F2 — Automaten und formale Sprachen

Matthias Jantzen
(nach und mit Folienvorlagen von Berndt Farwer)

Fachbereich Informatik
AB „Theoretische Grundlagen der Informatik“ (TGI)
Universität Hamburg
jantzen@informatik.uni-hamburg.de

Zielgruppe

Die Vorlesung wendet sich an Studierende der folgenden Studienrichtungen:

1. Informatik (2. Semester)
2. Wirtschaftsinformatik (2. Semester)
3. Nebenfach Informatik
4. Lehramt-Studium

Wichtig:

Beginn 10:15

Ende 11:45

Übungsgruppen

Mi 8–9	C-221	Heiko Rölke
Mi 9–10	C-221	Heiko Rölke
Mi 9–10	D-125a	Daniel Moldt
Mi 10–11	C-221	Michael Duvigneau
Mi 11–12	C-221	Matthias Jantzen
Mi 12–13	C-221	Daniel Moldt
Mi 12–13	F-132	Hedda Schmidtke
Mi 13–14	C-221	Michael Duvigneau
Mi 14–15	C-221	Michael Köhler
Mi 14–15	F-132	Michael Duvigneau
Mi 15–16	C-221	Michael Köhler
Mi 15–16	F-132	Jan Seedorf

Scheinkriterien

Bedingungen für die Ausstellung eines Scheines:

- 50% der erreichbaren Punkte (d.h. mindestens 80 von 160 Punkten)
- regelmäßige, **aktive** Teilnahme an den Übungsgruppen
- $2\times$ Vorrechnen an der Tafel
- max. zweimaliges unentschuldigtes Fehlen
- Bearbeiten aller 12 Übungszettel, auch wenn 50% schon erreicht worden ist!
- Gruppenarbeit: pro Gruppe *mindestens 2* und *höchstens 3* Personen

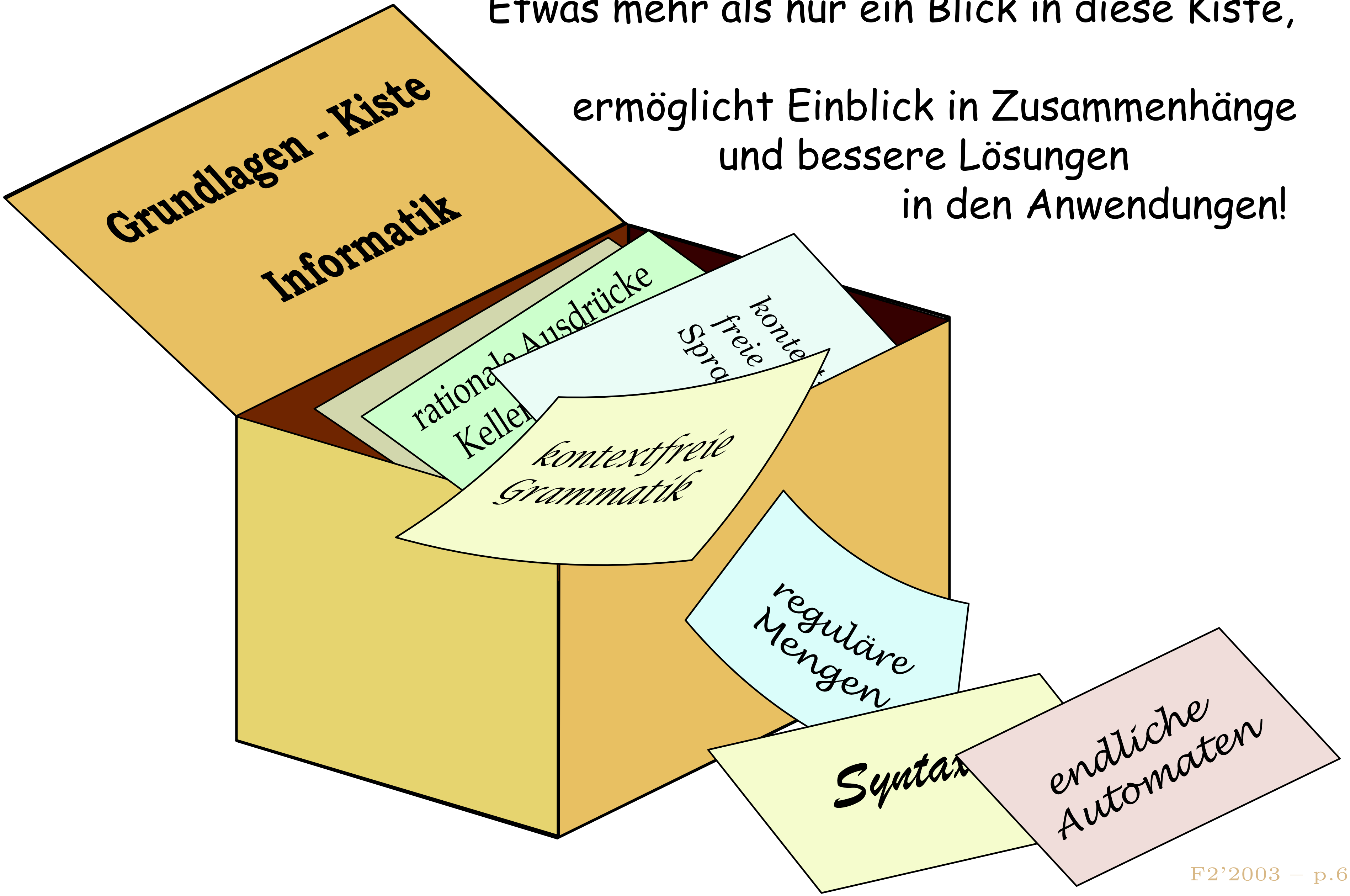
Skript

Das Skript zur Vorlesung ist erhältlich:

1. **gedruckt**
   - zum Mitnehmen im ersten Stock, Gebäude C, nördliches Treppenhaus (Ecke Haus C u. D!),
   - über die Übungsgruppe,
   - über das Sekretariat von TGI (C-218).
2. **elektronisch**
   - http://www.informatik.uni-hamburg.de
   - ⟶ Studium und Prüfungen ⟶ Skripte
     ... wird noch eingerichtet!
   - Die genaue URL lautet:
     http://www.informatik.uni-hamburg.de/TGI/lehre/vl/SS03/F2/index.html

Etwas mehr als nur ein Blick in diese Kiste,

ermöglicht Einblick in Zusammenhänge
und bessere Lösungen
in den Anwendungen!



Wissenschaft

was ist das?

Wissenschaft

was ist das?

- **Erkenntnisgewinn**

Wissenschaft

was ist das?

- **Erkenntnisgewinn**
- **systematische Entwicklung und Konstruktion von Neuem** mit gewünschten Eigenschaften und Verhalten.

Wissenschaft

was ist das?

- **Erkenntnisgewinn**
- **systematische Entwicklung und Konstruktion von Neuem** mit gewünschten Eigenschaften und Verhalten.
- **Dazu nötig ist:**

Wissenschaft

was ist das?

- **Erkenntnisgewinn**
- **systematische Entwicklung und Konstruktion von Neuem** mit gewünschten Eigenschaften und Verhalten.
- **Dazu nötig ist:**
  - Beschreibung dessen, was geistig erfasst wird. (Dinge, Sachverhalte, Zusammenhänge, . . .)

Wissenschaft

was ist das?

- **Erkenntnisgewinn**
- **systematische Entwicklung und Konstruktion von Neuem** mit gewünschten Eigenschaften und Verhalten.
- **Dazu nötig ist:**
  - Beschreibung dessen, was geistig erfasst wird. (Dinge, Sachverhalte, Zusammenhänge, . . .)
  - Erklärung dessen, was war und wie es zu dem wurde, was es nun ist. (Regeln der Art: Das muss so sein, weil . . .)

Wissenschaft

was ist das?

- **Erkenntnisgewinn**
- **systematische Entwicklung und Konstruktion von Neuem** mit gewünschten Eigenschaften und Verhalten.
- **Dazu nötig ist:**
  - Beschreibung dessen, was geistig erfasst wird. (Dinge, Sachverhalte, Zusammenhänge, . . .)
  - Erklärung dessen, was war und wie es zu dem wurde, was es nun ist. (Regeln der Art: Das muss so sein, weil . . .)
  - Vorhersage dessen, was sein wird. (Schlussfolgerungen, wenn–dann Aussagen, . . .)

Wie und womit?

Prinzipien

Wie und womit?

Prinzipien

- einfach

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Hilfsmittel

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Hilfsmittel

- Zeichen

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Hilfsmittel

- Zeichen
- Regeln

Wie und womit?

Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Hilfsmittel

- Zeichen
- Regeln
- Formalismen

Wie und womit?

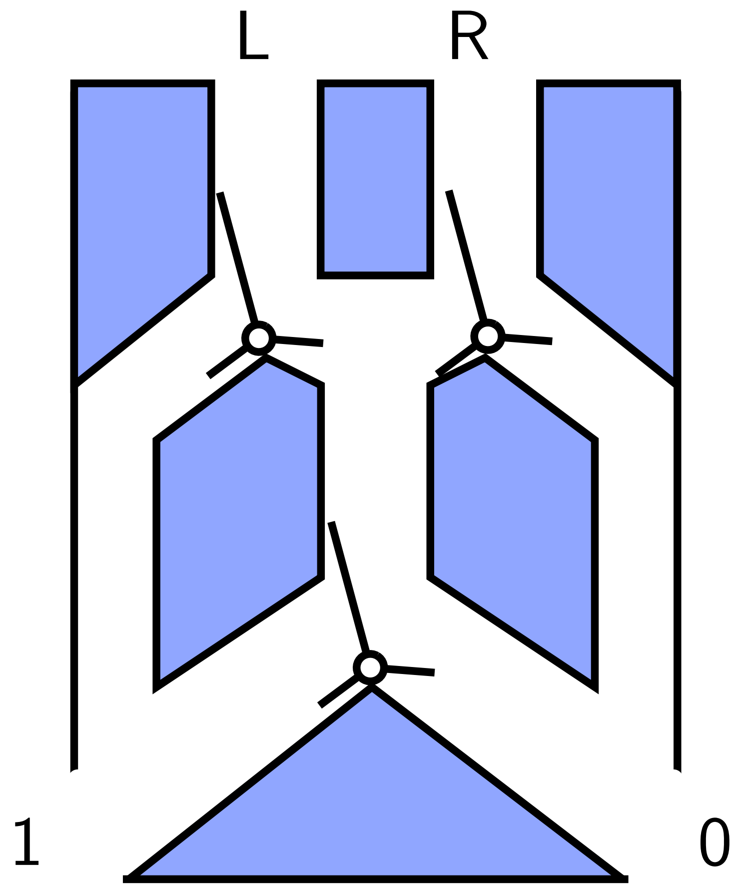
Prinzipien

- einfach
- nachvollziehbar
- eindeutig

Hilfsmittel

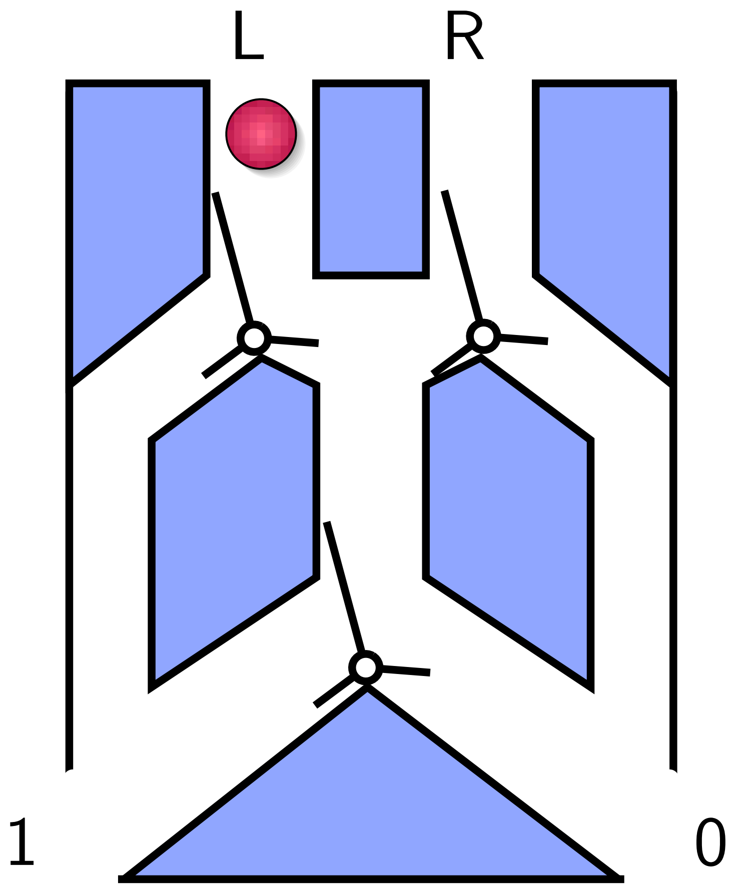
- Zeichen
- Regeln
- Formalismen
- Mathematik

Kugelautomat



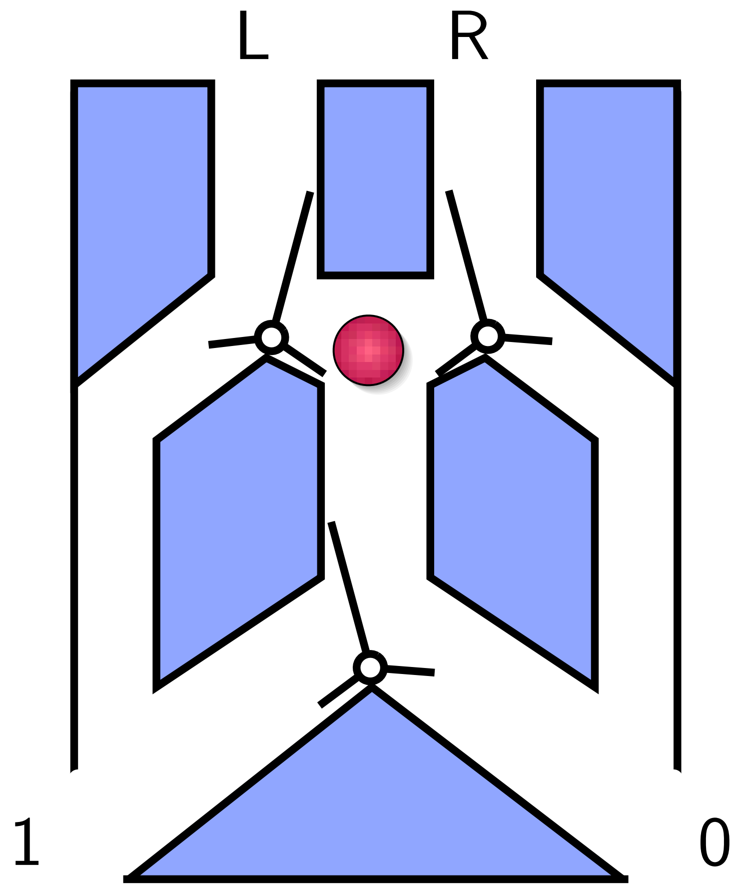
Gesucht ist eine *geeignete* **Darstellung** des möglichen Verhaltens . . .

Kugelautomat



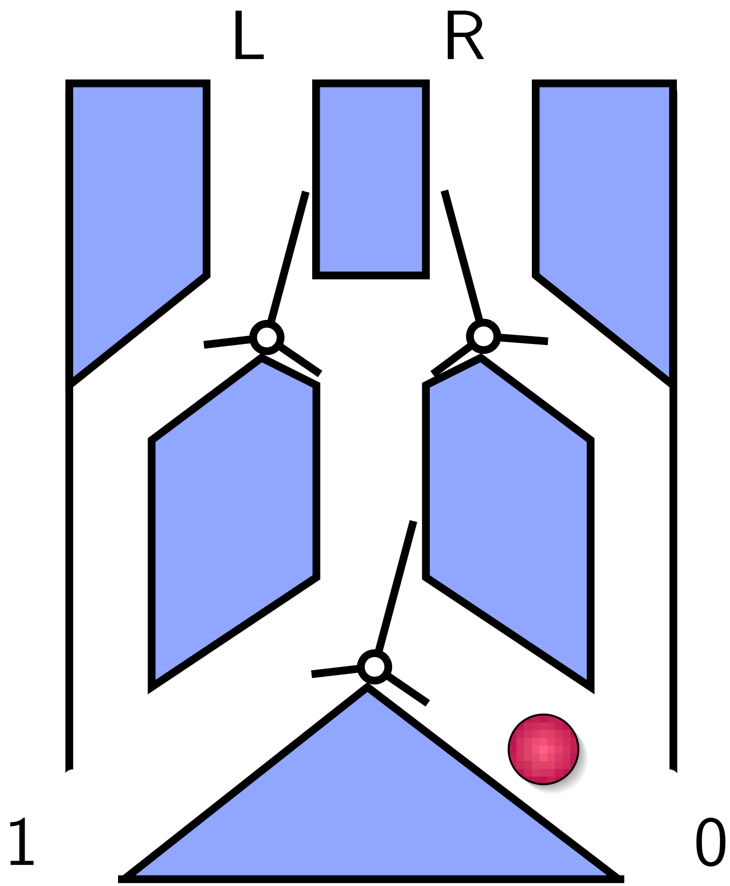
Werfen wir eine Kugel auf der linken Seite in den Kugelautomaten, . . .

Kugelautomat



. . . so verändert sich zunächst die Hebelstellung des linken oberen Hebels . . .

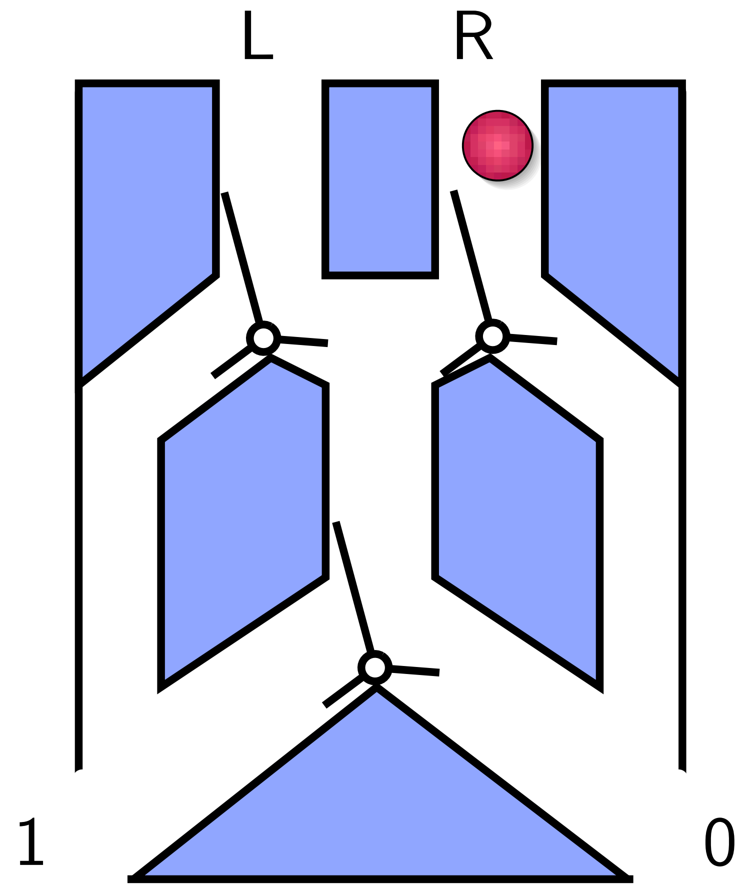
Kugelautomat



. . . und dann auch noch die des unteren Hebels.

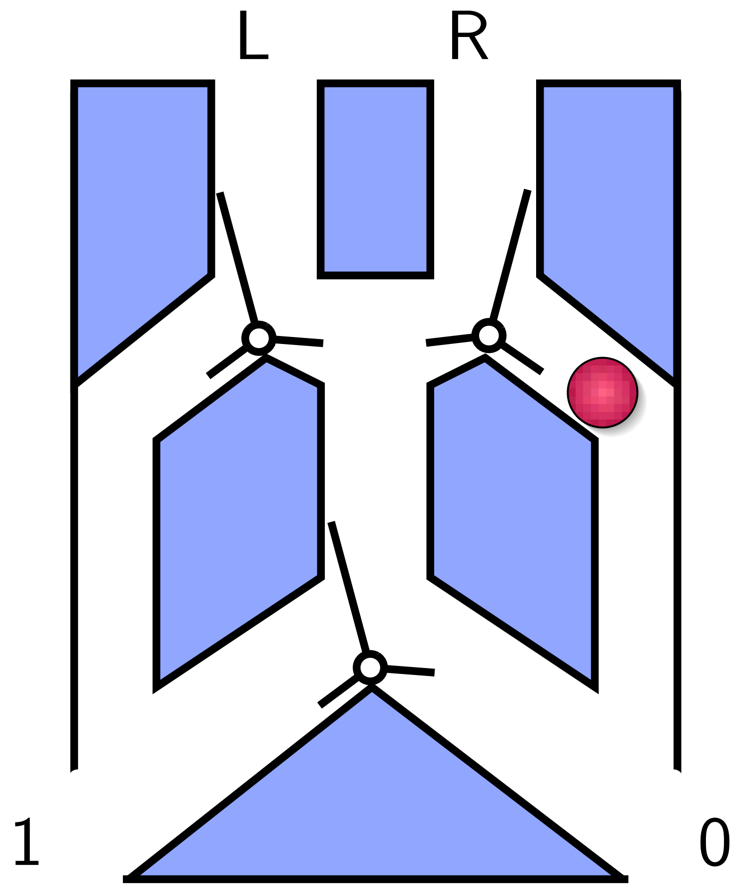
Die Kugel verläßt den Automaten bei Ausgang 0.

Kugelautomat



Werfen wir hingegen in der anfänglichen Situation rechts eine Kugel ein, . . .

Kugelautomat



. . . so ändert sich die Hebelstellung rechts oben und die Kugel verläßt den Automaten auf ebenfalls bei Ausgang 0.

Erste Erkenntnisse

Das Beispiel zeigt:

- Durch Aktionen wird der **interne Zustand** des Systems verändert.

Erste Erkenntnisse

Das Beispiel zeigt:

- Durch Aktionen wird der **interne Zustand** des Systems verändert.
- Unterschiedliche Eingaben können zu derselben Ausgabe führen.

Erste Erkenntnisse

Das Beispiel zeigt:

- Durch Aktionen wird der **interne Zustand** des Systems verändert.
- Unterschiedliche Eingaben können zu derselben Ausgabe führen.
- Frage: Führt dieselbe Eingabe immer zu derselben Ausgabe?

Erste Erkenntnisse

Das Beispiel zeigt:

- Durch Aktionen wird der **interne Zustand** des Systems verändert.
- Unterschiedliche Eingaben können zu derselben Ausgabe führen.
- Frage: Führt dieselbe Eingabe immer zu derselben Ausgabe?

Wir benötigen eine umfassende (formale) Beschreibung des Systems!

Interne Zustände

Der Kugelautomat befindet sich immer in einem

Zustand aus: $$Z := \left\{ \boxed{\begin{matrix}\backslash\ \backslash\\ \backslash\end{matrix}}, \boxed{\begin{matrix}\backslash\ /\\ \backslash\end{matrix}}, \boxed{\begin{matrix}\backslash\ \backslash\\ /\end{matrix}}, \boxed{\begin{matrix}\backslash\ /\\ /\end{matrix}}, \boxed{\begin{matrix}/\ \backslash\\ \backslash\end{matrix}}, \boxed{\begin{matrix}/\ /\\ \backslash\end{matrix}}, \boxed{\begin{matrix}/\ \backslash\\ /\end{matrix}}, \boxed{\begin{matrix}/\ /\\ /\end{matrix}} \right\}$$

- Sind alle diese Zustände auch erreichbar?

Interne Zustände

Der Kugelautomat befindet sich immer in einem

Zustand aus: $$Z := \left\{ \boxed{\begin{array}{c}\backslash\ \backslash\\ \backslash\end{array}}, \boxed{\begin{array}{c}\backslash\ /\\ \backslash\end{array}}, \boxed{\begin{array}{c}\backslash\ \backslash\\ /\end{array}}, \boxed{\begin{array}{c}\backslash\ /\\ /\end{array}}, \boxed{\begin{array}{c}/\ \backslash\\ \backslash\end{array}}, \boxed{\begin{array}{c}/\ /\\ \backslash\end{array}}, \boxed{\begin{array}{c}/\ \backslash\\ /\end{array}}, \boxed{\begin{array}{c}/\ /\\ /\end{array}} \right\}$$

- Sind alle diese Zustände auch erreichbar?
- Falls ja, was muß getan werden, um einen bestimmten Zustand herbeizuführen?

Interne Zustände

Der Kugelautomat befindet sich immer in einem

Zustand aus: $$Z := \left\{ \boxed{\substack{\backslash\ \backslash \\ \backslash}}, \boxed{\substack{\backslash\ / \\ \backslash}}, \boxed{\substack{\backslash\ \backslash \\ /}}, \boxed{\substack{\backslash\ / \\ /}}, \boxed{\substack{/\ \backslash \\ \backslash}}, \boxed{\substack{/\ / \\ \backslash}}, \boxed{\substack{/\ \backslash \\ /}}, \boxed{\substack{/\ / \\ /}} \right\}$$

- Sind alle diese Zustände auch erreichbar?
- Falls ja, was muß getan werden, um einen bestimmten Zustand herbeizuführen?
- Kann aus jedem beliebigen Zustand heraus wieder der Anfangszustand erreicht werden?

Interne Zustände

Der Kugelautomat befindet sich immer in einem

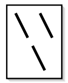
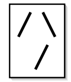
Zustand aus: $Z := \left\{ \boxed{\substack{\backslash\ \backslash \\ \backslash}}, \boxed{\substack{\backslash\ / \\ \backslash}}, \boxed{\substack{\backslash\ \backslash \\ /}}, \boxed{\substack{\backslash\ / \\ /}}, \boxed{\substack{/\ \backslash \\ \backslash}}, \boxed{\substack{/\ / \\ \backslash}}, \boxed{\substack{/\ \backslash \\ /}}, \boxed{\substack{/\ / \\ /}} \right\}$

- Sind alle diese Zustände auch erreichbar?
- Falls ja, was muß getan werden, um einen bestimmten Zustand herbeizuführen?
- Kann aus jedem beliebigen Zustand heraus wieder der Anfangszustand erreicht werden?

Zur Beschreibung des Systems gehören neben den Zuständen auch noch die **Aktionen** und deren **Wirkung**.

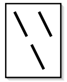
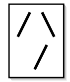
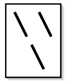
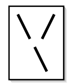
. . . mühsame Systembeschreibung

Wir können versuchen das Verhalten textuell zu beschreiben:

1. Wenn in der Situation  die Kugel bei **L** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .

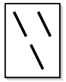
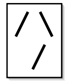
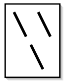
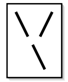
. . . mühsame Systembeschreibung

Wir können versuchen das Verhalten textuell zu beschreiben:

1. Wenn in der Situation  die Kugel bei **L** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .
2. Wenn in der Situation  die Kugel bei **R** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .

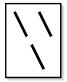
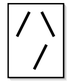
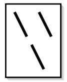
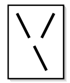
. . . mühsame Systembeschreibung

Wir können versuchen das Verhalten textuell zu beschreiben:

1. Wenn in der Situation  die Kugel bei **L** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .
2. Wenn in der Situation  die Kugel bei **R** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .
3. bis 16. (analog)

. . . mühsame Systembeschreibung

Wir können versuchen das Verhalten textuell zu beschreiben:

1. Wenn in der Situation  die Kugel bei **L** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .
2. Wenn in der Situation  die Kugel bei **R** eingeworfen wird, dann kommt sie beim Ausgang **0** heraus und die Hebelstellung danach ist: .
3. bis 16. (analog)

. . . Das ist ziemlich umständlich und zudem nicht gerade übersichtlich!!!

Alternative Beschreibungen

Das Verhalten des Kugelautomaten kann auch anders beschrieben werden:

(i) als Funktion:

$$\delta : Z \times \{L, R\} \longrightarrow Z \times \{0, 1\}$$

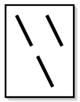
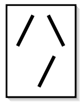
$$\text{mit} \quad \left(\boxed{\begin{array}{c} \backslash\ \backslash \\ \backslash \end{array}}, L \right) \mapsto \left(\boxed{\begin{array}{c} /\ \backslash \\ / \end{array}}, 0 \right), \ \ldots$$

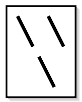
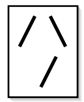
Alternative Beschreibungen

Das Verhalten des Kugelautomaten kann auch anders beschrieben werden:

(i) als Funktion:

$$\delta : Z \times \{L, R\} \longrightarrow Z \times \{0, 1\}$$

mit $\Big($  $, L\Big) \mapsto \Big($  $, 0\Big), \; \ldots$

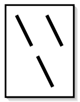
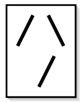
als Notation für $\delta\Big($  $, L\Big) := \Big($  $, 0\Big), \; \ldots$

Alternative Beschreibungen

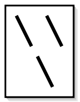
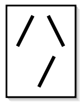
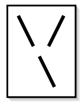
Das Verhalten des Kugelautomaten kann auch anders beschrieben werden:

(i) als Funktion:

$$\delta : Z \times \{L, R\} \longrightarrow Z \times \{0, 1\}$$

mit ( $, L) \mapsto ($  $, 0)$, . . .

(ii) als Tabelle:

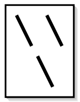
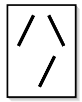
Zustand	Eingabe L	Eingabe R
	( $, 0)$	( $, 0)$
⋮	⋮	⋮

Alternative Beschreibungen

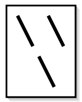
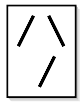
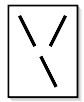
Das Verhalten des Kugelautomaten kann auch anders beschrieben werden:

(i) als Funktion:

$$\delta : Z \times \{L, R\} \longrightarrow Z \times \{0, 1\}$$

mit ( $, L$) $\mapsto$ ( $, 0$) $, \ \ldots$

(ii) als Tabelle:

Zustand	Eingabe L	Eingabe R
	( $, 0$)	( $, 0$)
$\vdots$	$\vdots$	$\vdots$

(iii) graphisch ...

Codierung

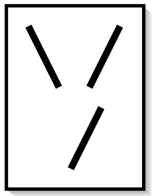
Zunächst eine abkürzende Schreibweise.

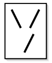
Wir kodieren Zustände durch Binärzahlen nach folgendem Schema:

Codierung

Zunächst eine abkürzende Schreibweise.

Wir kodieren Zustände durch Binärzahlen nach folgendem Schema:



code() :=

Codierung

Zunächst eine abkürzende Schreibweise.

Wir kodieren Zustände durch Binärzahlen nach folgendem Schema:

$$\boxed{\begin{matrix}\mathbf{0} & \mathbf{1} \\ \multicolumn{2}{c}{\mathbf{1}}\end{matrix}} \qquad \mathrm{code}(\boxed{\ }) := 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = [011]_2$$

Codierung

Zunächst eine abkürzende Schreibweise.

Wir kodieren Zustände durch Binärzahlen nach folgendem Schema:

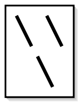
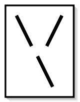
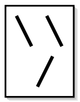
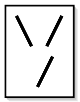
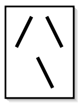
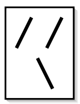
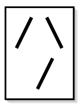
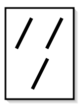
$$\boxed{\begin{array}{cc} \mathbf{0} & \mathbf{1} \\ \multicolumn{2}{c}{\mathbf{1}} \end{array}} \qquad \mathrm{code}(\boxed{\tiny\begin{array}{cc} \backslash & / \\ \multicolumn{2}{c}{/} \end{array}}) := 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = [011]_2$$

Allgemein:

$$\boxed{\begin{array}{cc} i_2 & i_0 \\ \multicolumn{2}{c}{i_1} \end{array}} \qquad \mathrm{code}(\boxed{\tiny\begin{array}{cc} i_2 & i_0 \\ \multicolumn{2}{c}{i_1} \end{array}}) := i_2 \cdot 2^2 + i_1 \cdot 2^1 + i_0 \cdot 2^0$$

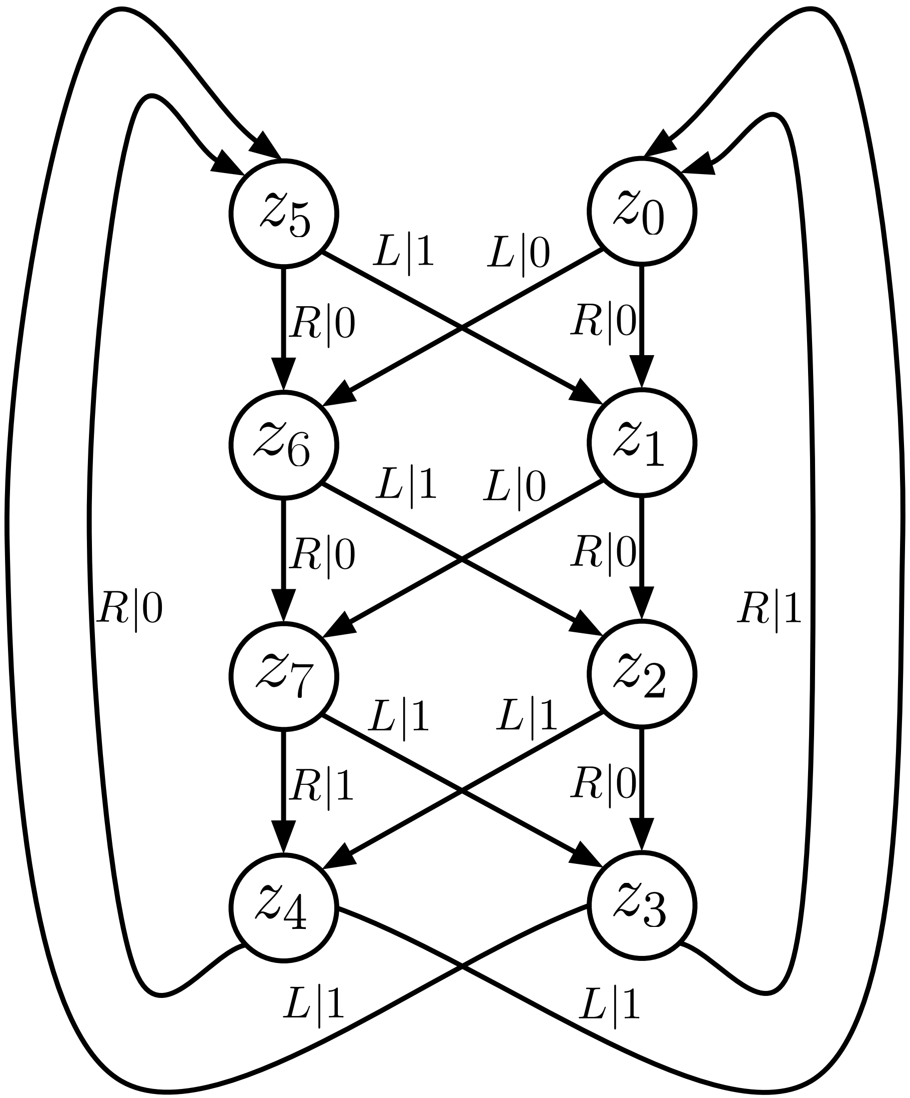
Zustandscodierung

. . . insgesamt ergeben sich folgende Codes:

Zustand	Binärdarstellung	Dezimaldarstellung
	$[000]_2$	$[0]_{10}$
	$[001]_2$	$[1]_{10}$
	$[010]_2$	$[2]_{10}$
	$[011]_2$	$[3]_{10}$
	$[100]_2$	$[4]_{10}$
	$[101]_2$	$[5]_{10}$
	$[110]_2$	$[6]_{10}$
	$[111]_2$	$[7]_{10}$

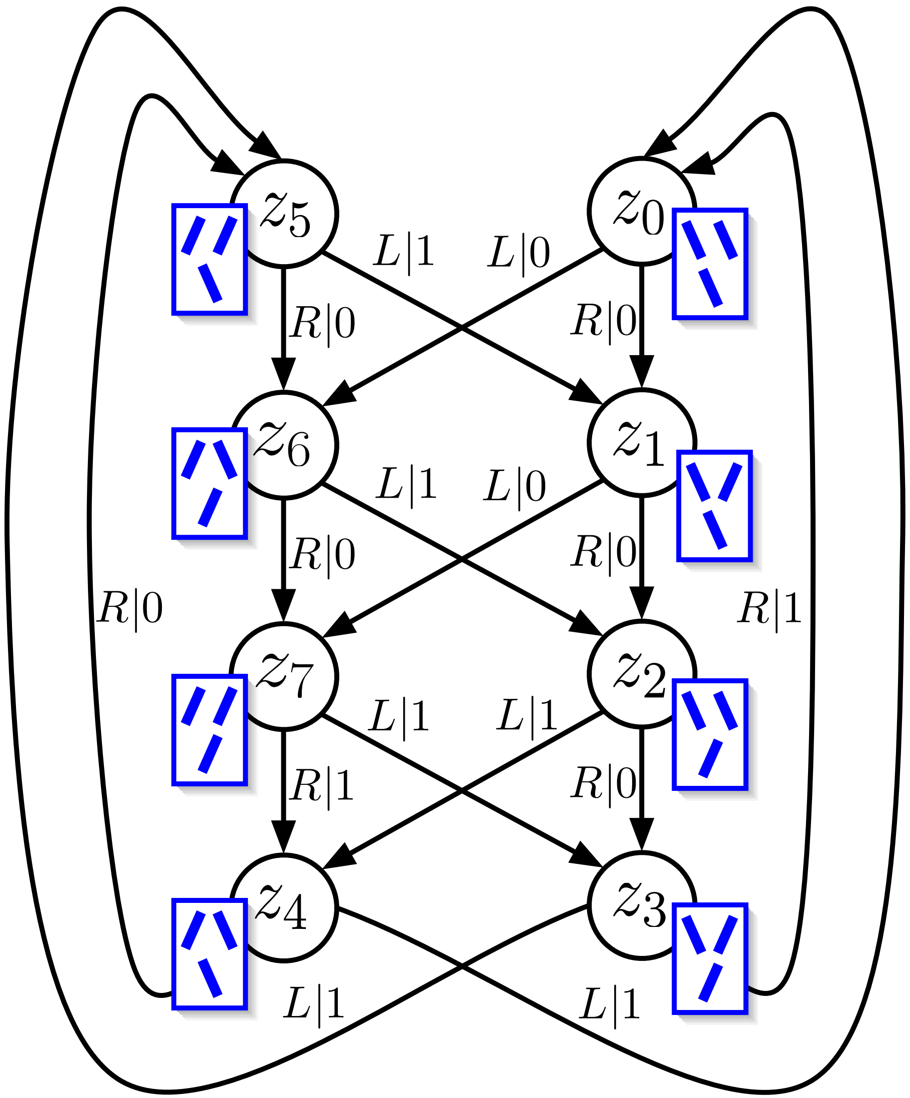
Graphische Beschreibung

Das **Zustandsübergangsdiagramm** für den Kugelautomaten hat die Form:



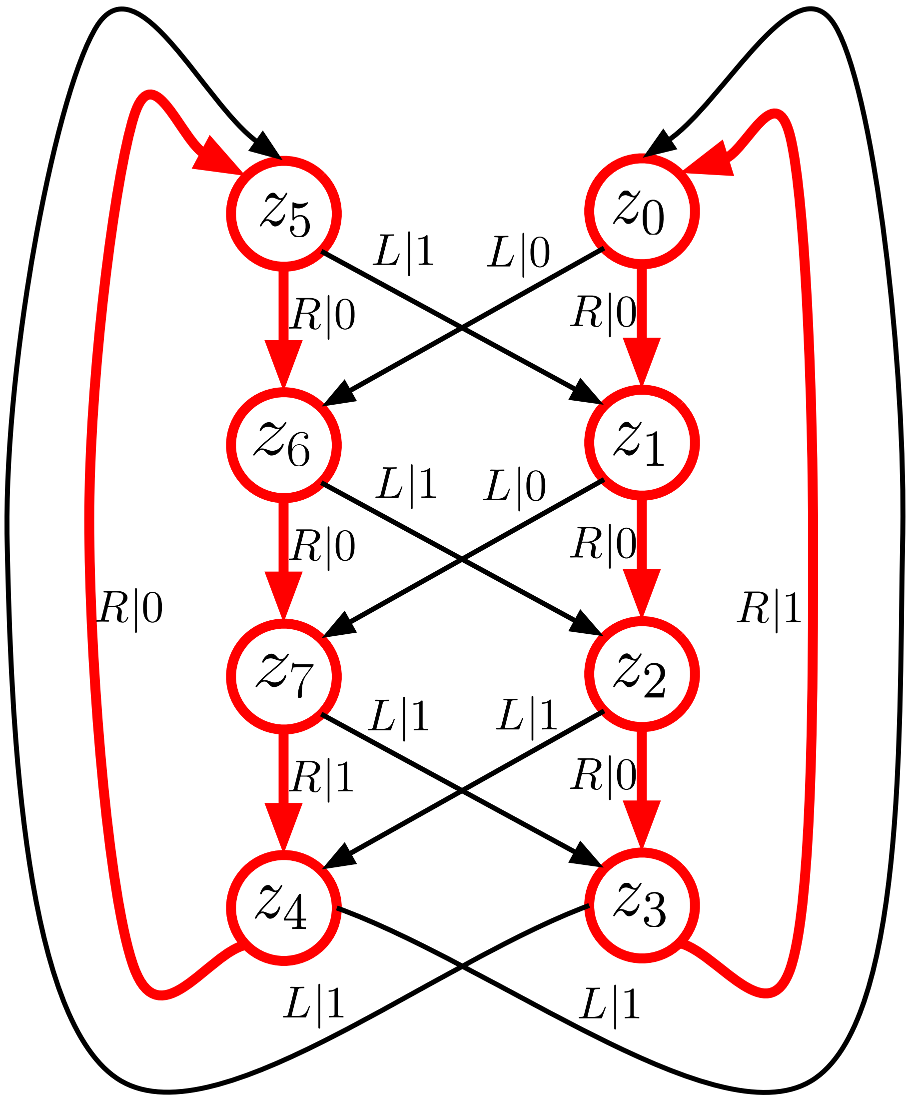
Graphische Beschreibung

Das **Zustandsübergangsdiagramm** für den Kugelautomaten hat die Form:



Systemeigenschaften

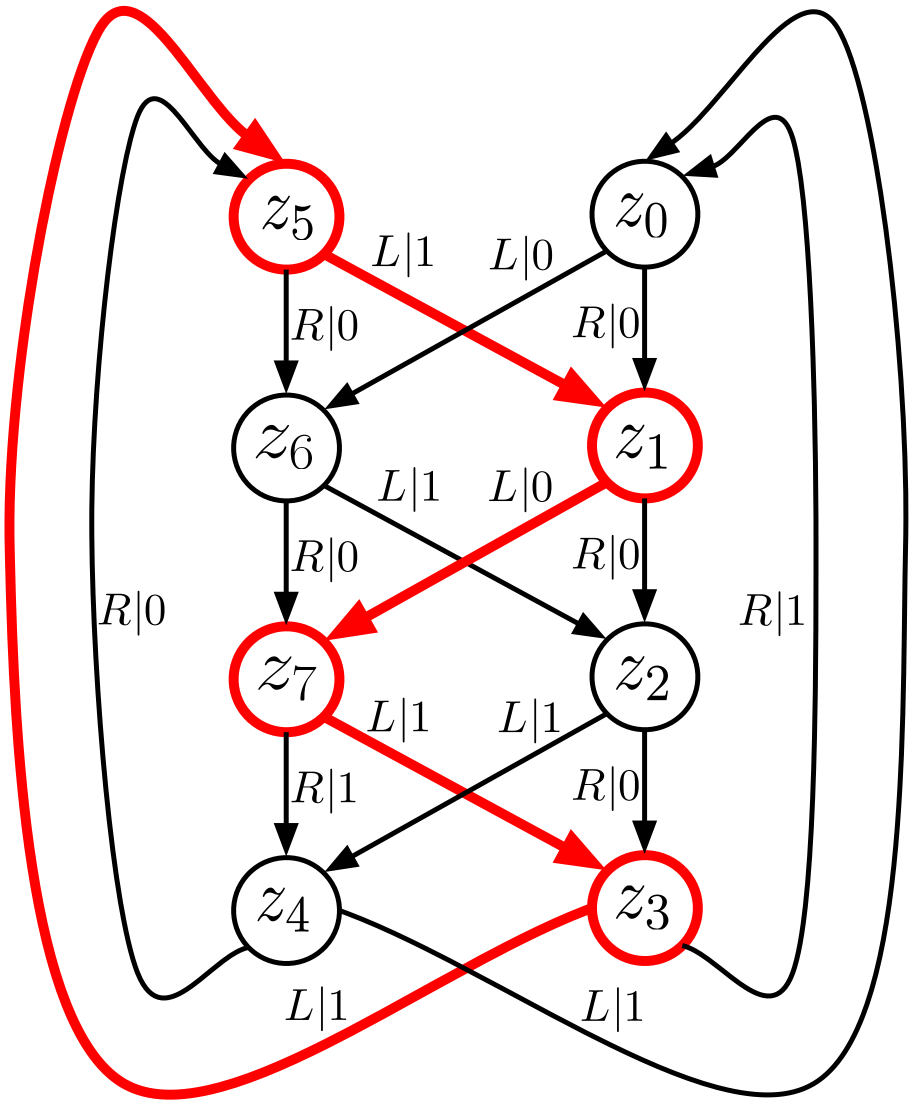
Einige Folgen von Aktionen führen zum Ausgangszustand zurück:



$RRRR$

Systemeigenschaften

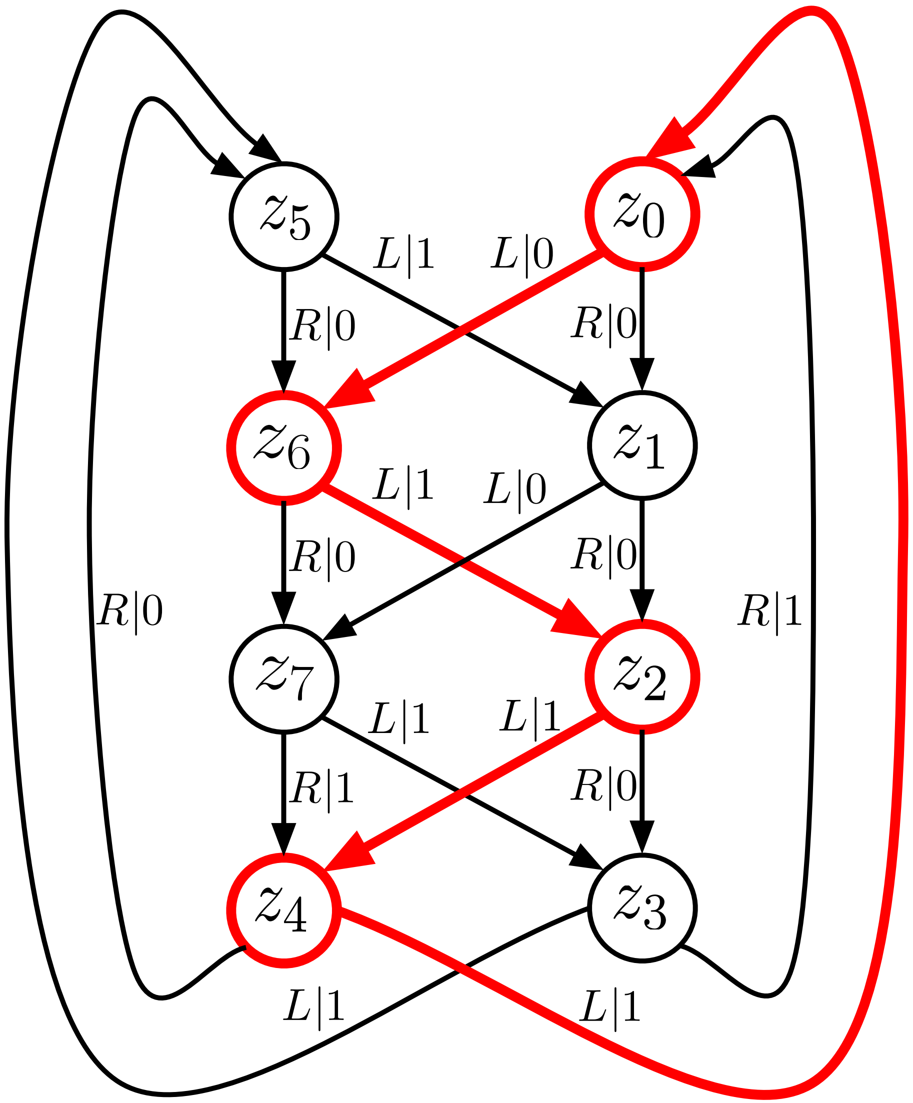
Einige Folgen von Aktionen führen zum Ausgangszustand zurück:



$LLLL$

Systemeigenschaften

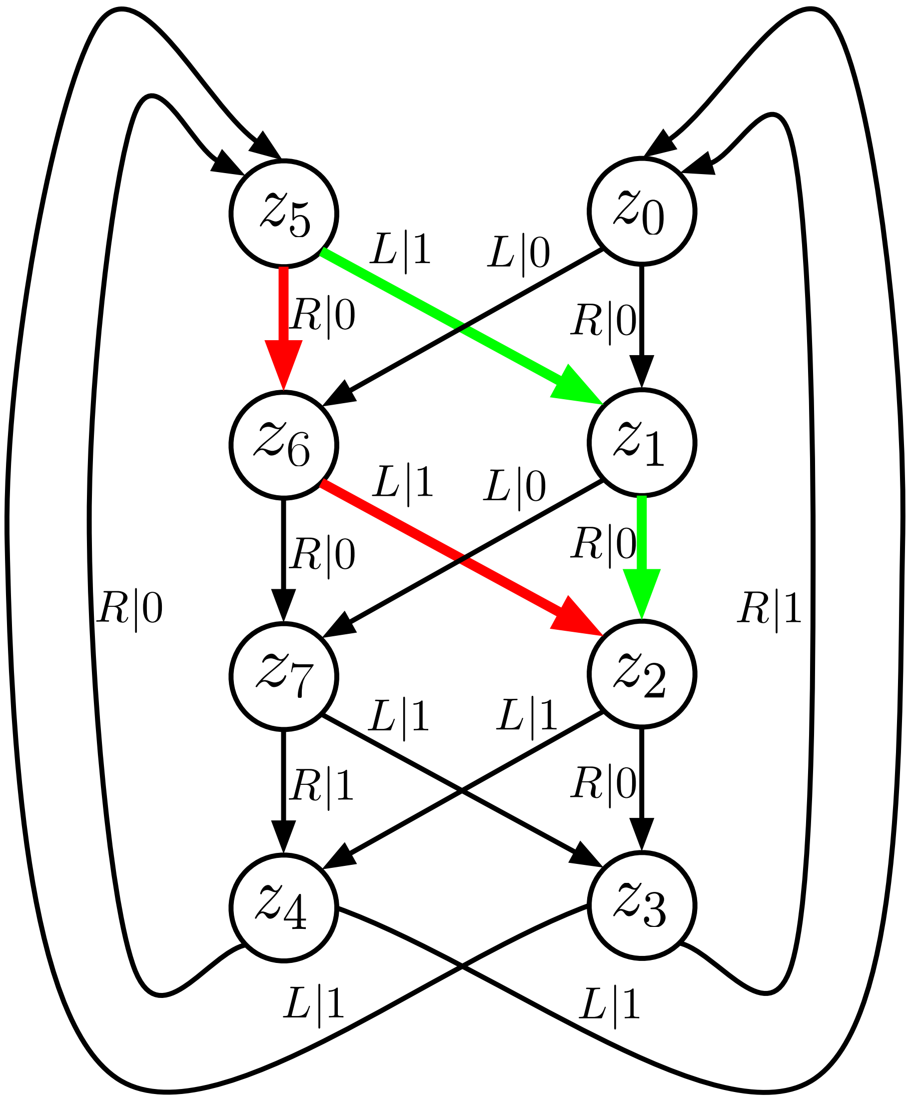
Einige Folgen von Aktionen führen zum Ausgangszustand zurück:



$LLLL$

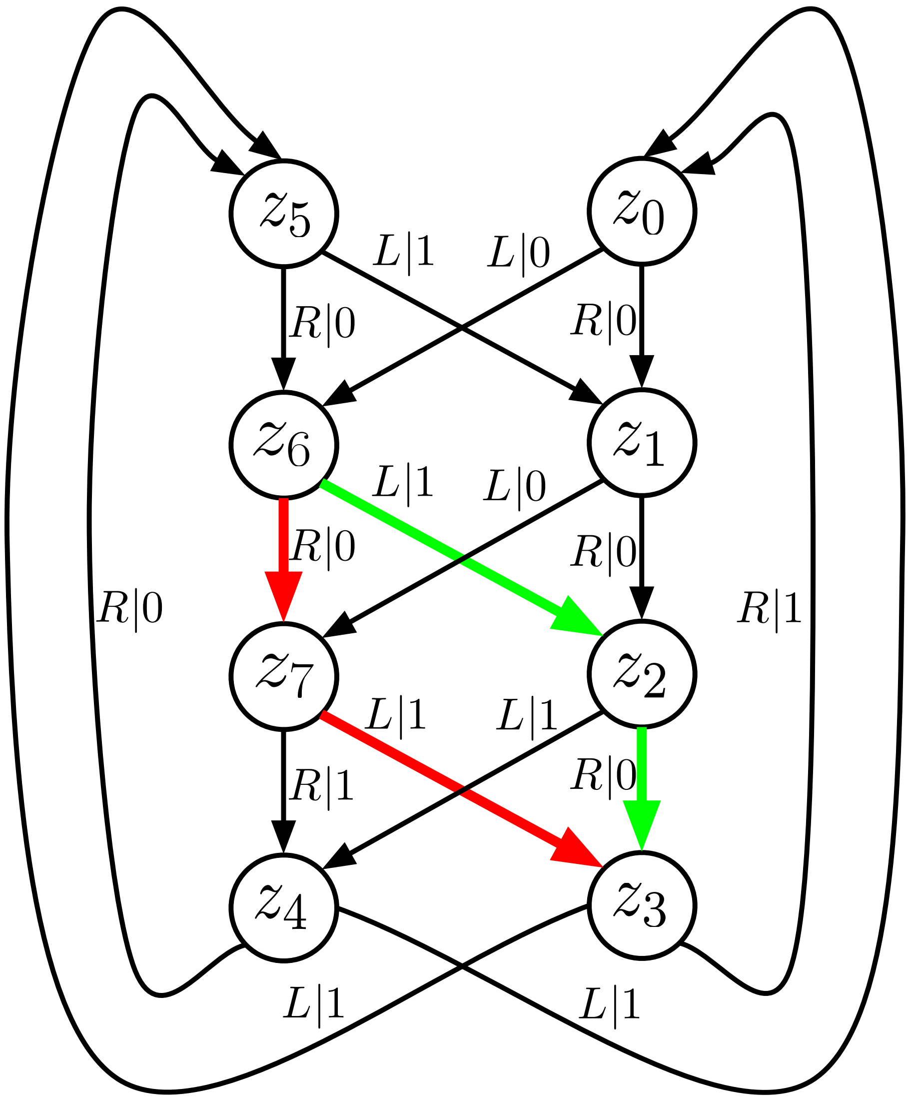
Äquivalenzen

Die Eingabe von LR und RL führt immer in den gleichen Zustand:



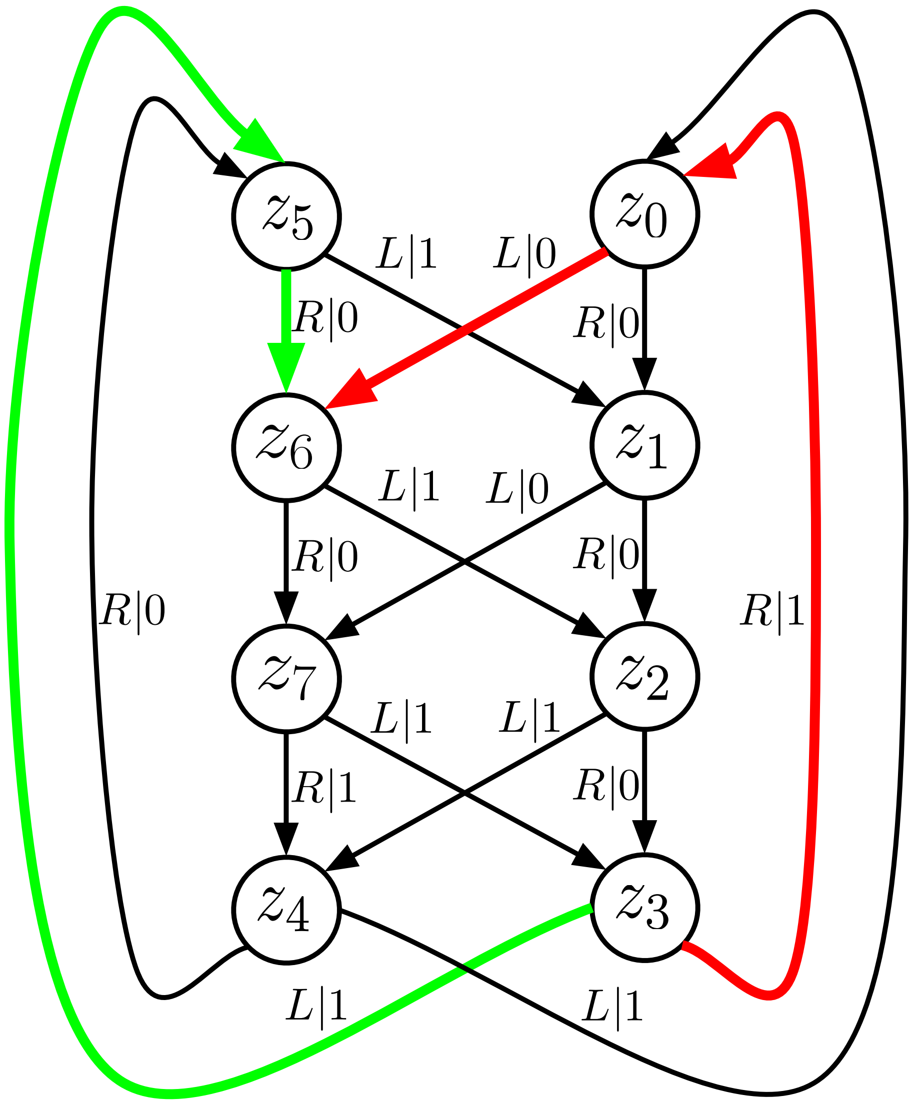
Äquivalenzen

Die Eingabe von LR und RL führt immer in den gleichen Zustand:



Äquivalenzen

Die Eingabe von LR und RL führt immer in den gleichen Zustand:



Konsequenzen

Möchten wir wissen, in welchen Zustand das Gerät gelangt, wenn wir beginnend im Zustand z_0 die Kugel z.B. in der Folge

$$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$$

einwerfen, so können wir dies am Zustandsdiagramm ablesen.

Konsequenzen

Möchten wir wissen, in welchen Zustand das Gerät gelangt, wenn wir beginnend im Zustand z_0 die Kugel z.B. in der Folge

$$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$$

einwerfen, so können wir dies am Zustandsdiagramm ablesen.

Alternative: **Vereinfachen der Folge**
Wende die folgenden Regeln solange an, wie dies möglich ist:

a) Wenn in der Folge eine der Sequenzen $LLLL$ oder $RRRR$ vorkommt, lösche diese. $(LLLL \longrightarrow \lambda)$

b) Wenn in der Folge die Sequenz RL auftritt, ersetze diese durch die Sequenz LR. $(RL \longrightarrow LR)$

Transformation

$$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$$

Transformation

$$LLRLRRLRLRLLRRLRLLRLRRLRRRLR\color{green}{RL}\color{black}{RRRLRR}$$

$$LLRLRRLRLRLLRRLRLLRLRRLRRRLRL\color{red}{RRRR}\color{black}{LRR}$$

Transformation

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLRRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLLRR$

Transformation

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLRRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLLRR$

$LLRLRRLRLRLLRRLRLLRLRLRRRRLRLLRR$

Transformation

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLRRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLLRR$

$LLRLRRLRLRLLRRLRLLRLRLRRRRLRLLRR$

$LLRLRRLRLRLLRRLRLLRLRLLRLLRR$

Transformation

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRRLRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLRRRRLRR$

$LLRLRRLRLRLLRRLRLLRLRRLRRRLRLLRR$

$LLRLRRLRLRLLRRLRLLRLRLRRRRLRLLRR$

$LLRLRRLRLRLLRRLRLLRLRLLRLLRR$

$$\vdots$$

$RLRLRLRR$

$LRLRLRRR$

$LLRLRRRR$

$LLLR$

…ist etwas einfacher als die ursprüngliche Folge!

Wozu Theoretische Informatik?

Wozu Theoretische Informatik?

Die schlechtesten Antworten:

- *Weil es in der Studienordnung steht!*
- *Weil es in der Prüfungsordnung steht!*

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

- **Fundierung** *versus* **Nutzung der Informatik!** (Um einen Computer zu benutzen muss man ihn nicht programmieren können.)

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

- **Fundierung** *versus* **Nutzung der Informatik!**
  (Um einen Computer zu benutzen muss man ihn nicht programmieren können.)
- **Konstruieren** *nicht* **Basteln!**

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

- **Fundierung** *versus* **Nutzung der Informatik!** (Um einen Computer zu benutzen muss man ihn nicht programmieren können.)
- **Konstruieren** *nicht* **Basteln!**
- präzise Analysen sind erforderlich!

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

- **Fundierung** *versus* **Nutzung der Informatik!** (Um einen Computer zu benutzen muss man ihn nicht programmieren können.)
- **Konstruieren** *nicht* **Basteln!**
- präzise Analysen sind erforderlich!
- Beherrschung von Komplexität anders nicht erreichbar!

Wozu Theoretische Informatik?

Bessere Begründungen und Ziele:

- **Fundierung** *versus* **Nutzung der Informatik!** (Um einen Computer zu benutzen muss man ihn nicht programmieren können.)
- **Konstruieren** *nicht* **Basteln!**
- präzise Analysen sind erforderlich!
- Beherrschung von Komplexität anders nicht erreichbar!
- Mathematik ist dafür ein **Denkzeug**!

Verwendete Notationen

Mengen

- **Darstellung** durch Aufzählung der Elemente oder Angabe von Eigenschaften:

$$M := \{0, 1, 2, \ldots, 7\} \quad \text{oder} \quad \{x \in I\!N \mid 0 \leq x \leq 7\}$$

Verwendete Notationen

Mengen

- **Darstellung** durch Aufzählung der Elemente oder Angabe von Eigenschaften:

  $$M := \{0, 1, 2, \ldots, 7\} \quad \text{oder} \quad \{x \in \mathbb{N} \mid 0 \leq x \leq 7\}$$

- $|M| = 8$ bezeichnet die **Kardinalität** von M.

Verwendete Notationen

Mengen

- **Darstellung** durch Aufzählung der Elemente oder Angabe von Eigenschaften:

  $M := \{0, 1, 2, \ldots, 7\}$ oder $\{x \in I\!N \mid 0 \leq x \leq 7\}$

- $|M| = 8$ bezeichnet die **Kardinalität** von M.
- **Operationen**: Seien A und B Mengen.
  - cartesisches Produkt: $A \times B$
  - Komplexprodukt: $A \cdot B$

Cartesisches Produkt

- Seien $A_1, A_2, \ldots, A_n$ Mengen und $x_1 \in A_1$, ..., $x_n \in A_n$, dann heißt $(x_1, \ldots, x_n)$ ein **(geordnetes)** n-**Tupel** von Elementen über $A_1, \ldots, A_n$.

Cartesisches Produkt

- Seien $A_1, A_2, \ldots, A_n$ Mengen und $x_1 \in A_1$, ..., $x_n \in A_n$, dann heißt $(x_1, \ldots, x_n)$ ein **(geordnetes)** n-**Tupel** von Elementen über $A_1, \ldots, A_n$.
- Die Menge aller geordneten n-Tupel $(x_1, \ldots, x_n)$ mit $x_1 \in A_1, \ldots, x_n \in A_n$ heißt **cartesisches Produkt** der Mengen A_1 bis A_n und wird geschrieben als $A_1 \times A_2 \ldots \times A_n$.

Cartesisches Produkt

- Seien $A_1, A_2, \ldots, A_n$ Mengen und $x_1 \in A_1$, ..., $x_n \in A_n$, dann heißt $(x_1, \ldots, x_n)$ ein **(geordnetes)** n-**Tupel** von Elementen über $A_1, \ldots, A_n$.
- Die Menge aller geordneten n-Tupel $(x_1, \ldots, x_n)$ mit $x_1 \in A_1, \ldots, x_n \in A_n$ heißt **cartesisches Produkt** der Mengen A_1 bis A_n und wird geschrieben als $A_1 \times A_2 \ldots \times A_n$.
- Es ist also $A_1 \times A_2 \times \ldots \times A_n := \{(x_1, \ldots, x_n) \mid x_1 \in A_1, \ldots, x_n \in A_n\}$.

Halbgruppe, Monoid

- Sei H eine Menge und $\odot : H \times H \longrightarrow H$ eine Abbildung für die das *Assoziativgesetz* gilt, d.h. $\forall a, b, c \in H : \; (a \odot b) \odot c = a \odot (b \odot c)$, dann heißt $(H, \odot)$ **Halbgruppe**.

Halbgruppe, Monoid

- Sei H eine Menge und $\odot : H \times H \longrightarrow H$ eine Abbildung für die das *Assoziativgesetz* gilt, d.h. $\forall a, b, c \in H : \ (a \odot b) \odot c = a \odot (b \odot c)$, dann heißt $(H, \odot)$ **Halbgruppe**.
- Eine Halbgruppe $(H, \odot)$ heißt **Monoid** genau dann, wenn es ein **neutrales Element** $e_H \in H$ gibt, so dass $e_H \odot m = m \odot e_H = m$ für jedes $m \in H$ gilt.

Halbgruppe, Monoid

- Sei H eine Menge und $\odot : H \times H \longrightarrow H$ eine Abbildung für die das *Assoziativgesetz* gilt, d.h. $\forall a, b, c \in H : \; (a \odot b) \odot c = a \odot (b \odot c)$, dann heißt $(H, \odot)$ **Halbgruppe**.
- Eine Halbgruppe $(H, \odot)$ heißt **Monoid** genau dann, wenn es ein **neutrales Element** $e_H \in H$ gibt, so dass $e_H \odot m = m \odot e_H = m$ für jedes $m \in H$ gilt.
- **Beispiel:** $(I\!N \setminus \{0\}, +)$ ist eine Halbgruppe, während $(I\!N, +)$ ein Monoid mit der Null als neutralem Element ist.

Komplexprodukt

- Für Teilmengen U, V einer Halbgruppe oder eines Monoides $(H, \odot)$ sei die Menge $U \cdot V := \{u \odot v \mid u \in U, v \in V\}$ das **Komplexprodukt** von U und V.

Komplexprodukt

- Für Teilmengen U, V einer Halbgruppe oder eines Monoides $(H, \odot)$ sei die Menge $U \cdot V := \{u \odot v \mid u \in U, v \in V\}$ das **Komplexprodukt** von U und V.
- Mit dem Komplexprodukt zweier Teilmengen einer Halbgruppe, wird die Halbgruppenoperation auf die elementweise Verknüpfung übernommen!

Komplexprodukt

- Für Teilmengen U, V einer Halbgruppe oder eines Monoides $(H, \odot)$ sei die Menge $U \cdot V := \{u \odot v \mid u \in U, v \in V\}$ das **Komplexprodukt** von U und V.
- Mit dem Komplexprodukt zweier Teilmengen einer Halbgruppe, wird die Halbgruppenoperation auf die elementweise Verknüpfung übernommen!
- **Beispiel:** Für Mengen von Wörtern ist die verwendete Operation die Konkatenation, so dass $U \cdot V$ alle Wörter enthält, die sich durch Hintereinanderschreiben eines Wortes aus U und eines Wortes aus V ergeben!

Relationen

- Eine Teilmenge $R \subseteq A_1 \times A_2 \times \ldots \times A_n$ heißt n-**stellige Relation** über A_1 bis A_n.

Relationen

- Eine Teilmenge $R \subseteq A_1 \times A_2 \times \ldots \times A_n$ heißt n-**stellige Relation** über A_1 bis A_n.
- Ist $n = 2$, so spricht man von einer **binären** statt von einer 2-stelligen **Relation**.

Relationen

- Eine Teilmenge $R \subseteq A_1 \times A_2 \times \ldots \times A_n$ heißt n-**stellige Relation** über A_1 bis A_n.
- Ist $n = 2$, so spricht man von einer **binären** statt von einer 2-stelligen **Relation**.
- **Beispiel:** Kugelautomat

$$\equiv \subseteq \{L, R\}^* \times \{L, R\}^*$$

mit

$$(LR, RL) \in \equiv \quad \text{und} \quad (RL, LR) \in \equiv$$

Eigenschaften von Relationen

Sei $R \subseteq A \times A$ eine Relation auf A. R heißt

- **reflexiv** gdw. $(a, a) \in R$ für alle $a \in A$ gilt.

Eigenschaften von Relationen

Sei $R \subseteq A \times A$ eine Relation auf A. R heißt

- **reflexiv** gdw. $(a, a) \in R$ für alle $a \in A$ gilt.
- **symmetrisch** gdw. für $(a, b) \in R$ stets auch $(b, a) \in R$ gilt.

Eigenschaften von Relationen

Sei $R \subseteq A \times A$ eine Relation auf A. R heißt

- **reflexiv** gdw. $(a, a) \in R$ für alle $a \in A$ gilt.
- **symmetrisch** gdw. für $(a, b) \in R$ stets auch $(b, a) \in R$ gilt.
- **transitiv** gdw. aus $(a, b) \in R$ und $(b, c) \in R$ stets $(a, c) \in R$ folgt. R ist also transitiv, wenn $R \cdot R \subseteq R$ gilt.

Eigenschaften von Relationen

Sei $R \subseteq A \times A$ eine Relation auf A. R heißt

- **reflexiv** gdw. $(a, a) \in R$ für alle $a \in A$ gilt.
- **symmetrisch** gdw. für $(a, b) \in R$ stets auch $(b, a) \in R$ gilt.
- **transitiv** gdw. aus $(a, b) \in R$ und $(b, c) \in R$ stets $(a, c) \in R$ folgt. R ist also transitiv, wenn $R \cdot R \subseteq R$ gilt.

Eine Relation heißt **Äquivalenzrelation** gdw. sie **reflexiv**, **symmetrisch** und **transitiv** ist.

Transitiver Abschluss

- Sei $R \subseteq A \times A$ eine Relation auf der Menge A. Dann seien R^+, der **transitive Abschluss** (auch **transitive Hülle**), und R^*, der **reflexive, transitive Abschluss**, von R wie folgt erklärt:

$$R^+ := \bigcup_{i \geq 1} R_i \text{ mit } R_1 := R \quad \text{und} \quad R_{i+1} := R_i \cdot R.$$

Transitiver Abschluss

- Sei $R \subseteq A \times A$ eine Relation auf der Menge A. Dann seien R^+, der **transitive Abschluss** (auch **transitive Hülle**), und R^*, der **reflexive, transitive Abschluss**, von R wie folgt erklärt:

$$R^+ := \bigcup_{i \geq 1} R_i \text{ mit } R_1 := R \quad \text{und} \quad R_{i+1} := R_i \cdot R.$$

Für Relationen R_1 und R_2 ist
$R_1 \cdot R_2 := \{(x, z) \mid \exists y : (x, y) \in R_1 \wedge (y, z) \in R_2\}$
das Relationenprodukt oder die Komposition. Wie auch beim Komplexprodukt, wird der Punkt ” $\cdot$ ” hier oft weggelassen.

Transitiver Abschluss

- Sei $R \subseteq A \times A$ eine Relation auf der Menge A. Dann seien R^+, der **transitive Abschluss** (auch **transitive Hülle**), und R^*, der **reflexive, transitive Abschluss**, von R wie folgt erklärt:

$$R^+ := \bigcup_{i \geq 1} R_i \text{ mit } R_1 := R \quad \text{und} \quad R_{i+1} := R_i \cdot R.$$

- Für eine Teilmenge $M \subseteq H$ einer Halbgruppe $(H, \odot)$ seien der **transitive Abschluss** M^+ sowie der **transitive und reflexive Abschluss** M^*

$$M^+ := \bigcup_{i \geq 1} M^i \text{ mit } M^1 := M \text{ und } M^{i+1} := M^i \cdot M$$

$$M^* := M^+ \cup \{e_H\}$$

Wortmengen

- Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen **Zeichen** (oder **Symbolen**).

Wortmengen

- Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen **Zeichen** (oder **Symbolen**).
- Für ein Alphabet Σ, ist Σ^* das **freie Monoid** mit der **Konkatenation** oder Hintereinanderschreibung der einzelnen Zeichen als Monoidoperation, und dem leeren Wort λ als neutralem Element. Für $w \in \Sigma^*$ schreiben wir w^k anstelle von $\underbrace{ww\cdots w}_{k}$.

  Jede Menge $L \subseteq \Sigma^*$ heißt **formale Sprache**.

Wortmengen

- Ein **Alphabet** ist eine (total geordnete) endliche Menge von unterschiedlichen **Zeichen** (oder **Symbolen**).
- Für ein Alphabet Σ, ist Σ^* das **freie Monoid** mit der **Konkatenation** oder Hintereinanderschreibung der einzelnen Zeichen als Monoidoperation, und dem leeren Wort λ als neutralem Element. Für $w \in \Sigma^*$ schreiben wir w^k anstelle von $\underbrace{ww \cdots w}_{k}$.

  Jede Menge $L \subseteq \Sigma^*$ heißt **formale Sprache**.
- Σ^* bezeichnet also die **Menge aller endlichen Wörter über dem Alphabet** Σ

Zahlensysteme

- Die b-**näre Darstellung** benutzt die Symbole der Ziffernmenge $B := \{0, 1, \ldots, b-1\}$. Ist $b = 2$, so spricht man von einer **binären** Zahlendarstellung.

Zahlensysteme

- Die b-**näre Darstellung** benutzt die Symbole der Ziffernmenge $B := \{0, 1, \ldots, b-1\}$. Ist $b = 2$, so spricht man von einer **binären** Zahlendarstellung.
- Eine Zahl $n \in I\!\!N$ wird zur Basis $b = |B|$ dargestellt durch eine Folge $a_k a_{k-1} \cdots a_0$ von Symbolen $a_i \in \{0, 1, \ldots, b-1\}$ gdw. $n = \sum_{i=0}^{k} a_i \cdot b^i$ gilt.

  Wir notieren dies durch $[a_k a_{k-1} \cdots a_0]_b = n$.

Ziele der Vorlesung

Wir wollen den Begriff der „**Berechenbarkeit**“ mit eingeschränkten Speichermodellen einführen:

1. konstanter Speicher

Ziele der Vorlesung

Wir wollen den Begriff der „**Berechenbarkeit**“ mit eingeschränkten Speichermodellen einführen:

1. konstanter Speicher
2. beliebig viel Speicher, aber eingeschränkter Zugriff

Ziele der Vorlesung

Wir wollen den Begriff der „**Berechenbarkeit**“ mit eingeschränkten Speichermodellen einführen:

1. konstanter Speicher
2. beliebig viel Speicher, aber eingeschränkter Zugriff
3. beliebig viel Speicher, beliebiger Zugriff

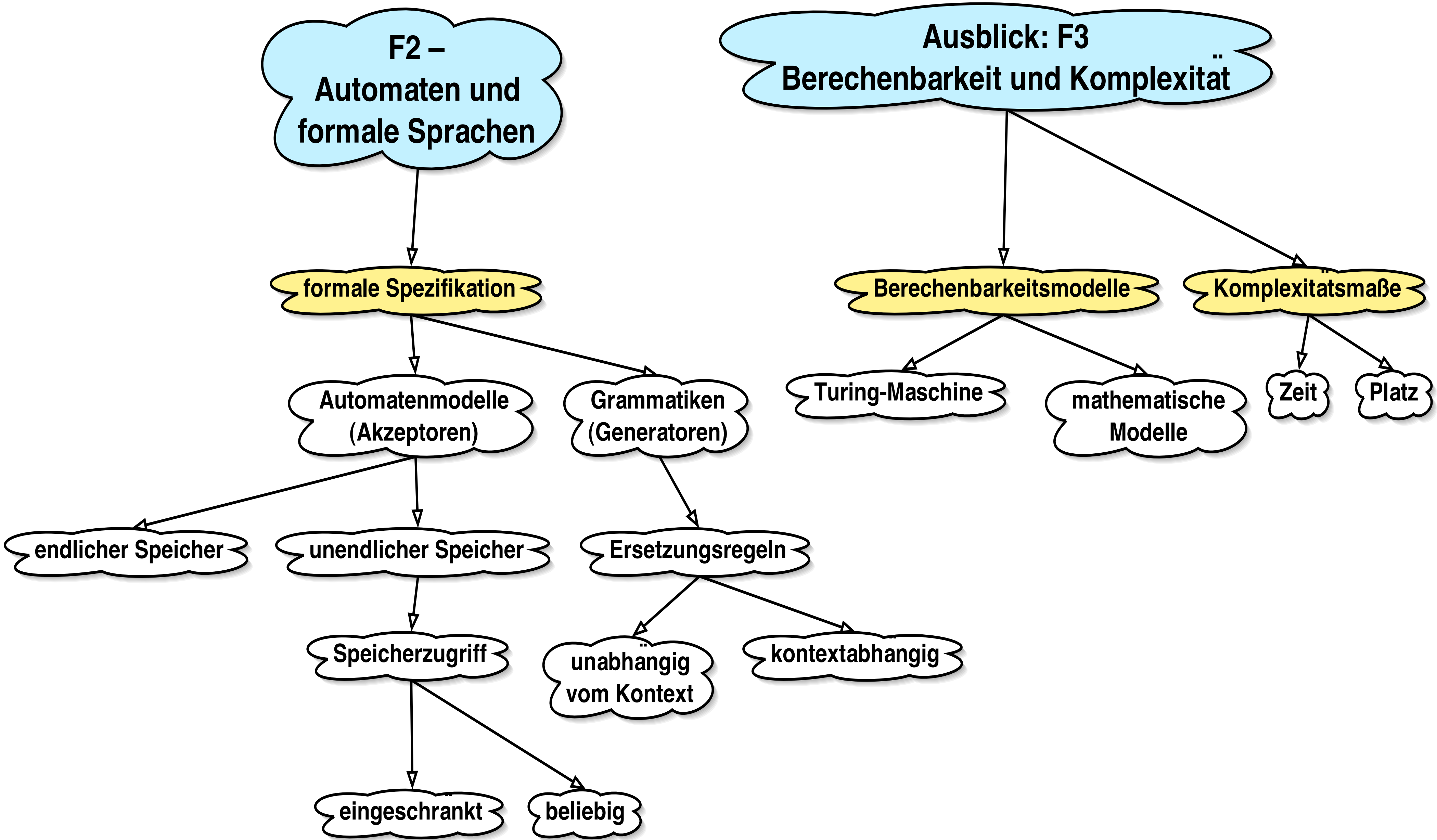
Ziele der Vorlesung

Wir wollen den Begriff der „**Berechenbarkeit**“ mit eingeschränkten Speichermodellen einführen:

1. konstanter Speicher
2. beliebig viel Speicher, aber eingeschränkter Zugriff
3. beliebig viel Speicher, beliebiger Zugriff

Ein allgemeiner Berechenbarkeitsbegriff wird in der Vorlesung F3 geprägt.

Überblick


F2 –
Automaten und
formale Sprachen
Ausblick: F3
Berechenbarkeit und Komplexität
formale Spezifikation
Berechenbarkeitsmodelle
Komplexitätsmaße
Automatenmodelle
(Akzeptoren)
Grammatiken
(Generatoren)
Turing-Maschine
mathematische
Modelle
Zeit
Platz
endlicher Speicher
unendlicher Speicher
Ersetzungsregeln
Speicherzugriff
unabhängig
vom Kontext
kontextabhängig
eingeschränkt
beliebig