# Automaten und Formale Sprachen
# 1. Vorlesung

Prof. Dr. Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Wintersemester 2023/24

1 Einführung

2 Grundbegriffe

3 Rechtslineare Sprachen

4 Kontextfreie Sprachen

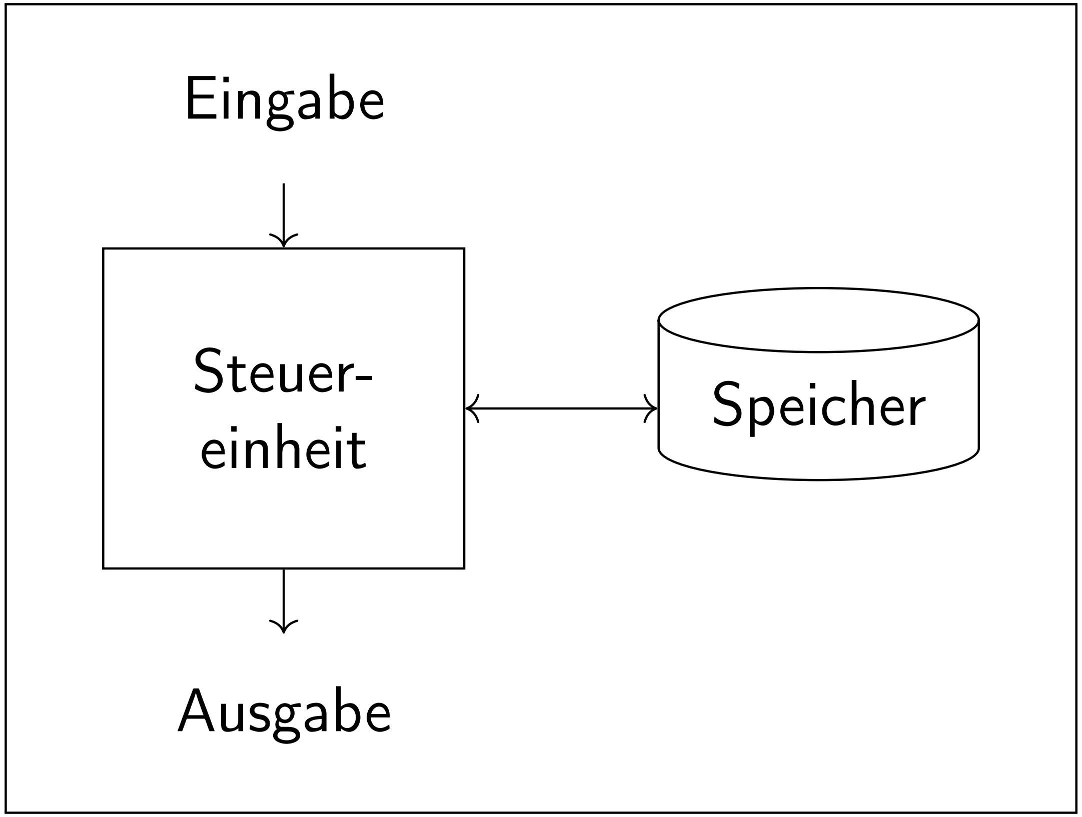
# Grundthema von AFS und BuK

### Grundfrage

Welche Probleme können von einem Algorithmus bzw. von einer Maschine mit beschränkten Ressourcen gelöst werden und welche können nicht gelöst werden?

- Was ist ein „Problem“?
- Was meinen wir mit „Maschine“?
- Welche „Ressourcen“ betrachten wir und wie werden diese „beschränkt“?

# Maschinen




Steuereinheit:

- endlich viele Zustände
- ändern sich in Abhängigkeit von Eingabe und Speicherkonfiguration
- produziert Ausgaben
- gibt „Speicherbefehle“

Eingabe: Folge von „Buchstaben“

Ausgabe: gut/schlecht, Folge von „Buchstaben“ oder $\mathbb{N}$

# Probleme

Wir betrachten Probleme als Abbildungen

$$\begin{array}{llcc} f: & \text{Menge der mgl. Eingaben} & \longrightarrow & \text{Menge der mgl. Ausgaben} \\ & E & \longrightarrow & A \end{array}$$

$E$ = $\underline{\text{alle „Wörter über Eingabealphabet“}}$ oder Tupel natürlicher Zahlen
$A$ = $\underline{\text{gut/schlecht } (\hat{=}\{1,0\})}$, alle „Wörter über Ausgabealphabet“, oder natürliche Zahlen

Spezialfall $A = \{0,1\}$
Solche Probleme $f$ heißen Entscheidungsprobleme.
Sie sind gegeben durch die Menge $\{w \in E \mid f(w) = 1\} \subseteq E$. Solche Mengen nennen wir „Sprachen“.

# beschränkte Ressourcen

- Art des Speicherzugriffs:
  verboten / Kellerspeicher / Register / Turing-Band
- Art der Steuereinheit: deterministisch / nichtdeterministisch
- Dauer der Berechnung: polynomiell / exponentiell / ...
- Größe des Speichers: fest / logarithmisch / polynomiell / ...

# Ressourcen für AFS

- 5 Leistungspunkte ergeben 150 h Arbeitsaufwand
(§4 PStO-AB Bachelor, Master und Diplom: „Ein Leistungspunkt entspricht … einer Arbeitszeit von 30 Stunden.“)
- das Semester hat 15 Wochen, also 10 h / Woche
- eine Vorlesung, $\frac{1}{2}$ Übung: $2\frac{1}{4}$ h / Woche Präsenzveranstaltungen
- es bleiben also

$$7\frac{3}{4} \text{ h / Woche für das Selbststudium}$$

= Vorlesungs- und Übungsvor- und -nachbereitung

# Organisatorisches zur Vorlesung

**Informationen**, **aktuelle Version der Folien**, **Übungsblätter** und **Mitschnitte der Vorlesungen**: Website der Veranstaltung (nicht moodle)

Damit wir Informationen an Sie schicken können, melden Sie sich bitte trotzdem im moodle-Kurs an (Passwort „abcd“).

**Prüfung:** 90-minütige Klausur im Februar/März 2023.

**Literaturempfehlung:** Uwe Schöning: Theoretische Informatik – kurz gefasst, Spektrum Akademischer Verlag

Die **Übungen** führt Herr M.Sc. Hugenroth durch.

**Bonuspunkte**: Die Lösungen der Übungsaufgaben werden abgegeben, (teilweise) korrigiert und sie ergeben (anteilig) 10% der Klausurpunkte.

**Abgabe der Übungsaufgaben**: bis Freitag 11 Uhr (ungerade Woche) in Briefkasten vor Z 1047 oder in Freitagsübung

Darüber hinaus führt er eine regelmäßige **Konsultation** durch, bei der Sie Fragen zum Vorstellungsstoff stellen können (Mo G 9 Uhr, OEC 5007).

# Arbeitsweise

1. Sie kommen natürlich zu jeder Vorlesung und hören aktiv zu.
2. Der anspruchsvolle Stoff kann nicht durch alleiniges Hören verstanden werden.
3. Daher werden Sie den Vorlesungsstoff semesterbegleitend nacharbeiten: Definitionen („Konzepte“) und Sätze („Sachverhalte“) herausschreiben und auswendig lernen, Beweise („Begründungen“) verstehen (= wiedergeben können), weitere Literatur zu Rate ziehen
4. Sie drucken die Übungsblätter frühzeitig aus, lesen sie genau, arbeiten die Lösung der Hausaufgaben aus und geben diese ab. Diese Lösungen und dabei auftretende Probleme werden in der Übung besprochen.
5. Auch Übungen werden semesterbegleitend nachgearbeitet.
6. Zu jeder Veranstaltung bringen Sie sämtliche Unterlagen zum Nachschlagen mit.
7. Bei Verständnisproblemen fragen Sie bitte frühzeitig!

1 Einführung

2 Grundbegriffe

3 Rechtslineare Sprachen

4 Kontextfreie Sprachen

natürliche Zahlen: $\mathbb{N} = \{0, 1, 2, 3, \dots\}$

## Definition

Sei $X$ eine Menge.

- $X^* = \bigcup_{n \in \mathbb{N}} X^n$ ist die Menge der Tupel beliebiger Länge über $X$.
- $X^+ = \bigcup_{n \geq 1} X^n = X^* \setminus \{()\}$ ist die Menge der nichtleeren Tupel über $X$.

## Beispiele

Elemente von $\mathbb{N}^*$: $(2, 3, 2, 2, 1, 6), (4, 3, 0, 1), (3, 0, 5), (3, 3), (1), ()$
Elemente von $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}^*$: $(\mathtt{a}, \mathtt{b}, \mathtt{a})$, $()$
Elemente von $\{\mathtt{A}, \dots, \mathtt{Z}, \mathtt{0}, \dots, \mathtt{9}\}^*$: $(\mathtt{A}, \mathtt{F}, \mathtt{S}, \mathtt{2}, \mathtt{0}, \mathtt{2}, \mathtt{3})$, $()$

**Bemerkungen:**

- $X^* \neq X^{\mathbb{N}}$, denn dies ist die Menge der unendlichen Folgen über $X$.
- $() \in X^*$ für alle $X$ (insbes. auch für $X = \emptyset$)

## Definition

Sei $X$ Menge und $w = (x_1, \dots, x_n) \in X^*$.

- $|w| = n$ ist die Länge von $w$.
- Für $x \in X$ ist $|w|_x$ die Anzahl der Vorkommen von $x$ in $w$, d.h. $|w|_x = \big|\{i \mid 1 \le i \le n, x_i = x\}\big|$.

## Bemerkung

Es gelten also

- $X^n = \big\{w \in X^* \bigm| |w| = n\big\}$,
- $|()| = 0$ und
- $|w| = \sum_{x \in X} |w|_x$ (falls $X$ eine endliche Menge ist).

## Definition

Ein Alphabet ist eine endliche nichtleere Menge.

üblicherweise heißen unsere Alphabete $\Sigma$, $\Gamma$, $\Delta$

## Beispiele

- Alphabete: $\{0\}$, $\{0, 1, 2\}$, ...
  $\{\mathtt{A}, \dots, \mathtt{Z}, \mathtt{0}, \dots, \mathtt{9}\}$
  $\{\mathtt{groß}, \mathtt{klein}\}$
- keine Alphabete: $\emptyset$, $\mathbb{N}$, $\mathbb{Q}$

Ist $\Sigma$ Alphabet, so nennen wir die Elemente oft Buchstaben.

## Beispiel

Das Alphabet $\{0, 1, 2\}$ hat also die drei Buchstaben 0, 1 und 2.
Das Alphabet $\{\mathtt{groß}, \mathtt{klein}\}$ hat die zwei Buchstaben `groß` und `klein`

Ist $\Sigma$ ein Alphabet, so heißen die Elemente von $\Sigma^*$ auch Wörter über $\Sigma$ (auch: String, Zeichenkette).

## Beispiel

- $(0)$, $(\,)$ und $(1, 2, 0, 0)$ sind also Wörter über dem Alphabet $\{0, 1, 2\}$.
- $(\texttt{groß})$, $(\texttt{klein}, \texttt{groß})$, $(\texttt{klein}, \texttt{groß}, \texttt{klein})$ und $(\,)$ sind Wörter über dem Alphabet $\{\texttt{groß}, \texttt{klein}\}$.

**Schreibweise:** Für das Wort $(a_0, a_1, a_2, \ldots, a_n)$ schreiben wir auch $a_0\, a_1\, a_2\, \ldots\, a_n$, typische Wörter: $u, v, w, x, y, z, w_0, w_1, \ldots$

## Beispiel

$$\begin{aligned} (1, 2, 0, 0) \text{ wird geschrieben als} \quad & 1\,2\,0\,0 \\ (1) \text{ wird geschrieben als} \quad & 1 \\ (\,) \text{ wird geschrieben als} \quad & \varepsilon \text{ (das leere Wort)} \end{aligned}$$

## Definition

Sind $u = (a_1, a_2, a_3, \dots, a_m)$ und $v = (b_1, b_2, b_3, \dots, b_n)$ Wörter, so ist $u \cdot v$ das Wort $(a_1, a_2, a_3, \dots, a_m, b_1, b_2, b_3, \dots, b_n)$; es wird als Verkettung oder Konkatenation von $u$ und $v$ bezeichnet.

**Schreibweise:** An Stelle von $u \cdot v$ schreibt man auch $uv$.

## Beobachtung

$\cdot : \Sigma^* \times \Sigma^* \to \Sigma^*$ ist eine Abbildung (oder eine zweistellige Operation auf $\Sigma^*$) mit

- $(u \cdot v) \cdot w = u \cdot (v \cdot w)$ („Assoziativität“ - Klammern unnötig)
- $\varepsilon \cdot u = u \cdot \varepsilon = u$ („neutrales Element“- in Produkten unnötig).

Kürzer: $(\Sigma^*, \cdot, \varepsilon)$ ist ein Monoid.

$(\mathbb{N}, \cdot, 1)$ ist auch ein Monoid.

## Definition

Für $w \in \Sigma^*$ und $n \in \mathbb{N}$ ist $w^n$ induktiv definiert:

$$w^n = \begin{cases} \varepsilon & \text{falls } n = 0 \\ w \cdot w^{n-1} & \text{falls } n > 0 \end{cases}$$

## Beispiel

$$\begin{aligned} 001100110011 &= 0011 \cdot 0011 \cdot 0011 = \left(0^2\,1^2\right)^3 \\ \texttt{abacabacabacabac} &= \texttt{abac} \cdot \texttt{abac} \cdot \texttt{abac} \cdot \texttt{abac} \\ &= (\texttt{abac})^4 \\ \texttt{abacccc} &= \texttt{aba} \cdot \texttt{cccc} \\ &= \texttt{abac}^4 \end{aligned}$$

d.h. die Potenzierung bindet stärker als die Verkettung.

## Definition

Seien $y, w$ Wörter über $\Sigma$. Dann heißt $y$

- Präfix/Anfangsstück von $w$, wenn es $z \in \Sigma^*$ gibt mit $yz = w$
- Infix/Faktor von $w$, wenn es $x, z \in \Sigma^*$ gibt mit $xyz = w$
- Suffix/Endstück von $w$, wenn es $x \in \Sigma^*$ gibt mit $xy = w$.

## Beispiel

- *aa* ist Präfix von *aabba* (und daher auch Faktor), aber kein Suffix
- *aa* ist kein Präfix, aber Suffix und Faktor von *abbaa*
- *bb* ist Faktor, aber weder Präfix noch Suffix von *aabba*
- *aa* ist Präfix, Faktor und Suffix von *aabbaa*
- Für jedes Wort $w$ sind $\varepsilon$ und $w$ Präfix, Faktor und Suffix von $w$

## Definition

- Sei $\Sigma$ ein Alphabet.
  Teilmengen von $\Sigma^*$ werden formale Sprachen über $\Sigma$ genannt.
- Eine Menge $L$ ist eine formale Sprache, wenn es ein Alphabet $\Sigma$ gibt, so daß $L$ formale Sprache über $\Sigma$ ist (d.h. so daß $L \subseteq \Sigma^*$).

## Beispiel

1. Die Menge $P_{10}$ der Dezimaldarstellungen von Primzahlen ist eine formale Sprache: das Alphabet ist $\{0, 1, 2, \ldots, 9\}$
2. Die Menge $P_2$ der Binärdarstellungen von Primzahlen ist eine formale Sprache: das Alphabet ist $\{0, 1\}$.
   Wegen $10 \in P_2 \setminus P_{10}$ gilt $P_2 \neq P_{10}$.
3. Die Menge $P$ der Primzahlen ist keine formale Sprache, denn was sollte das Alphabet sein?

- Die Menge der syntaktisch korrekten deutschen Sätze - wirklich?
  *Nun schon wieder*
  *Den eratmenden Schritt*
  *Mühsam Berg hinauf!*
  *Auf denn, nicht träge denn!*
  *Strebend und hoffend hinan!*

  (Aus: J.W. v. Goethe, „An Schwager Kronos“)

Es ist unklar, ob es sich hier um korrekte deutsche Sätze handelt. Die „Menge aller grammatisch korrekten Sätze der deutschen Sprache“ ist gar keine Menge, da nicht eindeutig feststeht, ob eine Zeichenreihe dazugehört oder nicht. Also ist „sie“ auch keine formale Sprache.

### Verabredung

Ab jetzt sagen wir oft einfach Sprache für „formale Sprache“.

## Definition

Sind $L_1$ und $L_2$ Sprachen, so heißt die Sprache

$$L_1 L_2 = \{w \mid \exists w_1 \in L_1, w_2 \in L_2 \colon w = w_1 w_2\} \qquad (\text{auch } L_1 \cdot L_2)$$

die Konkatenation oder Verkettung von $L_1$ und $L_2$.

## Beispiel

- $\{0\}^*\{1\}^* = \{0^i 1^j \mid i, j \geq 0\}$,
- $\{0\} \cup \{1\}\{0,1\}^*$ ist die Menge der Binärzahlen

## Beobachtungen

- Die Verkettung von Sprachen ist assoziativ: $(L_1\, L_2)\, L_3 = L_1\, (L_2\, L_3)$
- $\{\varepsilon\}$ ist neutrales Element: $L\,\{\varepsilon\} = \{\varepsilon\}\, L = L$
- $\emptyset$ ist auslöschendes Element: $\emptyset\, L = L\, \emptyset = \emptyset$.

## Definition

Sei $L$ Sprache und $n \in \mathbb{N}$. Dann ist $L^n$ induktiv definiert:

$$L^n = \begin{cases} \{\varepsilon\} & \text{falls } n = 0 \\ L\,L^{n-1} & \text{falls } n > 0 \end{cases}$$

Es gilt also $L^n := \{w_1 w_2 \cdots w_n \mid w_1, \ldots, w_n \in L\}$.

**Beachte:** $\{w^n \mid w \in L\} \subseteq L^n$, meist keine Gleichheit!

## Beispiel

- für $L = \{01, 0101\}$: $01\,0101 \in L^2$, aber $01\,0101 \notin \{0101\,0101, 01\,01\} = \{w^2 \mid w \in L\}$
- aber für $K = \{01\}$: $K^2 = \{01\,01\} = \{w^2 \mid w \in K\}$

## Definition

Sei $L$ eine Sprache. Dann ist $L^* = \bigcup_{n\geq 0} L^n$ der Kleene-Abschluß oder die Kleene-Iteration von $L$.
Weiter ist $L^+ = \bigcup_{n\geq 1} L^n$ die positive Iteration von $L$.

Selbststudium: $L^+ = L \cdot L^* = L^* \cdot L$

**Beachte:** $\{w^n \mid w \in L, n \geq 0\} \subseteq L^*$, meist keine Gleichheit!

## Beispiel

- für $L = \{01, 0101\}$: $$\begin{aligned} L^* &= \{\varepsilon, 01, 0101, 01\,0101, \dots\} \\ &= \left\{(01)^n \mid n \geq 0\right\} \\ &= \{w^n \mid w \in L, n \geq 0\} \end{aligned}$$
- aber für $K = \{0, 11\}$: $0\,11 \in K^* \setminus \{w^n \mid w \in K, n \geq 0\}$

## Beobachtung

Sei $\Sigma$ Alphabet.

- Sind $L_1$ und $L_2$ Sprachen über $\Sigma$, so auch
  - die Verkettung $L_1 L_2$,
  - die Kleene-Iteration $L_1^*$,
  - die positive Iteration $L_1^+$,
  - die Vereinigung $L_1 \cup L_2$,
  - die Differenz $L_1 \setminus L_2$
  - und der Schnitt $L_1 \cap L_2$.
- $\emptyset$, $\Sigma$ und $\Sigma^*$ sind Sprachen über $\Sigma$.

**Prioritätsregeln** für Operationen auf Sprachen:

- Potenz/Iteration binden stärker als Konkatenation
- Konkatenation stärker als Vereinigung/Durchschnitt/Differenz.

**Sprechweise**: „Klasse“ von Sprachen (nicht „Menge“).

### Beispiel

Die „Klasse aller Sprachen“,
die „Klasse der unendlichen Sprachen“,
die „Klasse der Sprachen über einem einelementigen Alphabet“,
die „Klasse der regulären Sprachen“, usw.

(Diese Gesamtheiten sind keine Mengen im strengen Sinne.)

# Zusammenfassung 1. Vorlesung

## in dieser Vorlesung neu

- Grundbegriffe: Alphabet, (formale) Sprache
- Operationen auf Sprachen

## kommende Vorlesung

- grundlegende Methode zur Beschreibung formaler Sprachen
- Chomsky-Hierarchie