# Formale Grundlagen der Informatik 1
## Kapitel 6
## *Abschlusseigenschaften und Grenzen regulärer Sprachen*

Frank Heitmann
heitmann@informatik.uni-hamburg.de

19. April 2016





## Die Familie der regulären Sprachen

**Satz**

Zu jeder regulären Sprache $L$ gibt es

- einen DFA $A$ mit $L(A) = L$
- einen NFA $B$ mit $L(B) = L$
- einen NFA mit $\lambda$-Kanten $C$ mit $L(C) = L$
- einen regulären Ausdruck $D$, der $L$ beschreibt ($M_D = L$)

DFAs, NFAs, NFAs mit $\lambda$-Kanten und reguläre Ausdrücke sind also äquivalent.

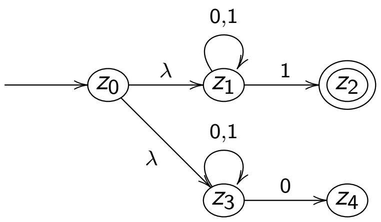




## NFAs mit Lambda-Kanten

Man kann einem NFA zusätzlich noch $\lambda$-Kanten (oder $\epsilon$-Kanten) erlauben:

- nichts vom Eingabeband lesen
- es findet nur ein Zustandswechsel statt








## Reguläre Ausdrücke

**Definition (Reguläre Ausdrücke)**

Sei $\Sigma$ ein Alphabet. Die **regulären Ausdrücke über** $\Sigma$ sind induktiv definiert durch:

1. $\emptyset$ ist ein regulärer Ausdruck, der die Menge $M_\emptyset = \emptyset$ beschreibt.
2. Für jedes $a \in \Sigma$ ist $a$ ein regulärer Ausdruck, der die Menge $M_a = \{a\}$ beschreibt.
3. Sind $X$ und $Y$ reguläre Ausdrücke, die die Mengen $M_X$ und $M_Y$ beschreiben, dann beschreibt
   - $(X + Y)$ die Menge $M_X \cup M_Y$
   - $(X \cdot Y)$ die Menge $M_X \cdot M_Y$
   - $X^*$ die Menge $M_X^*$ und
   - $X^+$ die Menge $M_X^+$.
4. Nur die so erzeugten Ausdrücke sind reguläre Ausdrücke.

# Reguläre Ausdrücke - Beispiele

**Anmerkung**

Wir nutzen reguläre Ausdrücke wie Mengen. Erlauben uns also z.B. $aaa \in a^*$ zu schreiben. Gemeint ist dann mit dem regulären Ausdruck gerade die Menge, die er beschreibt.

Ein paar Beispiele:

- Worte, die auf 1 enden: $(0+1)^* \cdot 1$
- Worte, die 1 enthalten: $(0+1)^* \cdot 1 \cdot (0+1)^*$
- Worte, die 00 oder 11 enthalten: $(0+1)^* \cdot (00+11) \cdot (0+1)^*$
- ...

Aber ist auch so etwas regulär wie *nicht* $(0+1)^* \cdot 1$?





# Abschlusseigenschaften

**Definition**

Sei $f_1$ eine einstellige Operation auf Mengen und $f_2$ eine zweistellige Operationen. D.h. wenn $M_1, M_2$ zwei Mengen sind, dann sind auch $f_1(M_1)$ und $f_2(M_1, M_2)$ Mengen.

- Eine Sprachfamilie $\mathcal{C}$ ist **abgeschlossen** gegenüber der Operation $f_1$ bzw. $f_2$, wenn für jedes $R \in \mathcal{C}$ auch $f_1(R) \in \mathcal{C}$ gilt bzw. wenn für $R_1, R_2 \in \mathcal{C}$ auch $f_2(R_1, R_2) \in \mathcal{C}$ gilt.

**Beispiel**

Typische Beispiele sind z.B. Vereinigung oder Komplementbildung. Oft schreibt man die Operatoren in Infix-Notation (d.h. den Operator zwischen die Argumente).





# Abschlusseigenschaften

**Ziel**

Wir wollen jetzt Abschlusseigenschaften für die regulären Sprachen untersuchen, also z.B. ob für zwei reguläre Sprachen $L_1, L_2 \in \text{REG}$ auch

- $L_1 \cup L_2 \in \text{REG}$
- $L_1^+ \in \text{REG}$
- $\overline{L} \in \text{REG}$
- ...

gilt.

**Vorgehen**

Die Argumentation ist dann immer "Seien $L_1$ und $L_2$ reguläre Sprachen, dann ..., also ist auch $L_1 \circ L_2$ regulär".





# Einfache Abschlusseigenschaften

Viele Abschlusseigenschaften kriegen wir durch reguläre Ausdrücke "geschenkt", da wir ja wissen, dass diese reguläre Sprachen beschreiben.

**Satz**

*Die regulären Sprachen sind abgeschlossen gegenüber* $\cup, \cdot, +, *$.

**Beweis.**

Seien $L_1, L_2 \in \text{REG}$. Dann gibt es reguläre Ausdrücke $A_1, A_2$, die $L_1$ bzw. $L_2$ beschreiben. Nach Definition sind nun auch $A_1 + A_2$, $A_1 \cdot A_2$, $A_1^+$, $A_1^*$ reguläre Ausdrücke, die die Mengen $L_1 \cup L_2$, $L_1 \cdot L_2$, $L_1^+$ und $L_1^*$ beschreiben. Diese sind folglich auch regulär. (Da jede Sprache, die von einem regulären Ausdruck beschrieben werden kann, regulär ist.) $\square$

## Komplement

**Satz**

*Die regulären Sprachen sind gegenüber Komplementbildung abgeschlossen, d.h. ist $L \in REG$, dann auch $\overline{L} \in REG$.*

**Beweis.**

Sei $L \in$ REG. Dann gibt es einen vollständigen DFA $A$ mit $L(A) = L$. Wir konstruieren nun einen vollständigen DFA $A'$ aus $A$ wie folgt:

- Alles von $A$ übernehmen
- $z \in Z$ in $A$ Endzustand, dann in $A'$ nicht
- $z \in Z$ in $A$ kein Endzustand, dann einer in $A'$





## Komplement

**Satz**

*Die regulären Sprachen sind gegenüber Komplementbildung abgeschlossen, d.h. ist $L \in REG$, dann auch $\overline{L} \in REG$.*

**Beweis.**

(Wir haben gerade End- und Nicht-Endzustände getauscht.) Wurde ein Wort nun von $A$ akzeptiert (die Rechnung endet in einem Endzustand), so wird es von $A'$ nicht akzeptiert (kein Endzustand). Wurde ein Wort von $A$ nicht akzeptiert (Rechnung endet in einem Nicht-Endzustand), so wird es von $A'$ akzeptiert (Endzustand). Damit gilt $L(A') = \overline{L}$. □





## Der Produktautomat

**Satz**

*Seien $L_1, L_2 \in REG$, dann ist auch $L_1 \cap L_2 \in REG$.*

**Beweis.**

Seien $A_1 = (Z_1, \Sigma_1, \delta_1, z_{1,0}, Z_{1,end})$ $A_2 = (Z_2, \Sigma_2, \delta_2, z_{2,0}, Z_{2,end})$ vollständige DFAs mit $L(A_1) = L_1$ und $L(A_2) = L_2$. Wir konstruieren $C = (Z_3, \Sigma_3, \delta_3, z_{3,0}, Z_{3,end})$ mit

$$\begin{aligned}
Z_3 &:= Z_1 \times Z_2 \\
\Sigma_3 &:= \Sigma_1 \cap \Sigma_2 \\
z_{3,0} &:= (z_{1,0}, z_{2,0}) \\
Z_{3,end} &:= Z_{1,end} \times Z_{2,end} \\
\delta_3((z_1, z_2), x) &:= (\delta_1(z_1, x), \delta(z_2, x))
\end{aligned}$$

□





## Der Produktautomat

Im Produktautomaten $C$ wird also

- in der ersten Komponente eine Rechnung von $A_1$ und
- in der zweiten Komponente eine Rechnung von $A_2$

gemacht. Daraus folgt schnell die Korrektheit:

$$\hat{\delta}_3(z_{3,0}, w) \in Z_{3,end} \Leftrightarrow \hat{\delta}_1(z_{1,0}, w) \in Z_{1,end} \wedge \hat{\delta}_2(z_{2,0}, w) \in Z_{2,end}$$

Dies wäre noch genauer zu zeigen. Z.B. wieder mit einer Induktion über die Wortlänge. Dafür bietet sich obiges aber nicht an. Warum? Was wäre eine bessere Induktionsbehauptung?

## Der Produktautomat - Konstruktion

Will man einen Produktautomaten konstruieren, so

- beginnt man in $(z_{1,0}, z_{2,0})$
- berechnet $\delta(z_{1,0}, x)$ und $\delta(z_{2,0}, x)$ für jedes $x \in \Sigma$ (sind für ein $x$ beide definiert, ist dies ein neuer Nachfolgezustand - so lässt sich das Verfahren auch auf nicht vollständige DFAs erweitern)
- So fährt man dann fort ...

**Bemerkung**

Im Grund genommen ähnlich zur Potenzautomatenkonstruktion, nur dass man hier im ersten Tupelelement immer wie im ersten Automaten und im zweiten Tupelelement immer wie im zweiten Automaten rechnet.





## Reverse-Operation

**Definition ($w^{rev}$)**

Das **Spiegelwort** $w^{rev}$ zu einem Wort $w \in \Sigma^*$ ist definiert durch

1. $\lambda^{rev} = \lambda$ und
2. $(ux)^{rev} = x \cdot u^{rev}$ mit $x \in \Sigma$ und $u \in \Sigma^*$

Bspw. ist für $w = 1011$ dann $w^{rev} = 1101$.
Für Mengen notieren wir $L^{rev} = \{w^{rev} \mid w \in L\}$.

**Satz**

*Ist $L$ regulär, dann ist auch $L^{rev}$ regulär.*

**Beweisidee**

Drehe alle Kanten um. Startzustand wird Endzustand. Endzustände werden Startzustände. (So entsteht ein NFA.)

Dies zeigt auch, dass *Sum* aus Kapitel 3 regulär ist!





## Abschlusseigenschaften - Zusammenfassung

**Satz**

*Die regulären Sprachen sind abgeschlossen gegenüber*

- *Vereinigung* $\cup$
- *Konkatenation* $\cdot$
- *“hoch $+$”*
- *“hoch $*$”*
- *Komplementbildung*
- *Durchschnitt* $\cap$
- *Reverse*

**Für Interessierte**

Es gelten weitere Abschlusseigenschaften. Siehe z.B. [HMU].





## Fragen...

Welcher reguläre Ausdruck beschreibt die Menge $\overline{\{a\}^*}$?

1. $(a+b)^* \cdot b \cdot (a+b)^*$
2. $b^*$
3. $(a+b)^* \setminus a^*$
4. $(a+b)^* \cdot b^* \cdot (a+b)^*$

## Fragen...

Sie $A$ ein NFA mit 6 Zuständen. Wie viele Zustände hat ein durch die Potenzautomatenkonstruktion gewonnener DFA maximal?

1. 6
2. 12
3. 36
4. 64





## Fragen...

Sei $A$ ein NFA mit 5 Zuständen. Wie viele Zustände hat ein äquivalenter DFA mindestens?

1. 1
2. 5
3. 25
4. 32





## Fragen...

Das Tempo der Vorlesung ist in den ersten drei Wochen bisher ...

1. Zu langsam
2. Etwas zu langsam
3. Ist ok ...
4. Etwas zu schnell
5. Zu schnell





## Zur Nachbereitung

**Zur Nachbereitung**

Richtige Antworten sind:

1. 1
2. 4
3. 1
4. -

## Möglichkeiten und Grenzen

Wir können nun sehr viele Sprachen konstruieren, die alle regulär sind. Sind z.B. $L_1, L_2$ regulär. Dann auch (in etwas schluderiger Schreibweise)

- $(L_1 + L_2)^* \cdot L_1 \cdot L_2^+ \cdot L_1^*$
- $(L_1^* \cdot a^3 \cdot L_2^*)^+$
- ...

dennoch wissen wir, dass wir nur endlich viele Informationen speichern können (der DFA hat nur endlich viele Zustände).

Bei welcher Sprache könnten wir Probleme kriegen?





## Grenzen regulärer Sprachen?

Es gibt viele Sprache, bei denen es uns schwerfällt, einen Automaten zu konstruieren.

**Beispiel**

Die Sprache aller korrekt geklammerten Ausdrücke ?

Z.B. wollen wir ()(()) akzeptieren, (() nicht.





## Grenzen regulärer Sprachen?

Mit unseren bisherigen Techniken scheitern wir hier...
... geht das überhaupt?!

Eine einfachere Sprache, die aber einen wichtigen Aspekt erfasst:

$$L := \{a^n b^n \mid n \in \mathbb{N}\}$$

Ist diese regulär ?

Ideen für einen DFA?
Welche? Oder was ist das Problem?





## Kellerautomaten

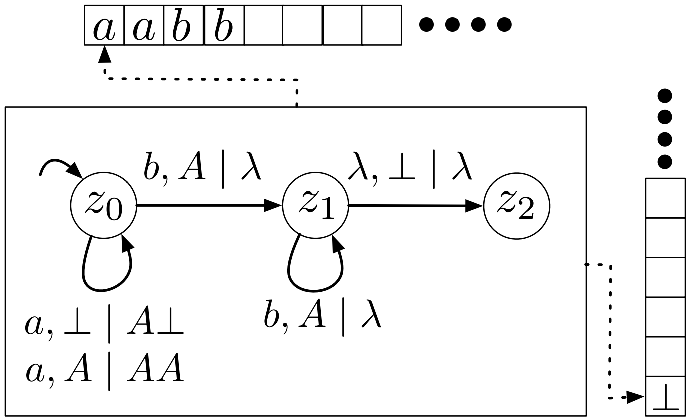
Wir erweitern den DFA um einen Keller, um uns Dinge zu merken.

Den so entstehenden Automaten nennen wir **Kellerautomat**.

**Anmerkung**

Wir zeigen den Kellerautomaten hier nur kurz. Es gibt dazu sehr viel mehr (siehe Literatur). Das führt hier aber zu weit.

## Der Kellerautomat








## Der Kellerautomat - Formal

**Definition (Kellerautomat (PDA = Push Down Automata))**

Ein **nichtdeterministischer Kellerautomat** (kurz PDA) ist ein 7-Tupel $A = (Z, \Sigma, \Gamma, \delta, Z_{start}, Z_{end}, \perp)$ mit

- Der endlichen Menge von *Zuständen* $Z$.
- Dem endlichen Alphabet $\Sigma$ von *Eingabesymbolen*.
- Dem endlichen Alphabet $\Gamma$ von *Kellersymbolen*.
- Der *Überführungsfunktion* $\delta : Z \times (\Sigma \cup \{\lambda\}) \times \Gamma \to 2^{Z \times \Gamma^*}$.
- Der Menge von *Startzuständen* $Z_{start} \subseteq Z$.
- Der Menge der *Endzustände* $Z_{end} \subseteq Z$.
- Dem *Kellerbodensymbol* $\perp \in \Gamma$.

**Anmerkung**

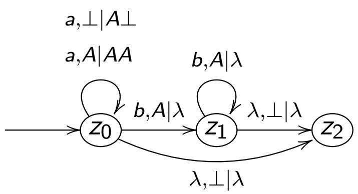
Dieses Automatenmodell ist *nichtdeterministisch*





## Beispiel $a^n b^n$

$$L = \{a^n b^n \mid n \in \mathbb{N}\}$$








## DFA vs. PDA

**Wichtige Anmerkung**

Alles was ein DFA kann, kann auch ein PDA! Jede Kante $(z, a, z')$ im DFA wird im PDA zur Kante $(z, a, \perp, \perp, z')$.
Der PDA ist also *mindestens so mächtig* wie der DFA. Die Frage ist nun noch, ob er tatsächlich mehr kann... (also weiterhin, ob es wirklich Sprachen gibt, die nicht regulär sind)

## Zum Pumping Lemma hin...

Wir haben jetzt einen PDA für $L := \{a^n b^n \mid n \in \mathbb{N}\}$, *aber* bisher wissen wir nicht mit Sicherheit, dass es hierfür keinen DFA gibt!

Können wir vielleicht eine Eigenschaft X finden, die alle regulären Sprachen haben müssen?

Dann könnten wir vielleicht nachweisen, dass $L$ oben die Eigenschaft $X$ nicht hat - und dann kann $L$ nicht regulär sein!





## Zum Pumping Lemma hin...

Wir wissen von regulären Sprachen, dass

- sie von endlichen Automaten akzeptiert werden.

Von diesen wissen wir, dass

- sie endlich viele Zustände haben.





## Zum Pumping Lemma hin...

- Wörter, die akzeptiert werden und
- die mehr Buchstaben haben als der Automat Zustände,
- müssen dann Schleifen im Automaten durchlaufen!
- Diese Schleifen könnte man öfter entlang gehen!
- Und damit neue Wörter erzeugen, die auch in der Sprache sein müssen!

Bei $a^n b^n$ wird das aber nicht gehen! Man überlege sich:

- Wenn es einen Automat gibt mit $k$ Zuständen für $L$
- betrachtet man $a^k b^k$
- Es müsste eine Schleife geben
- Aber welche sollte das sein, so dass man sie zweimal entlang gehen kann und trotzdem ein Wort der Form $a^j b^j$ herauskommt?





## Das Pumping Lemma

Wir werden jetzt die oben skizzierte Eigenschaft der regulären Sprachen genauer fassen und so das **Pumping Lemma** formulieren.

Dies werden wir dann nutzen, um tatsächlich zu zeigen, dass $L = \{a^n b^n\}$ nicht regulär ist!

## Das Pumping Lemma - Das Lemma

**Lemma**

*Sei $L \in REG$ eine reguläre Sprache. Dann gibt es ein $n \geq 1$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in die Form $z = uvw$ zerlegt werden kann, wobei*

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $uv^iw \in L$ *für jedes* $i \in \mathbb{N}$ *(inkl. der 0)*

*(Äquivalent zur letzten Aussage ist die Aussage $\{u\}\{v\}^*\{w\} \subseteq L$.)*

**Wichtige Anmerkung**

Das $v$ kann (mit $i = 0$) auch ganz weggelassen werden, d.h. die dritte Bedingung fordert, dass nicht nur $uvw, uvvw, uvvvw$ usw. in $L$ sein müssen, sondern auch $uv^0w = u\lambda w = uw$. Dies ist manchmal die einzige Möglichkeit einen Widerspruch herzustellen!





## Das Pumping Lemma - Der Beweis

**Beweis.**

Sei $L$ regulär. Dann gibt es einen vollständigen DFA $A$ mit $L(A) = L$. Sei $n$ die Anzahl der Zustände von $A$. Wir betrachten nun ein Wort $z \in L$ mit $|z| \geq n$ (gibt es kein solches $z$ ist nichts zu zeigen).

- Wir legen $A$ das Wort $z$ vor.
- $A$ beginnt im Startzustand $z_0$
- Da $A$ vollständig ist und $z$ akzeptiert wird, kann $z$ gelesen werden.
- Da $z$ aus mindestens $n$ Buchstaben besteht, finden mindestens $n$ Zustandsübergänge statt.
- D.h. es werden mindestens $n + 1$ Zustände besucht.
- Da es nur $n$ Zustände gibt, muss mindestens ein Zustand doppelt auftreten!





## Das Pumping Lemma - Der Beweis

**Beweis.**

Sei die Zustandsfolge vom Startzustand $z_0$ zu einem Endzustand $z_e$, die beim Lesen von $z$ besucht wird durch $z_0, z_1, z_2, \ldots, z_n, \ldots, z_e$ gegeben.

- Seien $z_i$ und $z_j$ ($i \neq j$ und $i < j$) in dieser Folge gleich und außerdem $z_j$ der erste Zustand, der ein zweites Mal auftritt!
- Nun ist $j \leq n$ (denn von $z_0$ bis $z_n$ sind es bereits $n + 1$ Zustände; einer davon muss doppelt auftreten!)
- und $j - i \geq 1$
- Sei $u$ das Wort das von $z_0$ bis $z_i$ gelesen wird
- $v$ das Wort das von $z_i$ bis $z_j$ gelesen wird
- und $w$ das Wort von $z_j$ bis $z_e$





## Das Pumping Lemma - Der Beweis

**Beweis.**

- Sei $u$ das Wort das von $z_0$ bis $z_i$ gelesen wird
- $v$ das Wort das von $z_i$ bis $z_j$ gelesen wird
- und $w$ das Wort von $z_j$ bis $z_e$

Es ist nun $|uv| \leq n$ (wegen $j \leq n$ oben) und $|v| \geq 1$ (wegen $j - i \geq 1$ oben). Ferner kann die Schleife von $z_i$ nach $z_j$ (man beachte: $i \neq j$, aber $z_i = z_j$!) beliebig oft durchlaufen werden und trotzdem kann dann von $z_j$ aus mit $w$ der Endzustand $z_n$ erreicht werden. Daraus folgt, dass auch $uv^iw$ für alle $i \geq 0$ akzeptiert werden. Und wir sind fertig! □

## Das Pumping Lemma - Das Lemma

**Lemma**

*Sei $L \in REG$ eine reguläre Sprache. Dann gibt es ein $n \geq 1$, so dass jedes Wort $z \in L$ mit $|z| \geq n$ in die Form $z = uvw$ zerlegt werden kann, wobei*

1. $|uv| \leq n$
2. $|v| \geq 1$
3. $uv^iw \in L$ *für jedes* $i \in \mathbb{N}$ *(inkl. der 0)*

*(Äquivalent zur letzten Aussage ist die Aussage $\{u\}\{v\}^*\{w\} \subseteq L$.)*

**Wichtige Anmerkung**

Das $v$ kann (mit $i = 0$) auch ganz weggelassen werden, d.h. die dritte Bedingung fordert, dass nicht nur $uvw, uvvw, uvvvw$ usw. in $L$ sein müssen, sondern auch $uv^0w = u\lambda w = uw$. Dies ist manchmal die einzige Möglichkeit einen Widerspruch herzustellen!





## Das Pumping Lemma - Der Ablauf

Der Ablauf beim Pumping Lemma:

1. Annehmen $L$ wäre regulär
2. Die Zahl aus dem Pumping Lemma benennen (z.B. $k$).
3. **Ein Wort** $z$ finden mit $|z| \geq k$ und $z \in L$.
   - Dieses Wort muss gut gewählt sein, damit der nachfolgende Widerspruch gelingt! Hier muss man also u.U. experimentieren!
4. **Für alle Zerlegungen** von $z$ in $uvw$ zeigen, dass sie im Widerspruch zur ersten, zweiten oder dritten Bedingung sind.
   - Üblicherweise betrachtet man alle Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigt, dass man dann einen Widerspruch zur dritten Bedingung erhält.
5. Also: Alle Zerlegungen $z = uvw$ mit $|uv| \leq k$ und $|v| \geq 1$ betrachten.
6. Zeigen, dass man **ein** $i$ findet mit $uv^iw \notin L$. (Dies kann auch $i = 0$ sein!)





## Das Pumping Lemma - Beispiel 1

**Behauptung**

$L := \{a^nb^n \mid n \in \mathbb{N}\}$ ist nicht regulär.

**Beweis.**

Angenommen $L$ wäre regulär. Dann gilt das Pumping Lemma. Sei $k$ die Zahl aus dem Pumping Lemma. Wir betrachten das Wort $z = a^kb^k$. Es gilt $z \in L$ und $|z| \geq k$. Damit muss es eine Zerlegung $z = uvw$ geben, die die drei Eigenschaften erfüllt. Wir betrachten nun nur jene Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigen, dass diese im Widerspruch zur dritten stehen. Wir haben dann gezeigt, dass jede Zerlegung von $z$ in $uvw$ im Widerspruch zu einer der drei Bedingungen steht und sind dann fertig. ...





## Das Pumping Lemma - Beispiel 1

**Behauptung**

$L := \{a^nb^n \mid n \in \mathbb{N}\}$ ist nicht regulär.

**Beweis.**

Sei also $z = a^kb^k = uvw$ eine Zerlegung mit $i)|uv| \leq k$ und $ii)|v| \geq 1$. Es muss dann $v \in \{a\}^+$ gelten, da $uv$ ja die ersten Buchstaben von $z$ ausmacht, aber davon nur maximal $k$ einnehmen kann und da ferner $v$ mindestens die Länge 1 hat. Es gibt sogar ein $j$ mit $1 \leq j \leq k$ derart, dass $v = a^j$ gilt. Wir betrachten nun das Wort $uv^2w$, dass nach der dritten Bedingung in $L$ sein müsste. Es ist aber $uv^2w = a^{k+j}b^k$ und somit $uv^2w \notin L$. Die ursprüngliche Annahme muss also falsch sein und daher ist $L$ nicht regulär. □

## Wichtige Folgerung

**Wichtige Anmerkung**

Damit ist bewiesen,
**dass der PDA tatsächlich mehr kann als der DFA!**
Die Sprachfamilie der von einem PDA akzeptierten Sprachen ist also echt größer als die Sprachfamilie REG (der von DFAs akzeptierten Sprachen).





## Fragen...

Wovon muss das gewählte Wort $z$ abhängen?

1. Von der Anzahl der Zustände $|Z|$ des DFA.
2. Von der Zahl aus dem Pumping Lemma $k$, die noch genauer spezifiziert werden muss.
3. Von der Zahl aus dem Pumping Lemma $k$, die nicht genauer spezifiziert werden muss.
4. Das Wort muss lediglich in der Sprache sein.





## Fragen...

Wieviele Zerlegungen von $z$ müssen betrachtet werden?

1. Eine muss zum Widerspruch geführt werden!
2. Alle Zerlegungen von $z$ müssen zum Widerspruch gebracht werden!
3. Zerlegungen mit bestimmten Eigenschaften müssen zum Widerspruch gebracht werden!
4. Man muss eine Zerlegung mit bestimmten Eigenschaften finden und diese zum Widerspruch führen!





## Fragen...

Kann mit dem Pumping Lemma gezeigt werden, dass eine Sprache regulär ist?

1. Ja!
2. Nein!

## Fragen...

Ist $L := \{ww^{rev} \mid w \in \{0,1\}^*\}$ regulär?

1. Ja!
2. Nein!





## Zur Nachbereitung

**Zur Nachbereitung**

Richtige Antworten sind:

1. 3
2. 2 und 3 ist beides ok
3. 2
4. 2 (Beweis folgt)





## Das Pumping Lemma - Beispiel 2

**Behauptung**

$L := \{ww^{rev} \mid w \in \{0,1\}^*\}$ ist nicht regulär.

**Beweis.**

Angenommen $L$ wäre regular. Dann gilt das Pumping Lemma. Sei $k$ die Zahl aus dem Pumping Lemma. Wir betrachten $z = 0^k1^k1^k0^k$. Es ist $z \in L$ und $|z| \geq k$. Sei $z = uvw$ eine Zerlegung von $z$ mit $|uv| \leq k$ und $|v| \geq 1$. Hieraus folgt $v = 0^j$ für ein $j$ mit $1 \leq j \leq k$. Nun ist aber $uv^0w = 0^{k-j}1^k1^k0^k$ nicht mehr in $L$ im Widerspruch zum Pumping Lemma. Die Annahme ist daher falsch und $L$ somit nicht regulär. $\square$





## Das Pumping Lemma - Nochmal der Ablauf

Der Ablauf beim Pumping Lemma:

1. Annehmen $L$ wäre regulär
2. Die Zahl aus dem Pumping Lemma benennen (z.B. $k$).
3. **Ein Wort** $z$ finden mit $|z| \geq k$ und $z \in L$.
   - Dieses Wort muss gut gewählt sein, damit der nachfolgende Widerspruch gelingt! Hier muss man also u.U. experimentieren!
4. **Für alle Zerlegungen** von $z$ in $uvw$ zeigen, dass sie im Widerspruch zur ersten, zweiten oder dritten Bedingung sind.
   - Üblicherweise betrachtet man alle Zerlegungen, die die erste und zweite Bedingung erfüllen und zeigt, dass man dann einen Widerspruch zur dritten Bedingung erhält.
5. Also: Alle Zerlegungen $z = uvw$ mit $|uv| \leq k$ und $|v| \geq 1$ betrachten.
6. Zeigen, dass man **ein** $i$ findet mit $uv^iw \notin L$. (Dies kann auch $i = 0$ sein!)

# Das Pumping Lemma - Anmerkung zum Ablauf

**Anmerkung**

Bei der Zerlegung (vorletzter Schritt eben) gelingt oft eine Parametrisierung. Bspw. konnte in den Beispielen oben $v = 0^j$ bzw. $v = a^j$ mit einem $j$ mit $1 \leq j \leq k$ gesetzt werden. Das $j$ wandert quasi den Bereich von 1 bis $k$ ab. Für jedes $j$ ist dies eine mögliche Zerlegung, die die ersten beiden Bedingungen erfüllt – und man führt im Grunde genommen *alle* diese Zerlegungen zum Widerspruch!





# Wiederholung

Wir haben gestern und heute:

- Mehr zu regulären Sprachen gesehen
  - NFA mit $\lambda$-Kanten
  - reguläre Ausdrücke
  - Abschlusseigenschaften
- Kellerautomaten (kurz) kennengelernt
- Das Pumping Lemma kennengelernt
- Damit bewiesen, dass Kellerautomaten mächtiger sind als endliche Automaten und damit auch gesehen, dass es Sprachen gibt, die nicht regulär sind.