

21.12.2012

Willkommen zur zehnten Übung zur Vorlesung

Grundbegriffe der Informatik



Matthias Janke

email: [matthias.janke@kit.edu](mailto:matthias.janke@kit.edu)

- ▶ Anmeldung für den Übungsschein nicht vergessen!
- ▶ Gestern waren 569 Personen angemeldet
- ▶ Anmeldung für die Klausur nicht vergessen!
- ▶ Gestern waren 315 Personen angemeldet
- ▶ Anmeldung über Studierendenportal: [studium.kit.edu](https://studium.kit.edu)
- ▶ Online Klausur-Anmeldung möglich für: INFO, INWI, MATH, PHYS

Master-Theorem

Endliche Automaten

Ergebnisse der Evaluation

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

Drei Kochrezepte, in denen  $f(n)$  und  $\log_b a$  eine Rolle spielen:

- Fall 1: Wenn  $f(n) \in O(n^{(\log_b a) - \varepsilon})$  für ein  $\varepsilon > 0$  ist, dann ist  $T(n) \in \Theta(n^{\log_b a})$ .
- Fall 2: Wenn  $f(n) \in \Theta(n^{\log_b a})$  ist, dann ist  $T(n) \in \Theta(n^{\log_b a} \log n)$ .
- Fall 3: Wenn  $f(n) \in \Omega(n^{(\log_b a) + \varepsilon})$  für ein  $\varepsilon > 0$  ist, und wenn es eine Konstante  $d$  gibt mit  $0 < d < 1$ , so dass für alle hinreichend großen  $n$  gilt  $af(n/b) \leq df(n)$ , dann ist  $T(n) \in \Theta(f(n))$ .

Einfachster Fall:

$$T(1) = 1$$

$$T(n) = aT(n/b) : T(b^k) = a^k \Rightarrow T(n) = n^{\log_b a}$$

Zweiteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  wenig ausmacht:

$$T(n) \in \Theta(n^{\log_b a})$$

Zweiteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  wenig ausmacht:

$$T(n) \in \Theta(n^{\log_b a})$$

Wenig ausmachen:

- ▶ Weniger als  $n^{\log_b a}$
- ▶ “Polynomial” weniger als  $n^{\log_b a}$

Zweiteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  wenig ausmacht:

$$T(n) \in \Theta(n^{\log_b a})$$

Wenig ausmachen:

$$f(n) \in O(n^c) \text{ mit } c < \log_b a$$



Zweiteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  wenig ausmacht:

$$T(n) \in \Theta(n^{\log_b a})$$

Wenig ausmachen:

$$f(n) \in O(n^c) \text{ mit } c < \log_b a$$

Abgleich mit Vorlesung:

$$c < \log_b a \Rightarrow \epsilon = \log_b a - c > 0 \Rightarrow f(n) \in O(n^{\log_b a - \epsilon})$$

Zweiteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  wenig ausmacht:

$$T(n) \in \Theta(n^{\log_b a})$$

Nicht wenig genug:  $f(n) = n^{\log_b a} / \log_2 n$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

$$T(1) = 0$$

$$T(2) = 2$$

$$T(4) = 6$$

$$T(8) = 44/3$$

$$T(16) = 100/3$$

$$T(32) = 1096/15$$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Oft hilfreich: Betrachte  $T(n)/n^{\log_b a}$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Oft hilfreich: Betrachte  $T(n)/n^{\log_b a}$

Hier also  $T(n)/n$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

$$T(1)/1 = 0/1 = 0$$

$$T(2)/2 = 2/2 = 1$$

$$T(4)/4 = 6/4 = 3/2$$

$$T(8)/8 = (44/3)/8 = 11/6$$

$$T(16)/16 = (100/3)/16 = 25/12$$

$$T(32)/32 = (1096/15)/32 = 137/60$$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Wenn man sonst nicht weiter weiß: Betrachte Differenzen!



Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

$$T(1)/1 = 0/1 = 0$$

$$T(2)/2 = 2/2 = 1 \rightarrow \text{Differenz } 1$$

$$T(4)/4 = 6/4 = 3/2 \rightarrow \text{Differenz } 1/2$$

$$T(8)/8 = (44/3)/8 = 11/6 \rightarrow \text{Differenz } 1/3$$

$$T(16)/16 = (100/3)/16 = 25/12 \rightarrow \text{Differenz } 1/4$$

$$T(32)/32 = (1096/15)/32 = 137/60 \rightarrow \text{Differenz } 1/5$$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Vollständige Induktion liefert  $T(2^k) = 2^k \cdot (\sum_{i=1}^k 1/i)$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Vollständige Induktion liefert  $T(2^k) = 2^k \cdot (\sum_{i=1}^k 1/i)$

$\sum_{i=1}^k 1/i \in \Theta(\log k)$

Beispiel:  $T(n) = 2T(n/2) + (n/\log_2 n)$

Vollständige Induktion liefert  $T(2^k) = 2^k \cdot (\sum_{i=1}^k 1/i)$

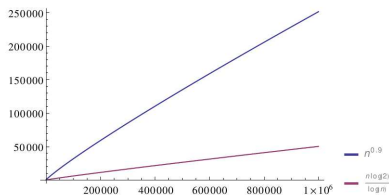
$\sum_{i=1}^k 1/i \in \Theta(\log k)$

Also  $T(n) \in \Theta(n \log_2(\log_2 n))$

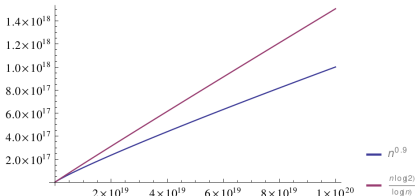
Fall 1: Wenn  $f(n) \in O(n^{(\log_b a) - \epsilon})$  für ein  $\epsilon > 0$  ist

- So etwas wie  $n/\log_2 n$  lässt sich nicht mit Fall 1 abdecken:

Plot:



Plot:



Dritteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  den größten Teil ausmacht.

Dann  $T(n) \in \Theta(f(n))$

Dritteinfachster Fall:

$T(n) = aT(n/b) + f(n)$ , wobei  $f(n)$  den größten Teil ausmacht.

Dann  $T(n) \in \Theta(f(n))$

Größten Teil ausmachen:

- ▶  $f(n) \in \Omega(n^c)$  mit  $c > \log_b a$

Master-Theorem

Endliche Automaten

Ergebnisse der Evaluation



Zwei Arten: Moore und Mealy-Automaten

Zwei Arten: Moore und Mealy-Automaten

Unterschied:

- ▶ Moore-Automaten: Ausgabefunktion hat ein Argument

Zwei Arten: Moore und Mealy-Automaten

Unterschied:

- ▶ Moore-Automaten: Ausgabefunktion hat ein Argument
- ▶ Mealy-Automaten: Ausgabefunktion hat zwei Argumente

Zwei Arten: Moore und Mealy-Automaten

Unterschied:

- ▶ Moore-Automaten: Ausgabefunktion hat ein Argument
- ▶ Mealy-Automaten: Ausgabefunktion hat zwei Argumente

Eselsbrücke: Anzahl der Argumente=Anzahl der Silben im Namen

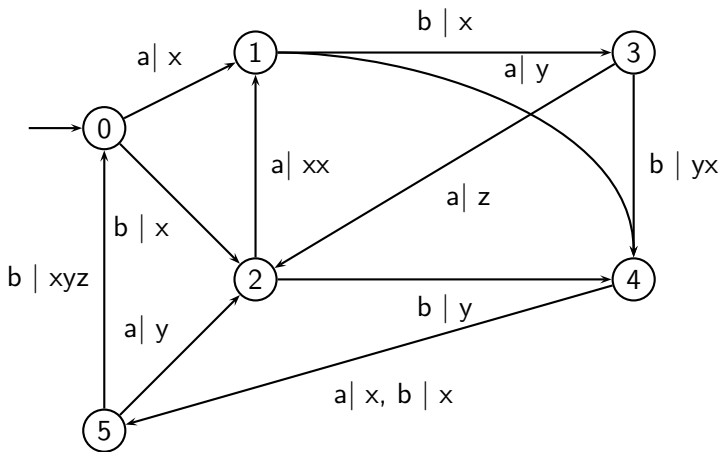
Moore/Mealy-Automat mit Anfangszustand  $z_0$ .

Ausgabe bei Eingabe  $w \in X^*$ : Üblicherweise  $g^{**}(z_0, w)$

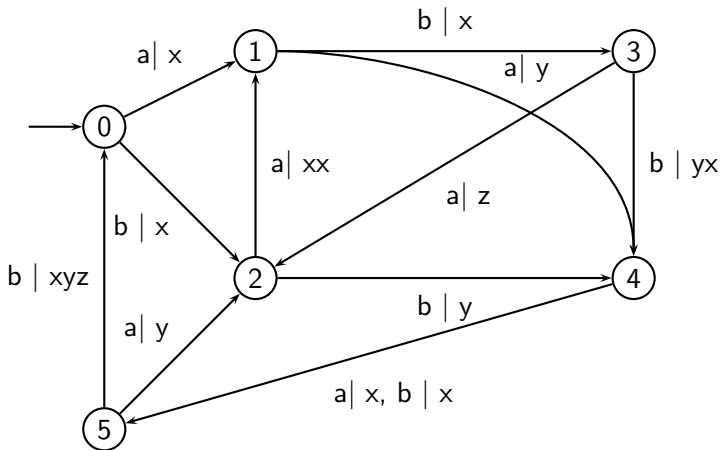
Moore/Mealy-Automat mit Anfangszustand  $z_0$ .

Ausgabe bei Eingabe  $w \in X^*$ : Üblicherweise  $g^{**}(z_0, w)$

Beachten Sie das  $z_0$ !

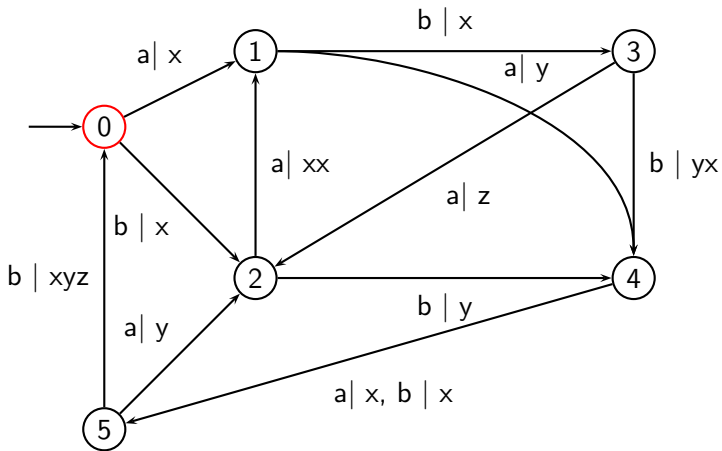


Eingabe: abbababba, **Ausgabe:**

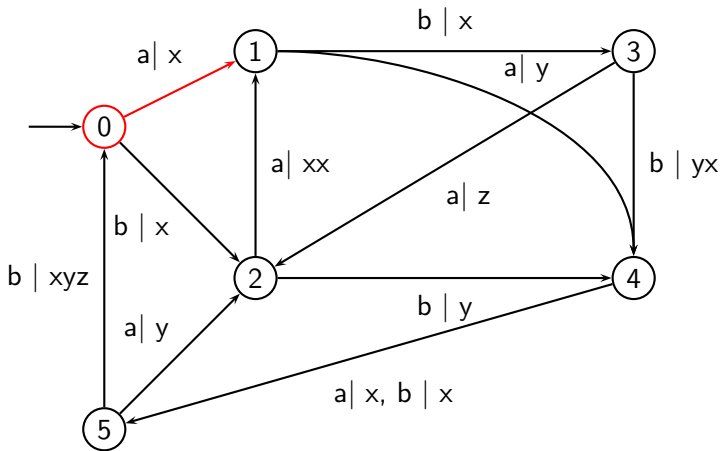




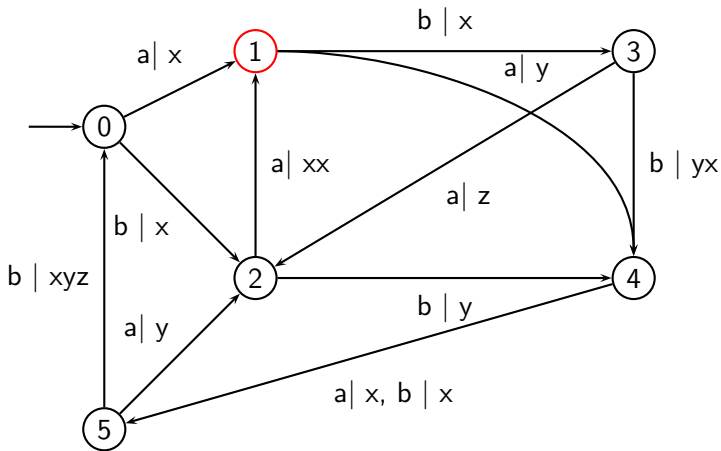
Eingabe: **abbababba**, **Ausgabe:**



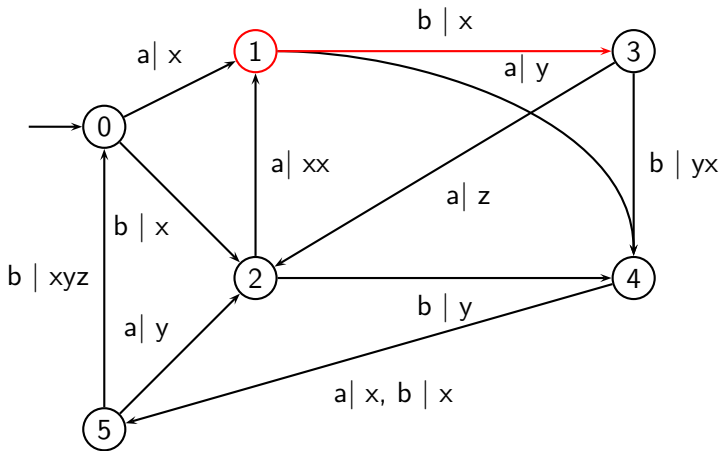
Eingabe: **abbababba**, Ausgabe: **x**



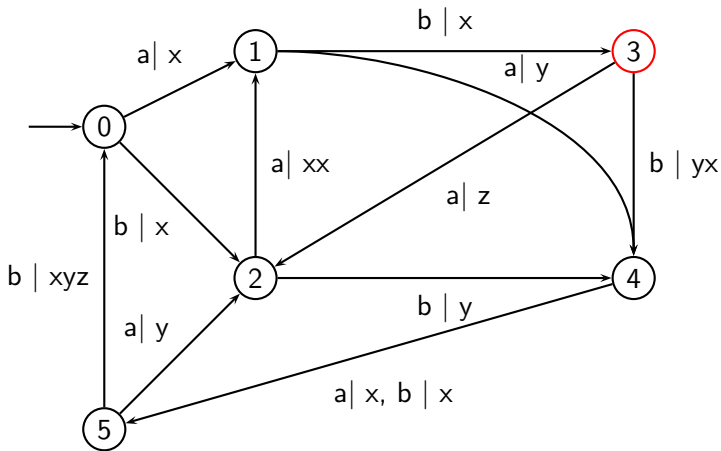
Eingabe: a**b**bababba, Ausgabe: x



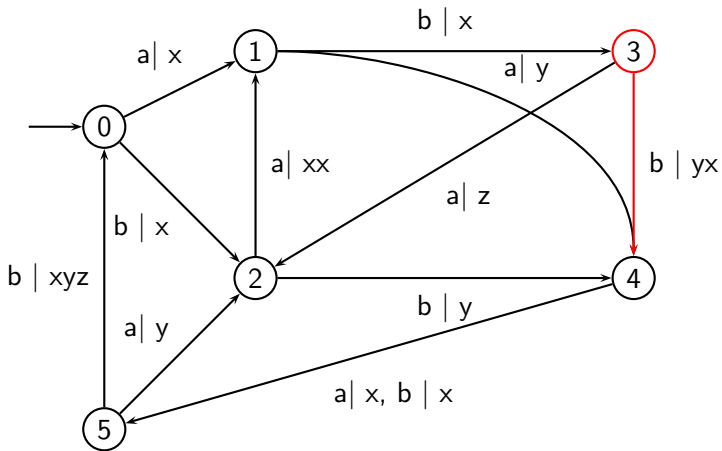
Eingabe: a**b**bababba, Ausgabe: **xx**



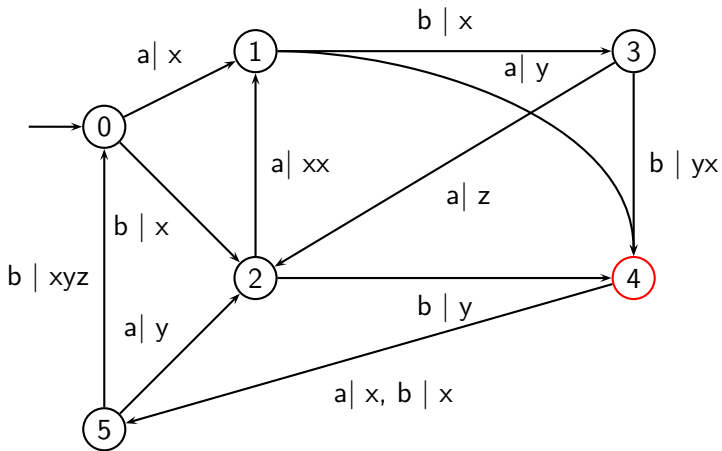
Eingabe: ab**b**ababba, Ausgabe: xx



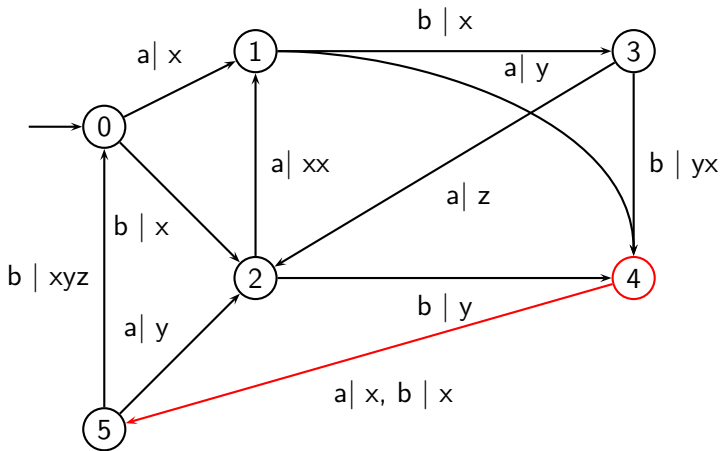
Eingabe: ab**b**ababba, Ausgabe: xxyx



Eingabe: abba**bab**ba, Ausgabe: **xx**yx

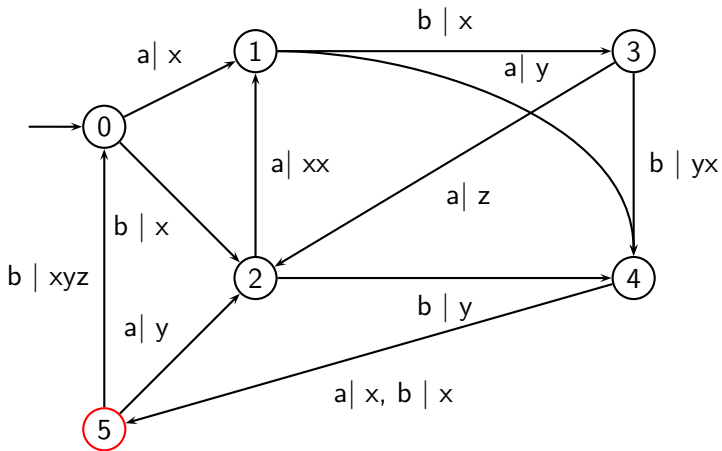


Eingabe: abba**bab**ba, Ausgabe: xxyxx

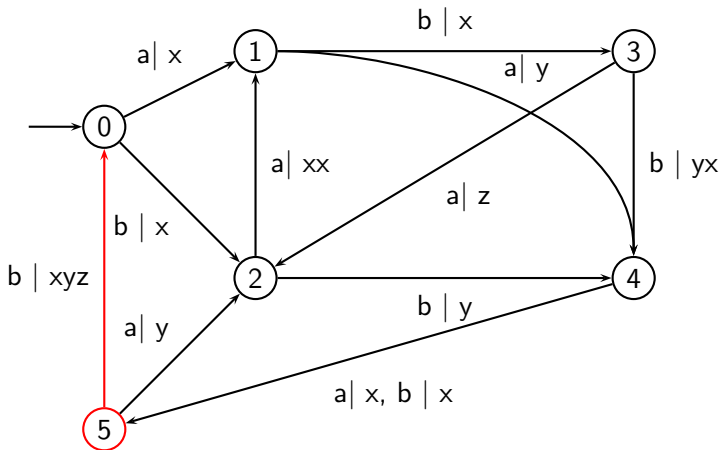




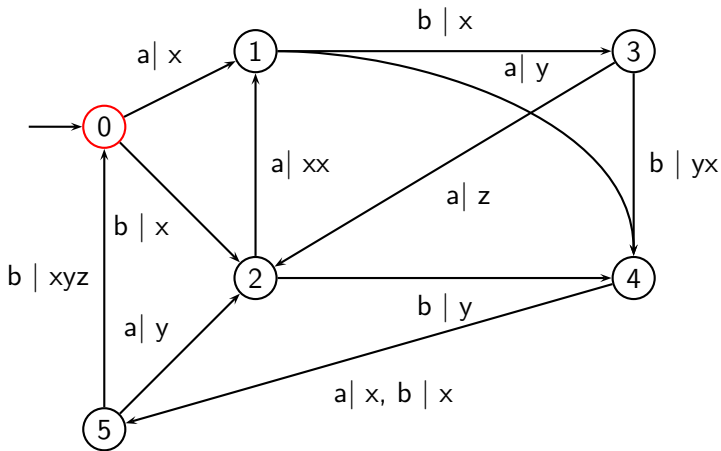
Eingabe: abba**b**abba, Ausgabe: xxyxx



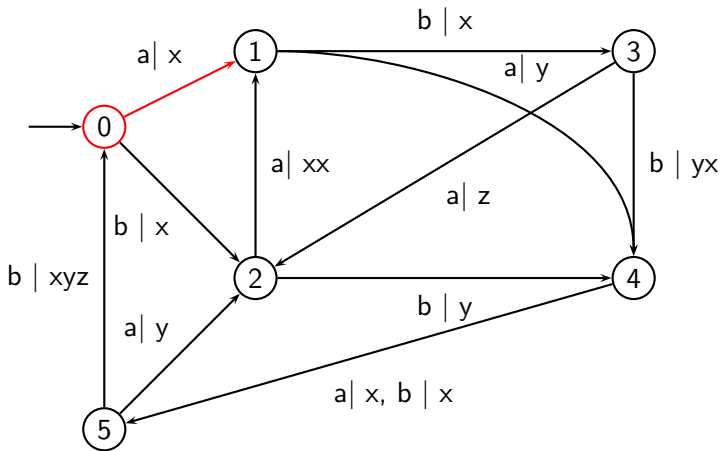
Eingabe: abba**b**abba, Ausgabe: xxyxxx**yz**



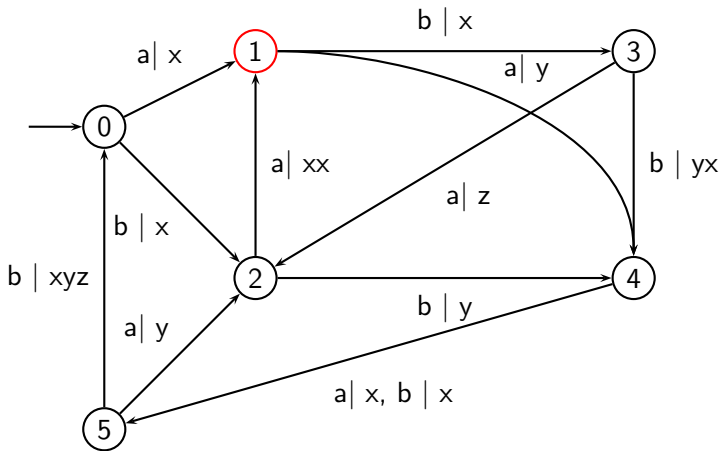
Eingabe: abbababba, Ausgabe: xxyxxxzy



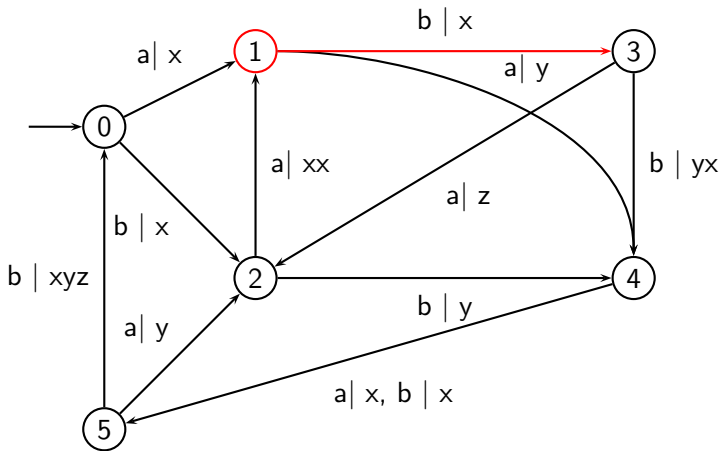
Eingabe: abbababba, Ausgabe: xxyxxxxyzx



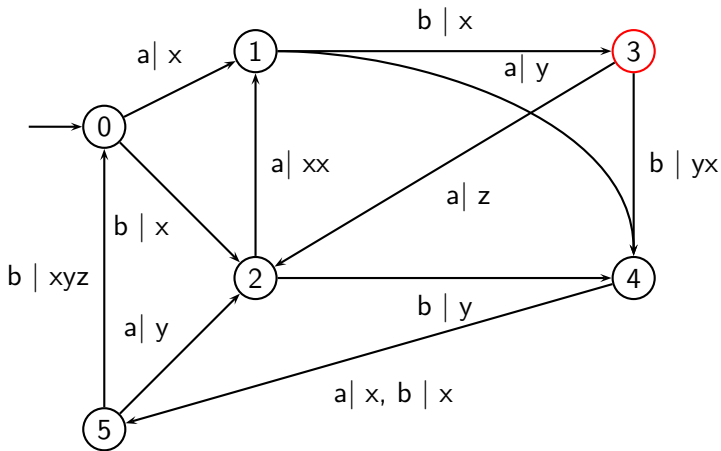
Eingabe: abbababba, Ausgabe: xxyxxxzyzx



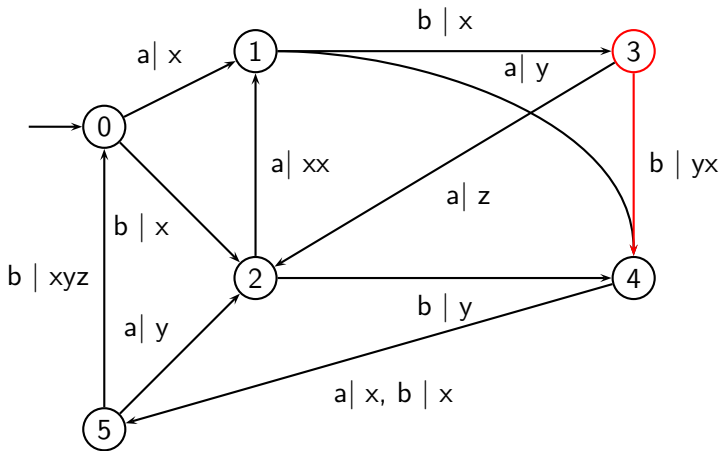
Eingabe: abbababba, Ausgabe: xxyxxxxyzxx



Eingabe: abbababba, Ausgabe: xxyxxxxyzxx

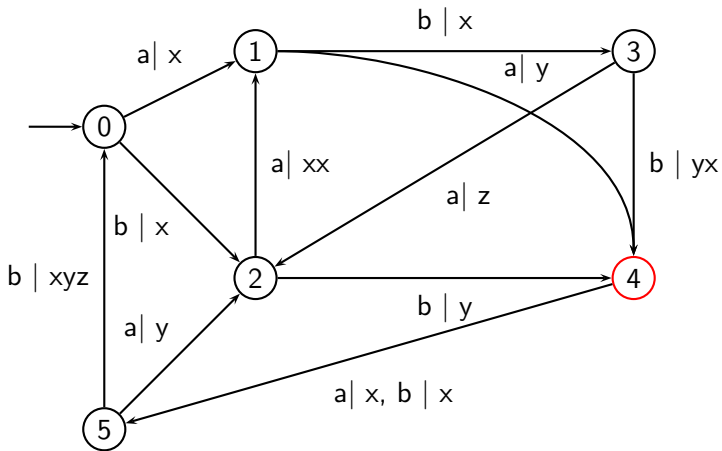


Eingabe: abbababba, Ausgabe: xxyxxxzyzxyx

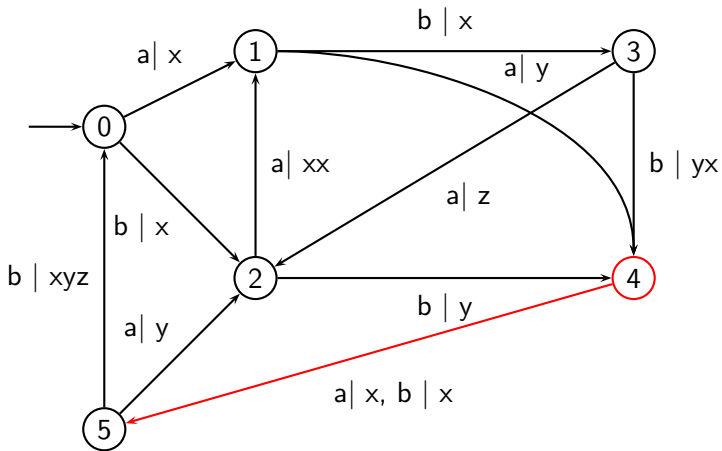




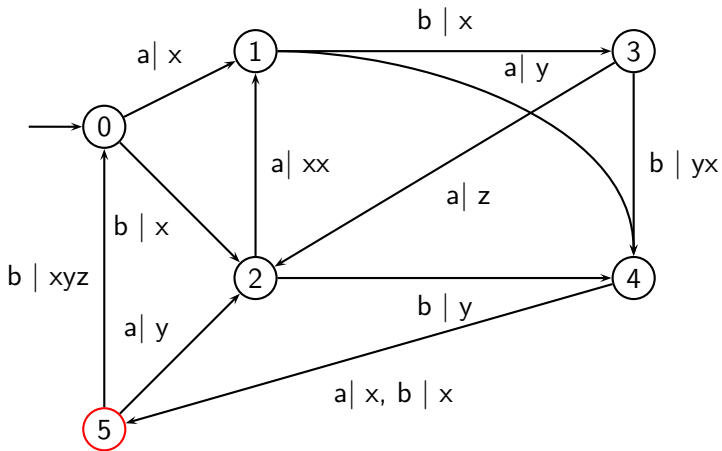
Eingabe: abbababba, Ausgabe: xxyxxxzyzxyx

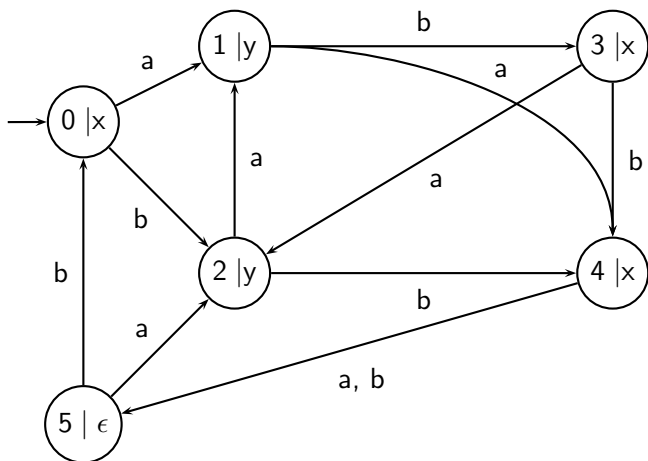


Eingabe: abbababba, Ausgabe: xxyxxxzyzxyxx

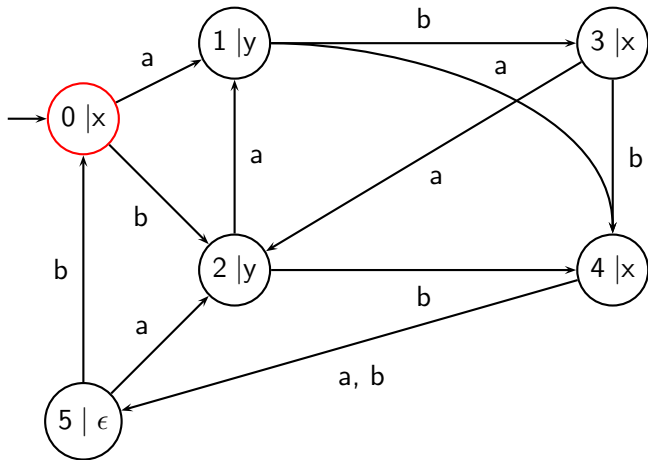


Eingabe: abbababba, Ausgabe: xxyxxxxyzxxyxx

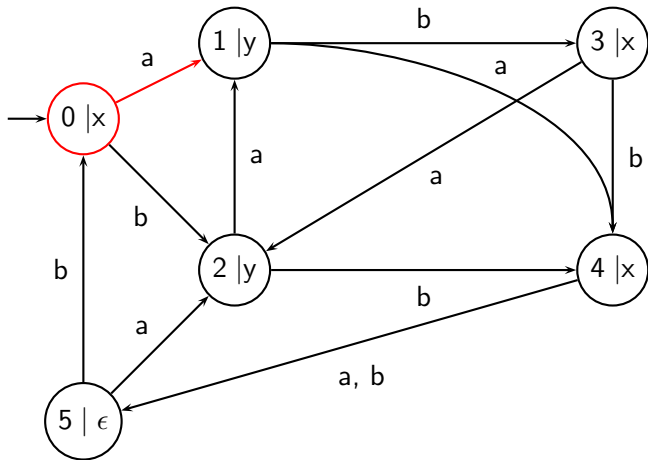




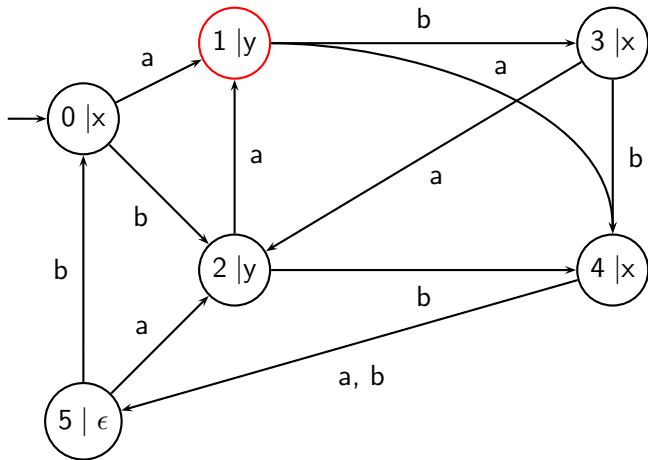
Eingabe: aabab, **Ausgabe: x**



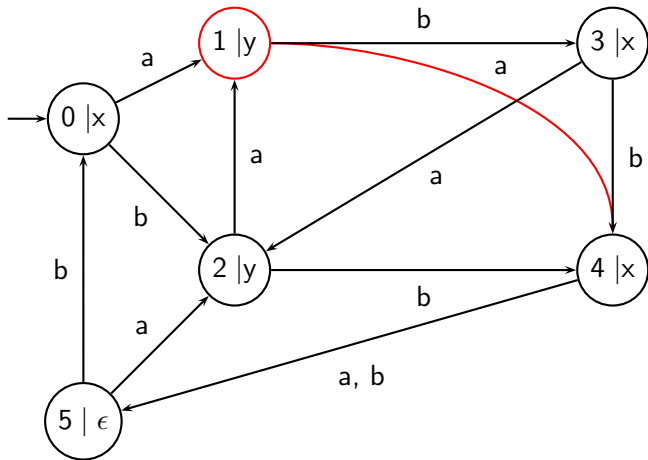
Eingabe: aabab, **Ausgabe: x**



Eingabe: aabab, Ausgabe: xy

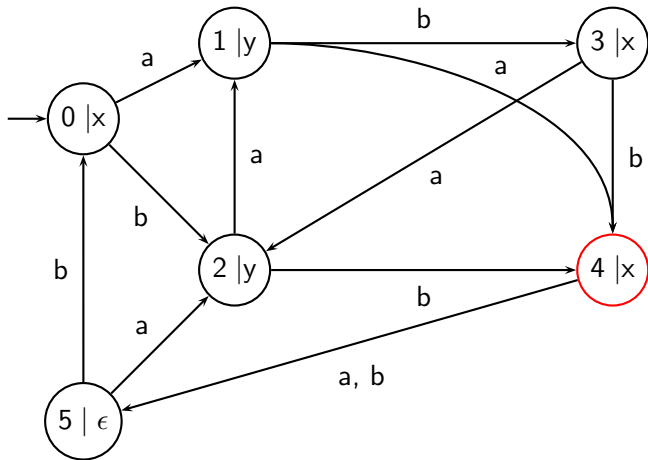


Eingabe: aabab, Ausgabe: xy

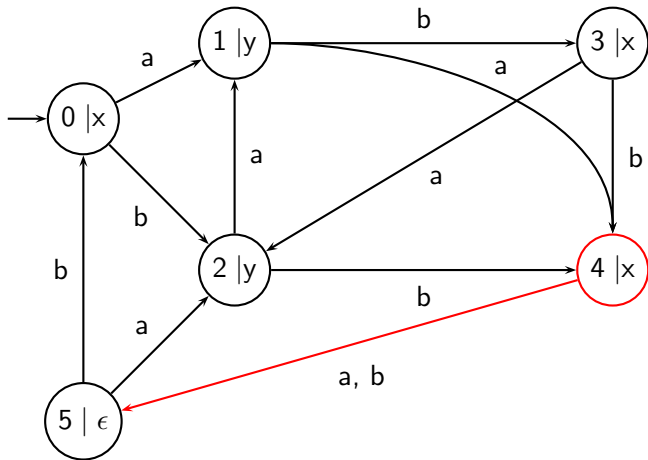




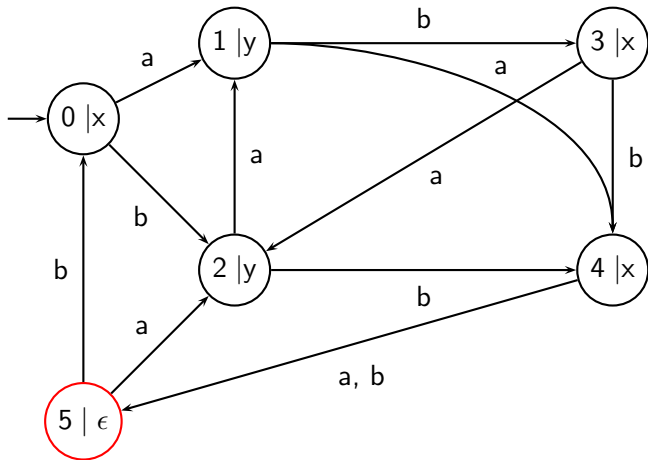
Eingabe: aabab, Ausgabe: xyx



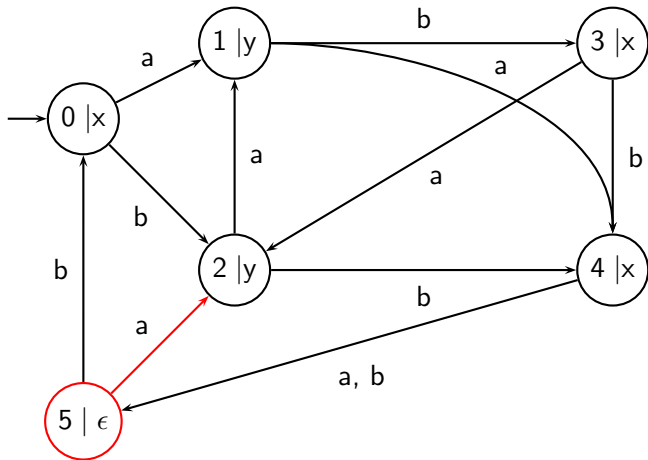
Eingabe: aabab, Ausgabe: xyx



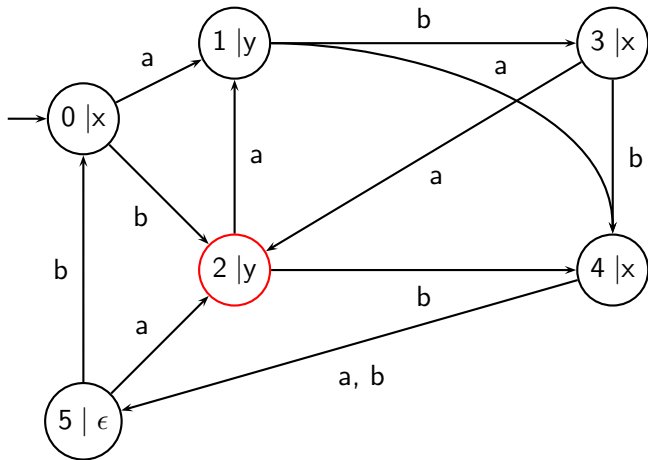
Eingabe: aabab, Ausgabe: xyx



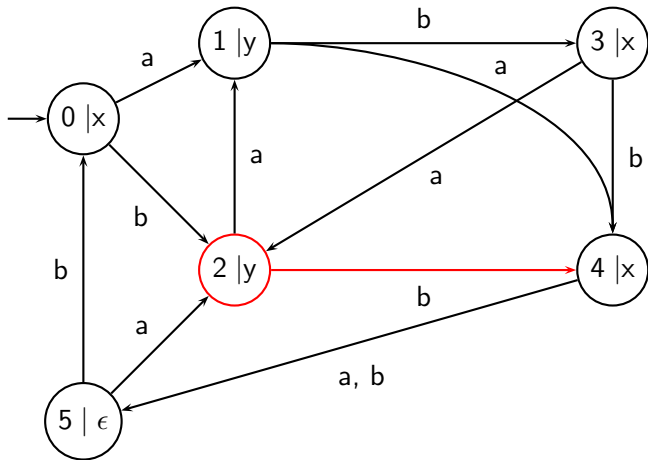
Eingabe: aabab, Ausgabe: xyx



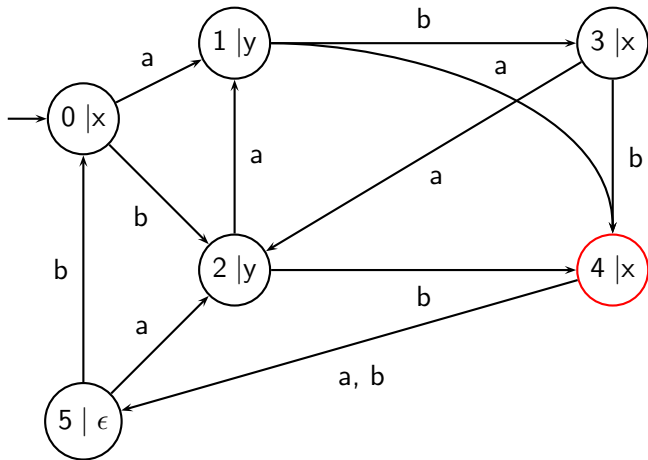
Eingabe: aabab, Ausgabe: xyxy



Eingabe: aabab, Ausgabe: xyxy



Eingabe: aabab, Ausgabe: xyxyx



*abb* ist ein bööööses Wort!



*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Eingabealphabet  $X = \{a, b, c\}$ , Ausgabealphabet  $Y = \{a, b, c, x\}$ .

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Eingabealphabet  $X = \{a, b, c\}$ , Ausgabealphabet  $Y = \{a, b, c, x\}$ .

Betrachte Wort *cbbaababbc*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*c**bbaababbc* Alles okay

→ Ausgabe *c*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*c****b****baababbc* Alles okay

→ Ausgabe *cb*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cb****b****aababbc* Alles okay

→ Ausgabe *cbb*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbb*a*babbbc* Könnte kritisch sein

→ Ausgabe *cbb*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbba*a*babbc* Könnte kritisch sein, aber *a* davor okay

→ Ausgabe *cbba*



*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbbaa**b**abbc* Alarmstufe **Rot**

→ Ausgabe *cbba*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbbaababb* Könnte kritisch sein, aber *ab* war okay

→ Ausgabe *cbbaab*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbbaaba**bbc* Alarmstufe Rot

→ Ausgabe *cbbaab*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbbaababbc* Böses Wort, ersetzen!

→ Ausgabe *cbbaabxxx*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

*cbbaababb*c Alles Okay

→ Ausgabe *cbbaabxxxxc*

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Zustände: Alles Okay, Könnte Kritisch sein, Alarmstufe Rot, Böses Wort!

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Zustände: Alles Okay, Könnte Kritisch sein, Alarmstufe Rot, Böses Wort!

<i>f</i>	OK	KK	AR	BW
<i>a</i>	KK	KK	KK	KK
<i>b</i>	OK	AR	BW	OK
<i>c</i>	OK	OK	OK	OK

*abb* ist ein bööööses Wort!

Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Zustände: Alles Okay, Könnte Kritisch sein, Alarmstufe Rot, Böses Wort!

<i>g</i>	OK	KK	AR	BW
<i>a</i>	$\epsilon$	<i>a</i>	<i>ab</i>	$\epsilon$
<i>b</i>	<i>b</i>	$\epsilon$	<i>xxx</i>	<i>b</i>
<i>c</i>	<i>c</i>	<i>ac</i>	<i>abc</i>	<i>c</i>

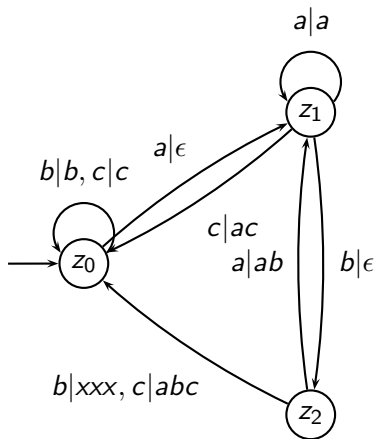


*abb* ist ein bööööses Wort!

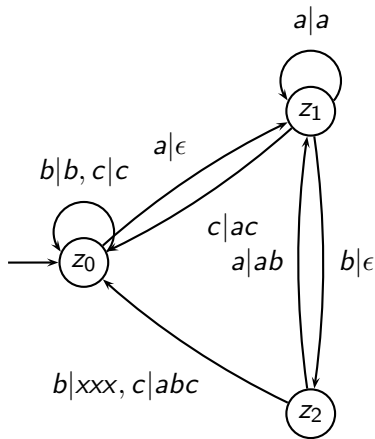
Falls *abb* in längerem Wort auftaucht, soll es durch *xxx* ersetzt werden.

Zustände: Alles Okay, Könnte Kritisch sein, Alarmstufe Rot, Böses Wort!

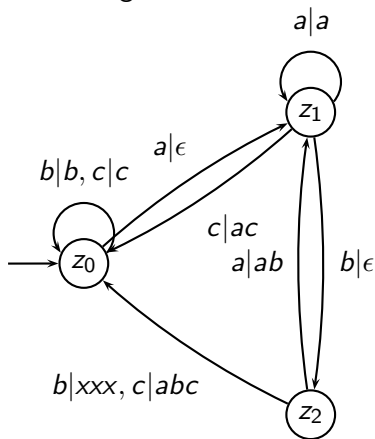
Feststellung: OK und BW machen das Gleiche!



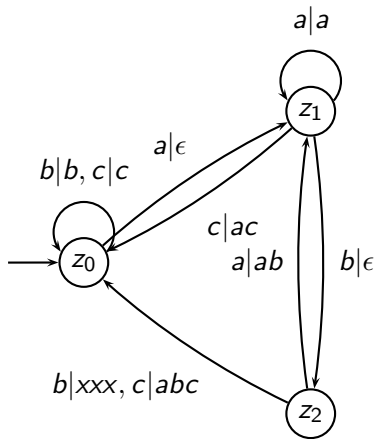
Eingabe *aababcabb*



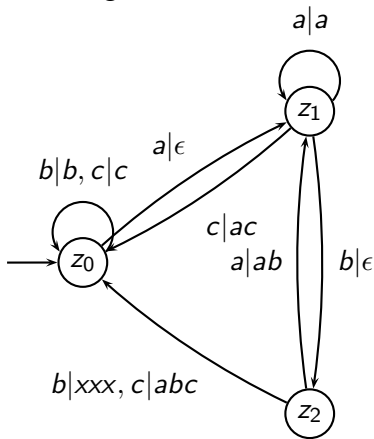
Eingabe *aababcabb* Ausgabe *aababcxxx*



Eingabe *aababcbab*



Eingabe *aababcbab* Ausgabe *aababbc*

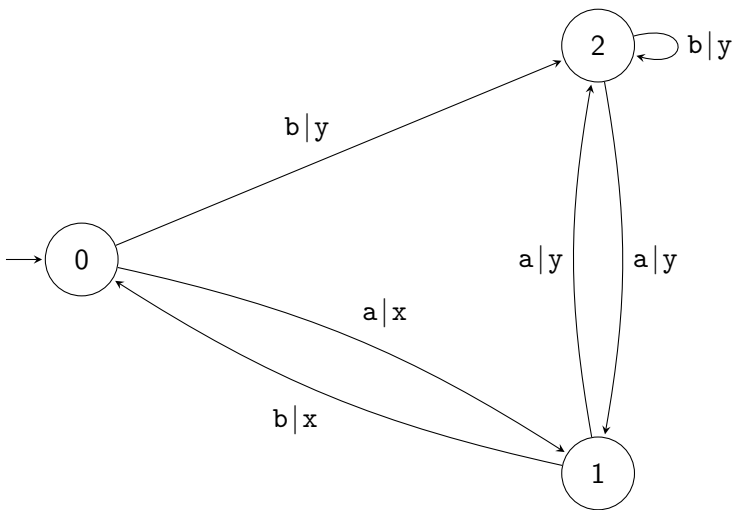


Eingabe *aababcab* Ausgabe *aababc*

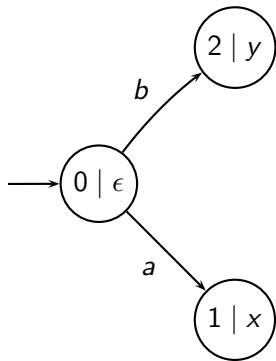
Allgemeines Problem: Endlicher Automat kann nicht feststellen, ob zuletzt gelesenes Zeichen letztes Zeichen war!

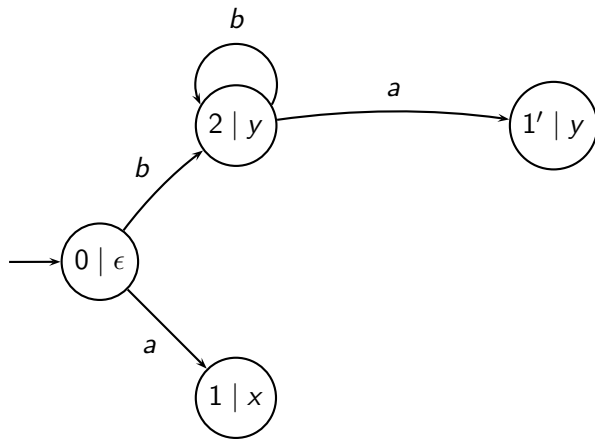
Randfälle beachten!

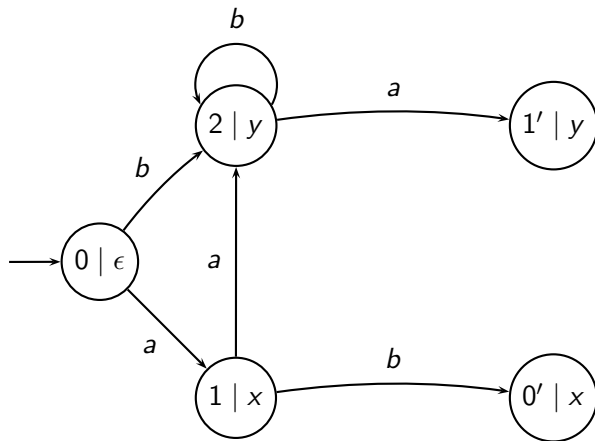
Gegeben ist folgender Mealy-Automat:



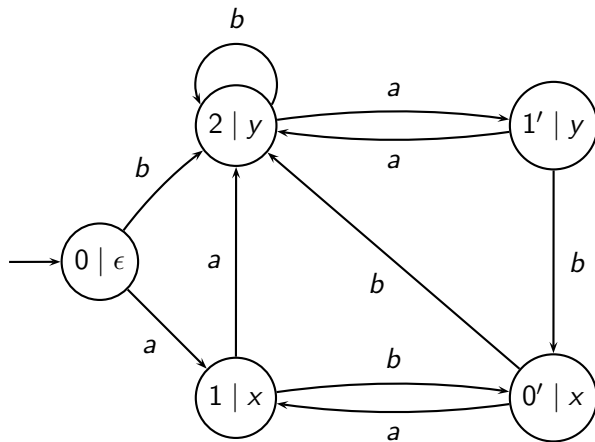








# Mealy-Automat $\leftrightarrow$ Moore-Automat



Master-Theorem

Endliche Automaten

Ergebnisse der Evaluation

- ▶ anschauliche Beispiele / Bezug zu Übungsblättern
- ▶ Übung geht nur 45 Minuten
- ▶ Folien sind online verfügbar
- ▶ Inhalte der Veranstaltung / Aufgaben
- ▶ das blaue Auge

- ▶ anschauliche Beispiele / Bezug zu Übungsblättern
- ▶ Übung geht nur 45 Minuten
- ▶ Folien sind online verfügbar
- ▶ Inhalte der Veranstaltung / Aufgaben
- ▶ das blaue Auge

- ▶ anschauliche Beispiele / Bezug zu Übungsblättern
- ▶ Übung geht nur 45 Minuten
- ▶ Folien sind online verfügbar
- ▶ Inhalte der Veranstaltung / Aufgaben
- ▶ das blaue Auge



- ▶ anschauliche Beispiele / Bezug zu Übungsblättern
- ▶ Übung geht nur 45 Minuten
- ▶ Folien sind online verfügbar
- ▶ Inhalte der Veranstaltung / Aufgaben
- ▶ das blaue Auge

- ▶ anschauliche Beispiele / Bezug zu Übungsblättern
- ▶ Übung geht nur 45 Minuten
- ▶ Folien sind online verfügbar
- ▶ Inhalte der Veranstaltung / Aufgaben
- ▶ das blaue Auge

## Nicht gefallen hat:

- ▶ Übung geht nur 45 Minuten
- ▶ Folien (Struktur, Schriftart)
- ▶ keine guten Beispiele
- ▶ Geschwindigkeit
- ▶ Mikrofon zu leise, Nuscheln, monotone Stimme, zu laut im Hörsaal

## Nicht gefallen hat:

- ▶ Übung geht nur 45 Minuten
- ▶ Folien (Struktur, Schriftart)
- ▶ keine guten Beispiele
- ▶ Geschwindigkeit
- ▶ Mikrofon zu leise, Nuscheln, monotone Stimme, zu laut im Hörsaal

## Nicht gefallen hat:

- ▶ Übung geht nur 45 Minuten
- ▶ Folien (Struktur, Schriftart)
- ▶ keine guten Beispiele
- ▶ Geschwindigkeit
- ▶ Mikrofon zu leise, Nuscheln, monotone Stimme, zu laut im Hörsaal

## Nicht gefallen hat:

- ▶ Übung geht nur 45 Minuten
- ▶ Folien (Struktur, Schriftart)
- ▶ keine guten Beispiele
- ▶ Geschwindigkeit
- ▶ Mikrofon zu leise, Nuscheln, monotone Stimme, zu laut im Hörsaal

## Nicht gefallen hat:

- ▶ Übung geht nur 45 Minuten
- ▶ Folien (Struktur, Schriftart)
- ▶ keine guten Beispiele
- ▶ Geschwindigkeit
- ▶ Mikrofon zu leise, Nuscheln, monotone Stimme, zu laut im Hörsaal

Themen für das zehnte Übungsblatt:

- ▶ Master-Theorem
- ▶ Mealy- und Moore-Automaten

Schöne Feiertage!