

1 Einleitung

Im Automatenteil wollen wir verschiedene Automaten(-typen) und Grammatiken betrachten, die in der Informatik (und auch darüberhinaus) an den verschiedensten Stellen zur Anwendung kommen. Bei den Automaten wollen wir insbesondere den endlichen Automaten betrachten, den Kellerautomaten und die Turingmaschine. Endliche Automaten dienen u.a. als Modellierungswerkzeug bspw. beim Entwurf von Kommunikationsprotokollen, sind eine wichtige Datenstruktur, um Mengen mit unendliche vielen Elementen zu beschreiben, werden bei der Texterkennung eingesetzt und vieles mehr. Zudem sind endliche Automaten ein ganz grundlegender Automatentyp auf dem viele weitere Automatentypen didaktisch aufgebaut werden können. Der Kellerautomat spielt z.B. beim Compilerbau eine wichtige Rolle und die Turingmaschine dient als abstraktes Modell für einen Computer bzw. eine einen Algorithmus ausführende Maschine. Mit ihr können grundsätzliche Überlegung über die Mächtigkeit von Algorithmen angestellt werden.

Bei den Grammatiken werden wir uns insb. mit kontextfreien Grammatiken beschäftigen. Diese treten z.B. bei der Beschreibung der Syntax von Programmiersprachen auf.

Auf viele der oben angesprochenen Themen werden wir nicht im einzelnen eingehen können, sie werden aber in anderen Vorlesungen in der Informatikausbildung immer wieder vorkommen, so dass ein hier erworbenes gutes Grundverständnis für das ganze Studium sehr nützlich sein kann.

Wir werden die einzelnen Modelle einführen und dann vor allem ihre Mächtigkeit betrachten. Kann ein Kellerautomat z.B. wirklich “mehr” als ein endlicher Automat? Kann eine Turingmaschine noch mehr? Im welchem Verhältnis stehen Grammatiken und Automaten zueinander? Dazu werden wir uns für die einzelnen Modelle definieren, was deren *Sprache* ist und darüber Sprachfamilien definieren. Wir können dann betrachten, welche Sprachen von welchen Modellen erkannt werden können und Werkzeuge entwickeln, mit denen wir sagen können, dass eine Sprache *nicht* von einem bestimmten Modell erfasst werden kann.

Die Turingmaschinen werden wir dann nutzen, um zu sehen, dass es Probleme gibt, die trotz ihrer Relevanz nicht algorithmisch gelöst werden können. Wir werden also eine für die Informatik fundamentale Grenze sehen.

Im Anschluss werden wir dann, nachdem wir gesehen haben, dass es Probleme gibt, die nicht algorithmisch gelöst werden können, noch sehen, dass es Probleme gibt, die algorithmisch nicht *effizient* gelöst werden können. D.h. es

gibt zwar einen Algorithmus, aber dieser braucht so lange, dass es für jede Anwendung unerheblich ist, dass er existiert. Mit diesem Thema werden wir einen kleinen Blick in das weite Gebiet der Komplexitätstheorie werfen.

Beginnen werden wir mit dem einfachsten Modell, dem deterministischen endlichen Automaten ...