Seminar-Themen

„Algorithmen und Komplexität“ und „Komplexitätstheorie“

Wintersemester 19/20 — Prof. Dr. Georg Schnitger

Hier finden Sie Themen für das Bachelor-Seminar „Algorithmen und Komplexität“ und das Master-Seminar „Komplexitätstheorie“ im WS 19/20. Bachelor-Studenten dürfen jedes Thema wählen, das mit B markiert ist, für Master-Studenten kommen alle Themen in Frage, die mit M markiert sind. Themen, die sowohl mit B als auch mit M markiert sind, stehen beiden Studiengängen offen, wobei der geforderte Umfang für Master-Studenten entsprechend größer ist. Wenn Sie ein eigenes Thema vorschlagen wollen, setzen Sie sich bitte zeitnah mit uns in Verbindung.

Beachten Sie bei Ihrer Themenwahl unbedingt die empfohlenen Vorkenntnisse der Themen! Falls Ihnen diese Vorkenntnisse fehlen, sollten Sie zusätzliche Zeit zur Aufarbeitung einplanen.

Inhaltsverzeichnis

Algorithmen 2

1 B SAT-Algorithmen: Schöning-Algorithmus . . . 2

2 B, M SAT-Algorithmen: Moser-Scheder-Algorithmus . . . 2

3 B The Small-World Phenomenon: An Algorithmic Perspective . . . 2

4 B, M A Universal Algorithm for Sequential Data Compression . . . 2

5 B, M Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications . . . 2

6 B, M Network Applications of Bloom Filters . . . 3

7 B, M Dynamische Programmierung und Belief Propagation . . . 3

8 B, M Low-Density Parity-Check Codes . . . 3

9 B The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals . 3

Komplexitätstheorie 4

10 B, M Interactive Proofs and Graph Isomorphism . . . 4

11 M Interaktive Beweise für unerfüllbare Formeln . . . 4

12 M Rational Proofs . . . 4

13 B, M Lower Bounds for Monotone Circuits . . . 5

14 M A Personal View of Average-Case Complexity . . . 5

15 B, M Algebraic Computation Models . . . 5

16 B Complexity of Counting and #P-Completeness . . . 5

17 B Smoothed Complexity and Pseudopolynomial-Time Algorithms . . . 5

18 B Not being (super)thin or solid is hard: A study of grid Hamiltonicity . . . 6

19 B Succinct Representations of Graphs . . . 6

20 B, M From Finite Automata to Regular Expressions and Back . . . 6

21 B, M Extended Regular Expressions: Succinctness and Decidability . . . 7

22 B Lower bounds for context-free grammars . . . 7

23 B, M More concise representation of regular languages by automata and reg. expressions . . 7

24 B, M Automata and Quantum Computing . . . 7

25 B, M Lineare Programmierung und die P=NP-Frage . . . 7

26 B Tetris is hard, even to approximate . . . 8

Algorithmen

① **SAT-Algorithmen: Schöning-Algorithmus** B

Uwe Schöning, Jacobo Torán
Das Erfüllbarkeitsproblem SAT – Algorithmen und Analysen. Mathematik für Anwendungen 1, Lehmann 2012. http://dblp.org/rec/books/daglib/0028796

Kurzbeschreibung: Im SAT-Problem ist eine KNF-Formel gegeben und auf Erfüllbarkeit zu prüfen. Da das Problem NP-vollständig ist, sind hier keine Wunder zu erwarten, aber man möchte die exponentielle Laufzeit so niedrig wie möglich drücken. Der randomisierte Schöning-WalkSAT-Algorithmus durchsucht mittels einer Markov-Kette den Belegungsraum und erreicht die Laufzeit $\mathcal{O}\big(\mathrm{poly}(n) \cdot (\frac{4}{3})^n\big)$ für 3-SAT.

Empfohlene Vorkenntnisse: Elementare Stochastik, Markov-Ketten

② **SAT-Algorithmen: Moser-Scheder-Algorithmus** B, M

Uwe Schöning, Jacobo Torán
Das Erfüllbarkeitsproblem SAT – Algorithmen und Analysen. Mathematik für Anwendungen 1, Lehmann 2012. http://dblp.org/rec/books/daglib/0028796

Kurzbeschreibung: Der Moser-Scheder-Algorithmus erreicht dieselbe Laufzeit wie der oben genannte Schöning-Algorithmus sogar deterministisch mithilfe von Überdeckungscodes und lokaler Suche.

Bemerkungen: Das Thema baut auf dem obigen („SAT-Algorithmen: Schöning-Algorithmus“) auf.

③ **The Small-World Phenomenon: An Algorithmic Perspective** B

Jon Kleinberg
STOC, 2000. https://dl.acm.org/citation.cfm?id=335325

Kurzbeschreibung: In sozialen Netzwerken tritt das sogenannte Kleine-Welt-Phänomen auf: Je zwei Personen sind über eine sehr kurze Kette von Kontakten miteinander verbunden. Diese Arbeit beschäftigt sich mit den algorithmischen Grundlagen von Milgrams Experiment „übermittle einen Brief an eine Zielperson, von der du nur den Wohnort kennst, indem du ihn an einen deiner Bekannten weitergibst, von dem du vermutest, dass er der Zielperson näher steht als du“.

Empfohlene Vorkenntnisse: parallele Algorithmen

④ **A Universal Algorithm for Sequential Data Compression** B, M

Jacob Ziv, Abraham Lempel
IEEE Trans. Information Theory 23(3): 337-343 (1977). https://dblp.org/rec/journals/tit/ZivL77

Kurzbeschreibung: Der Lempel-Ziv-Algorithmus (LZ77 bzw. LZ1) steht am Anfang einer Reihe von technischen Weiterentwicklungen bei der Kompression von sequentiellen Datenströmen. Er bildet die Basis für Grafikformate wie GIF oder PNG und den Lempel-Ziv-Markov-Algorithmus (LZMA), der bei verschiedenen Linux-Distributionen genutzt wird.

Bemerkungen: Als Masterthema ist neben LZ77 auch LZMA (siehe https://www.7-zip.org/sdk.html) zu bearbeiten. Insbesondere soll dargestellt werden, auf welchen theoretischen Grundlagen neuere Varianten des Lempel-Ziv-Algorithmus fußen.

Empfohlene Vorkenntnisse: String-Algorithmen, Lesen von technischen Spezifikationen und Quellcode

⑤ **Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications** B, M

Ion Stoica, Robert Tappan Morris, David R. Karger, M. Frans Kaashoek, Hari Balakrishnan
SIGCOMM 2001. https://dblp.org/rec/conf/sigcomm/StoicaMKKB01

Kurzbeschreibung: Chord ist eines der klassischen Protokolle für P2P-Anwendungen auf Basis von verteilten Hashtabellen (DHT, distributed hash tables): Finde zu einem gegebenen Schlüssel den Knoten, der den mit dem Schlüssel assoziierten Wert speichert.

Bemerkungen: Neben der theoretischen Analyse von Chord soll im Seminar ein kritischer Vergleich mit anderen verteilten Datenstrukturen wie etwa Blockchains vorgestellt werden.

Empfohlene Vorkenntnisse: Elementare Stochastik, Hashfunktionen

---

⑥ **Network Applications of Bloom Filters** B, M

Andrei Z. Broder, Michael Mitzenmacher
Internet Mathematics 1(4): 485-509 (2003). https://dblp.org/rec/journals/im/BroderM03

Kurzbeschreibung: Ein Bloom-Filter ist eine randomisierte Datenstruktur für das Wortproblem. Dabei erlaubt man sich mit kleiner Wahrscheinlichkeit eine falsche Antwort auf die Frage „Gehört w zur Sprache L?“ und spart dafür Speicherplatz ein.

Bemerkungen: Neben den theoretischen Grundlagen sollen beispielhaft verschiedene Anwendungen von Bloomfiltern untersucht werden.

Empfohlene Vorkenntnisse: Elementare Stochastik, Randomisierte Algorithmen

---

⑦ **Dynamische Programmierung und Belief Propagation** B, M

Georg Schnitger
Skript Computational Learning Theory, Sommersemester 2018, Abschnitt 15.2.2. http://thi.cs.uni-frankfurt.de/lehre/clt/sose18/skript_clt18.pdf

Kurzbeschreibung: Zahlreiche klassische Probleme der theoretischen Informatik lassen sich als Marginalisierung eines Produktes über einem Semiring auffassen. Belief Propagation ist ein Message-Passing-Algorithmus, der in vielen praktischen Anwendungen die Marginalisierung effizient approximiert, in einigen Fällen sogar exakte Lösungen ermittelt. Für Belief Propagation auf Bäumen entspricht der Algorithmus dem Ansatz der dynamischen Programmierung.

Empfohlene Vorkenntnisse: Dynamische Programmierung, Grundlagen der Algebra

---

⑧ **Low-Density Parity-Check Codes** B, M

David J.C. MacKay
Information Theory, Inference, and Learning Algorithms, Kapitel 47 bis inkl. 47.4. http://www.inference.phy.cam.ac.uk/itprnn/book.pdf

Kurzbeschreibung: LDPC-Codes sind fehlerkorrigierende lineare Blockcodes, die z. B. für WLAN, digitales Fernsehen und für andere Datenübertragungstechniken eingesetzt werden. Die Dekodierung von Codewörtern ist jedoch NP-vollständig. Als Dekodierungsheuristik wird der Belief-Propagation-Algorithmus verwendet. Neben Anwendungen im maschinellen Lernen und in der kombinatorischen Optimierung, ist die Dekodierung von LDPC-Codes damit eines der herausragenden Anwendungsbeispiele von Belief Propagation.

Bemerkungen: Elementare Stochastik, lineare Blockcodes

⑨	**The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals**	B

Jon Feldman, Matthias Ruhl
SIAM J. Comput. 36(2): 543-561 (2006). https://dblp.org/rec/journals/siamcomp/FeldmanR06

Kurzbeschreibung: Im Directed Steiner Network Problem sind ein gerichteter Graph mit Kantengewichten, sowie p Quellen $s_1, \ldots, s_p$ und p Senken $t_1, \ldots, t_p$ gegeben. Gefunden werden soll ein leichtester gerichteter Wald (d. h. eine leichteste Vereinigung von gerichteten Pfaden) sodass es von jeder Quelle s_i einen Weg zur Senke t_i gibt. Während das Problem für allgemeine p als NP-vollständig nachgewiesen wurde, lässt es sich für beschränkte p in Polynomialzeit lösen.

Komplexitätstheorie

⑩	**Interactive Proofs and Graph Isomorphism**	B, M

Kapitel 8 bis inkl. Abschnitt 8.3 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Gegeben seien eine Sprache L und eine Eingabe x. In einem Interactive Proof (IP) müssen zwei Personen kooperieren, um die Frage $x \in L$*?* zu beantworten. Der *Prover* betrachtet die Eingabe, rechnet beliebig lange und schickt einen „Beweis“ an den *Verifier*. Dieser muss dann mithilfe des Beweises in Polynomialzeit entscheiden, ob $x \in L$ gilt. Ein prominentes Problem im Kontext der IPs ist Graph Isomorphism (GI). Es wird einerseits gezeigt, dass GI vermutlich nicht NP-vollständig ist – sonst würde die Polynomialzeithierarchie kollabieren –, andererseits ist auch kein Polynomialzeitalgorithmus für GI bekannt. Es wird somit vermutet, das GI „zwischen“ P und NP-vollständig liegt.

Empfohlene Vorkenntnisse: Komplexitätsklassen wie P, NP und PSPACE, Elementare Stochastik

⑪	**Interaktive Beweise für unerfüllbare Formeln**	M

Scott Aaronson
Quantum Computing since Democritus (2013), Kapitel 17. https://www.scottaaronson.com/democritus/lec16.html

Kurzbeschreibung: *Wir haben eine aussagenlogische Formel ϕ und fragen uns, ob ϕ unerfüllbar ist. Eines Tages besuchen uns superintelligente Aliens auf der Erde. Wir misstrauen den Aliens und ihrer Technologie, aber bitten sie, uns zu* beweisen*, dass ϕ unerfüllbar ist. Gelingt es den Aliens, uns zu überzeugen?* Dieses Szenario wird durch das Modell der *Interactive Proofs* (IP) formalisiert.

Bemerkungen: Ausgehend von der *informellen* Darstellung in dem obigen Buch soll das Thema auch tiefergehend inkl. formalen Beweisen der Resultate bearbeitet werden. Siehe dazu z.B. Kapitel 8 (insb. 8.5) in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Empfohlene Vorkenntnisse: interaktive Beweise, Komplexitätsklassen

⑫	**Rational Proofs**	M

Pablo Daniel Azar, Silvio Micali
STOC 2012: 1017-1028. https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Proof%20Systems/RationalProofs.pdf

Kurzbeschreibung: *König Arthur möchte eine gegebene Funktion f auf einer Eingabe x auswerten und fragt seinen allwissenden Berater Merlin um Rat. Dieser kennt natürlich die Antwort, verrät sie aber nur gegen eine Bezahlung, deren Höhe von seiner Anwort abhängt. Arthur muss nun eine Bezahlungsfunktion geschickt wählen, damit Merlin – der natürlich möglichst viel verdienen möchte – die korrekte Antwort liefert.* In dieser Arbeit wird ergründet, welche Probleme f sich durch solche sog. *rationalen interaktiven Beweise* lösen lassen.

Empfohlene Vorkenntnisse: interaktive Beweise, Komplexitätsklassen wie PP und #P, Stochastik

(13) Lower Bounds for Monotone Circuits — B, M

Abschnitt 14.3 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Ein monotoner Schaltkreis besitzt OR- und AND-Gatter, aber keine NOT-Gatter. Eine Technik zur Herleitung unterer Schranken für die Größe monotoner Schaltkreise für Entscheidungsprobleme wird vorgestellt. Beispielsweise benötigt das Clique-Problem monotone Schaltkreise mindestens exponentieller Größe. Könnte man die Einschränkung der Monotonie weglassen, so wäre $\mathsf{P} \neq \mathsf{NP}$ gezeigt, doch leider benötigen monotone Schaltkreise auch exponentielle Größe für die Lösung des Matching-Problems.

Empfohlene Vorkenntnisse: Schaltkreise, Elementare Stochastik

(14) A Personal View of Average-Case Complexity — M

Russell Impagliazzo
Structure in Complexity Theory Conference 1995: 134-147. http://dblp.org/rec/conf/coco/Impagliazzo95

Kurzbeschreibung: Wie sähe unser Leben aus, wenn Hypothesen aus der Komplexitätstheorie wie bspw. „$\mathsf{P} = \mathsf{NP}$“ gälte oder Einwegfunktionen existierten? Auf umgangssprachliche Weise werden fünf mögliche Welten beschrieben: Algorithmica, Heuristica, Pessiland, Minicrypt und Cryptomania.

Bemerkungen: Diese Arbeit ist als Ausgangspunkt für eine Formalisierung der beschriebenen Welten (Definitionen, Sätze, Beweisideen) anhand der dort zitierten Literatur aufzufassen.
Es können bis zu zwei Vorträge zu dieser Arbeit vergeben werden. Dabei sollte der eine im Themengebiet Average-Case-Complexity (die ersten drei Welten) und der andere im Themengebiet One-Way-Functions (die letzten drei Welten) angesiedelt werden.

Empfohlene Vorkenntnisse: Grundkenntnisse in Kryptographie

(15) Algebraic Computation Models — B, M

Kapitel 16 exkl. Abschnitt 16.2 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Hier werden Berechnungsmodelle untersucht, die nicht auf Bits, sondern auf komplexen Zahlen rechnen. Die Klassen $\mathsf{P}_{\mathbb{C}}$ und $\mathsf{NP}_{\mathbb{C}}$ werden analog zu P und NP eingeführt. Das Entscheidungsproblem des Hilbertschen Nullstellensatzes („Gegeben m Polynome $p_1, \ldots, p_m$ vom Grad d, besitzen diese eine gemeinsame Nullstelle?“) wird als $\mathsf{NP}_{\mathbb{C}}$-vollständig nachgewiesen – selbst wenn die Eingabe aus konventionellen Bits besteht.

Empfohlene Vorkenntnisse: Schaltkreise, Komplexitätsklassen

(16) Complexity of Counting and #P-Completeness — B

Kapitel 17 bis inkl. Abschnitt 17.3.1 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/

Kurzbeschreibung: Im Gegensatz zu Entscheidungsproblemen, wo nach der *Existenz* einer Lösung gefragt ist, wird in Zählproblemen nach der *Anzahl* der Lösungen gefragt. Viele solcher Zählprobleme, beispielsweise das Zählen perfekter Matchings oder einfacher Kreise in einem Graphen, haben sich als sehr schwierig herausgestellt. Diese Schwierigkeit wird durch das Konzept der #P-Härte analog zur NP-Härte formalisiert.

Empfohlene Vorkenntnisse: Komplexitätsklassen, Polynomialzeitreduktionen

(17) Smoothed Complexity and Pseudopolynomial-Time Algorithms — B

Tim Roughgarden
Skripte (Lecture 17 und 18) zur Vorlesung CS264 Fall'17 (Beyond Worst-Case Analysis). Stanford University. http://theory.stanford.edu/~tim/w17/w17.html

Kurzbeschreibung: Smoothed Complexity ist ein Hybrid zwischen Worst-Case Complexity und Average-Case Complexity und beantwortet die Frage „Wie hoch ist die erwartete Laufzeit eines Algorithmus, wenn Worst-Case-Eingaben zufällig verrauscht werden?“. Im Rahmen der Vorlesung wird ein zentrales Resultat von Beier und Vöcking präsentiert: Ein binäres Optimierungsproblem hat genau dann polynomielle Smoothed Complexity, wenn es durch einen Algorithmus mit pseudopolynomieller erwarteter Laufzeit gelöst werden kann.

Bemerkungen: Aus Lecture 17 sind nur die Abschnitte 1 und 2 zu bearbeiten.

Empfohlene Vorkenntnisse: Laufzeitanalyse, elementare Stochastik, lineare Programmierung

(18) Not being (super)thin or solid is hard: A study of grid Hamiltonicity — B

Esther M. Arkin, Sándor P. Fekete, Kamrul Islam, Henk Meijer, Joseph S.B. Mitchell, Valentin Polishchuk, David Rappaport, Henry Xiao
Computational Geometry: Theory and Applications, 42(6-7), 582-605. https://www.sciencedirect.com/science/article/pii/S092577210800117X

Kurzbeschreibung: Das Hamiltonkreis-Problem ist eines der klassischen NP-vollständigen Probleme. Sofern $\mathsf{P} \neq \mathsf{NP}$ gilt, ist es im Allgemeinen nicht effizient lösbar. Doch wie sehen möglichst „einfache“ Graphklassen aus, für die das Hamiltonkreis-Problem trotzdem noch NP-vollständig ist? Und welche Graphklassen sind einfach und erlauben eine effiziente Lösung des Hamiltonkreis-Problems? In dieser Arbeit betrachten wir Gitter mit drei-, vier- und sechseckigen Maschen.

Empfohlene Vorkenntnisse: Polynomialzeitreduktionen

(19) Succinct Representations of Graphs — B

Bemerkung: Beide Quellen sind zu bearbeiten.

Hana Galperin, Avi Wigderson: *Succinct representations of graphs*, Information and Control, 56(3):183–198, 1983. https://dblp.org/rec/journals/iandc/GalperinW83

Kurzbeschreibung: Üblicherweise wird die Laufzeitkomplexität von Graphproblemen in Abhängigkeit von $|V|$ angegeben. Doch was passiert, wenn die Eingabeinstanz besonders platzsparend (mit polylog($|V|$) vielen Bits) beschrieben werden kann? Es stellt sich heraus, dass viele einfache Graphprobleme plötzlich NP-vollständig sind, wenn die Eingaben platzsparend beschrieben werden.

Christos H. Papadimitriou, Mihalis Yannakakis: *A Note on Succinct Representations of Graphs*, Information and Control 71(3): 181-185 (1986). https://dblp.org/rec/journals/iandc/PapadimitriouY86

Kurzbeschreibung: In dieser Arbeit wird gezeigt, dass klassische NP-vollständige Graphprobleme NEXP-vollständig sind, wenn ihre Eingaben platzsparend beschrieben werden.

Empfohlene Vorkenntnisse: Komplexitätsklassen, Reduktionen

(20) From Finite Automata to Regular Expressions and Back — B, M

Hermann Gruber, Markus Holzer
Int. J. Found. Comput. Sci. 26(8): 1009-1040 (2015). https://dblp.org/rec/journals/ijfcs/GruberH15

Kurzbeschreibung: Sowohl DFAs als auch reguläre Ausdrücke beschreiben die Klasse der regulären Sprachen. Jedoch gibt es Sprachen, die sich durch kleine DFAs, aber nur durch große reguläre Ausdrücke beschreiben lassen – und umgekehrt. Es wird untersucht, wie schlimm dieser „Blow-Up“ werden kann. Und wie zeigt man eigentlich untere Schranken für die Länge regulärer Ausdrücke?

Bemerkungen: Ausgehend von der oben genannten *Übersichtsarbeit* sollen auch die in den Originalarbeiten gegebenen Beweise bearbeitet werden. Dabei dürfen und sollen eigene Schwerpunkte gesetzt werden, nach Absprache mit dem Betreuer.

Empfohlene Vorkenntnisse: gute Kenntnisse in regulären Sprachen

㉑	**Extended Regular Expressions: Succinctness and Decidability**	B, M

Dominik D. Freydenberger
28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011). http://drops.dagstuhl.de/opus/volltexte/2011/3039/

Kurzbeschreibung: In der Praxis verbreitete Implementierungen von regulären Ausdrücken erlauben die Verwendung von Variablen bzw. Rückreferenzen. Dieses zusätzliche Feature der sogenannten erweiterten regulären Ausdrücke hat jedoch einen Haken: Der Ausdruck $\alpha = \big((\mathtt{a}|\mathtt{b})^*\big)\%x \cdot x$ beschreibt die (nichtreguläre) Copy-Sprache $L(\alpha) = \{ww : w \in \{a,b\}^*\}$. In dieser Arbeit werden die folgenden Fragestellungen für erweiterte reguläre Ausdrücke untersucht: Minimierbarkeit, Beschreibungslänge im Vergleich zu klassischen regulären Ausdrücken, Unentscheidbarkeit des Universalitäts-, Äquivalenz-, Inklusions-, Regularitäts- und Koendlichkeitsproblems.

Empfohlene Vorkenntnisse: Formale Sprachen, Berechenbarkeit, Entscheidbarkeit, Turing-Maschinen

㉒	**Lower bounds for context-free grammars**	B

Yuval Filmus
Inf. Process. Lett. 111(18): 895-898 (2011). http://www.cs.toronto.edu/~yuvalf/CFG-LB.pdf

Kurzbeschreibung: Wie viele Produktionen benötigt eine kontextfreie Grammatik (CFG), um eine gegebene Sprache L zu erzeugen? In dieser Arbeit wird eine Methode vorgestellt, mithilfe derer sich untere Schranken an die Größe von CFGs für endliche Sprachen zeigen lassen.

Empfohlene Vorkenntnisse: Kontextfreie Grammatiken

㉓	**More concise representation of regular languages by automata and reg. expressions**	B, M

Viliam Geffert, Carlo Mereghetti, Beatrice Palano
Inf. Comput. 208(4): 385-394 (2010). https://dblp.org/rec/html/journals/iandc/GeffertMP10

Kurzbeschreibung: Ein *constant height push down automat* ist ein Kellerautomat, dessen Kellergröße durch eine Konstante beschränkt ist. Solche Automaten akzeptieren stets *reguläre* Sprachen, können aber evtl. gegenüber DFAs oder sogar NFAs mit exponentiell weniger Zuständen auskommen. Das Modell der *straight line programs for regular expressions* hingegen stellt eine Verallgemeinerung regulärer Ausdrücke dar: hier können Teilausdrücke R' mehrfach „recycelt“ und so Ersparnisse in der Beschreibungslänge erzielt werden. In dieser Arbeit wird die Beschreibungskomplexität beider Modelle untersucht und ein überraschender Zusammenhang gezeigt.

Empfohlene Vorkenntnisse: DFAs, PDAs, Schaltkreise

㉔	**Automata and Quantum Computing**	B, M

Andris Ambainis, Abuzer Yakaryilmaz
CoRR abs/1507.01988 (2015.) https://arxiv.org/abs/1507.01988v2 bis inkl. Theorem 4.7

Kurzbeschreibung: Analog zu Quantencomputern lassen sich *Quanten endliche Automaten (QFAs)* definieren. Kann man jeden QFA durch DFAs oder probabilistische endliche Automaten (PFAs) simulieren? Und wenn ja, wie viele Zustände sind dafür jeweils notwendig und hinreichend?

Empfohlene Vorkenntnisse: Grundlagen des Quantencomputing oder der Quantenmechanik, Automatenmodelle (DFAs, PFAs)

㉕ Lineare Programmierung und die P=NP-Frage — B, M

Bemerkung: Beide Quellen sind zu bearbeiten (erste Quelle: Abschnitt 7.1; zweite Quelle: bis inkl. der Proposition auf Seite 446).

GEORG SCHNITGER: Skript zur Vorlesung Approximationsalgorithmen, Wintersemester 2013/14 http://thi.cs.uni-frankfurt.de/lehre/apa/ws1314/apa_ws1314_skript.pdf

Kurzbeschreibung: Lineare Programmierung ist eines der wichtigsten Werkzeuge im Bereich der kombinatorischen Optimierung. Ganzzahlige lineare Programmierung ist allerdings NP-vollständig. Für das Matching-Problem in bipartiten Graphen ist die Ganzzahligkeitsbedingung implizit erfüllt.

MIHALIS YANNAKAKIS: *Expressing Combinatorial Optimization Problems by Linear Programs*, J. Comput. Syst. Sci. 43(3): 441-466 (1991). https://dblp.org/rec/journals/jcss/Yannakakis91

Kurzbeschreibung: Wenn es gelingt, ein „kleines“ lineares Programm für ein NP-schweres Problem zu beschreiben, sodass die Ganzzahligkeitsbedingung implizit erfüllt ist, hat man P=NP gezeigt. Yannakakis zeigt, dass selbst für das in polynomieller Zeit lösbare Matching-Problem in nicht-bipartiten Graphen lineare Programme exponentieller Größe erforderlich sind.

Empfohlene Vorkenntnisse: Lineare Programmierung

㉖ Tetris is hard, even to approximate — B

RON BREUKELAAR, ERIK D. DEMAINE, SUSAN HOHENBERGER, HENDRIK JAN HOOGEBOOM, WALTER A. KOSTERS, DAVID LIBEN-NOWELL
Int. J. Comput. Geometry Appl. 14(1-2): 41-68 (2004) https://dblp.org/rec/journals/ijcga/BreukelaarDHHKL04

Kurzbeschreibung: Der Videospiel-Klassiker Tetris wird hinsichtlich seiner Komplexität untersucht. Zentrale Fragestellungen wie „maximiere die Anzahl der vervollständigten Reihen“ stellen sich als NP-hart heraus.

Empfohlene Vorkenntnisse: Polynomialzeitreduktionen