Proseminar WS 18/19 – SAT-Solving

Prof. Dr. Uwe Schöning,
Prof. Dr. Jacobo Torán,
Bogdan Dina,
Florian Wörz

Seminarüberblick

In diesem Proseminar wollen wir das SAT-Problem sowohl aus der theoretischen als auch praktischen (algorithmischen) Perspektive betrachten: SAT-*Solver* sind möglichst effiziente Algorithmen, die eine aussagenlogische Formel übergeben bekommen und untersuchen sollen, ob es eine erfüllende Belegung gibt. SAT ist jedoch NP-vollständig. Ob es also einen Algorithmus gibt, der SAT in Polynomialzeit löst, ist nicht bekannt (falls $\mathsf{P} \neq \mathsf{NP}$ ist, gibt es keinen solchen Algorithmus). Wir wollen in diesem Proseminar aber Möglichkeiten beleuchten, Algorithmen mit tolerabler Laufzeit (ohne $\mathsf{P} \neq \mathsf{NP}$ anzuzweifeln) für dieses Problem zu entwickeln. Grundlage für das Seminar werden ausgewählte Kapitel und Aspekte des Buches [ST13] sein.

Empfohlene Voraussetzungen für das Proseminar:

- Grundkenntnisse der Informatik z. B. aus *Formale Grundlagen* oder *Algorithmen und Datenstrukturen*: z. B. Landausche $\mathcal{O}$-Notation, Boolesche Formeln, etc.
- Grundlegende Mathematik, z. B. aus Analysis 1 für Informatiker und Ingenieure: Binomialkoeffizienten, Exponentialfunktionen, Logarithmen, etc.
- Die Bereitschaft sich fehlende Konzepte aus der Mathematik und Informatik selbstständig anzueignen. Teilweise brauchen wir z. B. etwas Grundlagenwissen über Wahrscheinlichkeitstheorie (Begriffe Wahrscheinlichkeit eines Ereignisses, Gegenereignis, Erwartungswert, Markov-Ungleichung, Jensen-Ungleichung, etc.).

Überblick über mögliche Seminarthemen

Weitere interessante Themen auf diesem Gebiet dürfen gerne vorab per E-Mail an florian.woerz@uni-ulm.de vorgeschlagen werden.

Mögliche Reservethemen: Algorithmus von Zhang.

Die Literaturlisten sind lediglich Vorschläge und Ausschnitte mögliche Literatur. Es ist Aufgabe der Teilnehmer eine umfassende Literaturrecherche zu betreiben.

1 Das SAT-Problem

Bemerkung: Wir wollen uns in diesen und folgenden Vorträgen möglichst an die Notationen aus [ST13] halten.

Im Vortrag sollen zunächst die Grundlagen für die weiteren Vorträge gelegt werden:

- Kurze Einführung von Booleschen Formeln, CNF, Belegungen, Literalen, Bitflips; Definition von SAT, CNF-SAT, k-SAT, etc. (möglichst mit Beamer, um hierauf nicht unnötig viel Zeit zu verlieren).
- Der Satz von Cook-Levin: Die NP-Vollständigkeit von SAT (je nach Zeit ggfs. mit kurzer Beweisskizze).
- Die Exponentialzeit-Hypothese von Impagliazzo & Paturi und Zusammenhang zum $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$-Problem.
- Optional: Praxisanwendungen von SAT.
- Autarke Belegungen.
- Craig-Interpolanten.

Für die Ausarbeitung ist zudem denkbar, dass das Prinzip der Tseitin-Transformation vorgestellt wird und das CSP-Problem und dessen Zusammenhang zu CNF-SAT studiert wird.

Mögliche Literatur

[BHMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren und Toby Walsh, Hrsg. *Handbook of Satisfiability.* Bd. 185. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, Feb. 2009.

[Coo71] Stephen A. Cook. „The Complexity of Theorem-Proving Procedures“. In: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC ’71). Shaker Heights, Ohio, USA, May 3–5, 1971.* Hrsg. von Michael A. Harrison, Ranan B. Banerji und Jeffrey D. Ullman. New York: ACM, 1971, S. 151–158.

[Lev73] Leonid Anatolevich Levin. „Universal Sequential Search Problems (Übersetzung von Russisch ins Englische)“. In: *Problemy Peredachi Informatsii (Problems of Information Transmission)* 9.3 (1973), S. 115–116.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses.* Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

[Tse83] Grigori S. Tseitin. „On the Complexity of Derivation in Propositional Calculus“. In: *Automation of Reasoning 2: Classical Papers on Computational Logic 1967–1970*. Hrsg. von Jörg Siekmann und Graham Wrightson. Symbolic Computation. Berlin et al.: Springer, 1983, S. 466–483.

2 Erfüllbarkeit durch Kombinatorik und das Lovász Local Lemma

Das Hauptaugenmerk dieses Vortrags soll auf der Erfüllbarkeit durch Kombinatorik liegen: In manchen Fällen kann man die Erfüllbarkeit einer aussagenlogischen Formeln nachweisen, in dem man lediglich Klauseln und Variablen zählt, deren Vorkommen in den Klauseln, oder die Größe der Klauseln betrachtet. Neben einfacheren Zusammenhängen wie in [ST13, § 1.7] dargestellt, soll als konkretes größeres Beispiel hierzu das Lovász Local Lemma in der Version von [Mos09] vorgestellt und bewiesen werden. Dabei soll vor allem das „Incompressibility"-Beweisargument und das KJonzept der Kolmogorov-Komplexität beleuchtet werden.

Mögliche Literatur

[BHMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren und Toby Walsh, Hrsg. *Handbook of Satisfiability.* Bd. 185. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, Feb. 2009.

[GMSW09] Heidi Gebauer, Robin A. Moser, Dominik Scheder und Emo Welzl. „The Lovász Local Lemma and Satisfiability". In: *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday.* Hrsg. von Susanne Albers, Helmut Alt und Stefan Näher. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 30–54.

[Mos09] Robin A. Moser. „A constructive proof of the Lovász local lemma". In: *STOC.* ACM, 2009, S. 343–350.

[MT10] Robin A. Moser und Gábor Tardos. „A constructive proof of the general lovász local lemma". In: *J. ACM* 57.2 (2010), 11:1–11:15.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses.* Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

3 Polynomial-lösbare Spezialfälle von SAT

Bei einigen Klassen von Klauselmengen lässt sich das Erfüllbarkeitsproblem in Polynomialzeit lösen. Bevor man also aufwändige SAT-Algorithmen startet, sollte man diese Spezialfälle prüfen.

Angelehnt an [ST13, § 3] sollen folgende Punkte beleuchtet werden:

- Backdoor-Variablen und Ausnutzung dieser.
- 2-CNF $\in$ P: Zwei verschiedene Beweise:
  - Ein graphentheoretischer Ansatz nach [APT79].
  - Ein Argument mittels Resolution.
- Im Speziellen: Papadimitrious Algorithmus für 2-SAT (inkl. Vorstellung der Komplexitätsklasse RP).
- Horn-Formeln und Renamable Horn Formeln.

Mögliche Literatur

[APT79] Bengt Aspvall, Michael F. Plass und Robert Endre Tarjan. „A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas“. In: *Inf. Process. Lett.* 8.3 (1979), S. 121–123.

[HW74] Lawrence J. Henschen und Larry Wos. „Unit Refutations and Horn Sets“. In: *J. ACM* 21.4 (1974), S. 590–605.

[Lew78] Harry R. Lewis. „Renaming a Set of Clauses as a Horn Set“. In: *J. ACM* 25.1 (1978), S. 134–135.

[Pap91] Christos H. Papadimitriou. „On Selecting a Satisfying Truth Assignment (Extended Abstract)“. In: *FOCS*. IEEE Computer Society, 1991, S. 163–169.

4 DPLL-artige Algorithmen I: Monien-Speckenmeyer

Die meisten SAT-Solver bauen auf einer sehr simplen Backtracking Prozedur auf. In [DP60] und [DLL62] wurde eine einfache Heuristik studiert, um die nächste „Verzweigungs-Variable“ festzulegen. Nach mehr 50 Jahren bildet dieser Algorithmus immer noch die Basis der erfolgreichsten vollständigen SAT-Solver.

Es sollte auf folgende Punkte eingegangen werden:

- Einführung von DPLL und evtl. Erwähnung von 1 oder 2 Heuristiken.
- Der Monien-Speckenmeyer-Algorithmus als Spezialfall eines DPLL-Algorithmus mit entsprechender Heuristik.
- Komplexitätsanalyse des Monien-Speckenmeyer-Algorithmus und Einführung der $\mathcal{O}^*$-Notation.

Mögliche Literatur

[BHMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren und Toby Walsh, Hrsg. *Handbook of Satisfiability.* Bd. 185. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, Feb. 2009.

[DLL62] Martin Davis, George Logemann und Donald W. Loveland. „A Machine Program for Theorem-Proving“. In: *Communications of the ACM* 5.7 (Juli 1962), S. 394–397.

[DP60] Martin Davis und Hilary Putnam. „A Computing Procedure for Quantification Theory“. In: *J. ACM* 7.3 (1960), S. 201–215.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks und Francis Zane. „An improved exponential-time algorithm for k-SAT“. In: *J. ACM* 52.3 (2005), S. 337–364.

[PPZ99] Ramamohan Paturi, Pavel Pudlák und Francis Zane. „Satisfiability Coding Lemma“. In: *Chicago J. Theor. Comput. Sci.* 1999 (1999).

[Sch01] Uwe Schöning. *Algorithmik (Spektrum Lehrbuch).* Heidelberg: Spektrum Akademischer Verlag, 2001.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses.* Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

5 `DPLL`-artige Algorithmen II: Paturi-Pudlák-Zane und `CDCL`

Der Paturi-Pudlák-Zane Algorithmus kann als probabilistische Version des `DPLL`-Algorithmus aufgefasst werden: sowohl die „Verzweigungs-Variable“ als auch ihre Belegung werden zufällig festgelegt. Statt Backtracking zu benutzen, wiederholt man den Algorithmus „genügend oft“.

Ziel des Vortrags ist die Vorstellung dieses Algorithmus, seiner Erfolgswahrscheinlichkeit und seiner Komplexität (bei einer konstante Fehlerwahrscheinlichkeit), d. h. folgenden Resultates samt Beweis:

> Der Patuti-Pudlák-Zane-Algorithmus findet für Formeln in k-SAT mit n Variablen mit Wahrscheinlichkeit $p \geq 2^{-n(1-1/k)}$ eine erfüllende Belegung. Die Komplexität des Verfahrens bei konstanter Fehlerwahrscheinlichkeit ist daher $\mathcal{O}^*(2^{n(1-1/k)})$.

Weiterhin soll vorgestellt werden, wie `DPLL` in der Praxis verwendet wird. Insbesondere soll auf `CDCL`-Solver und Implikationsgraphen eingegangen werden.

Mögliche Literatur

[BHMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren und Toby Walsh, Hrsg. *Handbook of Satisfiability.* Bd. 185. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, Feb. 2009.

[DLL62] Martin Davis, George Logemann und Donald W. Loveland. „A Machine Program for Theorem-Proving“. In: *Communications of the ACM* 5.7 (Juli 1962), S. 394–397.

[DP60] Martin Davis und Hilary Putnam. „A Computing Procedure for Quantification Theory“. In: *J. ACM* 7.3 (1960), S. 201–215.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks und Francis Zane. „An improved exponential-time algorithm for k-SAT“. In: *J. ACM* 52.3 (2005), S. 337–364.

[PPZ99] Ramamohan Paturi, Pavel Pudlák und Francis Zane. „Satisfiability Coding Lemma“. In: *Chicago J. Theor. Comput. Sci.* 1999 (1999).

[Sch01] Uwe Schöning. *Algorithmik (Spektrum Lehrbuch).* Heidelberg: Spektrum Akademischer Verlag, 2001.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses.* Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

6 Local Search I: Restarts und Covering Codes

In diesem Vortrag soll das sehr leichte Prinzip *Local Search* als Möglichkeit vorgestellt werden, das NP-schwere Problem SAT zu „lösen“: Man generiert zufällig einen möglichen Lösungkandidaten (eine Belegung) für eine Boolesche Formel, wobei sehr wahrscheinlich einige Klauseln der Formel noch nicht erfüllt sind. Diese Klauseln enthalten Variablen, die natürliche Kandidaten für die Abänderung des Lösungskandidaten durch „Bitflips“ darstellen. Die Hoffnung ist, dass man durch diese lokalen Änderungen des Lösungskandidaten „näher“ an eine erfüllende Belegung gelangt. Da das SAT-Problem NP-vollständig ist, sind keine Wunder zu erwarten. Dennoch möchte man die exponentielle Laufzeit so gering wie möglich halten. Durch zufällige Neuwahl von Anfangsbelegungen gelang es in [Sch98] für 3-SAT eine Laufzeit von

$$\mathcal{O}^*\left(\left(\frac{3}{2}\right)^n\right) = \mathcal{O}\left(\mathrm{poly}(n)\cdot\left(\frac{3}{2}\right)^n\right)$$

zu zeigen.

Folgende Punkte sind interessant:

- Vorstellung Grundidee Local Search
- Einführung von `Deterministic Limited Local Search` als nützliche Subroutine
- Probabilistischer Algorithmus für SAT via `Deterministic Limited Local Search` zusammen mit Restarts samt Komplexitätsanalyse für 3-SAT
- Covering Codes, um deterministische Startpunkte zu wählen

Hinweis: Dieser Vortrag soll [ST13, § 5.1–5.3] abdecken.

Mögliche Literatur

[DGH+02] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch, Ravi Kannan, Jon M. Kleinberg, Christos H. Papadimitriou, Prabhakar Raghavan und Uwe Schöning. „A deterministic (2-2/(k+1))$^{\mathrm{n}}$ algorithm for k-SAT based on local search“. In: *Theor. Comput. Sci.* 289.1 (2002), S. 69–83.

[DGHS00] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch und Uwe Schöning. „Deterministic Algorithms for k-SAT Based on Covering Codes and Local Search“. In: *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings.* 2000, S. 236–247.

[MS11] Robin A. Moser und Dominik Scheder. „A full derandomization of Schöning’s k-SAT algorithm“. In: *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011.* 2011, S. 245–252.

[Sch02] Uwe Schöning. „A Probabilistic Algorithm for k-SAT Based on Limited Local Search and Restart“. In: *Algorithmica* 32.4 (2002), S. 615–623.

[Sch98] Uwe Schöning. *On the complexity of constraint satisfaction problem.* Techn. Ber. Universität Ulm, 1998.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses.* Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

7 Local Search II: WalkSAT und der Moser-Scheder-Algorithmus

Dieser Vortrag baut auf Local Search I auf. Inhalt dieses zweiten Vortrags zu Local Search sollte sein:

- Schönings probabilistischer WalkSAT-Algorithmus
- Der Moser-Scheder-Algorithmus

Hinweis: Dieser Vortrag soll [ST13, § 5.1–5.3] abdecken. Dies ist der technischste Vortrag des Proseminars! Wir erlauben hier auch ggfs. mehr Zeit im Vortrag.

Mögliche Literatur: Wie bei Local Search I

8 Weitere Algorithmen: Divide-and-Conquer für 3-SAT und Stålmarcks Algorithmus

Neben den DPLL-artigen Algorithmen und den Algorithmen, die auf lokaler Suche suche aufbauen, gibt es noch weitere algorithmische Zugänge zum Erfüllbarkeitsproblem SAT.

In diesem Vortrag soll zunächst ein Zugang beleuchtet werden, der speziell auf 3-SAT zugeschnitten ist und von [Pud98] stammt. Der Vortrag soll diesen Ansatz verständlich darstellen (hierzu sind sicher Beispiele sinnvoll) und anschließend eine kleine Komplexitätsanalyse vornehmen.

Der Algorithmus von Stålmarck testet, ob eine gegebene Formel eine Tautologie ist. Von dem Algorithmus existieren verschiedene Varianten. Speziell für Formeln, die aus industriellen Anwendungen entstehen, liefert der Algorithmus gute Resultate. Im zweiten Teil dieses Vortrags soll die Version aus [SS98] vorgestellt werden. Es bietet sich an, ebenfalls den Resolutionskalkül kurz anzusprechen, da terminale Tripletts im Algorithmus die gleiche Rolle spielen wie die leere Klausel im Resolutionskalkül.

Mögliche Literatur

[Pud98] Pavel Pudlák. „Satisfiability - Algorithms and Logic“. In: *MFCS*. Bd. 1450. Lecture Notes in Computer Science. Springer, 1998, S. 129–141.

[SS98] Mary Sheeran und Gunnar Stålmarck. „A Tutorial on Stålmarcks's Proof Procedure for Propositional Logic“. In: *FMCAD*. Bd. 1522. Lecture Notes in Computer Science. Springer, 1998, S. 82–99.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses*. Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

9 Resolution und das Pigeon Hole Principle

Das Beweissystem der Resolution kann dafür verwendet werden, von einer Formel zu zeigen, dass sie unerfüllbar ist. Resolution ist manchen z. B. schon aus der Vorlesung Logik bekannt. Wir wollen hier einige theoretische Resultate studieren. Denkbare Aspekte dieses Vortrages:

- Der Resolutionskalkül (kurze Einführung ohne Beweise)
- Kurz: Zusammenhang zum $\mathsf{NP} \stackrel{?}{=} \mathsf{co\text{-}NP}$-Problem
- Kurz: Der Davis-Putnam-Algorithmus
- Das Hauptresultat des Vortrags sollte sein: Exponentielle untere Schranken an Resolutionswiderlegungen durch das Pigeon Hole Principle. Wir erwarten hier einen gut verständlich vorgetragenen Beweis des Resultates.

Denkbar für die Ausarbeitung ist aus das Vorstellen von Strategien & Einschränkungen.

Mögliche Literatur

[Bon18] Ilario Bonacina. *Space in Weak Propositional Proof Systems.* 1. Aufl. Cham, Schweiz: Springer International Publishing, Jan. 2018.

[BP96] Paul Beame und Toniann Pitassi. „Simplified and Improved Resolution Lower Bounds“. In: *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’96). Burlington, Vermont, USA, 14–16 October, 1996.* Los Alamitos, Kalifornien / Washington / Tōkyō: IEEE Computer Society Press, Okt. 1996, S. 274–282.

[DP60] Martin Davis und Hilary Putnam. „A Computing Procedure for Quantification Theory“. In: *J. ACM* 7.3 (1960), S. 201–215.

[Hak85] Armin Haken. „The Intractability of Resolution“. In: *Theoretical Computer Science* 39.2-3 (Aug. 1985), S. 297–308.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Hrsg. von William J. Cook, Ronald Graham, Bernhard Korte, László Lovász, Avi Wigderson und Günter M. Ziegler. Bd. 27. Algorithms and Combinatorics. Heidelberg et al.: Springer, Jan. 2012.

[KL94] Hans Kleine Büning und Theodor Lettmann. *Aussagenlogik: Deduktion und Algorithmen.* Hrsg. von Hans-Jürgen Appelrath, Volker Claus, Günther Hotz und Klaus Waldschmidt. Leitfäden und Monographien der Informatik. Stuttgart: B. G. Teubner Verlag, Jan. 1994.

[PBI93] Toniann Pitassi, Paul Beame und Russell Impagliazzo. „Exponential Lower Bounds for the Pigeonhole Principle“. In: *Computational Complexity* 3.2 (Juni 1993). Vorausgehende Version erschienen in STOC ’92, S. 97–140.

[Rob65] John Alan Robinson. „A Machine-Oriented Logic Based on the Resolution Principle“. In: *Journal of the ACM (JACM)* 12.1 (Jan. 1965), S. 23–41.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses*. Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

Vollständiges Literaturverzeichnis

[APT79] Bengt Aspvall, Michael F. Plass und Robert Endre Tarjan. „A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas“. In: *Inf. Process. Lett.* 8.3 (1979), S. 121–123.

[BFF+12] Leo Brueggeman, Michael Fellows, Rudolf Fleischer, Martin Lackner, Christian Komusiewicz, Yiannis Koutis, Andreas Pfandler und Frances Rosamond. „Train Marshalling Is Fixed Parameter Tractable“. In: *Fun with Algorithms*. Hrsg. von Evangelos Kranakis, Danny Krizanc und Flaminia Luccio. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, S. 51–56.

[BHMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren und Toby Walsh, Hrsg. *Handbook of Satisfiability*. Bd. 185. Frontiers in Artificial Intelligence and Applications. Amsterdam: IOS Press, Feb. 2009.

[Bon18] Ilario Bonacina. *Space in Weak Propositional Proof Systems*. 1. Aufl. Cham, Schweiz: Springer International Publishing, Jan. 2018.

[BP96] Paul Beame und Toniann Pitassi. „Simplified and Improved Resolution Lower Bounds“. In: *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’96). Burlington, Vermont, USA, 14–16 October, 1996*. Los Alamitos, Kalifornien / Washington / Tōkyō: IEEE Computer Society Press, Okt. 1996, S. 274–282.

[Bru15] Henning Bruhn-Fujimoto. *Combinatorial Optimisation II*. Vorlesungsmanuskript SoSe 2015. Institut für Optimierung und Operations Research, Universität Ulm, 2015.

[Chr76] Nicos Christofides. *Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem*. Carnegie Mellon University, Feb. 1976.

[Chv79] V. Chvatal. „A Greedy Heuristic for the Set-Covering Problem“. In: *Mathematics of Operations Research* 4.3 (1979), S. 233–235.

[CLRS01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest und Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press und McGraw-Hill Book Company, 2001.

[Coo71] Stephen A. Cook. „The Complexity of Theorem-Proving Procedures“. In: *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC ’71). Shaker Heights, Ohio, USA, May 3–5, 1971*. Hrsg. von Michael A. Harrison, Ranan B. Banerji und Jeffrey D. Ullman. New York: ACM, 1971, S. 151–158.

[DGH+02] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch, Ravi Kannan, Jon M. Kleinberg, Christos H. Papadimitriou, Prabhakar Raghavan und Uwe Schöning. „A deterministic $(2\text{-}2/(k+1))^n$ algorithm for *k*-SAT based on local search“. In: *Theor. Comput. Sci.* 289.1 (2002), S. 69–83.

[DGHS00] Evgeny Dantsin, Andreas Goerdt, Edward A. Hirsch und Uwe Schöning. „Deterministic Algorithms for *k*-SAT Based on Covering Codes and Local Search“. In: *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings.* 2000, S. 236–247.

[DHMR00] Elias Dahlhaus, Peter Horák, Mirka Miller und Joseph F. Ryan. „The train marshalling problem“. In: *Discrete Applied Mathematics* 103.1-3 (2000), S. 41–54.

[DLL62] Martin Davis, George Logemann und Donald W. Loveland. „A Machine Program for Theorem-Proving“. In: *Communications of the ACM* 5.7 (Juli 1962), S. 394–397.

[DP60] Martin Davis und Hilary Putnam. „A Computing Procedure for Quantification Theory“. In: *J. ACM* 7.3 (1960), S. 201–215.

[GMSW09] Heidi Gebauer, Robin A. Moser, Dominik Scheder und Emo Welzl. „The Lovász Local Lemma and Satisfiability“. In: *Efficient Algorithms: Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday.* Hrsg. von Susanne Albers, Helmut Alt und Stefan Näher. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, S. 30–54.

[Gra72] Ronald L. Graham. „An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set“. In: *Inf. Process. Lett.* 1.4 (1972), S. 132–133.

[GT88] Andrew V. Goldberg und Robert Endre Tarjan. „A new approach to the maximum-flow problem“. In: *J. ACM* 35.4 (1988), S. 921–940.

[Hak85] Armin Haken. „The Intractability of Resolution“. In: *Theoretical Computer Science* 39.2-3 (Aug. 1985), S. 297–308.

[Hus15] Thore Husfeldt. „Topics in Chromatic Graph Theory“. In: Hrsg. von Lowell W. Beineke und Robin J. Wilson. Cambridge, UK: Cambridge University Press, 2015. Kap. Graph Coloring Algorithms, S. 277–303.

[HW74] Lawrence J. Henschen und Larry Wos. „Unit Refutations and Horn Sets“. In: *J. ACM* 21.4 (1974), S. 590–605.

[Jar73] Ray A. Jarvis. „On the Identification of the Convex Hull of a Finite Set of Points in the Plane“. In: *Inf. Process. Lett.* 2.1 (1973), S. 18–21.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Hrsg. von William J. Cook, Ronald Graham, Bernhard Korte, László Lovász, Avi Wigderson und Günter M. Ziegler. Bd. 27. Algorithms and Combinatorics. Heidelberg et al.: Springer, Jan. 2012.

[Jun07] D. Jungnickel. *Graphs, Networks and Algorithms.* Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2007. ISBN: 9783540727804.

[KL94] Hans Kleine Büning und Theodor Lettmann. *Aussagenlogik: Deduktion und Algorithmen.* Hrsg. von Hans-Jürgen Appelrath, Volker Claus, Günther Hotz und Klaus Waldschmidt. Leitfäden und Monographien der Informatik. Stuttgart: B. G. Teubner Verlag, Jan. 1994.

[KV18] B. Korte und J. Vygen. *Combinatorial Optimization: Theory and Algorithms.* Algorithms and Combinatorics. Springer Berlin Heidelberg, 2018. ISBN: 9783662560396.

[Lev73] Leonid Anatolevich Levin. „Universal Sequential Search Problems (Übersetzung von Russisch ins Englische)“. In: *Problemy Peredachi Informatsii (Problems of Information Transmission)* 9.3 (1973), S. 115–116.

[Lew78] Harry R. Lewis. „Renaming a Set of Clauses as a Horn Set“. In: *J. ACM* 25.1 (1978), S. 134–135.

[Mos09] Robin A. Moser. „A constructive proof of the Lovász local lemma“. In: *STOC.* ACM, 2009, S. 343–350.

[MS11] Robin A. Moser und Dominik Scheder. „A full derandomization of Schöning's k-SAT algorithm“. In: *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011.* 2011, S. 245–252.

[MT10] Robin A. Moser und Gábor Tardos. „A constructive proof of the general lovász local lemma“. In: *J. ACM* 57.2 (2010), 11:1–11:15.

[Pap91] Christos H. Papadimitriou. „On Selecting a Satisfying Truth Assignment (Extended Abstract)“. In: *FOCS.* IEEE Computer Society, 1991, S. 163–169.

[PBI93] Toniann Pitassi, Paul Beame und Russell Impagliazzo. „Exponential Lower Bounds for the Pigeonhole Principle“. In: *Computational Complexity* 3.2 (Juni 1993). Vorausgehende Version erschienen in STOC '92, S. 97–140.

[PPSZ05] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks und Francis Zane. „An improved exponential-time algorithm for k-SAT“. In: *J. ACM* 52.3 (2005), S. 337–364.

[PPZ99] Ramamohan Paturi, Pavel Pudlák und Francis Zane. „Satisfiability Coding Lemma“. In: *Chicago J. Theor. Comput. Sci.* 1999 (1999).

[Pud98] Pavel Pudlák. „Satisfiability - Algorithms and Logic“. In: *MFCS*. Bd. 1450. Lecture Notes in Computer Science. Springer, 1998, S. 129–141.

[Rob65] John Alan Robinson. „A Machine-Oriented Logic Based on the Resolution Principle“. In: *Journal of the ACM (JACM)* 12.1 (Jan. 1965), S. 23–41.

[Sch01] Uwe Schöning. *Algorithmik (Spektrum Lehrbuch)*. Heidelberg: Spektrum Akademischer Verlag, 2001.

[Sch02] Uwe Schöning. „A Probabilistic Algorithm for k-SAT Based on Limited Local Search and Restart“. In: *Algorithmica* 32.4 (2002), S. 615–623.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and Combinatorics Bd. 1. Springer, 2003. ISBN: 9783540443896.

[Sch98] Uwe Schöning. *On the complexity of constraint satisfaction problem*. Techn. Ber. Universität Ulm, 1998.

[SS98] Mary Sheeran und Gunnar Stålmarck. „A Tutorial on Stålmarcks's Proof Procedure for Propositional Logic“. In: *FMCAD*. Bd. 1522. Lecture Notes in Computer Science. Springer, 1998, S. 82–99.

[ST13] Uwe Schöning und Jacobo Torán. *The Satisfiability Problem: Algorithms and Analyses*. Bd. 3. Mathematics for Applications (Mathematik für Anwendungen). Berlin: Lehmanns Media, Juli 2013.

[Tse83] Grigori S. Tseitin. „On the Complexity of Derivation in Propositional Calculus“. In: *Automation of Reasoning 2: Classical Papers on Computational Logic 1967–1970*. Hrsg. von Jörg Siekmann und Graham Wrightson. Symbolic Computation. Berlin et al.: Springer, 1983, S. 466–483.