# Einführung in die Computerlinguistik – Satz von Kleene

Dozentin: Wiebke Petersen

17.11.2009

# Satz von Kleene

(Stephen C. Kleene, 1909 - 1994)

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

# Wiederholung: reguläre Sprachen

## RE: syntax

The set of regular expressions $RE_\Sigma$ over an alphabet $\Sigma = \{a_1, \dots, a_n\}$ is defined by:

- $\underline{\emptyset}$ is a regular expression.
- $\epsilon$ is a regular expression.
- $a_1, \dots, a_n$ are regular expressions
- If $a$ and $b$ are regular expressions over $\Sigma$ then
  - $(a + b)$
  - $(a \bullet b)$
  - $(a^\star)$

  are regular expressions too.

# Wiederholung: reguläre Sprachen

## RE: semantics

Each regular expression $r$ over an alphabet $\Sigma$ describes a formal language $L(r) \subseteq \Sigma^*$.

Regular languages are those formal languages which can be described by a regular expression.

The function $L$ is defined inductively:

- $L(\underline{\emptyset}) = \emptyset$, $L(\epsilon) = \{\epsilon\}$, $L(a_i) = \{a_i\}$
- $L(a + b) = L(a) \cup L(b)$
- $L(a \bullet b) = L(a) \circ L(b)$
- $L(a^\star) = L(a)^*$

# Finite-state automatons accept regular languages

**Theorem (Kleene)**

*Every language accepted by a DFSA is regular and every regular language is accepted by some DFSA.*

# Finite-state automatons accept regular languages

**Theorem (Kleene)**

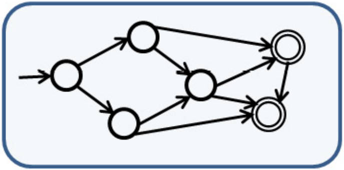
*Every language accepted by a DFSA is regular and every regular language is accepted by some DFSA.*

**proof idea (one direction):** Each regular language is accepted by a NDFSA (and therefore by a DFSA):

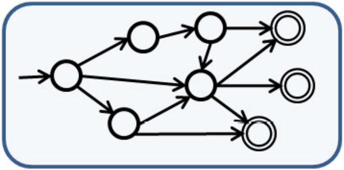

$L(\emptyset)$ $L(\epsilon)$ $L(a_i)$

# Proof of Kleene's theorem (cont.)

If $R_1$ and $R_2$ are two regular expressions such that the languages $L(R_1)$ and $L(R_2)$ are accepted by the automatons $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively, then $L(R_1 + R_2)$ is accepted by:



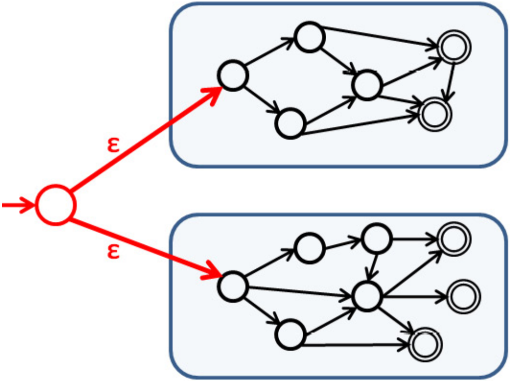
$L(R_1)$



$L(R_2)$

# Proof of Kleene's theorem (cont.)

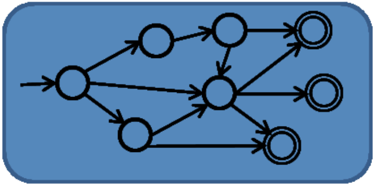
If $R_1$ and $R_2$ are two regular expressions such that the languages $L(R_1)$ and $L(R_2)$ are accepted by the automatons $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively, then $L(R_1 + R_2)$ is accepted by:



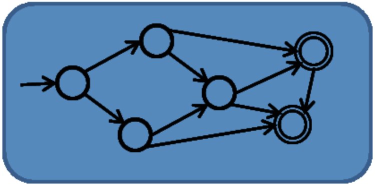

$L(R_1{+}R_2)$

# Proof of Kleene's theorem (cont.)

$L(R_1 \bullet R_2)$ is accepted by:



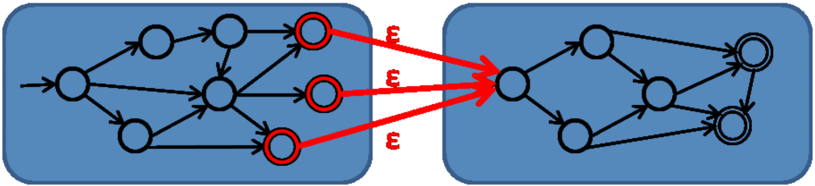
L(R₁)



L(R₂)

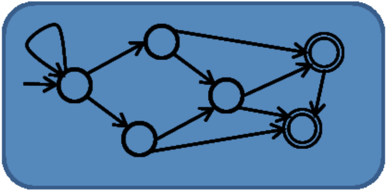
# Proof of Kleene's theorem (cont.)

$L(R_1 \bullet R_2)$ is accepted by:




L(R₁•R₂)

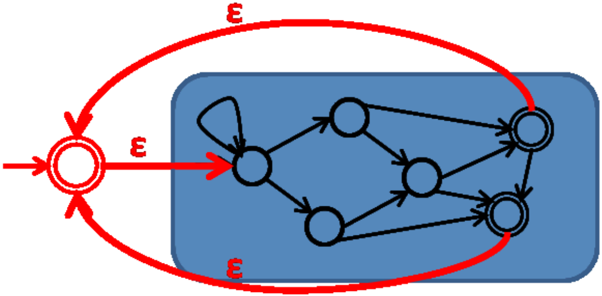
$L(R_1^*)$ is accepted by:



L(R₁)

# Proof of Kleene's theorem (cont.)

$L(R_1^*)$ is accepted by:




L(R₁*)

# Abschlußeigenschaften regulärer Sprachen

## Theorem

1. *If $L_1$ and $L_2$ are two regular languages, then*
   - *the union of $L_1$ and $L_2$ ($L_1 \cup L_2$) is a regular language too.*
   - *the intersection of $L_1$ and $L_2$ ($L_1 \cap L_2$) is a regular language too.*
   - *the concatenation of $L_1$ and $L_2$ ($L_1 \circ L_2$) is a regular language too.*
2. *The complement of every regular language is a regular language too.*
3. *If $L$ is a regular language, then $L^*$ is a regular language too.*

## Beweisidee

**a** Die Aussage für die Vereinigung, die Konkatenation und den Kleeneschen Stern folgt unmittelbar aus dem Satz von Kleene.

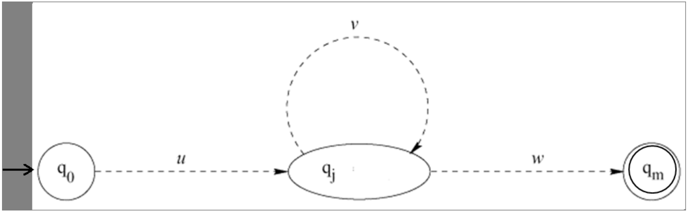
**b** Das Komplement einer regulären Sprache $L$ wird wie folgt konstruiert: (1) konstruiere einen deterministischen, endlichen Automaten mit vollständiger Übergangsfunktion, der $L$ akzeptiert. (2) Wechsle alle Nichtend- zu Endzuständen und umgekehrt. Der resultierende Automat akzeptiert $\bar{L}$.

**c** Die Aussage über den Schnitt zweier regulärer Sprachen folgt aus der Aussage über die Vereinigung, das Komplement und das Gesetz von De Morgan.

**Lemma (Pumping-Lemma)**

*Sei $L$ eine unendliche reguläre Sprache, dann gilt für jedes genügend lange Wort $z \in L$, daß es so in Teilworte $z = uvw$ $(u, w \in \Sigma^*, u, w \in \Sigma^+)$ zerlegt werden kann, dass jedes der Worte $uv^iw \in L$ $(i \geq 0)$ ein Wort der Sprache $L$ ist.*

**Beweisidee:**




Wenn ein Wort länger ist, als der Automat Zustände hat, dann muß bei der Verarbeitung des Wortes ein Zustand zweimal besucht werden. Es gibt somit eine Schleife, die beliebig oft durchlaufen werden kann; das Wort kann in dem Schleifenbereich aufgepumpt werden.

# $L = \{a^n b^n \ : \ n \geq 0\}$ ist nicht regulär

- $L = \{a^n b^n \ : \ n \geq 0\}$:
  $L$ ist unendlich. Wäre $L$ regulär, dann müßte es für genügend lange Worte die geforderte pumpbare Zerlegung geben: aber

# $L = \{a^n b^n \ : \ n \geq 0\}$ ist nicht regulär

- $L = \{a^n b^n \ : \ n \geq 0\}$:
  $L$ ist unendlich. Wäre $L$ regulär, dann müßte es für genügend lange Worte die geforderte pumpbare Zerlegung geben: aber
  1. das pumpbare Teilwort kann nicht nur aus $a$'s bestehen, sonst würden beim Pumpen zuviele $a$'s entstehen
     $(aa(aa)^2bbbb = aaaaaabbbb)$ .
  2. das pumpbare Teilwort kann nicht nur aus $b$'s bestehen, sonst würden beim Pumpen zuviele $b$'s entstehen.
     $(aaaab(bb)^2b = aaaabbbbbb)$
  3. das pumpbare Teilwort kann nicht aus $a$'s und $b$'s bestehen, da beim Pumpen die Sortierung der $a$'s und $b$'s verloren ginge.
     $(aaa(ab)^2bbb = aaaababbbb)$

## Hausaufgabe (Abgabe bis zum 24.11.2009; BN: entweder Aufgabe 1 oder Aufgabe 2 und 3)

1. Beschreiben sie mit ihren eigenen Worten, wie die Automaten für die Sprachen
   1. $L(R_1 + R_2)$,
   2. $L(R_1 \bullet R_2)$ und
   3. $L(R_1^\star)$

   systematisch aus den Automaten für die Sprachen $L(R_1)$ und $L(R_2)$ konstruiert werden können. (Wenn Ihnen die allgemeine Beschreibung schwerfällt, dann wählen Sie bitte als Beispiel zwei reguläre Ausdrücke $R_1$ und $R_2$ und bilden für dieses Beispiel systematisch $L(R_1 + R_2)$, $L(R_1 \bullet R_2)$ und $L(R_1^\star)$).
2. Bilden Sie mithilfe des auf den Folien angedeuteten Verfahrens das Komplement im Universum $L((a|b)^*)$ von
   1. $L(ab)$,
   2. $L(ba^*)$
3. Bilden Sie einen Automaten, der die Schnittmenge der beiden Sprachen $L(b^*a^*)$ und $L(ba^+b^*)$ akzeptiert.