

# SAT-3SAT

(SAT-3SAT)

**Stichwörter:** Polynomialzeitreduktion

## Exkurs: Sat

Das **Erfüllbarkeitsproblem der Aussagenlogik** SAT und  $\kappa$ -SAT mit  $k \geq 3, k \in \mathbb{N}$  (Satz von Cook) fragt, ob eine aussagenlogische Formel erfüllbar ist. Das Erfüllbarkeitsproblem der *Aussagenlogik* ist in exponentieller Zeit in Abhängigkeit der Anzahl der Variablen mit Hilfe einer Wahrheitstabelle entscheidbar. Diese *Wahrheitstabelle* kann nicht in polynomieller Zeit aufgestellt werden.

- (a) Wie zeigt man die aus der NP-Schwere des 3SAT-Problems die NP-Schwere des SAT-Problems?

Lösungsvorschlag

$$3\text{SAT} \preceq_p \text{SAT}$$

Jedes 3SAT-Problem ist auch ein SAT-Problem, weil  $3\text{SAT} \subset \text{SAT}$ . Damit braucht es keine Funktion (bzw. Identitäts-/Einheitsfunktion). Die Funktion ist korrekt, total und in Polynomialzeit anwendbar. Das SAT-Problem ist ebenfalls NP-schwer.

- (b) Wie zeigt man die aus der NP-Schwere des SAT-Problems die NP-Schwere des 3SAT-Problems?

Lösungsvorschlag

$$\text{SAT} \preceq_p 3\text{SAT}$$

Man muss eine Funktion finden, die eine allgemeine Aussagenlogik in eine Aussagenlogik mit 3 Literalen in konjunktiver Normalform umformt.

Durch die boolsche Algebra lässt sich jede logische Aussagenlogik in eine konjunktive Normalform bringen. Dies ist eine Konjunktion von Disjunktionstermen. Wir formen einen Disjunktionsterm mithilfe einer Funktion in ein 3SAT-Problem um. Diese Funktion kann auf jeden Disjunktionsterm angewendet werden und damit wird das gesamte SAT-Problem auf 3SAT reduziert.

Die Funktion formt Formel aus SAT mithilfe von Hilfsvariablen  $h_1, \dots, h_{n-2}$  derart um  $(a_1 \vee \dots \vee a_n) \rightarrow (a_1 \vee a_2 \vee h_1) \wedge (\neg h_1 \vee a_3 \vee h_2) \wedge (\neg h_2 \vee a_4 \vee h_3) \wedge \dots \wedge (\neg h_{n-2} \vee a_n)$

**total** Diese Funktion ist total, denn jede in SAT enthaltene Aussagenlogik kann so umgewandelt werden.

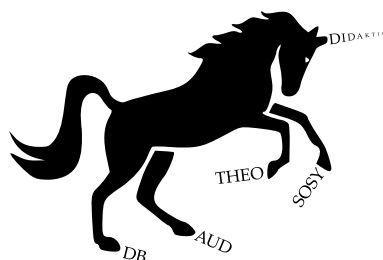
**Korrektheit:** Die Hilfsvariablen sind wahr, solange bis ein Literal  $a_x$  selber true ist. Ab diesem Zeitpunkt sind die Hilfsvariablen dann falsch.

**JA-Instanzen:** Der erste und alle mittleren Disjunktionstermen sind wahr, weil aufgrund der Nicht-Negierung und Negierung immer ein wahres Literal in den Disjunktionstermen. Somit ist dann auch der Disjunktionsterm wahr. Da es eine JA-Instanz ist, existiert ein  $a_x$  welches wahr ist. Somit sind

ab diesem Zeitpunkt die Hilfsvariablen falsch. Der letzte Disjunktionsterm wird dadurch sicher wahr, weil  $\neg h_{n-2}$  somit wahr ist.

**NEIN-Instanz:** Alle  $a_x$  sind falsch. Auch hier sind wieder der erste und alle mittleren Disjunktionsterme wahr (gleiche Begründung wie oben). Der letzte Disjunktionsterm ist allerdings falsch, weil die Hilfsvariablen durchgehend wahr bleiben und alle  $a_x$  falsch sind. Durch die Konjunktion der Disjunktionsterme ist dann auch die Gesamtaussage falsch.

**Polynomialzeit:** Der Algorithmus, der Formeln aus SAT nach 3SAT umformt liegt in  $\mathcal{O}(n)$  und somit in Polynomialzeit.



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der  $\text{\LaTeX}$ -Quelltext dieser Aufgabe kann unter folgender URL aufgerufen werden: [https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Module/70\\_THEO/40\\_Komplexitaet/Aufgabe\\_SAT-3SAT.tex](https://github.com/bschlangaul-sammlung/examens-aufgaben-tex/blob/main/Module/70_THEO/40_Komplexitaet/Aufgabe_SAT-3SAT.tex)