Weiteres Beispiel: Boole‘sche Erfüllbarkeit

- Eine Boole’sche Variable x kann Werte 0 und 1 (**`falsch`** und **`wahr`**) annehmen.
- Eine Boole’sche Formel Φ ist eine Verknüpfung von Boole’schen Variablen durch Boole’sche Operatoren, z.B. **AND** ($\wedge$) , **OR** ($\vee$), **NOT** ($\neg$).

Beispiel: $\Phi = (\neg x \vee y) \wedge (x \vee \neg z)$

ist eine Boole’sche Formel mit Variablen x, y, z.

- Φ ist erfüllbar, falls es eine Belegung der Variablen mit Werten 0, 1 gibt, die Φ **`wahr`** macht.

Beispiel: Φ ist erfüllbar, z. b. durch x:=1, y:=1, z:=0.

Anwendung: Optimierung von Logik-Schaltkreisen

Konjunktive Normalform (KNF)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Literal: Variable oder negierte Variable: y_i, $\neg y_i$
- Klausel: Disjunktion („**or**“) von Literalen:

  $C = x_1 \vee \ \dots \vee x_m$, x_i Literale
- Formel Φ in Konjunktiver Normalform (KNF): Konjunktion („**and**“) von Klauseln:

  $\Phi = C_1 \wedge \dots \wedge C_\ell$, C_i Klauseln
- **Value**:= $\{ \langle \Phi, \underline{x} \rangle : \Phi$ wird **wahr** bei Belegung $\underline{x} \}$
- **SAT** := $\{ \langle \Phi \rangle : \Phi$ erfüllbare Formel in KNF $\}$

Beispiel: $\langle\, (\neg x \vee y) \wedge (x \vee \neg z\,)\,,\, (1,1,0)\,\rangle \quad \in$ **Value**

$\langle\, (\neg x \vee y) \wedge (x \vee \neg z\,)\,,\, (0,0,1)\,\rangle \quad \notin$ **Value**

$\langle\, (\neg x \vee y) \wedge (x \vee \neg z\,)\,\rangle \quad \in$ **SAT**

$\langle\, (\neg x \vee y) \wedge (x \vee \neg z\,) \wedge (z \vee \neg y\,) \wedge \mathrm{x} \wedge (\neg y)\,\rangle \quad \notin$ **SAT**

Erfüllbarkeit (Satisfiability)

Konjunktive Normalform (KNF) =
Konjunktion ($\wedge$) von Disjunktion ($\vee$) von Literalen (x,$\neg x$)
Value = { $\langle\Phi,\underline{x}\rangle$: Formel Φ wird **wahr** bei Belegung $\underline{x}$ }
SAT = { $\langle\Phi\rangle$: KNF-Formel Φ ist erfüllbar }

- *k*-KNF Formel: Formel in KNF, in der jede Klausel aus $\leq k$ Literalen besteht.
- ***k*-SAT** := { $\langle\Phi\rangle$: Φ erfüllbare Formel in *k*-KNF }

Übung: 2-**SAT** $\in \mathcal{P}$

Fragen: SAT $\in \mathcal{P}$?   3-**SAT** $\in \mathcal{P}$?

Zweites Beispiel für Polynomielle Reduktion

a) $\mathbf{3SAT}$ ist polynomiell auf $\mathbf{SAT}$ reduzierbar:

$$\mathbf{3SAT} \leqslant_p \mathbf{SAT}$$

b) $\mathbf{SAT}$ ist polynomiell auf $\mathbf{3SAT}$ reduzierbar:

$$\mathbf{SAT} \leqslant_p \mathbf{3SAT}$$

Folge: Beide Probleme sind im Wesentlichen '*gleich schwer*'; ein polynomialzeit-Algorithmus für das eine würde einen ebensolchen für das andere liefern:

$$\mathbf{SAT} \in \mathcal{P} \;\Leftrightarrow\; \mathbf{3SAT} \in \mathcal{P}.$$

Die gleiche Situation wie bei $\mathbf{IS}$ und $\mathbf{CLIQUE}$…

a) 3SAT $\leqslant_p$ SAT   b) SAT $\leqslant_p$ 3SAT

Und was ist f(Eingaben, die keine 3-KNF Formel darstellen) ?

Beweis: *Was ist zu tun*?

a) Beschreibe eine in polynom[ieller Zeit berechenbare]
Funktion f, die aus einer 3-KNF Form[el eine] KNF-Formel Φ' macht so dass gilt:

Φ ist genau dann erfüllbar, wenn Φ' es ist. √

b) Gegeben KNF-Formel $\Phi = (a \vee b \vee c \vee d \vee e) \wedge \Phi_1$ mit Literalen *a,b,c,d,e.*

Betrachte neue Variablen x,y und Formel
$\Phi' = (a \vee b \vee x) \wedge (\neg x \vee c \vee y) \wedge (\neg y \vee d \vee e) \wedge \Phi'_1$

Klausel mit k Lit. $\rightarrow$ k-2 Klauseln a 3 Lit. & k-3 neue Var.

Φ' ist 3-KNF; und erfüllbar (d.h. in 3**SAT**)
genau dann, wenn Φ erfüllbar (d.h. in **SAT**) ist!

Reduktion IS $\leqslant_p$ SAT

Zu tun: Bei Eingabe von Graph G und $k \in \mathbb{N}$,
berechne in polynomieller Zeit eine KNF Formel Φ, so dass
Φ erfüllbar ist g.d.w. es k unabhängige Knoten in G gibt.

G habe Knoten $V=\{1,..,n\}$ und Kanten E.
Betrachte Bool'sche Variablen $x_{v,i}$, $v \in V$, $i=1\ldots k$

Knoten v ist i-ter der k unabhängigen.

dazu Klauseln $K_i := \bigvee_{v \in V} x_{v,i}$, $i=1\ldots k$

Es gibt einen i-ten unabhängigen.

und $\neg x_{v,i} \vee \neg x_{v,j}$, $v \in V$, $1 \leq i < j \leq k$

Knoten v kann nicht i-ter <u>und</u> j-ter sein.

und $\neg x_{u,i} \vee \neg x_{v,j}$, $\{u,v\} \in E$, $1 \leq i < j \leq k$

2 verbundene Knoten sind nicht unabhängig.

Größe von Φ: $O(k \cdot n + n \cdot k^2 + n^2 k^2) = O(n^2 k^2)$

Berechnungsaufwand von $(G,k) \rightarrow \Phi$: polynomiell in $n + \log k$
da $k \leq n$.