

# Mathematik 3 - Grundlagen der diskreten Mathematik

## Modulbeschreibung

Julia Stoll, small changes by Th. Dorssers

August 29, 2009



Hogeschool Techniek en Logistiek

# Contents

<b>1 Grundlagen der diskreten Mathematik (MAT3)</b>	<b>3</b>
1.1 Mengenlehre . . . . .	3
1.2 Relationen . . . . .	4
1.3 Funktionen . . . . .	4
1.4 Graphentheorie . . . . .	5
1.5 Sprachen, Grammatiken, Automaten . . . . .	5
<b>2 Organisation</b>	<b>5</b>
<b>3 Literatur</b>	<b>6</b>

# 1 Grundlagen der diskreten Mathematik (MAT3)

In dem Kurs Mathematik 3 beschäftigen wir uns mit Mengenlehren, Relationen und Funktionen aus der Sicht der Informatik, um Algorithmen als Berechnungsverfahren spezifizieren zu können. Hinzu kommen einfachen Konzepte von Automaten und formalen Sprachen. Der Kurs ist damit eine Ergänzung zu den Fächern “Algorithmen und Datenstrukturen“ (Fundamentals 1 / FND1) und “Compilerbau“ (Fundamentals 3 / FND3). Reguläre Ausdrücke werden auch im Bereich “Concurrency“ (FND2 / PRO3) benötigt. Einige Eigenschaften von endlichen Automaten sind im Modul “Modellieren 2“ (MOD 2) ebenfalls von Bedeutung.

Voraussetzung für diesen Kurs ist die erfolgreiche Teilnahme an den Mathematikkursen (MAT1 und MAT2) aus dem ersten Studienjahr.

## 1.1 Mengenlehre

In der Mengenlehre werden Mengen spezifiziert und dargestellt. Begriffe, wie universelle Menge (oder Grundmengen), Komplementmenge, Durchschnitts- und Vereinigungsmengen, werden eingeführt und aus der Sicht der Informatik diskutiert. Die Bedeutung von Zahlenmengen der natürlichen Zahlen wird betrachtet. Der Unterschied zwischen der Menge der rationalen Zahlen  $\mathbb{Q}$  und der Menge der reellen Zahlen  $\mathbb{R}$  in Bezug auf abzählbare und überabzählbare Mengen wird diskutiert. Eigenschaften von Mengen werden, insbesondere für Teilmengen untersucht, um schließlich Operationen auf Mengen zu definieren und zu üben. Eine besondere Rolle spielt die formale Notation und Darstellung von (Teil-)Mengen und damit der Spezifikation von so genannten Bereichen (oder Domänen, engl. domains). Als Anwendungsdomänen sind Mengen auch in der Analysephase eines jeden Software-Projekts zu spezifizieren. Damit werden die zu verarbeitenden “Elemente“ in einer Anwendung zu ermitteln. Die Fähigkeit wichtige Elemente im Anwendungsfall von Unwichtigem zu trennen, ist ein wesentlicher Ansatz um Probleme in einem möglichen Modell zu erfassen und schliesslich verarbeiten zu können. Die Fähigkeit der Abstraktion und Modellbildung wird somit geschult.

## 1.2 Relationen

Relationen beschreiben die Beziehungen zwischen Elementen von Mengen durch geordnete Paare über Produktmengen. Eigenschaften von Relationen werden untersucht. Dazu gehören z.B. die Komposition von Relationen. Eigenschaften von Äquivalenz- und Ordnungsrelationen werden betrachtet. Die Studierenden, die diesen Kurs besuchen haben bereits erste Erfahrungen in der Anwendung von Relationen gesammelt. Relationen in der Darstellung von Tabellen sind aus der Anwendung von Datenbanken bekannt. Relationen können aber nicht nur in Form Tabellen dargestellt werden, sondern auch durch Matrizen oder (gerichtete) Graphen. Graphentheorie und Graphenalgorithmen sind ein wesentlicher Bestandteil der Moduls “Algorithmen und Datenstrukturen“. Graphen ihrerseits werden oft in Form von Matrizen repräsentiert, so dass das Erlernen dieser Notationen, das Verständnis der später zu studierenden Algorithmen vereinfachen sollte. Die Studierenden sollen in der Lage sein, sich auf die Algorithmen zu konzentrieren. Typische Anwendungsfälle von Graphenalgorithmen sind Routing-Algorithmen in Netzwerken, Ticketvergabe in verteilten Systemen zur Synchronisation von Zuständen oder auch Kapazitätberechnungen im Bereich von Flussproblemen, die ihrerseits z.B. in den Bereich der Logistik fallen.

## 1.3 Funktionen

Funktionen ordnen einem Element einer Menge genau ein Element einer anderen Menge zu. Eigenschaften von Funktionen, wie Injektivität, Surjektivität und Bijektivität, werden eingeführt. Desweiteren werden Umkehrfunktionen betrachtet und ins Verhältnis zum Relationenbegriff gesetzt. Für Testverfahren von Software kann es wichtig sein zu wissen, ob eine Funktion umkehrbar ist. Mit der Einführung von rekursiven Funktionen soll das Verständnis geweckt werden, welche Funktionen berechenbar sind. Auch wenn eine Funktion rekursiv definiert werden kann, heißt das, dass sie berechnet werden kann; aber es bedeutet noch nicht, dass es auch ein effizientes Berechnungsverfahren gibt. Anschließend wird die Groß-*Oh*-Notation eingeführt, um die Komplexität von Algorithmen und damit eine erstes Effizienzmaß kennenzulernen. Dazu werden so genannte Standardfunktionen diskutiert und der Bezug zu den Maßen für Algorithmen hergestellt. Der Aufwand einiger Algorithmen wird abgeschätzt, so dass grundlegende Methoden der Komplexitätstheorie im Module “Algorithmen und Datenstrukturen“ (FND1) angewendet und vertieft werden können.

## 1.4 Graphentheorie

Graphen sind eine der wichtigsten Strukturen in der Informatik. Mit Graphen können insbesondere Abhängigkeiten zwischen Daten beschrieben werden. Graphen bestehen aus Knoten und Kanten. Oftmals repräsentieren Knoten Daten und die Kanten beschreiben die Beziehungen zwischen den Daten. Die Kanten können ungerichtet oder gerichtet sein. Typische Eigenschaften von ungerichteten und gerichteten Graphen werden untersucht. Dazu gehören u.a. die Wege, Zyklen, Hamiltonsche und Eulersche Kreise. Als Beispiele werden die Zusammenhänge zu Wege- und Routingproblemen hergestellt. Eine Unterform gerichteter Graphen sind Bäume. Mit Bäumen lassen sich insbesondere hierarchische Strukturen beschreiben. Hierarchische Strukturen werden in vielen Anwendungen benötigt, um Daten zu nach (vorgegebenen) Kriterien zu ordnen. Sowohl ungerichtete und gerichtete Graphen also auch Bäume werden als markierte Strukturen untersucht. Spezialfälle sind die Betrachtung von Flussproblemen und das so genannte Heiratsproblem. Bäume werden später auch als Ableitungsbäume eingeführt, so dass die Einführung von formalen Sprachen motiviert werden kann.

## 1.5 Sprachen, Grammatiken, Automaten

Im letzten Abschnitt werden Grundbegriffe der formalen Sprachen und Automatentheorie eingeführt. Zum einen werden die Konkatenation von Buchstaben zu Zeichenketten und die Bildung von Sprachen betrachtet. Zum anderen werden Grammatiken als erzeugenden Systeme für Sprachen und ihre unterliegenden Eigenschaften entlang der Chomsky-Hierarchie besprochen. Dabei liegt die Konzentration sowohl auf den Konzepten der regulären Ausdrücke, der regulären Sprachen und endlichen Automaten als auch den kontextfreien Sprachen und den Modellen nichtdeterministischer und deterministischer Kellerautomaten. Diese beiden Klassen von Automatenmodellen werden im anschließenden Kurs des Compilerbaus vertieft.

# 2 Organisation

**Voraussetzungen** MAT1 und MAT2 aus dem ersten Studienjahr

**Arbeitsorganisation** Das Modul hat einen Umfang von 4 Credits (d.h. für den Studierenden beträgt der Arbeitsaufwand ca. 112 Arbeitstunden). Im wöchentlichen Unterricht

von (jetzt) sieben Wochen werden die formalen Definitionen besprochen, (Anwendungs-)Beispiele gemeinsam betrachtet und Aufgaben gelöst. Aufgaben aus dem (vorausgegangenen) Selbststudium werden besprochen. Die Themen verteilen sich über die Wochen, wie folgt:

**Woche 1** - Mengenlehre

**Woche 2** - Relationen

**Woche 3** - Funktionen

**Woche 4** - Ungerichtete Graphen

**Woche 5** - Gerichtete Graphen

**Woche 6** - Reguläre Ausdrücke und reguläre Sprachen

**Woche 7** - Endliche Automaten und reguläre Sprachen

**Schriftliche Prüfung** Der Kurs schließt mit einer zweistündigen Klausur ab (120 min). Die Credits erwirbt der Studierenden mit dem Bestehen der Klausur am Ende des Quartesters.

### **3 Literatur**

S. Lipschutz, M. Lipson: Schaum's outline - Discrete Mathematics. 2007, 3. Aufl.