ell_1 \vee \ell_1$.

$\mathbf{k = 2}$: Ersetze $C = \ell_1 \vee \ell_2$ in φ' durch $\ell_1 \vee \ell_1 \vee \ell_2$.

k $\geq$ 4: Ersetze $C = \ell_1 \vee \ldots \vee \ell_k$ in φ' durch

- $\ell_1 \vee \ell_2 \vee y_1^C$,
- $\neg y_{i-2}^C \vee \ell_i \vee y_{i-1}^C$ für alle $i \in \{3, \ldots, k-2\}$,
- $\neg y_{k-3}^C \vee \ell_{k-1} \vee \ell_k$.

k $\geq$ 4: Ersetze $C = \ell_1 \vee \ldots \vee \ell_k$ in φ' durch

- $\ell_1 \vee \ell_2 \vee y_1^C$,
- $\neg y_{i-2}^C \vee \ell_i \vee y_{i-1}^C$ für alle $i \in \{3, \ldots, k-2\}$,
- $\neg y_{k-3}^C \vee \ell_{k-1} \vee \ell_k$.

$\varphi' = f(\varphi)$ kann **in polynomieller Zeit berechnet** werden.

k $\geq$ 4: Ersetze $C = \ell_1 \vee \ldots \vee \ell_k$ in φ' durch

- $\ell_1 \vee \ell_2 \vee y_1^C$,
- $\neg y_{i-2}^C \vee \ell_i \vee y_{i-1}^C$ für alle $i \in \{3, \ldots, k-2\}$,
- $\neg y_{k-3}^C \vee \ell_{k-1} \vee \ell_k$.

$\varphi' = f(\varphi)$ kann **in polynomieller Zeit berechnet** werden.

Beispiele: Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee \ell_4).$$

$k \geq 4$: Ersetze $C = \ell_1 \vee \ldots \vee \ell_k$ in φ' durch

- $\ell_1 \vee \ell_2 \vee y_1^C$,
- $\neg y_{i-2}^C \vee \ell_i \vee y_{i-1}^C$ für alle $i \in \{3, \ldots, k-2\}$,
- $\neg y_{k-3}^C \vee \ell_{k-1} \vee \ell_k$.

$\varphi' = f(\varphi)$ kann **in polynomieller Zeit berechnet** werden.

Beispiele: Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee \ell_4).$$

Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee \ell_5).$$

k $\geq$ 4: Ersetze $C = \ell_1 \vee \ldots \vee \ell_k$ in φ' durch

- $\ell_1 \vee \ell_2 \vee y_1^C$,
- $\neg y_{i-2}^C \vee \ell_i \vee y_{i-1}^C$ für alle $i \in \{3, \ldots, k-2\}$,
- $\neg y_{k-3}^C \vee \ell_{k-1} \vee \ell_k$.

$\varphi' = f(\varphi)$ kann **in polynomieller Zeit berechnet** werden.

Beispiele: Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee \ell_4).$$

Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee \ell_5).$$

Ersetze $C = \ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4 \vee \ell_5 \vee \ell_6$ durch

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6).$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':

Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$j \in \{1, 2\}$**, so setzen wir **alle Variablen y_i^C auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':

Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.

Gilt **$\mathbf{j} \in \{1, 2\}$**, so setzen wir **alle Variablen $\mathbf{y}_i^C$ auf 0**.

$$(\ell_1 \vee \boxed{\ell_2} \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$j \in \{1, 2\}$**, so setzen wir **alle Variablen y_i^C auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$\forall i : y_i^C = 0$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$j \in \{1, 2\}$**, so setzen wir **alle Variablen y_i^C auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$\forall i : y_i^C = 0$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$\mathbf{j} \in \{\mathbf{k}-1, \mathbf{k}\}$**, so setzen wir **alle Variablen $\mathbf{y}_i^C$ auf 1**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt $\mathbf{j \in \{k-1, k\}}$, so setzen wir **alle Variablen $\mathbf{y_i^C}$ auf 1**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$j \in \{k-1, k\}$**, so setzen wir **alle Variablen y_i^C auf 1**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$\forall i : y_i^C = 1$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$\mathbf{j} \in \{\mathbf{k}-1, \mathbf{k}\}$**, so setzen wir **alle Variablen $\mathbf{y}_i^C$ auf 1**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$\forall i : y_i^C = 1$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':

Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.

Gilt $\mathbf{j \notin \{1, 2, k-1, k\}}$, so setzen wir

die **Variablen $\mathbf{y_1^C, \ldots, y_{j-2}^C}$ auf 1** und die **Variablen $\mathbf{y_{j-1}^C, \ldots, y_{k-3}^C}$ auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':

Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.

Gilt **$j \notin \{1, 2, k-1, k\}$**, so setzen wir

die **Variablen $y_1^C, \ldots, y_{j-2}^C$ auf 1** und die **Variablen $y_{j-1}^C, \ldots, y_{k-3}^C$ auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$\mathbf{j} \notin \{1, 2, k-1, k\}$**, so setzen wir
die **Variablen $\mathbf{y}_1^C, \ldots, \mathbf{y}_{j-2}^C$ auf 1** und die **Variablen $\mathbf{y}_{j-1}^C, \ldots, \mathbf{y}_{k-3}^C$ auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = \ldots = y_{j-2}^C = 1 \qquad y_{j-1}^C = \ldots = y_{k-3}^C = 0$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt $\mathbf{j \notin \{1, 2, k-1, k\}}$, so setzen wir
die **Variablen $\mathbf{y_1^C, \ldots, y_{j-2}^C}$ auf 1** und die **Variablen $\mathbf{y_{j-1}^C, \ldots, y_{k-3}^C}$ auf 0**.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = \ldots = y_{j-2}^C = 1 \qquad y_{j-1}^C = \ldots = y_{k-3}^C = 0$$

Sei φ erfüllbar und $\mathbf{x}^*$ eine erfüllende Belegung.

Konstruiere erfüllende Belegung für φ':
Übernimm die Belegung x^*. Wähle Belegung der Variablen y_i^C wie folgt.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ mit $k \geq 4$ eine Klausel. Dann **erfüllt die Belegung $\mathbf{x}^*$ mindestens ein Literal aus C**. Sei dies ℓ_j für ein $j \in \{1, \ldots, k\}$.
Gilt **$\mathbf{j} \notin \{1, 2, k-1, k\}$**, so setzen wir
die **Variablen $\mathbf{y}_1^C, \ldots, \mathbf{y}_{j-2}^C$ auf 1** und die **Variablen $\mathbf{y}_{j-1}^C, \ldots, \mathbf{y}_{k-3}^C$ auf 0**.

$$(\ell_1 \vee \ell_2 \vee \colorbox{lightgreen}{y_1^C}) \wedge (\neg y_1^C \vee \ell_3 \vee \colorbox{lightgreen}{y_2^C}) \wedge (\neg y_2^C \vee \colorbox{lightgreen}{ℓ_4} \vee y_3^C) \wedge (\colorbox{lightgreen}{$\neg y_3^C$} \vee \ell_5 \vee \ell_6)$$

$$y_1^C = \ldots = y_{j-2}^C = 1 \qquad y_{j-1}^C = \ldots = y_{k-3}^C = 0$$

φ erfüllbar $\Rightarrow \varphi'$ ist erfüllbar

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\boxed{\ell_1} \vee \boxed{\ell_2} \vee y_1^C) \wedge (\neg y_1^C \vee \boxed{\ell_3} \vee y_2^C) \wedge (\neg y_2^C \vee \boxed{\ell_4} \vee y_3^C) \wedge (\neg y_3^C \vee \boxed{\ell_5} \vee \boxed{\ell_6})$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = 1$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = 1 \;\Rightarrow y_2^C = 1$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = 1 \;\Rightarrow y_2^C = 1 \;\Rightarrow y_3^C = 1$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = 1 \;\Rightarrow y_2^C = 1 \;\Rightarrow y_3^C = 1$$

$$\Rightarrow (x^*, y^*) \text{ ist keine erfüllende Belegung von } \varphi'$$

Sei φ' erfüllbar und $(\mathbf{x}^*, \mathbf{y}^*)$ eine erfüllende Belegung.

Behauptung: x^* ist erfüllende Belegung für φ.

Sei $C = \ell_1 \vee \ldots \vee \ell_k$ eine Klausel aus φ.

Annahme: Kein Literal aus C wird durch x^* erfüllt.

$$(\ell_1 \vee \ell_2 \vee y_1^C) \wedge (\neg y_1^C \vee \ell_3 \vee y_2^C) \wedge (\neg y_2^C \vee \ell_4 \vee y_3^C) \wedge (\neg y_3^C \vee \ell_5 \vee \ell_6)$$

$$y_1^C = 1 \;\Rightarrow y_2^C = 1 \;\Rightarrow y_3^C = 1$$
$$\Rightarrow (x^*, y^*) \text{ ist keine erfüllende Belegung von } \varphi'$$

φ' erfüllbar $\Rightarrow \varphi$ ist erfüllbar $\square$

Theorem 4.21

Es gilt 3-SAT $\leq_p$ CLIQUE.

Theorem 4.21

Es gilt 3-SAT $\leq_p$ CLIQUE.

Beweis: Es sei $\varphi = \bigwedge_{i=1}^{m} (\bigvee_{j=1}^{3} \ell_{i,j})$ eine **Eingabe für 3-SAT** mit m Klauseln mit Variablen $x_1, \ldots, x_n$.

Theorem 4.21

Es gilt 3-SAT $\leq_p$ CLIQUE.

Beweis: Es sei $\varphi = \bigwedge_{i=1}^{m}(\bigvee_{j=1}^{3} \ell_{i,j})$ eine **Eingabe für 3-SAT** mit m Klauseln mit Variablen $x_1, \ldots, x_n$.

Erzeuge Eingabe $\mathbf{f(\varphi) = (G, k)}$ für Clique:

$G = (V, E)$ enthält $3m$ viele Knoten und zwar einen Knoten (i, j) für jedes Literal $\ell_{i,j}$.

Zwei Knoten (i, j) und (i', j') sind durch Kante verbunden, wenn $i \neq i'$ und $\ell_{i,j} \neq \neg\ell_{i',j'}$.

Theorem 4.21

Es gilt 3-SAT $\leq_p$ CLIQUE.

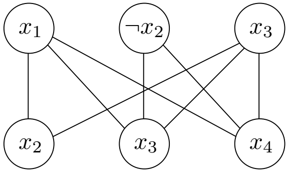
Beweis: Es sei $\varphi = \bigwedge_{i=1}^{m}(\bigvee_{j=1}^{3} \ell_{i,j})$ eine **Eingabe für 3-SAT** mit m Klauseln mit Variablen $x_1, \ldots, x_n$.

Erzeuge Eingabe $f(\varphi) = (G, k)$ für Clique:

$G = (V, E)$ enthält $3m$ viele Knoten und zwar einen Knoten (i, j) für jedes Literal $\ell_{i,j}$.

Zwei Knoten (i, j) und (i', j') sind durch Kante verbunden, wenn $i \neq i'$ und $\ell_{i,j} \neq \neg\ell_{i',j'}$.

$$(x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3 \vee x_4) \wedge \ldots$$



Sei φ erfüllbar und x^* eine erfüllende Belegung.

Sei φ erfüllbar und x* eine erfüllende Belegung.

x^* erfüllt in jeder Klausel mindestens ein Literal. Wir wählen **aus jeder Klausel ein beliebiges erfülltes Literal** aus und fügen den entsprechenden Knoten der Menge V' hinzu.

Sei φ erfüllbar und x* eine erfüllende Belegung.

x^* erfüllt in jeder Klausel mindestens ein Literal. Wir wählen **aus jeder Klausel ein beliebiges erfülltes Literal** aus und fügen den entsprechenden Knoten der Menge V' hinzu.

Per Konstruktion gilt $|V'| = k = m$.

Sei φ erfüllbar und x* eine erfüllende Belegung.

x^* erfüllt in jeder Klausel mindestens ein Literal. Wir wählen **aus jeder Klausel ein beliebiges erfülltes Literal** aus und fügen den entsprechenden Knoten der Menge V' hinzu.

Per Konstruktion gilt $|V'| = k = m$.

V' ist eine Clique, da sich erfüllte Literale nicht widersprechen.

Sei φ erfüllbar und x* eine erfüllende Belegung.

x^* erfüllt in jeder Klausel mindestens ein Literal. Wir wählen **aus jeder Klausel ein beliebiges erfülltes Literal** aus und fügen den entsprechenden Knoten der Menge V' hinzu.

Per Konstruktion gilt $|V'| = k = m$.

V' ist eine Clique, da sich erfüllte Literale nicht widersprechen.

φ erfüllbar $\Rightarrow$ G besitzt k-Clique.

Sei $V' \subseteq V$ eine k-Clique in G.

Sei $V' \subseteq V$ eine k-Clique in G.

V' kann **nicht zwei Knoten, die zur selben Klausel gehören**, enthalten.

Sei $V' \subseteq V$ eine k-Clique in G.

V' kann **nicht zwei Knoten, die zur selben Klausel gehören**, enthalten.

$\Rightarrow$ V' enthält für jede Klausel genau einen Knoten, da $k = m$.

Sei $V' \subseteq V$ eine k-Clique in G.

V' kann **nicht zwei Knoten, die zur selben Klausel gehören**, enthalten.

$\Rightarrow V'$ enthält für jede Klausel genau einen Knoten, da $k = m$.

V' enthält **für keine Variable x_j zwei Knoten, die die Literale x_j und $\neg x_j$ darstellen**.

Sei $V' \subseteq V$ eine k-Clique in G.

V' kann **nicht zwei Knoten, die zur selben Klausel gehören**, enthalten.

$\Rightarrow V'$ enthält für jede Klausel genau einen Knoten, da $k = m$.

V' enthält **für keine Variable x_j zwei Knoten, die die Literale x_j und $\neg x_j$ darstellen**.

Wir erhalten somit eine **erfüllende Belegung für φ**, indem wir alle Variablen x_j, für die das Literal x_j in V' enthalten ist, auf 1 setzen, und alle anderen auf 0.

Sei $V' \subseteq V$ eine k-Clique in G.

V' kann **nicht zwei Knoten, die zur selben Klausel gehören**, enthalten.

$\Rightarrow V'$ enthält für jede Klausel genau einen Knoten, da $k = m$.

V' enthält **für keine Variable x_j zwei Knoten, die die Literale x_j und $\neg x_j$ darstellen**.

Wir erhalten somit eine **erfüllende Belegung für φ**, indem wir alle Variablen x_j, für die das Literal x_j in V' enthalten ist, auf 1 setzen, und alle anderen auf 0.

G besitzt k-Clique $\Rightarrow \varphi$ erfüllbar

SubsetSum

Eingabe: Zahlen $z_1, \ldots, z_n \in \mathbb{N}$ sowie Zahl $b \in \mathbb{N}$

Frage: Gibt es Teilmenge $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} z_i = b$.

SubsetSum

Eingabe: Zahlen $z_1, \dots, z_n \in \mathbb{N}$ sowie Zahl $b \in \mathbb{N}$

Frage: Gibt es Teilmenge $I \subseteq \{1, \dots, n\}$ mit $\sum_{i \in I} z_i = b$.

Theorem 4.22

Das Problem SUBSETSUM ist NP-vollständig.

Beweis: Es gilt $\textsc{SubsetSum} \in \mathsf{NP}$.

Beweis: Es gilt SUBSETSUM $\in$ NP.

Es gilt **3-SAT $\leq_p$ SubsetSum**. Sei φ Eingabe für 3-SAT mit Variablen $x_1, \ldots, x_N$ und Klauseln $C_1, \ldots, C_M$. Wir erzeugen Eingabe für SUBSETSUM mit $n = 2N + 2M$ Zahlen (b nicht mitgezählt), in der jede Zahl mit **$N + M$ Ziffern im Dezimalsystem** codiert wird.

Beweis: Es gilt SUBSETSUM $\in$ NP.

Es gilt **3-SAT $\leq_p$ SubsetSum**. Sei φ Eingabe für 3-SAT mit Variablen $x_1, \ldots, x_N$ und Klauseln $C_1, \ldots, C_M$. Wir erzeugen Eingabe für SUBSETSUM mit $n = 2N + 2M$ Zahlen (b nicht mitgezählt), in der jede Zahl mit **N + M Ziffern im Dezimalsystem** codiert wird.

Für Zahl z und $j \in \{1, \ldots, N + M\}$ sei $z(j)$ die j-te Ziffer von z in Dezimaldarstellung.

Beweis: Es gilt SUBSETSUM $\in$ NP.

Es gilt **3-SAT $\leq_p$ SubsetSum**. Sei φ Eingabe für 3-SAT mit Variablen $x_1, \ldots, x_N$ und Klauseln $C_1, \ldots, C_M$. Wir erzeugen Eingabe für SUBSETSUM mit $n = 2N + 2M$ Zahlen (b nicht mitgezählt), in der jede Zahl mit **$N + M$ Ziffern im Dezimalsystem** codiert wird.

Für Zahl z und $j \in \{1, \ldots, N + M\}$ sei $z(j)$ die j-te Ziffer von z in Dezimaldarstellung.

- Für jede Variable x_i mit **$i \in \{1, \ldots N\}$ erzeugen wir zwei Zahlen a_i und $\overline{a_i}$**.
  Für $j \in \{1, \ldots, N\}$ setzen wir

$$a_i(j) = \overline{a_i}(j) = \begin{cases} 1 & \text{falls } i = j, \\ 0 & \text{falls } i \neq j. \end{cases}$$

Beweis: Es gilt $\textsc{SubsetSum} \in \mathrm{NP}$.

Es gilt **3-SAT $\leq_p$ SubsetSum**. Sei φ Eingabe für 3-SAT mit Variablen $x_1, \ldots, x_N$ und Klauseln $C_1, \ldots, C_M$. Wir erzeugen Eingabe für $\textsc{SubsetSum}$ mit $n = 2N + 2M$ Zahlen (b nicht mitgezählt), in der jede Zahl mit **$N + M$ Ziffern im Dezimalsystem** codiert wird.

Für Zahl z und $j \in \{1, \ldots, N + M\}$ sei $z(j)$ die j-te Ziffer von z in Dezimaldarstellung.

- Für jede Variable x_i mit **$i \in \{1, \ldots N\}$ erzeugen wir zwei Zahlen a_i und $\overline{a_i}$**.

  Für $j \in \{1, \ldots, N\}$ setzen wir

$$a_i(j) = \overline{a_i}(j) = \begin{cases} 1 & \text{falls } i = j, \\ 0 & \text{falls } i \neq j. \end{cases}$$

  Für $j \in \{1, \ldots, M\}$ setzen wir

$$a_i(N+j) = \begin{cases} 1 & \text{falls } x_i \text{ in } C_j \text{ enthalten,} \\ 0 & \text{sonst,} \end{cases} \qquad \overline{a_i}(N+j) = \begin{cases} 1 & \text{falls } \neg x_i \text{ in } C_j \text{ enthalten,} \\ 0 & \text{sonst.} \end{cases}$$

- Für jedes $i \in \{1, \ldots, M\}$ erzeugen wir zwei Zahlen h_i und h_i'. Es sei $h_i(N+i) = h_i'(N+i) = 1$ und alle anderen Ziffern seien 0.

- Für jedes $i \in \{1, \ldots, M\}$ erzeugen wir zwei Zahlen h_i und h'_i. Es sei $h_i(N+i) = h'_i(N+i) = 1$ und alle anderen Ziffern seien 0.
- Wir setzen

$$b(j) = \begin{cases} 1 & \text{falls } j \leq N, \\ 3 & \text{falls } j > N. \end{cases}$$

- Für jedes $i \in \{1, \dots, M\}$ erzeugen wir zwei Zahlen h_i und h'_i. Es sei $h_i(N+i) = h'_i(N+i) = 1$ und alle anderen Ziffern seien 0.
- Wir setzen

$$b(j) = \begin{cases} 1 & \text{falls } j \leq N, \\ 3 & \text{falls } j > N. \end{cases}$$

Diese Reduktion kann **in polynomieller Zeit** berechnet werden.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Beispiel:

$C_1 = x_1 \vee \neg x_2 \vee x_3$

und

$C_2 = x_2 \vee x_3 \vee \neg x_N$

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	$\mathbf{1}$	0	0	$\cdots$	0	$\mathbf{1}$	0	$\cdots$	$\cdots$
$\overline{a_1}$	$\mathbf{1}$	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	$\mathbf{1}$	0	$\cdots$	0	0	$\mathbf{1}$	$\cdots$	$\cdots$
$\overline{a_2}$	0	$\mathbf{1}$	0	$\cdots$	0	$\mathbf{1}$	0	$\cdots$	$\cdots$
a_3	0	0	$\mathbf{1}$	$\cdots$	0	$\mathbf{1}$	$\mathbf{1}$	$\cdots$	$\cdots$
$\overline{a_3}$	0	0	$\mathbf{1}$	$\cdots$	0	0	0	$\cdots$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
a_N	0	0	0	$\cdots$	$\mathbf{1}$	0	0	$\cdots$	$\cdots$
$\overline{a_N}$	0	0	0	$\cdots$	$\mathbf{1}$	0	$\mathbf{1}$	$\cdots$	$\cdots$
h_1	0	0	0	$\cdots$	0	$\mathbf{1}$	0	$\cdots$	0
h'_1	0	0	0	$\cdots$	0	$\mathbf{1}$	0	$\cdots$	0
h_2	0	0	0	$\cdots$	0	0	$\mathbf{1}$	$\cdots$	0
h'_2	0	0	0	$\cdots$	0	0	$\mathbf{1}$	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
h_M	0	0	0	$\cdots$	0	0	0	$\cdots$	$\mathbf{1}$
h'_M	0	0	0	$\cdots$	0	0	0	$\cdots$	$\mathbf{1}$
b	$\mathbf{1}$	$\mathbf{1}$	$\mathbf{1}$	$\cdots$	$\mathbf{1}$	$\mathbf{3}$	$\mathbf{3}$	$\cdots$	$\mathbf{3}$

Beispiel:

$C_1 = x_1 \vee \neg x_2 \vee x_3$

und

$C_2 = x_2 \vee x_3 \vee \neg x_N$

Allgemein gilt: Es treten **keine Überträge** auf.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei $x^\star$ **erfüllende Belegung**.

4.3 NP-vollständige Probleme

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei $x^\star$ **erfüllende Belegung**.

Konstruiere S mit

$\sum_{z \in S} z = b$ wie folgt:

$x_i^\star = 1 \Rightarrow a_i \in S$.

$x_i^\star = 0 \Rightarrow \overline{a_i} \in S$.

Fülle S ggf. mit h_i und h'_i auf.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei $x^\star$ **erfüllende Belegung**.

Konstruiere S mit

$\sum_{z \in S} z = b$ wie folgt:

$x_i^\star = 1 \Rightarrow a_i \in S$.

$x_i^\star = 0 \Rightarrow \overline{a_i} \in S$.

Fülle S ggf. mit h_i und h'_i auf.

Beispiel:

$C_1 = x_1 \vee \neg x_2 \vee x_3$

$C_2 = x_2 \vee x_3 \vee \neg x_N$

$x_1^\star = 1,\ x_2^\star = 0,$

$x_3^\star = 0,\ x_N^\star = 0$

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei $x^\star$ **erfüllende Belegung**.

Konstruiere S mit

$\sum_{z \in S} z = b$ wie folgt:

$x_i^\star = 1 \Rightarrow a_i \in S$.

$x_i^\star = 0 \Rightarrow \overline{a_i} \in S$.

Fülle S ggf. mit h_i und h'_i auf.

Beispiel:

$C_1 = x_1 \vee \neg x_2 \vee x_3$

$C_2 = x_2 \vee x_3 \vee \neg x_N$

$x_1^\star = 1,\ x_2^\star = 0,$

$x_3^\star = 0,\ x_N^\star = 0$

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei S mit $\sum_{z \in S} z = b$.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei S mit $\sum_{z \in S} z = b$.

Beobachtung: Für jedes j **genau eine der Zahlen a_j und $\overline{a_j}$** in S.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei S mit $\sum_{z \in S} z = b$.

Beobachtung: Für jedes j **genau eine der Zahlen a_j und $\overline{a_j}$** in S.

Setze

$$x_j^\star = \begin{cases} 1 & \text{falls } a_j \text{ in } S, \\ 0 & \text{falls } \overline{a_j} \text{ in } S. \end{cases}$$

$x^\star$ ist **erfüllende Belegung**.

	1	2	3	$\cdots$	N	$N+1$	$N+2$	$\cdots$	$N+M$
a_1	**1**	0	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
$\overline{a_1}$	**1**	0	0	$\cdots$	0	0	0	$\cdots$	$\cdots$
a_2	0	**1**	0	$\cdots$	0	0	**1**	$\cdots$	$\cdots$
$\overline{a_2}$	0	**1**	0	$\cdots$	0	**1**	0	$\cdots$	$\cdots$
a_3	0	0	**1**	$\cdots$	0	**1**	**1**	$\cdots$	$\cdots$
$\overline{a_3}$	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	$\cdots$	$\cdots$
$\overline{a_N}$	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**

Sei S mit $\sum_{z \in S} z = b$.

Beobachtung: Für jedes j **genau eine der Zahlen a_j und $\overline{a_j}$** in S.

Setze

$$x_j^\star = \begin{cases} 1 & \text{falls } a_j \text{ in } S, \\ 0 & \text{falls } \overline{a_j} \text{ in } S. \end{cases}$$

$x^\star$ ist **erfüllende Belegung**.

Beispiel:

$x_1^\star = 1$, $x_2^\star = 0$,

$x_3^\star = 0$, $x_N^\star = 0$ $\square$

PARTITION

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

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

PARTITION

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

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

Theorem 4.23

Das Problem PARTITION ist NP-vollständig.

PARTITION

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

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

Theorem 4.23

Das Problem PARTITION ist NP-vollständig.

Beweis: PARTITION ist ein Spezialfall von SUBSETSUM und gehört damit zu NP.

Es gilt $\textsc{SubsetSum} \leq_p \textsc{Partition}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine **Eingabe für** $\textsc{SubsetSum}$.

Es gilt $\textsc{SubsetSum} \leq_p \textsc{Partition}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine **Eingabe für** $\textsc{SubsetSum}$.

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

Wir konstruieren die folgende **Eingabe für** $\textsc{Partition}$ mit den Zahlen $a'_1, \ldots, a'_{n+2}$:

- Für $i \in \{1, \ldots, n\}$ sei $a'_i = a_i$.
- Es sei $a'_{n+1} = 2A - b$.
- Es sei $a'_{n+2} = A + b$.

Es gilt $\sum_{i=1}^{n+2} a'_i = 4A$.

Es gilt $\textsc{SubsetSum} \leq_p \textsc{Partition}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine **Eingabe für** $\textsc{SubsetSum}$.

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

Wir konstruieren die folgende **Eingabe für** $\textsc{Partition}$ mit den Zahlen $a'_1, \ldots, a'_{n+2}$:

- Für $i \in \{1, \ldots, n\}$ sei $a'_i = a_i$.
- Es sei $a'_{n+1} = 2A - b$.
- Es sei $a'_{n+2} = A + b$.

Es gilt $\sum_{i=1}^{n+2} a'_i = 4A$.

Diese Reduktion kann **in polynomieller Zeit** berechnet werden.

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.

$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.

$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

Sei $J \subseteq \{1, \ldots, n+2\}$ eine Menge mit $\sum_{i \in J} a'_i = \sum_{i \in \{1,\ldots,n+2\} \setminus J} a'_i = 2A$.

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.

$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

Sei J $\subseteq \{1, \ldots, n+2\}$ eine Menge mit $\sum_{i \in J} a'_i = \sum_{i \in \{1,\ldots,n+2\} \setminus J} a'_i = 2A$.

Wegen $\sum_{i=1}^{n} a'_i = A < 2A$ gilt entweder $(n+1) \in J$ oder $(n+2) \in J$. Es können aber nicht sowohl $n+1$ als auch $n+2$ zu J gehören, da $a'_{n+1} + a'_{n+2} = 3A > 2A$ gilt.

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.

$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

Sei $J \subseteq \{1, \ldots, n+2\}$ eine Menge mit $\sum_{i \in J} a'_i = \sum_{i \in \{1,\ldots,n+2\} \setminus J} a'_i = 2A$.

Wegen $\sum_{i=1}^{n} a'_i = A < 2A$ gilt entweder $(n+1) \in J$ oder $(n+2) \in J$. Es können aber nicht sowohl $n+1$ als auch $n+2$ zu J gehören, da $a'_{n+1} + a'_{n+2} = 3A > 2A$ gilt.

Sei o. B. d. A. $(n+1) \in J$. Setze $I = J \setminus \{n+1\}$. Dann gilt

$$2A = \sum_{i \in J} a'_i = \sum_{i \in I} a'_i + a'_{n+1} = \sum_{i \in I} a'_i + 2A - b,$$

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.
$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

Sei $J \subseteq \{1, \ldots, n+2\}$ eine Menge mit $\sum_{i \in J} a'_i = \sum_{i \in \{1,\ldots,n+2\} \setminus J} a'_i = 2A$.
Wegen $\sum_{i=1}^{n} a'_i = A < 2A$ gilt entweder $(n+1) \in J$ oder $(n+2) \in J$. Es können aber nicht sowohl $n+1$ als auch $n+2$ zu J gehören, da $a'_{n+1} + a'_{n+2} = 3A > 2A$ gilt.

Sei o. B. d. A. $(n+1) \in J$. Setze $I = J \setminus \{n+1\}$. Dann gilt

$$2A = \sum_{i \in J} a'_i = \sum_{i \in I} a'_i + a'_{n+1} = \sum_{i \in I} a'_i + 2A - b,$$

woraus $\sum_{i \in I} a'_i = \sum_{i \in I} a_i = b$ folgt.

zu zeigen: Es gibt $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$.

$\iff$ Es gibt $J \subseteq \{1, \ldots, n+2\}$ mit $\sum_{i \in J} a'_i = 2A$.

Sei $\mathbf{J} \subseteq \{1, \ldots, n+2\}$ eine Menge mit $\sum_{i \in J} a'_i = \sum_{i \in \{1,\ldots,n+2\} \setminus J} a'_i = 2A$.

Wegen $\sum_{i=1}^{n} a'_i = A < 2A$ gilt entweder $(n+1) \in J$ oder $(n+2) \in J$. Es können aber nicht sowohl $n+1$ als auch $n+2$ zu J gehören, da $a'_{n+1} + a'_{n+2} = 3A > 2A$ gilt.

Sei o. B. d. A. $(n+1) \in J$. Setze $I = J \setminus \{n+1\}$. Dann gilt

$$2A = \sum_{i \in J} a'_i = \sum_{i \in I} a'_i + a'_{n+1} = \sum_{i \in I} a'_i + 2A - b,$$

woraus $\sum_{i \in I} a'_i = \sum_{i \in I} a_i = b$ folgt.

Die Menge I ist also **eine Lösung für die gegebene Eingabe von SUBSETSUM**.

Sei nun umgekehrt **eine Menge $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$** gegeben.

Sei nun umgekehrt **eine Menge $\mathbf{I} \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = b$** gegeben.

Für $J = I \cup \{n+1\}$ gilt

$$\sum_{i \in J} a'_i = \sum_{i \in I} a'_i + a'_{n+1} = \sum_{i \in I} a_i + a'_{n+1} = b + (2A - b) = 2A.$$

Sei nun umgekehrt **eine Menge $\mathbf{I} \subseteq \{1, \dots, n\}$ mit $\sum_{i \in I} a_i = b$** gegeben.
Für $J = I \cup \{n+1\}$ gilt

$$\sum_{i \in J} a'_i = \sum_{i \in I} a'_i + a'_{n+1} = \sum_{i \in I} a_i + a'_{n+1} = b + (2A - b) = 2A.$$

Dementsprechend gilt auch $\sum_{i \in \{1,\dots,n+2\} \setminus J} a'_i = 4A - \sum_{i \in J} a'_i = 2A$ und **damit ist J eine Lösung für die konstruierte Instanz von PARTITION**. $\square$

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen SubsetSum $\leq_p$ KP.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für SubsetSum.

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen SUBSETSUM $\leq_p$ KP.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für SUBSETSUM.

Konstruiere **Eingabe für KP mit n Objekten**. Für $i \in \{1, \ldots, n\}$ sei $p_i = w_i = a_i$. Es sei $t = z = b$. Existiert $I \subseteq \{1, \ldots, n\}$ mit $w(I) \leq t = b$ und $p(I) \geq z = b$?

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen SUBSETSUM $\leq_p$ KP.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für SUBSETSUM.

Konstruiere **Eingabe für KP mit n Objekten**. Für $i \in \{1, \ldots, n\}$ sei $p_i = w_i = a_i$. Es sei $t = z = b$. Existiert $I \subseteq \{1, \ldots, n\}$ mit $w(I) \leq t = b$ und $p(I) \geq z = b$?

Sei $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = \sum_{i \in I} w_i \leq b$ und $\sum_{i \in I} a_i = \sum_{i \in I} p_i \geq b$. Dann ist $\sum_{i \in I} a_i = b$.

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen $\textsc{SubsetSum} \leq_p \text{KP}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für $\textsc{SubsetSum}$.

Konstruiere **Eingabe für KP mit n Objekten**. Für $i \in \{1, \ldots, n\}$ sei $p_i = w_i = a_i$. Es sei $t = z = b$. Existiert $I \subseteq \{1, \ldots, n\}$ mit $w(I) \leq t = b$ und $p(I) \geq z = b$?

Sei $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = \sum_{i \in I} w_i \leq b$ und $\sum_{i \in I} a_i = \sum_{i \in I} p_i \geq b$. Dann ist $\sum_{i \in I} a_i = b$. **Lösung für KP $\Rightarrow$ Lösung für SubsetSum**

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen $\textsc{SubsetSum} \leq_p \mathrm{KP}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für $\textsc{SubsetSum}$.

Konstruiere **Eingabe für KP mit n Objekten**. Für $i \in \{1, \ldots, n\}$ sei $p_i = w_i = a_i$. Es sei $t = z = b$. Existiert $I \subseteq \{1, \ldots, n\}$ mit $w(I) \leq t = b$ und $p(I) \geq z = b$?

Sei $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = \sum_{i \in I} w_i \leq b$ und $\sum_{i \in I} a_i = \sum_{i \in I} p_i \geq b$. Dann ist $\sum_{i \in I} a_i = b$. **Lösung für KP $\Rightarrow$ Lösung für SubsetSum**

Sei $I \subseteq \{1, \ldots, n\}$ eine Auswahl mit $\sum_{i \in I} a_i = b$. Dann gilt $\sum_{i \in I} w_i = \sum_{i \in I} a_i \leq b$ und $\sum_{i \in I} p_i = \sum_{i \in I} a_i \geq b$.

Theorem 4.24

Die Entscheidungsvariante des Rucksackproblems (KP) ist NP-vollständig.

Beweis: Das Rucksackproblem liegt in NP (Theorem 4.7).

Wir zeigen $\textsc{SubsetSum} \leq_p \mathrm{KP}$.

Sei $a_1, \ldots, a_n \in \mathbb{N}$, $b \in \mathbb{N}$ eine Eingabe für $\textsc{SubsetSum}$.

Konstruiere **Eingabe für KP mit n Objekten**. Für $i \in \{1, \ldots, n\}$ sei $p_i = w_i = a_i$. Es sei $t = z = b$. Existiert $I \subseteq \{1, \ldots, n\}$ mit $w(I) \leq t = b$ und $p(I) \geq z = b$?

Sei $I \subseteq \{1, \ldots, n\}$ mit $\sum_{i \in I} a_i = \sum_{i \in I} w_i \leq b$ und $\sum_{i \in I} a_i = \sum_{i \in I} p_i \geq b$. Dann ist $\sum_{i \in I} a_i = b$. **Lösung für KP $\Rightarrow$ Lösung für SubsetSum**

Sei $I \subseteq \{1, \ldots, n\}$ eine Auswahl mit $\sum_{i \in I} a_i = b$. Dann gilt $\sum_{i \in I} w_i = \sum_{i \in I} a_i \leq b$ und $\sum_{i \in I} p_i = \sum_{i \in I} a_i \geq b$. **Lösung für SubsetSum $\Rightarrow$ Lösung für KP** $\square$

Definition

Algorithmen, deren Laufzeiten polynomiell von der Eingabelänge und den in der Eingabe vorkommenden Zahlen abhängen, nennt man **pseudopolynomiell**.

Definition

Algorithmen, deren Laufzeiten polynomiell von der Eingabelänge und den in der Eingabe vorkommenden Zahlen abhängen, nennt man **pseudopolynomiell**.

Theorem 4.24 zeigt, dass es unter der Annahme $\mathrm{P} \neq \mathrm{NP}$ keinen polynomiellen Algorithmus für das Rucksackproblem gibt. Das dynamische Programm (Laufzeit $O(N^2 W)$) zeigt, dass das Rucksackproblem **nur dann schwer sein kann, wenn die Gewichte sehr groß** sind.

Definition

Algorithmen, deren Laufzeiten polynomiell von der Eingabelänge und den in der Eingabe vorkommenden Zahlen abhängen, nennt man **pseudopolynomiell**.

Theorem 4.24 zeigt, dass es unter der Annahme $P \neq NP$ keinen polynomiellen Algorithmus für das Rucksackproblem gibt. Das dynamische Programm (Laufzeit $O(N^2 W)$) zeigt, dass das Rucksackproblem **nur dann schwer sein kann, wenn die Gewichte sehr groß** sind.

Definition

NP-schwere Probleme, in deren Eingaben Zahlen vorkommen und die für Eingaben mit polynomiell großen Zahlen polynomiell lösbar sind, nennt man **schwach NP-schwere Probleme**. Probleme, die bereits für polynomiell in der Eingabegröße beschränkte Zahlen NP-schwer sind, heißen **stark NP-schwer**.

Definition

Algorithmen, deren Laufzeiten polynomiell von der Eingabelänge und den in der Eingabe vorkommenden Zahlen abhängen, nennt man **pseudopolynomiell**.

Theorem 4.24 zeigt, dass es unter der Annahme $P \neq NP$ keinen polynomiellen Algorithmus für das Rucksackproblem gibt. Das dynamische Programm (Laufzeit $O(N^2 W)$) zeigt, dass das Rucksackproblem **nur dann schwer sein kann, wenn die Gewichte sehr groß** sind.

Definition

NP-schwere Probleme, in deren Eingaben Zahlen vorkommen und die für Eingaben mit polynomiell großen Zahlen polynomiell lösbar sind, nennt man **schwach NP-schwere Probleme**. Probleme, die bereits für polynomiell in der Eingabegröße beschränkte Zahlen NP-schwer sind, heißen **stark NP-schwer**.

Beispiel: Das TSP ist stark NP-schwer.

Unter der Annahme P $\neq$ NP gibt es für NP-schwere Probleme **keine Algorithmen mit polynomieller Laufzeit**.

Dies bedeutet jedoch **nicht zwangsläufig, dass exponentielle Rechenzeit benötigt wird, um diese Probleme zu lösen**.

Unter der Annahme $P \neq NP$ gibt es für NP-schwere Probleme **keine Algorithmen mit polynomieller Laufzeit**.

Dies bedeutet jedoch **nicht zwangsläufig, dass exponentielle Rechenzeit benötigt wird, um diese Probleme zu lösen**.

Denkbar wäre beispielsweise ein Algorithmus für SAT mit einer Laufzeit von $O(n^{\log n})$.

Solche Algorithmen sind allerdings **nicht bekannt** und es gibt stärkere Annahmen, die die Existenz solcher Algorithmen ausschließen.

Unter der Annahme $P \neq NP$ gibt es für NP-schwere Probleme **keine Algorithmen mit polynomieller Laufzeit**.

Dies bedeutet jedoch **nicht zwangsläufig, dass exponentielle Rechenzeit benötigt wird, um diese Probleme zu lösen**.

Denkbar wäre beispielsweise ein Algorithmus für SAT mit einer Laufzeit von $O(n^{\log n})$.

Solche Algorithmen sind allerdings **nicht bekannt** und es gibt stärkere Annahmen, die die Existenz solcher Algorithmen ausschließen.

Die (unbewiesene) **Exponentialzeithypothese** besagt beispielsweise, dass es für 3-SAT eine Konstante $\delta > 0$ gibt, sodass SAT nicht in Zeit $O(2^{\delta n})$ gelöst werden kann.