

Komplexitätstheorie

Der Satz von Cook/Levin

Martin Dietzfelbinger

4. Mai 2009

Definitionen 1.2.1 und 1.3.1

$$\begin{aligned} \mathbf{NP} &:= \bigcup_{k \geq 1} \text{NTIME}(n^k) \\ &= \{L_M \mid \exists \text{ Polynom } p(n): M \text{ ist } p(n)\text{-zeitbeschr. NTM}\}. \end{aligned}$$

$L_1 \subseteq \Sigma_1^*$ heißt **polynomialzeit-reduzierbar** auf $L_2 \subseteq \Sigma_2^*$, wenn es ein $f \in \mathbf{FP}$ gibt, $f: \Sigma_1^* \rightarrow \Sigma_2^*$, mit

$$\forall x \in \Sigma_1^*: x \in L_1 \Leftrightarrow f(x) \in L_2.$$

NP-vollständige Sprachen

Definition 1.3.2

(a) L heißt **NP-vollständig** (engl. „NP-complete“), falls

- (i) $L \in \mathbf{NP}$
- (ii) für alle $L' \in \mathbf{NP}$ gilt $L' \leq_p L$
(„ L ist NP-schwer“ [„NP-hard“, „NP-hart“]).

(b) $\mathbf{NPC} := \{L \mid L \text{ Sprache, } L \text{ NP-vollständig}\}$.

Satz

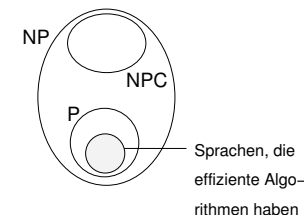
$$\mathbf{NPC} \cap \mathbf{P} \neq \emptyset \Leftrightarrow \mathbf{P} = \mathbf{NP}. \quad (1)$$

$$\mathbf{NPC} \cap \mathbf{P} = \emptyset \Leftrightarrow \mathbf{P} \neq \mathbf{NP}. \quad (2)$$

Vermutung: $\mathbf{NPC} \cap \mathbf{P} = \emptyset$ und $\mathbf{P} \neq \mathbf{NP}$.

(Offenes Problem. Geeignet als Hypothese für die Alltagsarbeit.)

Vermutete Situation:



1.4 Der Satz von Cook/Levin

Definition

Eine aussagenlogische Formel φ heißt **erfüllbar**, wenn $v(\varphi) = 1$ für **mindestens eine** Belegung v gilt. (Ergebnisspalte in der W.-tafel enthält mindestens eine 1.) Andernfalls heißt φ **unerfüllbar**.

Erfüllbarkeitsproblem für aussagenlogische Formeln:

Eingabe: Aussagenlogische Formel φ

Frage: Ist φ erfüllbar?

Spezielle $\{\vee, \wedge, \neg\}$ -Formeln:

konjunktive Normalform, KNF:

Konjunktion von Disjunktionen

Definition

- (a) **Boolesche Variable:** $X_0, X_1, X_2, X_3, \dots$
- (b) **Literale:** $X_0, \bar{X}_0, X_1, \bar{X}_1, X_2, \bar{X}_2, \dots$
(\bar{X}_i ist synonym mit $\neg X_i$)
- (c) **Klausel:** $(l_1 \vee \dots \vee l_s)$, wobei l_1, \dots, l_s Literale sind.
- (d) **Formel in KNF:** $C_1 \wedge \dots \wedge C_r$, wobei C_1, \dots, C_r Klauseln sind.

Beispiele:

Variable: X_1, X_{10}, X_{2009} .

Kodierung: [1], [1010], [11111011001].

Literale: X_1, \bar{X}_{10} .

Kodierung: [1], \neg [1010].

Klauseln: $(X_1 \vee X_3 \vee X_1 \vee \bar{X}_8)$ und (\bar{X}_2)

KNF-Formel: $(X_0) \wedge (X_1 \vee X_3 \vee \bar{X}_5) \wedge (\bar{X}_0 \vee \bar{X}_3)$

Kodierung: $([0]) \wedge ([1] \vee [11] \vee \neg[101]) \wedge (\neg[0] \vee \neg[11])$.

Definition 1.4.1(a)

Eine **Belegung** der Booleschen Variablen ist eine Abbildung

$$v : \{X_0, X_1, X_2, \dots\} \rightarrow \{0, 1\}.$$

Fortsetzung von v auf Literale, Klauseln, KNF-Formeln:

$$v(\bar{X}_i) = 1 - v(X_i) \quad (\text{Negation})$$

$$v((l_1 \vee \dots \vee l_s)) = \max\{v(l_1), \dots, v(l_s)\} \quad (\text{Oder})$$

$$v(C_1 \wedge \dots \wedge C_r) = \min\{v(C_1), \dots, v(C_r)\} \quad (\text{Und})$$

$v(C_1 \wedge \dots \wedge C_r) = 1$ („wahr“)

\Leftrightarrow jede Klausel C_i von $\varphi = C_1 \wedge \dots \wedge C_r$ enthält ein Literal, das unter v wahr wird.

($v(X_i)$ ist relevant nur, wenn X_i in φ vorkommt.)

Definition 1.4.1(b)

Eine KNF-Formel $\varphi = C_1 \wedge \dots \wedge C_r$ heißt **erfüllbar**,

falls es eine Belegung v gibt, so dass $v(\varphi) = 1$ ist.

KNF-Formeln sind über dem Alphabet $\Sigma = \{[,], (,), \wedge, \vee, \neg, 0, 1\}$ kodiert. Wir machen keinen Unterschied zwischen Formel und Kodierung.

Die Sprache

$$L_{KNF} = \{\varphi \in \{[,], (,), \wedge, \vee, \neg, 0, 1\}^* \mid \varphi \text{ ist KNF-Formel}\}$$

ist regulär.

(Syntaxcheck: DFA.)

Definition 1.4.1(c)

$$L_{SAT} = \{\varphi \mid \varphi \text{ ist erfüllbare KNF-Formel}\}.$$

Satz 1.4.2 (Satz von Cook/Levin, 1970)

L_{SAT} ist **NP**-vollständig.

Konsequenz: Wenn $\mathbf{P} \neq \mathbf{NP}$, besitzt das Erfüllbarkeitsproblem für KNF-Formeln keinen Polynomialzeitalgorithmus.

- Dreh- und Angelpunkt der NP-Vollständigkeits-Theorie
- Erfüllbarkeit von Formeln/Schaltkreisen praktisch relevant

1. Teil: $L_{SAT} \in \mathbf{NP}$.

Beweis: Beschreibe L_{SAT} als **Suchproblem**.

Zu KNF-Formel $\varphi = C_1 \wedge \dots \wedge C_r$ mit Variablen

$X_{i_1}, \dots, X_{i_m}, 1 \leq i_1 < \dots < i_m$

suche Belegung $v: \{X_{i_1}, \dots, X_{i_m}\} \rightarrow \{0, 1\}$

für X_{i_1}, \dots, X_{i_m} , so dass

$$v(\varphi) = 1$$

ist.

Klar: $m \leq |\varphi|$ und

$$\mathcal{R}_{SAT} = \{(\varphi, v) \mid v \in \text{SOL}(\varphi)\} \in \mathbf{P}.$$

Nach Bem. 1.2.3: $L_{SAT} \in \mathbf{NP}$.

2. Teil: L_{SAT} ist **NP**-schwer.

Wir müssen zeigen:

Für jede Sprache $L \in \mathbf{NP}$ gilt $L \leq_p L_{SAT}$.

Sei dazu L eine beliebige (ab jetzt feste) Sprache in **NP**.

Zu zeigen: $L \leq_p L_{SAT}$.

Was wissen wir über L ?

?

Es gibt eine NTM M' mit $L = L_{M'}$,

M' polynomiell zeitbeschränkt.

NTM M' mit $L = L_{M'}$, M' polynomiell zeitbeschränkt.

Baue M' um zu NTM M mit $L_M = L_{M'}$ mit

1. M hat **ein Band** (quadriert Laufzeit).
2. Der Kopf betritt **nie Zellen links der Eingabe**.
3. Anstatt zu halten, mit $\delta(q, a) = \emptyset$, bleibt M mit $\delta(q, a) = \{(q, a, N)\}$ in dieser Konfiguration stehen; wenn $q \in F$, nur Übergang $\delta(q, a) = \{(q, a, N)\}$.
4. M hat polynomielle Laufzeit: Es gibt c, k mit: für jedes $x \in \Sigma^*$ ist nach $t_M(x) \leq c|x|^k$ Schritten eine Haltekonfiguration wie in 3. erreicht.

Nun gilt für jedes $x \in \Sigma^*$:

$$x \in L \Leftrightarrow$$

es gibt eine Berechnung von M auf x ,

die nach $T = c|x|^k$ Schritten eine Konfiguration mit einem Zustand $q \in F$ erreicht hat.

Eine Berechnung ist eine Konfigurationenfolge

$$\text{init}_M(x) = k_0 \vdash k_1 \vdash \dots \vdash k_T.$$

Schema:

Schr. t	Zust. q	Kopfp. p	Band											
			1	2	3	...	n	$n+1$...	p'	...	T	$T+1$	
0	q_0	1	a_1	a_2	a_3	...	a_n	B	...	B	...	B	B	
1	q_5	2	d	a_2	a_3	...	a_n	B	...	B	...	B	B	
2	q_8	3	d	e	a_3	...	a_n	B	...	B	...	B	B	
3	q_6	3	d	e	b	...	a_n	B	...	B	...	B	B	
4	q_{10}	2	d	e	m	...	a_n	B	...	B	...	B	B	
...	:											
$t-1$	q	p'	...						a	...				
t	q'	$p'+1$...						a'	...				
...	:											
T	q_h	p_h	...											

Bandzellen von M mit 1, 2, 3, ... nummeriert.

In keiner Berechnung von M auf x , $|x| = n$, kann der Kopf eine Zelle $> T + 1$ erreichen.

Eine **Berechnung** entspricht einer „legalen“ (von δ erlaubten) Beschriftung dieses Schemas,

so dass der Zustand q_h in Zeile T in F liegt.

Wir beschreiben eine Reduktionsfunktion $\Phi \in \mathbf{FP}$ mit

$$L \leq_p L_{\text{SAT}} \text{ mittels } \Phi.$$

D.h.:

$$\Phi : \Sigma^* \rightarrow \{\varphi \mid \varphi \text{ KNF-Formel}\}, \quad x \mapsto \varphi_x$$

mit $\Phi \in \mathbf{FP}$, und

$$\forall x \in \Sigma^*: x \in L \Leftrightarrow \varphi_x \text{ ist erfüllbar.}$$

Dazu: φ_x so, dass gilt:

$$\exists \text{ akzeptierende Berechnung von } M \text{ auf } x$$

$$\Leftrightarrow$$

$$\exists \text{ Belegung } v \text{ mit } v(\varphi_x) = 1.$$

Idee: Eine (und jede) **erfüllende Belegung** von φ_x „**beschreibt**“ eine **akzeptierende Berechnung** von M auf x .

Dazu wird φ_x so gebaut, dass **jede Variable über ein Detail einer Berechnung spricht**.

Eine Belegung **aller** Variablen spricht dann über **alle Details** einer Berechnung.

Drei Typen von Variablen:

(a) $X_{t,p,a}$

(b) $Y_{t,q}$

(c) $Z_{t,p}$

$$0 \leq t \leq T, 1 \leq p \leq T+1, a \in \Gamma, q \in Q$$

(a) $X_{t,p,a}, 0 \leq t \leq T, 1 \leq p \leq T+1, a \in \Gamma$

Interpretation:

„Nach Schritt t steht in Zelle p das Zeichen a “

Diese Variable kann von v mit $1 \hat{=} wahr$ oder $0 \hat{=} falsch$ belegt werden.

Entsprechend sind Berechnungen gemeint, in denen nach Schritt t in Zelle p das Zeichen a steht oder eben nicht.

Beispiel: Wenn $v(X_{2,1,a}) = 1$ und $v(X_{2,1,b}) = 1$,

dann kann v **keine wirkliche Berechnung** beschreiben, weil in Zelle 2 nach Schritt 1 nicht sowohl a als auch b stehen kann.

Solche Belegungen dürfen nicht erfüllend sein!

Wenn nun φ_x die Klausel

$$(\overline{X}_{2,1,a} \vee \overline{X}_{2,1,b})$$

enthält, dann sind solche Belegungen nicht erfüllend.

Brauche dies für jedes Paar von Bandbuchstaben.

Weiter:

Irgendein Buchstabe **muss** nach Schritt 1 in Zelle 2 stehen.

Durch eine Klausel

$$\left(\bigvee_{a \in \Gamma} X_{1,2,a} \right)$$

in φ_x wird erreicht, dass mindestens eine dieser Variablen mit 1 belegt sein muss, wenn φ_x durch v erfüllt wird.

Abkürzung:

$$\left(\bigvee_{a \in \Gamma} \psi_a \right) = (\psi_{a_1} \vee \dots \vee \psi_{a_s}),$$

wo $\Gamma = \{a_1, \dots, a_s\}$

Analog: $\bigwedge_{a \in \Gamma} C_a$ usw.

Brauchen dies für jeden Schritt t , $0 \leq t \leq T + 1$,
jede Zelle p , $1 \leq p \leq T + 1$.

(1) Teilformel φ_1 :

$$\varphi_1 \equiv \bigwedge_{0 \leq t \leq T} \bigwedge_{1 \leq p \leq T+1} \left(\left(\bigvee_{a \in \Gamma} X_{t,p,a} \right) \wedge \bigwedge_{\substack{a, a' \in \Gamma \\ a \neq a'}} (\overline{X}_{t,p,a} \vee \overline{X}_{t,p,a'}) \right)$$

Beobachtung: Wenn v eine erfüllende Belegung für φ_1 ist, dann liefert die Standard-Interpretation dieser Belegung für jede Zelle und jeden Zeitpunkt genau einen Bandbuchstaben.

Schr. t	Zust. q	Kopfp. p	Band										
			1	2	3	...	n	$n+1$...	p'	...	T	$T+1$
0	q_0	1	a	b	a	...	c	d	...	a	...	e	a
1	q_5	2	c	a	c	...	e	a	...	a	...	b	d
2	q_8	3	e	b	c	...	b	b	...	d	...	a	e
3	q_6	3	d	e	b	...	a	a	...	c	...	a	c
4	q_{10}	2	d	e	m	...	c	a	...	d	...	e	e
...										
$t-1$	q	p'	...							c	...		
t	q'	$p'+1$...							a	...		
...										
T	q_h	p_h	...										

Jede Beschriftung der Bandzellen im Schema liefert eine Belegung für die $Z_{...}$ -Variablen, die φ_1 erfüllt.

Jede erfüllende Belegung für φ_1 liefert eine Beschriftung der Bandzellen-Abteilung des Schemas.

Hat mit Berechnungen leider noch wenig zu tun.

Brauchen noch mehr Teilformeln!

Gesamtstruktur:

$$\varphi_x \equiv \varphi_1 \wedge \varphi_2 \wedge \dots \wedge \varphi_8.$$

Weitere Variablen:

(b) $Y_{t,q}$, $0 \leq t \leq T$ und $q \in Q$.

Interpretation:

„Nach Schritt t ist M in Zustand q “

(2) Teilformel φ_2 :

$$\bigwedge_{0 \leq t \leq T} \left(\left(\bigvee_{q \in Q} Y_{t,q} \right) \wedge \bigwedge_{\substack{q, q' \in Q \\ q \neq q'}} (\overline{Y}_{t,q} \vee \overline{Y}_{t,q'}) \right)$$

Erfüllende Belegung für φ_2 ($Y_{...}$ -Variablen)

$\hat{=}$ Beschriftung der Zustands-Spalte im Schema

Weitere Variablen:

(c) $Z_{t,p}$, $0 \leq t \leq T$ und $1 \leq p \leq T+1$.

Interpretation:

„Nach Schritt t steht der Kopf von M in Zelle p “

(3) Teilformel φ_3 :

$$\bigwedge_{0 \leq t \leq T} \left(\left(\bigvee_{1 \leq p \leq T+1} Z_{t,p} \right) \wedge \bigwedge_{\substack{1 \leq p, p' \leq T+1 \\ p \neq p'}} (\bar{Z}_{t,p} \vee \bar{Z}_{t,p'}) \right)$$

Erfüllende Belegungen für φ_3 ($Z_{...}$ -Variablen)

$\hat{=}$ Beschriftung der Kopfpositions-Spalte im Schema

Schr. t	Zust. q	Kopfp. p	Band											
			1	2	3	...	n	$n+1$...	p'	...	T	$T+1$	
0	q_0	10	a	b	a	...	c	d	...	a	...	e	a	
1	q_5	2	c	a	c	...	e	a	...	a	...	b	d	
2	q_8	6	e	b	c	...	b	b	...	d	...	a	e	
3	q_6	9	d	e	b	...	a	a	...	c	...	a	c	
4	q_{10}	25	d	e	m	...	c	a	...	d	...	e	e	
...											
$t-1$	$q_{i_{t-1}}$	3	...								c	...		
t	q_{i_t}	9	...								a	...		
...											
T	q_{i_T}	p_h	...											

Erfüllende Belegung für $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$

($X_{...}$ -, $Y_{...}$ -, und $Z_{...}$ -Variablen)

$\hat{=}$ **Beschriftung** aller Zellen des Schemas.

Immer noch kein Bezug zu Berechnungen!

(4) Teilformel φ_4 : **Startkonfiguration**

$$\bigwedge_{1 \leq p \leq n} (X_{0,p,a_p}) \wedge \bigwedge_{n+1 \leq p \leq T} (X_{0,p,B}) \wedge (Y_{0,q_0}) \wedge (Z_{0,1})$$

Für v mit $v(\varphi_1 \wedge \varphi_2 \wedge \varphi_3) = 1$ gilt:

$v(\varphi_4) = 1 \Leftrightarrow$ die in v beschriebene Beschriftung des Schemas enthält in Zeile 0 die Startkonfiguration für x .

(Bandinschrift $x = a_1 \cdots a_n B \cdots B$, Startzustand q_0 , Kopf in Zelle 1)

(5) Teilformel φ_5 : **Endkonfiguration**

$$\varphi_5 \equiv \left(\bigvee_{q \in F} Y_{T,q} \right)$$

Für v mit $v(\varphi_1 \wedge \varphi_2 \wedge \varphi_3) = 1$ gilt:

$v(\varphi_5) = 1 \Leftrightarrow$ die in v beschriebene Beschriftung des Schemas enthält in Zeile T eine akzeptierende Konfiguration für x .

(Zustand $q_{i_T} \in F$)

Es fehlt noch:

Von Zeile $t-1$ zu Zeile t führt ein legaler Schritt von M .

(6) Teilformel φ_6 :

Wo der Kopf nicht ist, bleibt Bandbuchstabe erhalten

$$\bigwedge_{1 \leq t \leq T} \bigwedge_{1 \leq p \leq T+1} \bigwedge_{a \in \Gamma} (Z_{t-1,p} \vee \bar{X}_{t-1,p,a} \vee X_{t,p,a})$$

Für v mit $v(\varphi_1 \wedge \varphi_2 \wedge \varphi_3) = 1$ gilt:

$$v(\varphi_6) = 1$$

\Leftrightarrow

$[v(Z_{t-1,p}) = 0 \wedge v(X_{t-1,p,a}) = 1 \Rightarrow v(X_{t,p,a}) = 1]$, für alle t, p und a .

Noch eine (vierte) Sorte von Variablen:

Wir sagen: (q, a, q', a', D) ist „**Regel**“ in der Übergangsfunktion δ von M ,

wenn $(q', a', D) \in \delta(q, a)$.

Wenn M in Zustand q ist und der Kopf Buchstaben a „sieht“, dann *darf* M Buchstaben a' schreiben, in Zustand q' gehen, und den Kopf in Richtung $D \in \{L, R, N\}$ bewegen.

Neue Variable:

$$U_{t,q,a,q',a',D} = U_{t,r}, \text{ für } 1 \leq t \leq T, \\ r = (q, a, q', a', D) \text{ Regel}$$

Interpretation: $v(U_{t,r}) = 1$ soll bedeuten, dass in Schritt t der Berechnung Regel $r = (q, a, q', a', D)$ benutzt wird.

(7) Teilformel φ_7 :

In jedem Schritt genau eine Regel

$$\bigwedge_{1 \leq t \leq T} \left(\left(\bigvee_{r \text{ Regel}} U_{t,r} \right) \wedge \bigwedge_{\substack{r, \bar{r} \text{ Regeln} \\ r \neq \bar{r}}} (\bar{U}_{t,r} \vee \bar{U}_{t,\bar{r}}) \right)$$

Erfüllende Belegung für φ_7 (nur die U_{\dots} -Variablen)

$\hat{=}$ Folge r_1, \dots, r_T von Regeln.

(8) Teilformel φ_8 :

Übergang von Zeile $t-1$ zu Zeile t gemäß Regel r_t

$$\bigwedge_{1 \leq t \leq T} \bigwedge_{1 \leq p \leq T+1} \bigwedge_{\substack{r=(q,a,q',a',D) \\ \text{Regel}}} \left(\begin{array}{l} (\bar{U}_{t,r} \vee \bar{Z}_{t-1,p} \vee X_{t-1,p,a}) \quad \wedge \\ (\bar{U}_{t,r} \vee Y_{t-1,q}) \quad \wedge \\ (\bar{U}_{t,r} \vee \bar{Z}_{t-1,p} \vee X_{t,p,a'}) \quad \wedge \\ (\bar{U}_{t,r} \vee Y_{t,q'}) \quad \wedge \\ (\bar{U}_{t,r} \vee \bar{Z}_{t-1,p} \vee Z_{t,p+d(D)}) \end{array} \right)$$

$$\text{Dabei: } d(D) = \left\{ \begin{array}{ll} -1 & \text{für } D = L \\ 0 & \text{für } D = N \\ 1 & \text{für } D = R \end{array} \right\}$$

Feststellungen:

(*) Es sei v eine Belegung, die $\varphi_x \equiv \varphi_1 \wedge \dots \wedge \varphi_8$ erfüllt.

Dann liefern die Interpretationen der Variablen, die von v mit 1 belegt werden, eine Beschriftung des Schemas, das einer akzeptierenden Berechnung von M auf x entspricht.

(**) Eine akzeptierende Berechnung k_0, k_1, \dots, k_T von M auf x sei gegeben.

Wenn man eine Belegung v für die Variablen so definiert, dass diejenigen Variablen den Wert 1 erhalten, deren Interpretation in k_0, k_1, \dots, k_T zutrifft, die anderen den Wert 0,

dann ist v eine erfüllende Belegung für φ .

Fazit: Für φ_x gilt:

\exists akzeptierende Berechnung von M auf x

\Leftrightarrow

\exists Belegung v mit $v(\varphi_x) = 1$.

Wie gewünscht.

Damit ist

$$\Phi: x \mapsto \varphi_x$$

eine Reduktionsfunktion von L_M auf L_{SAT} .

Details:

- Muss die $X_{...}$ -, $Y_{...}$ -, $Z_{...}$ - und $U_{...}$ -Variablen als Standard-Variablen $X_j = [\text{bin}(j)]$ kodieren.

Beispiel: $X_{t,p,a} = [100 \text{ bin}(t) \text{ bin}(p) \text{ bin}(a)]$.

- Umfang von φ_x : Jeder Formelteil enthält $O(T^2)$ oder $O(T^3)$ (Fall (3)) Literale und Verknüpfungszeichen.

Darstellung einer Variablen: $O(\log T) = O(\log |x|)$ Zeichen.

$\Rightarrow |\varphi_x|$ polynomiell in $|x|$, weil $T = c|x|^k$.

- φ_x ist aus $|x|$ „leicht“ in polynomieller Zeit zu konstruieren. (Details: selbst überlegen.) Also: $\Phi \in \mathbf{FP}$.

Insgesamt:

$$L_M \leq_p L_{SAT} \text{ mittels } \Phi$$

L_M war L für $L \in \mathbf{NP}$ beliebig.

\Rightarrow für **jedes** $L \in \mathbf{NP}$ gilt $L_M \leq_p L_{SAT}$.

Damit Teil (ii) erledigt.

Satz von Cook/Levin: „ $L_{SAT} \in \mathbf{NPC}$ “ bewiesen. \square

Weitere **NP**-vollständige Probleme/Sprachen:

Beweis nur noch mit der **Reduktionsmethode**.

1.5 Reduktionsmethode, Beispiel 3-SAT

Beispiel: Klauseln der Länge 3:

$$(X_2 \vee \bar{X}_4 \vee X_7) \wedge (\bar{X}_0 \vee X_1 \vee X_4) \wedge (X_2 \vee \bar{X}_1 \vee X_6) \wedge (\bar{X}_1 \vee \bar{X}_2 \vee X_4).$$

Def.: Eine KNF-Formel φ ist **in 3-KNF-Form**, wenn jede Klausel genau drei Literale enthält.

Das Entscheidungsproblem $L_{3\text{-SAT}}$:

Eingabe: Formel φ in 3-KNF-Form

Frage: Ist φ erfüllbar?

$$L_{3\text{-SAT}} := \{\varphi \mid \varphi \text{ Formel in 3-KNF, } \varphi \text{ erfüllbar}\}.$$

Satz 1.5.1

$L_{3\text{-SAT}}$ ist **NP**-vollständig.

Beweis: (i) $L_{3\text{-SAT}} \in \mathbf{NP}$.

Die Entscheidungsvariante eines **NP**-Suchproblems, genau wie L_{SAT} .

Nur der Syntaxcheck („Ist x eine 3-KNF-Formel?“) ist leicht anders.

(ii) $L_{3\text{-SAT}}$ ist **NP**-schwer. heißt:

Für jedes $L' \in \mathbf{NP}$ gilt $L' \leq_p L_{3\text{-SAT}}$.

Könnte ziemlich mühsam sein

(siehe Beweis des Satzes von Cook/Levin!).

Reduktionsmethode

Lemma 1.5.2 Wenn (i) $L \in \mathbf{NP}$ und (ii)* $L' \leq_p L$ für eine **NP**-vollständige Sprache L' , dann ist L **NP**-vollständig.

Beweis: Es gelte (i), (ii)*.

Zu zeigen: (ii) $L'' \leq_p L$ für alle $L'' \in \mathbf{NP}$.

Sei $L'' \in \mathbf{NP}$ beliebig.

Weil L' **NP**-vollständig, gilt $L'' \leq_p L'$.

Weil \leq_p **transitiv**, folgt mit (ii)*: $L'' \leq_p L$. Fertig.

Rezept: Zeige (i) und (ii)*. Folgere: L ist **NP**-vollständig.

Hier: $L' = L_{\text{SAT}}$, $L = L_{3\text{-SAT}}$.

Sonst häufig: $L' = L_{3\text{-SAT}}$

Behauptung: $L_{\text{SAT}} \leq_p L_{3\text{-SAT}}$.

Müssen beliebige KNF-Formel φ transformieren in 3-KNF-Formel $f(\varphi) = \varphi^*$ mit:

$$\varphi \text{ erfüllbar} \Leftrightarrow \varphi^* \text{ erfüllbar}.$$

Achtung! φ, φ^* haben verschiedene Variablenmengen.

Sie sind **nicht äquivalent** (nur „**erfüllbarkeitsäquivalent**“).

Bemerkung 1: **Syntaxcheck.** Für Inputs x für f , die keine KNF-Formel sind, setzen wir $f(x) = 0$ (keine 3-KNF-Formel).

Bemerkung 2: **Vereinfachung.** Wenn in einer Klausel ein Literal mehrfach vorkommt: Wiederholungen streichen.

Gegeben: $\varphi = C_1 \wedge \dots \wedge C_r$

Wir bilden: $\varphi^* = \varphi_1^* \wedge \dots \wedge \varphi_r^*$

Dabei entstehen die φ_j^* aus den C_j , separat.

Beispiel: Aus

$$\varphi = \underbrace{(X_2 \vee \bar{X}_4)}_{C_1} \wedge \underbrace{(X_1 \vee \bar{X}_2 \vee \bar{X}_4 \vee X_5 \vee X_7)}_{C_2}$$

wird

$$\varphi^* = \underbrace{(X_2 \vee \bar{X}_4 \vee X_{101})}_{\varphi_1^*} \wedge \underbrace{(X_1 \vee \bar{X}_2 \vee X_{203})}_{\varphi_2^*} \wedge \underbrace{(\bar{X}_{203} \vee \bar{X}_4 \vee X_{204})}_{\varphi_2^*} \wedge \underbrace{(\bar{X}_{204} \vee X_5 \vee X_7)}_{\varphi_2^*}$$

Die Details folgen.

$$C_j = (l_1 \vee \dots \vee l_s)$$

1. Fall: $s = 1, C_j = (l_1)$.

$\varphi_j^* =$

$$(l_1 \vee Z_1 \vee Z_2) \wedge (l_1 \vee Z_1 \vee \bar{Z}_2) \wedge (l_1 \vee \bar{Z}_1 \vee Z_2) \wedge (l_1 \vee \bar{Z}_1 \vee \bar{Z}_2)$$

Dabei: Z_1, Z_2 neue Variable.

Anstelle von $(l_1 \vee l_1 \vee l_1)$ wählen wir diese etwas kompliziertere Variante, weil es für andere Reduktionen günstig ist, wenn man sagen kann:

„O.B.d.A. kommt jedes Literal in einer Klausel nur einmal vor.“

$$C_j = (l_1 \vee \dots \vee l_s)$$

2. Fall: $s = 2, C_j = (l_1 \vee l_2)$.

$$\varphi_j^* = (l_1 \vee l_2 \vee Z_1) \wedge (l_1 \vee l_2 \vee \bar{Z}_1)$$

Dabei: Z_1 neue Variable.

3. Fall: $s = 3, C_j = (l_1 \vee l_2 \vee l_3)$.

$$\varphi_j^* = C_j.$$

$$C_j = (l_1 \vee \dots \vee l_s)$$

4. Fall: $s \geq 4$.

$$\begin{aligned} \varphi_j^* = & (l_1 \vee l_2 \vee Z_3) \\ & \wedge (\bar{Z}_3 \vee l_3 \vee Z_4) \\ & \wedge (\bar{Z}_4 \vee l_4 \vee Z_5) \\ & \vdots \\ & \wedge (\bar{Z}_{s-2} \vee l_{s-2} \vee Z_{s-1}) \\ & \wedge (\bar{Z}_{s-1} \vee l_{s-1} \vee l_s) . \end{aligned}$$

Dabei: Z_3, Z_4, \dots, Z_{s-1} neue Variable.

Zu zeigen: Für jede KNF-Formel φ gilt:

φ erfüllbar $\Leftrightarrow \varphi^*$ erfüllbar.

Im Detail:

„ \Rightarrow “:

Starte mit Belegung v für die Variablen in φ mit $v(\varphi) = 1$.

Man kann v so zu einer Belegung v^* auch der neuen (Z_{\dots} -)Variablen **erweitern**, dass $v^*(\varphi^*) = 1$.

(Details gleich.)

„ \Leftarrow “:

Starte mit Belegung v^* für die Variablen in φ^* mit $v^*(\varphi^*) = 1$.

Man **beweist**: $v^*(\varphi) = 1$.

D.h.: v^* erfüllt auch φ .

(Details gleich.)

$$C_j = (l_1 \vee \dots \vee l_s)$$

2. Fall: $s = 2$, $C_j = (l_1 \vee l_2)$.

$$\varphi_j^* = (l_1 \vee l_2 \vee Z_1) \wedge (l_1 \vee l_2 \vee \overline{Z_1})$$

„ \Rightarrow “:

Sei $v(C_j) = 1$. Dann $v(l_1) = 1$ oder $v(l_2) = 1$.

Setze (z.B.) $v^*(Z_1) = 0$. Dann $v^*(\varphi_j^*) = 1$.

„ \Leftarrow “:

Sei $v^*(\varphi^*) = 1$. Dann ist $v^*(\varphi_j^*) = 1$.

Es ist nicht möglich, dass $v^*(l_1) = v^*(l_2) = 0$

(sonst hätte eine der beiden Klauseln in φ_j^* Wert 0).

Also: $v^*(C_j) = 1$.

1. Fall analog, **3. Fall** klar.

$$C_j = (l_1 \vee \dots \vee l_s)$$

4. Fall: $s \geq 4$.

„ \Rightarrow “:

Sei $v(l_r) = 1$. Dann setze

$v^*(Z_3) = \dots = v^*(Z_r) = 1$ und

$v^*(Z_{r+1}) = \dots = v^*(Z_{s-1}) = 0$.

Alle Klauseln von φ_j^* sind erfüllt.

„ \Leftarrow “:

Sei $v^*(\varphi_j^*) = 1$.

Annahme: $v^*(C_j) = 0$, also $v^*(l_1) = \dots = v^*(l_s) = 0$.

Man läuft die Klauseln von φ_j^* entlang und folgert der Reihe nach, dass $v^*(Z_3) = 1$, $v^*(Z_4) = 1$, \dots , $v^*(Z_{s-1}) = 1$ gilt.

Nun haben alle Literale in der letzten Klausel

$(\overline{Z_{s-1}} \vee l_{s-1} \vee l_s)$ den v^* -Wert 0, Widerspruch.