

11b

$\mathcal{NP}$ -Vollständigkeit von 3-CNF-SAT

Prof. Dr. Jasmin Blanchette

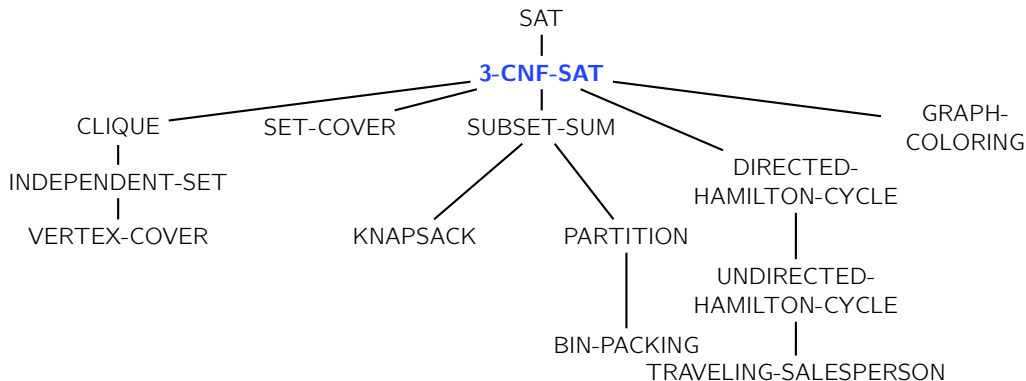
Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 11. Februar 2025
Basierend auf Folien von PD Dr. David Sabel



Überblick über $\mathcal{NP}$ -Vollständigkeitsbeweise

Wir weisen $\mathcal{NP}$ -Schwere nach durch Polynomialzeitreduktion bekannter $\mathcal{NP}$ -vollständiger Probleme auf das neue Problem.



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 (z.B. SAT).
Dann folgt die $\mathcal{NP}$ -Schwere von L .

Konjunktive Normalform

Die aussagenlogische Formel F ist in **konjunktiver Normalform (CNF)**, wenn sie von der folgenden Form ist:

$$\bigwedge_{i=1}^m \left(\bigvee_{j=1}^{n_i} L_{i,j} \right)$$

- ▶ Jedes **Literal** $L_{i,j}$ ist eine **aussagenlogische Variable** oder deren **Negation**.
- ▶ $(\bigvee_{j=1}^{n_i} L_{i,j})$ ist eine **Klausel**.

Konjunktive Normalform

Die aussagenlogische Formel F ist in **konjunktiver Normalform (CNF)**, wenn sie von der folgenden Form ist:

$$\bigwedge_{i=1}^m \left(\bigvee_{j=1}^{n_i} L_{i,j} \right)$$

- ▶ Jedes **Literal** $L_{i,j}$ ist eine **aussagenlogische Variable oder deren Negation**.
- ▶ $(\bigvee_{j=1}^{n_i} L_{i,j})$ ist eine **Klausel**.

Beispiel:

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

Konjunktive Normalform

Die aussagenlogische Formel F ist in **konjunktiver Normalform (CNF)**, wenn sie von der folgenden Form ist:

$$\bigwedge_{i=1}^m \left(\bigvee_{j=1}^{n_i} L_{i,j} \right)$$

- ▶ Jedes **Literal** $L_{i,j}$ ist eine **aussagenlogische Variable oder deren Negation**.
- ▶ $(\bigvee_{j=1}^{n_i} L_{i,j})$ ist eine **Klausel**.

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.

Konjunktive Normalform

Die aussagenlogische Formel F ist in **konjunktiver Normalform (CNF)**, wenn sie von der folgenden Form ist:

$$\bigwedge_{i=1}^m \left(\bigvee_{j=1}^{n_i} L_{i,j} \right)$$

- ▶ Jedes **Literal** $L_{i,j}$ ist eine **aussagenlogische Variable oder deren Negation**.
- ▶ $(\bigvee_{j=1}^{n_i} L_{i,j})$ ist eine **Klausel**.

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.

Definition

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

gegeben: eine aussagenlogische Formel F in CNF,
sodass jede Klausel höchstens 3 Literale enthält
(d.h. eine aussagenlogische Formel F in **3-CNF**)

gefragt: Ist F erfüllbar? Genauer: Gibt es eine erfüllende Belegung der Variablen,
sodass F den Wert 1 erhält?

Zugehörigkeit von 3-CNF-SAT zu $\mathcal{NP}$

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Zugehörigkeit von 3-CNF-SAT zu $\mathcal{NP}$

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis

1. 3-CNF-SAT ist in $\mathcal{NP}$:

Rate nichtdeterministisch eine Belegung der Variablen.

Zugehörigkeit von 3-CNF-SAT zu $\mathcal{NP}$

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis

1. 3-CNF-SAT ist in $\mathcal{NP}$:

Rate nichtdeterministisch eine Belegung der Variablen.

Prüfe deterministisch, ob die Belegung die 3-CNF wahr macht. Akzeptiere in diesem Fall, sonst verwirf.

Zugehörigkeit von 3-CNF-SAT zu $\mathcal{NP}$

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis

1. 3-CNF-SAT ist in $\mathcal{NP}$:

Rate nichtdeterministisch eine Belegung der Variablen.

Prüfe deterministisch, ob die Belegung die 3-CNF wahr macht. Akzeptiere in diesem Fall, sonst verwirf.

Dies geht in Polynomialzeit in der Größe der 3-CNF.

Zugehörigkeit von 3-CNF-SAT zu $\mathcal{NP}$

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis

1. 3-CNF-SAT ist in $\mathcal{NP}$:

Rate nichtdeterministisch eine Belegung der Variablen.

Prüfe deterministisch, ob die Belegung die 3-CNF wahr macht. Akzeptiere in diesem Fall, sonst verwirf.

Dies geht in Polynomialzeit in der Größe der 3-CNF.

Daher kann 3-CNF-SAT auf einer NTM in Polynomialzeit entschieden werden.

$\mathcal{NP}$ -Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist $\mathcal{NP}$ -schwer:

Wir zeigen $\text{SAT} \leq_p \text{3-CNF-SAT}$.

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist $\mathcal{NP}$ -schwer:

Wir zeigen $\text{SAT} \leq_p \text{3-CNF-SAT}$.

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.

CNF-Transformation

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

Schritte:

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

CNF-Transformation

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

Schritte:

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

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

CNF-Transformation

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

Schritte:

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

Zum Beispiel wird $x_1 \iff x_2$ 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.

CNF-Transformation

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

Schritte:

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

Zum Beispiel wird $x_1 \iff x_2$ 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.

$\mathcal{NP}$ -Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist $\mathcal{NP}$ -schwer:

Wir zeigen $\text{SAT} \leq_p \text{3-CNF-SAT}$.

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

$\mathcal{NP}$ -Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist $\mathcal{NP}$ -schwer:

Wir zeigen $\text{SAT} \leq_p \text{3-CNF-SAT}$.

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**.

$\mathcal{NP}$ -Schwere von 3-CNF-SAT

Satz

3-CNF-SAT ist $\mathcal{NP}$ -vollständig.

Beweis (Fortsetzung)

2. 3-CNF-SAT ist $\mathcal{NP}$ -schwer:

Wir zeigen $\text{SAT} \leq_p \text{3-CNF-SAT}$.

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 erfüllbarkeitsäquivalente 3-CNF $f(F)$ umzuformen, sodass $f(F)$ **polynomielle Größe** in $|F|$ hat.

Tseitin-Transformation

Schritte:

1. Schiebe alle Negationen nach innen vor die Literale, dabei werden die Regeln
 $\neg\neg F \rightsquigarrow F$, $\neg(F \wedge G) \rightsquigarrow \neg F \vee \neg G$, $\neg(F \vee G) \rightsquigarrow \neg F \wedge \neg G$,
 $\neg(F \iff G) \rightsquigarrow \neg F \iff G$ und $\neg(F \implies G) \rightsquigarrow F \wedge \neg G$ angewendet.

Tseitin-Transformation

Schritte:

1. Schiebe alle Negationen nach innen vor die Literale, dabei werden die Regeln $\neg\neg F \rightsquigarrow F$, $\neg(F \wedge G) \rightsquigarrow \neg F \vee \neg G$, $\neg(F \vee G) \rightsquigarrow \neg F \wedge \neg G$, $\neg(F \iff G) \rightsquigarrow \neg F \iff G$ und $\neg(F \implies G) \rightsquigarrow F \wedge \neg G$ angewendet.
2. Bilde den Syntaxbaum der Formel, mit Literalen auf den Blättern.

Tseitin-Transformation

Schritte:

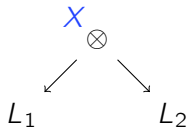
1. Schiebe alle Negationen nach innen vor die Literale, dabei werden die Regeln $\neg\neg F \rightsquigarrow F$, $\neg(F \wedge G) \rightsquigarrow \neg F \vee \neg G$, $\neg(F \vee G) \rightsquigarrow \neg F \wedge \neg G$, $\neg(F \iff G) \rightsquigarrow \neg F \iff G$ und $\neg(F \implies G) \rightsquigarrow F \wedge \neg G$ angewendet.
2. Bilde den Syntaxbaum der Formel, mit Literalen auf den Blättern.
3. Führe für jeden Nichtblatt-Knoten neue aussagenlogische Variable X ein, die als **Synonym für eine Teilformel** steht.

Tseitin-Transformation

Schritte:

1. Schiebe alle Negationen nach innen vor die Literale, dabei werden die Regeln $\neg\neg F \rightsquigarrow F$, $\neg(F \wedge G) \rightsquigarrow \neg F \vee \neg G$, $\neg(F \vee G) \rightsquigarrow \neg F \wedge \neg G$, $\neg(F \iff G) \rightsquigarrow \neg F \iff G$ und $\neg(F \implies G) \rightsquigarrow F \wedge \neg G$ angewendet.
2. Bilde den Syntaxbaum der Formel, mit Literalen auf den Blättern.
3. Führe für jeden Nichtblatt-Knoten neue aussagenlogische Variable X ein, die als **Synonym für eine Teilformel** steht.

4. Pro Gabelung



erzeuge die Formel $X \iff (L_1 \otimes L_2)$, wobei

$\otimes \in \{\wedge, \vee, \implies, \iff\}$ und L_1 bzw. L_2 entweder die neue Variable oder das Literal am Blatt ist.

5. Konjugiere die erzeugten Formeln zu F' und schließlich erzeuge $W \wedge F'$, wobei W die Variable für die Wurzel ist.
Die erzeugte Formel ist von der Form

$$W \wedge \bigwedge_{i,j,k} (X_i \iff (L_j \otimes_i L_k))$$

5. Konjugiere die erzeugten Formeln zu F' und schließlich erzeuge $W \wedge F'$, wobei W die Variable für die Wurzel ist.

Die erzeugte Formel ist von der Form

$$W \wedge \bigwedge_{i,j,k} (X_i \iff (L_j \otimes_i L_k))$$

6. Berechne für jede Teilformel $X_i \iff (L_j \otimes_i L_k)$ die CNF mit der üblichen CNF-Transformation.

5. Konjugiere die erzeugten Formeln zu F' und schließlich erzeuge $W \wedge F'$, wobei W die Variable für die Wurzel ist.
Die erzeugte Formel ist von der Form

$$W \wedge \bigwedge_{i,j,k} (X_i \iff (L_j \otimes_i L_k))$$

6. Berechne für jede Teilformel $X_i \iff (L_j \otimes_i L_k)$ die CNF mit der üblichen CNF-Transformation.
7. Lösche doppelte Vorkommen von Literalen.

5. Konjugiere die erzeugten Formeln zu F' und schließlich erzeuge $W \wedge F'$, wobei W die Variable für die Wurzel ist.
Die erzeugte Formel ist von der Form

$$W \wedge \bigwedge_{i,j,k} (X_i \iff (L_j \otimes_i L_k))$$

6. Berechne für jede Teilformel $X_i \iff (L_j \otimes_i L_k)$ die CNF mit der üblichen CNF-Transformation.
7. Lösche doppelte Vorkommen von Literalen.

Diese Prozedur ergibt eine Formel in 3-CNF,
da pro Klausel nur drei verschiedene Variablen vorkommen können.

Beispiel für die Tseitin-Transformation

Sei $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Beispiel für die Tseitin-Transformation

Sei $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Negationen nach innen schieben:

$$\neg(\neg(x_3 \implies \neg x_1) \vee x_2)$$

$$\rightsquigarrow \neg\neg(x_3 \implies \neg x_1) \wedge \neg x_2$$

$$\rightsquigarrow (x_3 \implies \neg x_1) \wedge \neg x_2$$

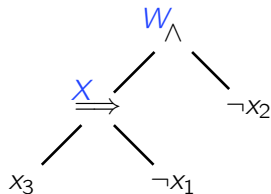
Beispiel für die Tseitin-Transformation

Sei $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Negationen nach innen schieben:

$$\begin{aligned} & \neg(\neg(x_3 \implies \neg x_1) \vee x_2) \\ \rightsquigarrow & \neg\neg(x_3 \implies \neg x_1) \wedge \neg x_2 \\ \rightsquigarrow & (x_3 \implies \neg x_1) \wedge \neg x_2 \end{aligned}$$

Syntaxbaum dazu:



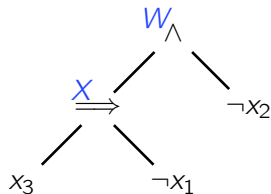
Beispiel für die Tseitin-Transformation

Sei $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Negationen nach innen schieben:

$$\begin{aligned} & \neg(\neg(x_3 \implies \neg x_1) \vee x_2) \\ \rightsquigarrow & \neg\neg(x_3 \implies \neg x_1) \wedge \neg x_2 \\ \rightsquigarrow & (x_3 \implies \neg x_1) \wedge \neg x_2 \end{aligned}$$

Syntaxbaum dazu:



Formel: $W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1))$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1))$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1))$$

$$\leadsto W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1))$$

$$\leadsto W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1))$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee \neg(\neg x_3 \vee \neg x_1)) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (\textcolor{red}{X} \vee \neg(\neg x_3 \vee \neg x_1)) \\ \leadsto & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (\textcolor{blue}{X} \vee (x_3 \wedge x_1)) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee \neg(\neg x_3 \vee \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (\neg X \vee (x_3 \wedge x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee x_3) \wedge (X \vee x_1) \end{aligned}$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee \neg(\neg x_3 \vee \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee (x_3 \wedge x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee x_3) \wedge (X \vee x_1) \end{aligned}$$

Die erzeugte Formel $f(F)$ ist erfüllbarkeitsäquivalent zu $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Zum Beispiel:

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee \neg(\neg x_3 \vee \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee (x_3 \wedge x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee x_3) \wedge (X \vee x_1) \end{aligned}$$

Die erzeugte Formel $f(F)$ ist erfüllbarkeitsäquivalent zu $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Zum Beispiel:

Erfüllende Belegung I für $f(F)$:

$$I(x_1) = 0, I(x_2) = 0, I(x_3) = 1, I(\textcolor{blue}{W}) = 1, I(\textcolor{blue}{X}) = 1$$

Beispiel für die Tseitin-Transformation

Berechnung der CNFs der kleinen Teilformeln:

$$\begin{aligned} & W \wedge (W \iff (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg(X \wedge \neg x_2)) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee (X \wedge \neg x_2)) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (X \iff (x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee (x_3 \implies \neg x_1)) \wedge (X \vee \neg(x_3 \implies \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee \neg(\neg x_3 \vee \neg x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee (x_3 \wedge x_1)) \\ \rightsquigarrow & W \wedge (W \vee \neg X \vee x_2) \wedge (\neg W \vee X) \wedge (\neg W \vee \neg x_2) \wedge (\neg X \vee \neg x_3 \vee \neg x_1) \wedge (X \vee x_3) \wedge (X \vee x_1) \end{aligned}$$

Die erzeugte Formel $f(F)$ ist erfüllbarkeitsäquivalent zu $F = \neg(\neg(x_3 \implies \neg x_1) \vee x_2)$.

Zum Beispiel:

Erfüllende Belegung I für $f(F)$:

$$I(x_1) = 0, I(x_2) = 0, I(x_3) = 1, I(W) = 1, I(X) = 1$$

Erfüllende Belegung I' für F :

$$I'(x_1) = 0, I'(x_2) = 0, I'(x_3) = 1$$

Erfüllbarkeitsäquivalenz der Tseitin-Transformation

Beweis (Fortsetzung)

Wir zeigen: F ist erfüllbar g.d.w. $f(F)$ ist erfüllbar.

Erfüllbarkeitsäquivalenz der Tseitin-Transformation

Beweis (Fortsetzung)

Wir zeigen: F ist erfüllbar g.d.w. $f(F)$ ist erfüllbar.

$\Leftarrow$ Sei I eine Belegung mit $I(f(F)) = 1$.

Für die Restriktion I' von I auf die Variablen von F gilt $I'(F) = 1$.

Erfüllbarkeitsäquivalenz der Tseitin-Transformation

Beweis (Fortsetzung)

Wir zeigen: F ist erfüllbar g.d.w. $f(F)$ ist erfüllbar.

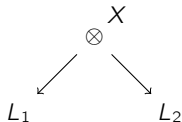
$\Leftarrow$ Sei I eine Belegung mit $I(f(F)) = 1$.

Für die Restriktion I' von I auf die Variablen von F gilt $I'(F) = 1$.

$\Rightarrow$ Sei I eine Belegung mit $I(F) = 1$.

Sei I' die Belegung mit

- ▶ $I'(x) = I(x)$ für alle Variablen x , die in F vorkommen
- ▶ $I'(X) = I'(L_1 \otimes L_2)$ für



Dann gilt $I'(f(F)) = 1$.

Komplexität der Tseitin-Transformation

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass f polynomiell berechenbar ist.

Komplexität der Tseitin-Transformation

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass f polynomiell berechenbar ist.

Die Größe der Teilformeln, die in CNF gebracht werden, ist konstant.

Die Anzahl der Teilformeln, die erzeugt werden, ist proportional zur Größe des Syntaxbaums.

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

Die Berechnung in **Polynomialzeit** geht.

Komplexität der Tseitin-Transformation

Beweis (Fortsetzung)

Es bleibt zu zeigen, dass f polynomiell berechenbar ist.

Die Größe der Teilformeln, die in CNF gebracht werden, ist konstant.

Die Anzahl der Teilformeln, die erzeugt werden, ist proportional zur Größe des Syntaxbaums.

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

Die Berechnung in **Polynomialzeit** geht.

Daher: $\text{SAT} \leq_p \text{3-CNF-SAT}$.

Da SAT $\mathcal{NP}$ -schwer ist, folgt dass 3-CNF-SAT $\mathcal{NP}$ -schwer ist. □