Mathematische Grundlagen der Computerlinguistik

formale Sprachen

Dozentin: Wiebke Petersen

3. Foliensatz

Alphabete und Wörter

Definition

- Alphabet Σ: endliche Menge von Symbolen / Zeichen.
- Wort: eine endliche Kette/Folge $x_1 \dots x_n$ von Symbolen/Zeichen eines Alphabets (mit $n \geq 0$). Das Wort, das aus null Zeichen besteht heißt leeres Wort und wird mit ε bezeichnet.
- Die Menge aller Wörter über einem Alphabet Σ bezeichnen wir mit Σ^*.

Alphabete und Wörter

Definition

- Alphabet Σ: endliche Menge von Symbolen / Zeichen.
- Wort: eine endliche Kette/Folge $x_1 \dots x_n$ von Symbolen/Zeichen eines Alphabets (mit $n \geq 0$). Das Wort, das aus null Zeichen besteht heißt leeres Wort und wird mit ε bezeichnet.
- Die Menge aller Wörter über einem Alphabet Σ bezeichnen wir mit Σ^*.

  $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ ist die Menge der nichtleeren Wörter.

Alphabete und Wörter

Definition

- Alphabet Σ: endliche Menge von Symbolen / Zeichen.
- Wort: eine endliche Kette/Folge $x_1 \dots x_n$ von Symbolen/Zeichen eines Alphabets (mit $n \geq 0$). Das Wort, das aus null Zeichen besteht heißt leeres Wort und wird mit ε bezeichnet.
- Die Menge aller Wörter über einem Alphabet Σ bezeichnen wir mit Σ^*.

  $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ ist die Menge der nichtleeren Wörter.
- Länge eines Wortes $|w|$: Gesamtzahl der Zeichen eines Wortes w ($|abbaca| = 6$, $|\varepsilon| = 0$)

Leersymbol, leeres Wort und leere Menge

Vorsicht Verwechslungsgefahr!

Das Leersymbol $\sqcup$ ist ein *Zeichen* des Alphabets, also ist ein Wort, das nur aus dem Leersymbol besteht, ein Wort der Länge 1.

Leersymbol, leeres Wort und leere Menge

Vorsicht Verwechslungsgefahr!

Das Leersymbol $\sqcup$ ist ein *Zeichen* des Alphabets, also ist ein Wort, das nur aus dem Leersymbol besteht, ein Wort der Länge 1.

Das leere Wort ε ist ein *Wort* der Länge 0.

Leersymbol, leeres Wort und leere Menge

Vorsicht Verwechslungsgefahr!

Das Leersymbol $\sqcup$ ist ein *Zeichen* des Alphabets, also ist ein Wort, das nur aus dem Leersymbol besteht, ein Wort der Länge 1.

Das leere Wort ε ist ein *Wort* der Länge 0.

Die leere Menge $\emptyset$ ist eine *Menge*.

Übung: Alphabete und Wörter

Sei $\Sigma = \{a, b, c\}$ ein Alphabet:

- Geben Sie zwei Wörter der Läge 4 über Σ an.
- Welche der folgenden Ausdrücke sind Wörter über Σ und welche Länge haben sie?:
  '*aa*', '*caab*', '*da*'
- Was ist der Unterschied zwischen Σ^*, Σ^+ und Σ?
- Wieviele Elemente haben Σ, Σ^* und Σ^+?

Operationen auf Wörtern

Verkettung / Konkatenation

Die Konkatenation / Verkettung zweier Wörter $u = a_1 a_2 \dots a_n$ und $v = b_1 b_2 \dots b_m$ mit $n, m \geq 0$ ist

$$u \circ v = a_1 \dots a_n b_1 \dots b_m$$

Häufig schreiben wir uv statt $u \circ v$.

Operationen auf Wörtern

Verkettung / Konkatenation

Die Konkatenation / Verkettung zweier Wörter $u = a_1 a_2 \dots a_n$ und $v = b_1 b_2 \dots b_m$ mit $n, m \geq 0$ ist

$$u \circ v = a_1 \dots a_n b_1 \dots b_m$$

Häufig schreiben wir uv statt $u \circ v$.

$$w \circ \varepsilon = \varepsilon \circ w = w \quad \text{Neutrales Element}$$

$$u \circ (v \circ w) = (u \circ v) \circ w \quad \text{Assoziativität}$$

Operationen auf Wörtern

Verkettung / Konkatenation

Die Konkatenation / Verkettung zweier Wörter $u = a_1 a_2 \dots a_n$ und $v = b_1 b_2 \dots b_m$ mit $n, m \geq 0$ ist

$$u \circ v = a_1 \dots a_n b_1 \dots b_m$$

Häufig schreiben wir uv statt $u \circ v$.

$$w \circ \varepsilon = \varepsilon \circ w = w \quad \text{Neutrales Element}$$

$$u \circ (v \circ w) = (u \circ v) \circ w \quad \text{Assoziativität}$$

Ist die Konkatenationsoperation kommutativ?

Symbolpolitik der Mathematik

Vorsicht:

- Obwohl die Symbole für die Komposition von Funktionen und die Konkatenation von Wörtern übereinstimmen, handelt es sich um unterschiedliche Operationen!
- In der Mathematik finden sie häufig mehrdeutige Symbole, deren Bedeutung sich aus dem jeweiligen Kontext ergibt.
- Sie müssen sich also bei dem Symbol $\circ$ immer fragen, ob es zwischen Funktionen oder Wörtern steht (wir werden auch noch eine Operation auf Mengen kennenlernen, die mit demselben Symbol bezeichnet wird).
- Bedenken Sie, dass die Alternative die Verwendung einer unbegrenzten Zahl verschiedener Symbole wäre, da es theoretisch unendlich viele Operationen gibt. Jedes dieser Symbole müsste in Zeichensätzen vorgehalten werden, was unmöglich ist, da Alphabete endlich sein müssen. Stellen Sie sich außerdem vor, ich würde an der Tafel versuchen eine Vielzahl von sehr ähnlichen Symbolen zu verwenden (Beispiel: Kreis mit dickem Punkt in der Mitte, Kreis mit kleinem Punkt, Kreis ohne Punkt, Kreis mit zwei Umrandungen,...), Sie würden das nicht lesen wollen!

Operationen auf Wörtern

Exponenten

- w^n: w wird n-mal mit sich selbst verkettet.
- $w^0 = \varepsilon$: w wird '0-mal' mit sich selbst verkettet.

Operationen auf Wörtern

Exponenten

- w^n: w wird n-mal mit sich selbst verkettet.
- $w^0 = \varepsilon$: w wird '0-mal' mit sich selbst verkettet.

Umkehrung

- Die Umkehrung eines Wortes w wird mit w^R bezeichnet. $(abcd)^R = dcba$.
- Ein Wort w, für das $w = w^R$ gilt, heißt Palindrom.

(madam, reliefpfeiler, otto, anna, regallager . . .)

Übung: Operationen auf Wörtern

Seien $w = aabc$ und $v = bcc$ Wörter, ermitteln Sie:

- $w \circ v$
- $((w^R \circ v)^R)^2$
- $w \circ (v^R \circ w^3)^0$

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot-, -\cdots, \ldots, --\cdot\cdot\}$

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot-, -\cdot\cdot\cdot, \ldots, --\cdot\cdot\}$
- Sprache L_{pal} der Palindrome im deutschen Duden $L_{pal} = \{\text{Madam, reliefpfeiler, } \ldots\}$

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot-, -\cdots, \ldots, --\cdot\cdot\}$
- Sprache L_{pal} der Palindrome im deutschen Duden $L_{pal} = \{\text{Madam, reliefpfeiler, } \ldots\}$
- Leere Menge

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot-, -\cdot\cdot\cdot, \ldots, --\cdot\cdot\}$
- Sprache L_{pal} der Palindrome im deutschen Duden $L_{pal} = \{\text{Madam, reliefpfeiler, } \ldots\}$
- Leere Menge
- Menge der Wörter der Länge 13 über dem Alphabet $\{a, b, c\}$

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot-, -\cdots, \ldots, --\cdot\cdot\}$
- Sprache L_{pal} der Palindrome im deutschen Duden $L_{pal} = \{\text{Madam, reliefpfeiler, } \ldots\}$
- Leere Menge
- Menge der Wörter der Länge 13 über dem Alphabet $\{a, b, c\}$
- Sprache der syntaktisch wohlgeformten Java-Programme

Formale Sprache

Definition

Eine formale Sprache L ist eine Menge von Wörtern über einem Alphabet Σ, also $L \subseteq \Sigma^*$.

Beispiele:

- Sprache L_{rom} der gültigen römischen Zahldarstellungen über dem Alphabet $\Sigma_{rom} = \{\mathbf{I}, \mathbf{V}, \mathbf{X}, \mathbf{L}, \mathbf{C}, \mathbf{D}, \mathbf{M}\}$.
- Sprache L_{Mors} der Buchstaben des lateinischen Alphabets dargestellt im Morsecode. $L_{Mors} = \{\cdot -, - \cdot \cdot\cdot, \ldots, - - \cdot\cdot\}$
- Sprache L_{pal} der Palindrome im deutschen Duden $L_{pal} = \{\text{Madam, reliefpfeiler, } \ldots\}$
- Leere Menge
- Menge der Wörter der Länge 13 über dem Alphabet $\{a, b, c\}$
- Sprache der syntaktisch wohlgeformten Java-Programme
- Deutsch?

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L =$

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L = \{abbbbb, abbab, abbb, aab\}$ und $L \circ K =$

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L = \{abbbbb, abbab, abbb, aab\}$ und $L \circ K = \{bbbabb, bbba, ababb, aba\}$
- $K \circ \emptyset =$

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L = \{abbbbb, abbab, abbb, aab\}$ und $L \circ K = \{bbbabb, bbba, ababb, aba\}$
- $K \circ \emptyset = \emptyset$
- $K \circ \{\varepsilon\} =$

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L = \{abbbbb, abbab, abbb, aab\}$ und $L \circ K = \{bbbabb, bbba, ababb, aba\}$
- $K \circ \emptyset = \emptyset$
- $K \circ \{\varepsilon\} = K$
- $K^2 =$

Operationen auf Sprachen

Seien $L \subseteq \Sigma^*$ und $K \subseteq \Sigma^*$ zwei Sprachen über dem Alphabet Σ, dann entstehen durch die Verknüpfung mit Mengenoperatoren neue Sprachen über Σ:

$$K \cup L,\ K \cap L,\ K \setminus L$$

Die Verkettung von Wörtern kann ausgedehnt werden auf die Verkettung von Sprachen:

$$K \circ L := \{v \circ w \in \Sigma^* | v \in K, w \in L\}$$

Beispiel: Sei $K = \{abb, a\}$ und $L = \{bbb, ab\}$

- $K \circ L = \{abbbbb, abbab, abbb, aab\}$ und $L \circ K = \{bbbabb, bbba, ababb, aba\}$
- $K \circ \emptyset = \emptyset$
- $K \circ \{\varepsilon\} = K$
- $K^2 = K \circ K = \{abbabb, abba, aabb, aa\}$

Potenzen von Sprachen, Iteration, Kleene-Stern

Die n-te Potenz einer Sprache L ist die n-fache Verkettung von L mit sich selbst:

$$L^n = \underbrace{L \circ L \circ L \ldots \circ L}_{n\text{-mal}}$$

Induktive Definition:

$$L^0 = \{\epsilon\}, \quad L^{n+1} = L^n \circ L$$

Potenzen von Sprachen, Iteration, Kleene-Stern

Die n-te Potenz einer Sprache L ist die n-fache Verkettung von L mit sich selbst:

$$L^n = \underbrace{L \circ L \circ L \ldots \circ L}_{n\text{-mal}}$$

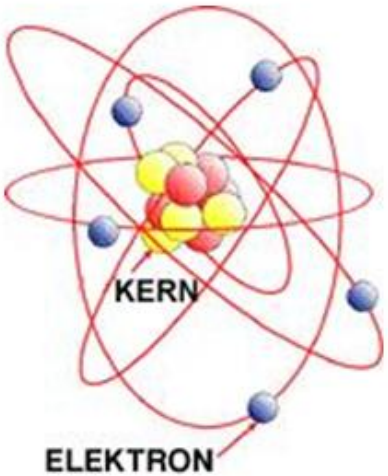
Induktive Definition:

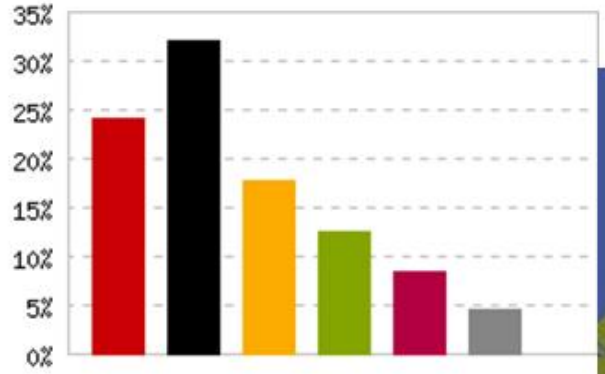
$$L^0 = \{\epsilon\}, \quad L^{n+1} = L^n \circ L$$

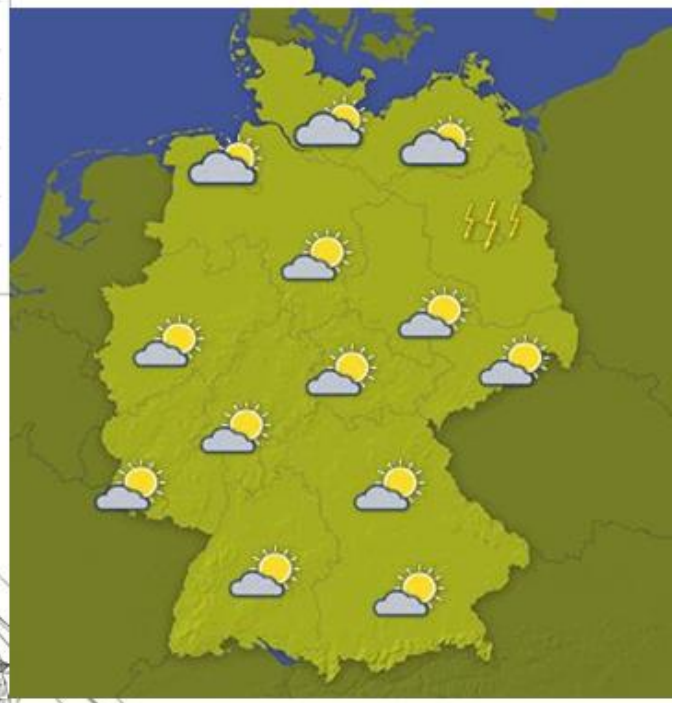
Die Iteration (Kleene-Stern) von L ist

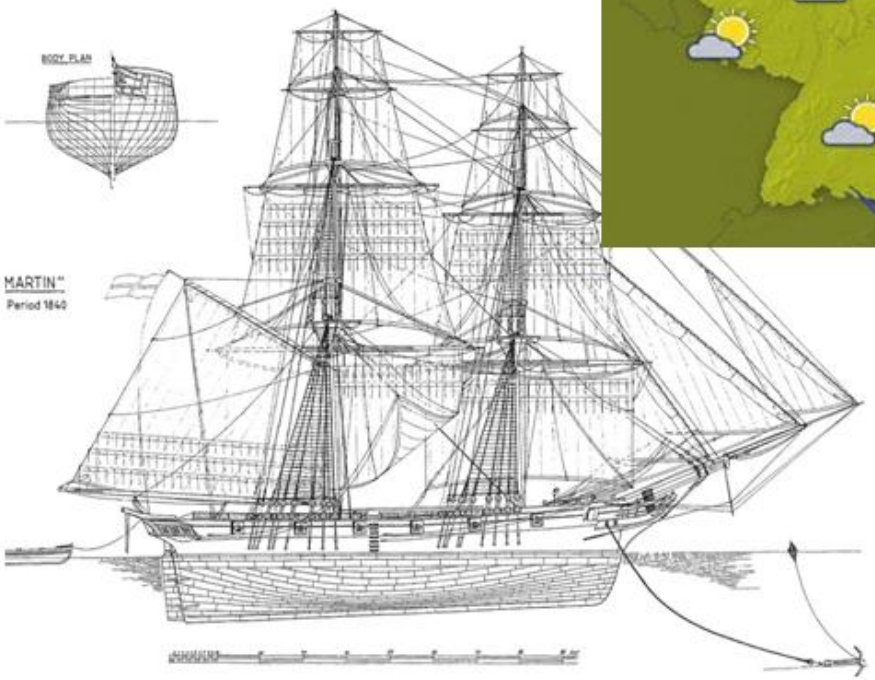
$$L^* := \bigcup_{n \geq 0} L^n$$

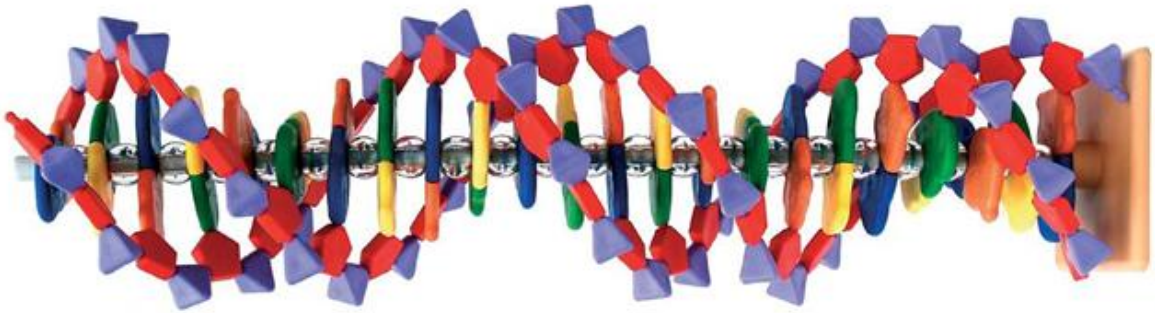
Für jede beliebige Sprache L gilt: $\epsilon \in L^*$

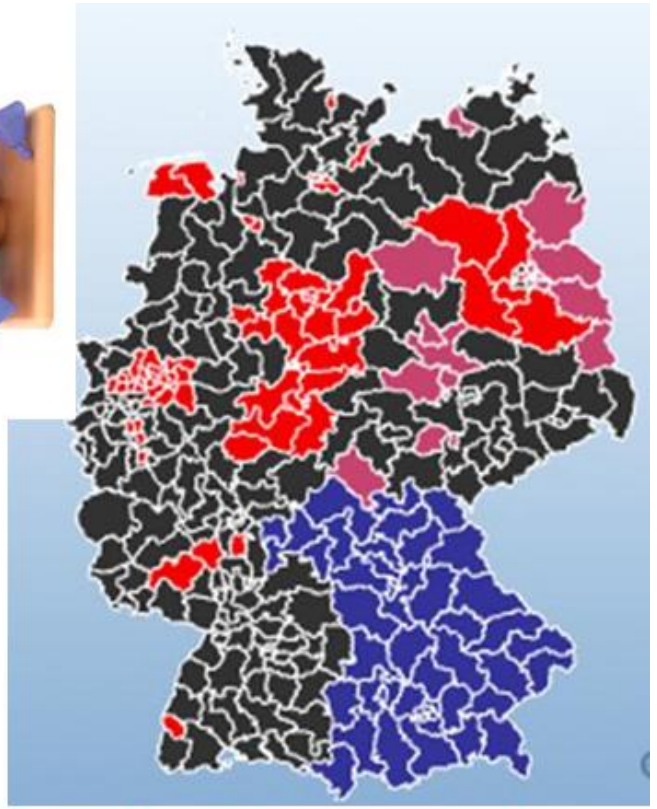

KERN
ELEKTRON

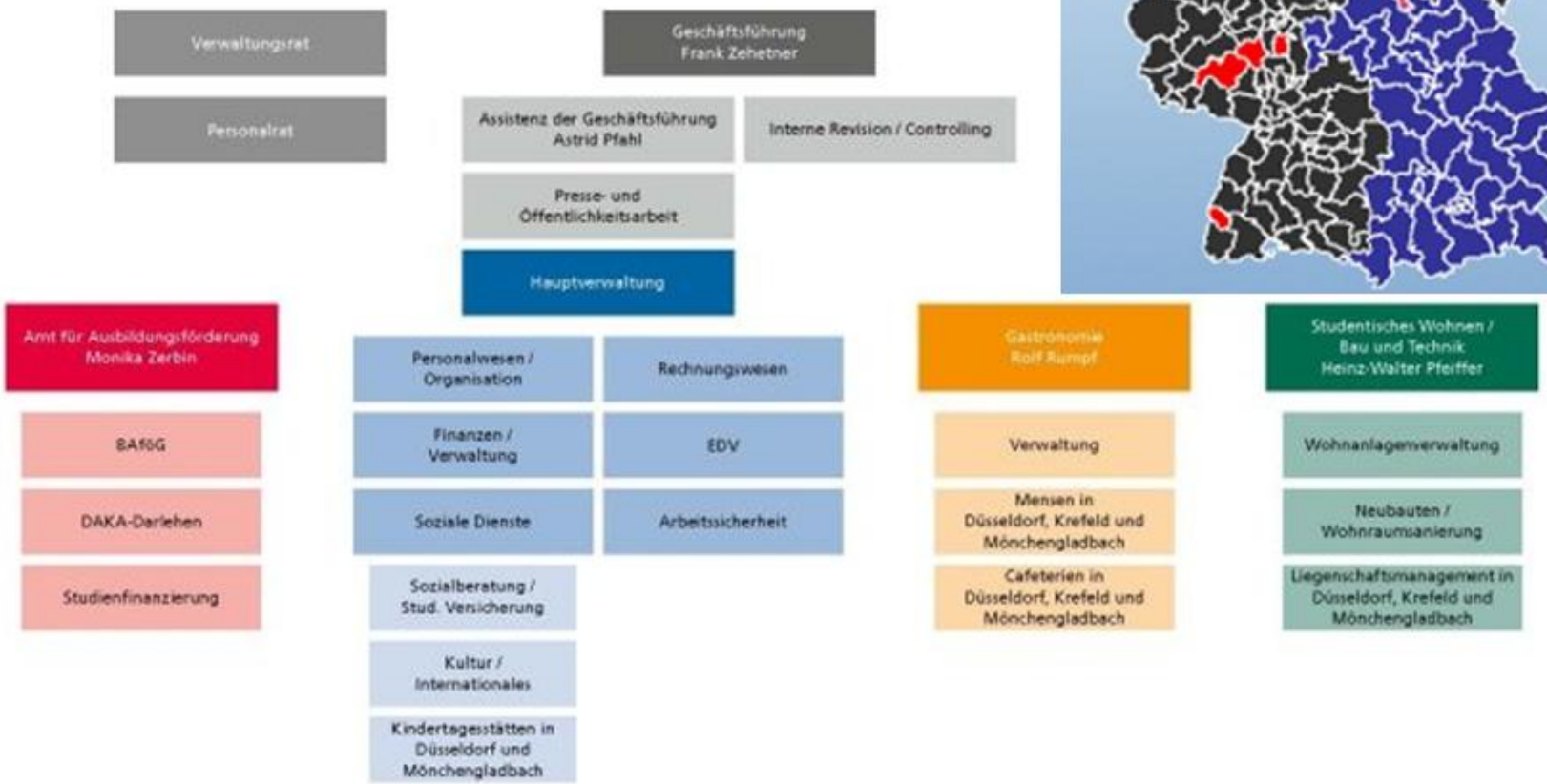

35%
30%
25%
20%
15%
10%
5%
0%



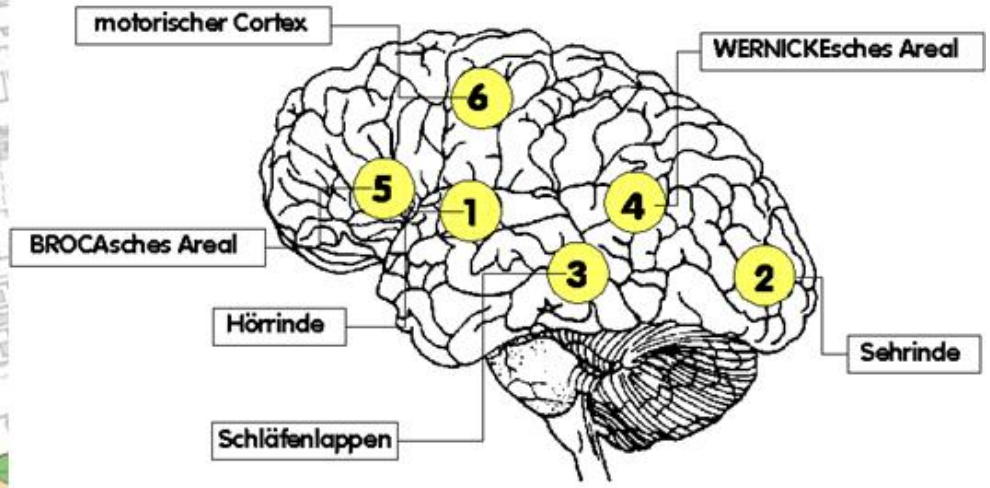

MARTIN"
Period 1840

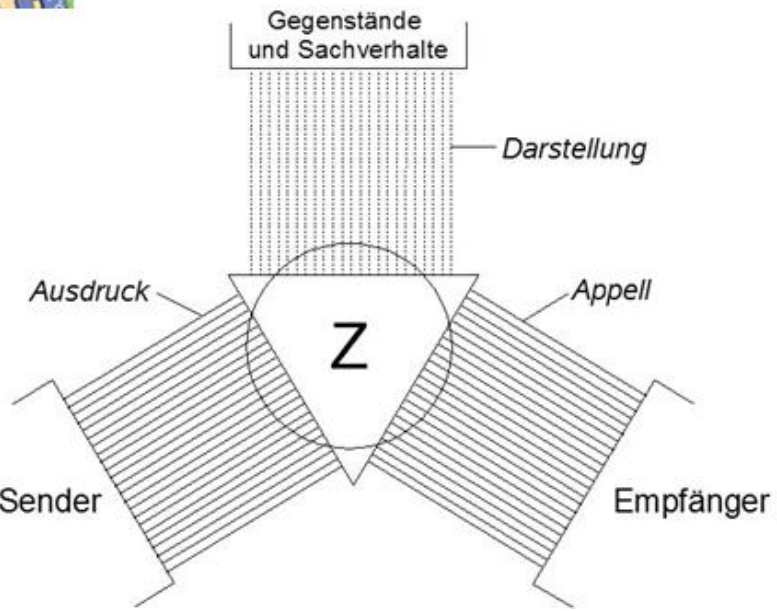





Verwaltungsrat
Personalrat
Geschäftsführung
Frank Zehetner
Assistenz der Geschäftsführung
Astrid Pfahl
Interne Revision / Controlling
Presse- und
Öffentlichkeitsarbeit
Hauptverwaltung
Amt für Ausbildungsförderung
Monika Zerbin
BAföG
DAKA-Darlehen
Studienfinanzierung
Personalwesen /
Organisation
Rechnungswesen
Finanzen /
Verwaltung
EDV
Soziale Dienste
Arbeitssicherheit
Sozialberatung /
Stud. Versicherung
Kultur /
Internationales
Kindertagesstätten in
Düsseldorf und
Mönchengladbach
Gastronomie
Rolf Rumpf
Verwaltung
Mensen in
Düsseldorf, Krefeld und
Mönchengladbach
Cafeterien in
Düsseldorf, Krefeld und
Mönchengladbach
Studentisches Wohnen /
Bau und Technik
Heinz-Walter Pfeiffer
Wohnanlagenverwaltung
Neubauten /
Wohnraumsanierung
Liegenschaftsmanagement in
Düsseldorf, Krefeld und
Mönchengladbach


Uni-Mensa
735
827
Uni-Mitte
735
827
835
836
Münchener
Uni-Süd
735
827
835


motorischer Cortex
WERNICKEsches Areal
BROCAsches Areal
Hörrinde
Sehrinde
Schläfenlappen
6
5
1
4
3
2

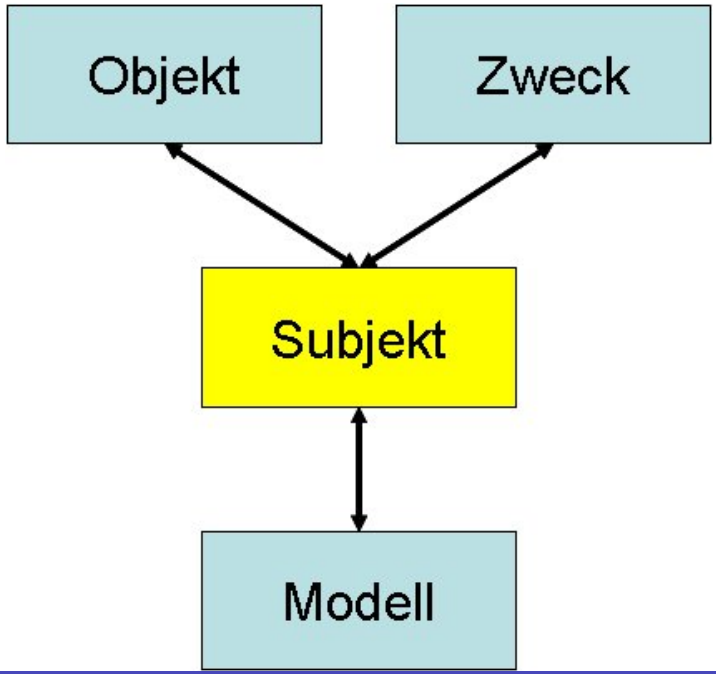

Gegenstände
und Sachverhalte
Darstellung
Ausdruck
Appell
Z
Sender
Empfänger

Modell

- künstlich geschaffen
- materiell oder immateriell
- vereinfachtes Abbild
- zweckgerichtet
- Abstraktion
- Repräsentation
- Modellierungsannahmen

Modell

- künstlich geschaffen
- materiell oder immateriell
- vereinfachtes Abbild
- zweckgerichtet
- Abstraktion
- Repräsentation
- Modellierungsannahmen



Modellierung

Ein Subjekt entwirft zu einem Orginal ein Modell zu einem bestimmten Zweck.

Modellierung natürlicher Sprachen

Formale Sprachen

Formale Sprachen sind Mengen von Wörtern (entspricht in natürlichen Sprachen den **Sätzen**), die ihrerseits aus Zeichen/Symbolen (in natürlichen Sprachen **Wörtern**) aufgebaut sind. Was in der Menge ist, ist ein "grammatisch korrektes Wort", alles andere nicht.

Modellierung natürlicher Sprachen

Formale Sprachen

Formale Sprachen sind Mengen von Wörtern (entspricht in natürlichen Sprachen den **Sätzen**), die ihrerseits aus Zeichen/Symbolen (in natürlichen Sprachen **Wörtern**) aufgebaut sind. Was in der Menge ist, ist ein "grammatisch korrektes Wort", alles andere nicht.

Für "strukturierte" formale Sprachen lassen sich endliche Mengen von Regeln/Grammatiken angeben, die diese beschreiben.

Modellierung natürlicher Sprachen

Formale Sprachen

Formale Sprachen sind Mengen von Wörtern (entspricht in natürlichen Sprachen den **Sätzen**), die ihrerseits aus Zeichen/Symbolen (in natürlichen Sprachen **Wörtern**) aufgebaut sind. Was in der Menge ist, ist ein "grammatisch korrektes Wort", alles andere nicht.

Für "strukturierte" formale Sprachen lassen sich endliche Mengen von Regeln/Grammatiken angeben, die diese beschreiben.

Sprachmodell

Formale Sprachen dienen als Modell für natürliche Sprachen.

Modellierung natürlicher Sprachen

Formale Sprachen

Formale Sprachen sind Mengen von Wörtern (entspricht in natürlichen Sprachen den **Sätzen**), die ihrerseits aus Zeichen/Symbolen (in natürlichen Sprachen **Wörtern**) aufgebaut sind. Was in der Menge ist, ist ein "grammatisch korrektes Wort", alles andere nicht.

Für "strukturierte" formale Sprachen lassen sich endliche Mengen von Regeln/Grammatiken angeben, die diese beschreiben.

Sprachmodell

Formale Sprachen dienen als Modell für natürliche Sprachen.

Wir gehen davon aus, daß alle natürlichen Sprachen durch endlich viele Regeln beschreibbar sind, da wir sie ansonsten nicht sprechen / verstehen könnten.

Modellierung natürlicher Sprachen

Formale Sprachen

Formale Sprachen sind Mengen von Wörtern (entspricht in natürlichen Sprachen den **Sätzen**), die ihrerseits aus Zeichen/Symbolen (in natürlichen Sprachen **Wörtern**) aufgebaut sind. Was in der Menge ist, ist ein "grammatisch korrektes Wort", alles andere nicht.

Für "strukturierte" formale Sprachen lassen sich endliche Mengen von Regeln/Grammatiken angeben, die diese beschreiben.

Sprachmodell

Formale Sprachen dienen als Modell für natürliche Sprachen.

Wir gehen davon aus, daß alle natürlichen Sprachen durch endlich viele Regeln beschreibbar sind, da wir sie ansonsten nicht sprechen / verstehen könnten.

Welche Modellannahmen werden hier implizit gemacht?

Sprachbeschreibung durch Aufzählung aller Wörter

- Peter says that Mary has fallen off the tree.
- Oskar says that Peter says that Mary has fallen off the tree.
- Lisa says that Oskar says that Peter says that Mary has fallen off the tree.
- ...

Sprachbeschreibung durch Aufzählung aller Wörter

- Peter says that Mary has fallen off the tree.
- Oskar says that Peter says that Mary has fallen off the tree.
- Lisa says that Oskar says that Peter says that Mary has fallen off the tree.
- ...

Scheitert bei unendlichen Sprachen.

Aufzählungen erfassen keine Generalisierungen.

Sprachbeschreibung durch Angabe einer Grammatik

Grammatik

- Eine formale Grammatik ist ein generativer Mechanismus zur Erzeugung von Zeichenketten.
- Grammatiken sind endliche Regelsysteme.
- Die Menge aller Ketten, die von einer Grammatik generiert werden, bilden die von der Grammatik beschriebene formale Sprache.

S	$\rightarrow$	NP VP	VP	$\rightarrow$	V	VP	$\rightarrow$	VP and VP
NP	$\rightarrow$	D N	NP	$\rightarrow$	NP and NP	D	$\rightarrow$	the
N	$\rightarrow$	cat	N	$\rightarrow$	dog	V	$\rightarrow$	sleeps
V	$\rightarrow$	dreams						

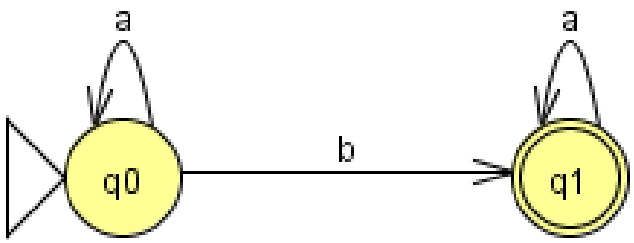
Generiert: the cat sleeps, the dog sleeps, the cat sleeps and dreams,...

aber auch: the cat and the dog sleeps and dreams, ...

Sprachbeschreibung durch Automaten

Automaten

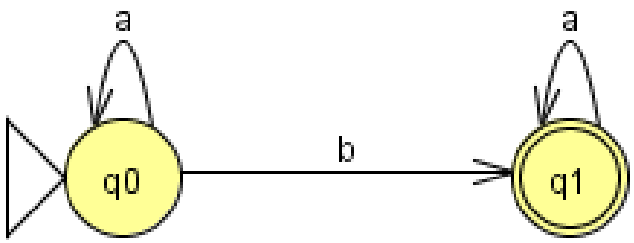
- Ein Automat ist eine abstrakte Maschine, die bestimmte Zeichenketten akzeptiert.
- Die Menge aller Ketten, die von einem Automaten akzeptiert werden, bilden die von dem Automaten beschriebene formale Sprache.



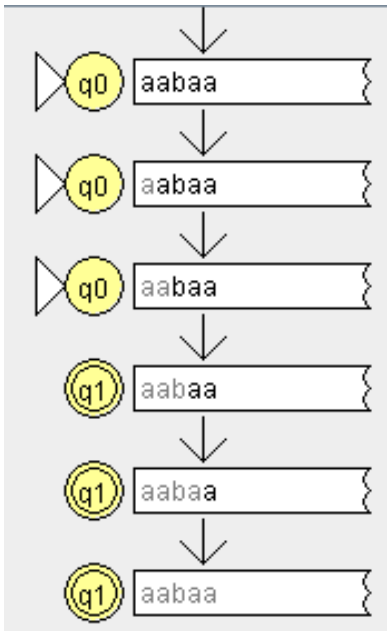
Sprachbeschreibung durch Automaten

Automaten

- Ein Automat ist eine abstrakte Maschine, die bestimmte Zeichenketten akzeptiert.
- Die Menge aller Ketten, die von einem Automaten akzeptiert werden, bilden die von dem Automaten beschriebene formale Sprache.



akzeptiert die Sprache $\{a\}^* \circ \{b\} \circ \{a\}^*$



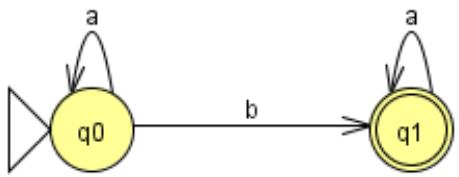
einfachstes Automatenmodell: endliche Automaten

Definition

Ein endlicher Automat ist ein 5-Tupel $(Q, \Sigma, \Delta, q_0, F)$ bestehend aus:

1. Q: Alphabet der Zustände
2. Σ: Eingabealphabet (Q und Σ müssen disjunkt sein)
3. Δ: Übergangsrelation ($\Delta \subseteq Q \times \Sigma \times Q$)
4. q_0: Startzustand ($q_0 \in Q$)
5. F: Menge der Endzustände $F \subseteq Q$.

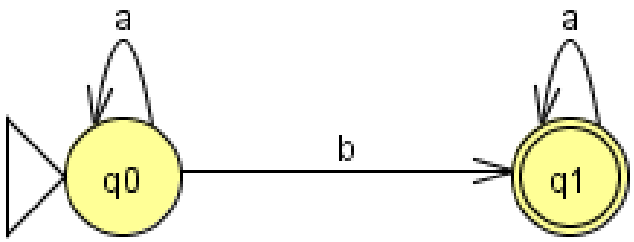
Der Automat heißt deterministisch, wenn die Übergangsrelation Δ eine (partielle) Funktion ist ($\Delta : Q \times \Sigma \to Q$).

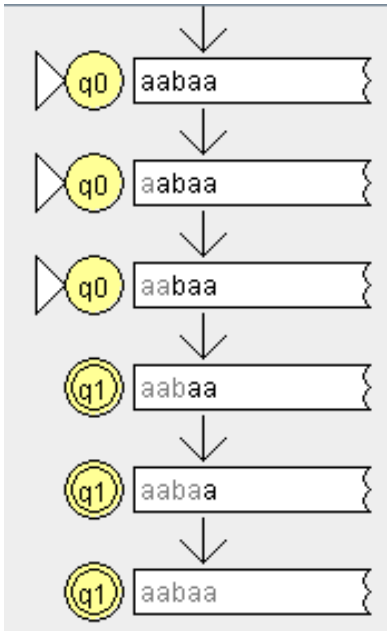


endliche Automaten: Akzeptanz von Wörtern

Ein endlicher Automat akzeptiert ein Wort w, wenn es möglich ist

- beginnend im Startzustand
- das Wort Symbol für Symbol abzuarbeiten, indem man den Zustand gemäß der Übergangsrelation wechselt
- bis das Wort vollständig abgearbeitet ist,
- und wenn man sich am Ende in einem Endzustand befindet.





Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_0, a, q_2),$
  $(q_1, a, q_3), (q_3, a, q_1),$
  $(q_2, b, q_2), (q_2, b, q_4)\}$
- $F = \{q_3, q_4\}$

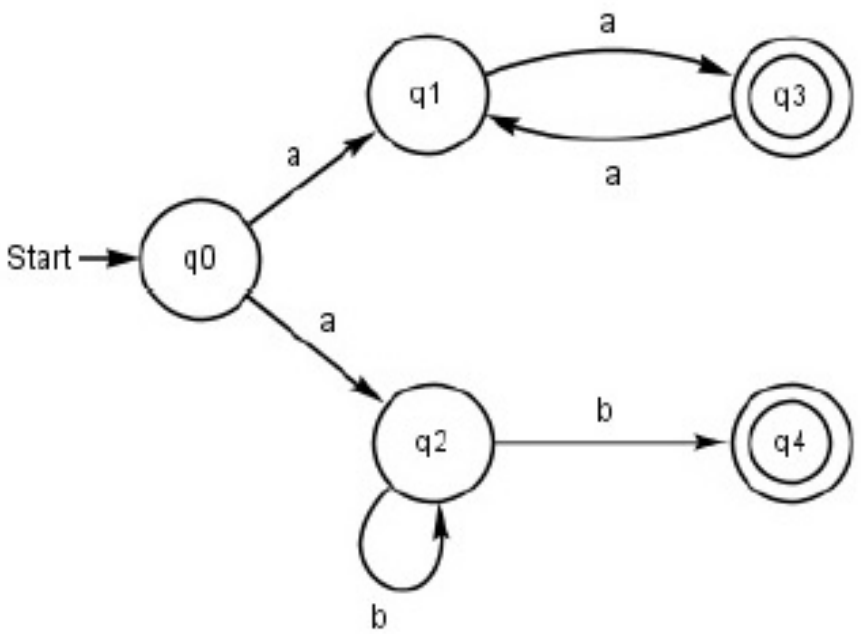
Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_0, a, q_2), (q_1, a, q_3), (q_3, a, q_1), (q_2, b, q_2), (q_2, b, q_4)\}$
- $F = \{q_3, q_4\}$

als Übergangsnetz:



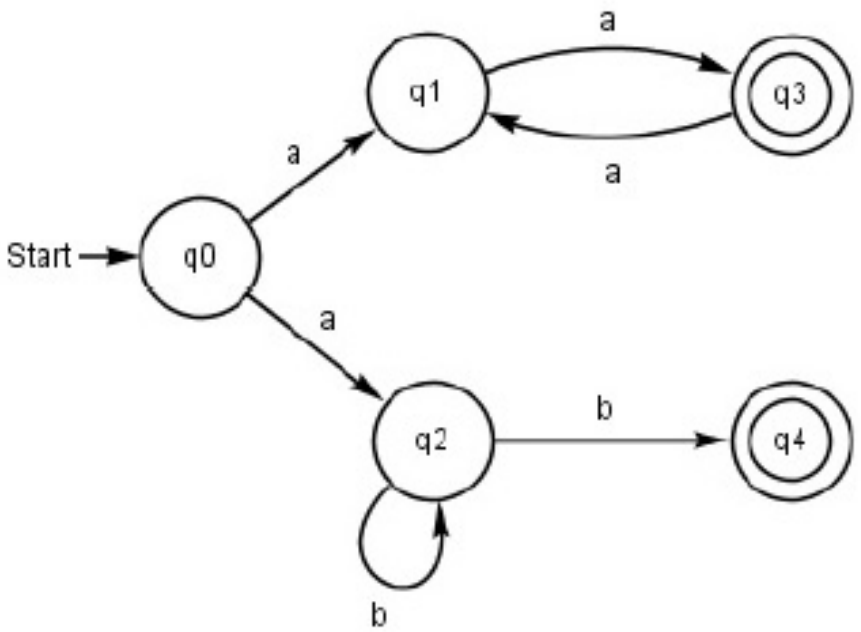
Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_0, a, q_2), (q_1, a, q_3), (q_3, a, q_1), (q_2, b, q_2), (q_2, b, q_4)\}$
- $F = \{q_3, q_4\}$

als Übergangsnetz:



Dieser Automat ist nicht deterministisch

(am Übergangsnetz ablesbar an identisch beschrifteten Kanten, die von demselben Knoten ausgehen)

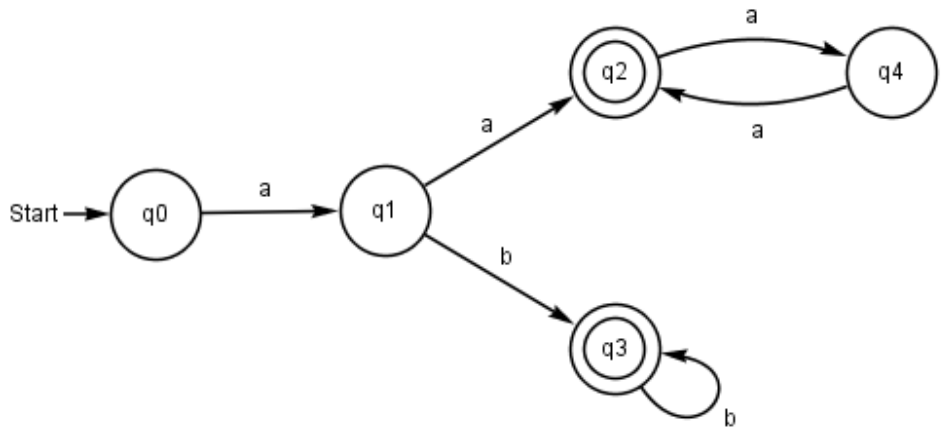
Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$
- $F = \{q_2, q_3\}$

als Übergangsnetz:



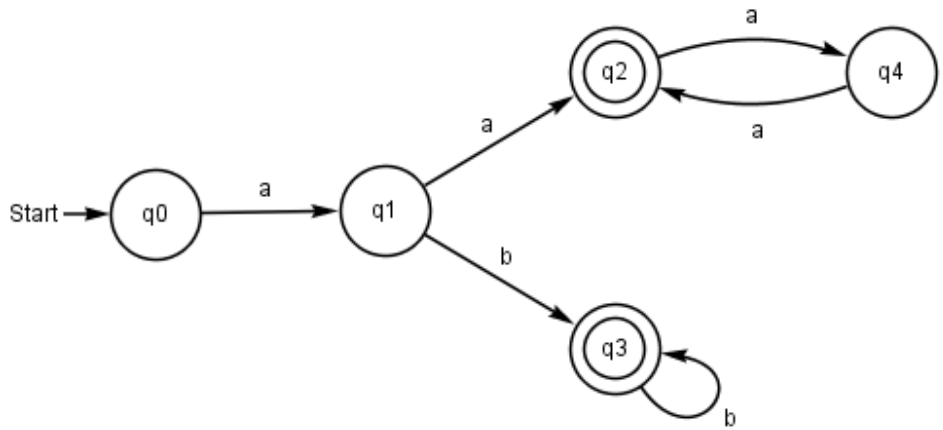
Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$
- $F = \{q_2, q_3\}$

als Übergangsnetz:



Dieser Automat ist deterministisch und akzeptiert dieselbe Sprache wie der Automat der vorangegangenen Folie, nämlich
$\{a\} \circ ((\{a\} \circ (\{a\} \circ \{a\})^*) \cup (\{b\} \circ \{b\}^*))$.

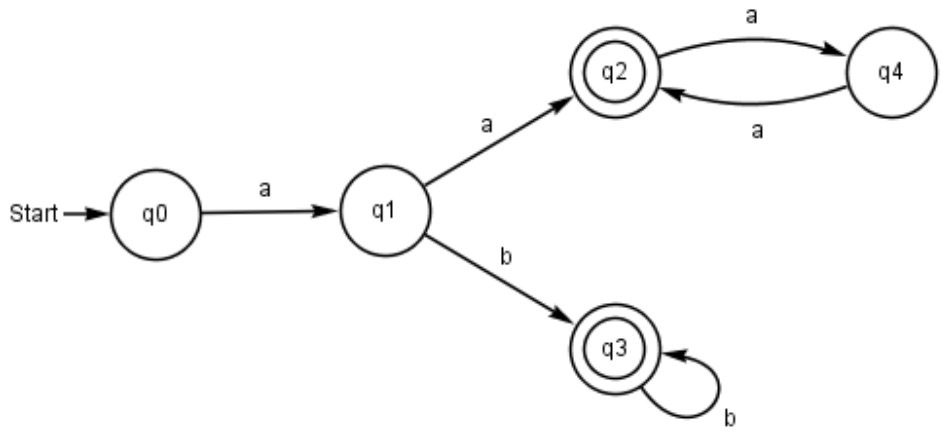
Beispiel: endlicher Automat

als 5-Tupel:

$(Q, \Sigma, \Delta, q_0, F)$ mit

- $Q = \{q_0, q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$
- $F = \{q_2, q_3\}$

als Übergangsnetz:



Dieser Automat ist deterministisch und akzeptiert dieselbe Sprache wie der Automat der vorangegangenen Folie, nämlich
$\{a\} \circ ((\{a\} \circ (\{a\} \circ \{a\})^*) \cup (\{b\} \circ \{b\}^*))$.
Dies ist die Sprache aller Wörter über dem Alphabet $\{a, b\}$, die aus einem a gefolgt von einer beliebigen, nichtleeren Kette von b's oder aus einer nichtleeren Kette von a's gerader Länge bestehen.

Endliche Automaten: Terminologie

- Zwei Automaten, die dieselbe Sprache akzeptieren, heißen äquivalent (Beispiel: die Automaten der letzten beiden Folien sind äquivalent)
- Satz: Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.
- Übergangsrelationen werden häufig als Übergangstabellen dargestellt.
  Beispiel: $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$

	a	b
q_0	q_1	
q_1	q_2	q_3
q_2	q_4	
q_3		q_3
q_4	q_2	

- Ist die Übergangsrelation eines endlichen Automaten eine totale Funktion (steht also in jeder Zelle der Übergangstabelle genau ein Element), so ist der Automat ein endlicher Automat mit vollständiger Übergangsfunktion

Endliche Automaten: Terminologie

- Zwei Automaten, die dieselbe Sprache akzeptieren, heißen äquivalent (Beispiel: die Automaten der letzten beiden Folien sind äquivalent)
- Satz: Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.
- Übergangsrelationen werden häufig als Übergangstabellen dargestellt.
  Beispiel: $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$

	a	b
q_0	q_1	
q_1	q_2	q_3
q_2	q_4	
q_3		q_3
q_4	q_2	

- Ist die Übergangsrelation eines endlichen Automaten eine totale Funktion (steht also in jeder Zelle der Übergangstabelle genau ein Element), so ist der Automat ein endlicher Automat mit vollständiger Übergangsfunktion

Sind endliche Automaten mit vollständiger Übergangsfunktion immer deterministisch?

Endliche Automaten: Terminologie

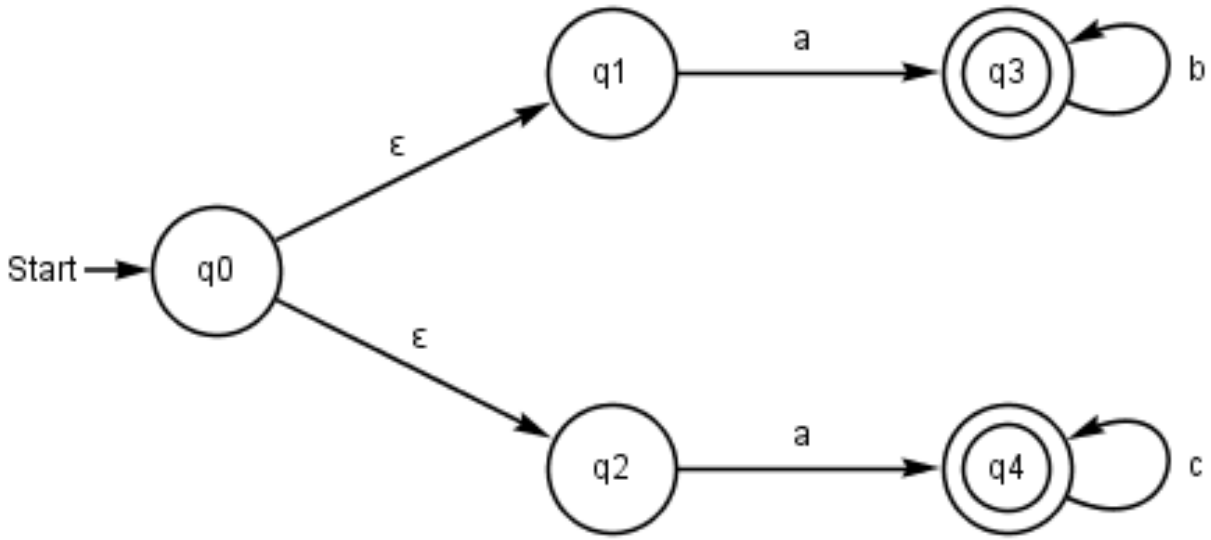
- Zwei Automaten, die dieselbe Sprache akzeptieren, heißen äquivalent (Beispiel: die Automaten der letzten beiden Folien sind äquivalent)
- Satz: Zu jedem nichtdeterministischen endlichen Automaten gibt es einen äquivalenten deterministischen endlichen Automaten.
- Übergangsrelationen werden häufig als Übergangstabellen dargestellt.
  Beispiel: $\Delta = \{(q_0, a, q_1), (q_1, a, q_2), (q_1, b, q_3), (q_3, b, q_3), (q_2, a, q_4), (q_4, a, q_2)\}$

	a	b
q_0	q_1	
q_1	q_2	q_3
q_2	q_4	
q_3		q_3
q_4	q_2	

- Ist die Übergangsrelation eines endlichen Automaten eine totale Funktion (steht also in jeder Zelle der Übergangstabelle genau ein Element), so ist der Automat ein endlicher Automat mit vollständiger Übergangsfunktion

Sind endliche Automaten mit vollständiger Übergangsfunktion immer deterministisch?

endliche Automaten mit ϵ-Übergängen



Zu jedem endlichen Automaten mit ϵ-Übergängen gibt es einen äquivalenten endlichen Automaten ohne ϵ-Übergänge.

Übung

Erstellen Sie endliche Automaten, die die folgenden Sprachen über dem Alphabet $\{a, b\}$ akzeptieren:

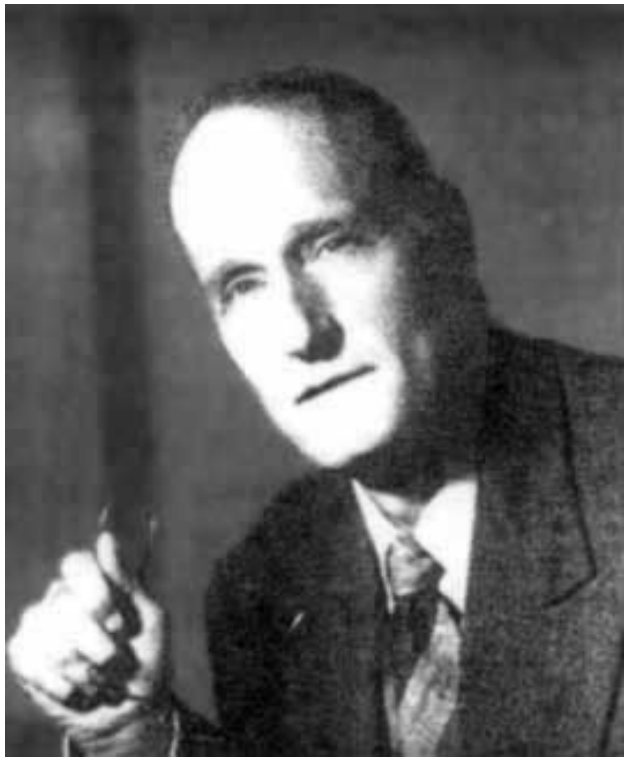
1. die Sprache aller Wörter, die nicht länger als 3 sind.
2. die Sprache aller Wörter, die mit 'ab' beginnen.
3. die Sprache aller Wörter, in denen die Kette 'aa' vorkommt.
4. die Sprache aller Wörter, die ungleich der Kette 'abb' sind.
5. die Sprache aller Wörter, die auf die Kette 'aa' enden.
6. die Sprache aller Wörter, in denen eine gerade Zahl von a's vorkommt.
7. die Sprache aller Wörter, in denen mindestens zwei a's vorkommen.

reguläre Sprache

Gegeben ein Alphabet Σ.

- $\emptyset$ ist eine reguläre Sprache über dem Alphabet Σ.
- $\{\epsilon\}$ ist eine reguläre Sprache über dem Alphabet Σ.
- Für jedes $a \in \Sigma$ ist $\{a\}$ eine reguläre Sprache über dem Alphabet Σ.
- Wenn A und B reguläre Sprachen sind, dann ist auch $A \cup B$ eine reguläre Sprache über dem Alphabet Σ.
- Wenn A und B reguläre Sprachen sind, dann ist auch $A \circ B$ eine reguläre Sprache über dem Alphabet Σ.
- Wenn A eine reguläre Sprachen ist, dann ist auch A^* eine reguläre Sprache über dem Alphabet Σ.
- Nichts sonst ist eine reguläre Sprache über dem Alphabet Σ.

Satz von Kleene



(Stephen C. Kleene, 1909 - 1994)

Jede Sprache, die von einem endlichen Automaten akzeptiert wird, ist regulär und jede reguläre Sprache wird von einem endlichen Automaten akzeptiert.

Endliche Automaten akzeptieren reguläre Sprachen

Theorem (Kleene)

Jede Sprache, die von einem endlichen Automaten akzeptiert wird ist regulär und jede reguläre Sprache wird von einem endlichen Automaten akzeptiert.

Endliche Automaten akzeptieren reguläre Sprachen

Theorem (Kleene)

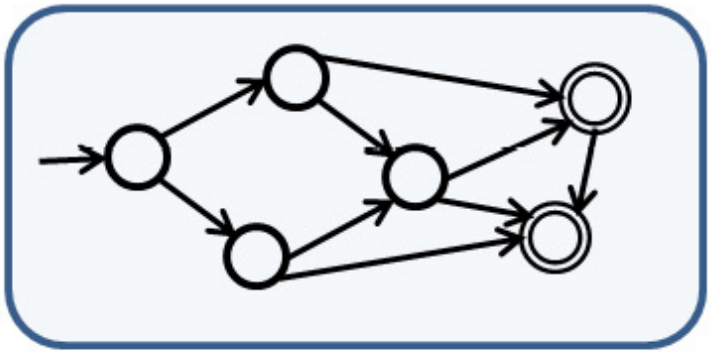
Jede Sprache, die von einem endlichen Automaten akzeptiert wird ist regulär und jede reguläre Sprache wird von einem endlichen Automaten akzeptiert.

Beweisidee (eine Richtung): Zu jeder regulären Sprache gibt es einen endlichen Automaten, der diese akzeptiert:

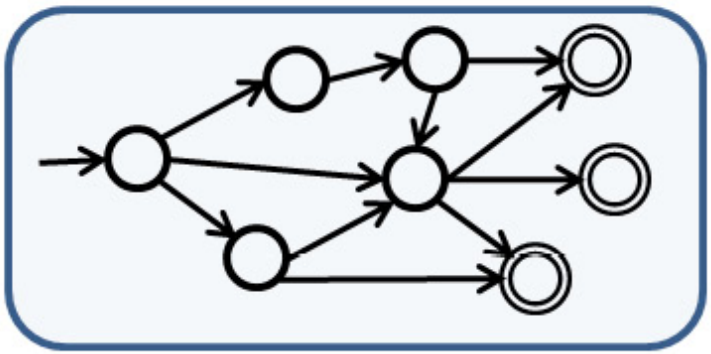
$\emptyset$ $\{\varepsilon\}$ $\{a_i\}$

Beweis des Satzes von Kleene (Fortsetzung)

Wenn A und B zwei reguläre Sprachen sind, die von den Automaten A_A und A_B akzeptiert werden, dann wird die reguläre Sprache $A \cup B$ von dem folgenden Automaten akzeptiert:



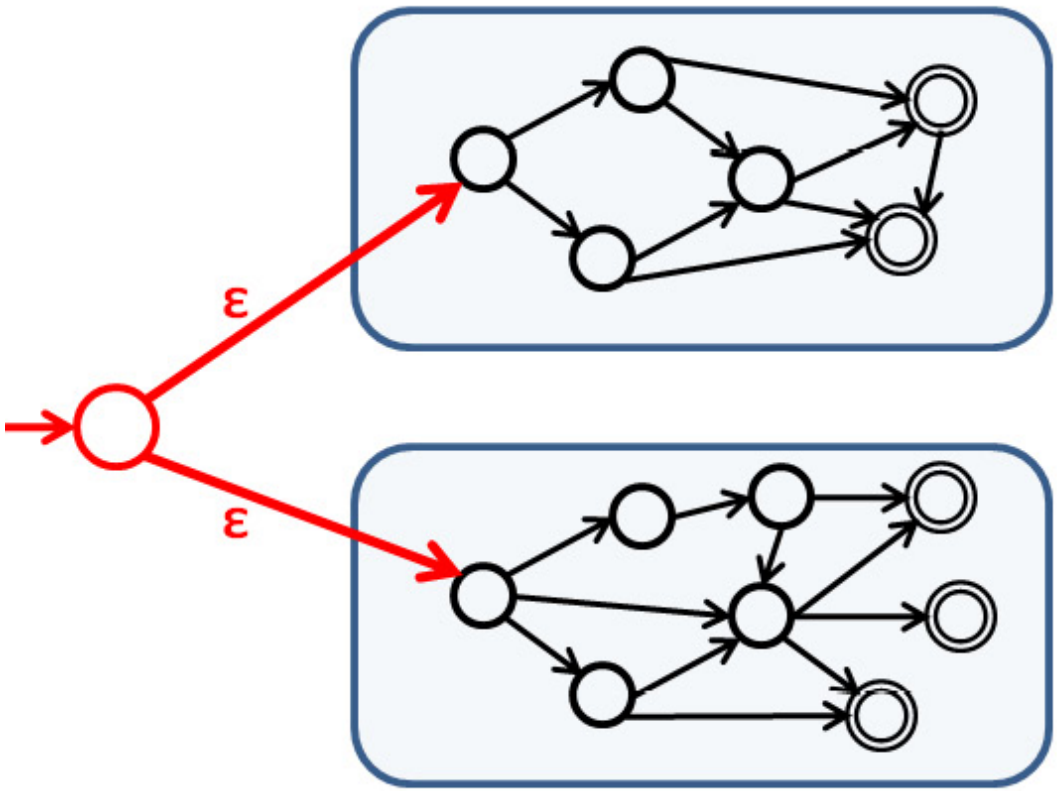
A



B

Beweis des Satzes von Kleene (Fortsetzung)

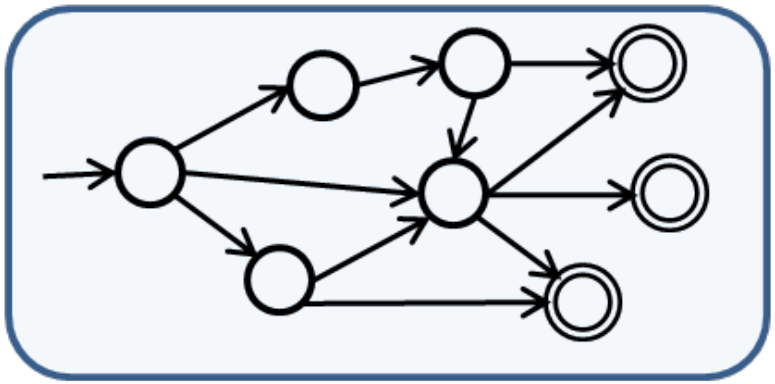
Wenn A und B zwei reguläre Sprachen sind, die von den Automaten A_A und A_B akzeptiert werden, dann wird die reguläre Sprache $A \cup B$ von dem folgenden Automaten akzeptiert:



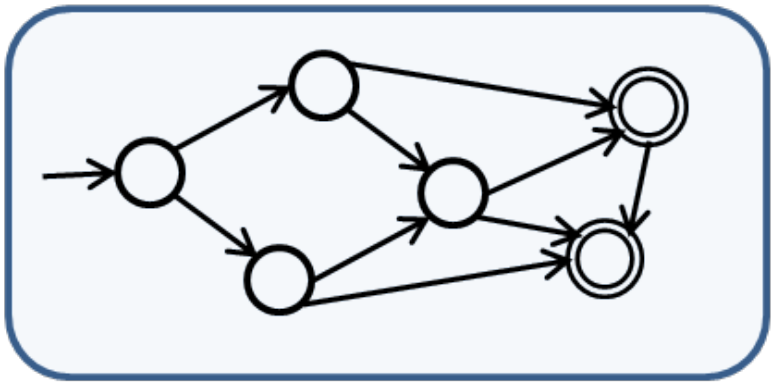
A ∪ B

Beweis des Satzes von Kleene (Fortsetzung)

Wenn A und B zwei reguläre Sprachen sind, die von den Automaten A_A und A_B akzeptiert werden, dann wird die reguläre Sprache $A \circ B$ von dem folgenden Automaten akzeptiert:



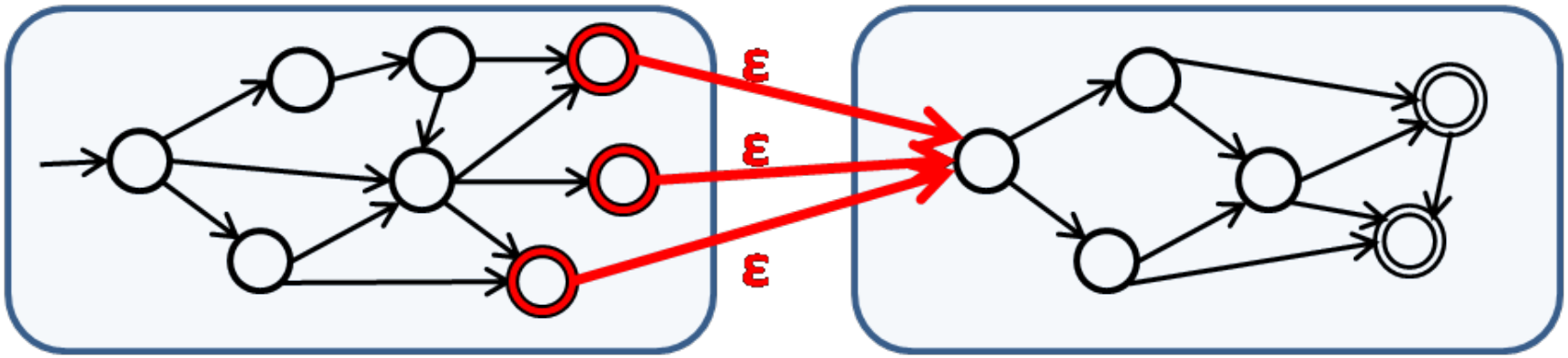
A



B

Beweis des Satzes von Kleene (Fortsetzung)

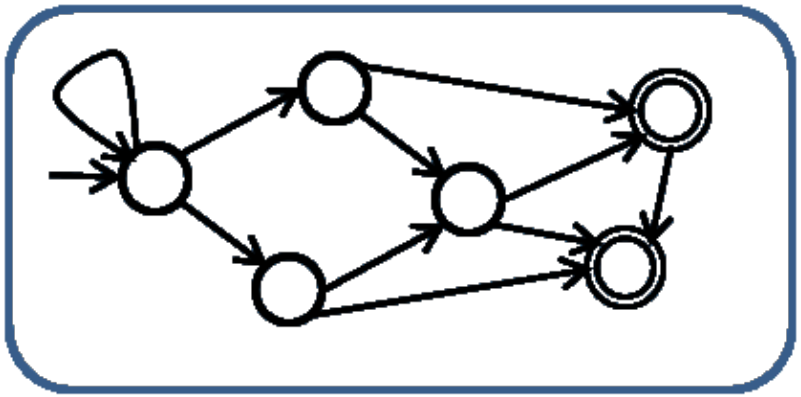
Wenn A und B zwei reguläre Sprachen sind, die von den Automaten A_A und A_B akzeptiert werden, dann wird die reguläre Sprache $A \circ B$ von dem folgenden Automaten akzeptiert:



A ∘ B

Beweis des Satzes von Kleene (Fortsetzung)

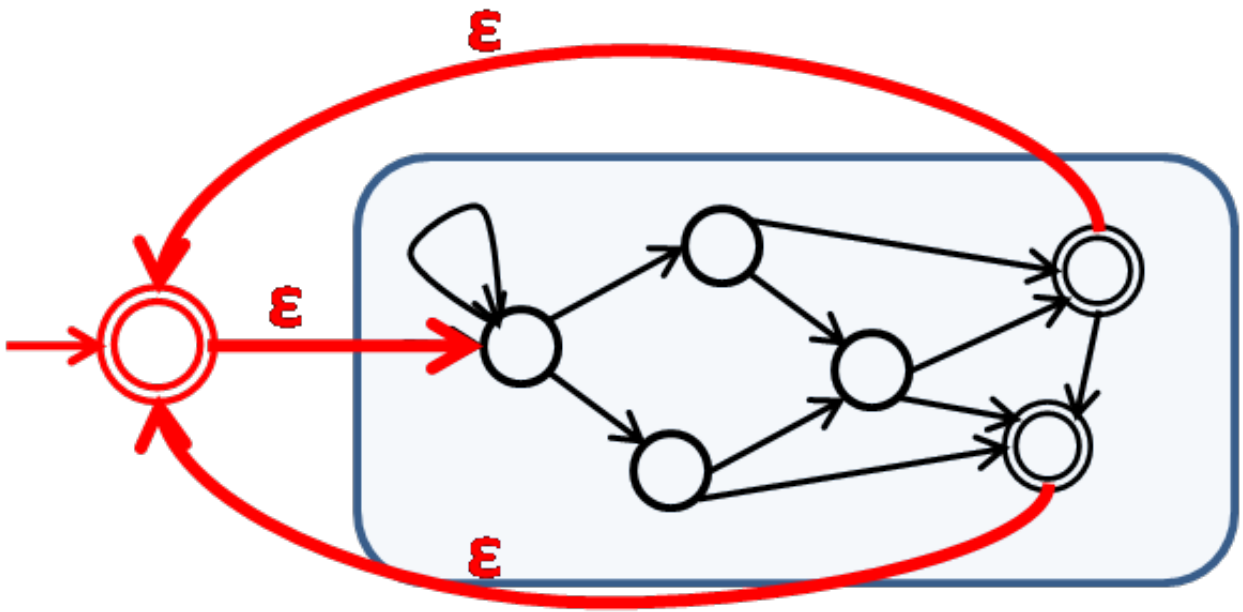
Wenn A eine reguläre Sprache ist, die von dem Automaten A_A akzeptiert wird, dann wird die reguläre Sprache A^* von dem folgenden Automaten akzeptiert:



A

Beweis des Satzes von Kleene (Fortsetzung)

Wenn A eine reguläre Sprache ist, die von dem Automaten A_A akzeptiert wird, dann wird die reguläre Sprache A^* von dem folgenden Automaten akzeptiert:



A^*

Formale Grammatik

Definition

Eine formale Grammatik ist ein 4-Tupel $G = (N, T, S, P)$ aus

- *einem Alphabet von Terminalsymbolen T (häufig auch Σ)*
- *einem Alphabet von Nichtterminalsymbolen N mit $N \cap T = \emptyset$*
- *einem Startsymbol $S \in N$*
- *einer Menge von Regeln/Produktionen $P \subseteq \{(\alpha, \beta) \mid \alpha, \beta \in (N \cup T)^* \; und \; \alpha \notin T^*\}$.*

Für eine Regel (α, β) schreiben wir auch $\alpha \rightarrow \beta$.
Formale Grammatiken werden auch Typ0- oder allgemeine Regelgrammatiken genannt.

S	$\rightarrow$	NP VP	VP	$\rightarrow$	V	NP	$\rightarrow$	D N
D	$\rightarrow$	the	N	$\rightarrow$	cat	V	$\rightarrow$	sleeps

Generiert: the cat sleeps

Terminologie

$G = (\{\text{S,NP,VP,N,V,D,N,EN}\}, \{\text{the, cat, peter, chases}\}, \text{S}, P)$

$$P = \left\{ \begin{array}{lcllcllcl} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V NP} & \text{NP} & \rightarrow & \text{D N} \\ \text{NP} & \rightarrow & \text{EN} & \text{D} & \rightarrow & \text{the} & \text{N} & \rightarrow & \text{cat} \\ \text{EN} & \rightarrow & \text{peter} & \text{V} & \rightarrow & \text{chases} & & & \end{array} \right\}$$

Terminologie

$G = (\{\text{S,NP,VP,N,V,D,N,EN}\}, \{\text{the, cat, peter, chases}\}, \text{S}, P)$

$$P = \left\{ \begin{array}{lcllcllcl} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V NP} & \text{NP} & \rightarrow & \text{D N} \\ \text{NP} & \rightarrow & \text{EN} & \text{D} & \rightarrow & \text{the} & \text{N} & \rightarrow & \text{cat} \\ \text{EN} & \rightarrow & \text{peter} & \text{V} & \rightarrow & \text{chases} & & & \end{array} \right\}$$

"NP VP" ist in einem Schritt ableitbar aus S

Terminologie

$G = (\{\text{S,NP,VP,N,V,D,N,EN}\}, \{\text{the, cat, peter, chases}\}, \text{S}, P)$

$$P = \left\{ \begin{array}{lcllcllcl} \text{S} & \to & \text{NP VP} & \text{VP} & \to & \text{V NP} & \text{NP} & \to & \text{D N} \\ \text{NP} & \to & \text{EN} & \text{D} & \to & \text{the} & \text{N} & \to & \text{cat} \\ \text{EN} & \to & \text{peter} & \text{V} & \to & \text{chases} & & & \end{array} \right\}$$

"NP VP" ist in einem Schritt ableitbar aus S

"the cat chases peter" ist ableitbar aus S:

$$\begin{array}{lll} \text{S} \to \text{NP VP} & \to \text{NP V NP} & \to \text{NP V EN} \\ \to \text{NP V peter} & \to \text{NP chases peter} & \to \text{D N chases peter} \\ \to \text{D cat chases peter} & \to \text{the cat chases peter} & \end{array}$$

Die Menge aller aus dem Startsymbol S ableitbarer Wörter ist die von der Grammatik G erzeugte Sprache $L(G)$.

$$L(G) = \left\{ \begin{array}{ll} \text{the cat chases peter} & \text{peter chases the cat} \\ \text{peter chases peter} & \text{the cat chases the cat} \end{array} \right\}$$

kontextfreie Grammatiken

Eine formale Grammatik in der jede linke Regelseite aus genau einem Nichtterminalsymbol besteht heißt kontextfrei.

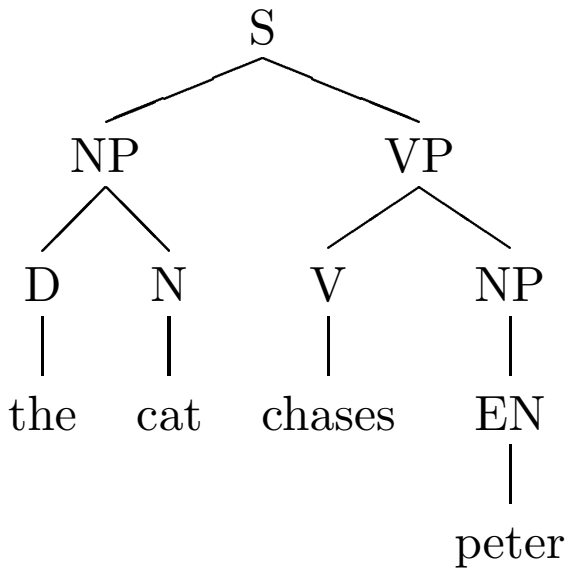
Beispiel:
$G = (\{\text{S,NP,VP,N,V,D,N,EN}\}, \{\text{the, cat, peter, chases}\}, \text{S}, P)$

$$P = \left\{ \begin{array}{rclrclrcl} \text{S} & \rightarrow & \text{NP VP} & \text{VP} & \rightarrow & \text{V NP} & \text{NP} & \rightarrow & \text{D N} \\ \text{NP} & \rightarrow & \text{EN} & \text{D} & \rightarrow & \text{the} & \text{N} & \rightarrow & \text{cat} \\ \text{EN} & \rightarrow & \text{peter} & \text{V} & \rightarrow & \text{chases} & & & \end{array} \right\}$$

Linksableitung (kontextfreie Grammatiken)

Gegeben eine kontextfreie Grammatik G. Eine Ableitung bei der stets das am weitesten links stehende nichtterminale Symbol ersetzt wird, heißt Linksableitung

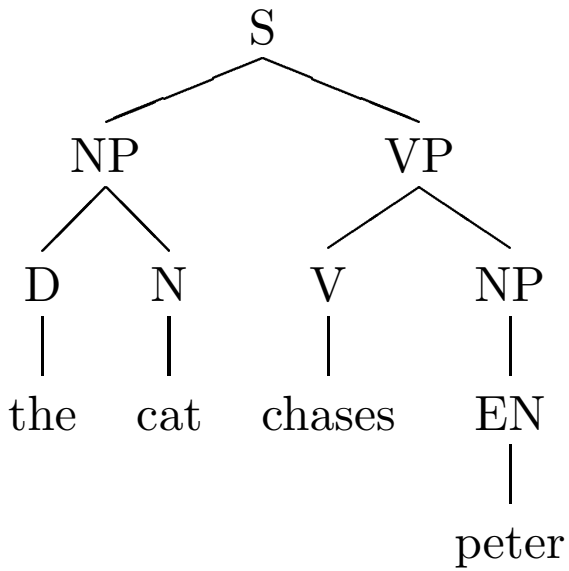
S $\rightarrow$ NP VP $\rightarrow$ D N VP $\rightarrow$ the N VP
$\rightarrow$ the cat VP $\rightarrow$ the cat V NP $\rightarrow$ the cat chases NP
$\rightarrow$ the cat chases EN $\rightarrow$ the cat chases peter



Linksableitung (kontextfreie Grammatiken)

Gegeben eine kontextfreie Grammatik G. Eine Ableitung bei der stets das am weitesten links stehende nichtterminale Symbol ersetzt wird, heißt Linksableitung

S → NP VP → D N VP → the N VP
→ the cat VP → the cat V NP → the cat chases NP
→ the cat chases EN → the cat chases peter



Zu jeder Linksableitung gibt es genau einen Ableitungsbaum und zu jedem Ableitungsbaum gibt es genau eine Linksableitung.

Chomskyhierarchie

- Wenn man die Form der Regeln einschränkt erhält man Teilmengen der Menge aller durch eine Grammatik erzeugten Sprachen.

Chomskyhierarchie

- Wenn man die Form der Regeln einschränkt erhält man Teilmengen der Menge aller durch eine Grammatik erzeugten Sprachen.
- Die Chomskyhierarchie ist eine Hierarchie über die Regelbedingungen (den verschiedenen Sprachklassen entsprechen Einschränkungen über die rechten und linken Regelseiten).

Chomskyhierarchie

- Wenn man die Form der Regeln einschränkt erhält man Teilmengen der Menge aller durch eine Grammatik erzeugten Sprachen.
- Die Chomskyhierarchie ist eine Hierarchie über die Regelbedingungen (den verschiedenen Sprachklassen entsprechen Einschränkungen über die rechten und linken Regelseiten).
- Die Chomskyhierarchie reflektiert eine spezielle Form der Komplexität, andere Kriterien sind denkbar und führen zu anderen Hierarchien.

Chomskyhierarchie

- Wenn man die Form der Regeln einschränkt erhält man Teilmengen der Menge aller durch eine Grammatik erzeugten Sprachen.
- Die Chomskyhierarchie ist eine Hierarchie über die Regelbedingungen (den verschiedenen Sprachklassen entsprechen Einschränkungen über die rechten und linken Regelseiten).
- Die Chomskyhierarchie reflektiert eine spezielle Form der Komplexität, andere Kriterien sind denkbar und führen zu anderen Hierarchien.
- Die Sprachklassen der Chomskyhierarchie sind in der Informatik intensiv untersucht worden (Berechnungskomplexität, effektive Parser).

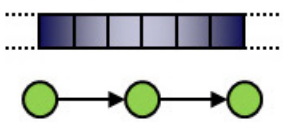
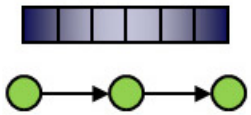
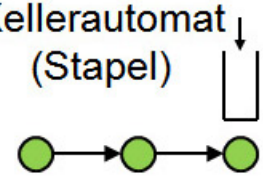
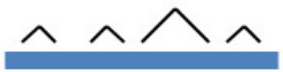
Chomskyhierarchie

- Wenn man die Form der Regeln einschränkt erhält man Teilmengen der Menge aller durch eine Grammatik erzeugten Sprachen.
- Die Chomskyhierarchie ist eine Hierarchie über die Regelbedingungen (den verschiedenen Sprachklassen entsprechen Einschränkungen über die rechten und linken Regelseiten).
- Die Chomskyhierarchie reflektiert eine spezielle Form der Komplexität, andere Kriterien sind denkbar und führen zu anderen Hierarchien.
- Die Sprachklassen der Chomskyhierarchie sind in der Informatik intensiv untersucht worden (Berechnungskomplexität, effektive Parser).
- Für Linguisten ist die Chomsky Hierarchie besonders interessant, da sie die Form der Regeln zentral stellt, und somit Aussagen über Grammatikformalismen zuläßt.

Noam Chomsky

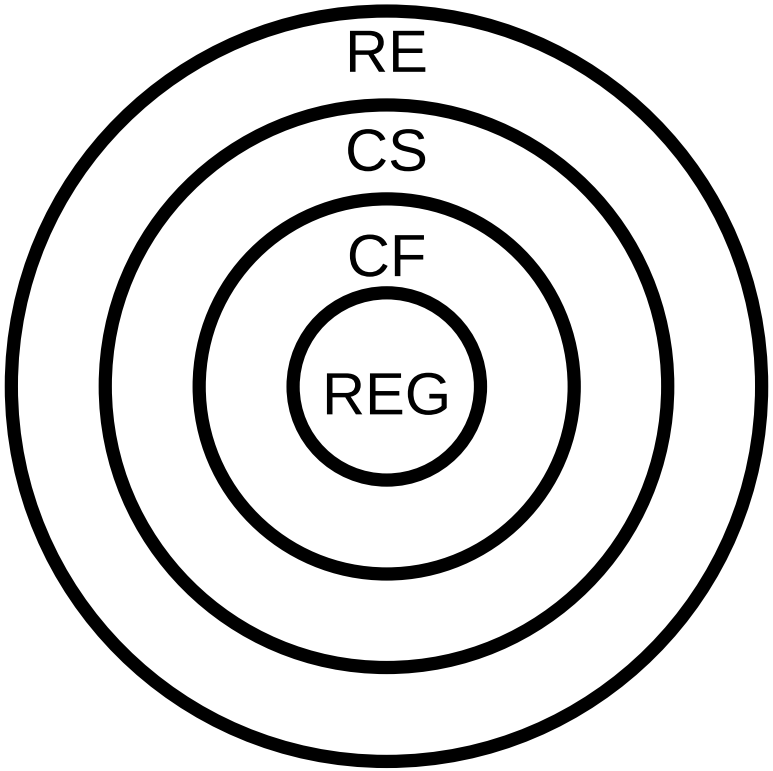
Noam Chomsky
(∗ 7.12.1928, Philadelphia)
Noam Chomsky, *Three Models for the Description of Language*,
IRE Transactions on Information Theory (1956).

Chomsky-Hierarchie & Automaten (grober Überblick)

Sprache	*Automat*	*Grammatik*	*Erkennung*	*Abhängigkeit*
rekursiv aufzählbar	Turing Maschine 	unbeschränkt Baa → ε	unentscheidbar	beliebig
kontext-sensitiv	linear gebunden 	kontext-sensitiv γAδ → γβδ	NP-vollständig 	überkreuzt 
kontext-frei	Kellerautomat (Stapel) 	kontextfrei C → bABa	polynomiell 	eingebettet 
regulär	endlicher Automat 	regulär A → bA	linear 	strikt lokal 

Chomskyhierarchie: Hauptsatz

$$\mathbf{REG} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{RE}$$



REG: reguläre Sprachen, CF: kontextfreie Sprachen, CS: kontextsensitive Sprachen, RE: rekursiv-aufzählbare Sprachen