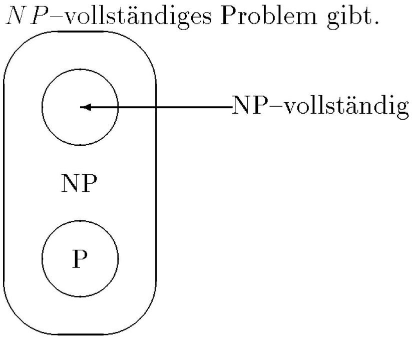
$\Rightarrow$ Zum Nachweis von $P = NP$ genügt die Angabe eines polynomialen Algorithmus für ein NP–vollständiges Problem:
Die Annahme $P \neq NP$ bedeutet, daß es keinen effizienten Algorithmus für ein NP–vollständiges Problem gibt.



Definition.
Das folgende Problem heißt "Erfüllbarkeitsproblem der Aussagenlogik" SAT.

Gegeben: Eine Formel F der Aussagenlogik mit n Variablen, $n \in \mathbb{N}$
Gefragt: Ist F erfüllbar?
(d.h. $\exists$ Belegung $a \in \{0,1\}^n$ mit $F(a) = 1$?)

Formal
$SAT = \{code(F) \in \Sigma^* : F$ ist erfüllbare Formel der Aussagenlogik$\}$

Theorem von Cook
Das Erfüllbarkeitsproblem der Aussagenlogik SAT ist NP–vollständig.

Beweis.
1.) Zu zeigen: $SAT \in NP$.
Angabe einer polynomial zeitbeschränkten NTM für SAT:
M stellt in einem Durchlauf über der Eingabe fest, welche Variablen in F vorkommen.
Seien dies $x_1, \ldots, x_k$.
M rät die Werte $a_1, \ldots, a_k \in \{0,1\}$ für $x_1, \ldots, x_k$ und setzt diese in F ein.
(es existieren 2^k mögliche unabhängige Berechnungen – für jede Belegung eine)
Für jede Belegung rechnet M jeweils deterministisch den Wert von F aus und akzeptiert, falls dieser 1 ist.
$F \in SAT \Leftrightarrow M$ akzeptiert F.
Wegen $k \leq |F|$ ist M polynomialzeit beschränkt.
$\Rightarrow SAT \in NP$.

2.) Zu zeigen: SAT ist NP–hart.
Sei $L \in NP$ beliebig.
M NTM für L der Rechenzeit p.
OBdA gelte: $\delta(z_e, a) \ni (z_e, a, N)$ (d.h. erreichte Endzustände werden nicht mehr

verlassen).
Sei $x = x_1, \ldots, x_n \in \Sigma^*$ die Eingabe von M.
Konstruktion einer Fromel F mit

$x \in L \Leftrightarrow F$ ist erfüllbar.

Sei $\Gamma = \{a_1, \ldots, a_l\}$ das Anfangsalphabet
$Z = \{z_1, \ldots, z_k\}$ die Zustandsmenge von M
F enthält folgende Variable:

Variable	Indizes	intendierte Bedeutung
$zust_{t,z}$	$t = 0, \ldots, p(n)$ $z \in Z$	$zust_{t,z} = 1 \Leftrightarrow$ nach t Schritten befindet sich M im Zustand z
$pos_{t,i}$	$t = 0, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$	$pos_{i,j} = 1 \Leftrightarrow$ M's Schreib–Lesekopf befindet sich nach t Schritten auf Position i
$band_{t,i,a}$	$t = 0, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$ $a \in \Gamma$	$band_{t,i,a} = 1 \Leftrightarrow$ nach t Schritten befindet sich auf Bandposition i das Zeichen a

F besteht aus mehreren Bauteilen:

$G(x_1, \ldots, x_m) = 1 \Leftrightarrow$ für geanu ein i ist $x_i = 1$.

Behauptung
G existiert und es gilt $size(G) = O(m^2)$.

Beweis.

$$G = (\bigvee_{i=1}^{k} x_i) \wedge (\bigwedge_{j=1}^{m-1} \bigwedge_{l=j+1}^{m} (\neg(x_j \wedge x_l))$$

Die erste Teilformel wird genau dann wahr, wenn mindestens eine Variable wahr ist.
Die zweite Teilformel wird genau dann wahr, wenn höchstens eine Variable wahr wird.

$$F = R \wedge A \wedge \ddot{U}_1 \wedge \ddot{U}_2 \wedge E \text{ , wobei}$$

R	für Randbedingung
A	für Anfangsbedingung
$\ddot{U}_1, U_2$	für Übergangsbedingung und
E	für Endbedingung steht.

R drückt aus:

- Zu jedem Zeitpunkt t ergibt sich für genau ein z $zust_{t,z} = 1$.
- Zu jedem Zeitpunkt t gibt es genau eine Bandposition i mit $pos_{t,i} = 1$.
- Zu jedem Zeitpunkt t und jeder Bandposition i gibt es genau ein a mit $band_{t,i,a} = 1$

$$R = \bigwedge_t [G(zust_{t,z_1}, \ldots, zust_{t,z_k}) \wedge G(pos_{t,-p(n)}, \ldots, pos_{t,p(n)}) \wedge \bigwedge_i G(band_{t,i,a_1}, \ldots, band_{t,i,a_l})]$$

A beschreibt den Status der Variablen für den Fall $t = 0$:

$$A = zust_{0,z_0} \wedge pos_{0,1} \wedge \bigwedge_{j=1}^{n} band_{0,j,x_j} \wedge \bigwedge_{j=-p(n)}^{0} band_{0,j,\Box} \wedge \bigwedge_{j=n+1}^{p(n)} band_{0,j,\Box}$$

$\ddot{U}_1$ beschreibt den Übergang von Zeitpunkt t nach $t+1$ an der Kopfposition ($y \in \{-1, 0, +1\}$):

$\ddot{U}_1 = \bigwedge_{t,z,i,a}[(zust_{t,z} \wedge pos_{t,i} \wedge band_{t,i,a}) \rightarrow \bigvee_{z',a',y \text{ mit } \delta(z,a) \ni (z',a',y)} (zust_{t+1,z'} \wedge pos_{t+1,i+y} \wedge band_{t+1,i,a'}]$

$\ddot{U}_2$ besagt, daß auf den übrigen Bandfeldern nichts passiert:

$$\ddot{U}_2 = \bigwedge_{t,i,a} ((\neg pos_{t,i} \wedge band_{t,i,a}) \rightarrow band_{t+1,i,a})$$

E überprüft, ob der Endzustand erreicht ist (wird auf jeden Fall im Zeitpunkt $p(n)$ erreicht):

$$E = \bigvee_{z \in E} zust_{p(n),z}$$

($\rightarrow$): Sei $x \in L$
$\Rightarrow$ $\exists$ nichtdeterministische Rechnung der Länge $p(n)$, die in einen Endzustand führt.
$\Rightarrow$ Alle Teilformeln von F erhalten den Wert 1.
$\Rightarrow$ $F(x)$ erhält den Wert 1.
$\Rightarrow$ $F(x)$ ist erfüllbar.

($\leftarrow$) : Sei $F(x)$ erfüllbar.
$\Rightarrow$ $\exists$ Belegung, die F und alle Teilformeln den Wert 1 annehmen läßt.

Insbesondere ist R erfüllt:
$\Rightarrow$ $zust_{t,z}, pos_{t,i}, band_{t,i,a}$ können $\forall t$ als Konfiguration von M interpretiert werden.
Insbesondere ist A erfüllt:

für $t = 0$ kann aus den Variablenwerten die Startkonfiguration von M abgelesen werden.

Insbesondere sind $\ddot{U}_1, \ddot{U}_2$ erfüllt:
$\Rightarrow$ zwischen t und $t+1$ ist die Nachfolgekonfigurationsbedingung erfüllt.
$\Rightarrow$ $\forall t = 0, 1, 2, \ldots$ ist eine mögliche nichtdeterministische Rechnung beschrieben.

Insbesondere ist E erfüllt:
$\Rightarrow$ Rechnung erreicht Endzustand.

Insgesamt gilt also :

$$x \in L(M)$$

Noch zu zeigen: F ist in polynomieller Zeit berechenbar:
Offenbar ist der Aufwand zur Erzeugung von F linear in der Länge von F.

$$\begin{array}{rcl} \text{Wegen } |R| & = & O(n^2) \\ |A| & = & O(n) \\ |\ddot{U}_1| & = & O(n^2) \\ |\ddot{U}_2| & = & O(n^2) \\ |E| & = & O(1) \\ & & \\ \text{gilt } |F| & = & O(n^2) \end{array}$$

■

NP–Berechnungen: "guess and check"
Alle deterministischen Algorithmen zur Berechnung von SAT haben Komplexität $2^{O(n)}$.
(Z.B. systematisches Durchprobieren aller Eingabeformeln).
Da SAT NP–hart ist folgt:

$$NP \subseteq \bigcup_{p \ Polynom} TIME(2^{p(n)})$$

2.4 Weitere NP–vollständige Probleme

2.4.1 $3SAT$

Definition. ($3SAT$)
Gegeben: Boolesche Formel F in KNF mit höchstens 3 Literalen pro Klausel.
Gefragt: Ist F erfüllbar?

Satz.
$3SAT$ ist NP–vollständig.

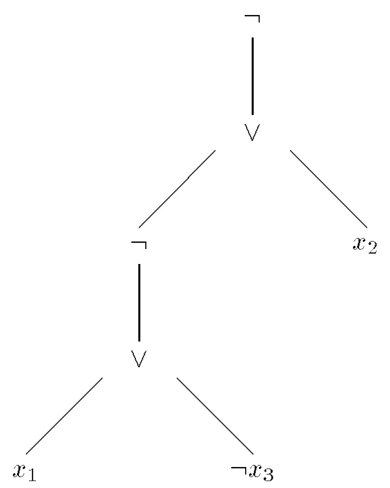
Beweis. 1.) $3SAT \in NP$, klar mit "guess and check"– Argument.
2.) $3SAT$ NP–hart.
Es reicht zu zeigen : $SAT \leq_p 3SAT$.
Angabe eines polynomiellen Verfahrens, das eine beliebige Formel F in eine Formel F' in KNF mit höchstens 3 Literalen pro Klausel umformt mit:

$$F \text{ erfüllbar} \Leftrightarrow F' \text{ erfüllbar.}$$

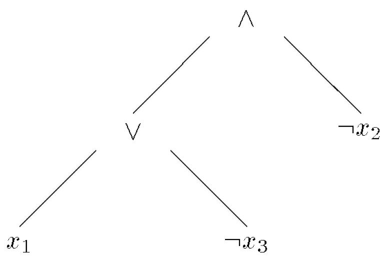
(Dabei genügt "Erfüllbarkeitsäquivalenz". "Äquivalenz ist nicht notwendig.)

Allgemeine Verfahren zur Überführung von F in äquivalente KNF benötigt i.a. exponentielle Zeit und garantiert nicht, daß alle Klauseln höchstens 3 Literale enthalten.

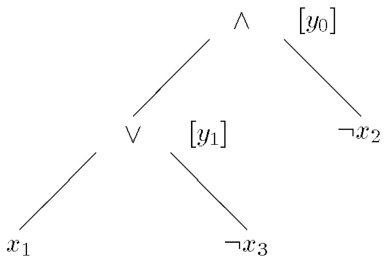
Erläuterung des Verfahrens am Beispiel:



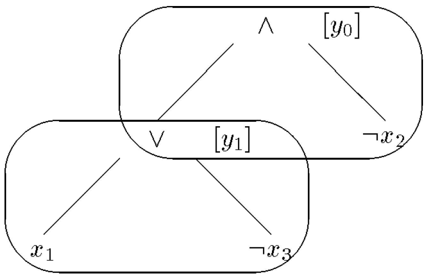
1. Schritt
Mit den DeMorgan'schen Regeln werden alle Negationszeichen zu den Variablen gebracht.



2. Schritt
Jedem inneren Knoten wird eine Variable $\{y_0, y_1, \ldots\}$ zugeordnet, wobei der Baumwurzel y_0 zugeordnet wird.



3. Schritt
Jedem inneren Knoten wird eine Teilformel der Form $(v \leftrightarrow (y \circ z))$, $\circ \in \{\wedge, \vee\}$ zugeordnet. Man erhält eine neue Formel F_1, indem man alle Teilformeln durch $\wedge$ verknüpft und für die Wurzel y_0 die Teilformel $[y_0]$ hinzunimmt.



Es gilt:

$$F \text{ erfüllbar } \Leftrightarrow F_1 \text{ erfüllbar}$$

($\rightarrow$) : Eine erfüllende Belegung von F liefert eine erfüllende Belegung von F_1.
($\leftarrow$) : Die Belegung der x–Variablen einer erfüllenden Belegung von F_1 liefert eine erfüllende belegung für F.

4. Schritt
Umformung jeder Teilformel in KNF.
Es enstehen Klauseln mit höchstens 3 Literalen.
Das Verfahren ist polynomial, da jede Teilformel in konstanter Länge umgeformt werden kann.

Beispiel $[a \leftrightarrow (b \vee c)] \mapsto (a \vee \neg b) \wedge (\neg a \vee b \vee c) \wedge (a \vee \neg c)$
$[a \leftrightarrow (b \wedge c)] \mapsto (\neg a \vee b) \wedge (\neg a \vee c) \wedge (a \vee \neg b \vee \neg c)$

$\Rightarrow F_1 = y_0 \wedge (\neg y_0 \vee y_1) \wedge (\neg y_0 \vee \neg x_2) \wedge (y_0 \vee \neg y_1 \vee x_2) \wedge (y_1 \vee \neg x_1) \wedge (\neg y_1 \vee x_1 \vee \neg x_3) \wedge (y_1 \vee x_3)$
mit
F_1 ist erfüllbarkeitsäquivalent mit F.

Umformung von F in F_1 ist in polynomieller Zeit möglich. ■

Bemerkung
Für ein analoges Problem gilt:
$2SAT \in P$, da es nur polynomiell viele verschiedene Klauseln mit höchstens 2 Literalen über $\{x_1, .., x_n\}$ gibt.

2.4.2 CLIQUE

Definition. (CLIQUE)
Gegeben: Ein ungerichteter Graph $G = (V, E)$, $k \in I\!N$.
Gefragt: Besitzt G eine "Clique" der Größe k?

Wobei Clique ein vollständiger Teilgraph $G' = (V', E')$ ist, mit

$$(u, v) \in E \quad \forall u, v \in V', \;\; u \neq v$$

Satz.
CLIQUE ist NP–vollständig.

Beweis.
1.) CLIQUE$\in NP$ mit "guess and check".
2.) CLIQUE ist NP–hart.
Sei F Formel in KNF mit (genau) 3 Literalen pro Klausel.
$F = (z_{1,1} \vee z_{1,2} \vee z_{1,3}) \wedge \ldots \wedge (z_{m,1} \vee z_{m,2} \vee z_{m,3})$ mit $z_{i,j} \in \{x_1, x_2, \ldots\} \wedge \{\neg x_1, \neg x_2, \ldots\}$