Seminar »Kryptographie und Komplexität«

Prof. Johannes Köbler Sebastian Kuhnert

Sommersemester 2011

In diesem Seminar werden aktuelle Themen der Theoretischen Informatik, insbesondere der Komplexitätstheorie und der Kryptologie behandelt. Hierbei gehen wir auch gern auf Teilnehmerwünsche ein.

In diesem Semester liegt der Schwerpunkt auf dem Graphisomorphieproblem, also der Frage, ob es für zwei gegebene Graphen G, H eine Bijektion zwischen deren Knotenmengen gibt, die Kanten auf Kanten und Nicht-Kanten auf Nicht-Kanten abbildet. Für dieses Problem ist kein Polynomialzeitalgorithmus bekannt, gleichzeitig gilt es als unwahrscheinlich, dass es NP-hart ist. Wegen der hohen praktischen Relevanz wurden zahlreiche Algorithmen entwickelt, die das Problem für eingeschränkte Graphklassen effizient lösen.

Vorkenntnisse aus der Komplexitätstheorie sind zum Besuch dieses Seminars nützlich, jedoch nicht notwendig.

Themen für Referate

Im Seminar sind Vorträge einführenden und vertiefenden Charakters zu folgenden Themenbereichen geplant:

1. **Isomorphie von Zufallsgraphen.** Für fast alle zufällig gewählten Paare von Eingabegraphen lässt sich das Graphisomorphieproblem in Linearzeit lösen.

   Inhalt: Wie und warum funktioniert der Algorithmus? Welche Eingabepaare sind »schwer«?

   Literatur: [CP08]

2. **Subgraph-Isomorphie** fragt bei Eingabe zweier Graphen G, H, ob es einen Teilgraphen von H gibt, der zu G isomorph ist.

   Inhalt: Warum ist Subgraph-Isomorphie NP-vollständig? Wie nutzt Ullmann die Struktur des Problems aus, um das Durchprobieren aller Möglichkeiten zu beschleunigen?

   Literatur: [Ull76]

3. **Isomorphie von Graphen mit beschränkten Farbklassen.** Jeder Knoten der Eingabegraphen ist gefärbt, jede Farbe kommt nur konstant oft vor, und Isomorphismen müssen jeden Knoten auf einen der gleichen Farbe abbilden.

   Inhalt: Wie funktioniert der randomisierte Algorithmus von Babai? Wie kann durch geschickten Einsatz von Algorithmen für Permutationsgruppen auf Zufall verzichtet werden?

   Literatur: [Bab79; FHL80]

4. **Isomorphie von Bäumen.** Wenn nur Bäume als Eingabegraphen zulässig sind, kann das Isomorphieproblem sogar mit logarithmisch wenig Platz entschieden werden.

   Inhalt: Wie funktioniert Lindells Algorithmus?

   Literatur: [Lin92]

5. **Isomorphie von partiellen k-Bäumen.** Die Klasse der partiellen k-Bäume ist eine Verallgemeinerung von Bäumen und beschreibt baumähnliche Graphen.

   Inhalt: Wie sind partielle k-Bäume definiert? Wie funktioniert Bodlaenders Algorithmus?

   Literatur: [Bod90]

6. **Isomorphie von Graphen mit beschränktem Grad.** Hier hat jeder Knoten nur konstant viele Nachbarn.

   Inhalt: Wie funktioniert Luks' Algorithmus?

   Literatur: [Luk82]

7. **Praktische Isomorphiealgorithmen für planare Graphen.**

   Inhalt: Wie funktioniert der Algorithmus? Wie schnell ist er im Vergleich zu anderen?

   Literatur: [KHC04]

Literatur

[Bab79] László Babai. *Monte-Carlo algorithms in graph isomorphism testing.* Tech. rep. 79–10. Université de Montréal, 1979.
URL: http://people.cs.uchicago.edu/~laci/lasvegas79.pdf
(visited on Mar. 21, 2011).

[Bod90] Hans L. Bodlaender. 'Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees'.
In: *Journal of Algorithms* 11.4 (Dec. 1990), pp. 631–643.
ISSN: 0196-6774. DOI: 10.1016/0196-6774(90)90013-5.

[Böt06] Martin Böttcher. *Einführung in das wissenschaftliche Arbeiten.* Universität Leipzig. 2006. URL: http://bis.informatik.uni-leipzig.de/de/Lehre/0506/SS/SemASKE/files?get=einfuehrung_in_das_wiss_arbeiten.pdf (besucht am 21. März 2011).

[CP08] Tomek Czajka and Gopal Pandurangan. 'Improved random graph isomorphism'. In: *Journal of Discrete Algorithms* 6.1 (Mar. 2008), pp. 85–92. ISSN: 1570-8667. DOI: 10.1016/j.jda.2007.01.002.

[FHL80] Merrik Furst, John Edward Hopcroft, and Eugene M. Luks. *Polynomial-time algorithms for permutation groups.* Tech. rep. Cornell University, Oct. 1980. HDL: 1813/6282.

[KHC04] Jacek P. Kukluk, Lawrence B. Holder, and Diane J. Cook. 'Algorithm and experiments in testing planar graphs for isomorphism'. In: *Journal of Graph Algorithms and Applications* 8.3 (2004), pp. 313–356. URL: http://www.emis.ams.org/journals/JGAA/accepted/2004/KuklukHolderCook2004.8.3.pdf (visited on Mar. 21, 2011).

[Lin92] Steven Lindell. 'A logspace algorithm for tree canonization. extended abstract'. In: *Proceedings of 24th annual ACM symposium on Theory of computing (STOC).* 1992, pp. 400–404. ISBN: 0-89791-511-9. DOI: 10.1145/129712.129750.

[Luk82] Eugene M. Luks. 'Isomorphism of graphs of bounded valence can be tested in polynomial time'. In: *Journal of Computer and System Sciences* 25.1 (Aug. 1982), pp. 42–65. ISSN: 0022-0000. DOI: 10.1016/0022-0000(82)90009-5.

[Mit07] Roland Mittermair. *Hinweise für korrektes Zitieren.* Institut für Informatik-Systeme, Universität Klagenfurt. 2007. URL: http://www.uni-klu.ac.at/tewi/downloads/Zitierhinweise.pdf (besucht am 21. März 2011).

[TWM10] Till Tantau, Joseph Wright, and Vedran Miletić. *The* BEAMER *class.* Version 3.10. 2010. URL: http://mirror.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf (visited on Mar. 21, 2011).

[Ull76] Julian R. Ullmann. 'An algorithm for subgraph isomorphism'. In: *Journal of the ACM* 23.1 (Jan. 1976), pp. 31–42. ISSN: 0004-5411. DOI: 10.1145/321921.321925.

Ablauf

- In der ersten Woche stellen wir euch die Referatsthemen vor und ihr wählt euer Thema aus. Außerdem geben wir euch Hinweise zur Gestaltung von Referaten und Ausarbeitungen.
- Im Lauf des Semesters haltet ihr **Referate**
  - Die Referate haben das Ziel, dass ihr (a) euch ein Thema erarbeitet, (b) euer Thema den anderen vermittelt, (c) von den Referaten der anderen lernt und (d) Vortragspraxis sammelt.
  - Einerseits sollen eure Referate *anschaulich* sein: Ihr führt die anderen in euer Thema ein. Bitte setzt dabei nicht mehr voraus, als sie schon wissen. Mit Beispielen und Bildern könnt ihr euren Zuhörern das Verstehen erleichtern. Eine gute Richtschnur für gute Erklärungen ist die Frage »Was hat mir selbst geholfen, das zu verstehen?«
  - Andererseits sollen eure Referate auch *präzise* sein: Klare Definitionen und die Details von Konstruktionen und Algorithmen gehören auch dazu.
  - Für euer Referat stehen euch ca. 90 Minuten zur Verfügung. Bitte plant Zeit für Rückfragen ein!
  - Nach jedem Referat gibt es eine Feedbackrunde.
- **Vorbereitung** des eigenen Referats:
  - Ihr arbeitet euch in das Thema ein, indem ihr die angegebene (und ggf. weitere) Literatur lest. Literatur, die es nicht in der Bibliothek oder im Netz gibt, kann bei uns kopiert werden.
  - Vor der Vorbereitung des Vortrags lest ihr am besten [TWM10, Abschnitt 5] – das lohnt sich auch dann, wenn ihr nicht LaTeX verwendet.
  - Eine Woche vor dem Referat kommt ihr in unsere Sprechstunde, um letzte Verständnisfragen zu stellen und den Ablauf des Referats durchzusprechen.
- Es ist ein zentrales Element eines Seminars, auch von den Referaten der anderen zu lernen. Deshalb solltet ihr möglichst immer **anwesend sein**. Wenn ihr mehr als einmal fehlt, zeigt uns bitte unaufgefordert ein Attest.
- Nach dem Referat fertigt ihr noch eine schriftliche **Ausarbeitung** zu eurem Thema an.
  - Die Ausarbeitungen haben das Ziel, (a) das im Seminar gesammelte Wissen zusammenzufassen, (b) Interessierten einen Einstieg in euer Thema zu ermöglichen und (c) euch die Gelegenheit zu geben, wissenschaftliches Schreiben zu üben (Vorbereitung auf Studien- und Diplomarbeit).
  - Wir werden eure Ausarbeitungen auf der Webseite des Seminars veröffentlichen, wenn ihr damit einverstanden seid.
  - Der Umfang eurer Ausarbeitung soll dem Umfang eures Referats entsprechen. Erfahrungsgemäß ergibt das 10–20 Seiten.
  - Hinweise zum wissenschaftlichen Schreiben findet ihr unter [Böt06] und [Mit07].