

Endliche Automaten III

Grundzüge digitaler Systeme

Vortrag von: Wolfgang Dvořák

Endliche Automaten & Reguläre Sprachen

- Endliche Automaten
- Weitere Typen von Automaten
 - Transducer
 - Mealy-Automaten
 - Moore-Automaten
 - Büchi-Automaten
- Umsetzung von Automaten als Schaltwerk
- Modellierungsbeispiele



Modellierung mit endlichen Automaten

Vorgangsweise:

- 1 Was sind die Zustände des Systems?
 - 1 Wieviele sind notwendig?
 - 2 Zustandsbezeichnungen?
 - 3 Identifiziere schlechte/ungültige Zustände
- 2 Startzustand? Endzustände?
- 3 Was sind die Aktionen/Eingaben, die zu Zustandsübergängen führen? Bezeichnung?
- 4 Was sind die Aktionen/Ausgaben, die bei Zustandsübergängen stattfinden? Bezeichnung?
- 5 Lege für jeden Zustand und jede Eingabe die Folgezustände und die Ausgaben fest.

Moore Automat für Bitfolge

Realisieren Sie einen Moore-Automaten, der

- „1“ ausgibt, wenn am Eingang die Bitfolge 1011 aufgetreten ist
- Überlappungen solcher Bitfolgen sollen nicht gewertet werden

Was sind die Zustände des Systems?

- Formen die letzten Eingaben ein Präfix des Musters?
- Jedes Präfix benötigt einen eigenen Zustand

Zustand	gelesene Folge
S_0	-
S_1	1
S_2	10
S_3	101
S_4	1011

Startzustand? Endzustände?

- Startzustand: S_0 (Die letzten Eingaben sind kein (nicht-leeres) Präfix des Musters)
- Endzustände: Moore-Automat definiert keine Endzustände

Moore Automat für Bitfolge

Realisieren Sie einen Moore-Automaten, der

- „1“ ausgibt, wenn am Eingang die Bitfolge 1011 aufgetreten ist
- Überlappungen solcher Bitfolgen sollen nicht gewertet werden

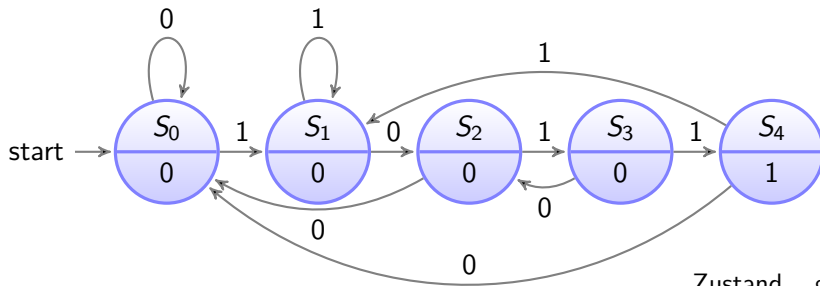
	Zustand	gelesene Folge
Was sind die Aktionen/Eingaben, die zu Zustandsübergängen führen?	S_0	-
	S_1	1
	S_2	10
	S_3	101
Was sind die Aktionen/Ausgaben, die bei Zustandsübergängen stattfinden?	S_4	1011

- Ausgabe 1 wenn das Muster gefunden wurde
- Ausgabe 0 sonst

Lege für jeden Zustand und jede Eingabe die Folgezustände und die Ausgaben fest.

...

Entwurf des Zustandsgraphen



Zustand	gelesene Folge
S_0	-
S_1	1
S_2	10
S_3	101
S_4	1011

Pantscherei

Pantscherei

3 Wasserkrüge à 3, 5 bzw. 7 Liter

Beginn: 3- und 7-Liter-Krug randvoll, 5-Liter-Krug leer

Ziel: Umleeren ohne Wasserverlust, bis sich in einem Krug genau ein Liter befindet.

Zustände:

- Wasserverteilung charakterisiert den Zustand.
- Tripel $(i, j, k) \dots i/j/k \ell$ in 3/5/7-Liter-Krug
 \hookrightarrow max. $4 \times 6 \times 8 = 192$ Zustände.
- Wasser geht nicht verloren $\Rightarrow i + j + k = 10$
 \hookrightarrow max. 18 Zustände $(0, 3, 7), (0, 4, 6), \dots, (3, 5, 2)$
- Mind. ein Krug ist immer leer oder voll
 \hookrightarrow max. 13 Zustände.
- **Startzustand:** $(3, 0, 7)$
- **Endzustände:** $(1, x, y), (x, 1, y), (\cancel{x}, y, 1)$.

Pantscherei

Pantscherei

3 Wasserkrüge à 3, 5 bzw. 7 Liter

Beginn: 3- und 7-Liter-Krug randvoll, 5-Liter-Krug leer

Ziel: Umleeren ohne Wasserverlust, bis sich in einem Krug genau ein Liter befindet.

Aktionen:

- (A,B) Fülle Krug A in Krug B
↪ 6 Aktionen: (3, 5), (3, 7), (5, 3), (5, 7), (7, 3), (7, 5)
- Ist Krug B voll oder Krug A leer passiert nichts
↪ Kein Fortschritt ⇒ Schleife oder verbotener Übergang
- Sei ℓ das Minimum aus der Restkapazität von Krug A und dem Inhalt von B
↪ Bei $(A, B) = (3, 5)$ wechselt der Zustand (i, j, k) zu $(i - \ell, j, k + \ell)$ usw.
- Keine Ausgabe
↪ DEA

Pantscherei

Pantscherei

3 Wasserkrüge à 3, 5 bzw. 7 Liter

Beginn: 3- und 7-Liter-Krug randvoll, 5-Liter-Krug leer

Ziel: Umleeren ohne Wasserverlust, bis sich in einem Krug genau ein Liter befindet.

Übergänge:

	δ	(3, 5)	(3, 7)	(5, 3)	(5, 7)	(7, 3)	(7, 5)
SZ	3, 0, 7	0, 3, 7	—	—	—	—	3, 5, 2
	0, 3, 7	—	—	3, 0, 7	—	3, 3, 4	0, 5, 5
	3, 5, 2	—	0, 5, 5	—	3, 0, 7	—	—
	3, 3, 4	1, 5, 4	0, 3, 7	—	3, 0, 7	—	3, 5, 2
	0, 5, 5	—	—	3, 2, 5	0, 3, 7	3, 2, 5	—
EZ	1, 5, 4	—	0, 5, 5	3, 3, 4	1, 2, 7	3, 5, 2	—
	3, 2, 5	0, 5, 5	1, 2, 7	—	3, 0, 7	—	3, 5, 2
EZ	1, 2, 7	0, 3, 7	—	3, 0, 7	—	3, 2, 5	1, 5, 4

5-Tupel $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$

- Zustände $Q = \{(3, 0, 7), (0, 3, 7), (3, 5, 2), (3, 3, 4), (0, 5, 5), (1, 5, 4), (3, 2, 5), (1, 2, 7)\}$
- Eingabealphabet: $\Sigma = \{(3, 5), (3, 7), (5, 3), (5, 7), (7, 3), (7, 5)\}$
- Übergangsfunktion: δ siehe Tabelle auf der vorherigen Folie
- Startzustand: $q_0 = (3, 0, 7)$
- Endzustände: $F = \{(1, 5, 4), (1, 2, 7)\}$

Die Sprache des Automaten $\mathcal{L}(\mathcal{A})$ gibt die Lösungen für unsere Pantscherei

Zählerei

Sei Σ ein Alphabet und $L' \subseteq \Sigma^*$ eine Sprache. Für ein Wort $w \in L'$ und ein atomares Symbol $s \in \Sigma$ bezeichnen wir mit $|w|_s$ die Anzahl an Symbolen s im Wort w .

Geben Sie (graphisch) einen **deterministischen** endlichen Automaten für folgende Sprache an:

$$\{w \in \{x, y, z\}^* \mid |w|_x - |w|_y \equiv 1 \pmod{5}\}$$

Zustände:

- Wir müssen uns die Differenz $|w|_x - |w|_y$ für die bisherige Eingabe w merken
 - Wir benötigen aber nicht den exakten Wert, sondern nur den Wert mod 5
- Fünf Zustände: $\{0, \dots, 4\}$
 - Zustand $i \dots |w|_x - |w|_y \equiv i \pmod{5}$
- Startzustand: 0
- Endzustand: 1

Zählerei

Zählerei

Sei Σ ein Alphabet und $L' \subseteq \Sigma^*$ eine Sprache. Für ein Wort $w \in L'$ und ein atomares Symbol $s \in \Sigma$ bezeichnen wir mit $|w|_s$ die Anzahl an Symbolen s im Wort w .

Geben Sie (graphisch) einen **deterministischen** endlichen Automaten für folgende Sprache an:

$$\{w \in \{x, y, z\}^* \mid |w|_x - |w|_y \equiv 1 \pmod{5}\}$$

Übergänge:

- Eingabe x Zähler inkrementieren
- Eingabe y Zähler dekrementieren
- Eingabe z : Nichts tun

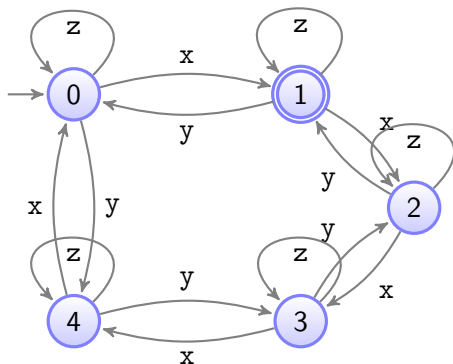
Zählerei

Zählerei

Sei Σ ein Alphabet und $L' \subseteq \Sigma^*$ eine Sprache. Für ein Wort $w \in L'$ und ein atomares Symbol $s \in \Sigma$ bezeichnen wir mit $|w|_s$ die Anzahl an Symbolen s im Wort w .

Geben Sie (graphisch) einen **deterministischen** endlichen Automaten für folgende Sprache an:

$$\{w \in \{x, y, z\}^* \mid |w|_x - |w|_y \equiv 1 \pmod{5}\}$$

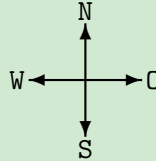
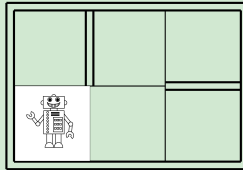


Roboter im Labyrinth

Ein Roboter wird in einem Labyrinth ausgesetzt, um es zu erkunden.

- Er erhält Steuerbefehle N (Nord), S (Süd), W (West) und O (Ost)
- Gelingt die Bewegung soll `ja` ausgegeben werden
- Ist die Bewegung durch eine Mauer blockiert soll `nein` ausgegeben werden

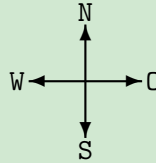
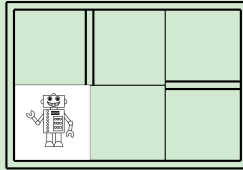
Geben Sie einen Automaten an, der das Antwort-Verhalten des Roboters für das folgende Labyrinth beschreibt.



Zustände

- Wir müssen die Position des Roboters speichern
- Ein Zustand pro Feld $Q = \{nw, sw, n, s, no, so\}$.
- Startzustand: `sw`

Roboter im Labyrinth



Zustände

- Wir müssen die Position des Roboters speichern
- Ein Zustand pro Feld $Q = \{nw, sw, n, s, no, so\}$.
- Startzustand: sw

Übergänge

- werden durch Steuersignale ausgelöst
- Eingabealphabet $\Sigma = \{N, S, W, O\}$.
- Es gibt eine Ausgabe
 - ↪ Transducer! Moore oder Mealy?
- Ausgabe hängt von Eingabe ab \Rightarrow Mealy

Roboter im Labyrinth

