11a

$\mathcal{NP}$-Vollständigkeit von SAT

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 8. April 2025
Basierend auf Folien von PD Dr. David Sabel und Dr. Jan Johannsen

Wiederholung: $\mathcal{NP}$-Vollständigkeit

Definition

Eine Sprache L heißt $\mathcal{NP}$-vollständig, wenn gilt

1. $L \in \mathcal{NP}$ und
2. L ist $\mathcal{NP}$-schwer: für alle $L' \in \mathcal{NP}$ gilt $L' \leq_p L$.

Wiederholung: $\mathcal{NP}$-Vollständigkeit

Definition

Eine Sprache L heißt $\mathcal{NP}$-vollständig, wenn gilt

1. $L \in \mathcal{NP}$ und
2. L ist $\mathcal{NP}$-schwer: für alle $L' \in \mathcal{NP}$ gilt $L' \leq_p L$.

Beweistechnik für $\mathcal{NP}$-Vollständigkeit von L:

1. Zeige $L \in \mathcal{NP}$.
2. Zeige $L_0 \leq_p L$ für ein bekanntes $\mathcal{NP}$-vollständiges Problem L_0.
   Dann folgt die $\mathcal{NP}$-Schwere von L.

SAT-Problem

Wir brauchen ein erstes Problem, dessen $\mathcal{NP}$-Vollständigkeit wir von Hand nachweisen müssen.

Dafür nehmen wir das Erfüllbarkeitsproblem der Aussagenlogik (kurz SAT).

Syntax der Aussagenlogik

Definition

Die aussagenlogischen Formeln über einer Menge X von Variablen sind durch folgende Regeln definiert:

- Jede Variable $x \in X$ ist eine Formel.
- Ist F eine Formel, dann auch $\neg F$.
- Sind F und G Formeln, dann auch $(F \wedge G)$, $(F \vee G)$, $(F \Longrightarrow G)$ und $(F \Longleftrightarrow G)$.

Semantik der Aussagenlogik

Definition

Sei $I : X \to \{0, 1\}$ eine Belegung der Variablen X mit den Wahrheitswerten 0 („falsch“) und 1 („wahr“).

Semantik der Aussagenlogik

Definition

Sei $I : X \to \{0, 1\}$ eine Belegung der Variablen X mit den Wahrheitswerten 0 („falsch“) und 1 („wahr“).

Der Wert $I(F)$ einer gegebenen Formel F wird durch folgende Regeln definiert:

- $I(x)$ ist durch I gegeben.
- $I(\neg F) := 1 - I(F)$.
- $I(F \wedge G) := \min\{I(F), I(G)\}$.
- $I(F \vee G) := \max\{I(F), I(G)\}$.
- $I(F \Longrightarrow G) := I(\neg F \vee G)$.
- $I(F \Longleftrightarrow G) := I((F \Longrightarrow G) \wedge (G \Longrightarrow F))$.

Semantik der Aussagenlogik

Definition

Sei $I : X \to \{0, 1\}$ eine Belegung der Variablen X mit den Wahrheitswerten 0 („falsch“) und 1 („wahr“).

Der Wert $I(F)$ einer gegebenen Formel F wird durch folgende Regeln definiert:

- $I(x)$ ist durch I gegeben.
- $I(\neg F) := 1 - I(F)$.
- $I(F \wedge G) := \min\{I(F), I(G)\}$.
- $I(F \vee G) := \max\{I(F), I(G)\}$.
- $I(F \implies G) := I(\neg F \vee G)$.
- $I(F \iff G) := I((F \implies G) \wedge (G \implies F))$.

I erfüllt F, wenn $I(F) = 1$.
F ist erfüllbar, wenn es eine Belegung I gibt mit $I(F) = 1$.

SAT-Problem

Definition

Das Erfüllbarkeitsproblem der Aussagenlogik (kurz SAT) lässt sich wie folgt formulieren.

gegeben: eine aussagenlogische Formel F

gefragt: Ist F erfüllbar, d.h. gibt es eine erfüllende Belegung der Variablen, sodass F den Wert 1 erhält?

SAT-Problem

Definition

Das Erfüllbarkeitsproblem der Aussagenlogik (kurz SAT) lässt sich wie folgt formulieren.

gegeben: eine aussagenlogische Formel F

gefragt: Ist F erfüllbar, d.h. gibt es eine erfüllende Belegung der Variablen, sodass F den Wert 1 erhält?

Als formale Sprache:
$\text{SAT} = \{code(F) \in \Sigma^* \mid F \text{ ist eine erfüllbare Formel der Aussagenlogik}\}$

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

M verwendet Nichtdeterminismus, um eine Belegung $I : X \to \{0, 1\}$ zu „raten“.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

M verwendet Nichtdeterminismus, um eine Belegung $I : X \to \{0, 1\}$ zu „raten“.

Jede der 2^n nichtdeterministischen Berechnungen berechnet einen Wert von $I(F)$.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

M verwendet Nichtdeterminismus, um eine Belegung $I : X \to \{0, 1\}$ zu „raten“.

Jede der 2^n nichtdeterministischen Berechnungen berechnet einen Wert von $I(F)$.

Akzeptanz, wenn $I(F) = 1$, sonst verwerfen.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

M verwendet Nichtdeterminismus, um eine Belegung $I : X \to \{0, 1\}$ zu „raten“.

Jede der 2^n nichtdeterministischen Berechnungen berechnet einen Wert von $I(F)$.

Akzeptanz, wenn $I(F) = 1$, sonst verwerfen.

Da jede Belegung geprüft wird, gilt:
M akzeptiert eine Formel F g.d.w. $code(F) \in$ SAT.

Zugehörigheit von SAT zu $\mathcal{NP}$

Lemma

SAT $\in \mathcal{NP}$.

Beweis Konstruiere eine NTM M mit $code(F)$ als Eingabe.

M berechnet, welche Variablen in F vorkommen. Seien diese $X = \{x_1, \ldots, x_n\}$.

M verwendet Nichtdeterminismus, um eine Belegung $I : X \to \{0, 1\}$ zu „raten“.

Jede der 2^n nichtdeterministischen Berechnungen berechnet einen Wert von $I(F)$.

Akzeptanz, wenn $I(F) = 1$, sonst verwerfen.

Da jede Belegung geprüft wird, gilt:
M akzeptiert eine Formel F g.d.w. $code(F) \in$ SAT.

Jeder Berechnungspfad von M läuft in Polynomialzeit in $|code(F)|$,
da die Anzahl der Variablen durch die Eingabegröße beschränkt ist. □

Polynomielle Verifizierbarkeit als Beweistechnik

Der Nachweis, dass eine Sprache in $\mathcal{NP}$ liegt, geht oft so wie bei SAT:

1. Verwende Nichtdeterminismus, um potentielle Lösung zu raten.
2. Zeige, dass eine Lösung in Polynomialzeit verifiziert werden kann.

$\mathcal{NP}$-Schwere von SAT

Wir müssen zeigen:

$$L \leq_p \text{SAT für alle } L \in \mathcal{NP}$$

$\mathcal{NP}$-Schwere von SAT

Wir müssen zeigen:

$$L \leq_p \text{SAT für alle } L \in \mathcal{NP}$$

Da $L \in \mathcal{NP}$, gibt es eine polynomiell zeitbeschränkte NTM M, die L akzeptiert.

$\mathcal{NP}$-Schwere von SAT

Wir müssen zeigen:

$$L \leq_p \text{SAT für alle } L \in \mathcal{NP}$$

Da $L \in \mathcal{NP}$, gibt es eine polynomiell zeitbeschränkte NTM M, die L akzeptiert.

Für Wort w erstelle Formel $f(w) = F$ (in deterministischer Polynomialzeit), sodass gilt:

$$F \text{ ist erfüllbar g.d.w. } M \text{ akzeptiert } w$$

$\mathcal{NP}$-Schwere von SAT

Wir müssen zeigen:

$$L \leq_p \text{SAT für alle } L \in \mathcal{NP}$$

Da $L \in \mathcal{NP}$, gibt es eine polynomiell zeitbeschränkte NTM M, die L akzeptiert.

Für Wort w erstelle Formel $f(w) = F$ (in deterministischer Polynomialzeit), sodass gilt:

$$F \text{ ist erfüllbar g.d.w. } M \text{ akzeptiert } w$$

In den nächsten Folien werden wir F definieren und zeigen, dass sie die erwünschte Eigenschaft hat.

$\mathcal{NP}$-Schwere von SAT

Wir müssen zeigen:

$$L \leq_p \text{SAT für alle } L \in \mathcal{NP}$$

Da $L \in \mathcal{NP}$, gibt es eine polynomiell zeitbeschränkte NTM M, die L akzeptiert.

Für Wort w erstelle Formel $f(w) = F$ (in deterministischer Polynomialzeit), sodass gilt:

$$F \text{ ist erfüllbar g.d.w. } M \text{ akzeptiert } w$$

In den nächsten Folien werden wir F definieren und zeigen, dass sie die erwünschte Eigenschaft hat.

F wird das Verhalten von M auf w kodieren. Aus einer erfüllende Belegung I für F wird sich einen akzeptierenden Lauf von M ablesen lassen und umgekehrt.

Hilfssatz für den $\mathcal{NP}$-Schwere-Beweis von SAT

Lemma

Für aussagenlogische Variablen $\{x_1, \ldots, x_n\}$ gibt es eine aussagenlogische Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$, sodass

$$I(\mathit{exactlyOne}(x_1, \ldots, x_n)) = 1 \text{ g.d.w. } I \text{ setzt genau eine der Variablen } x_i \text{ auf 1 und alle anderen auf 0}$$

Dabei ist die Größe der Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$ in $O(n^2)$.

Hilfssatz für den $\mathcal{NP}$-Schwere-Beweis von SAT

Lemma

Für aussagenlogische Variablen $\{x_1, \ldots, x_n\}$ gibt es eine aussagenlogische Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$, sodass

$$I(\mathit{exactlyOne}(x_1, \ldots, x_n)) = 1 \text{ g.d.w. } I \text{ setzt genau eine der Variablen } x_i \text{ auf 1 und alle anderen auf 0}$$

Dabei ist die Größe der Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$ in $O(n^2)$.

Beweis Wir nehmen

$$\mathit{exactlyOne}(x_1, \ldots, x_n) := (x_1 \vee \cdots \vee x_n) \wedge \bigwedge_{1 \leq i < j \leq n} \neg(x_i \wedge x_j)$$

Hilfssatz für den $\mathcal{NP}$-Schwere-Beweis von SAT

Lemma

Für aussagenlogische Variablen $\{x_1, \ldots, x_n\}$ gibt es eine aussagenlogische Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$, sodass

$$I(\mathit{exactlyOne}(x_1, \ldots, x_n)) = 1 \text{ g.d.w. } I \text{ setzt genau eine der Variablen } x_i \text{ auf 1 und alle anderen auf 0}$$

Dabei ist die Größe der Formel $\mathit{exactlyOne}(x_1, \ldots, x_n)$ in $O(n^2)$.

Beweis Wir nehmen

$$\mathit{exactlyOne}(x_1, \ldots, x_n) := (x_1 \vee \cdots \vee x_n) \wedge \bigwedge_{1 \le i < j \le n} \neg(x_i \wedge x_j)$$

- $(x_1 \vee \cdots \vee x_n)$ sichert „mindestens 1“ zu.
- $\bigwedge_{1 \le i < j \le n} \neg(x_i \wedge x_j)$ sichert „höchstens 1“ zu.

□

$\mathcal{NP}$-Schwere von SAT

Lemma

SAT ist $\mathcal{NP}$-schwer.

$\mathcal{NP}$-Schwere von SAT

Lemma

SAT ist $\mathcal{NP}$-schwer.

Beweis Sei $L \in \mathcal{NP}$ beliebig. Wir müssen zeigen: $L \leq_p \text{SAT}$.

$\mathcal{NP}$-Schwere von SAT

Lemma

SAT ist $\mathcal{NP}$-schwer.

Beweis Sei $L \in \mathcal{NP}$ beliebig. Wir müssen zeigen: $L \leq_p$ SAT.

Sei M eine NTM mit $L(M) = L$, w eine Eingabe für M und $ntime_M(w) \leq p(|w|)$.

$\mathcal{NP}$-Schwere von SAT

Lemma

SAT ist $\mathcal{NP}$-schwer.

Beweis Sei $L \in \mathcal{NP}$ beliebig. Wir müssen zeigen: $L \leq_p$ SAT.

Sei M eine NTM mit $L(M) = L$, w eine Eingabe für M und $ntime_M(w) \leq p(|w|)$.

Wir werden eine aussagenlogische Formel F konstruieren, sodass gilt

F ist erfüllbar g.d.w. M akzeptiert w

Dabei muss F in Polynomialzeit konstruierbar sein.

$\mathcal{NP}$-Schwere von SAT

Lemma

SAT ist $\mathcal{NP}$-schwer.

Beweis Sei $L \in \mathcal{NP}$ beliebig. Wir müssen zeigen: $L \leq_p$ SAT.

Sei M eine NTM mit $L(M) = L$, w eine Eingabe für M und $ntime_M(w) \leq p(|w|)$.

Wir werden eine aussagenlogische Formel F konstruieren, sodass gilt

F ist erfüllbar g.d.w. M akzeptiert w

Dabei muss F in Polynomialzeit konstruierbar sein.

D.h. wir geben eine in Polynomialzeit berechenbare Funktion $f(w)$ an, sodass $f(w) \in$ SAT g.d.w. M akzeptiert w.

$\mathcal{NP}$-Schwere von SAT

Notationen:

Eingabe:	$w = a_1 \cdots a_n \in \Sigma^*$	Zustände:	$Z = \{z_0, \ldots, z_k\}$
Bandalphabet:	$\Gamma = \{b_1, \ldots, b_\ell\}$	Startzustand:	z_0

$\mathcal{NP}$-Schwere von SAT

Notationen:

Eingabe:	$w = a_1 \cdots a_n \in \Sigma^*$	Zustände:	$Z = \{z_0, \ldots, z_k\}$
Bandalphabet:	$\Gamma = \{b_1, \ldots, b_\ell\}$	Startzustand:	z_0

Aussagenlogische Variablen in der Formel F:

Variable	Index-Bereich	Bedeutung
$State_{t,z}$	$t = 0, 1, \ldots, p(n)$ $z = z_0, \ldots, z_k$	$State_{t,z} = 1$ g.d.w. nach t Schritten ist M im Zustand z.
$Pos_{t,i}$	$t = 0, 1, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$	$Pos_{t,i} = 1$ g.d.w. nach t Schritten ist der Schreib-Lesekopf auf Position i
$Tape_{t,i,b}$	$t = 0, 1, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$ $b = b_1, \ldots, b_\ell$	$Tape_{t,i,b} = 1$ g.d.w. nach t Schritten steht in Position i das Zeichen b

$\mathcal{NP}$-Schwere von SAT

Notationen:

Eingabe:	$w = a_1 \cdots a_n \in \Sigma^*$	Zustände:	$Z = \{z_0, \ldots, z_k\}$
Bandalphabet:	$\Gamma = \{b_1, \ldots, b_\ell\}$	Startzustand:	z_0

Aussagenlogische Variablen in der Formel F:

Variable	Index-Bereich	Bedeutung
$State_{t,z}$	$t = 0, 1, \ldots, p(n)$ $z = z_0, \ldots, z_k$	$State_{t,z} = 1$ g.d.w. nach t Schritten ist M im Zustand z.
$Pos_{t,i}$	$t = 0, 1, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$	$Pos_{t,i} = 1$ g.d.w. nach t Schritten ist der Schreib-Lesekopf auf Position i
$Tape_{t,i,b}$	$t = 0, 1, \ldots, p(n)$ $i = -p(n), \ldots, p(n)$ $b = b_1, \ldots, b_\ell$	$Tape_{t,i,b} = 1$ g.d.w. nach t Schritten steht in Position i das Zeichen b

Die Bereiche reichen aus, da die TM nicht mehr als $p(n)$ Schritte macht.

$\mathcal{NP}$-Schwere von SAT

Aufbau der Formel F:

$$F = \mathit{Rand} \wedge \mathit{Anfang} \wedge \mathit{Übergang} \wedge \mathit{Ende}$$

- *Rand*: Randbedingungen
- *Anfang*: Anfangsbedingungen
- *Übergang*: Bedingungen für den Zustandsübergang
- *Ende*: Endbedingung

Randbedingungen

Zu jedem Zeitpunkt t

Randbedingungen

Zu jedem Zeitpunkt t

- ... ist M in genau einem Zustand z:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} \mathit{exactlyOne}(\mathit{State}_{t,z_0}, \ldots, \mathit{State}_{t,z_k})$$

Randbedingungen

Zu jedem Zeitpunkt t

- ... ist M in genau einem Zustand z:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} exactlyOne(State_{t,z_0}, \ldots, State_{t,z_k})$$

- ... ist der Schreib-Lesekopf von M in genau einer Position auf dem Band:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} exactlyOne(Pos_{t,-p(n)}, \ldots, Pos_{t,p(n)})$$

Randbedingungen

Zu jedem Zeitpunkt t

- ... ist M in genau einem Zustand z:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} exactlyOne(State_{t,z_0}, \ldots, State_{t,z_k})$$

- ... ist der Schreib-Lesekopf von M in genau einer Position auf dem Band:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} exactlyOne(Pos_{t,-p(n)}, \ldots, Pos_{t,p(n)})$$

- ... befindet sich in jeder Bandzelle genau ein Symbol aus Γ:

$$\bigwedge_{t \in \{0,\ldots,p(n)\}} \bigwedge_{i \in \{-p(n),\ldots,p(n)\}} exactlyOne(Tape_{t,i,b_1}, \ldots, Tape_{t,i,b_\ell})$$

Randbedingungen

Daher:

$$\mathit{Rand} := \bigwedge_{t \in \{0,\ldots,p(n)\}} \left(\begin{array}{l} \mathit{exactlyOne}(\mathit{State}_{t,z_0}, \ldots, \mathit{State}_{t,z_k}) \\ \wedge\ \mathit{exactlyOne}(\mathit{Pos}_{t,-p(n)}, \ldots, \mathit{Pos}_{t,p(n)}) \\ \wedge \bigwedge_{i \in \{-p(n),\ldots,p(n)\}} \mathit{exactlyOne}(\mathit{Tape}_{t,i,b_1}, \ldots, \mathit{Tape}_{t,i,b_\ell}) \end{array} \right)$$

Anfangsbedingungen

Die Bedingungen zum Zeitpunkt $t = 0$:

Anfangsbedingungen

Die Bedingungen zum Zeitpunkt $t = 0$:

- M ist im Startzustand: $State_{0,z_0}$

Anfangsbedingungen

Die Bedingungen zum Zeitpunkt $t = 0$:

- M ist im Startzustand: $State_{0,z_0}$
- Der Schreib-Lesekopf ist auf Position 0: $Pos_{0,0}$

Anfangsbedingungen

Die Bedingungen zum Zeitpunkt $t = 0$:

- M ist im Startzustand: $State_{0,z_0}$
- Der Schreib-Lesekopf ist auf Position 0: $Pos_{0,0}$
- Die Eingabe $w = a_1 \dots a_n$ steht auf dem Band und alle anderen Zellen enthalten das Blank-Symbol:

$$\left(\bigwedge_{i\in\{0,\dots,n-1\}} Tape_{0,i,a_{i+1}}\right) \wedge \left(\bigwedge_{i\in\{-p(n),\dots,-1\}} Tape_{0,i,\square}\right) \wedge \left(\bigwedge_{i\in\{n,\dots,p(n)\}} Tape_{0,i,\square}\right)$$

Anfangsbedingungen

Daher:

$$\mathit{Anfang} := \mathit{State}_{0,z_0} \wedge \mathit{Pos}_{0,0} \wedge \Big(\bigwedge_{i\in\{0,\ldots,n-1\}} \mathit{Tape}_{0,i,a_{i+1}}\Big) \wedge \Big(\bigwedge_{i\in\{-p(n),\ldots,-1\}} \mathit{Tape}_{0,i,\square}\Big) \wedge \Big(\bigwedge_{i\in\{n,\ldots,p(n)\}} \mathit{Tape}_{0,i,\square}\Big)$$

Übergangsbedingungen

Für Übergang von t zu $t+1$:

Übergangsbedingungen

Für Übergang von t zu $t+1$:

- Zustand, Bandinhalt, Position ändern.
  Sei $dir(N) = 0, dir(L) = -1, dir(R) = 1$:

$$\bigwedge_{\substack{t\in\{0,\ldots,p(n)-1\},z\in Z,\\ i\in\{-p(n)+1,\ldots,p(n)-1\},b\in\Gamma}} \left(\begin{array}{l} (State_{t,z} \wedge Pos_{t,i} \wedge Tape_{t,i,b}) \\ \Longrightarrow \bigvee_{(z',b',y)\in\delta(z,b)} (State_{t+1,z'} \wedge Pos_{t+1,i+dir(y)} \wedge Tape_{t+1,i,b'}) \end{array} \right)$$

Übergangsbedingungen

Für Übergang von t zu $t+1$:

- Zustand, Bandinhalt, Position ändern.
  Sei $dir(N) = 0, dir(L) = -1, dir(R) = 1$:

$$\bigwedge_{\substack{t\in\{0,\ldots,p(n)-1\},z\in Z,\\ i\in\{-p(n)+1,\ldots,p(n)-1\},b\in\Gamma}} \left(\begin{array}{l} (State_{t,z} \wedge Pos_{t,i} \wedge Tape_{t,i,b}) \\ \Longrightarrow \bigvee_{(z',b',y)\in\delta(z,b)} (State_{t+1,z'} \wedge Pos_{t+1,i+dir(y)} \wedge Tape_{t+1,i,b'}) \end{array} \right)$$

- Zellen auf denen der Schreib-Lesekopf nicht steht, bleiben unverändert:

$$\bigwedge_{\substack{t\in\{0,\ldots,p(n)-1\},\\ i\in\{-p(n),\ldots,p(n)\},\\ b\in\Gamma}} \left((\neg Pos_{t,i} \wedge Tape_{t,i,b}) \Longrightarrow Tape_{t+1,i,b} \right)$$

Übergangsbedingungen

Daher:

$$\textit{Übergang} := \bigwedge_{\substack{t\in\{0,\ldots,p(n)-1\},\\ z\in Z,\\ i\in\{-p(n)+1,\ldots,p(n)-1\},\\ b\in\Gamma}} \left(\begin{array}{l} (State_{t,z} \wedge Pos_{t,i} \wedge Tape_{t,i,b}) \\ \quad \Longrightarrow \bigvee_{(z',b',y)\in\delta(z,b)} (State_{t+1,z'} \wedge Pos_{t+1,i+dir(y)} \wedge Tape_{t+1,i,b'}) \end{array} \right)$$

$$\wedge \left(\bigwedge_{\substack{t\in\{0,\ldots,p(n)-1\},\\ i\in\{-p(n),\ldots,p(n)\},\\ b\in\Gamma}} \left((\neg Pos_{t,i} \wedge Tape_{t,i,b}) \Longrightarrow Tape_{t+1,i,b} \right) \right)$$

Endbedingung

Ein Endzustand wird erreicht:

$$\mathit{Ende} := \bigvee_{z \in E, t \in \{0,\ldots,p(n)\}} \mathit{State}_{t,z}$$

$\mathcal{NP}$-Schwere von SAT

Wenn es eine Belegung I der Variablen von F gibt mit $I(F) = 1$, dann kann daraus ein akzeptierender Lauf für M auf Eingabe w konstruiert werden.

$\mathcal{NP}$-Schwere von SAT

Wenn es eine Belegung I der Variablen von F gibt mit $I(F) = 1$, dann kann daraus ein akzeptierender Lauf für M auf Eingabe w konstruiert werden.

Umgekehrt: Wenn M die Eingabe w akzeptiert, dann $z_0 w \vdash_M^r u z_e v$ mit $z_e \in E$ und $r \leq p(n)$. Der Lauf liefert eine Belegung I mit $I(F) = 1$.

$\mathcal{NP}$-Schwere von SAT

Wenn es eine Belegung I der Variablen von F gibt mit $I(F) = 1$, dann kann daraus ein akzeptierender Lauf für M auf Eingabe w konstruiert werden.

Umgekehrt: Wenn M die Eingabe w akzeptiert, dann $z_0 w \vdash_M^r uz_e v$ mit $z_e \in E$ und $r \leq p(n)$. Der Lauf liefert eine Belegung I mit $I(F) = 1$.

Damit gilt: F ist erfüllbar g.d.w. M akzeptiert w.

$\mathcal{NP}$-Schwere von SAT

Wenn es eine Belegung I der Variablen von F gibt mit $I(F) = 1$, dann kann daraus ein akzeptierender Lauf für M auf Eingabe w konstruiert werden.

Umgekehrt: Wenn M die Eingabe w akzeptiert, dann $z_0 w \vdash_M^r uz_e v$ mit $z_e \in E$ und $r \leq p(n)$. Der Lauf liefert eine Belegung I mit $I(F) = 1$.

Damit gilt: F ist erfüllbar g.d.w. M akzeptiert w.

F kann in (deterministischer) Polynomialzeit berechnet werden:

Teilformel	Größe (Anzahl an Variablenvorkommen)
Rand	$O(p(n)^3)$
Anfang	$O(p(n))$
Übergang	$O(p(n)^2)$
Ende	$O(p(n))$
F	$O(p(n)^3)$

Daher: $L \leq_p \text{SAT}$ für alle $L \in \mathcal{NP}$. D.h. SAT ist $\mathcal{NP}$-schwer. □

Satz von Cook

Insgesamt haben wir gezeigt:

Satz (Satz von Cook)

Das Erfüllbarkeitsproblem der Aussagenlogik (SAT) ist $\mathcal{NP}$-vollständig.