Lektion 16

Alle Sprachen, die durch reguläre Ausdrücke beschrieben werden können, werden als *regulären Sprachen* gezeichnet. In dieser Lektion geht es darum, wie gezeigt werden kann, dass eine Sprache zu den regulären Sprachen gehört und wie auch bewiesen werden kann, dass eine Sprache nicht regulär ist.

9.4 Klasse der regulären Sprache

Definition 9.24 - Reguläre Sprache

Eine **Sprache** L heißt **regulär**, wenn es einen regulären Ausdruck R gibt, der die Sprache L beschreibt, d.h. für den $L = L(R)$ gilt.

Anmerkung

In den vorangehenden Lektionen haben Sie die Zusammenhänge zwischen der Beschreibung von Sprachen durch reguläre Ausdrücke, DEAs und ε-NEAs und Chomsky-Typ-3-Grammatiken kennengelernt. Somit ergibt sich folgendes Gesamtbild für die Klasse der *regulären Sprachen*.

Eigenschaft 9.25 - Reguläre Sprachen

Klasse der **regulären Sprachen**

= Klasse der von **regulären Ausdrücken** dargestellten Sprachen

= Klasse der von **DEAs** akzeptierten Sprachen

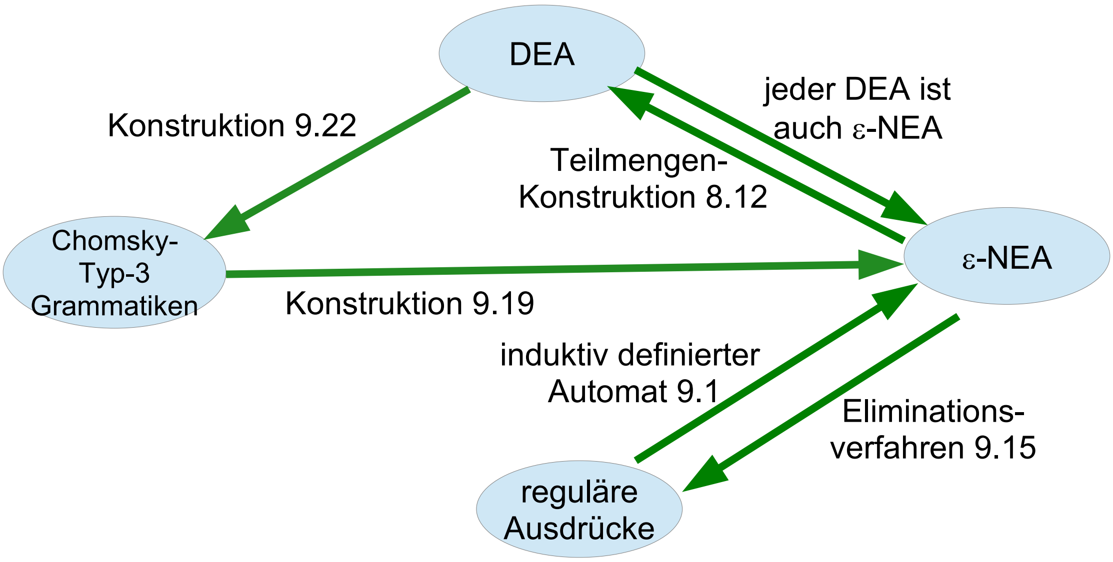
= Klasse der von **ε-NEAs** akzeptierten Sprachen

= Klasse der durch **Chomky-Typ-3-Grammatiken** generierbaren Sprachen

Begründung

Durch die bereits vorgestellten Umwandlungsverfahren lässt sich direkt oder indirekt jede dieser Beschreibungsformen in ein der anderen Formen äquivalent umwandeln, so wie wie hier in der Übersicht dargestellt:

Abbildung 9.26 - Beschreibungsmethoden für reguläre Sprachen



Zu jeden regulären Ausdruck kann beispielsweise auch eine Chomky-Typ-3-Grammatik gebildet werden, indem für den regulären Ausdruck zunächst ein ε-NEA gebildet wird, dieser ε-NEA mit der Teilmengenkonstruktion in einen DEA umgewandelt wird und dann daraus eine Chomsky-Typ-3-Grammatik abgleitet wird.

Nachweis für Regularität von Sprachen

Wie kann gezeigt werden, dass eine Sprache regulär ist? Wenn Sie einen regulären Ausdruck, einen DEA, einen ε-NEA oder eine Chomsky-Typ-3-Grammatik für die Sprache angeben können, ist sie regulär. Da alles diese Beschreibungsformen gleich mächtig sind, ist dann klar, dass die Sprache regulär ist. Im folgenden Teilkapitel werden Sie noch eine weitere Möglichkeit kennenlernen, um indirekt zu zeigen, dass eine Sprache regulär ist.

Aufgabe 9.27 - Reguläre Sprachen

Welche der folgenden Sprachen sind regulär?

$L_1 = \{ 0^n 1^{2k} \mid n \geq 0,\ \ k \geq 0\}$

$L_2 = \{ w \in \{0,1\}^* \mid w$ enthält eine ungerade Zahl von Nullen und eine ungerade Zahl von Einsen$\}$

$L_3 = \{ 0^n 110^n \mid n \geq 0 \}$

9.5 Abschlusseigenschaften regulärer Sprachen

Eine weitere Möglichkeit zu zeigen, dass eines Sprache regulär ist, besteht darin, dass man die nachfolgend vorgestellten *Abschlusseigenschaften* regulärer Sprachen verwendet. Diese Eigenschaften geben an, bei welchen Operationen auf Sprachen aus regulären Sprachen wieder reguläre Sprachen entstehen.

Wenn man zeigen kann, dass eine Sprache mit diesen Operationen aus Sprachen gebildet werden kann, von denen schon bekannt ist, dass sie regulär sind, ist sie somit auch regulär.

Eigenschaft 9.28 - Abschlusseigenschaften regulärer Sprachen

Sind L_1 und L_2 reguläre Sprachen, dann sind auch folgende Sprachen regulär:

$L_1 \cup L_2$	Vereinigung der Sprachen
$L_1 \cdot L_2$	Konkatenation der Sprachen
L_1^*	Kleene'sche-Hülle der Sprache
$L_1 \cap L_2$	Durchschnitt der Sprachen
$\overline{L_1}$	Komplement der Sprache ($\overline{L_1} = \Sigma^* \setminus L_1$, alle Wörter, die nicht in L_1 enthalten sind)

Beweis

- **Fälle $L_1 \cup L_2$, $L_1 \cdot L_2$ und L_1^***: Hier kann einfach über reguläre Ausdrücke argumentiert werden:

  Wenn L_1 und L_2 regulär sind, dann gibt es reguläre Ausdrücke R_1 bzw. R_2, die diese Sprachen darstellen.

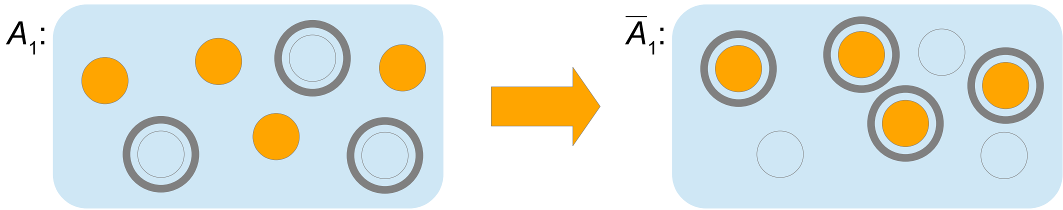
  Dann ist

$R_1 \mid R_2$,	regulärer Ausdruck für die Vereinigung $L_1 \cup L_2$
R_1R_2	regulärer Ausdruck für die Konkatenation $L_1 \cdot L_2$
$(R_1)^*$	regulärer Ausdruck für L_1^*

  Somit sind auch die Sprachen $L_1 \cup L_2$, $L_1 \cdot L_2$ und L_1^* regulär, da auch reguläre Ausdrücke dafür existieren.

- **Fall $\overline{L_1}$**: Der Beweis kann hier mittels DEAs geführt werden. Ist die Sprache L_1 regulär, dann gibt es einen DEA A_1, der die Sprache akzeptiert.

  Aus diesem Automaten kann ein DEA $\overline{A_1}$ gebildet werden, indem die Endzustände zu Nicht-Endzuständen gewandelt werden und umgekehrt.



$\overline{A}_1$ ist auch ein DEA, der genau die Wörter akzeptiert, die von A_1 nicht akzeptiert werden, d.h. die akzeptierte Sprache des Automaten ist $\overline{L}_1$. Somit existiert ein DEA für die Sprache $\overline{L}_1$, also ist die Sprache regulär.

- **Fall $L_1 \cap L_2$** : Eine naheliegende Argumentation über reguläre Ausdrücke ist hier nicht möglich, das es keinen entsprechenden Operator bei regulären Ausdrücken gibt.

  Diese Abschlusseigenschaft kann auf zwei Arten bewiesen werden: mit Hilfe der schon gezeigten Abschlusseigenschaften für reguläre Sprachen oder auch direkt mittels DEAs.

  - **Beweisvariante 1 mittels Abschlusseigenschaften**: Für Mengen gilt folgende Eigenschaft:

    $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$

    Da reguläre Sprachen unter Vereinigung und Komplementbildung abgeschlossen sind (s.o.) ist somit auch $L_1 \cap L_2$ regulär.

  - **Beweisvariante 2 mittels DEAs**: Sind die Sprachen L_1 und L_2 regulär, dann gibt es DEAs $A_1 = (Z_1, \Sigma, \delta_1, z_{01}, E_1)$ und $A_2 = (Z_2, \Sigma, \delta_2, z_{02}, E_2)$, die diese Sprachen akzeptieren.

    Die Kernidee besteht nun darin, dass man beide Automaten parallel ausführt, um ein Wort *w* zu prüfen. Erreichen beide Automaten mit *w* einen Endzustand, dann liegt das Wort in der Schnittmenge der Sprachen.

    Die parallele Ausführung von Automaten kann über einen sog. *Produkt-automaten* $A_1 \times A_2$ formalisiert werden. Die Zustände des Produktautomaten sind Paare aus einem Zustand von A_1 und einem Zustand von A_2.

    $A_1 \times A_2 = (Z_1 \times Z_2, \Sigma, \delta_{12}, (z_{01}, z_{02}), E_{12})$ wobei

    $Z_1 \times Z_2$: Zustände sind alle Paare (z_1, z_2) aus einem Zustand z_1 des ersten Automaten und einem Zustand z_2 des zweiten Automaten:

    Σ Eingabealphabet (ist bei beiden Automaten gleich)

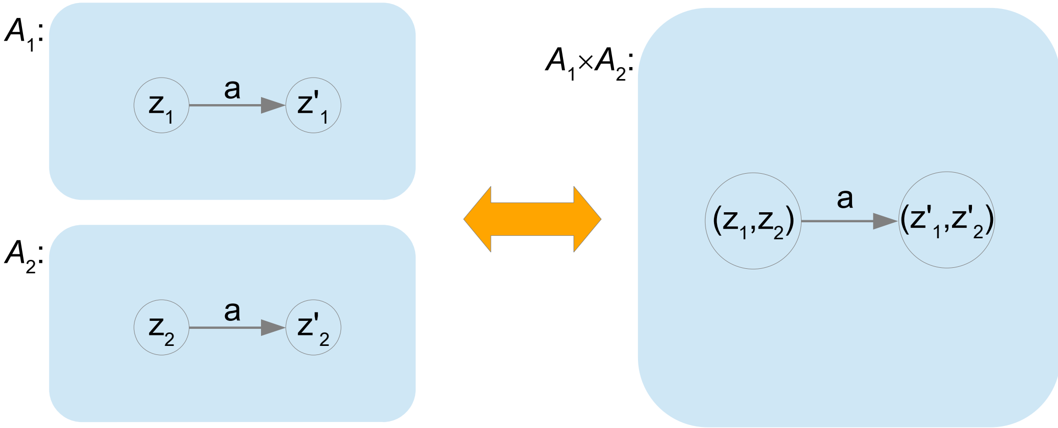
    δ_{12} Zustandsübergangsfunktion:
    für alle Zeichen $a \in \Sigma$ und alle Zustände $z_1 \in Z_1$, $z_2 \in Z_2$ gilt

    $\delta_{12}((z_1, z_2), a) = (z'_1, z'_2)$ genau dann, wenn

    $\delta_1(z_1, a) = z'_1$ und

    $\delta_2(z_2, a) = z'_2$

Veranschaulichung für Zustandsübergang im Produktautomaten:



Startzustand (z_{01}, z_{02}):

Das Paar aus den beiden Startzuständen der Automaten.

$E_{12} = \{(z_{e1}, z_{e2}) \mid z_{e1} \in E_1, z_{e2} \in E_2\}$

Endzustände sind alle Paare bestehend aus zwei Endzuständen.

Der so definierte Automat $A_1 \times A_2$ ist ein DEA. Ein Wort w wird von diesem Produktautomaten $A_1 \times A_2$ genau dann akzeptiert, wenn es sowohl von Automat A_1 als auch von A_2 akzeptiert wird. D.h. die von $A_1 \times A_2$ akzeptierte Sprache ist $L_1 \cap L_2$. Somit existiert ein DEA für die Sprache, also ist sie regulär.

Beispiel 9.29 - Abschlusseigenschaften regulärer Sprachen

Sei das Alphabet $\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}\}$ gegeben. Die Sprache

$L_1 = \{ w \in \Sigma^* \mid w \text{ enthält } \texttt{abab}, \text{ aber nicht } \texttt{abba}\}$

ist regulär. Dies lässt sich so mit Hilfe der Abschlusseigenschaften begründen:

- Die Sprache

  $L_{abab} = \{ w \in \Sigma^* \mid w \text{ enthält } \texttt{abab} \}$

  ist regulär, da die Sprache durch den regulären Ausdruck $(\texttt{a}|\texttt{b}|\texttt{c})^*\texttt{abab}(\texttt{a}|\texttt{b}|\texttt{c})^*$ darstellbar ist.

- Ebenso ist die Sprache

  $L_{abba} = \{ w \in \Sigma^* \mid w \text{ enthält } \texttt{abba} \}$

  regulär, da sie durch den regulären Ausdruck $(\texttt{a}|\texttt{b}|\texttt{c})^*\texttt{abba}(\texttt{a}|\texttt{b}|\texttt{c})^*$ darstellbar ist.

Es gilt $L_1 = L_{abab} \cap \overline{L}_{abba}$.

Da reguläre Sprachen unter Komplement- und Durchschnittsbildung abgeschlossen sind, ist somit auch L_1 regulär.

Aufgabe 9.30 - Abschlusseigenschaften regulärer Sprachen

Sei $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. Zeigen Sie mit Hilfe der Abschlusseigenschaften regulärer Sprachen, dass die Sprache

$L_2 = \{\ w \in \Sigma^* \mid w \text{ endet } \textit{nicht} \text{ mit } \mathtt{cba}\ \}$

regulär ist.

9.6 Das Pumping-Lemma für reguläre Sprachen

In den vorangehenden Abschnitten wurde betrachtet, wie nachgewiesen werden kann, dass eine Sprache regulär ist. Wie kann aber das Gegenteil bewiesen werden, d.h. dass eine Sprache *nicht regulär* ist, dass es also keinen endlichen Automaten, keinen regulären Ausdruck und keine Chomsky-Typ-3-Grammatik für eine Sprache geben kann?

Beispielsweise sind folgende Sprachen nicht regulär:

$L_1 = \{\ \mathrm{a}^n\mathrm{b}^n \mid n \geq 0\ \}$

$L_2 = \{\ w \in \{0,1\}^* \mid w \text{ ist Palindrom }\}$

Eine Methode, mit der die Nichtregularität bewiesen werden kann, wird nachfolgend vorgestellt. Es handelt sich dabei um Beweise durch Widerspruch:

(1) Man nimmt zuerst an, dass die Sprache regulär ist.

(2) Wenn die Sprache regulär ist, kann man bestimmte gültige Folgerungen ableiten.

(3) Es zeigt sich dann, dass ein gefolgerter Sachverhalt aber nicht zutrifft.

(4) Somit muss die Annahme falsch gewesen sein (da die Folgerungsschritte korrekt waren).

Dabei wird dann folgender Satz für Folgerungen verwendet, das sog. *Pumping-Lemma*.

Satz 9.31 - Pumping-Lemma für reguläre Sprachen

Wenn L eine reguläre Sprache ist, dann gibt es eine sprachabhängige Mindestlänge $N \geq 0$, so dass sich jedes Wort $x \in L$ mit Länge $|x| \geq N$ in drei Teilwörter u, v, w zerlegen lässt,

$x = uvw$

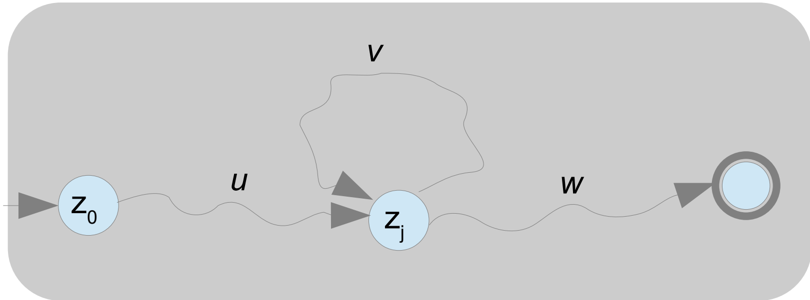
für die gilt:

1. $|uv| \leq N$
2. $|v| \geq 1$
3. für alle $k \geq 0$ ist auch uv^kw in der Sprache L enthalten.

Beweisskizze

Ist L eine reguläre Sprache, dann gibt es einen DEA mit endliche vielen Zuständen, der die Wörter der Sprache akzeptiert.

Betrachtet man nun "lange" Wörter der Sprache, dann ergibt sich bei der Verarbeitung der Wörter folgende Situation:



Jedes Wort *x*, dessen Länge mindestens so groß ist wie die Anzahl der Zustände, bewirkt, dass mindestens ein Zustand z_j bei der Verarbeitung zweimal erreicht wird, d.h. ein Zyklus durchlaufen wird. Dies liefert, wie oben dargestellt, die Aufteilung in drei Teilwörter *u*, *v*, *w*. Der mittlere Teil *v* muss dabei mindestens ein Zeichen lang sein, die Teile *u* und *w* können ggf. auch das leere Wort sein, wenn der Zyklus ganz beim Startzustand oder beim Endzustand liegt. *u* und *v* zusammen können aber nicht länger als die Zustandszahl sein.

Dieser Zyklus entsprechend *v* kann dann beliebig oft durchlaufen werden, d.h. jedes Wort uv^kw führt dann auch zu einem Endzustand, gehört somit auch zur Sprache des Automaten.

Warum "Pumping Lemma"?

- Ein Lemma ist in der Mathematik ein Hilfssatz, der nur dazu verwendet wird, eine andere, wichtigere Eigenschaft zu beweisen.
- Die Bezeichnung "Pumping" kommt daher, dass dabei Wörter gedanklich "aufgepumpt" werden: aus kürzeren Wörtern der Sprache werden längere Wörter der Sprache abgeleitet.

Vorgehensweise: Beweis der Nichtregularität mittels Pumping-Lemma

Mit Hilfe des Pumping-Lemmas kann in folgender Weise ein Beweis durch Widerspruch geführt werden:

(1) Annahme: Sprache *L* ist regulär.

(2) Wähle ein geeignetes Wort *x* aus *L,* abhängig von der (unbekannten) Mindestlänge *N*, die es nach Pumping-Lemma gibt.

(3) Zeige, dass *für jede beliebige Aufteilung* des Worts $x = uvw$ in drei Teilwörter mit $|uv| < N$ und $|v| \geq 1$ mindestens ein *k* existiert, so dass uv^kw nicht in der Sprache *L* enthalten ist.

Wenn die Annahme richtig wäre, müsste nach Pumping-Lemma für mindestens eine der Aufteilungen x=*uvw* für beliebiges *k* das Wort uv^kw in der Sprache *L* sein. Das ist aber nicht der Fall. Somit hat sich ein Widerspruch ergeben.

(4) Konsequenz: Die Annahme, dass *L* regulär ist, muss falsch gewesen sein!

Satz 9.32 - a^nb^n nicht regulär

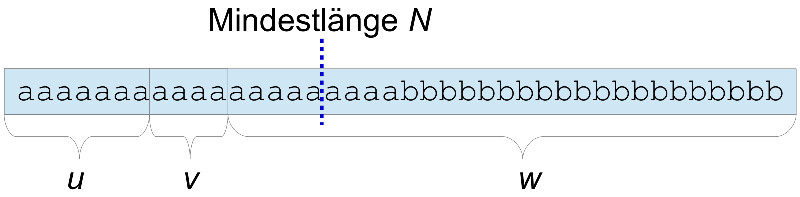
Die Sprache $L = \{ a^nb^n \mid n \geq 0\}$ ist nicht regulär.

Beweis durch Widerspruch mit Hilfe des Pumping-Lemmas:

(1) Annahme: Sprache *L* sei regulär. Dann würde das Pumping-Lemma für *L* gelten.

(2) Sei *N* die (unbekannte) sprachspezifische Mindestlänge gem. Pumping-Lemma Wir wählen ein Wort a^kb^k mit $k > N$. Dieses Wort ist in *L* enthalten.

(3) Für jede mögliche Aufteilung x = *uvw* mit $|uv| < N$ und $|v| \geq 1$ ergibt sich folgende Situation:



v hat Länge $j \geq 1$.

Die Teilwörter *u* und v können dann nur aus a-Zeichen bestehen. Der mittlere Teil *v* besteht somit aus mindestens einem Zeichen a.

Nach Pumping-Lemma muss es mindestens eine solche Aufteilung geben, so dass dann auch $uv^2w = a^{k+j}\,b^k$ ein Wort der Sprache ist. Das "aufgepumpte" Wort uv^2w enthält aber nun mehr a-Zeichen als b-Zeichen, ist somit garantiert nicht in *L* enthalten.

Da das "Aufpumpen" des Worts bei jeder möglichen Aufteilung dazu führt, dass das Wort nicht mehr in der Sprache *L* ist, gilt das Pumping-Lemma offensichtlich nicht, d.h. die Annahme, dass die Sprache regulär ist, muss falsch gewesen sein.

Aufgabe 9.33 - Anwendung des Pumping-Lemmas

Beweisen Sie mit Hilfe des Pumping-Lemmas, dass die Sprache

$L_2 = \{ w \in \{0,1\}^* \mid w \text{ ist Palindrom } \}$

nicht regulär ist.

Anmerkung

Das Pumping-Lemma und seine Anwendung zu verstehen, ist erfahrungsgemäß ein der besonderen Herausforderungen in der TINF-Vorlesung. Sie sollten zumindest grob die Bedeutung des Lemmas und die Vorgehensweise verstehen. In der Klausur wird aber kein Beweis mittels Pumping-Lemma verlangt.

Zusammenfassung zu Lektion 16

Diese Fragen sollten Sie nun beantworten können

- Was sind reguläre Sprachen?
- Wie kann gezeigt werden, ob eine Sprache regulär ist?
- Welche Operationen auf Sprachen erhalten die Eigenschaft, dass die Sprachen regulär sind?
- Was sagt das Pumping-Lemma aus?
- Wie kann das Pumping-Lemma verwendet werden, um zu beweisen, dass eine Sprache nicht regulär ist?