1a

Begrüßung, Organisatorisches, Inhaltsübersicht und Grundlagen

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 18. April 2024
Basiert auf Folien von PD Dr. David Sabel

Personen

Dozent

- Prof. Dr. Jasmin Blanchette
  jasmin.blanchette@ifi.lmu.de

Wissenschaftliche Mitarbeiter

- Jannis Limperg
  jannis.limperg@ifi.lmu.de
- Luca Maio
  luca.maio@ifi.lmu.de

Tutorinnen und Tutoren

- Julia Doktor, Luis Gambarte, Massin Guerdi, Paul Köhler, Elisabeth Lempa, Jannis Limperg, Luca Maio und Korbinian Meier

Korrektorinnen und Korrektoren

- Karl Deck, Julia Doktor, Jakob Friedrich, Paul Köhler, Matthias Königer, Magdalena Mansfeld, Korbinian Meier und Zhiyang Song

Zielgruppe der Veranstaltung

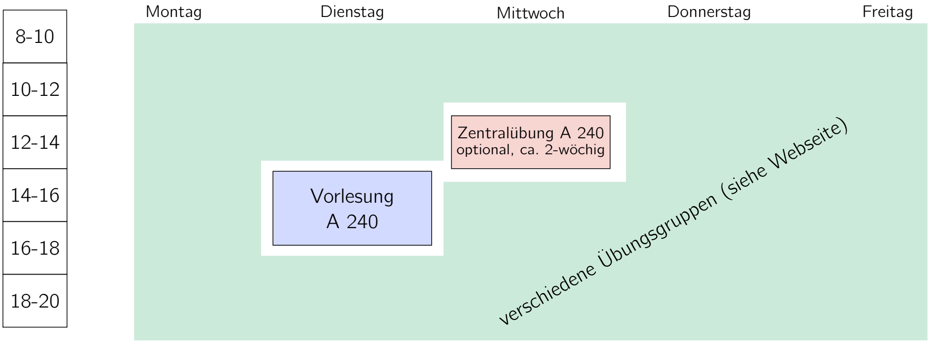
Formale Sprachen und Komplexität (FSK)

- Studierende der Informatik
- Studierende der Bioinformatik
- Studierende im Lehramt
- Studierende im Nebenfach Informatik

Theoretische Informatik für Studierende der Medieninformatik (TIMI)

- Studierende der Medieninformatik

Struktur der Veranstaltung



- **Vorlesung:** FSK 3V, TIMI 2V (integriert, Plan auf Webseite)
- **Zentralübung:** Zusatzangebot, Fragestunde und Beispiele (Plan auf Webseite)
- **Übungen:** in Präsenz; Besprechung der Hausaufgaben; FSK: 2Ü, TIMI: 1Ü

Internetauftritt

Webseiten

www.tcs.ifi.lmu.de/lehre/ss-2024/fsk_de.html (FSK)
www.tcs.ifi.lmu.de/lehre/ss-2024/timi_de.html (TIMI)

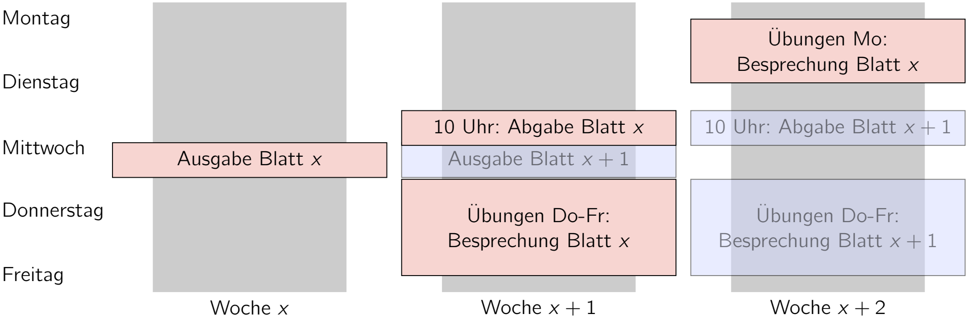
Moodle

moodle.lmu.de/course/view.php?id=32444 (FSK)
moodle.lmu.de/course/view.php?id=32603 (TIMI)
Anmeldung ist notwendig für Abgabe und Korrektur der Hausaufgaben.

Zulip-Chat

Server-Adresse: chat.ifi.lmu.de
Stream: TCS-24S-FSK-TIMI
Fragen und Kommentare am besten dort stellen.

Hausaufgaben



- Die erste Übung ist zum Kennenlernen und zur Besprechung von Übungsblatt 0 (ohne Abgabe) da.
- Sie wählen Ihre Übungsgruppe selbst.
- Abgabe und Korrektur der Übungsblätter erfolgt elektronisch über Moodle.

Korrektur

- Ausgewählte Hausaufgaben werden bepunktet.
- Für jede Lösung zu einer bepunkteten Aufgabe gibt es 0 oder 1 oder 2 Punkte.
- Diese Punkte dienen als Feedback und zählen **nicht** zu einem Notenbonus.

Klausuren

- Die Bearbeitungszeit der Präsenzklausur beträgt 120 Minuten.
- Anmeldung zur Prüfung wird noch freigeschaltet.
- Teilnahme an der Nachholklausur ist auch ohne Teilnahme an der Erstklausur möglich.
- Die Erstklausur wird am 5. August 2024 ab 15:00 Uhr stattfinden.
- Das Datum der Nachholklausur ist noch nicht bekannt.

Material

Auf der Webseite verfügbar

- Vorlesungsfolien
- Vorlesungsskript
  - Die nicht TIMI-relevanten Teile sind mit $\star$ markiert.
  - Kapitel 2 (Grundlagen) ist zum Teil dem Selbststudium überlassen.
- Übungsblätter

Lehrbücher

Wesentliche Quelle

- Uwe Schöning: Theoretische Informatik – kurz gefasst, 5. Auflage, Spektrum Akademischer Verlag, 2008
  Zum Teil zu kurz gefasst

Weitere Literatur

- Alexander Asteroth und Christel Baier: Theoretische Informatik, Pearson, 2002.
  Aufbau in anderer Reihenfolge, Zugriff über UB
- John E. Hopcroft, Rajeev Motwani und Jeffrey D. Ullman: Introduction to Automata Theory, Languages, and Computation, 3. Auflage, Pearson, 2006
  Der Klassiker, umfangreich, Erstauflage 1979
- Ingo Wegener: Theoretische Informatik – eine algorithmenorientierte Einführung, 3. Auflage, Teubner Verlag, 2005.
  Algorithmen stehen im Vordergrund, Zugriff über UB

Ziel der Veranstaltung

Ziel ist die Vermittlung von:

Theorie

- Die Theorie sagt uns, was Computer (wie schnell) können und was nicht.
- Viele Konzepte haben praktische Anwendungen.
- Die Theorie an sich ist zum Teil sehr schön.

Fähigkeiten

- Sie werden lernen, mit abstrakten Konzepten umzugehen.
- Sie werden lernen, sorgfältig und präzise zu arbeiten.
- Sie werden Fähigkeiten zur Beweisführung entwickeln.

Inhalte der Veranstaltung

Drei große Themen der Theoretischen Informatik:

1. Formale Sprachen und Automatentheorie
   Wie stellt man Entscheidungsprobleme formal dar?
   Insbesondere: Wie kann man Programmiersprachen u.Ä. erkennen?
2. Berechenbarkeitstheorie
   Welche Probleme kann man algorithmisch
   (bzw. mit dem Computer) überhaupt lösen?
3. Komplexitätstheorie
   Welche Probleme kann man in annehmbarer Zeit lösen?

Thema: Formale Sprachen und Automatentheorie

1. Formale Sprachen und Automatentheorie
   Wie stellt man Entscheidungsprobleme formal dar?
   Insbesondere: Wie kann man Programmiersprachen u.Ä. erkennen?

Schlüsselkonzepte:

- Formale Sprachen und Entscheidungsprobleme
- Reguläre Ausdrücke (z.B. für Lexer)
- Grammatiken (z.B. für Parser)
- Automaten

Thema: Berechenbarkeitstheorie

2. Berechenbarkeitstheorie
   Welche Probleme kann man algorithmisch
   (bzw. mit dem Computer) überhaupt lösen?

Schlüsselkonzepte:

- Intuitive Berechenbarkeit
- Turingberechenbarkeit
- Imperative Programme (LOOP, WHILE, GOTO)
- Rekursive Funktionen
- Unentscheidbarkeit

Thema: Komplexitätstheorie

3. Komplexitätstheorie
   Welche Probleme kann man in annehmbarer Zeit lösen?

Schlüsselkonzepte:

- Die Klassen $\mathcal{P}$ und $\mathcal{NP}$
- $\mathcal{NP}$-Schwere, $\mathcal{NP}$-Vollständigkeit
- Konkrete $\mathcal{NP}$-vollständige Probleme

Grundlagen: Wörter

Definition

Ein Alphabet Σ ist eine endliche nicht leere Menge von Zeichen (oder Symbolen).

Z.B. $\Sigma = \{a, b, c, d, e\}$.

Grundlagen: Wörter

Definition

Ein Alphabet Σ ist eine endliche nicht leere Menge von Zeichen (oder Symbolen).

Z.B. $\Sigma = \{a, b, c, d, e\}$.

Definition

Ein Wort w über Σ ist eine endliche Folge von Zeichen aus Σ.

Beispiele:

- $bade$ ist ein Wort über $\{a, b, c, d, e\}$.
- $baden$ ist kein Wort über $\{a, b, c, d, e\}$.

Weitere Notationen zu Wörtern

- Das leere Wort wird als ε notiert.
- Für $w = a_1 \cdots a_n$ ist $|w| = n$ die Länge des Wortes.
- Für $1 \leq i \leq |w|$ ist $w[i]$ das Zeichen an der i-ten Position in w.
- Für $a \in \Sigma$ und w ein Wort über Σ sei $\#_a(w) \in \mathbb{N}$ die Anzahl an Vorkommen des Zeichens a im Wort w.

Weitere Notationen zu Wörtern

- Das leere Wort wird als ε notiert.
- Für $w = a_1 \cdots a_n$ ist $|w| = n$ die Länge des Wortes.
- Für $1 \leq i \leq |w|$ ist $w[i]$ das Zeichen an der i-ten Position in w.
- Für $a \in \Sigma$ und w ein Wort über Σ sei $\#_a(w) \in \mathbb{N}$ die Anzahl an Vorkommen des Zeichens a im Wort w.

Beispiele:

- Es gilt $|\varepsilon| = 0$ und $\#_a(\varepsilon) = 0$ für alle $a \in \Sigma$.
- Für $\Sigma = \{a, b, c\}$ ist
  - $|abbccc| = 6$
  - $|aabbbccc| = 8$
  - $\#_a(abbccc) = 1$
  - $\#_c(aabbbccc) = 3$.
- Für $w = abbbcd$ ist $w[1] = a$, $w[5] = c$ und $w[7]$ undefiniert.

Konkatenation und Kleene-Stern

Definition

Das Wort $u \cdot v$ (alternativ uv) entsteht, indem Wort v hinten an Wort u angehängt wird.

Konkatenation und Kleene-Stern

Definition

Das Wort $u \cdot v$ (alternativ uv) entsteht, indem Wort v hinten an Wort u angehängt wird.

Die Konkatenation hilft folgende Mengen von Wörtern über Σ zu definieren:

Definition

Sei Σ ein Alphabet, dann definieren wir:

$$\begin{aligned}
\Sigma^0 &:= \{\varepsilon\} \\
\Sigma^i &:= \{aw \mid a \in \Sigma, w \in \Sigma^{i-1}\} \text{ für } i > 0 \\
\Sigma^* &:= \bigcup_{i \in \mathbb{N}} \Sigma^i \\
\Sigma^+ &:= \bigcup_{i \in \mathbb{N}_{>0}} \Sigma^i
\end{aligned}$$

Beachte: $\mathbb{N} = \{0, 1, 2, \ldots\}$ und $\mathbb{N}_{>0} = \{1, 2, \ldots\}$.

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist
$\Sigma^0 = \{\varepsilon\}$,

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist
$\Sigma^0 = \{\varepsilon\}$,
$\Sigma^1 = \Sigma = \{a, b\}$,

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist
$\Sigma^0 = \{\varepsilon\}$,
$\Sigma^1 = \Sigma = \{a, b\}$,
$\Sigma^2 = \{aa, ab, ba, bb\}$,

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist
$\Sigma^0 = \{\varepsilon\}$,
$\Sigma^1 = \Sigma = \{a, b\}$,
$\Sigma^2 = \{aa, ab, ba, bb\}$,
$\Sigma^3 = \{xw \mid x \in \{a, b\}, w \in \Sigma^2\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

Beispiele für Konkatenation und Kleene-Stern

Sei $\Sigma = \{a, b\}$.

Dann ist
$\Sigma^0 = \{\varepsilon\}$,
$\Sigma^1 = \Sigma = \{a, b\}$,
$\Sigma^2 = \{aa, ab, ba, bb\}$,
$\Sigma^3 = \{xw \mid x \in \{a, b\}, w \in \Sigma^2\} = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$

$\vdots$

und
$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, aba, abb, baa, bab, bba, bbb, aaaa, \ldots\}$.

Weitere Notationen und Begriffe

Sei w ein Wort über Σ.

- w^m entsteht aus m-maligen Konkatenieren von w, d.h.

$$w^0 = \varepsilon \text{ und } w^m = ww^{m-1} \text{ für } m > 0$$

- $\overline{w}$ ist das rückwärts gelesene Wort w, d.h.

$$\overline{\varepsilon} = \varepsilon \text{ und für } w = a_1 \cdots a_n \text{ ist } \overline{w} = a_n \cdots a_1$$

- w ist ein Palindrom g.d.w. $w = \overline{w}$.
  Beispiele für Palindrome: anna, reliefpfeiler, lagerregal, annasusanna.

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.

$$\underbrace{u \mid w}_{v}$$

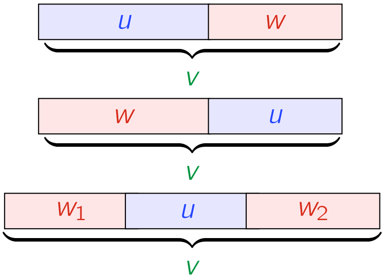
$$\underbrace{w \mid u}_{v}$$

$$\underbrace{w_1 \mid u \mid w_2}_{v}$$

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.

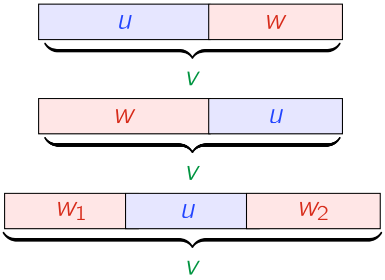


Beispiel: Sei $w = ababbaba$.

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.



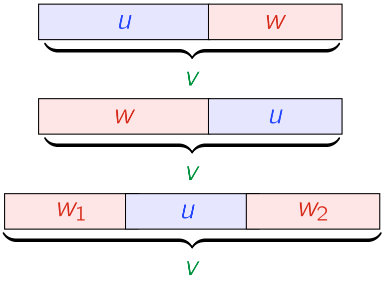
Beispiel: Sei $w = ababbaba$.

- w ist ein Präfix, Suffix und Teilwort von w.

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.



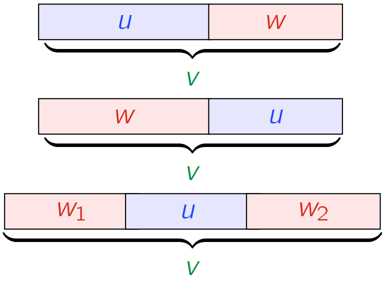
Beispiel: Sei $w = ababbaba$.

- w ist ein Präfix, Suffix und Teilwort von w.
- aba ist ein Präfix, Suffix und Teilwort von w.

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.



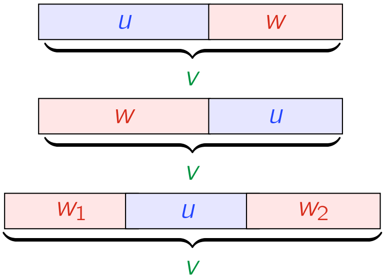
Beispiel: Sei $w = ababbaba$.

- w ist ein Präfix, Suffix und Teilwort von w.
- aba ist ein Präfix, Suffix und Teilwort von w.
- $ababb$ ist ein Präfix und Teilwort von w, aber kein Suffix von w.

Sprechweisen: Präfix, Suffix, Teilwort

Seien u, v Wörter über einem Alphabet Σ.

- u ist ein Präfix von v, wenn es ein Wort w gibt mit $uw = v$.
- u ist ein Suffix von v, wenn es ein Wort w gibt mit $wu = v$.
- u ist ein Teilwort von v, wenn es Wörter w_1, w_2 gibt mit $w_1 u w_2 = v$.



Beispiel: Sei $w = ababbaba$.

- w ist ein Präfix, Suffix und Teilwort von w.
- aba ist ein Präfix, Suffix und Teilwort von w.
- $ababb$ ist ein Präfix und Teilwort von w, aber kein Suffix von w.
- bab ist ein Teilwort von w, aber weder ein Präfix noch ein Suffix.

Grundlagen: Formale Sprache

Definition

Eine (formale) Sprache L über dem Alphabet Σ ist eine Teilmenge von Σ^*, d.h. $L \subseteq \Sigma^*$.

Beachte: L steht für „language“.

Grundlagen: Formale Sprache

Definition

Eine (formale) Sprache L über dem Alphabet Σ ist eine Teilmenge von Σ^*, d.h. $L \subseteq \Sigma^*$.

Beachte: L steht für „language“.

Definition

Seien L, L_1, L_2 formale Sprachen über Σ.

- Vereinigung: $L_1 \cup L_2 := \{w \mid w \in L_1 \text{ oder } w \in L_2\}$
- Schnitt: $L_1 \cap L_2 := \{w \mid w \in L_1 \text{ und } w \in L_2\}$
- Komplement zu L: $\overline{L} := \Sigma^* \setminus L$
- Produkt: $L_1 \cdot L_2 = L_1 L_2 = \{uv \mid u \in L_1 \text{ und } v \in L_2\}$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$\begin{aligned} L_1 \cup L_2 &= ? \\ L_1 \cap L_2 &= ? \\ \overline{L_1} &= ? \\ L_1 L_2 &= ? \\ L_2 L_1 &= ? \\ L_1 L_1 &= ? \end{aligned}$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$\begin{aligned}
L_1 \cup L_2 &= \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen} \\
L_1 \cap L_2 &= \text{?} \\
\overline{L_1} &= \text{?} \\
L_1 L_2 &= \text{?} \\
L_2 L_1 &= \text{?} \\
L_1 L_1 &= \text{?}
\end{aligned}$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$
\begin{aligned}
L_1 \cup L_2 &= \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen} \\
L_1 \cap L_2 &= \{\varepsilon\} \neq \emptyset \\
\overline{L_1} &= ? \\
L_1 L_2 &= ? \\
L_2 L_1 &= ? \\
L_1 L_1 &= ?
\end{aligned}
$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$L_1 \cup L_2 = \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen}$$

$$L_1 \cap L_2 = \{\varepsilon\} \neq \emptyset$$

$$\overline{L_1} = \text{Sprache der Wörter, die mindestens ein } b \text{ enthalten}$$

$$L_1 L_2 = ?$$

$$L_2 L_1 = ?$$

$$L_1 L_1 = ?$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$
\begin{aligned}
L_1 \cup L_2 &= \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen} \\
L_1 \cap L_2 &= \{\varepsilon\} \neq \emptyset \\
\overline{L_1} &= \text{Sprache der Wörter, die mindestens ein } b \text{ enthalten} \\
L_1 L_2 &= \{a^i b^j \mid i, j \in \mathbb{N}\} \\
L_2 L_1 &= ? \\
L_1 L_1 &= ?
\end{aligned}
$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$\begin{aligned}
L_1 \cup L_2 &= \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen} \\
L_1 \cap L_2 &= \{\varepsilon\} \neq \emptyset \\
\overline{L_1} &= \text{Sprache der Wörter, die mindestens ein } b \text{ enthalten} \\
L_1 L_2 &= \{a^i b^j \mid i, j \in \mathbb{N}\} \\
L_2 L_1 &= \{b^i a^j \mid i, j \in \mathbb{N}\} \\
L_1 L_1 &= \, ?
\end{aligned}$$

Beispiele für Operationen auf formalen Sprachen

Seien $\Sigma = \{a, b\}$, $L_1 = \{a^i \mid i \in \mathbb{N}\}$ und $L_2 = \{b^i \mid i \in \mathbb{N}\}$.

$$L_1 \cup L_2 = \text{Sprache der Wörter, die nur aus } a\text{'s oder nur aus } b\text{'s bestehen}$$
$$L_1 \cap L_2 = \{\varepsilon\} \neq \emptyset$$
$$\overline{L_1} = \text{Sprache der Wörter, die mindestens ein } b \text{ enthalten}$$
$$L_1 L_2 = \{a^i b^j \mid i, j \in \mathbb{N}\}$$
$$L_2 L_1 = \{b^i a^j \mid i, j \in \mathbb{N}\}$$
$$L_1 L_1 = L_1$$

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$

$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$
$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Beispiel: Sei $L = \{ab, ac\}$.

$L^0 = ?$

$L^1 = ?$

$L^2 = ?$

$L^3 = ?$

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad\qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$
$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad\qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Beispiel: Sei $L = \{ab, ac\}$.

$$L^0 = \{\varepsilon\}$$
$$L^1 = ?$$
$$L^2 = ?$$
$$L^3 = ?$$

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$
$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Beispiel: Sei $L = \{ab, ac\}$.

$$L^0 = \{\varepsilon\}$$
$$L^1 = L \cdot L^0 = L = \{ab, ac\}$$
$$L^2 = ?$$
$$L^3 = ?$$

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$
$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Beispiel: Sei $L = \{ab, ac\}$.

$$L^0 = \{\varepsilon\}$$
$$L^1 = L \cdot L^0 = L = \{ab, ac\}$$
$$L^2 = L \cdot L^1 = \{abab, abac, acab, acac\}$$
$$L^3 = ?$$

Kleenescher Abschluss

Sei L eine Sprache. Dann ist:

$$L^0 := \{\varepsilon\} \qquad L^* := \bigcup_{i \in \mathbb{N}} L^i$$
$$L^i := L \cdot L^{i-1} \text{ für } i > 0 \qquad L^+ := \bigcup_{i \in \mathbb{N}_{>0}} L^i$$

Die Sprache L^* nennt man auch den Kleeneschen Abschluss von L, benannt nach Stephen Cole Kleene.

Beispiel: Sei $L = \{ab, ac\}$.

$$L^0 = \{\varepsilon\}$$
$$L^1 = L \cdot L^0 = L = \{ab, ac\}$$
$$L^2 = L \cdot L^1 = \{abab, abac, acab, acac\}$$
$$L^3 = L \cdot L^2 = \{ababab, ababac, abacab, abacac, acabab, acabac, acacab, acacac\}$$

Weitere Beispiele für Operationen auf formalen Sprachen

$$((\{\varepsilon, 1\} \cdot \{0, \dots, 9\}) \cup (\{2\} \cdot \{0, 1, 2, 3\})) \cdot \{:\} \cdot \{0, 1, 2, 3, 4, 5\} \cdot \{0, \dots, 9\}$$

Beschriebene Sprache = ?

$$\{0\} \cup (\{1, \dots, 9\} \cdot \{0, \dots, 9\}^*)$$

Beschriebene Sprache = ?

Weitere Beispiele für Operationen auf formalen Sprachen

$$((\{\varepsilon, 1\} \cdot \{0, \ldots, 9\}) \cup (\{2\} \cdot \{0, 1, 2, 3\})) \cdot \{:\} \cdot \{0, 1, 2, 3, 4, 5\} \cdot \{0, \ldots, 9\}$$

Beschriebene Sprache = Sprache aller gültigen Uhrzeiten

$$\{0\} \cup (\{1, \ldots, 9\} \cdot \{0, \ldots, 9\}^*)$$

Beschriebene Sprache = ?

Weitere Beispiele für Operationen auf formalen Sprachen

$$((\{\varepsilon, 1\} \cdot \{0, \ldots, 9\}) \cup (\{2\} \cdot \{0, 1, 2, 3\})) \cdot \{:\} \cdot \{0, 1, 2, 3, 4, 5\} \cdot \{0, \ldots, 9\}$$

Beschriebene Sprache = Sprache aller gültigen Uhrzeiten

$$\{0\} \cup (\{1, \ldots, 9\} \cdot \{0, \ldots, 9\}^*)$$

Beschriebene Sprache = Sprache aller natürlichen Zahlen