

Komplexitätstheorie (SS 2006)

Markus Holzer

Institut für Informatik, Technische Universität München,
Boltzmannstraße 3, D-85748 Garching bei München, Germany
email: holzer@in.tum.de

1 Allgemeines

Bereich: 4+2 SWS im Bereich Informatik III

2 Inhaltsangabe und Literaturverzeichnis

1. Allgemeines
 - (a) Grundlegende Fragestellungen, einfache Beispiele wie Traveling Salesman, Komplexität von Brettspielen oder Bleistiftspielen, Wachstum von Funktionen (O -Notation) und Laufzeitvergleiche
2. Turingmaschinen und Komplexitätsklassen
 - (a) Definition Turingmaschine mit mehreren Arbeitsbändern, Konfiguration, Konfigurationsübergang, akzeptierende Berechnung, akzeptierte Sprache, Definition von Platz- und Zeitschranken, Bildung von Platz- und Zeitkomplexitätsklassen mittels det. und nichtdet. Turingmaschinen, Definition von Standardkomplexitätsklassen L, NL, P, NP, PSPACE.
 - (b) Bandreduktion und Bandkompression für platzbeschränkte Klassen, lineare Beschleunigung und Bandreduktion auf ein bzw. zwei Bändern bei zeitbeschränkten Klassen.
 - (c) Definition von konstruierbaren Funktionen, einfache Inklusionsbeziehungen zwischen Klassen (Enthaltensein durch mehr Platz bzw. mehr Zeit, det. in nichtdet.), Zeit induziert Platzschranke, Platz läßt sich in Zeit exponentiell in der Platzschranke simulieren, Konfigurationsgraph und Erreichbarkeitsargument.
 - (d) Satz von Savitch, divide and conquer Algorithmus, Definition von PSPACE über det. bzw. nichtdet. Turingmaschinen möglich.
 - (e) Hierarchiesätze für det. Platz- und Zeitklassen, Voraussetzungen zu den Hierarchiesätzen (Konstruierbarkeit der Schranke, Komplementabschluß, universelle Turingmaschine), Diagonalisierung, L echt in PSPACE.
 - (f) Komplementabschluß nichtdet. Platzklassen, nichtdet. und Komplementierung allgemein, det. Klassen sind unter Komplementbildung abgeschlossen, Erreichbarkeitsargument mittels Zählargument überprüfbar.
 - (g) Translationstechniken, Definition Padding, Translationssatz für Platz- und Zeitklassen, Downward Separation, P nicht gleich DLBA, Anwendung des Translationslemmas zur Bildung einer nichtdet. Platzhierarchie.

3. Reduktion und vollständige Probleme
 - (a) Begriff der Reduktion many-one Reduktion, log-Platz und pol-Zeit Reduktionen, einfache Eigenschaften (reflexiv, transitiv), Definition von Härte und Vollständigkeit, Grapherreichbarkeit ist NL-vollständig.
 - (b) Es gibt NP vollständige Problem in NTIME(n), generisches vollständiges NP Problem.
 - (c) Der Satz von Cook, Definition Boolesche Formel, SAT ist NP-vollständig, Trick mit Rechentepich, KNF versus DNF, weitere NP-vollständige Problem: SAT in KNF, 3SAT, lineare Programmierung über ganzen Zahlen, Vertex Cover, Hamiltonpfad; nicht vollständige Mengen innerhalb von NP ohne P.
 - (d) Vollständige Problem für P und PSPACE: Leerheit von kontextfreien Sprachen, Auswertung von Booleschen Schaltkreisen, quantifizierte Boolesche Formeln.
4. Alternierende Turingmaschinen
 - (a) Definition alternierende Turingmaschine, Berechnungsbaum, akzeptierende Berechnung, akzeptierende Sprache, Klassenbildung, Standardkompleititätsklassen AL und AP.
 - (b) Grundlegende Techniken lassen sich auf alternierende Turingmaschinen übertragen, nichtdet. Platz in alternierender Zeit mit quadratischem Aufwand, alternierende Zeit in det. Platz ohne Verlust, Satz von Savitch als Spezialfall, alternierender Platz in det. Zeit mit exponentiellem Aufwand, P gleich AL, alternierende Turingmaschinen mit sublinearer Laufzeit, indexed tape Turingmaschine.
 - (c) Beschränkte Alternierung, Klassenbildung, polynomielle Zeithierarchie, Inklusionsstruktur, vollständige Problem auf Basis von QBF, Konsequenzen bei Gleichheiten wie z.B. P gleich NP, oder NP gleich co-NP usw., logarithmische Plathierarchie, Kollaps durch NL gleich co-NL.
5. Orakel Turingmaschinen
 - (a) Definition von \exists^P und \forall^P , Quantorencharakterisierung der polynomiellen Zeithierarchie.
 - (b) Definition Orakel Turingmaschine, Orakelcharakterisierung der polynomiellen Zeithierarchie, Trick guess and delay query.
 - (c) Relativierung, Existenz eines Orakels so daß P gleich NP, Konstruktion eines Orakels so daß P ungleich NP, weitere Konsequenzen, Orakel für P ungleich PSPACE, P ungleich co-NP, NP ungleich co-NP, P ungleich NP aber NP gleich co-NP, P ungleich NP aber NP gleich PSPACE.

Literatur

1. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1988.
2. J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*, volume 22 of *EATCS Monographs on Theoretical Computer Science*. Springer, 1990.
3. J. E. Hopcroft and J. D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley, 1968.

4. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.
5. C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
6. K. R. Reischuk. *Einführung in die Komplexitätstheorie*. Teubner, 1990.

Weiterführende Literatur

1. L. A. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. EATCS Monographs in Theoretical Computer Science. Springer, 2002.
2. K. Wagner and G. Wechsung. *Computational Complexity*. Mathematics and its applications (East Europeans series). VEB Deutscher Verlag der Wissenschaften, Berlin, 1986.