

## Aufgabe 1

## Aufgabe 2

(a)

Sei  $\varphi$  eine Formel in 3-KNF mit Literalen  $L$  und  $k$  Klauseln und habe die Form

$$\varphi = (x_{11} \vee x_{12} \vee x_{13}) \wedge (x_{21} \vee x_{22} \vee x_{23}) \dots$$

Dabei sind  $x_{ij} \in \{l, \neg l \mid l \in L\}$ .

Sei nun  $G(\varphi) = (V, E, K = 2k + |L|)$  ein Graph mit

$$V = \{l, \neg l \mid L \in L\} \cup \{x_{ij}\}$$

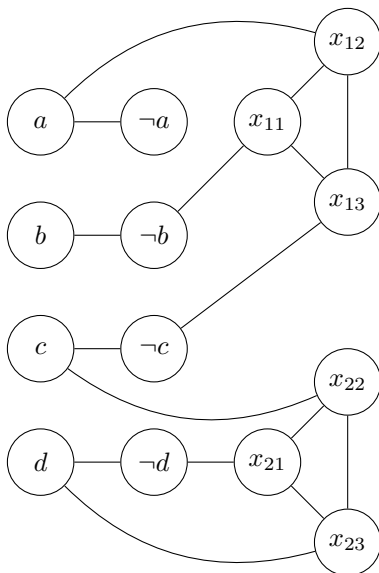
– wobei die Knoten aus  $L$  *Literalknoten* und die Knoten  $x_{ij}$  *Klauselknoten* heißen – und

$$E = \{(x_{i1}, x_{i2}), (x_{i1}, x_{i3}), (x_{i2}, x_{i3}) \mid i = 1 \dots k\} \cup \{(L(x_{ij}), x_{ij})\}$$

wobei  $L(x_{ij})$  das Literal (oder die Negation des Literals) von  $x_{ij}$  darstellt. Die  $i$  Dreiecke aus Klauselknoten nennen wir *Cluster*.

Auf  $G(\varphi)$  muss ein VertexCover der Größe  $K$  gefunden werden.

Für die Formel  $\varphi = (\neg b \vee a \vee \neg c) \wedge (\neg d \vee c \vee d)$  ergibt sich also der Graph



Behauptung: **VertexCover** hat auf  $G(\varphi)$  genau dann eine Lösung, wenn  $\varphi$  erfüllbar ist (3-SAT hat eine Lösung).

Beweis:

$\Leftarrow$ :

Wir konstruieren ein Vertex Cover  $W \subseteq V$  der Größe  $K$ . Seien  $L_1$  die Literale, denen in der Lösung von 3-Sat der Wahrheitswert wahr zugewiesen wird. Dann sind alle  $l \in L_1$  auch in  $W$ . Wähle nun aus jeder 3-Clique, die eine Klausel beschreibt, zwei beliebige Knoten, so dass der verbleibende mit einem bereits markierten Literal-Knoten verbunden ist. So ein Knoten muss existieren, da mindestens ein Literal (oder eine Negation eines Literals) in einer Klausel wahr sein muss, ein Knoten also mit einem bereits markierten Knoten verbunden sein muss.

$\Rightarrow$ :

Beobachtung: Jedes Vertex Cover in  $G(\varphi)$  muss mindestens einen Literalknoten jedes Literals sowie mindestens zwei Klauselknoten enthalten. Ein Vertex Cover der Größe  $K = 2k + |L|$  enthält somit genau einen Literalknoten jedes Literals sowie genau zwei Klauselknoten jedes Dreiecks. Seien nun  $L_1$  die Literalknoten des Vertex Covers und wie in  $\Leftarrow$  die Menge der Literale mit dem Wahrheitswert wahr. Die entsprechende Formel  $\varphi$  ist dann erfüllbar, da jedem Literal ein Wahrheitswert zugewiesen ist und jede Klausel erfüllbar ist: Jede Klausel enthält genau einen Knoten, der nicht im Vertex Cover enthalten ist, dessen Kante mit einem Literalknoten aber abgedeckt ist. Der inzidente Literalknoten muss daher im Vertex Cover enthalten sein, das zum Klauselknoten gehörende Literal also wahr sein.  $\square$

(b)

Behauptung:  $C \subseteq V$  ist genau dann ein **VertexCover**, wenn  $I = V \setminus C$  ein **IndependentSet** ist.

Aus der Behauptung folgt sofort die Äquivalenz der Problem hinsichtlich NP-Vollständigkeit.

Beweis:

$\Rightarrow$ :

Ist  $C \subseteq V$  ein Vertex Cover, so sind alle Kanten des Graphen zu mindestens einem Knoten in  $C$  adjazent. Somit gibt es keine Kante, die zu zwei Knoten aus  $I = V \setminus C$  adjazent ist. Also gibt es keine zwei inzidenten Knoten in  $I$ .

$\Leftarrow$ :

Ist  $I \subseteq V$  ein Independent Set, so existiert keine Kante im Graphen zwischen zwei Knoten in  $I$ . Jede Kante ist somit zu mindestens einem Knoten in  $C = V \setminus I$  adjazent. Also deckt  $C$  alle Kanten ab.  $\square$

### Aufgabe 3

Betrachte den Lauf für eine beliebige Eingabe der Länge  $n$ . Sei  $\alpha_i$  mit  $i \in \mathbb{Z}$  das Zeichen an der  $i$ ten Bandposition. Sei  $\delta_k$  die  $k$ te Transitionsregel. Der Unterstrich markiert die aktuelle Bandposition.

Die TM startet in  $q_0$  auf einer beliebigen Eingabe

$q_0 \_ \$INPUT\$ \_$

Solange das aktuelle Zeichen kein Blank-Symbol ist, geht die TM nach rechts, ohne etwas zu verändern ( $\delta_1, \delta_2$ ).

Hat die TM das Ende des Wortes erreicht, so erscheint das erste Blanksymbol:

$\_ \$INPUT\$ q_0 \_$

Die TM wechselt nun nach  $q_1$ , schreibt ein  $\#$  und geht nach rechts:

$\_ \$INPUT\$ \# q_1 \_$

Ein Blank: Sie wechselt nach  $q_2$ , schreibt eine 1 und geht nach rechts:

$\_ \$INPUT\$ \# 1 q_2 \_$

Ein Blank: Sie bleibt in  $q_2$  und schreibt eine 0:

$\_ \$INPUT\$ \# 1 q_2 0$

Eine 0: Sie wechselt nach  $q_3$  und geht nach rechts:

$\_ \$INPUT\$ \# 1 0 q_3 \_$

Ein Blank: Sie schreibt eine 1, wechselt nach  $q_4$  und geht nach rechts:

$\_ \$INPUT\$ \# 1 0 1 q_4 \_$

Ein Blank: Sie wechselt nach  $q_1$ , schreibt eine 0 und geht nach links:

$\_ \$INPUT\$ \# 1 0 q_1 1 0 \_$

Eine 1: Sie geht nach rechts:

$\_ \$INPUT\$ \# 1 0 1 q_1 0 \_$

Eine 0: Sie geht nach rechts und wechselt nach  $q_5$ :

$\_ \$INPUT\$ \# 1 0 1 0 q_5 \_$

Ein Blank: Sie schreibt eine 1:

$\_ \$INPUT\$ \# 1 0 1 0 q_5 1$

Eine 1: Sie wechselt nach  $q_6$  und geht nach rechts:

$\sqcup \$ \text{INPUT} \$ \# 10101 q_6 \sqcup$

Ein Blank: Sie wechselt nach  $q_3$  und schreibt eine 0:

$\sqcup \$ \text{INPUT} \$ \# 10101 q_3 0$

Eine 0: Sie wechselt nach  $q_F$  und geht nach rechts:

$\sqcup \$ \text{INPUT} \$ \# 101010 q_F \sqcup$

Die Endkonfiguration der TM ist also für jedes Eingabewort die folgende: Sie endet im Zustand  $q_F$ , an die Eingabe wurden die Zeichen  $\#101010$  angehängt und der Lesekopf befindet sich beim ersten Blank hinter diesen Zeichen.

## Aufgabe 4