# Übung 6

1) 3SAT ist NP-vollständig bzgl. LogSpace-many-one-Reduktionen

2) 3-Colorability ist NP-vollst. bzgl. LogSpace-m.-o.-R.

## 3SAT

Gegeben: $F$ in KNF mit höchstens 3 Literalen pro Klausel

Frage: Gibt es eine Belegung $\varphi: Vars(F) \to \{0,1\}$ mit $F[X/\varphi(X)] \equiv 1$

$3SAT \in NP$: Ein Algorithmus, der nichtdeterministisch $\varphi$ rät und anschließend prüft, ob die Formel davon erfüllt wird.

$SAT \leq_m^{log} 3SAT$: Gesucht ist $f$: KNF $\to$ 3KNF mit $F \in SAT \Leftrightarrow f(F) \in 3SAT$.

Idee: $A \Leftrightarrow (B \vee C) \equiv (A \to (B \vee C)) \wedge ((B \vee C) \to A)$

$\equiv (\neg A \vee B \vee C) \wedge (\neg B \vee A) \wedge (\neg C \vee A)$

Beispiel $C = L_1 \vee L_2 \vee L_3 \vee L_4 \equiv_{SAT} (L_1 \vee L_2 \vee K) \wedge (\neg K \vee L_3 \vee L_4) \wedge (\neg L_3 \vee K) \wedge (\neg L_4 \vee K)$

Für jede Klausel $C = \bigvee_{j=1}^{n} L_j$ , falls $n \leq 3$, tue nichts. $C' := C$

sonst erzeuge neue Variablen $K_1 \dots K_{n-3}$ und die Formel

$$C \equiv_{SAT} (L_1 \vee L_2 \vee K_1) \wedge \bigwedge_{j=1}^{n-4} (K_j \Leftrightarrow L_{j+2} \vee K_{j+1}) \wedge (K_{n-3} \Leftrightarrow L_{n-1} \vee L_n)$$

$$\equiv (L_1 \vee L_2 \vee K_1) \wedge \bigwedge_{j=1}^{n-4} \left( (\neg K_j \vee L_{j+2} \vee K_{j+1}) \wedge (\neg L_{j+2} \vee K_j) \wedge (\neg K_{j+1} \vee K_j) \right) \cancel{\vee (\neg K_{n-3}}$$

$$\wedge (\neg K_{n-3} \vee L_{n-1} \vee L_n) \wedge (\neg L_{n-1} \vee K_{n-3}) \wedge (\neg L_n \vee K_{n-3}) =: C'$$

Jede Klausel erzeugt somit höchstens linear viele Klauseln, deren Konjunktion erfüllbarkeits-äquivalent sind. Wird diese Methode auf alle Klauseln von $F$ angewendet, erhält man eine erf.-äquiv. Formel in 3KNF.

$f$ sei nun eine Funktion, die $F = \bigwedge_{i=1}^{m} C_i$ nach $f(F) = \bigwedge_{i=1}^{m} C_i'$ überführt.

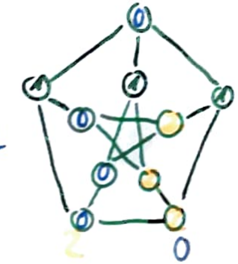
$f$ ist LogSpace-berechenbar. und $F \in SAT \Leftrightarrow f(F) \in 3SAT$. $\square$

## 2) 3-Färbbarkeit (3-Colorability)

Gegeben: Ein ungerichteter Graph $\langle V, \leftrightarrow \rangle$

Frage: Gibt es eine Färbung $c: V \to \{0,1,2\}$, sodass für alle $v \leftrightarrow w$ gilt $c(v) \neq c(w)$

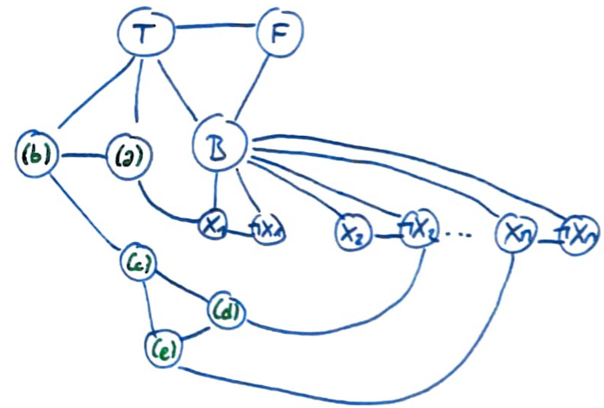
3-Colorability $\in NP$: Ein Algorithmus, der c rät und anschließend testen, ob c eine gültige 3-Färbung des Graphen ist, benötigt polynomiell (vielleicht quadratisch) viel Zeit.



3SAT $\le_m^{log}$ 3-Colorability: Gesucht ist $f$: 3KNF $\rightarrow$ U.G. mit $F \in$ 3SAT $\Leftrightarrow f(F) \in$ 3C..

Idee: Wir erzeugen eine Palette

Dazu Inputs für alle Variablen:




Und „Gatter" für alle Klauseln:

z.B. $(x_1 \vee \neg x_2 \vee x_n)$

Diese Konstruktion erzwingt, dass jede 3-Färbung für eine Belegung für F steht. Falls diese Belegung eine der Klauseln nicht erfüllt, dann kann der Graph innerhalb des „Gatters" dieser Klausel nicht 3-gefärbt werden. Sobald mindestens eines der Literale „true" ist. (3 Fälle zu unterscheiden) dann ist das Gatter 3-färbbar.

Falls F erfüllbar, dann gibt es eine erfüllende Belegung $\varphi$.
Diese diktiert partiell eine 3-Färbung $c_\varphi$ für $F(f)$. ObdA gewähle
$c_\varphi(T)=1$, $c_\varphi(F)=0$, $c_\varphi(B)=2$, $c_\varphi(X_i)=\varphi(X_i)$, $c_\varphi(\neg X_i)=1-\varphi(X_i)$ $\forall i \in \{1 \ldots n\}$
Jedes „Gatter" ist mit mind. einem Literal mit $c_\varphi(L)=1$ verbunden und damit wie oben erklärt 3-Färbbar. Erweitere $c_\varphi$ entsprechend. Sobald dies für alle „Gatter" getan wurde, ist $c_\varphi$ eine gültige 3-Färbung für $f(F) \Rightarrow f(F) \in$ 3-Color...

Falls $f(F) \in$ 3Colorability, dann gibt es eine 3-Färbung $c$. Für alle Variablen $X_i$ gilt $c(X_i) \neq c(B)$. setze $\varphi_c(X_i) = \begin{cases} 1 & \text{falls } c(X_i)=c(T) \\ 0 & \text{falls } c(X_i)=c(F) \end{cases}$
Diese Belegung erfüllt jede Klausel, da sonst das „Gatter" nicht 3-färbbar gewesen wäre. Also erfüllt sie F $\Rightarrow$ $F \in$ 3SAT.

Die ~~die~~ Funktion f, die entsprechen der obigen Konstruktion F auf einen solchen ungerichteten Graphen abbildet, ist LogSpace-berechenbar.

$\square$