Automaten und Formale Sprachen
6. Vorlesung

Prof. Dr. Dietrich Kuske

FG Automaten und Logik, TU Ilmenau

Wintersemester 2023/24

Nicht-reguläre Sprachen

Ist vielleicht jede Sprache regulär?

Wir werden zeigen: Es gibt eine Sprache, die von keiner Grammatik erzeugt wird, also insbes. von keiner rechtslinearen. Damit gibt es eine Sprache, die nicht regulär ist.

Hierzu zeigen wir für jedes Alphabet Σ:

1. Es gibt nur abzählbar unendlich viele Sprachen über Σ, die Sprache einer Grammatik sind.
2. Es gibt überabzählbar viele Sprachen über Σ.

Damit werden wir erhalten: Es gibt eine nicht erzeugbare Sprache, ohne daß wir eine konkrete angeben.

Anzahl der Grammatiken

Wir fixieren ein Alphabet Σ und Symbole $\#, \S, \%, \&, \$ \notin \Sigma$. Setze $\Gamma = \Sigma \cup \{\#, \S, \%, \&, \$\}$.

Sei $G = (V, \Sigma, P, S)$ eine Grammatik mit

- $V = \{A_1, A_2, \ldots, A_n\}$ und
- $P = \{(\ell_1, r_1), (\ell_2, r_2), \ldots, (\ell_m, r_m)\}$.

Dann definiere:

- für $A_i \in V$ setze $\mathrm{code}(A_i) = \#^i$
- für $a \in \Sigma$ setze $\mathrm{code}(a) = a$
- für $w = a_1 a_2 \ldots a_k \in (V \cup \Sigma)^*$ setze $\mathrm{code}(w) = \S\,\mathrm{code}(a_1)\,\S\,\mathrm{code}(a_2)\,\ldots\,\S\,\mathrm{code}(a_k)$
- für $\ell, r \in (V \cup \Sigma)^*$ setze $\mathrm{code}(\ell, r) = \mathrm{code}(\ell)\,\%\,\mathrm{code}(r)$
- $\mathrm{code}(G) = \left(\prod_{1 \le i \le n} \&\,\mathrm{code}(A_i)\right) \$ \left(\prod_{1 \le i \le m} \&\mathrm{code}(\ell_i, r_i)\right) \$\,\mathrm{code}(S)$

Es gelten:

1. $\mathrm{code}(G) \in \Gamma^*$. In der Übung wird gezeigt, daß Γ^* abzählbar unendlich ist.
   $\Longrightarrow$ es gibt nur abzählbar unendlich viele Codes von Grammatiken über Σ
2. für alle Grammatiken G und G' mit $\mathrm{code}(G) = \mathrm{code}(G')$ gilt $G = G'$
   $\Longrightarrow$ es gibt nur abzählbar unendlich viele Grammatiken über Σ

Damit erhalten wir:

Lemma

Für jedes Alphabet Σ ist die Menge $\{L(G) \mid G \text{ Grammatik über } \Sigma\}$ abzählbar unendlich.

Anzahl der Sprachen

Satz

Für jedes Alphabet Σ ist die Menge $\mathcal{P}(\Sigma^*) = \{L \mid L \text{ Sprache über } \Sigma\}$ überabzählbar, d.h. es gibt keine bijektive Funktion $F\colon \mathbb{N} \to \mathcal{P}(\Sigma^*)$.

Beweis: Indirekt. **(„Diagonalisierung“)**
angenommen, $\mathcal{P}(\Sigma^*)$ wäre abzählbar unendlich, d.h. es gäbe eine bijektive Funktion $F\colon \mathbb{N} \to \mathcal{P}(\Sigma^*)$.
Da Σ^* abzählbar unendlich ist (wird in Übung gezeigt), gibt es eine bijektive Funktion $f\colon \mathbb{N} \to \Sigma^*$.
Also ist $g = F \circ f^{-1}\colon \Sigma^* \to \mathcal{P}(\Sigma^*)\colon w \mapsto F\big(f^{-1}(w)\big)$ eine Bijektion.

Wir definieren eine neue Sprache:

$$L^\times = \{w \in \Sigma^* \mid w \notin g(w)\} \in \mathcal{P}(\Sigma^*)$$

Da g surjektiv ist, existiert $v \in \Sigma^*$ mit $L^\times = g(v)$. Dann gilt

$$v \in L^\times \iff v \notin g(v) = L^\times,$$

ein Widerspruch. $\square$

Veranschaulichung: wir stellen uns dies als unendliche Tabelle vor, deren Zeilen und Spalten den Wörtern aus Σ^* entsprechen. In der Zelle $(v, w) \in \Sigma^* \times \Sigma^*$ steht, ob $v \in g(w)$ gilt oder nicht (die Spalte w stellt also die Sprache $g(w)$ dar):

Zum Beispiel:

	ε	a	b	aa	ab	$\dots$
ε	nein	ja	ja	nein	nein	
a	ja	ja	ja	nein	ja	
b	ja	nein	nein	ja	ja	
aa	nein	ja	nein	nein	nein	
ab	ja	nein	nein	nein	ja	
$\vdots$						

Veranschaulichung: wir stellen uns dies als unendliche Tabelle vor, deren Zeilen und Spalten den Wörtern aus Σ^* entsprechen. In der Zelle $(v, w) \in \Sigma^* \times \Sigma^*$ steht, ob $v \in g(w)$ gilt oder nicht (die Spalte w stellt also die Sprache $g(w)$ dar):

Alle Einträge auf der Diagonalen negieren. Dadurch erhält man die Sprache $L^\times$.

	ε	a	b	aa	ab	$\ldots$
ε	nein ja	ja	ja	nein	nein	
a	ja	ja nein	ja	nein	ja	
b	ja	nein	nein ja	ja	ja	
aa	nein	ja	nein	nein ja	nein	
ab	ja	nein	nein	nein	ja nein	
$\vdots$						

Veranschaulichung: wir stellen uns dies als unendliche Tabelle vor, deren Zeilen und Spalten den Wörtern aus Σ^* entsprechen. In der Zelle $(v, w) \in \Sigma^* \times \Sigma^*$ steht, ob $v \in g(w)$ gilt oder nicht (die Spalte w stellt also die Sprache $g(w)$ dar):

Die Sprache $L^\times$ kann aufgrund der Konstruktion mit keiner der Sprachen $g(w)$ (d.h. Spalten) übereinstimmen.

	ε	a	b	aa	ab	$\dots$
ε	nein ja	ja	ja	nein	nein	
a	ja	ja nein	ja	nein	ja	
b	ja	nein	nein ja	ja	ja	
aa	nein	ja	nein	nein ja	nein	
ab	ja	nein	nein	nein	ja nein	
$\vdots$						

Veranschaulichung: wir stellen uns dies als unendliche Tabelle vor, deren Zeilen und Spalten den Wörtern aus Σ^* entsprechen. In der Zelle $(v, w) \in \Sigma^* \times \Sigma^*$ steht, ob $v \in g(w)$ gilt oder nicht (die Spalte w stellt also die Sprache $g(w)$ dar):

Die Sprache $L^\times$ kann aufgrund der Konstruktion mit keiner der Sprachen $g(w)$ (d.h. Spalten) übereinstimmen.

	ε	a	b	aa	ab	$\dots$
ε	nein ja	ja	ja	nein	nein	
a	ja	ja nein	ja	nein	ja	
b	ja	nein	nein ja	ja	ja	
aa	nein	ja	nein	nein ja	nein	
ab	ja	nein	nein	nein	ja nein	
$\vdots$						

Diese „selbstbezüglichen“ Beweisen nennt man aufgrund ihrer Veranschaulichung oft Diagonalisierungsbeweise.

Korollar

Für jedes Alphabet Σ existiert eine Sprache L über Σ, die von keiner Grammatik G erzeugt wird.

Beweis:

Lemma auf Folie 6.4: $\{L(G) \mid G \text{ Grammatik über } \Sigma\}$ ist abzählbar

Satz auf Folie 6.5: es gibt überabzählbar viele Sprachen über Σ

Konsequenz: es gibt eine (sogar überabzählbar viele) Sprachen über Σ, die nicht von einer Grammatik erzeugt werden. $\square$

Insbesondere gibt es (sehr viele) Sprachen, die nicht regulär sind. Unser Beweis erlaubt es uns aber nicht, ein einziges Beispiel zu finden. Offen bleibt auch, ob z.B. wenigstens jede kontextfreie Sprache regulär ist.

Konkrete nicht-reguläre Sprachen

Um zu zeigen, daß eine konkrete Sprache L regulär ist, kann man

- einen NFA M angeben mit $L(M) = L$, oder
- eine rechtslineare Grammatik G angeben mit $L(G) = L$, oder
- einen regulären Ausdruck γ angeben mit $L(\gamma) = L$, oder
- zeigen, daß $L = L_1 \cap L_2$ ist und daß L_1 und L_2 regulär sind, oder
- $\ldots$

Aber wie kann man zeigen, daß eine konkrete Sprache L **nicht** regulär ist?

Während Beweise der Existenz vom Grundsatz her einfach sind (gib einfach ein ... an), sind Beweise der Nicht-Existenz schwierig. Sie werden oft über „notwendige Bedingungen“ geführt. Im folgenden lernen wir zwei solche „notwendigen Bedingungen für die Regularität einer Sprache L“ kennen.

Das Pumping Lemma

Pumping Lemma (M. Rabin, D. Scott 1964)

Wenn L eine **reguläre** Sprache ist, **dann** gilt die folgende Aussage:

es gibt $n \geq 1$ derart,

daß für alle $x \in L$ mit $|x| \geq n$ gilt:

es gibt Wörter $u, v, w \in \Sigma^*$ mit

(i) $x = uvw$,

(ii) $|uv| \leq n$,

(iii) $|v| \geq 1$ und

(iv) $uv^iw \in L$ für alle $i \geq 0$.

Dieses Lemma sagt:

„Nur wenn es $n \geq 1$ gibt mit ..., ist es möglich, daß L regulär ist“.

Existiert kein $n \geq 1$ mit ..., so kann L also nicht regulär sein.

Beweis des Pumping-Lemmas:

Sei L eine reguläre Sprache.

Dann existiert ein NFA $M = (Z, \Sigma, S, \delta, E)$ mit $L = L(M)$, sei $n = |Z|$.

Sei nun x ein beliebiges Wort mit $x \in L$ und $|x| \geq n$, d.h. $x = a_1 a_2 \cdots a_m$ mit $m \geq n$ und $a_1, a_2, \ldots, a_m \in \Sigma$.

Da $x \in L(M)$, existieren Zustände $z_0, z_1, \ldots, z_m \in Z$ mit

$$z_0 \in S, \quad z_j \in \delta(z_{j-1}, a_j) \text{ für } 1 \leq j \leq m, \quad z_m \in E\,.$$

Wegen $|Z| = n \leq m$ existieren nach dem Schubfachprinzip $0 \leq j < k \leq n$ mit $z_j = z_k$.

Setze $u = a_1 \cdots a_j$, $v = a_{j+1} \cdots a_k$ und $w = a_{k+1} \cdots a_m$.

Dann gilt:

(i) $x = a_1 \cdots a_j\, a_{j+1} \cdots a_k\, a_{k+1} \cdots a_m = uvw$

(ii) $|uv| = |a_1 \cdots a_k| = k \leq n$

(iii) $|v| = k - (j+1) + 1 = k - j > 0$ (da $j < k$)

(iv) Sei $i \geq 0$ beliebig. Es gelten

$$z_j \in \widehat{\delta}(z_0, u),\ z_j = z_k \in \widehat{\delta}(z_j, v) \text{ und } z_m \in \widehat{\delta}(z_k, w) \cap E\,.$$

Also gilt $z_j \in \widehat{\delta}(z_j, v^i)$ für alle $i \in \mathbb{N}$.

Damit erhält man aber $z_m \in \widehat{\delta}(z_0, uv^i w) \cap E$, d.h., $uv^i w \in L(M) = L$.

□

Beispiel

$L_1 = \{0^m 1^m \mid m \in \mathbb{N}\}$ ist nicht regulär.

Beweis: indirekt
Angenommen, L_1 wäre regulär.

Nach dem Pumping-Lemma gibt es ein $n \geq 1$, so daß die folgende Behauptung gilt:

> Für jedes $x \in L_1$, $|x| \geq n$, gibt es $u, v, w \in \Sigma^*$ mit (i), (ii), (iii) und (iv). (*)

Wir wählen nun $x = 0^n 1^n$.
Dann ist $x \in L_1$ und $|x| = 2n > n$.
Nach der Behauptung (*) gibt es also $u, v, w \in \Sigma^*$ mit

(i) $x = uvw$,
(ii) $|uv| \leq n$,
(iii) $|v| \geq 1$ und
(iv) $uv^i w \in L_1$ für alle $i \geq 0$.

$$uvw \overset{(i)}{=} x = \underbrace{00000000\cdots0000}_{n-\text{mal}}\underbrace{11111111\cdots1111}_{n-\text{mal}}$$

Wegen $|uv| \leq n$ nach (ii) besteht uv nur aus Nullen.

Insbesondere besteht $v \in \{0\}^*$ nur aus Nullen.

Also gilt $uv^0w = uw = 0^{n-|v|}1^n$.

Aus $|v| \geq 1$ nach (iii) folgt $n - |v| < n$ und damit $uv^0w = 0^{n-|v|}1^n \notin L_1$.

Widerspruch zu (iv)! Also ist L_1 nicht regulär. $\square$

Unser Beweis, daß L_1 nicht regulär ist, folgt dem folgenden Schema:

Behauptung: Die Sprache L ist nicht regulär.

[0] (wörtlich) **Beweis:** indirekt. Angenommen, L wäre regulär. Nach dem Pumping-Lemma gibt es ein $n \geq 1$, so daß die folgende Behauptung (*) gilt:

Für jedes $x \in L$, $|x| \geq n$, gibt es $u, v, w \in \Sigma^*$ mit (i)-(iv).

[1] (problemspezifisch) Wir wählen ein **geeignetes** $x \in L$ mit $|x| \geq n$, so daß Schritt [3] ausführbar ist.

[2] (wörtlich) Nach Behauptung (*) gibt es $u, v, w \in \Sigma^*$ mit (i)–(iv).

[3] (problemspezifisch) Wir wählen zu u, v, w ein **passendes** $i \geq 0$ und zeigen, daß $uv^i w$ nicht in L sein kann.

[4] (wörtlich) Widerspruch zu (iv)! Also ist L nicht regulär. $\square$

Mithilfe logischer Operatoren kann das Pumping-Lemma auch wie folgt geschrieben werden:

L regulär
$$\rightarrow \quad \exists n \;\; \forall x \in L \text{ mit } |x| \geq n \;\; \exists u, v, w \text{ mit (i–iii)} \;\; \forall i : uv^i w \in L$$

Diese Aussage ist logisch äquivalent zu:

$$\forall n \quad \exists x \in L \text{ mit } |x| \geq n \quad \forall u, v, w \text{ mit (i-iii)} \quad \exists i : uv^i w \notin L$$
$$\rightarrow L \text{ ist nicht regulär}$$

Um zu zeigen, daß eine Sprache L nicht regulär ist, reicht es also zu zeigen, daß es für alle n ein $x \in L$ gibt ...

Dies können wir auch in dem folgenden **Spielschema** fassen:

Wir (die Beweiser oder Braven) wollen zeigen, daß die Sprache L nicht regulär ist. Dazu müssen wir das folgende Spiel (gegen den Gegner oder den Gemeinen) gewinnen:

Runde 1 G wählt eine Zahl $n \geq 1$.

Runde 2 B wählt ein $x \in L$ mit $|x| \geq n$

Runde 3 G wählt u, v, w mit
(i) $x = uvw$, (ii) $|uv| \leq n$ und (iii) $|v| \geq 1$.

Runde 4 B wählt ein i und zeigt, daß $uv^iw \notin L$.

Die Sprache L ist **nicht** regulär, falls B unabhängig von den Wahlen von G in Runden 1 und 3 immer so wählen kann (in Runden 2 und 4), daß schließlich $uv^iw \notin L$ gilt.

Beispiel

$L_2 = \{0^m 1 0^\ell 1 0^{m+\ell} \mid m, \ell \in \mathbb{N}\}$ ist nicht regulär.

Beweis: wir zeigen, daß B im Spielschema immer so wählen kann, daß $uv^iw \notin L_2$ gilt:

Runde 1 G wählt eine Zahl $n \geq 1$.

Runde 2 B wählt $x = 0^n 1 1 0^n$ (natürlich gelten $x \in L_2$ und $|x| \geq n$)

Runde 3 G wählt u, v, w mit
(i) $x = uvw$, (ii) $|uv| \leq n$ und (iii) $|v| \geq 1$.

Runde 4 B wählt $i = 0$ und zeigt, daß $uv^iw \notin L_2$:
x beginnt mit genau $n \geq |uv|$ Nullen. Also besteht v nur aus Nullen. Damit ergibt sich $uv^iw = 0^{n-|v|} 1 1 0^n \notin L_2$, denn $|v| \geq 1$ impliziert $n - |v| \neq n$.

Also kann B so wählen, daß $uv^iw \notin L_2$, d.h., L_2 ist tatsächlich nicht regulär. □

Beispiel

$L_3 = \{0^n\colon n$ ist quadratfrei, d.h. $m^2 \mid n \;\Rightarrow\; m = 1\}$ ist nicht regulär.

Beweis: wir zeigen, daß B im Spielschema immer so wählen kann, daß $uv^iw \notin L_3$ gilt:

Runde 1 G wählt eine Zahl $n \geq 1$.

Runde 2 B wählt $x = 0^p$ für eine Primzahl $p > n$ (natürlich gelten $x \in L_3$ und $|x| \geq n$)

Runde 3 G wählt u, v, w mit
(i) $x = uvw$, (ii) $|uv| \leq n$ und (iii) $|v| \geq 1$.

Runde 4 B wählt $i = |uw| \cdot \big(|v| + 2\big)$ und zeigt, daß $uv^iw \notin L_3$:
Mit dieser Wahl von i erhalten wir

$$\begin{aligned} |uv^iw| &= |uw| + i \cdot |v| = |uw| + |uw| \cdot \big(|v| + 2\big) \cdot |v| \\ &= |uw|\big(|v|^2 + 2|v| + 1\big) = |uw|\big(|v| + 1\big)^2 . \end{aligned}$$

Wegen (iii) gilt $|v| + 1 > 1$, also ist $|uv^iw|$ nicht quadratfrei.

Also kann B so wählen, daß $uv^iw \notin L_3$, d.h., L_3 ist nicht regulär. □

Vorsicht!

Das Pumping-Lemma hat die Form „Wenn L regulär ist, so gilt ...“, es handelt sich also nur eine notwendige Bedingung.
Wir zeigen, daß diese Bedingung nicht hinreichend ist:

Beispiel

Sei $L_4 = \{a^k b^\ell c^\ell \mid k \geq 1, \ell \in \mathbb{N}\} \cup \{b, c\}^*$. Dann ist L_4 nicht regulär, es gibt aber $n \geq 1$, so daß für jedes $x \in L_4$ mit $|x| \geq n$ Wörter u, v, w mit (i)-(iv) existieren.

Beweis: Angenommen, L_4 wäre regulär. Nach dem Satz auf Folie 4.9 wäre dann auch $L_4 \cap L(ab^*c^*) = \{ab^\ell c^\ell \mid \ell \in \mathbb{N}\}$ regulär. Auf Folie 7.7 werden wir sehen, daß dies nicht der Fall ist (alternativ kann der Beweis zum Beispiel auf Folie 6.13 variiert werden). Also ist L_4 nicht regulär.

Sei $n = 1$.

Sei nun $x \in L_4$ beliebig mit $|x| \geq n = 1$.

Wir faktorisieren x, indem wir setzen: $u = \varepsilon$, $v =$ erster Buchstabe von x, $w =$ Rest von x.

Sei $i \in \mathbb{N}$ beliebig. Wir zeigen $uv^iw \in L_4$ durch Fallunterscheidung:

1. Fall: $v = a$. Dann existieren $k, \ell \in \mathbb{N}$ mit $x = aa^kb^\ell c^\ell$. Es folgt

$$uv^iw = a^ia^kb^\ell c^\ell \in L_4$$

2. Fall $v \neq a$. Dann gilt $uv^iw \in \{b, c\}^* \subseteq L_4$. $\square$

Zusammenfassung Pumping-Lemma

Das Pumping-Lemma kann genutzt werden, um zu zeigen, daß eine Sprache **nicht** regulär ist (z.B. mit Hilfe des Spielschemas). Es gibt aber nichtreguläre Sprachen, deren Nicht-Regularität nicht durch das Pumping-Lemma bewiesen werden kann.

Insbesondere kann das Pumping-Lemma nicht genutzt werden, um die Regularität einer Sprache L zu zeigen.

Zusammenfassung 6. Vorlesung

in dieser Vorlesung neu

- über jedem Alphabet gibt es eine nicht-reguläre Sprache
- das Pumping-Lemma erlaubt es manchmal, die Nicht-Regularität zu zeigen

kommende Vorlesung

- eine weitere notwendige Bedingung für die Regularität eine Sprache
- „optimale“ DFAs