NP-Vollständigkeit wichtiger Probleme

Sebastian Bernauer

22.03.2019

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Komplexitätsklassen

Probleme in der Informatik werden Probleme in verschiedene Komplexitätsklassen aufgeteilt.
Der Operator $L_1 \leq_p L_2$ bedeutet, dass das L_1 polynomiell auf L_2 reduzierbar ist. Dies ist der Fall, wenn es eine polynomielle Transformation von L_1 nach L_2 gibt, so das heißt, wenn es eine von einer DTM in polynomieller Zeit berechenbare Funktion $f : \sum_1^* \to \sum_2^*$ gibt, so dass für alle $w \in \sum_1^*$ gilt:

$$w \in L_1 \leftrightarrow f(w) \in L_2$$

Dieser Operator $\leq_p$ kann nun Probleme in diverse Problemklassen einteilen. In der Tabelle 1 sind häufige Problemklassen dargestellt.

Komplexitätsklasse	**Beschreibung**
P	Polynomiell lösbar
NP	Nichtdeterministisch polynomiell lösbar
NP-vollständig	Schwierigste Probleme in NP
NP-schwierig	Genauso schwer wie NP-vollständig, das Problem muss aber nicht in NP enthalten sein
nicht rekursiv	Sehr, sehr schwere Probleme

Tabelle 1: Häufige Problemklassen

1.1 NP-vollständig

Ein Problem/Sprache L ist NP-vollständig, wenn folgendes gilt:

- $L \in NP$
- $\forall L' \in NP : L' \leq_p L$

1.2 NP-schwierig

Ein Problem/Sprache L ist NP-schwierig, wenn folgendes gilt:

- $\forall L' \in NP : L' \leq_p L$

2 Satisfiability Problem (SAT)

2.1 Problembeschreibung

Für natürliche Zahlen n und m seien m Klauseln über n Variablen gegeben. Eine Klausel ist die Disjunktion [Veroderung] von einigen Literalen x_i bzw. $\overline{x_i}$ mit $i, j \in \{1, ..., n\}$. Es soll entschieden werden, ob es eine Belegung $a =$

$\{a_1, ..., a_n\} \in \{0,1\}^n$ der Variablen $x_1, ..., x_n$ gibt, so dass alle Klauseln erfüllt sind.
Eine mögliche Fragestellung für das SAT-Problem ist: Existiert eine Wahrheitsbelegung der Variablen $x_1, ..., x_n$, so dass alle Klauseln erfüllt sind?
Der Beweis, dass das SAT-Problem ist der Satz von Cook, der in der vorherigen Präsentation gezeigt wurde. Er ist unter [3] in dem Kapitel 3.3.7 zu finden.

2.2 3-SAT

3-SAT ist ein Spezialfall von SAT, bei denen jede Klausel die Länge 3 hat, also 2 Literale besitzt. Im Folgenden wird bewiesen, dass das allgemeine SAT-Problem durch das 3-SAT Problem abbildbar ist. Durch den Beweis wird gezeigt, dass die zwei Probleme gleich komplex und somit beide NP-vollständig sind (SAT wurde ja durch den Satz von Cook als NP-vollständig bewiesen).

2.3 Beweis: 3-SAT $\leq_p$ SAT

Das grundsätzliche Vorgehen, um SAT-Probleme in 3-SAT-Probleme abzubilden ist, alle Klauseln in neue Klauseln der Länge 3 zu überführen. Haben die ursprüngliche, allgemeinen Klauseln die Länge ≤ 3, sind die Umformungen trivial:

- Länge 1: Die Klausel besteht aus einem Literal z
  Neue Klausel der Länge 3: $z \vee z \vee z$
  Durch die Umformung wird die semantische Bedeutung der Klausel nicht verändert, dies könnte z.B. durch eine Wahrheitstabelle bewiesen werden.
- Länge 2: Die Klausel besteht aus 2 Literalen $z \vee y$
  Neue Klausel: $z \vee z \vee y$
  Hier gilt das Gleiche wie bei der Umformung einer Klausel der Länge 1.
- Länge 3: Die Klausel besteht aus 3 Literalen $z \vee y \vee z$
  Neue Klausel: $z \vee y \vee z$
  An dieser Stelle besteht die Klausel bereits aus 3 Literalen, es muss keine Umformung durchgeführt werden.

Hat die ursprüngliche Klausel eine Länge ≥ 4, wird die Umformung schwieriger. Hier wird für das Verständnis zunächst eine beispielhafte Umformung gezeigt, anschließend wird die allgemeine Umformung definiert.
Beispiel: Klausel der Länge $k = 7$ mit den Literalen $z_1 \vee ... \vee z_k$ wird umgeformt zu folgenden 5 Klauseln der Länge 3:

1. $z_1 \vee z_2 \vee y_1$
2. $\overline{y_1} \vee z_3 \vee y_2$
3. $\overline{y_2} \vee z_4 \vee y_3$
4. $\overline{y_3} \vee z_5 \vee y_4$

5. $\overline{y_4} \vee z_6 \vee z_7$

Die Literale $y_1, ..., y_4$ sind eingeführte "Hilfsvariablen" und werden benötigt, um die Klausel aufzutrennen ohne die Bedeutung zu verändern. Betrachten wir die entstandene Klausel 3: Es ist zu erkennen, dass durch das Literal $\overline{y_2}$ entweder die vorherige Klausel (2) oder die aktuelle Klausel (3) erfüllt werden muss. Dies entspricht dem Operator, der in der ursprünglichen Klausel zwischen den Literalen stand, dem logischen ODER. Das zweite Literal in der neuen Klausel - z_5 - wurde von der ursprünglichen Klausel übernommen. Mit dem dritten und letztem Literal y_4 wurde die ODER-Verknüpfung zu der nächsten entstandenen Klausel geschaffen.
Das war ein konkretes Beispiel, um eine Klausel der Länge $k = 7$ in Klauseln der Länge 3 zu überführen. Die allgemeine Umformung für Klauseln der Länge $k \geq 4$ sieht wie folgt aus: c ist die Nummer der Klausel und wurde eingeführt, um die Hilfsvariablen der Klauseln formal zu trennen.

1. $z_1 \vee z_2 \vee y_{c,1}$
2. $\overline{y_{c,l}} \vee z_{l+2} \vee y_{c,l+1}$ für $1 \leq l \leq k-4$
3. $\overline{y_{c,k-3}} \vee z_{k-1} \vee z_k$

Das Prinzip bleibt das Gleiche, wie bei dem Beispiel mit $k = 7$, die Umformung ist nur allgemein definiert.
Durch die Transformation eines SAT-Problem in ein 3-SAT-Problem wurde bewiesen, dass das 3-SAT-Problem NP-vollständig ist.

3 Clique Problem

3.1 Problemstellung

In einem ungerichteten Graphen $G = (V, E)$ bildet die Knotenmenge $V' \subseteq V$ eine Clique, wenn für alle $v, v' \in V'$ gilt:

$$v, v' \in E$$

In den Abbildungen 1 und 2 auf der nächsten Seite sind 2 Graphen mit Cliquen dargestellt.

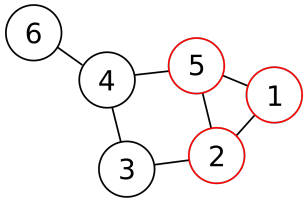

Abbildung 1: Ein Graph mit einer Clique der Größe 3. Bildquelle: [2]

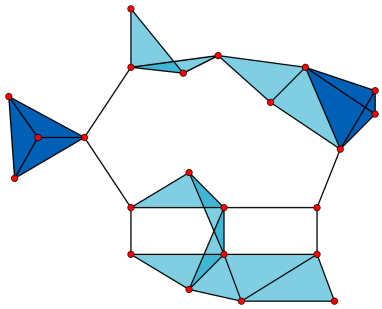

Abbildung 2: Ein Graph mit zwei Cliquen der Größe 4 (dunkelblau) und vielen Cliquen der Größe 3 (türkis). Bildquelle: [1]

3.2 Fragestellungen

Es gibt für die meisten Probleme 3 Fragestellungen, diese sind in Tabelle 2 aufgeführt.

Fragestellung	**Beschreibung**
Entscheidungsproblem	Liefert wahr oder falsch
Werteproblem	Liefert eine Zahl
Optimierungsproblem	Liefert die Lösung des Problems, z.B. eine Menge an Knoten

Tabelle 2: Die 3 Fragestellungen bei Problemen

Für das Clique-Problem gibt es die folgenden 3 Fragestellungen:

1. Entscheidungsproblem: Gibt es eine Clique der Größe k?

2. Werteproblem: Berechne das größte k, so dass eine Clique der Größe k vorhanden ist.

3. Optimierungsproblem: Berechne eine Clique mit dem größten k.

3.3 Beweis: Clique ist NP-vollständig

Der Bewies, dass Clique NP-vollständig ist, teilt sich in 2 Teile auf:

1. Clique ist in NP enthalten

2. SAT $\leq_p$ Clique

Die Beweise werden in den nächsten 2 Kapiteln durchgeführt.

3.3.1 Beweis: Clique ist in NP enthalten

Das Clique-Problem kann mit einer NTM gelöst werden. Dies geht wie folgt vor:

1. NTM zählt Anzahl n der Knoten im Graphen
2. Rät Wort $w \in \{0,1\}^n$
3. Das Wort wird als Knotenauswahl interpretiert, V' enthält alle Knoten i mit $w_i = 1$
4. Es wird getestet, ob
   (a) V' genau k Knoten beinhaltet.
   (b) G eine Clique auf V' enthält

Der Rechenaufwand hierfür ist polynomiell in der Knotenzahl n. Es wurde bereits bewiesen, dass NTM durch DTM abgebildet werden können. Die polynomielle Laufzeit des Entscheidens, die angegebenen Knoten eine Clique bilden, kombiniert mit dem Nichtdeterminismus ergibt die Problemklasse NP für das Cliquen-Problem.

3.3.2 Beweis: SAT $\leq_p$ Clique

Konstruiere einen Graphen, der mittels *Clique* ein Problem löst, welches ein *SAT*-Problem ist. Dadurch wird das SAT-Problem in ein Clique-Problem transferiert.

1. Füge für jedes Literal in den Klauseln einen Knoten hinzu.
2. Verbinde alle Literale außer folgende Kanten:
   - Klauselgruppen untereinander
   - Gegensätzliche Literale (z.B. x_1 und $\overline{x_1}$)
3. Suche eine Clique der Größe k, k ist die Anzahl der Klauseln. Da die Knoten einer Klauselgruppe nicht verbunden sind, muss aus jeder Klausel ein Literal "wahr" sein. Da die Literale in den Klauseln ODER-verknüpft sind, sind alle Klauseln erfüllt.

In Abbildung 3 auf der nächsten Seite ist ein beispielhaftes SAT-Problem als Clique-Graph dargestellt.
Der Beweis wurde [4] entnommen. Die Transformation von SAT in Clique (Schritte 1 + 2) findet in polynomieller Zeit statt, weshalb gilt: SAT $\leq_p$ Clique.
Damit ist bewiesen: Clique ist NP-vollständig.

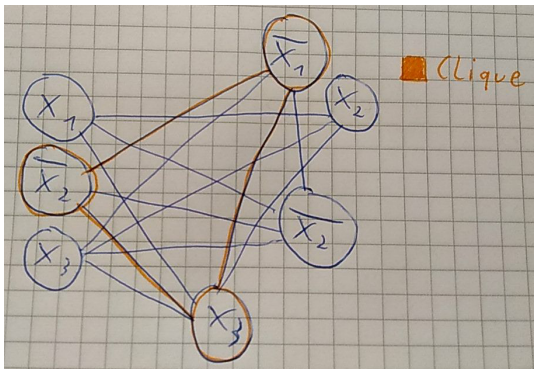

Abbildung 3: Beispielhafte Darstellung SAT-Problem als Clique-Problem

Literaturverzeichnis

[1] David Eppstein. *File:VR complex.svg*. 2007. URL: https://upload.wikimedia.org/wikipedia/commons/d/d0/VR_complex.svg (besucht am 17. 03. 2019) (siehe S. 6).

[2] Erin Silversmith. *Datei:6n-graf-clique.svg*. 2006. URL: https://upload.wikimedia.org/wikipedia/commons/8/86/6n-graf-clique.svg (besucht am 17. 03. 2019) (siehe S. 5).

[3] Ingo Wegener. *Theoretische Informatik. Eine algorithmenorientierte Einführung*. Teubner, 2005 (siehe S. 4).

[4] Weitz. *Clique ist NP-vollständig*. 2017. URL: https://www.youtube.com/watch?v=D3gkCTRMcKU (besucht am 25. 02. 2019) (siehe S. 7).