Seminar-Themen

„Algorithmen und Komplexität“

Sommersemester 20 — Prof. Dr. Georg Schnitger

Hier finden Sie Themen für das Bachelor-Seminar „Algorithmen und Komplexität“ im WS 19/20. Wählen Sie aus den unten genannten Themen Ihre drei Wunschthemen aus. 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

① SAT-Algorithmen: Schöning-Algorithmus

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 $\mathcal{O}\big(\text{poly}(n) \cdot (\frac{4}{3})^n\big)$ für 3-SAT.

Empfohlene Vorkenntnisse: Elementare Stochastik, Markov-Ketten

② SAT-Algorithmen: Moser-Scheder-Algorithmus

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

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 Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications

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 Peer, der den mit dem Schlüssel assoziierten Wert speichert.

Bemerkungen: Elementare Stochastik, Hashfunktionen

⑤ Network Applications of Bloom Filters

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

⑥ **Dynamische Programmierung und Belief Propagation**

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

⑦ **Low-Density Parity-Check Codes**

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.

Bemerkungen: Elementare Stochastik, lineare Blockcodes

⑧ **Diskrete Fourier-Transformation und schnelle Multiplikation von Polynomen**

Cormen, Leiserson und Rivest
Kapitel 30 von Introduction to algorithms, 3. Auflage, The MIT Press, 2009. https://hds.hebis.de/ubffm/Record/HEB390398470

Kurzbeschreibung: Diskrete Fourier-Transformation (DFT) ist eine fundamentale Technik in algorithmischen Anwendungen. Mithilfe der schnellen Fourier-Transformation (FFT) gelingt z. B. die Multiplikation von Polynomen in Laufzeit $\Theta(n \log n)$ statt $\Theta(n^2)$. Neben den mathematischen Grundlagen der DFT sollen prominente Beispiele für Anwendungen der FFT vorgestellt werden.

Empfohlene Vorkenntnisse: mathematische Grundlagen

Komplexitätstheorie

⑨ **Lower Bounds for Monotone Circuits**

Abschnitt 13.3 und 13.4 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/ (Draft, 2007)

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

(10) Complexity of Counting and #P-Completeness

Kapitel 9 bis inkl. Abschnitt 9.3.1 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/ (Draft, 2007)

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

(11) Nicht-uniforme Schaltkreise und die Klasse P/poly

Kapitel 6 bis inkl. Abschnitt 6.1 und Abschnitt 7.6 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/ (Draft, 2007)

Kurzbeschreibung: P/poly ist die Klasse aller Sprachen, die polynomiell großen Schaltkreisfamilien akzeptiert werden können. Hier wird also ein uniformer Berechnungsbegriff (ein Algorithmus für **alle** Eingabelängen) durch einen nicht-uniformen Berechnungsberechnung (für jede Eingabelänge ein eigener Algorithmus) ersetzt. Es wird zuerst gezeigt, dass $\mathsf{P} \subseteq \mathsf{P/poly}$ gilt. Außerdem ist es möglich, und das ist der wesentliche Thema des Vortrags, effiziente randomisierte Algorithmen zu *derandomisieren*, wenn man auf Uniformität verzichtet, d. h. $\mathsf{BPP} \subseteq \mathsf{P/poly}$ gilt.

Empfohlene Vorkenntnisse: Komplexitätsklassen, Polynomialzeitreduktionen

(12) Das Verhältnis zwischen P und NP

Abschnitte 3.4 bis 3.5 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/ (Draft, 2007)

Kurzbeschreibung: Im Satz von Ladner wird gezeigt, dass es Probleme gibt, die zwar in $\mathsf{NP} \setminus \mathsf{P}$ liegen, aber nicht NP-vollständig sind, sofern $P \neq \mathsf{NP}$ gilt. Unter der Annahme $\mathsf{P} \neq \mathsf{NP}$ gibt es also „mittelschwere“ Probleme. Wir betrachten außerdem Berechnungswelten, in denen Turingmaschinen ein Orakel A befragen können, das ihnen in Polynomialzeit eine Antwort liefert. Analog zu P und NP sind die Komplexitätsklassen P^A und NP^A definiert. Dann gibt es Orakel A und B, sodass $\mathsf{P}^A = \mathsf{NP}^A$ und $\mathsf{P}^B \neq \mathsf{NP}^B$ gilt.

Bemerkungen: Es können bis zu zwei Vorträge zu diesem Thema gehalten werden. Für jeden Vortrag werden die Inhalte in Absprache mit dem Betreuer eingegrenzt.

Empfohlene Vorkenntnisse: Komplexitätsklassen, Orakel, Diagonalisierung

(13) BPP is in PH

Abschnitt 7.7 in *Computational Complexity: A Modern Approach*, Cambridge University Press 2009 von Sanjeev Arora und Baoz Barak. http://theory.cs.princeton.edu/complexity/ (Draft, 2007)

Kurzbeschreibung: Was haben effiziente randomisierte Algorithmen mit Raten und Verifizieren zu tun? Eine ganze Menge! Wir zeigen, dass $\mathsf{BPP} \subseteq \Sigma_2^p \cap \Pi_2^p \subseteq \mathsf{PH}$ gilt, d. h. BPP liegt in der polynomiellen Hierarchie.

Empfohlene Vorkenntnisse: Komplexitätsklassen, Orakel, Diagonalisierung

(14) Smoothed Complexity and Pseudopolynomial-Time Algorithms

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?“.
In Lecture 17 wird gezeigt: „The smoothed complexity of the 2-OPT heuristic for the TSP in the plane with the ℓ_1 metric is polynomial.“
In Lecture 18 wird gezeigt, dass ein binäres Optimierungsproblem genau dann polynomielle Smoothed Complexity hat, wenn es durch einen Algorithmus mit pseudopolynomieller erwarteter Laufzeit gelöst werden kann.

Bemerkungen: Es können bis zu zwei Vorträge zu diesem Thema gehalten werden. Für jeden Vortrag werden die Inhalte in Absprache mit dem Betreuer eingegrenzt.

Empfohlene Vorkenntnisse: Laufzeitanalyse, elementare Stochastik, TSP, lineare Programmierung

(15) Succinct Representations of Graphs

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 der Knotenzahl $|V|$ angegeben. Doch was passiert, wenn die Eingabeinstanz besonders platzsparend (mit $\text{polylog}(|V|)$ vielen Bits) beschrieben werden kann? Es stellt sich heraus, dass platzsparenden Darstellungen viele einfache Graphprobleme plötzlich NP-vollständig sind.

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

(16) Regular Expression Length via Arithmetic Formula Complexity

Ehud Cseresnyes, Hannes Seiwert
noch nicht veröffentlicht (wird bei Interesse zur Verfügung gestellt)

Kurzbeschreibung: Wie „lang“ können reguläre Ausdrücke für endliche Sprachen werden? Mithilfe unterer Schranken für die Größe monotoner arithmetischer Formeln werden untere Schranken für die Länge regulärer Ausdrücke für die sogenannte Binomialsprache aller Wörter mit k Einsen und $n-k$ Nullen hergeleitet. Außerdem wird eine Beweistechnik für untere Schranken monotoner arithmetischer Formeln von Hrubeš und Yehudayoff direkt auf reguläre Ausdrücke übertragen, um eine scharfe untere Schranke für die Größe regulärer Ausdrücke für die Sprache aller Binärzahlen mit n Bits, die durch eine ungerade Zahl p teilbar sind, zu zeigen.
Der Schwerpunkt des Vortrags liegt in der Übersetzung der unteren Schranken für monotone arithmetische Formeln in untere Schranken für reguläre Ausdrücke.

Empfohlene Vorkenntnisse: reguläre Sprachen, Schaltkreise

(17) Lower bounds for context-free grammars

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

⑱ Tetris is hard, even to approximate

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