NP-Vollständigkeit
Proseminar Theoretische Informatik
(WS2010/11)

Maurice Wothe

Vortrag vom 7.12.2010

1 3-SAT ist NP-vollständig

Wir zeigen, dass SAT auf 3-SAT reduzierbar ist. Sei $C = (c_1, \ldots, c_n)$ eine Eingabe für SAT, wobei c_i eine Klausel über $x_1, \overline{x_1}, \ldots, x_n, \overline{x_n}$ darstelle. Nun betrachten wir die einzelnen Klauseln. Besteht c_i nur aus einem Literal z, so ersetzen wir die Klausel durch $z \vee z \vee z$. Besitzt sie die Form $z \vee z'$, so ersetzen wir sie durch $z \vee z \vee z'$. Besitzt sie schon 3 Literale, übernehmen wir sie ohne Veränderung. Sollte sie nun die Form $c = z_1 \vee \cdots \vee z_k$ mit $k \geq 4$ besitzen, so erreichen wir c durch $k-2$ Klauseln und benutzen $k-3$ neue Variablen. Die Ersetzung hat die folgende Form:

$z_1 \vee z_2 \vee yc_1$
$\overline{y_{c,l}} \vee z_{l+2} \vee y_{c,l+1} \quad$ für $1 \leq l \leq k-4$
$\overline{y_{c,k-3}} \vee z_{k-1} \vee z_k$

Diese Konstruktion kann offensichtlich in polynomieller Zeit erzeugt werden. Um die Funktionsweise zu veranschaulichen.

Hier die Klauseln für $k = 7$

$z_1 \vee z_2 \vee y_{c,1}$
$\overline{y_{c,1}} \vee z_3 \vee y_{c,2}$
$\overline{y_{c,2}} \vee z_4 \vee y_{c,3}$
$\overline{y_{c,3}} \vee z_5 \vee y_{c,4}$
$\overline{y_{c,4}} \vee z_6 \vee z_7$

Wir müssen nun Zeigen, dass die Konstruierte Klauselmenge genau dann erfüllbar ist, wenn eine der z-Variablen dem Wert 1 erhällt. Falls z_1 oder z_2 den Wert 1 haben setzen wir alle $y_{c,j} = 0$. Falls $z_6 = 1$ oder $z_7 = 1$ setzen wir alle $y_{c,j} = 1$. Falls $z_i = 1 \quad (3 \leq i \leq 5)$, setzen wir alle $y_{c,j} = \cdots = y_{c,i-2} = 1$ und alle $y_{c,i-1} = \cdots = y_{c,4} = 0$. Wenn alle $z_i = 0$ sind können durch $y_{c,j} = 1$ nur die ersten 4 Klauseln erfüllbar werden. Die letzte Klausel bleibt unerfüllbar. Bei $k \neq 7$ erfolgt die Argumentation analog.

Falls also ein $a \in \{0,1\}^n$ existiert, sodass für $x_i = a_i$ alle Klauseln C erfüllt sind, können auch alle Klauseln an der zugeordneten Eingabe für 3-SAT erfüllt werden. Falls aber für alle $a \in \{0,1\}^n$ gilt, dass für $x_i = a_i$ mindestens eine Klausel unerfüllbar ist, können auch die Klauseln in der zugeorneten Eingabe für 3-SAT nicht gleichzeitig erfüllt werden.

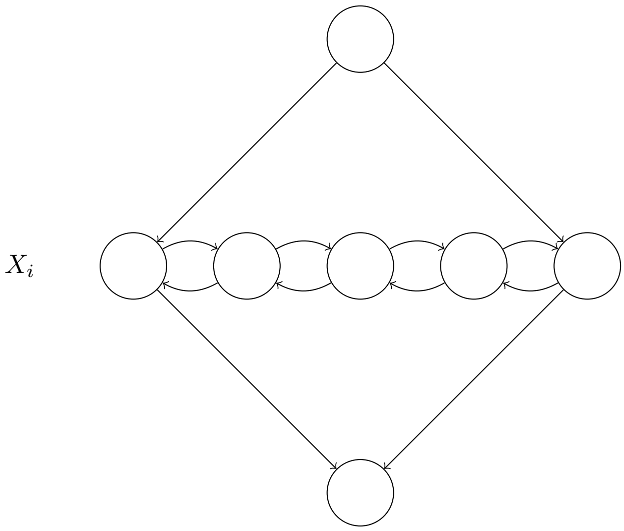
2 Hamillton-Weg

Da wir schon bewiesen haben, dass Hamilton-weg $\in$ NP müssen wir nur noch 3SAT $\leq p$ Hamilton-Weg zeigen. Wir konstruieren aus einer 3CNF Φ einen Grafen G mit den Knoten s und t, der genau dann einen Hamilton-Weg besitzt, wenn Φ erfüllbar ist.

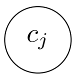
$$\Phi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \cdots \wedge (a_k \vee b_k \vee c_k)$$

Die a, b, c sind Literale der Form x_i oder $\overline{x_i}$. Dabei seien $x_1, \ldots x_c$ die l-Variablen von Φ.

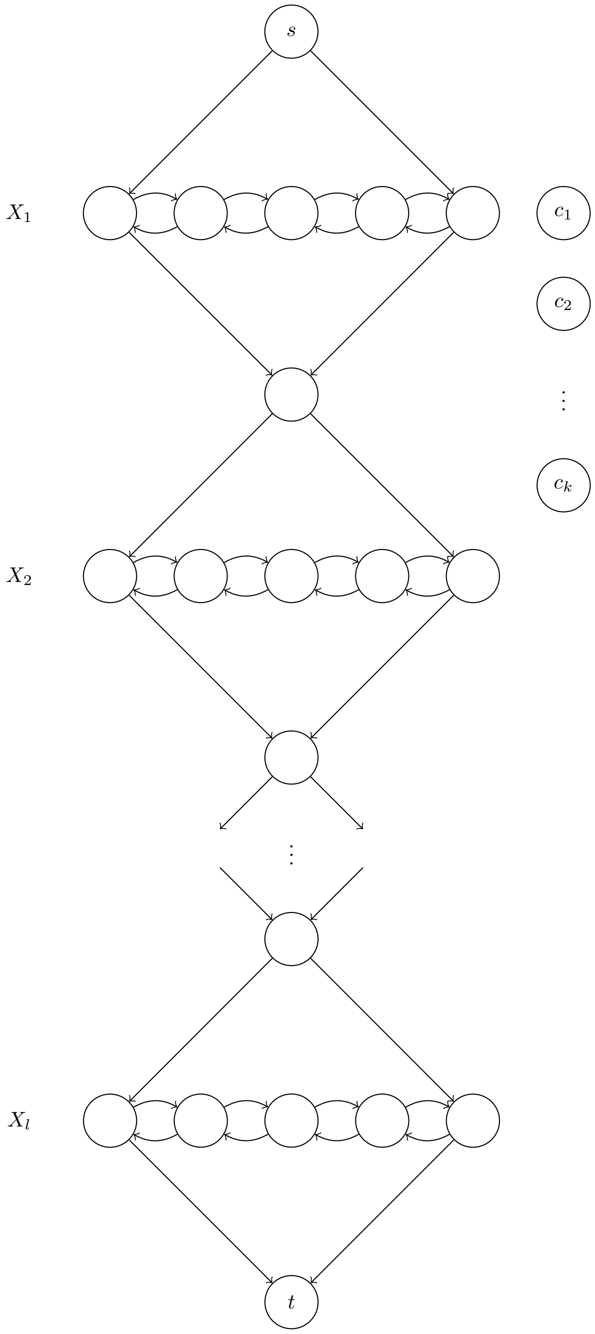
Jede Variable x_i wird durch ein diamantförmigen Graphen mit einer in der Mitte befindlichen Reihe aus in beiden Richtungen verbundenen Knoten repräsentiert. Die Anzahl der Knoten in der mittleren Reihe spezifizieren wir später.



Jede Klausel von Φ wird durch einen einzelnen Knoten repräsentiert.

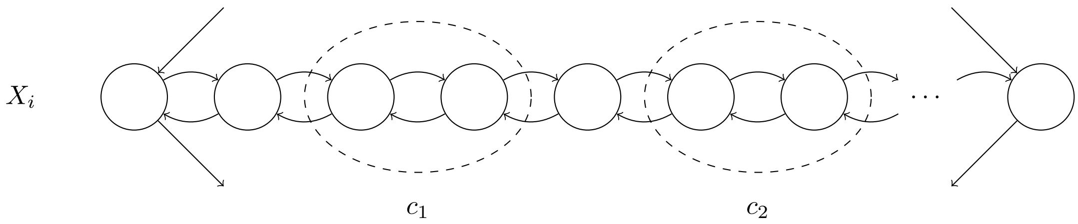


G sieht ohne die Berücksichtigung der Verhältnisse der Variablen zu den sie beinhaltenden Klauseln wie folgt aus:

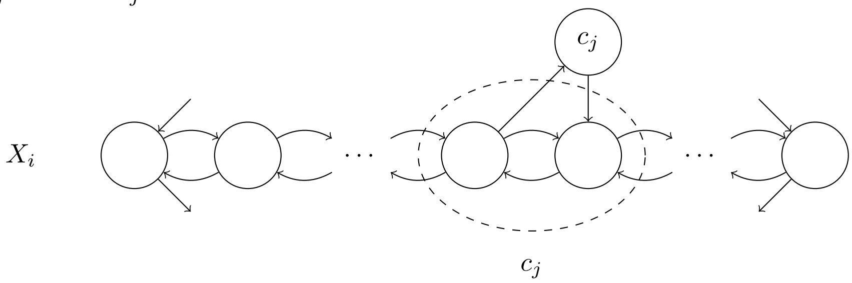


Nun ergänzen wir G um die Kanten, die die Diamantenstruktur mit den Knoten, die die Klauseln repräsentieren verbinden. Hierfür müssen wir zuerst die Mittelreihen der Diamanten genauer spezifizieren.

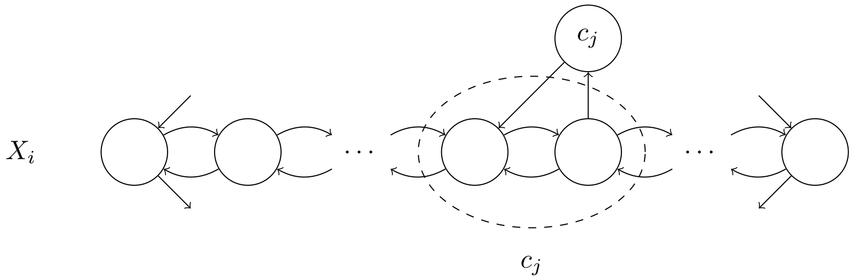
Jede Reihe enthällt $3k + 1$ Knoten, zusätzlich zu den 2, die den Diamanten Bilden. Für jede Klausel wird nun ein Paar aus zwei Knoten gebildet. Jedes Paar wird an beiden Seiten von einem Trennknoten flankiert.



Falls nun x_i in einer Klausel c_j auftritt fügen wir z' zwei Kanten vom j-ten Paar im i-ten Diamanten zum j-en Knoten ein und zwar eine vom linken Knoten zu c_j und von c_j zum rechten Knoten.



Falls $\overline{x_i}$ in der Klausel vorkommt, ist die Richtung der zugefügten Knoten umgekehrt.



G ist nun vollständig konstruiert. Nun wollen wir zeigen, dass wenn Φ erfüllbar ist G einen Hamiltonweg aufweist.

Zuerst betrachten wir wieder nur die Struktur aus Diamanten. Um die Knoten in der der Mitte der Diamanten zu vebinden kann man von links nach rechts $(l - r)$ oder von rechts nach links $(r - l)$ gehen. Falls $x_i = 1$ wird $l - r$, falls $x_i = 0$ $r - l$ angewandt.

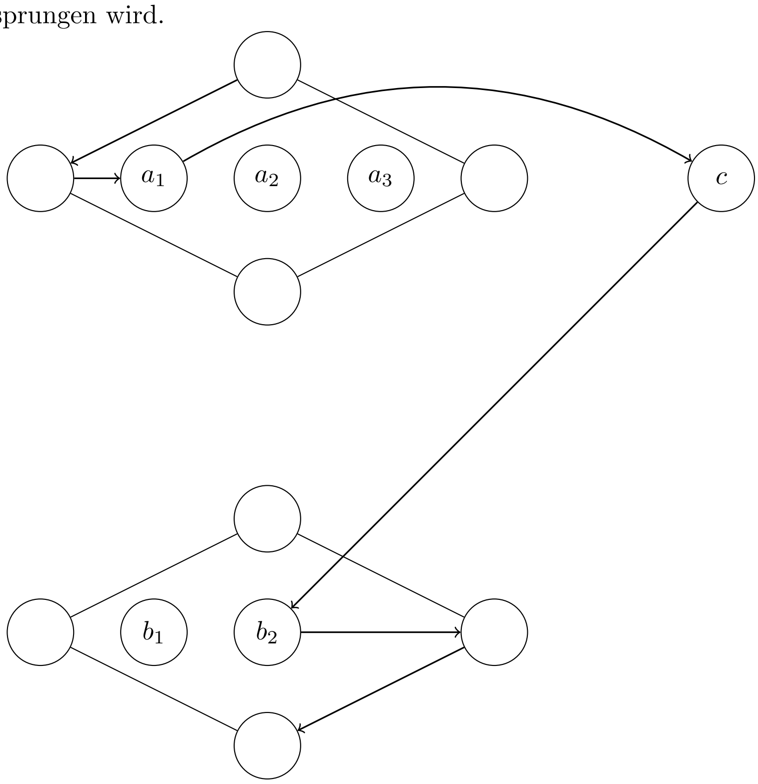
Um nun auch die Knoten für die Klauseln mit einzubinden wählen wir in jeder Klausel genau ein Literal aus, welches den Wert 1 hat. Falls wir x_i in c_j auswählen wird der Hamiltonweg mit gewählt, da $x_i = 1$ sein muss und so im Diamanten $l - r$ vorliegt, so geben wir also im c_j-en Paar vom linken Knoten zum c_j-en Knoten und von da zum rechten Knoten und können von da aus normal fortfahren.

Im Falle von $\overline{x_i}$ sind links und rechts lediglich vertauscht. Auf diese Weise erhalten wir einen Hamiltonweg.

Bei der Gegenrichtung nehmen wir an, dass eine Hamiltonweg von s zu t existiert und zeigen, dass eine akzeptierende Belegung für Φ existieren.

Hierfür nehmen wir nun an, dass der Graph in Normalform ist, also von s zu t durch die Diamanten geht, bis auf die Umwege über die Klauselknoten. Über $r - l$ und $l - r$ können wir die Variablen ihre Werte zuweisen und über die Richtung der Umwege zu den Klauselknoten können wir bestimmen, welches Literal den Wert 1 der Klausel erzeugt.

Nun bleibt zu zeigen, dass ein Hamiltonweg Normalform haben muss. Das System kann nur dadurch zerstört werden, wenn ein den Klauselknoten, der durch ein Diamantenauge betreten wird, aber davon in einen anderen zurückgesprungen wird.



Der Weg geht von a_1 zu c und dann statt zu a_2 zu b_2. Entweder a_2 oder a_3 müssen Trennknoten sein. Falls a_2 der Trennknoten ist, führen nur Kanten von a_1 und a_3 nach a_2. Falls a_3 der Trennknoten wäre, würden a_1 und a_2 zum gleichen Paar gehören und die einzige eingehende Kante von a_2 wären von a_1, a_3 und c. Da der gesamte Weg ein Hamiltonweg sein so müsste er auch a_2 endhalten, allerdings kann a_2 nicht von a_1 oder c erreicht werden, weil deren ausgehenden Kanten schon mit einem anderen Knoten verbunden sind. Somit bleibt nur noch a_3 als möglicher Ausgangspunkt, dies ist ebenfalls nicht möglich, da a_2 nur noch mit a_3 als noch nicht in den Weg einbezogenen Knoten verbunden ist und so man in a_2 stehen bleiben würde, somit kann so kein Hamiltonweg zustande kommen, woraus folgt, dass alle Hamiltonwege normal sein müssen.

Quellen:

- Sipsor, Mirhael: Introduction to the Theory of computing, 2.Auflage. Boston: Course technolog, 2006
- Wegener, Ingo: Theoretische Informatik, 2. Auflage, Stuttgart, Leipzig, B.G.Teutner 1999