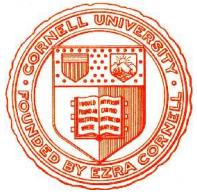


Theoretische Informatik I



Einheit 2

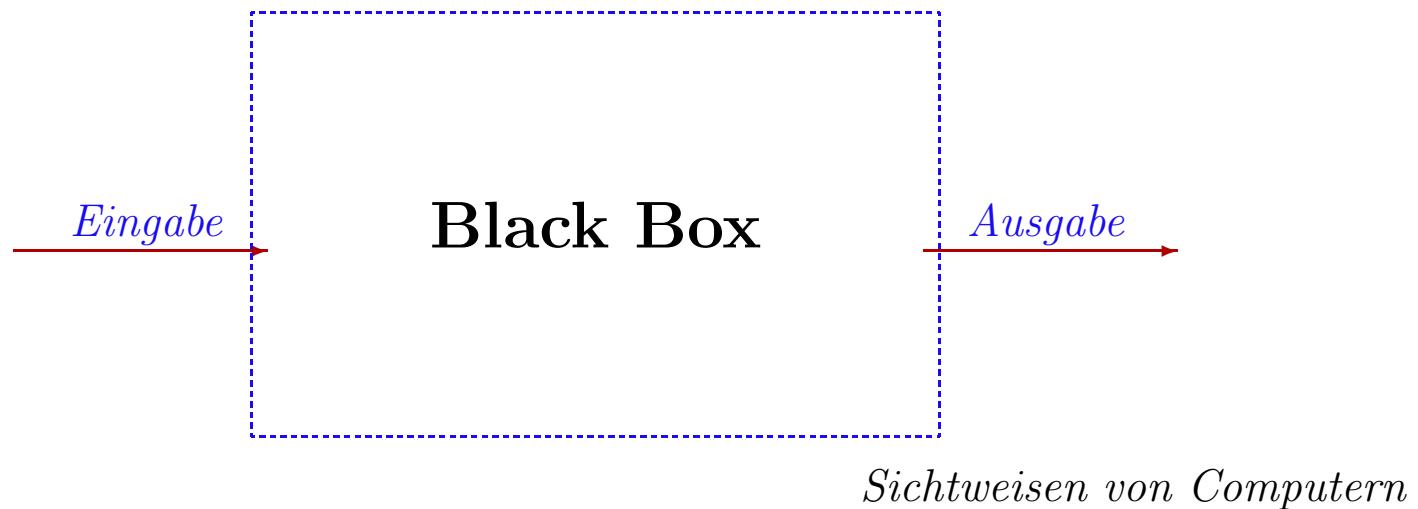
Endliche Automaten



1. Deterministische endliche Automaten
2. Nichtdeterministische endliche Automaten
3. Automaten mit ϵ -Übergängen

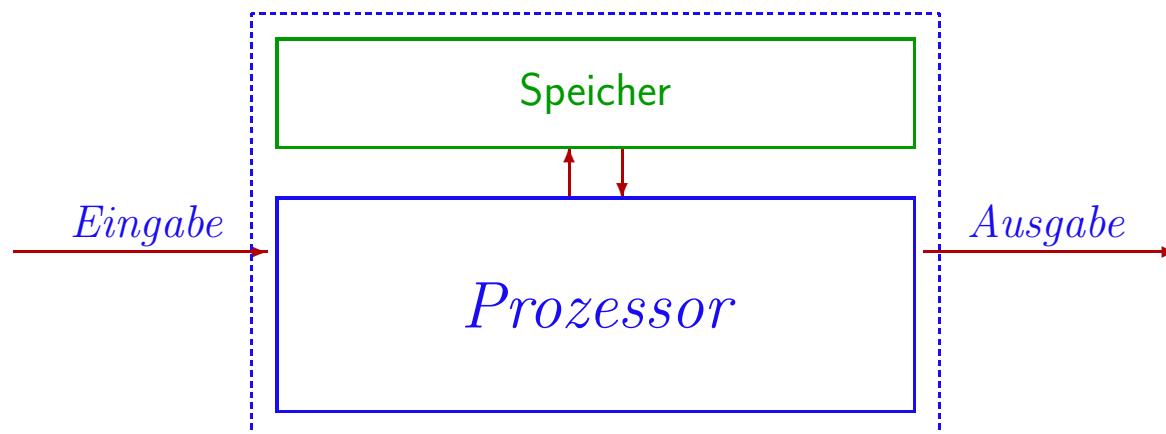
AUTOMATEN: DAS EINFACHSTE MASCHINENMODELL

Betrachtung von außen



AUTOMATEN: DAS EINFACHSTE MASCHINENMODELL

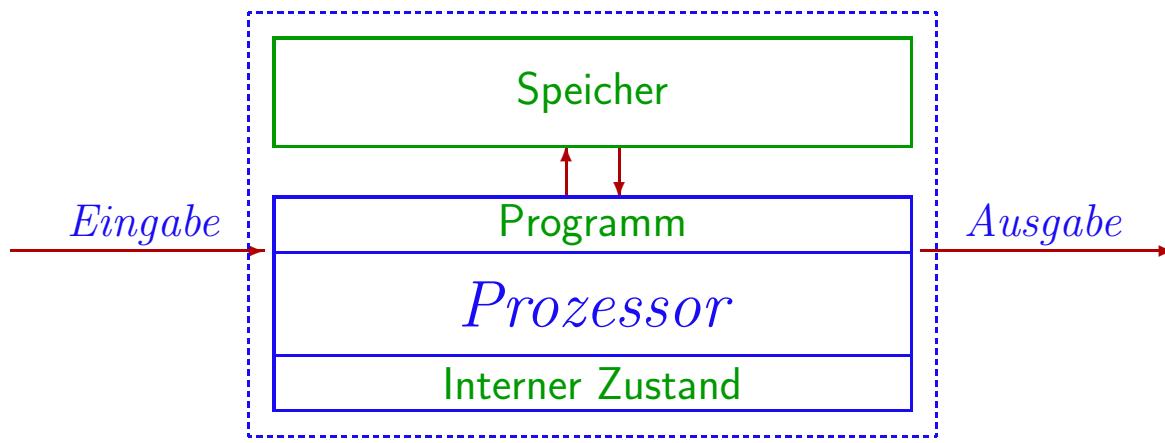
Abstrakte Maschinenarchitektur



Sichtweisen von Computern

AUTOMATEN: DAS EINFACHSTE MASCHINENMODELL

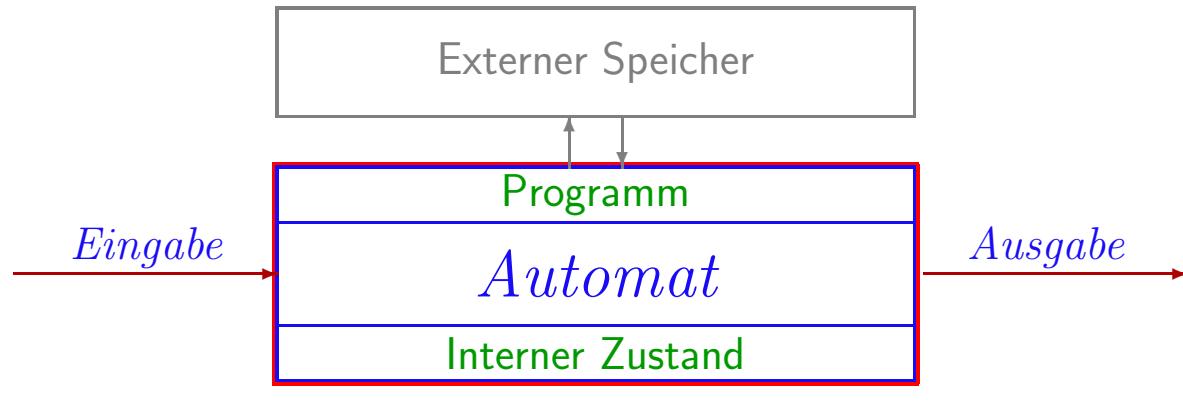
Details des Berechnungsmoduls



Sichtweisen von Computern

AUTOMATEN: DAS EINFACHSTE MASCHINENMODELL

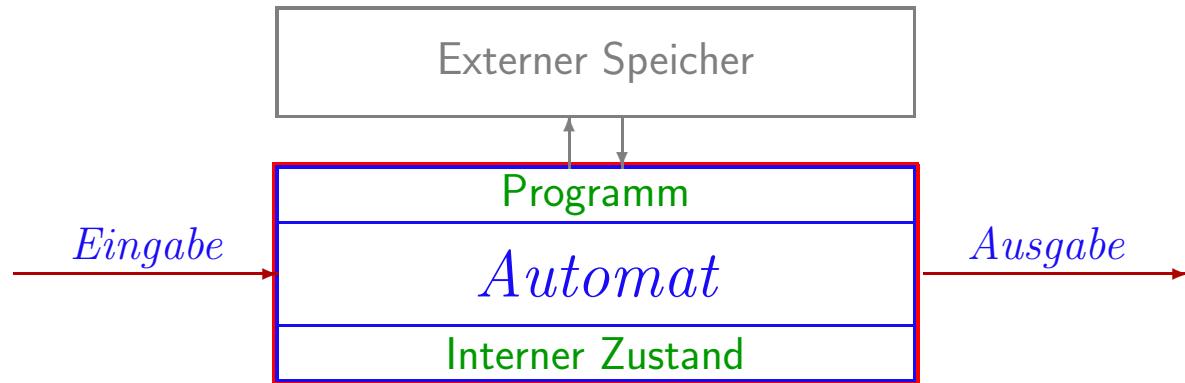
Aus der Sicht des Berechnungsmoduls



Sichtweisen von Computern

- **Automaten stehen im Kern jeder Berechnung**
 - Schnelle, direkte Verarbeitung von Eingaben
 - Keine interne Speicherung von Daten
 - Speicher sind Teil der Umgebung

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

Basismodell für Hardware und viele Arten von Software

- **Steuerungsautomaten**

- Alle Formen rein Hardware-gesteuerter automatischer Maschinen
Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren

Basismodell für Hardware und viele Arten von Software

- **Steuerungsautomaten**

- Alle Formen rein Hardware-gesteuerter automatischer Maschinen
Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren

- **Entwurf und Überprüfung digitaler Schaltungen**

- Entwicklungswerkzeuge und Testsoftware beschreiben endliches Verhalten

Basismodell für Hardware und viele Arten von Software

- **Steuerungsautomaten**

- Alle Formen rein Hardware-gesteuerter automatischer Maschinen
Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren

- **Entwurf und Überprüfung digitaler Schaltungen**

- Entwicklungswerkzeuge und Testsoftware beschreiben endliches Verhalten

- **Lexikalische Analyse in Compilern**

- Schnelle Identifizierung von Bezeichnern, Schlüsselwörtern, ...

Basismodell für Hardware und viele Arten von Software

- **Steuerungsautomaten**

- Alle Formen rein Hardware-gesteuerter automatischer Maschinen
Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren

- **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 Schlüsselworten

Basismodell für Hardware und viele Arten von Software

- **Steuerungsautomaten**

- Alle Formen rein Hardware-gesteuerter automatischer Maschinen
Waschmaschinen, Autos, Unterhaltungselektronik, Ampelanlagen, Computerprozessoren

- **Entwurf und Überprüfung digitaler Schaltungen**

- Entwicklungswerkzeuge und Testsoftware beschreiben endliches Verhalten

- **Lexikalische Analyse in Compilern**

- Schnelle Identifizierung von Bezeichnern, Schlüsselwörtern, ...

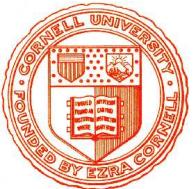
- **Textsuche in umfangreichen Dokumenten**

- Z.B. Suche nach Webseiten mithilfe von Schlüsselworten

- **Software mit endlichen Alternativen**

- Kommunikationsprotokolle, Protokolle zum sicheren Datenaustausch ...

Theoretische Informatik I



Einheit 2.1

Deterministische Endliche Automaten



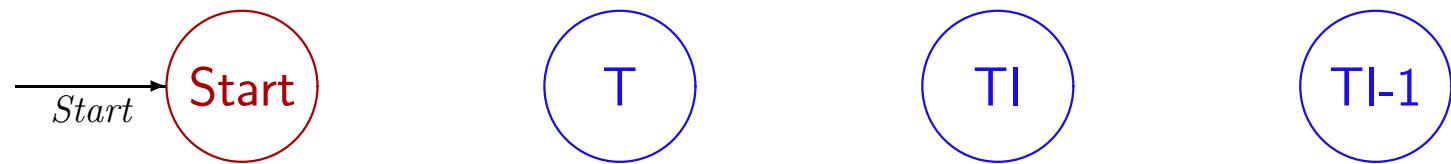
1. Arbeitsweise
2. Akzeptierte Sprache
3. Entwurf und Analyse

ERKENNUNG VON WORTEN MIT AUTOMATEN



- Endliche Anzahl von Zuständen

ERKENNUNG VON WORTEN MIT AUTOMATEN



- Endliche Anzahl von Zuständen
- Ein Startzustand

ERKENNUNG VON WORTEN MIT AUTOMATEN



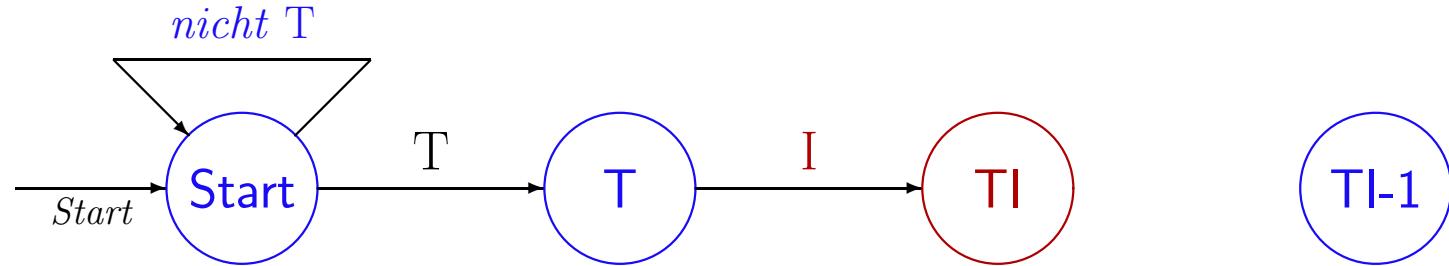
- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge

ERKENNUNG VON WORTEN MIT AUTOMATEN



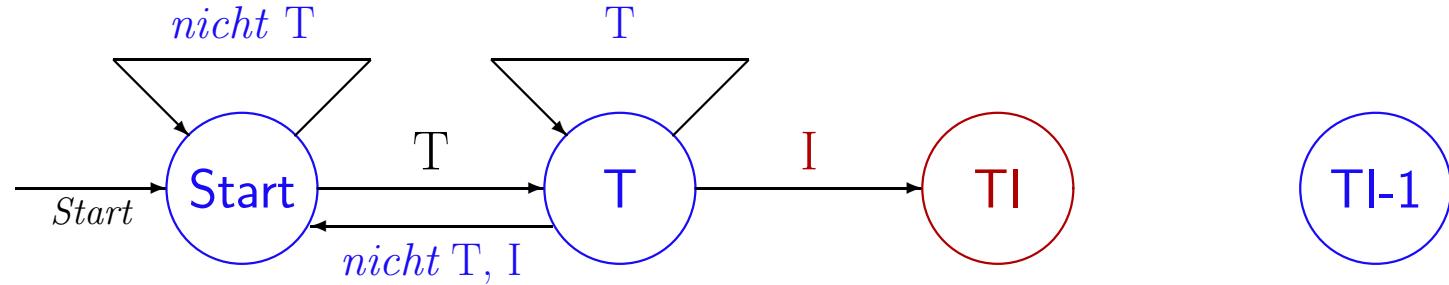
- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge
- Eingabealphabet: $\{A, \dots, Z, a, \dots, z, \ , ?, !, \dots\}$

ERKENNUNG VON WORTEN MIT AUTOMATEN



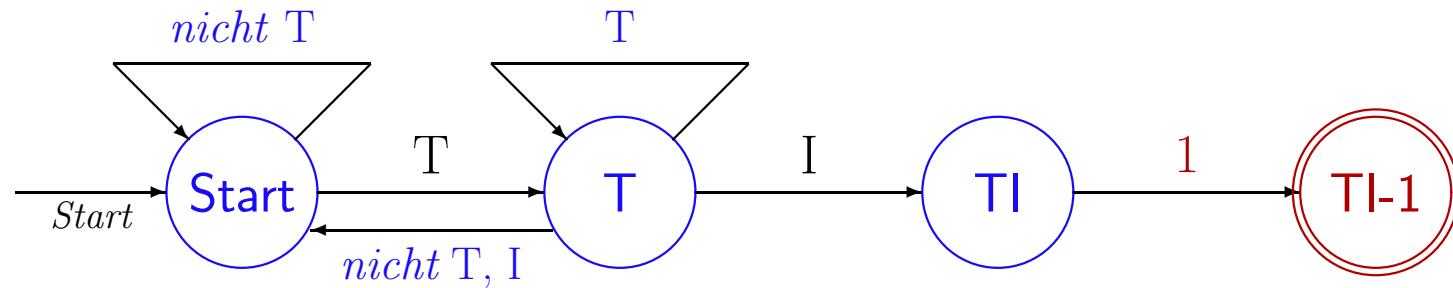
- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge
- Eingabealphabet: $\{A, \dots, Z, a, \dots, z, \ , ?, !, \dots\}$

ERKENNUNG VON WORTEN MIT AUTOMATEN



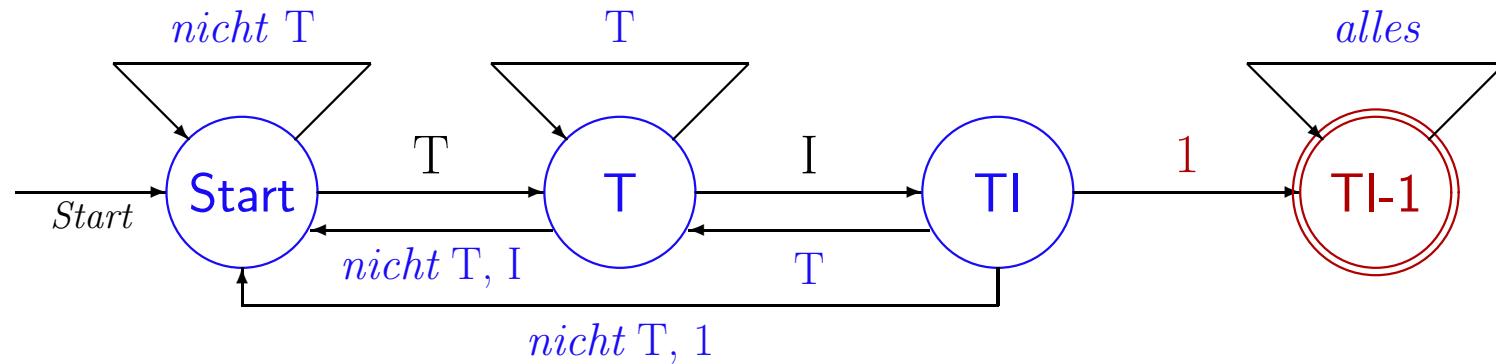
- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge
- Eingabealphabet: $\{A, \dots, Z, a, \dots, z, \ , ?, !, \dots\}$

ERKENNUNG VON WORTEN MIT AUTOMATEN

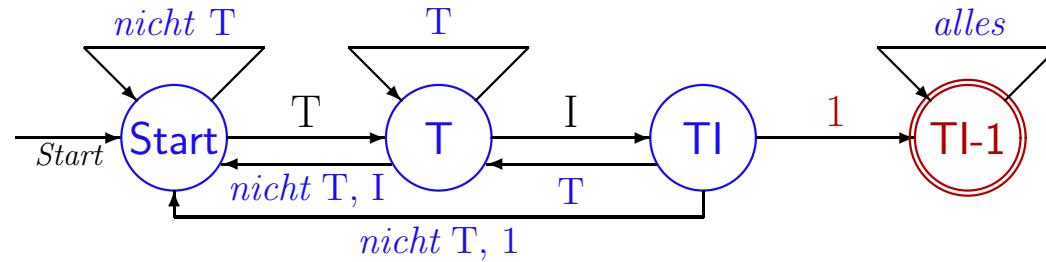


- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge
- Eingabealphabet: $\{A, \dots, Z, a, \dots, z, \ , ?, !, \dots\}$
- Ein oder mehrere akzeptierende Endzustände

ERKENNUNG VON WORTEN MIT AUTOMATEN



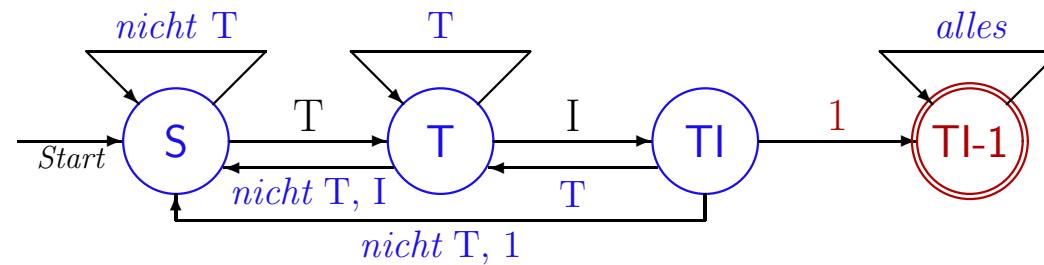
- Endliche Anzahl von Zuständen
- Ein Startzustand
- Regeln für Zustandsübergänge
- Eingabealphabet: $\{A, \dots, Z, a, \dots, z, \ , ?, !, \dots\}$
- Ein oder mehrere akzeptierende Endzustände



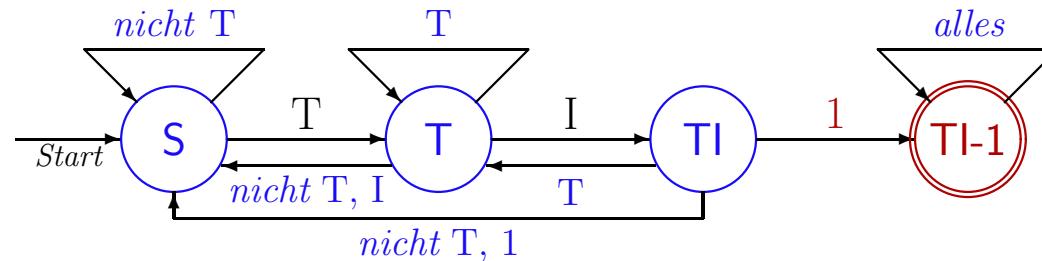
Ein **Deterministischer Endlicher Automat (DFA)**
ist ein 5-Tupel $\mathbf{A} = (Q, \Sigma, \delta, q_0, F)$ mit

- Q nichtleere endliche **Zustandsmenge**
- Σ **Eingabealphabet**
- $\delta: Q \times \Sigma \rightarrow Q$ **Zustandsübergbungsfunktion**
- $q_0 \in Q$ **Startzustand** (Anfangszustand)
- $F \subseteq Q$ Menge von **akzeptierenden** (finalen) **Zuständen** (Endzustände)

- Übergangsdiagramme

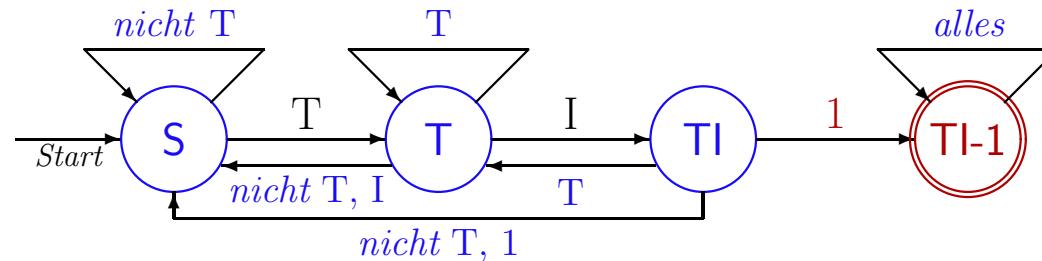


- Übergangsdiagramme



- Jeder Zustand in Q wird durch einen Knoten (Kreise) 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 die Diagramm bestimmt

• Übergangsdiagramme

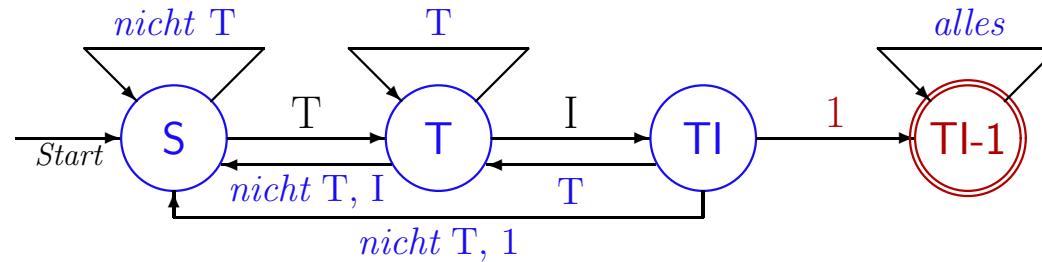


- Jeder Zustand in Q wird durch einen Knoten (Kreise) dargestellt
- Ist $\delta(q, a) = p$, so verläuft eine Kante von q nach p mit Beschriftung a (mehrere Beschriftungen derselben Kante möglich)
- q_0 wird durch einen mit *Start* beschrifteten Pfeil angezeigt
- Endzustände in F werden durch doppelte Kreise gekennzeichnet
- Σ meist implizit durch die Diagramm bestimmt

• Übergangstabellen

	T	I	1	sonst
\rightarrow	S	T	S	S
	T	T	I	S
	I	T	I	1
*	1	1	1	1

• Übergangsdiagramme



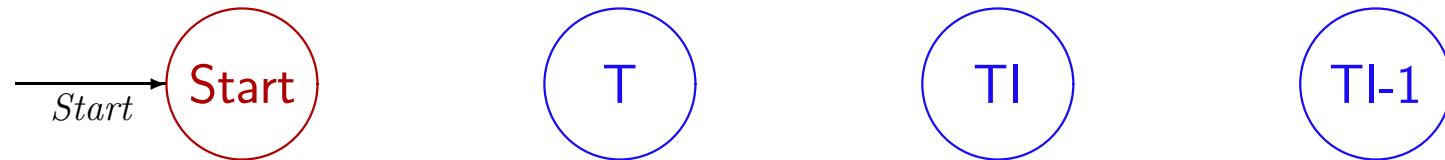
- Jeder Zustand in Q wird durch einen Knoten (Kreise) dargestellt
- Ist $\delta(q, a) = p$, so verläuft eine Kante von q nach p mit Beschriftung a (mehrere Beschriftungen derselben Kante möglich)
- q_0 wird durch einen mit *Start* beschrifteten Pfeil angezeigt
- Endzustände in F werden durch doppelte Kreise gekennzeichnet
- Σ meist implizit durch die Diagramm bestimmt

• Übergangstabellen

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

	T	I	1	sonst
\rightarrow	S	T	S	S
	T	T	I	S
	I	T	I	1
*	1	1	1	1

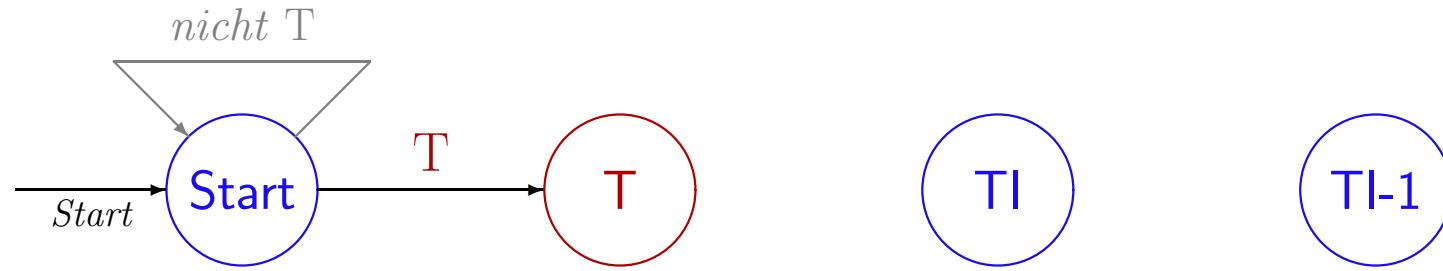
ARBEITSWEISE VON ENDLICHEN AUTOMATEN



● Anfangssituation

- Automat befindet sich im Startzustand q_0

ARBEITSWEISE VON ENDLICHEN AUTOMATEN



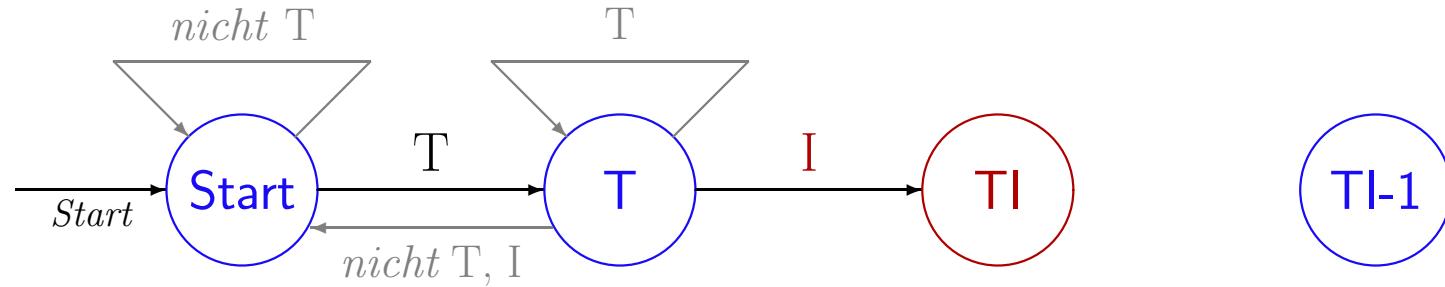
● Anfangssituation

- Automat befindet sich im Startzustand q_0

● Arbeitsschritt

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

ARBEITSWEISE VON ENDLICHEN AUTOMATEN



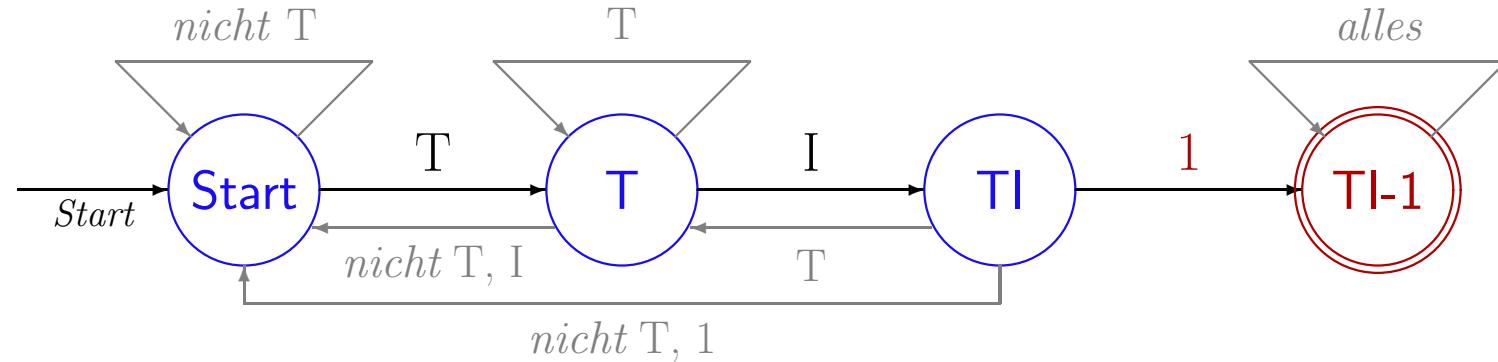
● Anfangssituation

- Automat befindet sich im Startzustand q_0

● Arbeitschritt

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

ARBEITSWEISE VON ENDLICHEN AUTOMATEN



● Anfangssituation

- Automat befindet sich im Startzustand q_0

● Arbeitsschritt

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

● Terminierung

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

● Ergebnis

- Eingabewort w wird akzeptiert, wenn $q_n \in F$, sonst wird w abgewiesen

- Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$

- Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts

- Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$

- Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts
- Formal: Induktive Definition

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

- **Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$**

- Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts
- Formal: Induktive Definition

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

- **Von A akzeptierte Sprache**

- Menge der Eingabeworte w für die $\hat{\delta}(q_0, w)$ akzeptierender Zustand ist

● Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$

- Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts
- Formal: Induktive Definition

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

● Von A akzeptierte Sprache

- Menge der Eingabeworte w für die $\hat{\delta}(q_0, w)$ akzeptierender Zustand ist

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

- Erweiterte Überführungsfunktion $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$

- Schrittweise Abarbeitung der Eingabe mit δ von links nach rechts
- Formal: Induktive Definition

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

- Von A akzeptierte Sprache

- Menge der Eingabeworte 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

ENTWURF UND ANALYSE ENDLICHER AUTOMATEN

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$
 - Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

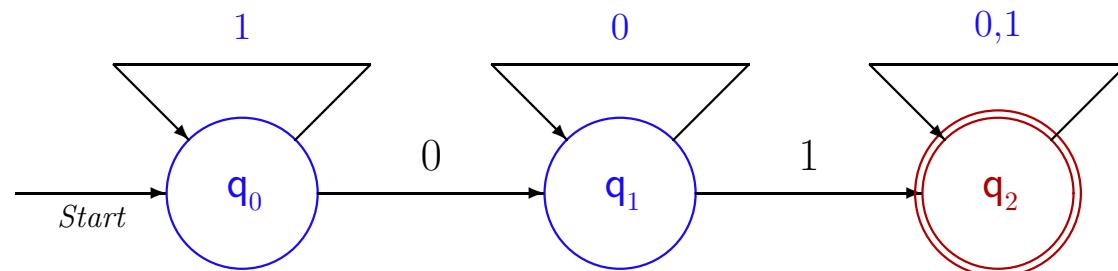
- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen $1^i 0^{j+1}$ bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen $1^i 0^j 01v$ bleibt in q_2

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

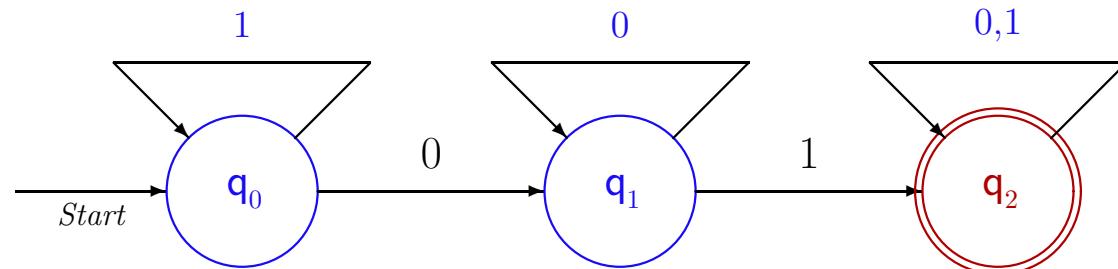
- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2

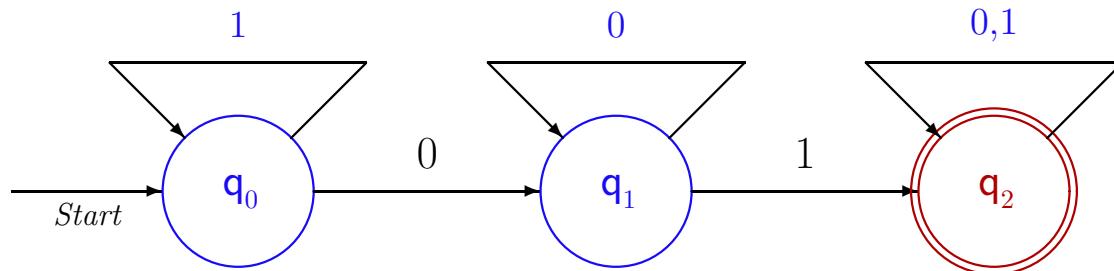


- Analyse der erweiterten Überführungsfunktion

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



- Analyse der erweiterten Überführungsfunktion

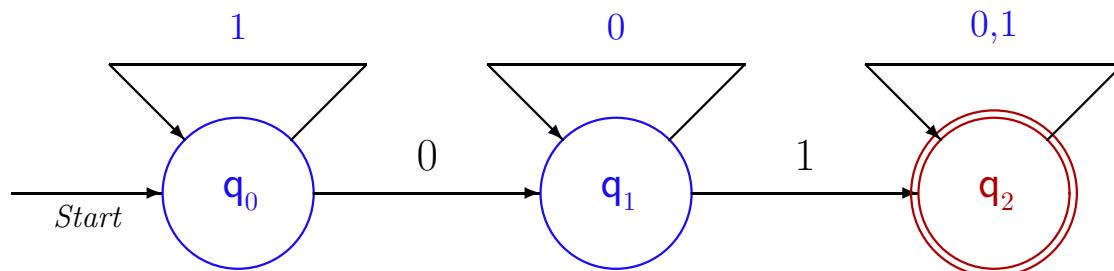
- Zeige durch Induktion für alle $i, j \in \mathbb{N}$ und alle $v \in \Sigma^*$

$$\hat{\delta}(q_0, 1^i) = q_0$$

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



- Analyse der erweiterten Überführungsfunktion

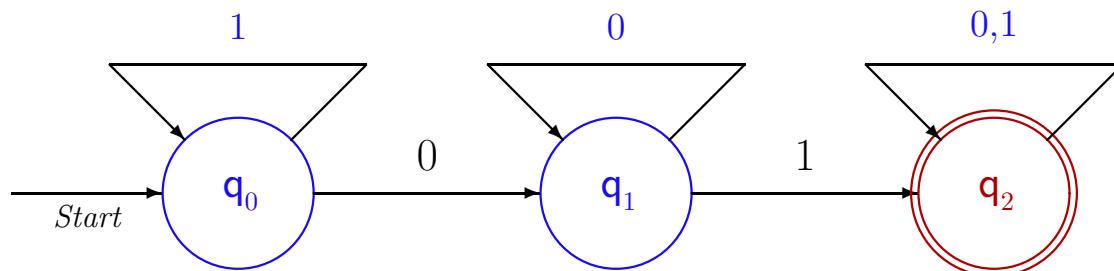
- Zeige durch Induktion für alle $i, j \in \mathbb{N}$ und alle $v \in \Sigma^*$

$$\hat{\delta}(q_0, 1^i) = q_0 \quad \hat{\delta}(q_0, 1^i0^{j+1}) = q_1$$

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



- Analyse der erweiterten Überführungsfunktion

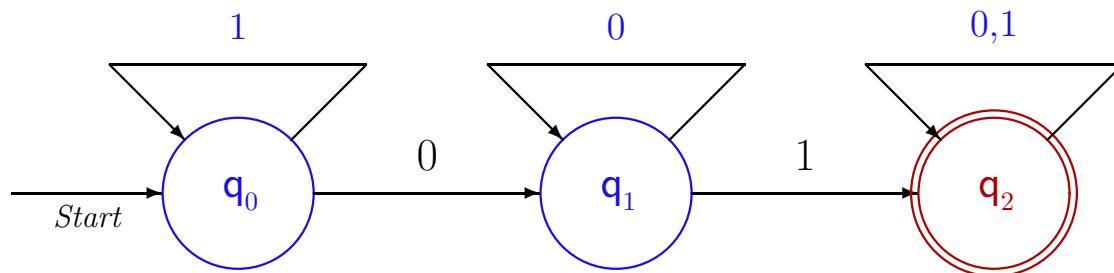
- Zeige durch Induktion für alle $i, j \in \mathbb{N}$ und alle $v \in \Sigma^*$

$$\hat{\delta}(q_0, 1^i) = q_0 \quad \hat{\delta}(q_0, 1^i0^{j+1}) = q_1 \quad \hat{\delta}(q_0, 1^i0^j01v) = q_2$$

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



- Analyse der erweiterten Überführungsfunktion

- Zeige durch Induktion für alle $i, j \in \mathbb{N}$ und alle $v \in \Sigma^*$

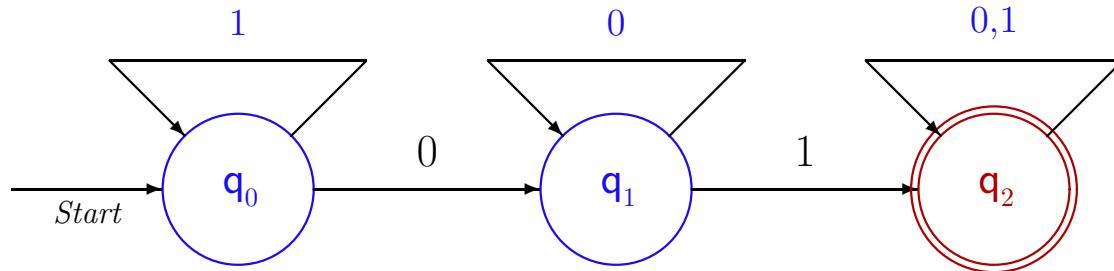
$$\hat{\delta}(q_0, 1^i) = q_0 \quad \hat{\delta}(q_0, 1^i0^{j+1}) = q_1 \quad \hat{\delta}(q_0, 1^i0^j01v) = q_2$$

- Es gilt: $w \in L$ g.d.w. es gibt $i, j \in \mathbb{N}, v \in \Sigma^*$ mit $w = 1^i0^j01v$ (Beweis?)

Entwerfe DEA für $L = \{u01v \mid u, v \in \{0, 1\}^*\}$

- Entwurf eines Automaten mit Alphabet $\Sigma = \{0, 1\}$

- Zustand q_0 : A hat noch keine 0 gelesen 1^i bleibt in q_0
- Zustand q_1 : A hat eine 0 aber noch keine 1 gelesen 1^i0^{j+1} bleibt in q_1
- Zustand q_2 : A hat eine Zeichenkette 01 gelesen 1^i0^j01v bleibt in q_2



- Analyse der erweiterten Überführungsfunktion

- Zeige durch Induktion für alle $i, j \in \mathbb{N}$ und alle $v \in \Sigma^*$

$$\hat{\delta}(q_0, 1^i) = q_0 \quad \hat{\delta}(q_0, 1^i0^{j+1}) = q_1 \quad \hat{\delta}(q_0, 1^i0^j01v) = q_2$$
- Es gilt: $w \in L$ g.d.w. es gibt $i, j \in \mathbb{N}, v \in \Sigma^*$ mit $w = 1^i0^j01v$ (Beweis?)
- Es folgt $L = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) = q_2 \in F\} = L(A)$

BEISPIELE ENDLICHER AUTOMATEN

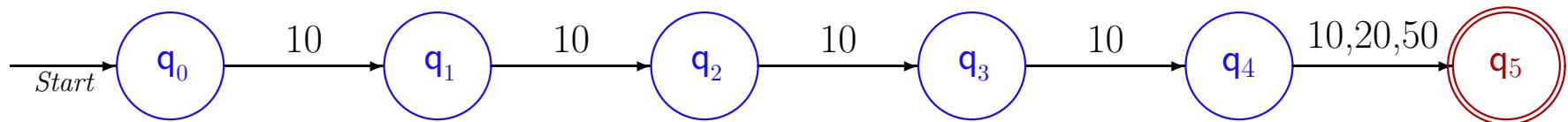
• 50c Kaffeeautomat

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

BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

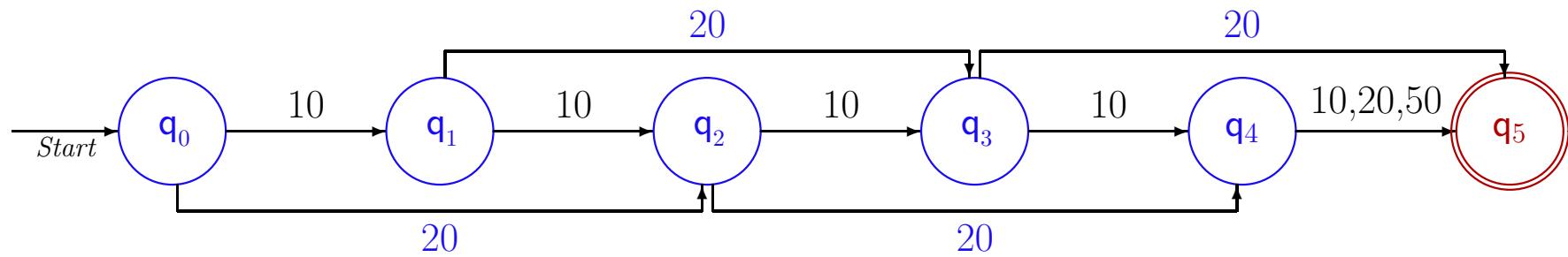
- Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

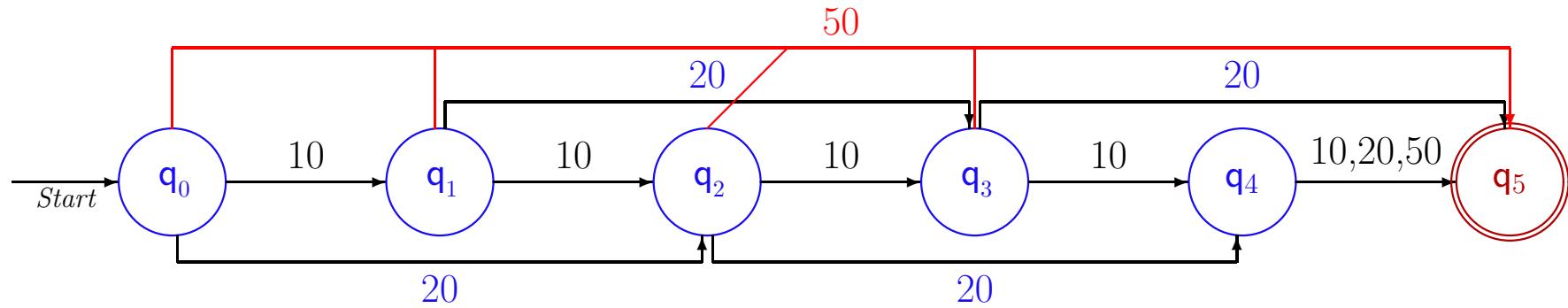
- Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

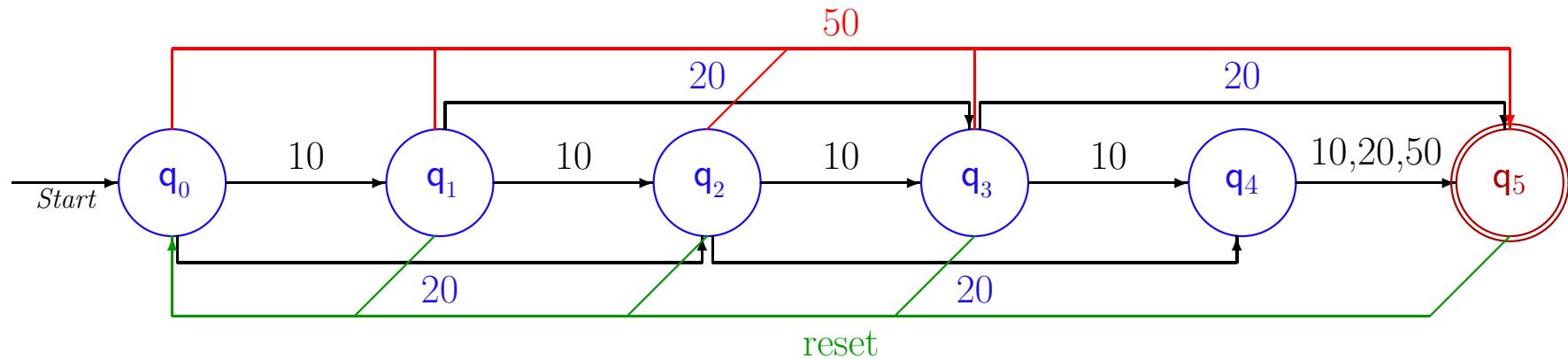
- Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

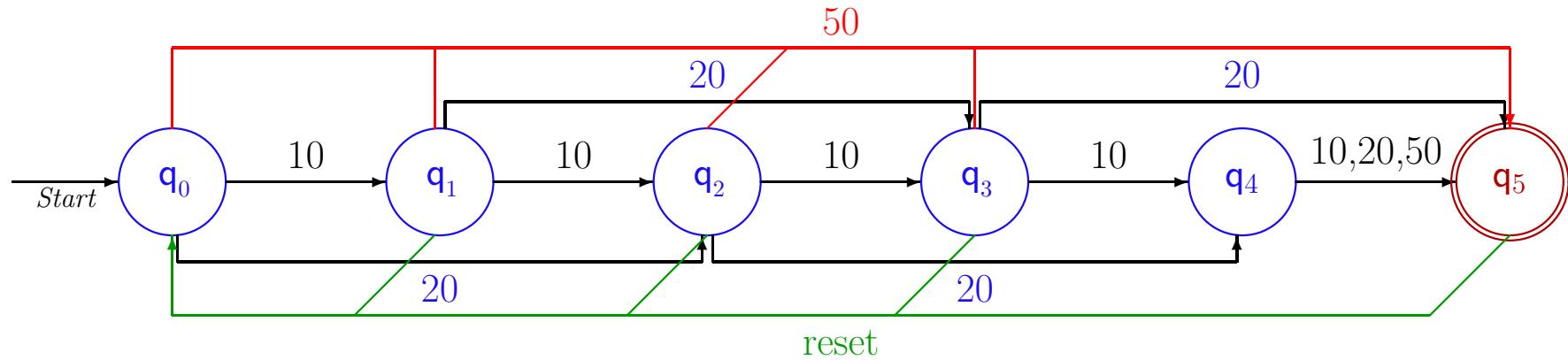
- Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

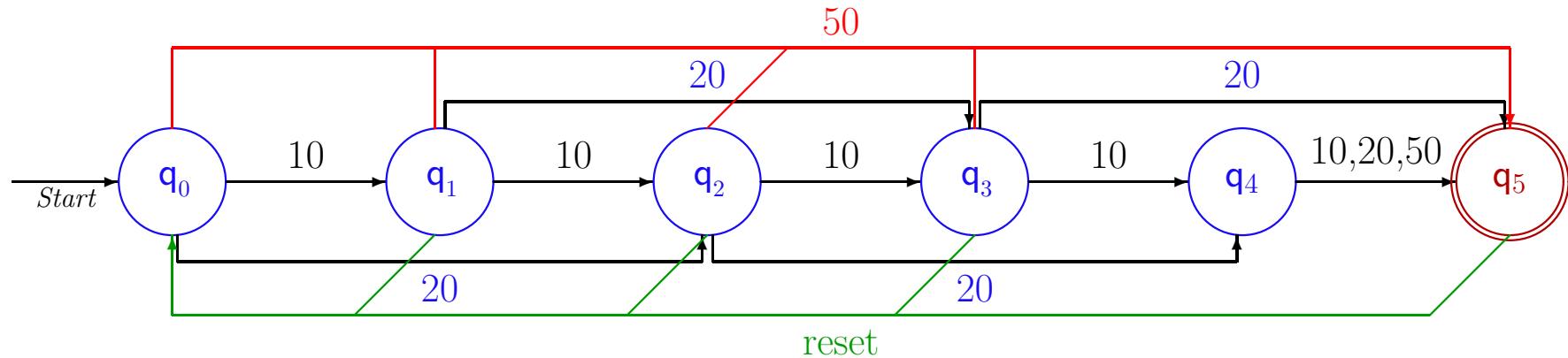
- Akzeptiert 10,20,50c Münzen, gibt kein Geld zurück, mit Reset-Taste



BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

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

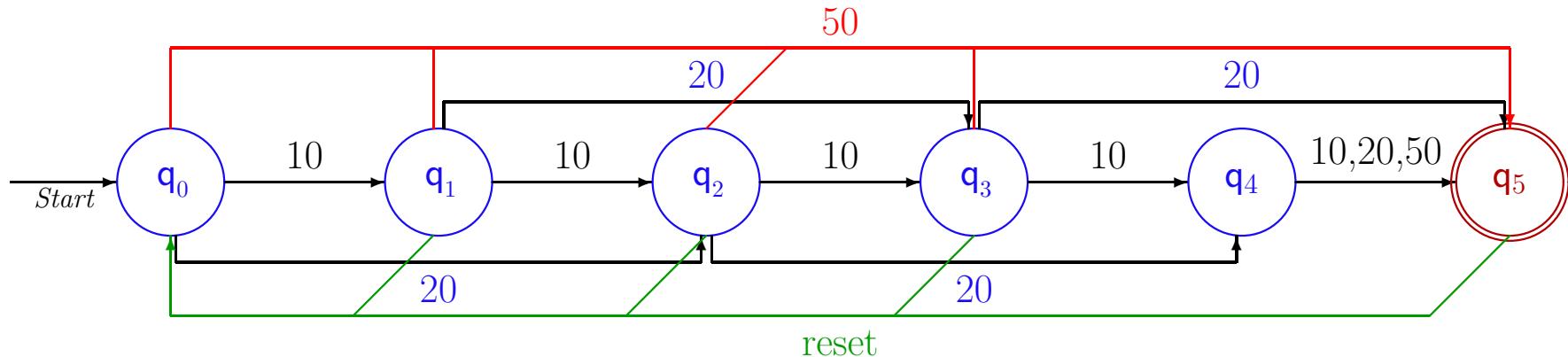


- $L = \{w \in \{0, 1\}^* \mid w \text{ enthält gerade Anzahl von } 0 \text{ und } 1\}$

BEISPIELE ENDLICHER AUTOMATEN

• 50c Kaffeeautomat

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



• $L = \{w \in \{0, 1\}^* \mid w \text{ enthält gerade Anzahl von } 0 \text{ und } 1\}$

- 4 Zustände codieren Anzahl der Vorkommen von Nullen und Einsen

$q_0 = (\text{gerade, gerade})$

$q_1 = (\text{gerade, ungerade})$

$q_2 = (\text{ungerade, gerade})$

$q_3 = (\text{ungerade, ungerade})$

