Teil II

Grundlagen und Notationen

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen .
2. Ein Alphabet Σ ist eine endliche, nicht-leere Menge von Buchstaben

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen .
2. Ein Alphabet Σ ist eine endliche, nicht-leere Menge von Buchstaben
3. $\Sigma^n := \{a_1 a_2 \ldots a_n \mid a_i \in \Sigma\}$ ist die Menge aller Wörter der Länge n über Σ

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen .
2. Ein Alphabet Σ ist eine endliche, nicht-leere Menge von Buchstaben
3. $\Sigma^n := \{a_1 a_2 \ldots a_n \mid a_i \in \Sigma\}$ ist die Menge aller Wörter der Länge n über Σ
4. ε ist das leere Wort, $\Sigma^0 = \{\varepsilon\}$

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen .
2. Ein Alphabet Σ ist eine endliche, nicht-leere Menge von Buchstaben
3. $\Sigma^n := \{a_1 a_2 \ldots a_n \mid a_i \in \Sigma\}$ ist die Menge aller Wörter der Länge n über Σ
4. ε ist das leere Wort, $\Sigma^0 = \{\varepsilon\}$
5. $\Sigma^* := \bigcup_{n \in \mathbb{N}} \Sigma^n$ ist die Menge aller Wörter über Σ
   $\Sigma^+ := \bigcup_{n \in \mathbb{N}_{>0}} \Sigma^n$ ist die Menge aller nicht leeren Wörter über Σ

Alphabete, Wörter, Sprachen

1. $\mathbb{N} := \{0, 1, 2, 3 \ldots\}$ ist die Menge der natürlichen Zahlen.
   $\mathbb{N}_{>0} := \{1, 2, 3 \ldots\}$ ist die Menge der positiven natürlichen Zahlen .
2. Ein Alphabet Σ ist eine endliche, nicht-leere Menge von Buchstaben
3. $\Sigma^n := \{a_1 a_2 \ldots a_n \mid a_i \in \Sigma\}$ ist die Menge aller Wörter der Länge n über Σ
4. ε ist das leere Wort, $\Sigma^0 = \{\varepsilon\}$
5. $\Sigma^* := \bigcup_{n \in \mathbb{N}} \Sigma^n$ ist die Menge aller Wörter über Σ
   $\Sigma^+ := \bigcup_{n \in \mathbb{N}_{>0}} \Sigma^n$ ist die Menge aller nicht leeren Wörter über Σ

Eine (formale) **Sprache** (über Σ) ist eine Teilmenge von Σ^*.

Also: Eine Sprache ist eine Menge von Wörtern.

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

1. **C++** ist die Menge aller syntaktisch richtig aufgebauten C++ Programme. Das Alphabet ist die Menge aller **ASCII-Symbole**.

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

1. **C++** ist die Menge aller syntaktisch richtig aufgebauten C++ Programme. Das Alphabet ist die Menge aller **ASCII-Symbole**.
2. die Menge der im **Duden** aufgeführten Wörter über dem Alphabet {**a,A,b,B,. . . ,z,Z,ä,Ä,ö,Ö,ü,Ü,ß,-**}

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

1. **C++** ist die Menge aller syntaktisch richtig aufgebauten C++ Programme. Das Alphabet ist die Menge aller **ASCII-Symbole**.
2. die Menge der im **Duden** aufgeführten Wörter über dem Alphabet $\{$**a,A,b,B,. . . ,z,Z,ä,Ä,ö,Ö,ü,Ü,ß,-**$\}$
3. die Menge aller Sätze in der natürlichen Sprache **Deutsch**, das Alphabet ist die **Menge aller deutschen Wörter**

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

1. **C++** ist die Menge aller syntaktisch richtig aufgebauten C++ Programme. Das Alphabet ist die Menge aller **ASCII-Symbole**.
2. die Menge der im **Duden** aufgeführten Wörter über dem Alphabet $\{$**a,A,b,B,. . . ,z,Z,ä,Ä,ö,Ö,ü,Ü,ß,-**$\}$
3. die Menge aller Sätze in der natürlichen Sprache **Deutsch**, das Alphabet ist die **Menge aller deutschen Wörter**
4. die Menge aller **DNA(DNS)-Sequenzen**, Alphabet ist $\{$**A,C,G,T**$\}$

Modellierung mit formale Sprachen

Wozu eignen sich formale Sprachen?

- **Wörter** eignen sich zur Modellierung von Objekten, die als **endliche** Folge von diskreten Elementen aufgefasst werden können.
- Formale **Sprachen** eigenen sich zur Modellierung von Mengen von Objekten, die sich als Wörter modellieren lassen.

1. **C++** ist die Menge aller syntaktisch richtig aufgebauten C++ Programme. Das Alphabet ist die Menge aller **ASCII-Symbole**.
2. die Menge der im **Duden** aufgeführten Wörter über dem Alphabet **{a,A,b,B,...,z,Z,ä,Ä,ö,Ö,ü,Ü,ß,-}**
3. die Menge aller Sätze in der natürlichen Sprache **Deutsch**, das Alphabet ist die **Menge aller deutschen Wörter**
4. die Menge aller **DNA(DNS)-Sequenzen**, Alphabet ist **{A,C,G,T}**
5. **XML**-Dokumente, **HTML**-Dokumente, ...

Modellierung von Sprachen

Vorsicht

Nicht jede dieser Sprachen ist eindeutig definiert.

Modellierung von Sprachen

Vorsicht

Nicht jede dieser Sprachen ist eindeutig definiert.

Fragen

- Wie definiert man Sprachen?
- Welche Formalismen eignen sich?
- Was macht einen „guten“ Formalismus aus?

Modellierung von Sprachen

Vorsicht

Nicht jede dieser Sprachen ist eindeutig definiert.

Fragen

- Wie definiert man Sprachen?
- Welche Formalismen eignen sich?
- Was macht einen „guten“ Formalismus aus?

- **Ausdrucksstärke:** Der Formalimus sollte möglichst viele verschiedene Sprachen beschreiben können.

Modellierung von Sprachen

Vorsicht

Nicht jede dieser Sprachen ist eindeutig definiert.

Fragen

- Wie definiert man Sprachen?
- Welche Formalismen eignen sich?
- Was macht einen „guten“ Formalismus aus?

- **Ausdrucksstärke:** Der Formalimus sollte möglichst viele verschiedene Sprachen beschreiben können.
- **Handhabbarkeit:** Der Formalismus soll (schnell) ausgewertet werden können; außerdem möchte man Aussagen darüber treffen können (z.B. Verifikation, Minimierung).

Modellierung von Sprachen

Vorsicht

Nicht jede dieser Sprachen ist eindeutig definiert.

Fragen

- Wie definiert man Sprachen?
- Welche Formalismen eignen sich?
- Was macht einen „guten“ Formalismus aus?

- **Ausdrucksstärke:** Der Formalimus sollte möglichst viele verschiedene Sprachen beschreiben können.
- **Handhabbarkeit:** Der Formalismus soll (schnell) ausgewertet werden können; außerdem möchte man Aussagen darüber treffen können (z.B. Verifikation, Minimierung).

Wortproblem für L

Eingabe: $w \in \Sigma^*$
Frage: Ist $w \in L$?

Einschränkungen

Beispiel C++

1. die Menge aller syntaktisch richtig aufgebauten C++ Programme
2. die Menge aller syntaktisch richtig aufgebauten C++ Programme, **die immer terminieren**

Frage

Wie schwer ist das Wortproblem für diese beiden Sprachen?

Einschränkungen

Beispiel C++

1. die Menge aller syntaktisch richtig aufgebauten C++ Programme
2. die Menge aller syntaktisch richtig aufgebauten C++ Programme, **die immer terminieren**

Frage

Wie schwer ist das Wortproblem für diese beiden Sprachen?

1. 1. Sprache: Wird von jedem C++ Compiler gelöst
2. 2. Sprache: unentscheidbar

Einschränkungen

Beispiel C++

1. die Menge aller syntaktisch richtig aufgebauten C++ Programme
2. die Menge aller syntaktisch richtig aufgebauten C++ Programme, **die immer terminieren**

Frage

Wie schwer ist das Wortproblem für diese beiden Sprachen?

1. 1. Sprache: Wird von jedem C++ Compiler gelöst
2. 2. Sprache: unentscheidbar

Grundidee

Um handhabbare Formalismen zu bekommen, muss man Ausdrucksstärke opfern.

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid \text{ex. } v \in \Sigma^* \text{ mit } uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid \text{ex. } u \in \Sigma^* \text{ mit } uv = w\}$

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid \text{ex. } v \in \Sigma^* \text{ mit } uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid \text{ex. } u \in \Sigma^* \text{ mit } uv = w\}$

Operationen auf Sprachen

- $\cup$, $\cap$, $-$, $\overline{L}$ (Mengenoperationen)

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid \text{ex. } v \in \Sigma^* \text{ mit } uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid \text{ex. } u \in \Sigma^* \text{ mit } uv = w\}$

Operationen auf Sprachen

- $\cup$, $\cap$, $-$, $\overline{L}$ (Mengenoperationen)
- $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid \text{ex. } v \in \Sigma^* \text{ mit } uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid \text{ex. } u \in \Sigma^* \text{ mit } uv = w\}$

Operationen auf Sprachen

- $\cup$, $\cap$, $-$, $\overline{L}$ (Mengenoperationen)
- $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$
- $\mathrm{prefix}(L) := \bigcup_{w \in L} \mathrm{prefix}(w)$
- $\mathrm{suffix}(L) := \bigcup_{w \in L} \mathrm{suffix}(w)$

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid \text{ex. } v \in \Sigma^* \text{ mit } uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid \text{ex. } u \in \Sigma^* \text{ mit } uv = w\}$

Operationen auf Sprachen

- $\cup$, $\cap$, $-$, $\overline{L}$ (Mengenoperationen)
- $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$
- $\mathrm{prefix}(L) := \bigcup_{w \in L} \mathrm{prefix}(w)$
- $\mathrm{suffix}(L) := \bigcup_{w \in L} \mathrm{suffix}(w)$
- für $n \in \mathbb{N}$: $L^n := \{w_1 \cdot \ldots \cdot w_n \mid w_i \in L\}$ (n-fache Konkatenation)

Operationen auf Wörtern und Sprachen

Operationen auf Wörtern

- Konkatenation: $u \cdot v := uv$
- Menge aller Präfixe: $\mathrm{prefix}(w) := \{u \in \Sigma^* \mid$ ex. $v \in \Sigma^*$ mit $uv = w\}$
- Menge aller Suffixe: $\mathrm{suffix}(w) := \{v \in \Sigma^* \mid$ ex. $u \in \Sigma^*$ mit $uv = w\}$

Operationen auf Sprachen

- $\cup$, $\cap$, $-$, $\overline{L}$ (Mengenoperationen)
- $L_1 \cdot L_2 := \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$
- $\mathrm{prefix}(L) := \bigcup_{w \in L} \mathrm{prefix}(w)$
- $\mathrm{suffix}(L) := \bigcup_{w \in L} \mathrm{suffix}(w)$
- für $n \in \mathbb{N}$: $L^n := \{w_1 \cdot \ldots \cdot w_n \mid w_i \in L\}$ (n-fache Konkatenation)
- $L^* := \bigcup_{n \in \mathbb{N}} L^n$ (Kleene-Stern)
- $L^+ := \bigcup_{n \in \mathbb{N}_{>0}} L^n$ (Kleene-Plus)

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klasse aller endlichen Sprachen

$$\mathrm{FIN}_\Sigma := \{L \subset \Sigma^* \mid L \text{ ist endlich}\},$$

$$\mathrm{FIN} := \bigcup_{\Sigma \text{ ist ein Alphabet}} \mathrm{FIN}_\Sigma \,.$$

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klasse aller endlichen Sprachen

$$\mathrm{FIN}_\Sigma := \{L \subset \Sigma^* \mid L \text{ ist endlich}\},$$

$$\mathrm{FIN} := \bigcup_{\Sigma \text{ ist ein Alphabet}} \mathrm{FIN}_\Sigma .$$

Lemma

Die Klasse FIN ist

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klasse aller endlichen Sprachen

$$\mathrm{FIN}_\Sigma := \{L \subset \Sigma^* \mid L \text{ ist endlich}\},$$

$$\mathrm{FIN} := \bigcup_{\Sigma \text{ ist ein Alphabet}} \mathrm{FIN}_\Sigma .$$

Lemma

Die Klasse FIN ist

1. abgeschlossen unter $\cup$, $\cap$, $-$, $\cdot$, prefix und suffix,

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klasse aller endlichen Sprachen

$$\mathrm{FIN}_\Sigma := \{L \subset \Sigma^* \mid L \text{ ist endlich}\},$$

$$\mathrm{FIN} := \bigcup_{\Sigma \text{ ist ein Alphabet}} \mathrm{FIN}_\Sigma\,.$$

Lemma

Die Klasse FIN ist

1. abgeschlossen unter $\cup$, $\cap$, $-$, $\cdot$, prefix und suffix,
2. für jedes $n \in \mathbb{N}$ abgeschlossen unter L^n,

Klassen und Abschlusseigenschaften

- Eine Menge von Sprachen heißt auch **Klasse**.
- Wir interessieren uns besonders für Klassen, deren Sprachen gemeinsame Eigenschaften haben.

Abgeschlossen

Sei Op eine n-stellige Operation auf Sprachen ($n \geq 1$). Eine Klasse $\mathcal{C}$ ist **abgeschlossen unter** Op wenn $\mathrm{Op}(L_1, \ldots, L_n) \in \mathcal{C}$ für alle $L_i \in \mathcal{C}$.

Klasse aller endlichen Sprachen

$$\mathrm{FIN}_\Sigma := \{L \subset \Sigma^* \mid L \text{ ist endlich}\},$$

$$\mathrm{FIN} := \bigcup_{\Sigma \text{ ist ein Alphabet}} \mathrm{FIN}_\Sigma\,.$$

Lemma

Die Klasse FIN ist

1. abgeschlossen unter $\cup$, $\cap$, $-$, $\cdot$, prefix und suffix,
2. für jedes $n \in \mathbb{N}$ abgeschlossen unter L^n,
3. nicht abgeschlossen unter $\overline{L}$, $^+$ und *

Beweis: Tafel/Übung