Konzepte der theoretischen Informatik

(GK 14 Stunden, LK 30 Stunden)

Tino Hempel / Lutz Hellmig

Mecklenburg Vorpommern
Institut für Qualitätsentwicklung

Informatische Bildung in Mecklenburg-Vorpommern

Universität Rostock
Traditio et Innovatio

Formale Sprachen und Automaten (integrativ)

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Endliche Automaten • mithilfe ihrer Definition interpretieren • eine Simulation zum Analysieren, Visualisieren und Implementieren nutzen • Aufbau und Arbeitsweise anhand eines anschaulichen Modells beschreiben Formale Sprache • 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	Die Überführungsfunktion und ggf. die Ausgabefunktion können sowohl in grafischer als auch tabellarischer Darstellung vorliegen.

Werkzeug-Tip 1: Die Kara-Familie

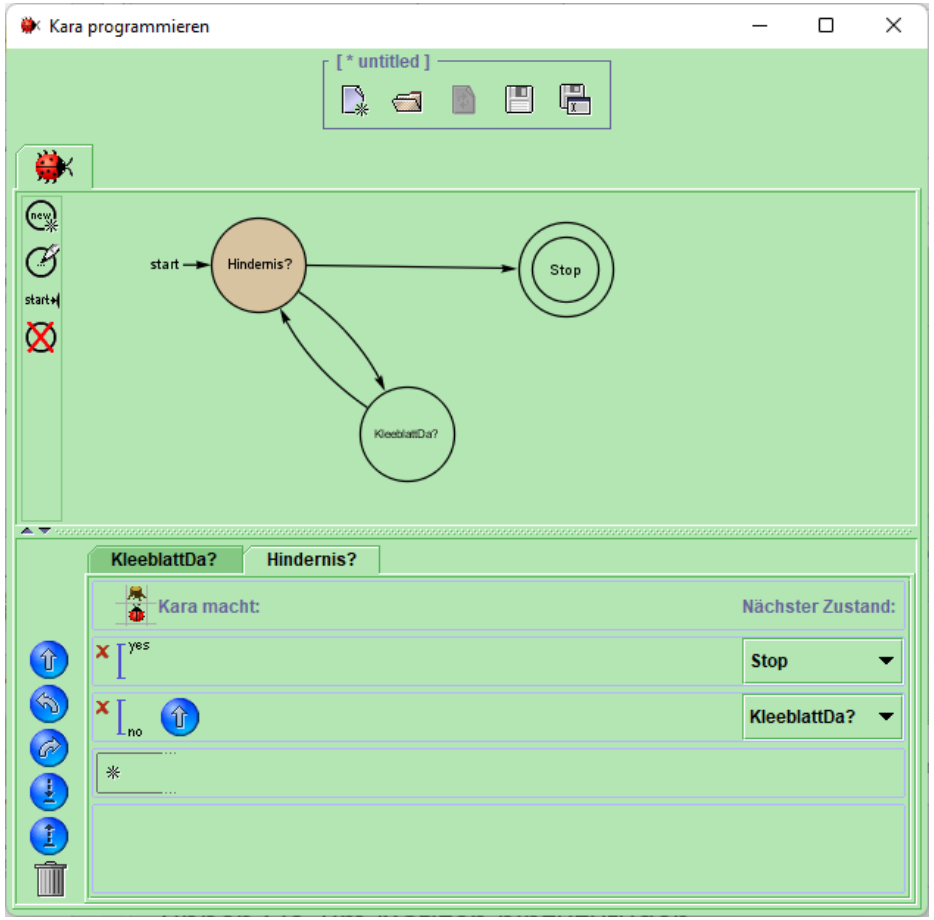
https://www.swisseduc.ch/informatik/karatojava/index.html

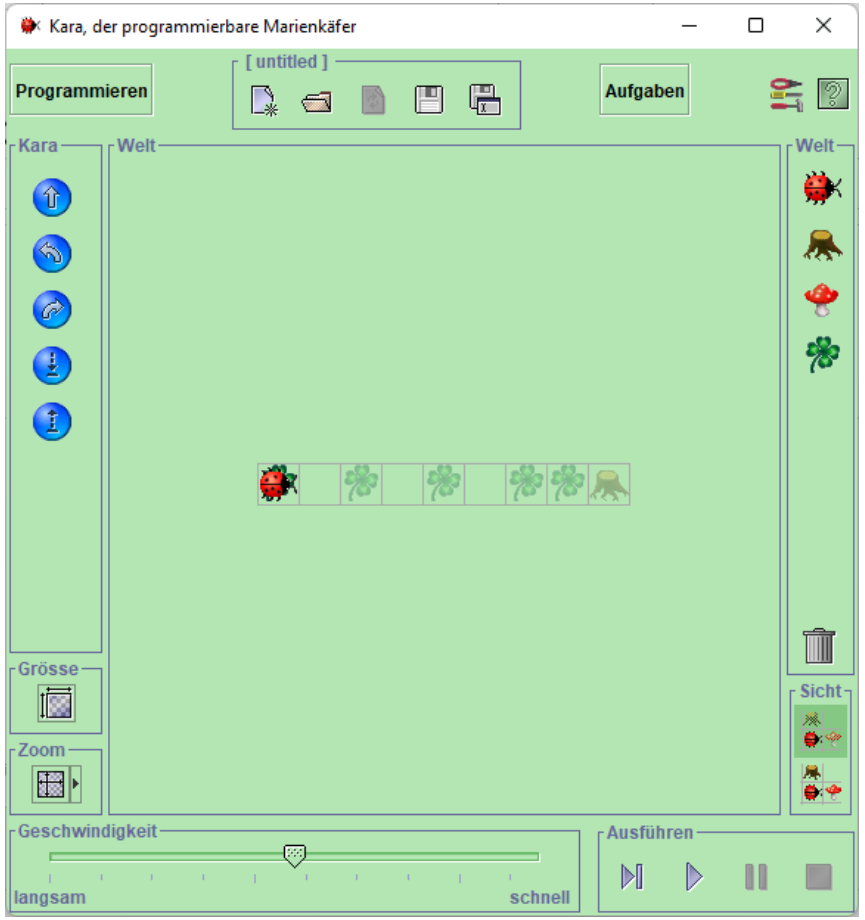
Programmieren mit endlichen Automaten

Turing-Kara

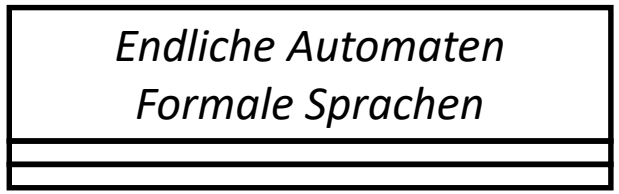
Wo steckt der Fehler?

1_Kleeblatt_sammeln.kara





Vom realen Automaten zur Turingmaschine



Unterscheidung endlicher Automaten

Transduktoren

Explizite Ausgabe

- Moore-Automat
- Mealy-Automat
- Turingmaschine
  (Betrachtung des Bandinhalts)

Akzeptoren

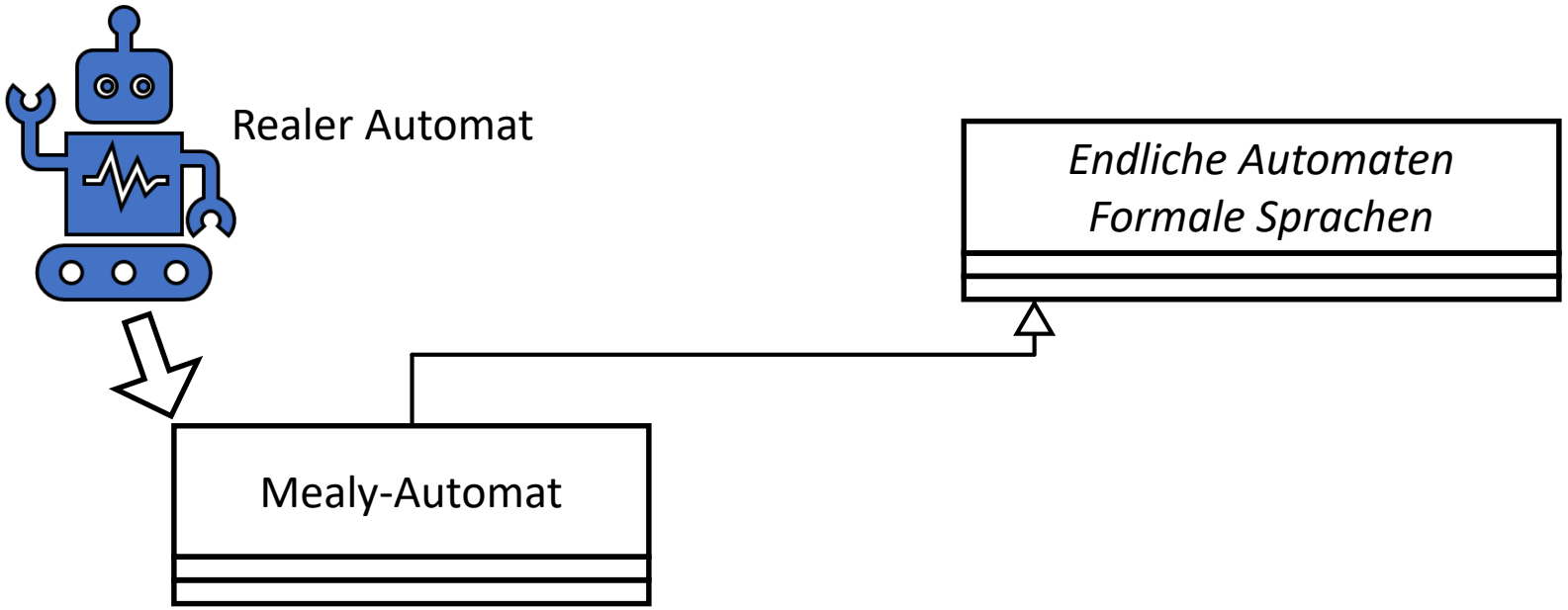
Implizite Ausgabe mit Endzustand

- Medwedew-Automat
- Kellerautomat
- Turingmaschine
  (Betrachtung des Endzustands)

Endliche Automaten (14/20 Stunden)

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Mealy-Automat MA = (X, Y, Z, δ, λ, z_0) • Überführungs- und Ausgabefunktion tabellarisch und grafisch darstellen • einen Mealy-Automaten modellieren	Eine Analyse realer Automaten bildet den Ausgangspunkt für die Abstraktion zum Mealy-Automaten.

Vom realen Automaten zur Turingmaschine



Beispiel: Der Kunstautomat

Münzeinwurf

Korrekturknopf

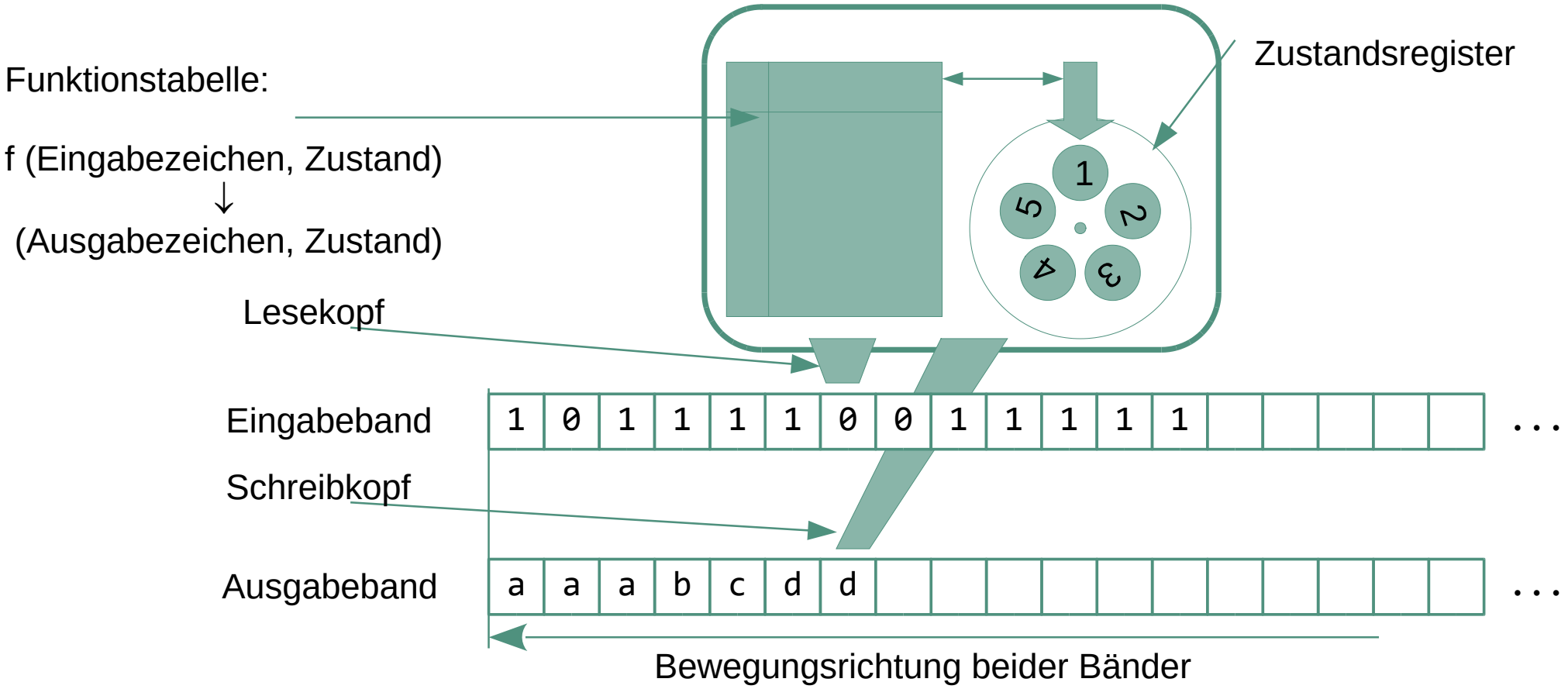
Warenknopf

Warenausgabe

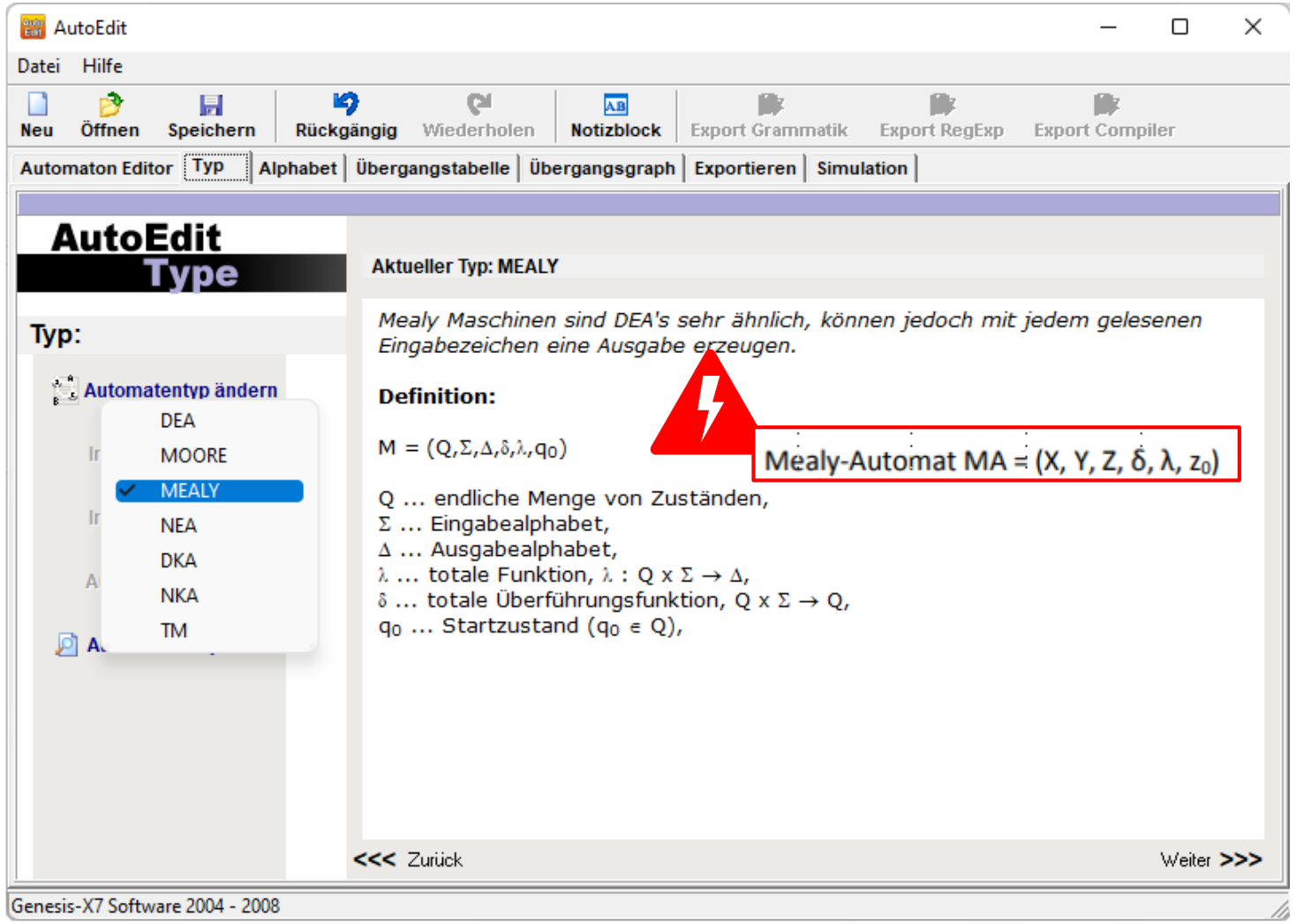
Münzrückgabe

https://www.inf-schule.de/automaten-sprachen/zustandsmodellierung/endlicheautomaten

Mealy-Automat: anschaulich

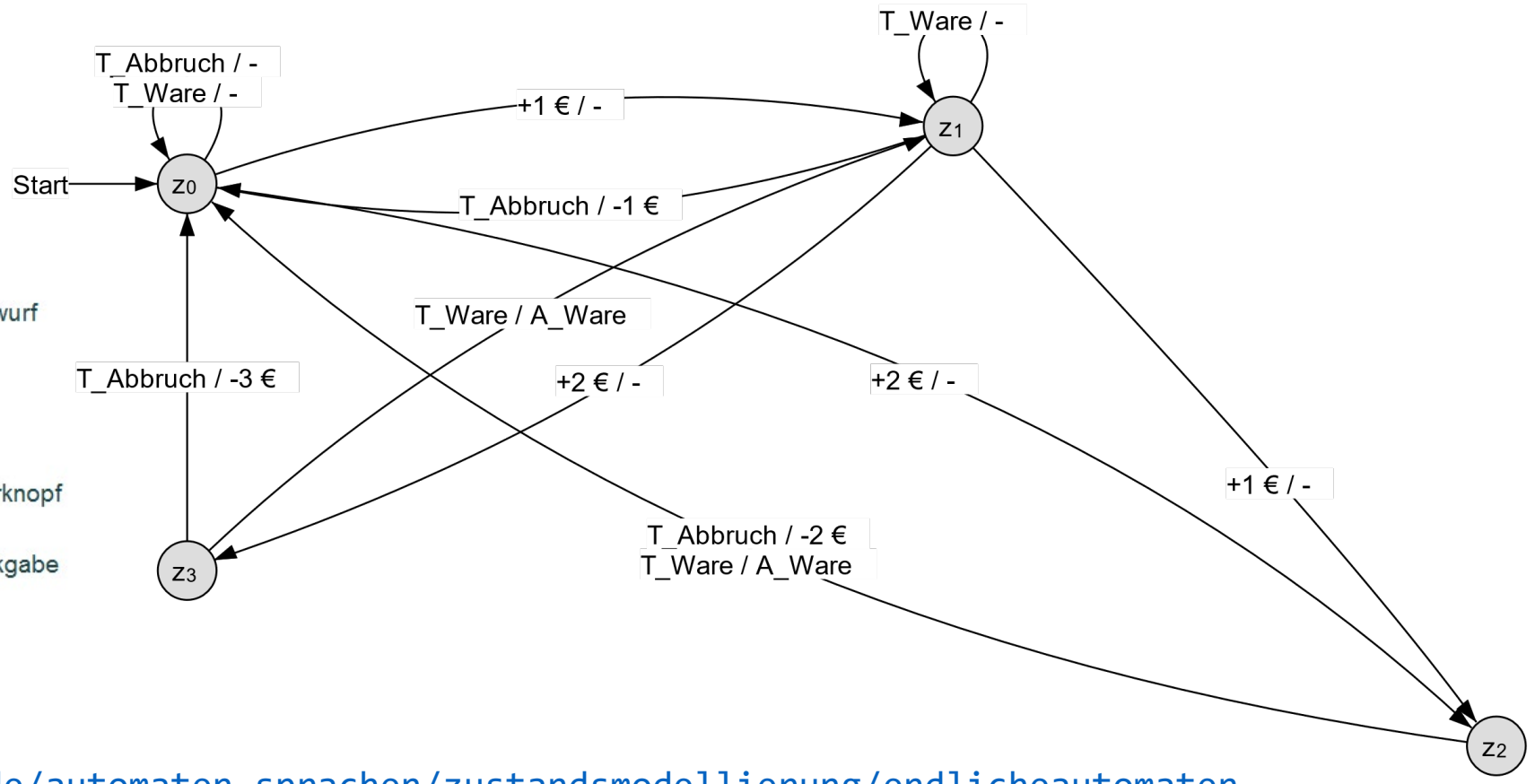


Werkzeug-Tip 2: AtoCC-Autoedit



Beispiel: Der Kunstautomat

2_Kunstautomat_vorgabe.xml

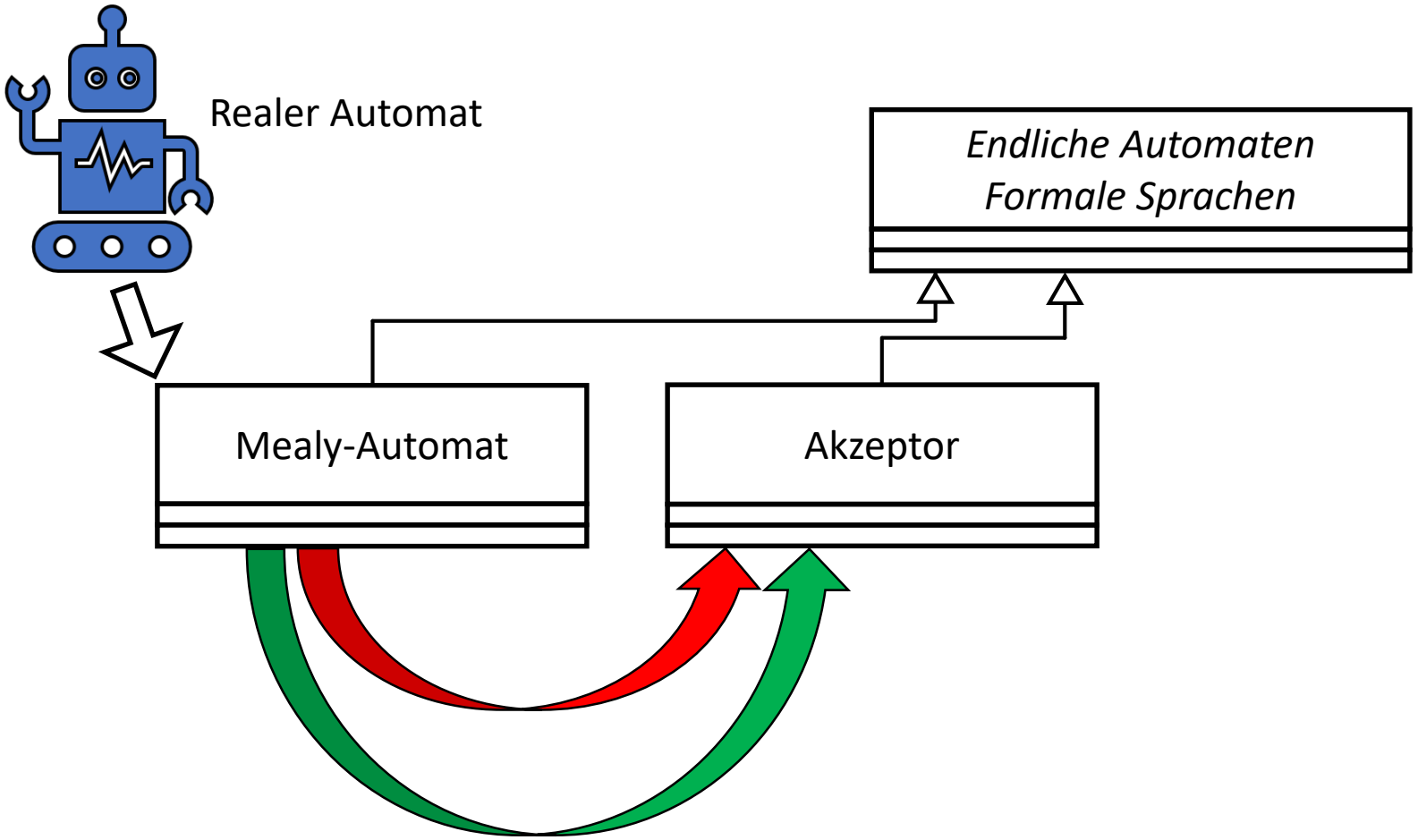


https://www.inf-schule.de/automaten-sprachen/zustandsmodellierung/endlicheautomaten

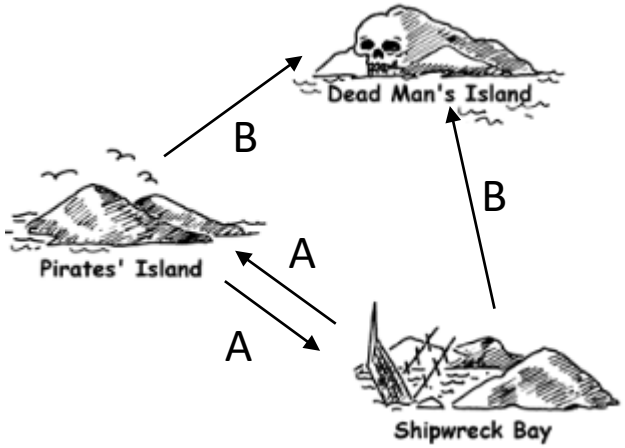
Endliche Automaten (14/20 Stunden)

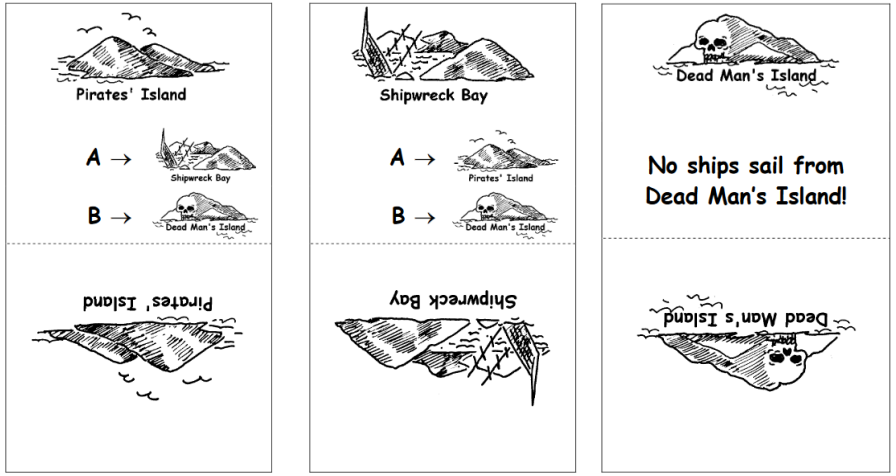
Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Mealy-Automat MA = (X, Y, Z, δ, λ, z_0) • Überführungs- und Ausgabefunktion tabellarisch und grafisch darstellen • einen Mealy-Automaten modellieren	Eine Analyse realer Automaten bildet den Ausgangspunkt für die Abstraktion zum Mealy-Automaten.
Akzeptor A = (X, Z, δ, z_0, Z_E) • Überführungsfunktion tabellarisch und grafisch darstellen • einen Akzeptor anhand einer gegebenen Sprache modellieren • eine Grenze von Akzeptoren erläutern	z. B. anhand der Sprache L(A) = $\{a^n b^n \mid n \in \mathbb{N}\}$

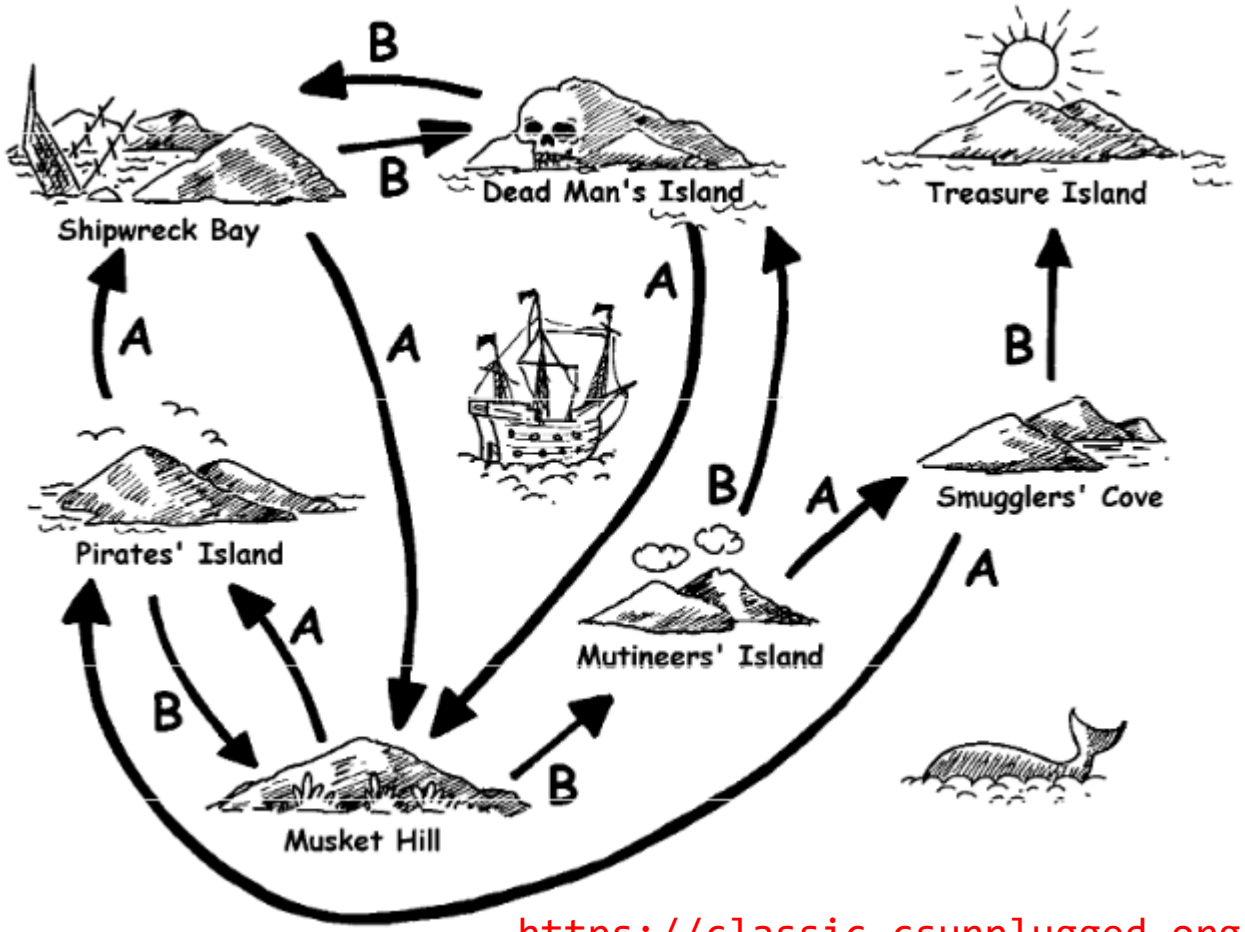
Vom realen Automaten zur Turingmaschine



CS unplugged – Treasure Island

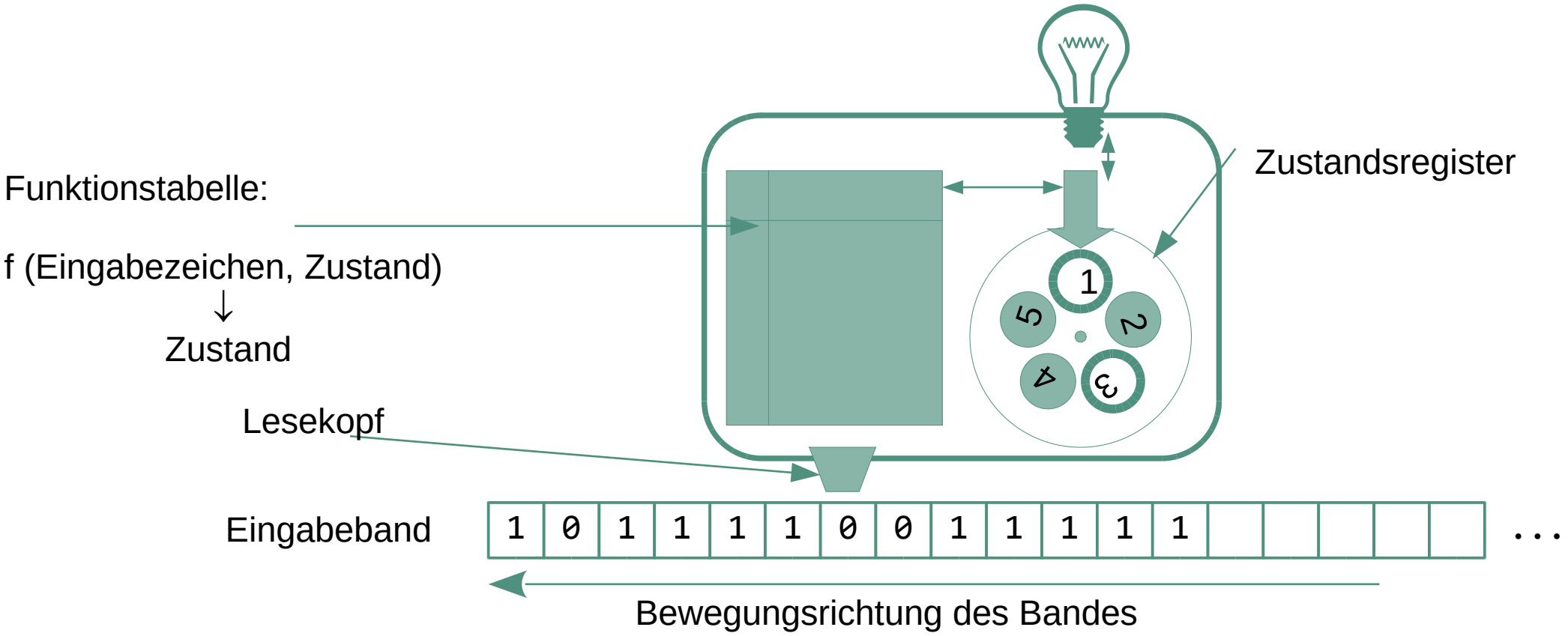






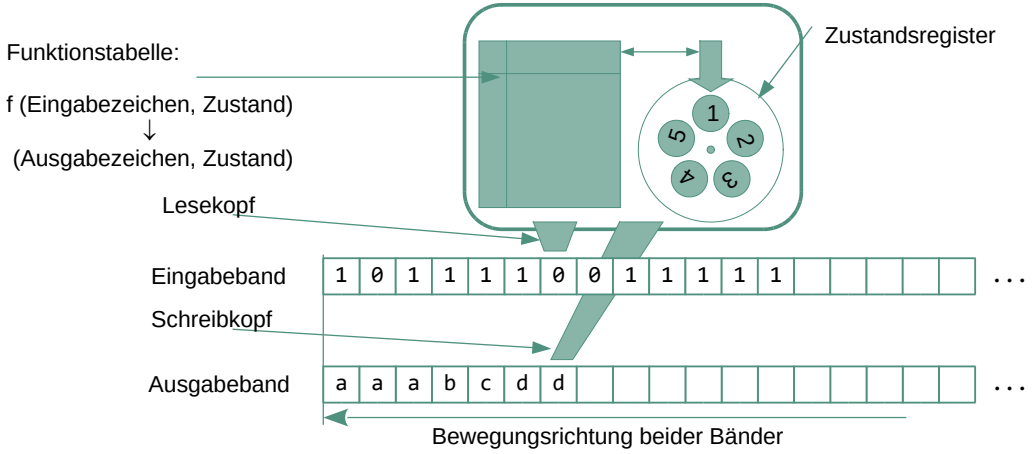
https://classic.csunplugged.org

Akzeptor: anschaulich

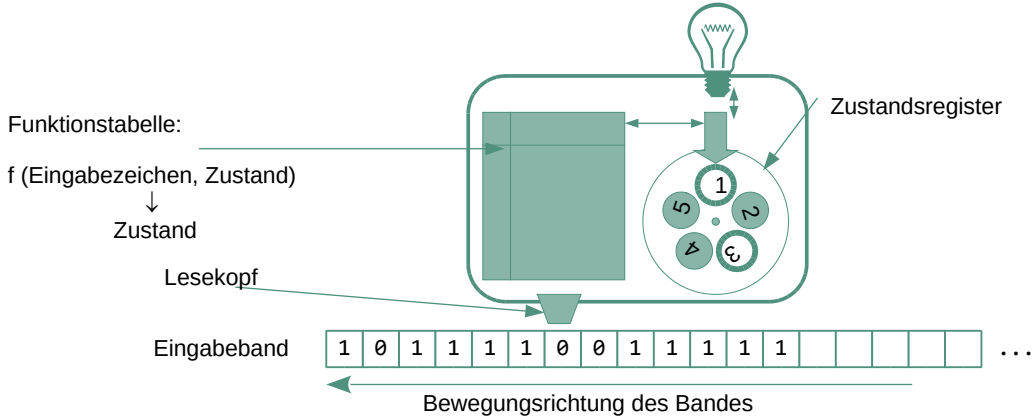


Transduktor und Akzeptor im Vergleich

Mealy-Automat

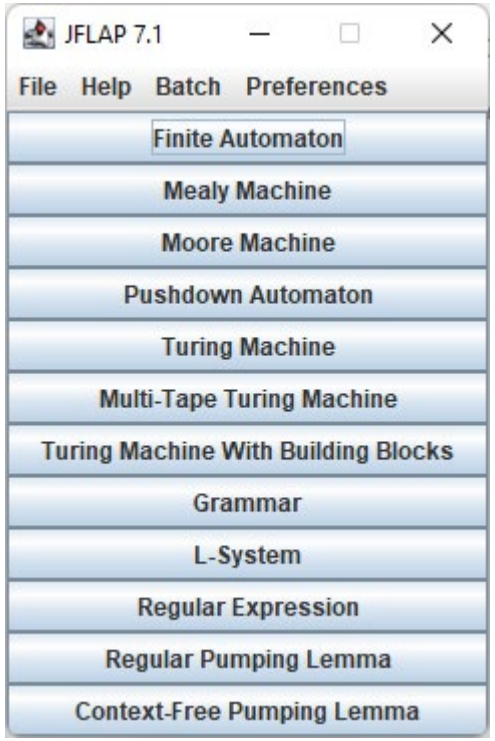


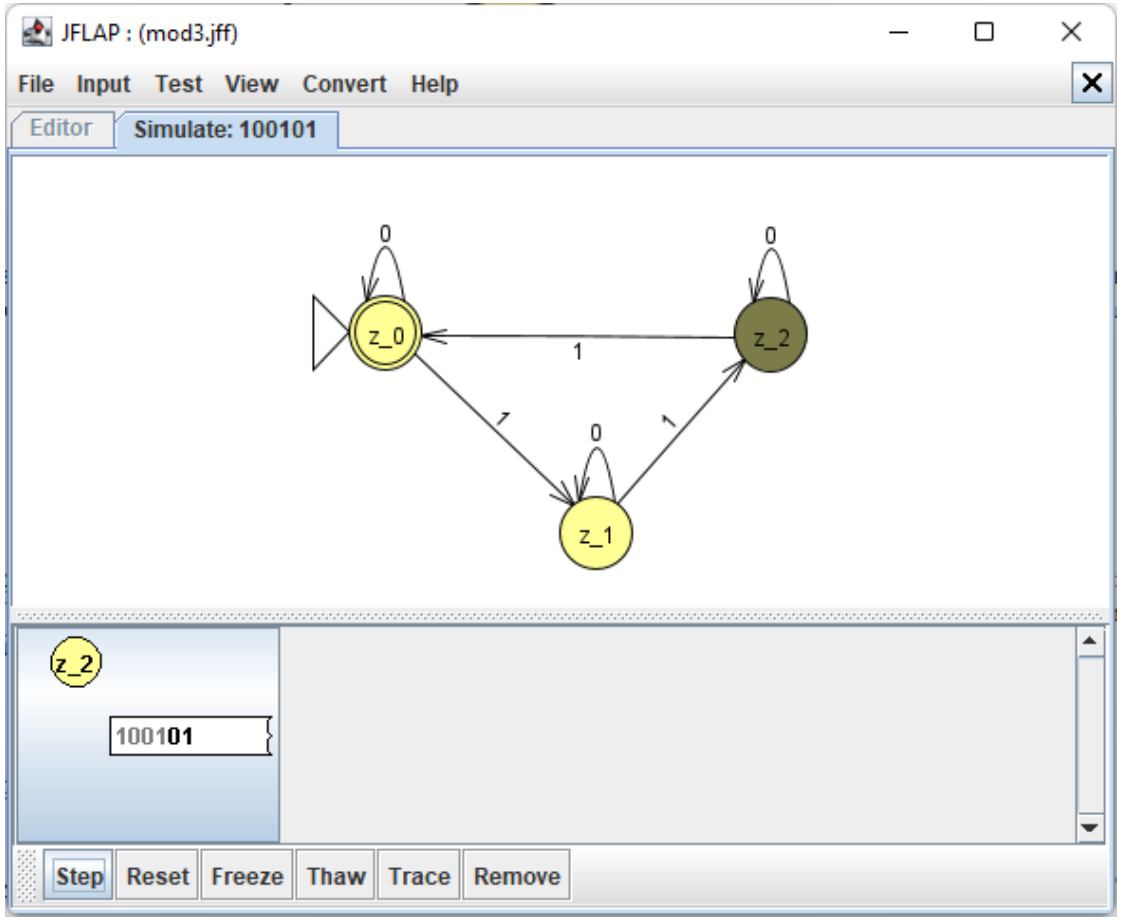
Akzeptor



Werkzeug-Tip 3: JFLAP

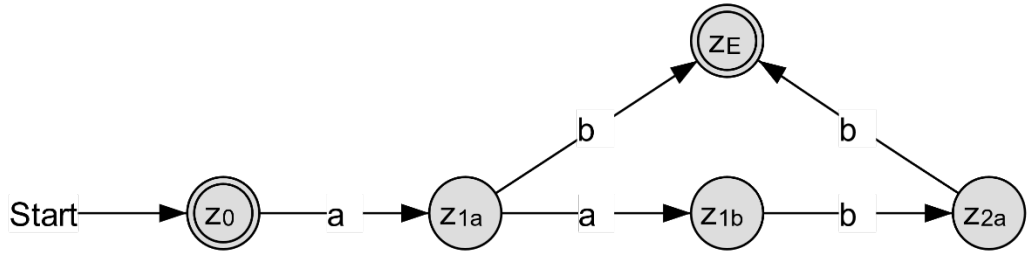
3_mod3.jff

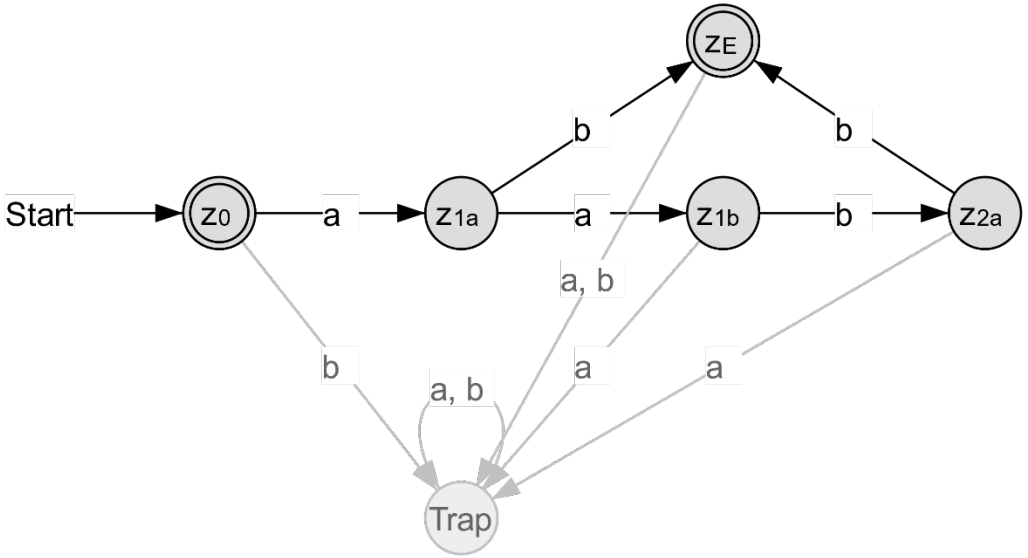




Grenzen anhand der Sprache
L(A) = $\{a^n b^n | n \in \mathbb{N}\}$

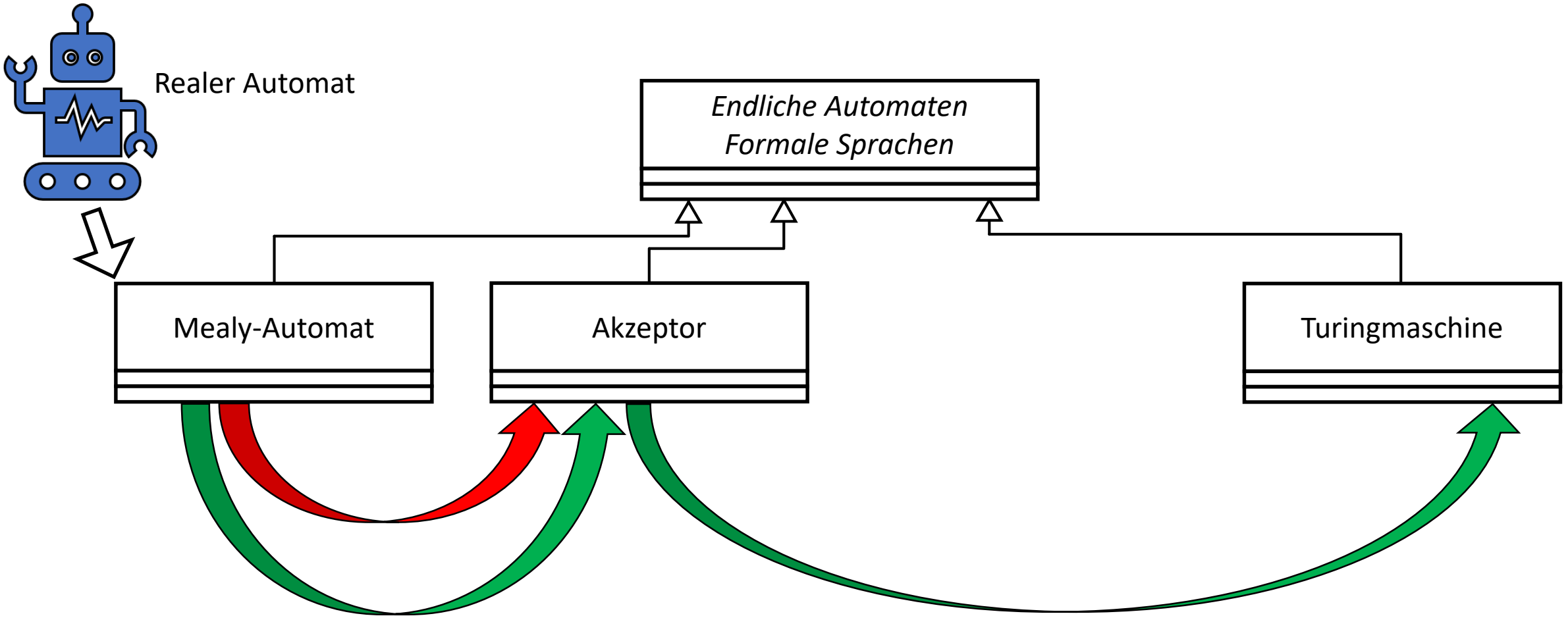
4_DEA_anbn_Vorgabe.xml



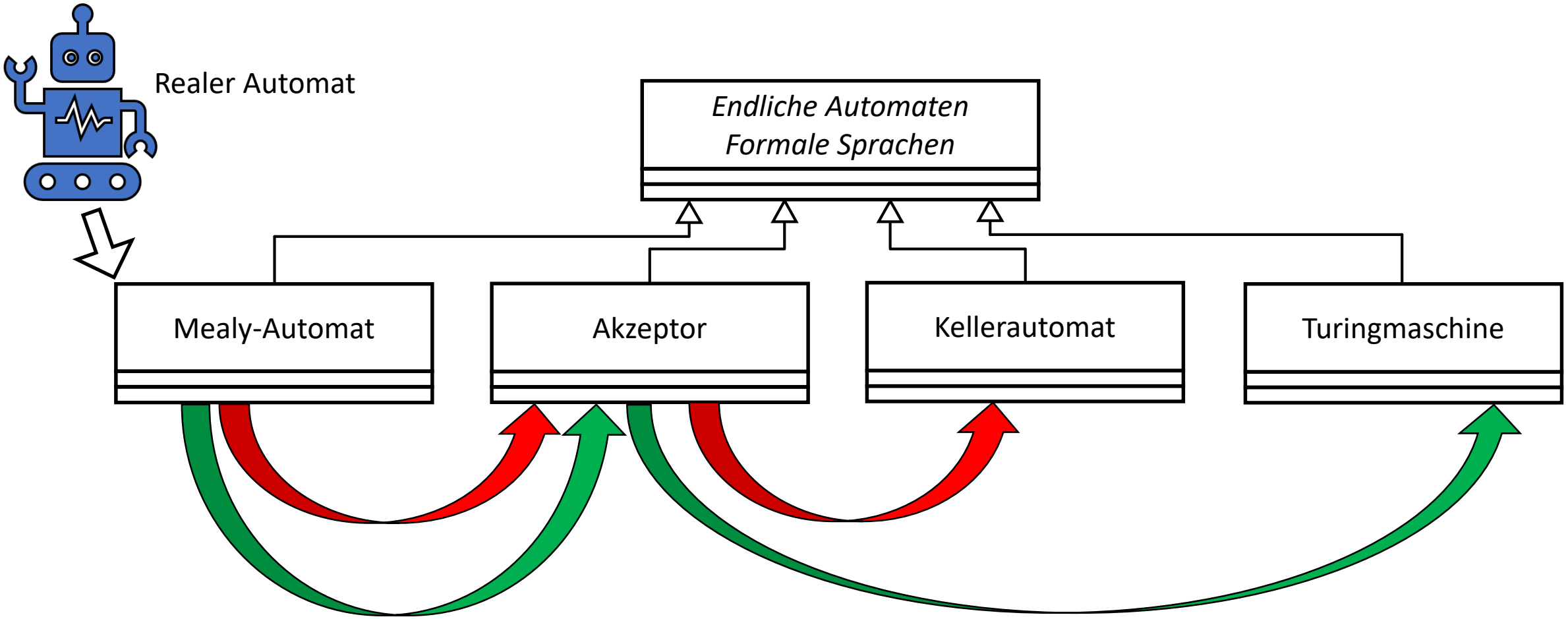


ab
aabb
aaabbb
aaaaaaaaaaaabbbbbbbbbbbb

Vom realen Automaten zur Turingmaschine



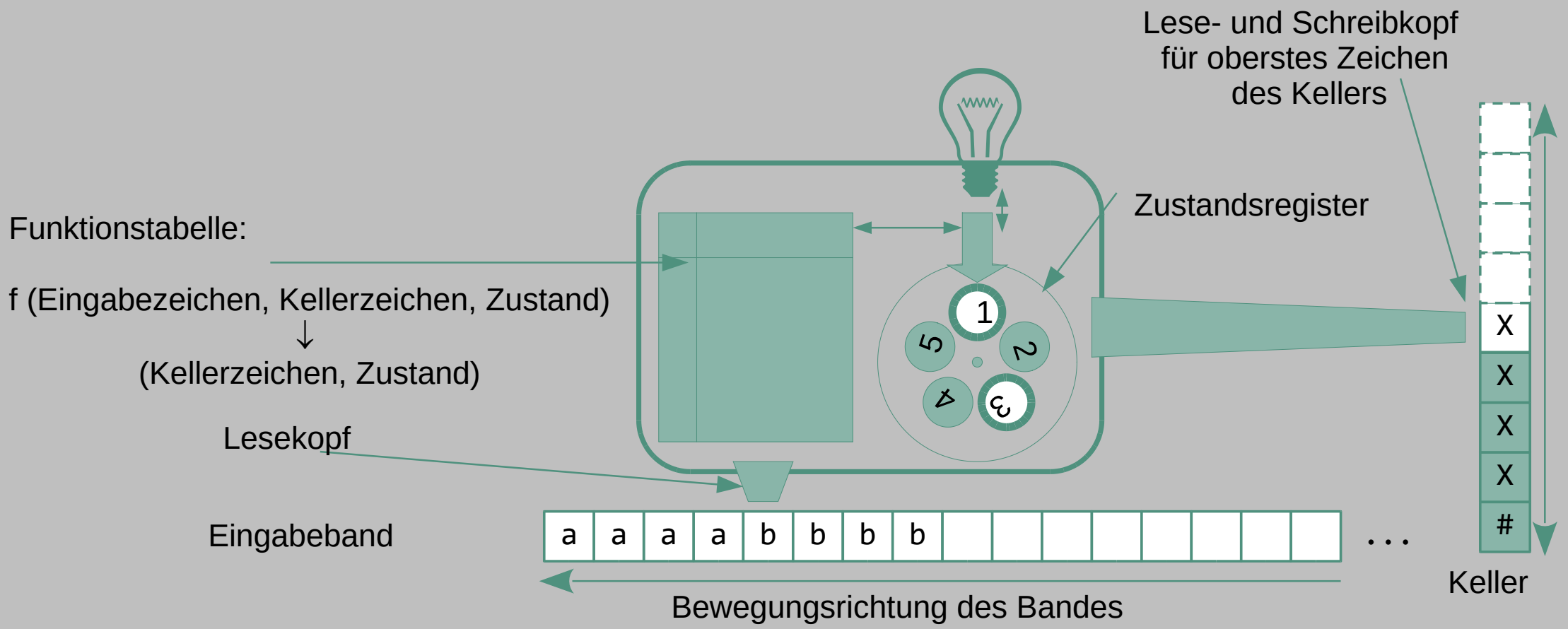
Vom realen Automaten zur Turingmaschine



Endliche Automaten

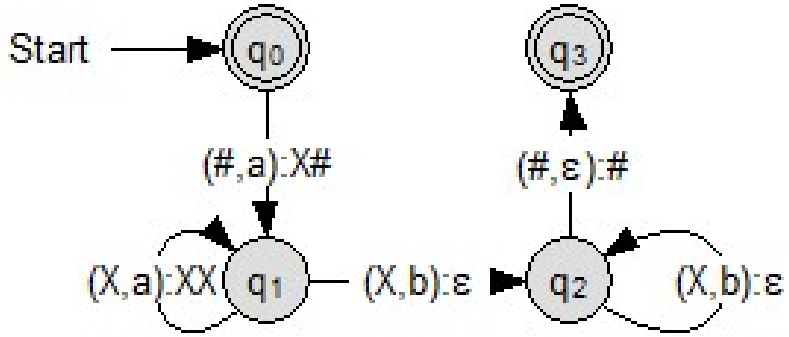
Verbindliche Ziele und Inhalte	Hinweise und Anregungen
zusätzlich für den Leistungskurs	
Kellerautomat KA = $(X, Z, \Gamma, \delta, z_0, k_0, Z_E)$ • das Prinzip eines Kellerspeichers und die Kelleroperationen push, pop und nop erläutern • einen Kellerautomaten anhand einer gegebenen Sprache modellieren • Prinzip des Nichtdeterminismus anhand der Erkennung von Palindromen erläutern • eine Grenze von Kellerautomaten erläutern	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 $L(A) = \{a^n b^n c^n \mid n \in \mathbb{N}\}$

Kellerautomat – anschaulich



Kellerautomat für $a^n b^n$ mit Autoedit

Transition Graph:



Definition: $M = (\{q_0,q_1,q_2,q_3\},\{a,b\},\{\#,X\},\delta,q_0,\#,\{q_0,q_3\})$

Transitions:

$\delta(q_0,a,\#) = (q_1,X\#)$
$\delta(q_1,a,X) = (q_1,XX)$
$\delta(q_1,b,X) = (q_2,\varepsilon)$
$\delta(q_2,b,X) = (q_2,\varepsilon)$
$\delta(q_2,\varepsilon,\#) = (q_3,\#)$

aaaabbbb ✔

aaaaabb ✘

aaabbbbb ?

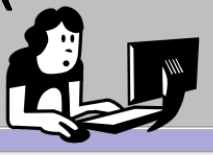
5_DKA_anbn.xml

Grenzen von Kellerautomaten

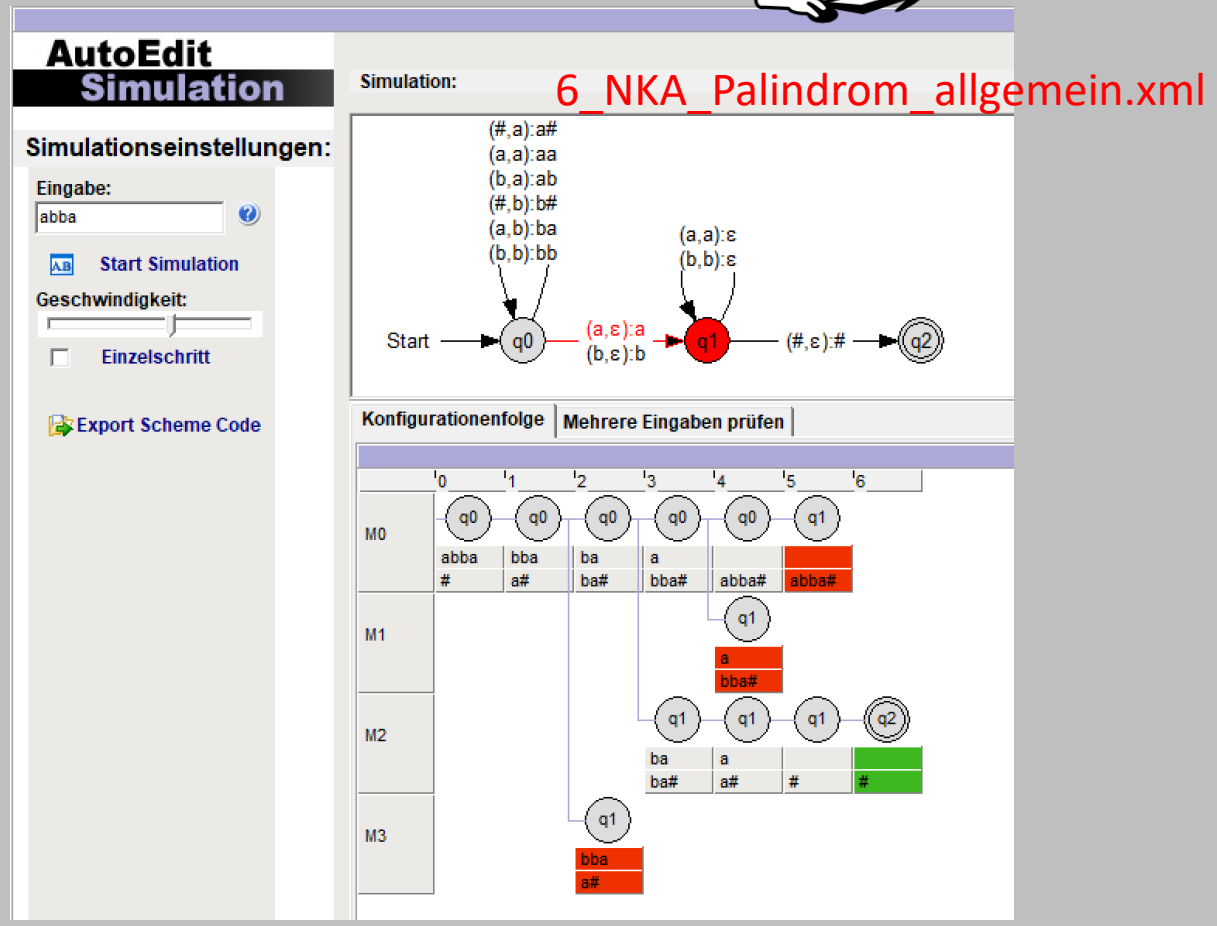
Untersuchung von Palindromen (z. B. SAIPPUAKUPPIN**IPPUKAUPPIAS)**

- trivial bei bekannter Mitte
- Ist die Mitte vorher nicht bekannt
  → nicht mit *deterministischen* Kellerautomaten lösbar

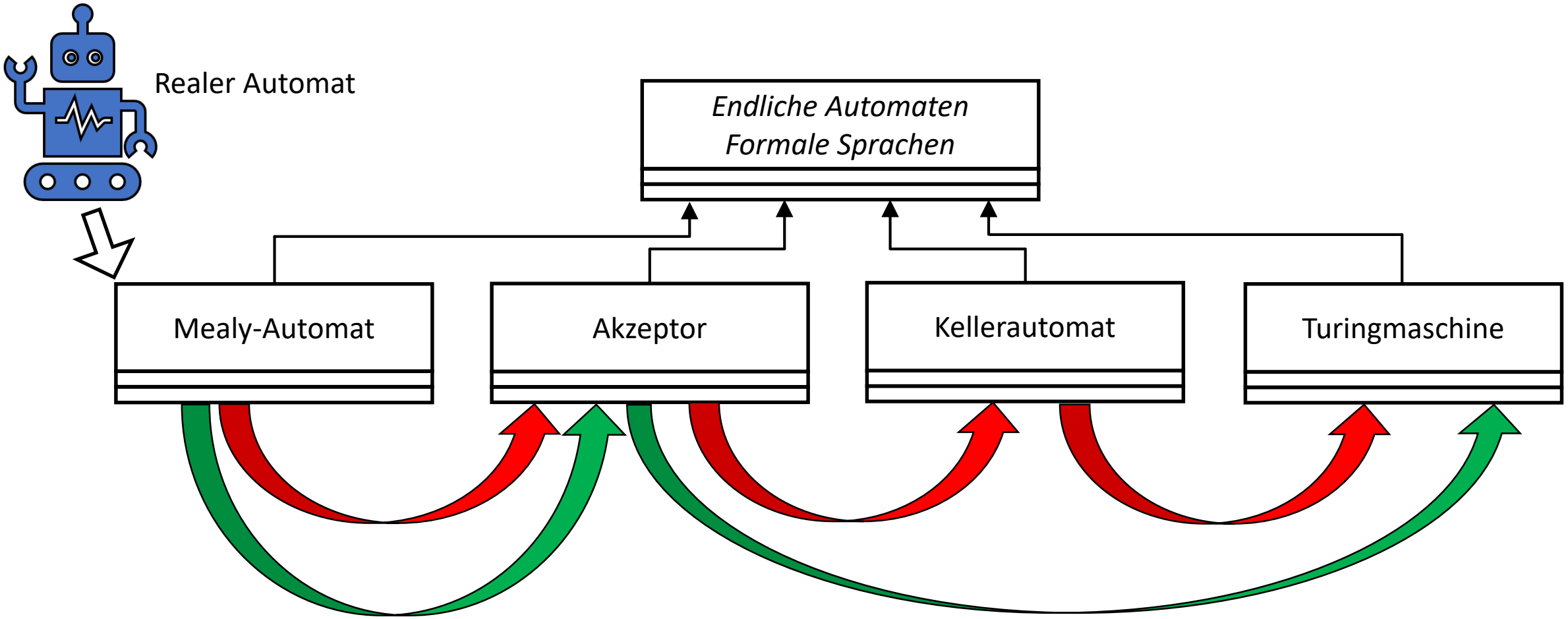
Nichtdeterministische Kellerautomaten (NKA)



- Mehrere Zustandsübergänge bei gleichen Voraussetzungen möglich → Nichtdeterminiertheit
- „Raten“ der richtigen Lösung
- Kultivierung der Brute-Force-Methode



Vom realen Automaten zur Turingmaschine



Endliche Automaten

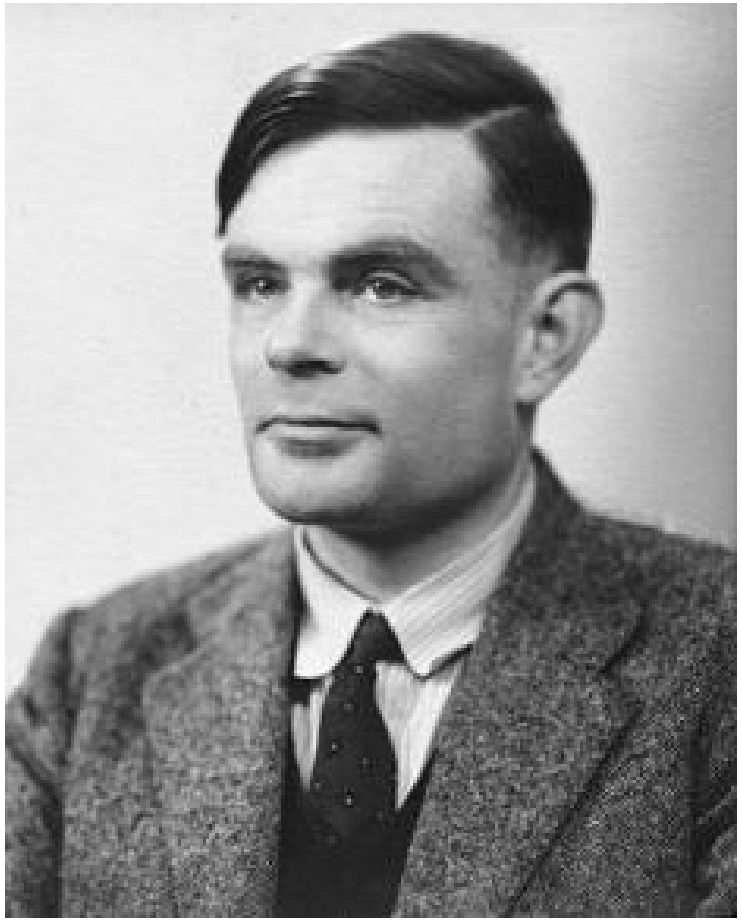
Verbindliche Ziele und Inhalte	Hinweise und Anregungen
Turingmaschine TM = (X, Z, Γ, δ, z_0, k_0, $, Z_E) • den Wert des Modells anhand der Church-Turing-These begründen • die Grenzen einer Turingmaschine anhand des Halteproblems erklären	Die Turingmaschine ist sowohl als Akzeptor als auch als rechnende Maschine mit Ausgabe zu betrachten.
Turingmaschine • eine Turingmaschine anhand einer gegebenen Sprache modellieren	

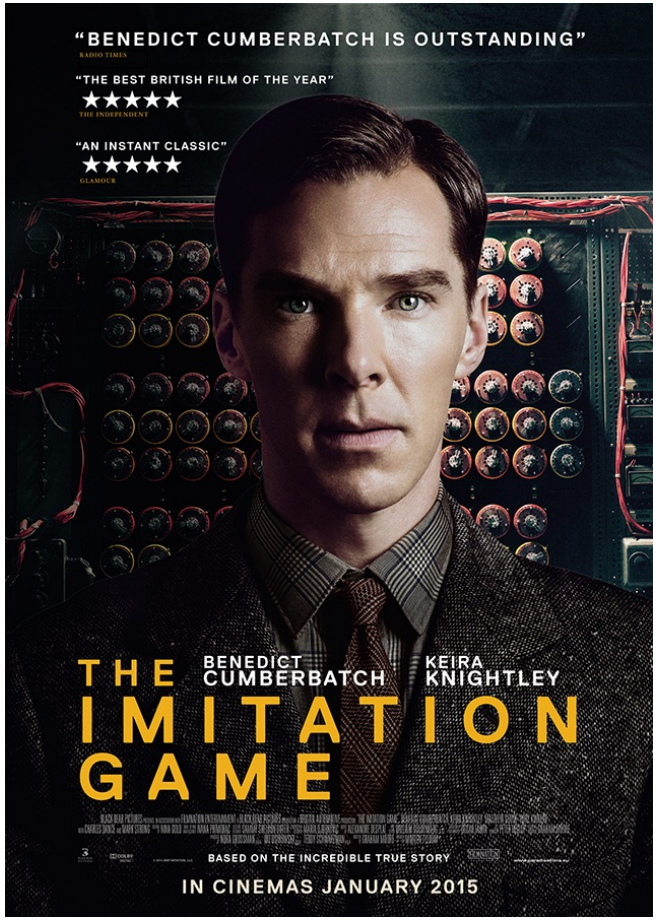
Alan Turing (1912-1954)

1936 Turing-Maschine

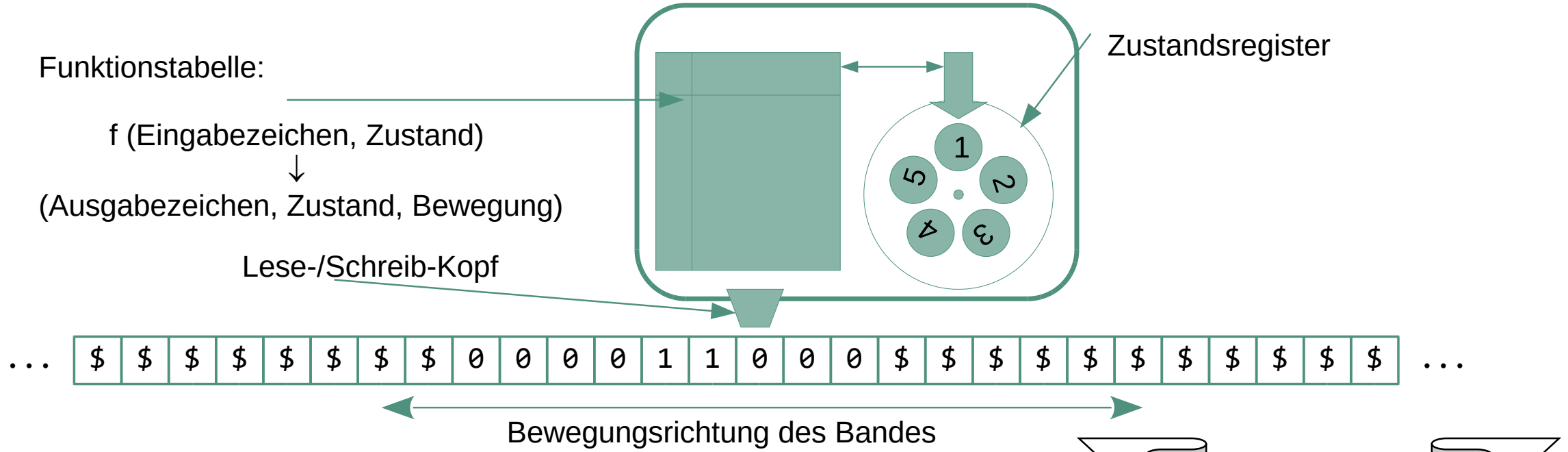
1940 Enigma

1950 Turing-Test

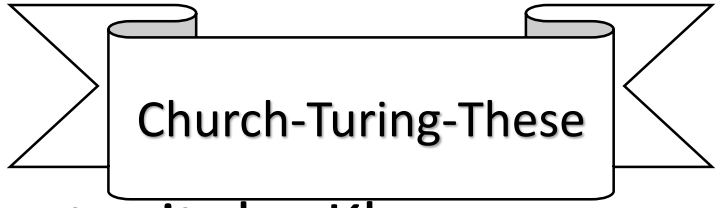




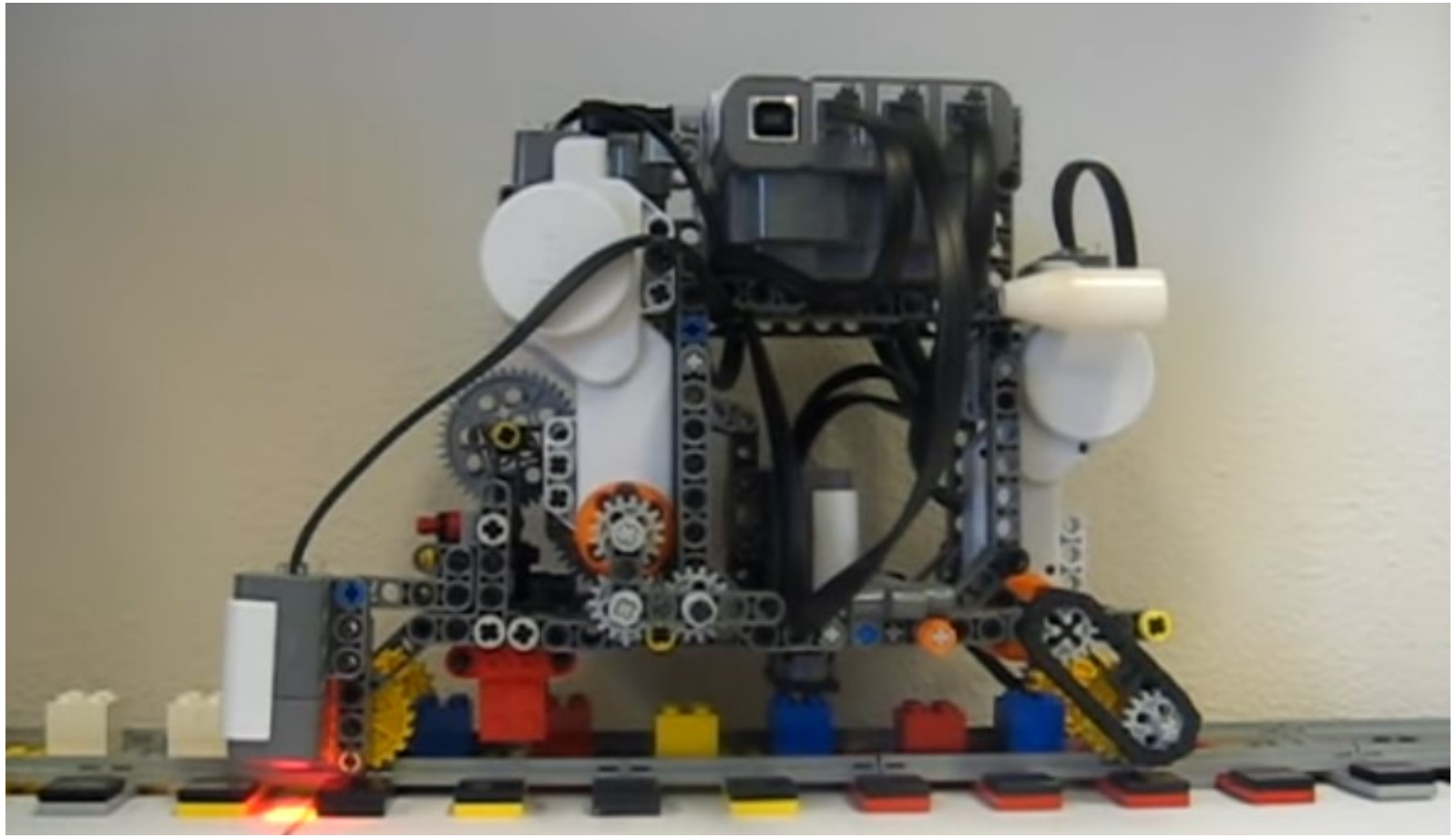
Turing-Maschine (1936)



- allgemeines, formalisiertes Modell für Computer
- „Die Klasse der Turing-berechenbaren Funktionen stimmt mit der Klasse der *intuitiv berechenbaren* Funktionen überein.“ (nicht beweisbar!)



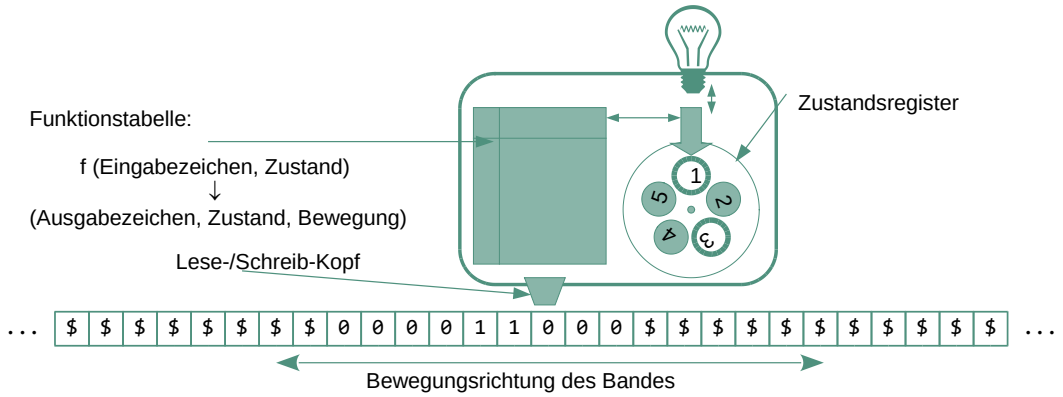
Eine Turingmaschine aus Lego



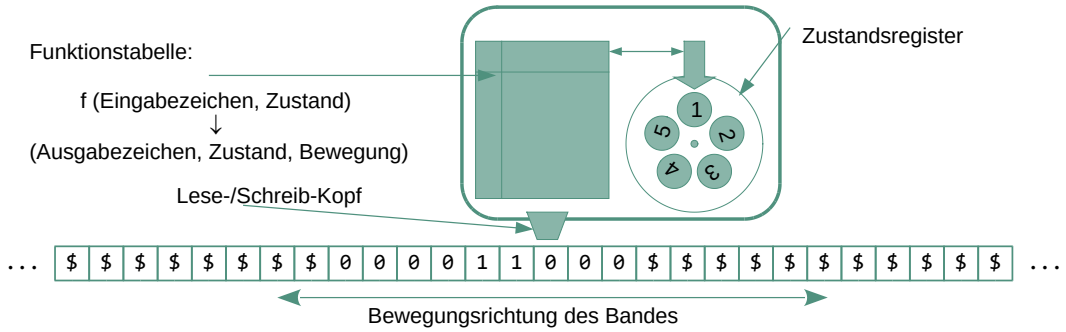
https://www.youtube.com/watch?v=cYw2ewoO6c4

TM als Akzeptor und Transduktor

Akzeptor



Transduktor



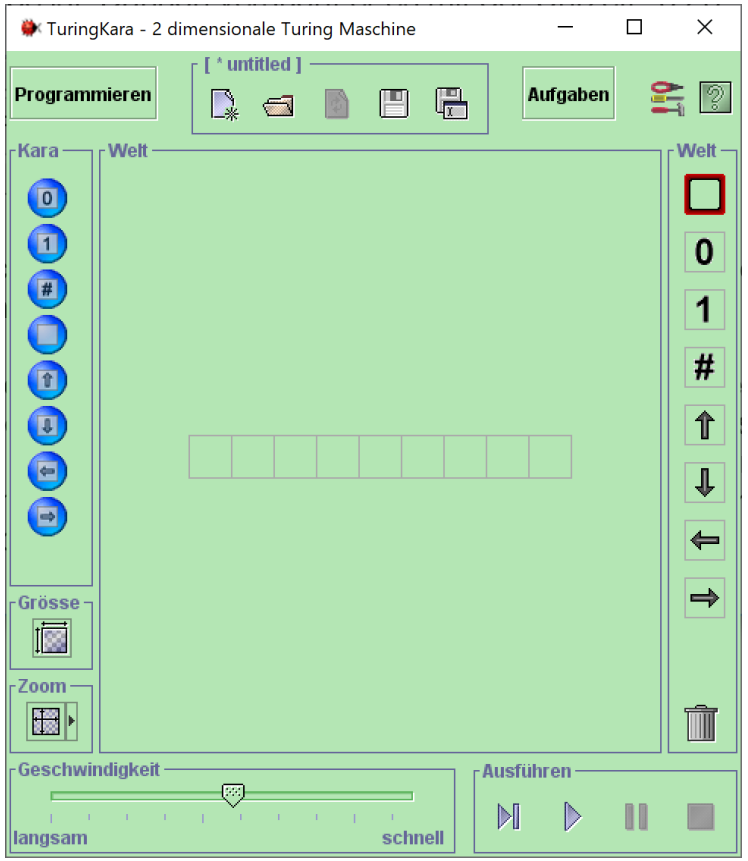
Werkzeug-Tip 5: Die Kara-Familie

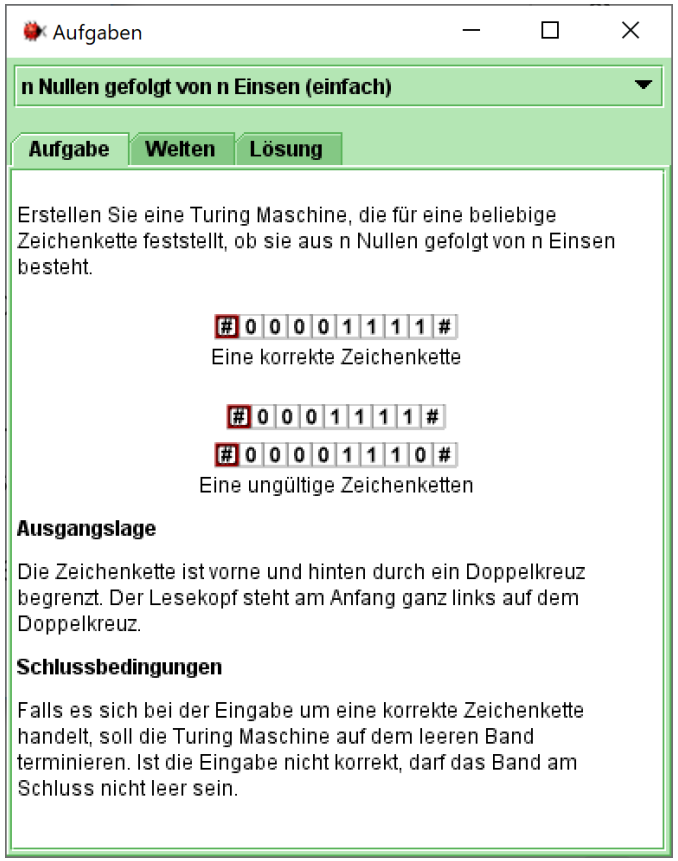
https://www.swisseduc.ch/informatik/karatojava/turingkara/

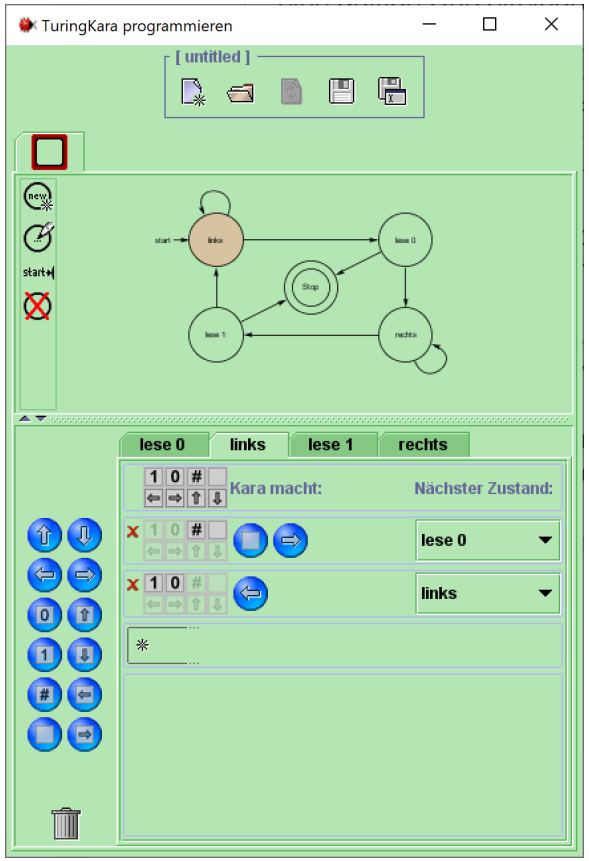
Programmieren mit endlichen Automaten

Turing-Kara

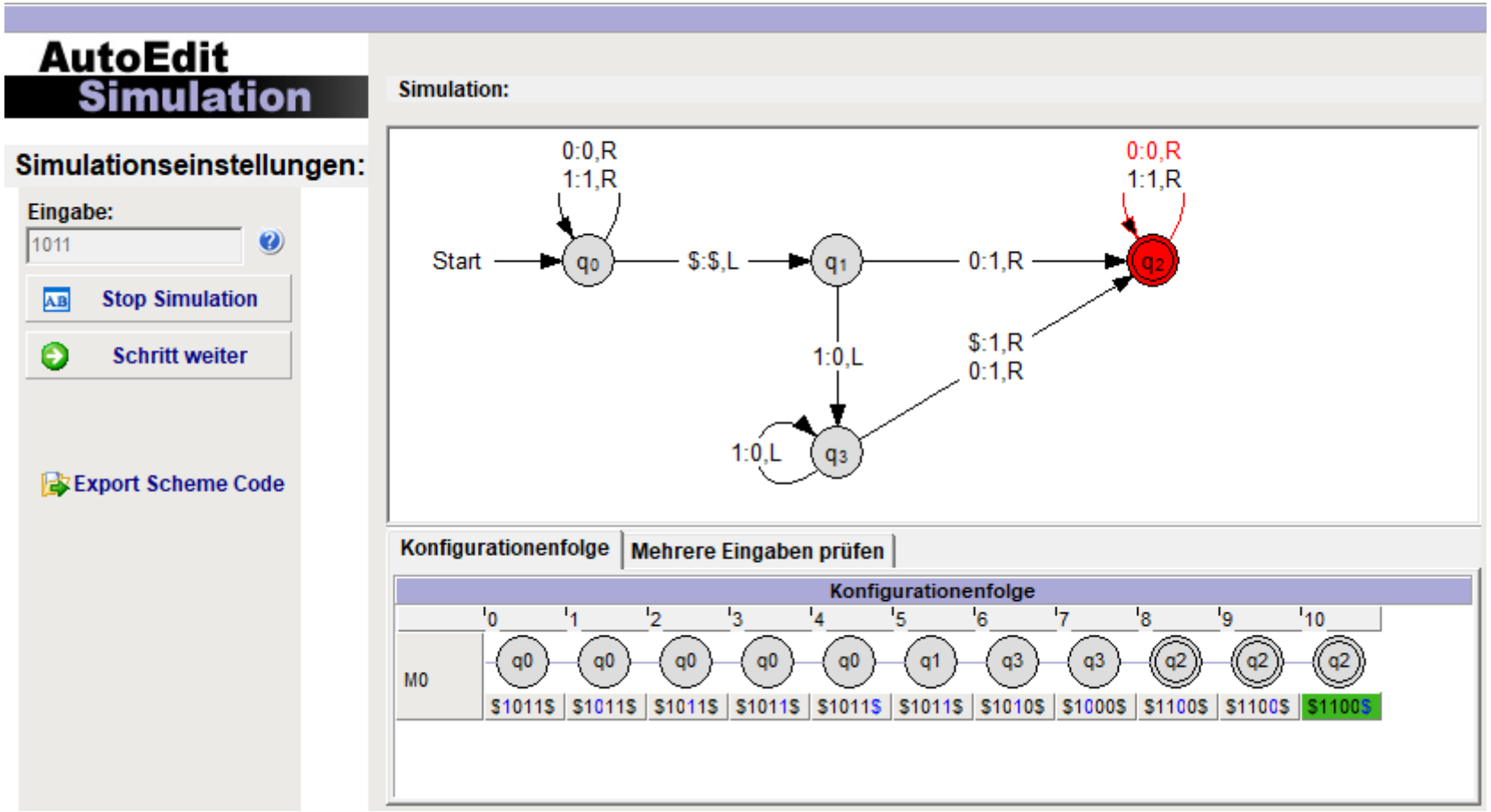
Turingkara (am besten ohne 2D-Option)







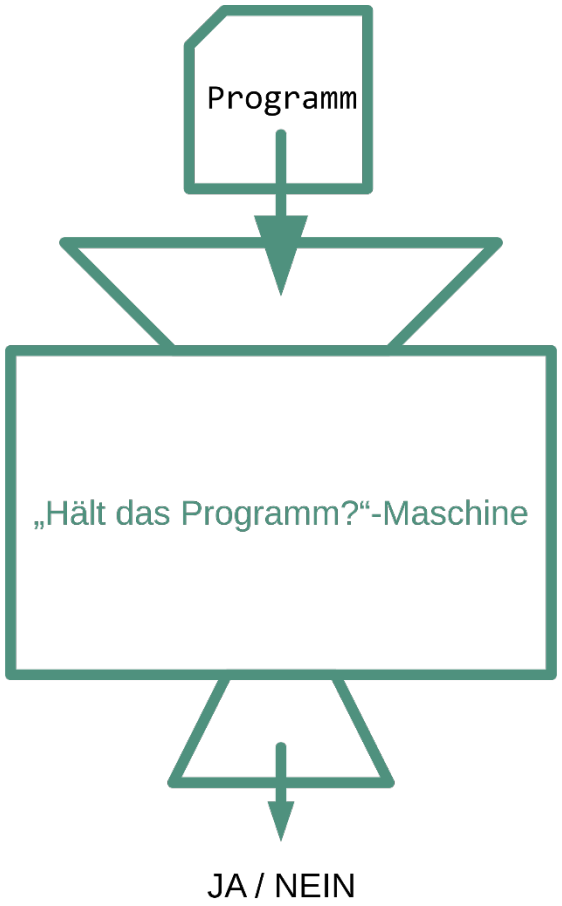
TM als Transduktor: Nachfolger einer Binärzahl



Das Halteproblem

Angenommen, es GÄBE eine Turingmaschine *STOPS_IT?*, die von jedem Algorithmus prüfen kann, ob er terminiert.

`STOPS_IT?(Program)` → `YES / NO`



Halteproblem – 2 Programme zur Einführung

Programm 1

```
1.WENN N == N+1 GEHE ZU Zeile 1
2.Ende.
```

Das Programm terminiert.

Programm 2

```
1.WENN N == N GEHE ZU Zeile 1
2.Ende.
```

Das Programm terminiert nicht.

Halteproblem – Teste Dich selbst.

Programm C lautet:

```
1. WENN (STOPS_IT?(Programm C)) GEHE ZU Zeile 1
2. Ende.
```

Offenbar prüft sich Programm C damit selbst.

Fall 1
- Programm C terminiert nicht.
- Dann wird Zeile 2 erreicht.

→ Programm C terminiert doch.

Fall 2
- Programm C terminiert.
- Dann wird Zeile 2 nicht erreicht.

→ Programm C terminiert doch nicht.

Grenzen von Turing-Maschinen

Als Schlussfolgerung aus dem Halteproblem ergibt sich.

Die Turingmaschine *Stops_it?* ist bewiesenermaßen nicht konstruierbar.

Damit gibt es Probleme, die prinzipiell, auch durch einen Rechner, nicht lösbar sind.

Das Halteproblem zeigt, dass es ist insbesondere nicht möglich ist, die Fehlerfreiheit aller Programme zu beweisen.

Implikationen des Halteproblems

David Hilbert

Kurt Gödel

1. Unvollständigkeitssatz (Gödel)

„Jedes hinreichend mächtige, rekursiv aufzählbare formale System ist entweder widersprüchlich oder unvollständig.“

2. Unvollständigkeitssatz (Gödel)

„Jedes hinreichend mächtige konsistente formale System kann die eigene Konsistenz nicht beweisen.“

Was bedeutet das für…

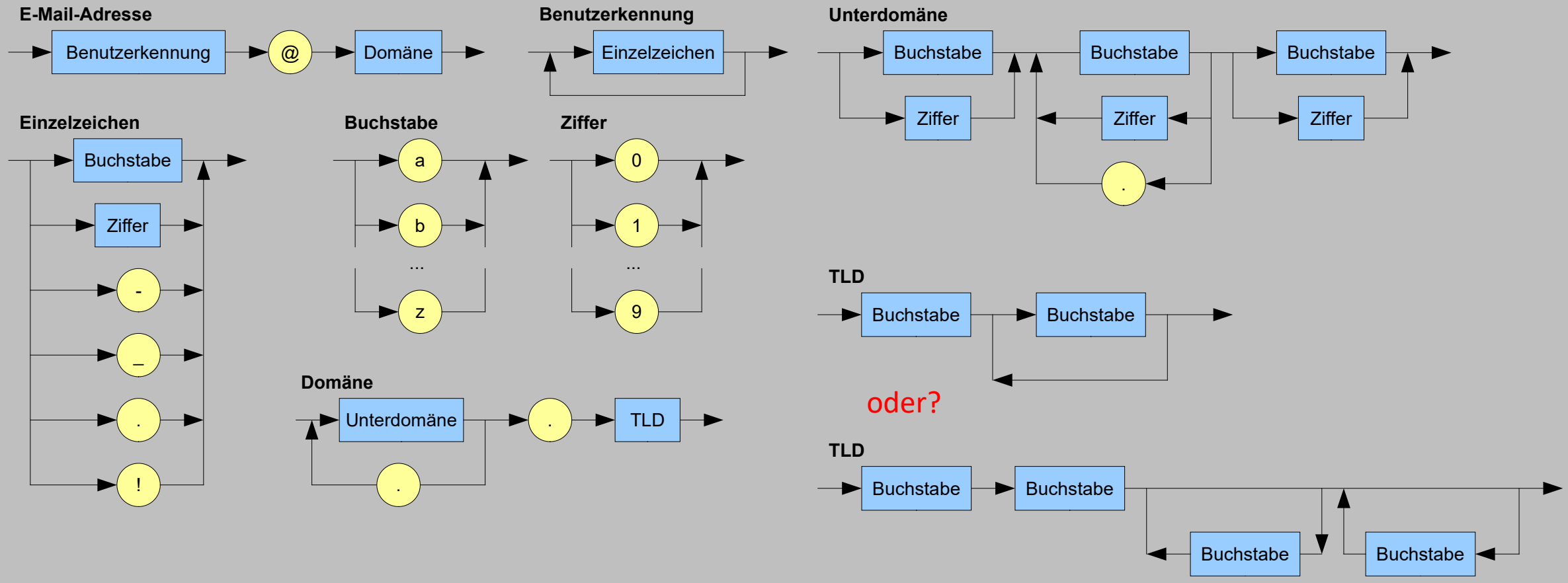
- die Fehlerfreiheit von Informatiksystemen?
- die Zuverlässigkeit von Künstlicher Intelligenz?

Formale Sprachen und Grammatiken (0/10)

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
ausschließlich für den Leistungskurs Grammatik G = (T, N, P, S) • mithilfe der Definition beschreiben • Wörter nachvollziehbar ableiten • die Ableitbarkeit von Wörtern untersuchen • die erzeugte Sprache bestimmen • zu einer gegebenen regulären Sprache eine Grammatik entwickeln	
Chomsky-Hierarchie [Deutsch] [Philosophie] • den Typ einer Grammatik bestimmen • den Zusammenhang zwischen Grammatik, Sprache und Automat mithilfe der Chomsky-Hierarchie beschreiben • für eine aus einer regulären oder kontext-freien Grammatik erzeugte Sprache einen erkennenden Automaten angeben	

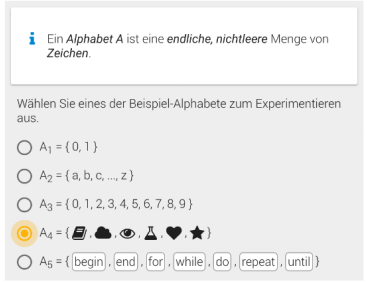
T: Terminale
N: Nichtterminale
P: Produktionsregeln
S: Startsymbol
(Abweichende Definitionen möglich)

Syntaxdiagramme



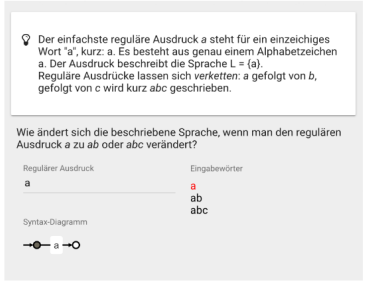
Werkzeug-Tip 4: FLACI (https://flaci.com)

Formale Sprachen



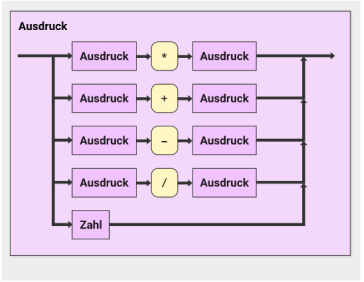
Interaktives Minitutorial zu den Grundbegriffen formaler Sprachen

Reguläre Ausdrücke



Interaktives Minitutorial zu regulären Ausdrücken

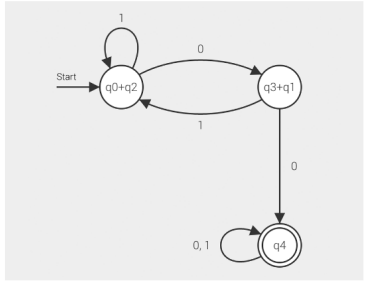
Formale Grammatiken



Kontextfreie Grammatiken entwickeln, transformieren und konvertieren

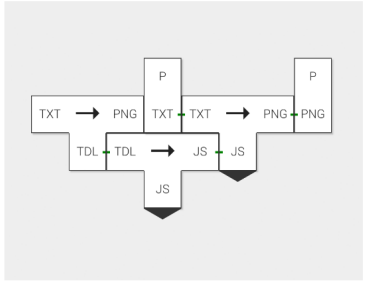
→ Formale Grammatiken
→ Beispielsammlung
→ Grammatiken
→ Rechner

Abstrakte Automaten



Abstrakte Automaten konstruieren, simulieren, transformieren und konvertieren

Compiler und Interpreter



Modellieren von Übersetzungsprozessen und Entwicklung von Compilern und Interpretern

Wie viele Sätze enthält eine UN-Resolution?

Genau einen Satz aus drei Teilen:

- **Die Bezeichnung des Gremiums,**
  das die Resolution verabschiedet (beispielsweise der Sicherheitsrat, die UN-Vollversammlung, ein der Vollversammlung untergeordnetes Organ oder eine andere Organisation), ist das Subjekt des Satzes.
- **Die präamblen Klauseln**
  verweisen auf die Beweggründe der Resolution, vergleichbar mit einer Präambel als Vorbemerkung in anderen Dokumenten.
  Die präamblen Klauseln sind unnummerierte Stichpunkte. Sie beginnen mit kursiv gesetzten Verben im Partizip I (gelegentlich in Verbindung mit einem Adverb), gefolgt von den Beweggründen. Die präamblen Klauseln werden durch Komma getrennt.
- **Die operativen Klauseln**
  enthalten die vom Gremium beschlossenen Maßnahmen oder gegebenen Forderungen.
  Sie sind nummeriert und beginnen jeweils mit einem kursiv gesetzten Verb in der Aktivform, gefolgt von der Forderung bzw. Maßnahme. Sie werden mit einem Semikolon beendet – mit Ausnahme der letzten Klausel, die mit einem Punkt abschließt.

Wie viele Sätze enthält eine UN-Resolution?

Der Aufbau einer UN-Resolution kann formal beschrieben werden.

1. Geben Sie aufgrund der obigen Beschreibung eine Grammatik zur Beschreibung der *Struktur* einer UN-Resolution an und erklären Sie Ihre Lösung.
2. Entscheiden Sie, ob es sich bei der von Ihnen gefundenen Grammatik um eine reguläre Grammatik handelt.

Grammatik einer UN-Resolution

<UN-RESOLUTION> := <SUBJEKT><ABSATZ> ,<PRÄAMBLE_KLAUSELN>,<OPERATIONALE_KLAUSELN>.

<PRÄAMBLE_KLAUSELN> := <PRÄAMBLE_KLAUSELN> <PRÄAMBLE_KLAUSEL>

<PRÄAMBLE_KLAUSELN> := <PRÄAMBLE_KLAUSEL>

<PRÄAMBLE_KLAUSEL> := <PARTIZIP> <OBJEKT>,<ABSATZ>

<OPERATIONALE_KLAUSELN> := <OPERATIONALE_KLAUSELN>;<ABSATZ> <OPERATIONALE_KLAUSEL>

<OPERATIONALE_KLAUSELN> := <OPERATIONALE_KLAUSEL>

<OPERATIONALE_KLAUSEL> := <LFD_NR>.<VERB><OBJEKT>

Formale Sprachen und Grammatiken (0/10)

Verbindliche Ziele und Inhalte	Hinweise und Anregungen
ausschließlich für den Leistungskurs	
Grammatik G = (T, N, P, S) • mithilfe der Definition beschreiben • Wörter nachvollziehbar ableiten • die Ableitbarkeit von Wörtern untersuchen • die erzeugte Sprache bestimmen • zu einer gegebenen regulären Sprache eine Grammatik entwickeln	
Chomsky-Hierarchie [Deutsch] [Philosophie] • den Typ einer Grammatik bestimmen • den Zusammenhang zwischen Grammatik, Sprache und Automat mithilfe der Chomsky-Hierarchie beschreiben • für eine aus einer regulären oder kontext-freien Grammatik erzeugte Sprache einen erkennenden Automaten angeben	

Noam Chomsky (*1928)

- libertärer Sozialist mit Sympathien für den Anarchosyndikalismus
- kritischer Medienwissenschaftler
- Linguist
- Kompetenz-Erfinder
- Informatiker

Zwischen 1980 und 1992 die am häufigsten zitierte lebende Person der Welt
(Arts and Humanities Citation Index 1992)

Sprache, Grammatik, Automat

Die Chomsky-Hierarchie

Grammatik	Sprache	Automat
Typ 3	Reguläre Sprache	Akzeptor (Medwedew)
Typ 2	Kontextfreie Sprache	nichtdeterministischer Kellerautomat
Typ 1	Kontextsensitive Sprache	nichtdeterministische, linear beschränkte Turingmaschine
Typ 0	Rekursiv aufzählbar	Turingmaschine

Aufgabenbeispiele

für Grund- und Leistungskurs diskutieren wir in den Breakouträumen.