Übung "Grundbegriffe der Informatik"

Karlsruher Institut für Technologie

Matthias Schulz, Gebäude 50.34, Raum 247

email: schulz@ira.uka.de

Matthias Janke, Gebäude 50.34, Raum 249

email: matthias.janke@kit.edu

Master-Theorem

Einfachster Fall:

$T(1) = 1$
$T(n) = aT(n/b) : T(b^k) = a^k \Rightarrow T(n) = n^{\log_b a}$

.

Master-Theorem

Zweiteinfachster Fall:

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

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

.

Master-Theorem

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 $n^{\log_b a}$

.

Master-Theorem

Zweiteinfachster Fall:

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

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

Wenig ausmachen:

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

.

Master-Theorem

Zweiteinfachster Fall:

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

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

Wenig ausmachen:

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

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

.

Master-Theorem

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

.

Master-Theorem

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

.

Master-Theorem

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

$T(1) = 0$
$T(2) = 2$
$T(4) = 6$
$T(8) = 44/3$
$T(16) = 100/3$
$T(32) = 1096/15$

.

Master-Theorem

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

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

.

Master-Theorem

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

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

Hier also $T(n)/n$

.

Master-Theorem

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

$T(1)/1 = 0/1 = 0$
$T(2)/2 = 2/2 = 1$
$T(4)/4 = 6/4 = 3/2$
$T(8)/8 = (44/3)/8 = 11/6$
$T(16)/16 = (100/3)/16 = 25/12$
$T(32)/32 = (1096/15)/32 = 137/60$

.

Master-Theorem

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

Wenn man sonst nicht weiter weiß: Betrachte Differenzen!

.

Master-Theorem

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

$$T(1)/1 = 0/1 = 0$$
$$T(2)/2 = 2/2 = 1 \rightarrow \text{Differenz } 1$$
$$T(4)/4 = 6/4 = 3/2 \rightarrow \text{Differenz } 1/2$$
$$T(8)/8 = (44/3)/8 = 11/6 \rightarrow \text{Differenz } 1/3$$
$$T(16)/16 = (100/3)/16 = 25/12 \rightarrow \text{Differenz } 1/4$$
$$T(32)/32 = (1096/15)/32 = 137/60 \rightarrow \text{Differenz } 1/5$$

.

Master-Theorem

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

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

.

Master-Theorem

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

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

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

.

Master-Theorem

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

.

Master-Theorem

Dritteinfachster Fall:

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

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

.

Master-Theorem

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 $c > \log_b a$
- $af(n/b) \leq df(n)$ für $0 < d < 1$ und hinreichend große n.

.

Endliche Automaten

Zwei Arten: Moore und Mealy-Automaten

.

Endliche Automaten

Zwei Arten: Moore und Mealy-Automaten

Unterschied:

Moore-Automaten: Ausgabefunktion hat ein Argument

.

Endliche Automaten

Zwei Arten: Moore und Mealy-Automaten

Unterschied:

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

.

Endliche Automaten

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

.

Endliche Automaten

Moore/Mealy-Automat mit Anfangszustand z_0.

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

.

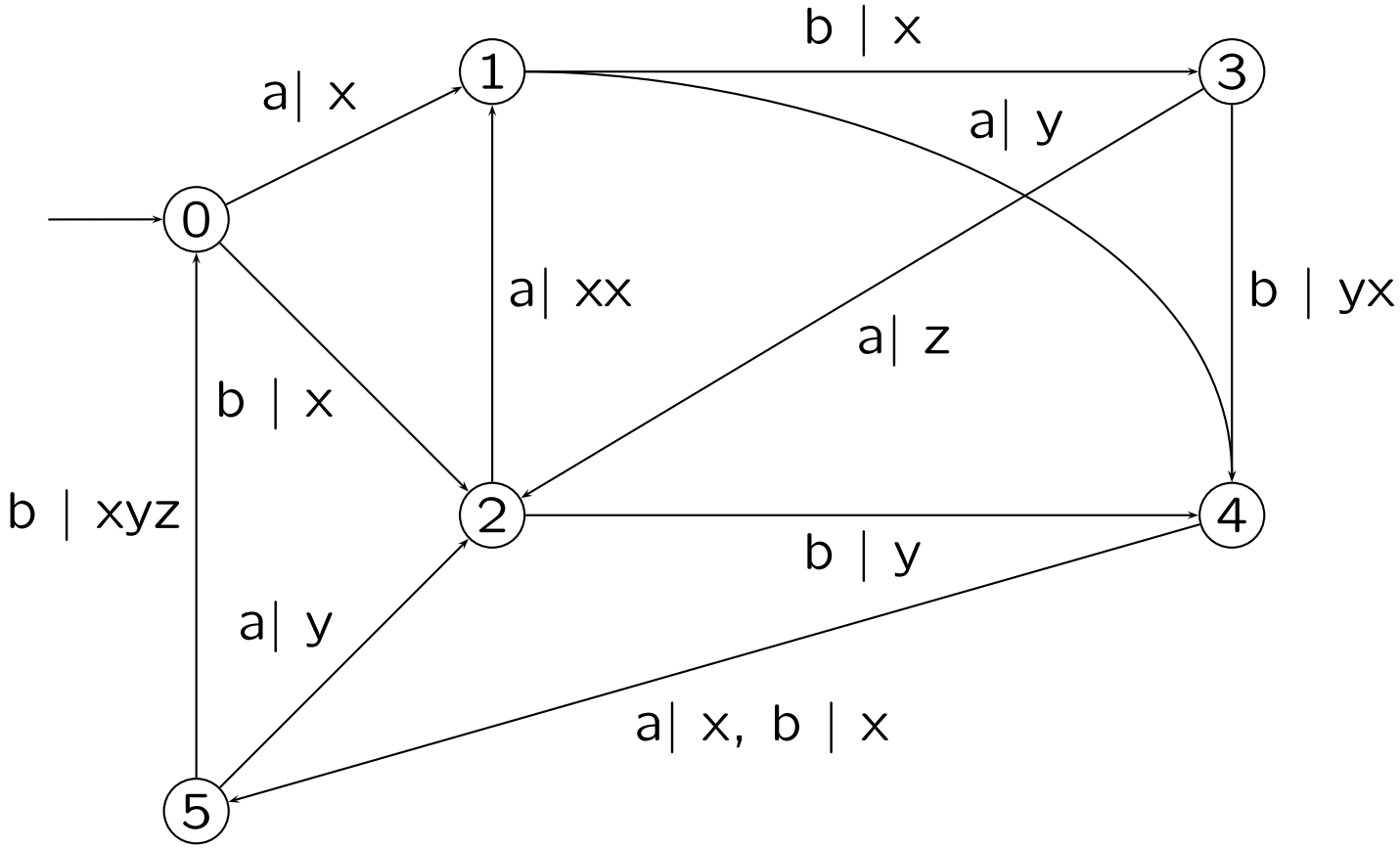
Endliche Automaten

Moore/Mealy-Automat mit Anfangszustand z_0.

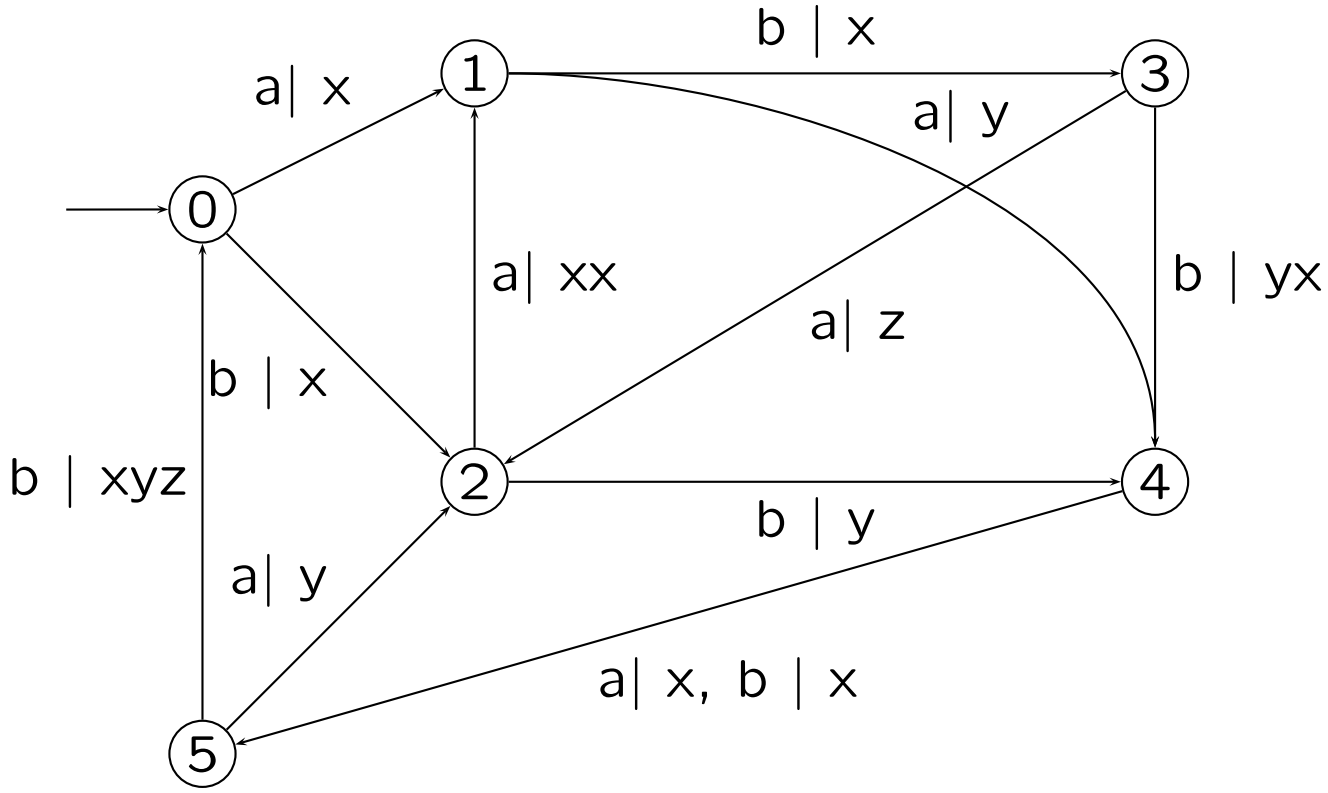
Ausgabe bei Eingabe $w \in X^*$: Üblicherweise $g^{**}(z_0, w)$

Beachten Sie das z_0!

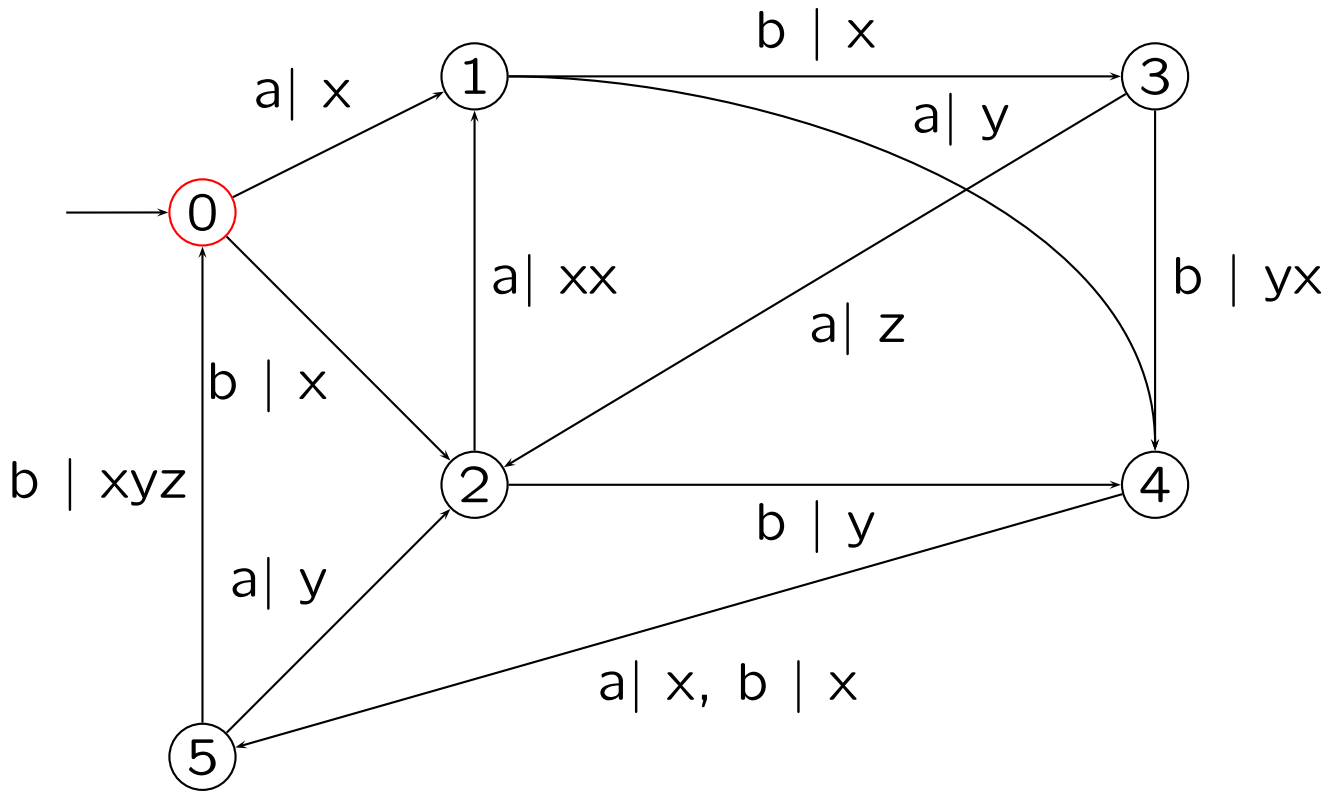
.


0
1
2
3
4
5
a| x
b | x
a| y
a| xx
b | yx
a| z
b | x
b | xyz
b | y
a| y
a| x, b | x

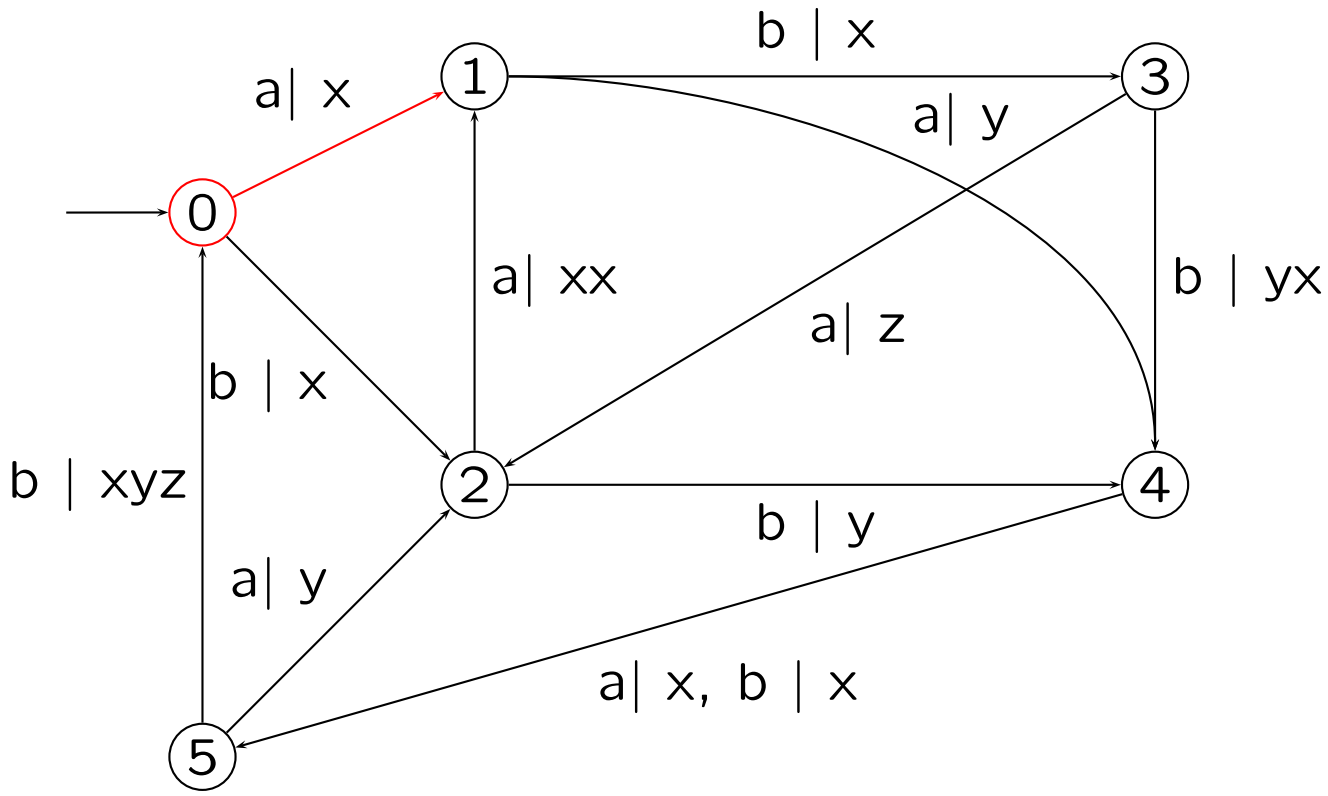
Eingabe: abbababba, Ausgabe:



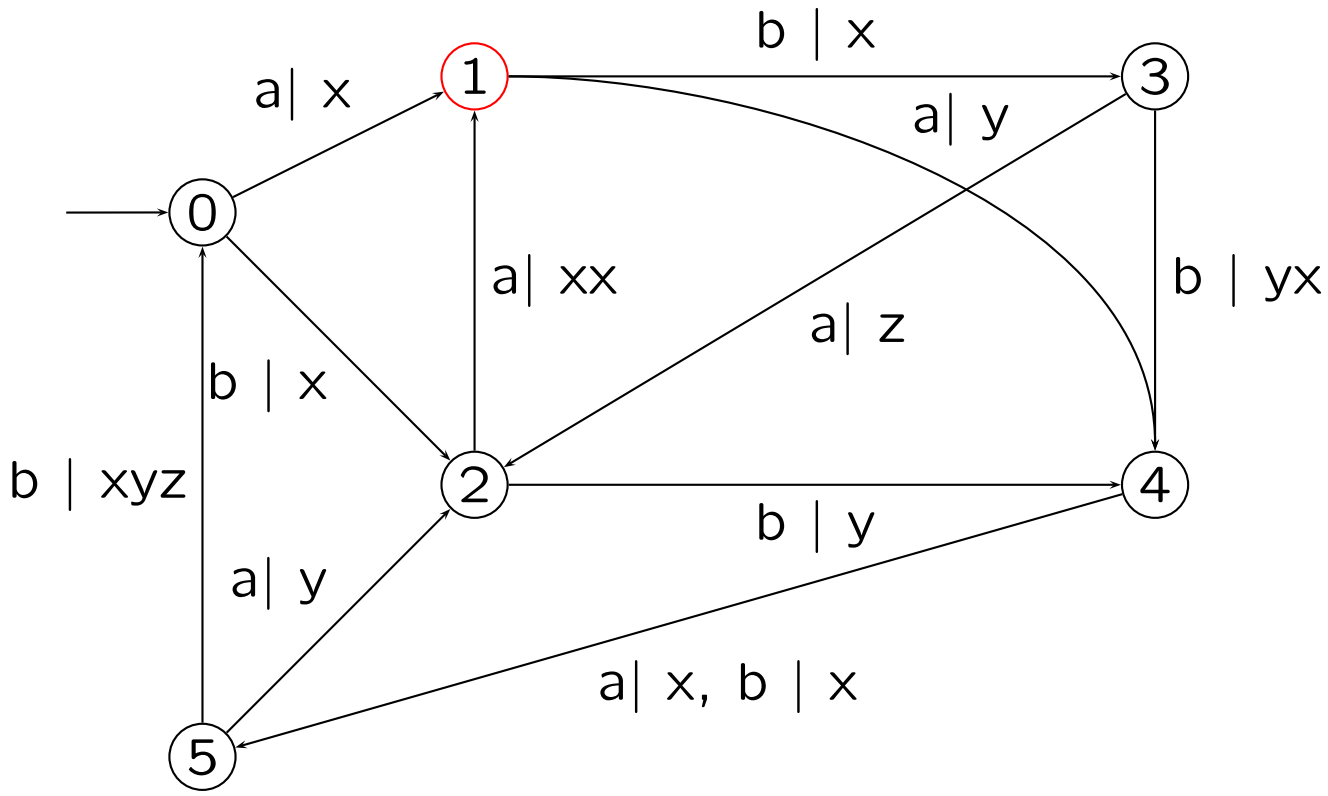
Eingabe: abbababba, Ausgabe:



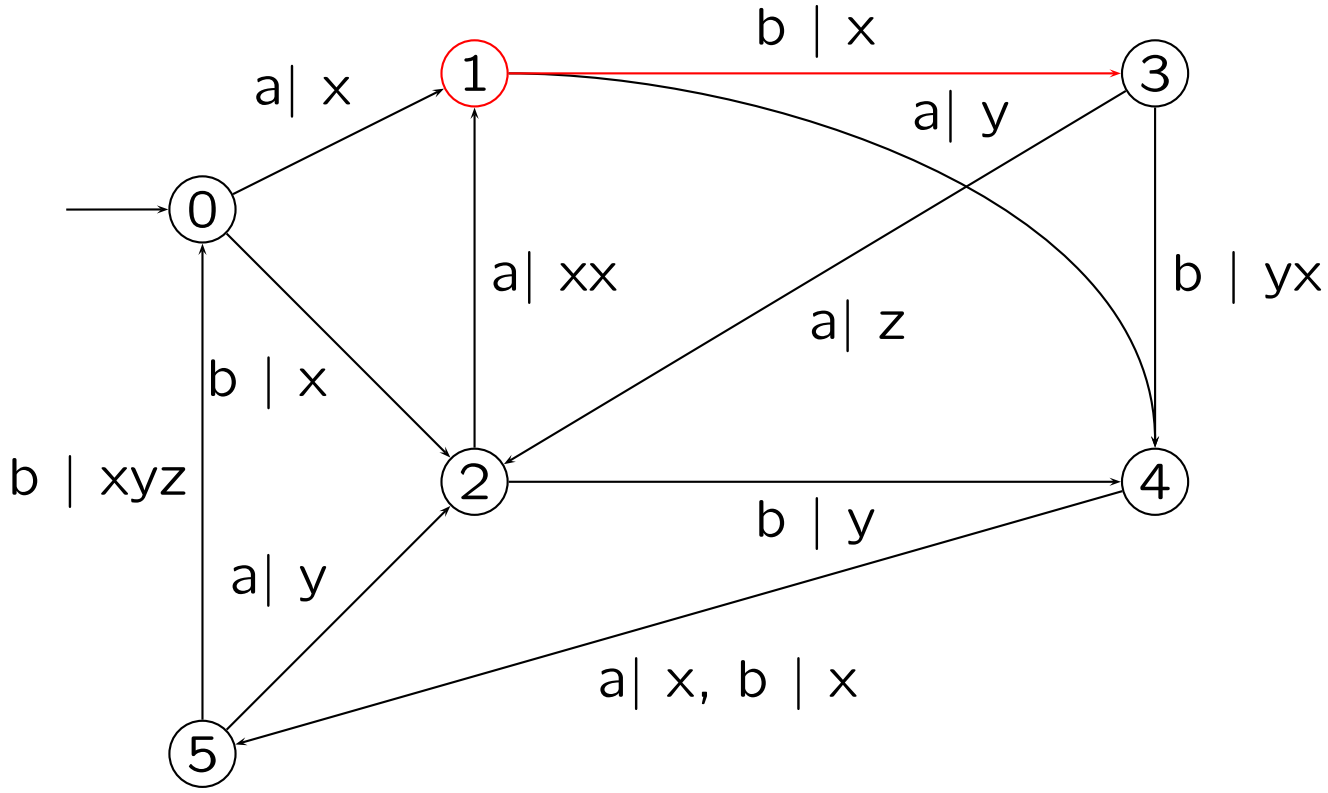
Eingabe: abbababba, Ausgabe: x



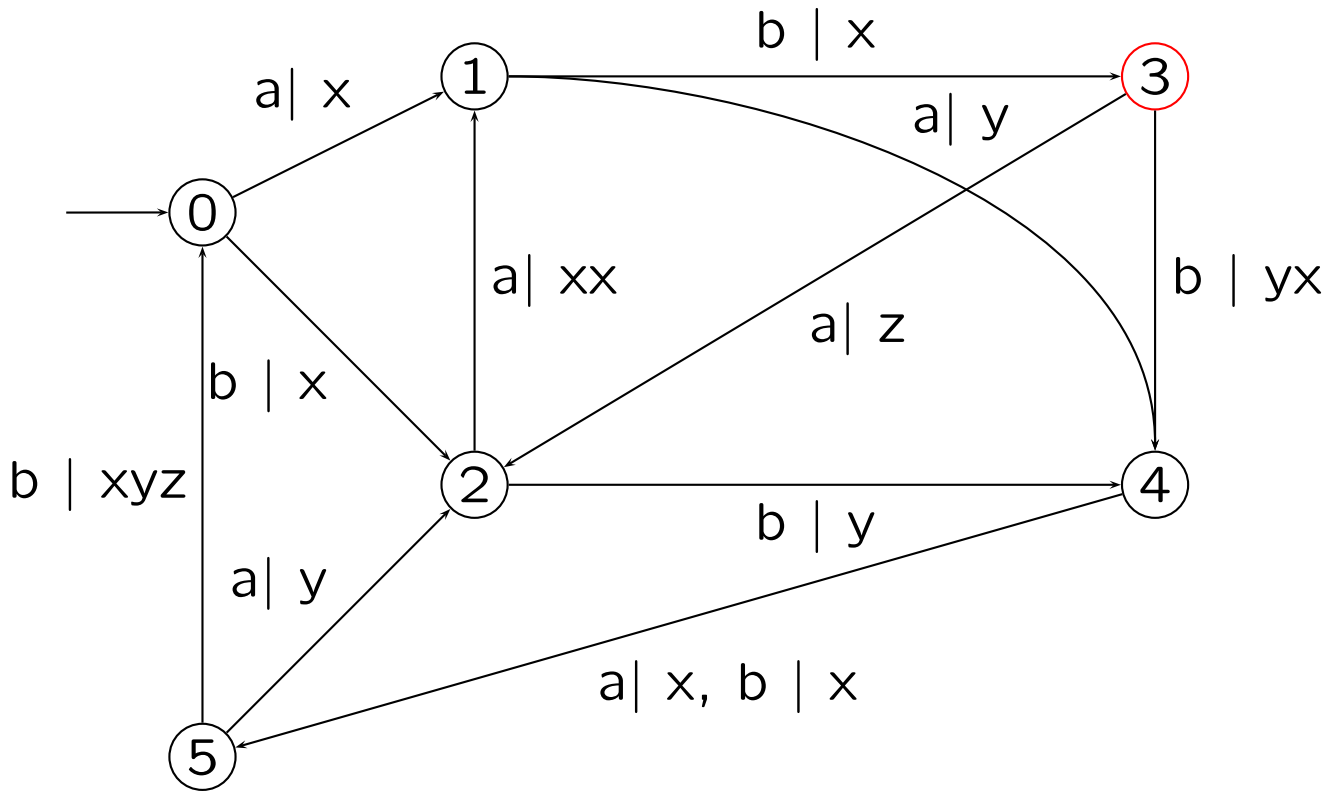
Eingabe: abbababba, Ausgabe: x



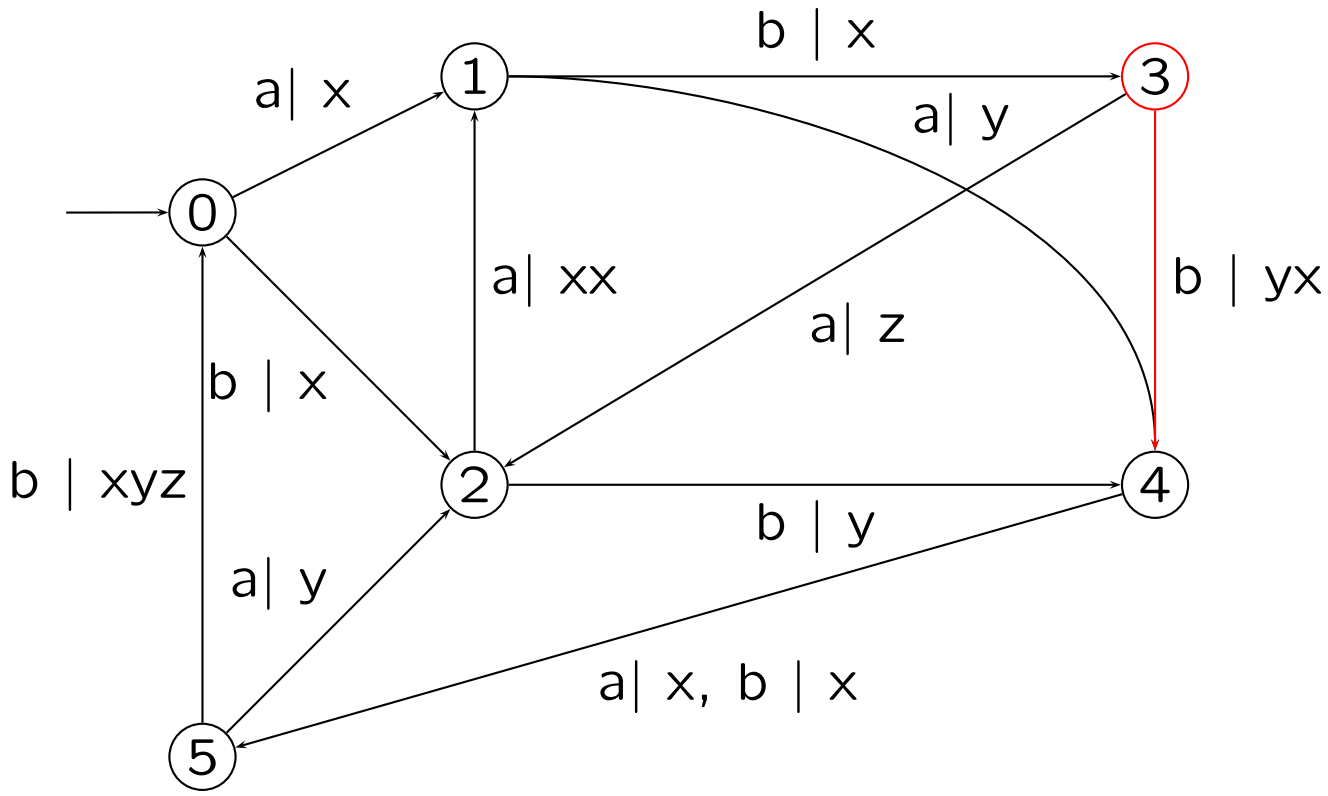
Eingabe: abbababba, Ausgabe: xx



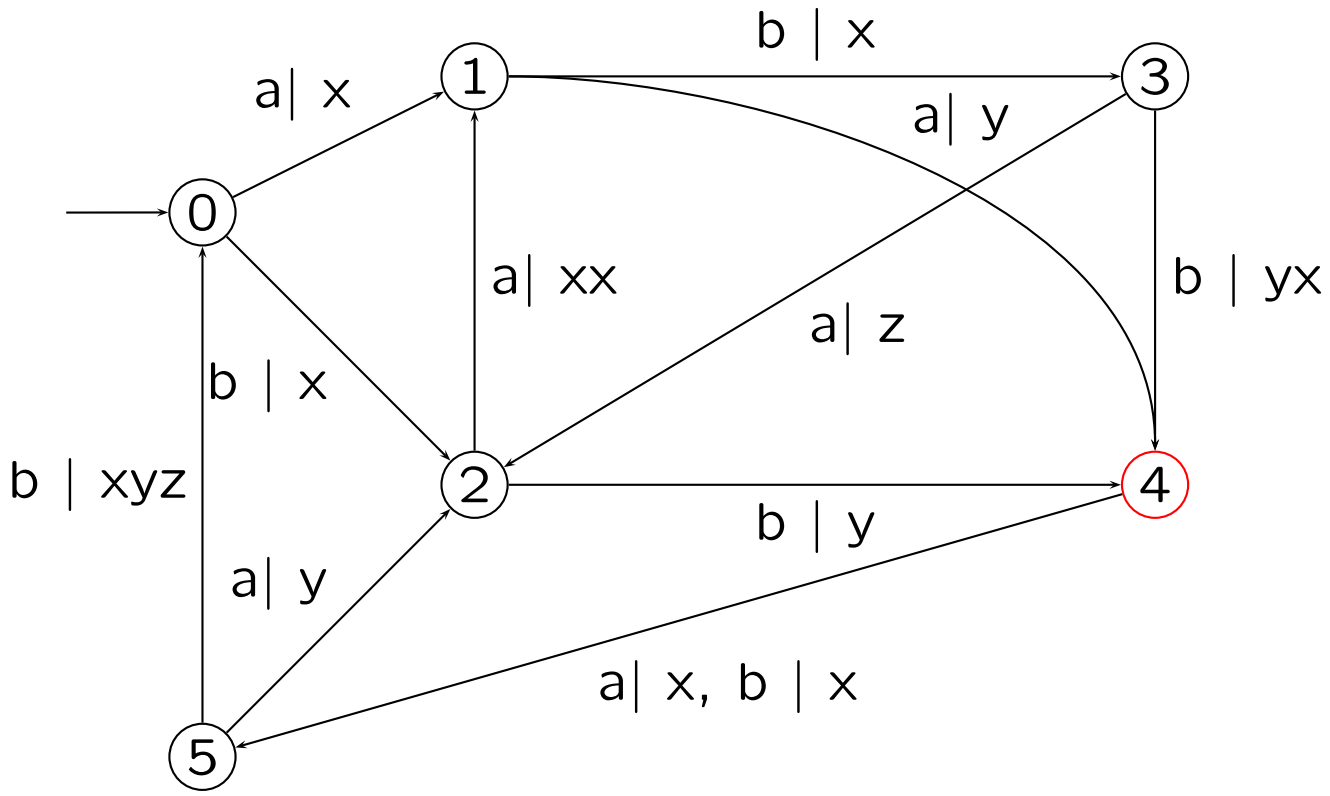
Eingabe: abbababba, Ausgabe: xx



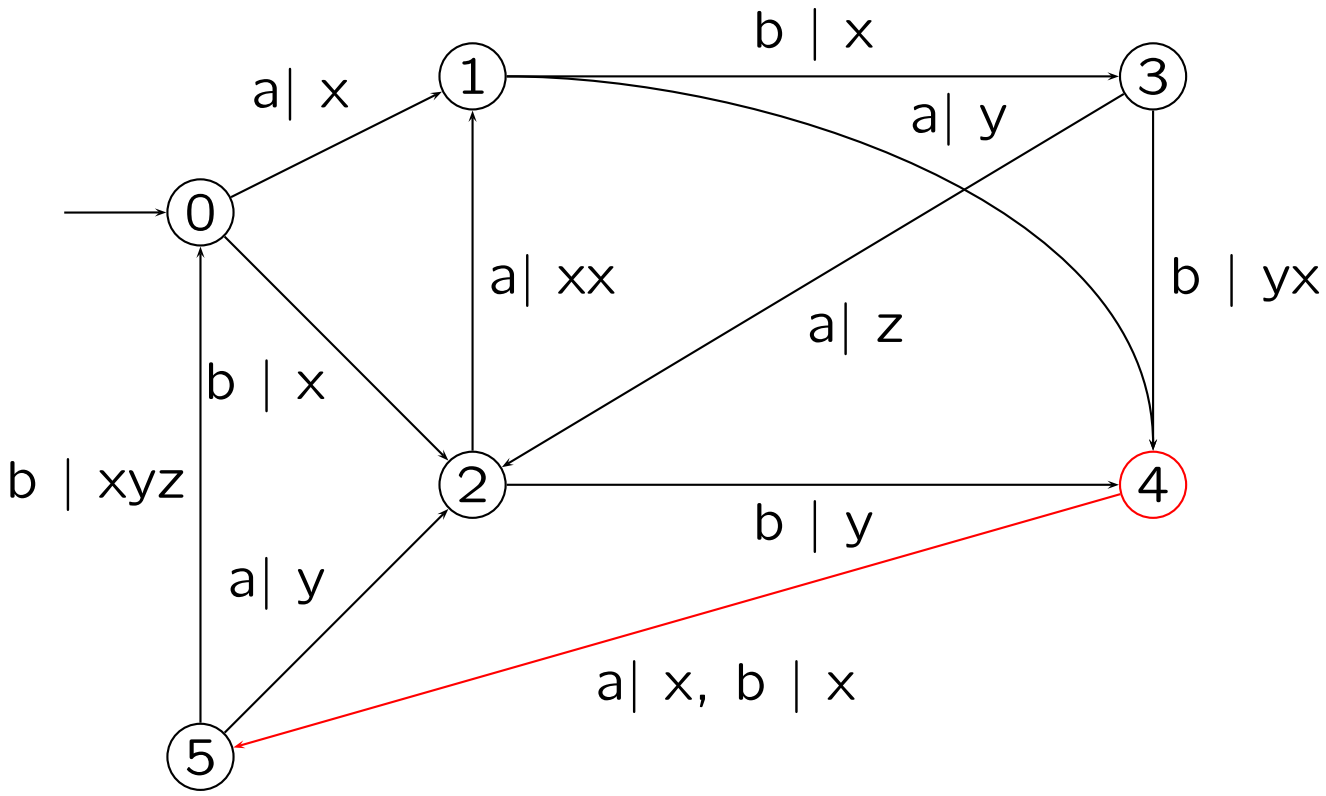
Eingabe: abbababba, Ausgabe: xxyx



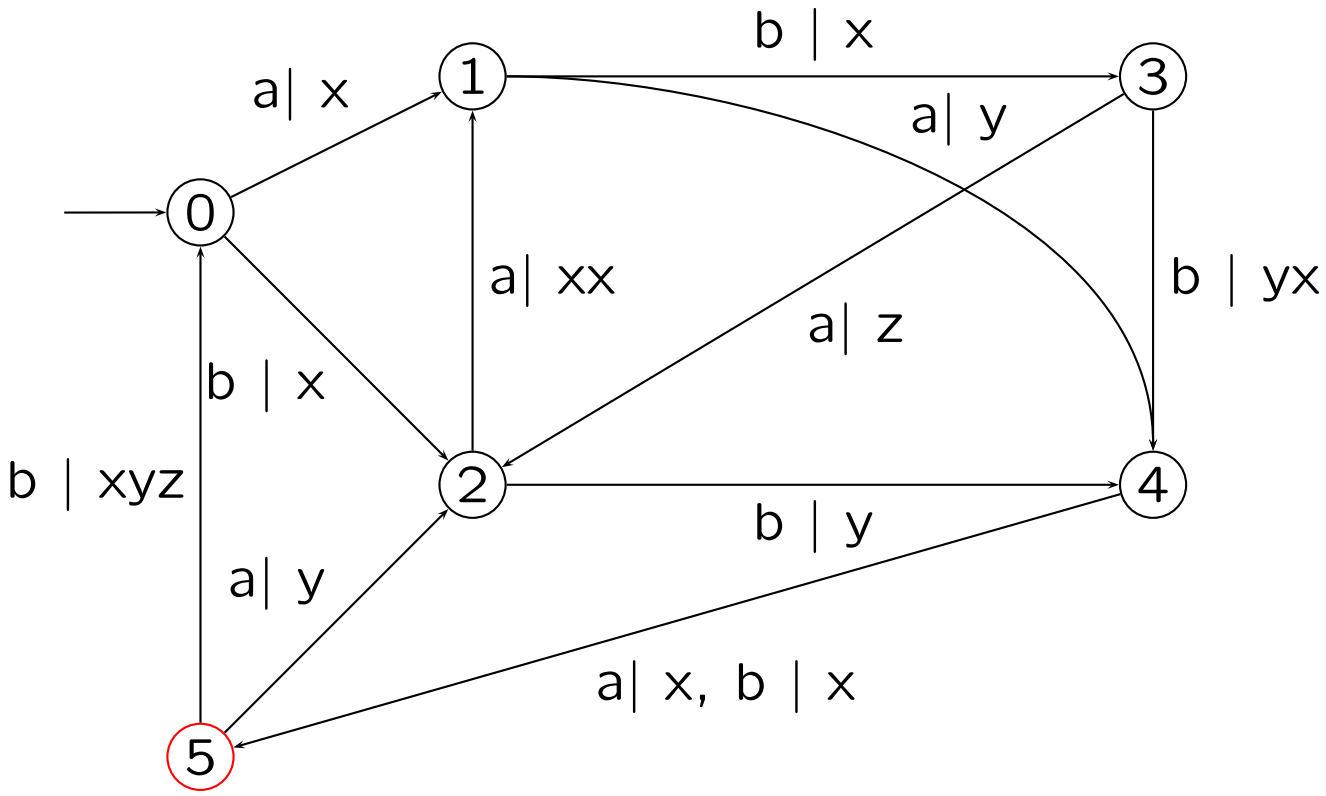
Eingabe: abbababba, Ausgabe: xxyx



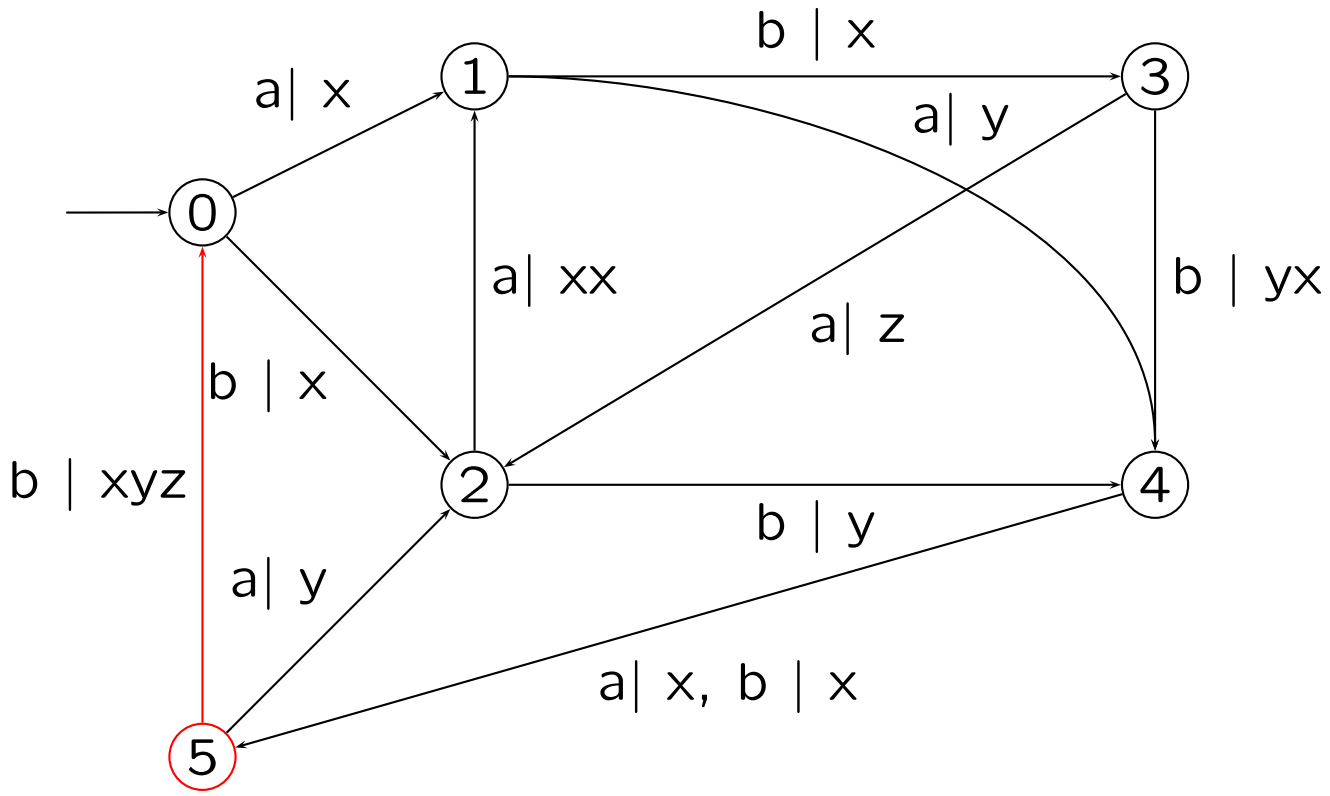
Eingabe: abbababba, Ausgabe: xxyxx



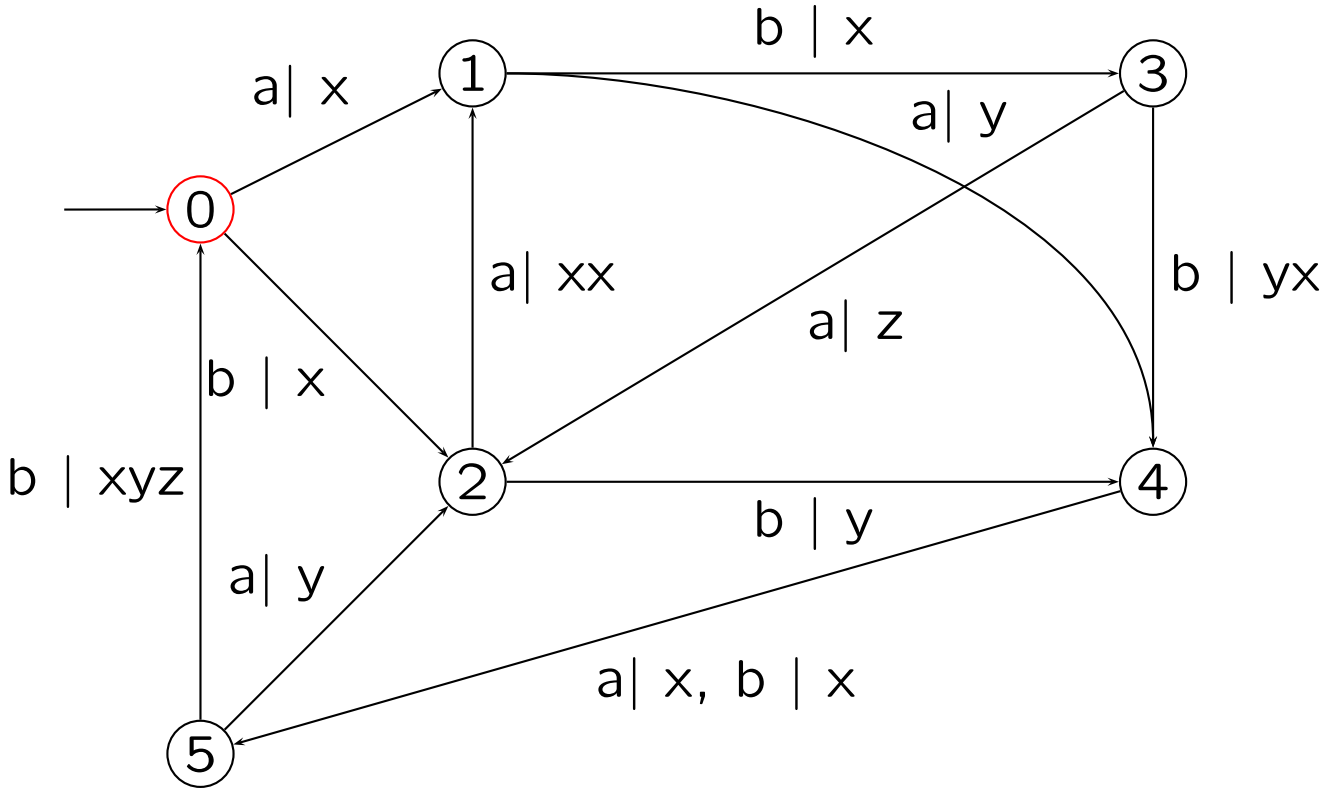
Eingabe: abbababba, Ausgabe: xxyxx



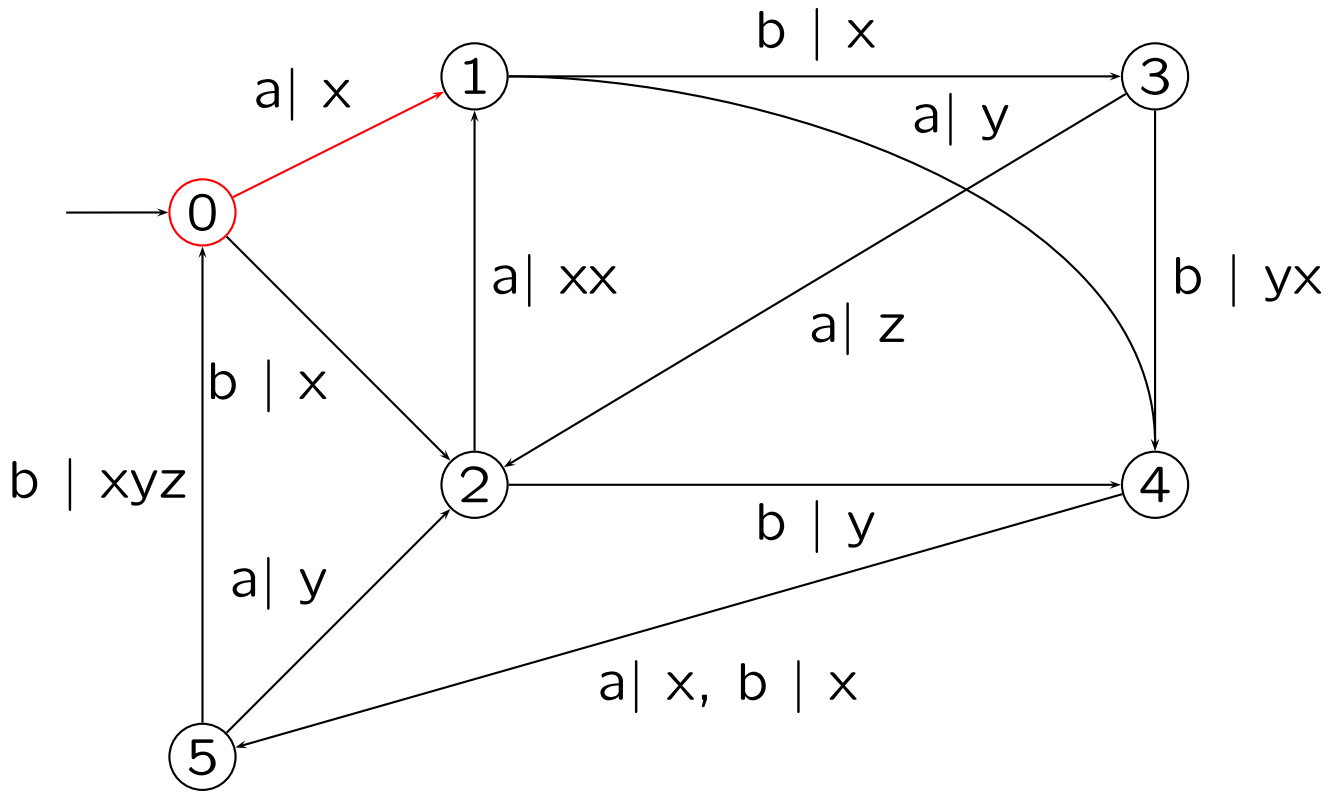
Eingabe: abbababba, Ausgabe: xxyxxxyz



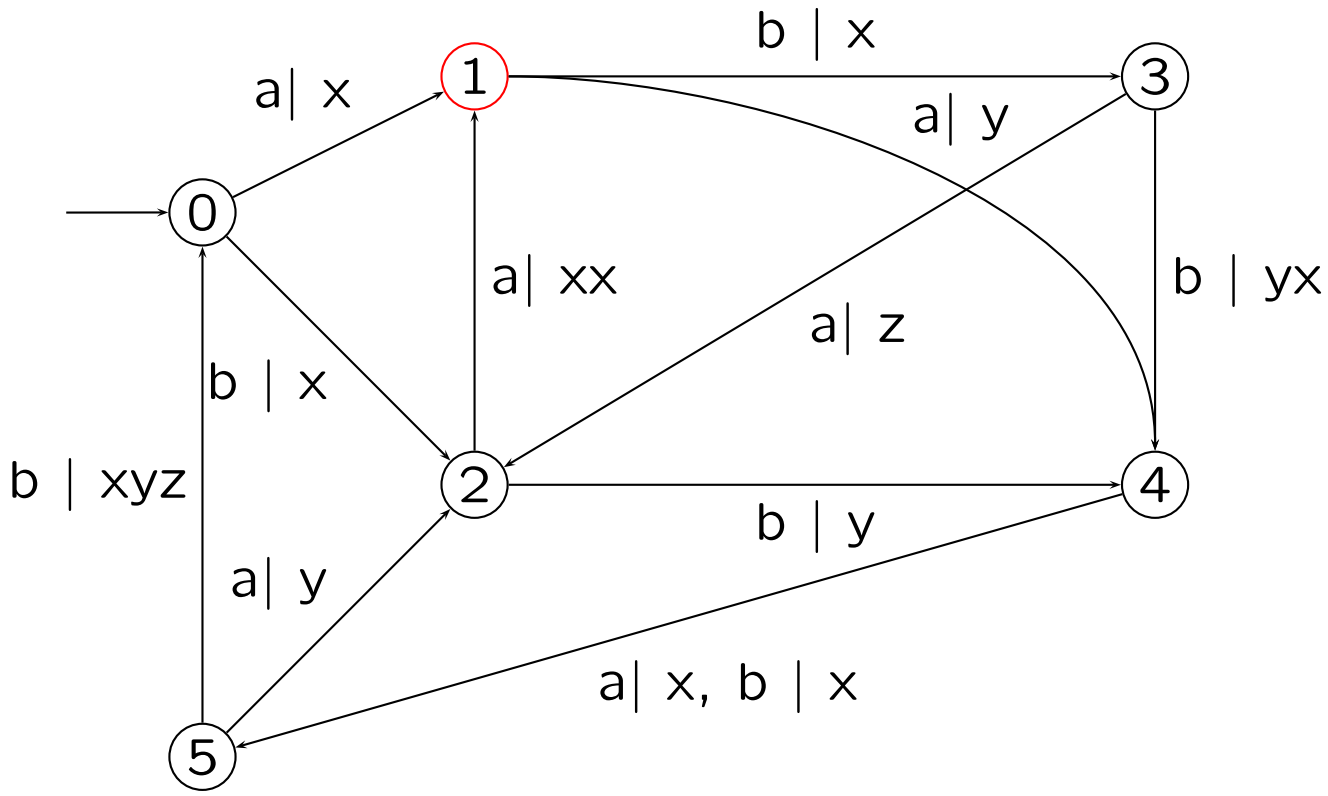
Eingabe: abbababba, Ausgabe: xxyxxxyz



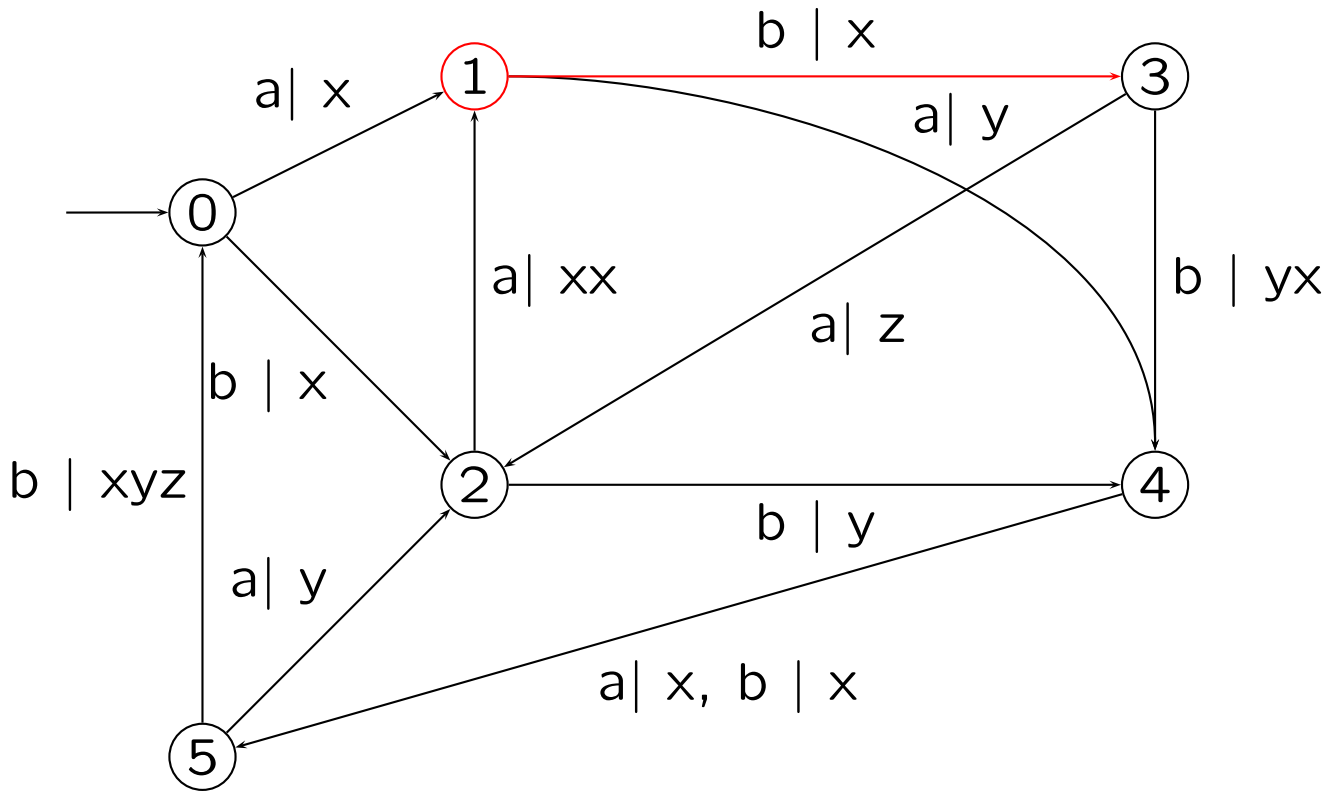
Eingabe: abbababba, Ausgabe: xxyxxxyzx



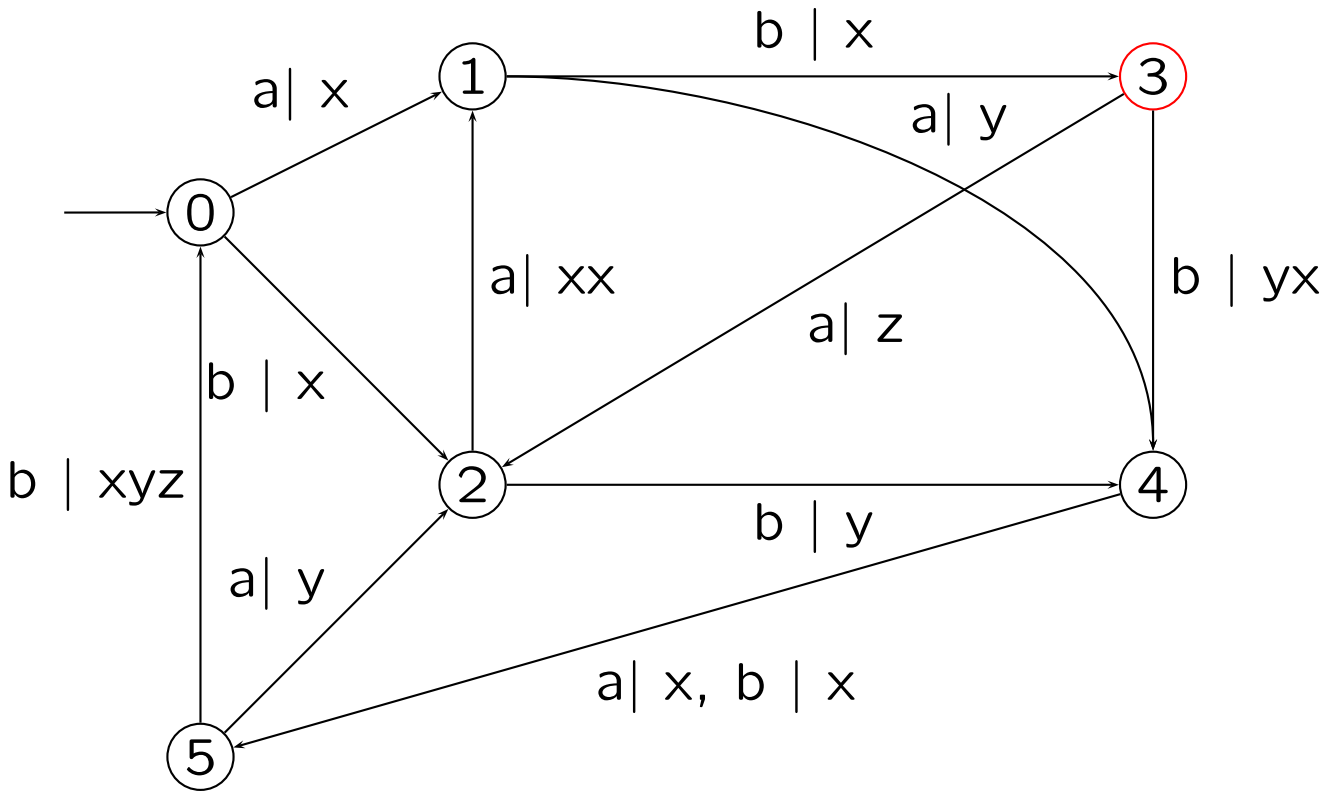
Eingabe: abbababba, Ausgabe: xxyxxxyzx



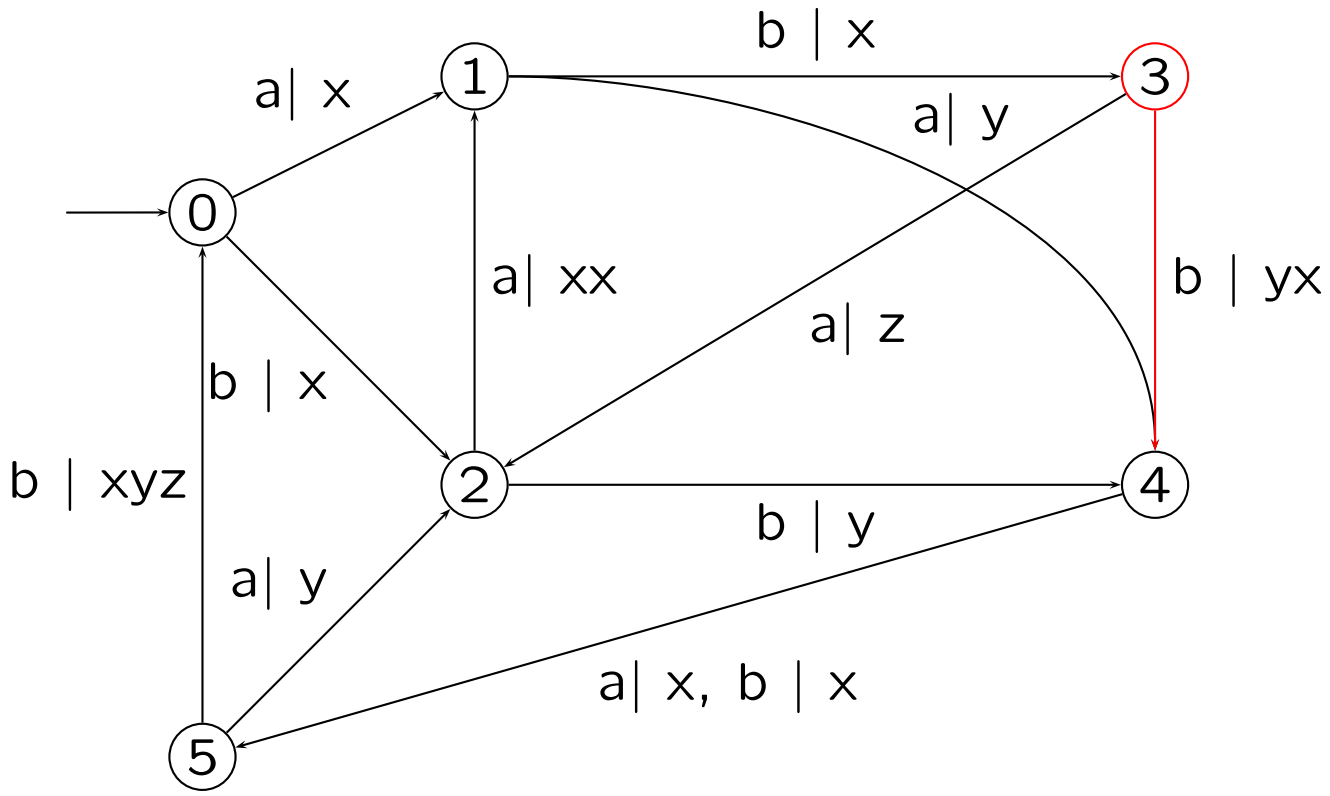
Eingabe: abbababba, Ausgabe: xxyxxxyzxx



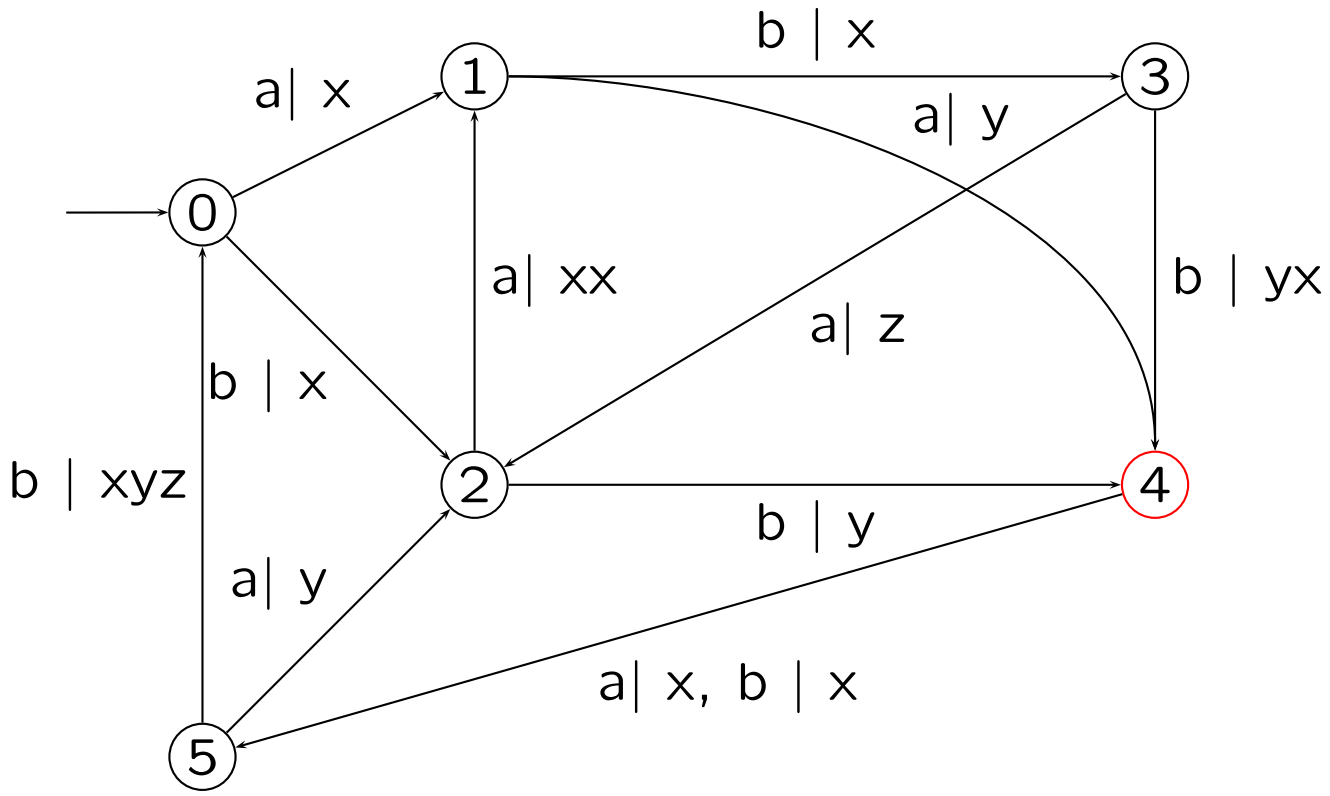
Eingabe: abbababba, Ausgabe: xxyxxxyzxx



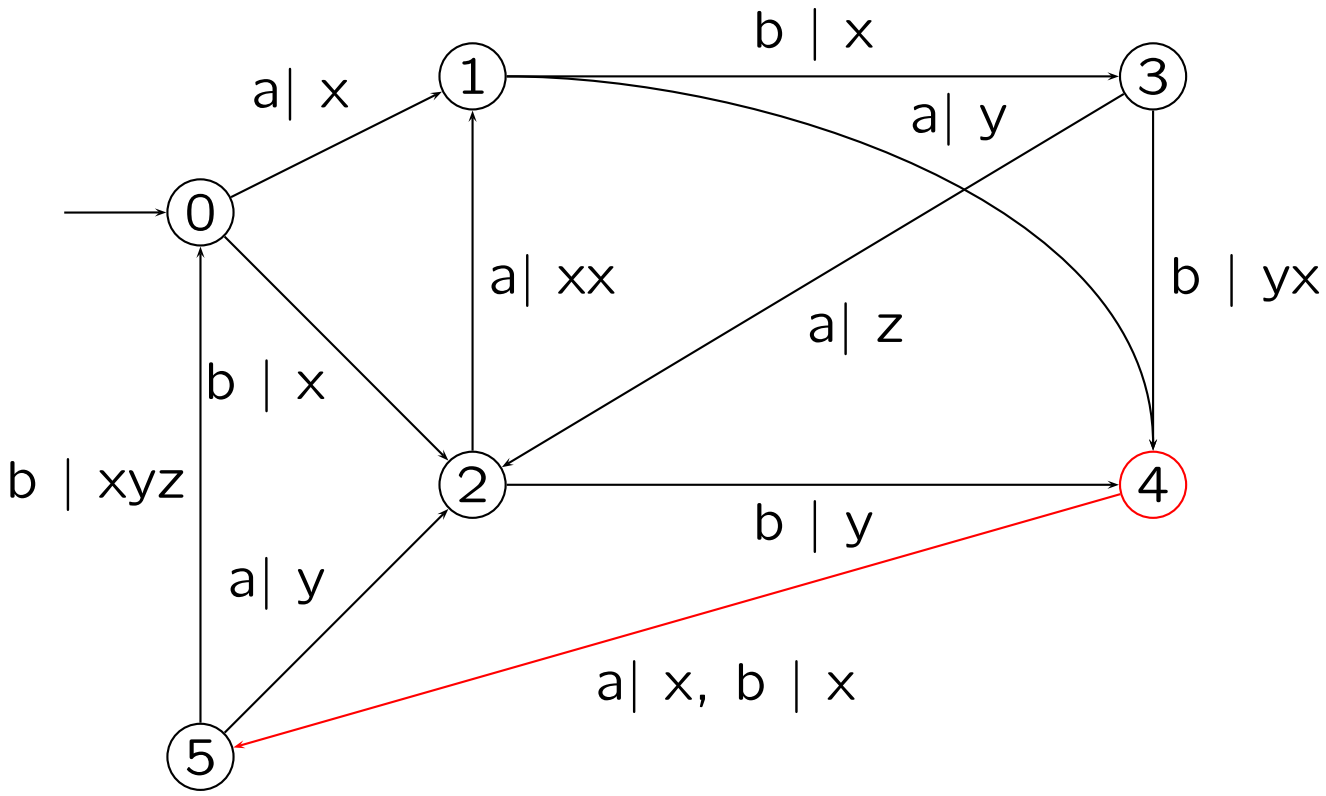
Eingabe: abbababba, Ausgabe: xxyxxxyzxxyx



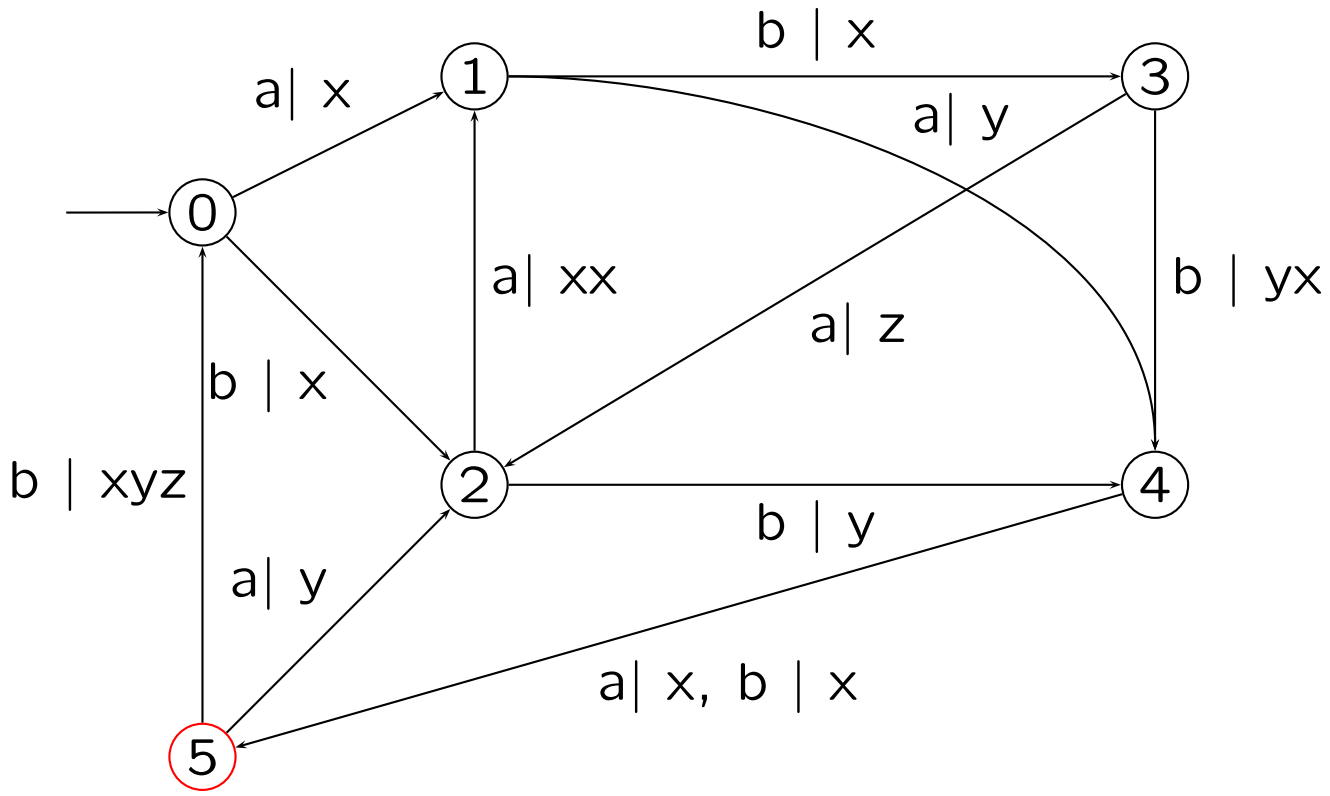
Eingabe: abbababba, Ausgabe: xxyxxxyzxxyx

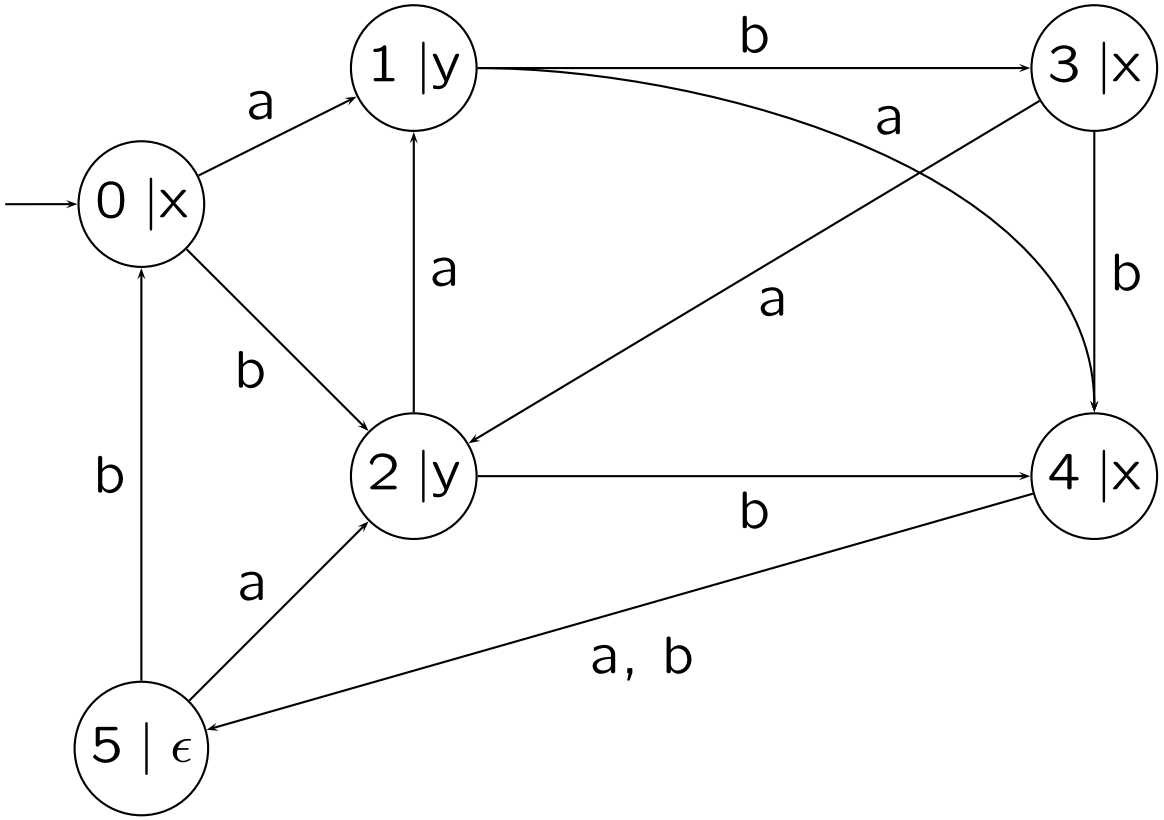


Eingabe: abbababba, Ausgabe: xxyxxxyzxxyxx

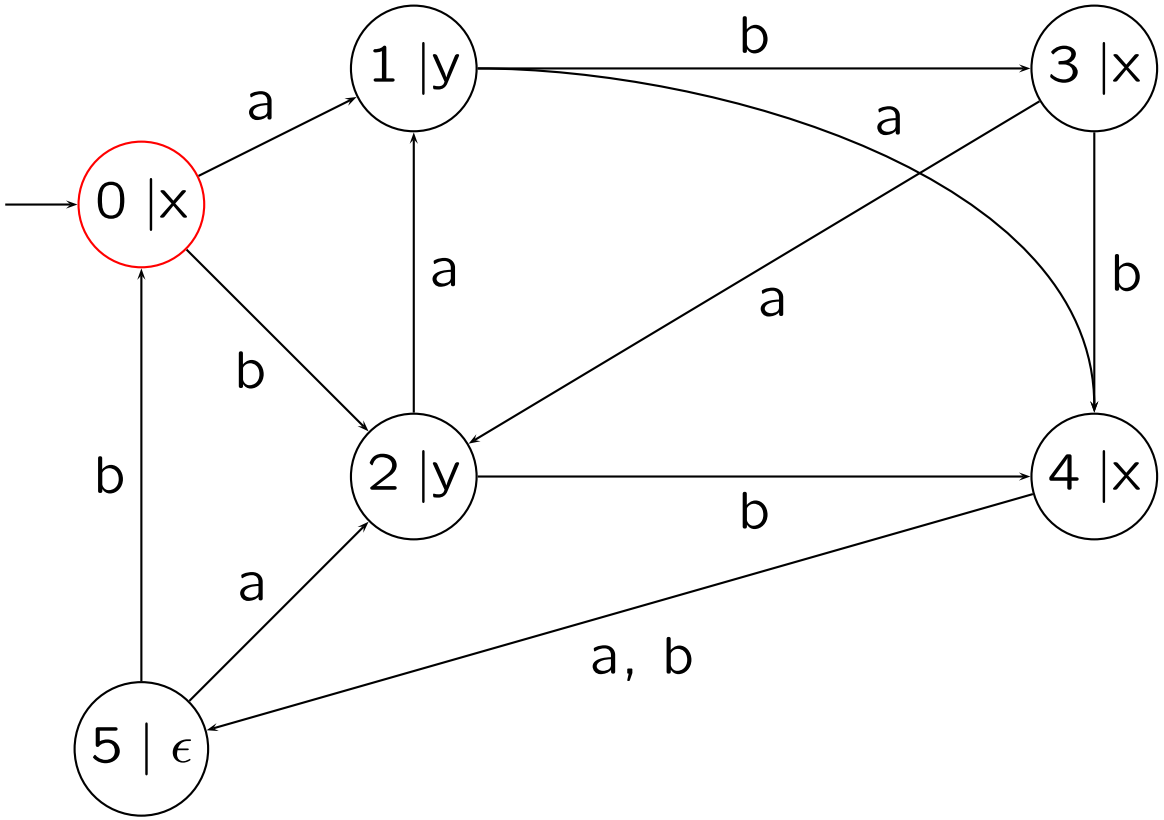


Eingabe: abbababba, Ausgabe: xxyxxxyzxxyxx

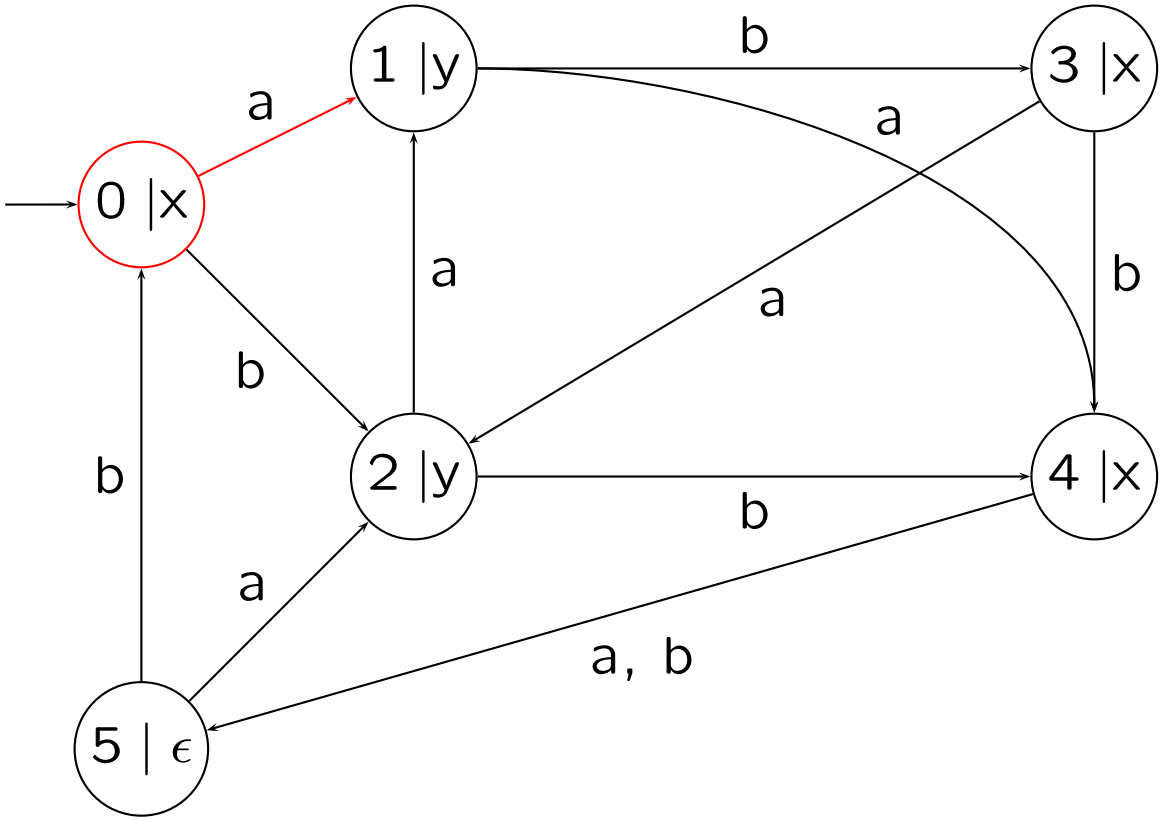



1 |y
b
3 |x
a
a
0 |x
a
b
a
b
b
2 |y
4 |x
b
a
a, b
5 | ϵ

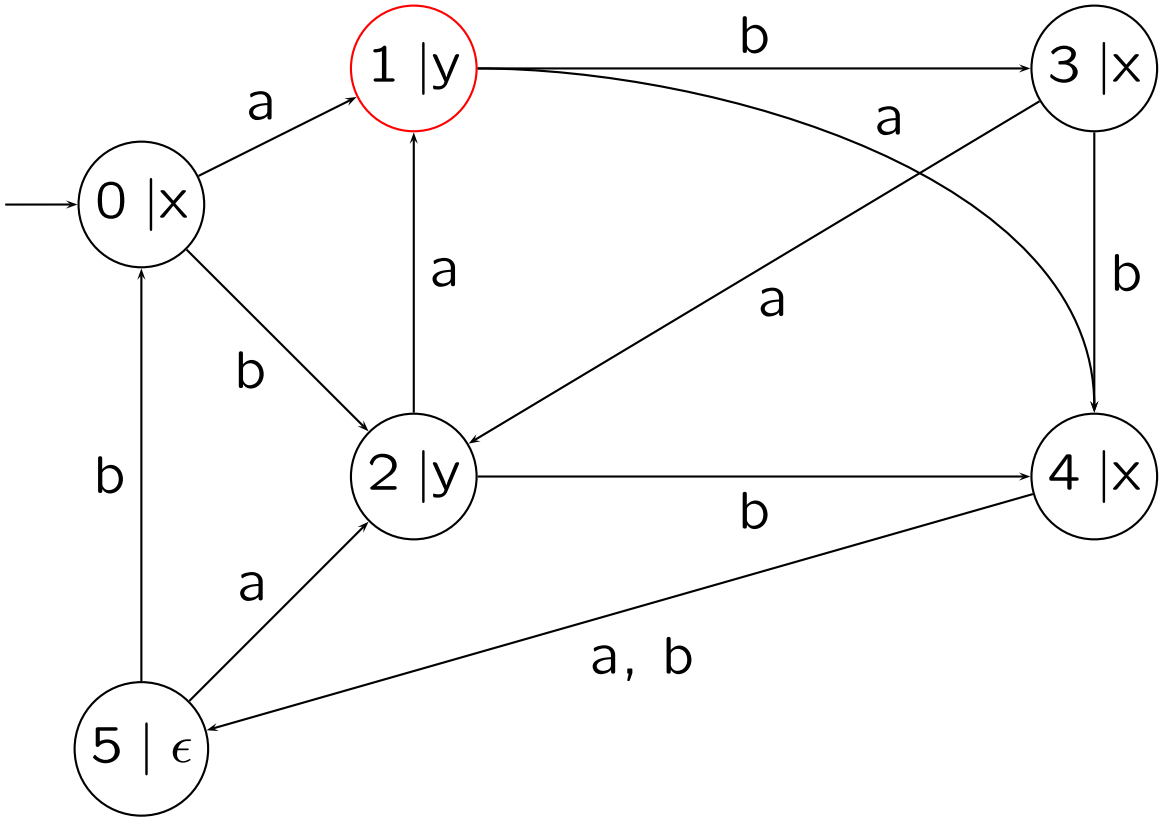
Eingabe: aabab, Ausgabe: x



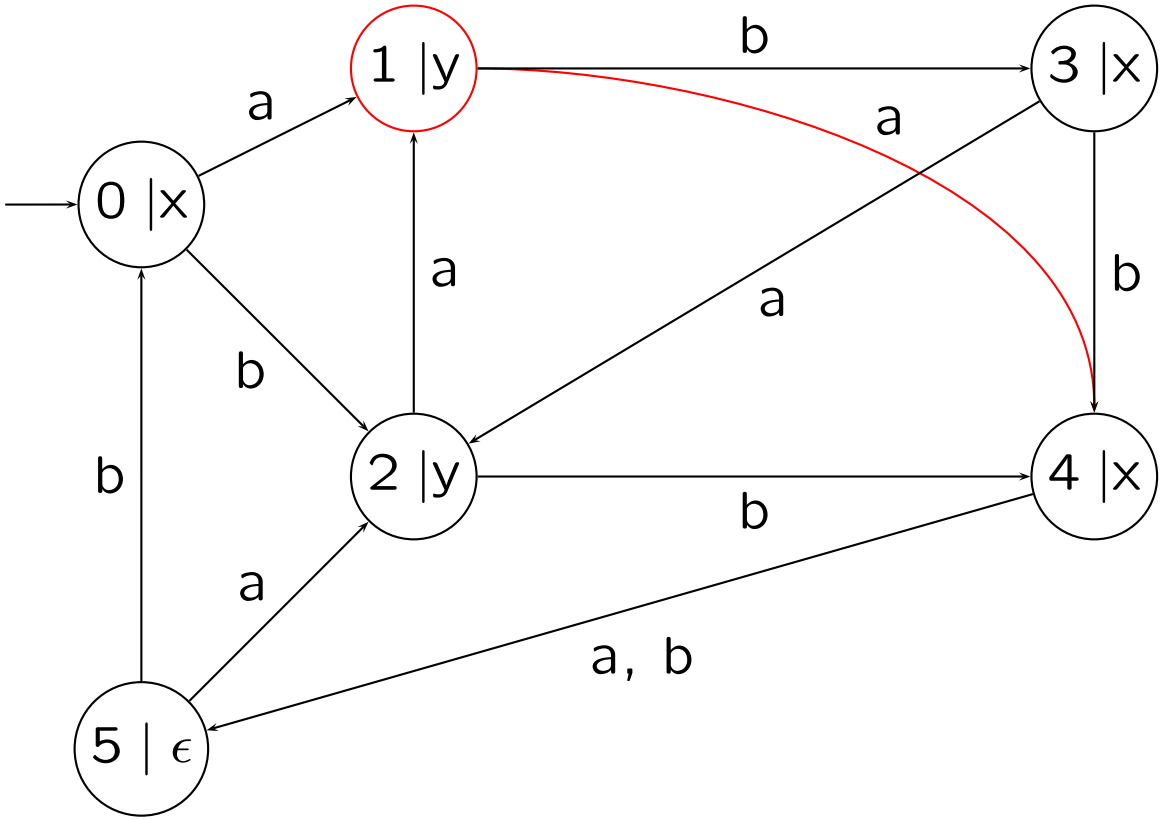
Eingabe: aabab, Ausgabe: x



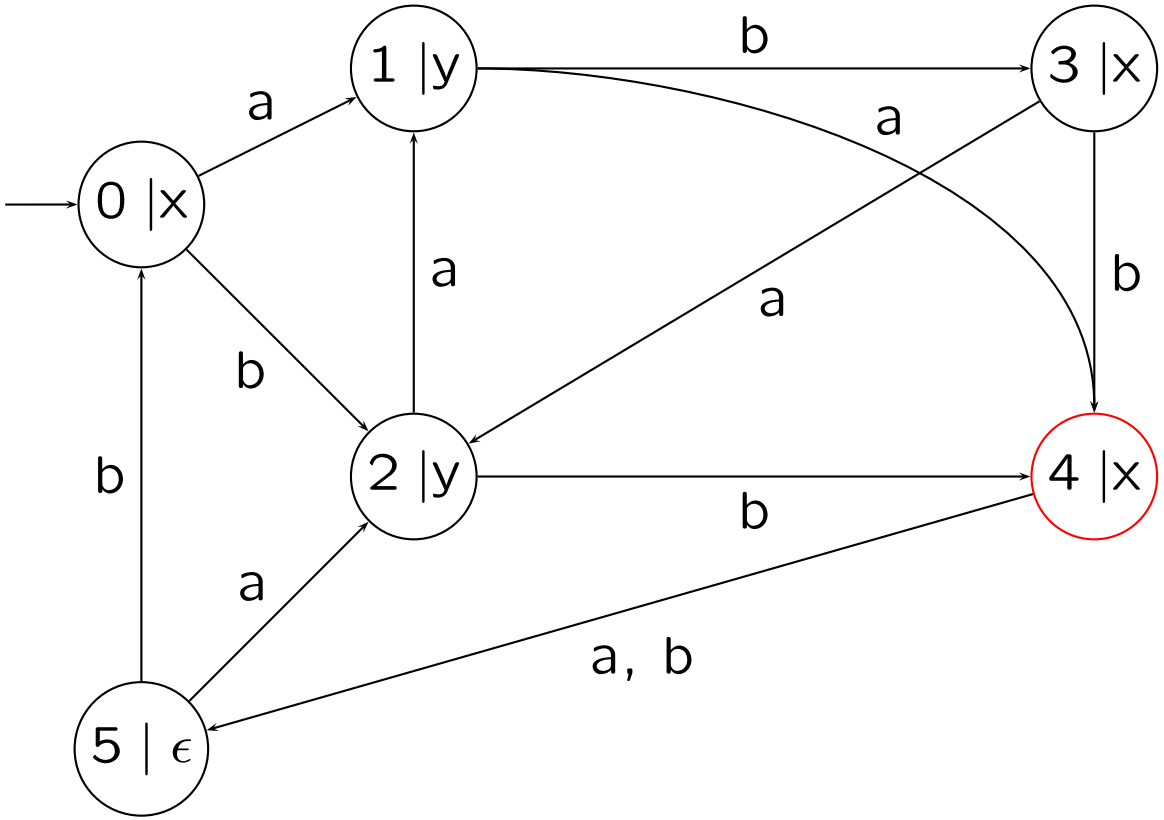
Eingabe: aabab, Ausgabe: xy



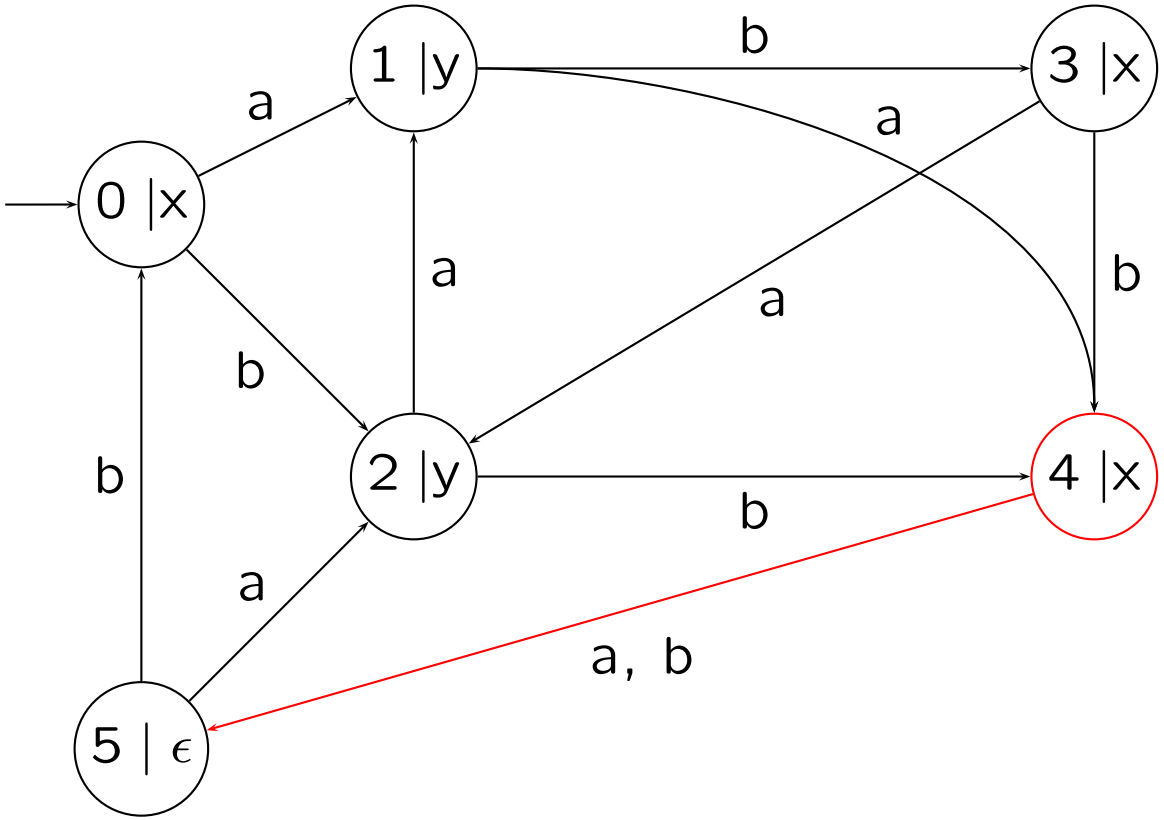
Eingabe: aabab, Ausgabe: xy



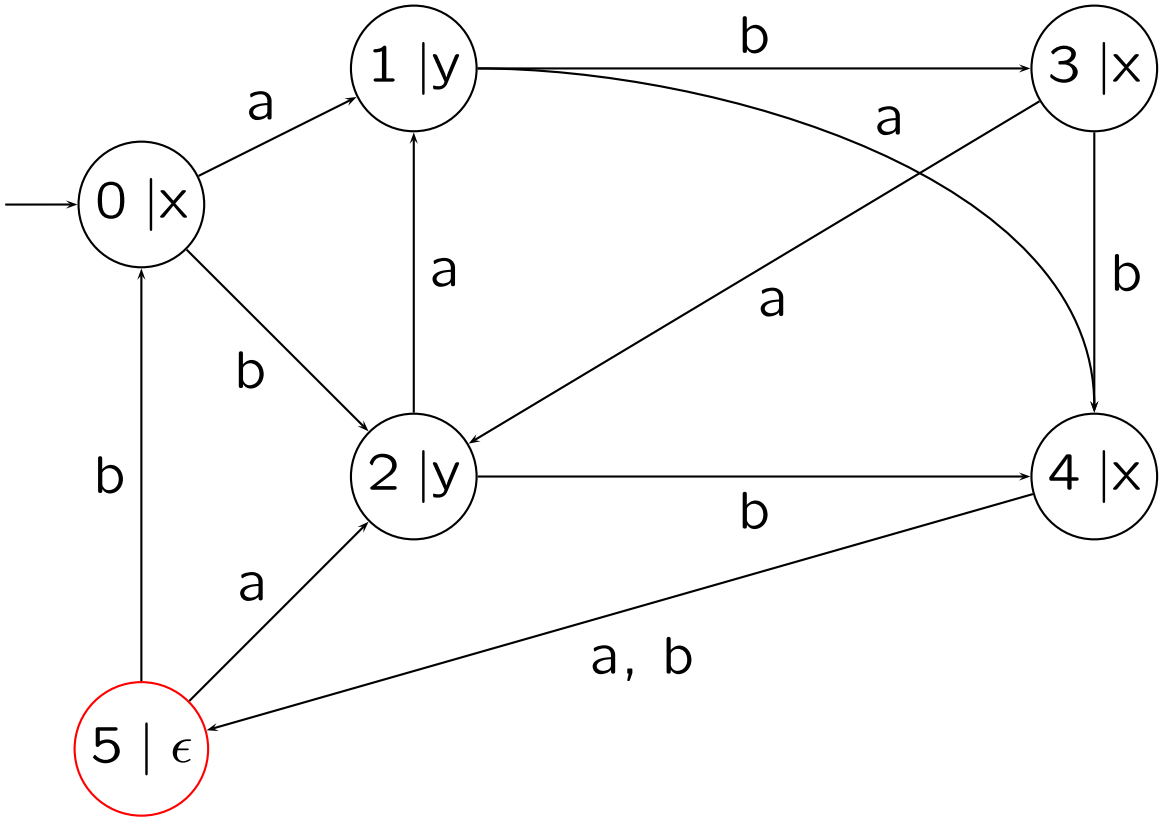
Eingabe: aabab, Ausgabe: xyx



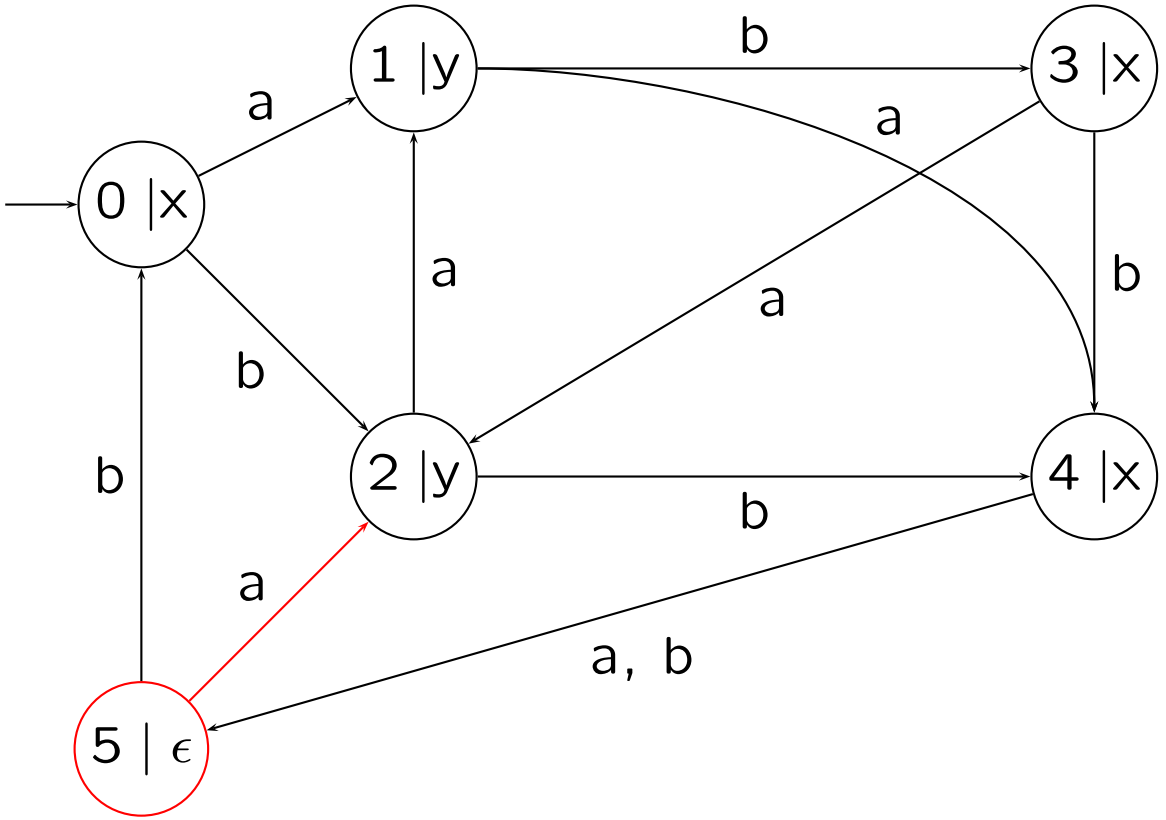
Eingabe: aabab, Ausgabe: xyx



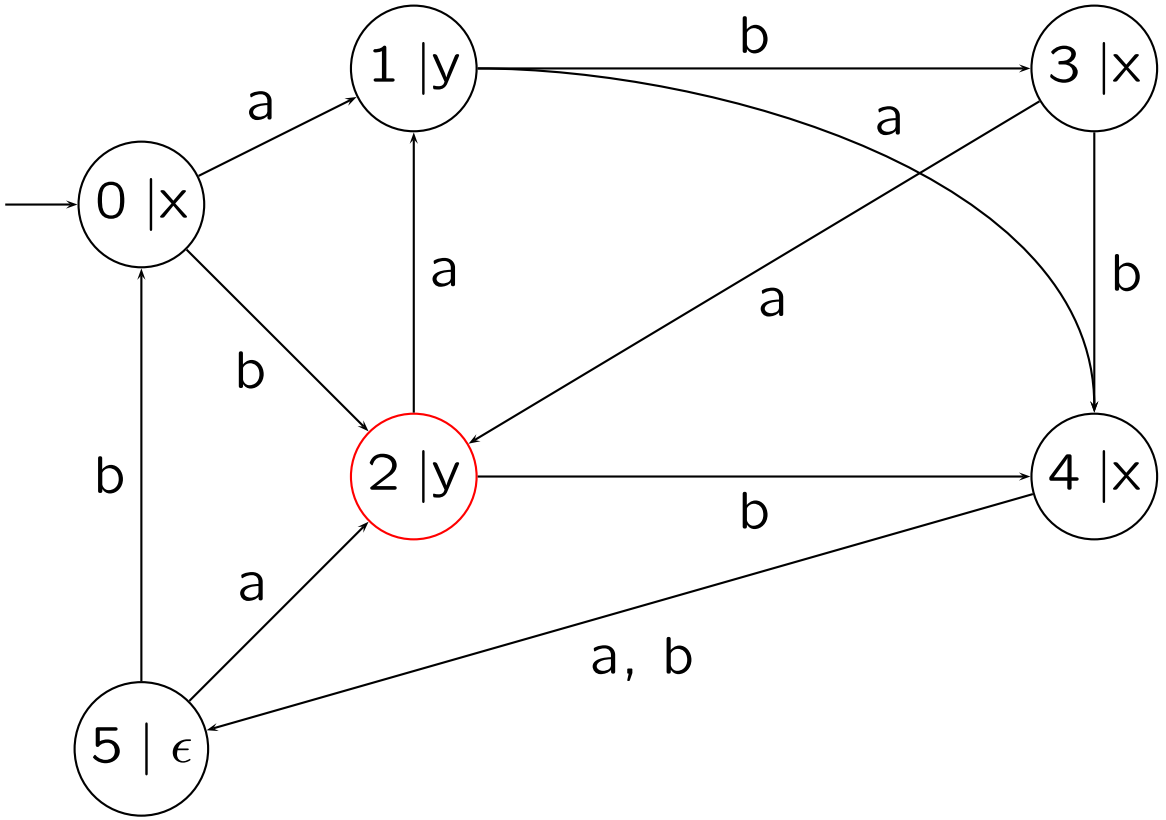
Eingabe: aabab, Ausgabe: xyx



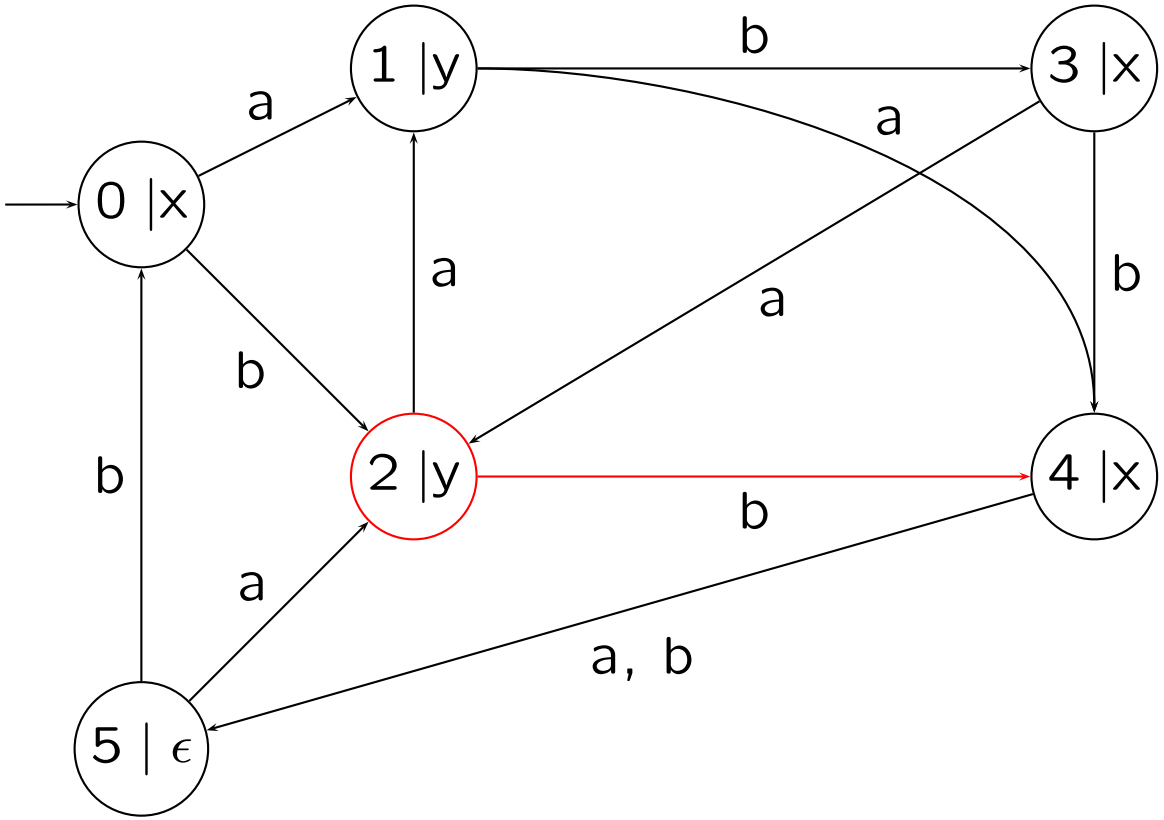
Eingabe: aabab, Ausgabe: xyx



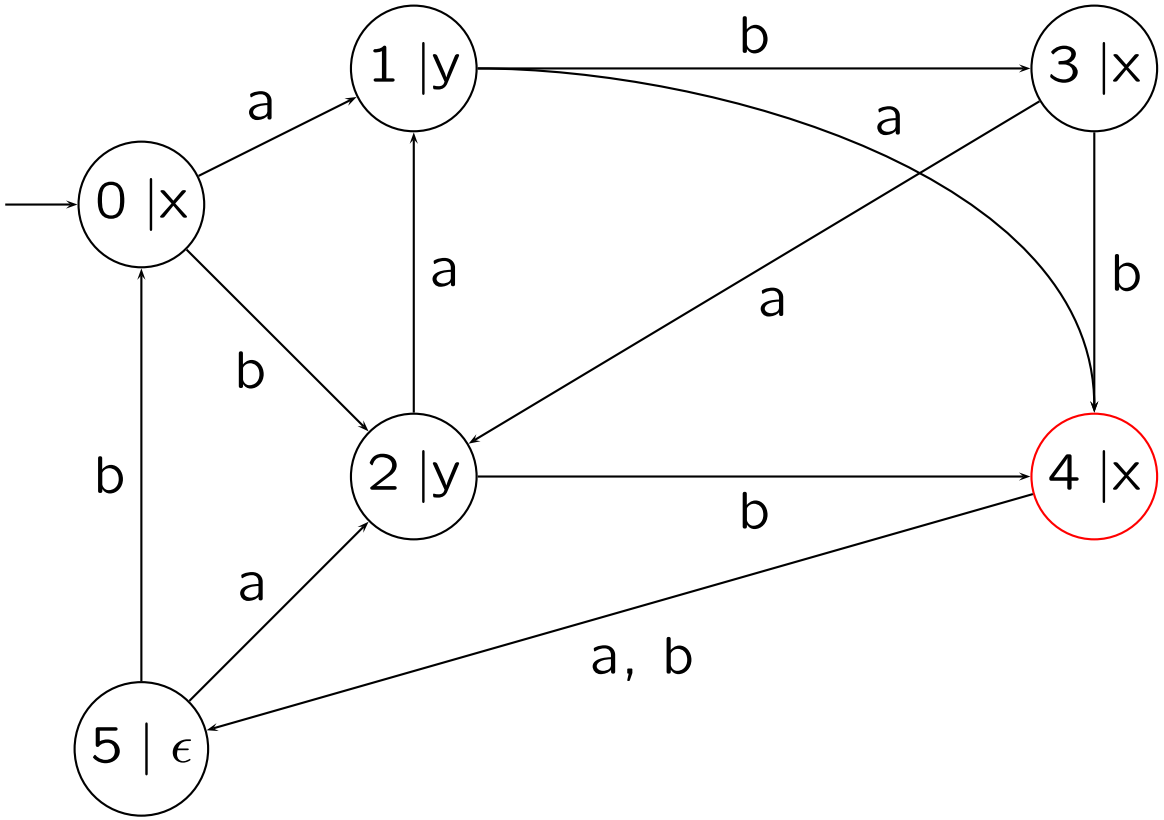
Eingabe: aabab, Ausgabe: xyxy



Eingabe: aabab, Ausgabe: xyxy



Eingabe: aabab, Ausgabe: xyxyx



Wortverarbeitung

abb ist ein böööööses Wort!

.

Wortverarbeitung

abb ist ein böööööses Wort!

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

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

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

.

Wortverarbeitung

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 $cbbaababbc$

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Alles okay
$\rightarrow$ Ausgabe c

.

Wortverarbeitung

abb ist ein böööööses Wort!

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

$cbbaababbc$ Alles okay
$\rightarrow$ Ausgabe cb

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Alles okay
$\rightarrow$ Ausgabe cbb

.

Wortverarbeitung

abb ist ein böööööses Wort!

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

$cbbaababbc$ Könnte kritisch sein
$\rightarrow$ Ausgabe cbb

.

Wortverarbeitung

abb ist ein böööööses Wort!

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

$cbbaababbc$ Könnte kritisch sein, aber a davor okay
$\rightarrow$ Ausgabe $cbba$

.

Wortverarbeitung

abb ist ein böööööses Wort!

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

$cbbaabbabbc$ Alarmstufe Rot
$\rightarrow$ Ausgabe $cbba$

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Könnte kritisch sein, aber ab war okay
$\rightarrow$ Ausgabe $cbbaab$

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Alarmstufe Rot
$\rightarrow$ Ausgabe $cbbaab$

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Böses Wort, ersetzen!
$\rightarrow$ Ausgabe $cbbaabxxx$

.

Wortverarbeitung

abb ist ein bööööses Wort!

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

$cbbaababbc$ Alles Okay
$\rightarrow$ Ausgabe $cbbaabxxxc$

.

Wortverarbeitung

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!

.

Wortverarbeitung

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!

f	OK	KK	AR	BW
a	KK	KK	KK	KK
b	OK	AR	BW	OK
c	OK	OK	OK	OK

.

Wortverarbeitung

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!

g	OK	KK	AR	BW
a	ϵ	a	ab	ϵ
b	b	ϵ	xxx	b
c	c	ac	abc	c

.

Wortverarbeitung

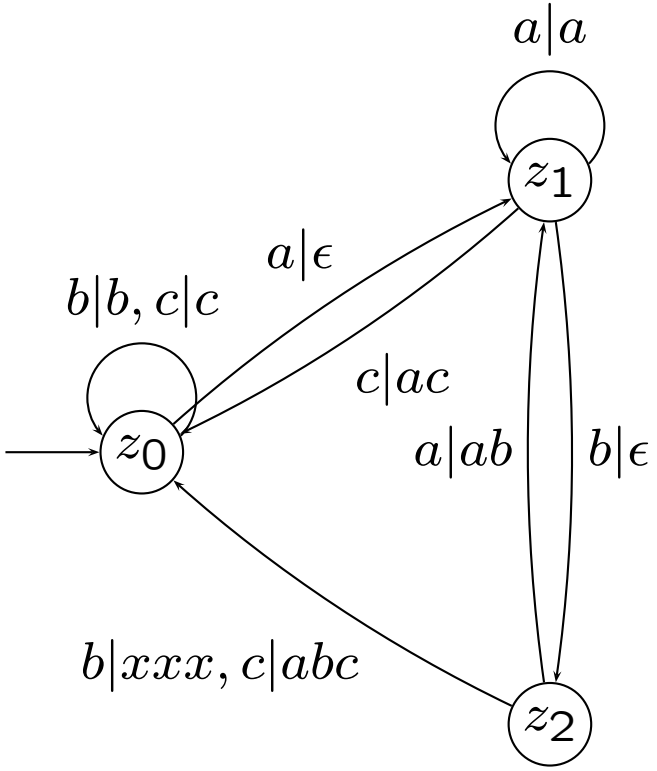
abb ist ein böööööses Wort!

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

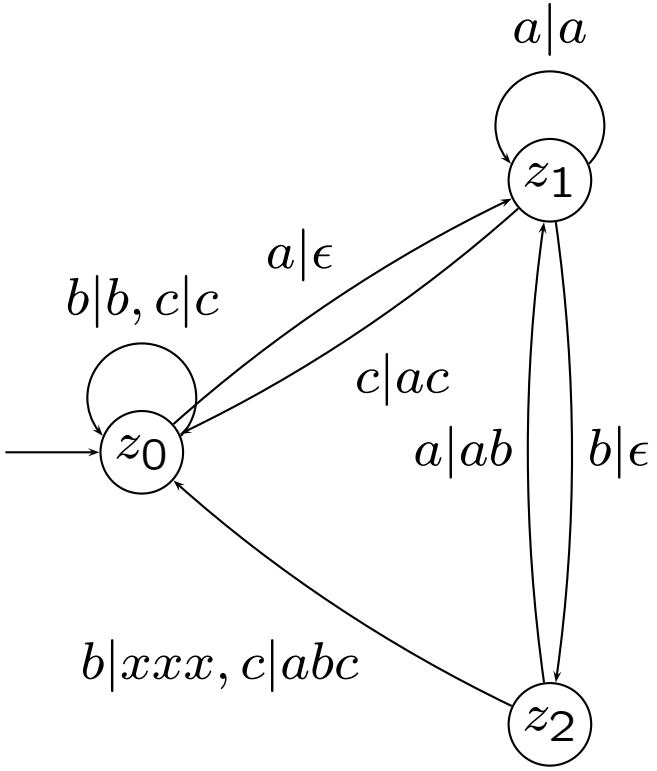
Zustände: Alles Okay, Könnte Kritisch sein, Alarmstufe Rot, Böses Wort!

Feststellung: OK und BW machen das Gleiche!

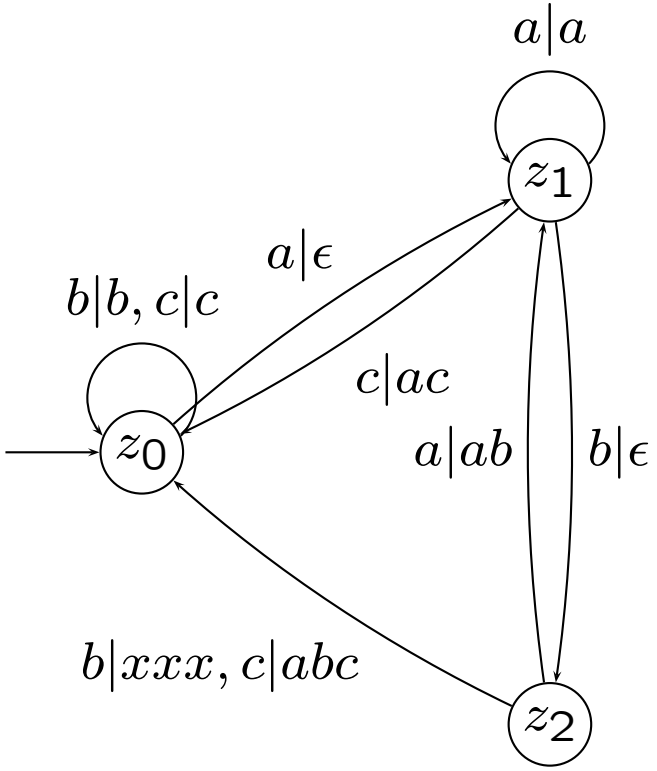
.


$a|a$
z_1
$a|\epsilon$
$b|b, c|c$
$c|ac$
z_0
$a|ab$
$b|\epsilon$
$b|xxx, c|abc$
z_2

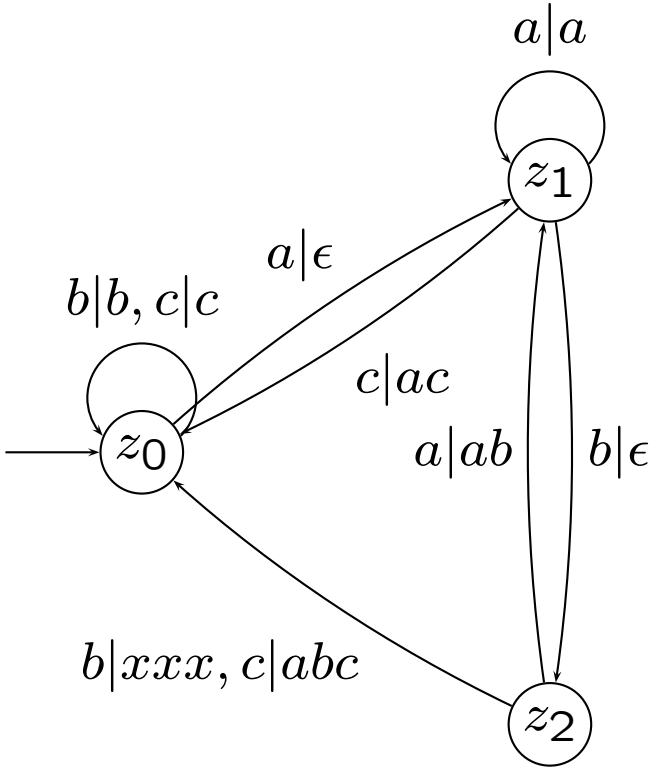
Eingabe $aababcabb$



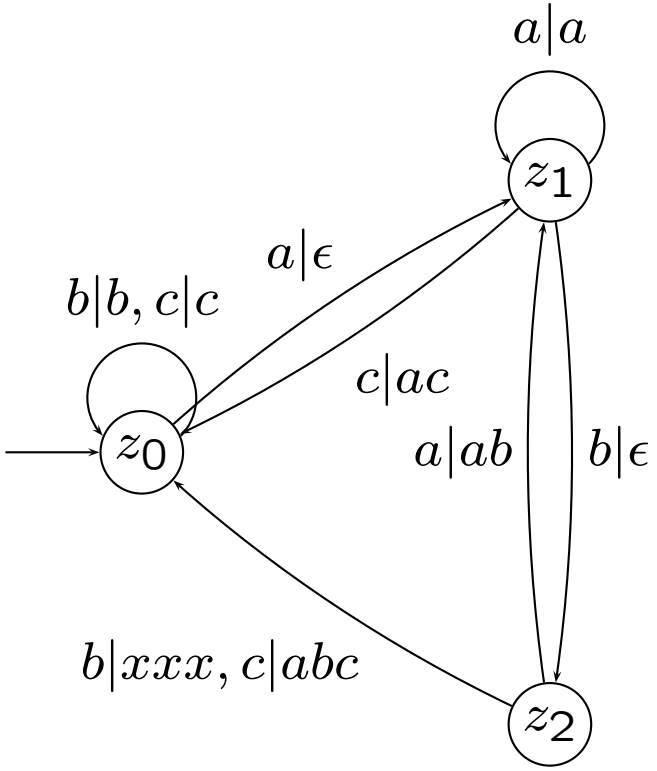
Eingabe $aababcabb$ Ausgabe $aababcxxx$



Eingabe $aababcab$



Eingabe $aababcab$ Ausgabe $aababc$



Eingabe $aababcab$ Ausgabe $aababc$

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

Randfälle beachten!