

Informatik Q3 Abels



Formale Sprachen

Natürliche Sprachen

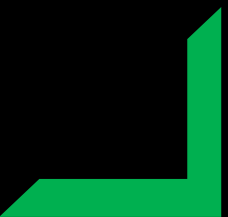
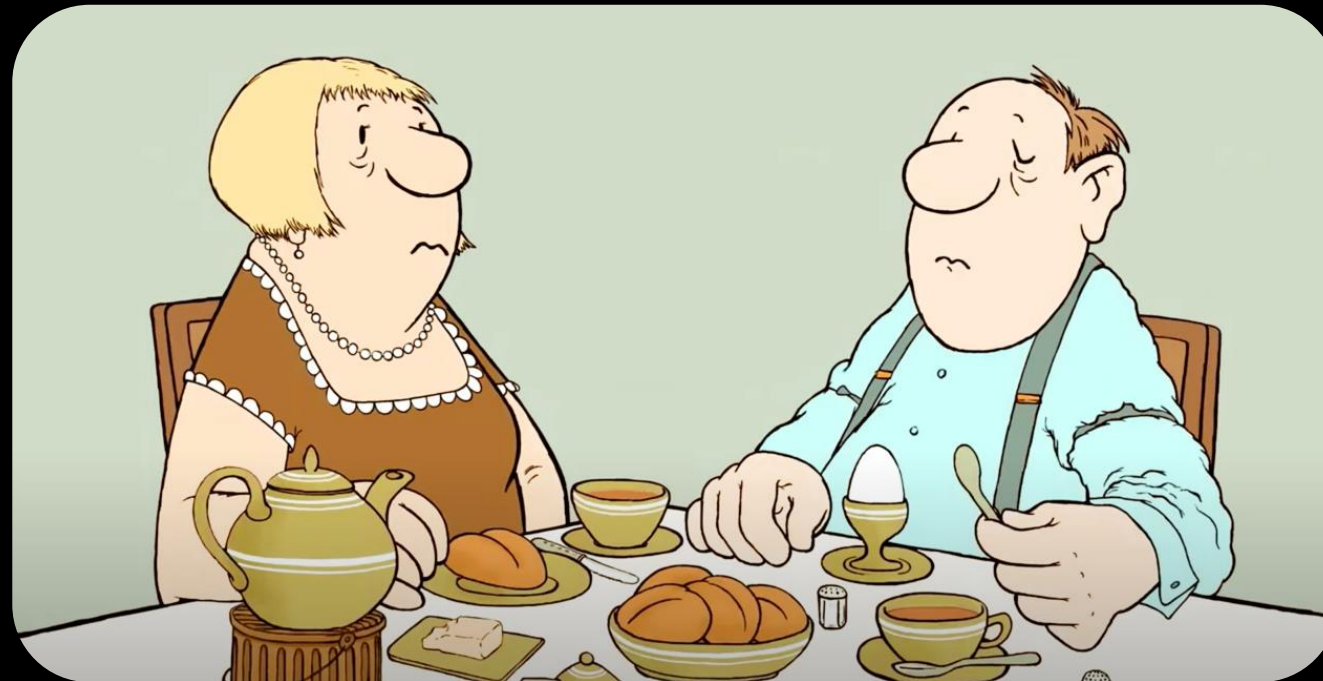
Das Frühstücksei (von Lorient)





Übung 1

- Worin liegt das Problem bei der Kommunikation des Ehepaares?
- Welche Anforderungen sollten an eine Sprache gestellt werden, die dazu verwendet wird, mit Maschinen zu kommunizieren?



Formale Sprachen

Beispiel: "ABC"



Code.abc

```
X: 1
T: Brother John
C: Traditional
L: 1/4
K: C
CDEC | CDEC | EFGz |\ % continues
w: Are you slee-ping, Are you slee-ping, Bro-ther John!\ % continues
EFGz |
w: Bro-ther John!
G/ A/ G/ F/ EC | G/ A/ G/ F/ EC |\ % continues
w: Mor-ning bells are rin-ging, Mor-ning bells are rin-ging,\
FHEz | FHEz|]
w: ding ding dong, ding ding dong!
```

Formale Sprachen

Beispiel: "ABC"



Code.abc

```
X: 1
T: Brother John
C: Traditional
L: 1/4
K: C
CDEC | CDEC | EFGz |\ % continues
w: Are you slee-ping, Are you slee-ping, Bro-ther John!\ % continues
EFGz |
w: Bro-ther John!
G/ A/ G/ F/ EC | G/ A/ G/ F/ EC |\ % continues
w: Mor-ning bells are rin-ging, Mor-ning bells are rin-ging,\
FHEz | FHEz|]
w: ding ding dong, ding ding dong!
```



The image shows two staves of musical notation. The first staff is in treble clef with a key signature of one sharp (F#) and a 4/4 time signature. It contains the melody for the first line of the song, with lyrics underneath: "Are you slee - ping, Are you slee - ping, Bro - ther John! Bro - ther John!". The second staff is also in treble clef with the same key signature and time signature. It contains the melody for the second line of the song, with lyrics underneath: "Mor-ning bells are rin-ging, Mor-ning bells are rin-ging, ding ding dong, ding ding dong!".

Formale Sprachen

Beispiel: "Portable Graymap"



Code.pgm

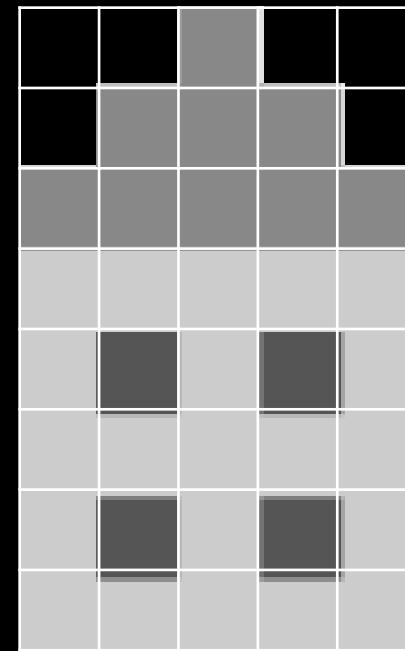
```
# Haus
P2
5 8
15
15 15 8 15 15
15 8 8 8 15
8 8 8 8 8
12 12 12 12 12
12 5 12 5 12
12 12 12 12 12
12 5 12 5 12
12 12 12 12 12
```

Formale Sprachen

Beispiel: "Portable Graymap"

```
Code.pgm

# Haus
P2
5 8
15
15 15 8 15 15
15 8 8 8 15
8 8 8 8 8
12 12 12 12 12
12 5 12 5 12
12 12 12 12 12
12 5 12 5 12
12 12 12 12 12
```



Formale Sprachen

Ein Alphabet Σ ist eine endliche Menge von Symbolen.

Ein Wort $\omega \in \Sigma^*$ ist eine beliebige Kombination von Zeichen des Alphabets.

Das leere Wort wird mit ϵ bezeichnet.

Eine formale Sprache L über Σ ist eine Teilmenge von Worten $L \subseteq \Sigma^*$.

Formale Sprachen

Beispiel: "Alphabetisch geordnete a's und b's"

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\epsilon, a, b, \overset{\omega_0}{aa}, \overset{\omega_1}{ab}, \overset{\omega_2}{ba}, \overset{\omega_3}{bb}, \dots, \overset{\omega_0}{aaa}, \overset{\omega_1}{aab}, \overset{\omega_2}{aba}, \overset{\omega_3}{baa}, \overset{\omega_4}{abb}, \overset{\omega_5}{bab}, \overset{\omega_6}{bba}, \overset{\omega_7}{bbb}, \overset{\omega_8}{aaaa}, \dots\}$$

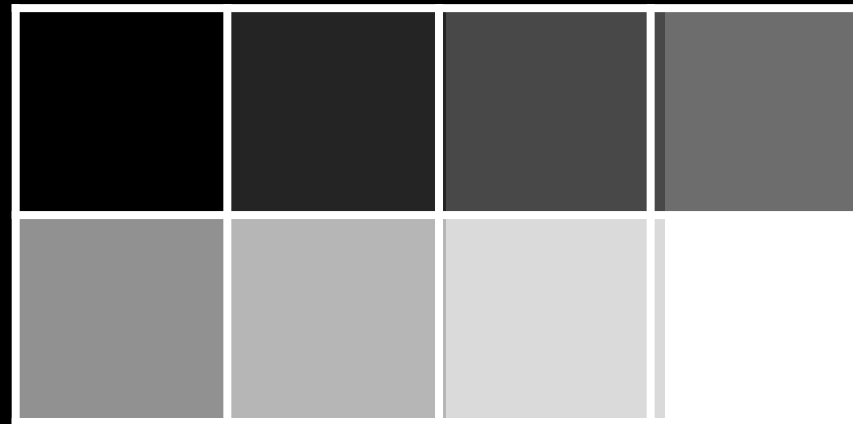
$$\mathbf{L} = \{\epsilon, a, b, aa, ab, bb, aaa, aab, abb, bbb, aaaa, \dots\}$$



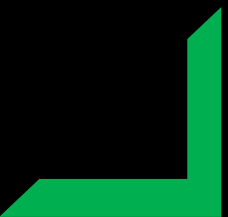
Übung 2

Gegeben ist folgendes Graustufenbild.

- Übersetze das Bild in die Sprache "Portable Graymap".
- Gib das Alphabet der Sprache an.
- Nenne je drei Wörter, die der Sprache an- bzw. nicht angehören.



Formuliere zwei eigene Sprachen inklusive Alphabet.





Übung 2

$\Sigma = \{P2, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\omega_1 = "P21100" \in L$

$\omega_2 = "P221101" \in L$

$\omega_3 = "P2235321441" \in L$

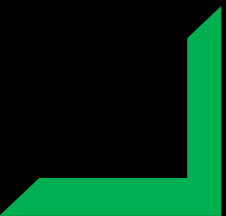
$\omega_4 = "P11100" \notin L$

$\omega_5 = "P211000" \notin L$

$\omega_6 = "P2213014" \notin L$



```
Code.pgm
P2
4 2
7
0 1 2 3
4 5 6 7
```





Übung 2



Beispiel: "Palindrome"

$$\Sigma = \{a, b, c, \dots, x, y, z\}$$

$$L = \{\epsilon, abba, otto, reittier, \dots\}$$

Beispiel: "Primzahlen"

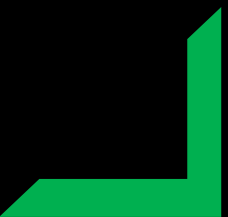
$$\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$L = \{\epsilon, 2, 3, 5, 7, 11, 13, \dots\}$$

Beispiel: "Klammersprache (Dyck-Sprache D_2)"

$$\Sigma = \{(,), [,]\}$$

$$L = \{\epsilon, (), [(), ()[(), \dots\}$$



Formale Sprachen

Bei der Kommunikation mit natürlichen Sprachen können Mehrdeutigkeiten und Missverständnisse auftreten.

Dies kann man durch die Verwendung einer formalen Sprache verhindern, die als Menge zulässiger Zeichenketten über einem Alphabet definiert wird.

Die Regeln zur Bildung zulässiger Zeichenketten stellen die **Syntax** einer Sprache dar.

Die Bedeutung eines Wortes nennt man **Semantik**.

Beachte: Die Semantik ist unabhängig von der Syntax, sie muss separat festgelegt werden!



Übung 3



Code1.java

```
public void SageHallo (String name) {  
    System.out.println("Hallo " +name);  
}
```



Code2.java

```
public void SageHallo [name] (  
    System.out.println("Hallo " +name);  
)
```



Code3.html

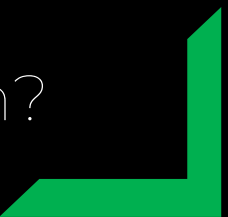
```
<h1> Gedicht </h2>  
<p> Rosen sind rot, </p>  
<p> Veilchen sind blau, </p>  
<p> Aber ohne <b>Kaffee</p>, </b>  
<li> Wäre alles ganz grau. </p>
```



Code4.html

```
<h1> Gedicht </h1>  
<p> Rosen sind rot, </p>  
<p> Veilchen sind blau, </p>  
<p> Aber ohne <b>Kaffee</b>, </p>  
<p> Wäre alles ganz grau. </p>
```

- Welche Codeausschnitte sind syntaktisch korrekt?
- Sind Java und HTML Formale Sprachen?
- Welche Semantik lässt sich aus dem korrekten Java-Code ablesen?





Übung 3



Code1.java

```
public void SageHallo (String name) {  
    System.out.println("Hallo " +name);  
}
```



Code2.java

```
public void SageHallo ([name] (  
    System.out.println("Hallo " +name);  
)
```



Code3.html

```
<h1> Gedicht </h2>  
<p> Rosen sind rot, </p>  
<p> Veilchen sind blau, </p>  
<p> Aber ohne <b>Kaffee</p>, </b>  
<li> Wäre alles ganz grau. </p>
```



Code4.html

```
<h1> Gedicht </h1>  
<p> Rosen sind rot, </p>  
<p> Veilchen sind blau, </p>  
<p> Aber ohne <b>Kaffee</b>, </p>  
<p> Wäre alles ganz grau. </p>
```





Übung 3

Sind höhere Programmiersprachen wie Java oder html formale Sprachen?



Ja, höhere Programmiersprachen wie Java und HTML sind formale Sprachen. Eine formale Sprache ist eine Sprache, die durch präzise und formale Regeln definiert wird, die bestimmen, welche Zeichenfolgen (oder Sätze) zur Sprache gehören.

Hier ist eine kurze Erklärung, warum Java und HTML formale Sprachen sind:

1. Java:

- **Syntax:** Java hat eine formale Syntax, die durch die Java Language Specification definiert ist. Diese Spezifikation legt genau fest, welche Programme syntaktisch korrekt sind.
- **Semantik:** Die Bedeutung (Semantik) der syntaktisch korrekten Programme ist ebenfalls formal definiert, sodass klar ist, wie Programme interpretiert und ausgeführt werden.
- **Compiler/Interpreter:** Java-Code wird von Compilern verarbeitet, die strikt die formalen Regeln befolgen, um den Quellcode in Bytecode zu übersetzen, der dann von der Java Virtual Machine (JVM) ausgeführt wird.

2. HTML:

- **Syntax:** HTML hat eine formale Syntax, die durch Spezifikationen wie HTML5 definiert wird. Diese legen fest, welche Tags und Attribute in HTML-Dokumenten verwendet werden können und wie sie korrekt strukturiert sind.
- **Dokument-Objekt-Modell (DOM):** Die Bedeutung der HTML-Strukturen ist durch das DOM formal definiert, welches beschreibt, wie HTML-Dokumente als strukturierte Objekte interpretiert und von Browsern dargestellt werden.
- **Parser:** Webbrowser enthalten Parser, die HTML-Dokumente entsprechend den formalen Spezifikationen analysieren und darstellen.

Zusammengefasst sind sowohl Java als auch HTML formale Sprachen, weil sie durch klar definierte Regeln bestimmt sind, die sowohl ihre Syntax als auch ihre Semantik festlegen.

$\Sigma = \{if, for, class, public, \dots, (,), \dots\}$



```
Code1.java
public void SageHallo (String name) {
    System.out.println("Hallo " +name);
}
```

$L = \{ \dots, \dots \}$

$\Sigma = \{< p >, </p >, < h1 >, \dots\}$



```
Code4.html
<h1> Gedicht </h1>
<p> Rosen sind rot, </p>
<p> Veilchen sind blau, </p>
<p> Aber ohne <b>Kaffee</b>, </p>
<p> Wäre alles ganz grau. </p>
```

$L = \{ \dots, \dots \}$



Übung 3



Welche Semantik lässt sich aus dem folgenden Code ablesen?

```
public void SageHallo (String name){System.out.println( "Hallo "
+name);}
```



Der gegebene Code ist ein Java-Methodendefinition. Um die Semantik zu verstehen, schauen wir uns die verschiedenen Teile des Codes an:

```
java Code kopieren
public void SageHallo (String name){
    System.out.println("Hallo " + name);
}
```

Detaillierte Analyse:

1. Methodensignatur:

- **public:** Dies ist ein Zugriffsmodifizierer, der angibt, dass die Methode von überall aus aufgerufen werden kann.
- **void:** Der Rückgabebetyp der Methode. In diesem Fall gibt die Methode keinen Wert zurück.
- **SageHallo:** Der Name der Methode. Methoden werden in Java üblicherweise in camelCase benannt, also wäre `sageHallo` üblicher.
- **(String name):** Dies sind die Parameter der Methode. Die Methode nimmt einen einzelnen Parameter vom Typ `String` mit dem Namen `name` entgegen.

2. Methodenkörper:

- **System.out.println("Hallo " + name);:** Dies ist der Anweisungsblock der Methode.
 - `System.out` ist ein Ausgabestream, der üblicherweise die Konsole darstellt.
 - `.println` ist eine Methode von `PrintStream`, die die übergebene Zeichenkette ausgibt und anschließend einen Zeilenumbruch einfügt.
 - `"Hallo " + name` ist eine Zeichenkette, die mit dem Wert von `name` verkettet wird. Wenn beispielsweise `name` den Wert "Welt" hat, ergibt der Ausdruck `"Hallo Welt"`.

Semantische Bedeutung:

Die Methode `sageHallo` nimmt einen String-Parameter `name` entgegen und gibt eine Begrüßung in der Konsole aus, die "Hallo " gefolgt vom Wert des Parameters `name` enthält.

Beispiel:

Wenn die Methode mit `sageHallo("Welt")` aufgerufen wird, wird folgende Ausgabe in der Konsole erscheinen:

```
Code kopieren
```

```
Hallo Welt
```

Zusammenfassung der Semantik:

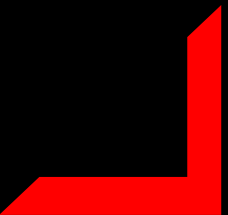
- Die Methode `sageHallo` druckt eine personalisierte Begrüßung auf die Konsole.
- Sie akzeptiert einen Parameter vom Typ `String`, der den Namen der Person repräsentiert, die begrüßt wird.
- Die Ausgabe besteht aus dem Wort "Hallo" gefolgt von einem Leerzeichen und dem Wert des übergebenen Namens.

Dies ist eine einfache, aber typische Anwendung von Methoden in Java zur Veranschaulichung der grundlegenden Syntax und Semantik.



Tagebucheintrag

Formale Sprachen





Wochenübung

In Ipogesen hört man im Mathematikunterricht der 1. Klasse ständig folgende Wörter:

ipigisi, isipigisisi, ipisigisisi, isisipigisisisi, isipisigisisisi, ipisisigisisisi, ...

Nachdem der Schüler Ipo das Wort **isipisisigisisisisisi** sagt, gibt es ein Aufmurren.

- Wie lautet das Alphabet der “ipigisi-Sprache”?
- Wie sind die Wörter der “ipigisi-Sprache” aufgebaut?
- Was bedeuten die Wörter der “ipigisi-Sprache”?
Tipp: Es handelt sich um Additionsaufgaben der Form $1+1=2$.
- Es wäre auch eine Deutung als Subtraktionsaufgaben möglich (**p** = “ist gleich” und **g** = “subtrahiert von”).
Erläutere an diesem Beispiel den Unterschied zwischen Syntax und Semantik.