Gliederung

Reguläre Ausdrücke – induktive Definition

- Es sei Σ das zugrunde liegende Alphabet.

Einfache reguläre Ausdrücke:

- Das Zeichen $\varnothing$ ist ein regulärer Ausdruck.
- [Das Zeichen ε ist ein regulärer Ausdruck.][1]
- Jedes Zeichen $a \in \Sigma$ ist ein regulärer Ausdruck.

[1]) verzichtbar, da durch $(\varnothing)^*$ ersetzbar.

Operatoren zum Bilden komplexerer regulärer Ausdrücke:

Es seien α und β reguläre Ausdrücke. Dann ist …

- die Zeichenkette $\alpha \circ \beta$ ein regulärer Ausdruck[2] (Verkettung),
- die Zeichenkette $(\alpha+\beta)$ ein regulärer Ausdruck[3] (Auswahl),
- die Zeichenkette $(\alpha)^*$ ein regulärer Ausdruck[4] (Kleene-Hülle).

Reguläre Ausdrücke in der Praxis (1)
Anwendungen

... werden verwendet zur Definition von „Mustern“ in:

- Unix
- Java
- PHP
- Perl
- Python
- Select
- ...

Reguläre Ausdrücke in der Praxis (1)
Schreibweisen – Varianten

Man sieht auch oft …

- $\alpha\beta$ anstelle von $\alpha \circ \beta$
- $(\alpha|\beta)$ anstelle von $(\alpha+\beta)$
- $(\alpha)^+$ anstelle von $\alpha \circ (\alpha)^*$.

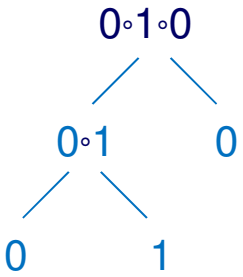
Reguläre Ausdrücke werden oft abgespeckt:

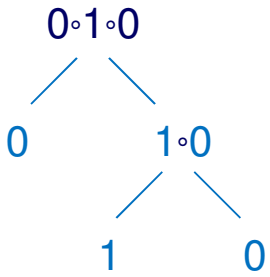
- äußerste Klammern weglassen, z.B. $\alpha+\alpha\circ\beta$ für $(\alpha+\alpha\circ\beta)$
- Doppelklammerpaare nur einfach, z.B. $(\alpha+\beta)^*$ für $((\alpha+\beta))^*$
- Prioritäten erlauben Klammerersparnis:
  - Oft * *vor* Verkettung *vor* Auswahl.
    Dann z.B. $0\circ1^*$ [oder (s.o.) 01^*] anstelle von $(0\circ1)^*$.
- Klammern in $\circ$ - Verkettungen weglassen, z.B. $0\circ1\circ0$ [oder 010] für $((0\circ1)\circ0)$

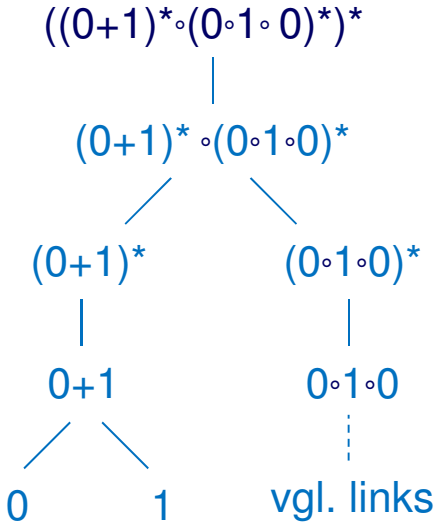
Beispiele für reguläre Ausdrücke

- Es sei $\Sigma = \{ 0,1 \}$ das zugrunde liegende Alphabet.

Beispiele (mit induktivem Aufbau, äußerste Klammern weggelassen):







Erinnerung: Operationen über Sprachen

- Es seien Σ das zugrunde liegende Alphabet und $L_1, L_2 \subseteq \Sigma^*$.

  Dann definieren wir:

$L_1 \circ L_2 = \{ u \circ v \mid u \in L_1, v \in L_2 \}$ (Verkettung von L_1 und L_2)

$(L_1)^* = \{ \varepsilon \} \cup \{ u_1 \circ \ldots \circ u_k \mid k \geq 1, u_1 \in L_1, \ldots, u_k \in L_1 \}$ (Kleene-Hülle von L_1)

Klammereinsparung wie bei regulären Ausdrücken möglich.

Durch reguläre Ausdrücke beschriebene Sprachen

- Es sei Σ das zugrunde liegende Alphabet.
- Zu jedem regulären Ausdruck γ bezeichnet $L(\gamma)$ die von γ beschriebene Sprache.

Einfache reguläre Ausdrücke ($a \in \Sigma$):

- $L(\varnothing) = \varnothing$
- $L(a) = \{ a \}$
- ($L(\varepsilon) = \{ \varepsilon \}$)

Komplexere regulärer Ausdrücke (α, β *reg. Ausdr.*):

- $L(\alpha \circ \beta) = L(\alpha) \circ L(\beta)$ (Verkettung)
- $L((\alpha+\beta)) = L(\alpha) \cup L(\beta)$ (Auswahl)
- $L((\alpha)^*) = (L(\alpha))^*$ (Kleene-Hülle)

- Beispiele für durch reguläre Ausdrücke beschriebene Sprachen

Es sei $\Sigma = \{ 0,1 \}$ das zugrunde liegende Alphabet.

- $L(010) = \{ 010 \}$
- $L((0+1)^*) = \Sigma^*$
- $L(\varnothing^*) = \{\varepsilon\}$
- $L((0+1)^*1) = \{ w \circ 1 \mid w \in \Sigma^* \}$
- $L((0+1)^*101(0+1)^*) = \{ v \circ 101 \circ w \mid v,w \in \Sigma^* \}$
- $L(1^*0\,(1^*01^*0)^*\,1^*)$ = Sprache aller 0/1-Folgen mit ungerade vielen 0en

Zwischenbilanz

- Reguläre Ausdrücke sind ein in der Praxis weit verbreitetes Mittel, um formale Sprachen zu beschreiben.
- Reguläre Ausdrücke lassen sich oft „einfacher“ notieren als Grammatiken (ein wenig Übung vorausgesetzt ...).

- Noch zu klären bleibt ...
  - welche formalen Sprachen sich mithilfe von regulären Ausdrücken beschreiben lassen;
  - ob man das Wortproblem für diese Sprachen lösen kann und, wenn ja, wie effizient das möglich ist.

- zentrales Resultat (die Antwort auf beide Fragen)

Es seien Σ ein endliches Alphabet und $L \subseteq \Sigma^*$.

Dann sind die folgenden Aussagen äquivalent:

1. L ist eine reguläre Sprache.
2. Es gibt einen regulären Ausdruck α mit $L(\alpha) = L$.

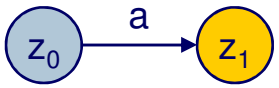
Es handelt sich wieder um zwei Implikationen.

- Implikation reg. Ausdruck → reg. Sprache (Überblick)

  - Das ist der einfachere Teil, da wir beim Beweis davon Gebrauch machen können, dass die Menge der regulären Ausdrücke induktiv definiert ist.
  - Grundidee:
    - Wir zeigen, dass jede durch einen **einfachen regulären** Ausdruck α definiert Sprache $L(\alpha)$ regulär ist (trivial, wie wir gleich sehen).
    - Beim Nachweis, dass die durch einen komplexeren regulären Ausdruck $\gamma = \alpha\beta$, $\gamma = (\alpha+\beta)$ bzw. $\gamma = (\alpha)^*$ definierte Sprache $L(\gamma)$ regulär ist, benutzen wir dann, dass $L(\alpha)$ und $L(\beta)$ regulär sind.
    - Es würde genügen, wenn für beliebige reguläre Sprachen L_1 und L_2 die Sprachen $L = L_1L_2$, $L = L_1 \cup L_2$ und $L = L_1^*$ regulär wären. Da wir die **Abgeschlossenheit** gegenüber Vereinigung bereits gezeigt haben, ist diese jetzt nur noch gegenüber **Verkettung und Hüllenoperation** zu zeigen.

- ... für (einfache) reguläre Ausdrücke
  - nichtdeterministischer endlicher Automat für …
    - ε: 

    - $\emptyset$: 

    - a mit $a \in \Sigma$: 

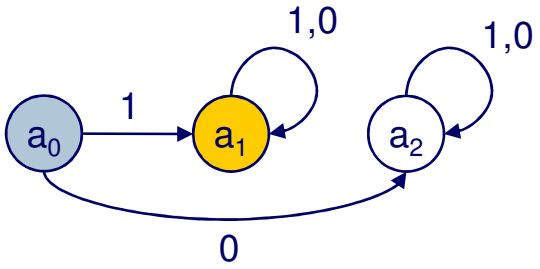
Verkettung – am Beispiel

- Es seien $\Sigma = \{ 0,1 \}$,
- $L_1 = \{ 1 \circ w \mid w \in \Sigma^* \}$, die Wörter mit Präfix 1, und
- $L_2 = \{ w \circ 0 \mid w \in \Sigma^* \}$, die Wörter mit Suffix 0.

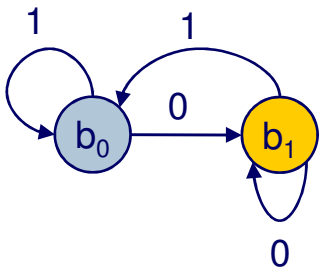
Frage: *Ist die Sprache L_1L_2 auch regulär?*
(L_1L_2 enthält alle Wörter aus Σ^*,
die das Präfix 1 und das Suffix 0 haben.)

- Verkettung – am Beispiel

A_1 mit $L(A_1) = L_1$

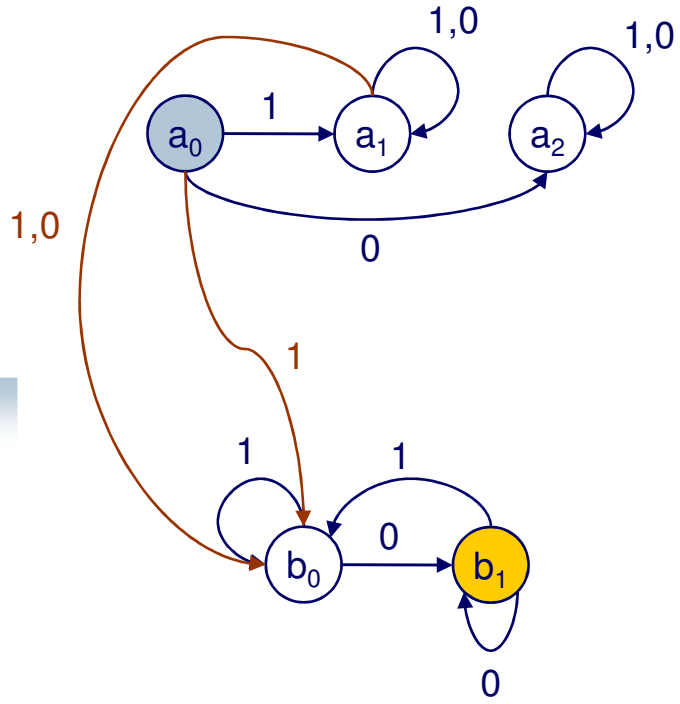


A_2 mit $L(A_2) = L_2$



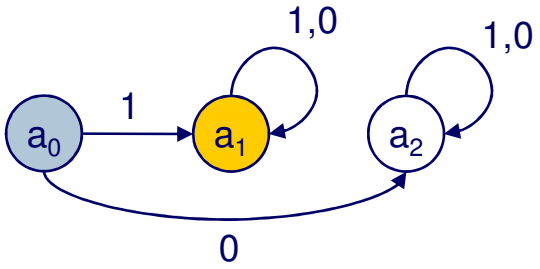
Was geschieht hier?

nichtdet. endl. Automat B
mit $L(B) = L_1L_2$

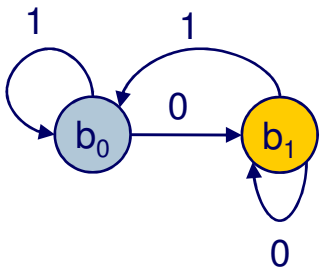


Verkettung – am Beispiel

A_1 mit $L(A_1) = L_1$

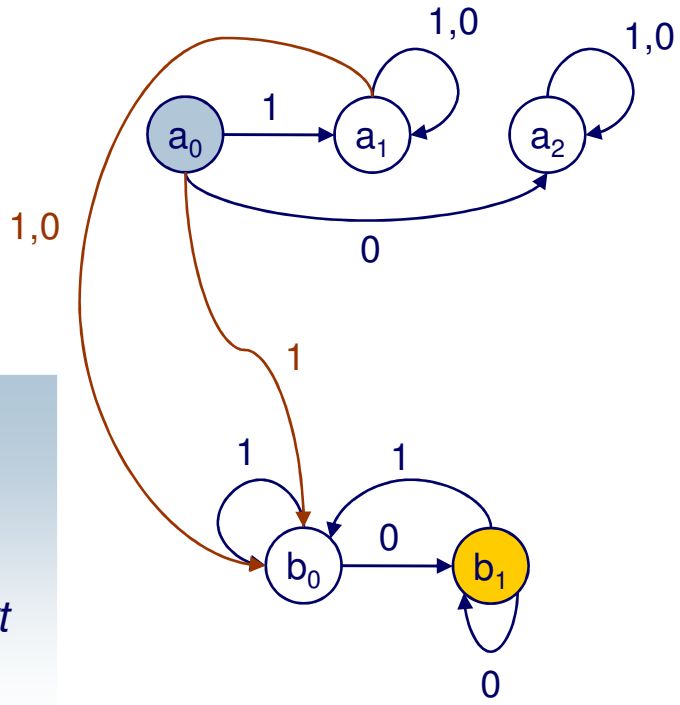


A_2 mit $L(A_2) = L_2$



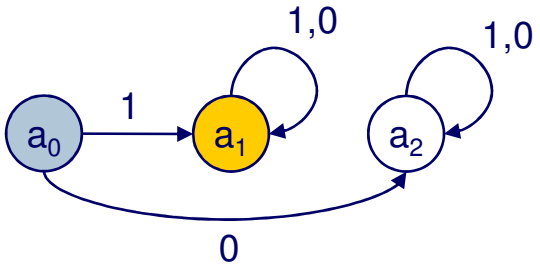
In A_1 wird nicht mehr akzeptiert, in A_2 nicht mehr gestartet. Was in A_1 zum Akzeptieren führte, führt nun auch zum ehemaligen Start von A_2.

nichtdet. endl. Automat B
mit $L(B) = L_1L_2$

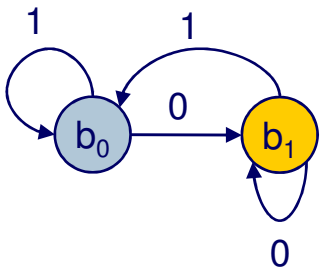


Exkurs: Elegante Alternative

A_1 mit $L(A_1) = L_1$

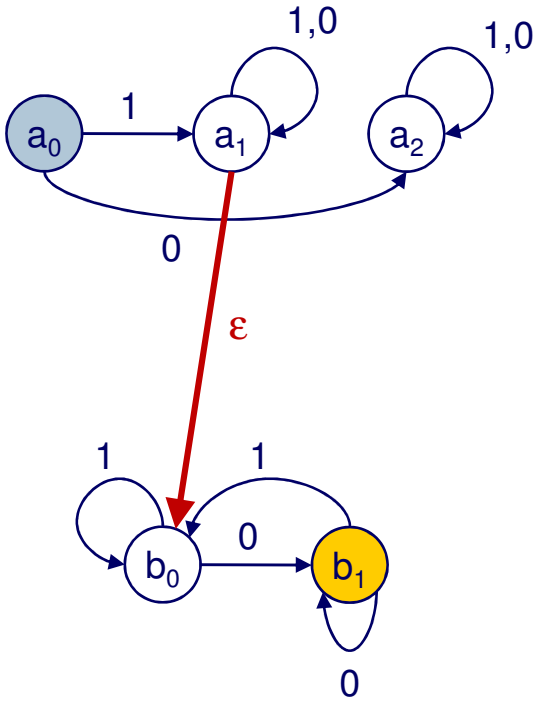


A_2 mit $L(A_2) = L_2$



sog. ε-Automat, macht evtl. spontane Übergänge, ohne ein Zeichen zu verarbeiten

nichtdet. endl. Automat B mit $L(B) = L_1L_2$



Beweisidee für die Operation Verkettung

- Sei $A_1 = [Z,\Sigma,z_0,F_1,\delta_1]$ ein (deterministischer) endlicher Automat für L_1
- Sei $A_2 = [Q,\Sigma,q_0,F_2,\delta_2]$ ein (deterministischer) endlicher Automat für L_2
- Es gelte $Z \cap Q = \varnothing$.
- Definiere den nichtdeterministischen endlichen Automaten $A' = [Z',\Sigma,z'_0,F',\Delta']$ wie folgt:

  - $Z' = Z \cup Q$
  - $z'_0 = z_0$
  - $F' = F_2$
  - $\Delta' = \Delta'_1 \cup \Delta'_2 \cup \Delta'_3$ mit:
    - $\Delta'_1 = \{ (z,a,\delta_1(z,a)) \mid z \in Z \text{ und } a \in \Sigma \}$
    - $\Delta'_2 = \{ (q,a,\delta_2(q,a)) \mid q \in Q \text{ und } a \in \Sigma \}$
    - $\Delta'_3 = \{ (z,a,q_0) \mid z \in Z, a \in \Sigma \text{ und } \delta_1(z,a) \in F_1 \}$

- Man kann sich nun leicht überlegen, dass $L(A') = L(A_1) \circ L(A_2)$ gilt.

Reguläre Grammatik für $L(G_1) \circ L(G_2)$ am Beispiel

- Es seien $\Sigma = \{ 0,1 \}$,
- $L_1 = \{ 1 \circ w \mid w \in \{0\}^* \}$, die Wörter mit Präfix 1 und dann höchstens 0en,
- $L_2 = \{ w \circ 0 \mid w \in \{1\}^* \}$, die Wörter mit Suffix 0 und davor höchstens 1en.
- Die generieren wir z.B. mit Regelmengen

R1:
$S \rightarrow 1 \mid 1A$
$A \rightarrow 0 \mid 0A$

R2:
$S \rightarrow 0 \mid 1S$

Welche Regelmenge generiert L_1L_2? (Ad-hoc-Übung)

R12:
$S \rightarrow 1T \mid 1A$
$A \rightarrow 0T \mid 0A$
$T \rightarrow 0 \mid 1T$

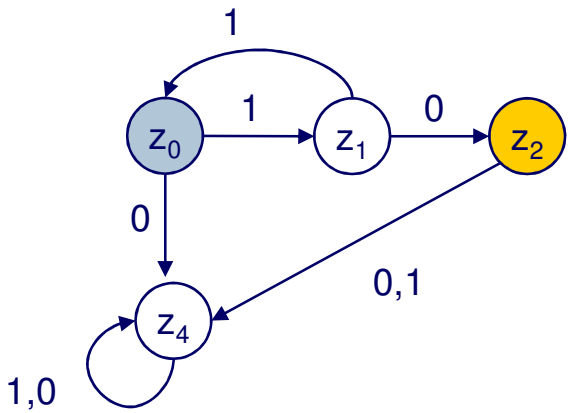
1. R2 mit „neuen Variablennamen“

2. R2-Start nach jedem „R1-Stop“

Welche Regelmenge generiert $(L_1L_2)\backslash\{\varepsilon\}$ für beliebige reguläre L_1, L_2? → Übung 4

Kleenesche Hülle – am Beispiel

- Es sei $\Sigma = \{ 0,1 \}$.
- Es sei $L = \{ 1^{2n-1} \circ 0 \mid n \geq 1 \}$, die Wörter aus ungerade vielen Einsen plus einer Null am Ende.
- Ein Automat für L

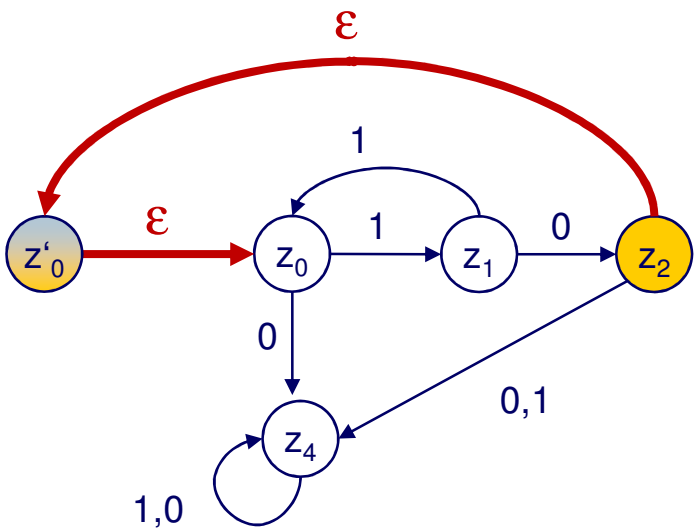


Für L* würde man gerne

- evtl. gleich am Start ε akzeptieren (aber nicht im z_0 - sonst 11 akzeptiert),
- evtl. vom akzeptierenden z_2 aus sofort wieder beginnen.

Kleenesche Hülle – am Beispiel

- Es sei $\Sigma = \{ 0,1 \}$.
- Es sei $L = \{ 1^{2n-1} \circ 0 \mid n \geq 1 \}$, die Wörter aus ungerade vielen Einsen plus einer Null am Ende.
- Ein **ε-Automat** für L^*



Jetzt würde man gerne

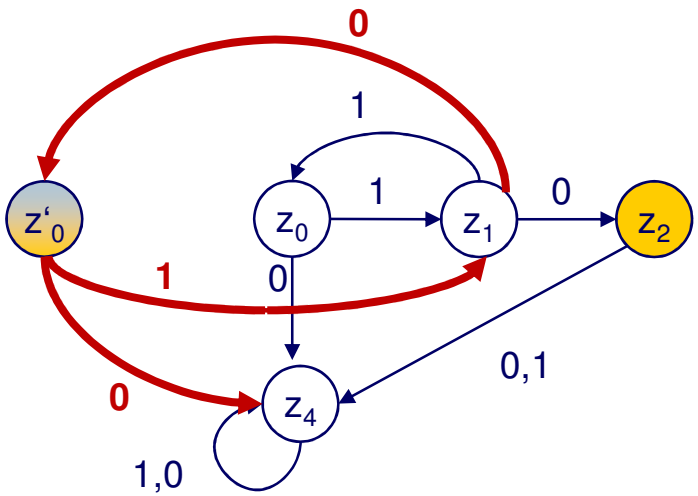
- das gleiche ohne ε-Übergänge erreichen,

wozu man

- die ε-Übergänge evtl. „überspringt“.

Kleenesche Hülle – vom Beispiel zur Konstruktionsvorschrift

- Es sei $\Sigma = \{ 0,1 \}$.
- Es sei $L = \{ 1^{2n-1} \circ 0 \mid n \geq 1 \}$, die Wörter aus ungerade vielen Einsen plus einer Null am Ende.
- Ein **ND-Automat** für L*



Der ND-Automat kann sogar noch abgespeckt werden. Wie?

Rezept
Vom endlichen Automaten
$A = [Z,\Sigma,z_0,F,\delta]$ zum NDA
$A' = [Z',\Sigma,z'_0,F', \Delta']$ mit $L(A')=L(A)^*$

$Z' := Z \cup \{ z'_0 \}$
neuer Startzustand := z'_0
$F' := F \cup \{ z'_0 \}$
$\Delta' := \Delta'_1 \cup \Delta'_2 \cup \Delta'_3$ mit:
$\Delta'_1 = \{ (z,a,\delta(z,a)) \mid z \in Z \text{ und } a \in \Sigma \}$
$\Delta'_2 = \{ (z'_0,a,\delta(z_0,a)) \mid a \in \Sigma \}$
$\Delta'_3 = \{ (z,a,z'_0) \mid z \in Z, a \in \Sigma \text{ und } \delta(z,a) \in F \}$

- Implikation reg. Sprache → reg. Ausdruck

  - Das wäre der kompliziertere Teil …
  - … wenn wir nicht bereits aus der Graphentheorie das Zwischenknoten-Lemma für Wege kennen würden.
  - Trotzdem bleibt die manuelle algorithmische Umsetzung i.a. langwierig.

Es sei A eine endlicher Automat für eine reguläre Sprache L .

Wir werden zeigen, dass man die Menge der akzeptierten Wörter mit Hilfe von regulären Ausdrücken beschreiben kann.

- Implikation reg. Sprache → reg. Ausdruck (Hilfsbegriff)

  - Es sei Σ das zugrunde liegende Alphabet.
  - Es sei A ein endlicher Automat mit der Zustandsmenge Z = { 1,...,n }, dem Anfangszustand 1 und der Zustandsüberführungsfunktion $\delta(.,.)$.
  - Es seien i,j,k natürliche Zahlen mit $1 \le i,j \le n$ und $0 \le k \le n$

$$R_{i,j,k} = \{ w \in \Sigma^* \mid \delta^*(i,w) = j \text{ und bei der Verarbeitung von w wird kein „Zwischenzustand“ } > k \text{ besucht } \}$$

... d.h. es gibt in A einen mit w markierten Weg vom Zustand i zum Zustand j, in dem nur die „Zwischenzustände“ 1,...,k vorkommen … also im Spezialfall k=0: gar keiner, d.h. bei m Kanten von i nach j ($0 \le m \le n$):

  - $R_{i,j,0} = \{\varepsilon\} \cup \{x \in \Sigma \mid \delta(i,x)=j \} = L(\varepsilon + x_1 + x_2 + \ldots + x_m)$

Also ist $R_{i,j,0}$ Sprache eines regulären Ausdrucks.

- Implikation reg. Sprache → reg. Ausdruck
  (warum der Hilfsbegriff sinnvoll ist)

  - Es sei Σ das zugrunde liegende Alphabet.
  - Es sei A ein endlicher Automat mit der Zustandsmenge $Z = \{ 1,...,n \}$, dem Anfangszustand 1 und der Zustandsüberführungsfunktion $\delta(.,.)$ und $F \subseteq Z$ als Menge der akzeptierenden Zustände von A.

Beobachtung 1:

Es gilt: $L(A) = \bigcup_{j \in F} R_{1,j,n}$.

Beobachtung 2:

Falls es für jede Menge $R_{1,j,n}$ auf der rechten Seite
einen regulären Ausdruck α' mit $L(\alpha') = R_{1,j,n}$ gibt,
so gibt es auch einen regulären Ausdruck α mit $L(\alpha) = L(A)$.

- Implikation reg. Sprache → reg. Ausdruck
  (warum der Hilfsbegriff sinnvoll ist)

  - Es seien i,j,k mit $1 \le i,j \le n$ und $0 \le k \le n-1$.

Beobachtung 3:

Es gilt $R_{i,j,k+1} = R_{i,j,k} \cup R_{i,k+1,k} (R_{k+1,k+1,k})^* R_{k+1,j,k}$

Wie beim Zwischenknoten-Lemma ist ein „Wort von i nach j“ mit Zwischenzuständen ≤ k+1 entweder

- *ein „Wort von i nach j“ mit „Zwischenzuständen“ ≤ k oder*
- *zerlegbar in*
  a) ein „Wort von i nach k+1“ mit „Zwischenzuständen“ ≤ k,
  b) eine (eventuell leere) Folge von „Worten von k+1 nach k+1“ mit „Zwischenzuständen“ ≤ k und
  c) ein „Wort von k+1 nach j“ mit „Zwischenzuständen“ ≤ k .

- Implikation reg. Sprache → reg. Ausdruck
  (warum der Hilfsbegriff sinnvoll ist)

  - Es seien i,j,k mit $1 \leq i,j \leq n$ und $0 \leq k \leq n-1$.

Beobachtung 3:

Es gilt $R_{i,j,k+1} = R_{i,j,k} \cup R_{i,k+1,k} (R_{k+1,k+1,k})^* R_{k+1,j,k}$

Beobachtung 4:

Falls es für jede Menge R' auf der rechten Seite einen regulären Ausdruck α' mit $L(\alpha') = R'$ gibt, so gibt es auch einen regulären Ausdruck α mit $L(\alpha) = R_{i,j,k+1}$.

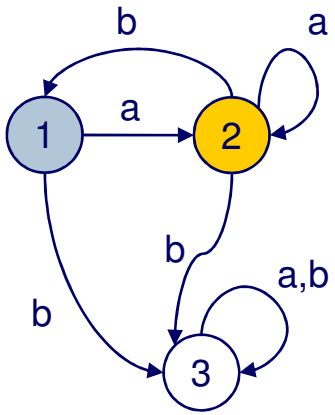
Beobachtung 5:

Wiederholte Anwendung auf die rechten Seiten („Auflösung“) führt von Sprachen $R_{i,j,k}$ zu regulären Sprachen $R_{i,j,0}$.

- Implikation reg. Sprache → reg. Ausdruck
  (warum der Hilfsbegriff sinnvoll ist)

  - Es sei $\Sigma = \{ a,b \}$.
  - Es sei $L_3 = L(A_3)$

A_3



$$L(A) = R_{1,2,3} = R_{1,2,2} \cup R_{1,3,2} (R_{3,3,2})^* R_{3,2,2}$$
$$= R_{1,2,2} \quad (\text{ da } R_{3,2,2} = \varnothing \text{ ! })$$

$$R_{1,2,2} = R_{1,2,1} \cup R_{1,2,1} (R_{2,2,1})^* R_{2,2,1}$$

$$R_{1,2,1} = R_{1,2,0} \cup R_{1,1,0} (R_{1,1,0})^* R_{1,2,0}$$

$$R_{2,2,1} = R_{2,2,0} \cup R_{2,1,0} (R_{1,1,0})^* R_{1,2,0}$$

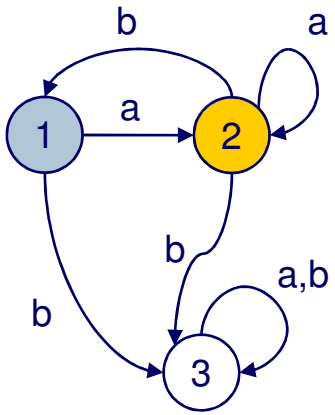
- Implikation reg. Sprache → reg. Ausdruck
  (Beispiel für Auflösung)

  - Es sei $\Sigma = \{ a,b \}$.
  - Es sei $L_3 = L(A_3)$

$R_{1,1,0} = \{ \varepsilon \}$ $R_{1,2,0} = \{ a \}$

$R_{2,1,0} = \{ b \}$ $R_{2,2,0} = \{ \varepsilon,a \}$

A_3



$$R_{1,2,1} = R_{1,2,0} \cup R_{1,1,0} (R_{1,1,0})^* R_{1,2,0}$$
$$= \{ a \} \cup \{ \varepsilon \} (\{ \varepsilon \})^* \{ a \} = \{ a \}$$

$$R_{2,2,1} = R_{2,2,0} \cup R_{2,1,0} (R_{1,1,0})^* R_{1,2,0}$$
$$= \{ \varepsilon,a \} \cup \{ b \} (\{ \varepsilon \})^* \{a \} = \{ \varepsilon,a,ba \}$$

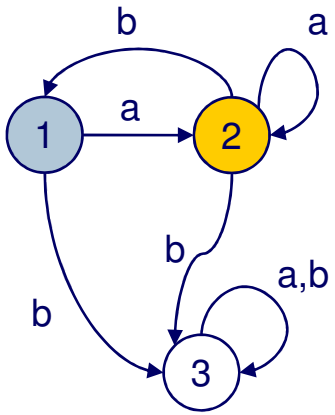
$$L(A) = R_{1,2,2} = R_{1,2,1} \cup R_{1,2,1} (R_{2,2,1})^* R_{2,2,1}$$
$$= \{ a \} \cup \{ a \}(\{ \varepsilon,a,ba \})^* \{ \varepsilon,a,ba \}$$

- Implikation reg. Sprache → reg. Ausdruck
  (Beispiel für Auflösung, Vereinfachen, Umwandeln)

  - Es sei $\Sigma = \{ a,b \}$.
  - Es sei $L_3 = L(A_3)$

$$L(A) = R_{1,2,2} = R_{1,2,1} \cup R_{1,2,1} (R_{2,2,1})^* R_{2,2,1}$$
$$= \{ a \} \cup \{ a \}(\{ \varepsilon,a,ba \})^* \{ \varepsilon,a,ba \}$$

A_3



Es gilt $L(A) = L(a + (a(\varepsilon + a + ba)^* (\varepsilon + a + ba)).$
$= L(a (a + ba)^*)$

wie man überlegt

Beobachtung 6:

Ohne die Abkürzungen und Vereinfachungen zwischendurch ...

- *Wie viele Folien mehr hätten wir gebraucht?*
- *Wie lang wäre der reg. Ausdruck geworden?*

- Zwei angenehme Nachrichten

  - **Erste gute Nachricht:**

    Wir haben es geschafft und uns klar gemacht, dass endliche Automaten und reguläre Ausdrücke (und deshalb auch dazu noch nichtdeterministische Automaten und reguläre Grammatiken) dieselbe Menge von Sprachen beschreiben.

    Die zweite Beweisrichtung war dabei etwas unangenehm.

  - **Zweite gute Nachricht:**

    Zur zweiten Beweisrichtung mit dem Zwischenknotenlemma verzichten wir auf Übungen und Klausuraufgaben.