



Vorlesungsinhalt

Semester: Wintersemester 2008/09
Vorlesung: Algorithmen und Datenstrukturen (EI) (4+2, 6 ECTS)
(mit Zentralübung)
Dozent: Prof. Dr. Ernst W. Mayr
Übungsleitung: Stefan Schmid

- Texte:** Alfred V. Aho and John E. Hopcroft and Jeffrey D. Ullman:
“The design and analysis of computer algorithms”
Addison-Wesley Publishing Company: Reading (MA), 1974
- Thomas H. Cormen and Charles E. Leiserson and R.L. Rivest
and Clifford Stein:
“Introduction to algorithms”
McGraw-Hill, 1990
- Hartmut Ernst:
“Grundkurs Informatik: Grundlagen und Konzepte für die
erfolgreiche IT-Praxis — Eine umfassende, praxisorientierte
Einführung”
Vieweg Verlag: Braunschweig-Wiesbaden, 3. Auflage, 2003
- Volker Heun:
“Grundlegende Algorithmen: Einführung in den Entwurf und die
Analyse effizienter Algorithmen”
2. Aufl., Vieweg: Braunschweig-Wiesbaden, 2003
- John E. Hopcroft, Jeffrey D. Ullman:
“Introduction to Automata Theory, Languages, and Computation”
Donald E. Knuth:
“The art of computer programming. V. 1: Fundamental algorithms”
3rd ed., Addison-Wesley Publishing Company: Reading (MA), 1997
- Kurt Mehlhorn, Peter Sanders:
“Algorithms and Data Structures — The Basic Toolbox”
Springer-Verlag: Berlin-Heidelberg, 2008
- Steven S. Skiena:
“The algorithm design manual”
Springer-Verlag: Berlin-Heidelberg-New York, 1998
- Robert E. Tarjan:
“Data Structures and Network Algorithms”
CBMS-NSF Regional Conference Series in Applied Mathematics,
SIAM, Philadelphia, PA, 1983

Vorlesungsinhalt:

0. Organisatorisches

1. Termine
2. Literatur

I. Historisches und Begriffliches

1. EI, Informatik und ihre Struktur
2. Abstraktionsebenen

II. Boolesche Algebra und Logik

1. Einleitung
2. Algebren
 - 2.1 Grundbegriffe
 - 2.2 Eigenschaften
 - 2.3 Vereinbarung
 - 2.4 Schreibweisen und Sprachgebrauch
 - 2.5 Beispiele für Axiome einer Algebra
 - 2.6 Boolescher Verband
 - 2.7 Boolesche Algebren
 - 2.8 Boolesche Ausdrücke und Funktionen
 1. Boolesche Funktionen mit einem Argument
 2. Boolesche Funktionen mit zwei Argumenten
3. Normalformen Boolescher Funktionen
 - 3.1 Vollkonjunktion und disjunktive Normalform (DNF)
 - 3.2 Ableitung der disjunktiven Normalform aus einer Wertetabelle
 - 3.3 Volldisjunktion und konjunktive Normalform (KNF/CNF)
 - 3.4 Ableitung der konjunktiven Normalform aus einer Wertetabelle
 - 3.5 Vergleich von DNF und KNF
 - 3.6 Bemerkungen zur Umformung boolescher Formeln (NAND)
 - 3.7 Umformung zur NOR-Darstellung
 - 3.8 Addierer als Anwendungsbeispiel
4. Morphismen
 - 4.1 Isomorphismus
 - 4.2 Wie viele Boolesche Algebren gibt es?
5. Ergänzungen zu booleschen Funktionen

III. Automatentheorie

1. Begriff des Automaten
2. Eigenschaften von Automaten
 - 2.1 Ein- und Ausgabe
 - 2.2 Innere Zustände
 - 2.3 Ausgabefunktion
 - 2.4 Zustandsübergangsfunktion
 - 2.5 Determinismus
 - 2.6 Verdeutlichung verwendeter Begriffe
3. Darstellung
 - 3.1 Graphische Darstellung
 - 3.2 Tabellarische Darstellung
 - 3.3 Mathematische Darstellung
4. Deterministische endliche Automaten
 - 4.1 Asynchroner Automat
5. Turing-Maschinen
 - 5.1 Alan Turing
 - 5.2 Struktur einer Turing-Maschine
6. Linear beschränkte Automaten
7. Kellerautomaten
8. Deterministische Kellerautomaten

IV. Formale Sprachen und Grammatiken

1. Begriffe und Notationen
2. Sprachkonzept
 - 2.1 Erzeugung und Sprachschatz
 - 2.2 Darstellungsformen
3. Die Chomsky-Hierarchie
 - 3.1 Phrasenstrukturgrammatik, Chomsky-Grammatik
 - 3.2 Die Chomsky-Hierarchie
4. Das Wortproblem
 - 4.1 Die Existenz von Ableitungen eines Wortes
 - 4.2 Ableitungsgraph und Ableitungsbaum
5. Eigenschaften regulärer Automaten und Sprachen
 - 5.1 Äquivalenz von NFA und DFA

- 5.2 NFAs mit ϵ -Übergängen
- 5.3 Entfernen von ϵ -Übergängen
- 5.4 Endliche Automaten und reguläre Sprachen
- 5.5 Reguläre Ausdrücke
- 5.6 Abschlusseigenschaften regulärer Sprachen
- 5.7 Konstruktion minimaler endlicher Automaten
- 6. Einige Ergebnisse zur Entscheidbarkeit
- 7. Das Wortproblem für kontextfreie Sprachen
 - 7.1 Die Chomsky-Normalform
 - 7.2 Der Cocke-Younger-Kasami-Algorithmus
 - 7.3 Das Pumping-Lemma für kontextfreie Sprachen
 - 7.4 Algorithmen für kontextfreie Sprachen/Algorithmen
 - 7.5 Greibach-Normalform
 - 7.6 Kellerautomaten und kontextfreie Sprachen
 - 7.7 LR(k)-Grammatiken
 - 7.8 LL(k)-Grammatiken
- 8. Das Halteproblem
- 9. Unentscheidbarkeit

V. Algorithmen und Datenstrukturen

- 1. Analyse von Algorithmen
 - 1.1 Referenzmaschine
 - 1.2 Zeit- und Platzkomplexität
 - 1.3 Worst Case-Analyse
 - 1.4 Average Case-Analyse
- 2. Sortierverfahren
 - 2.1 Selection-Sort
 - 2.2 Insertion-Sort
 - 2.3 Merge-Sort
 - 2.4 Quick-Sort
 - 2.5 Heap-Sort
 - 2.6 Vergleichsbasierte Sortierverfahren
 - 2.7 Bucket-Sort
- 3. Suchverfahren
 - 3.1 Binäre/natürliche Suchbäume
 - 3.2 AVL-Bäume
 - 3.3 (a, b) -Bäume

- 3.4 Hashing
 - 1. Kollisionsauflösung
 - 2. Universelle Hashfunktionen
- 4. Vorrangwarteschlangen (priority queues)
- 5. Union/Find-Datenstrukturen
 - 5.1 Motivation
 - 5.2 Union/Find-Datenstruktur
 - 1. Intrees
 - 2. Gewichtete Union (erste Verbesserung)
 - 3. Pfad-Kompression mit gewichteter Union (zweite Verbesserung)
- 6. Traversierung von Graphen
 - 6.1 DFS-Algorithmus
 - 6.2 BFS-Algorithmus
- 7. Minimale Spannbäume
 - 7.1 Konstruktion von minimalen Spannbäumen
 - 7.2 Generischer minimaler Spannbaum-Algorithmus
 - 7.3 Kruskals Algorithmus
 - 7.4 Prims Algorithmus
- 8. Kürzeste Pfade
 - 8.1 Grundlegende Begriffe
 - 8.2 Das single-source-shortest-path-Problem
 - 1. Dijkstras Algorithmus
 - 2. Bellman-Ford-Algorithmus
 - 8.3 Floyds Algorithmus für das all-pairs-shortest-path-Problem
 - 8.4 Digraphen mit negativen Kantengewichten
 - 1. Grundsätzliches
 - 2. Modifikation des Bellman-Ford-Algorithmus
 - 3. Modifikation des Floyd-Algorithmus
 - 4. Der Algorithmus von Johnson
 - 8.5 Zusammenfassung