# Globalübung FoSAP

10. Juli 2017

Jan Dreier
Philipp Kuinke
Marcel Hark

*Kontakt*: `tcs-teaching@cs.rwth-aachen.de`

*Sprechzeiten*: Jan u. Philipp Mittwochs 10:00 - 11:00,
Marcel Dienstags 14:00 - 15:00

*Website*:
`http://tcs.rwth-aachen.de/lehre/FSAP/SS2017`

# Reguläre Ausdrücke

## Definition

Es sei $\Sigma$ ein Alphabet.

# Reguläre Ausdrücke

## Definition

Es sei $\Sigma$ ein Alphabet.

1. $\emptyset$ ist ein regulärer Ausdruck.
2. $\epsilon$ ist ein regulärer Ausdruck.
3. $a$ ist ein regulärer Ausdruck, falls $a \in \Sigma$.

# Reguläre Ausdrücke

## Definition

Es sei $\Sigma$ ein Alphabet.

1. $\emptyset$ ist ein regulärer Ausdruck.
2. $\epsilon$ ist ein regulärer Ausdruck.
3. $a$ ist ein regulärer Ausdruck, falls $a \in \Sigma$.
4. $rs$ ist ein regulärer Ausdruck, falls $r$ und $s$ reguläre Ausdrücke sind.

# Reguläre Ausdrücke

## Definition

Es sei $\Sigma$ ein Alphabet.

1. $\emptyset$ ist ein regulärer Ausdruck.
2. $\epsilon$ ist ein regulärer Ausdruck.
3. $a$ ist ein regulärer Ausdruck, falls $a \in \Sigma$.
4. $rs$ ist ein regulärer Ausdruck, falls $r$ und $s$ reguläre Ausdrücke sind.
5. $r + s$ ist ein regulärer Ausdruck, falls $r$ und $s$ reguläre Ausdrücke sind.

# Reguläre Ausdrücke

## Definition

Es sei $\Sigma$ ein Alphabet.

1. $\emptyset$ ist ein regulärer Ausdruck.
2. $\epsilon$ ist ein regulärer Ausdruck.
3. $a$ ist ein regulärer Ausdruck, falls $a \in \Sigma$.
4. $rs$ ist ein regulärer Ausdruck, falls $r$ und $s$ reguläre Ausdrücke sind.
5. $r + s$ ist ein regulärer Ausdruck, falls $r$ und $s$ reguläre Ausdrücke sind.
6. $r^*$ ist ein regulärer Ausdruck, falls $r$ ein regulärer Ausdruck ist.

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $r_1 = a$, $r_2 = b$, $r_3 = c$

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $r_1 = a$, $r_2 = b$, $r_3 = c$
2. $r_4 = r_1 + r_2 = (a + b)$

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $r_1 = a$, $r_2 = b$, $r_3 = c$
2. $r_4 = r_1 + r_2 = (a + b)$
3. $r_5 = r_4^* = (a + b)^*$

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a, \ b, \ c\}$

1. $r_1 = a$, $r_2 = b$, $r_3 = c$
2. $r_4 = r_1 + r_2 = (a + b)$
3. $r_5 = r_4^* = (a + b)^*$
4. $r_6 = r_5 r_3 = (a + b)^* c$

# Reguläre Ausdrücke

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $r_1 = a$, $r_2 = b$, $r_3 = c$
2. $r_4 = r_1 + r_2 = (a + b)$
3. $r_5 = r_4^* = (a + b)^*$
4. $r_6 = r_5 r_3 = (a + b)^* c$
5. $r_7 = (r_1^* r_2^*)^* r_3 = (a^* b^*)^* c$

## Definition

Es seien $A, B \subseteq \Sigma^*$ und $w \in \Sigma^*$.

# Definition

Es seien $A, B \subseteq \Sigma^*$ und $w \in \Sigma^*$.

- $AB := \{\, uv \mid u \in A \text{ und } v \in B \,\}$

## Definition

Es seien $A, B \subseteq \Sigma^*$ und $w \in \Sigma^*$.

- $AB := \{\, uv \mid u \in A$ und $v \in B \,\}$
- $wA := \{w\}A$ und $Aw := A\{w\}$

# Definition

Es seien $A, B \subseteq \Sigma^*$ und $w \in \Sigma^*$.

- $AB := \{\, uv \mid u \in A \text{ und } v \in B \,\}$
- $wA := \{w\}A$ und $Aw := A\{w\}$
- $A^i := \begin{cases} \{\epsilon\} & \text{falls } i = 0 \\ A & \text{falls } i = 1 \\ AA^{i-1} & \text{falls } i > 1 \end{cases}$

# Definition

Es seien $A, B \subseteq \Sigma^*$ und $w \in \Sigma^*$.

- $AB := \{\, uv \mid u \in A \text{ und } v \in B \,\}$
- $wA := \{w\}A$ und $Aw := A\{w\}$
- $A^i := \begin{cases} \{\epsilon\} & \text{falls } i = 0 \\ A & \text{falls } i = 1 \\ AA^{i-1} & \text{falls } i > 1 \end{cases}$
- $A^* := \bigcup_{n \geq 0} A^n$
- $A^+ := \bigcup_{n \geq 1} A^n$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$,

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\}$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$,

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$, $A^3 := AA^2 := AAA = \{ababab\}$,

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$, $A^3 := AA^2 := AAA = \{ababab\}$,
   $A^n = \{(ab)^n\}$ *für* $n \geq 1$.

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$, $A^3 := AA^2 := AAA = \{ababab\}$,
   $A^n = \{(ab)^n\}$ *für* $n \geq 1$.
3. $A^* := \bigcup_{n \geq 0} A^n$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$, $A^3 := AA^2 := AAA = \{ababab\}$,
   $A^n = \{(ab)^n\}$ *für* $n \geq 1$.
3. $A^* := \bigcup_{n \geq 0} A^n = \{\epsilon\} \cup \underbrace{\{(ab)^n \mid \ n \geq 1\}}_{A^+}$

## Beispiel

*Sei wieder* $\Sigma := \{a,\ b,\ c\}$, $A := \{ab\} \subseteq \Sigma^*$

1. $Ac := A\{c\} = \{abc\}$
2. $A^0 := \{\epsilon\}$, $A^3 := AA^2 := AAA = \{ababab\}$,
   $A^n = \{(ab)^n\}$ *für* $n \geq 1$.
3. $A^* := \bigcup_{n \geq 0} A^n = \{\epsilon\} \cup \underbrace{\{(ab)^n \mid\ n \geq 1\}}_{A^+} =$
   $\{\epsilon,\ ab,\ abab,\ \dots\}$

# Die Sprache eines regulären Ausdrucks

## Definition

Wir ordnen jedem regulärer Ausdruck $r$ seine *Sprache* $L(r)$ zu:

1. $L(\emptyset) = \emptyset$
2. $L(\epsilon) = \{\epsilon\}$
3. $L(a) = \{a\}$

# Die Sprache eines regulären Ausdrucks

## Definition

Wir ordnen jedem regulärer Ausdruck $r$ seine *Sprache* $L(r)$ zu:

1. $L(\emptyset) = \emptyset$
2. $L(\epsilon) = \{\epsilon\}$
3. $L(a) = \{a\}$
4. $L(rs) = L(r)L(s)$
5. $L(r + s) = L(r) \cup L(s)$
6. $L(r^*) = L(r)^*$

# Die Sprache eines regulären Ausdrucks

### Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

# Die Sprache eines regulären Ausdrucks

### Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right)$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right) = \{a,\ b\}$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a, \ b, \ c\}$

1. $L\left((a+b)\right) \ = \{a, \ b\}$
2. $L\left((a+b)^*\right)$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right) = \{a,\ b\}$
2. $L\left((a+b)^*\right) = \{a, b\}^*$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right) = \{a,\ b\}$
2. $L\left((a+b)^*\right) = \{a,b\}^* = \{w \in \Sigma^* | |w|_c = 0\}$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right)\ = \{a,\ b\}$
2. $L\left((a+b)^*\right)\ = \{a,b\}^* = \{w \in \Sigma^* \| |w|_c = 0\}$
3. $L\left(\emptyset^*\right) = L\left(\emptyset\right)^* =$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right) \ = \{a,\ b\}$
2. $L\left((a+b)^*\right) \ = \{a,b\}^* = \{w \in \Sigma^* \| w|_c = 0\}$
3. $L\left(\emptyset^*\right) = L\left(\emptyset\right)^* = \{\epsilon\}$

# Die Sprache eines regulären Ausdrucks

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

1. $L\left((a+b)\right)\ = \{a,\ b\}$
2. $L\left((a+b)^*\right)\ = \{a, b\}^* = \{w \in \Sigma^* \| |w|_c = 0\}$
3. $L\left(\emptyset^*\right) = L\left(\emptyset\right)^* = \{\epsilon\}$
4. $L\left(\epsilon^*\right) = L\left(\epsilon\right)^* = \{\epsilon\}$

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \;:\Leftrightarrow L(r_1) = L(r_2).$$

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- $\emptyset^* = \epsilon^*$

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*
- *Ja. Es gilt sicherlich* $(a+b)^*c = (a^*b^*)^*c \Leftrightarrow (a+b)^* = (a^*b^*)^*$.

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*
- *Ja. Es gilt sicherlich* $(a+b)^*c = (a^*b^*)^*c \Leftrightarrow (a+b)^* = (a^*b^*)^*$.
- *Wir zeigen* $(a+b)^* = (a^*b^*)^*$, *also* $\{a,\ b\}^* = L\left((a+b)^*\right) = L\left((a^*b^*)^*\right) = L\left((a^*b^*)\right)^*$.

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*
- *Ja. Es gilt sicherlich* $(a+b)^*c = (a^*b^*)^*c \Leftrightarrow (a+b)^* = (a^*b^*)^*$.
- *Wir zeigen* $(a+b)^* = (a^*b^*)^*$, *also* $\{a,\ b\}^* = L\left((a+b)^*\right) = L\left((a^*b^*)^*\right) = L\left((a^*b^*)\right)^*$.
- *Es ist* $\underbrace{L\left((a+b)^*\right)}_{\text{Alle Wörter in } a,\ b} \supseteq L\left((a^*b^*)\right)^*$.

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*
- *Ja. Es gilt sicherlich*
  $(a+b)^*c = (a^*b^*)^*c \Leftrightarrow (a+b)^* = (a^*b^*)^*$.
- *Wir zeigen* $(a+b)^* = (a^*b^*)^*$, *also*
  $\{a,\ b\}^* = L\left((a+b)^*\right) = L\left((a^*b^*)^*\right) = L\left((a^*b^*)\right)^*$.
- *Es ist* $\underbrace{L\left((a+b)^*\right)}_{\text{Alle Wörter in } a,\ b} \supseteq L\left((a^*b^*)\right)^*$.
- *Es ist außerdem* $a,\ b \in L\left((a^*b^*)\right)$.

## Definition

Es sei $\Sigma$ ein Alphabet und $r_1,\ r_2$ seien reguläre Ausdrücke über $\Sigma$.

$$r_1 = r_2 \ :\Leftrightarrow L(r_1) = L(r_2).$$

## Beispiel

*Sei* $\Sigma := \{a,\ b,\ c\}$

- *Gilt* $(a+b)^*c = (a^*b^*)^*c$*?*
- *Ja. Es gilt sicherlich* $(a+b)^*c = (a^*b^*)^*c \Leftrightarrow (a+b)^* = (a^*b^*)^*$.
- *Wir zeigen* $(a+b)^* = (a^*b^*)^*$*, also* $\{a,\ b\}^* = L\left((a+b)^*\right) = L\left((a^*b^*)^*\right) = L\left((a^*b^*)\right)^*$.
- *Es ist* $\underbrace{L\left((a+b)^*\right)}_{\text{Alle Wörter in } a,\ b} \supseteq L\left((a^*b^*)\right)^*$.
- *Es ist außerdem* $a,\ b \in L\left((a^*b^*)\right)$.
- *Also gilt* $\{a,\ b\}^* = L\left((a+b)^*\right) \subseteq L\left((a^*b^*)^*\right)$.

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\mathtt{isNonEmpty}(\emptyset) = \mathtt{False}$,

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\texttt{isNonEmpty}(\emptyset) = \texttt{False}$, $\texttt{isNonEmpty}(a) = \texttt{True}$, $\texttt{isNonEmpty}(\epsilon) = \texttt{True}$

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\texttt{isNonEmpty}(\emptyset) = \texttt{False}$, $\texttt{isNonEmpty}(a) = \texttt{True}$, $\texttt{isNonEmpty}(\epsilon) = \texttt{True}$
2. $\texttt{isNonEmpty}(r_1 + r_2) = \texttt{isNonEmpty}(r_1) \vee \texttt{isNonEmpty}(r_2)$

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\mathtt{isNonEmpty}(\emptyset) = \mathtt{False}$, $\mathtt{isNonEmpty}(a) = \mathtt{True}$, $\mathtt{isNonEmpty}(\epsilon) = \mathtt{True}$
2. $\mathtt{isNonEmpty}(r_1 + r_2) = \mathtt{isNonEmpty}(r_1) \vee \mathtt{isNonEmpty}(r_2)$
3. $\mathtt{isNonEmpty}(rs) = \mathtt{isNonEmpty}(r) \wedge \mathtt{isNonEmpty}(s)$

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\texttt{isNonEmpty}(\emptyset) = \texttt{False}$, $\texttt{isNonEmpty}(a) = \texttt{True}$, $\texttt{isNonEmpty}(\epsilon) = \texttt{True}$
2. $\texttt{isNonEmpty}(r_1 + r_2) = \texttt{isNonEmpty}(r_1) \vee \texttt{isNonEmpty}(r_2)$
3. $\texttt{isNonEmpty}(rs) = \texttt{isNonEmpty}(r) \wedge \texttt{isNonEmpty}(s)$
4. $\texttt{isNonEmpty}(r^+) = \texttt{isNonEmpty}(r)$, $\texttt{isNonEmpty}(r^*) =$

## Problem

*Konstruktion eines Algorithmus* `isNonEmpty` *zur Entscheidung von* $L(r) \neq \emptyset$.

## Lösung

1. $\texttt{isNonEmpty}(\emptyset) = \texttt{False}$, $\texttt{isNonEmpty}(a) = \texttt{True}$, $\texttt{isNonEmpty}(\epsilon) = \texttt{True}$
2. $\texttt{isNonEmpty}(r_1 + r_2) = \texttt{isNonEmpty}(r_1) \vee \texttt{isNonEmpty}(r_2)$
3. $\texttt{isNonEmpty}(rs) = \texttt{isNonEmpty}(r) \wedge \texttt{isNonEmpty}(s)$
4. $\texttt{isNonEmpty}(r^+) = \texttt{isNonEmpty}(r)$, $\texttt{isNonEmpty}(r^*) = \texttt{True}$

`^` start of line
`$` end of line
`.` any character (except newline)
`(a|b)` $a$ or $b$
`abc` concatination of $a, b, c$
`a?` $a$ or $\epsilon$
`a*` sequence of $a$
`a+` one or more $a$

`[0-5]` a digit between 0 and 5
`[0-9A-E]` any digit or $A, B, C, D, E$
`[^0-9]` Any non-digit
`a{50}` 50 times $a$
`a{50,60}` between 50 and 60 $a$

Capital letter as first, digit as third character or ends with $bbb$:

```
(^[A-Z].[0-9]|bbb$)
```

Regular expression for floating point numbers?

Regular expression for floating point numbers?

(42, 4.2, -415.00, +55, +.99, 1e-7, 3.5e20)

Regular expression for floating point numbers?

(42, 4.2, -415.00, +55, +.99, 1e-7, 3.5e20)

```
(\-|\+)?[0-9]*\.?[0-9]+((e|E)(\-|\+)?[0-9]+)?
```

Regular expression for email?

Regular expression for email?

## General Email Regex (RFC 5322 Official Standard)

```
(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x2
1\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z
0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]
|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x
0b\x0c\x0e-\x7f])+)\])
```

# Regex Golf


Quelle: xkcd.com