

Universität
Rostock



Traditio et Innovatio

5. ILTiS 2024
Workshop 8

**Konzepte der Theoretischen Informatik
mit FLACI (Sek II)**


Ulrike Heiduck



28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

1

Universität
Rostock



Traditio et Innovatio

Arbeitsbereich: Formale Sprachen und Automaten

Sekundarstufe II, Gk & Lk

- Der Rahmenplan der Sekundarstufe II sieht für die Behandlung des Themas nur ca. 15 Unterrichtsstunden vor. Wie soll man 3 Automaten (Mealy-Automat, Akzeptor, Turing-Maschine) mit Definition, Aufbau und Arbeitsweise in dieser kurzen Zeit umfassend behandeln?
- Im Workshop wird ein Gang durch die Unterrichtseinheit diskutiert und das Werkzeug FLACI vorgestellt, mit dem Automaten effizient und anschaulich definiert und simuliert werden können.

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

2



**Universität
Rostock**
Traditio et Innovatio

Konzepte der theoretischen Informatik


Arbeitsbereich: Formale Sprachen und Automaten [MD5]

ca. 14/30 Unterrichtsstunden

integrativ

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
<p>Endliche Automaten</p> <ul style="list-style-type: none"> mithilfe ihrer Definition interpretieren eine Simulation zum Analysieren, Visualisieren und Implementieren nutzen Aufbau und Arbeitsweise anhand eines anschaulichen Modells beschreiben <p>Formale Sprache</p> <ul style="list-style-type: none"> verbal, durch Angabe eines Musters oder aller Wörter beschreiben mathematische Darstellungen einer Sprache interpretieren aus einem Syntaxdiagramm ableiten die von einem Automaten akzeptierte Sprache bestimmen 	<p>Die Überföhrungsfunktion und ggf. die Ausgabefunktion können sowohl in grafischer als auch tabellarischer Darstellung vorliegen.</p>

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTIS 2024 | Ulrike Heiduck
3



**Universität
Rostock**
Traditio et Innovatio

**Gk:
&
Lk:**


Endliche Automaten [MD5]

ca. 14/20 Unterrichtsstunden

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
<p>Mealy-Automat $MA = (X, Y, Z, \delta, \lambda, z_0)$</p> <ul style="list-style-type: none"> Überföhrungs- und Ausgabefunktion tabellarisch und grafisch darstellen einen Mealy-Automaten modellieren <p>Akzeptor $A = (X, Z, \delta, z_0, Z_E)$</p> <ul style="list-style-type: none"> Überföhrungsfunktion tabellarisch und grafisch darstellen einen Akzeptor anhand einer gegebenen Sprache modellieren eine Grenze von Akzeptoren erläutern <p>Turingmaschine $TM = (X, Z, \Gamma, \delta, z_0, \\$, Z_E)$</p> <ul style="list-style-type: none"> den Wert des Modells anhand der Church-Turing-These begründen die Grenzen einer Turingmaschine anhand des Halteproblems erklären 	<p>Eine Analyse realer Automaten bildet den Ausgangspunkt für die Abstraktion zum Mealy-Automaten</p> <p style="text-align: center; font-weight: bold; font-size: 1.2em;">Keine Grammatik/ keine Sprachentypen!</p> <p>z. B. anhand der Sprache $L(A) = \{a^n b^n \mid n \in \mathbb{N}\}$</p> <p>Die Turingmaschine ist sowohl als Akzeptor als auch als rechnende Maschine mit Ausgabe zu betrachten.</p>
zusätzlich für den Leistungskurs	

28.02.2024
© 2024 UNIVERSITÄT ROSTOCK | ILTIS 2024 | Ulrike Heiduck
4

Universität
Rostock



Traditio et Innovatio

Lk:

zusätzlich für den Leistungskurs

Kellerautomat $KA = (X, Z, \Gamma, \delta, z_0, k_0, Z_E)$

- das Prinzip eines Kellerspeichers und die Kelleroperationen push, pop und nop erläutern
- einen Kellerautomaten anhand einer gegebenen Sprache modellieren
- Prinzip des Nichtdeterminismus anhand der Erkennung von Palindromen erläutern
- eine Grenze von Kellerautomaten erläutern

Turingmaschine

- eine Turingmaschine anhand einer gegebenen Sprache modellieren

Es wird von folgender Variante eines Kellerautomaten ausgegangen: Ein Wort wird akzeptiert, wenn das Eingabewort vollständig gelesen wurde, der Keller leer ist und ein Endzustand erreicht wurde.

z. B. anhand der Sprache $L(A) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$

Ausschließlich für den Leistungskurs außerdem:

Formale Sprachen und Grammatiken (ca. 10 Unterrichtsstunden)


- Grammatik
- Chomsky-Hierarchie

28.02.2024

© 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

5

Universität
Rostock



Traditio et Innovatio

Ausschließlich Lk:

Formale Sprachen und Grammatiken [MD5]

ca. 10 Unterrichtsstunden

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
<div><div>ausschließlich für den Leistungskurs</div><div><div>Grammatik $G = (T, N, P, S)$</div><div><ul style="list-style-type: none">mithilfe der Definition beschreibenWörter nachvollziehbar ableitendie Ableitbarkeit von Wörtern untersuchendie erzeugte Sprache bestimmenzu einer gegebenen regulären Sprache eine Grammatik entwickeln</div><div>Chomsky-Hierarchie [Deutsch] [Philosophie]</div><div><ul style="list-style-type: none">den Typ einer Grammatik bestimmenden Zusammenhang zwischen Grammatik, Sprache und Automat mithilfe der Chomsky-Hierarchie beschreibenfür eine aus einer regulären oder kontextfreien Grammatik erzeugte Sprache einen erkennenden Automaten angeben</div></div></div>	


28.02.2024

© 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

6

3

**Universität
Rostock**



Traditio et Innovatio

Vorstellung der Workshopteilnehmer*innen

Wer ist gekommen, um nur FLACI auszuprobieren?


- Arbeitsblatt
- Ergänzung Tafelwerk

Kurzer Abriss über einen möglichen Ablauf des Unterrichts nach Rahmenplan

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

7

**Universität
Rostock**



Traditio et Innovatio

Vorschlag einer Aufteilung

Formale Sprachen und Automaten (14/20)

Gk 3 WS, Lk 5 WS

- Doppelstundenprinzip mit A/B-Woche
- Oder tatsächlich jede Woche auch eine Einzelstunde?

→ Gk 7 Doppelstunden

→ Lk 10 Doppelstunden

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

8

Mögliche Verteilung der Themen

	Gk & Lk		Lk
1	Zustandsorientierte Modellierung		
2	Mealy-Automat		
3	Darstellungsformen für formale Sprachen; Akzeptor		
4	Beispiele: Akzeptor anhand einer gegebenen Sprache modellieren		
5	Weitere Beispiele und Grenzen	5	Kellerautomat
		6	Beispiele: KA anhand einer gegebenen Sprache modellieren
		7	Beispiel Palindrome: Prinzip des Nichtdeterminismus
		8	Grenzen von KA; Turingmaschine
6	Turingmaschine; Church-Turing-These	9	TM anhand einer gegebenen Sprache modellieren
7	Grenzen der TM: Halteproblem	10	Grenzen der TM: Halteproblem

1. Einstieg

Was bedeutet überhaupt Theoretische Informatik?

- Gegenstand
- Zustandsorientierte Modellierung
 - Systemverhalten
 - Verhaltensbeschreibung

Zustandsorientierte Modellierung

→ <https://www.inf-schule.de/automaten-sprachen/zustandsmodellierung>

Einladung zum kurzen Durchspielen:

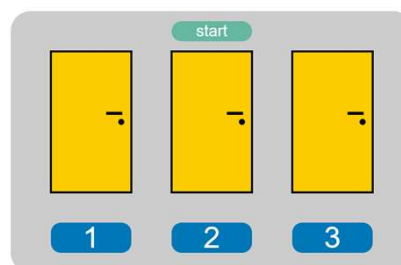
Erkundung Schaltsysteme

1. Systemverhalten
2. Verhaltensbeschreibung


Erkundung Schaltsysteme

Aufgabe:

Wie kann man mit den Schaltern die drei Türen öffnen?



Universität
Rostock



Traditio et Innovatio

Verhaltensbeschreibung


Lösung einfacher Systeme

Gedrückter Schalter	Türkonstellation
1	XXO
2	XOX
3	OXX

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

13

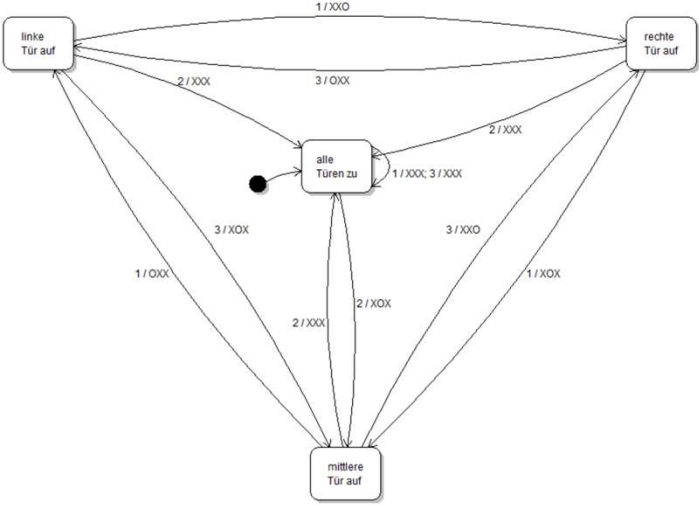
Universität
Rostock



Traditio et Innovatio

Verhaltensbeschreibung

Lösung bei notwendigen Bedingungen wie im System 3



28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

14

2. Realer Automat

aus Fahrgastsicht:



Übergang endlicher Automat

Beispiel: Kunstautomat

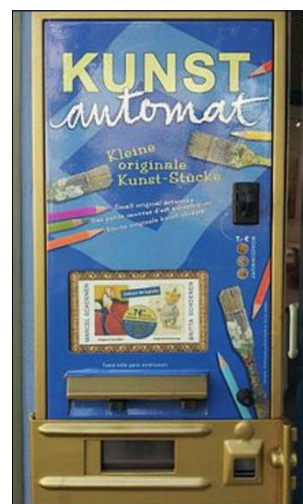
Ein Kunstautomat ist - nach Wikipedia ein Selbstbedienungsautomat, der kleine Kunstwerke oder Kunstbotschaften als Unikate spendet.

Auf inf-schule.de:

→ <https://www.inf-schule.de/automaten-sprachen/zustandsmodellierung/zustandsbasiertesysteme/erkundung-automaten/kunstautomat>

Anschauung bei youtube:

→ <https://www.youtube.com/watch?v=PKLj-cklqpE>





Universität
Rostock

Traditio et Innovatio

Informelle Beschreibung des Kunstautomaten:

Bestandteile
und Verhalten



Münzeinwurf

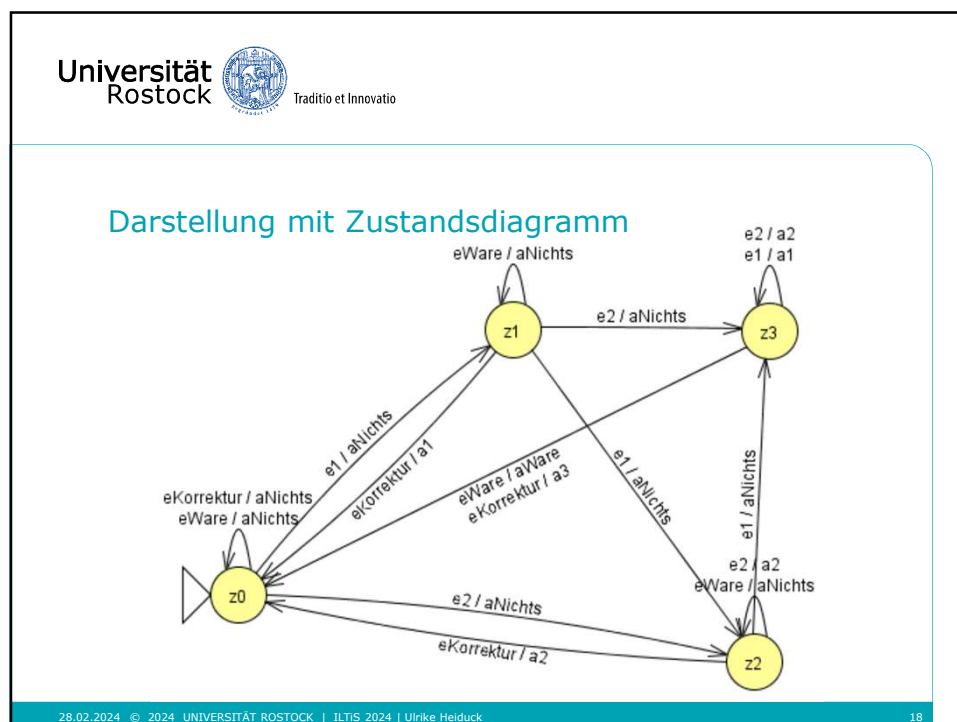
Korrekturknopf

Münzrückgabe

Warenknopf

Warenausgabe

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTIS 2024 | Ulrike Heiduck
17



Mealy-Automat

Definition

Ein **endlicher Automat mit Ausgabe** ist ein 6-Tupel $A = (X, Y, Z, \delta, \lambda, z_0)$, wobei gilt:

- X ist eine nichtleere, endliche Menge, das **Eingabealphabet**
- Y ist eine nichtleere, endliche Menge, das **Ausgabealphabet**
- Z ist eine nichtleere, endliche Menge, die **Zustandsmenge**
- $\delta: X \times Z \rightarrow Z$ ist die **Überföhrungsfunktion**, welche jedem Paar (Eingabezeichen, Zustand) einen **Folgezustand** zuordnet
- $\lambda: X \times Z \rightarrow Y^*$ ist die **Ausgabefunktion**, welche jedem Paar (Eingabezeichen, Zustand) ein **Ausgabewort** zuordnet.
- z_0 ist der **Anfangszustand**

FLACI

DIE MODULE

Formale Sprachen



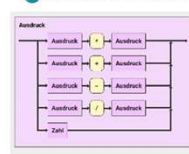
Interaktives Minitutorial zu den Grundbegriffen formaler Sprachen

Reguläre Ausdrücke



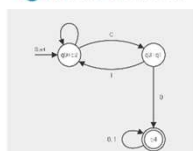
Interaktives Minitutorial zu regulären Ausdrücken

Formale Grammatiken



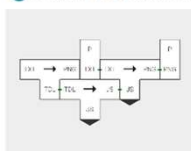
Kontextfreie Grammatiken entwickeln, transformieren und konvertieren

Abstrakte Automaten




Abstrakte Automaten konstruieren, simulieren, transformieren und konvertieren

Compiler und Interpreter



Modellieren von Übersetzungsprozessen und Entwicklung von Compilern und Interpretern

Universität
Rostock



Traditio et Innovatio

Offline-Version
im Abitur!

Download-Bereich:

FLACI Desktop Apps

Mit den offline-Versionen von FLACI können Sie auch ohne Internet arbeiten.


Windows 64-Bit

Windows 32-Bit

Linux dmg


MAC Silicon (M1)

MAC Intel



28.02.2024
© 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck
21

Universität
Rostock



Traditio et Innovatio

Zustandsdiagramme

Aufgaben mit FLACI

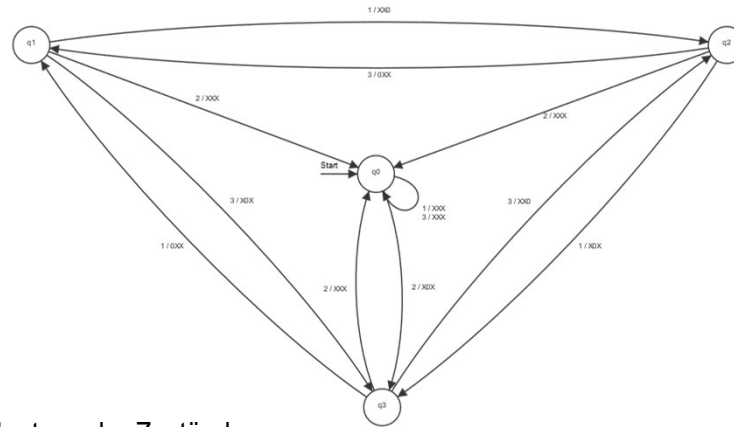
1. Beispiel zum Nachbauen oder zum Interpretieren:
Türen öffnen durch die drei vorgegebenen Schalter.
 Beschreiben Sie das Verhalten der Türen im System 3 mithilfe des Zustandsdiagramms (aus FLACI).

2. Beispiel:
Kunstaautomat
 Beschreiben Sie das Verhalten des Kunstaautomaten mit einem Zustandsdiagramm.
 Verwenden Sie dazu FLACI.

28.02.2024
© 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck
22

Verhaltensbeschreibung der Türen mit FLACI

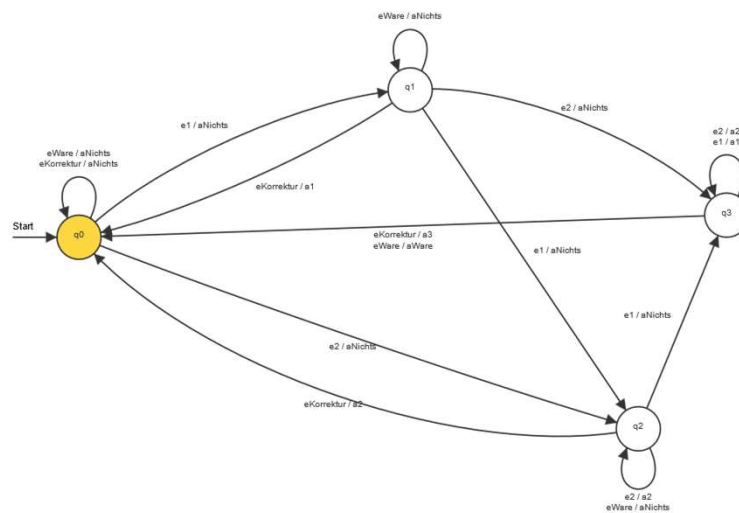
Lösung bei notwendigen Bedingungen wie im System 3



Bedeutung der Zustände:

q_0 : alle Türen zu; q_1 : linke Tür auf; q_2 : rechte Tür auf; q_3 : mittlere Tür auf

Zustandsdiagramm Kunstautomat mit FLACI



3. Kurze Einführung in Formale Sprachen

- Wie können formale Sprachen in der Informatik dargestellt werden?
- Interaktives Minitutorial zu den Grundbegriffen formaler Sprachen → <https://flaci.com/languages>
- Was ist eine von einem Automaten akzeptierte Sprache?
- Theorie: Was ist ein Akzeptor?

Akzeptoren


Ein Endlicher Automat ohne Ausgabe ist ein 5-Tupel

$$A = (X, Z, \delta, E, z_0),$$

wobei gilt:

- X ist eine nichtleere, endliche Menge, das **Eingabealphabet**
- Z ist eine nichtleere, endliche Menge, die **Zustandsmenge**
- $\delta: X \times Z \rightarrow Z$ ist die **Überföhrungsfunktion**, welche jedem Paar (Eingabezeichen, Zustand) einen **Folgezustand** zuordnet
- $E \subseteq Z \rightarrow$ ist die **Menge der Endzustände**,
- z_0 ist der **Anfangszustand**

Universität
Rostock



Traditio et Innovatio

4. Beispiele für Endliche Automaten

Aufgaben 3 - 5

Lösung Aufgabe 3

Lösung Aufgabe 4

Lösung Aufgabe 5

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

27

Universität
Rostock



Traditio et Innovatio

5. Grenzen von endlichen Automaten

Warum reichen DEAs nicht aus?

Beispiel → z. B. anhand der Sprache $L = \{a^n b^n \mid n \in \mathbb{N}\}$

Lösung durch SuS: Erweiterung des Automatenmodells durch die Möglichkeit des Speicherns

→ **Kellerautomat**

→ **Nur im Lk!**

→ Aufgabe 6

Lösung Aufgabe 6

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

28

Grenzen des Kellerautomaten

Beispiel → z. B. anhand der Sprache $L = \{0^n 1^n 2^n \mid n \in \mathbb{N}\}$
 → wegen des eingeschränkten Speicherzugriffs stößt auch der KA
 an Grenzen
 → Turingmaschine

6. Leben des Alan Turing

1912 - 1954



https://upload.wikimedia.org/wikipedia/commons/7/79/Alan_Turing_az_1930-es_%2C3%25Abveken.jpg

Film aus der SWR-Mediathek: Geniale Mathematiker (26 Minuten)

→ <https://www.swr.de/swr2/wissen/alan-turing-geniale-mathematiker-100.html>

→ Film: The Imitation Game (2014)

→ Film: Enigma (2001)

→ Lk: evtl. Referat zu Alan Turing



Berechenbarkeit

Algorithmusbegriff

Gibt es überhaupt einen Algorithmus, der dieses Problem löst ?

Verneinung einer solchen Frage erfordert, den Begriff Algorithmus *mathematisch exakt* zu definieren.

→ verschiedene Explikate des Begriffs Algorithmus:

Turing - Maschinen, Typ 0 - Grammatiken, rekursive Funktionen,
while - Programme.

Es zeigt sich:

Diese Begriffe sind in dem Sinne äquivalent, dass jedes Problem, das mit einem dieser Mechanismen gelöst werden kann, auch mit jedem anderen dieser Mechanismen gelöst werden kann.



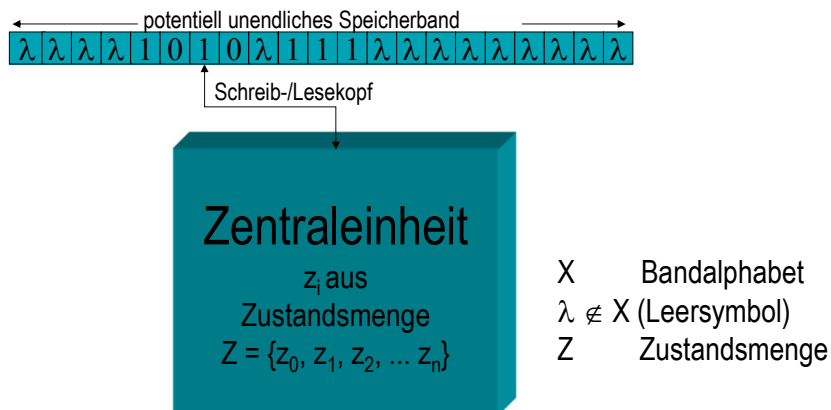
Church-Turing-These

Turing-Berechenbarkeit

Die Klasse der Turing-berechenbaren Funktionen stimmt mit der Klasse der intuitiv berechenbaren Funktionen überein.

Turing-Maschine – erstes theoretisches Maschinenmodell

Konzept wurde 1936 von Alan Turing entwickelt



Turingmaschine

Formale Definition

Eine (zweiseitige deterministische) Turingmaschine

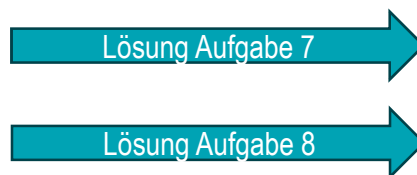
$T = (X, Z, \Gamma, \lambda, z_A, f, Z_E)$ ist bestimmt durch

- eine endliche Menge von Eingabezeichen X
- eine endliche Menge von Zuständen Z
- eine endliche Menge von Bandsymbolen Γ
Dabei enthält Γ das Leerzeichen λ und es ist $X \subseteq \Gamma \setminus \{\lambda\}$.
- einen Anfangszustand $z_A \in Z$
- eine i.a. nur partiell definierte Übergangsfunktion f :
 $f: Z \times \Gamma \rightarrow Z \times \Gamma \times \{L, R, K\}$
- eine Menge Z_E von Endzuständen; $Z_E \subseteq Z$



6. Beispiel Turingmaschine

Aufgaben 7 - 8



7. Halteproblem

Algorithmisch unlösbare Probleme

(1) Halteproblem:

Gibt es ein Programm, mit dessen Hilfe man für jedes Programm entscheiden kann, ob es mit jeder beliebigen Eingabe nach endlich vielen Schritten stoppt oder nicht?

Ersatzproblem:

(2) Selbstanwendbarkeitsproblem:

Gibt es ein Programm, das von jedem Programm, das auf sich selbst angewendet wird, entscheidet, ob es nach endlich vielen Schritten stoppt oder nicht?

Da (2) ein Spezialfall von (1) ist, gilt:

Wenn es kein Programm für (2) gibt, dann gibt es kein Programm für (1)



Indirekter Beweis der algorithmischen Unlösbarkeit des Halteproblems

Definition: Ein Programm heiße **selbststoppend**, wenn es bei Anwendung auf sich selbst nach endlich vielen Schritten stoppt.



Annahme:

Es existiert ein Python-Programm, das von jedem Python-Programm entscheiden kann, ob es selbststoppend ist oder nicht.

PROGRAMM Stopptest

```
# Funktionsdefinitionen
def stopp( Quelltext ):
    ...
    return ...

if stopp == TRUE:
    print „Das Programm ist selbststoppend.“
else:
    print „Das Programm ist nicht selbststoppend.“
# ENDE
```

testet in diesem Bereich, ob das eingegebene Programm selbststoppend ist.
Wenn ja, dann stopp := TRUE, sonst stopp := FALSE.

Annahme zum Widerspruch führen

PROGRAMM **Seltsam**

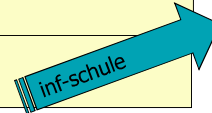
```
# Funktionsdefinitionen
...
def stopp:
    ...
    return ...
...

if stopp == TRUE:
    print „Das Programm ist selbststoppend.“
    while TRUE pass

print „Das Programm ist nicht selbststoppend.“
# ENDE
```

In diesem Bereich völlig identisch mit dem Programm Stopptest; testet wieder, ob das eingegebene Programm selbststoppend ist. Wenn ja, dann stopp := TRUE, sonst stopp := FALSE.

Welche Eigenschaft hat das Programm *Seltsam*?




1. Fall: Angenommen, *Seltsam* ist selbststoppend.
Variable stopp wird TRUE gesetzt; Ausgabe: Das Programm ist selbststoppend; dann Endlosschleife; Programm stoppt nicht
2. Fall: Angenommen, *Seltsam* ist nicht selbststoppend.
Variable stopp wird FALSE gesetzt; Ausgabe: Das Programm ist nicht selbststoppend; Programm stoppt.

Widerspruch!!

Widerspruch!!

**Universität
Rostock**



Traditio et Innovatio

Prüfungsaufgaben mit FLACI?

Diskussion

→ Abituraufgaben aus MV...

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

39

**Universität
Rostock**



Traditio et Innovatio

Lehrmittel

Literatur:

- Schöningh: Informatik 2 → Kapitel 4
- Oldenbourg: Informatik Oberstufe 2 → Kapitel 1, 4
- Klett: Informatik 5 → Kapitel 1, 5


Inf-Schule: Sprachen und Automaten

Inf-Schule: Halteproblem und Turingmaschine

Software

AtoCC

Flaci



28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

40

Universität
Rostock



Traditio et Innovatio


Vielen Dank für Ihre Aufmerksamkeit!



28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

41

Universität
Rostock



Traditio et Innovatio

Verwendete Materialien/Quellen:

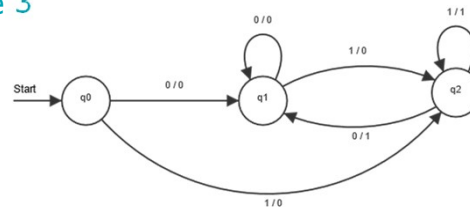
- <https://www.inf-schule.de>
- <https://www.swr.de/swr2/wissen/alan-turing-und-die-moeglichkeiten-der-maschine-100.html>
- "Theoretische Grundlagen der Informatik" von Rolf Socher, Fachbuchverlag Leipzig 2003
- "Einführung in die Theoretische Informatik" von U. Hedtstück, Oldenbourg-Verlag München 2000
- "Theoretische Informatik - kurzgefasst" von Uwe Schöning, Spektrum Akademischer Verlag Heidelberg 1999
- Schullehrbücher (vgl. Folie 40)

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

42

Lösung Aufgabe 3

Mealy-Automat



Ablaufprotokolle:

01011 \vdash^* 00101

11111 \vdash^* 01111

1010 \vdash^* 0101

Eingabe: 11 → Ausgabe: 5

Eingabe: 31 → Ausgabe: 15

Eingabe: 10 → Ausgabe: 5

Der Automat schiebt alle Ziffern um 1 Stelle nach rechts,
die letzte Ziffer entfällt
→ entspricht bei Dualzahlen der Ganzzahldivision durch 2.



Lösung Aufgabe 4

Akzeptoren

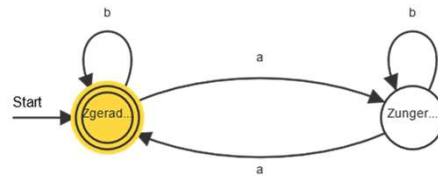
a) $L(A) = \{aax^* \mid x \in \{a,b\}^*\}$

b) $L(A) = \{w \mid w \text{ ist ein Wort aus } a\text{'s und } b\text{'s,}\br/>
\text{wobei die Anzahl jeweils gerade ist}\}$
 insbesondere $\varepsilon \in L(A)$



Lösung Aufgabe 5

Akzeptor



$$A = (X = \{a, b\}, Z = \{z_{\text{gerade}}, z_{\text{ungerade}}\}, z_{\text{gerade}}, \{z_{\text{gerade}}\}, \delta)$$

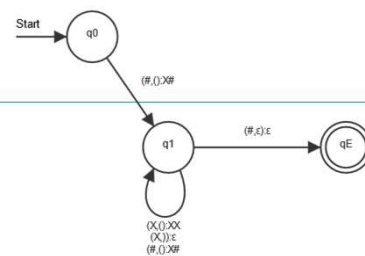
mit δ :

	a	b
z_{gerade}	z_{ungerade}	z_{gerade}
z_{ungerade}	z_{gerade}	z_{ungerade}



Lösung Aufgabe 6:


Kellerautomat



$$\mathbf{KA} = (\{ (,) \}, \{ \#, X \}, \{ q_0, q_1, q_E \}, q_0, \delta, \#, \{ q_E \}) \text{ mit } \delta$$

Zustand	eingelese- nes Zeichen	Keller- zeichen	→	neuer Zustand	Kelleraktion
q ₀	(#		q ₁	push X → X#
q ₁	(X		q ₁	push X → XX
q ₁	(#		q ₁	push X → X#
q ₁)	X		q ₁	pop → ε
q ₁	ε	#		q _E	nop → ε



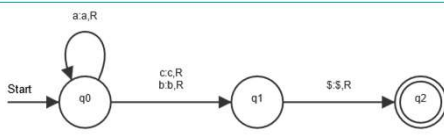


Universität
Rostock

Traditio et Innovatio


Lösung Aufgabe 7

Turingmaschine



$TM = (X = \{a, b, c\}, Z = \{q_0, q_1, q_2\}, \Gamma = \{a, b, c, \$\}, q_0, f, \$, Z_E = \{q_2\})$
 mit f:

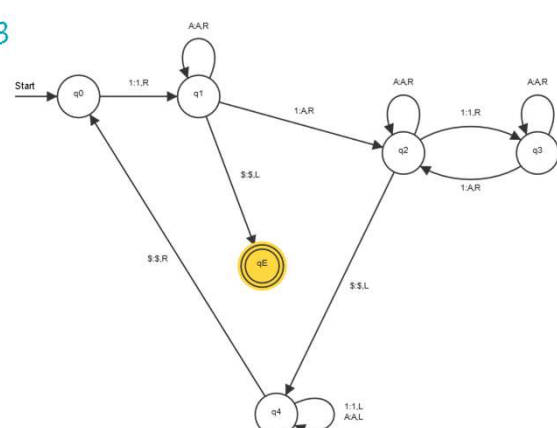
	a	b	c	\$
q ₀	q ₀ , a, R	q ₁ , b, R	q ₁ , c, R	
q ₁				q ₂ , \$, R
q ₂				



28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck
47

Lösung Aufgabe 8


Turingmaschine




a)

b) $q_A 1111 \vdash 1q_1 111 \vdash 1Aq_2 11 \vdash 1A1q_3 1 \vdash 1A1Aq_2 \$ \vdash 1A1q_4 A \vdash 1Aq_4 1A \vdash 1q_4 A1A \vdash q_4 1A1A \vdash q_4 \$1A1A \vdash q_0 1A1A \vdash 1q_1 A1A \vdash 1Aq_1 1A \vdash 1AAq_2 A \vdash 1AAAq_2 \$ \vdash 1AAq_4 A \vdash 1Aq_4 AA \vdash 1q_4 AAA \vdash q_4 1AAA \vdash q_4 \$1AAA \vdash q_0 1AAA \vdash 1q_1 AAA \vdash 1Aq_1 AA \vdash 1AAq_1 A \vdash 1AAq_1 \$ \vdash 1AAq_E A$

c) $L(T) = \{ 1^n \mid n = 2^k, k \in \mathbb{N} \}$





Universität
Rostock

Traditio et Innovatio

28.02.2024 © 2024 UNIVERSITÄT ROSTOCK | ILTiS 2024 | Ulrike Heiduck

48