



1. Einleitung

Die Theoretische Informatik untersucht grundlegende Konzepte, Modelle und Vorgehensweisen, die allen Bereichen der Informatik zugrunde liegen. Sie liegt nahe bei der Mathematik: Ihre Gegenstände werden formal definiert, und es werden viele Beweise geführt. Damit liefert sie eine formale Ausdrucksweise, die überall in der Informatik dann verwendet wird, wenn man etwas exakt beschreiben will. Die Theoretische Informatik ist aber anschaulicher als die Mathematik, weil immer wieder die praktischen Probleme durchscheinen, die modelliert werden, und sie ist auch algorithmischer, und gerade in dieser Schwellenposition zwischen Theorie und Praxis liegt ihr Reiz.

Es geht in diesem Buch um zwei große Themengebiete:

- Mit *formalen Sprachen* untersucht man den Aufbau von Sprachen und teilt Sprachphänomene in verschiedene Grade von Strukturkomplexität auf. Formale Sprachen werden im Compilerbau einerseits beim Entwurf von Programmiersprachen benutzt – sie helfen, unnötig komplizierte Konstrukte zu vermeiden – und andererseits bei der Analyse von Programmen. Ihren Ursprung haben die formalen Sprachen allerdings in der Beschreibung und Verarbeitung natürlicher Sprachen.
- In der *Theorie der Berechenbarkeit* betrachtet man kleine, abstrakte Modelle eines Computers, die viel einfacher gebaut sind als ein konkreter Computer, aber genauso mächtig sind. Mit diesen Modellen kann man z.B. untersuchen, welche Probleme überhaupt von einem Computer lösbar sind. Wir betrachten zum einen Modelle, die Verwandtschaft zu üblichen Konzepten aus der Praktischen Informatik haben, zum anderen aber auch ganz andere, exotische Modelle, die eine Vorstellung davon geben, welche vielfältigen Formen eine „Berechnung“ annehmen kann.

Interessant ist bei lösbaren Problemen die Frage nach der *Komplexität*, der Rechenzeit, die man aufwenden muss, um sie zu lösen. Man kann Berechnungsverfahren in verschiedene Klassen einteilen, je nachdem, wie schwierig sie sind bzw. welche Laufzeit sie brauchen.