

# Diskrete Mathematik

Arne Dür      Kurt Girstmair      Simon Legner  
Georg Moser      Harald Zankl

Fakultät für Mathematik, Informatik und Physik @ UIBK  
Sommersemester 2011



# Zusammenfassung der letzten LV

## Definition

ein  $\epsilon$ -NEA ist gegeben durch

- 1 eine endliche Menge  $Q$ , den Zuständen
- 2 eine endliche Menge  $\Sigma$ , dem Eingabealphabet
- 3 eine Abbildung

$$\delta: Q \times \Sigma \cup \{\epsilon\} \rightarrow \mathcal{P}(Q)$$

die **Übergangsfunktion**

- 4 einen ausgezeichneten Zustand, den Startzustand
- 5 eine Teilmenge  $F \subseteq Q$ , den akzeptierenden Zuständen

um Verwechslungen auszuschließen, fordern wir dass  $\epsilon \notin \Sigma$

## Epsilon-Hülle

### Definition

betrachte den Zustandsgraphen des Automaten, setze  $S = \{q\}$   
 der folgende Algorithmus markiert alle Zustände in  $\epsilon$ -Hülle( $q$ ):

- 1 markiere die Zustände in  $S$
- 2 solange  $S \neq \emptyset$ , wiederhole:
  - wähle einen Zustand  $p$  aus  $S$  und entferne  $p$
  - bestimme alle unmarkierten Nachfolger von  $p$  die **mit einer  $\epsilon$ -Kante erreichbar sind**
  - markiere diese und füge sie zu  $S$  hinzu

### Definition

die **Sprache** von  $\epsilon$ -NEA  $E = (Q, \Sigma, \delta, q_0, F)$ :

$$L(E) := \{x \mid \widehat{\delta}(q_0, x) \cap F \neq \emptyset\}$$

# Übersicht

## Endliche Automaten

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion,  $\epsilon$ -NEAs, Umwandlung endlicher Automaten in reguläre Ausdrücke, Pumpinglemma, Minimierung

## Berechenbarkeitstheorie

Einführung in die Berechenbarkeitstheorie, Turingmaschinen, Entscheidungsprobleme, Universelle Maschinen und Diagonalisierung,

## Komplexitätstheorie

Einführung in die Komplexitätstheorie, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

# Übersicht

## Endliche Automaten

Automaten, reguläre Sprachen und Grammatiken, (nicht)-deterministische endliche Automaten, Teilmengenkonstruktion,  $\epsilon$ -NEAs, Umwandlung endlicher Automaten in reguläre Ausdrücke, Pumpinglemma, Minimierung

## Berechenbarkeitstheorie

Einführung in die Berechenbarkeitstheorie, Turingmaschinen, Entscheidungsprobleme, Universelle Maschinen und Diagonalisierung,

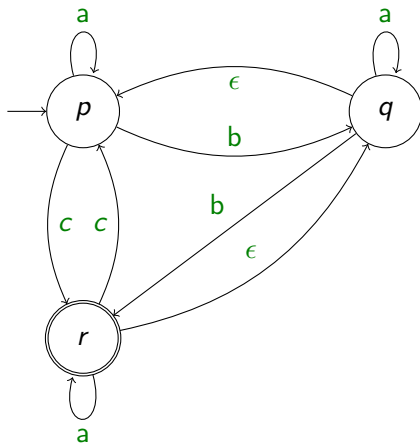
## Komplexitätstheorie

Einführung in die Komplexitätstheorie, die Klassen P und NP, logarithmisch platzbeschränkte Reduktionen, Speicherplatzkomplexität

# Beispiel (1)

Beispiel

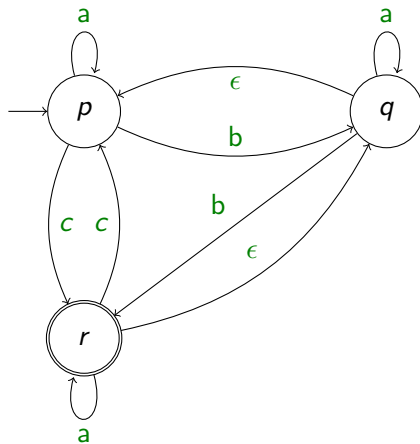
betrachte  $\epsilon$ -NEA  $A$



# Beispiel (1)

Beispiel

betrachte  $\epsilon$ -NEA  $A$



Frage

welche Sprache akzeptiert  $A$ ?

## Beispiel (2)

	$\epsilon$	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
*r	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$



## Beispiel (2)

	$\epsilon$	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
*r	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

## Epsilon-Hüllen

$$\epsilon\text{-Hülle}(p) = \{p\} \quad \epsilon\text{-Hülle}(q) = \{q, p\} \quad \epsilon\text{-Hülle}(r) = \{r, q, p\}$$

## Beispiel (2)

	$\epsilon$	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
*r	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

## Epsilon-Hüllen

$$\epsilon\text{-Hülle}(p) = \{p\} \quad \epsilon\text{-Hülle}(q) = \{q, p\} \quad \epsilon\text{-Hülle}(r) = \{r, q, p\}$$

## Zustandstabelle

	a	b	c
$\rightarrow \{p\}$	$\{p\}$	$\{p, q\}$	$\{p, q, r\}$

## Beispiel (2)

	$\epsilon$	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
*r	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

## Epsilon-Hüllen

$$\epsilon\text{-Hülle}(p) = \{p\} \quad \epsilon\text{-Hülle}(q) = \{q, p\} \quad \epsilon\text{-Hülle}(r) = \{r, q, p\}$$

## Zustandstabelle

	a	b	c
$\rightarrow \{p\}$	$\{p\}$	$\{p, q\}$	$\{p, q, r\}$
$\{p, q\}$	$\{p, q\}$	$\{p, q, r\}$	$\{p, q, r\}$

## Beispiel (2)

	$\epsilon$	a	b	c
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
*r	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

## Epsilon-Hüllen

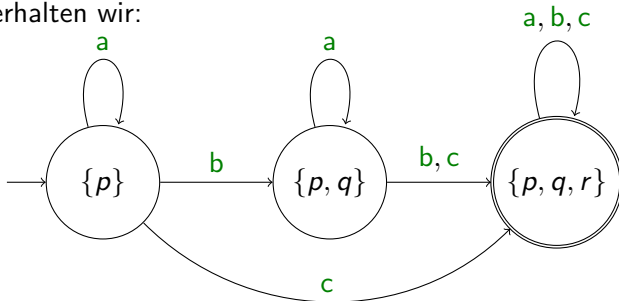
$$\epsilon\text{-Hülle}(p) = \{p\} \quad \epsilon\text{-Hülle}(q) = \{q, p\} \quad \epsilon\text{-Hülle}(r) = \{r, q, p\}$$

## Zustandstabelle

	a	b	c
$\rightarrow \{p\}$	$\{p\}$	$\{p, q\}$	$\{p, q, r\}$
$\{p, q\}$	$\{p, q\}$	$\{p, q, r\}$	$\{p, q, r\}$
* $\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$

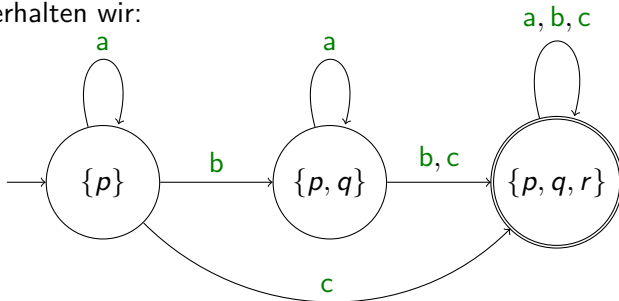
## Beispiel (3)

in Summe erhalten wir:



## Beispiel (3)

in Summe erhalten wir:



### Antwort

A akzeptiert alle Wörter über  $\{a, b, c\}$ , sodass

- 1 entweder zwei **b**s oder
- 2 ein **c** auftreten

# Äquivalenz von $\epsilon$ -NEAs und DEAs

## Satz

sei  $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$  der DEA, der mit der Teilmengenkonstruktion aus  $\epsilon$ -NEA  $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$  konstruiert ist, dann gilt  $L(D) = L(N)$

## Beweisansatz

wie für die Korrektheit der Teilmengenkonstruktion für NEAs ■

# Äquivalenz von $\epsilon$ -NEAs und DEAs

## Satz

sei  $D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$  der DEA, der mit der Teilmengenkonstruktion aus  $\epsilon$ -NEA  $E = (Q_E, \Sigma, \delta_E, q_0, F_E)$  konstruiert ist, dann gilt  $L(D) = L(N)$

## Beweisansatz

wie für die Korrektheit der Teilmengenkonstruktion für NEAs ■

## Satz

eine Sprache  $L$  wird genau dann von einem  $\epsilon$ -NEA akzeptiert, wenn  $L$  von einem DEA akzeptiert wird.

## Beweis.

der Satz folgt aus der Teilmengenkonstruktion und der einfachen Einsicht, dass jeder DEA in einen  $\epsilon$ -NEA umgeschrieben werden kann ■



# Anwendung von Endlichen Automaten

## Anwendung

- Softwarebasiertes Entwickeln und Testen von Schaltkreisen
- Compilerbau: Lexikalische Analyse
- Textsuche; Pattern Matching
- Softwareverifikation von Protokollen
- Spielengine von Computerspielen

# Anwendung von Endlichen Automaten

## Anwendung

- Softwarebasiertes Entwickeln und Testen von Schaltkreisen
- Compilerbau: Lexikalische Analyse
- Textsuche; Pattern Matching
- Softwareverifikation von Protokollen
- Spielengine von Computerspiele

# Anwendung von Endlichen Automaten

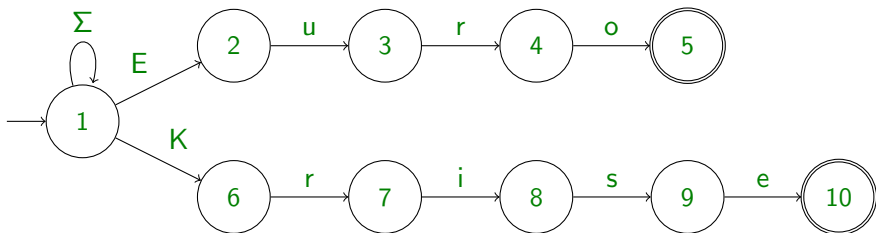
## Anwendung

- Softwarebasiertes Entwickeln und Testen von Schaltkreisen
- Compilerbau: Lexikalische Analyse
- Textsuche; Pattern Matching
- Softwareverifikation von Protokollen
- Spielengine von Computerspiele

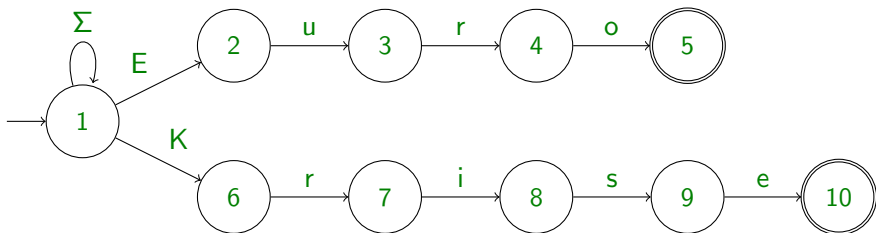
## Beispiel

- gesucht sei eine Liste von Schlüsselwörter in einem Text oder HTML/XML Dokument
- Inhalt des Textes ändert sich täglich, sodass Indizierung zu teuer
- suche die Worte Euro oder Krise in einer Online-Zeitung

## Beispiel



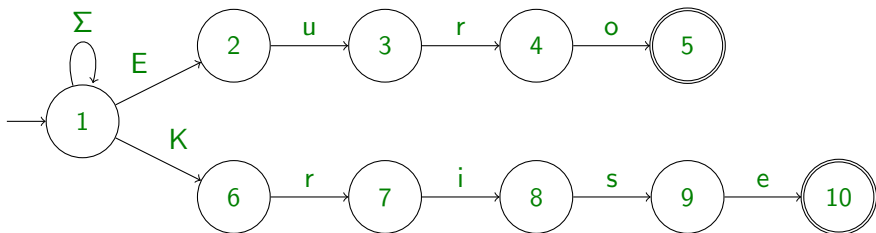
## Beispiel



## Implementierung

- wir können  $N$  direkt simulieren, indem wir alle Möglichkeiten aufzählen

## Beispiel



## Implementierung

- wir können  $N$  direkt simulieren, indem wir alle Möglichkeiten aufzählen
- oder wir wanden  $N$  in einen DEA  $D$  um und implementieren  $D$

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:



## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1 sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1** sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2** Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1** sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2** Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn
  - entweder in  $N$  mit  $a$  markierte Kante von  $p_i$  nach  $q_j$ , oder

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1** sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2** Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn
  - entweder in  $N$  mit  $a$  markierte Kante von  $p_i$  nach  $q_j$ , oder
  - in  $N$  mit  $a$  markierte Kante von 1 nach  $q_j$ , wenn keine Kante von  $p_i$  nach  $q_j$  mit  $a$  markiert

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1** sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2** Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn
  - entweder in  $N$  mit  $a$  markierte Kante von  $p_i$  nach  $q_j$ , oder
  - in  $N$  mit  $a$  markierte Kante von 1 nach  $q_j$ , wenn keine Kante von  $p_i$  nach  $q_j$  mit  $a$  markiert

## Beispiel

## Lemma

*der so erhaltene DEA hat maximal soviele Zustände wie der NEA*

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1 sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
 korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
 jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2 Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn
  - entweder in  $N$  mit  $a$  markierte Kante von  $p_i$  nach  $q_j$ , oder
  - in  $N$  mit  $a$  markierte Kante von 1 nach  $q_j$ , wenn keine Kante von  $p_i$   
 nach  $q_j$  mit  $a$  markiert

## Beispiel

- $D$  enthält zB die Zustände:  $\{1\}$ ,  $\{1, 2\}$ ,  $\{1, 3\}$ ,  $\{1, 4\}$ ,  $\{1, 5\}$

## Lemma

der so erhaltene DEA hat maximal soviele Zustände wie der NEA

## Beweis.

Analyse der Teilmengenkonstruktion ergibt:

- 1 sei  $p$  ein Zustand in  $N$ , erreichbar beim Lesen von  $a_1 \dots a_m$   
 korrespondierende Zustand in  $D$  besteht aus 1 und  $p$ , sowie  
 jedem Zustand aus  $N$  der durch einen Suffix von  $a_1 \dots a_m$  erreichbar
- 2 Kanten in  $D$  von  $\{1, p_1, \dots, p_n\}$  nach  $\{1, q_1, \dots, q_m\}$ , wenn
  - entweder in  $N$  mit  $a$  markierte Kante von  $p_i$  nach  $q_j$ , oder
  - in  $N$  mit  $a$  markierte Kante von 1 nach  $q_j$ , wenn keine Kante von  $p_i$  nach  $q_j$  mit  $a$  markiert

## Beispiel

- $D$  enthält zB die Zustände:  $\{1\}$ ,  $\{1, 2\}$ ,  $\{1, 3\}$ ,  $\{1, 4\}$ ,  $\{1, 5\}$
- Kante von  $\{1\}$  nach  $\{1, 2\}$  mit E markiert; Kante von  $\{1, 2\}$  nach  $\{1\}$  mit  $\Sigma \setminus \{E, K, u\}$  markiert

## Beispiel

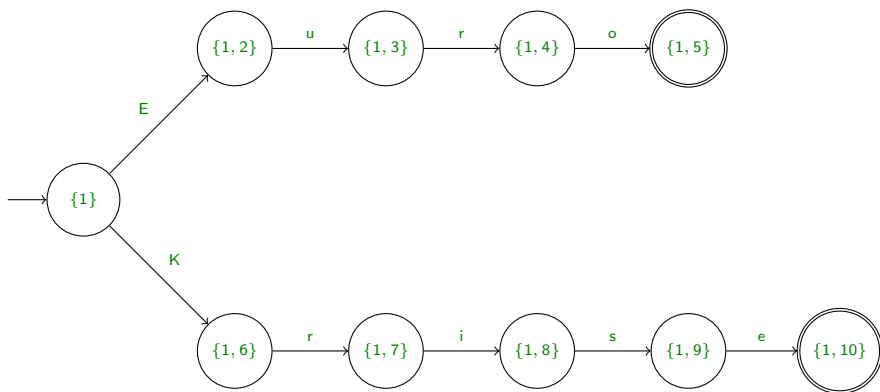
durch die Teilmengenkonstruktion erhalten wir den folgenden DEA  $D$





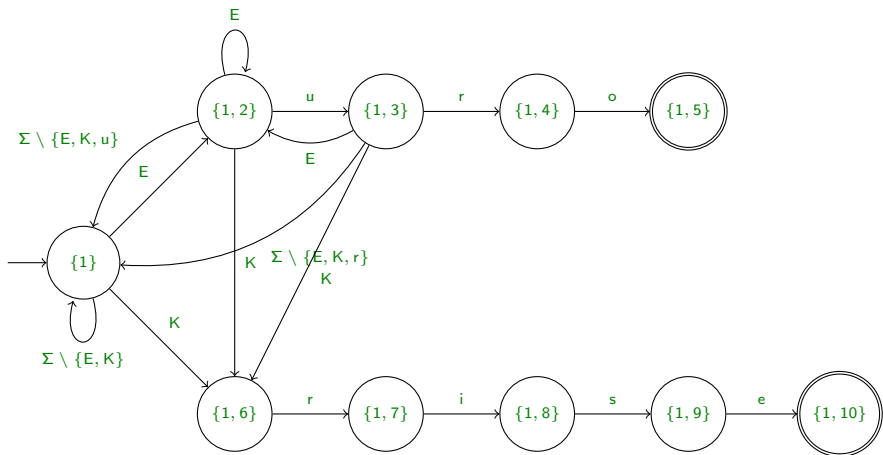
## Beispiel

durch die Teilmengenkonstruktion erhalten wir den folgenden DEA  $D$



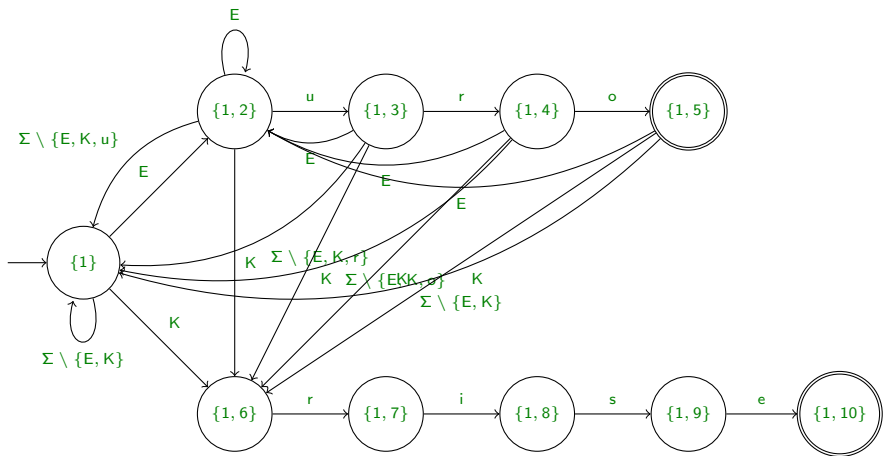
## Beispiel

durch die Teilmengenkonstruktion erhalten wir den folgenden DEA  $D$



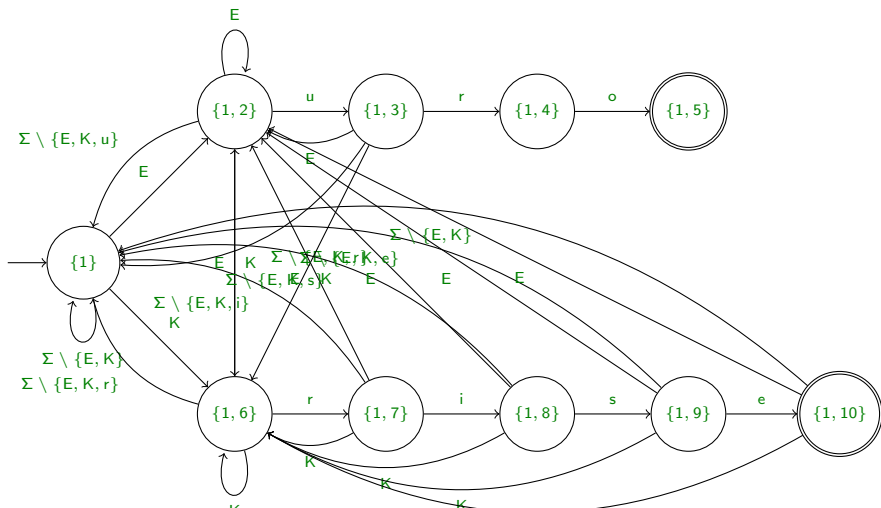
## Beispiel

durch die Teilmengenkonstruktion erhalten wir den folgenden DEA  $D$



## Beispiel

durch die Teilmengenkonstruktion erhalten wir den folgenden DEA  $D$



seien  $L, M$  formale Sprachen

seien  $L, M$  formale Sprachen

### Definition

die **Vereinigung**  $L \cup M$  von  $L$  und  $M$ , ist die Menge der Wörter, die entweder in  $L$  oder in  $M$  liegen

seien  $L, M$  formale Sprachen

### Definition

die **Vereinigung**  $L \cup M$  von  $L$  und  $M$ , ist die Menge der Wörter, die entweder in  $L$  oder in  $M$  liegen

### Definition

die **Konkatenation**  $L \cdot M$  von  $L$  und  $M$ , ist die Menge der Wörter, die gebildet werden können, indem wir ein Wort aus  $L$  mit einem Wort aus  $M$  verketten

seien  $L, M$  formale Sprachen

### Definition

die **Vereinigung**  $L \cup M$  von  $L$  und  $M$ , ist die Menge der Wörter, die entweder in  $L$  oder in  $M$  liegen

### Definition

die **Konkatenation**  $L \cdot M$  von  $L$  und  $M$ , ist die Menge der Wörter, die gebildet werden können, indem wir ein Wort aus  $L$  mit einem Wort aus  $M$  verketten

### Definition

der **Abschluss**  $L^*$  von  $L$  ist die Menge der Wörter, die gebildet werden können durch die Verkettung von beliebig vielen Elementen aus  $L$



seien  $L, M$  formale Sprachen

### Definition

die **Vereinigung**  $L \cup M$  von  $L$  und  $M$ , ist die Menge der Wörter, die entweder in  $L$  oder in  $M$  liegen

### Definition

die **Konkatenation**  $L \cdot M$  von  $L$  und  $M$ , ist die Menge der Wörter, die gebildet werden können, indem wir ein Wort aus  $L$  mit einem Wort aus  $M$  verketten

### Definition

der **Abschluss**  $L^*$  von  $L$  ist die Menge der Wörter, die gebildet werden können durch die Verkettung von beliebig vielen Elementen aus  $L$

### Beispiel

Algebra  $(\Sigma^*, \cdot, \epsilon)$ , sodass  $\Sigma^*$  die Menge aller Wörter über  $\Sigma$ ,  $\cdot$  die Verkettung und  $\epsilon$  das neutrale Element, heißt **Wortmonoid**

# Reguläre Ausdrücke

## Erinnerung

sei  $\Sigma$  ein endliches Alphabet; wir definieren reguläre Ausdrücke induktiv

## Basis

- 1  $\emptyset$  ist ein regulärer Ausdruck (kurz: RA)
- 2  $\epsilon$  ist ein RA
- 3 für jedes Symbol  $a$  ist  $a$  ein RA

## Schritt

- 1 für jeden RA  $E$  ist  $E^*$  ein RA
- 2 für RAs  $E$  und  $F$  ist  $EF$  ein RA
- 3 für RAs  $E$  und  $F$  ist  $E + F$  ein RA
- 4 wenn  $E$  ein RA ist, dann ist  $(E)$  ein RA

# Reguläre Ausdrücke

## Erinnerung

sei  $\Sigma$  ein endliches Alphabet; wir definieren reguläre Ausdrücke induktiv

## Basis

- 1  $\emptyset$  ist ein regulärer Ausdruck (kurz: RA)
- 2  $\epsilon$  ist ein RA
- 3 für jedes Symbol  $a$  ist  $\mathbf{a}$  ein RA

## Sprache von $\emptyset$

$$\widehat{L(\emptyset)} := \emptyset$$

$$L(\epsilon) := \{\epsilon\}$$

$$L(\mathbf{a}) := \{a\}$$

## Schritt

- 1 für jeden RA  $E$  ist  $E^*$  ein RA
- 2 für RAs  $E$  und  $F$  ist  $EF$  ein RA
- 3 für RAs  $E$  und  $F$  ist  $E + F$  ein RA
- 4 wenn  $E$  ein RA ist, dann ist  $(E)$  ein RA

$$L(E^*) := L(E)^*$$

$$L(EF) := L(E) \cdot L(F)$$

$$L(E + F) := L(E) \cup L(F)$$

$$L((E)) := L(E)$$

## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden  $0$ en und  $1$ en enthält

## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden 0en und 1en enthält

## Lösung

- regulärer Ausdruck  $10$  beschreibt den String 10

## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden 0en und 1en enthält

## Lösung

- regulärer Ausdruck  $10$  beschreibt den String 10
- also beschreibt  $(01)^*$  alle Strings der Form

01010101...

## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden 0en und 1en enthält

## Lösung

- regulärer Ausdruck  $10$  beschreibt den String  $10$
- also beschreibt  $(01)^*$  alle Strings der Form

$01010101 \dots$

- und  $(10)^*$  beschreibt

$10101010 \dots$

## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden 0en und 1en enthält

## Lösung

- regulärer Ausdruck  $\mathbf{10}$  beschreibt den String 10
- also beschreibt  $\mathbf{(01)^*}$  alle Strings der Form

0101010101...

- und  $\mathbf{(10)^*}$  beschreibt

1010101010...

$$L = \mathbf{(01)^*} + \mathbf{(10)^*} + \mathbf{0(10)^*} + \mathbf{1(01)^*}$$



## Aufgabe

wir wollen einen regulären Ausdruck formulieren,  
dessen Sprache  $L$  alle Strings mit abwechselnden 0en und 1en enthält

## Lösung

- regulärer Ausdruck  $10$  beschreibt den String 10
- also beschreibt  $(01)^*$  alle Strings der Form

0101010101...

- und  $(10)^*$  beschreibt

1010101010...

$$\begin{aligned} L &= (01)^* + (10)^* + 0(10)^* + 1(01)^* \\ &= (\epsilon + 1)(01)^*(\epsilon + 0) \end{aligned}$$

# Algebraische Gesetze für reguläre Ausdrücke

Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

# Algebraische Gesetze für reguläre Ausdrücke

Beispiel

$$\mathbf{0^*1} + (\mathbf{0^*1})(\mathbf{0^*1} + \epsilon)^*(\mathbf{0^*1}) \equiv (\mathbf{0^*1})^+$$

seien  $L, M, N$  beliebige reguläre Ausdrücke

# Algebraische Gesetze für reguläre Ausdrücke

## Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

seien  $L, M, N$  beliebige reguläre Ausdrücke

## Assoziativität und Kommutativität

$$\mathbf{1} \quad L(L + M) = L(M + L)$$

Kommutativität von  $+$

# Algebraische Gesetze für reguläre Ausdrücke

## Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

seien  $L, M, N$  beliebige reguläre Ausdrücke

## Assoziativität und Kommutativität

$$\mathbf{1} \quad L(L + M) = L(M + L)$$

Kommutativität von +

$$\mathbf{2} \quad L((L + M) + N) = L(L + (M + N))$$

Assoziativität von +

# Algebraische Gesetze für reguläre Ausdrücke

## Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

seien  $L, M, N$  beliebige reguläre Ausdrücke

## Assoziativität und Kommutativität

$$1 \quad L(L + M) = L(M + L)$$

Kommutativität von  $+$

$$2 \quad L((L + M) + N) = L(L + (M + N))$$

Assoziativität von  $+$

$$3 \quad L((LM)N) = L(L(MN))$$

Assoziativität der Verkettung

# Algebraische Gesetze für reguläre Ausdrücke

## Beispiel

$$0^*1 + (0^*1)(0^*1 + \epsilon)^*(0^*1) \equiv (0^*1)^+$$

seien  $L, M, N$  beliebige reguläre Ausdrücke

## Assoziativität und Kommutativität

$$\mathbf{1} \quad L(L + M) = L(M + L)$$

Kommutativität von  $+$

$$\mathbf{2} \quad L((L + M) + N) = L(L + (M + N))$$

Assoziativität von  $+$

$$\mathbf{3} \quad L((LM)N) = L(L(MN))$$

Assoziativität der Verkettung

## Erinnerung

Kommutativität der Verkettung gilt nicht

## Neutrales Element und Löscher

### Erinnerung

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst



## Neutrales Element und Löscher

### Erinnerung

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst

### Lemma

$$\mathbf{1} \quad L(\emptyset + L) = L(L + \emptyset) = L(L) \quad \emptyset \text{ ist das } \textit{neutrale Element} \text{ für } +$$

# Neutrales Element und Löscher

## Erinnerung

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst

## Lemma

$$\mathbf{1} \quad L(\emptyset + L) = L(L + \emptyset) = L(L)$$

$\emptyset$  ist das **neutrale Element** für  $+$

$$\mathbf{2} \quad L(\epsilon L) = L(L\epsilon) = L(L)$$

$\epsilon$  ist das **neutrale Element** von  $\cdot$

## Neutrales Element und Löscher

### Erinnerung

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst

### Lemma

$$\mathbf{1} \quad L(\emptyset + L) = L(L + \emptyset) = L(L) \quad \emptyset \text{ ist das } \textit{neutrale Element} \text{ für } +$$

$$\mathbf{2} \quad L(\epsilon L) = L(L\epsilon) = L(L) \quad \epsilon \text{ ist das } \textit{neutrale Element} \text{ von } \cdot$$

### Definition

ein **Löscher** (Annihilator) für einen Operator ist ein Element das die Operation zunichte macht

## Neutrales Element und Löscher

### Erinnerung

ein **neutrales Element** für einen Operator ist ein Element das die Operation nicht beeinflusst

### Lemma

$$\mathbf{1} \quad L(\emptyset + L) = L(L + \emptyset) = L(L) \quad \emptyset \text{ ist das } \textit{neutrale Element} \text{ für } +$$

$$\mathbf{2} \quad L(\epsilon L) = L(L\epsilon) = L(L) \quad \epsilon \text{ ist das } \textit{neutrale Element} \text{ von } \cdot$$

### Definition

ein **Löscher** (Annihilator) für einen Operator ist ein Element das die Operation zunichte macht

### Lemma

$$\mathbf{1} \quad L(\emptyset L) = L(L\emptyset) = \emptyset \quad \emptyset \text{ ist ein } \textit{Löscher} \text{ der Verkettung}$$

# Distributivgesetze und Idempotenzgesetz

Lemma

$$\mathbf{1} \quad L(L(M + N)) = L(LM + LN)$$

*Links distributivität*

# Distributivgesetze und Idempotenzgesetz

## Lemma

$$\mathbf{1} \quad L(L(M + N)) = L(LM + LN)$$

*Links distributivität*

$$\mathbf{2} \quad L((M + N)L) = L(ML + NL)$$

*Rechts distributivität*

# Distributivgesetze und Idempotenzgesetz

## Lemma

$$\mathbf{1} \quad L(L(M + N)) = L(LM + LN)$$

*Links distributivität*

$$\mathbf{2} \quad L((M + N)L) = L(ML + NL)$$

*Rechts distributivität*

$$\mathbf{0} + \mathbf{01}^* \equiv \mathbf{0}\epsilon + \mathbf{01}^* \equiv \mathbf{0}(\epsilon + \mathbf{1}^*) \equiv \mathbf{01}^*$$

# Distributivgesetze und Idempotenzgesetz

## Lemma

$$\mathbf{1} \quad L(L(M + N)) = L(LM + LN)$$

*Links-distributivität*

$$\mathbf{2} \quad L((M + N)L) = L(ML + NL)$$

*Rechts-distributivität*

$$\mathbf{0} + \mathbf{01}^* \equiv \mathbf{0}\epsilon + \mathbf{01}^* \equiv \mathbf{0}(\epsilon + \mathbf{1}^*) \equiv \mathbf{01}^*$$

## Lemma

$$\mathbf{1} \quad L(L + L) = L(L)$$

*Idempotenzgesetz von +*



# Distributivgesetze und Idempotenzgesetz

## Lemma

$$\mathbf{1} \quad L(L(M + N)) = L(LM + LN)$$

*Links-distributivität*

$$\mathbf{2} \quad L((M + N)L) = L(ML + NL)$$

*Rechts-distributivität*

$$\mathbf{0} + \mathbf{01}^* \equiv \mathbf{0\epsilon} + \mathbf{01}^* \equiv \mathbf{0(\epsilon + 1^*)} \equiv \mathbf{01}^*$$

## Lemma

$$\mathbf{1} \quad L(L + L) = L(L)$$

*Idempotenzgesetz von +*

$$\mathbf{0}^* + \mathbf{0}^* \equiv \mathbf{0}^*$$

# Gesetze für den Kleene-Stern

Lemma

$$\mathbf{1} \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

# Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

# Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

# Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

# Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L?) = L(\epsilon + L)$$

## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L^?) = L(\epsilon + L)$$

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$

## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

*Definition*

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L^?) = L(\epsilon + L)$$

*Definition*

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$



## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

*Definition*

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L^?) = L(\epsilon + L)$$

*Definition*

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$

$$(0^* + 1^?)^*$$

## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

*Definition*

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L?) = L(\epsilon + L)$$

*Definition*

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$

$$(0^* + 1?)^* \equiv ((0^*)^* (1?)^*)^*$$

## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

*Definition*

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L?) = L(\epsilon + L)$$

*Definition*

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$

$$(0^* + 1?)^* \equiv ((0^*)^* (1?)^*)^* \equiv (0^* (\epsilon + 1)^*)^*$$

## Gesetze für den Kleene-Stern

## Lemma

$$1 \quad L(L^*) = L(L^* L^*) = L((L^*)^*)$$

$$2 \quad L(\emptyset^*) = L(\epsilon)$$

$$3 \quad L(L^+) = L(LL^*) = L(L^*L)$$

*Definition*

$$4 \quad L(L^*) = L(L^+ + \epsilon)$$

$$5 \quad L(L?) = L(\epsilon + L)$$

*Definition*

$$6 \quad L((E + F)^*) = L((E^* + F^*)^*) = \\ = L((E^* F^*)^*) = L((E^* F)^* E^*) = L(E^* (F E^*)^*)$$

$$(\mathbf{0}^* + \mathbf{1}?)^* \equiv ((\mathbf{0}^*)^* (\mathbf{1}?)^*)^* \equiv (\mathbf{0}^* (\epsilon + \mathbf{1})^*)^* \equiv (\mathbf{0}^* \mathbf{1}^*)^*$$

## Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*)$$

## Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*)$$

Beweis.

zunächst  $L(L)^* \subseteq L(L)^* L(L)^*$ :

## Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

Beweis.

zunächst  $L(L)^* \subseteq L(L^*)L(L)^*$ :

## Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$



## Gesetze für den Kleene-Stern (2)

Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*L(L)^*$

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*L(L)^*$
- $\exists y, z$  aus  $L(L)^*$ , sodass  $x = yz$

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*L(L)^*$
- $\exists y, z$  aus  $L(L)^*$ , sodass  $x = yz$
- $\exists k, l$  sodass  $y \in L(L)^k$ ,  $z \in L(L)^l$  und  $x \in L(L)^{k+l}$

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*L(L)^*$
- $\exists y, z$  aus  $L(L)^*$ , sodass  $x = yz$
- $\exists k, l$  sodass  $y \in L(L)^k$ ,  $z \in L(L)^l$  und  $x \in L(L)^{k+l}$
- somit  $x \in L(L)^*$  und die Behauptung folgt

## Gesetze für den Kleene-Stern (2)

## Lemma

$$L(L^*) = L(L^*L^*) \quad (= L(L^*)L(L^*) = L(L)^*L(L)^*)$$

## Beweis.

zunächst  $L(L)^* \subseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*$ , dann  $x = x\epsilon \in L(L)^*L(L)^*$
- also folgt die Behauptung

zunächst  $L(L)^* \supseteq L(L)^*L(L)^*$ :

- sei  $x \in L(L)^*L(L)^*$
- $\exists y, z$  aus  $L(L)^*$ , sodass  $x = yz$
- $\exists k, l$  sodass  $y \in L(L)^k$ ,  $z \in L(L)^l$  und  $x \in L(L)^{k+l}$
- somit  $x \in L(L)^*$  und die Behauptung folgt

