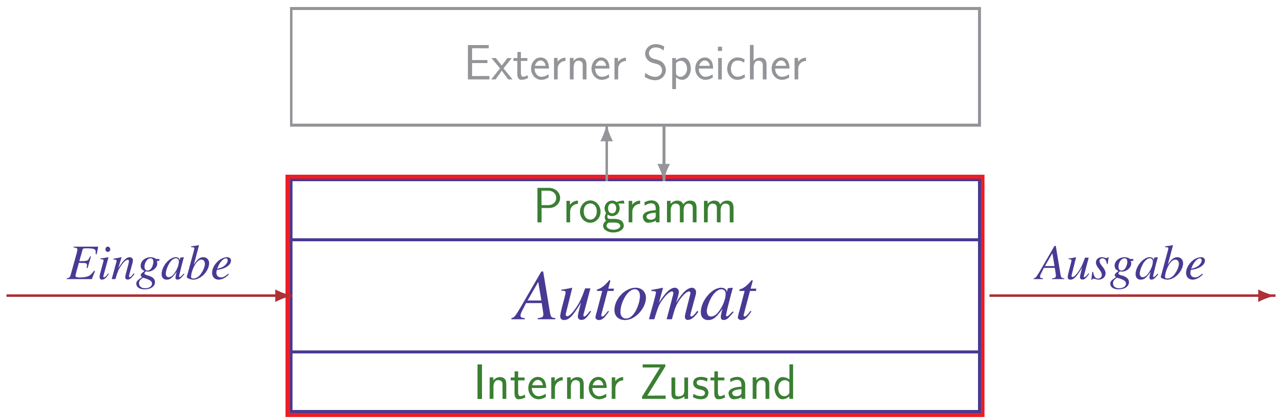
Theoretische Informatik I

Einheit 2

Endliche Automaten
&
Reguläre Sprachen

1. Deterministische endliche Automaten
2. Nichtdeterministische Automaten
3. Reguläre Ausdrücke
4. Grammatiken
5. Eigenschaften regulärer Sprachen

Automaten: das einfachste Maschinenmodell



Sichtweisen von Computern

- **Automaten stehen im Kern jeder Berechnung**
  - Schnelle, direkte Verarbeitung von Eingaben
  - Keine interne Speicherung von Daten
  - Speicher sind Teil der Umgebung
- **Endliche Automaten sind leicht zu analysieren**
  - Jede Berechnung endet nach einer festen Anzahl von Schritten
  - Keine Schleifen oder Seiteneffekte

Verwendungszwecke für endliche Automaten

Grundlegendes und vielseitiges Modellierungskonzept für viele Arten von Hard- & Software

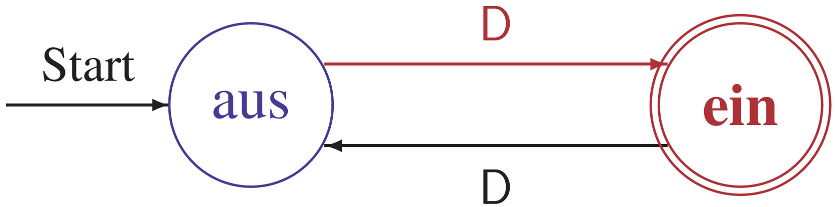
- **Steuerungsautomaten**
  - Alle Formen rein Hardware-gesteuerter automatischer Maschinen
    Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren
- **Entwurf und Überprüfung digitaler Schaltungen**
  - Entwicklungswerkzeuge & Testsoftware beschreiben endliches Verhalten
- **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 . . .

Automaten beschreiben Sprachen

- **Generierte Sprache**
  - Menge aller möglichen Ausgaben des Automaten
- **Erkannte Sprache**
  - Menge aller Eingaben, die zur Ausgabe “ja” führen
  - Alternativ: letzter Zustand des Automaten muß ein “Endzustand” sein
- **Sprachen endlicher Automaten sind einfach**
  - Nur sehr einfach strukturierte Sprachen können beschrieben werden
  - Durch endliche Automaten beschreibbare Sprachen heißen **regulär**

Modelle zur Beschreibung regulärer Sprachen

- **Automaten: Erkennen von Wörtern**
  - z.B. Wechselschalter: Verarbeitung von “Drück”-Eingaben



  - Zustände: aus, ein – Startzustand: aus – Endzustand: ein
  - Eingabe: Drücken eines Schalters, dargestellt als Symbol D
  - Endzustand wird erreicht bei ungerader Anzahl von Drück-Eingaben
- **Mathematische Mengennotation**
  - z.B.: $\{w \in \{\mathsf{D}\}^* \mid \exists i \in \mathbb{N}. |w| = 2i+1\}$, kurz $\{\mathsf{D}^{2i+1} \mid i \in \mathbb{N}\}$
- **Reguläre Ausdrücke: algebraische Strukturen**
  - z.B.: $(\mathsf{DD})^*\mathsf{D}$
- **Grammatiken: Vorschriften für Spracherzeugung**
  - z.B.: $S \to \mathsf{D}$, $S \to S\mathsf{DD}$
  - Erzeugt nur eine ungerade Anzahl von D-Symbolen

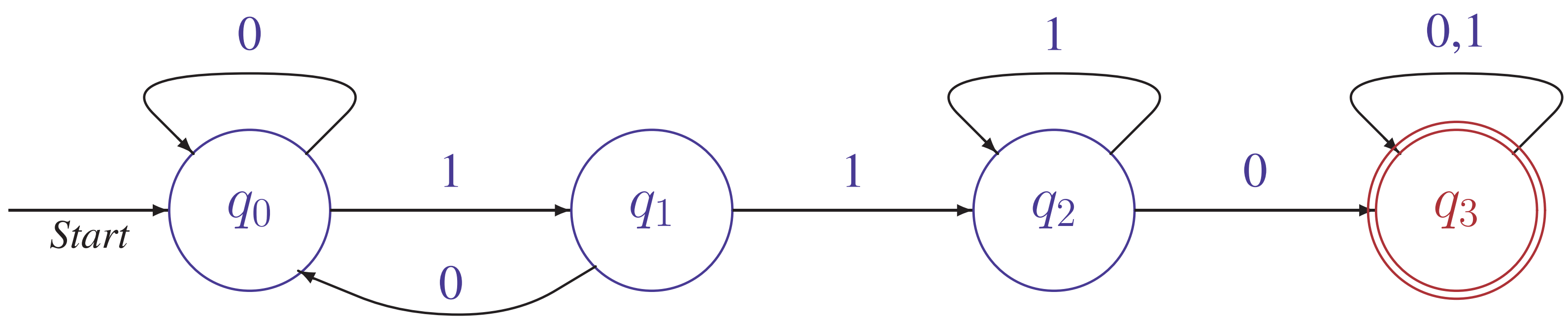
Theoretische Informatik I

Einheit 2.1

Deterministische Endliche Automaten

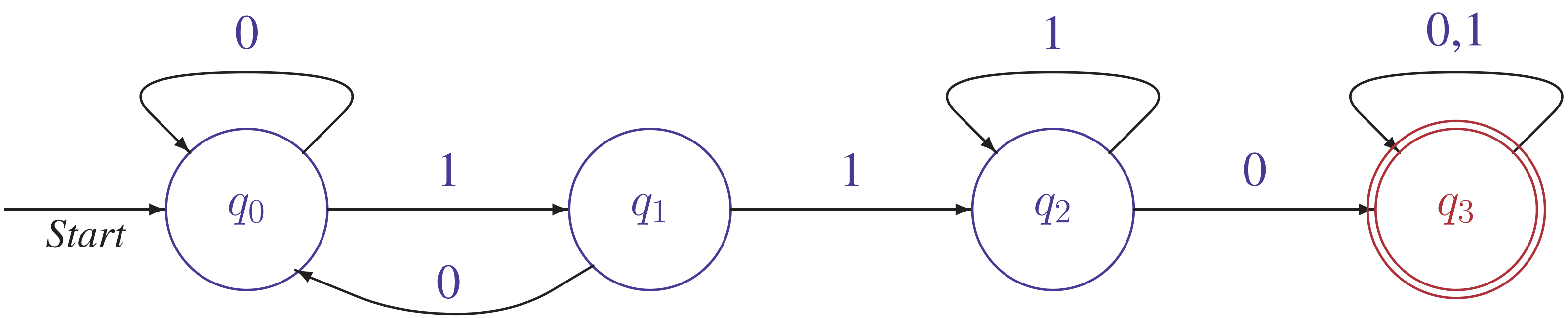
1. Arbeitsweise
2. Akzeptierte Sprache
3. Entwurf und Analyse
4. Automaten mit Ausgabe

Erkennung von Wörtern mit Automaten



- **Endliche Anzahl von Zuständen :** $\{q_0, q_1, q_2, q_3\}$
- **Ein Startzustand :** q_0
- **Regeln für Zustandsübergänge :** siehe Diagramm
- **Eingabealphabet:** $\{0, 1\}$
- **Ein oder mehrere akzeptierende Endzustände :** $\{q_3\}$

Endliche Automaten – mathematisch präzisiert



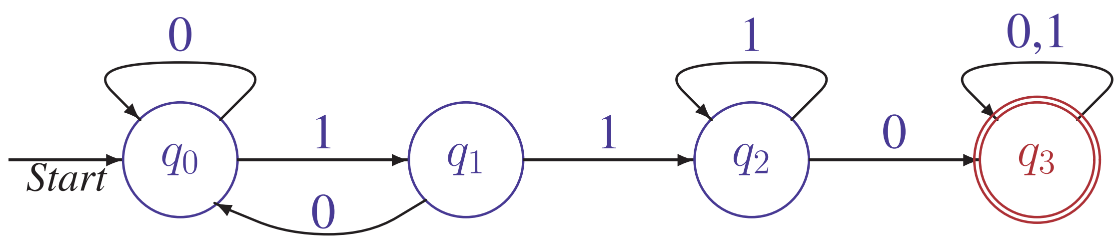
Ein **Deterministischer Endlicher Automat** (**DEA**)

ist ein 5-Tupel $A = (Q, \Sigma, \delta, q_0, F)$ mit

- Q nichtleere endliche **Zustandsmenge**
- Σ (endliches) **Eingabealphabet**
- $\delta : Q \times \Sigma \to Q$ **Zustandsüberführungsfunktion**
- $q_0 \in Q$ **Startzustand** (Anfangszustand)
- $F \subseteq Q$ Menge von **akzeptierenden Zuständen** (Endzustände) (Finale Zustände)

BESCHREIBUNG VON ENDLICHEN AUTOMATEN

- **Übergangsdiagramm**



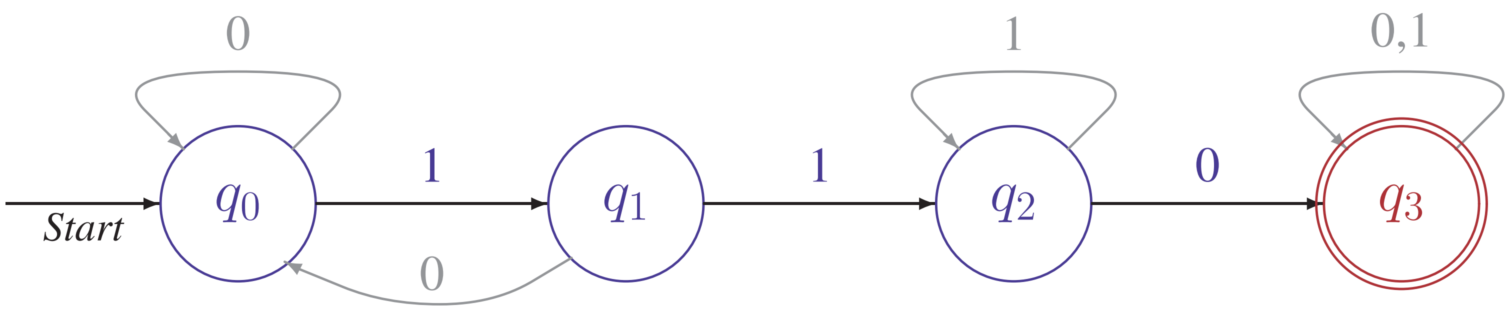
  – Jeder Zustand in Q wird durch einen Knoten (Kreis + Name) dargestellt
  – Ist $\delta(q, a) = p$, so verläuft eine Kante von q nach p mit Beschriftung a (mehrere Beschriftungen derselben Kante möglich)
  – q_0 wird durch einen mit *Start* beschrifteten Pfeil angezeigt
  – Endzustände in F werden durch doppelte Kreise gekennzeichnet
  – Σ meist implizit durch Diagramm bestimmt (Im Beispiel kommt nur 0,1 vor)

- **Übergangstabelle**

  – Tabellarische Darstellung der Funktion δ
  – Kennzeichnung von q_0 durch einen Pfeil
  – Kennzeichnung von F durch Sterne
  – Σ und Q meist implizit durch Tabelle bestimmt

	0	*1*
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_2
q_2	q_3	q_2
$* q_3$	q_3	q_3

Arbeitsweise von Endlichen Automaten



- **Anfangssituation**
  - Automat befindet sich im Startzustand q_0
- **Arbeitschritt**
  - Im Zustand q lese Eingabesymbol a (von externem Medium, Tastatur, Schalter, ...)
    Nur Symbole des Eingabealphabets können eingegeben werden
  - Bestimme $\delta(q,a){=}p$ und wechsele in neuen Zustand p (Echtzeitverarbeitung)
- **Terminierung und Ergebnis**
  - Automat stoppt, wenn Eingabewort $w = a_1..a_n$ komplett gelesen ist
    Der Automat befindet sich jetzt in einem Zustand q_n
  - Eingabe w wird akzeptiert, wenn $q_n \in F$, sonst wird w abgewiesen

Arbeitsweise von DEAs – mathematisch präzisiert

- **Erweiterte Überführungsfunktion $\hat{\delta}$: $Q\times\Sigma^*\to Q$**
  - Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts
  - Informal: $\hat{\delta}(q, w_1 w_2 ... w_n) = \delta(...(\delta(\delta(q, w_1),w_2),...),w_n)$
  - Mathematisch präzise Beschreibung benötigt induktive Definition

$$\hat{\delta}(q,w) = \begin{cases} q & \text{falls } w{=}\epsilon, \\ \delta(\hat{\delta}(q,v),a) & \text{falls } w{=}v\,a \;\text{ für ein } v\in\Sigma^*,\, a\in\Sigma \end{cases}$$

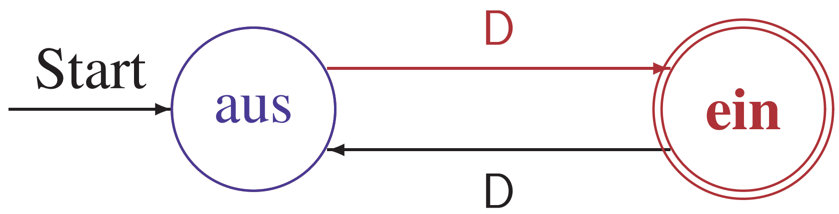
- **Von A akzeptierte Sprache**
  - Menge der Eingabewörter w, für die $\hat{\delta}(q_0, w)$ akzeptierender Zustand ist

$$L(A) = \{w\in\Sigma^* \mid \hat{\delta}(q_0,w)\in F\}$$

  - Auch: die von A erkannte Sprache

- **Reguläre Sprache**
  - Sprache, die von einem DEA A akzeptiert wird

Analyse der Sprache des Wechselschalters



- **Sprache: Eingaben, für die Automat eingeschaltet ist**
  – Teilmenge der Wörter über dem Alphabet $\Sigma = \{\mathsf{D}\}$

- **Automat *A* ist ein Wechselschalter**

  Nach n-fachem Drücken ist A aus(ein)geschaltet, wenn n (un)gerade ist

  Zwei Behauptungen zu Eigenschaften von A sind zu zeigen:

  – $B_1(n)$: Ist n gerade, so ist $\hat{\delta}(aus, \mathsf{D}^n) = aus$

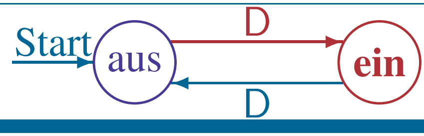
  – $B_2(n)$: Ist n ungerade, so ist $\hat{\delta}(aus, \mathsf{D}^n) = ein$

  Beweis: simultane Induktion analog zu Einheit 1, Folie 29/30, Details auf nächster Folie

- **Formale Beschreibung der Sprache von *A***

  $\boldsymbol{L(A)} = \{w \in \mathsf{D}^* \mid \hat{\delta}(aus,w) \in \{ein\}\} = \boldsymbol{\{}\mathsf{D}^{2i+1} | \boldsymbol{i} \in \mathbb{N}\boldsymbol{\}}$

Detaillierter Induktionsbeweis



Um zu zeigen, daß der Automat ein Wechselschalter ist, zeigen wir durch Induktion, daß für alle $n \in \mathbb{N}$ die folgenden beiden Aussagen gelten

$B_1(n)$**:** n gerade $\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^n) = \text{aus}$

$B_2(n)$**:** n ungerade $\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^n) = \text{ein}$

Induktionsanfang $n{=}0$**:**

$B_1(n)$**:** 0 ist gerade und $\hat{\delta}(\text{aus}, \text{D}^0) = \hat{\delta}(\text{aus}, \epsilon) = \text{aus}$, also gilt Aussage $B_1(n)$.

$B_2(n)$**:** 0 ist nicht ungerade und $\hat{\delta}(\text{aus}, \text{D}^0) \neq \text{ein}$, also gilt die Äquivalenz $B_2(n)$, da jeweils die rechte und linke Seite falsch ist.

Induktionsannahme: $B_1(m)$ und $B_2(m)$ seien für ein beliebiges $m \in \mathbb{N}$ gezeigt.

Induktionsschritt: Es sei $n = m{+}1$.

$B_1(n)$**:** Es ist $n = m{+}1$ gerade

$\Leftrightarrow$ m ist ungerade

$\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^m) = \text{ein}$ (Induktionsannahme $B_2(m)$)

$\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^n) = \text{aus}$

$B_2(n)$**:** Es ist $n = m{+}1$ ungerade

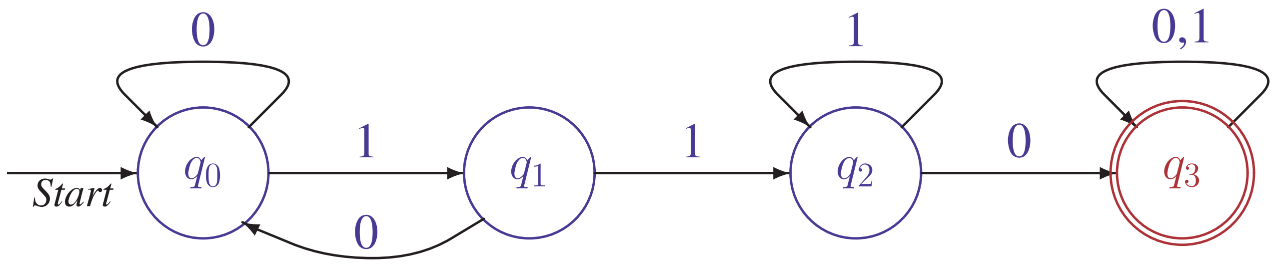
$\Leftrightarrow$ m ist gerade

$\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^m) = \text{aus}$ (Induktionsannahme $B_1(m)$)

$\Leftrightarrow$ $\hat{\delta}(\text{aus}, \text{D}^n) = \text{ein}$

Aufgrund des Induktionsprinzips gilt $B_1(n)$ und $B_2(n)$ für alle $n \in \mathbb{N}$

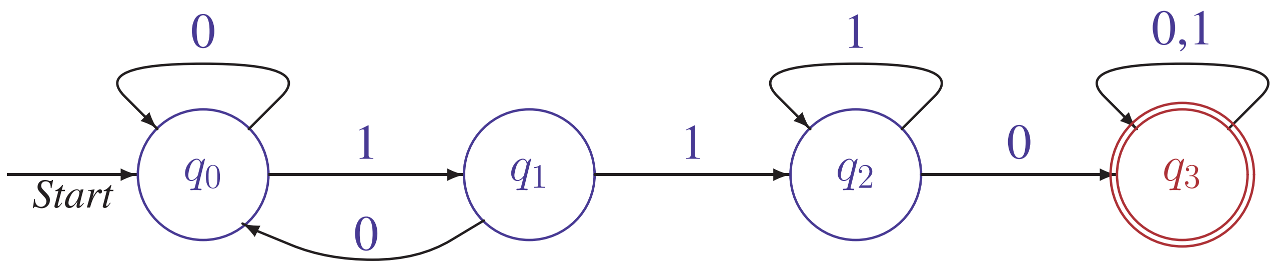
Analyse eines komplexeren Automaten



- **Teste Verhalten auf Eingabe 01110**
  - Automat startet in q_0 und bleibt bei Verarbeitung der 0 in q_0,
    wechselt bei Verarbeitung der 1 nach q_1,
    wechselt bei Verarbeitung der 1 nach q_2,
    bleibt bei Verarbeitung der 1 in q_2,
    wechselt bei Verarbeitung der 0 nach q_3
  - nach Verarbeitung ist der Automat in einem akkzeptierenden Zustand
    01110 gehört zur Sprache des Automaten

- **Formalere Analyse von 01010 mit erweiterter Überführungsfunktion**
  - $\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0) = \delta(q_0, 0) = q_0$
  - $\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1) = \delta(q_0, 1) = q_1$
  - $\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0) = \delta(q_1, 0) = q_0$
  - $\hat{\delta}(q_0, 0101) = \delta(\hat{\delta}(q_0, 010), 1) = \delta(q_0, 1) = q_1$
  - $\hat{\delta}(q_0, 01010) = \delta(\hat{\delta}(q_0, 0101), 0) = \delta(q_1, 0) = q_0 \notin F$ $\boxed{01010 \notin \boldsymbol{L(A)}}$

Analyse eines komplexeren Automaten II



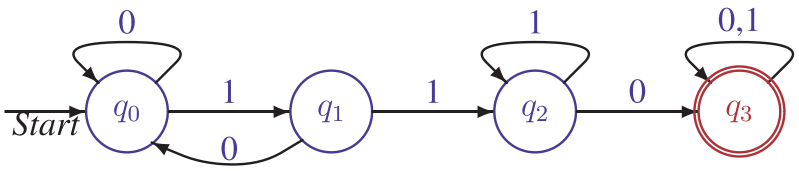
- **Sprache: Eingaben, die den Zustand q_3 erreichen**
  - Beobachtung: Letztes Symbol muß 0 sein, davor mindestens zwei 1
  - Vermutung: Menge der Wörter, die 110 als Teilwort enthalten

    Formale Beschreibung: $L(A) = \{w \in \{0,1\}^* \mid \exists u,v \in \{0,1\}^*.\ w{=}u110v\}$
  - Zu beweisen ist also: $\hat{\delta}(q_0, w) \in \{q_3\} \Leftrightarrow \exists u,v \in \{0,1\}^*.\ w{=}u110v$

- **Beweis zeigt, welche Wörter welchen Zustand erreichen**

  Beweise vier Eigenschaften durch simultane Induktion: (nächste Folie)
  - $B_3(w)$: $\hat{\delta}(q_0, w){=}q_3 \Leftrightarrow \exists u,v \in \{0,1\}^*.\ w{=}u110v$
  - $B_2(w)$: $\hat{\delta}(q_0, w){=}q_2 \Leftrightarrow \exists u \in \{0,1\}^*.\ w{=}u11 \wedge \forall u,v \in \{0,1\}^*.\ w{\neq}u110v$
  - $B_1(w)$: $\hat{\delta}(q_0, w){=}q_1 \Leftrightarrow \exists u \in \{0,1\}^*.\ w{=}u1 \wedge \forall u,v \in \{0,1\}^*.\ w{\neq}u110v \wedge w{\neq}u11$
  - $B_0(w)$: $\hat{\delta}(q_0, w){=}q_0 \Leftrightarrow \forall u,v \in \{0,1\}^*.\ w{\neq}u110v \wedge w{\neq}u11 \wedge w{\neq}u1$

Induktionsbeweis der vier Eigenschaften



Beweise durch simultane (strukturelle) Induktion:

– $B_3(w)$: $\hat\delta(q_0,w){=}q_3 \Leftrightarrow \exists u,v.\ w{=}u110v$
– $B_2(w)$: $\hat\delta(q_0,w){=}q_2 \Leftrightarrow \exists u.\ w{=}u11 \wedge \forall u,v.\ w{\neq}u110v$
– $B_1(w)$: $\hat\delta(q_0,w){=}q_1 \Leftrightarrow \exists u.\ w{=}u1 \wedge \forall u,v.\ w{\neq}u110v \wedge w{\neq}u11$
– $B_0(w)$: $\hat\delta(q_0,w){=}q_0 \Leftrightarrow \forall u,v.\ w{\neq}u110v \wedge w{\neq}u11 \wedge w{\neq}u1$

Induktionsanfang $w{=}\epsilon$**:**

$B_0(w)$**:** $\hat\delta(q_0,\epsilon){=}q_0$ gilt und $w{=}\epsilon$ hat keine der drei “verbotenen” Formen ✓

$B_1(w)$**:** $\hat\delta(q_0,\epsilon){=}q_1$ gilt nicht und $w{=}\epsilon$ endet nicht mit 1 ✓

$B_2(w)$**:** $\hat\delta(q_0,\epsilon){=}q_2$ gilt nicht und $w{=}\epsilon$ endet nicht mit 11 ✓

$B_3(w)$**:** $\hat\delta(q_0,\epsilon){=}q_3$ gilt nicht und $w{=}\epsilon$ enthält nicht das Teilwort 110 ✓

Induktionsannahme: die Aussagen $B_0(w')$–$B_3(w')$ seien für ein beliebiges $w' \in \{0,1\}^*$ gezeigt

Induktionsschritt: Es sei $\boldsymbol{w = w'a}$ für ein beliebiges $a \in \{0,1\}$.

$B_0(w)$**:** $\hat\delta(q_0,w){=}q_0 \ \Leftrightarrow\ (\hat\delta(q_0,w'){=}q_0 \wedge a{=}0) \vee (\hat\delta(q_0,w'){=}q_1 \wedge a{=}0)$
$\Leftrightarrow\ \forall u,v \in \{0,1\}^*.\ w{\neq}u110v \wedge w{\neq}u11 \wedge w{\neq}u1$ mit $B_0(w')$, $B_1(w')$ ✓

$B_1(w)$**:** $\hat\delta(q_0,w){=}q_1 \ \Leftrightarrow\ (\hat\delta(q_0,w'){=}q_0 \wedge a = 1)$
$\Leftrightarrow\ \exists u \in \{0,1\}^*.\ w{=}u1 \wedge \forall u,v \in \{0,1\}^*.\ w{\neq}u110v$ mit $B_0(w')$ ✓

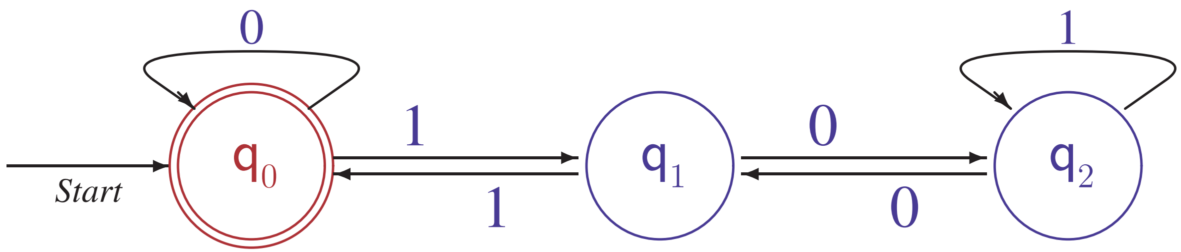
$B_2(w)$**:** $\hat\delta(q_0,w){=}q_2 \ \Leftrightarrow\ (\hat\delta(q_0,w'){=}q_1 \wedge a = 1) \vee (\hat\delta(q_0,w'){=}q_2 \wedge a{=}1)$
$\Leftrightarrow\ \exists u \in \{0,1\}^*.\ w{=}u11 \wedge \forall u,v \in \{0,1\}^*.\ w{\neq}u110v$ mit $B_1(w')$, $B_2(w')$ ✓

$B_3(w)$**:** $\hat\delta(q_0,w){=}q_2 \ \Leftrightarrow\ (\hat\delta(q_0,w'){=}q_2 \wedge a = 0) \vee (\hat\delta(q_0,w'){=}q_3)$
$\Leftrightarrow\ \exists u,v \in \{0,1\}^*.\ w{\neq}u110v$ mit $B_2(w')$, $B_3(w')$ ✓

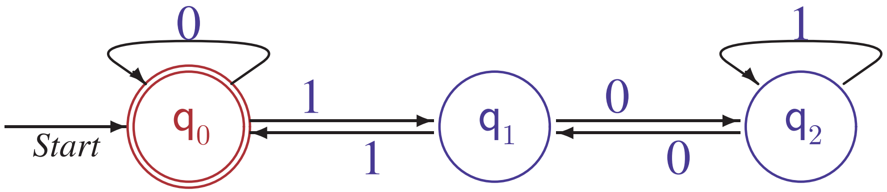
Aufgrund des Induktionsprinzips gilt $B_0(w) - B_3(w)$ für alle $w \in \{0,1\}^*$.

Erkenne, ob eine Binärzahl durch 3 teilbar ist

- **Binärzahl wird von links nach rechts gelesen**
  - Eingabewort $w = w_0..w_n$ entspricht der Zahl $n = r_b(w) = \Sigma_{j=0}^{n} w_j{\cdot}2^{n-j}$
- **Konstruktion des Automaten aus induktiver Beweisidee**
  - Das Eingabewort $w = \epsilon$ entspricht der Zahl $n = 0$
  - Entspricht v der Zahl i, dann entspricht $w = va$ der Zahl $n = 2i{+}a$
  - Die Zahl n ist durch 3 teilbar wenn $n \bmod 3 = 0$ (Divisionsrest)
  - Wir wissen $2i{+}a \bmod 3 = 2(i \bmod 3) + a \bmod 3$
- **Drei Zustände sind erforderlich**
  - Zustand $q_i \ \hat{=}\ $ bisher gelesene Binärzahl hat Divisionsrest i (modulo 3)
  - Zustandsübergänge erhalten "Bedeutung": $\delta(q_i, a) = q_{2i+a \bmod 3}$
  - Resultierender DEA über Alphabet $\Sigma = \{0, 1\}$



ZEIGE $L(A) = \{w \mid r_b(w) \bmod 3 = 0\}$



- **Zeige durch simultane strukturelle Induktion über w**
  – $B_j(w)$: $\hat{\delta}(q_0, w){=}q_j \Leftrightarrow r_b(w) \bmod 3 = j$ für $j \in \{0, 1, 2\}$
- **Induktionsanfang $w{=}\epsilon$:**
  – Es ist $\hat{\delta}(q_0, \epsilon){=}q_0$ und $r_b(\epsilon) \bmod 3 = 0$
  – Damit gilt $B_0(w)$ und auch $B_1(w)$ und $B_2(w)$ (beide Seiten von $\Leftrightarrow$ sind falsch) $\surd$
- **Induktionsannahme: $B_j(w')$ sei gezeigt für $w' \in \Sigma^*$ und $j \in \{0, 1, 2\}$**
- **Induktionsschritt: sei $w = w'a$ für ein beliebiges $a \in \Sigma$**

$$\begin{aligned}
&\hat{\delta}(q_0, w){=}q_j \\
\Leftrightarrow\ &\exists i.\hat{\delta}(q_0, w'){=}q_i \wedge \delta(q_i, a) = q_j \\
\Leftrightarrow\ &\exists i.r_b(w') \bmod 3 = i \wedge j = 2i{+}a \bmod 3 \\
\Leftrightarrow\ &r_b(w) \bmod 3 \ = \ 2r_b(w'){+}a \bmod 3 \ = \ 2(r_b(w') \bmod 3){+}a \bmod 3 \ = \ j \quad \surd
\end{aligned}$$

- **Damit gilt $B_j(w)$ für alle $w \in \Sigma^*$ und $j \in \{0, 1, 2\}$ und es folgt**
  $L(A) = \{w \mid \hat{\delta}(q_0, w){=}q_0\} = \{w \mid r_b(w) \bmod 3 = 0\}$

ALTERNATIVE BESCHREIBUNG DER ARBEITSWEISE VON DEAs

- **Konfiguration: 'Gesamtzustand' von Automaten**
  - Mehr als $q \in Q$: auch die noch unverarbeitete Eingabe zählt
  - Formal dargestellt als Tupel $\boldsymbol{K} = (\boldsymbol{q},\boldsymbol{w}) \in Q \times \Sigma^*$
- **Konfigurationsübergangsrelation $\vdash^*$**
  - Wechsel zwischen Konfigurationen durch Abarbeitung von Wörtern
  - $(\boldsymbol{q},\boldsymbol{aw}) \vdash (\boldsymbol{p},\boldsymbol{w})$, falls $\delta(q, a) = p$
  - $\boldsymbol{K_1} \vdash^* \boldsymbol{K_2}$, falls $K_1 = K_2$ oder

    es gibt eine Konfiguration K mit $K_1 \vdash^* K$ und $K \vdash K_2$
- **Akzeptierte Sprache**
  - Menge der Eingaben, für die $\vdash^*$ zu akzeptierendem Zustand führt

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

Für DEAs weniger intuitiv, aber leichter zu verallgemeinern

DEA FÜR $L = \{w \in \{0,1\}^* \mid |w|$ *ist durch 5 teilbar* $\}$

Codiere Anzahl der gelesenen Symbole im Zustand

– $|w|$ ist durch 5 teilbar gdw. $|w| \mod 5 = 0$

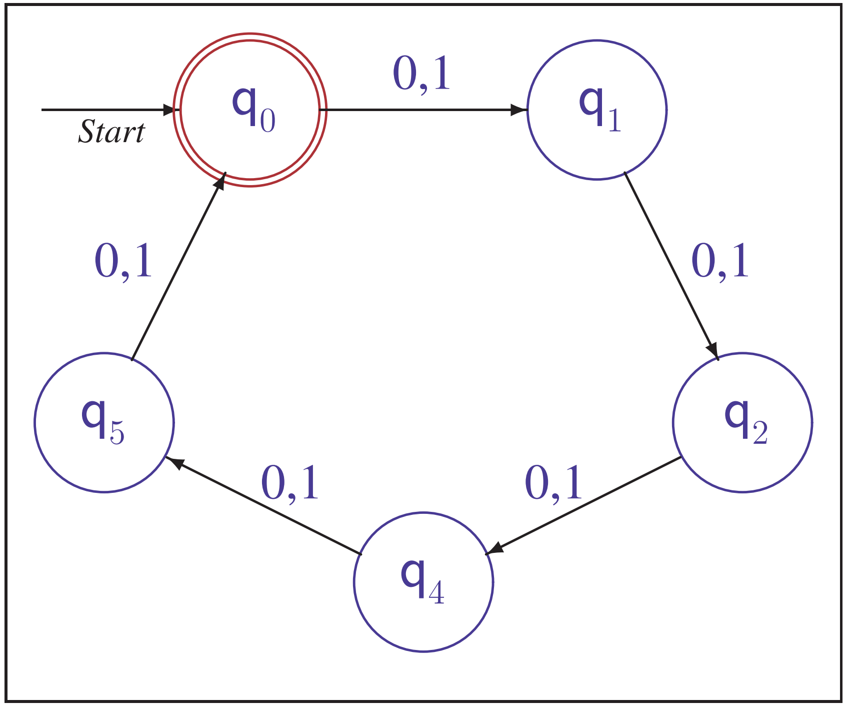
– Zustand $q_i \hat{=}$ bisher gelesene Wortlänge hat Divisionsrest i (modulo 5)

q_0 wird Start- und Endzustand

– Verarbeiten eines Symbols erhöht Divisionsrest

Formal: $\delta(q_i, a) = q_{i+1 \mod 5}$

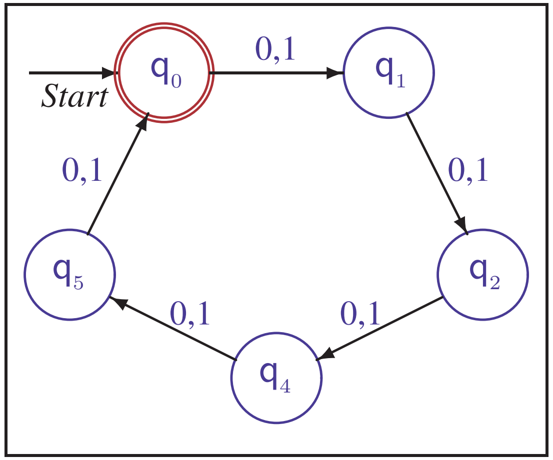
– Zugehöriges Überführungsdiagramm



– Korrektheit beweisbar durch gegenseitige strukturelle Induktion

(nächste Folie)

Korrektheitsbeweis mit Konfigurationen



- **Zeige simultan für alle Wörter $w, v \in \{0,1\}^*$:**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; |w| \bmod 5 = 0$
  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\; |w| \bmod 5 = 1$
  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\; |w| \bmod 5 = 2$
  $B_3(w)$: $(q_0, w\,v) \vdash^* (q_3, v) \;\Leftrightarrow\; |w| \bmod 5 = 3$
  $B_4(w)$: $(q_0, w\,v) \vdash^* (q_4, v) \;\Leftrightarrow\; |w| \bmod 5 = 4$

  Kurz: $\boldsymbol{B_i(w)}$: $\boldsymbol{(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\; |w| \bmod 5 = i}$

- **Basisfall: Sei $w = \epsilon$**
  – Per Definition ist $(q_0, \epsilon v) \vdash^* (q_0, \epsilon v)$ und $|\epsilon| = 0$, also gilt $B_0(\epsilon)..B_4(\epsilon)$ ✓

- **Annahme: $B_0(w')..B_4(w')$ sei bewiesen für ein $w' \in \{0,1\}^*$**

- **Schrittfall: Sei $w = w'a$ für ein $a \in \{0,1\}$**

  $B_i(w)$, $\Rightarrow$: Es gelte $(q_0, w\,v) \vdash^* (q_i, v)$
  Dann gilt $(q_0, w'\,a\,v) \vdash^* (q_j, a\,v) \vdash (q_i, v)$ für ein j und $i = j{+}1 \bmod 5$
  Wegen $B_j(w')$ folgt $|w'| \bmod 5 = j$ und $|w| \bmod 5 = j{+}1 \bmod 5 = i$

  $B_i(w)$, $\Leftarrow$: Es gelte $|w| \bmod 5 = i$ (Gegenrichtung kehrt Argument um)
  Dann gilt $|w'| \bmod 5 = j$ für ein j und $i = |w'|{+}1 \bmod 5 = j{+}1 \bmod 5$
  Wegen $B_j(w')$ folgt $(q_0, w'\,a\,v) \vdash^* (q_j, a\,v) \vdash (q_i, v)$

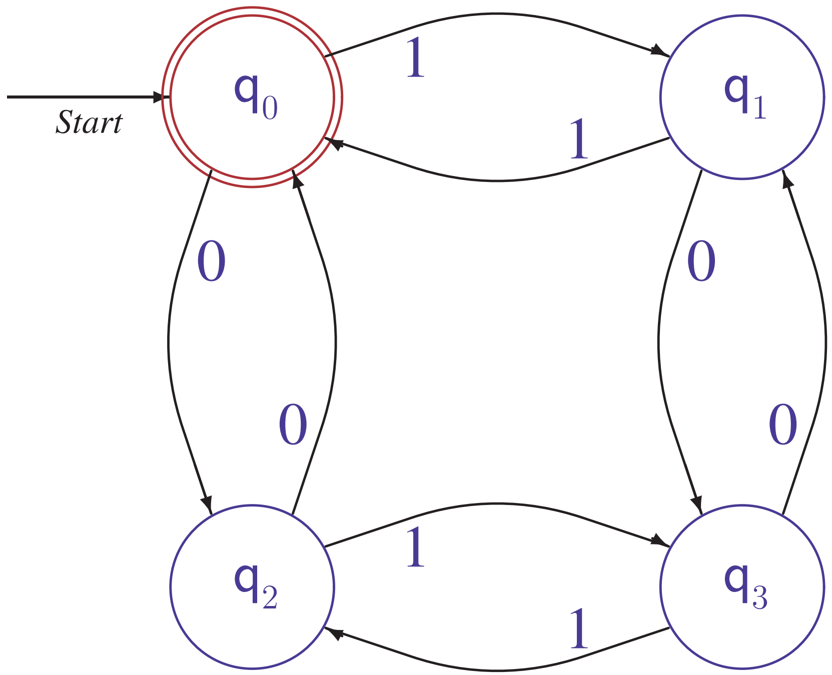
- **Es folgt $w \in L(A) \;\Leftrightarrow\; (q_0, w) \vdash^* (q_0, \epsilon) \;\Leftrightarrow\; |w| \bmod 5 = 0 \;\Leftrightarrow\; w \in L$**

DEA für $L = \{w \in \{0,1\}^* \mid \textit{w enthält gerade Anzahl von 0 und 1}\}$

Codiere Anzahl der gelesenen 0/1 im Zustand

$q_0 \hat{=}$ (gerade,gerade) $q_1 \hat{=}$ (gerade,ungerade)

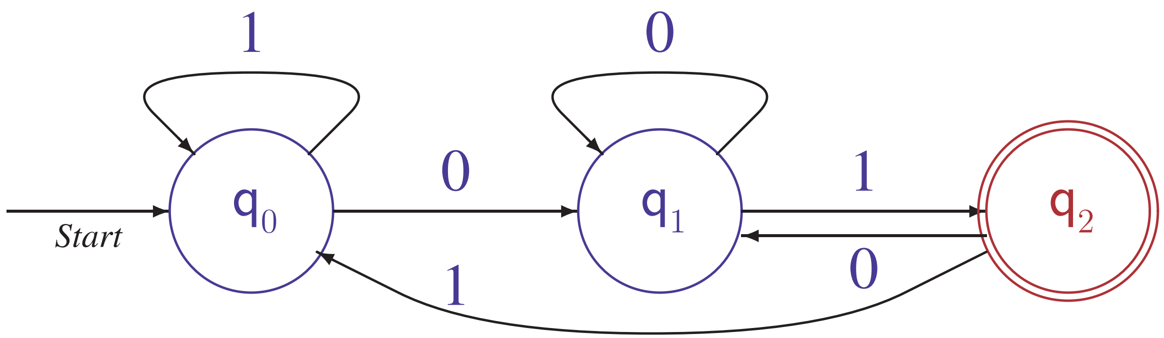
$q_2 \hat{=}$ (ungerade,gerade) $q_3 \hat{=}$ (ungerade,ungerade)



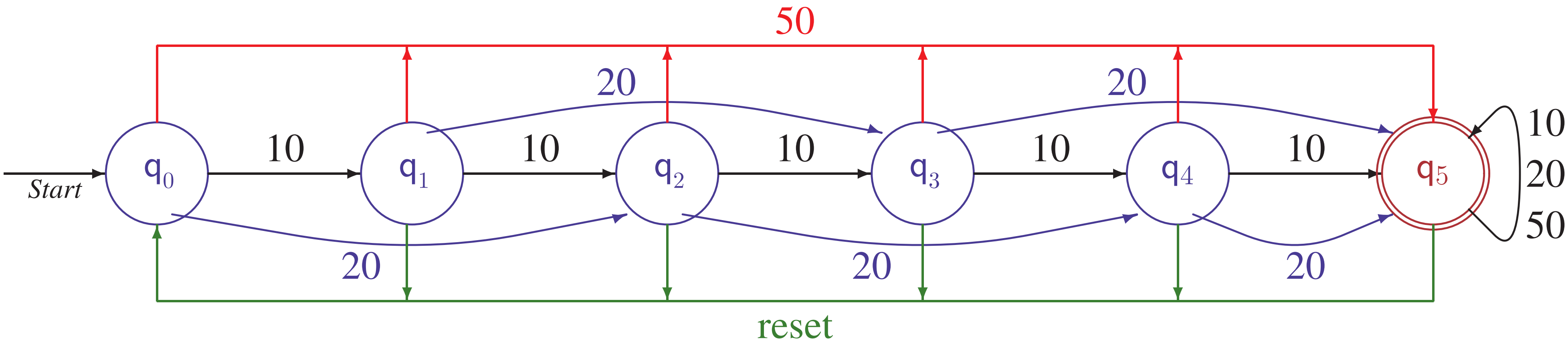
Korrektheit: gegenseitige strukturelle Induktion (Anhang, Folie 22)

WEITERE BEISPIELE ENDLICHER AUTOMATEN

- **Erkenne Strings, die mit `01` enden**

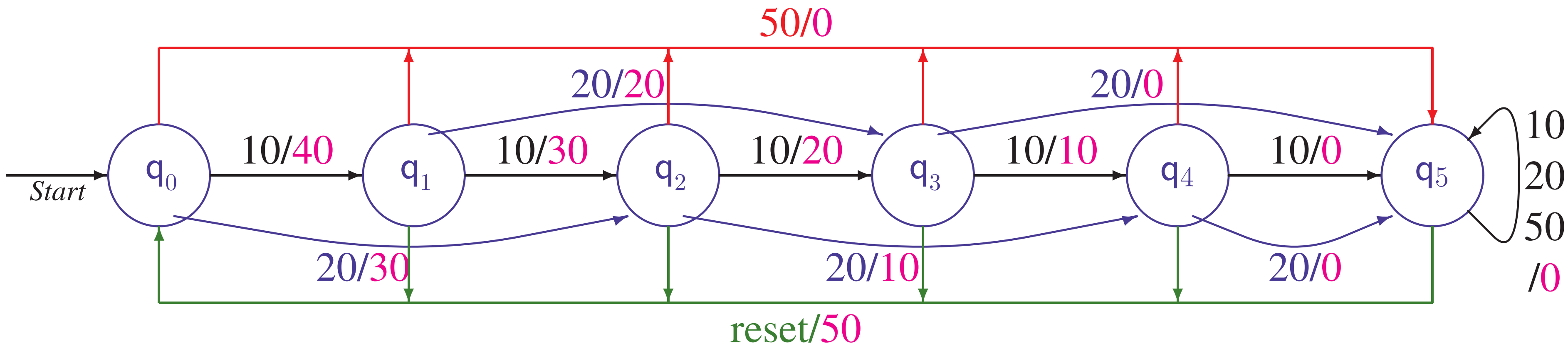


- **50c Kaffeeautomat**
  - Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



Endliche Automaten mit Ausgabefunktion

- **50c Kaffeeautomat mit Restbetragsanzeige**



  – Münzeinwurf führt zu Zustandsänderung und erzeugt Ausgabe

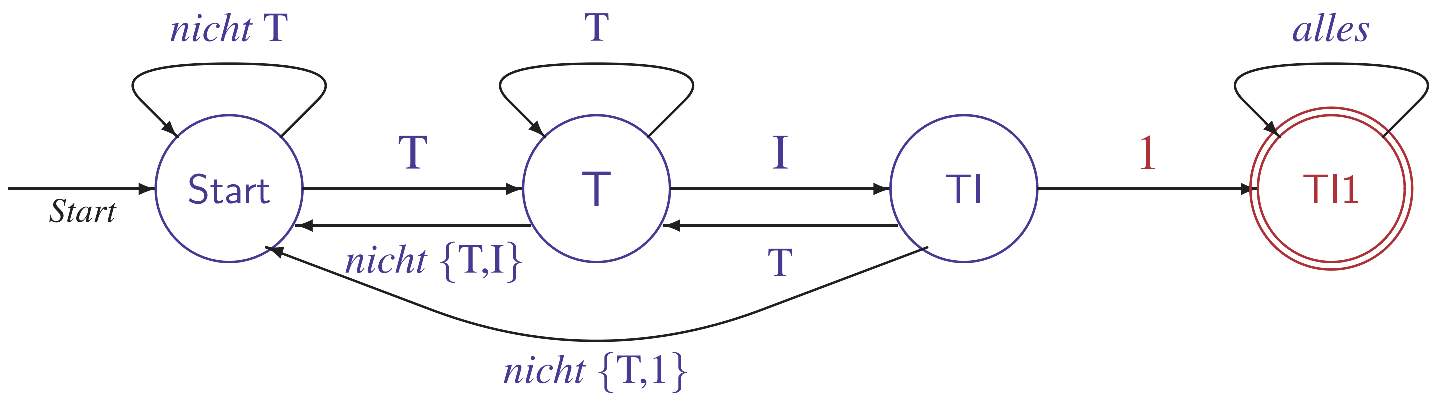
- **Formalisierungen von Automaten mit Ausgabe**
  – Mealy-Automaten: Ausgabefunktion abhängig von Eingabe & Zustand
  – Moore-Automaten: Ausgabefunktion nur von Zustand abhängig

- **Automaten mit Ausgabe sind keine echte Erweiterung**
  – Mealy- und Moore-Automaten sind äquivalent
  – DEAs können Mealy-/Moore-Automaten simulieren und umgekehrt

Mehr dazu im Anhang

ANHANG

Analyse eines "TI1"-Automaten



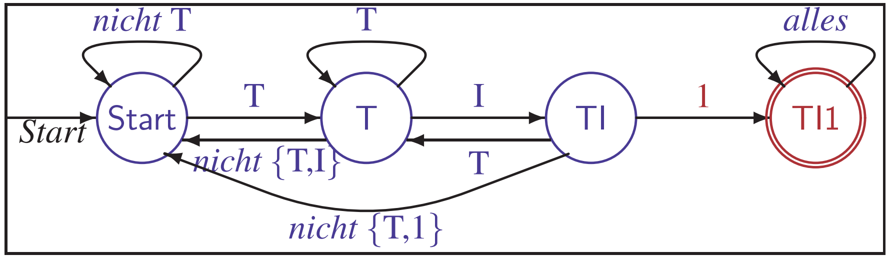
- **Sprache: Eingaben, die den Zustand TI1 erreichen**
  - Vermutung: Menge der Wörter, die TI1 als Teilwort enthalten
  - Formale Beschreibung: $L(A) = \{w \in \Sigma^* \mid \exists u, v \in \Sigma^*.\ w{=}u\texttt{TI1}v\}$
  - Zu beweisen ist also: $\hat{\delta}(\mathsf{Start}, w) \in \{\mathsf{TI1}\} \Leftrightarrow \exists u, v \in \Sigma^*.\ w{=}u\texttt{TI1}v$

- **Beweis zeigt, welche Wörter welchen Zustand erreichen**

  Beweise drei Eigenschaften durch simultane Induktion: (nächste Folie)
  - $B_3(w)$: $\hat{\delta}(\mathsf{Start}, w){=}\mathsf{TI} \Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\texttt{TI} \wedge \forall u, v \in \Sigma^*.\ w{\neq}u\texttt{TI1}v$
  - $B_2(w)$: $\hat{\delta}(\mathsf{Start}, w){=}\mathsf{T} \Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\texttt{T} \wedge \forall u, v \in \Sigma^*.\ w{\neq}u\texttt{TI1}v$
  - $B_1(w)$: $\hat{\delta}(\mathsf{Start}, w){=}\mathsf{Start} \Leftrightarrow \forall u, v \in \Sigma^*.\ w{\neq}u\texttt{TI1}v \wedge w{\neq}u\texttt{TI} \wedge w{\neq}u\texttt{T}$

  Die Behauptung folgt dann aus B_3 (und einer Induktion innerhalb von TI1)

Beweis der Sprache des "TI1"-Automaten



Beweise durch simultane (strukturelle) Induktion:

$B_3(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{TI} \Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\mathtt{TI} \wedge \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v$

$B_2(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{T} \Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\mathtt{T} \wedge \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v$

$B_1(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{Start} \Leftrightarrow \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v \wedge w{\neq}u\mathtt{TI} \wedge w{\neq}u\mathtt{T}$

Induktionsanfang $w{=}\epsilon$:

$B_1(w)$: $\hat\delta(\mathsf{Start}, \epsilon){=}\mathsf{Start}$ gilt und $w{=}\epsilon$ hat keine der drei "verbotenen" Formen ✓

$B_2(w)$: $\hat\delta(\mathsf{Start}, \epsilon){=}\mathsf{T}$ gilt nicht und $w{=}\epsilon$ endet nicht mit $\mathtt{T}$ ✓

$B_3(w)$: $\hat\delta(\mathsf{Start}, \epsilon){=}\mathsf{TI}$ gilt nicht und $w{=}\epsilon$ endet nicht mit $\mathtt{TI}$ ✓

Induktionsannahme: die Aussagen $B_1(w')$–$B_3(w')$ seien für ein beliebiges $w' \in \Sigma^*$ gezeigt.

Induktionsschritt: Es sei $w = w'a$ für ein beliebiges $a \in \Sigma$.

$B_1(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{Start}$

$\Leftrightarrow (\hat\delta(\mathsf{Start}, w'){=}\mathsf{Start} \wedge a{\neq}\mathtt{T}) \vee (\hat\delta(\mathsf{Start}, w'){=}\mathsf{T} \wedge a \notin \{\mathtt{T}, \mathtt{I}\}) \vee (\hat\delta(\mathsf{Start}, w'){=}\mathsf{TI} \wedge a \notin \{\mathtt{T}, \mathtt{1}\})$

$\Leftrightarrow \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v \wedge w{\neq}u\mathtt{TI} \wedge w{\neq}u\mathtt{T}$ mit $B_1(w')$, $B_2(w')$, $B_3(w')$ ✓

$B_2(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{T}$

$\Leftrightarrow (\hat\delta(\mathsf{Start}, w'){=}\mathsf{Start} \wedge a = \mathtt{T}) \vee (\hat\delta(\mathsf{Start}, w'){=}\mathsf{T} \wedge a = \mathtt{T}) \vee (\hat\delta(\mathsf{Start}, w'){=}\mathsf{TI} \wedge a = \mathtt{T})$

$\Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\mathtt{T} \wedge \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v$ mit $B_1(w')$, $B_2(w')$, $B_3(w')$ ✓

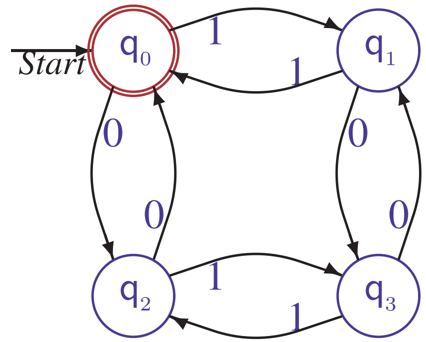
$B_3(w)$: $\hat\delta(\mathsf{Start}, w){=}\mathsf{TI}$

$\Leftrightarrow (\hat\delta(\mathsf{Start}, w'){=}\mathsf{T} \wedge a = \mathtt{I})$

$\Leftrightarrow \exists u \in \Sigma^*.\ w{=}u\mathtt{TI} \wedge \forall u,v \in \Sigma^*.\ w{\neq}u\mathtt{TI1}v$ mit $B_2(w')$ ✓

Aufgrund des Induktionsprinzips gilt $B_1(w)$, $B_2(w)$ und $B_3(w)$ für alle $w \in \Sigma^*$.

Korrektheitsbeweis für den Automat von Folie 16



- **Zeige simultan für alle Wörter $w, v \in \{0,1\}^*$:**

  $B_0(w)$: $(q_0, w\,v) \vdash^* (q_0, v) \;\Leftrightarrow\;$ es gilt $g_0(w)$ und $g_1(w)$

  $B_1(w)$: $(q_0, w\,v) \vdash^* (q_1, v) \;\Leftrightarrow\;$ es gilt $g_0(w)$ und $u_1(w)$

  $B_2(w)$: $(q_0, w\,v) \vdash^* (q_2, v) \;\Leftrightarrow\;$ es gilt $u_0(w)$ und $g_1(w)$

  $B_3(w)$: $(q_0, w\,v) \vdash^* (q_3, v) \;\Leftrightarrow\;$ es gilt $u_0(w)$ und $u_1(w)$

  $g_0(w) \mathrel{\hat{=}} w$ hat gerade Anzahl von Nullen, $u_0(w) \mathrel{\hat{=}} w$ hat ungerade Anzahl von Nullen, ...

- **Basisfall: Sei $w = \epsilon$**
  - Per Definition ist $(q_0, \epsilon v) \vdash^* (q_0, \epsilon v)$, $g_0(\epsilon)$ und $g_1(\epsilon)$, also $B_0(\epsilon)..B_3(\epsilon)$ $\surd$

- **Annahme: $B_0(w')..B_3(w')$ sei bewiesen für ein $w' \in \{0,1\}^*$**

- **Schrittfall: Sei $w = w'a$ für ein $a \in \{0,1\}$**

  $B_0(w)$, $\Rightarrow$: Es gelte $(q_0, w\,v) \vdash^* (q_0, v)$.

  Dann gilt $(q_0, w'\,a\,v) \vdash^* (p, a\,v) \vdash (q_0, v)$ für einen Zustand p.

  Falls $a = 0$, dann ist $p = q_2$ und wegen $B_2(w')$ folgt $u_0(w')$ und $g_1(w')$.

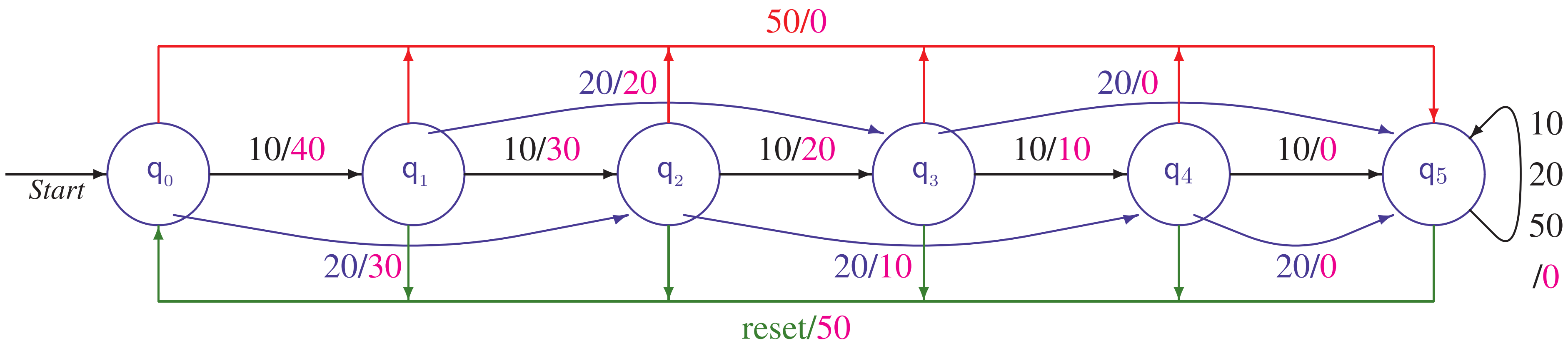
  Falls $a = 1$, dann ist $p = q_1$ und wegen $B_1(w')$ folgt $g_0(w')$ und $u_1(w')$.

  Für $w = w'a$ folgt somit $g_0(w)$ und $g_1(w)$. $\surd$

  Gegenrichtung durch Umkehrung des Arguments. $B_1(w), B_2(w), B_3(w)$ analog.

- **Es folgt $w \in L(A) \;\Leftrightarrow\; (q_0, w) \vdash^* (q_0, \epsilon) \;\Leftrightarrow\; g_0(w) \wedge g_1(w) \;\Leftrightarrow\; w \in L$**

Mealy-Automaten – mathematisch präzisiert



Ein **Mealy-Automat** ist ein 6-Tupel $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

- Q nichtleere endliche **Zustandsmenge**
- Σ (endliches) **Eingabealphabet**
- Δ (endliches) **Ausgabealphabet**
- $\delta{:}Q\times\Sigma \rightarrow Q$ **Zustandsüberführungsfunktion**
- $\lambda{:}Q\times\Sigma \rightarrow \Delta$ **Ausgabefunktion**
- $q_0 \in Q$ **Startzustand**

Arbeitsweise von Mealy-Automaten analog zu DEAs

- **Anfangssituation:** Automat im Startzustand q_0
- **Arbeitschritt**
  - Im Zustand q lese Eingabesymbol a,
  - Bestimme $\delta(q,a)=p$ und wechsele in neuen Zustand p
  - Bestimme $x = \lambda(q,a)$ und gebe dieses Symbol aus
- **Terminierung:** Eingabewort $w = a_1..a_n$ ist komplett gelesen
- **Ausgabewort:** Verkettung der ausgegebenen Symbole $x_1..x_n$

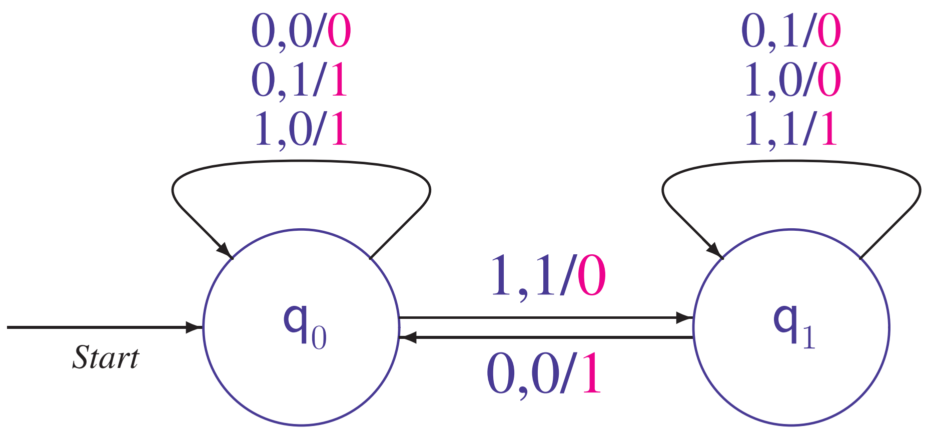
- **Erweiterte Ausgabefunktion $\hat{\lambda}$: $Q\times\Sigma^*\to\Delta^*$**
  - Schrittweise Erzeugung der Ausgabe mit Abarbeitung der Eingabe
  - Formal: Induktive Definition

$$\hat{\lambda}(q,w) = \begin{cases} \epsilon & \text{falls } w{=}\epsilon, \\ \hat{\lambda}(q,v)\circ\lambda(\hat{\delta}(q,v),a) & \text{falls } w{=}v\,a \text{ für ein } a\in\Sigma \end{cases}$$

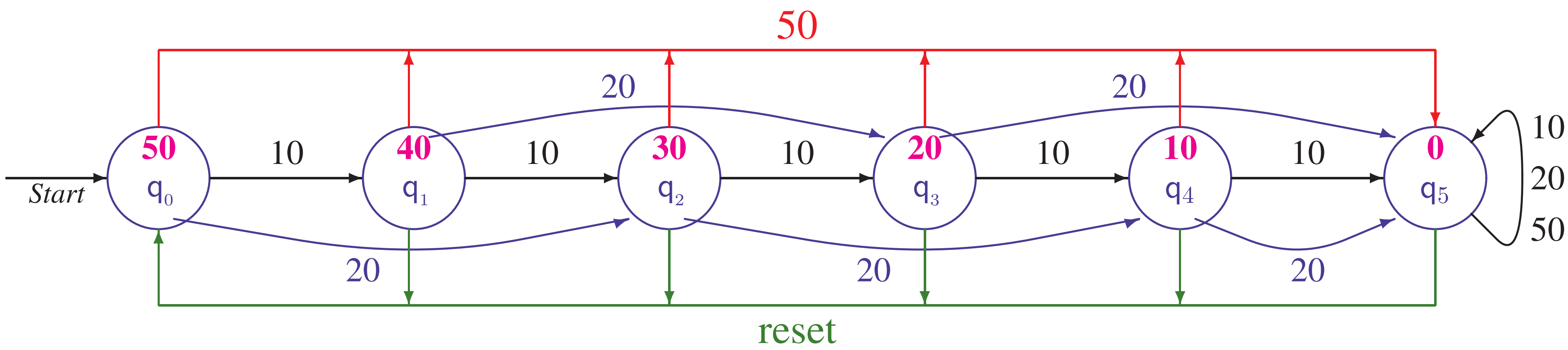
- **Von M berechnete Funktion: $f_M(w) = \hat{\lambda}(q_0,w)$**

Mealy-Automat für (inverse) Binäraddition

- **Addition von Bitpaaren von rechts nach links**
  - Eingabealphabet $\Sigma = \{0,1\}\times\{0,1\}$
  - Ausgabealphabet $\Delta = \{0,1\}$
- **Zwei Zustände sind erforderlich**
  - Zustand q_0: A kann Addition zweier Bits direkt ausführen
  - Zustand q_1: A hat bei Addition einen Übertrag zu berücksichtigen
- **Zugehöriger Mealy-Automat**



Moore-Automaten – mathematisch präzisiert



Ein **Moore-Automat** ist ein 6-Tupel $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$

- Q nichtleere endliche **Zustandsmenge**
- Σ (endliches) **Eingabealphabet**
- Δ (endliches) **Ausgabealphabet**
- $\delta{:}Q\times\Sigma \to Q$ **Zustandsüberführungsfunktion**
- $\lambda{:}Q \to \Delta$ **Ausgabefunktion**
- $q_0 \in Q$ **Startzustand**

ARBEITSWEISE VON MOORE-AUTOMATEN ANALOG ZU DEAS

- **Anfangssituation:** Automat im Startzustand q_0, Ausgabe $x_0 = \lambda(q_0)$
- **Arbeitschritt**
  - Im Zustand q lese Eingabesymbol a,
  - Bestimme $\delta(q,a)=p$ und wechsele in neuen Zustand p
  - Bestimme $x = \lambda(p)$ und gebe dieses Symbol aus
- **Terminierung:** Eingabewort $w = a_1..a_n$ ist komplett gelesen
- **Ausgabewort:** Verkettung der ausgegebenen Symbole $x_0x_1..x_n$

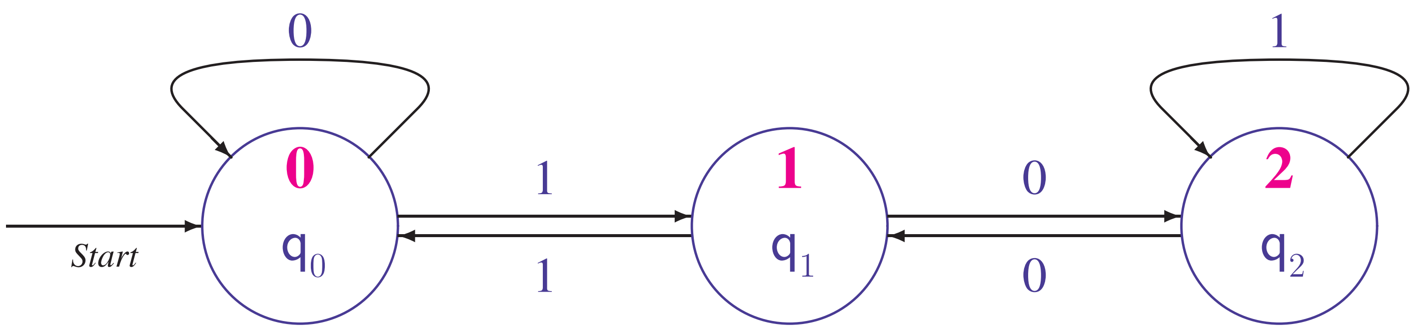
- **Erweiterte Ausgabefunktion $\hat{\lambda} : Q\times\Sigma^*\to\Delta^*$**
  - Schrittweise Erzeugung der Ausgabe mit Abarbeitung der Eingabe
  - Formal: Induktive Definition

$$\hat{\lambda}(q,w) = \begin{cases} \lambda(q) & \text{falls } w{=}\epsilon, \\ \hat{\lambda}(q,v)\circ\lambda(\hat{\delta}(q,v\,a)) & \text{falls } w{=}v\,a \text{ für ein } a\in\Sigma \end{cases}$$

- **Von M berechnete Funktion: $f_M(w) = \hat{\lambda}(q_0,w)$**

Moore-Automat für Divisionsrest

- **Eingabe einer Bitfolge von links nach rechts**
  - Bisher eingegebene Bitfolge ist Binärdarstellung einer Zahl n
  - Ausgabe ist jeweils "n mod 3"
  - Eingabealphabet $\Sigma = \{0, 1\}$, Ausgabealphabet $\Delta = \{0, 1, 2\}$
- **Drei Zustände sind erforderlich**
  - Zustand q_0: Bisheriger Divisionsrest ist 0 (Ausgabe 0)
  - Zustand q_1: Bisheriger Divisionsrest ist 1 (Ausgabe 1)
  - Zustand q_2: Bisheriger Divisionsrest ist 2 (Ausgabe 2)
  - Zustandsüberführungsregel $\delta(q_i, j) = q_{2*i+j \, mod \, 3}$
- **Zugehöriger Moore-Automat**



Moore-Automaten sind Äquivalent zu DEAs

Gegenseitige Simulation ist möglich

- **Jede Sprache L ist als Funktion beschreibbar**
  - $\chi_L(w) = \begin{cases} 1 & \text{falls } w \in L, \\ 0 & \text{sonst} \end{cases}$ charakteristische Funktion von L
  - Charakteristische Funktionen akzeptierter Sprachen sind berechenbar

  Satz: **L regulär $\Leftrightarrow$ χ_L "Moore-berechenbar"**

- **Jede Funktion f ist als Menge beschreibbar**
  - **graph(f)** $= \{(w, v) \mid f(w) = v\}$
  - **graph*(f)** $= \{(w_1, v_0, v_1)..(w_n, v_n) \mid f(w_1..w_n) = v_0..v_n\}$
  - DEAs können Graphen berechneter Funktionen akzeptieren

  Satz: **f Moore-berechenbar $\Leftrightarrow$ graph*(f) reguläre Sprache**

Beweis der Äquivalenz (Skizze)

- **L regulär $\Leftrightarrow$ χ_L "Moore-berechenbar"**
  - Zu $A = (Q, \Sigma, \delta, q_0, F)$ konstruiere $M = (Q, \Sigma, \{0{,}1\}, \delta, \lambda, q_0)$

    mit $\lambda(q) = \begin{cases} 1 & \text{falls } q \in F, \\ 0 & \text{sonst} \end{cases}$
  - Dann ist $w \in L(A)$ genau dann, wenn $f_M(w) = v1$ für ein $v \in \{0,1\}^*$

    $\chi_L(w)$ ist das letzte Ausgabesymbol von $f_M(w)$

- **f Moore-berechenbar $\Leftrightarrow$ graph$^*(f)$ regulär**
  - Zu $M = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ konstruiere $A = (Q \cup \{q_s, q_f\}, \Sigma', \delta', q_s, Q)$

    mit $\Sigma' = \Sigma \times (\Delta \cup \{\lambda(q_0)\} \times \Delta)$

    $$\delta'(q, (a, b)) = \begin{cases} \delta(q_0, a) & \text{falls } q{=}q_s, b = (\lambda(q_0), \lambda(\delta(q_0, a)), \\ \delta(q, a) & \text{falls } \lambda(\delta(q, a)) = b, \\ q_f & \text{sonst} \end{cases}$$
  - Dann $f_M(w_1..w_n) = v_0..v_n$ genau dann, wenn $(w_1, v_0, v_1)..(w_n, v_n) \in L(A)$

Mehr zu Automaten mit Ausgabe im Buch von Vossen & Witt