Gliederung

Erfüllbarkeitsproblem

SAT

Eingabe: aussagenlogische Formel F

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0, 1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Erfüllbarkeitsproblem

SAT

Eingabe: aussagenlogische Formel F

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Beispiele

$0, 1,\qquad x_1, x_2, \overline{x_3},\qquad (x_1 \wedge \overline{x_2}),\qquad (\overline{(x_1 \wedge \overline{x_2})} \vee x_2 \vee \overline{x_3})$

Erfüllbarkeitsproblem

SAT

Eingabe: aussagenlogische Formel F

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Beispiele

$0, 1,$ $\quad x_1, x_2, \overline{x_3},$ $\quad (x_1 \wedge \overline{x_2}),$ $\quad (\overline{(x_1 \wedge \overline{x_2})} \vee x_2 \vee \overline{x_3})$

Theorem (Satz von Cook und Levin)

SAT ist NP-vollständig.

Erfüllbarkeitsproblem

SAT

Eingabe: aussagenlogische Formel F

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Beispiele

$$0, 1, \qquad x_1, x_2, \overline{x_3}, \qquad (x_1 \wedge \overline{x_2}), \qquad (\overline{(x_1 \wedge \overline{x_2})} \vee x_2 \vee \overline{x_3})$$

Theorem (Satz von Cook und Levin)

SAT ist NP-vollständig.

„guess & check“

Beweis (Idee, Details später)

Teil 1: „SAT $\in$ NP“: rate erfüllende Belegung (Zertifikat) und verifiziere sie.

Teil 2: „SAT ist NP-schwer“: mit $L \in \mathrm{NP}$ beliebig,
transformiere NTM N mit $T(N) = L$ in Formel $\varphi(x)$ sodass $x \in L \Leftrightarrow \varphi(x) \in \mathrm{SAT}$.

CNF-SAT ist NP-vollständig

Konjunktion v. Disjunktionen

$(x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge \ldots$

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0, 1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

$\mathrm{CNF\text{-}SAT}$ ist NP-vollständig

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0, 1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

$\mathrm{SAT} \leq_m^p \mathrm{CNF\text{-}SAT}$ ($\leadsto$ $\mathrm{CNF\text{-}SAT}$ NP-vollständig)

CNF-SAT ist NP-vollständig

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

$\mathrm{SAT} \leq_m^p \mathrm{CNF\text{-}SAT}$ ($\rightsquigarrow$ $\mathrm{CNF\text{-}SAT}$ NP-vollständig)

Beweis (Skizze)

Reduktion: $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente Formel ψ:

CNF-SAT ist NP-vollständig

$\boxed{x \to y \equiv \overline{x} \vee y}$

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform"

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

SAT $\leq^p_m$ CNF-SAT ($\rightsquigarrow$ CNF-SAT NP-vollständig)

Beweis (Skizze)

Reduktion: $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente Formel ψ:

$$\psi = (y_1 \leftrightarrow (\overline{x_1} \wedge x_2 \wedge x_1)) \wedge (y_2 \leftrightarrow (x_1 \wedge \overline{x_3})) \wedge y_5 \wedge (y_3 \leftrightarrow \ldots) \wedge (y_4 \leftrightarrow (y_1 \vee y_2)) \wedge (y_5 \leftrightarrow (y_4 \wedge \overline{y_3}))$$

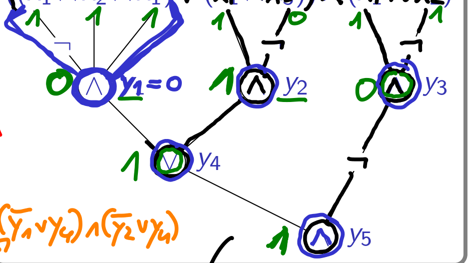
$$y_4 \leftrightarrow (y_1 \vee y_2) \equiv (y_4 \to (y_1 \vee y_2)) \wedge ((y_1 \vee y_2) \to y_4)$$

$$(\overline{y_4} \vee y_1 \vee y_2) \wedge ((\overline{y_1} \wedge \overline{y_2}) \vee y_4)$$

Beispiel

$x_1 = 1 \quad x_2 = 1 \quad x_3 = 0$

$$\varphi = ((\overline{x_1} \wedge x_2 \wedge x_1) \vee (x_1 \wedge \overline{x_3})) \wedge \overline{(x_1 \wedge \overline{x_2})}$$



$(\overline{y_1} \vee y_4) \wedge (\overline{y_2} \vee y_4)$

„Formelbaum"

CNF-SAT ist NP-vollständig

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

$\mathrm{SAT} \leq_m^p \mathrm{CNF\text{-}SAT}$ ($\rightsquigarrow$ $\mathrm{CNF\text{-}SAT}$ NP-vollständig)

Beweis (Skizze)

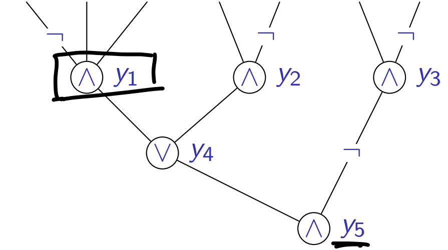
Reduktion: $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente Formel ψ:

(1) neue Variable y_i für jeden Knoten im “Formelbaum” mit “äquivalentem Wahrheitswert”

(2) neue Klausel für die Wurzel

Beispiel

$((\overline{x_1} \wedge x_2 \wedge x_1) \vee (x_1 \wedge \overline{x_3})) \wedge \overline{(x_1 \wedge \overline{x_2})}$



CNF-SAT ist NP-vollständig

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0, 1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

SAT $\leq_m^p$ CNF-SAT ($\leadsto$ CNF-SAT NP-vollständig)

Beweis (Skizze)

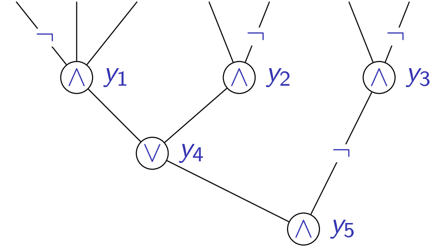
Reduktion: $\varphi \leadsto$ **erfüllbarkeits**-äquivalente Formel ψ:
(1) neue Variable y_i für jeden Knoten im “Formelbaum” mit “äquivalentem Wahrheitswert”
(2) neue Klausel für die Wurzel

ψ erfüllbar $\Leftrightarrow$ φ erfüllbar ✓

→ KNF

Beispiel

$$((\overline{x_1} \wedge x_2 \wedge x_1) \vee (x_1 \wedge \overline{x_3})) \wedge \overline{(x_1 \wedge \overline{x_2})}$$



CNF-SAT ist NP-vollständig

CNF-SAT

Eingabe: aussagenlogische Formel F in „konjunktiver Normalform“

Frage: Ist F **erfüllbar**, d.h. gibt es eine $\{0,1\}$-wertige Belegung der in F verwendeten Booleschen Variablen derart, dass F zu **wahr** (d.h. 1) ausgewertet wird?

Theorem

$\mathrm{SAT} \leq_m^p \mathrm{CNF\text{-}SAT}$ ($\leadsto$ CNF-SAT NP-vollständig)

Beweis (Skizze)

Reduktion: $\varphi \leadsto$ **erfüllbarkeits**-äquivalente Formel ψ:

(1) neue Variable y_i für jeden Knoten im “Formelbaum” mit “äquivalentem Wahrheitswert”

(2) neue Klausel für die Wurzel

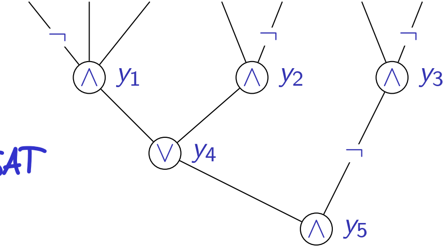
$\langle\varphi\rangle \in \mathrm{SAT} \Leftrightarrow \langle\psi\rangle \in \mathrm{CNF\text{-}SAT}$

ψ erfüllbar $\Leftrightarrow$ φ erfüllbar ✓

poly-time computable ✓ (ab jetzt implizit)

Beispiel

$((\overline{x_1} \wedge x_2 \wedge x_1) \vee (x_1 \wedge \overline{x_3})) \wedge \overline{(x_1 \wedge \overline{x_2})}$



3-SAT ist NP-vollständig

$(x_1 \vee \bar{x}_2 \vee x_3) \wedge (\quad) \wedge (\quad)$

≤ 3 Literale

Formel in 3-KNF

Theorem

CNF-SAT $\leq_m^p$ 3-SAT (also ist 3-SAT NP-vollständig).

3-SAT ist NP-vollständig

Theorem

$\text{CNF-SAT} \leq_m^p 3\text{-SAT}$ (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ
Für jede Klausel $c_j = (\ell_1 \vee \ell_2 \vee \ldots \vee \ell_r) \in \varphi$,

3-SAT ist NP-vollständig

Theorem

CNF-SAT $\leq_m^p$ 3-SAT (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ

Für jede Klausel $c_j = (\ell_1 \vee \ell_2 \vee \ldots \vee \ell_r) \in \varphi$,

- falls $r \leq 3$, dann füge c_j zu ψ hinzu;

3-SAT ist NP-vollständig

Theorem

CNF-SAT $\leq_m^p$ 3-SAT (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ

Für jede Klausel $c_j = (\ell_1 \vee \ell_2 \vee \ldots \vee \ell_r) \in \varphi$,

- falls $r \leq 3$, dann füge c_j zu ψ hinzu;
- sonst füge c'_j hinzu mit

$\underbrace{(\overline{y_1} \vee \ell_3 \vee \ell_4 \vee \ldots \vee \ell_r)}$ $\rightarrow$ Länge $r-2+1 = r-1$

$$c'_j := (\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y_1} \vee \ell_3 \vee y_2) \wedge (\overline{y_2} \vee \ell_4 \vee y_3) \ldots (\overline{y_{r-3}} \vee \ell_{r-1} \vee \ell_r)$$

wobei $y_1, \ldots, y_{r-3}$ neue Variablen sind.

„$\Rightarrow$" c_j erfüllt von β $\rightsquigarrow \exists$ Literal ℓ_i s.d. $\beta(\ell_i) = 1$. Fall 1 $i \leq 2 \rightsquigarrow \beta \cup \{(y_1, 0)\}$ erfüllt
Fall 2 $i > 2 \rightsquigarrow \beta \cup \{(y_1, 1)\}$ erfüllt

„$\Leftarrow$" c'_j erfüllt v. β. Fall 1: $\beta(y_1) = 0 \rightsquigarrow \beta(\ell_1) = 1$ oder $\beta(\ell_2) = 1 \rightsquigarrow \beta$ erfüllt c_j
Fall 2: $\beta(y_1) = 1 \rightsquigarrow \exists_{i>2}\, \beta(\ell_i) = 1 \rightsquigarrow \beta$ erfüllt c_j

3-SAT ist NP-vollständig

Theorem

$\text{CNF-SAT} \leq_m^p 3\text{-SAT}$ (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ
Für jede Klausel $c_j = (\ell_1 \vee \ell_2 \vee \ldots \vee \ell_r) \in \varphi$,

- falls $r \leq 3$, dann füge c_j zu ψ hinzu;
- sonst füge c_j' hinzu mit

$$c_j' := (\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y_1} \vee \ell_3 \vee y_2) \wedge (\overline{y_2} \vee \ell_4 \vee y_3) \ldots (\overline{y_{r-3}} \vee \ell_{r-1} \vee \ell_r)$$

wobei $y_1, \ldots, y_{r-3}$ neue Variablen sind.

$\rightsquigarrow$ Belegung β erfüllt $c_j \Leftrightarrow$ Erweiterung von β erfüllt c_j'

3-SAT ist NP-vollständig

Theorem

$\text{CNF-SAT} \leq_m^p 3\text{-SAT}$ (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \rightsquigarrow$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ
Für jede Klausel $c_j = (\ell_1 \vee \ell_2 \vee \ldots \vee \ell_r) \in \varphi$,

- falls $r \leq 3$, dann füge c_j zu ψ hinzu;
- sonst füge c_j' hinzu mit

$$c_j' := (\ell_1 \vee \ell_2 \vee y_1) \wedge (\overline{y_1} \vee \ell_3 \vee y_2) \wedge (\overline{y_2} \vee \ell_4 \vee y_3) \ldots (\overline{y_{r-3}} \vee \ell_{r-1} \vee \ell_r)$$

  wobei $y_1, \ldots, y_{r-3}$ neue Variablen sind.
  $\rightsquigarrow$ Belegung β erfüllt $c_j \Leftrightarrow$ Erweiterung von β erfüllt c_j'

ψ erfüllbar $\Leftrightarrow$ φ erfüllbar ✓

3-SAT ist NP-vollständig

Theorem

CNF-SAT $\leq_m^p$ 3-SAT (also ist 3-SAT NP-vollständig).

Beweis (Skizze)

Reduktion: CNF-Formel $\varphi \leadsto$ **erfüllbarkeits**-äquivalente 3CNF-Formel ψ
Für jede Klausel $c_j = (\ell_1 \lor \ell_2 \lor \ldots \lor \ell_r) \in \varphi$,

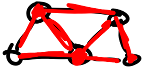
- falls $r \leq 3$, dann füge c_j zu ψ hinzu;
- sonst füge c_j' hinzu mit

$$c_j' := (\ell_1 \lor \ell_2 \lor y_1) \land (\overline{y_1} \lor \ell_3 \lor y_2) \land (\overline{y_2} \lor \ell_4 \lor y_3) \ldots (\overline{y_{r-3}} \lor \ell_{r-1} \lor \ell_r)$$

wobei $y_1, \ldots, y_{r-3}$ neue Variablen sind.
$\leadsto$ Belegung β erfüllt $c_j \Leftrightarrow$ Erweiterung von β erfüllt c_j'

ψ erfüllbar $\Leftrightarrow \varphi$ erfüllbar ✓
Bemerkung: $|\psi| \leq 3|\varphi|$

Vertex Cover ist NP-vollständig



Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Vertex Cover ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\mathrm{Var} + 2\#\mathrm{Klauseln})$

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$

Vertex Cover ist NP-vollständig

Theorem

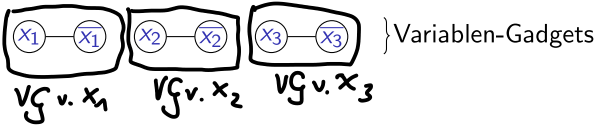
3-SAT $\leq_m^p$ Vertex Cover.

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



Vertex Cover ist NP-vollständig

Theorem

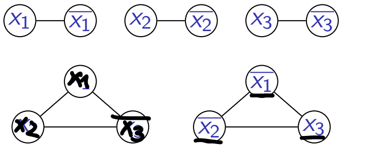
3-SAT $\leq_m^p$ Vertex Cover.

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \lor \ell_{i_2} \lor \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$

Beispiel: $(x_1 \lor x_2 \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor x_3)$

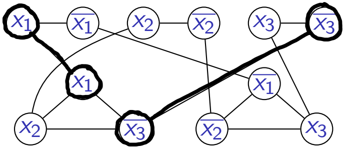


}Variablen-Gadgets

}Klausel-Gadgets

Vertex Cover ist NP-vollständig

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



Variablen-Gadgets

Klausel-Gadgets

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung zwischen Var-Gadgets & Klausel-Gadgets

Vertex Cover ist NP-vollständig

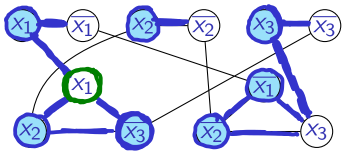
Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

Beispiel: $(\overset{1}{x_1} \lor \overset{1}{x_2} \lor \overline{x_3}) \land (\overline{x_1} \lor \overline{x_2} \lor \overset{1}{x_3})$



}Variablen-Gadgets

}Klausel-Gadgets

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \lor \ell_{i_2} \lor \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung

„$\Rightarrow$“: aus Variablen-Gadget, wähle entsprechend der Belegung

$\rightsquigarrow$ alle anderen Kanten mit 2 Knoten aus jedem Klausel-Gadget überdeckt

(1) Kanten in Var.-Gadgets ✓

(2) Kanten in Klausel-Gadgets ✓

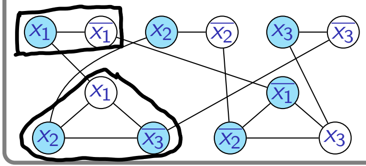
(3) Kanten zwischen VGs & KGs ✓

Vertex Cover ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung

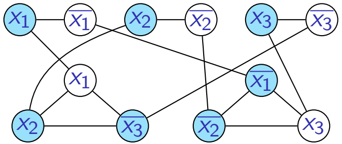
„$\Rightarrow$“: aus Variablen-Gadget, wähle entsprechend der Belegung
$\rightsquigarrow$ alle anderen Kanten mit 2 Knoten aus jedem Klausel-Gadget überdeckt
„$\Leftarrow$“:
(a) ≥ 1 Knoten von jedem Variablen-Gadget in jeder VC-Lösung
(b) ≥ 2 Knoten von jedem Klausel-Gadget in jeder VC-Lösung.

Vertex Cover ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



Variablen-Gadgets

Klausel-Gadgets

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung

„$\Rightarrow$“: aus Variablen-Gadget, wähle entsprechend der Belegung

$\rightsquigarrow$ alle anderen Kanten mit 2 Knoten aus jedem Klausel-Gadget überdeckt

„$\Leftarrow$“:

(a) $= 1$ Knoten von jedem Variablen-Gadget in jeder VC-Lösung

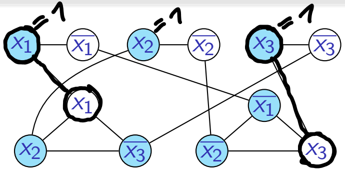
(b) $= 2$ Knoten von jedem Klausel-Gadget in jeder VC-Lösung.

Vertex Cover ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$

B =


} Variablen-Gadgets

} Klausel-Gadgets

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung

„$\Rightarrow$“: aus Variablen-Gadget, wähle entsprechend der Belegung

$\rightsquigarrow$ alle anderen Kanten mit 2 Knoten aus jedem Klausel-Gadget überdeckt

„$\Leftarrow$“:

(a) $= 1$ Knoten von jedem Variablen-Gadget in jeder VC-Lösung

(b) $= 2$ Knoten von jedem Klausel-Gadget in jeder VC-Lösung.

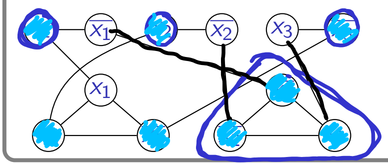
$\rightsquigarrow$ jedes Klausel-Gadget benachbart zu einem Knoten in VC-Lösung

Vertex Cover ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Vertex Cover.

Beispiel: $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_3)$



}Variablen-Gadgets

$\rightsquigarrow 8$

}Klausel-Gadgets

Beweis (Skizze)

Formel $\varphi \rightsquigarrow (G, k = \#\text{Var} + 2\#\text{Klauseln})$

1. Variablen-Gadget: Variable $x_i \rightsquigarrow$ 2 benachbarte Knoten mit Beschriftungen x_i und $\overline{x_i}$
2. Klausel-Gadget: Klausel $(\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3}) \rightsquigarrow$ Dreieck mit Beschriftungen $\ell_{i_1}, \ell_{i_2}, \ell_{i_3}$
3. Verbinde Knoten mit gleicher Beschriftung

„$\Rightarrow$“: aus Variablen-Gadget, wähle entsprechend der Belegung

$\rightsquigarrow$ alle anderen Kanten mit 2 Knoten aus jedem Klausel-Gadget überdeckt

„$\Leftarrow$“:

(a) $= 1$ Knoten von jedem Variablen-Gadget in jeder VC-Lösung

(b) $= 2$ Knoten von jedem Klausel-Gadget in jeder VC-Lösung.

$\rightsquigarrow$ jedes Klausel-Gadget benachbart zu einem Knoten in VC-Lösung

$\rightsquigarrow$ entsprechende Belegung erfüllt die Formel!

Dominating Set ist NP-vollständig

Theorem

Vertex Cover $\leq_m^p$ Dominating Set.

DOMINATING SET ist NP-vollständig

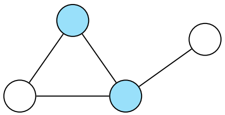
Theorem

VERTEX COVER $\leq_m^p$ DOMINATING SET.

Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

Beispiel



$k = 2$

VERTEX COVER

$k = 2$

DOMINATING SET

Dominating Set ist NP-vollständig

Theorem

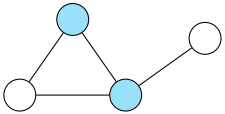
Vertex Cover $\leq_m^p$ Dominating Set.

Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

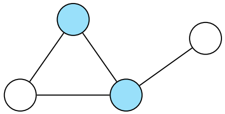
1. setze initial $G' = G$

Beispiel



$k = 2$

Vertex Cover



$k = 2$

Dominating Set

DOMINATING SET ist NP-vollständig

Theorem

VERTEX COVER $\leq_m^p$ DOMINATING SET.

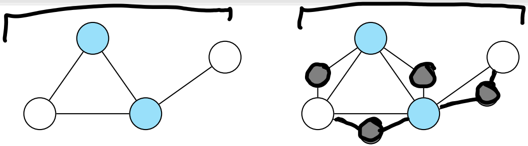
Beweis (Skizze)

$(G, k) \leadsto (G', k)$

1. setze initial $G' = G$
2. für jede Kante $e = \{u, v\}$ in G:

   erzeuge einen neuen (grauen) Knoten in G' und verbinde ihn mit u und v

Beispiel



$k = 2$ VERTEX COVER

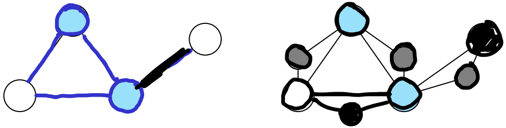
$k = 2$ DOMINATING SET

Dominating Set ist NP-vollständig

Theorem

Vertex Cover $\leq_m^p$ Dominating Set.

Beispiel



$k = 2$

Vertex Cover

$k = 2$

Dominating Set

Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

1. setze initial $G' = G$
2. für jede Kante $e = \{u, v\}$ in G:

   erzeuge einen neuen (grauen) Knoten in G' und verbinde ihn mit u und v

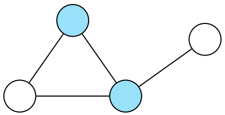
Korrektheit: „$\Rightarrow$“: VC-Lösung in G ist auch DS-Lösung in G'

Dominating Set ist NP-vollständig

Theorem

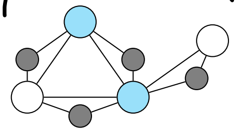
Vertex Cover $\leq_m^p$ Dominating Set.

Beispiel



$k = 2$

Vertex Cover



$k = 2$

Dominating Set

Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

1. setze initial $G' = G$
2. für jede Kante $e = \{u, v\}$ in G:

   erzeuge einen neuen (grauen) Knoten in G' und verbinde ihn mit u und v

Korrektheit: „$\Rightarrow$“: VC-Lösung in G ist auch DS-Lösung in G'

„$\Leftarrow$“: Sei $X \subseteq V(G')$ eine DS-Lösung für G' mit $|X| \leq k$

DOMINATING SET ist NP-vollständig

Theorem

VERTEX COVER $\leq_m^p$ DOMINATING SET.

Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

1. setze initial $G' = G$
2. für jede Kante $e = \{u, v\}$ in G:
   erzeuge einen neuen (grauen) Knoten in G' und verbinde ihn mit u und v

Korrektheit: „$\Rightarrow$“: VC-Lösung in G ist auch DS-Lösung in G'

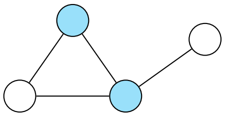
„$\Leftarrow$“: Sei $X \subseteq V(G')$ eine DS-Lösung für G' mit $|X| \leq k$

(a) neuer (grauer) Knoten u $\in$ DS-Lösung $\rightsquigarrow$ mit weißem Nachbarn v tauschen

$\rightsquigarrow$ Lösung ohne graue Knoten

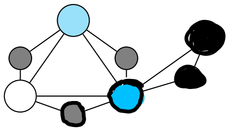
$(X \setminus \{u\}) \cup \{v\}$ ist DS-Lösung

Beispiel



$k = 2$

VERTEX COVER



$k = 2$

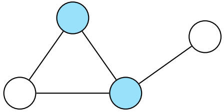
DOMINATING SET

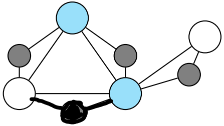
DOMINATING SET ist NP-vollständig

Theorem

VERTEX COVER $\leq_m^p$ DOMINATING SET.

Beispiel





Beweis (Skizze)

$(G, k) \rightsquigarrow (G', k)$

1. setze initial $G' = G$
2. für jede Kante $e = \{u, v\}$ in G:

   erzeuge einen neuen (grauen) Knoten in G' und verbinde ihn mit u und v

Korrektheit: „$\Rightarrow$“: VC-Lösung in G ist auch DS-Lösung in G'

„$\Leftarrow$“: Sei $X \subseteq V(G')$ eine DS-Lösung für G' mit $|X| \leq k$

(a) neuer (grauer) Knoten $\in$ DS-Lösung $\rightsquigarrow$ mit weißem Nachbarn tauschen

$\rightsquigarrow$ Lösung ohne graue Knoten

(b) graue Knoten dominiert $\rightsquigarrow$ jede Kante in G hat Endpunkt in X

$\rightsquigarrow$ X ist vertex cover in G

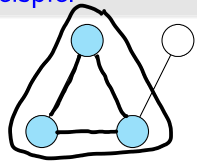
CLIQUE ist NP-vollständig

Clique

Eingabe: ungerichteter Graph G und Zahl $k \in \mathbb{N}$

Frage: Hat G einen vollständigen Teilgraph G' mit $\geq k$ Knoten?

Beispiel



$k = 3$

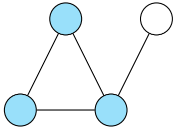
CLIQUE ist NP-vollständig

Clique

Eingabe: ungerichteter Graph G und Zahl $k \in \mathbb{N}$

Frage: Hat G einen vollständigen Teilgraph G' mit $\geq k$ Knoten?

Beispiel



$k = 3$

$\underline{VC} \leq_m^p \underline{IS}$

Theorem

$\underline{\text{INDEPENDENT SET}} \leq_m^p \underline{\text{CLIQUE}}$.

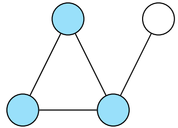
Clique ist NP-vollständig

Clique

Eingabe: ungerichteter Graph G und Zahl $k \in \mathbb{N}$

Frage: Hat G einen vollständigen Teilgraph G' mit $\geq k$ Knoten?

Beispiel



$k = 3$

Theorem

Independent Set $\leq_m^p$ Clique.

Beweis (Skizze)

$(G = (V, E), k) \rightsquigarrow (\overline{G} = (V, \binom{V}{2} \setminus E), k)$

ungeordnete Knotenpaare

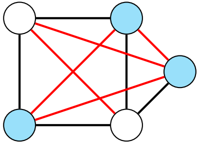
Korrektheit:

Jede unabhängige Knotenmenge in G bildet eine Clique in $\overline{G}$ und umgekehrt, also:

$(G, k) \in$ Independent Set $\Leftrightarrow (\overline{G}, k) \in$ Clique

keine Kante zwischen u & v in G $\Leftrightarrow$ Kante zwischen u & v in $\overline{G}$

Beispiel



$k = 3$

Independent Set
Clique

Wenn ein "Dominostein" fiele...

SAT $\leq_m^p$ 3-SAT $\leq_m^p$ VC $\leq_m^p$ IS $\leq_m^p$ CLIQUE $\leq_m^p$ SAT

$\overset{?}{\in} P$

Quelle: pixabay.com/en/mikado-domino-stones-pay-steinchen-1013878/

Netzwerk polynomieller Reduktionen I

SAT
CNF-SAT
3-SAT

Vertex Cover

Independent Set

Dominating Set

Clique

...

HITTING SET und SET COVER

Eingabe:

(1) Grundmenge („Universum“) $U := \{x_1, x_2, \ldots, x_n\}$,

(2) eine Teilmengenfamilie $\mathcal{F} := \{S_1, S_2, \ldots, S_m\}$ mit $S_i \subseteq U$ für $1 \leq i \leq m$ und

(3) ein $k \in \mathbb{N}$

Hitting Set und Set Cover

Eingabe:

(1) Grundmenge („Universum“) $U := \{x_1, x_2, \ldots, x_n\}$,

(2) eine Teilmengenfamilie $\mathcal{F} := \{S_1, S_2, \ldots, S_m\}$ mit $S_i \subseteq U$ für $1 \leq i \leq m$ und

(3) ein $k \in \mathbb{N}$

Hitting Set

Frage: Existiert eine Teilmenge $X \subseteq U$ mit $|X| \leq k$ und $X \cap S_i \neq \varnothing$ für jedes S_i?

Hitting Set und Set Cover

Eingabe:

(1) Grundmenge („Universum“) $U := \{x_1, x_2, \ldots, x_n\}$,

(2) eine Teilmengenfamilie $\mathcal{F} := \{S_1, S_2, \ldots, S_m\}$ mit $S_i \subseteq U$ für $1 \leq i \leq n$ und

(3) ein $k \in \mathbb{N}$

Hitting Set

Frage: Existiert eine Teilmenge $X \subseteq U$ mit $|X| \leq k$ und $X \cap S_i \neq \emptyset$ für jedes S_i?

Set Cover

Frage: Existiert ein $\mathcal{Z} \subseteq \mathcal{F}$ mit $|\mathcal{Z}| \leq k$ und $\bigcup_{S \in \mathcal{Z}} S = U$?

Hitting Set und Set Cover

Eingabe:

(1) Grundmenge („Universum") $U := \{x_1, x_2, \ldots, x_n\}$,
(2) eine Teilmengenfamilie $\mathcal{F} := \{S_1, S_2, \ldots, S_m\}$ mit $S_i \subseteq U$ für $1 \leq i \leq m$ und
(3) ein $k \in \mathbb{N}$

Hitting Set

Frage: Existiert eine Teilmenge $X \subseteq U$ mit $|X| \leq k$ und $X \cap S_i \neq \varnothing$ für jedes S_i?

Set Cover

Frage: Existiert ein $\mathcal{Z} \subseteq \mathcal{F}$ mit $|\mathcal{Z}| \leq k$ und $\bigcup_{S \in \mathcal{Z}} S = U$?

Beispiel

(1) $U = \{1, 2, 3, 4, 5, 6\}$,
(2) $S_1 = \{1, 3\}$, $S_2 = \{3, 4\}$, $S_3 = \{1, 5\}$, $S_4 = \{2, 4, 6\}$, $S_5 = \{1, 3, 5\}$
(3) $k = 2$

1, 2, 3, 4, 5, 6

$S_4 \cup S_5 = U$

Hitting Set und Set Cover

Eingabe:
(1) Grundmenge („Universum“) $U := \{x_1, x_2, \ldots, x_n\}$,
(2) eine Teilmengenfamilie $\mathcal{F} := \{S_1, S_2, \ldots, S_m\}$ mit $S_i \subseteq U$ für $1 \leq i \leq n$ und
(3) ein $k \in \mathbb{N}$

Hitting Set

Frage: Existiert eine Teilmenge $X \subseteq U$ mit $|X| \leq k$ und $X \cap S_i \neq \emptyset$ für jedes S_i?

Set Cover

Frage: Existiert ein $\mathcal{Z} \subseteq \mathcal{F}$ mit $|\mathcal{Z}| \leq k$ und $\bigcup_{S \in \mathcal{Z}} S = U$?

Beispiel

(1) $U = \{1, 2, 3, 4, 5, 6\}$,
(2) $S_1 = \{1, 3\}$, $S_2 = \{3, 4\}$, $S_3 = \{1, 5\}$, $S_4 = \{2, 4, 6\}$, $S_5 = \{1, 3, 5\}$
(3) $k = 2$

$\leadsto X = \{1, 4\}$, $\mathcal{Z} = \{S_4, S_5\}$.

Hitting Set ist NP-vollständig

Theorem

Vertex Cover $\leq_m^p$ Hitting Set.

Hitting Set ist NP-vollständig

Theorem

Vertex Cover $\leq_m^p$ Hitting Set.

Beweis (Skizze)

$(G = (V, E), k) \rightsquigarrow (U = V, \mathcal{F} = E, k)$

Korrektheit: klar

In der Tat ist Vertex Cover auch bekannt als „2-Hitting Set".

Hitting Set ist NP-vollständig

Theorem

Vertex Cover $\leq_m^p$ Hitting Set.

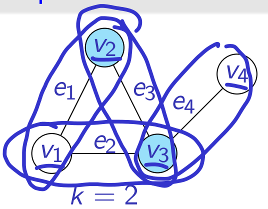
Beweis (Skizze)

$(G = (V, E), k) \rightsquigarrow (U = V, \mathcal{F} = E, k)$

Korrektheit: klar

In der Tat ist Vertex Cover auch bekannt als „2-Hitting Set“.

Beispiel



$U = \{v_1, v_2, v_3, v_4\}$
$\mathcal{F} = \{\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_3, v_4\}\}$

Set Cover ist NP-vollständig

Theorem

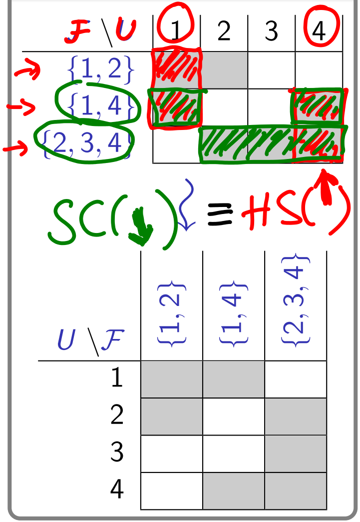
Hitting Set $\leq_m^p$ Set Cover.

Set Cover ist NP-vollständig

Theorem

Hitting Set $\leq_m^p$ Set Cover.

HS/SC Dualität



Set Cover ist NP-vollständig

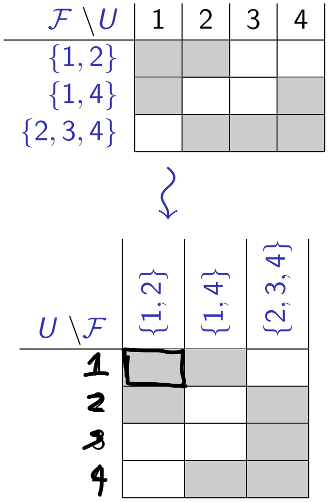
Theorem

Hitting Set $\leq^p_m$ Set Cover.

Beweis (Skizze)

$(U, \mathcal{F}, k) \rightsquigarrow (U_{SC} = \mathcal{F}, \mathcal{F}_{SC} = \{F_x \mid x \in U\}, k)$
mit $F_x := \{S_i \in \mathcal{F} \mid x \in S_i\}$

HS/SC Dualität



Set Cover ist NP-vollständig

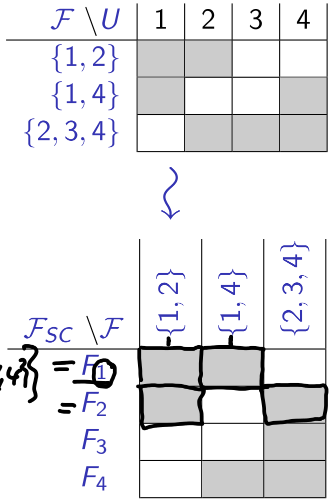
Theorem

Hitting Set $\leq_m^p$ Set Cover.

Beweis (Skizze)

$(U, \mathcal{F}, k) \rightsquigarrow (U_{SC} = \mathcal{F}, \mathcal{F}_{SC} = \{F_x \mid x \in U\}, k)$
mit $F_x := \{S_i \in \mathcal{F} \mid x \in S_i\}$

HS/SC Dualität



$\{\{1,2\},\{1,4\}\} = F_1$

$= F_2$

Set Cover ist NP-vollständig

Theorem

Hitting Set $\leq_m^p$ Set Cover.

Beweis (Skizze)

$(U, \mathcal{F}, k) \rightsquigarrow (U_{SC} = \mathcal{F}, \mathcal{F}_{SC} = \{F_x \mid x \in U\}, k)$
mit $F_x := \{S_i \in \mathcal{F} \mid x \in S_i\}$

Korrektheit:

df HS

$X \subseteq U$ ist ein Hitting Set für $\mathcal{F}$

$\Leftrightarrow \forall_{S_i \in \mathcal{F}} \exists_{x \in X}\ x \in S_i \iff \forall_{S_i \in S} \exists_{x \in X}\ S_i \in F_x$

$\Leftrightarrow \bigcup_{x \in X} F_x = \mathcal{F}$

$\Leftrightarrow \mathcal{Z} := \{F_x \mid x \in X\}$ ist ein Set Cover für $\mathcal{F} = U_{SC}$

df. SC

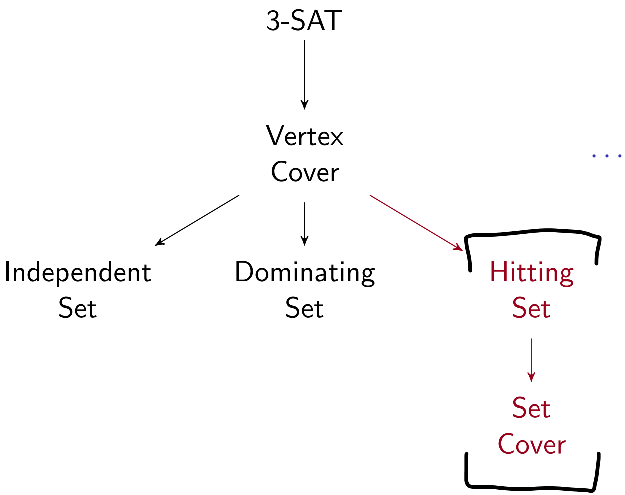
HS/SC Dualität

$\mathcal{F} \setminus U$	1	2	3	4
$\{1,2\}$	■	■		
$\{1,4\}$	■			■
$\{2,3,4\}$		■	■	■

poly Zeit

$\mathcal{F}_{SC} \setminus \mathcal{F}$	$\{1,2\}$	$\{1,4\}$	$\{2,3,4\}$
F_1	■	■	
F_2	■		■
F_3			■
F_4		■	■

Netzwerk polynomieller Reduktionen II



Subset Sum

Ein Problem u.a. aus dem Bereich „Scheduling“ (Ablaufsteuerung).

Subset Sum

Eingabe: Multi-Menge $U := \{u_1, u_2, \ldots, u_n\}$ von natürlichen Zahlen und eine Zahl $B \in \mathbb{N}$

Frage: Existiert eine Teilmenge $X \subseteq U$, die sich zu B summiert, d.h. $\sum_{u \in X} u = B$?

Beispiel

$U = \{4, 4, 11, 16, 21\}$ und $B = 29$.

Subset Sum

Ein Problem u.a. aus dem Bereich „Scheduling“ (Ablaufsteuerung).

Subset Sum

Eingabe: Multi-Menge $U := \{u_1, u_2, \ldots, u_n\}$ von natürlichen Zahlen und eine Zahl $B \in \mathbb{N}$
Frage: Existiert eine Teilmenge $X \subseteq U$, die sich zu B summiert, d.h. $\sum_{u \in X} u = B$?

Beispiel

$U = \{4, 4, 11, 16, 21\}$ und $B = 29$.
$\rightsquigarrow X = \{4, 4, 21\}$.

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

SUBSET SUM ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ SUBSET SUM.

Beweis (Skizze)

Konstruktion: Variablen $\underline{x_1, \ldots, x_n}$, Klauseln $\underline{c_1, \ldots, c_m}$

Beispiel

$$c_1 : x_1 \lor x_2 \lor \overline{x_3}$$
$$c_2 : \overline{x_1} \lor x_2 \lor x_3$$
$$c_3 : \overline{x_1} \lor \overline{x_2} \lor \overline{x_3}$$

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \ldots, x_n$, Klauseln $c_1, \ldots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.

Beispiel

$c_1 : \cancel{x_1} \vee x_2 \vee \overline{x_3}$

$c_2 : \cancel{\overline{x_1}} \vee x_2 \vee x_3$

$c_3 : \cancel{\overline{x_1}} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1

Subset Sum ist NP-vollständig

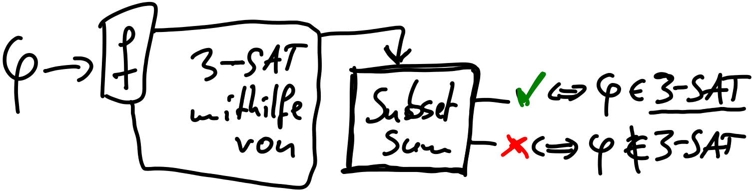
Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \ldots, x_n$, Klauseln $c_1, \ldots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.



Beispiel

$c_1 : x_1 \vee x_2 \vee \overline{x_3}$

$c_2 : \overline{x_1} \vee x_2 \vee x_3$

$c_3 : \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
B:	1	1	1	1	1	1

x_2 x_3

2

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \ldots, x_n$, Klauseln $c_1, \ldots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.
2. Für jede Klausel c_j, bilde zwei **dezimale** „Füllzahlen“ g_j, h_j

Beispiel

$c_1: x_1 \vee x_2 \vee \overline{x_3}$

$c_2: \overline{x_1} \vee x_2 \vee x_3$

$c_3: \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
g_1:	0	0	0	1	0	0
h_1:	0	0	0	1	0	0

B

3

$\textsc{Subset Sum}$ ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ $\textsc{Subset Sum}$.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \ldots, x_n$, Klauseln $c_1, \ldots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.
2. Für jede Klausel c_j, bilde zwei **dezimale** „Füllzahlen“ g_j, h_j

Beispiel

$c_1: x_1 \vee x_2 \vee \overline{x_3}$

$c_2: \overline{x_1} \vee x_2 \vee x_3$

$c_3: \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
g_1:	0	0	0	1	0	0
h_1:	0	0	0	1	0	0
g_2:	0	0	0	0	1	0
h_2:	0	0	0	0	1	0
g_3:	0	0	0	0	0	1
h_3:	0	0	0	0	0	1

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \ldots, x_n$, Klauseln $c_1, \ldots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.
2. Für jede Klausel c_j, bilde zwei **dezimale** „Füllzahlen“ g_j, h_j
3. Setze Dezimalzahl $B := \underbrace{1\ldots1}_{n}\underbrace{3\ldots3}_{m}$.

Beispiel

$c_1 : x_1 \vee x_2 \vee \overline{x_3}$

$c_2 : \overline{x_1} \vee x_2 \vee x_3$

$c_3 : \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
g_1:	0	0	0	1	0	0
h_1:	0	0	0	1	0	0
g_2:	0	0	0	0	1	0
h_2:	0	0	0	0	1	0
g_3:	0	0	0	0	0	1
h_3:	0	0	0	0	0	1
B:	1	1	1	3	3	3

1 0

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \dots, x_n$, Klauseln $c_1, \dots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.
2. Für jede Klausel c_j, bilde zwei **dezimale** „Füllzahlen“ g_j, h_j
3. Setze Dezimalzahl $B := \underbrace{1\dots1}_{n}\underbrace{3\dots3}_{m}$.

Korrektheit „$\Rightarrow$“: Sei β eine erfüllende Belegung.

$\rightsquigarrow$ Lösung $= \{y_i \mid \beta(x_i) = 1\} \cup \{z_i \mid \beta(x_i) = 0\}$ + geeignete g_i & h_i

Beispiel

$c_1 : x_1 \vee x_2 \vee \overline{x_3}$

$c_2 : \overline{x_1} \vee x_2 \vee x_3$

$c_3 : \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	c_1	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
g_1:	0	0	0	1	0	0
h_1:	0	0	0	1	0	0
g_2:	0	0	0	0	1	0
h_2:	0	0	0	0	1	0
g_3:	0	0	0	0	0	1
h_3:	0	0	0	0	0	1
B:	1	1	1	3	3	3
	1	1	1	3	3	3

Subset Sum ist NP-vollständig

Theorem

3-SAT $\leq_m^p$ Subset Sum.

Beweis (Skizze)

Konstruktion: Variablen $x_1, \dots, x_n$, Klauseln $c_1, \dots, c_m$

1. Für jedes x_i bilde zwei Dezimalzahlen $y_i, z_i \in \{0,1\}^{n+m}$ mit:
   Vordere n Ziffern: i-te Stelle von y_i und z_i ist 1, alle anderen sind 0.
   Hintere m Ziffern: j-te Stelle von y_i ist 1 falls $x_i \in c_j$, und sonst 0.
   j-te Stelle von z_i ist 1 falls $\overline{x_i} \in c_j$, und sonst 0.
2. Für jede Klausel c_j, bilde zwei **dezimale** „Füllzahlen“ g_j, h_j
3. Setze Dezimalzahl $B := \underbrace{1 \dots 1}_{n}\underbrace{3 \dots 3}_{m}$.

Korrektheit „$\Rightarrow$“: Sei β eine erfüllende Belegung.
$\rightsquigarrow$ Lösung $= \{y_i \mid \beta(x_i) = 1\} \cup \{z_i \mid \beta(x_i) = 0\}$ + geeignete g_i & h_i
„$\Leftarrow$“: Sei X eine Menge von Zahlen mit $\sum_{u \in X} u = B$.
erste n Ziffern $\rightsquigarrow$ $y_i \in X \Leftrightarrow z_i \notin X$
Die Belegung β mit $\beta(x_i) = 1$ falls $y_i \in X$, und $\beta(x_i) = 0$ sonst, ist erfüllend.

Beispiel

$c_1 : x_1 \vee x_2 \vee \overline{x_3}$
$c_2 : \overline{x_1} \vee x_2 \vee x_3$
$c_3 : \overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$

	x_1	x_2	x_3	$\boxed{c_1}$	c_2	c_3
y_1:	1	0	0	1	0	0
z_1:	1	0	0	0	1	1
y_2:	0	1	0	1	1	0
z_2:	0	1	0	0	0	1
y_3:	0	0	1	0	1	0
z_3:	0	0	1	1	0	1
g_1:	0	0	0	1	0	0
h_1:	0	0	0	1	0	0
g_2:	0	0	0	0	1	0
h_2:	0	0	0	0	1	0
g_3:	0	0	0	0	0	1
h_3:	0	0	0	0	0	1
B:	1	1	1	3	3	3
				2	2	2
				1	1	1

Netzwerk polynomieller Reduktionen III

$$
\begin{tikzcd}
 & \text{3-SAT} \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{Subset Sum} & \text{Vertex Cover} \arrow[dl] \arrow[d] \arrow[dr] & \dots \\
\text{Independent Set} & \text{Dominating Set} & \text{Hitting Set} \arrow[d] \\
 & & \text{Set Cover}
\end{tikzcd}
$$

in Praxis:

1. Heuristiken
2. spezialisierte „solver" (ILP, SAT, ...)
3. Approximation
4. Parameterisierte Algorithmen

VC lösen in $2^k \cdot poly(n)$