Grundbegriffe der Informatik

Einheit 14: Endliche Automaten

Thomas Worsch

Universität Karlsruhe, Fakultät für Informatik

Wintersemester 2008/2009

Überblick

Überblick

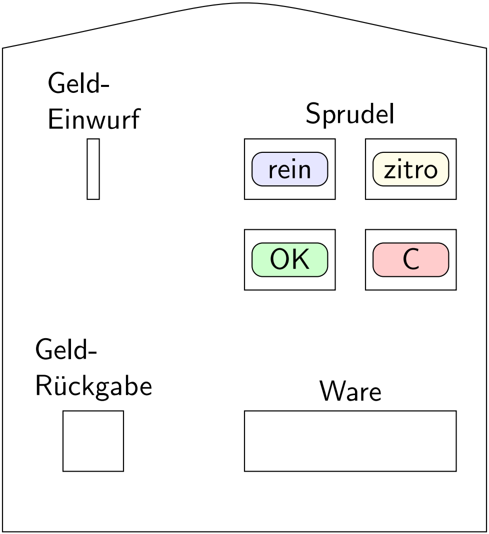
Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

- Allgemeines Modell
- Spezialfall: Akzeptoren

Ein primitiver Getränkeautomat



Getränkeautomat: umgangssprachliche Beschreibung

- nur 1-Euro-Stücke einzuwerfen
- vier Tasten
  - zwei Wahltasten für Mineralwasser rein und Zitronensprudel zitro
  - OK-Taste und Abbruch-Taste C
- Jede Flasche kostet 1 Euro.
- Guthaben von 1 Euro kann gespeichert werden
- weitere Euro-Stücke werden sofort wieder ausgegeben
- Drücken von rein/zitro: letzter wird Wunsch gespeichert
- C -Taste: bereits eingeworfener Euro wird zurückgegeben und kein Getränkewunsch mehr gespeichert.
- OK -Taste
  - ignoriert, solange noch kein Euro eingeworfen oder keine Getränkesorte ausgewählt
  - andernfalls das gewünschte Getränk ausgeworfen

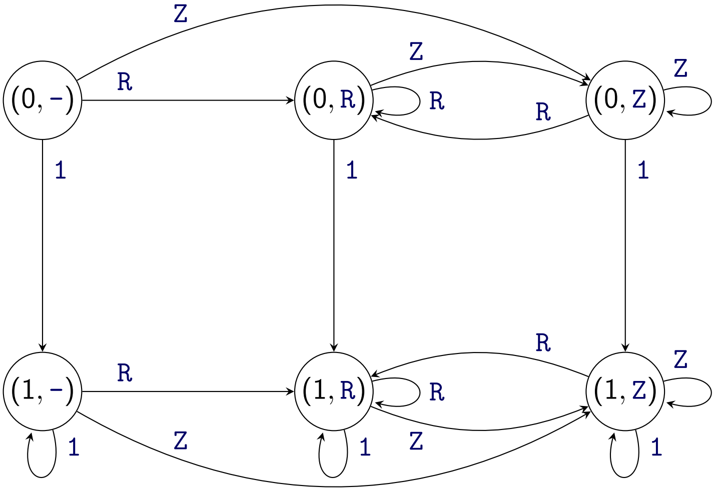
Formalisierung des Getränkeautomaten: Zustände

- 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 $y \in \{\texttt{-}, \texttt{R}, \texttt{Z}\}$: Getränkewahl
  - Zustandsmenge $Z = \{0, 1\} \times \{\texttt{-}, \texttt{R}, \texttt{Z}\}$

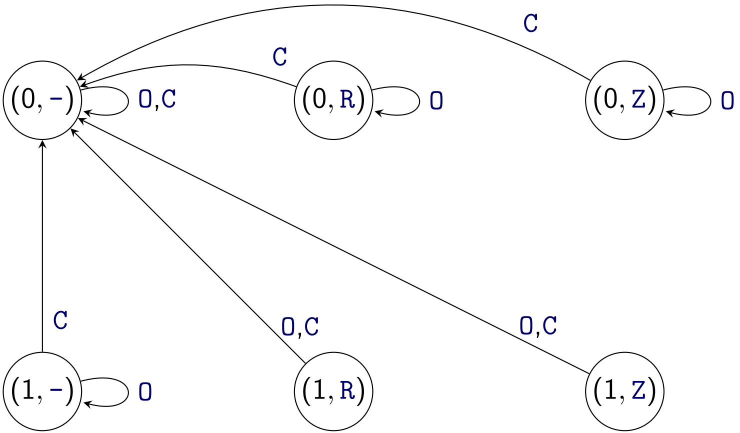
Formalisierung des Getränkeautomaten: Eingaben

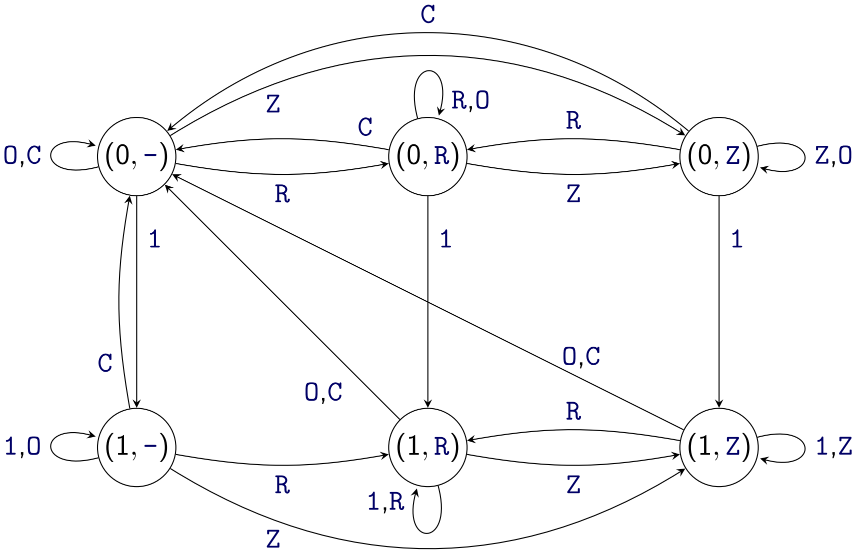
- erster wesentlicher Aspekt jedes Automaten: Eingaben (Einflüsse „von außen“) führen zu *Zustandsänderungen*.
- Eingaben hier:
  - Einwurf eines Euros
  - Drücken einer der vier Tasten
    - ignoriere gleichzeitige Tastendrücke („Abstraktion“)
- Modellierung der Eingaben: Symbole `1`, `R`, `Z`, `C` und `O`
- Eingabealphabet $X = \{\mathtt{1}, \mathtt{R}, \mathtt{Z}, \mathtt{C}, \mathtt{O}\}$

- 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: Zustandsüberführungsfunktion $f : Z \times X \to Z$
  - oft in Darstellung als Graph spezifiziert


(0, -)
(0, R)
(0, Z)
(1, -)
(1, R)
(1, Z)
Z
R
1

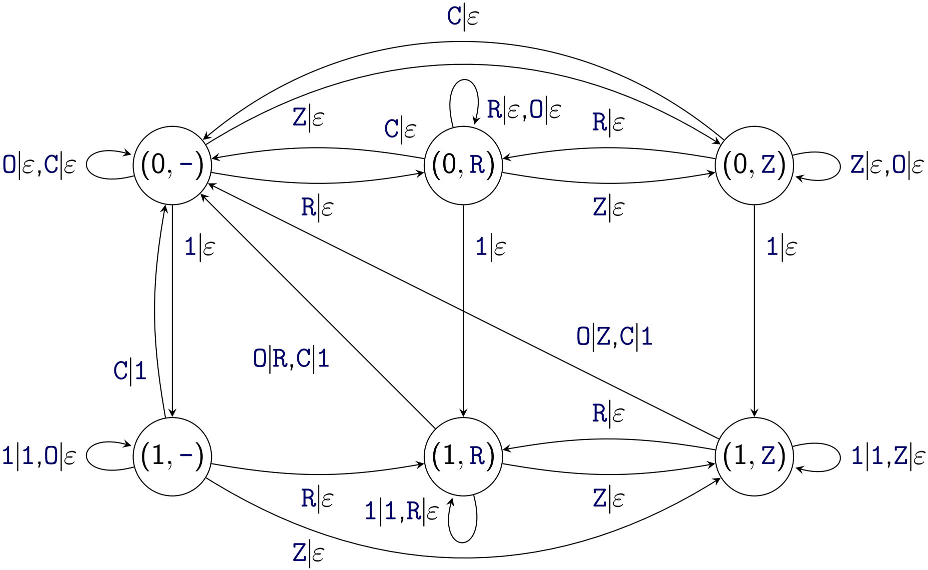
Formalisierung des Getränkeautomaten: Zustandsübergänge (2b)




C
Z
R,O
C
R
O,C
(0, -)
(0, R)
(0, Z)
Z,O
R
Z
1
1
1
C
O,C
O,C
R
1,O
(1, -)
(1, R)
(1, Z)
1,Z
R
Z
1,R
Z

Formalisierung des Getränkeautomaten: Ausgaben (1)

- zweiter wesentlicher Aspekt jedes Automaten: Ausgaben
  - wozu sollte man ihn sonst laufen lassen
  - jedenfalls ab und zu
- Ausgaben hier:
  - Euro Rückgeld
  - gewählte Flasche
- Ausgabealphabet $Y = \{\mathtt{1}, \mathtt{R}, \mathtt{Z}\}$
- 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 $g : Z \times X \to Y^*$
    - Funktionswert ε: „keine Ausgabe“
- Auch g üblicherweise in den Zustandsübergangsdiagrammen mit angegeben in der Form $x|g(z,x)$


C|ε
R|ε,O|ε
Z|ε
C|ε
R|ε
O|ε,C|ε
(0, -)
(0, R)
(0, Z)
Z|ε,O|ε
R|ε
Z|ε
1|ε
1|ε
1|ε
O|Z,C|1
O|R,C|1
C|1
R|ε
1|1,O|ε
(1, -)
(1, R)
(1, Z)
1|1,Z|ε
R|ε
1|1,R|ε
Z|ε
Z|ε

Was ist wichtig

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

Überblick

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

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

Darstellung als Graph:

- Knoten: Zustände
- Kanten: $z \xrightarrow{x|g(z,x)} f(z,x)$
- Anfangszustand: in der Darstellung durch kleinen Pfeil gekennzeichnet: $\longrightarrow z$

- erreichter Zustand nach Eingabe eines ganzen Wortes $w \in X^*$?
- alle durchlaufenen Zustände Eingabe eines ganzen Wortes $w \in X^*$?
- 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 $f^* : Z \times X^* \to Z$:

$$\begin{aligned} f^*(z,\varepsilon) &= z \\ \forall w \in X^* : \forall x \in X : \quad f^*(z,wx) &= f(f^*(z,w),x) \end{aligned}$$

- alternativ:

$$\begin{aligned} \bar{f}^*(z,\varepsilon) &= z \\ \forall w \in X^* : \forall x \in X : \quad \bar{f}^*(z,xw) &= \bar{f}^*(f(z,x),w) \end{aligned}$$

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

- erreichter Zustand nach Eingabe eines ganzen Wortes $w \in X^*$?
- alle durchlaufenen Zustände Eingabe eines ganzen Wortes $w \in X^*$?
- 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 $f^* : Z \times X^* \to Z$:

$$\begin{aligned} & f^*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : \quad & f^*(z, wx) = f(f^*(z, w), x) \end{aligned}$$

- alternativ:

$$\begin{aligned} & \bar{f}^*(z, \varepsilon) = z \\ \forall w \in X^* : \forall x \in X : \quad & \bar{f}^*(z, xw) = \bar{f}^*(f(z, x), w) \end{aligned}$$

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

Verallgemeinerte Zustandsübergangsfunktionen (1)

- erreichter Zustand nach Eingabe eines ganzen Wortes $w \in X^*$?
- alle durchlaufenen Zustände Eingabe eines ganzen Wortes $w \in X^*$?
- 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 $f^* : Z \times X^* \to Z$:

$$\begin{aligned} f^*(z, \varepsilon) &= z \\ \forall w \in X^* : \forall x \in X : \quad f^*(z, wx) &= f(f^*(z, w), x) \end{aligned}$$

- alternativ:

$$\begin{aligned} \bar{f}^*(z, \varepsilon) &= z \\ \forall w \in X^* : \forall x \in X : \quad \bar{f}^*(z, xw) &= \bar{f}^*(f(z, x), w) \end{aligned}$$

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

alle durchlaufenen Zustände:

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

$$\begin{aligned} f^{**}(z, \varepsilon) &= z \\ \forall w \in X^* : x \in X : \qquad f^{**}(z, wx) &= f^{**}(z, w) \cdot f(f^*(z, w), x) \end{aligned}$$

- auch hier wieder eine alternative Definitionsmöglichkeit ...
- das wird hoffentlich gleich noch nützlich sein
  (wenn die Zeit reicht)

Verallgemeinerte Ausgabefunktionen

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

$$\begin{aligned} g^*(z, \varepsilon) &= \varepsilon \\ g^*(z, wx) &= g(f^*(z, w), x) \end{aligned}$$

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

$$\begin{aligned} g^{**}(z, \varepsilon) &= \varepsilon \\ g^{**}(z, wx) &= g^{**}(z, w) \cdot g^*(z, wx) \end{aligned}$$

  - das wird am Freitag nützlich sein beim Codieren/Decodieren mit Mealy-Automaten

Was ist wichtig

Das sollten Sie mitnehmen:

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

Das sollten Sie üben:

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

Überblick

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

- Allgemeines Modell
- Spezialfall: Akzeptoren

Überblick

Erstes Beispiel: ein Getränkeautomat

Mealy-Automaten

Moore-Automaten

- Allgemeines Modell
- Spezialfall: Akzeptoren

Moore-Automat

- manchmal näherliegend: Automat produziert „in jedem Zustand“ eine Ausgabe (nicht bei Zustandsübergang)
- (endlicher) Moore-Automat festgelegt durch
  - eine endliche *Zustandsmenge* Z,
  - einen *Anfangszustand* $z_0 \in Z$,
  - ein *Eingabealphabet* X,
  - eine *Zustandsüberführungsfunktion* $f : Z \times X \to Z$,
  - ein *Ausgabealphabet* Y,
  - eine *Ausgabefunktion* $h : Z \to Y^*$
- Definition von f^* und f^{**} wie bei Mealy-Automaten
- Definition von h^* und h^{**} analog zu g^* und g^{**}
- Darstellung als Graph mit Knoten $z|h(z)$ oder $\dfrac{z}{h(z)}$

Überblick

Endliche Akzeptoren

wichtiger Sonderfall: 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 der *akzeptierenden Zustände*:
  - $F = \{z \mid h(z) = 1\}$
- in graphischen Darstellungen

  akzeptierende Zustände mit doppeltem Kringel gemalt: 

Einfache „Syntaxanalyse“ mit endlichen Akzeptoren

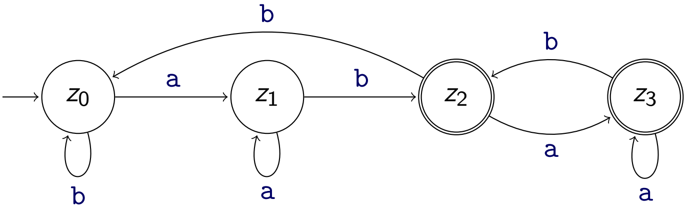
- Wort $w \in X^*$ wird *akzeptiert*, falls $f^*(z_0, w) \in F$.
- Die von einem Akzeptor A *akzeptierte formale Sprache* ist

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

Beispiel einer erkennbaren Sprache

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

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



Beispiel einer *nicht* erkennbaren Sprache

- Behauptung: Die formale Sprache

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

  kann von keinem endlichen Akzeptor erkannt werden.
- Beweisskizze:
  - indirekt: angenommen A wäre ein Akzeptor mit $L(A) = L$.
  - sei $k = |Z|$
  - betrachte die Eingabe $w = \mathtt{a}^k \mathtt{b}^k$
  - $f^{**}(z_0, w)$ besteht aus $k + 1$ Zuständen
  - soviele verschiedene gibt es gar nicht
  - also kommt mindestens einer doppelt vor:
    A läuft in einer Schleife; Länge sei $\ell > 0$
  - entferne das Teilwort für einen Schleifendurchlauf aus w:
    $w' = \mathtt{a}^{k-\ell} \mathtt{b}^k \notin L$
  - $f^*(z_0, w') = f^*(z_0, \mathtt{a}^{k-\ell}\mathtt{b}^k) = f^*(f^*(z_0, \mathtt{a}^{k-\ell}), \mathtt{b}^k) =$
    $f^*(f^*(z_0, \mathtt{a}^k), \mathtt{b}^k) \in F$, aber $w' \notin L$ Widerspruch

Was ist wichtig

Das sollten Sie mitnehmen:

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

Das sollten Sie üben:

- zu vorgegebener formaler Sprache Akzeptor konstruieren, der sie erkennt
- von vorgebenenem Akzeptor erkannte formale Sprache herausfinden

Zusammenfassung

- Mealy-Automaten
- Moore-Automaten
- insbesonderen Akzeptoren:
  - primitive Syntaxanalyse
  - aber oft nützlich