Informatik IV
Theoretische Informatik

Kapitel 13
NP-Vollständigkeit und der Satz von Cook

Sommersemester 2019

Dozent: Prof. Dr. J. Rothe

Wie schwer ist SAT?

Ziel: Nachweis, dass SAT eines der schwersten Probleme in NP ist: SAT ist das erste bekannte Beispiel eines NP-vollständigen Problems.

- Demnach kann SAT – mit bisher bekannten Techniken – nur in Exponentialzeit gelöst werden, ist also ein „störrisches“ Problem.
- Da man in der Praxis aber sehr an „möglichst effizienten“ Algorithmen für SAT interessiert ist, versucht man:
  - subexponentielle Algorithmen (d.h. Laufzeiten wie z. B. $2^{\sqrt{n}}$ oder $2^{\mathcal{O}\left(\sqrt{n \log n}\right)}$) oder
  - effiziente Heuristiken, die für die „praktisch relevanten“ Eingaben korrekt sind, oder
  - Exponentialzeit-Algorithmen, die den trivialen $\mathcal{O}(2^n)$-Algorithmus für SAT verbessern,

  zu finden.

Vergleich zweier Exponentialzeit-Algorithmen für SAT

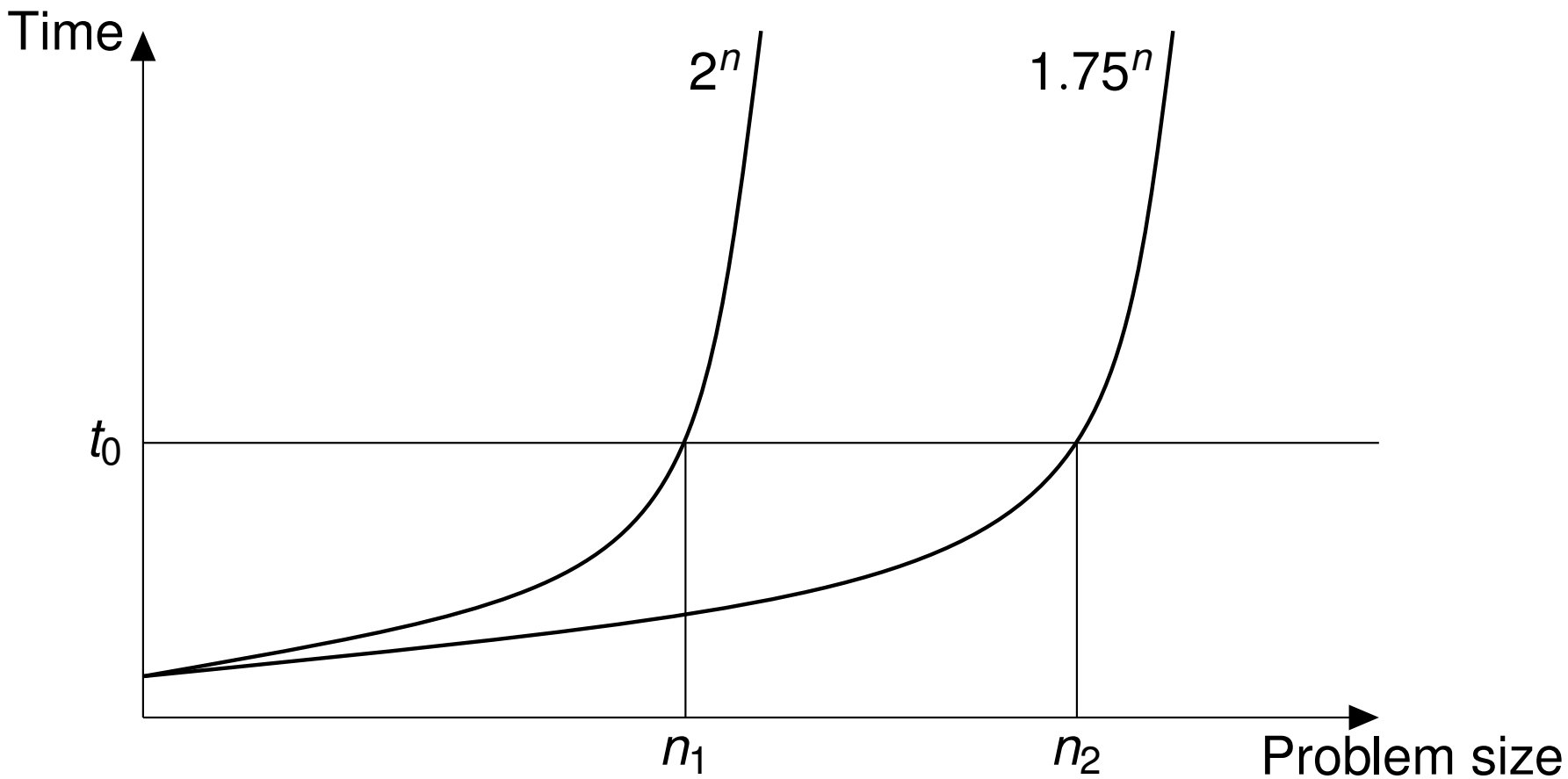

Abbildung: Verbesserter Exponentialzeit-Algorithmus für SAT

Vergleich zweier Exponentialzeit-Algorithmen für SAT

Motivation:

- Läuft der triviale Algorithmus T für SAT in der Zeit 2^n, dann kann ein c^n-Algorithmus A mit $c < 2$ größere Probleminstanzen als T in derselben Zeit bearbeiten.
- Ist $c = \sqrt{2}$, dann *kann A doppelt so große Eingaben in derselben Zeit wie der triviale Algorithmus bearbeiten*, denn

$$\left(\sqrt{2}\right)^{2n} = 2^n.$$

- Dieser Unterschied kann in der Praxis signifikant sein!
- **Vergleiche:** Ein 10-mal schnellerer Computer mit dem 2^n-Algorithmus, würde n_1 in Abbildung 1 nur geringfügig zu n_2 verbessern: $2^{n_2} = 10 \cdot 2^{n_1}$, d.h., $n_2 = n_1 + \log_2 10$.

NP-Vollständigkeit

Definition

- FP bezeichne die *Menge der in Polynomialzeit berechenbaren, totalen Funktionen* von Σ^* in Σ^* für ein Alphabet Σ. Formal:

$$\mathrm{FP} = \left\{ f : \Sigma^* \to \Sigma^* \;\middle|\; \begin{array}{l} f \text{ ist total und es gibt ein Polynom } p \in \mathbb{P}\mathrm{ol} \text{ und} \\ \text{eine deterministische TM } M, \text{ die } f \text{ berechnet,} \\ \text{und so dass } \mathrm{Time}_M(n) \leq p(n) \text{ für alle } n \end{array} \right\}.$$

- Seien $A, B \subseteq \Sigma^*$ Mengen. A ist *in Polynomialzeit many-one-reduzierbar* auf B (symbolisch: $A \leq_{\mathrm{m}}^{\mathrm{p}} B$), falls gilt:

$$(\exists f \in \mathrm{FP})\,(\forall x \in \Sigma^*)\,[x \in A \iff f(x) \in B].$$

NP-Vollständigkeit

Definition

- Eine Menge B heißt NP-*vollständig* (bzgl. $\leq_{\mathrm{m}}^{\mathrm{p}}$), falls gilt:
  1. $B \in \mathrm{NP}$.
  2. $(\forall A \in \mathrm{NP})\,[A \leq_{\mathrm{m}}^{\mathrm{p}} B]$. (Sprich: B ist NP-*hart* (oder NP-*schwer*).)

Bemerkung:

- Die NP-vollständigen Mengen sind also die schwersten Probleme in NP.
- NP-Härte wird intuitiv mit „Ineffizienz“ gleichgesetzt, denn bis heute ist für keine NP-harte Menge ein effizienter Algorithmus bekannt.

NP-Vollständigkeit

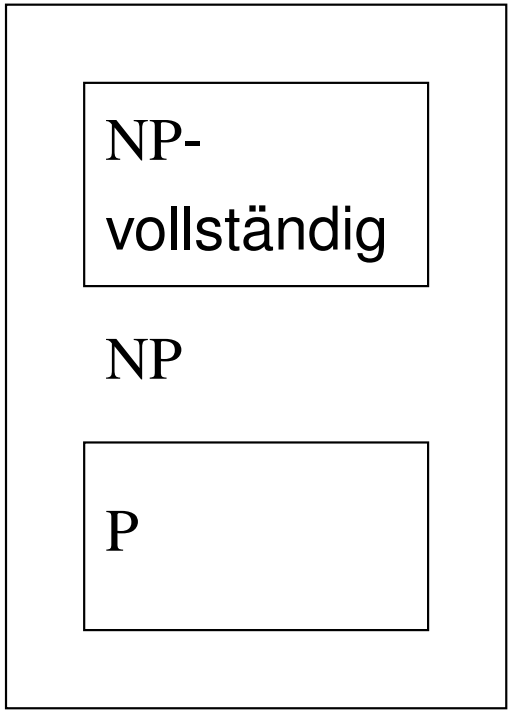

Abbildung: Beziehungen zwischen den Klassen NP, P und der Menge aller NP-vollständigen Probleme (sofern $\mathrm{P} \neq \mathrm{NP}$)

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Lemma

Es seien $A, B \subseteq \Sigma^*$ Mengen.

1. *Die Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$ ist reflexiv und transitiv, aber nicht antisymmetrisch.*

2. P *und* NP *sind $\leq_{\mathrm{m}}^{\mathrm{p}}$-abgeschlossen, d.h.,*

$$(A \leq_{\mathrm{m}}^{\mathrm{p}} B \wedge B \in \mathrm{P}) \quad \Rightarrow \quad A \in \mathrm{P}$$
$$(A \leq_{\mathrm{m}}^{\mathrm{p}} B \wedge B \in \mathrm{NP}) \quad \Rightarrow \quad A \in \mathrm{NP}.$$

Das heißt, Mitgliedschaft in P *und* NP *(„effiziente Lösbarkeit durch (nicht-)deterministische Maschinen“) vererbt sich bzgl. $\leq_{\mathrm{m}}^{\mathrm{p}}$ nach unten.*

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Lemma

3. NP-*Härte („Ineffizienz“) vererbt sich bzgl.* $\leq_{\mathrm{m}}^{\mathrm{p}}$ *nach oben, d.h.:*

$$(A \leq_{\mathrm{m}}^{\mathrm{p}} B \wedge A \text{ ist NP-hart}) \Rightarrow B \text{ ist NP-hart}.$$

4. $A \leq_{\mathrm{m}}^{\mathrm{p}} B \Rightarrow \overline{A} \leq_{\mathrm{m}}^{\mathrm{p}} \overline{B}$, *aber i.a. gilt nicht:* $A \leq_{\mathrm{m}}^{\mathrm{p}} \overline{A}$.

5. *Ist* B NP-*vollständig, dann gilt:* $B \in \mathrm{P} \iff \mathrm{NP} = \mathrm{P}$.

Beweis:

1. Übung.

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Beweis:

2.
   - Es sei $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ mittels $f \in \mathrm{FP}$.
   - Die deterministische Turingmaschine M_f berechne f in Polynomialzeit p.
   - Es sei $B \in \mathrm{P}$ mittels der deterministischen Turingmaschine M_B mit polynomieller Rechenzeit q.
   - Aus diesen beiden Turingmaschinen konstruieren wir eine Turingmaschine M_A, die A entscheidet:
     - Eine Instanz x für M_A wird zuerst in $f(x)$ umgewandelt und
     - dann wird $M_B(f(x)) = M_A(x)$ berechnet.
   - Die Laufzeit ist polynomiell: $p(|x|) + q(|f(x)|) \leq p(|x|) + q(p(|x|))$.
   - Falls $B \in \mathrm{NP}$, so sind die Maschinen nichtdeterministisch.

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Beweis:

3.
   - Da A NP-hart ist, gilt $\forall A' \in \mathrm{NP} : A' \leq_{\mathrm{m}}^{\mathrm{p}} A$.
   - Da nach Voraussetzung $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ gilt, folgt aus der Transitivität von $\leq_{\mathrm{m}}^{\mathrm{p}}$, dass
   $$\forall A' \in \mathrm{NP} : A' \leq_{\mathrm{m}}^{\mathrm{p}} B.$$
   - Damit folgt unmittelbar aus der Definition, dass B NP-hart ist.
4. Übung.

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Beweis:

5 $(\Rightarrow)$

- Es sei $B \in \mathrm{P}$.
- Da B NP-vollständig ist, gilt für alle Sprachen $A \in \mathrm{NP}$: $A \leq_{\mathrm{m}}^{\mathrm{p}} B$.
- Da $B \in \mathrm{P}$, gilt $\forall A \in \mathrm{NP} : A \in \mathrm{P}$ nach Teil (2).
- Da $\mathrm{P} \subseteq \mathrm{NP}$, folgt $\mathrm{P} = \mathrm{NP}$.

$(\Leftarrow)$

- Da B NP-vollständig ist, gilt $B \in \mathrm{NP}$, und
- da nach Voraussetzung $\mathrm{NP} = \mathrm{P}$ gilt, folgt $B \in \mathrm{P}$. ❑

Eigenschaften der Relation $\leq_{\mathrm{m}}^{\mathrm{p}}$

Bemerkung:

- Eigenschaft (5) des Lemmas sagt, dass ein polynomieller (effizienter) Algorithmus für (irgend)ein NP-hartes Problem impliziert, dass NP auf P kollabiert, und das P-NP-Problem wäre somit gelöst.
- Die weit verbreitete Annahme, dass $\mathrm{P} \neq \mathrm{NP}$ gilt, bedeutet, dass es *vermutlich* keinen effizienten Algorithmus für ein NP-hartes Problem gibt.

Der Satz von Cook

Theorem (Satz von Cook)

SAT *ist* NP*-vollständig.*

Beweis: Nach Definition sind die folgenden zwei Aussagen zu zeigen:

1. SAT ist in NP: siehe früheres Lemma.
2. SAT ist NP-hart.

- Sei A eine beliebige Menge in NP, und
- sei $M = (\Sigma, \Gamma, Z, \delta, \square, s_0, F)$ eine nichtdeterministische Turingmaschine, die A in Polynomialzeit akzeptiert.
- Wir nehmen an, dass M bei jeder Eingabe x der Länge n *genau* $p(n) \geq n$ Takte rechnet, für ein $p \in \mathbb{P}\mathrm{ol}$.

Der Satz von Cook

- Unser Ziel ist es, eine Reduktion $f \in \mathrm{FP}$ anzugeben, so dass für jedes $x \in \Sigma^*$ gilt:

$$x \in A \quad \Longleftrightarrow \quad f(x) = F_x \in \mathrm{SAT}, \tag{1}$$

  wobei F_x eine Boolesche Formel ist, deren Struktur und deren Variablen nun beschrieben werden.

- Sei ein Eingabewort $x = x_1 x_2 \cdots x_n$ mit $x_i \in \Sigma$ für alle i gegeben.
- Da M in der Zeit $p(n)$ arbeitet, kann sich der Kopf nicht weiter als um $p(n)$ Bandzellen nach links oder rechts bewegen.
- Nummeriere entsprechend die relevanten Bandzellen von $-p(n)$ bis $p(n)$.

Der Satz von Cook

$\cdots$	$\square$	$\cdots$	$\square$	x_1	x_2	$\cdots$	x_n	$\square$	$\cdots$	$\square$	$\cdots$
$\cdots$	$-p(n)$	$\cdots$	-1	0	1	$\cdots$	$n-1$	n	$\cdots$	$p(n)$	$\cdots$

Abbildung: Nummerierung der Bandzellen der NTM M bei Eingabe x

- Die obige Abbildung zeigt das Band von M zu Beginn der Rechnung mit der Startkonfiguration:
  - die Eingabesymbole $x_1 x_2 \cdots x_n$ stehen auf den Bandzellen 0 bis $n-1$,
  - der Kopf liest die Zelle mit Nr. 0 und
  - der Startzustand ist s_0.

Der Satz von Cook

- Wir konstruieren die Formel F_x, so dass (1) gilt.
- Die folgende Tabelle gibt die Variablen von F_x, den Bereich ihrer Indizes und ihre Bedeutung an. Es gibt drei Typen von Variablen:
  - $\text{state}_{t,s}$ repräsentiert den Zustand s von M in Takt t;
  - $\text{head}_{t,i}$ repräsentiert die Nummer i der Bandzelle, die der Kopf von M in Takt t liest;
  - $\text{tape}_{t,i,a}$ repräsentiert das Symbol $a \in \Gamma$ in der Bandzelle Nr. i in Takt t.

Der Satz von Cook

Variable	Indexbereich	Bedeutung
$\text{state}_{t,s}$	$t \in \{0,1,\ldots,p(n)\}$ $s \in Z$	$\text{state}_{t,s}$ ist wahr $\Longleftrightarrow$ M ist im Takt t in Zustand s
$\text{head}_{t,i}$	$t \in \{0,1,\ldots,p(n)\}$ $i \in \{-p(n),\ldots,p(n)\}$	$\text{head}_{t,i}$ ist wahr $\Longleftrightarrow$ M's Kopf liest im Takt t die Bandzelle Nr. i
$\text{tape}_{t,i,a}$	$t \in \{0,1,\ldots,p(n)\}$ $i \in \{-p(n),\ldots,p(n)\}$ $a \in \Gamma$	$\text{tape}_{t,i,a}$ ist wahr $\Longleftrightarrow$ Symbol a steht im Takt t in Bandzelle Nr. i

Tabelle: Die Booleschen Variablen von F_x und ihre Bedeutung in der Cook-Reduktion

Der Satz von Cook

F_x hat die Form:

$$F_x = S \wedge \ddot{U}_1 \wedge \ddot{U}_2 \wedge E \wedge K,$$

wobei diese Teilformeln von F_x den folgenden Zweck haben:

- S ist erfüllbar $\iff$ die Rechnung von $M(x)$ *startet* korrekt;
- $\ddot{U}_1$ ist erfüllbar $\iff$ die *Übergänge* der Rechnung von $M(x)$ sind in jedem Takt korrekt an den Bandfeldern mit Kopf;
- $\ddot{U}_2$ ist erfüllbar $\iff$ die *Übergänge* der Rechnung von $M(x)$ sind in jedem Takt korrekt an den Bandfeldern ohne Kopf;

Der Satz von Cook

- E ist erfüllbar $\iff$ die Rechnung von $M(x)$ *endet* korrekt (d.h., M akzeptiert x);
- K ist erfüllbar $\iff$ die allgemeine *Korrektheit* der Rechnung von $M(x)$ ist gegeben, d.h.:
  - in jedem Takt der Rechnung von $M(x)$ ist M in genau einem Zustand und der Kopf von M steht auf genau einem Feld und
  - in jedem Takt und für jede Bandposition gibt es genau ein Symbol auf diesem Feld des Bandes).

Der Satz von Cook

- Um diese Teilformeln von F_x formal zu beschreiben, seien die Zustandsmenge Z und das Arbeitsalphabet Γ von M gegeben durch:

$$\begin{aligned} Z &= \{s_0, s_1, \ldots, s_k\} \\ \Gamma &= \{\Box, a_1, a_2, \ldots, a_\ell\}. \end{aligned}$$

Dabei gilt $\Sigma \subseteq \Gamma$.

Der Satz von Cook

- Definiere die Teilformel K von F_x, die die *allgemeine Korrektheit* beschreibt, durch:

$$K = \bigwedge_{0 \le t \le p(n)} [D_1(\text{state}_{t,s_0}, \text{state}_{t,s_1}, \ldots, \text{state}_{t,s_k}) \wedge D_2(\text{head}_{t,-p(n)}, \text{head}_{t,-p(n)+1}, \ldots, \text{head}_{t,p(n)}) \wedge \bigwedge_{-p(n) \le i \le p(n)} D_3(\text{tape}_{t,i,\square}, \text{tape}_{t,i,a_1}, \ldots, \text{tape}_{t,i,a_\ell})], \quad (2)$$

wobei die Struktur der drei Teilformeln D_i von K in (2) mittels des folgenden Lemmas spezifiziert wird. Für $i \in \{1, 2, 3\}$ gilt:

(a) D_i ist genau dann wahr, wenn *genau* eine der Variablen von D_i wahr ist, und

(b) die Größe von D_i – und folglich auch die Größe von K – ist polynomiell in n.

Der Satz von Cook

Lemma

Für jedes m gibt es eine Boolesche Formel D mit m Variablen, so dass gilt:

1. $D(v_1, v_2, \ldots, v_m)$ *ist wahr* $\iff$ *genau eine Variable* v_i *ist wahr;*
2. *die Größe von D (d.h., die Zahl der Vorkommen von Variablen in D) ist in* $\mathcal{O}(m^2)$.

Der Satz von Cook

Beweis:

- Für festes $m \geq 1$ sei die Formel D definiert durch

$$D(v_1, v_2, \ldots, v_m) \quad = \quad \underbrace{\left(\bigvee_{i=1}^{m} v_i \right)}_{D_{\geq}} \wedge \underbrace{\left(\bigwedge_{j=1}^{m-1} \bigwedge_{k=j+1}^{m} \neg(v_j \wedge v_k) \right)}_{D_{\leq}} .$$

- Die beiden Teilformeln $D_{\geq}$ und $D_{\leq}$ von D haben die folgenden Eigenschaften:

$$D_{\geq}(v_1, v_2, \ldots, v_m) \text{ ist wahr} \quad \Leftrightarrow \quad \text{mindestens eine Variable } v_i \text{ ist wahr;} \tag{3}$$

$$D_{\leq}(v_1, v_2, \ldots, v_m) \text{ ist wahr} \quad \Leftrightarrow \quad \text{höchstens eine Variable } v_i \text{ ist wahr.} \tag{4}$$

Der Satz von Cook

- Die Äquivalenz (3) ist offensichtlich.
- Dass auch die Äquivalenz (4) gilt, ergibt sich so:

$$
\begin{aligned}
& D_{\leq}(v_1, v_2, \ldots, v_m) \\
= \; & (\neg v_1 \vee \neg v_2) \wedge (\neg v_1 \vee \neg v_3) \wedge \cdots \wedge (\neg v_1 \vee \neg v_m) \\
& \wedge (\neg v_2 \vee \neg v_3) \wedge \cdots \wedge (\neg v_2 \vee \neg v_m) \\
& \qquad\qquad \ddots \qquad\qquad \vdots \\
& \wedge (\neg v_{m-1} \vee \neg v_m).
\end{aligned}
$$

- (3) und (4) zusammen implizieren, dass $D(v_1, v_2, \ldots, v_m)$ genau dann wahr ist, wenn genau eine Variable v_i wahr ist.
- Offenbar ist die Größe von D in $\mathcal{O}(m^2)$. ❑

Der Satz von Cook

- Um den Beweis des Satzes fortzusetzen, definiere die Teilformel S von F_x, die für Takt $t = 0$ den korrekten Beginn der Rechnung $M(x)$ beschreibt (siehe die obige Abbildung), durch

$$S \;=\; \text{state}_{0,s_0} \wedge \text{head}_{0,0} \wedge \bigwedge_{i=-p(n)}^{-1} \text{tape}_{0,i,\square} \wedge \bigwedge_{i=0}^{n-1} \text{tape}_{0,i,x_{i+1}} \wedge \bigwedge_{i=n}^{p(n)} \text{tape}_{0,i,\square}.$$

Der Satz von Cook

- Definiere die Teilformel $\ddot{U}_1$ von F_x, die den korrekten Übergang von Takt t zu Takt $t+1$ für die aktuell von M's Kopf gelesenen Bandzellen beschreibt, durch

$$\ddot{U}_1 = \bigwedge_{t,s,i,a} \Big(\big(\text{state}_{t,s} \wedge \text{head}_{t,i} \wedge \text{tape}_{t,i,a} \big) \Longrightarrow \bigvee_{\substack{\hat{s} \in Z, \hat{a} \in \Gamma, y \in \{-1,0,1\} \\ \text{mit } (\hat{s},\hat{a},y) \in \delta(s,a)}} \big(\text{state}_{t+1,\hat{s}} \wedge \text{head}_{t+1,i+y} \wedge \text{tape}_{t+1,i,\hat{a}} \big) \Big) ,$$

wobei δ die Überführungsfunktion von M ist und $y \in \{-1,0,1\}$ die Kopfbewegung repräsentiert.

Der Satz von Cook

- Definiere die Teilformel $\ddot{U}_2$ von F_x, die den korrekten Übergang von Takt t zu Takt $t+1$ für die aktuell von M's Kopf *nicht* gelesenen Bandzellen beschreibt, durch

$$\ddot{U}_2 \quad = \quad \bigwedge_{t,i,a} \left(\left(\neg \mathrm{head}_{t,i} \wedge \mathrm{tape}_{t,i,a} \right) \Longrightarrow \mathrm{tape}_{t+1,i,a} \right).$$

- Definiere die Teilformel E von F_x, die das korrekte Ende der Berechnung $M(x)$ beschreibt und überprüft, ob M die Eingabe x akzeptiert:

$$E \quad = \quad \bigvee_{s \in F} \mathrm{state}_{p(n),s},$$

wobei F die Menge der Endzustände von M ist.

Der Satz von Cook

- Damit ist die Reduktion f beschrieben.
- Analysiert man die Struktur der Formel $f(x) = F_x$ und benutzt das obige Lemma, so folgt $f \in \mathrm{FP}$.
- Es bleibt zu zeigen, dass (1) gilt:

$$x \in A \iff f(x) = F_x \text{ ist erfüllbar.}$$

Der Satz von Cook

($\Rightarrow$)

- Sei $x \in A$.
- Dann gibt es einen akzeptierenden Berechnungspfad α von $M(x)$.
- Weist man nun jeder Variablen von F_x gemäß ihrer beabsichtigten Bedeutung der Variablen aus der obigen Tabelle einen α entsprechenden Wahrheitswert zu, dann erfüllt diese Belegung jede Teilformel von F_x, also auch F_x selbst.
- Somit ist $F_x \in \mathrm{SAT}$.

Der Satz von Cook

($\Leftarrow$)

- Sei nun $F_x \in \mathrm{SAT}$.
- Dann gibt es eine Belegung τ, die F_x erfüllt.
- Die Variablen $\mathrm{state}_{t,s}$, $\mathrm{head}_{t,i}$ und $\mathrm{tape}_{t,i,a}$ von F_x können gemäß τ als eine Folge $K_0, K_1, \ldots, K_{p(n)}$ von Konfigurationen von $M(x)$ entlang eines Berechnungspfades interpretiert werden.
- Da $\tau(F_x) = 1$, muss τ jede Teilformel von F_x erfüllen.

Der Satz von Cook

- Somit gilt:
  - $\tau(S) = 1$ impliziert, dass K_0 die Startkonfiguration von $M(x)$ ist,
  - $\tau(\ddot{U}_1) = \tau(\ddot{U}_2) = \tau(K) = 1$ impliziert, dass $K_{t-1} \vdash_M K_t$ für jedes t mit $1 \leq t \leq p(n)$ gilt, und
  - $\tau(E) = 1$ impliziert, dass $K_{p(n)}$ eine akzeptierende Endkonfiguration von $M(x)$ ist.
- Folglich ist $x \in A$.

Die Äquivalenz (1) ist gezeigt.

Der Satz von Cook

- Es bleibt zu zeigen, dass F_x in polynomieller Zeit berechenbar ist.
- Der Aufwand zu Erzeugung von F_x ist offenbar linear in der Länge von F_x.

$$
\begin{array}{lcl}
|S| & \in & \mathcal{O}(p(n)) \\
|\ddot{U}_1| & \in & \mathcal{O}((p(n))^2) \\
|\ddot{U}_2| & \in & \mathcal{O}((p(n))^2) \\
|E| & \in & \mathcal{O}(1) \\
|K| & \in & \mathcal{O}((p(n))^3) \\
\hline
|F_x| & \in & \mathcal{O}((p(n))^3)
\end{array}
$$

Damit ist der Satz bewiesen. ❑

Der Satz von Cook

Da die im Beweis von Satz 5 konstruierte Boolesche Formel F_x sogar in KNF ist bzw. ohne zu große Aufblähung der Formel in KNF gebracht werden kann, erhalten wir die folgende Folgerung.

Corollary

KNF-SAT *ist* NP-*vollständig.*

Der Satz von Cook

Bemerkung:

- Die bekannten deterministischen Algorithmen zur Berechnung von SAT (z.B. systematisches Durchprobieren aller Belegungen) haben alle eine Zeitkomplexität von $2^{\mathcal{O}(n)}$.
- Da nach dem Satz von Cook jede Sprache $L \in \mathrm{NP}$ auf SAT reduziert werden kann ($\forall L \in \mathrm{NP} : L \leq_{\mathrm{m}}^{\mathrm{p}} \mathrm{SAT}$), kann die deterministische Zeitkomplexität von L nach oben durch $2^{p(n)}$ für ein Polynom p abgeschätzt werden.
- Das heißt,

$$\mathrm{NP} \subseteq \bigcup_{p \in \mathbb{P}\mathrm{ol}} \mathrm{TIME}(2^{p(n)}).$$