

## Formale Sprachen im Alltag

### Syntax korrekt?

#### Plausibilitätstests in Online-Formularen

The image shows a typical online registration form with the following fields:

- \*Name:
- \*Vorname:
- \*Straße:  \*Hausnummer:
- \*PLZ:  \*Ort:
- Telefonnummer:
- \*Email:
- \*Passwort:  \*Passwort wiederholen:

At the bottom, there is a note: "Alle mit \* gekennzeichneten Felder sind Pflichtfelder." and a blue button labeled "Abschicken".

Abbildung 1: typisches Eingabe-Formular

Sie kennen das sicherlich aus dem ein- oder anderen Online-Formular. Nach Eingabe aller benötigten Daten erfolgt eine Art Plausibilitätsprüfung. Dabei überprüft ein Computerprogramm, ob die Nutzereingabe *syntaktisch* plausibel ist. So besteht eine deutsche Postleitzahl beispielsweise aus fünf Ziffern, bei mehr oder anderen Zeichen muss die Nutzereingabe fehlerhaft sein. Eine weitere Plausibilitätsprüfung kann bei der Email-Adresse erfolgen: diese enthält zwingend das Zeichen @ – ohne dieses kann es sich bei einer Nutzereingabe um keine existierende Email-Adresse handeln. Man spricht in diesem Zusammenhang von *formalen Sprachen*. Die „Sprache“ der Postleitzahlen ist sehr eingeschränkt: sie besteht nur aus fünf Zeichen, die jeweils Ziffern sein müssen. Die „Sprache“ der Email-Adressen ist schon etwas komplizierter zu beschreiben. Ein erster Ansatz wäre zu sagen, dass sie aus beliebigen Zeichenfolgen besteht, die jeweils genau ein @-Zeichen enthalten.

### Aufgaben

- 1) Die formale Sprache „Geburtsdatum“ wird häufig durch TT.MM.JJJJ beschrieben. Erläutern Sie, was damit gemeint ist.
- 2) Beschreiben Sie eine mögliche „Sprache“ der Hausnummern.
- 3) Geben Sie jeweils auch mögliche „Sprachen“ der anderen Eingabefelder aus Abbildung 1 wie etwa eine mögliche „Sprache“ der Nachnamen usw. an.
- 4) Erläutern Sie Unterschiede von formalen Sprachen zu natürlichen Sprachen wie beispielsweise Ihrer Muttersprache.
- 5) Geben Sie mögliche Gründe an, warum man sich überhaupt für formale Sprachen interessiert.

## Definition „Formale Sprache“

Sie haben bereits ein paar Beispiele für formale Sprachen kennengelernt. Umgangssprachlich kann man sagen, dass eine formale Sprache eine sehr eingeschränkte Sprache ist, die nur aus ganz bestimmten Zeichenfolgen, den „Wörtern“ der Sprache besteht. Auch bei formalen Sprachen spricht man von einem Alphabet und von Wörtern. Die Definitionen lauten:

Ein **Alphabet**  $\Sigma$  ist eine endliche Menge von Zeichen bzw. Symbolen. Der griechische Buchstabe  $\Sigma$  (gesprochen „Sigma“) steht dabei für die gesamte Menge.

**Beispiel:**  $\Sigma = \{ '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.' \}$  (Menge der Ziffern zusammen mit dem Punkt-Zeichen)

Ein **Wort**  $\omega$  (gesprochen „omega“) ist eine Aneinanderreihung beliebig vieler Zeichen des Alphabets. Man schreibt dafür  $\omega \in \Sigma^*$ .

**Beispiel:**  $10.07.1980 \in \Sigma^*$

Eine **formale Sprache**  $L$  ist eine Teilmenge aller Wörter über dem Alphabet  $\Sigma$ , man schreibt hierfür  $L \subset \Sigma^*$ .

**Beispiel:** Die Menge aller möglichen Datumsangaben. So gehört beispielsweise 3.114.224.14 nicht zur Sprache, 10.07.1980 dagegen schon.

## Unterschied Syntax zu Semantik

Eine formale Sprache sagt nur etwas über die Syntax der Sprache aus. Das bedeutet, es geht nur darum, aus welchen Zeichen die einzelnen Wörter der Sprache aufgebaut sind, aber nicht um ihre Bedeutung, der sogenannten Semantik. Den Unterschied kann man sich mit der Plausibilitätsprüfung des Online-Formulars verdeutlichen: Interessiert man sich für die Syntax möglicher E-Mail-Adressen, so schaut man beispielsweise, ob die Eingaben das @-Zeichen enthalten. Interessiert man sich stattdessen für die Semantik, also die Bedeutung einer Eingabe, würde man möglicherweise zusätzlich eine Mail an die angegebene Adresse schicken und auf eine Bestätigung dieser warten.

## Beispiele für formale Sprachen

Textbasierte Programmiersprachen stellen sicherlich ein Standard-Beispiel für formale Sprachen dar. Nur wenn ein Quelltext den Regeln der jeweiligen Programmiersprache genügt, also zur durch die Programmiersprache vorgegebenen formalen Sprache gehört, kann dieser weiterverarbeitet und interpretiert werden. Ähnlich verhält es sich mit Taschenrechnereingaben – auch hier werden diese nur ausgewertet, wenn die Syntax korrekt verwendet wurde, die Taschenrechnereingaben also zur formalen Sprache der gültigen Rechenausdrücke gehören.

Formale Sprachen begegnen uns aber auch sehr häufig im Alltag, ohne dass wir sie bewusst wahrnehmen. Sie stellen beispielsweise eine einheitliche oder standardisierte Schreibweise für Sachverhalte dar oder sind so aufgebaut, dass sie leichter von Maschinen ausgewertet werden.

**Aufgaben:**

- 1) Überlegen Sie sich selbst ein paar Beispiele, bevor Sie weiterlesen.



- 2) Kennengelernt haben wir bereits die durch das Formular in Abbildung 1 definierten Sprachen wie die der Postleitzahlen oder Email-Adressen. Im Folgenden werden weitere Beispiele für formale Sprachen im Alltag aufgelistet. Versuchen Sie jeweils die zugehörige formale Sprache mit eigenen Worten zu beschreiben.
- a. Kfz-Kennzeichen
  - b. Twitter-Nutzernamen
  - c. Hashtags
  - d. Kleidergrößen wie beispielsweise L, XS, M,...
  - e. Lebensmittel-Kennzeichnungen wie beispielsweise bei Eiern
  - f. Raumnummern, beispielsweise Ihrer Schule
  - g. gültige Dateinamen (oder eingeschränkter: gültige Bilddateien)
  - h. Dezimalzahlen
  - i. Barcodes
  - j. IBAN
  - k. ...
- 3) Schreiben Sie ein Programm, welches erkennt, ob eine Zeichenfolge das Zeichen @ enthält.
- 4) Ein Programm soll erkennen, ob eine Zeichenfolge zur formalen Sprache der Kfz-Kennzeichen gehört. Diskutieren Sie mit einem Partner, was eine Implementierung hierzu alles berücksichtigen muss.

### Ausblick: Möglichkeiten zur Beschreibung formaler Sprachen

Vielleicht haben Sie bei der Bearbeitung obiger Aufgaben bereits gemerkt, dass verbale Formulierungen zur Beschreibung formaler Sprachen schnell an ihre Grenzen stoßen. Daher werden in der Praxis häufig andere Darstellungsarten verwendet. Es gibt verschiedene Klassen von formalen Sprachen. Diese können jeweils durch sogenannte Grammatiken beschrieben werden. Einige dieser Sprachen (die sogenannten regulären Sprachen) können auch durch endliche Automaten beschrieben werden. Im weiteren Verlauf werden Sie all diese Darstellungsarten und unterschiedlichen Sprachen kennenlernen. Auch werden Sie lernen, wie ein endlicher Automat in ein Programm umgesetzt werden kann, das erkennt, ob eine Eingabe zur jeweiligen formalen Sprache gehört oder nicht.

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

