6 FORMALE SPRACHEN

6.1 FORMALE SPRACHEN

Eine natürliche Sprache umfasst mehrere Aspekte, z. B. Aussprache und Stil, also z. B. Wortwahl und Satzbau. Dafür ist es auch notwendig zu wissen, welche Formulierungen syntaktisch korrekt sind. Neben den anderen genannten und ungenannten Punkten spielt *syntaktsiche Korrektheit* auch in der Informatik an vielen Stellen eine Rolle.

Bei der Formulierung von Programmen ist das jedem klar. Aber auch der Text, der beim Senden einer Email über das Netz transportiert wird oder der Quelltext einer HTML-Seite müssen bestimmten Anforderungen genügen. Praktisch immer, wenn ein Programm Eingaben liest, sei es aus einer Datei oder direkt vom Benutzer, müssen diese Eingaben gewissen Regeln genügen, sofern sie weiterverarbeitet werden können sollen. Wird z. B. vom Programm die Darstellung einer Zahl benötigt, dann ist vermutlich „`101`“ in Ordnung, aber „`a*&W`“ nicht. Aber natürlich (?) sind es bei jeder Anwendung andere Richtlinien, die eingehalten werden müssen.

Es ist daher nicht verwunderlich, wenn

- syntaktische Korrektheit,
- Möglichkeiten zu spezifizieren, was korrekt ist und was nicht, und
- Möglichkeiten, syntaktische Korrektheit von Texten zu überprüfen,

von großer Bedeutung in der Informatik sind.

Man definiert: Eine *formale Sprache* (über einem Alphabet A) ist eine Teilmenge $L \subseteq A^*$. *formale Sprache*

Immer, wenn es um syntaktische Korrektheit geht, bilden die syntaktisch korrekten Gebilde eine formale Sprache L, während die syntaktisch falschen Gebilde eben *nicht* zu L gehören.

Beispiele:

- Es sei $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -\}$. Die formale Sprache der Dezimaldarstellungen ganzer Zahlen enthält zum Beispiel die Wörter „`1`“, „`-22`“ und „`192837465`“, aber nicht „`2-3--41`“.
- Die formale Sprache der syntaktisch korrekten Java-Programme über dem Unicode-Alphabet enthält zum Beispiel nicht das Wort „`[2] class int)(`“ (aber eben alle Java-Programme).

6.2 OPERATIONEN AUF FORMALEN SPRACHEN

6.2.1 Produkt oder Konkatenation formaler Sprachen

Wir haben schon definiert, was die Konkatenation zweier Wörter ist. Das erweitern wir nun auf eine übliche Art und Weise auf Mengen von Wörtern: Für zwei formale Sprachen L_1 und L_2 heißt

$$L_1 \cdot L_2 = \{w_1 w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$$

Produkt von Sprachen

das *Produkt der Sprachen L_1 und L_2*.

6.1 Lemma. Für jede formale Sprache L ist

$$L \cdot \{\varepsilon\} = L = \{\varepsilon\} \cdot L\,.$$

6.2 Beweis. Einfaches Nachrechnen:

$$\begin{aligned} L \cdot \{\varepsilon\} &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in \{\varepsilon\}\} \\ &= \{w_1 w_2 \mid w_1 \in L \wedge w_2 = \varepsilon\} \\ &= \{w_1 \varepsilon \mid w_1 \in L\} \\ &= \{w_1 \mid w_1 \in L\} \\ &= L \end{aligned}$$

Analog zeigt man $L = \{\varepsilon\} \cdot L$. ∎

In Abschnitt 4.4.3 haben wir ein erstes Mal über den Aufbau von E-Mails gesprochen. Produkte formaler Sprachen könnte man nun benutzen, um folgende Festlegung zu treffen:

- Die formale Sprache L_{email} der syntaktisch korrekten E-Mails ist

$$L_{email} = L_{header} \cdot L_{leer} \cdot L_{body}$$

- Dabei sei
  - L_{header} die formale Sprache der syntaktisch korrekten E-Mail-Köpfe,
  - L_{leer} die formale Sprache, die nur die Leerzeile enthält, also $L_{leer} = \{\boxed{\texttt{CR}}\boxed{\texttt{LF}}\}$ und
  - L_{body} die formale Sprache der syntaktisch korrekten E-Mail-Rümpfe.

L_{header} und L_{body} muss man dann natürlich auch noch definieren. Eine Möglichkeit, das bequem zu machen, werden wir in einem späteren Kapitel kennenlernen.

Potenzen L^k

Wie bei Wörtern will man *Potenzen L^k* definieren. Der „Trick“ besteht darin,

für den Fall $k = 0$ etwas Sinnvolles zu finden — Lemma 6.1 gibt einen Hinweis. Die Definition geht (wer hätte es gedacht?) wieder induktiv:

$$L^0 = \{\varepsilon\}$$
$$\forall k \in \mathbb{N}_0 : \; L^{k+1} = L \cdot L^k$$

Wie auch schon bei der Konkatenation einzelner Wörter kann man auch hier wieder nachrechnen, dass z. B. gilt:

$$L^1 = L$$
$$L^2 = L \cdot L$$
$$L^3 = L \cdot L \cdot L$$

Genau genommen hätten wir in der dritten Zeile $L \cdot (L \cdot L)$ schreiben müssen. Aber Sie dürfen glauben (oder nachrechnen), dass sich die Assoziativität vom Produkt von Wörtern auf das Produkt von Sprachen überträgt.

Als einfaches Beispiel betrachte man $L = \{\mathtt{aa}, \mathtt{b}\}$. Dann ist

$$\begin{aligned}
L^0 &= \{\varepsilon\} \\
L^1 &= \{\mathtt{aa}, \mathtt{b}\} \\
L^2 &= \{\mathtt{aa}, \mathtt{b}\} \cdot \{\mathtt{aa}, \mathtt{b}\} = \{\mathtt{aa} \cdot \mathtt{aa}, \mathtt{aa} \cdot \mathtt{b}, \mathtt{b} \cdot \mathtt{aa}, \mathtt{b} \cdot \mathtt{b}\} \\
&= \{\mathtt{aaaa}, \mathtt{aab}, \mathtt{baa}, \mathtt{bb}\} \\
L^3 &= \{\mathtt{aa} \cdot \mathtt{aa} \cdot \mathtt{aa},\ \mathtt{aa} \cdot \mathtt{aa} \cdot \mathtt{b},\ \mathtt{aa} \cdot \mathtt{b} \cdot \mathtt{aa},\ \mathtt{aa} \cdot \mathtt{b} \cdot \mathtt{b}, \\
&\qquad \mathtt{b} \cdot \mathtt{aa} \cdot \mathtt{aa},\ \mathtt{b} \cdot \mathtt{aa} \cdot \mathtt{b},\ \mathtt{b} \cdot \mathtt{b} \cdot \mathtt{aa},\ \mathtt{b} \cdot \mathtt{b} \cdot \mathtt{b}\} \\
&= \{\mathtt{aaaaaa}, \mathtt{aaaab}, \mathtt{aabaa}, \mathtt{aabb}, \mathtt{baaaa}, \mathtt{baab}, \mathtt{bbaa}, \mathtt{bbb}\}
\end{aligned}$$

In diesem Beispiel ist L endlich. Man beachte aber, dass die Potenzen auch definiert sind, wenn L unendlich ist. Betrachten wir etwa den Fall

$$L = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}_+\} \,,$$

es ist also (angedeutet)

$$L = \{\mathtt{ab}, \mathtt{aabb}, \mathtt{aaabbb}, \mathtt{aaaabbbb}, \ldots\} \,.$$

Welche Wörter sind in L^2? Die Definion besagt, dass man alle Produkte $w_1 w_2$ von Wörtern $w_1 \in L$ und $w_2 \in L$ bilden muss. Man erhält also (erst mal ungenau

hingeschrieben)

$$\begin{aligned} L^2 = \quad & \{\mathtt{ab}\cdot\mathtt{ab}, \mathtt{ab}\cdot\mathtt{aabb}, \mathtt{ab}\cdot\mathtt{aaabbb}, \dots\} \\ \cup\ & \{\mathtt{aabb}\cdot\mathtt{ab}, \mathtt{aabb}\cdot\mathtt{aabb}, \mathtt{aabb}\cdot\mathtt{aaabbb}, \dots\} \\ \cup\ & \{\mathtt{aaabbb}\cdot\mathtt{ab}, \mathtt{aaabbb}\cdot\mathtt{aabb}, \mathtt{aaabbb}\cdot\mathtt{aaabbb}, \dots\} \\ & \vdots \end{aligned}$$

Mit anderen Worten ist

$$L^2 = \{\mathtt{a}^{n_1}\mathtt{b}^{n_1}\mathtt{a}^{n_2}\mathtt{b}^{n_2} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+\} .$$

Man beachte, dass bei die Exponenten n_1 „vorne“ und n_2 „hinten“ verschieden sein dürfen.

Für ein Alphabet A und für $i \in \mathbb{N}_0$ hatten wir auch die Potenzen A^i definiert. Und man kann jedes Alphabet ja auch als eine formale Sprache auffassen, die genau alle Wörter der Länge 1 über A enthält. Machen Sie sich klar, dass die beiden Definitionen für Potenzen konsistent sind, d. h. A^i ergibt immer die gleiche formale Sprache, egal, welche Definition man zu Grunde legt.

6.2.2 Konkatenationsabschluss einer formalen Sprache

Bei Alphabeten hatten wir neben den A^i auch noch A^* definiert und darauf hingewiesen, dass für ein Alphabet A gilt:

$$A^* = \bigcup_{i=0}^{\infty} A^i .$$

Konkatenationsabschluss L^* von L
ε-freier Konkatenationsabschluss L^+ von L

Das nehmen wir zum Anlass nun den *Konkatenationsabschluss L^* von L* und den *ε-freien Konkatenationsabschluss L^+ von L* definieren:

$$L^+ = \bigcup_{i=1}^{\infty} L^i \qquad \text{und} \qquad L^* = \bigcup_{i=0}^{\infty} L^i$$

Wie man sieht, ist $L^* = L^0 \cup L^+$. In L^* sind also alle Wörter, die sich als Produkt einer beliebigen Zahl (einschließlich 0) von Wörtern schreiben lassen, die alle Element von L sind.

Als Beispiel betrachten wieder $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}_+\}$. Weiter vorne hatten wir schon gesehen:

$$L^2 = \{\mathtt{a}^{n_1}\mathtt{b}^{n_1}\mathtt{a}^{n_2}\mathtt{b}^{n_2} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+\} .$$

Analog ist

$$L^3 = \{\mathtt{a}^{n_1}\mathtt{b}^{n_1}\mathtt{a}^{n_2}\mathtt{b}^{n_2}\mathtt{a}^{n_3}\mathtt{b}^{n_3} \mid n_1 \in \mathbb{N}_+ \wedge n_2 \in \mathbb{N}_+ \wedge n_3 \in \mathbb{N}_+\} .$$

Wenn wir uns erlauben, Pünktchen zu schreiben, dann ist allgemein

$$L^i = \{\mathtt{a}^{n_1}\mathtt{b}^{n_1} \cdots \mathtt{a}^{n_i}\mathtt{b}^{n_i} \mid n_1 \ldots, n_i \in \mathbb{N}_+\} .$$

Und für L^+ könnte man vielleicht notieren:

$$L^+ = \{\mathtt{a}^{n_1}\mathtt{b}^{n_1} \cdots \mathtt{a}^{n_i}\mathtt{b}^{n_i} \mid i \in \mathbb{N}_+ \wedge n_1 \ldots, n_i \in \mathbb{N}_+\} .$$

Aber man merkt (hoffentlich!) an dieser Stelle doch, dass uns $^+$ und * die Möglichkeit geben, etwas erstens präzise und zweitens auch noch kürzer zu notieren, als wir es sonst könnten.

Zum Abschluss wollen wir noch darauf hinweisen, dass die Bezeichnung ε-freier Konkatenationsabschluss für L^+ leider (etwas?) irreführend ist. Wie steht es um das leere Wort bei L^+ und L^*? Klar sollte inzwischen sein, dass für *jede* formale Sprache L gilt: $\varepsilon \in L^*$. Das ist so, weil ja $\varepsilon \in L^0 \subseteq L^*$ ist. Nun läuft zwar in der Definition von L^+ die Vereinigung der L^i nur ab $i = 1$. Es kann aber natürlich sein, dass $\varepsilon \in L$ ist. In diesem Fall ist dann aber offensichtlich $\varepsilon \in L = L^1 \subseteq L^+$. Also kann L^+ sehr wohl das leere Wort enthalten.

Außerdem sei erwähnt, dass die Definition von L^* zur Folge hat, dass gilt:

$$\{\}^* = \{\varepsilon\}$$

6.3 ZUSAMMENFASSUNG

In dieser Einheit wurden *formale Sprachen* eingeführt, ihr *Produkt* und der *Konkatenationsabschluss*.

Wir haben gesehen, dass man damit jedenfalls manche formalen Sprachen kurz und verständlich beschreiben kann. Dass diese Notationsmöglichkeiten auch in der Praxis Verwendung finden, werden wir in der nächsten Einheit sehen.

Manchmal reicht das, was wir bisher an Notationsmöglichkeiten haben, aber noch nicht. Deshalb werden wir in späteren Einheiten mächtigere Hilfsmittel kennenlernen.