Informatik Q3 Abels

Reguläre Ausdrücke

Reguläre Ausdrücke

Definition

Mit einem regulären Ausdruck wird (alternativ zur Tabelle oder zum Graphen) eine formale Sprache definiert.

Reguläre Ausdrücke

Beispiel: " $\boldsymbol{\Sigma = \{0, 1\}}$ "

Regulärer Ausdruck	Wortmenge / Formale Sprache	Beschreibung: " Die Sprache, die ..."
$\varnothing$	$\{\}$	... nichts enthält (leere Menge).
ϵ	$\{\epsilon\}$	... nur das leere Wort enthält.
0	$\{0\}$	... nur eine 0 enthält.
1	$\{1\}$	... nur eine 1 enthält.
10	$\{10\}$	... nur das Wort "10" enthält.
$1\|0$	$\{1,0\}$	... nur aus einer 0 und einer 1 besteht.
1^*	$\{\epsilon, 1, 11, 111, 1111, \dots\}$	... aus allen Wörtern mit beliebig vielen (auch gar keine) 1en besteht.
01^*	$\{0, 01, 011, 0111, 01111, \dots\}$	... aus allen Wörtern besteht, bei denen auf eine 0 beliebig viele (auch gar keine) 1en folgen.
0^*1^*	$\{\epsilon, 0, 00, \dots, 1, 01, 001, \dots, 11, 011, 0011, \dots\}$	... aus allen Wörtern besteht, bei denen auf beliebig viele (auch gar keine) 0en beliebig viele (auch gar keine) 1en folgen.
$0^*\|1^*$	$\{\epsilon, 0, 00, 000, \dots, 1, 11, 111, \dots\}$	... aus allen Wörtern mit entweder beliebig vielen (auch gar keinen) 0en oder beliebig vielen (auch gar keinen) 1en besteht.
$0\|(1(0\|1)^*)$	$\{0, 1, 10, 11, 100, 101, 110, 111, \dots\}$	... die Binärzahlen darstellt.
$1+$	$\{1, 11, 111, 1111, \dots\}$	... aus Wörtern nur aus 1en mit mindestens einer 1 besteht. (nicht verwechseln mit $\mathbf{1^*}$)
$1?$	$\{\epsilon, 1\}$	... nur aus 1 oder dem leeren Wort besteht. Das ? bedeutet, dass das Zeichen davor optional vorkommen kann.

Reguläre Ausdrücke

Regeln

- $\emptyset$ ist ein regulärer Ausdruck. Er beschreibt die leere Wortmenge $\{\}$.
- $\boldsymbol{\epsilon}$ ist ein regulärer Ausdruck. Er beschreibt die Wortmenge $\{\epsilon\}$, in der nur das leere Wort vorkommt.
- Für jedes $\boldsymbol{a} \in \Sigma$ ist $\boldsymbol{a}$ ein regulärer Ausdruck. Der reguläre Ausdruck $\boldsymbol{a}$ beschreibt die Wortmenge $\{a\}$.
- Wenn $\boldsymbol{\alpha}$ und $\boldsymbol{\beta}$ reguläre Ausdrücke sind, dann ist auch die Konkatenation $\boldsymbol{\alpha\beta}$ ein regulärer Ausdruck. Wenn $\boldsymbol{\alpha}$ die Wortmenge A und $\boldsymbol{\beta}$ die Wortmenge B beschreibt, dann beschreibt die Konkatenation $\boldsymbol{\alpha\beta}$ die Menge $\{ab|a \in A \; und \; b \in B\}$ aller Wörter, die mit einem Wort aus A beginnen und mit einem Wort aus B enden.
- Wenn $\boldsymbol{\alpha}$ und $\boldsymbol{\beta}$ reguläre Ausdrücke sind, dann ist auch die Alternative $\boldsymbol{\alpha|\beta}$ ein regulärer Ausdruck. Wenn $\boldsymbol{\alpha}$ die Wortmenge A und $\boldsymbol{\beta}$ die Wortmenge B beschreibt, dann beschreibt die Alternative $\boldsymbol{\alpha|\beta}$ die Menge $\{\omega|\omega \in A \; oder \; \omega \in B\}$ aller Wörter, die in A oder in B vorkommen.
- Wenn $\boldsymbol{\alpha}$ ein regulärer Ausdruck ist, dann ist auch die Iteration $\boldsymbol{\alpha^*}$ ein regulärer Ausdruck. Wenn $\boldsymbol{\alpha}$ die Wortmenge A beschreibt, dann beschreibt die Iteration $\boldsymbol{\alpha^*}$ die Menge A^* aller Wörter, die durch endlich häufiges Aneinanderfügen (auch keinmal) von Wörtern aus A entstehen.

Übung 1

Erkläre jeweils, welche Wortmenge beschrieben wird:

a) $R_1 = (halli)^*hallo$

b) $R_2 = Too^*r$

c) $R_3 = (5|6|7|8|9|10)(a|b|c|d)$

d) $R_4 = \big((0|1)(0|1|2|3|4|5|6|7|8|9)\big)|\big(2(0|1|2|3)\big){:}\,(0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9)$

Übung 1

a) $\mathrm{R}_1 = (halli)^* hallo$

$\mathrm{L}_1 = \{hallo, hallihallo, hallihallihallo, hallihallihallihallo, \dots\}$

b) $\mathrm{R}_2 = Too^* r$

$\mathrm{L}_2 = \{Tor, Toor, Tooor, Toooor, Tooooor, Toooooor, Tooooooor, \dots\}$

c) $\mathrm{R}_3 = (5|6|7|8|9|10)(a|b|c|d)$

$\mathrm{L}_3 = \{5a, 5b, 5c, 5d, 6a, 6b, 6c, 6d, 7a, 7b, 7c, 7d, 8a, 8b, 8c, 8d, 9a, \dots, 10d\}$

d) $\mathrm{R}_4 = \big((0|1)(0|1|2|3|4|5|6|7|8|9)\big)|\big(2(0|1|2|3)\big): (0|1|2|3|4|5)(0|1|2|3|4|5|6|7|8|9)$

$L_4 = \{00:00, \dots, 16:23, \dots, 21:45, \dots, 23:59\}$

Übung 2

Gib einen regulären Ausdruck zur Beschreibung der Menge aller vereinfachten E-Mail-Adressen, die nach den folgenden Regeln gebildet werden, an:

Eine vereinfachte E-Mail-Adresse besteht aus einem User-Namen gefolgt vom @-Symbol und einer Domain-Angabe. Der User-Name soll nur aus b´s bestehen. Die Domainangabe soll aus Subdomains mit einer Topleveldomain aufgebaut sein, die jeweils mit einem Punkt getrennt werden. Während die Subdomains aus beliebig vielen b´s bestehen, soll die Topleveldomain aus genau zwei b´s bestehen.

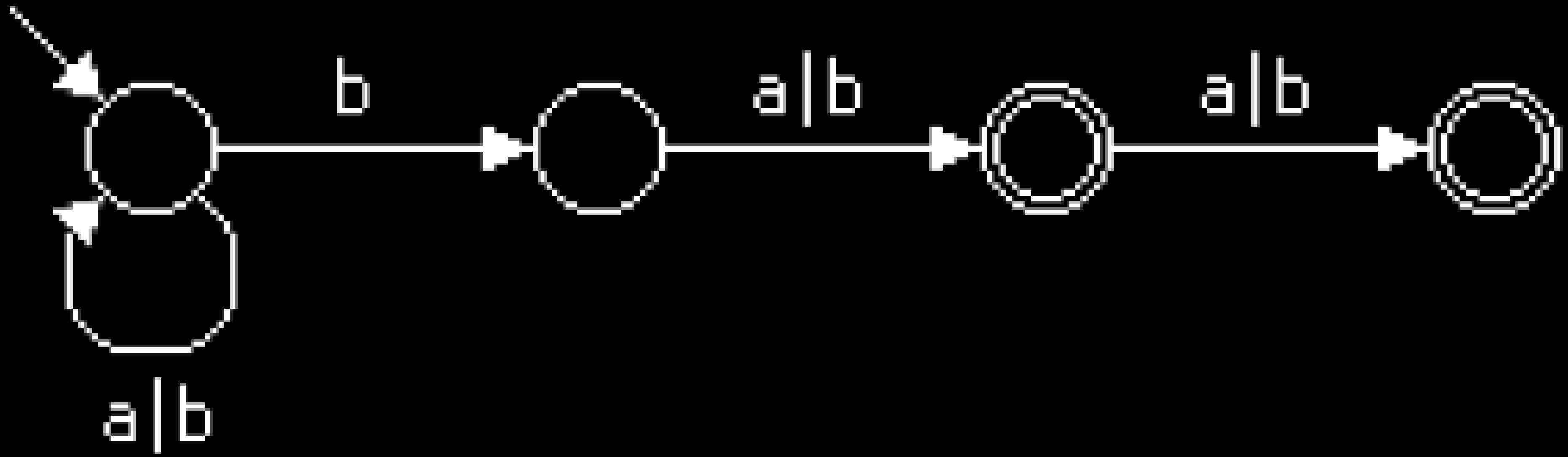
Beispiel: bb@b.bbb.bb

Übung 2

$$R = b + @(b+.)^{*}.bb$$

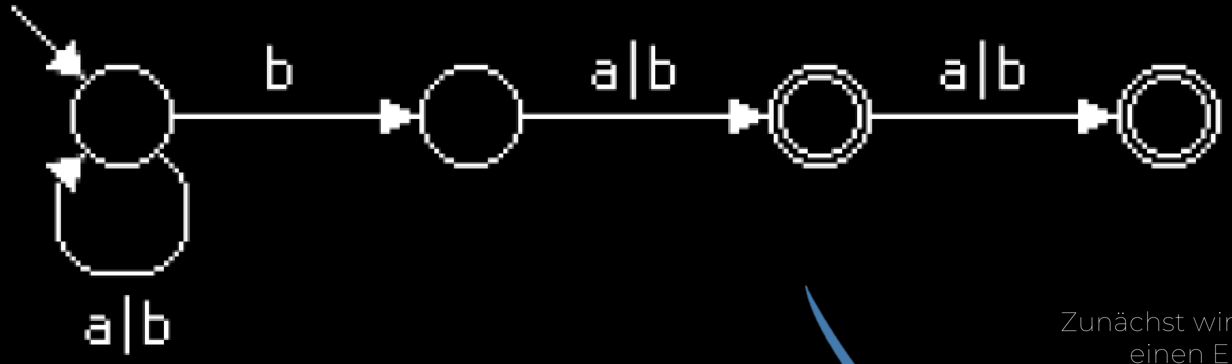
https://www.w3schools.com/tags/att_input_pattern.asp

Akzeptor → Regulärer Ausdruck

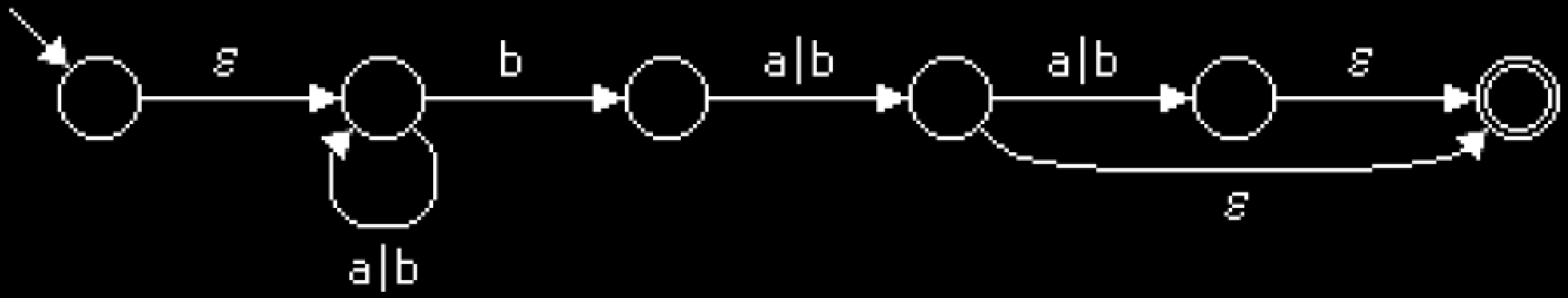


Akzeptor → Regulärer Ausdruck

1

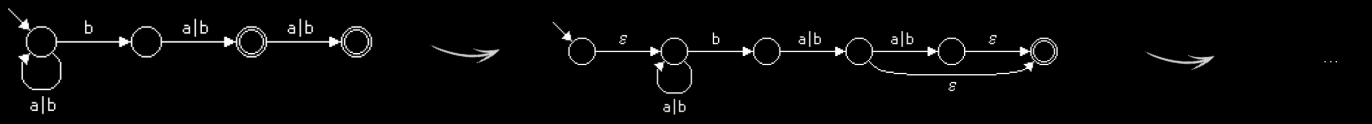


Zunächst wird ein neuer Startzustand eingeführt und durch einen Epsilon-Übergang mit dem ursprünglichen Startzustand verbunden. Ferner wird ein neuer Endzustand eingeführt. Alle bisherigen Endzustände verlieren ihre Endzustandseigenschaft; sie werden mit dem neuen Endzustand durch Epsilon-Übergänge verbunden.



Akzeptor → Regulärer Ausdruck

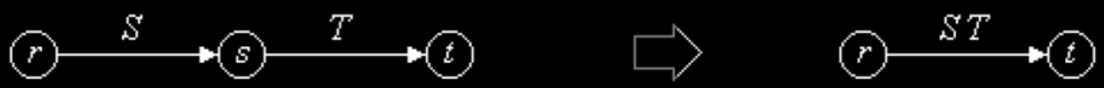
2



Außer dem Startzustand und dem Endzustand werden nun schrittweise alle Zustände nach folgenden Regeln entfernt. Wenn ein Zustand *s* entfernt wird, so wird jeder Kantenzug (*r*, *s*)(*s*, *t*) mit *r*, *t* ≠ *s* durch eine Kante (*r*,*t*) ersetzt. Sind hierbei die Kanten (*r*, *s*) und (*s*, *t*) mit den regulären Ausdrücken *S* bzw. *T* beschriftet, so wird die neue Kante (*r*,*t*) …

a) … mit dem regulären Ausdruck *ST* beschriftet, wenn keine Kante (*s*, *s*) vorhanden ist.

a)



b) … mit dem regulären Ausdruck *SX***T* beschriftet, wenn eine Kante (*s*, *s*) vorhanden ist, die mit dem regulären Ausdruck X beschriftet ist.

b)

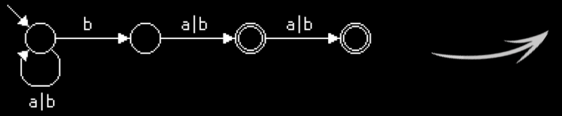
c)

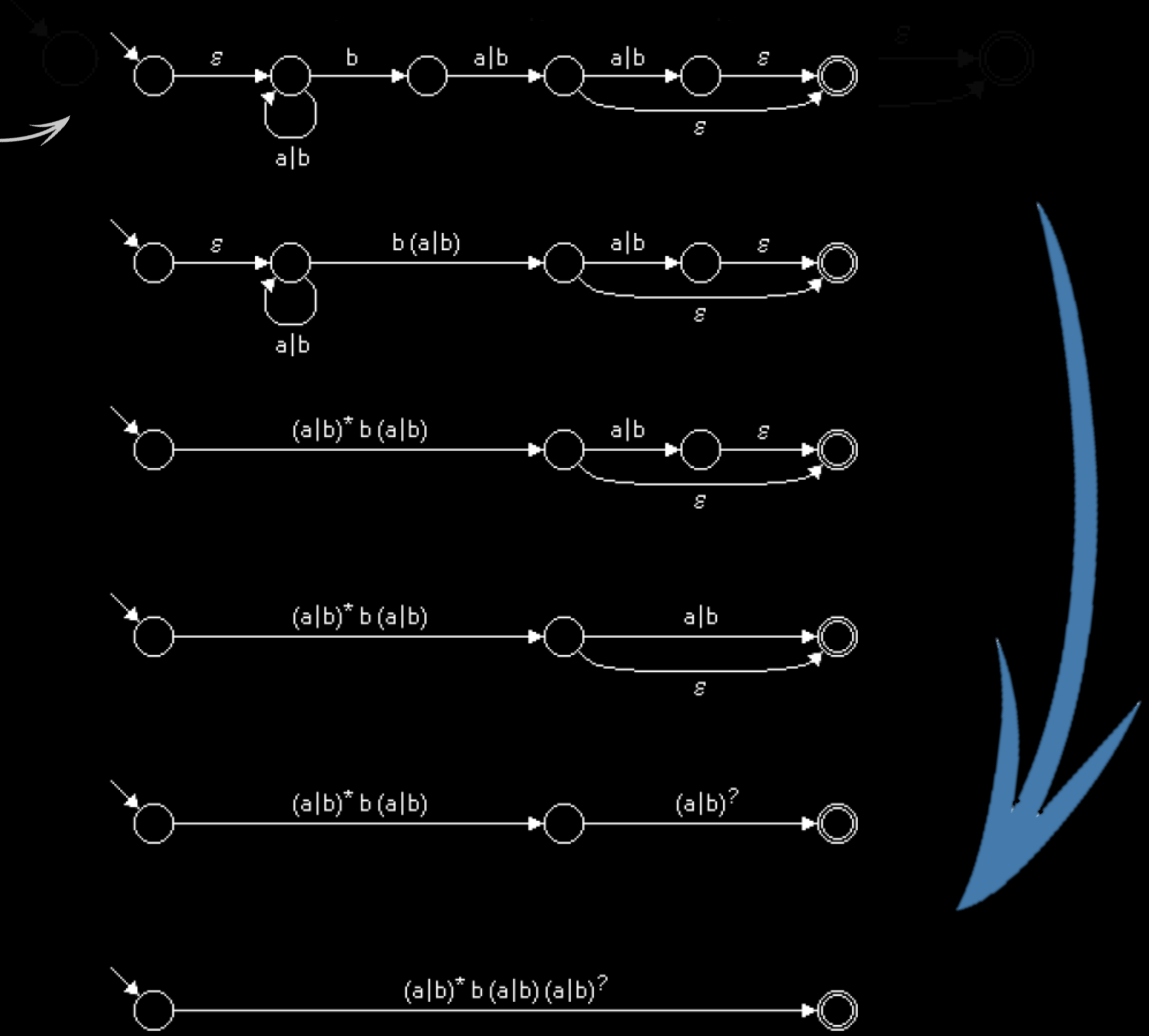
Ist nach dem Entfernen eines Zustands eine Doppelkante zwischen zwei Zuständen vorhanden, so wird diese durch eine einfache Kante ersetzt. Die Beschriftung dieser neuen Kante ist S|T, wenn S bzw. T die Beschriftungen der Doppelkante waren.

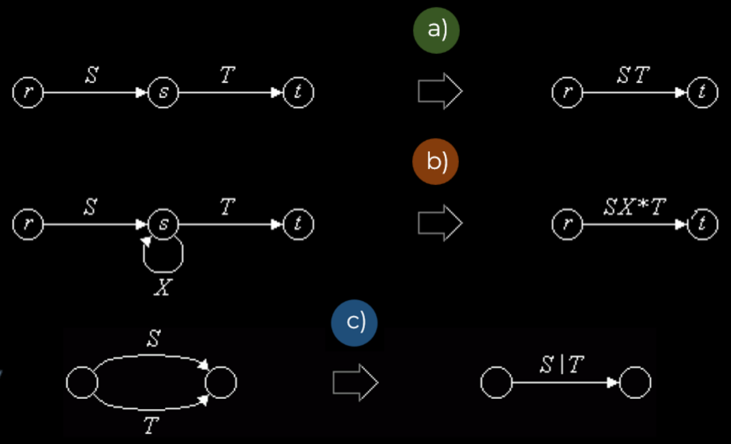
c)

Akzeptor → Regulärer Ausdruck

2

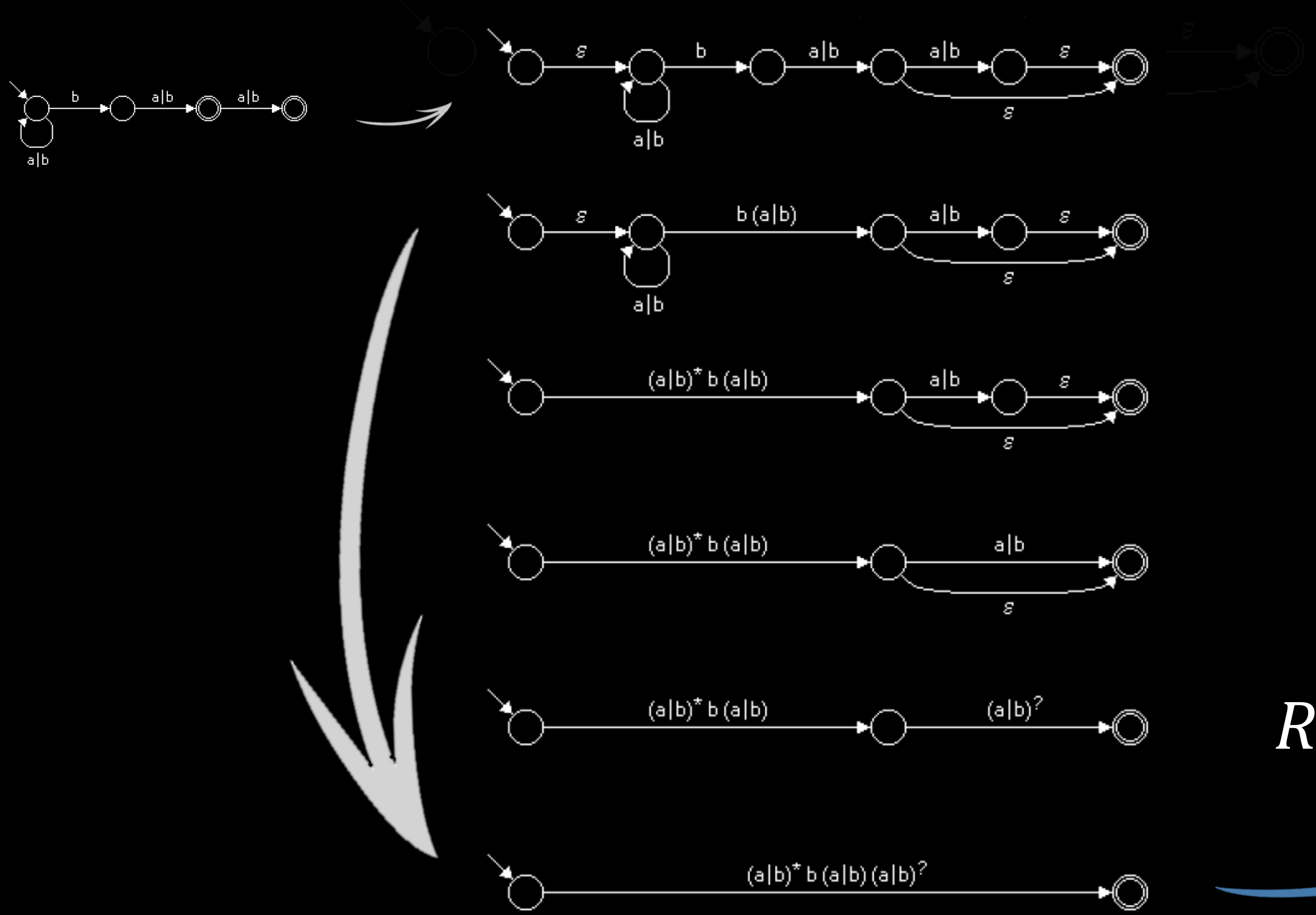






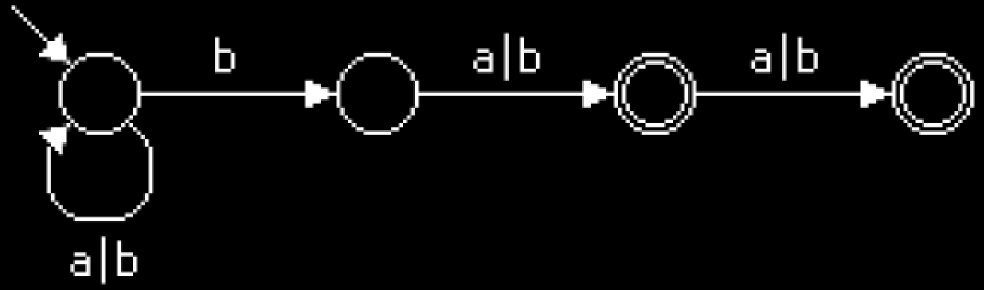
Akzeptor → Regulärer Ausdruck

3



$$R = (a|b)^{*}b(a|b)(a|b)?$$

Akzeptor → Regulärer Ausdruck

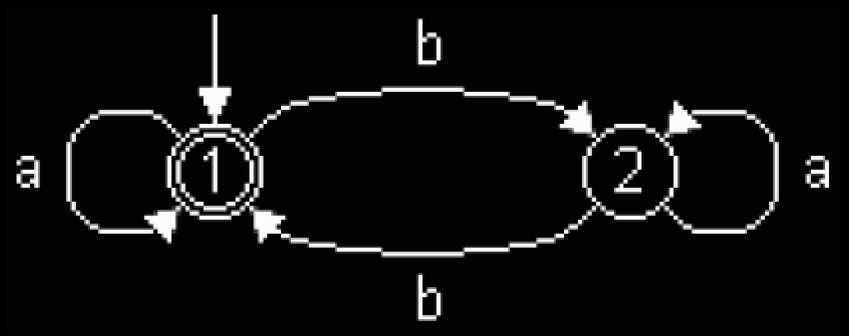


$$R = (a|b)^* b(a|b)(a|b)?$$

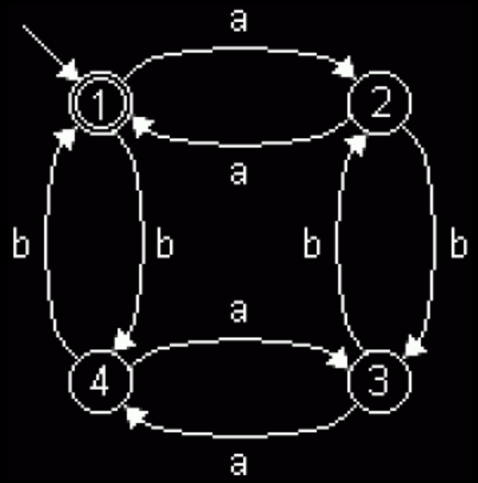
Übung 3

Erzeuge zu den gegebenen DEA's jeweils den zugehörigen regulären Ausdruck:

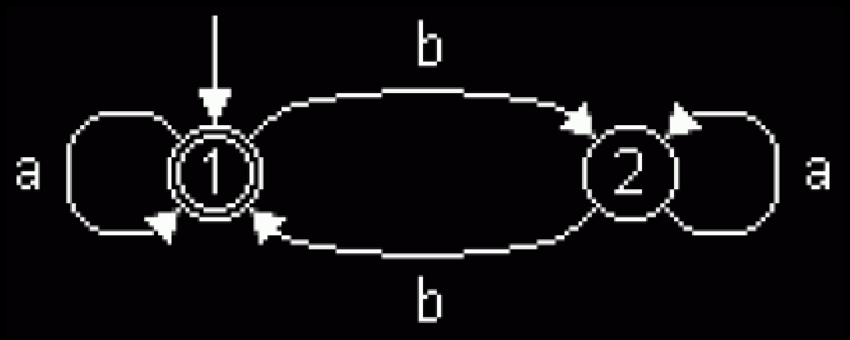
a)

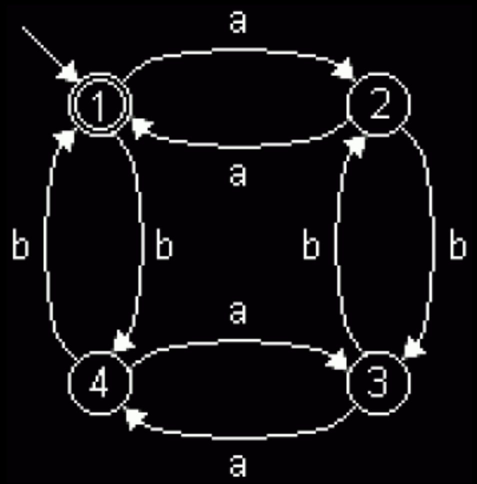


b)



Übung 3





$$R_a = (a|ba^*b)^*$$

Regulärer Ausdruck → Akzeptor

$$R = a(a|b)^*a|a$$

Regulärer Ausdruck → Akzeptor

1

$$R = a(a|b)^{*}a|a$$

Als erstes wird ein Zustandsgraph mit einem Startzustand und einem Endzustand erzeugt. Startzustand und Endzustand sind durch einen Zustandsübergang miteinander verbunden, der mit dem regulären Ausdruck R beschriftet ist (Bild 1). Dieser Zustandsgraph beschreibt offenbar das Verhalten eines Automaten, der die regulären Sprache L(R) erkennt – allerdings nur ganz grob, die inneren Zustände des Automaten bleiben zunächst unbestimmt.

Regulärer Ausdruck → Akzeptor

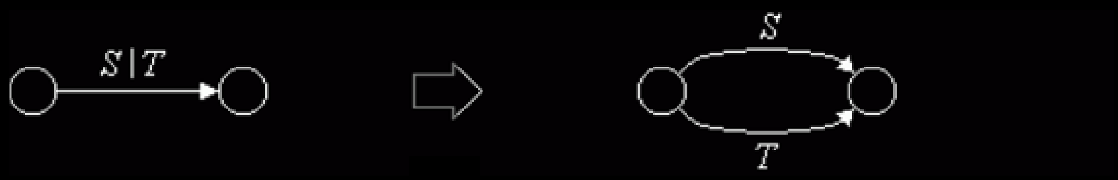
2

$R = a(a|b)^*a|a$

…

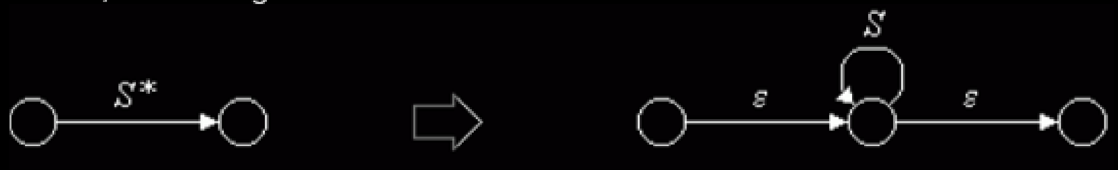
Durch Zerlegung dieses Zustandsübergangs werden nach und nach die inneren Zustände des Automaten erzeugt. Dabei gelten folgende Regeln:

a) Jeder Zustandsübergang, der mit $S|T$ beschriftet ist, wird in zwei parallele Zustands-übergänge zerlegt, die mit S bzw. mit T beschriftet sind.



b) Jeder Zustandsübergang, der mit ST beschriftet ist, wird in zwei aufeinander folgende Zustandsübergänge zerlegt, die mit S bzw. mit T beschriftet sind.

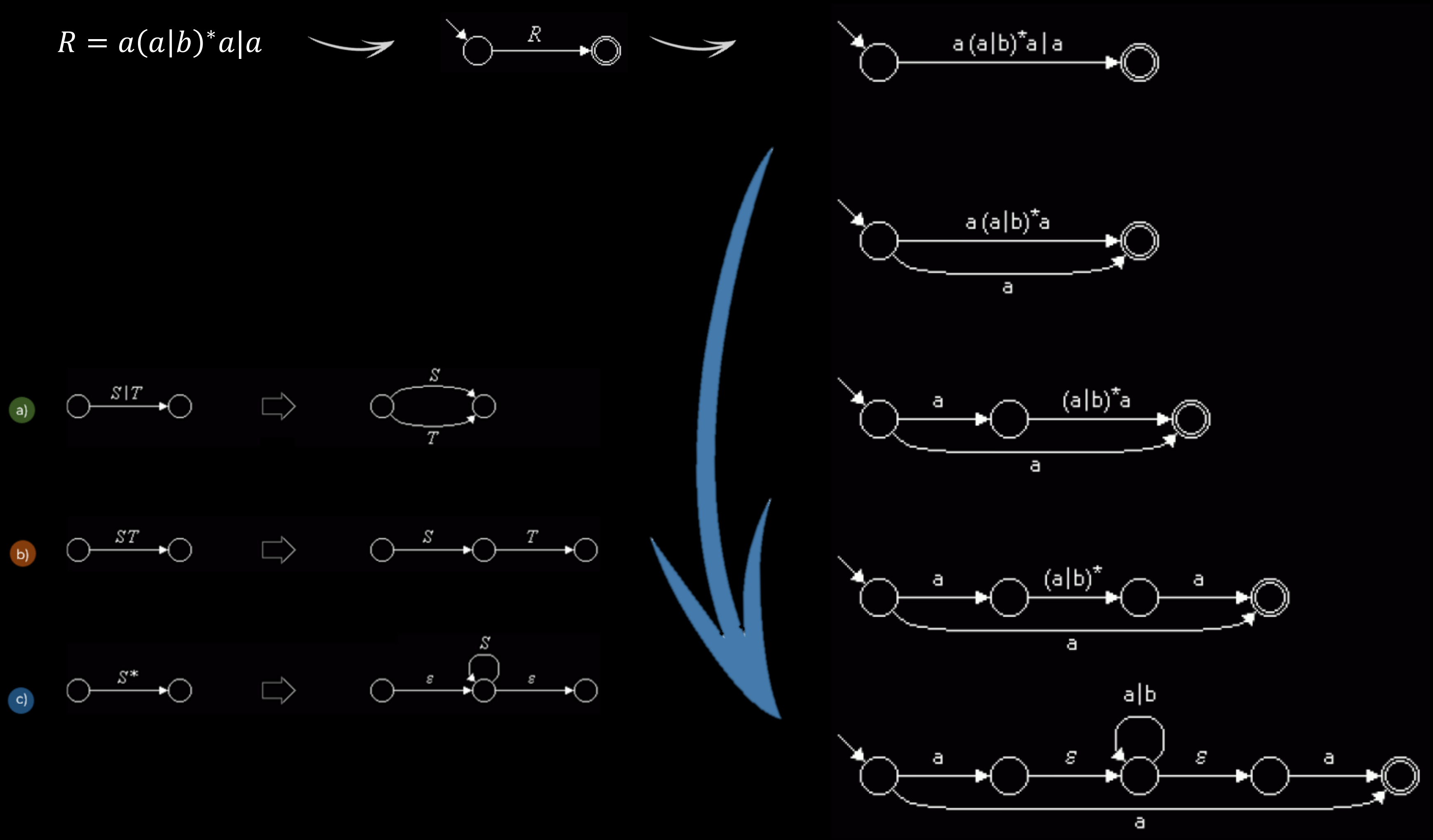
c) Jeder Zustandsübergang, der mit S^* beschriftet ist, wird zerlegt.



Regulärer Ausdruck → Akzeptor

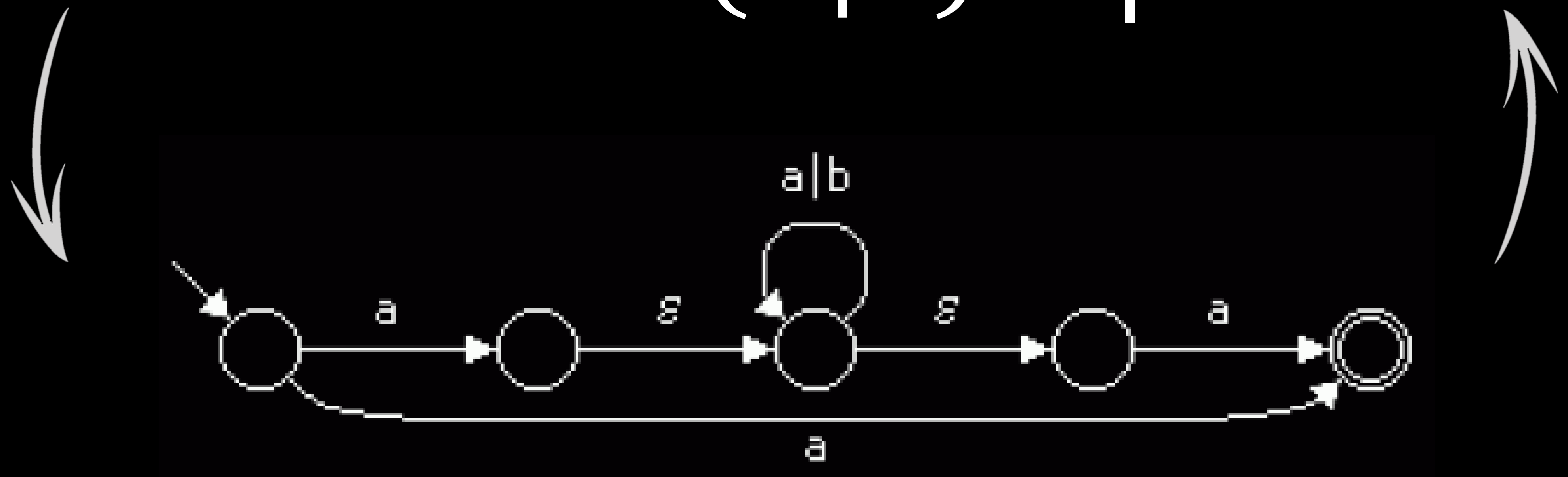
2

$R = a(a|b)^*a|a$



Regulärer Ausdruck → Akzeptor

$$R = a(a|b)^*a|a$$



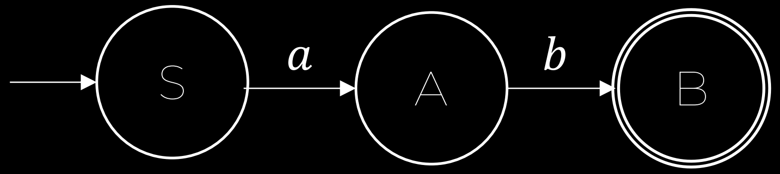
Übung 4

Erstelle zu den folgenden regulären Ausdrücken jeweils das zum Automaten gehörige Zustandsdiagramm:

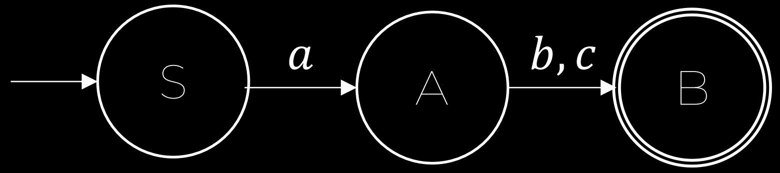
a) ab
b) $a(b|c)$
c) 10^*1
d) $0|1(0|1)^*$

Übung 4

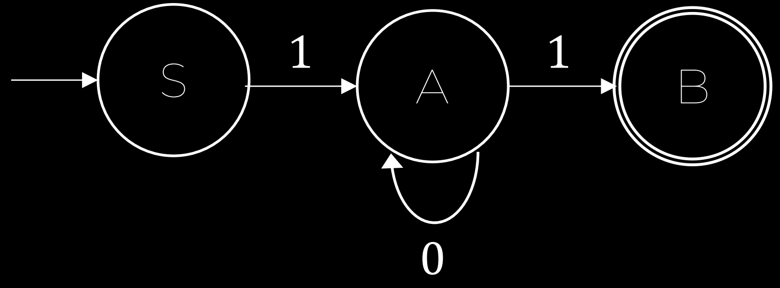
a) ab



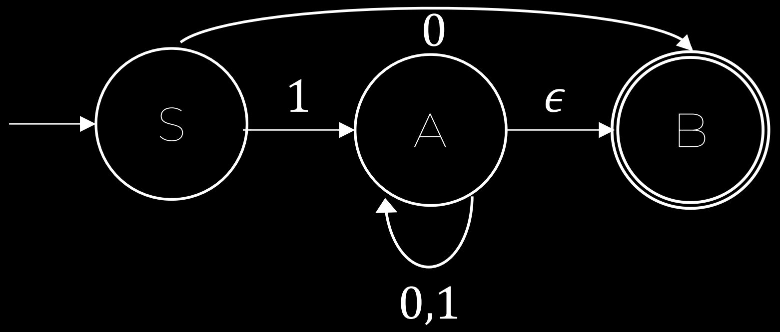
b) $a(b|c)$



c) 10^*1



d) $0|1(0|1)^*$



Satz

reguläre Sprache

Die Klasse der Sprachen, die mit einer regulären Grammatik beschrieben werden können, ist identisch mit der Klasse der Sprachen, die mit einem regulären Ausdruck beschrieben werden können. Sie ist ebenso identisch mit der Klasse der Sprachen, die von deterministischen endlichen Automaten bzw. von nichtdeterministischen endlichen Automaten erkannt werden können.

Tagebucheintrag

Reguläre Ausdrücke

Wochenübung

Konstruiere einen NEA aus folgendem regulären Ausdruck …

$$R = (a|ba|bba)^*(\epsilon|b|bb)$$

… über dem Alphabet $\Sigma = \{a, b\}$.

Die erzeugte Sprache umfasst alle Wörter über Σ, die höchstens zwei b's hintereinander enthalten.