

Seminar Komplexität und Kryptologie

Fixed Parameter Tractability

Prof. Johannes Köbler Sebastian Kuhnert

Wintersemester 2013/14

Montag 13:15–14:45, RUD 26, 1'308

In diesem Seminar werden aktuelle Forschungsthemen der Gebiete Komplexitätstheorie und Kryptografie vorgestellt und diskutiert. Hierbei gehen wir auch gern auf Teilnehmerwünsche ein. Vorkenntnisse aus dem Bereich Komplexitätstheorie und Graphalgorithmen sind hilfreich, aber nicht notwendig. Das Seminar eignet sich gut zur Vorbereitung auf Abschlussarbeiten.

In diesem Semester liegt der Schwerpunkt auf *Fixed Parameter Tractability (FPT)*. Hierbei handelt es sich um einen Ansatz, im Allgemeinen nicht effizient lösbare Probleme mit vertretbarem Aufwand zu lösen, sofern ein *Parameter* der Eingabeinstanz hinreichend klein ist.

Themen für Referate

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

1. Grundlagen

Inhalt: Wie ist Fixed Parameter Tractability definiert? Welchen Einfluss hat die Wahl des Parameters? Welche Ansätze für Algorithmen sind bekannt? Wie verhalten sich die Komplexitätsklassen FPT und $W[1]$ zueinander?

Literatur: [Nie06, Kapitel 1 und 3]

2. Problemkerne

Inhalt: Wie sind Problemkerne definiert? Wie kann die Größe von MAXSAT- und VERTEXCOVER-Instanzen reduziert werden, um FPT-Algorithmen zu erhalten?

Literatur: [Nie06, Kapitel 7]

3. Suchbäume mit beschränkter Tiefe

Inhalt: Wie können durch geschicktes Verkleinern des Suchbaums FPT-Algorithmen gefunden werden? Wie können CLUSTEREDITING, VERTEXCOVER und d -HITTINGSET auf diese Weise in FPT gelöst werden?

Literatur: [Nie06, Kapitel 8; DF99, Kapitel 3.1]

4. Dynamische Programmierung

Inhalt: Wie kann Dynamische Programmierung eingesetzt werden, um FPT-Algorithmen zu finden? Wie können BINARYKNAPSACK, STEINERTREE und MDFT auf diese Weise in FPT gelöst werden?

Literatur: [Nie06, Kapitel 9; DF99, Kapitel 3.2]

5. Baumdekompositionen

Inhalt: Was ist die Baumweite eines Graphen? Wie kann diese ausgenutzt werden, um FPT-Algorithmen für VERTEXCOVER und DOMINATINGSET in planaren Graphen anzugeben?

Literatur: [Nie06, Kapitel 10]

6. Color Coding und Hashing

Inhalt: Wie können zufällige Färbungen verwendet werden, um einen randomisierten FPT-Algorithmus für LONGESTPATH anzugeben? Was sind perfekte Hash-Familien, wie können diese zur Derandomisierung dieses Algorithmus eingesetzt werden, und wie können sie konstruiert werden?

Literatur: [Nie06, Kapitel 11.1; DF99, Kapitel 8.3]

7. Approximation und FPT

Inhalt: Welche Arten der Approximation von Optimierungsproblemen sind bekannt? Welche Bezüge gibt es zu FPT-Algorithmen?

Literatur: [DF99, Kapitel 4]

8. $W[1]$ -Härte

Inhalt: Wie können Reduktionen im parametrisierten Kontext definiert werden? Wie ist die Klasse $W[1]$ definiert? Was sind Beispiele von $W[1]$ -harten Problemen?

Literatur: [DF99, Kapitel 9–10]

Ablauf

- In der ersten Vorlesungswoche 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 [TWM13, 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 Abschlussarbeiten).
 - 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].
 - Der **Abgabeschluss** für Ausarbeitungen ist der erste Tag der Vorlesungszeit im folgenden Semester.

Literatur

- [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 1. Okt. 2013).
- [DF99] Rod G. Downey und Michael R. Fellows. *Parameterized Complexity*. New York: Springer, 1999. ISBN: 978-1-4612-6798-0.
- [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 1. Okt. 2013).
- [Nie06] Rolf Niedermeier. *Invitation to fixed-parameter algorithms*. Oxford University Press, 2006. ISBN: 0-19-856607-7.
- [TWM13] Till Tantau, Joseph Wright und Vedran Miletic. *The BEAMER class*. Version 3.27. 18. Juni 2013. URL: <http://mirror.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf> (besucht am 1. Okt. 2013).