Formale Sprachen

Dozent: Prof. Dr. Michael Eichberg

Kontakt: michael.eichberg@dhbw.de, Raum 149B

Version: 1.0

Quelle: Die Folien sind teilweise inspiriert von oder basierend auf Lehrmaterial von Prof. Dr. Ritterbusch und Theoretische Informatik - kurzgefasst von Prof. Dr. Uwe Schöning.

Folien: https://delors.github.io/theo-algo-formale_sprachen/folien.de.rst.html

https://delors.github.io/theo-algo-formale_sprachen/folien.de.rst.html.pdf

Fehler melden: https://github.com/Delors/delors.github.io/issues

Einführung

Alphabete und Sprachen

Formale Sprachen sind ein zentraler Aspekt der theoretischen Informatik.

- Nutzungsinterface zwischen Computer und Mensch
- Grundlage für Programmiersprachen

Es gibt unterschiedliche Klassen und Modelle formaler Sprachen:

- Erkennbarkeit und Ausdruckskraft
- Anforderungen an Computermodelle zur Erkennbarkeit
- Komplexität von Verfahren zur Erkennung

Alphabete

Definition

Ein Alphabet $\Sigma = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ ist eine endliche Menge von Zeichen / Symbolen.

Beispiel

Abzählbare Mengen

- $\Sigma_{lat} = \{a, b, c, \ldots, z\}$
- $\Sigma_{ziffer} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $\Sigma_{unicode} = \{x | x \text{ ist ein Unicode-Zeichen}\}$
- $\Sigma_{logik} = \{0, 1, (,), \wedge, \vee, \neg, (,)\} \cup \Sigma_{lat}$

Kartesisches Produkt

Definition

Ein kartesisches Produkt wie $A \times B$ oder A^n für $n \in \mathbb{N}$ von Mengen oder Alphabeten bezeichnet die Menge der Tupel (a, b) oder $(a_1, \ldots, a_n)$ von Elementen der Mengen:

$$\begin{aligned} A \times B &:= && \{(a, b) | a \in A, b \in B\} \\ A^n &:= \underbrace{A \times \ldots \times A}_{n \text{ Faktoren}} &=& \{(a1, \ldots, an) | a1, \ldots, an \in A\} \end{aligned}$$

Beispiel

- $\Sigma_{lat} \times \Sigma_{lat} = \{(a, a), (a, b), \ldots, (z, z)\}$
- $\Sigma_{lat}^3 = \{(a, a, a), (a, a, b), \ldots, (z, z, z)\}$

Kleene-Abschluss

Definition

Ein Wort ω ist ein endliches — ggf. leeres — Tupel $(w_1, w_2, \ldots, w_n) \in \Sigma^n$ von Zeichen $w_k \in \Sigma$ eines Alphabets mit Länge $|\omega| = n$ der Anzahl der Zeichen.

- Wörter werden meist ohne Klammern geschrieben; d. h. $\omega = w_1 w_2 \ldots w_n$.
- Das leere Wort (das Wort ohne Zeichen) wird mit ε bezeichnet.
- Besondere Wortmengen:
- $\Sigma^0 = \{\varepsilon\}$
- $\Sigma^* = \bigcup_{n=0}^{\infty} \Sigma^n$
- $\Sigma^+ = \bigcup_{n=1}^{\infty} \Sigma^n$

Die Operationen M^* und M^+ auf einer Menge M werden als

- Kleene-$*$-Abschluss oder
- Kleene-$+$-Abschluss bezeichnet.

Beispiel

- $\Sigma_{lat}^* = \{\varepsilon, a, b, \ldots, z, aa, ab, \ldots, zz, aaa, \ldots\}$
- $\Sigma_{lat}^+ = \{a, b, \ldots, z, aa, ab, \ldots, zz, aaa, \ldots\}$

Beispiel

Sei $M = \{01, 2\}$, so ergeben sich u.a. diese Wortmengen:

$$\begin{array}{lcl} M^0 & = & \varepsilon \\ M^1 & = & 01, 2 \\ M^2 & = & 0101, 012, 201, 22 \\ M^3 & = & 010101, 01012, 01201, 0122, 20101, 2012, 2201, 222 \\ & \ldots & \\ M^+ & = & M^1 \cup M^2 \cup \ldots = 01, 2, 0101, 012, 201, 22, 010101, 01012, \ldots \\ M^* & = & M^0 \cup M^+ = \varepsilon, 01, 2, 0101, 012, 201, 22, 010101, 01012, \ldots \end{array}$$

Beobachtung

Die Wortlänge $|\omega|$ für ein $\omega \in L^*$ hängt von der Definition des Alphabets ab. So ist in diesem Beispiel $|222| = 3$ während $|0101| = 2$ ist.

Produkt und Konkatenation

Definition

Die Konkatenation von zwei Wörtern $\omega = (\omega_1, \dots, \omega_n)$ und $\upsilon = (\upsilon_1, \dots, \upsilon_m)$ ist definiert als das Wort, das durch ein aneinanderreihen der beiden Wörter entsteht:

$$\omega \cdot \upsilon = \omega\upsilon = (\omega_1, \dots, \omega_n) \cdot (\upsilon_1, \dots, \upsilon_m) = w_1 \dots w_n \upsilon_1 \dots \upsilon_m$$

Das leere Wort ist $\omega^0 = \varepsilon$ und die n-te Potenz von ω ist:

$$\omega^n = \underbrace{\omega \cdot \dots \cdot \omega}_{n \text{ Faktoren}} \text{ für } n > 0$$

Beispiel

Sei $\Sigma = a, e, n, r$, sowie $\omega = \text{na} \in \Sigma^*$ und $\upsilon = \text{er} \in \Sigma^*$.

$\omega^2 = \text{nana}$, $\upsilon\omega = \text{erna}$ und $\upsilon\omega^2\upsilon = \text{ernanaer}$

Abschluss-Eigenschaften

Bemerkung

Der Begriff *Abschluss in obiger Definition* bedeutet:

Auf einer Menge mit einer Verknüpfung liefert jede Anwendung der Operation mit Elementen wieder ein Element aus der Menge.

Beispiel

- die Subtraktion ist auf den natürlichen Zahlen nicht abgeschlossen,
- der Abschluss der natürlichen Zahlen bezüglich der Subtraktion sind die ganzen Zahlen.

Die Kleene-Abschlüsse und Multiplikationen werden später in regulären Ausdrücken auf Wörtern verwendet, damit ist dann der Abschluss oder das kartesische Produkt der Menge mit genau diesem Wort gemeint.

Beispiel

$$\begin{aligned} (ab)^+ &= \{ab\}^+ &&= \{ab, abab, ababab, \dots\} \\ cd^*e &= \{c\} \times \{d\}^* \times \{e\} &&= \{ce, cde, cdde, cddde, \dots\} \end{aligned}$$

Übung

Alphabet Σ = {a,el,en,g,l,ste}

Gegeben sei das Alphabet $\Sigma = \{a, el, en, g, l, ste\}$. Welche der folgenden Worte liegen in Σ^4?

ω_1 = galgen, ω_2 = stelle, ω_3 = sagen, ω_4 = lagen, ω_5 = allen, ω_6 = aalen

Alphabet Σ = {e,en,in,r,t,u}

Gegeben sei das Alphabet $\Sigma = e, en, in, r, t, u$. Welche der folgenden Worte liegen in Σ^5?

ω_1 = reiner, ω_2 = teurer, ω_3 = treuer, ω_4 = teuren, ω_5 = retten, ω_6 = teuer

Übung

Alphabet Σ = {e,g,in,l,s,ter}

Gegeben sei das Alphabet $\Sigma = e, g, in, l, s, ter$. Welche der folgenden Worte liegen in Σ^*?

ω_1 = tester, ω_2 = seile, ω_3 = lines, ω_4 = segel, ω_5 = seinen, ω_6 = erster

Formale Sprachen

Definition

Jede Teilmenge $L \subseteq \Sigma^*$ ist eine formale Sprache über dem Alphabet Σ.

Beispiel

Sei $\Sigma = \{0, 1, 2\}$, dann ist Σ^* die Menge oder Sprache von Wörtern aus den Ziffern 0, 1 oder 2 beliebiger Länge wie 101 oder auch 0001.

1

Die Menge $M \subset \Sigma^*$ der binären Zahlen ohne führende Nullen:

$$M = \{0\} \cup \{1\} \times \{0, 1\}^* = \{0, 1, 10, 11, 100, 101, 110, 111, 1000, \dots\}$$

2

Die Menge $M \subset \Sigma^*$ von einer gleichen Anzahl von 0 und 1 in dieser Reihenfolge:

$$M = \{0^n 1^n | n \in \mathbb{N}\} = \{01, 0011, 000111, 00001111, 0000011111, \dots\}$$

3

Die Wörter $M \subset \Sigma^*$ mit gleicher Anzahl von 0, 1 und 2 in dieser Reihenfolge:

$$M = \{0^n 1^n 2^n | n \in \mathbb{N}\} = \{012, 001122, 000111222, 000011112222, \dots\}$$

4

Die Menge $M \subset \Sigma^*$ mit Wörtern der Länge von Zweierpotenzen:

$$M = \{w \in \Sigma^* || w| = 2^n, n \in \mathbb{N}\} = \{0, 1, 2, 00, 01, \dots, 21, 22, 0000, \dots\}$$

Übung

Wörter bestimmen

Bestimmen Sie die Wörter der folgenden Sprache:

$$L = \{acx^m(zq)^n | n \in \{0, 1\}, m \in \{1, 2\}\}$$

Wörter bestimmen

Bestimmen Sie die Wörter der folgenden Sprache:

$$L = \{(b^m a)^l za | m \in \{0, 1\}, l \in \{1, 2, 3\}\}$$

Abzählbar (unendlich)

Beobachtung

Selbst mit endlichen Alphabeten können formale Sprachen unendlich groß sein.

Definition

Eine Menge M ist *abzählbar*, wenn die einzelnen Elemente abzählbar sind, es also eine bijektive Funktion $f : N \to M$ von den natürlichen Zahlen $N = \mathbb{N}$ oder einer Teilmenge der natürlichen Zahlen $N \subset \mathbb{N}$ auf M gibt.

Wenn es keine solche Funktion geben kann, so ist die Menge *überabzählbar unendlich*.

Satz

Jede endliche Menge ist abzählbar.

Beweis

Eine endliche Menge M hat eine endliche Anzahl $n = |M|$ von Elementen.

Wird nun beginnend von $M_0 = M$ und $k = 1$ in n Schritten jeweils ein Element m_k der Menge M_{k-1} entnommen mit $M_k = M_{k-1} \setminus \{m_k\}$, so ist induktiv $|M_k| = |M_{k-1}| - 1 = n - k$ und es ist $M_n = \emptyset$.

Die Bijektion lautet dann $f : N \to M$ mit $f(k) = m_k$ mit $N = \{1, \ldots, n\}$. ∎

Satz

Jede Teilmenge $M \subseteq N$ einer abzählbaren Menge $N = \{n_1, n_2, \ldots\}$ ist abzählbar.

Beweis

Sei $f(k) = n_k$ die Abzählung der Menge N. Sei $R = \{k \in \mathbb{N} | n_k \in M\}$; d. h. die Menge der Indizes der Elemente aus N, die in M sind. Dann ist die Einschränkung $f_{|R} : R \to M$ von f genau die Abzählung, die die Abzählbarkeit von M beweist. ∎

Beispiel

Eine abzählbar unendliche Menge sind — zum Beispiel:

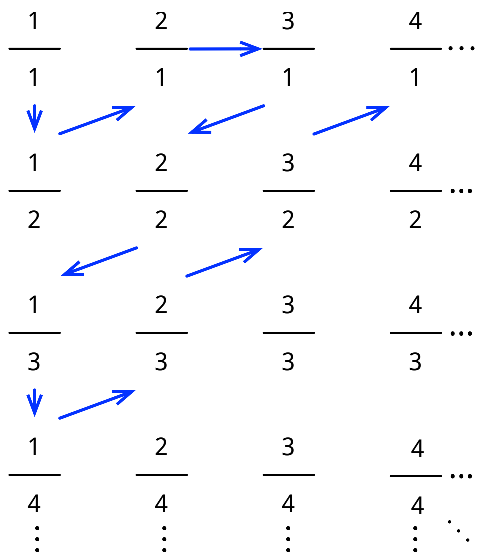
- die geraden Zahlen $\{2n | n \in \mathbb{N}\}$
- die Quadratzahlen $\{n^2 | n \in \mathbb{N}\}$
- die Menge der Fakultäten $\{n! | n \in \mathbb{N}\}$
- die ganzen Zahlen $\mathbb{Z}$ mit der Funktion:

$$f(n) = \begin{cases} n/2 & \text{für } n \text{ gerade} \\ -(n+1)/2 & \text{für } n \text{ ungerade} \end{cases}$$

$$f(1) = 0,\ f(2) = 1,\ f(3) = -1,\ f(4) = 2,\ f(5) = -2,\ \ldots$$

Beispiel

Die rationalen Zahlen $\mathbb{Q}$ sind abzählbar unendlich.



Rationale Zahlen können als Brüche dargestellt werden und mit Hilfe des Diagonalisierungsverfahren von Cantor (auch: Cantors erstes Diagonalargument) in eine Bijektion zu den natürlichen Zahlen gebracht werden.

Die 0 und alle negativen Brüche können wie zuvor eingeschoben werden. Auch alle rationalen Vektoren $\mathbb{Q}^n$ in beliebiger Dimension $n \in \mathbb{N}$ sind so abzählbar.

Satz

Für jede endliche Menge oder Alphabet Σ ist deren Kleene-Abschluss Σ^* abzählbar.

Beweis

Ist das Alphabet Σ leer, so ist auch Σ^* leer, und damit für $N = \emptyset$ trivial abzählbar.

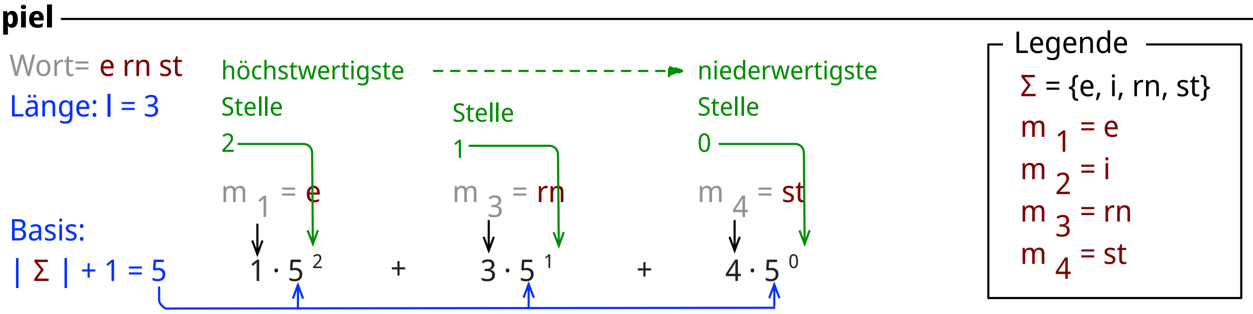
Ist Σ nicht leer, dann besitzt Σ mit Größe $n = |\Sigma|$ eine Aufzählung m_k mit $k = 1, \ldots, n$.

Jedes Wort $w = m_{k_1} m_{k_2} \ldots m_{k_l}$ kann dann im Stellenwertsystem zur Basis $n+1$ dargestellt werden:

$$1 + k_1 \cdot (n+1)^{l-1} + k_2(n+1)^{l-2} + \ldots + k_l(n+1)^0$$

und somit der Zahl $1 + (k_1 k_2 \ldots k_l)_{(n+1)}$[1] zugeordnet werden.

Beispiel



Die Abbildung $f : N \to \Sigma^*$ mit $N \subseteq \mathbb{N}$ ergibt sich für $f(x)$ aus der Stellenwertdarstellung von $x - 1 > 0$ zur Basis $n+1$ beginnend mit der höchstwertigen Ziffer k_1 bis zur letzten Stelle k_l.

Das Bild $f(x)$ ist dann das Wort $m_{k_1} m_{k_2} \ldots m_{k_l}$.

Das leere Wort ε wird von 1 abgebildet und entsprechend ist $f(1) = \varepsilon$. ∎

Beispiel

Sei $\Sigma = \{e, i, rn, st\}$ mit Aufzählung $m_1 = e$, $m_2 = i$, $m_3 = rn$, und $m_4 = st$, dann haben die folgenden Wörter diese Abzählung nach Stellenwert:

x	1	2	3	4	5

	1	$1 \quad + \quad 1$ $1_5 \quad + \quad 1_5$ [2]	$1 \quad + \quad 2$ $1_5 \quad + \quad 2_5$	$1 \quad + \quad 3$ $1_5 \quad + \quad 3_5$	$1 \quad + \quad 4$ $1_5 \quad + \quad 4_5$
Wort	ϵ	e	i	rn	st
$f(x)$	$f(1) = \epsilon$	$f(2) = e$	$f(3) = i$	$f(4) = rn$	$f(5) = st$ *(Anm.: k ist 4 für st)*

x	...	$7 = 1 + 6$ $12_5 = 1_5 + 11_5$	$8 = 1 + 7$ $13_5 = 1_5 + 12_5$	...	$45 = 1 + 44$ $140_5 = 1_5 + 134_5$	...
	...	$1 \quad + \quad 1 \cdot 5 + 1$ $1 \quad + \quad 11_5$	$1 \quad + \quad 1 \cdot 5 + 2$ $1 \quad + \quad 12_5$	...	$1 \quad + \quad 1 \cdot 25 + 3 \cdot 5 + 4$ $1 \quad + \quad 134_5$	...
Wort	...	ee	ei	...	$e\,rn\,st$	...

Unbesetzt bleibt, wo eine 0 in der Stellenwertdarstellung vorliegt. Zum Beispiel ist $f(6) = 1 + 1 \cdot 5^1 + 0 \cdot 5^0 = 1_5 + 10_5$.

Satz

Jede formale Sprache is abzählbar.

Beweis

Da jede formale Sprache L über einem endlichen Alphabet Σ definiert ist, ist das eine direkte Folge aus vorherigem Satz, dass Σ^* abzählbar ist, und wie zuvor gezeigt damit auch die Teilmenge $L \subseteq \Sigma^*$ abzählbar ist. ∎

Abzählen mit Hilfe von Gödelnummern

Gödelnummern unterstützen abzählbarer un-/endliche Mengen. Letzeres (abzähbar unendlich) ist mit einem einfachen Stellenwertsystem zur Basis der Anzahl der Elemente und des somit (zwangsweise) endlichen Alphabets nicht möglich.

Definition

Sei (p_n) die Folge der Primzahlen:

$p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7, p_5 = 11, p_6 = 13, \ldots$

Für eine abzählbare Menge $M = m_1, m_2, \ldots$ ist die Gödelnummer $c_M : M^* \to \mathbb{N}$ des Tupels $w = (m_{k_1}, m_{k_2}, \ldots, m_{k_l})$ gegeben durch

$$c_M(w) = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_l^{k_l} = \prod_{i=1}^{l} p_i^{k_i}$$

Beispiel

Sei $\Sigma = \{e, i, rn, st\}$ mit Aufzählung $m_1 = e, m_2 = i, m_3 = rn$ und $m_4 = st$, dann haben die folgenden Wörter diese Gödelnummern:

Wort	ϵ	e	i	rn	st	$e\,rn\,st$
$c_M(w)$	$2^0 = 1$	2^1 $p_1^{k_1=1}$	2^2 $p_1^{k_2=2}$	2^3	2^4	$2^1 \cdot 3^3 \cdot 5^4 = 33750$

Beobachtung
Unbesetzt bleibt, wo bis zum höchsten Primzahlfaktor davor eine Primzahlpotenz 0 ist.

Z. B. ist die Primzahlzerlegung von $10 = 2^1 \cdot 3^0 \cdot 5^1$. Somite gäbe es an der zweiten Stelle *kein Zeichen* was unsinnig ist.

Satz
Die Menge von endlichen Folgen $P = \{p = (w_1, w_2, \ldots, w_n) | w_k \in L, n \in \mathbb{N}\}$ aus Wörtern einer formalen Sprache $L \subseteq \Sigma^*$ (also Programmen) über einem Alphabet Σ ist abzählbar.

Beweis
Jede formale Sprache $L \subseteq \Sigma^*$ ist abzählbar. Damit kann nach Definition für jede Folge $p \in P$ injektiv eine Gödelnummer $c_L(p)$ über L bestimmt werden.

Auf der Menge $N = \{x = c_L(p) | p \in P\}$ kann die Umkehrung $f : N \to P$ von c_L auf P eingeschränkten bijektiven Funktion $c_{L|P} : P \to N$ bestimmt werden, und damit ist P abzählbar. ∎

---

[2] Wir haben immer $1 + \ldots$, da wir noch das leere Wort ε haben.

[1] Die Darstellung $(k_1 k_2 \ldots k_l)_{(n+1)}$ ist die Stellenwertdarstellung zur Basis $n + 1$ des Wortes w.

Überabzählbar unendlich

Satz

Die Menge der reellen Zahlen $r \in (0,1) \subset \mathbb{R}$ ist überabzählbar unendlich.

Beweis

Cantor's (zweites) Diagonalargument

Angenommen die reellen Zahlen sind als Binärbrüche wie folgt abzählbar:

$$\begin{array}{ccl} r_1 & = & 0, x_{11}x_{13}x_{13}x_{14}x_{15}\ldots \\ r_2 & = & 0, x_{21}x_{23}x_{23}x_{24}x_{25}\ldots \\ r_3 & = & 0, x_{31}x_{33}x_{33}x_{34}x_{35}\ldots \\ r_4 & = & 0, x_{41}x_{43}x_{43}x_{44}x_{45}\ldots \\ \vdots & & \vdots \end{array}$$

Sei jetzt $r = 0, \overline{x_{11}}\,\overline{x_{22}}\,\overline{x_{33}}\,\overline{x_{44}}\,\overline{x_{55}}\ldots \in (0,1)$, dann ist r nicht in der Abzählung und es liegt ein Widerspruch zur Annahme vor. $\mathbb{R}$ ist also überabzählbar unendlich.

$\bar{x}$ ist das einfache Komplement von x. Das bedeutet, dass 0 durch 1 und 1 durch 0 ersetzt wird.

Beachte, dass r über die gesamte (unendliche) Diagonale definiert ist und dadurch zu jeder bestehenden Zahl unterschiedlich sein muss; d. h. r ist nicht gleich zu r_1 in der ersten Stelle, nicht gleich zu r_2 in der zweiten Stelle, nicht gleich zu r_3 in der dritten Stelle, … und nicht gleich r_n in der n-ten Stelle.

Die Kardinalität (bereits) der Menge der reellen Zahlen im Bereich $(0,1)$ ist also größer als die der natürlichen Zahlen.

∎

Schlussfolgerungen aus der Überabzählbarkeit

1

Angenommen:

- jedes in einer formalen Sprache geschriebenes Programm löst ein Problem
- wir interpretieren dies als Berechnung einer Lösung

So sind dies verschwindend wenige lösbare Probleme verglichen schon mit der Reichhaltigkeit der reellen Zahlen im Intervall $(0, 1)$.

Schlussfolgerung

Soweit davon auszugehen ist, dass die Teilmenge der in der Realität tatsächlich relevanten reellen Zahlen tatsächlich auch überabzählbar ist, wird es nie möglich sein, für alle Fragestellungen über solche Zahlen Lösungen in der Form von Programmen über einer gegebenen formalen Sprache zu formulieren.

2

Schlussfolgerung

Gleichzeitig ist aber auch die Anzahl der formalen Sprachen sehr groß.

Beweis

Für jede reelle Zahl $x \in R$ mit Nachkommastellen $r1r2...$ gibt es eine formale Sprache L_x über Σ_{Zahl}:

$$L_x = \{r_1r_2...r_n \in \Sigma^*_{\text{Zahl}} | x \text{ hat die ersten } n \text{ Nachkommastellen } r_1...r_n\}$$

Beispielsweise ist $L_\pi = 1, 14, 141, 1415, 14159, 141592, 1415926, ...$. Damit ist die Anzahl der formalen Sprachen mindestens so groß, wie die Anzahl reeller Zahlen im Intervall $(0, 1)$, also aller möglichen Nachkommastellen in $\mathbb{R}$, zuzüglich der 0, und damit nach vorherigem Satz überabzählbar unendlich. ∎

Übung

Stellenwerte I

Gegeben sei das Alphabet $\Sigma = a, gen, i, re$ mit Aufzählung in dieser Reihenfolge. Bestimmen Sie die Zahlen n nach Stellenwert mit Bild $f(n)$ der Wörter $regen$, $aare$ und die Worte mit Stellenwert 15, 118.

Stellenwerte II

Gegeben sei das Alphabet $\Sigma = e, h, r, ste$ mit Aufzählung in dieser Reihenfolge. Bestimmen Sie die Zahlen n nach Stellenwert mit Bild $f(n)$ der Wörter $steh$, $rehe$ und die Worte mit Stellenwert 45, 1417.

Übung

Gödelnummern I

Gegeben sei das Alphabet $\Sigma = e, l, ste, te$ mit Aufzählung in dieser Reihenfolge. Bestimmen Sie die Gödelnummer $c(w)$ der Wörter $este$, $elle$ und die Worte mit Gödelnummer 720, 12600.

Gödelnummern II

Gegeben sei das Alphabet $\Sigma = h, he, re, ste$ mit Aufzählung in dieser Reihenfolge. Bestimmen Sie die Gödelnummer $c(w)$ der Wörter $steh$, $reste$ und die Worte mit Gödelnummer 144, 1500.

Übung

Gödelnummern und ChatGPT

Eine Befragung von ChatGPT zum Thema Gödelnummern ergab, dass ChatGPT vorgeschlagen hat allen Zeichen $a \in \Sigma$ eine Primzahl zuzuordnen und dann für das Vorkommen eines Zeichens a an Stelle i den aktuellen Wert mit der Primzahl des Zeichens hoch i zu multiplizieren.

Beispiel

Sei $\Sigma = \{a, b, c, d\}$

Zuweisung von Primzahlen an Symbole: $a \to 2$, $b \to 3$, $c \to 5$, $d \to 7$

Für das Wort: $abac$ wäre nach dem von ChatGPT vorgeschlagenen Verfahren die Gödelnummer $c(abac) = 2^1 \cdot 3^2 \cdot 2^3 \cdot 5^4 = 90\,000$.

Bewerten Sie diesen Vorschlag.

Verknüpfungen und Entscheidbarkeit

Verknüpfungen von formalen Sprachen

Satz

Sind L_1 und L_2 zwei formale Sprachen über den Alphabeten Σ_1 und Σ_2, so gilt:

1. Die Vereinigung $L_\cup = L_1 \cup L_2$ ist eine formale Sprache über dem Alphabet $\Sigma_1 \cup \Sigma_2$.
2. Der Schnitt $L_\cap = L_1 \cap L_2$ ist eine formale Sprache über dem Alphabet $\Sigma_1 \cup \Sigma_2$.

Beweis

Die Vereinigung der Alphabete $\Sigma = \Sigma_1 \cup \Sigma_2$, also zweier endlicher Mengen, ist wieder eine endliche Menge und damit ein Alphabet. Da sowohl $\Sigma_k \subseteq \Sigma$ für $k = 1, 2$, sind L_1 und L_2 auch Sprachen über Σ und es gilt $L_k \subseteq \Sigma_k^* \subseteq \Sigma^*$, da die Teilmengenbeziehung in jeder Mengenpotenz und damit auch in deren Vereinigung gilt. Damit sind auch $L_1 \cup L_2 \subseteq \Sigma^*$ und $L_1 \cap L_2 \subseteq \Sigma^*$ und damit Sprachen über $\Sigma = \Sigma_1 \cup \Sigma_2$. ∎

Satz

Sind L_1 und L_2 zwei formale Sprachen über den Alphabeten Σ_1 und Σ_2, so gilt:

3. Das Komplement $\overline{L_k} = \Sigma_k^* \setminus L_k, k = 1, 2$ ist formale Sprache über Alphabet Σ_k.

Beweis

Nach Definition der Mengendifferenz gilt $\Sigma_k^* \setminus L_k \subseteq \Sigma_k^*$. Somit ist $\overline{L_k} \subseteq \Sigma_k^*$ und somit eine Sprache über Σ_k. ∎

Satz

Sind L_1 und L_2 zwei formale Sprachen über den Alphabeten Σ_1 und Σ_2, so gilt:

4. Das Produkt $L_1 L_2 = \{w_1 w_2 | w_1 \in L_1, w_2 \in L_2\}$ ist eine formale Sprache über $\Sigma_1 \cup \Sigma_2$.

Beweis

Für $L = L_1 \cup L_2$ ist $L_1 \subseteq L$ und $L_2 \subseteq L$.

L ist somit eine Sprache über $\Sigma = \Sigma_1 \cup \Sigma_2$ nach Satz 1. Damit ist ist jedes Wort $w \in L \subseteq \Sigma^*$ in einem $w \in \Sigma^k$ für ein bestimmtes k enthalten. Ebenso ist damit $w_1 w_2 \in \Sigma^{k_1}\Sigma^{k_2} = \Sigma^{k_1+k_2} \subseteq \Sigma^*$. Damit ist $LL \subseteq \Sigma^*$ und damit $L_1 L_2 \subseteq LL \subseteq \Sigma^*$ Sprache über Σ. ∎

Satz

Sind L_1 und L_2 zwei formale Sprachen über den Alphabeten Σ_1 und Σ_2, so gilt:

5. Kleensche Abschlüsse L_k^* und L_k^+ , $k = 1, 2$ sind formale Sprachen über Σ_k.

Beweis

Beobachtung

Zunächst ist $\varepsilon \in \Sigma_k^*$, somit reicht es für L_k^+ zu argumentieren.

- Jedes Wort $w \in L_k^+$ ist in $w \in L_k^n$ für ein bestimmtes n.
- Damit gibt es Teilworte $m_1 m_2 \ldots m_n = w$ mit $m_i \in L_k$
- Da $L_k \subseteq \Sigma_k^*$ gibt es p_i , so dass $m_i \in \Sigma_k^{p_i}$
- Damit ergibt sich, dass $m_1 \ldots m_n \in \Sigma_k^{\Sigma p_i}$ liegt und damit in $\Sigma_k^{\Sigma p_i} \subseteq \Sigma_k^*$
- Damit dies für alle Worte in L_k^+ gilt, ist $L_k^+ \subseteq \Sigma_k^*$ und damit eine Sprache über Σ_k

∎

Beispiel

Komplement einer Sprache

Gegeben

Alphabet: $\Sigma_k = \{a, b\}$

Sprache: math:*L_k*: Alle Wörter, die mit dem Symbol a beginnen.

$$L_k = \{a, aa, ab, aaa, aab, \ldots\}$$

Komplement der Sprache:

Das Komplement $\overline{L_k}$ enthält alle Wörter aus Σ_k^*, die *nicht* mit a beginnen. Das bedeutet:

$$\overline{L_k} = \{\epsilon, b, bb, ba, bba, bbb, \ldots\}$$

Existenz der Abzählbarkeit

Wiederholung

Sind L_1 und L_2 abzählbar, so sind mit entsprechenden Anpassungen auch

- Vereinigung,
- Schnitt und
- Produkt

abzählbar.

Beobachtung

Die Abzählbarkeit des Komplements kann nicht so einfach beantwortet werden!

Dies ist jedoch kein Problem, da jede formale Sprache abzählbar ist und damit auch ihr Komplement.

Frage

Kann mit dem Wissen der Existenz auch die tatsächliche Abzählung angegeben werden?

Zusammenfassung

Wir unterscheiden deswegen die einfache und nicht konstruktivistische Erkenntnis einer Abzählbarkeit von einer konstruktiven und praktischen Aufzählbarkeit.

Definition

Eine Menge oder Sprache M ist **aufzählbar** oder **rekursiv aufzählbar**, wenn eine surjektive Abbildung $f : N \to M$ bekannt ist, die nach endlichen Schritten für jedes $n \in N$ die Berechnung von $f(n)$ ermöglicht, falls $M \neq \emptyset$. Daraus ergibt sich eine Aufzählung von M durch die Folge $(f(1), f(2), \ldots)$.

Bemerkung

Die Bedeutung der „Berechenbarkeit“ wird später im Sinne eines „Programms“ erklärt.

„aufzählbar“: bezieht sich auf die Existenz der Aufzählung als „berechenbare Funktion“,

„rekursiv aufzählbar“: bezieht sich auf die Existenz eines „Programms“, was aber hier äquivalent ist.

Aufzählbarkeit

Satz

Sei Σ ein Alphabet, dann ist Σ^* aufzählbar.

Beweis

Die Konstruktion aus dem früheren Satz zur Abzählbarkeit von Σ^* ist schon eine konstruktive Aufzählung von Σ^*.

Die nicht zugeordneten natürlichen Zahlen werden beispielsweise auf das jeweils zuletzt zugeordneten Wort abgebildet. ∎

Zusammenfassung

Zwischen den Bezeichnungen „aufzählbar“ zu „abzählbar“ ist der relevante Unterschied in der konstruktiven Kenntnis der Aufzählbarkeit im Gegensatz von der nicht konstruktiven Gewissheit der Abzählbarkeit.

Achtung!

Es ist aber kein Verfahren bekannt, wie aus einer allgemeinen Aufzählung einer Sprache konstruktiv eine Aufzählung des Komplements abgeleitet werden kann. Das Gleiche gilt bei zwei aufgezählten Sprachen für deren Schnitt.

Die Übertragung der Eigenschaft der Aufzählbarkeit muss mit Angabe eines ausführbaren Algorithmus erfolgen.

So kann - wie bei der Aufzählung von $\mathbb{Z}$ - bei der Vereinigung abwechselnd die eine oder die andere Aufzählung verwendet werden. Die Aufzählung der rationalen Zahlen kann nach dem vorgestellten Verfahren von Cantor erfolgen. Die gilt ggf. auch für das Produkt.

Entscheidungsproblem

Definition

Das *Entscheidungsproblem* bezeichnet die Frage, ob für ein Problem ein ausführbares Verfahren angegeben werden kann, mit dem in endlich vielen Schritten eine Entscheidung für das Problem bestimmt wird.

Ein Problem ist …

entscheidbar: wenn ein solches Verfahren existiert

nicht-entscheidbar: wenn es ein solches Verfahren nicht geben kann

semi-entscheidbar: wenn ein Verfahren existiert, das nach endlich vielen Schritten die Entscheidung für eine Klasse von möglichen Antworten bestimmt

Wortproblem

(Ein Beispiel für ein Entscheidbarkeitsproblem.)

Definition

Sei L eine Sprache über Σ und $w \in \Sigma^*$. Das Wortproblem bezeichnet die Frage, ob w Teil der Sprache ist, also entweder $w \in L$ oder $w \notin L$ gilt.

Satz

Sind L und $\bar{L}$ aufzählbare Sprachen über dem Alphabet Σ, so ist das Wortproblem $w \overset{?}{\in} L$ für ein $w \in \Sigma^*$ entscheidbar.

Dann werden L und $\bar{L}$ als *entscheidbare Sprachen* oder *rekursive Sprachen* bezeichnet.

Beweis

Es seien $f_L : \mathbb{N} \to L$ und $f_{\bar{L}} : \mathbb{N} \to \bar{L}$ die Aufzählungen von L und $\bar{L}$.

Abwechselnd wird aufsteigend — beginnend bei $k = 1$ — das Wort w mit $f_L(k)$ und $f_{\bar{L}}(k)$ verglichen. Nach endlicher Anzahl von Schritten ist $f_L(k) = w$, dann ist $w \in L$, oder $f_{\bar{L}}(k) = w$, dann ist $w \notin L$. ∎

Ist L aufzählbar, doch $\bar{L}$ nicht, so endet das Verfahren, genau dann wenn $w \in L$ ist. Daher ist Wortproblem aufzählbarer Sprachen semi-entscheidbar.

Satz

Jede entscheidbare Sprache ist aufzählbar.

Beweis

Jede formale Sprache L basiert auf einem Alphabet Σ_L. Damit ist der Abschluss Σ_L^* mit f_{Σ^*} aufzählbar und $L \subseteq \Sigma_L^*$. Entweder ist die Sprache L leer, oder es gibt ein Wort $w_0 \in L$.

Wenn L entscheidbar ist, so kann für jedes $n \in \mathbb{N}$ in endlichen Schritten bestimmt werden, ob $f_{\Sigma_L^*}(n) \in L$ ist. Wenn ja, so ist $f_L(n) = f_{\Sigma_L^*}(n)$, und sonst $f_L(n) = w_0$. ∎

Damit gilt:

rekursive bzw. entscheidbare Sprache ⇒ rekursiv aufzählbare Sprache

semi-entscheidbare Sprache ⇐ rekursiv aufzählbare Sprache

Beobachtung

Eine rekursive Aufzählung kann die Sprache völlig durcheinander aufzählen.

Es ist nie sicher, ob frühe Lücken zur Stellenwertaufzählung später aufgefüllt werden.

Das Collatz-Problem

Definition

Die Collatz-Funktion $f : \mathbb{N} \to \mathbb{N}$ ist definiert als:

$$f(n) = \begin{cases} n/2 & \text{für gerade } n \\ 3n+1 & \text{für ungerade } n \end{cases}$$

Das Collatz-Problem besteht darin, für ein gegebenes n die Folge $f(n), f(f(n)), f(f(f(n))), \ldots$ zu betrachten und zu entscheiden, ob die Folge irgendwann den Wert 1 erreicht.

Beispiel

$$f(6) = 3, f(3) = 10, f(10) = 5, f(5) = 16,$$

$$f(16) = 8, f(8) = 4, f(4) = 2, f(2) = 1, \ldots$$

Die Folge erreicht für $n = 6$ also den Wert 1 nach 8 Schritten.

Satz

Das Collatz-Problem ist semi-entscheidbar.

Beweis

Die Collatz-Folge kann für ein gegebenes n in endlich vielen Schritten berechnet werden.

Wenn die Folge den Wert 1 erreicht, so ist das Problem entschieden.

Wenn die Folge nicht den Wert 1 erreicht, so ist das Problem nicht entschieden, aber es ist auch nicht sicher, ob die Folge den Wert 1 nicht doch noch erreicht.

∎

Das Collatz-Problem kann direkt in eine Collatz-Sprache über Σ_{Zahl} übertragen werden:

$$L_{\text{Collatz}} = \{n \in N | \exists k \in \mathbb{N}_0 : f^k(n) = 1\}$$

Das Wortproblem auf dieser Sprache ist damit — hier nach Definition des Problems statt einer Aufzählung — ebenso mindestens semi-entscheidbar.
Ob das Problem auch entscheidbar ist, konnte bisher niemand beantworten. Die naive Methode des Ausprobierens, ob es überhaupt ein $w \in N$ mit $w \notin L_{\text{Collatz}}$ gibt, hat trotz intensiver Suche bisher nicht geendet.

Das Halteproblem

Definition

Das Halteproblem ist die Fragestellung, ob die Ausführung eines Programms p bei gegebenen Eingabedaten x nach endlichen Schritten terminiert.

Das Halteproblem ist die verallgemeinerte Fragestellung zum Collatz-Problem. Entsprechend ist die äquivalente Sprache:

$$L_{\text{Halteproblem}} = \{(p, x) \in \Sigma^*_{\text{Unicode}} \times \Sigma^*_{\text{Unicode}} | p(x) \text{ terminiert nach endlichen Schritten } \}$$

nur semi-entscheidbar, da durch Ausführung des Programms nur $(p, x) \in L_{\text{Halteproblem}}$ gezeigt werden kann.

Bemerkung

Alan Turing konnte beweisen, dass es keinen Algorithmus gibt, der die Entscheidung $(p, x) \notin L_{\text{Halteproblem}}$ für beliebige p und x in endlicher Zeit beantwortenkann.

Übung

Collatz-Funktion

Die parametrisierte Collatz-Funktion $f_{\alpha,\beta}(n) : \mathbb{N} \to \mathbb{N}$ laute für $\alpha, \beta \in \mathbb{N}$:

$$f_{\alpha,\beta}(n) = \begin{cases} n/2 & \text{für } n \text{ gerade} \\ \alpha \cdot n + \beta & \text{sonst} \end{cases}$$

1. Bestimmen Sie mit einem Programm das kleinste $k \in \mathbb{N}$ für das $f_{3,1}^k(27) = 1$ ist.
2. Sei die Sprache $L_{\text{Collatz}_{3,7}} = \{n \in \mathbb{N} | \exists k \in \mathbb{N}_0 : f_{3,7}^k(n) = 1\}$.

   Bestimmen Sie mit einem Programm die Menge $M = \bar{L}_{3,7} \cap [1, 20]$.

Übung

Rekursive Sprachen

Es seien L_1 und L_2 rekursive Sprachen über dem Alphabet Σ. Sei $L = L_1 \setminus L_2$.

Zeigen oder widerlegen Sie, dass L eine rekursive Sprache über Σ sei.

Grammatiken

Englische Grammatik (Beispielhaft)

$$\underbrace{\underbrace{\underbrace{The}_{\text{Bestimmungswort}}\ \underbrace{professor}_{\text{Substantiv}}}_{\text{Substantivphrase}}\ \underbrace{\underbrace{should}_{\text{Modalverb}}\ \underbrace{teach}_{\text{Hauptwort}}}_{\text{Verbphrase}}}_{\text{Satz}}.$$

$$\underbrace{\underbrace{\underbrace{One}_{\text{Bestimmungswort}}\ \underbrace{student}_{\text{Substantiv}}}_{\text{Substantivphrase}}\ \underbrace{\underbrace{might}_{\text{Modalverb}}\ \underbrace{listen}_{\text{Hauptwort}}}_{\text{Verbphrase}}}_{\text{Satz}}.$$

Ein Satz `S` wird mit diesen Regeln `R` gebildet:

- Ein Satz besteht aus einer Substantivphrase und einer Verbphrase.
- Eine Substantivphrase hat ein optionales Bestimmungswort und ein Substantiv.
- Eine Verbphrase besteht aus optionalem Modalverb und einem Hauptverb.
- Ein Bestimmungswort ist 🇺🇸 *the* oder 🇺🇸 *one*.
- Ein Substantiv ist 🇺🇸 *student* oder 🇺🇸 *professor*.
- Ein Modalverb ist 🇺🇸 *should* oder 🇺🇸 *might*.
- Ein Hauptverb ist 🇺🇸 *listen* oder 🇺🇸 *teach*.

Darin wurden diese Variablen `V` und Symbole `T` verwendet:

`V`:	{Satz,Substantivphrase,Verbphrase,Bestimmungswort,Substantiv, Modalverb,Hauptverb}
`T`:	{the,one,student,professor,should,might,listen,teach}.

Grammatiken

Definition

Eine Grammatik ist ein Tupel $G = (V, T, R, S)$, wo

V:	das Alphabet der Variablen,
T:	das Alphabet der Terminalen Symbole mit $V \cap T = \emptyset$,
$R = r_1, \ldots, r_n$:	die endliche Menge der Regeln
	r_k: $(V \cup T)^* \setminus T^* \to (V \cup T)^*$
$S \in V$:	das Startsymbol ist.

Die Regeln von Grammatiken werden auch Produktionen genannt

Ableitungen

Definition

Sei $G = (V, T, R, S)$ eine Grammatik. Eine Ableitung ist die Anwendung einer Regel $r \in R$ mit $a \mapsto b$ auf das Wort $w_1 \in (V \cup T)^*$ zum Wort $w_2 \in (V \cup T)^*$, geschrieben $w_1 \overset{r}{\Rightarrow} w_2$, wenn es $x, y \in (V \cup T)^*$ gibt, so dass:

$$\begin{array}{ccccc} w_1 & = & x & a & y \\ \Downarrow_r & & & \downarrow_r & \\ w_2 & = & x & b & y \end{array}$$

Definition

Eine transitive Ableitung $w_1 \overset{*}{\Rightarrow} w_n$ ist die Anwendung keiner oder beliebig vieler Regeln $r \in R$, um von w_1 auf w_n zu schließen. Die Sprache einer Grammatik $L(G)$ ist die Menge aller möglichen Wörter, die durch die Regeln der Grammatik transitiv aus dem Startsymbol S abgeleitet werden können:

$$L(G) := \{w \in T^* | S \overset{*}{\Rightarrow} w\}$$

Zusammenfassung

Ableitungen aus einer Grammatik definieren eine Sprache.

Eine Grammatik für boolsche Ausdrücke

Eine Grammatik für boolesche Terme ist $G_{\text{Logik}} = (V, T, R, S)$ mit

$$\begin{array}{rcl}
V & = & \{\text{Term}, \text{Literal}, \text{Variable}\} \\
T & = & \Sigma_{\text{Logik}} = \{\vee, \wedge, \neg, (,), 0, 1, a, \ldots, z\} \\
R & = & \{r_1, r_2, r_3, r_4\}, wo \\
 & & r_1 : \text{Term} \mapsto \text{Literal}|\text{Variable}|\neg\text{Term}|(\text{Term}) \\
 & & r_2 : \text{Term} \mapsto \text{Term} \vee \text{Term}|\text{Term} \wedge \text{Term} \\
 & & r_3 : \text{Literal} \mapsto 0|1 \\
 & & r_4 : \text{Variable} \mapsto a|\ldots|z \\
S & = & \text{Term}
\end{array}$$

Bemerkung

$r2 : \text{Term} \mapsto \text{Term} \vee \text{Term}|\text{Term} \wedge \text{Term}$ ist zu interpretieren als:

$$\begin{cases} r2.1 : \text{Term} \mapsto \text{Term} \vee \text{Term} \\ r2.2 : \text{Term} \mapsto \text{Term} \wedge \text{Term} \end{cases}$$

Eine Ableitung des Terms $S \overset{*}{\Rightarrow} (a \wedge b) \vee c \in L(G_{\text{Logik}})$ kann dann so ablaufen:

Regel					
				S = Term	
r2.1				↧	
		Term		∨	Term
r1.4,r1.2		↧			↧
		(Term)		∨	Variable
r2.2,r4		↧			↧
	(Term	∧	Term)	∨	c
r1.2,r1.2	↧		↧		
	(Variable	∧	Variable)	∨	c
r4,r4	↧		↧		
	(a	∧	b)	∨	c

Übung

Sprache bestimmen: ersw

Bestimmen Sie die Sprache $L(G)$ für $G = (V, T, R, S)$:

$$
\begin{aligned}
V &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \\
T &= \{e, r, s, w\} \\
R &= \{r_1, r_2, r_3\}, \\
& \quad r_1 : \mathrm{A} \mapsto \mathrm{B}w|ws\mathrm{C} \\
& \quad r_2 : \mathrm{B} \mapsto \mathrm{C}r \\
& \quad r_3 : \mathrm{C} \mapsto e|s \\
S &= \mathrm{A}
\end{aligned}
$$

Übung

Sprache bestimmen: kot

Bestimmen Sie die Sprache $L(G)$ für $G = (V, T, R, S)$:

$$
\begin{aligned}
V &= \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \\
T &= \{k, o, t\} \\
R &= \{r_1, r_2, r_3, r_4\}, \\
& \quad r_1 : \mathrm{A} \mapsto \mathrm{B}t|\mathrm{C}o \\
& \quad r_2 : \mathrm{B} \mapsto \mathrm{C}t \\
& \quad r_3 : \mathrm{C} \mapsto k|o \\
& \quad r_4 : \mathrm{C}tt \mapsto o|ok \\
S &= \mathrm{A}
\end{aligned}
$$

Wenn auf der linken Seite einer Regel ein komplexer Ausdruck steht, dann erfolgt die Ersetzung für den Ausdruck als Ganzes.

D. h. Sei das aktuelle Wort $w = Ctt$, dann wird $w \overset{r_4}{\Rightarrow} \mathrm{o}|\mathrm{ok}$.

Übung

Ableitung finden: ewtiewet

Wie wird das Wort $ewtiewet$ aus der Grammatik $G = (V, T, R, S)$ abgeleitet?

$$
\begin{aligned}
V &= \{\mathrm{P}, \mathrm{Q}, \mathrm{R}, \mathrm{S}\} \\
T &= \{e, i, t, w\} \\
R &= \{r_1, r_2, r_3, r_4, r_5\}, \\
& \quad r_1 : \mathrm{P} \mapsto i|w\mathrm{Q} \\
& \quad r_2 : \mathrm{Q} \mapsto et|we|wit \\
& \quad r_3 : \mathrm{R} \mapsto \mathrm{Q}wt|tie\mathrm{P} \\
& \quad r_4 : \mathrm{S} \mapsto \mathrm{P}e|ew\mathrm{R}|i|w\mathrm{Q}we \\
& \quad r_5 : wtie\mathrm{P} \mapsto wtietie \\
S &= \mathrm{S}
\end{aligned}
$$

Übung

Ableitung finden: etrrtse

Wie wird das Wort $etrrtse$ aus der Grammatik $G = (V, T, R, S)$ abgeleitet?

$$\begin{aligned}
V &= \{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\} \\
T &= \{e, r, s, t\} \\
R &= \{r_1, r_2, r_3\}, \\
&\quad r_1 : \mathrm{X} \mapsto rts \\
&\quad r_2 : \mathrm{Y} \mapsto et\mathrm{Z} | re\mathrm{X} \\
&\quad r_3 : \mathrm{Z} \mapsto r\mathrm{X}e | srt | tse \\
S &= \mathrm{Y}
\end{aligned}$$

Grammatiken für die vorhergehenden Beispiele

$M_3 = \{0\} \cup \{1\} \times \{0,1\}^* = \{0, 1, 10, 11, 100, 101, 110, 111, \dots\} = L(G)$:

$$
\begin{aligned}
G &= (V, T, R, S)\\
V &= \{\text{Start}, \text{A}\}\\
T &= \{0, 1\}\\
R &= \{r_1, r_2\},\\
& \quad r_1 : \text{Start} \mapsto 0|1|1\text{A}\\
& \quad r_2 : \text{A} \mapsto 0|1|0\text{A}|1\text{A}\\
S &= \text{Start}
\end{aligned}
$$

$M_2 = \{0^n 1^n | n \in \mathbb{N}\} = \{01, 0011, 000111, \dots\} = L(G)$:

$$
\begin{aligned}
G &= (V, T, R, S)\\
V &= \{\text{S}\}\\
T &= \{0, 1\}\\
R &= \{r_1\},\\
& \quad r_1 : \text{S} \mapsto 0\,\text{S}\,1|01\\
S &= \text{S}
\end{aligned}
$$

$M_1 = \{0^n 1^n 2^n | n \in \mathbb{N}\} = \{012, 001122, 000111222, \dots\} = L(G)$:

$$
\begin{aligned}
G &= (V, T, R, S)\\
V &= \{\text{S}, \text{B}, \text{C}\}\\
T &= \{0, 1, 2\}\\
R = \{r_1, r_2, r_3, r_4, r_5, r_6\} \quad &, \quad r_1 : \text{S} \mapsto 0\text{SBC}|0\text{BC}\\
& \quad r_2 : \text{CB} \mapsto \text{BC}\\
& \quad r_3 : 0\text{B} \mapsto 01\\
& \quad r_3 : 1\text{B} \mapsto 11\\
& \quad r_3 : 1\text{C} \mapsto 12\\
& \quad r_3 : 2\text{C} \mapsto 22\\
S &= \text{S}
\end{aligned}
$$

Chomsky-Hierarchie

Aufbau der Chomsky-Hierarchie

Definition

Unterteilung der formalen Grammatiken $G = (V, T, R, S)$ in vier Klassen:

0

Typ-0: In einer allgemeinen Chomsky-Grammatik oder Typ-0 Grammatik sind alle Regeln zugelassen.

$r_k : (V \cup T)^* \setminus T^* \to (V \cup T)^*$

1

Typ-1: In einer kontextsensitiven Grammatik oder Typ-1 Grammatik müssen die Regeln Prefix und Postfix vor und nach der Ersetzung erhalten, und die Länge des Wortes erhalten oder wachsen lassen, also

$r_k : uAv \mapsto uwv$ mit $u, v \in (V \cup T)^*$, $A \in V$ und $w \in (V \cup T)^+$.

Einmalig ist die Regel $S \mapsto \varepsilon$ erlaubt, dann darf aber S auf keiner rechten Seite einer anderen Regel auftreten.

2

Typ-2: In einer kontextfreien Grammatik oder Typ-2 Grammatik dürfen Regeln links nur aus einer Variablen bestehen, also

$rk : A \mapsto w$ mit $A \in V$ und $w \in (V \cup T)^+$.

Einmalig ist die Regel $S \mapsto \varepsilon$ erlaubt, dann darf aber S auf keiner rechten Seite einer anderen Regel auftreten.

3

Typ-3: In einer regulären Grammatik oder Typ-3 Grammatik dürfen Regeln links nur aus einer Variablen bestehen, und auf der rechten Seite aus einem terminalen Symbol und optional einer Variable, die bei allen Regeln nur links für *links-lineare Grammatiken* oder nur rechts für *rechts-lineare Grammatiken* stehen darf:

$rk : A \mapsto aB$ (rechts-linear) oder $A \mapsto Ba$ (links-linear) oder $A \mapsto a$ mit $A, B \in V$, $a \in T$.

Einmalig ist die Regel $S \mapsto \varepsilon$ erlaubt, dann darf aber S auf keiner rechten Seite einer anderen Regel auftreten.

Chomsky-Typ einer Sprache

Beobachtung

Regeln von Grammatiken mit höherem Typ erfüllen immer auch „tiefere" Bedingungen.

Eine relevante Frage ist: Welches ist der höchste Grammatik-Typ einer erzeugten Sprache?

Definition

Eine formale Sprache L ist von einem bestimmten *Chomsky-Typ* und entsprechend kontextsensitiv, kontextfrei oder regulär, wenn es eine Grammatik G gibt, die die Sprache $L = L(G)$ erzeugt.

Zusammenfassung

Da Sprachen höheren Typs auch die Kriterien tieferen Typs erfüllen, sind somit reguläre Sprachen auch kontextfrei, sowie kontextfreie Sprachen auch kontextsensitiv.

Einordnung von Grammatiken in die Chomsky-Hierarchie

Frage

Welchen Typ hat die folgende Grammatik $G = (V, T, R, S)$?

$$\begin{array}{rcl} V & = & \{\mathrm{Start}, \mathrm{A}\} \\ T & = & \{0, 1\} \\ R & = & \{r_1, r_2\}, \\ & & r_1 : \mathrm{Start} \mapsto 0|1|1\mathrm{A} \\ & & r_2 : \mathrm{A} \mapsto 0|1|0\mathrm{A}|1\mathrm{A} \\ S & = & \mathrm{Start} \end{array}$$

Frage

Welchen Typ hat die folgende Grammatik $G = (V, T, R, S)$?

$$\begin{array}{rcl} V & = & \{\mathrm{S}\} \\ T & = & \{0, 1\} \\ R & = & \{r_1\}, \\ & & r_1 : \mathrm{S} \mapsto 0\,\mathrm{S}\,1|01 \\ S & = & \mathrm{S} \end{array}$$

Frage

Welchen Typ hat die folgende Grammatik $G = (V, T, R, S)$?

$$\begin{array}{rcccl} V = \{\mathrm{S}, \mathrm{B}, \mathrm{C}\} & , & S = \mathrm{S} & , & T = \{0, 1, 2\} \\ R = \{r_1, r_2, r_3, r_4, r_5, r_6\} & & & , & r_1 : \mathrm{S} \mapsto 0\mathrm{SBC}|0\mathrm{BC} \\ & & & & r_2 : \mathrm{CB} \mapsto \mathrm{BC} \\ & & & & r_3 : 0\mathrm{B} \mapsto 01 \\ & & & & r_3 : 1\mathrm{B} \mapsto 11 \\ & & & & r_3 : 1\mathrm{C} \mapsto 12 \\ & & & & r_3 : 2\mathrm{C} \mapsto 22 \end{array}$$

Können wir die Grammatik umformulieren, damit dies eine Type 1 Grammatik wird?

Umformulierung einer allgemeinen Vertauschung in eine kontextsensitive Regel (der Kontext ist hierbei nicht explizit definiert kann aber natürlich ergänzt werden):

Gegeben sei die Regel $r_2 : \mathrm{CB} \mapsto \mathrm{BC}$.

Umformulierung in eine kontextsensitive Regel:

$$\begin{array}{rcl} r_{2'.1} & : & CB \mapsto CX \\ r_{2'.2} & : & CX \mapsto YX \\ r_{2'.3} & : & YX \mapsto YC \\ r_{2'.4} & : & YC \mapsto BC \end{array}$$

In jeder Regel wird nur eine Variable ersetzt!

Frage

Welchen Typ hat die folgende Grammatik $G = (V, T, R, S)$?

$$
\begin{array}{rcl}
V = \{Start, o, >, <, \#, *\} & , & T = \{0, 1, 2\} \quad , \quad S = Start \\
 & & \begin{array}{rrcl}
r_1: & Start & \mapsto & \# < o\# \\
r_2: & \# < & \mapsto & \# > |* \\
r_3: & > o & \mapsto & oo > \\
r_4: & > \# & \mapsto & < \# \\
r_5: & o < & \mapsto & < o \\
r_6: & *o & \mapsto & 0 * | 1 * | 2* \\
r_7: & *\# & \mapsto & \varepsilon
\end{array} \\
R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\} & , &
\end{array}
$$

Die Grammatik erzeugt die Sprache:

$$
\begin{aligned}
M_0 &= \{w \in \Sigma^* | |w| = 2^n, n \in \mathbb{N}\} \\
&= \{0, 1, 2, 00, 01, \ldots, 21, 22, 0000, 0001, \ldots\} \\
&= L(G)
\end{aligned}
$$

Übung

Chomsky-Typ: ikos

Bestimmen Sie den Chomsky-Typ der Grammatik $G = (V, T, R, S)$ und geben Sie eine Ableitung für das Wort $okoik$ an.

$$
\begin{array}{rcl}
V & = & \{X, Y, Z\} \\
T & = & \{i, k, o, s\} \\
R & = & \{r_1, r_2, r_3, r_4, r_5\}, \\
 & & r_1 : X \mapsto io|isk|ok \\
 & & r_2 : Xo \mapsto ikso|ko|okio|oso \\
 & & r_3 : Y \mapsto Xoik|k|o|s \\
 & & r_4 : Z \mapsto oY \\
 & & r_5 : oXo \mapsto oko|osioo \\
S & = & Z
\end{array}
$$

Übung

Chomsky-Typ: ru

Bestimmen Sie den Chomsky-Typ von $G = (V, T, R, S)$ und die Sprache $L(G)$:

$$
\begin{aligned}
V &= \{A, B, C\} \\
T &= \{r, u\} \\
R &= \{r_1, r_2, r_3, r_4, r_5\} \\
& \quad r_1 : A \mapsto uB \\
& \quad r_2 : B \mapsto r \\
& \quad r_3 : Bir \mapsto ru|u|ur \\
& \quad r_4 : C \mapsto AiB|r|rB|u \\
& \quad r_5 : riB \mapsto u \\
S &= C
\end{aligned}
$$

Übung

Chomsky-Typ: iosu

Bestimmen Sie den Chomsky-Typ von $G = (V, T, R, S)$ und die Sprache $L(G)$:

$$
\begin{aligned}
V &= \{A, B, C, D\} \\
T &= \{i, o, s, u\} \\
R &= \{r_1, r_2, r_3, r_4\} \\
& r_1 : A \mapsto Co|o \\
& r_2 : B \mapsto iCu|iDu|uA \\
& r_3 : C \mapsto is \\
& r_4 : D \mapsto usoA \\
S &= B
\end{aligned}
$$