

Allgemeines

<i>Dozent:</i>	Prof. Dr. Hartmut Ernst / Prof. Dr. Bernhard Holaubek
<i>Verantwortlich:</i>	Prof. Dr. Hartmut Ernst / Prof. Dr. Bernhard Holaubek
<i>Studiengang:</i>	Bachelor, Diplom
<i>Kennung:</i>	Bachelor / Diplom: Pflichtfach im Grundstudium
<i>Voraussetzungen:</i>	keine
<i>Sprache:</i>	Deutsch
<i>Lehrform:</i>	2 SWS seminaristischer Unterricht und Übungen
<i>Arbeitsaufwand:</i>	30 Stunden Präsenzzeit, 30 Stunden Selbststudium
<i>Leistungspunkte:</i>	2
<i>Medienform:</i>	Vortrag mit Tablet-PC und Beamer, Skriptum (PDF), veranstaltungsspezifische Seiten im ELRON-Server, Bearbeitung von Aufgaben in den Übungen.
<i>Prüfung:</i>	Schriftliche Prüfung, 90 Min., mit allen Unterlagen

Lernziele und Inhalt

Richtziel

Die Studierenden lernen die Grundlagen der theoretischen Informatik kennen, insbesondere Automatentheorie, formale Sprachen und Komplexität.

Inhaltsübersicht

Nach einer Einführung in die Automatentheorie einschließlich Turingmaschinen und die Grundlagen der formalen Sprachen wird auf einige Aspekte der Komplexitätstheorie und Algorithmentheorie eingegangen.

Inhalt

1. *Automatentheorie*
 - 1.1 Grundbegriffe der Automatentheorie
 - 1.2 Automaten und Halbgruppen
 - 1.2 Turing-Maschinen
2. *Einführung in die Theorie der formalen Sprachen*
 - 2.1 Definition von formalen Sprachen
 - 2.2 Die Chomsky-Hierarchie
 - 2.3 Das Pumping-Theorem
 - 2.4 Die Analyse von Wörter
 - 2.5 Compiler
3. *Berechenbarkeit*
 - 3.1 Problemstellung
 - 3.2 Entscheidungsproblem und Church-Turing-These
 - 3.3 Das Halteproblem
 - 3.4 Primitiv rekursive und μ -rekursive Funktionen
4. *Komplexität*
 - 4.1 Einführung
 - 4.2 Klassifikation der Komplexität
 - 4.3 Ausführbare und nicht-ausführbare Algorithmen

5. *Optimierung von Algorithmen*
 - 5.1 Vereinfachen komplexer Operationen
 - 5.2 Teile und Herrsche
 - 5.3 Greedy-Strategien
 - 5.4 Genetische Algorithmen
 - 5.5 Probabilistische und heuristische Algorithmen
6. *Rekursion*
 - 6.1 Definition und einführende Beispiele
 - 6.2 Rekursive Programmierung und Iteration
 - 6.3 Backtracking

Literatur

Besonders empfohlen

1. Eirund, H., Müller, B. und Schreiber, G.: *Formale Beschreibungsverfahren der Informatik – ein Arbeitsbuch für die Praxis*. Teubner (2000)
2. Ernst, H.: *Grundlagen und Konzepte der Informatik*. Vieweg (2003)
3. Hopcroft, J.E., Motwani, R. und Ullmann, J.D.: *Einführung in die Automatentheorie, formalen Sprachen und Komplexitätstheorie*. Pearson Studium (2002)
4. Schöneburg, E., Heinzmann, F. und Feddersen, S.: *Genetische Algorithmen und Evolutionsstrategien*. Addison-Wesley (1994)
5. Schöning, U.: *Theoretische Informatik - kurz gefasst*. Spektrum Akad. Verlag (2001)

Zusätzlich empfohlen

6. Aho, A.V. et al.: *Compilerbau 1 und 2*. Oldenbourg (1999)
7. Erk, K. und Priese, L.: *Theoretische Inf. Eine umfassende Einführung*. Springer (2001)
8. Goos, G.: *Vorlesungen über Informatik 1, 2, 3, 4*. Springer (2000)
9. Hofstadter, D.R.: *Gödel, Escher, Bach*. Klett-Cotta (1989)
10. Knuth, D.E.: *The Art of Computer Programming*. Addison-Wesley (1981)
11. Press, W. H. et al.: *Numerical Recipes in C*. Cambridge University Press (1994)
12. Sedgewick, R.: *Algorithms*. Addison-Wesley (1988)
14. Wirth, N.: *Grundlagen und Techniken des Compilerbaus*. Oldenbourg (1995)