Das Pumping-Lemma

Wir wissen schon, dass endliche deterministische Automaten $\mathcal{A}$ (DEA), also Akzeptoren, und reguläre Ausdrücke die gleiche reguläre Sprache beschreiben. Beide Beschreibungen sind hinsichtlich ihrer formalen Sprache äquivalent. Zu jedem regulären Ausdruck r lässt sich ein Akzeptor finden, der diesen regulären Ausdruck akzeptiert. Streng genommen erzeugt der reguläre Ausdruck eine Wortmenge $\{r\}$ über diesen regulären Ausdruck. Die formale Sprache zu r ist $\mathcal{L}(\{r\})$. Man schreibt aber kürzer $\mathcal{L}(r)$.
Andererseits kann ein DEA in einen regulären Ausdruck überführt werden, der die gleiche reguläre Sprache beschreibt.

Somit lässt sich formulieren:

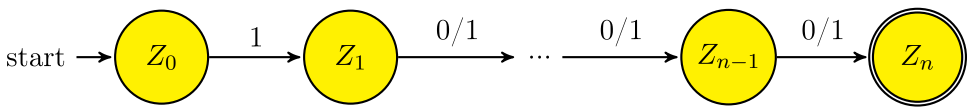
Satz: Äquivalenz von regulären Ausdrücken und DEA

Sei $\mathcal{A}$ ein DEA und r ein regulärer Ausdruck. Dann gilt: $\mathcal{L}(A) = \mathcal{L}(r)$.

Hinweis: Streng genommen gilt die Überführung eines regulären Ausdrucks in einen Nichtdeterministischen endlichen Automaten (NEA). Man kann aber zeigen, dass jeder NEA in einen äquivalenten DEA überführbar ist.

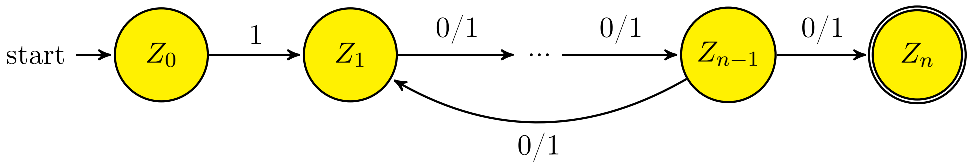
Worin liegt nun das Besondere dieses Satzes?
Eine formale Sprache $\mathcal{L}$ über einem Alphabet Σ enthält unendlich viele Worte, die über diesem Alphabet gebildet werden können. Ein endlicher Automat muss daher beim Überprüfen der Worte Zustände doppelt verwenden, es entstehen Zyklen.

Beispiel: Sei $\Sigma = \{0, 1\}$ und der DEA habe n Zustände.
Prüfe $w = \underbrace{100...1001}_{n-Zeichen}$



Wenn mit jedem neuen Zeichen nun ein neuer Zustand eingenommen wird, können nur Wörter w mit $|w| = n$ geprüft werden. Für Wörter w' mit $|w'| > n$ müssen also Zustände mehrfach durchlaufen werden (Schubfachprinzip).

Prüfe $w' = \underbrace{10011...001001}_{n+k-Zeichen}$



In regulären Ausdrücken wird dieses Prinzip durch die Verwendung des Kleene-Sterns * deutlich. Es können durch diese Schleifen beliebig lange Wörter konstruiert werden. Und das ist das Prinzip, welches das *Pumping-Lemma* nutzt. Bei Wörtern einer regulären Sprache mit einer bestimmten Länge (der Pumping-Zahl n) kann jedes andere Wort w' mit $|w'| > n$ zerlegt werden in drei Anteile $w' = xyz$. Ein Teil x vor der Schleife, die Schleife y selbst (y kann kein oder mehrfach durchlaufen werden), und einem Anteil z nach der Schleife. Finde ich ein Wort der Sprache, wo dies nicht gelingt, dann ist die Sprache nicht regulär.

Satz: Pumping-Lemma für reguläre Sprachen

Sei $L \subseteq \Sigma^*$ eine reguläre Sprache. Dann gibt es eine ganze Zahl $n \geq 1$ (Pumping-Zahl), so daß jedes Wort $|w| \geq n$ wie folgt zerlegt werden kann:
$w = xyz$ mit Wörtern $x, y, z \in \Sigma^*$, so daß

1. $|y| \geq 1$ (bzw. $y \neq \epsilon$)
2. $|xy| \leq n$
3. $xy^kz \in L$ für alle $k \in \mathbb{N}$.

Eine etwas formalere Notation dieses Satzes lautet:

Satz: Pumping-Lemma für reguläre Sprachen (formale Fassung)

Sei $L \subseteq \Sigma^*$ eine Sprache. Dann gilt:
L regulär $\Rightarrow \exists n \in \mathbb{N}^* \; \forall w \in L, |w| \geq n \; \exists x, y, z \in \Sigma^*, w = xyz$ mit
$|y| \geq 1 \wedge |xy| \leq n \wedge \forall k \in \mathbb{N} : xy^kz \in L.$

Somit können alle diejenigen Wörter nicht geprüft werden, die das Zählen von Zeichen notwendig macht. Eine Folge aus der endlichen Anzahl der Zustände eines DEAs und der fehlende Möglichkeit des Speicherns. Und das ist auch schon die Beweisidee:

Beweis:

Sei $w = a_1a_2 \ldots a_m$ mit $|w| = m > n$ ein Wort in L, wobei n die Anzahl der Zustände des DEAs ist (Pumping-Zahl).
Dann ist $Z_0Z_1 \ldots Z_m$ die Folge der Zustände im DEA, die bei Prüfung des Wortes w durchlaufen wird. Es ist $Z_m \in F$ ein Endzustand.

Da die Anzahl der Zustände n ist und die Menge der durchlaufenden Zustände m größer ist, müssen Zustände mehrfach besucht werden. D.h. wir durchlaufen eine Schleife im Graphen für das Teilwort y.

Es gibt Indizes $i, j, k \in \mathbb{N}$ mit $(0 < i < j < n)$ und eine Zerlegung für $w = xyz$ mit:

$$x = a_1, a_2, \ldots, a_i\ ,\ \ y = a_{i+1}, \ldots, a_j\ ,\ \ z = a_{j+1}, \ldots, a_n$$

1. Wegen $i < j$ gilt $|y| \geq 1$.
2. Wegen $j \leq n$ gilt $|xy| = |a_1, \ldots, a_j| = j \leq n$.
3. Wir können das Wort in xy^kz unterteilen, da der Zustand zu dem Zeichen a_{i+1} gehörig, mehrfach besucht wird und wir beliebig häufig diesen besuchen können.

Wegen $Z_m \in F$ ist jedes dieser Worte in der Sprache enthalten. q.e.d.

Das Pumping-Lemma ist ein notwendiges Kriterium für reguläre Sprachen. Somit kann ein Beweis der Regularität nicht erfolgen. Aber die Nichtregularität lässt sich mit diesem Satz zeigen. Um nun zu zeigen, dass Sprachen **nicht** regulär sind, kann der indirekte Beweis oder die Kontraposition des Satzes genutzt werden.

Satz: Kontraposition des Pumping-Lemmas für reguläre Sprachen

Sei $L \subseteq \Sigma^*$ eine Sprache. Dann gilt:
Wenn $\forall n \in \mathbb{N}\ \exists w \in L, |w| \geq n\ \forall x, y, z \in \Sigma^*, w = xyz$ gilt:

$$|y| = 0\ \vee\ |xy| > n\ \vee\ \exists k > 0 : xy^kz \notin L \Rightarrow \text{L nicht regulär.}$$

Nun kann man die drei Oder-verknüpften Bedingungen noch zu einer Implikation umwandeln: $a \Rightarrow b \Leftrightarrow \overline{a} \vee b$. Somit lautet dann der Satz:

Satz: Kontraposition des Pumping-Lemmas für reguläre Sprachen, 2. Fassung

Sei $L \subseteq \Sigma^*$ eine Sprache. Dann gilt:
Wenn $\forall n \in \mathbb{N}\ \exists w \in L, |w| \geq n\ \forall x, y, z \in \Sigma^*, w = xyz$ gilt:

$$(|y| \neq 0 \wedge |xy| < n) \Rightarrow (\exists k \geq 0 : xy^kz \notin L), \Rightarrow \text{L nicht regulär.}$$

Einige Beispiele sollen die Verwendung des Pumping-Lemmas zeigen.

Beispiel 1: Gegeben sei $L = \{a^k : k \in \mathbb{N}\}$. L ist regulär (Warum?)
Wir wählen für $w = xyz$ folgende Zerlegung

- $x = a\ ,\ \ y = a\ ,\ \ z = a \rightarrow$ Bedingung 0 (bedeutet die Existenz der drei Worte x,y,z) ist erfüllt,
- $|y| = |a| = 1 \neq 0 \rightarrow$ Bedingung 1 ist erfüllt,

- $|xy| \leq n$ (jedes Wort hat mindestens 3 Zeichen) $\rightarrow$ Bedingung 2 ist auch erfüllt (z. B. für $n = 4$)

Da wir das a beliebig wiederholen dürfen und das zusammengesetzte Wort immer noch in der Sprache ist, gilt also $xy^k z \in L$. (Bedingung 3 ist erfüllt)

Beispiel 2:

$$L = \{a^k b^l : k, l \in \mathbb{N}\}$$

Diese Sprache ist regulär. Wir wählen

$$x = a \;\; y = a \;\; z = b^l$$

und alle Regeln sind erfüllt (Prüfen!). Wie lautet die Pumping-Zahl?

Beispiel 3:

$$L = \{a^n b^n : n \in \mathbb{N}\}$$

Diese Sprache ist nicht regulär, da wir die n's nicht zählen können. Wie zeigt man das mithilfe eines indirekten Beweises?
Sei p die Zahl aus dem Pumping-Lemma. Wir wählen $w = a^p b^p$ und wir nehmen an, w ist regulär. Es gilt $|w| > p$. Für jede Zerlegung $w = xyz$ wissen wir $y \neq \epsilon$ und $|xy| \leq p$. x kann nur aus a's bestehen, sogar xy beteht nur aus a's. Dann muß z alle b's enthalten. Nach dem Pumping-Lemma ist $xz \in L$, was für $k = 0$ gilt. Da aber auch mindestens $y = a$ (wegen $y \neq \epsilon$), hat xz weniger als p a's. Widerspruch!
Hinweis: Für jedes andere k (außer $k = 1$) können auch Wörter $a^{p+l} b^p : l > 0$ gebildet werden. Dies ist ein Widerspruch zur Annahme, das L regulär ist.

Noch ein letztes Beispiel soll hier zeigen, wie man mit einem Widerspruchsbeweis die Nichtregularität zeigen kann.

Beispiel 4:

$$L = \{a^n b^m c^k \,|\, n, m, k \geq 1, m \geq n + k\}$$

Übung 1: Sei L die Sprache aller Wörter über dem Alphabet $A = \{a, b\}$, die aus gleich vielen $a's$ und $b's$ bestehen, d.h. $L = \{w \in \{a, b\}^* | \; |w|_a = |w|_b\}$

Übung 2: Sei L die Sprache aller Wörter über dem Alphabet $A = \{1\}$ und $L = \{w \in A^* | \quad x = 1^{i^2}$ für $i \geq 0\}$