



Hochschule Furtwangen
Fakultät Informatik

Automaten und Formale Sprachen

Prof. Dr. Bernhard Hollunder
Bernhard.Hollunder@hs-furtwangen.de

September 2024

- Skript Nr.: 025
- Immer bis Freitag Zeit zum Abgeben der Aufgaben

Automaten und Formale Sprachen

Prof. Dr. Bernhard Hollunder
bernhard.hollunder@hs-furtwangen.de
Fakultät Informatik
Hochschule Furtwangen

Wintersemester 2024/25
Version vom 20. September 2024



Literatur



Zusätzliche Informationen zu den Vorlesungsinhalten wie etwa ausführliche textuelle Beschreibungen, weitere Beispiele sowie Beweise zu Theoremen finden Sie im folgenden Buch:

Gottfried Vossen und Kurt-Ulrich Witt.
Grundkurs Theoretische Informatik Eine anwendungsbezogene Einführung - Für Studierende in allen Informatik-Studiengängen.
Springer Fachmedien, 2016.

Das Buch ist in der HFU-Bibliothek digital als pdf-Dokument verfügbar.

Einordnung des Moduls



- Das Modul behandelt Kernthemen der theoretischen Informatik.
- Legt die Grundlagen, um zentrale Konzepte der Informatik zu verstehen.
- Hoher Anwendungsbezug, z.B. Definition von Programmiersprachen, Beschreibung von Datenstrukturen, Modellierung von Abläufen, Abstraktionen, ...
- Erfassen von Zusammenhängen zwischen scheinbar sehr unterschiedlichen Themengebieten.
- Was ist Berechenbarkeit und was für Berechenbarkeitsmodelle gibt es?
- Gibt es Probleme, die algorithmisch nicht gelöst werden können?
- Gibt es Grenzen der Berechenbarkeit? Und wenn ja, was bedeutet dies für die Informatik?
- Ausgewählte Beispiele auf den folgenden Folien zeigen die Themenbreite des Moduls.

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

3 / 100

Beispiel 1: Darstellung von Datumsformaten



ISO 8601 ist ein internationaler Standard der ISO zur Darstellung von Datumsformaten und Zeitangaben. Verschiedene Repräsentationen werden spezifiziert. Beispiele:

Erzeugung in einer Kommandozeile:
> date -Iseconds
> 2024-07-27T10:58:03+02:00

Programmatisch in Python:
> print(datetime.datetime.now())
> 2024-07-27 12:19:40.552175

Fragen:

- Wie kann ein gültiges Datumsformat erkannt und verarbeitet werden?
- Wie kann die Menge aller gültigen Datumsformate auf einfache Weise präzise definiert werden?

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

4 / 100

- Rechner braucht formale Handhabung für Formate
 - Beschreibung im Klartext kann nicht verarbeitet werden

Beispiel 2: Gültige Programme



Wird das folgende Java-Programm erfolgreich übersetzt?

```
public class IsThisValidJavaProgram {  
    int i = 0,0,0,0;  
    public static void main(String[] args) {  
        for (;;) {  
        }  
    }  
}
```

Weitergehende Fragen:

- Wir kann (schnell) geprüft werden, dass ein gegebenes Java-Programm erfolgreich übersetzt werden kann?
- Wie kann die Menge aller gültigen Java-Programme spezifiziert werden?

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

5 / 100

Beispiel 3: Collatz-Funktion



Können Sie das Verhalten der folgenden Funktion beschreiben?

```
public long collatz(long n) {  
    if (n <= 1)  
        return 1;  
    if (n % 2 == 0) {  
        return collatz(n / 2);  
    }  
    return collatz(3 * n + 1);  
}
```

Weitergehende Fragen:

- Wie kann geprüft werden, für welche Eingaben die Funktion collatz nicht terminiert?
- Welche Informationen können grundsätzlich von einem Compiler bei der statischen Codeanalyse abgeleitet werden?

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

6 / 100

Beispiel 4: Effiziente Suche



In einem Text (z.B. in einem pdf-Dokument) soll nach einem bestimmten Wort gesucht werden. Ein einfaches Verfahren ist wie folgt:

```
public class Teilworterkennung {  
    boolean search(String text, String searchText) {  
        for (int i = 0; i < text.length(); i++) {  
            for (int j = 0; j < searchText.length() &&  
                i + j < text.length(); j++) {  
                if (text.charAt(i+j) != searchText.charAt(j))  
                    break;  
                if (j == searchText.length() - 1)  
                    return true;  
            }  
        }  
        return false;  
    }  
}
```

Lässt sich diese Strategie signifikant verbessern?

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

7 / 100

- $O(n^2)$
 - Strategie lässt sich verbessern zu $O(n \cdot \log(n))$

Begrifflichkeiten



Automaten: Mehr oder weniger komplexe Rechenmaschinen, meist geringere Komplexität als "volle" Programmiersprachen.

Formale Sprachen: Definieren Mengen von Wörtern (d.h. Zeichenfolgen), z.B. die Menge aller gültigen Java-Programme oder Datumswerte.

Voraussetzung: Sicherer Umgang mit

- Mengen
 - z.B. $\{a, b, aba\}$, $\{\}$, $\{\{\}\}$, $\{0, 1, 2, 3, 4, \dots\}$
- Potenzmengen
 - $\{\{\}, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$
- Relationen
 - z.B. binäre Relation $R \subseteq A \times B = \{(a, b) \mid a \in A, b \in B\}$
- Funktionen
 - z.B. diff. $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ mit $\text{diff}(m, n) = \begin{cases} m - n & \text{falls } m \geq n \\ 0 & \text{sonst} \end{cases}$

Beachte: Im folgenden gilt: $\mathbb{N} = \{0, 1, 2, 3, \dots\}$.

Bernhard Hubander – Fakultät Informatik Automaten und Formale Sprachen

8 / 100