Das Pumping-Lemma – Formulierung

Sei L reguläre Sprache. Dann gibt es ein $n \in \mathbb{N}$ mit:

jedes Wort $w \in L$ mit $|w| \geq n$ kann zerlegt werden

in $w = xyz$, so dass gilt:

1. $|xy| \leq n$
2. $|y| \geq 1$
3. für alle $k \geq 0$ ist $xy^k z \in L$.

Anschaulich gesprochen heisst das:

Ist w ein hinreichend langes Wort in L, dann gibt es ein Teilwort y in w, das nicht zu lang und nicht zu weit vom Anfang entfernt ist, und das beliebig oft wiederholt werden kann.

Dies widerspricht nicht der Tatsache, dass endliche Sprachen regulär sind. (Das ist so, weil $L = \{w_1, \ldots, w_k\}$ einfach durch den regulären Ausdruck $w_1 + \ldots + w_k$ beschrieben wird.) Denn in einer endlichen Sprache L gibt es ein längstes Wort $w_{\max}$. Wählt man in der Aussage des Pumping-Lemmas $n > |w_{\max}|$, so ist die Aussage (leererweise) erfüllt: es gibt gar kein Wort $w \in L$ mit $|w| \geq n$.

Das Pumping-Lemma – Beweis

Wähle DFA $A = (Q, \Sigma, \delta, q_0, F)$ mit $L(A) = L$, und setze $n := |Q|$.

Sei $w = a_1 \dots a_m \in L$ mit $m \geq n$. Definiere für $i \leq m$:

- $w_i := a_1 \dots a_i \qquad (w_0 = \epsilon)$
- $q_i := \hat{\delta}(q_0, w_i)$.

Da $|Q| = n$, gibt es $i < j \leq n$ mit $q_i = q_j$, also $\hat{\delta}(q_0, w_i) = \hat{\delta}(q_0, w_j)$. Definiere

- $x = w_i = a_1 \dots a_i$
- $y = a_{i+1} \dots a_j$
- $z = a_{j+1} \dots a_m$

Dann ist $\hat{\delta}(q_0, x) = q_i = \hat{\delta}(q_0, xy) = \hat{\delta}(q_0, xy^k)$ für alle $k \geq 0$.

Da $w \in L(A)$, ist $\hat{\delta}(q_i, z) = q_m \in F$, also auch $\hat{\delta}(q_0, xy^k z) = q_m \in F$ für alle $k \geq 1$. □

Die Zustände $q_0, \dots, q_n$ sind $n+1$ viele, es gibt aber nur $|Q| = n$ Zustände in A, also müssen nach dem *Schubfachprinzip* mindestens zwei davon gleich sein. Der Automat durchläuft also beim Lesen von $w_n = a_1 \dots a_n$ eine Schleife. Dann kann aber diese Schleife auch beliebig oft wiederholt werden, und der Zustand am Ende bleibt gleich.

Das Pumping-Lemma – Anwendung

Um zu zeigen, dass L nicht regulär ist, zeige:

- Für alle $n \in \mathbb{N}$
- gibt es $w \in L$ mit $|w| \geq n$ und
- für jede Zerlegung $w = xyz$ mit $|xy| \leq n$ und $|y| \geq 1$
- gibt es ein $k \geq 0$ so dass $xy^k z \notin L$.

Das Pumping-Lemma liefert eine notwendige Bedingung dafür, dass eine Sprache regulär ist: Ist L regulär, dann hat L die Pumpeigenschaft, also die in der Konklusion des Lemmas beschriebene Eigenschaft.

Es kann also verwendet werden, um zu zeigen, dass eine Sprache *nicht* regulär ist, indem man zeigt, dass sie die Pumpeigenschaft verletzt.

Auf der Folie oben steht, was dazu gezeigt werden muss, nämlich die Negation der Pumpeigenschaft. Auf den folgenden zwei Folien sind zwei Beispiele, wobei das zweite hauptsächlich zeigen soll, dass man in manchen Fällen tatsächlich im letzten Schritt ein $k > 1$ wählen muss.

Erste Anwendung

Theorem

$L_{eq} = \{ w \in \{0,1\}^* ; |w|_0 = |w|_1 \}$ *ist nicht regulär.*

Beweis: Sei $n \in \mathbb{N}$ gegeben.

Wähle $w = 0^n 1^n \in L_{eq}$ mit $|w| = 2n \geq n$.

Sei eine Zerlegung $w = xyz$ mit $|xy| \leq n$ und $|y| = j \geq 1$ gegeben.

Da $|xy| \leq n$ ist, ist $x = 0^i$ und $y = 0^j$ und $z = 0^{(n-i-j)}1^n$.

Wähle $k = 0$, dann ist $xy^0z = xz = 0^{(n-j)}1^n \notin L_{eq}$,
da $j > 0$ und somit $|xz|_0 = n - j < n = |xz|_1$. □

Zweite Anwendung

Theorem

$L_{prim} = \{ w \in \{1\}^* ; |w|$ *ist prim* $\}$ *ist nicht regulär.*

Beweis: Sei $n \in \mathbb{N}$ gegeben.

Wähle Primzahl $p \geq n + 2$ und $w = 1^p$.

Sei eine Zerlegung $w = xyz$ mit $|xy| \leq n$ und $|y| = j \geq 1$ gegeben.

Es ist $|xz| = p - j$. Betrachte das Wort $w' = xy^k z$ für $k = p - j$.

$|w'| = |xz| + (p-j)|y| = (p-j) + (p-j)j = (p-j)(j+1)$.

Außerdem ist $j \geq 1$, also $(j+1) \geq 2$, sowie $j \leq n \leq p-2$,
also auch $(p-j) \geq 2$.

Daher ist $w' \notin L_{\text{prim}}$.

Abschlusseigenschaften regulärer Sprachen

Sind L, L_1 und L_2 reguläre Sprache, dann auch:

Boolesche Operationen:

$L_1 \cup L_2$	(Vereinigung)
$\Sigma^* \setminus L$	(Komplement)
$L_1 \cap L_2$	(Durchschnitt)

Reguläre Operationen:

$L_1 \cdot L_2$	(Konkatenation)
L^*	(Kleenesche Hülle)

Reversion:

$L^R := \{ w^R ; w \in L \}$
wobei $w^R := a_n \dots a_1$ ist, wenn $w = a_1 \dots a_n$

Abschluss unter Vereinigung folgt wegen des $+$ in regulären Ausdrücken.

Zum Abschluss unter Negation: Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein DFA, der L akzeptiert. Dann akzeptiert $\bar{A} = (Q, \Sigma, \delta, q_0, \bar{F})$ mit $\bar{F} = Q \setminus F$ die Sprache $\Sigma^* \setminus L$, denn es gilt: $w \notin L \leftrightarrow \hat{\delta}(q_0, w) \notin F \leftrightarrow \hat{\delta}(q_0, w) \in \bar{F} \leftrightarrow w \in L(\bar{A})$

Der Abschluss unter Durchschnitt folgt daraus schon mittels der DeMorgan-Regel $A \cap B = \overline{\bar{A} \cup \bar{B}}$. Er kann aber auch direkt durch die Konstruktion eines *Produktautomaten* gezeigt werden, der Automaten für L_1 und L_2 parallel ablaufen lässt:

Sei also $A_i = (Q_i, \Sigma, \delta_i, q_{0,i}, F_i)$ für $i = 1, 2$ ein DFA mit $L(A_i) = L_i$. Konstruiere daraus den Produktautomaten $A = (Q, \Sigma, \delta, q_0, F)$ mit:

Zuständen $Q = Q_1 \times Q_2$, also Paare von Zuständen (p, q) mit $p \in Q_1$ und $q \in Q_2$.

Anfangszustand ist $q_0 = (q_{0,1}, q_{0,2})$, das Paar aus den beiden Anfangszuständen, d.h. beide Automaten starten im Anfangszustand.

Übergangsfunktion $\delta\big((p, q), a\big) = \big(\delta_1(p, a), \delta_2(q, a)\big)$, d.h. beide Automaten führen ihren jeweiligen Übergang aus.

Endzustände sind $F = F_1 \times F_2$, d.h. der Produktautomat ist im Endzustand, wenn beide Automaten im Endzustand sind.

Es ist nun leicht durch Induktion über $|w|$ zu zeigen, dass $\hat{\delta}(q_0, w) = \big(\hat{\delta}_1(q_{0,1}, w), \hat{\delta}_2(q_{0,2}, w)\big)$ ist, also akzeptiert A gdw. beide A_1 und A_2 akzeptieren.

Klammerausdrücke

Sei $L_{kl} := \{ w \in \{<, >\}^* ; w \text{ korrekt geklammerter Ausdruck} \}$

$$<<><<>>><> \in L_{kl} \qquad <<>>><<> \notin L_{kl}$$

Theorem

L_{kl} ist nicht regulär.

Beweis: Angenommmen, L_{kl} sei regulär.

Dann wäre auch die folgende Sprache regulär:

$$L_{kl} \cap <^*>^* = \{ <^n>^n ; n \in \mathbb{N} \}$$

Beweis, dass L_{eq} nicht regulär ist, zeigt, dass dies nicht der Fall ist — Widerspruch !

Es folgt wegen des Abschlusses unter Durschnitt, dass L_{kl} nicht regulär ist. Der Beweis, dass L_{eq} nicht regulär ist, zeigt eigentlich, dass die Sprache $\{ 0^n 1^n ; n \in \mathbb{N} \}$ nicht regulär ist. Dies gilt, unabhängig davon ob die Symbole 0 und 1 oder $<$ und $>$ heißen.

Dieses Beispiel ist wichtig, weil korrekt geklammerte Ausdrücke Bestandteil vieler für die Informatik wichtiger Sprachen sind, z.B. Programmiersprachen, Abfragesprachen, Markup-Sprachen.

Diese Sprachen sind also allesamt nicht regulär, und man braucht raffiniertere Techniken als endliche Automaten, um sie auf syntaktische Korrektheit zu überprüfen.