REGULÄRE AUSDRÜCKE UND AUTOMATEN

Reguläre Ausdrücke und Automaten

• Endliche Automaten (DEA, NEA, NEA/ε)	Reguläre Ausdrücke
A	r
L(A)	L(r)

Zu zeigen:
„Für A gibt es r mit L(A) = L(r).“ und „Für r gibt es A (NEA/ε) mit L(r) = L(A).“,

$L \subset \sum^*$ heißt regulär, falls es einen endlichen Automaten A bzw. einen regulären Ausdruck r gibt mit L = L(A) = L(r)

$r \in REG_\Sigma \rightarrow$ endlicher Automat A mit L(r) = L(A)

© Prof. Dr. Juho Mäkiö – juho.maekioe@hs-emden-leer.de

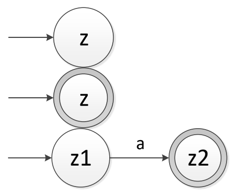
Reguläre Ausdrücke und Automaten

- Analog zur „inneren“ rekursiven Struktur von r wird ein Automat konstruiert
- $\emptyset$, ε ∈ REG_Σ , a ∈ REG_Σ (a ∈ ∑)
- Automaten:

∅

ε

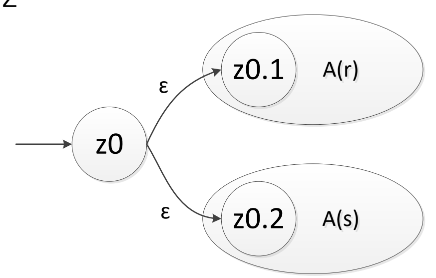
a



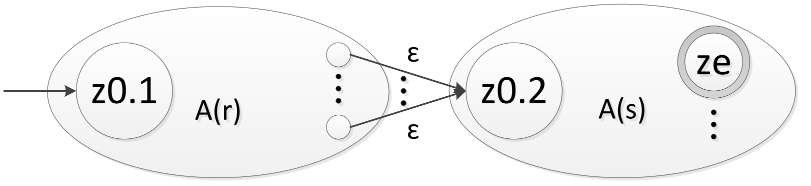
Reguläre Ausdrücke und Automaten

- r, s ∈ REG_Σ => r + s ∈ REG_Σ
- Automat A(r) und Automat A(s) müssen zusammen montiert werden
- r, s ∈ REG_Σ => r•s ∈ REG_Σ

A(r) A(s)



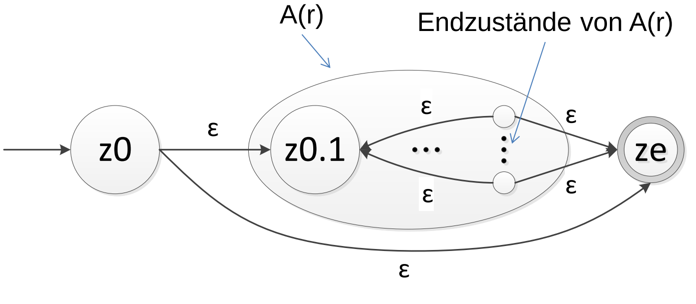
Reguläre Ausdrücke und Automaten



- Alle Endzustände von A(r) werden über einen ε-Übergang mit dem Startzustand von A(s) verbunden.

Reguläre Ausdrücke und Automaten

$r \in REG_\Sigma \Rightarrow r^* \in REG_\Sigma$



r – Schleife - kann oft verkürzt werden

Reguläre Ausdrücke und Automaten

- Beispiel:

  Für r = 10*+01* soll ein NEA/ε konstruiert werden

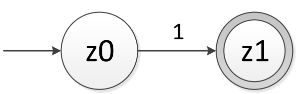
  10* + 01* -> A(10*+01*)

  A(10*) A(01*)

  A(1) A(0*) A(0) A(1*)

Reguläre Ausdrücke und Automaten

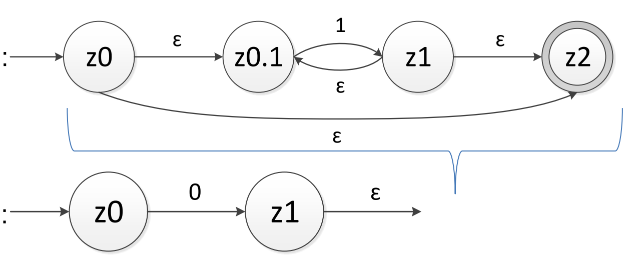
A(1)



A(0) entsprechend

A(1*):

A(01*):

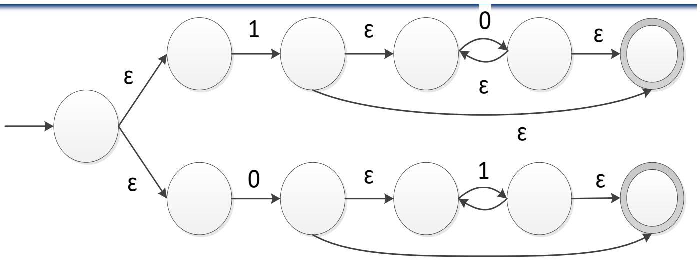


A(10*) entsprechend

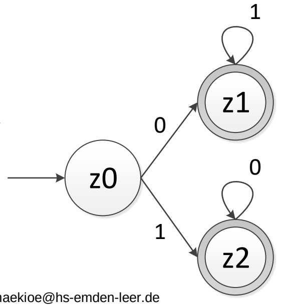
Damit entsteht der folgende NEA/ε:
(ohne Zustandsnamen)…

Reguläre Ausdrücke und Automaten

A(10*+01*):



Viele ε-Übergänge, aber von der Konstruktion her eindeutig: → DEA



Übung

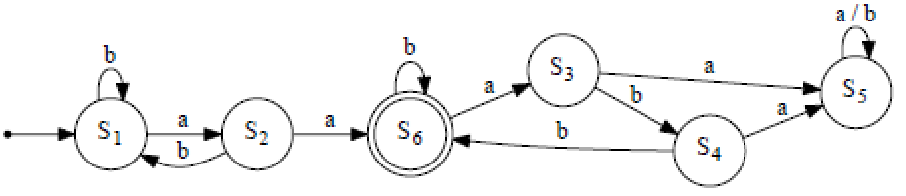
a) Konstruieren Sie Automaten A für die Sprache
L(A) = { x ∈ {a,b}* | x enthält aa oder bb}
Geben Sie einen regulären Ausdruck r an, so dass es gilt:
L(r) = L(A)

b) Konstruieren Sie Automaten A für die Sprache
L(A) = { x ∈ {0,1}* | $|x|_1$ mod 2 = 1}
Geben Sie einen regulären Ausdruck r an, so dass es gilt:
L(r) = L(A)

c) Konstruieren Sie Automaten A für die Sprache
L(A) = { x ∈ {a,b}* | x endet mit bb}
Geben Sie einen regulären Ausdruck r an, so dass es gilt:
L(r) = L(A)

Übung

- Gegeben sei der reguläre Ausdruck RA({a, b}) : α = (b + ab)*aa(abb + b)*
- Konstruieren Sie einen deterministischen endlichen Automaten A mit L(A) = L(α). Geben Sie A vollständig an.



Übung

- Geben Sie die regulären Ausdrücke für die folgenden Sprachen an:
  - Die Wörter der Sprache haben höchstens 4 Zeichen: L = {α ϵ {a, b}* | |α| ≤ 4}
  - In den Wörter der Sprache dürfen a‘s nur alleine vorkommen: L = {α ϵ {a, b}* | ∀ u, v ϵ {a, b}* : α ≠ uaav}
  - Die Wörter der Sprache dürfen keine drei oder mehr a‘s hintereinander auftreten: RA({a, b}) : L = {α ϵ {a, b}* | ∀ u, v ϵ {a, b}* : α ≠ uaaav}
  - Die Wörter der Sprache haben eine gerade Anzahl von Zeichen, wobei es dürfen nie gleiche Zeichen hintereinander stehen: L = {α ϵ $\{a, b\}^{2n}$ | n ϵ N_0; ∀ t, l ϵ {a, b}*, ∀ z ϵ {a, b} : α ≠ tzzl}

KONTEXTSENSITIVE GRAMMATIKEN (TYP-1-GRAMMATIK)

- Eine Grammatik heißt <u>kontextsensitiv (Typ-1-Grammatik)</u>, wenn alle Regeln von folgender Form sind:

S -> ε oder

l -> r mit $l \in (\Sigma \cup V)^{+} \setminus \Sigma^{*}$

$r \in (\Sigma \cup V)^{*}$

und $|l| \leq |r|$

[l enthält mindestens eine Variable, |l| ≤ |r| bedeutet, dass es keine Verkürzung gibt]

- Anmerkungen:

Jede Typ-3-/Typ-2-Grammatik ist auch vom Typ-1.
(Die Umkehrung gilt nicht.)

l = $\underbrace{. . . B . . .}_{l}$-> r (andere Notationen -> Literatur)

abhängig vom „Kontext“ . . . . . . wird B ersetzt.

Beispiel

G = ({S, B, C}, {a, b, c}, P, S)
S -> aSBC|aBC
[B, C -> b, c]
→ a^n (BC)^n = a^n BC BC …
CB -> BC
→ a^n B^n C^n (geht nicht kontextfrei)

$\left.\begin{array}{l} aB \text{ -> } ab \\ bB \text{ -> } bb \\ bC \text{ -> } bc \\ cC \text{ -> } cc \end{array}\right\}$ Erzeugung der [a’s] b’s, c‘s -> a^n b^n c^n

L(G) = {a^n b^n c^n | n ≥ 1}

Yes we can…

a_1
a_2
a_3

- $\mathbb{N}$ $\mathbb{Z}$ $\mathbb{R}$ $\mathbb{Q}$