

Modul “Theoretische Informatik II”

Ziel

Einführung in Berechenbarkeits-, Automaten-, Komplexitätstheorie

Inhalt

Im ersten Teil der Vorlesung werden verschiedene Sprachklassen (z.B. reguläre und kontextfreie Sprachen) eingeführt, die u.a. deswegen wichtig sind, weil sie zur Beschreibung der Syntax von Programmiersprachen verwendet werden. Parallel dazu werden je Sprachklasse dazugehörige Automatenmodelle (z.B. endliche Automaten, Kellerautomaten, Turingmaschinen) vorgestellt, die als abstrakte Algorithmenbeschreibung, wie die jeweiligen Sprachen akzeptiert werden, aufgefasst werden können. Zur Klassifikation dieser Sprachen dienen Grammatiken.

Im zweiten Teil der Vorlesung wird untersucht, welche Funktionen algorithmisch berechenbar bzw. welche Probleme algorithmisch entscheidbar sind. Dazu wird der Begriff des Algorithmus formalisiert. Turingmaschinen und Grammatiken stellen sich als äquivalente Ansätze heraus. Es wird gezeigt, dass es Probleme gibt, die nicht algorithmisch entscheidbar sind. Dazu gehören leider auch viele Probleme von praktischem Interesse.

Im dritten Teil der Vorlesung geht es um die Komplexität von Algorithmen, d.h. wieviel Zeit und Speicherplatz zum Lösen einer Aufgabe benötigt wird. Insbesondere werden Probleme betrachtet, die deterministisch oder nichtdeterministisch in polynomieller Zeit lösbar sind. Diese Problemklassen sind unter den Namen P und NP bekannt.

Literatur

- V. Claus & E.-R. Olderog, *Grundbegriffe der Theoretischen Informatik*, Skript.
- U. Schöningh, *Theoretische Informatik – kurzgefasst*, Spektrum, 5. Auflage, 2008.
- J.E. Hopcroft, R. Motwani & J.D. Ullman: *Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie*, Addison Wesley, 2nd Edition, 2001.