

# Grundbegriffe der Informatik

## Einheit 14: Endliche Automaten

Thomas Worsch

Karlsruher Institut für Technologie, Fakultät für Informatik

Wintersemester 2010/2011

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

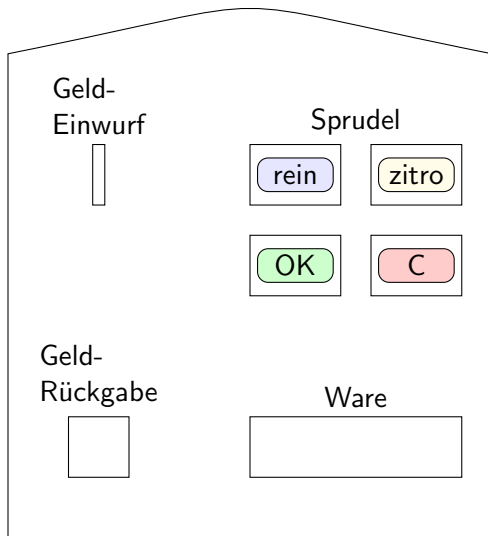
Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

# Ein primitiver Getränkeautomat



- ▶ Geld
  - ▶ nur 1-Euro-Stücke einwerfen
  - ▶ erster Euro wird gespeichert
  - ▶ weitere Geld-Stücke sofort wieder ausgegeben
- ▶ vier Tasten
  - ▶ für Mineralwasser **rein** (1 Euro)  
und Zitronensprudel **zitro** (1 Euro)
  - ▶ **rein**/**zitro**: letzter Getränkewunsch gemerkt
  - ▶ **C** -Taste: ggf. Geld zurück, ggf. Getränkewunsch gelöscht
  - ▶ **OK** -Taste: Geld und Getränkewunsch  $\leadsto$  Getränk

- ▶ Automat muss zwischen den Eingaben, die sein Verhalten beeinflussen können (Geldeinwürfe und Getränkewahl), gewisse Nachrichten speichern.
- ▶ und zwar
  - ▶ Wurde schon ein 1-Euro-Stück eingeworfen?
  - ▶ Wurde schon ein Getränk ausgewählt?
  - ▶ Wenn ja: welches?
- ▶ Modellierung: durch Paare  $(x, y)$ 
  - ▶ Komponente  $x \in \{0, 1\}$ : schon eingeworfener Geldbetrag
  - ▶ Komponente  $y \in \{-, R, Z\}$ : Getränkewahl
  - ▶ Zustandsmenge  $Z = \{0, 1\} \times \{-, R, Z\}$

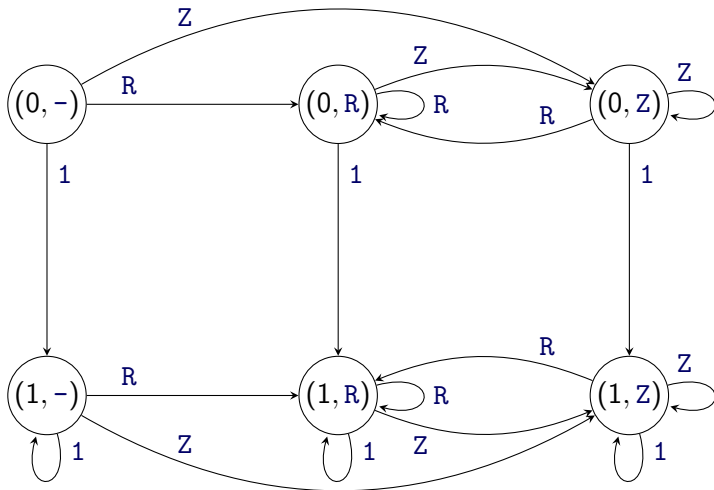
- ▶ erster wesentlicher Aspekt jedes Automaten:  
*Eingaben führen zu Zustandsänderungen.*
- ▶ Eingaben hier:
  - ▶ Einwurf eines Euros
  - ▶ Drücken einer der vier Tasten
- ▶ Modellierung der Eingaben: Symbole 1, R, Z, C und 0
- ▶ Eingabealphabet  $X = \{1, R, Z, C, 0\}$

## ► Zustandsübergang

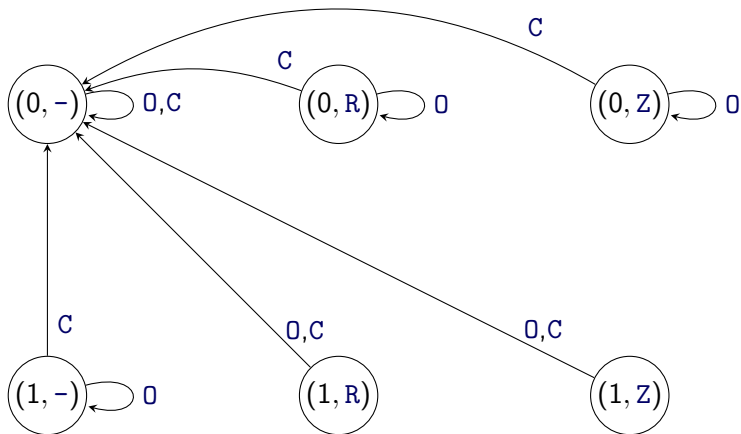
- abhängig von aktuellem Zustand  $z \in Z$  und aktuellem Eingabesymbol  $x \in X$
- $z$  und  $x$  legen eindeutig den neuen Zustand fest.
  - also *immer* und *eindeutig*
  - jedenfalls bei dem Getränkeautomaten
- Formalisierung: **Zustandsüberföhrungsfunktion**  $f : Z \times X \rightarrow Z$
- oft in Darstellung als Graph spezifiziert



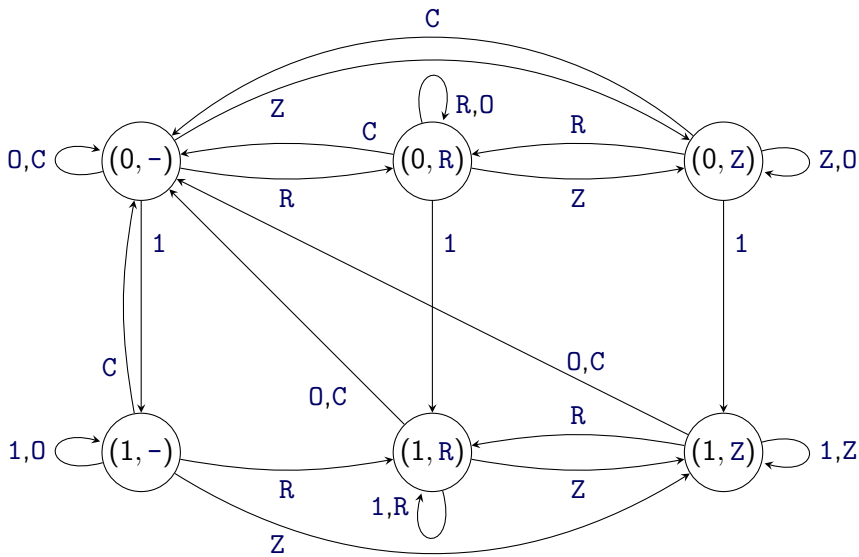
# Formalisierung des Getränkeautomaten: Zustandsübergänge (2a)



## Formalisierung des Getränkeautomaten: Zustandsübergänge (2b)

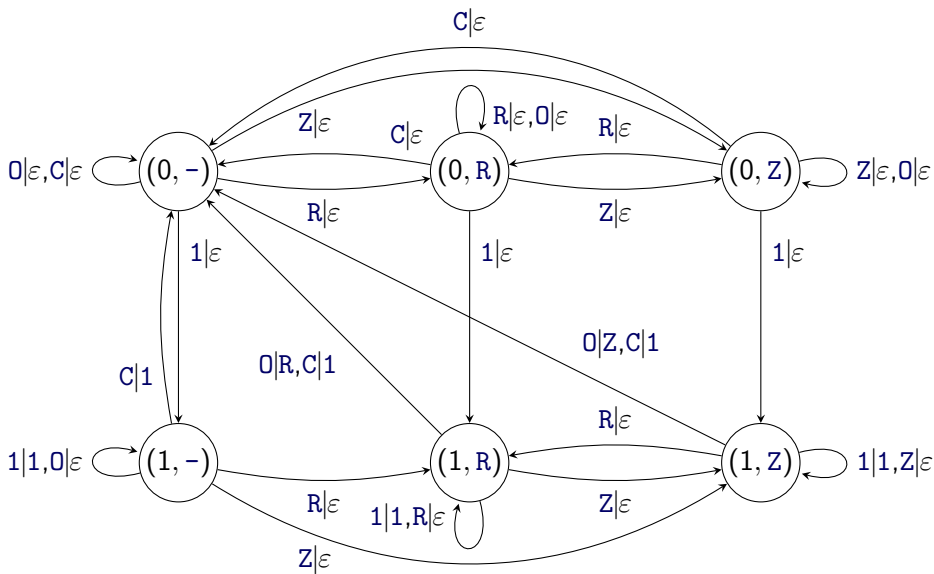


# Formalisierung des Getränkeautomaten: Zustandsübergänge (2c)



- ▶ zweiter wesentlicher Aspekt jedes Automaten: **Ausgaben**
- ▶ hier: Rückgeld, gewählte Flasche
- ▶ **Ausgabealphabet**  $Y = \{1, R, Z\}$
- ▶ Formalisierung der Ausgaben
  - ▶ abhängig von aktuellem Zustand  $z \in Z$  und aktuellem Eingabesymbol  $x \in X$
  - ▶  $z$  und  $x$  legen eindeutig die Ausgabe fest.
    - ▶ also *immer* und *eindeutig*
    - ▶ jedenfalls bei dem Getränkeautomaten
    - ▶ im allgemeinen Wörter über  $Y$
  - ▶ Formalisierung: **Ausgabefunktion**  $g : Z \times X \rightarrow Y^*$ 
    - ▶ Funktionswert  $\varepsilon$ : „keine Ausgabe“
- ▶  $g$  im Zustandsübergangsdiagramm:  $x|g(z, x)$

# Formalisierung des Getränkeautomaten: Ausgaben (2)



## Das sollten Sie mitnehmen:

- ▶ alles endlich
- ▶ alles endlich beschreibbar
- ▶ im Beispiel Getränkeautomat:
  - ▶ aktueller Zustand und aktuelle Eingabe legen immer und eindeutig fest:
    - ▶ nächsten Zustand
    - ▶ Ausgabe

## Das sollten Sie üben:

- ▶ Zustandsdiagramm lesen

Erstes Beispiel: ein Getränkeautomat

**Mealy-Automaten**

Moore-Automaten

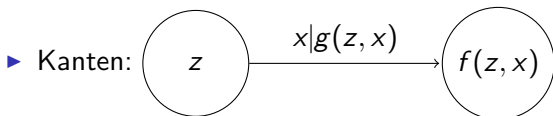
Spezialfall: endliche Akzeptoren

Ein (*endlicher*) *Mealy-Automat* ist festgelegt durch

- ▶ eine endliche *Zustandsmenge*  $Z$ ,
- ▶ einen *Anfangszustand*  $z_0 \in Z$ ,
- ▶ ein *Eingabealphabet*  $X$ ,
- ▶ eine *Zustandsüberföhrungsfunktion*  $f : Z \times X \rightarrow Z$ ,
- ▶ ein *Ausgabealphabet*  $Y$ ,
- ▶ eine *Ausgabefunktion*  $g : Z \times X \rightarrow Y^*$

Darstellung als Graph:

- ▶ Knoten: Zustände



- ▶ Anfangszustand: in der Darstellung durch kleinen Pfeil gekennzeichnet:  $\longrightarrow \textcircled{z}$



# Verallgemeinerte Zustandsübergangsfunktionen (1)

- ▶ nach Eingabe eines ganzen Wortes  $w \in X^*$ :
  - ▶ Was ist der erreichte Zustand?
  - ▶ Welches sind alle durchlaufenen Zustände?
- ▶ definiere passende Funktionen  $f^*$  und  $f^{**}$ 
  1. Stern: zweite Argument ist ganzes Wort von Eingabesymbolen
  2. Stern: interessieren uns für alle durchlaufenen Zuständen Zuständen
- ▶ definiere  $f^* : Z \times X^* \rightarrow Z$ :

$$f^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : f^*(z, wx) = f(f^*(z, w), x)$$

- ▶ alternativ:

$$\bar{f}^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : \bar{f}^*(z, xw) = \bar{f}^*(f(z, x), w)$$

- ▶ beide Definitionen liefern das Gleiche:  $f^* = \bar{f}^*$ .  
Man nimmt die bequemere.

# Verallgemeinerte Zustandsübergangsfunktionen (1)

- ▶ nach Eingabe eines ganzen Wortes  $w \in X^*$ :
  - ▶ Was ist der erreichte Zustand?
  - ▶ Welches sind alle durchlaufenen Zustände?
- ▶ definiere passende Funktionen  $f^*$  und  $f^{**}$ 
  1. Stern: zweite Argument ist ganzes Wort von Eingabesymbolen
  2. Stern: interessieren uns für alle durchlaufenen Zuständen Zuständen
- ▶ definiere  $f^* : Z \times X^* \rightarrow Z$ :

$$f^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : f^*(z, wx) = f(f^*(z, w), x)$$

- ▶ alternativ:

$$\bar{f}^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : \bar{f}^*(z, xw) = \bar{f}^*(f(z, x), w)$$

- ▶ beide Definitionen liefern das Gleiche:  $f^* = \bar{f}^*$ .  
Man nimmt die bequemere.

# Verallgemeinerte Zustandsübergangsfunktionen (1)

- ▶ nach Eingabe eines ganzen Wortes  $w \in X^*$ :
  - ▶ Was ist der erreichte Zustand?
  - ▶ Welches sind alle durchlaufenen Zustände?
- ▶ definiere passende Funktionen  $f^*$  und  $f^{**}$ 
  1. Stern: zweite Argument ist ganzes Wort von Eingabesymbolen
  2. Stern: interessieren uns für alle durchlaufenen Zuständen Zuständen
- ▶ definiere  $f^* : Z \times X^* \rightarrow Z$ :

$$f^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : f^*(z, wx) = f(f^*(z, w), x)$$

- ▶ alternativ:

$$\bar{f}^*(z, \varepsilon) = z$$

$$\forall w \in X^* : \forall x \in X : \bar{f}^*(z, xw) = \bar{f}^*(f(z, x), w)$$

- ▶ beide Definitionen liefern das Gleiche:  $f^* = \bar{f}^*$ .  
Man nimmt die bequemere.

alle durchlaufenen Zustände:

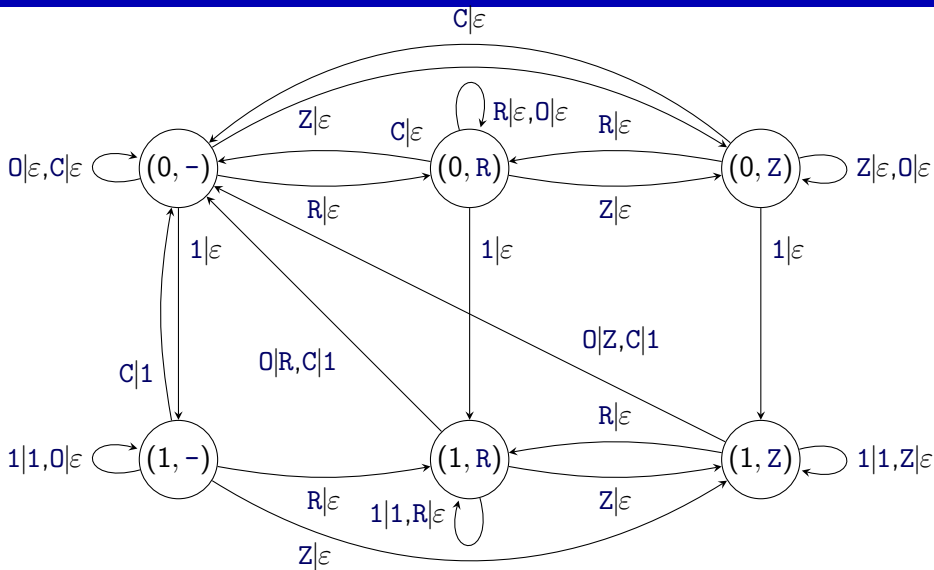
- ▶ definiere  $f^{**} : Z \times X^* \rightarrow Z^*$ :

$$f^{**}(z, \varepsilon) = z$$

$$\forall w \in X^* : x \in X : \quad f^{**}(z, wx) = f^{**}(z, w) \cdot f(f^*(z, w), x)$$

- ▶ auch hier wieder eine alternative Definitionsmöglichkeit

# Bespiel Getränkeautomaten



$$f^{**}((0, -), R1RZO) = (0, -) (0, R) (1, R) (1, R) (1, Z) (0, -)$$

- ▶ für „die letzte“ Ausgabe:  $g^* : Z \times X^* \rightarrow Y^*$

$$g^*(z, \varepsilon) = \varepsilon$$

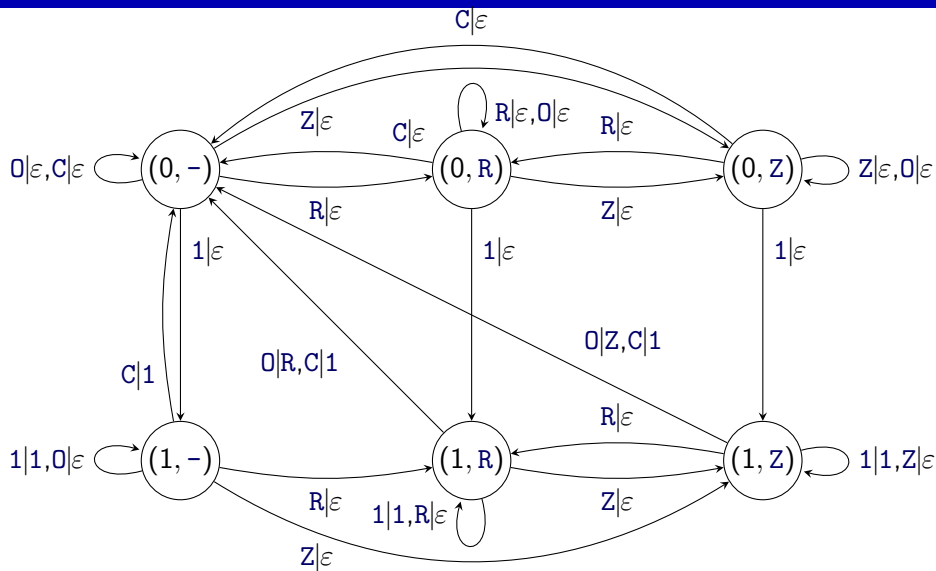
$$g^*(z, wx) = g(f^*(z, w), x)$$

- ▶ Für „alle Ausgaben konkateniert“:  $g^{**} : Z \times X^* \rightarrow Y^*$ :

$$g^{**}(z, \varepsilon) = \varepsilon$$

$$g^{**}(z, wx) = g^{**}(z, w) \cdot g^*(z, wx)$$

# Bespiel Getränkeautomaten



$$g^{**}((0, -), R1RZ0) = \epsilon\epsilon\epsilon\epsilon Z = Z$$

## Das sollten Sie mitnehmen:

- ▶ „offizielle“ Definition von Mealy-Automat:
  - ▶  $Z$
  - ▶  $z_0 \in Z$
  - ▶  $X$
  - ▶  $f : Z \times X \rightarrow Z$
  - ▶  $Y$
  - ▶  $g : Z \times X \rightarrow Y^*$       Ausgabe hängt von der Eingabe ab

## Das sollten Sie üben:

- ▶ zu vorgegebenem „Verhalten“ Beispielautomaten konstruieren
- ▶ von vorgegebenen Automaten ihr Verhalten verstehen



Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

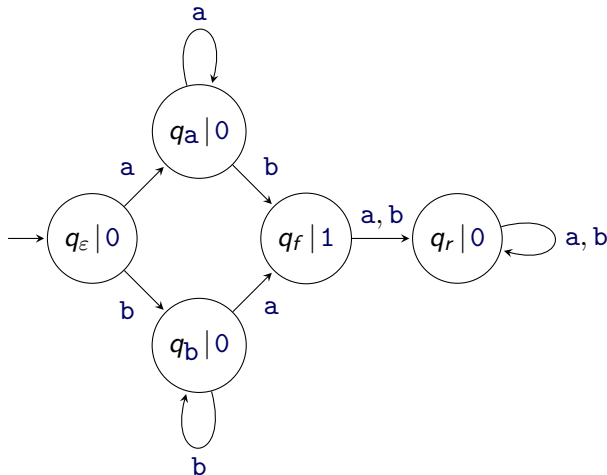
**Moore-Automaten**

Spezialfall: endliche Akzeptoren

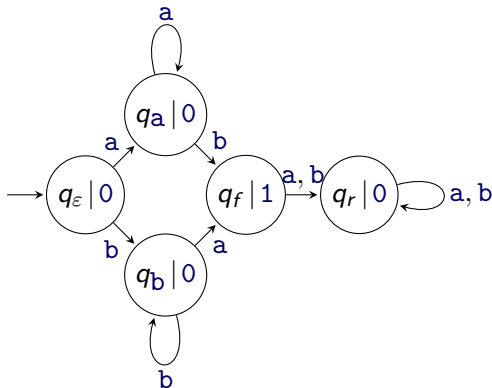
- ▶ manchmal praktischer:  
Ausgabe „im Zustand“ statt beim Zustandsübergang
- ▶ (endlicher) Moore-Automat festgelegt durch
  - ▶ eine endliche *Zustandsmenge*  $Z$ ,
  - ▶ einen *Anfangszustand*  $z_0 \in Z$ ,
  - ▶ ein *Eingabealphabet*  $X$ ,
  - ▶ eine *Zustandsüberföhrungsfunktion*  $f : Z \times X \rightarrow Z$ ,
  - ▶ ein *Ausgabealphabet*  $Y$ ,
  - ▶ eine *Ausgabefunktion*  $h : Z \rightarrow Y^*$

# Moore-Automat: Beispiel (aus der Dokumentation zu tikz)

graphische Darstellung analog zu Mealy-Automaten,  
nur die Ausgaben in den Zuständen:



Definition von  $f^*$  und  $f^{**}$  wie bei Mealy-Automaten



Beispiel:

- ▶  $f^*(q_\epsilon, \text{aaaba}) = q_r$
- ▶  $f^{**}(q_\epsilon, \text{aaaba}) = q_\epsilon q_a q_a q_a q_f q_r$

- ▶ etwas einfacher als bei Mealy-Automaten
- ▶ „letzte Ausgabe“  $g^* = h \circ f^*$ :

$$\forall (z, w) \in Z \times X^* : g^*(z, w) = h(f^*(z, w))$$

- ▶ „alle Ausgaben“:  $g^{**} = h^{**} \circ f^{**}$

$$\forall (z, w) \in Z \times X^* : g^{**}(z, w) = h^{**}(f^{**}(z, w))$$

- ▶ Beispiel:

- ▶  $f^*(q_\varepsilon, \text{aaaba}) = q_r$
- ▶  $f^{**}(q_\varepsilon, \text{aaaba}) = q_\varepsilon q_a q_a q_a q_f q_r$

also

- ▶  $g^*(q_\varepsilon, \text{aaaba}) = h(f^*(q_\varepsilon, \text{aaaba})) = h(q_r) = 0$
- ▶  $g^{**}(q_\varepsilon, \text{aaaba}) = h^{**}(f^{**}(q_\varepsilon, \text{aaaba}))$   
 $= h^{**}(q_\varepsilon q_a q_a q_a q_f q_r)$   
 $= h(q_\varepsilon)h(q_a)h(q_a)h(q_a)h(q_f)h(q_r)$   
 $= 000010$

## Das sollten Sie mitnehmen:

- ▶ „offizielle“ Definition von Moore-Automat:
  - ▶  $Z$
  - ▶  $z_0 \in Z$
  - ▶  $X$
  - ▶  $f : Z \times X \rightarrow Z$
  - ▶  $Y$
  - ▶  $h : Z \rightarrow Y^*$       Ausgabe hängt *nicht* von der Eingabe ab

## Das sollten Sie üben:

- ▶ zu vorgegebenem Verhalten Moore-Automaten konstruieren, der es realisiert
- ▶ von vorgegebenem Moore-Automaten realisiertes Verhalten herausfinden

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

Spezialfall: endliche Akzeptoren

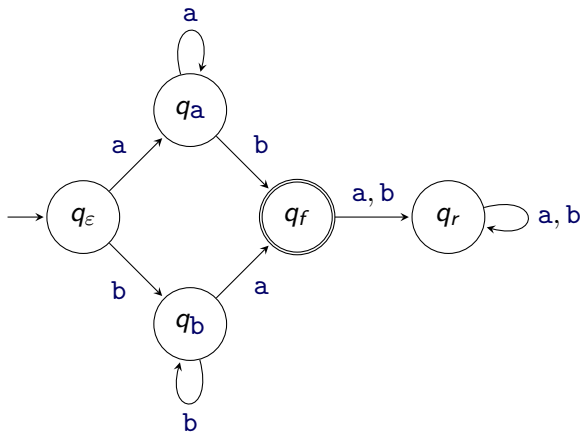
wichtiger Sonderfall von Moore-Automaten:

sogenannte **endliche Akzeptoren**

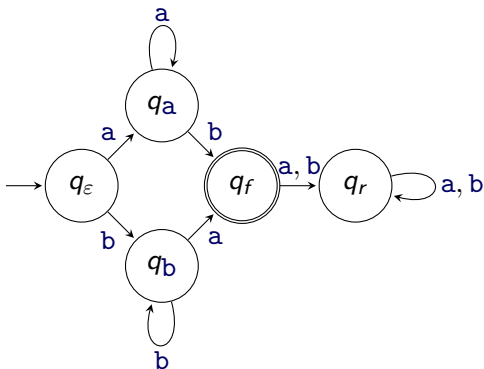
- ▶ Moore-Automaten mit immer genau einem Bit Ausgabe:
  - ▶  $Y = \{0, 1\}$  und
  - ▶  $\forall z : h(z) \in Y$
- ▶ Interpretation der Ausgabe:
  - ▶ Eingabe war „gut“ oder „schlecht“ bzw.
  - ▶ „syntaktisch korrekt“ oder „syntaktisch falsch“  
(für eine gerade interessierende Syntax)
- ▶ bequemere Formalisierung:
  - ▶ Spezifikation der Menge  $F$ : **akzeptierende Zustände**
  - ▶  $F = \{z \mid h(z) = 1\}$
  - ▶ die anderen heißen **ablehnende Zustände**
- ▶ in graphischen Darstellungen  
akzeptierende Zustände mit doppeltem Kringel gemalt: 



der Moore-Automat aus dem vorangegangenen Abschnitt:



- ▶ Wort  $w \in X^*$  wird *akzeptiert*, falls  $f^*(z_0, w) \in F$ .
- ▶ Wort  $w \in X^*$  wird *abgelehnt*, falls  $f^*(z_0, w) \notin F$ .



- ▶ **aaaba** wird abgelehnt, denn  $f^*(z_0, \mathbf{aaaba}) = q_r \notin F$
- ▶ **aaab** wird akzeptiert, denn  $f^*(z_0, \mathbf{aaab}) = q_f \in F$ .
- ▶ allgemein:
  - ▶ Alle Wörter der Form  $\mathbf{a^k b}$  für ein  $k \in \mathbb{N}_+$  werden akzeptiert.
  - ▶ Alle Wörter der Form  $\mathbf{b^k a}$  für ein  $k \in \mathbb{N}_+$  werden akzeptiert.
  - ▶ Keine anderen Wörter werden akzeptiert.

- Die von einem Akzeptor  $A = (Z, z_0, X, f, F)$  *akzeptierte* oder *erkannte formale Sprache* ist

$$L(A) = \{w \in X^* \mid f^*(z_0, w) \in F\}$$

- Das ist ganz einfache „Syntaxanalyse“.
- in unserem Beispiel:

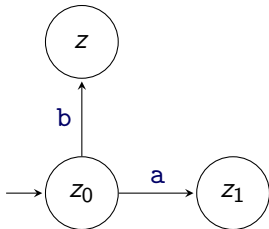
$$L(A) = \{a\}^+ \{b\} \cup \{b\}^+ \{a\}$$

## Beispiel 2 einer erkennbaren Sprache

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.

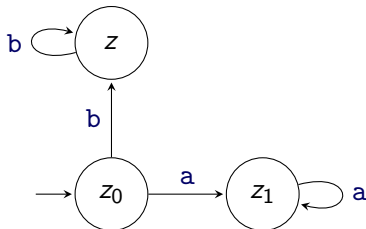
## Beispiel 2 einer erkennbaren Sprache (1)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: erster Ansatz



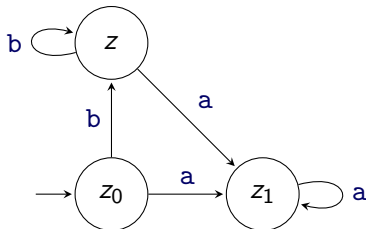
## Beispiel 2 einer erkennbaren Sprache (2)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt



## Beispiel 2 einer erkennbaren Sprache (3)

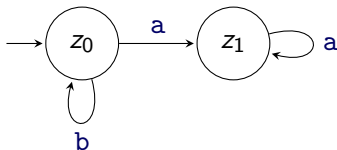
- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt





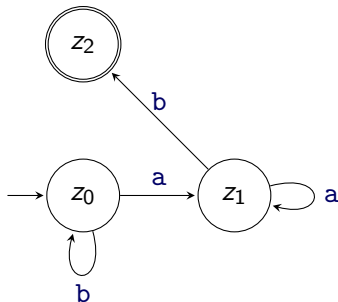
## Beispiel 2 einer erkennbaren Sprache (4)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt



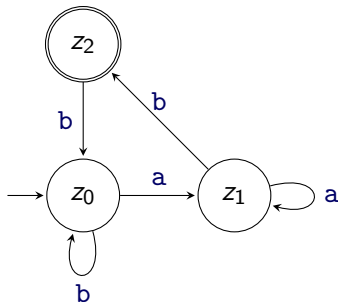
## Beispiel 2 einer erkennbaren Sprache (5)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt



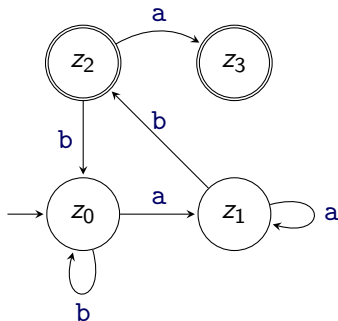
## Beispiel 2 einer erkennbaren Sprache (6)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt



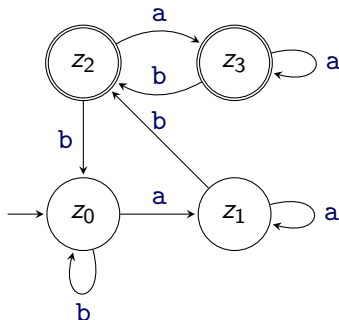
## Beispiel 2 einer erkennbaren Sprache (7)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: nächster Schritt



## Beispiel 2 einer erkennbaren Sprache (8)

- ▶ formale Sprache  $L$  aller Wörter  $w \in \{a, b\}^*$  mit den Eigenschaften:
  - ▶ in  $w$  kommt mindestens ein  $b$  vor und
  - ▶ vor dem letzten  $b$  steht ein  $a$
- ▶ **Behauptung:** Es gibt einen endlichen Akzeptor, der  $L$  erkennt.
- ▶ Konstruktion: letzter Schritt



## Beispiel 3 einer erkennbaren Sprache

- ▶ Aufgabe aus dem Informatiker-Alltag
  - ▶ gegeben: Textdatei mit vielen Zeilen
  - ▶ gesucht: die Zeilen, in denen ein gewisses Wort  $m$  vorkommt
- ▶ mit anderen Worten
  - ▶ gegeben: Zeichenkette  $w$  und Textmuster  $m$
  - ▶ gesucht: Algorithmus, der feststellt, ob  $m$  in  $w$  vorkommt
  - ▶ das geht mit einem endlichen Akzeptor
- ▶ Beispiel:
  - ▶ Eingabealphabet  $X = \{a, b\}$
  - ▶ Textmuster  $m = ababb$
  - ▶ Ziel: endlicher Akzeptor  $A$  mit
$$L(A) = \{w_1 ababb w_2 \mid w_1, w_2 \in \{a, b\}^*\}$$

## Beispiel 3 einer erkennbaren Sprache

- ▶ Aufgabe aus dem Informatiker-Alltag
  - ▶ gegeben: Textdatei mit vielen Zeilen
  - ▶ gesucht: die Zeilen, in denen ein gewisses Wort  $m$  vorkommt
- ▶ mit anderen Worten
  - ▶ gegeben: Zeichenkette  $w$  und Textmuster  $m$
  - ▶ gesucht: Algorithmus, der feststellt, ob  $m$  in  $w$  vorkommt
  - ▶ das geht mit einem endlichen Akzeptor
- ▶ Beispiel:
  - ▶ Eingabealphabet  $X = \{a, b\}$
  - ▶ Textmuster  $m = ababb$
  - ▶ Ziel: endlicher Akzeptor  $A$  mit
$$L(A) = \{w_1 ababb w_2 \mid w_1, w_2 \in \{a, b\}^*\}$$

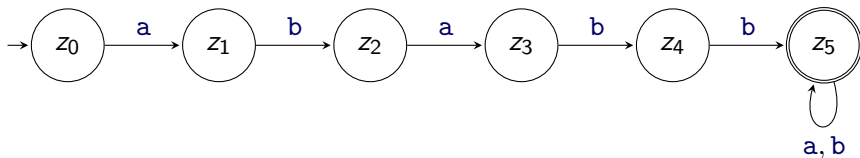
## Beispiel 3 einer erkennbaren Sprache

- ▶ Aufgabe aus dem Informatiker-Alltag
  - ▶ gegeben: Textdatei mit vielen Zeilen
  - ▶ gesucht: die Zeilen, in denen ein gewisses Wort  $m$  vorkommt
- ▶ mit anderen Worten
  - ▶ gegeben: Zeichenkette  $w$  und Textmuster  $m$
  - ▶ gesucht: Algorithmus, der feststellt, ob  $m$  in  $w$  vorkommt
  - ▶ das geht mit einem endlichen Akzeptor
- ▶ Beispiel:
  - ▶ Eingabealphabet  $X = \{a, b\}$
  - ▶ Textmuster  $m = ababb$
  - ▶ Ziel: endlicher Akzeptor  $A$  mit
$$L(A) = \{w_1 ababb w_2 \mid w_1, w_2 \in \{a, b\}^*\}$$



## Beispiel 3 einer erkennbaren Sprache (2)

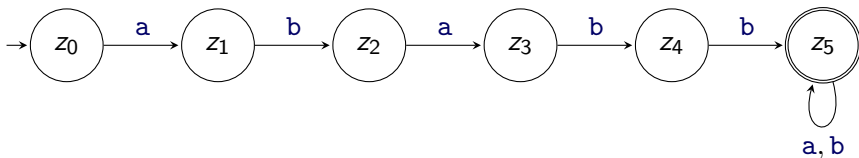
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (2)

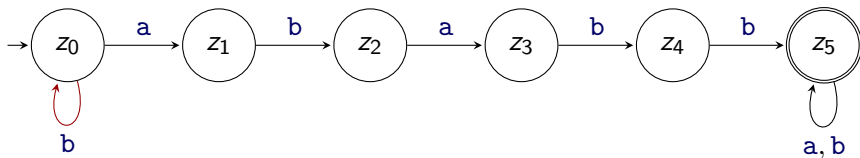
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (3)

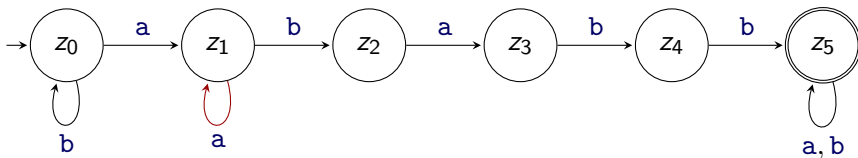
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ **bbbababb**
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (4)

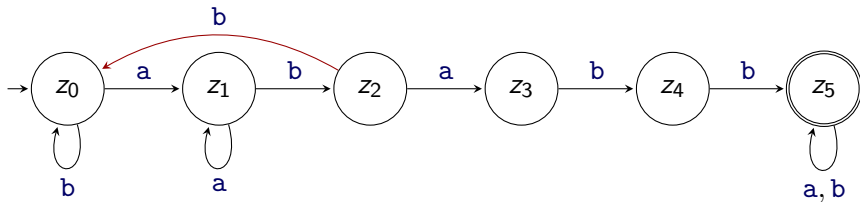
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (5)

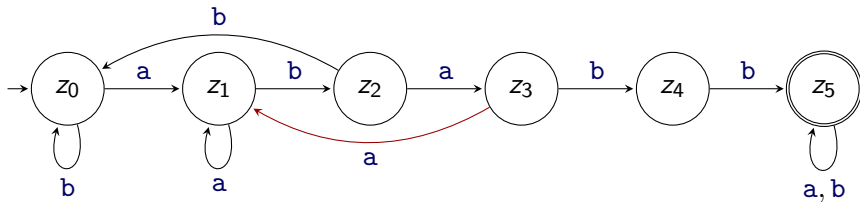
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ **abbababb**
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (6)

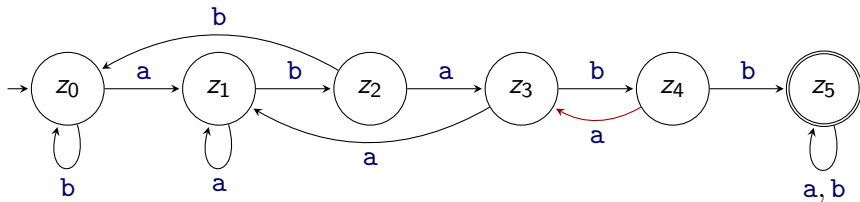
ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

## Beispiel 3 einer erkennbaren Sprache (7)

ein erster Ansatz:



- ▶ es fehlen noch diverse Übergänge
- ▶ Was ist z. B. mit folgenden Wörtern?
  - ▶ bbbababb
  - ▶ aaaababb
  - ▶ abbababb
  - ▶ abaababb
  - ▶ abababb

- **Behauptung:** Die formale Sprache

$$L = \{a^k b^k \mid k \in \mathbb{N}_0\}$$

kann von keinem endlichen Akzeptor erkannt werden.

- **Beweis**

- „schwieriger“ als einen Akzeptor hinzumalen:

- betrachte „beliebigen“ endlichen Akzeptor  $A$  und
- zeige:  $L(A) \neq L$

- Was bedeutet  $L(A) \neq L$ ?

- $L \not\subseteq L(A)$       oder
- $L(A) \not\subseteq L$

- folgenden Vorgehensweise ist „zielführend“:

Fallunterscheidung:

- 1. Fall:  $L \not\subseteq L(A)$ : dann offensichtlich  $L(A) \neq L$
- 2. Fall:  $L \subseteq L(A)$ : zeige: dann aber  $L(A) \not\subseteq L$ ,  
d. h. ein „falsches“ Wort wird akzeptiert



- **Behauptung:** Die formale Sprache

$$L = \{a^k b^k \mid k \in \mathbb{N}_0\}$$

kann von keinem endlichen Akzeptor erkannt werden.

- **Beweis**

- „schwieriger“ als einen Akzeptor hinzumalen:
  - betrachte „beliebigen“ endlichen Akzeptor  $A$  und
  - zeige:  $L(A) \neq L$
- Was bedeutet  $L(A) \neq L$ ?
  - $L \not\subseteq L(A)$       oder
  - $L(A) \not\subseteq L$
- folgenden Vorgehensweise ist „zielführend“:  
Fallunterscheidung:
  - 1. Fall:  $L \not\subseteq L(A)$ : dann offensichtlich  $L(A) \neq L$
  - 2. Fall:  $L \subseteq L(A)$ : zeige: dann aber  $L(A) \not\subseteq L$ ,  
d. h. ein „falsches“ Wort wird akzeptiert

## Beispiel einer *nicht* erkennbaren Sprache (2)

- ▶  $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$
- ▶ „nur“ zu zeigen: wenn  $L \subseteq L(A)$ , dann  $L(A) \not\subseteq L$
- ▶ sei  $m = |Z|$
- ▶ betrachte die Eingabe  $w = a^m b^m$
- ▶  $f^{**}(z_0, a^m)$  besteht aus  $m + 1$  Zuständen:
  - ▶  $z_0$
  - ▶  $z_1 = f(z_0, a)$
  - ▶  $z_2 = f(z_1, a)$
  - ▶  $\vdots$
  - ▶  $z_m = f(z_{m-1}, a)$
- ▶ ein Zustand muss doppelt vorkommen:  
A läuft in einer Schleife
- ▶ seien  $i \geq 0$  und  $\ell \geq 1$  die kleinsten Zahlen mit  
 $z_i = z_{i+\ell}$ , also  $f^*(z_0, a^i) = f^*(z_0, a^{i+\ell})$

## Beispiel einer *nicht* erkennbaren Sprache (2)

- ▶  $L = \{a^k b^k \mid k \in \mathbb{N}_0\}$
- ▶ „nur“ zu zeigen: wenn  $L \subseteq L(A)$ , dann  $L(A) \not\subseteq L$
- ▶ sei  $m = |Z|$
- ▶ betrachte die Eingabe  $w = a^m b^m$
- ▶  $f^{**}(z_0, a^m)$  besteht aus  $m + 1$  Zuständen:
  - ▶  $z_0$
  - ▶  $z_1 = f(z_0, a)$
  - ▶  $z_2 = f(z_1, a)$
  - ▶  $\vdots$
  - ▶  $z_m = f(z_{m-1}, a)$
- ▶ ein Zustand muss doppelt vorkommen:  
A läuft in einer **Schleife**
- ▶ seien  $i \geq 0$  und  $\ell \geq 1$  die kleinsten Zahlen mit  
 $z_i = z_{i+\ell}$ , also  $f^*(z_0, a^i) = f^*(z_0, a^{i+\ell})$

## Beispiel einer *nicht* erkennbaren Sprache (3)

- ▶ seien  $i \geq 0$  und  $\ell \geq 1$  die kleinsten Zahlen mit  $z_i = z_{i+\ell}$ , also  $f^*(z_0, \mathbf{a}^i) = f^*(z_0, \mathbf{a}^{i+\ell})$
- ▶ dann auch
  - ▶  $z_{i+1} = z_{i+\ell+1}$
  - ▶  $z_{i+2} = z_{i+\ell+2}$
  - ▶  $\vdots$
  - ▶  $z_{m-\ell} = z_m$
- ▶ also  $f^*(z_0, \mathbf{a}^{m-\ell}) = f^*(z_0, \mathbf{a}^m)$
- ▶ A „unterscheidet nicht“, ob  $m$  oder  $m - \ell$   $\mathbf{a}$  in der Eingabe
- ▶ betrachte:  $w' = \mathbf{a}^{m-\ell}\mathbf{b}^m \notin L$
- ▶ 
$$\begin{aligned} f^*(z_0, w') &= f^*(z_0, \mathbf{a}^{m-\ell}\mathbf{b}^m) = f^*(f^*(z_0, \mathbf{a}^{m-\ell}), \mathbf{b}^m) \\ &= f^*(f^*(z_0, \mathbf{a}^m), \mathbf{b}^m) = f^*(z_0, \mathbf{a}^m\mathbf{b}^m) \in F \end{aligned}$$
- ▶  $w' \in L(A)$  aber  $w' \notin L$

## Beispiel einer *nicht* erkennbaren Sprache (3)

- ▶ seien  $i \geq 0$  und  $\ell \geq 1$  die kleinsten Zahlen mit  $z_i = z_{i+\ell}$ , also  $f^*(z_0, \mathbf{a}^i) = f^*(z_0, \mathbf{a}^{i+\ell})$
- ▶ dann auch
  - ▶  $z_{i+1} = z_{i+\ell+1}$
  - ▶  $z_{i+2} = z_{i+\ell+2}$
  - ▶  $\vdots$
  - ▶  $z_{m-\ell} = z_m$
- ▶ also  $f^*(z_0, \mathbf{a}^{m-\ell}) = f^*(z_0, \mathbf{a}^m)$
- ▶ A „unterscheidet nicht“, ob  $m$  oder  $m - \ell$   $\mathbf{a}$  in der Eingabe
- ▶ betrachte:  $w' = \mathbf{a}^{m-\ell}\mathbf{b}^m \notin L$
- ▶  $f^*(z_0, w') = f^*(z_0, \mathbf{a}^{m-\ell}\mathbf{b}^m) = f^*(f^*(z_0, \mathbf{a}^{m-\ell}), \mathbf{b}^m)$   
 $= f^*(f^*(z_0, \mathbf{a}^m), \mathbf{b}^m) = f^*(z_0, \mathbf{a}^m\mathbf{b}^m) \in F$
- ▶  $w' \in L(A)$  aber  $w' \notin L$

## Das sollten Sie mitnehmen:

- ▶ „offizielle“ Definition endlicher Akzeptoren:
  - ▶  $Z$
  - ▶  $z_0 \in Z$
  - ▶  $X$
  - ▶  $f : Z \times X \rightarrow Z$
  - ▶  $F \subseteq Z$
- ▶ Wenn ein „sehr langes“ Wort akzeptiert wird,
  - ▶ dann läuft der Automat in einer Schleife,
  - ▶ die beliebig oft durchlaufen werden kann ohne Akzeptanz zu ändern

## Das sollten Sie üben:

- ▶ gegeben  $L$ : konstruiere  $A$  mit  $L(A) = L$
- ▶ gegeben  $A$ : bestimme  $L(A)$

- ▶ Mealy-Automaten
- ▶ Moore-Automaten
  - ▶ taucht im Zusammenhang mit diversen *Protokollen* z. B. in Betriebssystemen und bei Kommunikationssystemen auf
- ▶ insbesondere Akzeptoren
  - ▶ primitive Syntaxanalyse
  - ▶ aber oft nützlich, z. B. bei Compilerbau-Werkzeugen, Suche nach Text-Vorkommen, etc.