

Script generated by TTT

Title: Seidl: Theoretische_Informatik
(18.04.2013)

Date: Thu Apr 18 16:01:19 CEST 2013

Duration: 89:45 min

Pages: 93

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

L_0	_____
L_1	
L_2	
\vdots	

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0					
L_1					
L_2					
\vdots					

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1					
L_2					
\vdots					

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1	1	0	0	\dots	
L_2	1	1	1	\dots	
\vdots					

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1	1	0	0	\dots	
L_2	1	1	1	\dots	
\vdots	\vdots	\vdots	\vdots	\ddots	

Eine Sprache L , die nicht in der Tabelle ist:

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1	1	0	0	\dots	
L_2	1	1	1	\dots	
\vdots	\vdots	\vdots	\vdots	\ddots	

Eine Sprache L , die nicht in der Tabelle ist: \neg Diagonale!

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1	1	0	0	\dots	
L_2	1	1	1	\dots	
\vdots	\vdots	\vdots	\vdots	\ddots	

Eine Sprache L , die nicht in der Tabelle ist: \neg Diagonale!

$L \mid 1 \quad 1 \quad 0 \quad \dots$

Widerspruch!

Formal: $L := \{w_i \mid w_i \notin L_i\}$

Satz 1.8

Die Menge der Sprachen über einem nicht-leeren Alphabet ist überabzählbar.

Beweis:

Indirekt. Nimm an, die Menge der Sprachen sei abzählbar.

	w_0	w_1	w_2	\dots	$= \Sigma^*$
L_0	0	1	1	\dots	
L_1	1	0	0	\dots	
L_2	1	1	1	\dots	
\vdots	\vdots	\vdots	\vdots	\ddots	

Eine Sprache L , die nicht in der Tabelle ist: \neg Diagonale!

$L \mid 1 \quad 1 \quad 0 \quad \dots$

Widerspruch!

Formal: $L := \{w_i \mid w_i \notin L_i\}$

Dann $L \neq L_i$ für alle i , da $w_i \in L \Leftrightarrow w_i \notin L_i$. □

Definition 1.9

Eine Sprache $M \subseteq \Sigma^*$ ist **entscheidbar** gdw es einen Algorithmus gibt, der ihre **charakteristische Funktion**

$$\chi_M(x) = \begin{cases} 1 & \text{falls } x \in M \\ 0 & \text{falls } x \notin M \end{cases}$$

berechnet.

Definition 1.9

Eine Sprache $M \subseteq \Sigma^*$ ist **entscheidbar** gdw es einen Algorithmus gibt, der ihre **charakteristische Funktion**

$$\chi_M(x) = \begin{cases} 1 & \text{falls } x \in M \\ 0 & \text{falls } x \notin M \end{cases}$$

berechnet.

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)
- $\{(ab)^n \mid n \in \mathbb{N}\}$

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)
- $\{(ab)^n \mid n \in \mathbb{N}\}$
- $\{a^n b^n \mid n \in \mathbb{N}\}$

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)
- $\{(ab)^n \mid n \in \mathbb{N}\}$
- $\{a^n b^n \mid n \in \mathbb{N}\}$
- Die Menge aller Primzahlen (dezimal kodiert)

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)
- $\{(ab)^n \mid n \in \mathbb{N}\}$
- $\{a^n b^n \mid n \in \mathbb{N}\}$
- Die Menge aller Primzahlen (dezimal kodiert)
- \emptyset

Beispiel 1.10 (Entscheidbare Sprachen)

- Die Menge aller Wörter im Duden (24. Aufl.)
- $\{(ab)^n \mid n \in \mathbb{N}\}$
- $\{a^n b^n \mid n \in \mathbb{N}\}$
- Die Menge aller Primzahlen (dezimal kodiert)
- \emptyset
- $\{\epsilon\}$

Satz 1.11

Es gibt nicht-entscheidbare Sprachen.

Satz 1.11

Es gibt nicht-entscheidbare Sprachen.

Beweis:

Jeder Algorithmus ist ein Wort.

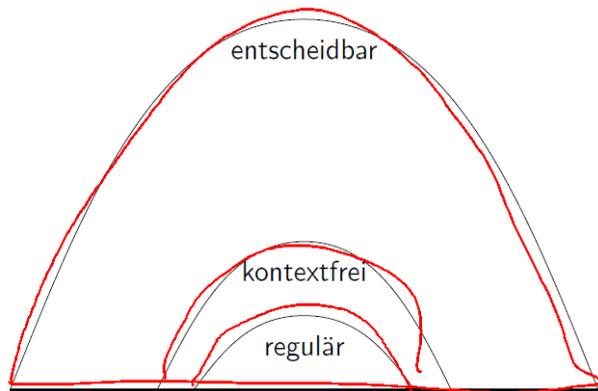
Daher gibt es nur abzählbar viele Algorithmen.

Aber überabzählbar viele Sprachen, also mehr als Algorithmen. \square

Wir haben sogar gezeigt:

- Es gibt höchstens *abzählbar viele entscheidbare Sprachen*.

alle formalen Sprachen

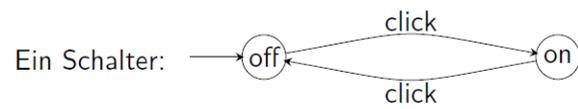


2. Reguläre Sprachen

2.1 Deterministische endliche Automaten

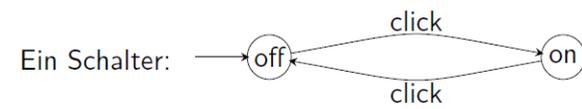
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



2. Reguläre Sprachen

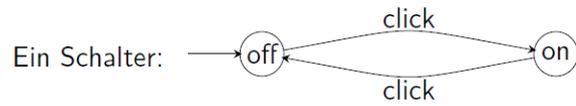
2.1 Deterministische endliche Automaten



UML state charts sind endliche Automaten.

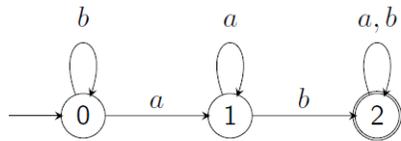
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



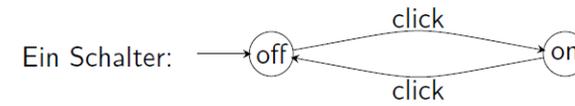
UML state charts sind endliche Automaten.

Ein endlicher Automat mit Endzustand:



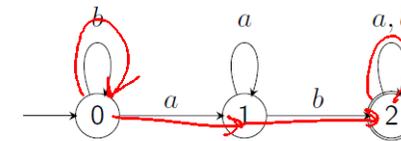
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



UML state charts sind endliche Automaten.

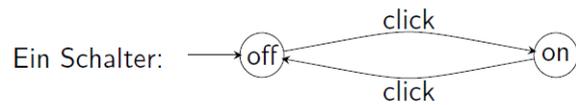
Ein endlicher Automat mit Endzustand:



Eingabewort *baba* \rightsquigarrow Zustandsfolge 0,0,1,2,2.

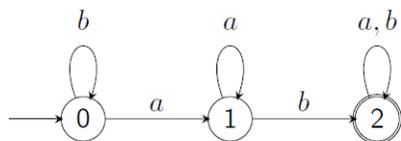
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



UML state charts sind endliche Automaten.

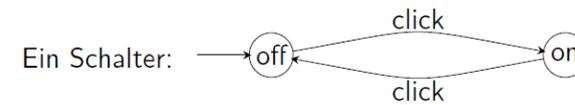
Ein endlicher Automat mit Endzustand:



Eingabewort *baba* \rightsquigarrow Zustandsfolge 0,0,1,2,2.
Menge der Wörter, die vom Start- in den Endzustand führen?

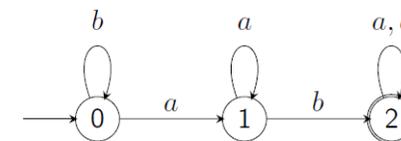
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



UML state charts sind endliche Automaten.

Ein endlicher Automat mit Endzustand:



Eingabewort *baba* \rightsquigarrow Zustandsfolge 0,0,1,2,2.
Menge der Wörter, die vom Start- in den Endzustand führen?
Alle Wörter, die ϵ enthalten.

Definition 2.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

Definition 2.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

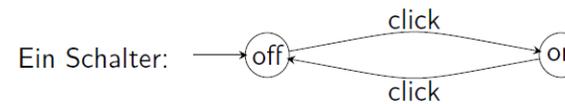
- einer endlichen Menge von Zuständen Q ,

Definition 2.1

Ein deterministischer endlicher Automat (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

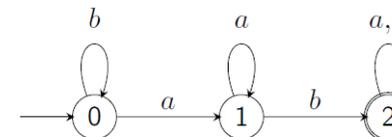
2. Reguläre Sprachen

2.1 Deterministische endliche Automaten



UML state charts sind endliche Automaten.

Ein endlicher Automat mit Endzustand:



Eingabewort $baba \rightsquigarrow$ Zustandsfolge 0,0,1,2,2.

Menge der Wörter, die vom Start- in den Endzustand führen?

Alle Wörter, die ab enthalten.

Definition 2.1

Ein **deterministischer endlicher Automat** (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von **Zuständen** Q ,
- einem (endlichen) **Eingabealphabet** Σ ,
- einer (totalen!) **Übergangsfunktion** $\delta : Q \times \Sigma \rightarrow Q$,
- einem **Startzustand** $q_0 \in Q$, und

Definition 2.1

Ein **deterministischer endlicher Automat** (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von **Zuständen** Q ,
- einem (endlichen) **Eingabealphabet** Σ ,
- einer (totalen!) **Übergangsfunktion** $\delta : Q \times \Sigma \rightarrow Q$,
- einem **Startzustand** $q_0 \in Q$, und
- einer Menge von **Endzuständen** $F \subseteq Q$.

Definition 2.1

Ein **deterministischer endlicher Automat** (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von **Zuständen** Q ,
- einem (endlichen) **Eingabealphabet** Σ ,
- einer (totalen!) **Übergangsfunktion** $\delta : Q \times \Sigma \rightarrow Q$,
- einem **Startzustand** $q_0 \in Q$, und
- einer Menge von **Endzuständen** $F \subseteq Q$.

Endzustände heißen auch **akzeptierenden Zustände**.

Definition 2.1

Ein **deterministischer endlicher Automat** (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von **Zuständen** Q ,
- einem (endlichen) **Eingabealphabet** Σ ,
- einer (totalen!) **Übergangsfunktion** $\delta : Q \times \Sigma \rightarrow Q$,
- einem **Startzustand** $q_0 \in Q$, und
- einer Menge von **Endzuständen** $F \subseteq Q$.

Endzustände heißen auch **akzeptierenden Zustände**.

Die von M **akzeptierte Sprache** ist

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

Definition 2.1

Ein **deterministischer endlicher Automat** (*deterministic finite automaton*, DFA) $M = (Q, \Sigma, \delta, q_0, F)$ besteht aus

- einer endlichen Menge von **Zuständen** Q ,
- einem (endlichen) **Eingabealphabet** Σ ,
- einer (totalen!) **Übergangsfunktion** $\delta : Q \times \Sigma \rightarrow Q$,
- einem **Startzustand** $q_0 \in Q$, und
- einer Menge von **Endzuständen** $F \subseteq Q$.

Endzustände heißen auch **akzeptierenden Zustände**.

Die von M **akzeptierte Sprache** ist

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

wobei $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ rekursiv definiert ist durch

$$\begin{aligned} \hat{\delta}(q, \epsilon) &= q \\ \hat{\delta}(q, aw) &= \hat{\delta}(\delta(q, a), w) \quad \text{für } a \in \Sigma, w \in \Sigma^* \end{aligned}$$

Definition 2.2

Eine Sprache ist **regulär** gdw sie von einem DFA akzeptiert wird.

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen
 - Kanten $\hat{=}$ Übergängen: $p \xrightarrow{a} q \hat{=} \delta(p, a) = q$

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen
 - Kanten $\hat{=}$ Übergängen: $p \xrightarrow{a} q \hat{=} \delta(p, a) = q$

Der Anfangszustand wird durch einen Pfeil, Endzustände werden durch doppelte Kreise gekennzeichnet.

- Die Operation $\hat{\cdot}$ ist ein Funktional. In OCaml:

```
let rec dach f q = function
  [] -> q |
  a::w -> dach f (f q a) w
```

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen
 - Kanten $\hat{=}$ Übergängen: $p \xrightarrow{a} q \hat{=} \delta(p, a) = q$

Der Anfangszustand wird durch einen Pfeil, Endzustände werden durch doppelte Kreise gekennzeichnet.

- Die Operation $\hat{\cdot}$ ist ein Funktional. In OCaml:

```
let rec dach f q = function
  [] -> q |
  a::w -> dach f (f q a) w
```

D.h. dach = fold_left !

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen
 - Kanten $\hat{=}$ Übergängen: $p \xrightarrow{a} q \hat{=} \delta(p, a) = q$

Der Anfangszustand wird durch einen Pfeil, Endzustände werden durch doppelte Kreise gekennzeichnet.

- Die Operation $\hat{\cdot}$ ist ein Funktional. In OCaml:

```
let rec dach f q = function
  [] -> q |
  a::w -> dach f (f q a) w
```

D.h. dach = fold_left !

- Die Erweiterung von f auf \hat{f} funktioniert für beliebige $f : A \times B \rightarrow A$

- Für $a \in \Sigma$ gilt $\hat{\delta}(q, a)$

Bemerkungen

- Endliche Automaten können durch gerichtete und markierte **Zustandsgraphen** veranschaulicht werden:
 - Knoten $\hat{=}$ Zuständen
 - Kanten $\hat{=}$ Übergängen: $p \xrightarrow{a} q \hat{=} \delta(p, a) = q$

Der Anfangszustand wird durch einen Pfeil, Endzustände werden durch doppelte Kreise gekennzeichnet.

- Die Operation $\hat{\cdot}$ ist ein Funktional. In OCaml:

```
let rec dach f q = function
  [] -> q |
  a::w -> dach f (f q a) w
```

D.h. dach = fold_left !

- Die Erweiterung von f auf \hat{f} funktioniert für beliebige $f : A \times B \rightarrow A$

- Für $a \in \Sigma$ gilt $\hat{\delta}(q, a) = \hat{\delta}(q, a\epsilon) = \hat{\delta}(\delta(q, a), \epsilon) = \delta(q, a)$

Das Beweisprinzip **Induktion über die Länge von Wörtern** hat folgenden häufigen Spezialfall:



Definition 2.2

Eine Sprache ist **regulär** gdw sie von einem DFA akzeptiert wird.

Das Beweisprinzip **Induktion über die Länge von Wörtern** hat folgenden häufigen Spezialfall:

- **Basis:** Zeige die Behauptung für $w = \epsilon$
- **Induktionsannahme:** Die Behauptung stimmt für w .

Das Beweisprinzip **Induktion über die Länge von Wörtern** hat folgenden häufigen Spezialfall:

- **Basis:** Zeige die Behauptung für $w = \epsilon$
- **Induktionsannahme:** Die Behauptung stimmt für w .
Schritt: Zeige die Behauptung für aw , $a \in \Sigma$.

Das Beweisprinzip **Induktion über die Länge von Wörtern** hat folgenden häufigen Spezialfall:

- **Basis:** Zeige die Behauptung für $w = \epsilon$
- **Induktionsannahme:** Die Behauptung stimmt für w .
Schritt: Zeige die Behauptung für aw , $a \in \Sigma$.
Alternative: Zeige die Behauptung für wa , $a \in \Sigma$.

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- **Basis:** $w = \epsilon$:
 $\hat{\delta}(q, \epsilon x) =$

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- **Basis:** $w = \epsilon$:
 $\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) =$

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- **Basis:** $w = \epsilon$:
 $\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) =$

Lemma 2.4

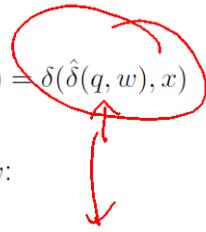
Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:

$$\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$$



Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

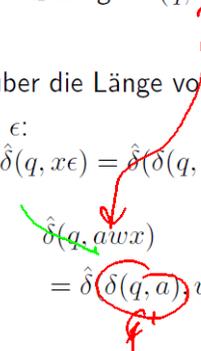
Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:

$$\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$$

- Schritt:

$$\begin{aligned} & \hat{\delta}(q, awx) \\ &= \hat{\delta}(\hat{\delta}(q, a), wx) \quad \text{Def. von } \hat{\delta} \end{aligned}$$



Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:

$$\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$$

- Schritt:

$$\begin{aligned} & \hat{\delta}(q, awx) \\ &= \hat{\delta}(\delta(q, a), wx) \quad \text{Def. von } \hat{\delta} \\ &= \delta(\hat{\delta}(\delta(q, a), w), x) \quad \text{Ind.Hyp.} \end{aligned}$$

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:

$$\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$$

- Schritt:

$$\begin{aligned} & \hat{\delta}(q, awx) \\ &= \hat{\delta}(\delta(q, a), wx) \quad \text{Def. von } \hat{\delta} \\ &= \delta(\hat{\delta}(\delta(q, a), w), x) \quad \text{Ind.Hyp.} \\ &= \delta(\hat{\delta}(q, aw), x) \quad \text{Def. von } \hat{\delta} \end{aligned}$$

□

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:
 $\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$

- Schritt:

$$\begin{aligned} & \hat{\delta}(q, awx) \\ &= \hat{\delta}(\delta(q, a), wx) && \text{Def. von } \hat{\delta} \\ &= \delta(\hat{\delta}(\delta(q, a), w), x) && \text{Ind.Hyp.} \\ &= \delta(\hat{\delta}(q, aw), x) && \text{Def. von } \hat{\delta} \end{aligned}$$

□

Lemma 2.4

Für $x \in \Sigma$ und $w \in \Sigma^*$ gilt $\hat{\delta}(q, wx) = \delta(\hat{\delta}(q, w), x)$

Beweis:

Mit Induktion über die Länge von w :

- Basis: $w = \epsilon$:
 $\hat{\delta}(q, \epsilon x) = \hat{\delta}(q, x\epsilon) = \hat{\delta}(\delta(q, x), \epsilon) = \delta(q, x) = \delta(\hat{\delta}(q, \epsilon), x)$

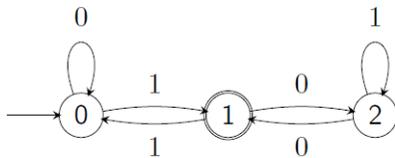
- Schritt:

$$\begin{aligned} & \hat{\delta}(q, awx) \\ &= \hat{\delta}(\delta(q, a), wx) && \text{Def. von } \hat{\delta} \\ &= \delta(\hat{\delta}(\delta(q, a), w), x) && \text{Ind.Hyp.} \\ &= \delta(\hat{\delta}(q, aw), x) && \text{Def. von } \hat{\delta} \end{aligned}$$

□

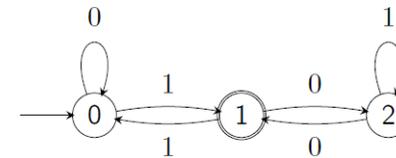
Beispiel 2.5

Ein DFA M :



Beispiel 2.5

Ein DFA M :

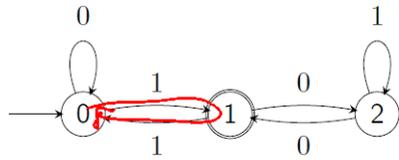


Für $w \in \{0, 1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl, zB $\#100 = 4$.

$$L(M) = \{w \mid \quad \quad \quad \}$$

Beispiel 2.5

Ein DFA M :

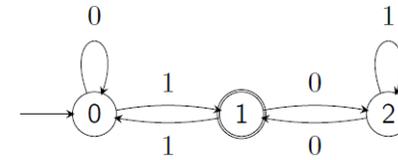


Für $w \in \{0, 1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl, zB $\#100 = 4$.

$$L(M) = \{w \mid \#w \bmod 3 = 1\}$$

Beispiel 2.5

Ein DFA M :



Für $w \in \{0, 1\}^*$ sei $\#w$ die von w binär repräsentierte Zahl, zB $\#100 = 4$.

$$L(M) = \{w \mid \#w \bmod 3 = 1\}$$

Beweis:

1. $\delta(q, b) =$
2. $\hat{\delta}(0, w) =$

□

2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

[$\mathcal{P}(Q)$ = Potenzmenge = Menge aller Teilmengen = 2^Q]

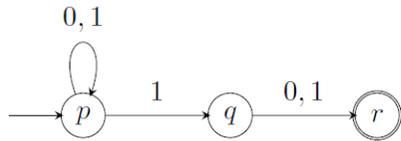
2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

[$\mathcal{P}(Q)$ = Potenzmenge = Menge aller Teilmengen = 2^Q]

Beispiel 2.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:



$$\delta(r, 1) = \emptyset$$

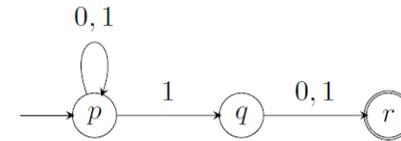
2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

[$\mathcal{P}(Q)$ = Potenzmenge = Menge aller Teilmengen = 2^Q]

Beispiel 2.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:



Wort wird akzeptiert gdw es einen Weg zum Endzustand gibt.

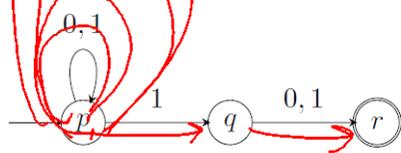
2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

[$\mathcal{P}(Q)$ = Potenzmenge = Menge aller Teilmengen = 2^Q]

Beispiel 2.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:



Wort wird akzeptiert gdw es einen Weg zum Endzustand gibt.

100110

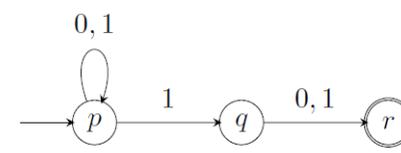
2.2 Nichtdeterministische endliche Automaten

Verallgemeinerung: $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

[$\mathcal{P}(Q)$ = Potenzmenge = Menge aller Teilmengen = 2^Q]

Beispiel 2.6

Erkennung der Binärzahlen, deren vorletzte Ziffer 1 ist:



Wort wird akzeptiert gdw es einen Weg zum Endzustand gibt.

Intuitive Vorstellung:

- Der Automat durchsucht alle möglichen Wege (parallel/sequentiell).
- Der Automat "rät" den richtigen Weg.

Definition 2.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

Definition 2.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA

Definition 2.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

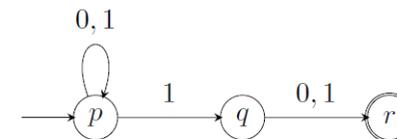
- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

Definition 2.7

Ein nichtdeterministischer endlicher Automat (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

Beispiel



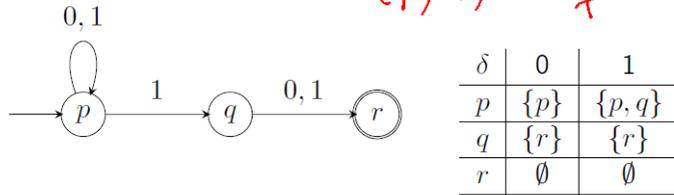
δ	0	1
p	{p}	{p, q}
q	{r}	{r}
r	\emptyset	\emptyset

Definition 2.7

Ein **nichtdeterministischer endlicher Automat** (*nondeterministic finite automaton*, NFA) ist ein 5-Tupel $N = (Q, \Sigma, \delta, q_0, F)$, so dass

- Q, Σ, q_0 und F sind wie bei einem DFA
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

Beispiel



Alternative: Relation $\delta \subseteq Q \times \Sigma \times Q$

Erweiterung von $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$:

Erweiterung von $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

Erweiterung von $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$$\Rightarrow \hat{\delta} : \mathcal{P}(Q) \times \Sigma^* \rightarrow \mathcal{P}(Q)$$

Erweiterung von $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
 auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$$\Rightarrow \hat{\delta} : \mathcal{P}(Q) \times \Sigma^* \rightarrow \mathcal{P}(Q)$$

Intuition:

$\hat{\delta}(S, w)$ ist Menge aller Zustände,
 die sich von einem Zustand in S aus mit w erreichen lassen.

Erweiterung von $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$
 auf $\bar{\delta} : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$:

$$\bar{\delta}(S, a) := \bigcup_{q \in S} \delta(q, a)$$

$$\Rightarrow \hat{\delta} : \mathcal{P}(Q) \times \Sigma^* \rightarrow \mathcal{P}(Q)$$

Intuition:

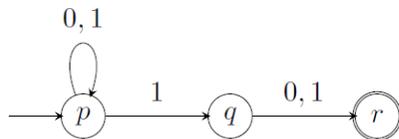
$\hat{\delta}(S, w)$ ist Menge aller Zustände,
 die sich von einem Zustand in S aus mit w erreichen lassen.

Die von $N = (Q, \Sigma, \delta, q_0, F)$ akzeptierte Sprache ist

$$L(N) := \{w \in \Sigma^* \mid \hat{\delta}(\{q_0\}, w) \cap F \neq \emptyset\}$$

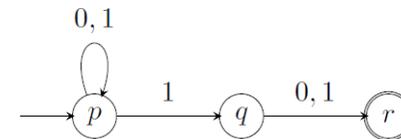


Beispiel



$$\hat{\delta}(\{p, q\}, 10) =$$

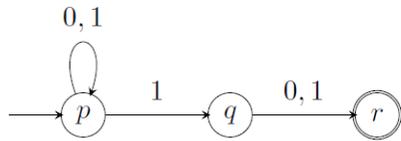
Beispiel



$$\hat{\delta}(\{p, q\}, 10) =$$

$$\hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) =$$

Beispiel

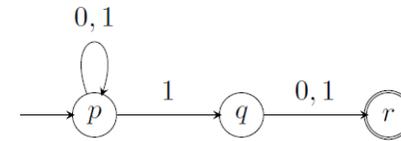


$$\hat{\delta}(\{p, q\}, 10) =$$

$$\hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) =$$

$$\hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) =$$

Beispiel



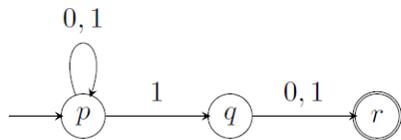
$$\hat{\delta}(\{p, q\}, 10) =$$

$$\hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) =$$

$$\hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) =$$

$$\hat{\delta}(\{p, q, r\}, 0) =$$

Beispiel



$$\hat{\delta}(\{p, q\}, 10) =$$

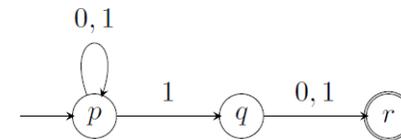
$$\hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) =$$

$$\hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) =$$

$$\hat{\delta}(\{p, q, r\}, 0) =$$

$$\bar{\delta}(\{p, q, r\}, 0) =$$

Beispiel



$$\hat{\delta}(\{p, q\}, 10) =$$

$$\hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) =$$

$$\hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) =$$

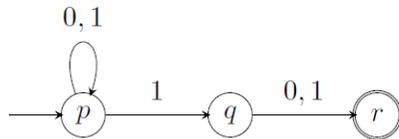
$$\hat{\delta}(\{p, q, r\}, 0) =$$

$$\bar{\delta}(\{p, q, r\}, 0) =$$

$$\delta(p, 0) \cup \delta(q, 0) \cup \delta(r, 0) =$$

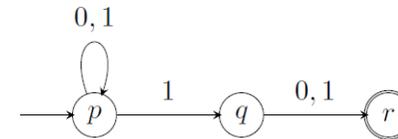
p r \emptyset

Beispiel



$$\begin{aligned}\hat{\delta}(\{p, q\}, 10) &= \\ \hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) &= \\ \hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) &= \\ \hat{\delta}(\{p, q, r\}, 0) &= \\ \bar{\delta}(\{p, q, r\}, 0) &= \\ \delta(p, 0) \cup \delta(q, 0) \cup \delta(r, 0) &= \\ \{p\} \cup \{r\} \cup \emptyset &= \{p, r\}\end{aligned}$$

Beispiel



$$\begin{aligned}\hat{\delta}(\{p, q\}, 10) &= \\ \hat{\delta}(\bar{\delta}(\{p, q\}, 1), 0) &= \\ \hat{\delta}(\delta(p, 1) \cup \delta(q, 1), 0) &= \\ \hat{\delta}(\{p, q, r\}, 0) &= \\ \bar{\delta}(\{p, q, r\}, 0) &= \\ \delta(p, 0) \cup \delta(q, 0) \cup \delta(r, 0) &= \\ \{p\} \cup \{r\} \cup \emptyset &= \{p, r\}\end{aligned}$$

2.3 Äquivalenz von NFA und DFA

Satz 2.8

Für jede von einem NFA akzeptierte Sprache L gibt es einen DFA M mit

$$L = L(M) .$$

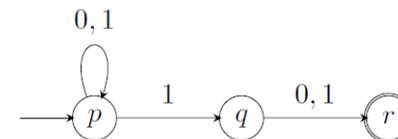
2.3 Äquivalenz von NFA und DFA

Satz 2.8

Für jede von einem NFA akzeptierte Sprache L gibt es einen DFA M mit

$$L = L(M) .$$

Beispiel



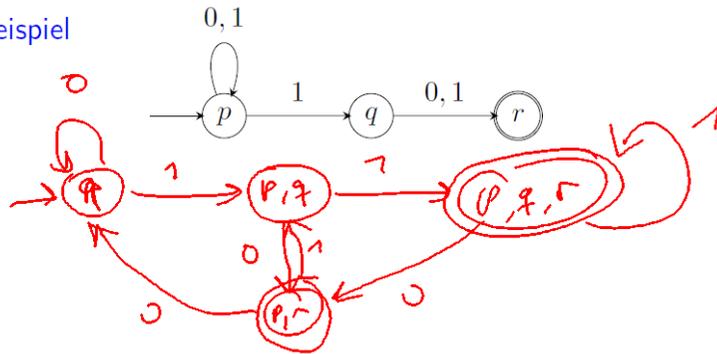
2.3 Äquivalenz von NFA und DFA

Satz 2.8

Für jede von einem NFA akzeptierte Sprache L gibt es einen DFA M mit

$$L = L(M).$$

Beispiel



Beweis:

Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Beweis:

Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Beweis:

Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$w \in L(N) \Leftrightarrow \hat{\delta}(\{q_0\}, w) \cap F \neq \emptyset \quad \text{Def.}$$

Beweis:

Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{aligned} w \in L(N) &\Leftrightarrow \hat{\delta}(\{q_0\}, w) \cap F \neq \emptyset && \text{Def.} \\ &\Leftrightarrow \hat{\delta}(\{q_0\}, w) \in F_M && \text{Def.} \end{aligned}$$

Beweis:

Sei $N = (Q, \Sigma, \delta, q_0, F)$ ein NFA.

Definiere den DFA $M = (\mathcal{P}(Q), \Sigma, \bar{\delta}, \{q_0\}, F_M)$:

$$F_M := \{S \subseteq Q \mid S \cap F \neq \emptyset\}$$

Dann gilt:

$$\begin{aligned} w \in L(N) &\Leftrightarrow \hat{\delta}(\{q_0\}, w) \cap F \neq \emptyset && \text{Def.} \\ &\Leftrightarrow \hat{\delta}(\{q_0\}, w) \in F_M && \text{Def.} \\ &\Leftrightarrow w \in L(M) && \text{Def.} \end{aligned}$$

□