

Unerfüllbarkeit „langer“ Formeln der Aussagenlogik

Bachelorarbeit
am Fachbereich Mathematik und Informatik
der Philipps-Universität Marburg

vorgelegt von
Johanna Wiehe

März 2015
Betreuer: Prof. Dr. Volkmar Welker

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	4
2.1	Aussagenlogik	4
2.2	Stochastik	9
2.3	Graphentheorie	12
3	Erfüllbarkeit in der Aussagenlogik	16
3.1	Lokale und globale Eigenschaften	16
3.2	Das Lovász Local Lemma	20
3.3	Binärbäume und KNF-Formeln	24
4	Schlussbetrachtung	31

1 Einleitung

In dieser Arbeit betrachten wir Formeln der Aussagenlogik. Das sind Aussagen, die entweder wahr oder falsch sind. Die Inhalte der Aussagen sind dabei nicht von Belang, es geht nur um deren Wahrheitswert. Durch bestimmte aussagenlogische Verknüpfungen können Aussagen miteinander verbunden werden. Dabei wird der Wahrheitswert jeder zusammengesetzten Aussage eindeutig durch die Wahrheitswerte ihrer Teilaussagen bestimmt. Je nach Wahrheitsgehalt der Teilaussagen kann dann eine Aussage, völlig unabhängig von ihrer inhaltlichen Bedeutung, mal als wahr und mal als nicht wahr interpretiert werden. Es gibt aber auch Aussagen, die immer wahr und welche, die immer falsch sind.

Wissen wir, dass eine Aussage nie wahr werden kann, so wissen wir gleichzeitig auch, dass deren Negation immer wahr ist. So lassen sich auch Zusammenhänge zwischen immer falschen (später: unerfüllbaren) und zumindest einmal wahren (später: erfüllbaren) Formeln finden.

Diese Arbeit behandelt notwendige syntaktische Eigenschaften beliebig langer unerfüllbarer Formeln und die daraus resultierenden hinreichenden Bedingungen für die Erfüllbarkeit aussagenlogischer Formeln. Dabei gehen wir wie folgt vor:

Im Kapitel „Grundlagen“ werden zunächst Hilfsmittel auch aus den vorerst vermeintlich weit entfernten mathematischen Gebieten Stochastik und Graphentheorie bereitgestellt, sowie Verbindungen zur Aussagenlogik und zum Problem der Erfüllbarkeit hergestellt.

Diese werden dann im Kapitel „Erfüllbarkeit in der Aussagenlogik“ benötigt, um einige Erkenntnisse über hinreichende Bedingungen der Erfüllbarkeit zu formulieren und anschließend zu beweisen.

Diese Bedingungen können sich sowohl global als auch lokal auf die Formel beziehen und durch reine Existenzaussagen oder durch konstruktive Algorithmen bewiesen werden.

Im Kapitel „Lokale und globale Eigenschaften“ wird außerdem noch eine Methode vorgestellt, mit der man die Unerfüllbarkeit bestimmter Formeln (sogenannter KNF-Formeln) testen kann. Damit wird sich zeigen, dass eine unerfüllbare KNF-Formel nicht nur eine gewisse Länge haben muss, sondern auch eine gewisse innere Abhängigkeit der einzelnen Terme untereinander verlangt.

Das *Lovász Local Lemma*, eine eigentlich probabilistische Aussage, beschäftigt sich näher mit dieser inneren Abhängigkeit und stellt eines der Hauptthemen dieser Arbeit dar.

Am Ende werden wir im Kapitel „Binärbäume und KNF-Formeln“ einen konstruktiven Algorithmus kennenlernen, mit dessen Hilfe man sowohl minimal

unerfüllbare also auch erfüllbare Formeln konstruieren kann.

Allgemein orientiert sich diese Arbeit an der unter [2] angegebenen Quelle. Die dort vorausgesetzten Grundlagen und Querverbindungen zu anderen Gebieten wurden hier hinzugefügt, sowie Beweise und Erklärungen vervollständigt und erstellt.

2 Grundlagen

In diesem Kapitel werden die wichtigsten Grundlagen aus den drei Themengebieten Aussagenlogik, Stochastik und Graphentheorie, auf denen die folgende Arbeit aufbaut, erarbeitet. Vorausgesetzt werden lediglich Grundkenntnisse der Linearen Algebra und Analysis.

2.1 Aussagenlogik

Wir beginnen nun mit der Einführung der für diese Arbeit wichtigsten Begriffe der Aussagenlogik. Die Aussagenlogik ist ein Teilgebiet der Logik, in dem Aussagen unabhängig von deren Inhalt nur im Hinblick auf deren Wahrheitsgehalt untersucht werden.

Wie jede Sprache kann man auch die Aussagenlogik aus zwei Blickwinkeln betrachten. Während sich die Syntax mit dem formalen Aufbau von Formeln beschäftigt, geht es in der Semantik um die Bedeutung der Formeln, speziell um deren Wahrheitswerte.

Ausgehend von Elementaraussagen, das sind Aussagen die entweder wahr oder falsch sind und sich nicht weiter zerlegen lassen, und Junktoren, die diese verknüpfen, lässt sich die Menge der aussagenlogischen Formeln induktiv aufbauen.

Sei dazu $V = \{A_1, \dots, A_n\}$ eine endliche nichtleere Menge solcher Elementaraussagen, auch Atome genannt. Wir verwenden die Symbole \neg, \wedge, \vee für die aussagenlogischen Verknüpfungen Negation, Konjunktion und Disjunktion.

Wir definieren die Menge der aussagenlogischen Formeln über V :

Jedes Atom aus V ist eine aussagenlogische Formel über V . Diese Atome nennen wir ab jetzt atomare Formeln. Weiter ist jede Verknüpfung aussagen-

logischer Formeln wieder eine Formel. Formal:

$$\begin{aligned}
 At &::= A_1 | A_2 | \dots \\
 Fm &::= At \\
 &| \neg Fm \\
 &| (Fm \wedge Fm) \\
 &| (Fm \vee Fm)
 \end{aligned}$$

Zwecks besserer Lesbarkeit lassen wir ab jetzt gelegentlich die Außenklammern in Formeln weg.

Semantisch gesehen nimmt jede Formel genau einen der Wahrheitswerte „wahr“ oder „falsch“ an. Dabei halten wir uns an die Konvention, „wahr“ mit der 1 und „falsch“ mit der 0 zu identifizieren.

Man definiert: Eine Belegung $\alpha: V \rightarrow \{0, 1\}$ ist eine Abbildung, die jeder atomaren Formel A aus V einen Wahrheitswert zuordnet. Man erweitert α auf die Menge der aussagenlogischen Formeln über V indem man für Formeln F_1 und F_2 induktiv setzt:

$$\begin{aligned}
 \alpha(\neg F_1) &:= 1 - \alpha(F_1) \\
 \alpha(F_1 \wedge F_2) &:= \min_{i=1,2} \alpha(F_i) \\
 \alpha(F_1 \vee F_2) &:= \max_{i=1,2} \alpha(F_i)
 \end{aligned}$$

Man sagt, α erfüllt eine Formel F , wenn $\alpha(F) = 1$ gilt. F heißt erfüllbar, wenn es eine Belegung α gibt, die F erfüllt, andernfalls heißt F unerfüllbar. Eine Tautologie ist eine Formel F , die von jeder Belegung erfüllt wird, F heißt dann auch gültig.

Beispiel : $F_1 = (A_1 \vee \neg A_1)$ ist erfüllbar und sogar gültig.

$F_2 = (A_1 \wedge \neg A_1)$ ist unerfüllbar und somit nicht gültig.

$F_3 = (A_1 \vee A_2)$ ist erfüllbar aber nicht gültig.

Um zwei Formeln semantisch sinnvoll zu vergleichen, verwendet man folgende Definition:

Zwei Formeln F und G heißen (semantisch) äquivalent, wenn für jede Belegung α gilt: $\alpha(F) = \alpha(G)$. In diesem Fall schreiben wir $F \equiv G$. Insbesondere sind alle Tautologien, sowie alle unerfüllbaren Formeln zueinander äquivalent. Weiter gelten folgende Regeln:

Doppelnegation $\neg\neg F \equiv F$

Assoziativität $\begin{cases} ((F \wedge G) \wedge H) \equiv (F \wedge (G \wedge H)) , \text{ abkürzend: } F \wedge G \wedge H \\ ((F \vee G) \vee H) \equiv (F \vee (G \vee H)) , \text{ abkürzend: } F \vee G \vee H \end{cases}$

Idempotenz $\begin{cases} (F \wedge F) \equiv F \\ (F \vee F) \equiv F \end{cases}$

Kommutativität $\begin{cases} (F \wedge G) \equiv (G \wedge F) \\ (F \vee G) \equiv (G \vee F) \end{cases}$

de Morgan $\begin{cases} \neg(F \wedge G) \equiv (\neg F \vee \neg G) \\ \neg(F \vee G) \equiv (\neg F \wedge \neg G) \end{cases}$

Beweis: Beispielhaft für Doppelnegation per Definition:

$$\alpha(\neg\neg F) = 1 - \alpha(\neg F) = 1 - (1 - \alpha(F)) = \alpha(F)$$

Und die 1. Regel von de Morgan per Wahrheitstafel:

F	G	$F \wedge G$	$\neg(F \wedge G)$	$\neg F$	$\neg G$	$\neg F \vee \neg G$
1	1	1	0	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	0	1
0	0	0	1	1	1	1

Hier sind links alle möglichen Belegungen von F und G eingetragen und rechts die entsprechenden Werte von $\neg(F \wedge G)$ und $\neg F \vee \neg G$. Diese stimmen in jeder Zeile überein, daher gilt: $\alpha(\neg(F \wedge G)) = \alpha(\neg F \vee \neg G)$. ■

Sei nun $\bar{V} := \{\neg A \mid A \in V\}$ die Menge der negierten atomaren Formeln aus V. Dann nennen wir die Elemente von $V \cup \bar{V}$ Literale über V, wobei die Elemente von V positive und die von \bar{V} negative Literale heißen. Da $\neg\neg A$ semantisch äquivalent zu A ist, betrachten wir auch mehrfach negierte Atome als Literale, verwenden der Einfachheit halber aber nur semantisch äquivalente atomare Formeln bzw. deren Negation.

Eine Klausel $C \subset (V \cup \bar{V})$ ist nun eine endliche Menge von Literalen. Ein Disjunktionsterm ist wiederum eine Disjunktion einzelner Literalen, wobei jedes Literal maximal einmal vorkommt und zwar entweder positiv oder negativ. Jeder Disjunktionsterm entspricht einer Klausel, umgekehrt gilt jedoch

nicht, dass jede Klausel einem Disjunktionsterm entspricht, da in einer Klausel dasselbe Literal positiv und negativ auftauchen kann.

Beispiel : $C = \{A_1, \neg A_2, A_3\} \hat{=} (A_1 \vee \neg A_2 \vee A_3)$ ist ein Disjunktionsterm,
aber $D = \{A_1, A_2, \neg A_2\} \hat{=} (A_1 \vee A_2 \vee \neg A_2)$ ist kein Disjunktionsterm.

Ist $C = \{L_1, \dots, L_n\}$ eine Klausel, so schreiben wir $\overline{C} := \{\neg L_1, \dots, \neg L_n\}$ und $\neg C := \neg(L_1 \vee \dots \vee L_n)$. Beachte: Für $n > 1$ gilt wegen der Rechenregel von de Morgan $\overline{\overline{C}} \neq C$.

Da sie später eine wichtige Rolle spielen, werden an dieser Stelle noch zwei Extremfälle definiert: Die leere Klausel ist eine leere Disjunktion und wird als unerfüllbar definiert und die leere Formel ist eine leere Konjunktion und wird als gültig definiert. Später wird sich mithilfe stochastischer Methoden zeigen, wieso diese Definitionen sinnvoll sind.

Eine Formel F heißt in konjunktiver Normalform (KNF), wenn sie eine Konjunktion von Disjunktionstermen ist. Insbesondere ist die leere Formel als Konjunktion von null Disjunktionstermen in konjunktiver Normalform. Fasst man die Disjunktionsterme als Klauseln auf, so entspricht F der Menge dieser Klauseln. Wenn man dann von Elementen der Formel spricht, so sind die Elemente der entsprechenden Klauselmengen, also die Klauseln gemeint.

Beispiel :

$$F = (A_1 \vee A_2 \vee \neg A_3) \wedge (A_1 \vee A_2 \vee A_3) \wedge (A_1 \vee A_4 \vee \neg A_5) \wedge (\neg A_1 \vee A_3 \vee \neg A_4) \\ \hat{=} \{\{A_1, A_2, \neg A_3\}, \{A_1, A_2, A_3\}, \{A_1, A_4, \neg A_5\}, \{\neg A_1, A_3, \neg A_4\}\}$$

Eine Belegung α erfüllt eine Klausel C , wenn mindestens eins der Literale aus C auf 1 abgebildet wird. Ferner erfüllt α eine KNF-Formel F , wenn alle Klauseln aus F auf 1 abgebildet werden.

Insbesondere heißt eine KNF-Formel k-KNF-Formel, wenn jede in ihr vorkommende Klausel k Elemente besitzt.

Im Beispiel ist F also eine 3-KNF-Formel. Sie ist erfüllbar, zum Beispiel für $\alpha(A_1) = \alpha(A_2) = \alpha(A_3) = 1$ und $\alpha(A_4) = \alpha(A_5) = 0$. Wir werden später sehen, dass diese Formel schon aufgrund ihrer Länge nicht unerfüllbar sein kann, da sie weniger als $2^3 = 8$ Klauseln besitzt.

Um generell sinnvoll von der „Länge“ einer k -KNF-Formel zu reden, reduzieren wir ab jetzt alle Formeln auf deren semantisch äquivalente kürzeste Form in k -KNF. Dazu werden zunächst die Disjunktionsterme so normiert, dass innere Klammern wegfallen und die Literale dem Index der Atome nach geordnet werden. Dies führt wegen der Assoziativität und Kommutativität zu semantisch äquivalenten Disjunktionstermen. Dann darf wegen der Idempotenz einer von zwei gleichen Disjunktionstermen weggelassen werden ohne

die ursprüngliche Formel semantisch zu verändern. Die Länge einer Formel in k-KNF ist dann die Anzahl der (echt verschiedenen) Disjunktionsterme. Nichts anderes passiert im Übrigen bei der Reduktion von Disjunktionstermen und KNF-Formeln auf Klauseln und Klauselmengen.

Je nach Kontext werden wir Disjunktionsterme und Formeln in konjunktiver Normalform manchmal als Klauseln bzw. Klauselmengen und manchmal als aussagenlogische Formeln betrachten. Dies schlägt sich dann in der Notation nieder, indem wir $F \cup \{C\}$ oder $F \wedge C$ schreiben.

Da sich jede Formel der Aussagenlogik in eine semantisch äquivalente Formel in k-KNF umformen lässt, kann man immer ohne Einschränkung annehmen, dass die Formel in k-KNF vorliegt. Dabei kann die Umformung mittels semantischer Äquivalenzen oder mittels Wahrheitstabellen erfolgen. Genauer wird in [4] beschrieben.

Ein wichtiges Problem der Aussagenlogik ist das sogenannte Erfüllbarkeitsproblem SAT (aus dem Englischen „satisfiability“). Es fragt, ob eine in konjunktiver Normalform gegebene Formel erfüllbar ist. Zwar gibt es einen Algorithmus, der eine korrekte Antwort liefert, die Laufzeit kann aber gegebenenfalls sehr lang sein. Zum Beispiel kann man zur gegebenen Formel F eine Wahrheitstafel, in der alle möglichen Belegungen aufgeführt sind, erstellen und jede Belegung einmal ausprobieren. Dann erhält man in jedem Fall eine korrekte Antwort, muss dafür aber eventuell 2^n Belegungen testen, wobei n die Anzahl der in F vorkommenden atomaren Formeln ist. Damit wird der Zeitaufwand exponentiell hoch. Wir werden später noch eine weitere Methode kennenlernen, die die Unerfüllbarkeit einer KNF-Formel testet. Das *Lovász Local Lemma* bietet eine Möglichkeit, die Erfüllbarkeit einer Formel in KNF unter einer bestimmten Bedingung sicherzustellen. Dafür benötigen wir aber noch die folgenden Definitionen:

Sei $vbl(C) := \{A \mid A \in C \text{ oder } \neg A \in C\}$ die Menge aller in einer Klausel C auftretenden Atome. Weiter sei für eine Formel F in KNF

$$vbl(F) := \bigcup_{C \in F} vbl(C) = \bigcup_{C \in F} \{A \mid A \in C \text{ oder } \neg A \in C\}$$

die Menge aller in F auftretenden Atome. Damit können wir nun den wesentlichen Begriff der Nachbarschaft Γ sowie den der Konfliktnachbarschaft Γ' einer Klausel C definieren:

$$\begin{aligned} \Gamma(C) &:= \{D \in F \mid vbl(D) \cap vbl(C) \neq \emptyset\} \\ \Gamma'(C) &:= \{D \in F \mid C \cap \overline{D} \neq \emptyset\}. \end{aligned}$$

$$\text{Beispiel : } F = \{ \underbrace{\{A_1, \neg A_2, A_3\}}_{:= C_1}, \underbrace{\{\neg A_1, A_2, \neg A_4\}}_{:= C_2}, \underbrace{\{A_4, \neg A_5, A_6\}}_{:= C_3}, \underbrace{\{A_1, A_4, A_5\}}_{:= C_4} \}$$

Dann ist $\Gamma(C_1) = \{C_1, C_2, C_4\}$ und $\Gamma'(C_1) = \{C_2\} \subset \Gamma(C_1)$.

Wir werden zeigen, dass eine begrenzte Nachbarschaft, also eine begrenzte Anzahl von Klauseln, die gemeinsame Atome beinhalten, Schlüsse auf die Erfüllbarkeit einer KNF-Formel zulässt.

2.2 Stochastik

Ursprünglich ist das *Lovász Local Lemma* ein Hilfssatz aus der Wahrscheinlichkeitstheorie, einem Teilgebiet der Stochastik. Auch wenn es hier auf die Aussagenlogik angewandt wird, benötigen wir Hilfsmittel aus der Stochastik, um es zu beweisen. Diese werden nun zusammengestellt.

Zunächst definieren wir einen endlichen Wahrscheinlichkeitsraum:
Ist Ω eine nichtleere endliche Menge, $\mathcal{P}(\Omega)$ deren Potenzmenge und $P : \mathcal{P}(\Omega) \rightarrow [0, 1]$ eine Abbildung mit:

$$P(\Omega) = 1 \text{ und für je zwei disjunkte Teilmengen } X \text{ und } Y \text{ von } \Omega \text{ gilt:}$$

$$P(X \cup Y) = P(X) + P(Y).$$

Dann heißt (Ω, P) ein endlicher Wahrscheinlichkeitsraum. Die Elemente von Ω heißen Elementarereignisse, die von $\mathcal{P}(\Omega)$ heißen Ereignisse und P heißt Wahrscheinlichkeitsmaß.

Beispiel: Sei $V = \{A_1, \dots, A_n\}$ eine endliche, nichtleere Menge atomarer Formeln. Wir definieren $\Omega := \{\alpha \mid \alpha : V \rightarrow \{0, 1\}\}$ die Menge der Belegungen von V , und

$$P : \mathcal{P}(\Omega) \rightarrow [0, 1] \text{ mit } P(X) := \frac{|X|}{2^n}.$$

Dann ist (Ω, P) ein endlicher Wahrscheinlichkeitsraum, denn:

$$P(\Omega) = \frac{|\Omega|}{2^n} = \frac{2^n}{2^n} = 1$$

und für zwei disjunkte Ereignisse X und Y aus $\mathcal{P}(\Omega)$ gilt:

$$P(X \cup Y) = \frac{|X \cup Y|}{2^n} = \frac{|X| + |Y|}{2^n} = \frac{|X|}{2^n} + \frac{|Y|}{2^n} = P(X) + P(Y).$$

Da jedes Elementarereignis mit gleicher Wahrscheinlichkeit eintritt, spricht man auch von einer Gleichverteilung auf Ω .

Mit folgender Interpretation wird dieser Wahrscheinlichkeitsraum noch häufig benutzt:

Für eine Formel F über V ist $\{\alpha | \alpha(F) = 1\}$ das Ereignis, das F erfüllt und die Wahrscheinlichkeit dieses Ereignisses ist $P(\{\alpha | \alpha(F) = 1\})$. Hierfür schreiben wir auch abkürzend $P(F)$.

$P(F)$ gibt also die Wahrscheinlichkeit an, dass eine zufällig gewählte Belegung von V die Formel F erfüllt. Ist $P(F) = 0$, so gibt es keine Belegung, die F erfüllt, ist $P(F) > 0$ gibt es eine, und ist $P(F) = 1$, so ist F gültig.

Konkret gilt für eine k -elementige Klausel C aus einer KNF-Formel über V :

$$P(\neg C) = P(\{\alpha | \alpha(\neg C) = 1\}) = \frac{|\{\alpha | \alpha(\neg C) = 1\}|}{2^n} = \frac{2^{n-k}}{2^n} = \frac{1}{2^k},$$

da es nur eine Belegung gibt, die $\neg C$ über $\text{vbl}(C)$ erfüllt und somit 2^{n-k} Belegungen, die $\neg C$ über V erfüllen. Ist $k=0$, so erhalten wir

$$P(\neg C) = \frac{1}{2^0} = \frac{1}{1} = 1.$$

Dies deckt sich mit der Definition der leeren Formel, da eine negierte Klausel eine Konjunktion darstellt und die leere Konjunktion als gültig definiert wurde.

In Wahrscheinlichkeitsräumen gelten folgende Rechenregeln für $A, B \in \mathcal{P}(\Omega)$:

$$\begin{aligned} P(A^c) &= 1 - P(A) \\ P(A \setminus B) &= P(A) - P(A \cap B) \\ P(A \cup B) &= P(A) + P(B), \text{ falls } A \cap B = \emptyset \\ P(A) &\leq P(B), \text{ falls } A \subset B \end{aligned}$$

Einen Beweis findet man in [3].

Beispiel(Fortsetzung): Übertragen auf den obigen Wahrscheinlichkeitsraum ergeben sich folgende Regeln für Formeln F, G und Klauselmengen C, D :

$$\begin{aligned} P(\neg F) &= 1 - P(F) \\ P(F \wedge \neg G) &= P(F) - P(F \wedge G) \\ P(C \vee D) &= P(C) + P(D), \text{ falls } C \cap D = \emptyset \\ \text{Ist } D \subset C, \text{ so folgt } \{\alpha | \alpha(C) = 1\} &\subset \{\alpha | \alpha(D) = 1\} \text{ und daher } P(C) \leq P(D). \end{aligned}$$

Somit gilt insbesondere für eine k-elementige Klausel C einer KNF-Formel:

$$P(C) = 1 - P(\neg C) = 1 - \frac{1}{2^k}.$$

Ist $k=0$, so erhalten wir hier

$$P(C) = 1 - \frac{1}{2^0} = 1 - \frac{1}{1} = 0.$$

Dies deckt sich mit der Definition der leeren Klausel, die als unerfüllbar definiert wurde.

Ein wichtiger Begriff der Wahrscheinlichkeitstheorie ist der der stochastischen Unabhängigkeit. Zwei Ereignisse sollen stochastisch unabhängig voneinander sein, wenn sich die Wahrscheinlichkeit, dass das eine Ereignis eintritt nicht dadurch ändert, dass das andere eingetreten ist bzw. nicht eingetreten ist.

Formal:

$A, B \in \mathcal{P}(\Omega)$ heißen stochastisch unabhängig, falls $P(A \cap B) = P(A)P(B)$ gilt.

Beispiel(Fortsetzung): Gegeben sei der Wahrscheinlichkeitsraum aus obigem Beispiel. Dann sind die Ereignisse $\{\alpha | \alpha(F) = 1\}$ und $\{\alpha | \alpha(G) = 1\}$ stochastisch unabhängig, wenn die Formeln F und G keine gemeinsamen Atome besitzen, also wenn $vbl(F) \cap vbl(G) = \emptyset$ gilt.

Abkürzend sagen wir F und G sind stochastisch unabhängig und es gilt

$$P(F \wedge G) = P(F)P(G).$$

Umgekehrt gilt diese Aussage jedoch nicht, da zum Beispiel für die Tautologie $F = (A_1 \vee \neg A_1)$ gilt:

$$P(F \wedge F) = P(F) = 1 = P(F)P(F),$$

F also von sich selbst stochastisch unabhängig wäre, obwohl offenbar $vbl(F) \cap vbl(F) \neq \emptyset$ gilt.

2.3 Graphentheorie

Um die Struktur einer aussagenlogischen Formel in konjunktiver Normalform zu veranschaulichen, bedienen wir uns der Graphentheorie, welche Zusammenhänge bildlich darstellt und somit auch lokale Verknüpfungen verdeutlicht. Es folgt eine kurze Einführung in die Graphentheorie und der Übergang zur Erfüllbarkeit aussagenlogischer Formeln:

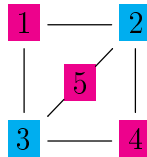
Ein Hypergraph H ist ein Paar (V, E) , bestehend aus einer endlichen, nicht-leeren Menge V und einer Menge $E \subset \mathcal{P}(V)$, mit $|e| > 1$ für alle e aus E . Die Elemente von V heißen Ecken, die von E Kanten. Eine Kante $e \in E$ verbindet alle Ecken, die in e enthalten sind und hat keine Richtung.

Gilt $|e| = k$ für alle $e \in E$, so heißt H k-uniform.

Weiter heißt H 2-färbbar, wenn es eine Abbildung $f : V \rightarrow \{\text{blau}, \text{rot}\}$ gibt, sodass jede Kante von H eine Ecke v und eine Ecke w enthält mit $f(v)=\text{blau}$ und $f(w)=\text{rot}$. f heißt dann (gültige) 2-Färbung von H .

Beispiel:

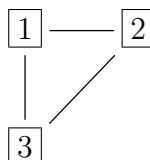
- Sei $H=(V,E)$ mit $V=\{1, 2, 3, 4, 5\}$
und $E=\{\{1, 2\}, \{2, 4\}, \{3, 4\}, \{3, 5\}, \{1, 3\}, \{2, 5\}\}$



Dieser Hypergraph ist 2-uniform und 2-färbbar mit der Färbung $f : V \rightarrow \{\text{blau}, \text{rot}\}$

$$f(x) = \begin{cases} \text{blau} & , \text{ falls } x \in \{2, 3\} \\ \text{rot} & , \text{ falls } x \in \{1, 4, 5\}. \end{cases}$$

- $V = \{1, 2, 3\}, E = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$



Dieser Hypergraph ist 2-uniform aber nicht 2-färbbar.

Zum Zusammenhang zwischen einem 2-färbbaren Hypergraphen und einer erfüllbaren KNF-Formel:

Betrachtet man die Menge der Ecken V eines Hypergraphen $H=(V,E)$ als atomare Formeln $\{A_1, \dots, A_n\}$ und die Menge der Kanten E als Klauselmengen, so kann man eine KNF-Formel als Klauselmengen wie folgt definieren:

$$F = F(H) := E \cup \{\overline{C} \mid C \in E\}.$$

F entspricht einer KNF-Formel über $V = \text{vbl}(E)$ und ist C eine Klausel aus F , so auch \overline{C} .

Satz 2.1 *Ist $F = E \cup \{\overline{C} \mid C \in E\}$ erfüllbar, so ist der Hypergraph $H=(V,E)$ 2-färbbar.*

Beweis: Ist F erfüllbar, so gibt es eine Belegung $\alpha : V \rightarrow \{0, 1\}$ mit $\alpha(F) = 1$. Sei $f : V \rightarrow \{\text{blau}, \text{rot}\}$ definiert durch

$$f(A_i) = \begin{cases} \text{blau} & , \text{ falls } \alpha(A_i) = 1 \\ \text{rot} & , \text{ falls } \alpha(A_i) = 0. \end{cases}$$

Sei nun C eine beliebige Kante aus E .

Dann ist $C \in F$ und es gilt $\alpha(C) = 1$. Folglich enthält C ein $A_i \in V$ mit $\alpha(A_i) = 1$. Damit enthält C ein A_i mit $f(A_i) = \text{blau}$.

Ferner ist $\overline{C} \in F$ und es gilt $\alpha(\overline{C}) = 1$. Folglich enthält \overline{C} ein $\neg A_j, j \neq i$ mit $\alpha(\neg A_j) = 1$. Also enthält C ein A_j mit $f(A_j) = \text{rot}$.

Somit enthält jede Kante von H zwei Ecken A_i, A_j mit $f(A_i) = \text{blau}$ und $f(A_j) = \text{rot}$. f ist also eine gültige 2-Färbung von H und H ist 2-färbbar. ■

Umgekehrt gilt:

Satz 2.2 *Ist $H=(V,E)$ ein 2-färbbarer Hypergraph, so ist die Klauselmengen $F = F(H) := E \cup \{\overline{C} \mid C \in E\}$ erfüllbar.*

Beweis: Ist H 2-färbbar, so existiert eine Färbung $f : V \rightarrow \{\text{blau}, \text{rot}\}$, sodass jede Kante in E zwei Ecken A_i, A_j enthält mit $f(A_i) = \text{blau}$ und $f(A_j) = \text{rot}$. Sei $\alpha : V \rightarrow \{0, 1\}$ wie folgt definiert:

$$\alpha(A) := \begin{cases} 1 & , \text{ falls } f(A) = \text{blau} \\ 0 & , \text{ falls } f(A) = \text{rot}. \end{cases}$$

Sei nun D eine beliebige Klausel aus F .

Fall 1: $D \in E$

D enthält also ein A_i mit $f(A_i) = \text{blau}$. Somit gibt es ein Literal in D , nämlich A_i , mit $\alpha(A_i) = 1$.

Fall 2: $D \in \{\overline{C} \mid C \in E\}$

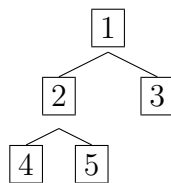
Dann gibt es eine Kante K mit $K = \overline{D}$. K enthält wiederum ein A_j mit $f(A_j) = \text{rot}$. Es gibt also ein Literal in \overline{D} , nämlich A_j , mit $\alpha(A_j) = 0$. Somit gibt es auch ein Literal in D , nämlich $\neg A_j$ mit $\alpha(\neg A_j) = 1$.

In beiden Fällen gilt also $\alpha(C) = 1$ und F ist erfüllbar. ■

Wir wollen nun spezielle 2-uniforme Hypergraphen $T=(V,E)$, sogenannte Binärbäume betrachten:

Beginnend bei einer obersten Ecke, Wurzel genannt, baut sich der Binärbaum nach unten auf, wobei jede Ecke entweder keine oder zwei Ecken, sogenannte Kinder, hat, die durch Kanten mit ihr verbunden werden. In einem Binärbaum gibt es keine geschlossenen „Rundwege“. Ecken, die in nur einer Kante vorkommen, heißen auch Blätter, alle anderen Ecken heißen (innere) Knoten.

Beispiel : $T = (V, E)$ mit $V = \{1, 2, 3, 4, 5\}$, $E = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}\}$



T ist ein Binärbaum. Dabei ist 1 die Wurzel, 1 und 2 sind innere Knoten, 3, 4 und 5 sind Blätter.

Angelehnt an die in einem Stammbaum üblichen Begriffe verwenden wir auch hier den Begriff der Vorfahren einer Ecke v für die Knoten die sich auf dem Weg von v zur Wurzel befinden und den der Geschwister für Kinder mit denselben Vorfahren. Im Beispiel sind 1 und 2 Vorfahren der Geschwister 4 und 5. Sei u nun eine Ecke und v ein Vorfahre von u .

Der Abstand $\text{dist}(u,v)$ zwischen u und v sei die (minimale) Anzahl der dazwischenliegenden Kanten. Da die Wurzel W für jedes Blatt u der am weitesten entfernte Vorfahre von u ist, wird $\text{dist}(u,v)$ für $v=W$ maximal.

Die Tiefe $t(v)$ einer Ecke $v \in V$ sei nun die (minimale) Anzahl der Kanten zwischen v und der Wurzel W , also $t(v):=\text{dist}(v,W)$.

Es gilt immer $t(W)=0$, im Beispiel gilt $t(2)=t(3)=1$ und $t(4)=t(5)=2$.

Wir betrachten nun eine spezielle Klasse von Binärbäumen, sogenannte (k,d) -Bäume. Diese sind dadurch charakterisiert, dass ihre Blätter besonders tief aber dennoch nah an jedem inneren Knoten liegen:

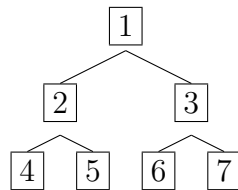
Für $k, d \in \mathbb{N}$ heißt $T=(V,E)$ ein (k,d)-Baum, falls

1. jedes Blatt mindestens Tiefe $k-1$ hat, also:
 $t(u) \geq k-1$ für alle Blätter u und es
2. für jeden Knoten maximal d Blätter gibt, die höchstens $k-1$ Kanten entfernt liegen, also: für jeden Knoten $v \in V$ gilt:
 $|\{ u \mid u \text{ ist Blatt mit } \text{dist}(u,v) \leq k-1 \}| \leq d$.

Beispiel:

1. Der Binärbaum im obigen Beispiel ist ein (2,2)-Baum. Denn:
 $t(3) = 1 \geq 2 - 1 = 1$, $t(4) = t(5) = 2 \geq 1$ und
für den Knoten 1 ist die Anzahl der Blätter u mit $\text{dist}(u,1) \leq 1$ gleich 1 (nämlich für $u=3$) und
für den Knoten 2 ist die Anzahl der Blätter u mit $\text{dist}(u,2) \leq 1$ gleich 2 (nämlich für $u=4$ und $u=5$).

2.



Dies ist ein (3,4)-Baum, da $t(u) \geq 2$ für jedes Blatt u gilt, und es insgesamt nur 4 Blätter gibt.

Für später benötigen wir noch folgenden

Satz 2.3 Ist b_T die Anzahl der Blätter und k_T die Anzahl der inneren Knoten eines Binärbaums T , so gilt:

$$k_T = b_T - 1$$

Beweis: per Induktion über k_T :

Induktionsanfang : $k_T = 0$

Es gibt keinen inneren Knoten, demnach muss es wegen $V \neq \emptyset$ ein Blatt, die Wurzel, geben:

$$k_T = 0 = 1 - 1 = b_T - 1$$

Induktionsvoraussetzung : Für Binärbäume mit k_T inneren Knoten gilt:

$$b_T = k_T + 1$$

Induktionsschritt : $k_T \rightarrow k_T + 1$

T' sei ein Binärbaum mit $k_{T'} := k_T + 1$ inneren Knoten. Nach Induktionsvoraussetzung gibt es also vorerst $k_T + 1$ Blätter, wobei durch das Anfügen des $k_T + 1$ -ten Knotens ein Blatt zunächst abgezogen, dafür aber zwei neue addiert werden. Für die Anzahl der Blätter $b_{T'}$ von T' gilt somit:

$$b_{T'} = k_T + 1 - 1 + 2 = k_T + 2 = k_{T'} + 1$$

■

3 Erfüllbarkeit in der Aussagenlogik

In diesem Kapitel werden zunächst zwei kleinere Sätze vorgestellt, die sich mit der Erfüllbarkeit von k -KNF-Formeln unter gewissen Voraussetzungen beschäftigen. Diese Voraussetzungen können sich global oder lokal auf die Formel beziehen. Satz 3.1 ist ein Beispiel für eine globale, Satz 3.2 für eine lokale Eigenschaft.

3.1 Lokale und globale Eigenschaften

Satz 3.1 *Jede unerfüllbare k -KNF-Formel hat mindestens 2^k Klauseln.*

Beweis: Sei $F = \{C_1, \dots, C_m\}$, $m < 2^k$ eine Klauselmengung, die einer k -KNF-Formel über V entspricht und $\alpha : V \rightarrow \{0, 1\}$ eine zufällig gewählte Belegung. Im Bezug auf den im Abschnitt 2.2 definierten Wahrscheinlichkeitsraum (Ω, P) betrachten wir die Wahrscheinlichkeit, dass α eine k -elementige Klausel

$C_i \in F$ nicht erfüllt:

$$P(\neg C_i) = \frac{|\{\alpha \mid \alpha(C_i) = 0\}|}{2^k} = \frac{1}{2^k} .$$

Damit ist die Wahrscheinlichkeit, dass F nicht von α erfüllt wird

$$P(\neg F) = P(\neg C_1 \vee \dots \vee \neg C_m) = \sum_{i=1}^m P(\neg C_i) = \sum_{i=1}^m \frac{1}{2^k} = \frac{m}{2^k} < 1.$$

Umgekehrt ist also die Wahrscheinlichkeit, dass F von α erfüllt wird

$$P(F) = 1 - P(\neg F) > 0$$

und es gibt eine Belegung, die F erfüllt. ■

Satz 3.2 Ist F eine Klauselmengemenge, die einer k -KNF-Formel entspricht, wobei je zwei Klauseln entweder gar keinen oder mindestens zwei Konflikte haben, so ist F erfüllbar.

Um diese Aussage beweisen zu können, benötigen wir noch eine Methode aus der Aussagenlogik: Die Resolventenmethode:

Die Resolventenmethode ist ein rein syntaktisches Verfahren, das die Erfüllbarkeit einer Klauselmengemenge testet.

Seien C_1 und C_2 zwei Klauseln. Gibt es $L_1 \in C_1$ und $L_2 \in C_2$, so dass L_1 das positive und L_2 das negative Literal zur selben atomaren Formel ist, dann heißt

$$R := (C_1 \setminus \{L_1\}) \cup (C_2 \setminus \{L_2\})$$

eine Resolvente von C_1 und C_2 .

Beispiel: $C_1 = \{A_1, A_2, A_3\}$, $C_2 = \{\neg A_1, A_2, \neg A_3\}$

Dann ist mit $L_1 = A_1 \in C_1$, $L_2 = \neg A_1 \in C_2$

$$R_1 = (C_1 \setminus \{L_1\}) \cup (C_2 \setminus \{L_2\}) = \{A_2, A_3, \neg A_3\}$$

eine Resolvente von C_1 und C_2 . Mit $L_1 = A_3$ und $L_2 = \neg A_3$ ist

$$R_2 = \{A_1, A_2, \neg A_1\} \text{ eine weitere Resolvente.}$$

In dem Fall, dass $C_1 = \{L\}$ und $C_2 = \{\neg L\}$ gilt, ist die Resolvente beider Klauseln die leere Klausel, die mit dem Symbol \square bezeichnet wird. Sie ist per Definition unerfüllbar und somit auch jede Klauselmengemenge, die sie enthält. Außerdem wird zu einer Klauselmengemenge F

$$Res(F) := F \cup \{R \mid R \text{ ist Resolvente zweier Klauseln in } F\} \text{ definiert.}$$

Weiter setzen wir induktiv

$$Res^0(F) := F,$$

$$Res^{n+1}(F) := Res(Res^n(F)) \text{ für } n \geq 0 \text{ und schließlich}$$

$$Res^*(F) := \bigcup_{n \geq 1} Res^n(F).$$

Beispiel: $K = \{\{A_1, A_2\}, \{\neg A_1, A_2\}, \{A_1, \neg A_2\}, \{\neg A_1, \neg A_2\}\}$ Dann ist

$$Res^0(K) = K,$$

$$Res^1(K) = K \cup \{\{A_2\}, \{\neg A_2\}, \{A_2, \neg A_2\}, \{A_1\}, \{\neg A_1\}, \{A_1, \neg A_1\}\},$$

und schließlich $Res^2(K) = Res^1(K) \cup \{\square\} = \dots = Res^*(K)$.

Die Methode ist nun folgende: Man resolviert so lange je zwei Klauseln, in denen dasselbe Literal einmal positiv und einmal negativ vorkommt, bis entweder die leere Klausel resolviert wird oder keine Resolution mehr möglich ist. Im letzten Fall ist die ursprüngliche Klauselmenge erfüllbar, im ersten nicht.

Dies sagt der

Resolutionssatz: *Eine endliche Klauselmenge K ist genau dann unerfüllbar, wenn $\square \in Res^*(K)$.*

Einen Beweis findet man in [4].

Nun zum Beweis von Satz 3.2:

Sei $F = \{C_1, \dots, C_n\}$ eine Klauselmenge, die einer k-KNF-Formel entspricht, wobei je zwei Klauseln entweder gar keinen oder mindestens zwei Konflikte haben. Wir zeigen, dass die leere Klausel mit der Resolventenmethode nicht resolviert werden kann. Daraus folgt dann mit dem Resolutionssatz die Erfüllbarkeit.

Seien C_i und C_j zwei beliebige Klauseln aus F . Haben diese keinen Konflikt, so kann man nicht resolvidieren, insbesondere nicht die leere Klausel.

Ohne Einschränkung seien also L_1 und L_2 zwei Konflikte von C_i und C_j mit $L_1, L_2 \in C_i$ und $\neg L_1, \neg L_2 \in C_j$. Dann ist

$$R = (C_i \setminus \{L_1\}) \cup (C_j \setminus \{\neg L_1\})$$

eine Resolvente von C_i und C_j . Sie enthält sowohl $L_2 \in C_i$ als auch $\neg L_2 \in C_j$ und ist daher eine Tautologie. Da wir zwei beliebige Klauseln resolviert haben, sind alle Resolventen zweier Klauseln mit mindestens zwei Konflikten Tautologien.

$Res^1(F)$ lässt sich demnach schreiben als $S_1 \cup T_1$, wobei in S_1 nur Oberklauseln, das sind Klauseln, die mindestens eine Klausel aus F als Teilmenge enthalten, und in T_1 nur Tautologien enthalten sind.

Lassen sich nun zwei Klauseln C und D aus $Res^1(F)$ resolvidieren, so gibt es drei Fälle:

1. Fall *beide Klauseln sind aus S_1*

Seien $C, D \in S_1$, etwa:

$$C = C_i \cup X \text{ und } D = C_j \cup Y \text{ mit } C_i, C_j \in F$$

(Dabei sind X und Y gegebenenfalls leer.)

Gibt es nun zwischen C und D mehr als einen Konflikt, so sind alle Resolventen Tautologien.

Es gebe also genau einen Konflikt zwischen C und D . Da C_i und C_j nach Voraussetzung keinen oder mindestens zwei Konflikte haben, muss dieser Konflikt zwischen X und C_j , Y und C_i oder X und Y liegen.

Alle Resolventen beinhalten dann immer noch C_i , C_j oder C_i und C_j , sind also Oberklauseln.

2. Fall *beide Klauseln sind aus T_1*

Dann enthalten beide Klauseln ein Literal sowohl positiv als auch negativ, etwa:

$$C = \{L_i, \neg L_i\} \cup X \text{ und } D = \{L_j, \neg L_j\} \cup Y$$

Jede Resolvente enthält dann weiterhin mindestens ein Literal (L_i oder L_j) sowohl positiv als auch negativ.

3. Fall *eine Klausel ist aus S_1 und eine ist aus T_1*

Sei etwa

$$C = C_i \cup X \in S_1 \text{ und } D = \{L, \neg L\} \cup Y \in T_1$$

Nun gibt es drei Fälle:

(a) *C_i und Y haben einen Konflikt oder X und Y haben einen Konflikt:*
Dann enthält die Resolvente immer noch $\{L, \neg L\}$, ist also eine Tautologie.

(b) *X und $\{L, \neg L\}$ haben einen Konflikt:*
Dann enthält die Resolvente immer noch C_i und ist somit eine Oberklausel.

(c) *C_i und $\{L, \neg L\}$ haben einen Konflikt:*
Dann muss der Konflikt durch L oder $\neg L$ verursacht worden sein, C_i enthält also $\neg L$ oder L . Die Resolvente ist in beiden Fällen

$$(C_i \setminus \{\neg L\}) \cup (\{L, \neg L\} \setminus \{L\}) = (C_i \setminus \{L\}) \cup (\{L, \neg L\} \setminus \{\neg L\}) = C_i$$

wieder eine Oberklausel von C_i .

Somit sind alle Resolventen entweder Oberklauseln oder Tautologien, und $Res^2(F)$ lässt sich schreiben als $S_2 \cup T_2$, wobei S_2 nur aus Oberklauseln und T_2 nur aus Tautologien besteht.

Schließlich lässt sich $Res^*(F)$ als $S_* \cup T_*$ schreiben, wobei in S_* nur Oberklauseln und in T_* nur Tautologien enthalten sind.

Insgesamt sind Klauseln aus S_* nicht leer, da sie nichtleere Klauseln aus F als Teilmengen enthalten. Klauseln aus T_* sind nicht leer da sie mindestens ein Literal positiv und negativ enthalten. Die leere Klausel kann folglich nicht resolviert werden und F ist erfüllbar. ■

Zusammenfassend ist die Resolventenmethode ein weiteres Verfahren, mit dem man sowohl die Unerfüllbarkeit als auch die Erfüllbarkeit einer aussagenlogischen Formel in endlicher Zeit feststellen kann. Allerdings ist auch hier der Zeitaufwand eventuell exponentiell hoch und daher vergleichbar mit dem Zeitaufwand der Wahrheitstafelmethode. Es wurde bereits nachgewiesen, dass das Erfüllbarkeitsproblem SAT zur Klasse der sogenannten NP-vollständigen Probleme gehört. Diese lassen sich vermutlich nicht effizient lösen.

3.2 Das Lovász Local Lemma

Das *Lovász Local Lemma* ist eines der Hauptbestandteile dieser Arbeit. Es wurde 1975 von László Lovász und Paul Erdős bewiesen und findet seitdem hauptsächlich Anwendung in der Stochastik, um Existenzbeweise zu führen. Hier wird es auf aussagenlogische Formeln in konjunktiver Normalform angewandt, um deren Erfüllbarkeit zu testen. Der Name beruht darauf, dass die lokale Eigenschaft der begrenzten Nachbarschaft zu dem globalen Ergebnis, der Erfüllbarkeit, zusammengeführt wird. Γ bezeichnet die in Abschnitt 2.1 definierte Nachbarschaft, e die eulersche Zahl:

Satz 3.3 *Sei $k \in \mathbb{N}, k > 1$ und F eine k -KNF-Formel. Gilt $|\Gamma(C)| \leq \frac{2^k}{e} - 1$ für alle Klauseln $C \in F$, so ist F erfüllbar.*

Für den Beweis benötigen wir noch eine Abschätzung, die hier vorangestellt wird:

Sei $x \in (1, \infty)$. Aus der Analysis ist bekannt, dass $\exp(y) \geq y + 1$ für alle $y \in \mathbb{R}$ gilt. Daher gilt insbesondere für $y = \frac{1}{x-1}$

$$\begin{aligned} \exp\left(\frac{1}{x-1}\right) &\geq \frac{1}{x-1} + 1 = \frac{x}{x-1}, \text{ also} \\ \exp\left(\frac{-1}{x-1}\right) &\leq \frac{x-1}{x}. \end{aligned}$$

Dieser Ausdruck ist größer als Null, daher kann man den monoton wachsenden natürlichen Logarithmus \ln auf beide Seiten anwenden und erhält

$$\frac{-1}{x-1} \leq \ln\left(\frac{x-1}{x}\right) \Leftrightarrow -1 \leq (x-1) \ln\left(1 - \frac{1}{x}\right).$$

Da die Exponentialfunktion \exp monoton wachsend ist folgt weiter

$$e^{-1} \leq \exp\left((x-1) \ln\left(1 - \frac{1}{x}\right)\right).$$

Für $a > 0$ und $r \in \mathbb{R}$ gilt per Definition $a^r = \exp(r \ln(a))$ und daher insgesamt

$$e^{-1} \leq \left(1 - \frac{1}{x}\right)^{x-1}. \quad (1)$$

Nun zum eigentlichen Beweis:

(Ω, P) sei wieder der Wahrscheinlichkeitsraum aus den Beispielen aus Kapitel 2.2. Wir zeigen, dass die Wahrscheinlichkeit $P(F)$ echt größer Null ist, daraus folgt dann die Existenz einer Belegung α die F erfüllt und daraus die Behauptung.

Sei $F' \subsetneq F$ eine erfüllbare Klauselmengung und $C \in F \setminus F'$ eine Klausel. Ein solches F' mit $P(F') > 0$ existiert, wegen Satz 3.1 spätestens ab $|F'| \leq 2^k$.

Zwischenbehauptung: $P(F' \wedge \neg C) \leq e 2^{-k} P(F')$

Diese ist wegen $P(F' \wedge \neg C) = P(F') - P(F' \wedge C)$ äquivalent zu

$$\begin{aligned} P(F' \wedge \neg C) &\leq e 2^{-k} P(F') \\ \Leftrightarrow P(F') - P(F' \wedge C) &\leq e 2^{-k} P(F') \\ \Leftrightarrow P(F' \wedge C) &\geq (1 - e 2^{-k}) P(F'). \end{aligned}$$

Erweitert man also, unter Voraussetzung der Zwischenbehauptung, F' um die Klausel C , so sinkt die Wahrscheinlichkeit, dass F' erfüllt wird nicht mehr als um den konstanten Faktor $(1 - e 2^{-k}) > 0$. Erweitern wir nun sukzessive F' solange um die fehlenden Klauseln C aus F , so folgt schließlich, da $P(F') > 0$, auch $P(F) > 0$.

Die Zwischenbehauptung wird nun mittels Induktion über $|F'|$ bewiesen:

Induktionsanfang: $|F'| = 0$

Dann ist F' die leere Formel und mit $P(F') = 1$ folgt:

$$P(F' \wedge \neg C) = P(\neg C) = 2^{-k} \leq e 2^{-k} = e 2^{-k} P(F')$$

Induktionsvoraussetzung : Für alle F'' mit $|F''| < |F'|$ gelte

$$P(F'' \wedge C) \geq (1 - e 2^{-k})P(F'').$$

Induktionsschritt : $|F''| \rightarrow |F'|$

Fall 1: $vbl(F') \cap vbl(C) = \emptyset$

Da $vbl(C) = vbl(\neg C)$ sind F' und $\neg C$ stochastisch unabhängig und es gilt

$$P(F' \wedge \neg C) = P(F')P(\neg C) = 2^{-k}P(F') \leq e 2^{-k}P(F').$$

Fall 2: $vbl(F') \cap vbl(C) \neq \emptyset$

Definiere $F'' := F' \setminus \Gamma(C)$. Dann gilt $vbl(F'') \cap vbl(C) = \emptyset$ und somit

$$P(F'' \wedge \neg C) = P(F'')P(\neg C) = 2^{-k}P(F'') \quad (2)$$

Wenden wir nun die Induktionsvoraussetzung sooft auf F'' an, bis wir F'' auf F' erweitert haben, so müssen wir dies $|\Gamma(C)|$ mal tun, also nach Voraussetzung maximal $d := \frac{2^k}{e} - 1$ mal. Es folgt:

$$P(F') \geq (1 - e 2^{-k})^d P(F'')$$

Und mit der vorangestellten Abschätzung (1) mit $x = \frac{2^k}{e} > 1$ gilt:

$$P(F') \geq e^{-1}P(F'') \quad (3)$$

Da $F'' \subset F'$ und wegen (2) gilt andererseits

$$P(F' \wedge \neg C) \leq P(F'' \wedge \neg C) = 2^{-k}P(F''). \quad (4)$$

Ist $P(F'') = 0$, so folgt an dieser Stelle direkt die Zwischenbehauptung:

$$P(F' \wedge \neg C) \leq 0 \leq e 2^{-k}P(F').$$

Sei also $P(F'') > 0$, dann folgt mit (3) und (4) insgesamt

$$\frac{P(F' \wedge \neg C)}{P(F')} \leq \frac{2^{-k}P(F'')}{e^{-1}P(F'')} \Leftrightarrow P(F' \wedge \neg C) \leq e 2^{-k}P(F').$$

■

Dies ist ein nicht konstruktiver Beweis. Wir wissen jetzt, dass eine erfüllende Belegung existiert, wie genau diese aussieht oder wie man eine erzeugt jedoch nicht. In [2] wird ein Algorithmus vorgestellt, der genau dies tut. Der Beweis, dass dieser in endlicher Zeit terminiert ist allerdings etwas aufwändiger und wird daher hier nicht behandelt.

Beispiel :

1. $F = (A_1 \vee A_2 \vee A_3 \vee A_4) \wedge (\neg A_1 \vee A_2 \vee A_4 \vee A_5) \wedge (A_4 \vee A_5 \vee A_6 \vee A_7)$
 Dann ist F nach Satz 3.1 erfüllbar, da die Anzahl der Klauseln kleiner als $2^4 = 16$ ist. Außerdem gilt $|\Gamma(C)| = 3$ für alle Klauseln C aus F und weil $\frac{2^4}{e} - 1 > 4$ gilt, ist F auch nach dem Local Lemma erfüllbar. Satz 3.2 kann man hier dagegen nicht anwenden, da zum Beispiel die ersten beiden Klauseln genau einen Konflikt haben.
2. $G = (A_1 \vee A_2 \vee A_3) \wedge (\neg A_1 \vee \neg A_2 \vee A_3) \wedge (A_3 \vee A_4 \vee A_5) \wedge (\neg A_4 \vee \neg A_5 \vee A_6)$
 Hier kann man wieder mit Satz 3.1 die Erfüllbarkeit zeigen, genauso aber auch mit Satz 3.2, da je zwei Klauseln keinen oder mindestens zwei Konflikte haben. Das Local Lemma dagegen ist nicht anwendbar, da zum Beispiel $|\Gamma(\{A_1, A_2, A_3\})| = 3$ ist, das Lemma aber $|\Gamma(C)| \leq \frac{2^3}{e} - 1 < 2$ fordert.

Eine stärkere Variante ist das sogenannte *Lopsided Local Lemma*:

Satz 3.4 Sei $k \in \mathbb{N}, k > 1$ und F eine k -KNF-Formel. Gilt $|\Gamma'(C)| \leq \frac{2^k}{e} - 1$ für alle $C \in F$, so ist F erfüllbar.

Beweis: Wir gehen analog zu obigem Beweis vor, bis zum Fall 2 : $vbl(F') \cap vbl(C) \neq \emptyset$.

Definiere nun $F'' := F' \setminus \Gamma'(C)$.

Jetzt gilt allerdings nicht wie oben, dass $vbl(F'') \cap vbl(C) = \emptyset$ ist und somit $P(F'' \wedge \neg C) = P(F'')P(\neg C)$ gilt. Um im obigen Beweis bei (4) fortzufahren, genügt es allerdings zu zeigen, dass

$$P(F'' \wedge \neg C) \leq P(F'')P(\neg C)$$

gilt. Sei dazu:

$$P(F'' | \neg C) := \frac{P(F'' \wedge \neg C)}{P(\neg C)}$$

die Wahrscheinlichkeit, dass F'' erfüllt wird, unter der Bedingung, dass C nicht erfüllt wird. ($P(\neg C) > 0$, da C einem Disjunktionsterm entspricht.)

Damit ist $P(F'' \wedge \neg C) = P(F'' | \neg C)P(\neg C)$.

Sei nun $\alpha(\neg C) = 1$ gegeben. Dann gilt $\alpha(L') = 1$ für alle $L' \in \overline{C}$. Weiter sei $K \in F''$ eine Klausel und $L \in K$ ein Literal. Wegen $\Gamma'(C) \not\subseteq F''$ kann L nicht gleich L' sein für alle $L' \in \overline{C}$. Andernfalls hätten F'' und C ja einen Konflikt. Ist nun $L = \neg L'$, so gilt $\alpha(L) = 0$. Damit ist die Wahrscheinlichkeit, dass F'' unter der Bedingung, dass $\alpha(\neg C) = 1$ gilt, erfüllt wird, kleiner als die Wahrscheinlichkeit, dass F'' ohne diese Bedingung erfüllt wird. Ist $L \notin vbl(C)$,

ändert sich an den Wahrscheinlichkeiten natürlich nichts. Zusammengefasst gilt also:

$$P(F''|\neg C) \leq P(F'').$$

Damit können wir den obigen Beweis analog fortsetzen, wobei die Induktionsvoraussetzung natürlich nur $|\Gamma'(C)|$ mal angewandt wird. ■

Das *Lopsided Local Lemma* kann zum Beispiel verwendet werden, um die Anzahl der in einer k -KNF-Formel vorkommenden atomaren Formeln so zu begrenzen, dass die Erfüllbarkeit der Formel garantiert ist. In [1] wurde beispielsweise für $k \geq 6$ gezeigt, dass k -KNF-Formeln, in denen jede atomare Formel genau $k + 1$ mal vorkommt, erfüllbar sind.

3.3 Binärbäume und KNF-Formeln

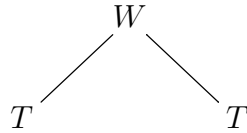
In diesem Abschnitt soll noch ein weiterer Blick auf KNF-Formeln aus einer anderen Sicht geworfen werden und veranschaulichen, aus wie vielen verschiedenen Richtungen man sich der Erfüllbarkeit einer Formel nähern kann.

Satz 3.5 *Seien $k, d \in \mathbb{N}$ und T ein (k, d) -Baum. Dann gibt es eine k -KNF-Formel $F := F(T)$ mit:*

1. *Ist b_T die Anzahl der Blätter von T , so ist die Anzahl der Klauseln in F gleich $|F| = 2b_T$.*
2. *Jedes Literal aus F kommt in maximal d Klauseln in F vor.*
3. *Ist $vbl(C) \cap vbl(D) \neq \emptyset$ für Klauseln $C, D \in F$, $C \neq D$, so existiert genau ein Literal L mit $L \in C$ und $\neg L \in D$.*
4. *F ist unerfüllbar.*
5. *Gibt es für jeden Knoten v aus T mindestens ein Blatt u mit $dist(u, v) \leq k-1$, so ist $|F| = |vbl(F)| + 1$.*
6. *F ist minimal unerfüllbar, d.h. für jede Klausel $C \in F$ ist $F' := F \setminus \{C\}$ erfüllbar.*

Beweis: Um eine Formel F mit den gewünschten Eigenschaften zu konstruieren, kopieren wir zunächst den Binärbaum T und setzen ihn und seine

Kopie anschließend, verbunden durch eine neue Wurzel W , zu einem neuen Baum $\hat{T} = (\hat{V}, \hat{E})$ zusammen:



\hat{T} ist ein (k,d) -Baum, wobei sich die Tiefe jedes Blattes nun um eins erhöht hat, die 2. Bedingung für einen (k,d) -Baum überträgt sich auf \hat{T} .

zu 1.: Ist b_T die Anzahl der Blätter von T , so hat \hat{T} offensichtlich $2b_T$ Blätter und wegen Satz 2.3 ($k_T = b_T - 1$) $2b_T - 1$ innere Knoten, also insgesamt $4b_T - 1$ Ecken. Vergessen wir die Wurzel W , so bleiben $4b_T - 2$ Ecken, die wir in $2b_T - 1$ Geschwisterpaare aufteilen.

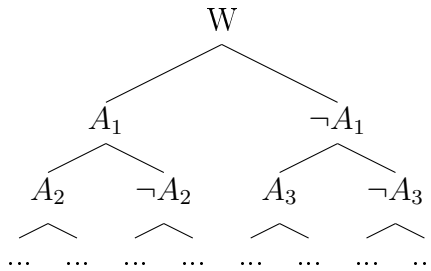
Sei nun $V \cup \bar{V} = \{A_1, \dots, A_{2b_T-1}\} \cup \{\neg A_1, \dots, \neg A_{2b_T-1}\}$ eine Menge von $2b_T - 1$ atomaren Formeln vereinigt mit der Menge ihrer Negationen. Wegen $|(\hat{V} \setminus \{W\})| = 4b_T - 2 = |V \cup \bar{V}|$ kann man alle Ecken von \hat{T} bis auf die Wurzel durch eine bijektive Abbildung

$$\phi : (\hat{V} \setminus \{W\}) \rightarrow V \cup \bar{V}$$

so umbenennen, dass für Geschwister u und v gilt:

$$\text{aus } \phi(u) = L \text{ folgt } \phi(v) = \neg L.$$

Damit identifizieren wir jede Ecke von $\hat{V} \setminus \{W\}$ eindeutig mit einem Literal aus $V \cup \bar{V}$ und umgekehrt. Zum Beispiel:



Zu jedem Blatt v wird nun eine Klausel C_v konstruiert: Beginnend bei v geht man einen $k-1$ langen Weg entlang der Vorfahren von v in Richtung der Wurzel. Da jedes Blatt mindestens Tiefe k hat, erreicht der Weg nie die Wurzel. C_v bestehe nun aus v und dessen $k-1$ nächsten Vorfahren. Da jedes Literal nur einmal vorkommt hat C_v genau k Elemente. Mit

$$F = F(T) := \bigcup_{v \in B_{\hat{T}}} C_v,$$

wobei $B_{\hat{T}}$ die Menge aller Blätter in \hat{T} sei, ist F eine Klauselmenge. Sie entspricht einer k -KNF-Formel mit $2b_T$ Disjunktionstermen. Also: $|F| = 2b_T$.

zu 2.: Wegen der 2. Eigenschaft eines (k,d) -Baums gilt: Zu jedem inneren Knoten v gibt es maximal d Blätter u mit $\text{dist}(u,v) \leq k-1$. Da jeder innere Knoten einem Literal $L \in V \cup \bar{V}$ entspricht und aus jedem Blatt eine Klausel konstruiert wurde, gibt es auch maximal d Klauseln in F in denen L vorkommt.

zu 3.: Sei $vbl(C_u) \cap vbl(C_v) \neq \emptyset$ für zwei Klauseln $C_u, C_v \in F, u \neq v$. Dann gibt es ein Atom A mit $A \in vbl(C_u)$ und $A \in vbl(C_v)$. Betrachten wir den nächsten gemeinsamen Vorfahren w von u und v , so muss ein Kind von w in C_u und eins in C_v liegen (wegen $u \neq v$, kann w kein Blatt sein). Diese Kinder haben nach Konstruktion komplementäre Vorzeichen, es gibt also ein Literal L mit $L \in C_u$ und $\neg L \in C_v$. Angenommen, es gäbe ein weiteres Literal L' mit $L' \in C_u$ und $\neg L' \in C_v$. Dann muss, aufgrund der Konstruktion von C_u , A ein Vorfahre von L' oder umgekehrt sein. Ohne Einschränkung sei A ein Vorfahre von L' . Da L' und $\neg L'$ Geschwister sind, muss also A auch ein Vorfahre von $\neg L'$ sein, dann kann aber $\neg A$ kein Vorfahre oder Nachkomme von $\neg L'$ sein und somit nicht in C_v enthalten sein. Daraus folgt die Eindeutigkeit.

zu 4.: Sei $\alpha : V \rightarrow \{0, 1\}$ eine beliebige Belegung. Beginnend bei der Wurzel definiert α einen Weg über Literale L zu einem Blatt v , indem man immer entlang $\alpha(L) = 0$ geht. Offensichtlich gilt dann $\alpha(C_v) = 0$. Es gibt also immer eine Klausel C_v , die nicht erfüllt wird. Damit ist F unerfüllbar.

zu 5.: Es gebe für jeden Knoten v von T mindestens ein Blatt u mit $\text{dist}(u,v) \leq k-1$. Damit kommt jedes Literal aus $V \cup \bar{V}$ zumindest in einer Klausel C_u vor. Mit 1. folgt:

$$|vbl(F)| + 1 = |V| + 1 = 2b_T - 1 + 1 = 2b_T = |F|.$$

zu 6.: Zu jedem Blatt v von \hat{T} sei $F_v := F \setminus \{C_v\}$. Zu zeigen ist nun die Erfüllbarkeit von F_v .

Im Folgenden wird eine Belegung $\alpha : vbl(F) \rightarrow \{0, 1\}$ konstruiert und eine Abbildung

$$f : \{L \in V \cup \bar{V} \mid \alpha(L) = 1\} \rightarrow F_v$$

definiert. Ist f bijektiv, so gilt auch $\alpha(C) = 1$ für alle $C \in F_v$ und insgesamt $\alpha(F_v) = 1$.

Initialisiere zunächst $u := W$, $v_u := v$ und setze $\alpha(L) = 0$ für alle Literale L auf dem Weg von v_u nach u (einschließlich v_u , ausschließlich u).

Definiere nun die Menge

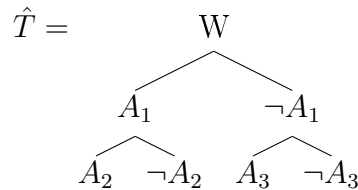
$$G_{v_u, u} := \{L' \in V \cup \bar{V} \mid L' \text{ ist Geschwister eines der Literale auf dem Weg von } v_u \text{ nach } u\}$$

der Geschwister dieser Literale und initialisiere $S := G_{v, W}$. Die Menge S wird im Laufe des Algorithmus häufig verändert und ihre Mächtigkeit dient unter anderem als Abbruchkriterium.

Verfahre nun wie folgt:

- (i) ist $S = \emptyset$: STOP!
sonst: wähle ein Element aus S , überschreibe u damit und überschreibe $S := S \setminus \{u\}$
- (ii) setze $\alpha(u) = 1$
- (iii) ist u ein Blatt, so setze $v_u := u$ und gehe zu (vi), sonst wähle ein Blatt v_u mit $\text{dist}(v_u, u) \leq k - 1$
- (iv) setze $\alpha(L) = 0$ für alle Literale L , die sich auf dem Weg von v_u nach u befinden
- (v) überschreibe $S := S \cup G_{v_u, u}$
- (vi) setze schließlich $f(u) := C_{v_u}$ und gehe zu (i)

Beispiel:



\hat{T} ist wie im Beispiel aus 2.3 ein (3,4)-Baum.

Weiter entspricht F der unerfüllbaren Formel

$$(A_1 \vee A_2) \wedge (A_1 \vee \neg A_2) \wedge (\neg A_1 \vee A_3) \wedge (\neg A_1 \vee \neg A_3).$$

Sei nun $v = A_2$, also $C_v = \{A_1, A_2\}$ und $F_v := F \setminus \{C_v\}$.

Weiter sei $\alpha : \{A_1, A_2, A_3\} \rightarrow \{0, 1\}$ eine Abbildung mit $\alpha(A_1) = \alpha(A_2) = 0$ und $S := \{\neg A_1, \neg A_2\} \neq \emptyset$.

- (i) $S \neq \emptyset$, wähle etwa $\neg A_2 \in S$, überschreibe $u := \neg A_2$ und $S := S \setminus \{u\} = \{\neg A_1\}$
- (ii) setze $\alpha(u) = \alpha(\neg A_2) = 1$
- (iii) u ist ein Blatt, setze also $v_u := u$
- (vi) setze $f(u) = f(\neg A_2) := C_{\neg A_2}$

- (i) $S \neq \emptyset$, wähle also $u := \neg A_1$ und überschreibe $S := \emptyset$
- (ii) setze $\alpha(u) = \alpha(\neg A_1) = 1$
- (iii) u ist kein Blatt, wähle v_u so, dass $\text{dist}(v_u, u) \leq 2 = k - 1$ ist, etwa $v_u := A_3$
- (iv) setze $\alpha(L) = 0$ für alle Literale L , die sich auf dem Weg von v_u nach u befinden, das ist nur v_u selbst, also: $\alpha(A_3) = 0$
- (v) überschreibe $S := S \cup G_{v_u, u} = \{\neg A_3\} \neq \emptyset$
- (vi) setze $f(\neg A_1) := C_{A_3}$

- (i) $S \neq \emptyset$, wähle also $u := \neg A_3$ und überschreibe $S := \emptyset$
- (ii) setze $\alpha(\neg A_3) = 1$
- (iii) u ist ein Blatt, setze also $v_u := u$
- (vi) setze $f(\neg A_3) := C_{\neg A_3}$

- (i) $S = \emptyset$: STOP!

In der Tat ist α eine Belegung, die F_v erfüllt und f eine Bijektion zwischen der Menge der Ecken, die mit 1 belegt werden und der Menge der Klauseln aus F_v .

Dies gilt es jetzt für allgemeine Binärbäume T zu beweisen. Wir zeigen:

- (a) $\alpha : \text{vbl}(F) \rightarrow \{0, 1\}$ ist eine Belegung,
- (b) $|\{L \in V \cup \bar{V} \mid \alpha(L) = 1\}| = 2b_T - 1 = |F_v|$ und schließlich
- (c) $f : \{L \in V \cup \bar{V} \mid \alpha(L) = 1\} \rightarrow F_v$ ist bijektiv.

Wie oben schon bemerkt, folgt dann mit der Bijektivität von f , dass die Belegung α jede Klausel aus F_v , insgesamt also auch F_v , erfüllt.

zu (a) Ein Literal $L \in V \cup \bar{V}$ wird genau dann auf 1 abgebildet, wenn es im Laufe des Algorithmus' in S auftritt. L wird genau dann auf 0 abgebildet, wenn $\neg L$ im Laufe des Algorithmus' in S auftritt. Angenommen, L wird im Laufe des Algorithmus' weder auf 1 noch auf 0 abgebildet, dann wird wegen Schritt (v) auch $\neg L$ weder auf 1 noch auf 0 abgebildet. Sei w der nächste gemeinsame Vorfahre der Geschwister L und $\neg L$. Dann gibt es drei Fälle:

Fall 1: $\alpha(w) = 1$

Dann ist w im Laufe des Algorithmus' in S und L oder $\neg L$ liegt auf dem Weg eines Blattes v_w zu w , wird somit auf 0 abgebildet.

Fall 2: $\alpha(w) = 0$

Dann liegt w auf dem Weg eines Blattes v_u nach $u \in S$ und da w selbst kein Blatt ist, muss auch L oder $\neg L$ auf diesem Weg liegen und somit auf 0 abgebildet werden.

Fall 3: w bleibt unbelegt

Ist $w=W$, kann dies nach der Anfangssituation des Algorithmus' nicht sein. Ist $w \neq W$, so muss wieder wegen Schritt (v) auch $\neg w$ unbelegt bleiben und wir befinden uns in der Anfangssituation dieses Beweises.

Da \hat{T} ein Binärbaum ist, muss w irgendwann die Wurzel sein und der Widerspruch ist gefunden. Zusammengefasst wird also jedes Literal genau einmal abgebildet und komplementäre Literale, also Geschwister, haben unterschiedliche Bilder in $\{0, 1\}$. Damit ist α eine Belegung.

zu (b) Wegen (a) wird genau die Hälfte aller Literale aus $V \cup \bar{V}$ auf 1 und die andere Hälfte auf 0 abgebildet. Jedes Literal entspricht genau einer Ecke aus $\hat{V} \setminus \{W\}$, das sind (vgl. „zu 1.“) $4b_T - 2$ Ecken. Also: $|\{L \in V \cup \bar{V} \mid \alpha(L) = 1\}| = \frac{1}{2}(4b_T - 2) = 2b_T - 1 = |F_v|$.

zu (c) Seien $u, v \in V \cup \bar{V}$ mit $\alpha(u) = \alpha(v) = 1$ und $f(u) = f(v) = C_r$ für ein gewisses Blatt r .

Dann sind $u, v \in S$, $\text{dist}(r, u) \leq k-1$ und $\text{dist}(r, v) \leq k-1$. Damit sind u und v Vorfahren von r und deshalb muss auch u Vorfahre von v oder v Vorfahre von u sein. Sei ohne Einschränkung u Vorfahre von v . Dann muss aber wegen (iv) $\alpha(v) = 0$ oder $u=v$ gelten.

Ersteres wäre ein Widerspruch, also ist f injektiv.

Wegen (b) sind die Mengen $\{L \in V \cup \bar{V} \mid \alpha(L) = 1\}$ und F_v gleichmächtig und nach dem sogenannten Schubladenprinzip für injektive Funktionen auf endlichen, gleichmächtigen Mengen folgt schließlich die Bijektivität von f .

■

Sei T ein (k,d) -Baum und $F := F(T)$ wie oben. Definiert man nun für in F vorkommende Literale c :

$$occ(c) := |\{C \in F \mid c \in C\}|$$

als die Anzahl der Klauseln aus F , die c enthalten, so gilt für die Anzahl der Konfliktnachbarschaften einer Klausel C :

$$|\Gamma'(C)| = \sum_{c \in C} |\{D \in F \mid \neg c \in D\}| = \sum_{c \in C} occ(\neg c),$$

denn: Zunächst gilt immer

$$\Gamma(C) = \{D \in F \mid \bar{D} \cap C \neq \emptyset\} = \bigcup_{c \in C} \{D \in F \mid \neg c \in D\}.$$

Seien nun $c, d \in C$ und angenommen, es gäbe ein

$$K \in \{D \in F \mid \neg c \in D\} \cap \{D \in F \mid \neg d \in D\},$$

dann ist $\neg c \in K$ und $\neg d \in K$ und da C einem Disjunktionsterm entspricht, muss $K \neq C$ gelten. Da $vbl(C) \cap vbl(K) \neq \emptyset$ sind die Voraussetzungen von 3. erfüllt und es gibt genau ein Literal L mit $L \in C$ und $\neg L \in K$. Damit ist $c=d$ und die Mengen $\{D \in F \mid \neg c \in D\}$ sind für alle $c \in C$ disjunkt. Daraus folgt dann die obige Gleichheit.

Ist weiter $D \in \Gamma(C)$ ein Nachbar von C und $D \neq C$, so gilt wieder die Voraussetzung aus 3.:

$$vbl(C) \cap vbl(D) \neq \emptyset$$

und es existiert genau ein Literal L mit $L \in C$ und $\neg L \in D$.

Folglich ist D sogar ein Konfliktnachbar von C und es gilt, da C und D beliebig gewählt waren:

$$\Gamma(C) = \Gamma'(C) \text{ für alle } C \in F.$$

Da nun jede Klausel k Elemente hat und weil jedes Literal in maximal d Klauseln vorkommt (2.), können wir für alle $C \in F$ schließen:

$$|\Gamma(C)| = |\Gamma'(C)| = \sum_{c \in C} \text{occ}(\neg c) \leq kd.$$

Bezeichnet $l(k)$ die größte natürliche Zahl, sodass jede k -KNF-Formel G mit $|\Gamma(C)| \leq l(k)$ für alle $C \in G$ erfüllbar ist, so wissen wir mit den vorigen Überlegungen, dass

$$l(k) \leq kd - 1$$

gilt. Schließlich gibt es eine unerfüllbare k -KNF-Formel, nämlich F , mit $|\Gamma(C)| \leq kd$.

4 Schlussbetrachtung

Diese Arbeit liefert nur einen Einblick in das Gebiet der Erfüllbarkeit aussagenlogischer Formeln. Nichtsdestotrotz haben wir wichtige Erkenntnisse über notwendige Bedingungen der Unerfüllbarkeit erlangt und somit auch hinreichende Voraussetzungen der Erfüllbarkeit gefunden. Diese haben sich mal global und mal lokal auf die Formel bezogen und waren teils kombinatorischer, teils analytischer Natur. Mit der Resolventenmethode haben wir einen kurzen Blick auf das Entscheidungsproblem SAT geworfen und ein Verfahren vorgestellt, mit dem man die Erfüllbarkeit einer Formel der Aussagenlogik in endlicher Zeit sicher feststellen kann.

Außerdem wurden Querverbindungen zu anderen Themengebieten hergestellt, um beispielsweise das *Lovász Local Lemma* zu beweisen, mit dem noch weitere erfüllbarkeitsgarantierende Bedingungen an Formeln gestellt werden konnten.

Schließlich konnten minimal unerfüllbare Formeln mithilfe von Binärbäumen konstruiert werden, die den Unterschied zwischen Erfüllbarkeit und Unerfüllbarkeit durch das Weglassen nur einer Klausel verdeutlicht haben.

Alles in allem konnten wir uns dem Problem der Erfüllbarkeit beliebig langer Formeln von vielen unterschiedlichen Seiten aus nähern und somit einen Einblick in die Komplexität dieses Themengebiets gewinnen.

Literatur

- [1] Berman, Piotr ; Karpinski, Marek ; Scott, Alex D.: *Approximation hardness and satisfiability of bounded occurrence instances of SAT*. In: Electronic Colloquium on Computational Complexity (ECCC), 10(022), 2003
- [2] Gebauer, Heidi ; Moser, Robin ; Scheder, Dominik ; Welzl, Emo: *The Lovász Local Lemma and Satisfiability*. In: Albers, Susanne (Hrsg.) ; Alt, Helmut (Hrsg.) ; Näher, Stefan (Hrsg.): *Efficient Algorithms* Bd. 5760. Springer Berlin / Heidelberg, 2009, S. 30–54
- [3] Kütting, Herbert ; Sauer, Martin J.: *Elementare Stochastik*. Spektrum. Akademischer Verlag, 2008
- [4] Schönig, Uwe: *Logik für Informatiker*. Spektrum, Akademischer Verlag, 2000
- [5] Tittmann, Peter: *Graphentheorie*. Carl Hanser Verlag, 2011

Eidesstattliche Versicherung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, ganz oder in Teilen noch nicht als Prüfungsleistung vorgelegt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die benutzten Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellenangaben kenntlich gemacht. Mir ist bewusst, dass es sich bei Plagiarismus um akademisches Fehlverhalten handelt, das sanktioniert werden kann.

Marburg, den