Kapitel 2

Reguläre Sprachen

2.1

Deterministischer Endlicher Automat (DEA)

Prof. Dr. Robert Preis
Fachbereich Informatik
Fachhochschule Dortmund
Robert.Preis@fh-dortmund.de

Wdh.: Typ 3+?: Deterministische reguläre (rechtslineare) Grammatiken

Bei einer deterministischen rechtslinearen Grammatik gibt es immer genau eine Regel (ohne ε), die man anwenden kann:

G = ({S,A,B,C}, {a,b}, P, S) mit

P = { S → aA | bB,
A → aA | bC | ε,
B → aC | bB | ε,
C → aA | bC}

Ableitung z.B. S→aA→abC→abbC→abbaA→abba

Deterministische reguläre Grammatiken sind sehr einfach, weil

1. man das Wort immer von links nach rechts abarbeitet,
2. dabei immer genau einen Buchstaben weitergeht,
3. es nach jedem Schritt immer ganz rechts genau eine Variable gibt,
4. es dabei immer genau eine Regel gibt, die man anwenden kann,
5. man zum Schluss einfach nur schauen muss, ob es eine ε-Regel gibt.

Frage: Kann man das Wort „bbabbaaa“ ableiten?

Was muss man sich bei einer Typ 3+ Ableitung merken?

G = ({S,A,B,C}, {a,b}, P, S) mit

P = { S → aA | bB,
A → aA | bC | ε,
B → aC | bB | ε,
C → aA | bC}

Ableitung z.B. $\underline{S}$→a$\underline{A}$→ab$\underline{C}$→abb$\underline{C}$→abba$\underline{A}$→abba

Beobachtungen:

- Das Wort wird automatisch von links nach rechts aufgebaut.
- Es gibt immer rechts eine Variable, d.h. man muss sich einfach nur merken, welche Variable es gerade ist.

Wir merken uns während der Ableitung mit der Variable den Zustand, in dem wir gerade sind.

Ein Automat für das Wortproblem für determ. rechtslineare Grammatiken

Wortproblem für deterministische rechtslineare Grammatiken:

Kann die Grammatik G ein Eingabewort w erzeugen?

Lösung:

1. Durch Ausprobieren.
2. Durch Benutzung einer Maschine/eines Rechners/eines Automaten.

Eingabe
(Eingabewort w) →

Sprache dargestellt durch deterministische rechtslineare Grammatik

$G = (\{S,A,B,C\}, \{a,b\}, P, S)$

$$
\begin{aligned}
P = \{ \quad & S \rightarrow aA \mid bB, \\
& A \rightarrow aA \mid bC \mid \varepsilon, \\
& B \rightarrow aC \mid bB \mid \varepsilon, \\
& C \rightarrow aA \mid bC \}
\end{aligned}
$$

Automat

→ **Ausgabe**
(Enthalten? Ja/Nein)

Ein Automat ist ein einfaches Maschinenmodell, das ohne Speicher die Eingabe direkt verarbeitet.

Von det. rechtslinearer Grammatik zu Tabelle und Diagramm …

Deterministische rechtslineare Grammatik G = (V, T, P, S)

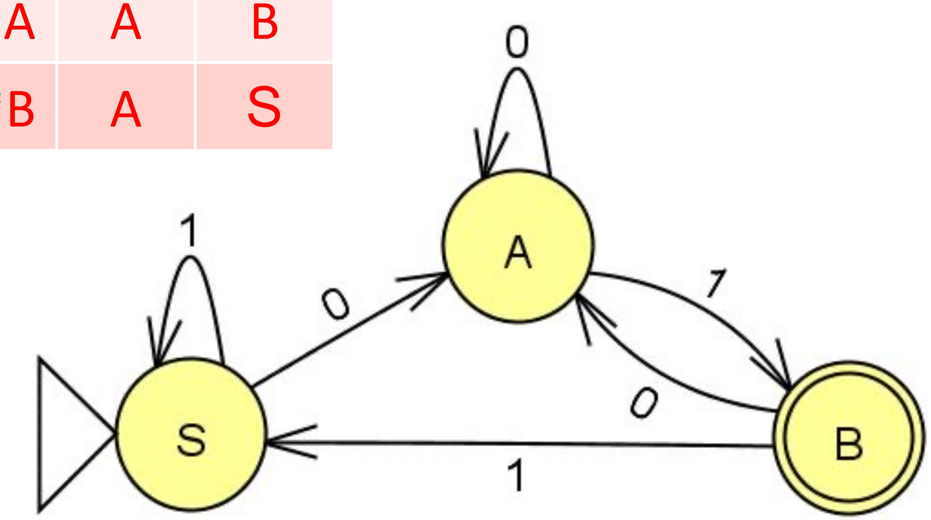
1. **als Grammatik:** G = ({S,A,B}, {0,1}, P, S) mit

   P = { S → 0A | 1S,
   A → 0A | 1B,
   B → 0A | 1S | ε }

2. **als Automat mit Tabelle:**
   Ein Automat ist eine tabellarische Darstellung einer det. rechtslinearen Grammatik.

	0	1
→S	A	S
A	A	B
*B	A	S

3. **als Automat mit Diagramm:**
   Ein Automat ist eine grafische Darstellung einer det. rechtslinearen Grammatik.



Ein Automat für eine deterministische reguläre Grammatik

Frage: *Ist die Zahl 4711 durch 3 teilbar?*

Wortproblem: *4711 $\in$ {Alle Zahlen, die durch 3 teilbar sind.} ?*

Grammatik: G = ({S,A,B}, {0,1,2,3,4,5,6,7,8,9}, P, S) mit P = {

S→0S|1A|2B|3S|4A|5B|6S|7A|8B|9S|ε, (*S=bisher Quers. durch 3 teilbar*)

A→0A|1B|2S|3A|4B|5S|6A|7B|8S|9A, (*A=bisher Quers. 1 zu viel*)

B→0B|1S|2A|3B|4S|5A|6B|7S|8A|9B} (*B=bisher Quers. 2 zu viel*)

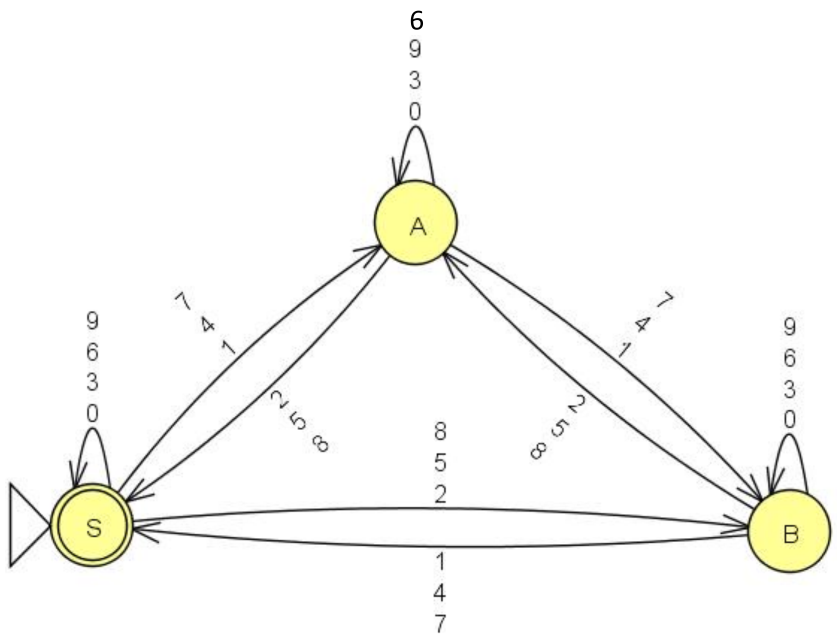
Ableitung: S→4A→47B→471S→4711A

Automat: *Ich gehe das Wort von links nach rechts durch und ich merke mir, wo ich gerade bin, d.h. in welchem Zustand ich bin:*

S: *Bisherige Quersumme ist durch 3 teilbar.*

A: *Bisherige Quersumme ist 1 zu viel.*

B: *Bisherige Quersumme ist 2 zu viel.*



Wort im Text finden

Frage: *Enthält der Text*

*„01101011010101000111010100110101001011000110101010001110010001110
010101010010101010101010100010110001110010101001101010101001010101
001010101001010101001010100101001100101000101111001010101001010101
010010101001010100101010010100101010000111100101010010101010100101
00101011100100101001101001011111000“*

das Teilwort „011001“ ?

Wie gehen Sie dabei vor?

Einfach: Bei jeder „0“ prüfen, ob es der Anfang von „011001“ ist.

Problem: Man muss die Stellen hinter jeder „0“ mehrfach betrachten…

Wie gehen Sie vor, wenn Sie den Text nur einmal von links nach rechts lesen dürfen?

Ein Automat zur Worterkennung

Frage: *Enthält der Text*

„GESTERN SAGTE THORSTEN, DASS DAS THEMA IN DER VORLESUNG SEHR THEORETISCH WAR“

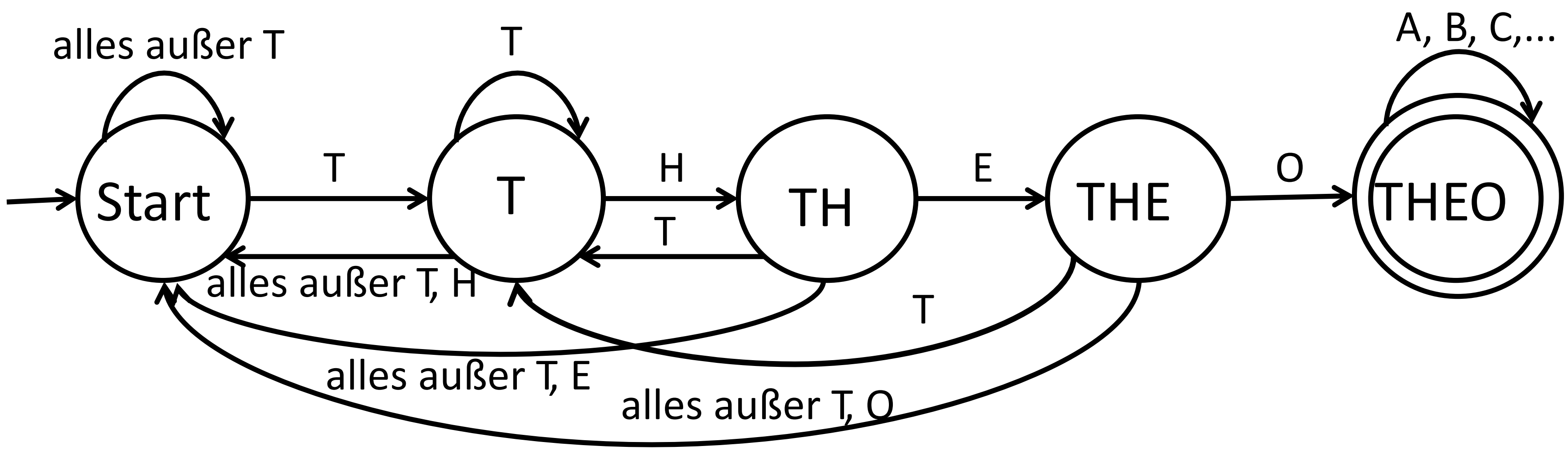
das Wort „THEO“?

Wortproblem: *Text $\in$ {Alle Sätze, in denen das Wort „THEO“ ist.} ?*

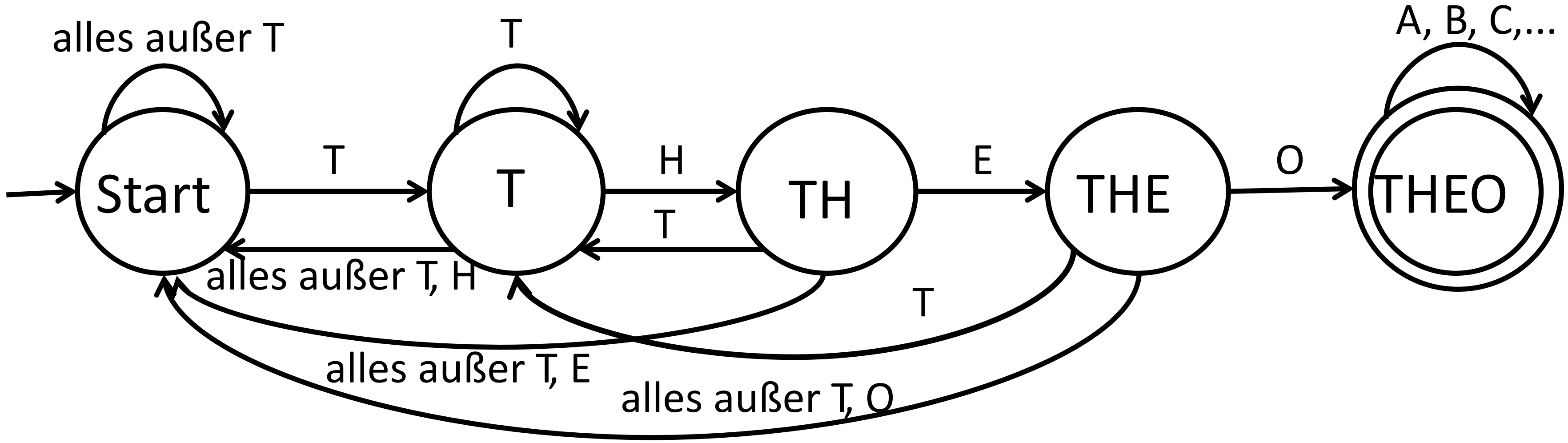
Wie sieht dazu eine deterministische rechtslineare Grammatik aus?

Automat:

Gehe Zeichen für Zeichen durch und wechsle entsprechend den Zustand!



Deterministischer Endlicher Automat (DEA)



Ein Deterministischer Endlicher Automat (DEA) ist ein 5-Tupel

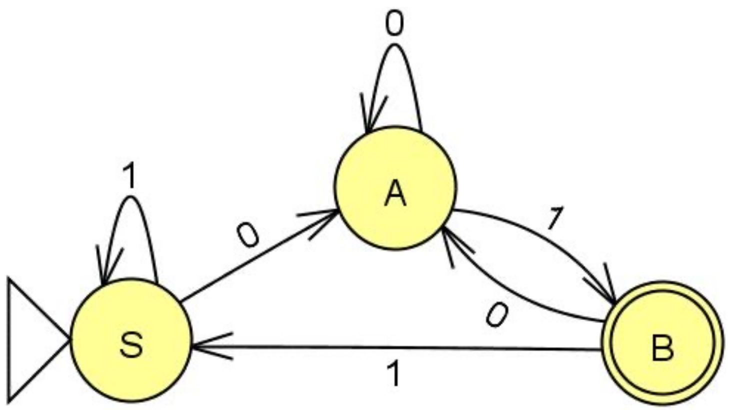
$$A = (Q, \Sigma, \delta, q_0, F)$$

mit

Q: nichtleere, endliche Menge von Zuständen	(Variablen: Kreise)
∑: endliches Eingabealphabet	(Alphabet: an den Pfeilen)
δ: $Q \times \Sigma \rightarrow Q$ Übergangsfunktion	(Regeln: Pfeile)
$q_0 \in Q$: Startzustand	(Startvariable: Startpfeil)
$F \subseteq Q$: Menge von Endzuständen	(ε-Regel: Doppelkreis)

Übergangsdiagramm und Übergangstabelle eines DEA

Man kann die Übergangsfunktion eines DEA mit einem Übergangs-diagramm oder einer Übergangstabelle darstellen.



$A = (Q, \sum, \delta, q_0, F)$

- Zustandsmenge Q:
  1. Spalte
- Alphabet ∑:
  1. Zeile
- Übergangsfunktion δ:
  Inhalt der Tabelle
  (genau ein Übergang pro Eintrag)
- Startzustand q_0:
  Pfeil an dem Zustand
- Endzustände F:
  Sterne an den Zuständen

δ	0	1
→S	A	S
A	A	B
*B	A	S

$A = (\{S,A,B\}, \{0,1\}, \delta, S, \{B\})$

Äquivalenz deterministische rechtslineare Grammatik und DEA

Deterministische rechtslineare Grammatik

G = (V, T, P, S)

DEA

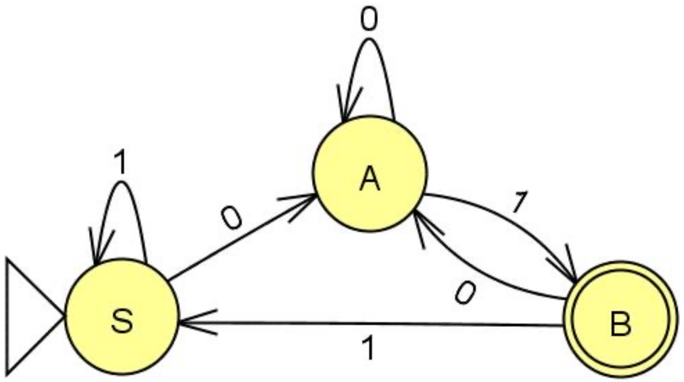
A = (V, T, δ, S, F)

G = ({S,A,B}, {0,1}, P, S) mit

P = { S → 0A | 1S,
A → 0A | 1B,
B → 0A | 1S | ε}

A = ({S,A,B}, {0,1}, δ, S, {B}) mit

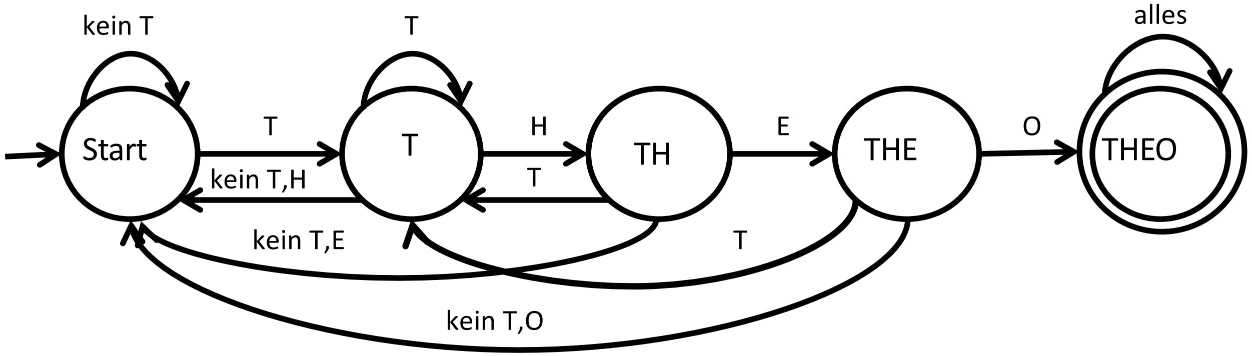
δ	0	1
→S	A	S
A	A	B
*B	A	S



- Startvariable wird Startzustand und umgekehrt
- ε-Regeln werden zu Endzuständen und umgekehrt

Man kann jede deterministische rechtslineare Grammatik in einen äquivalenten DEA und umgekehrt umwandeln !

Arbeitsweise von endlichen Automaten



Anfangssituation:

- Automat befindet sich im Startzustand q_0.

Arbeitsschritt:

- Im Zustand q lese Eingabesymbol a,
- Bestimme $\delta(q,a)=p$ und wechsele in neuen Zustand p.

Terminierung:

- Eingabewort $w = a_1..a_n$ ist komplett gelesen, Automat im Zustand q_n.

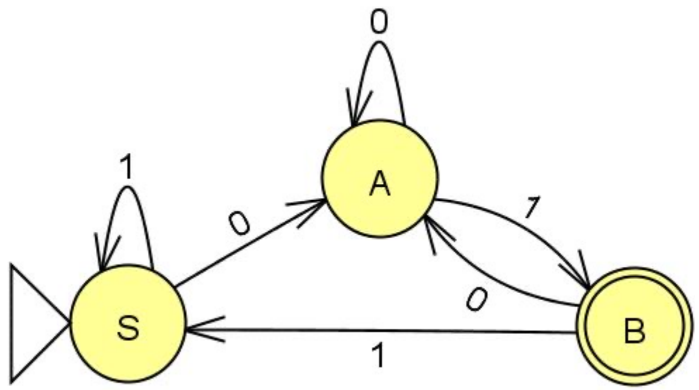
Ergebnis:

- Falls $q_n \in F$, dann akzeptiere, sonst weise w ab.

Konfigurationen und Übergang

Ableitung (zeigt, was schon gemacht wurde und wo man jetzt ist):

$\underline{S} \rightarrow 1\underline{S} \rightarrow 10\underline{A} \rightarrow 100\underline{A} \rightarrow 1001\underline{B} \rightarrow 1001$



Eine Konfiguration ist ein 2er-Tupel

$(q,w) \in Q \times \Sigma^*$,

der den aktuellen Gesamtzustand eines Automaten, darstellt, d.h. der Automat ist im Zustand q und muss noch das Wort w abarbeiten.

Durch die Anwendung eines Übergangs wechselt der Automat in eine neue Konfiguration:

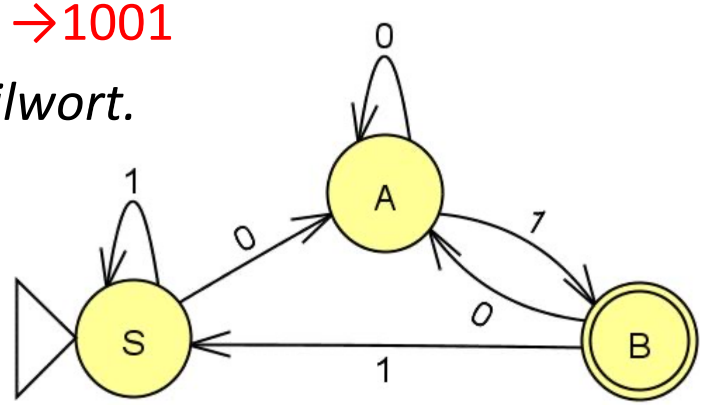
$(q,av) \vdash (p,v)$, falls $\delta(q, a) = p$

Konfigurationsübergänge und akzeptierte Sprache

Ableitung (zeigt, was schon gemacht wurde und wo man jetzt ist):

S → 1S → 10A → 100A → 1001B → 1001

Ableitung zeigt Variablen und schon abgeleitetes Teilwort.



Konfigurationsübergänge:

(S,1001) ⊢ (S,001) ⊢ (A,01) ⊢ (A,1) ⊢ (B,ε)

Konfigurationsübergänge zeigen Variablen und noch abzuleitendes Teilwort.

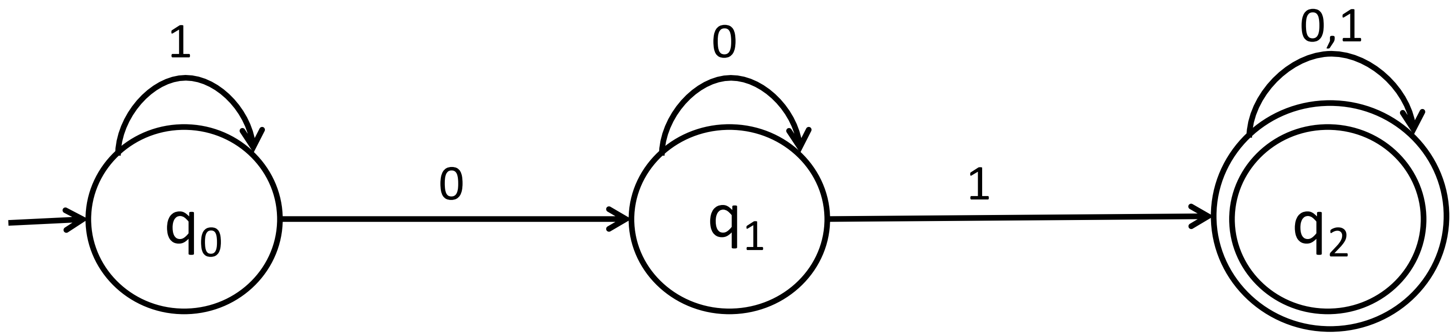
Akzeptierte Sprache eines Automaten A:

Menge der Eingaben w, für die $\vdash^*$ von dem Startzustand q_0 zu irgendeinen akzeptierenden Zustand p führt:

$$L(A) = \{w \in \Sigma^* \mid \exists p \in F: (q_0,w) \vdash^* (p,\varepsilon)\}$$

Entwurf eines Automaten

L = {w∈{0,1}* | w enthält das Teilwort 01 } = {w∈{0,1}* | w=u01v; u,v∈{0,1}* }



$G = (\{q_0,q_1,q_2\}, \{0,1\}, P, q_0)$

$A = (\{q_0,q_1,q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$

$P = \{ \quad q_0 \to 0q_1 \mid 1q_0,$
$\quad q_1 \to 0q_1 \mid 1q_2,$
$\quad q_2 \to 0q_2 \mid 1q_2 \mid \varepsilon\}$

	0	**1**
$\to q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_2	q_2

Die Zustände haben die folgenden Bedeutungen

q_0: A hat noch keine 0 gelesen

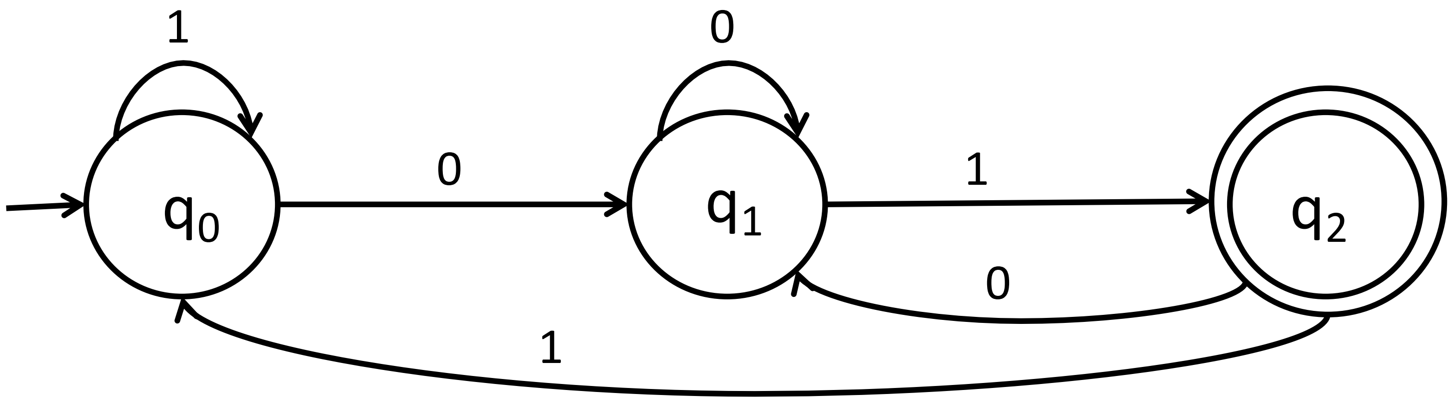
q_1: A hat eine 0, aber noch keine 1 gelesen

q_2: A hat das Teilwort 01 bereits gelesen

Entwurf eines Automaten

Entwurf eines Automaten für die Sprache

$L = \{ w \in \{0,1\}^* \mid w \text{ endet mit } 01 \} = \{ w \in \{0,1\}^* \mid w = u01;\ u \in \{0,1\}^* \}$



	0	1
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
$*q_2$	q_1	q_0

Die Zustände haben die folgenden Bedeutungen

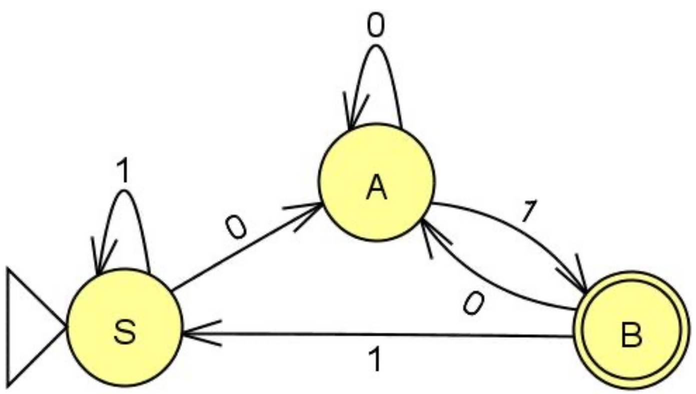
q_0: A hat noch keine 0 gelesen oder zuletzt 11 gelesen

q_1: A hat als letztes eine 0 gelesen

q_2: A hat das Teilwort 01 als letztes gelesen

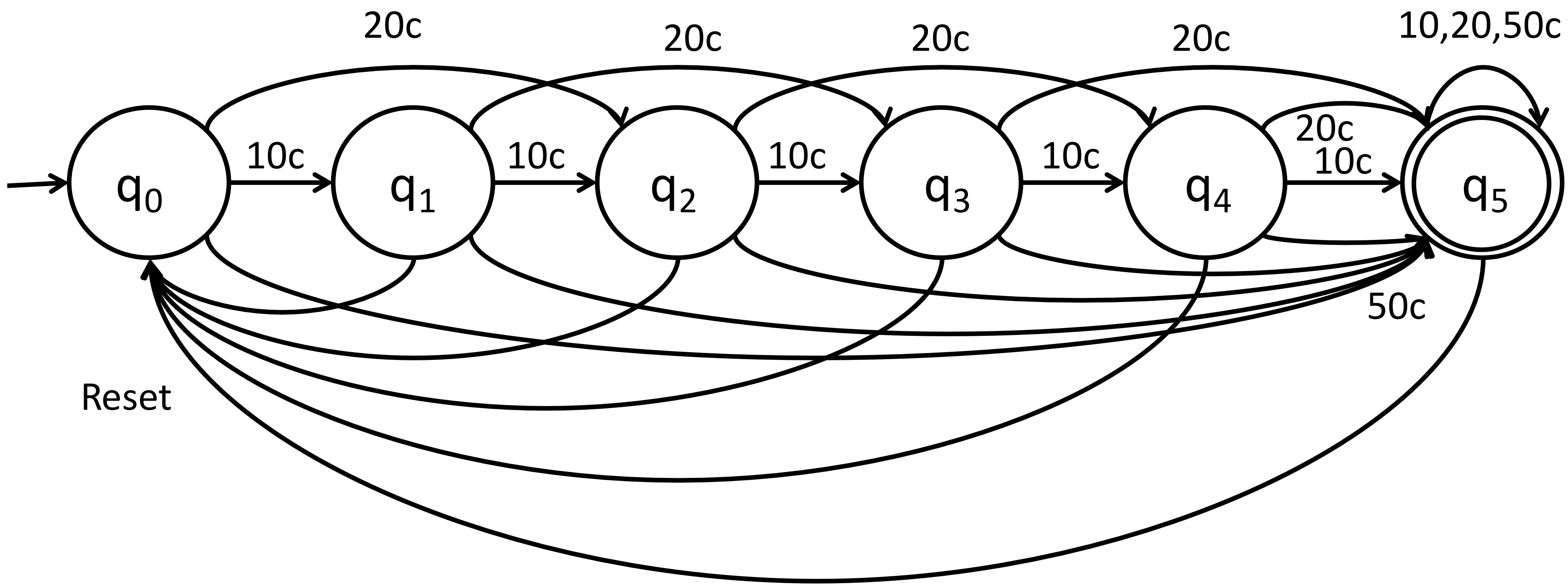
$A = (\{q_0,q_1,q_2\}, \{0,1\}, \delta, q_0, \{q_2\})$

Das ist derselbe Automat wie vorhin,
nur mit anderen Zustandsbezeichnern:



Beispiel: 50 cent Kaffeeautomat

- Ein Kaffee in einem Kaffeeautomat kostet 50 cent.
- Man muss mit 10 cent, 20 cent oder 50 cent Münzen bezahlen, es gibt kein Rückgeld.
- Es gibt aber eine Reset Taste.



Verwendungszweck für endliche Automaten

- Steuerungsautomaten:
  alle Formen rein Hardware-gesteuerter automatischer Maschinen (Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen,…)
- Entwurf und Überprüfung digitaler Schaltungen:
  Entwicklungswerkzeuge und Testsoftware beschreiben endliches Verhalten
- Lexikalische Analyse in Compilern:
  Schnelle Identifizierung von Bezeichnern, Schlüsselwörtern, . . .
- Textsuche in umfangreichen Dokumenten:
  Z.B. Suche nach Webseiten mithilfe von Schlüsselwörtern
- Software mit endlichen Alternativen:
  Kommunikationsprotokolle, Protokolle zum sicheren Datenaustausch, …

Zusammenfassung

- Bei einer deterministischen rechtslinearen Grammatik kann man leicht analysieren, ob ein Wort ableitbar ist oder nicht.
- Ein Deterministischer Endlicher Automat (DEA) ist eine Maschine zur Analyse von deterministischen rechtslinearen Grammatiken.
- Ein DEA ist ein 5-Tupel $A = (Q, \Sigma, \delta, q_0, F)$.
- Ein DEA kann entweder als Übergangsdiagramm oder als Übergangstabelle dargestellt werden.
- Man kann jede deterministische rechtslineare Grammatik in einen äquivalenten DEA und umgekehrt umwandeln.
- Die von einem Automaten akzeptierte Sprache sind alle Wörter, die nach deren Abarbeitung in einem Endzustand sind.