Seminar »Komplexität und Kryptologie«

PCP und Approximation

Prof. Johannes Köbler Sebastian Kuhnert

Wintersemester 2011/2012

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 PCP-Theorem und Approximationsalgorithmen.

PCP steht für *probabilistically checkable proofs*: Das sind Beweise, die man nicht vollständig lesen muss, um ihre Richtigkeit zu überprüfen – stattdessen genügt das Betrachten weniger, zufällig ausgewählter Bits, um gültige und ungültige Beweise mit hoher Wahrscheinlichkeit zu unterscheiden. Das PCP-Theorem besagt, dass für jede Sprache in NP (für die also Beweise existieren, die in polynomieller Zeit geprüft werden können) auch PCPs existieren.

Approximationsalgorithmen kommen zum Einsatz, wenn die exakte Lösung eines Optimierungsproblems nicht effizient möglich ist. Dies betrifft beispielsweise Max-3-Sat, wo für eine in 3-KNF gegebene Formel nach der maximalen Anzahl gleichzeitig erfüllbarer Klauseln gefragt wird. Einerseits sind für viele praktisch relevante Optimierungsprobleme gute Approximationsalgorithmen bekannt, andererseits lassen sich in vielen Fällen mithilfe des PCP-Theorems Grenzen der Approximierbarkeit zeigen.

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. **Approximation von Optimierungsproblemen**
   Diskrete Optimierungsprobleme haben eine hohe praktische Relevanz, exakte Lösungen können aber häufig nicht effizient ermittelt werden. Approximierungsalgorithmen ermöglichen in vielen dieser Fälle effiziente Lösungen.

   Inhalt: Was ist ein ρ-Approximationsalgorithmus?
   Wie kann SetCover approximiert werden?

   Literatur: [AB09, Kapitel 11.1; Vaz01, Kapitel 1 und 2]

2. **PCP und Nichtapproximierbarkeit**
   Das PCP-Theorem kann aus zwei unterschiedlichen Blickwinkeln betrachtet werden: Einmal als Aussage über probabilistisch überprüfbare Beweise, und einmal als Aussage über die Härte von Approximationsproblemen.

   Inhalt: Wie können diese beiden Blickwinkel formalisiert werden?
   Warum sind die beiden Aussagen äquivalent?
   Warum sind VertexCover und SetCover schwer zu approximieren?

   Literatur: [AB09, Kapitel 11.2 bis 11.4 und 22.8]

3. **Das PCP-Theorem (Teil 1)**
   Beweis eines Vorläufers des PCP-Theorems, der auch im Beweis des unabgeschwächten Theorems verwendet wird.

   Inhalt: Warum gilt $\mathsf{NP} \subseteq \mathsf{PCP}(\mathrm{poly}(n), 1)$?
   Wie kann die Größe des Alphabets reduziert werden?

   Literatur: [AB09, Kapitel 11.5 und 22.2.5]

4. **Das PCP-Theorem (Teil 2)**
   Beweis des PCP-Theorems.

   Inhalt: Warum gilt $\mathsf{NP} = \mathsf{PCP}(\log(n), 1)$?

   Literatur: [AB09, Kapitel 22.2]

5. **Håstads 3-Bit-PCP**
   In dieser verschärften Variante des PCP-Theorems werden nur drei Bits des Beweises überprüft. Sie impliziert, dass $(7/8 + \epsilon)$-Approximation von Max-3-Sat NP-hart ist.

   Inhalt: Warum reicht es aus, 3 Bits eines Beweises anzusehen?

   Literatur: [AB09, Kapitel 22.4 bis 22.7]

6. **Approximationsalgorithmen durch Lineare Programme**
   Lineare Programmierung kann als Werkzeug für den Entwurf von Approximationsalgorithmen verwendet werden.

   Inhalt: Warum gilt der Dualitätssatz für Lineare Programme?
   Wie kann Lineare Programmierung genutzt werden, um Approximationalgorithmen für Max-3-Sat und SetCover zu gewinnen?

   Literatur: [Vaz01, Kapitel 12; WS11, Kapitel 1]

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].

Literatur

[AB08] Sanjeev Arora and Boaz Barak. *Computational complexity. A modern approach.* Web draft of [AB09]. Princeton University, 2008. URL: http://www.cs.princeton.edu/theory/index.php/Compbook/Draft (visited on Aug. 31, 2011).

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity. A modern approach.* Cambridge University Press, 2009. ISBN: 978-0-521-42426-4.

[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 31. Aug. 2011).

[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 31. Aug. 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 Aug. 31, 2011).

[Vaz01] Vijay V. Vazirani. *Approximation algorithms.* Berlin et al.: Springer, 2001. ISBN: 3-540-65367-8.

[WS11] David P. Williamson and David B. Shmoys. *The design of approximation algorithms.* Camebridge University Press, 2011. ISBN: 978-0-521-195270. URL: http://designofapproxalgs.com/book.pdf.