12b

NP-Vollständigkeit von 3-CNF-SAT

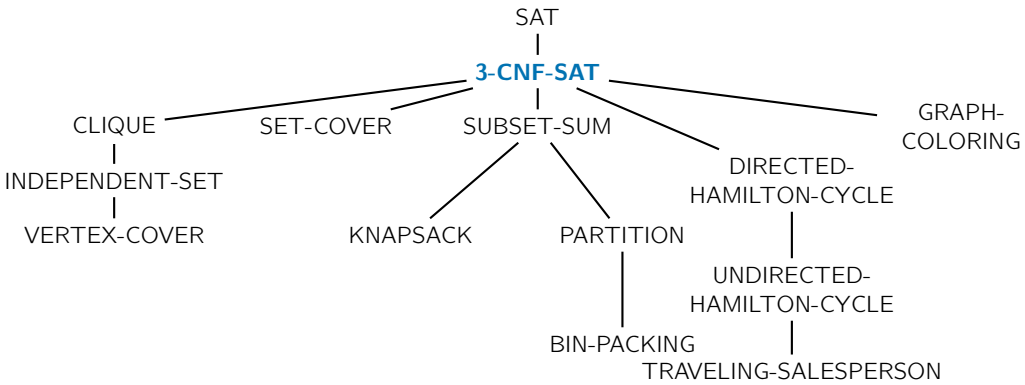
PD Dr. Jan Johannsen

Institut für Informatik

Stand: 7. Juli 2026
Basierend auf Folien von Prof. Dr. David Sabel und Prof. Dr. Jasmin Blanchette

Überblick über NP-Vollständigkeitsbeweise

Wir weisen NP-Schwere nach durch Polynomialzeitreduktion bekannter NP-vollständiger Probleme auf das neue Problem.



Wiederholung: NP-Vollständigkeit

Definition

Eine Sprache L heißt NP-vollständig, wenn gilt

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

Beweistechnik für NP-Vollständigkeit von L:

1. Zeige $L \in \mathsf{NP}$.
2. Zeige $L_0 \leq_p L$ für ein bekanntes NP-vollständiges Problem L_0 (z.B. SAT). Dann folgt die NP-Schwere von L.

Konjunktive Normalform

Ein Literal ist eine Variable x oder eine negierte Variable $\neg x$.

Eine Klausel ist eine Disjunktion $a_1 \vee \ldots \vee a_k$ von Literalen.

Eine Formel in konjunktiver Normalform (CNF) ist
eine Konjunktion $C_1 \wedge \ldots \wedge C_m$ von Klauseln.

Beispiel:

$$x_1 \wedge (x_2 \vee \neg x_4 \vee x_3) \wedge (x_2 \vee x_4)$$

Eine erfüllende Belegung muss mindestens 1 Literal pro Klausel wahr machen.

Wir nehmen an, dass es keine Klauseln gibt, die x und $\neg x$ enthalten.
Diese Klauseln sind immer wahr und können gelöscht werden.

3-CNF-SAT

Eine Formel in CNF ist in k-CNF, wenn jede Klausel höchstens k Literale enthält.

Definition

Das 3-CNF-SAT-Problem lässt sich wie folgt formulieren.

gegeben: eine aussagenlogische Formel F in 3-CNF,

Frage: Ist F erfüllbar? Genauer: Gibt es eine Belegung α mit $\alpha(F) = 1$?

Zugehörigkeit von 3-CNF-SAT zu NP

Satz

3-CNF-SAT ist NP-vollständig.

Beweis

1. 3-CNF-SAT ist in NP:

   Ein Zeuge für $F \in$ 3-CNF-SAT ist eine Bewertung α mit $\alpha \models F$.

   Da α nur Variablen bewerten muss, die in F vorkommen, ist $|\mathbf{dom}(\alpha)| \leq |F|$, und daher $|\alpha| \leq O(|F|)$.

   Die Prüfrelation $R_{\text{3-CNF-SAT}}$ ist gegeben durch $(\alpha, F) \in R_{\text{3-CNF-SAT}}$ gdw. $\alpha \models F$.

   Dies ist wie im Fall von SAT in Zeit $O(|F|^2)$ entscheidbar.

NP-Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist NP-vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist NP-schwer:
   Wir zeigen SAT $\leq_p$ 3-CNF-SAT.

   Gesucht wird eine polynomiell berechenbare, totale Funktion f, sodass:
   F ist erfüllbar g.d.w. die 3-CNF $f(F)$ ist erfüllbar.

CNF-Transformation

Die CNF-Transformation ist ein Verfahren,
um F in eine äquivalente CNF $f(F)$ umzuformen.

Schritte:

1. Löse $\rightarrow$ auf.
2. Schiebe Negationen nach innen und anschließend multipliziere aus.
   (Wende Distributivität, Kommutativität an, um CNF herzustellen.)

Zum Beispiel wird $x_1 \leftrightarrow x_2$ zu $(\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)$.

Aber der Algorithmus hat im schlechtesten Fall exponentielle Laufzeit und kann Klauseln erzeugen, die mehr als drei Literale enthalten.

Er kann daher nicht direkt für eine Polynomialzeitreduktion von SAT auf 3-CNF-SAT verwendet werden.

NP-Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist NP-vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist NP-schwer:
   Wir zeigen SAT $\leq_p$ 3-CNF-SAT.

   Gesucht wird eine polynomiell berechenbare, totale Funktion f, sodass:
   F ist erfüllbar g.d.w. die 3-CNF $f(F)$ ist erfüllbar.

   Beachte: f muss die Erfüllbarkeit erhalten, aber nicht die Äquivalenz.

   Die Tseitin-Transformation ist ein Verfahren,
   um F in eine erfüllbarkeits-äquivalente 3-CNF $E(F)$ umzuformen,
   sodass $E(F)$ lineare Größe in $|F|$ hat.

Negations-Normalform

Eine Formel F ist in Negations-Normalform (NNF).

Transformation einer Formel in eine äquivalente Formal in NNF:

- Ersetze alle Teilformeln $F \to G$ durch $\neg F \vee G$.
- Wende die folgenden Transformationsregeln an, bis keine mehr anwendbar ist:
  - Ersetze Teilformel $\neg(F \wedge G)$ durch $\neg F \vee \neg G$.
  - Ersetze Teilformel $\neg(F \vee G)$ durch $\neg F \wedge \neg G$.
  - Ersetze Teilformel $\neg\neg F$ durch F.

Tseitin-Transformation

Sei F in NNF.

Für jede Teilformel G von F definiere
eine neue Variable v_G, und die definierende Formeln D_G:

- $G = a$ Literal:

$$\begin{aligned} D_a &= (v_a \leftrightarrow a) = (v_a \rightarrow a) \wedge (a \rightarrow v_a) \\ &= (\neg v_a \vee a) \wedge (\neg a \vee v_a) \end{aligned}$$

Tseitin-Transformation

- $G = H_1 \vee H_2$ Disjunktion:

$$\begin{aligned} D_G &= (v_G \leftrightarrow (v_{H_1} \vee v_{H_2})) \\ &= (\neg v_G \vee v_{H_1} \vee v_{H_2}) \wedge (\neg v_{H_1} \vee v_G) \wedge (\neg v_{H_2} \vee v_G) \end{aligned}$$

- $G = H_1 \wedge H_2$ Konjunktion:

$$\begin{aligned} D_G &= (v_G \leftrightarrow (v_{H_1} \wedge v_{H_2})) \\ &= (\neg v_G \vee v_{H_1}) \wedge (\neg v_G \vee v_{H_2}) \wedge (\neg v_{H_1} \vee \neg v_{H_2} \vee v_G) \end{aligned}$$

$$E(F) = v_F \wedge \bigwedge_{G \text{ Teilformel von } F} D_G$$

Erfüllbarkeitsäquivalenz der Tseitin-Transformation

Zu zeigen: $E(F)$ ist erüllbar genau dann, wenn F erfüllbar ist.

Sei $\alpha \models E(F)$.

Zeige durch Induktion: $\alpha(G) = \alpha(v_G)$ für jede Teilformel G von F.

Da $\alpha \models E(F)$, ist $\alpha(v_F) = 1$, also auch $\alpha \models F$.

Sei nun $\alpha \models F$.

Definiere Erweiterung α^* von α durch:
$\alpha^*(v_G) := \alpha(G)$ für jede Teilformel G von F.

Dann gilt nach Konstruktion: $\alpha^* \models D_G$ für jede Teilformel G von F
außerdem $\alpha^*(v_F) = 1$, also insgesamt $\alpha^* \models E(F)$.

Komplexität der Tseitin-Transformation

$E(F)$ ist offensichtlich in 3-CNF.
Es bleibt zu zeigen, dass $E(F)$ polynomiell berechenbar ist.

Die Größe der Teilformeln D_G, ist konstant.

Die Anzahl der Teilformeln D_G, die erzeugt werden, ist gleich der Anzahl der logischen Zeichen in F, also kleiner als $|F|$.

Die Größe der 3-CNF ist also linear in der Größe der ursprünglichen Formel F.

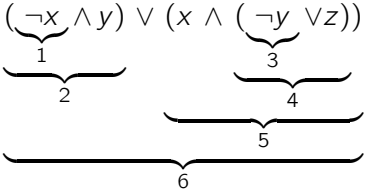
Die Berechnung geht in Polynomialzeit.

Daher: SAT $\leq_p$ 3-CNF-SAT.
Da SAT **NP**-schwer ist, folgt dass 3-CNF-SAT **NP**-schwer ist. □

Beispiel für die Tseitin-Transformation

Sei F die Formel $(\neg x \wedge y) \vee (x \wedge (y \rightarrow z))$.

Nach Transformation in NNF erhalten wir:



Wir führen neue Variablen $v_1, \ldots, v_6$ ein

und definieren die Formeln $D_1, \ldots, D_6$ für die jeweiligen Teilformeln.

Z.B. ist $D_3 \equiv (v_3 \leftrightarrow \neg y)$ und $D_4 \equiv v_4 \leftrightarrow (v_3 \vee z)$.

Beispiel für die Tseitin-Transformation

Die Formeln D_i sind:

$$
\begin{aligned}
D_1 &= (v_1 \leftrightarrow \neg x) \\
&= (v_1 \vee x) \wedge (\neg v_1 \vee \neg x) \\
D_2 &= (v_2 \leftrightarrow (v_1 \wedge y)) \\
&= (\neg v_2 \vee v_1) \wedge (\neg v_2 \vee y) \wedge (v_2 \vee \neg v_1 \vee \neg y) \\
D_3 &= (v_3 \vee y) \wedge (\neg v_1 \vee \neg y) \\
D_4 &= (v_4 \leftrightarrow (v_3 \vee z)) \\
&= (v_4 \vee \neg v_3) \wedge (v_4 \vee \neg z) \wedge (\neg v_4 \vee v_3 \vee z) \\
D_5 &= (\neg v_5 \vee x) \wedge (\neg v_5 \vee v_4) \wedge (v_5 \vee \neg x \vee \neg v_4) \\
D_6 &= (v_6 \vee \neg v_2) \wedge (v_6 \vee \neg v_5) \wedge (\neg v_6 \vee v_2 \vee v_5)
\end{aligned}
$$

Die Formel $E(F)$ ist: $\quad v_6 \wedge D_1 \wedge \ldots \wedge D_6$.