

Begriff „Grammatik“ (aus dem Duden Informatik)

Jede Sprache ist nach bestimmten Regeln aufgebaut. Die zulässige Form der Wörter und Sätze einer Sprache nennt man **Syntax**, die Bedeutung wird durch die **Semantik** beschrieben.

Zur Festlegung der Syntax einer Sprache verwendet man Grammatiken. Eine Grammatik ist eine Menge von Regeln, die bestimmen, welche Sätze zu einer Sprache gehören und welche nicht. Eine Grammatik ist eine Möglichkeit, eine formale Sprachen zu beschreiben – und damit eine Alternative z.B. zu endlichen Automaten.

Beispielgrammatik

Die „Grammatik einfacher deutscher Sätze“ legt fest, dass Sätze syntaktisch korrekt sind, wenn sie nach folgenden Regeln aufgebaut sind (das Zeichen → ist zu lesen als „wird ersetzt durch“):

- (1) <Satz> → <Subjekt> <Prädikat> <Objekt>
- (2) <Subjekt> → <Artikel> <Substantiv>
- (3) <Objekt> → <Artikel> <Substantiv>

In unserem sehr vereinfachten Beispiel besteht die deutsche Sprache nur aus folgenden Wörtern: der, die, das, Hund, Katze, Maus, jagt, verfolgt, erschreckt, beißt. Daher müssen noch folgende Regeln ergänzt werden (der senkrechte Strich | bedeutet „oder“):

- (4) <Artikel> → der | die | das
- (5) <Substantiv> → Hund | Katze | Maus
- (6) <Prädikat> → beißt | verfolgt | erschreckt | jagt | sieht

Die Wörter in den Spitzklammern < > werden **Nichtterminalsymbole** genannt (Hilfszeichen), die anderen Wörter heißen **Terminalsymbole**. Jeder Satz der Sprache ist eine Folge von Terminalsymbolen. Die Nichtterminalsymbole gehören nicht zur Sprache.

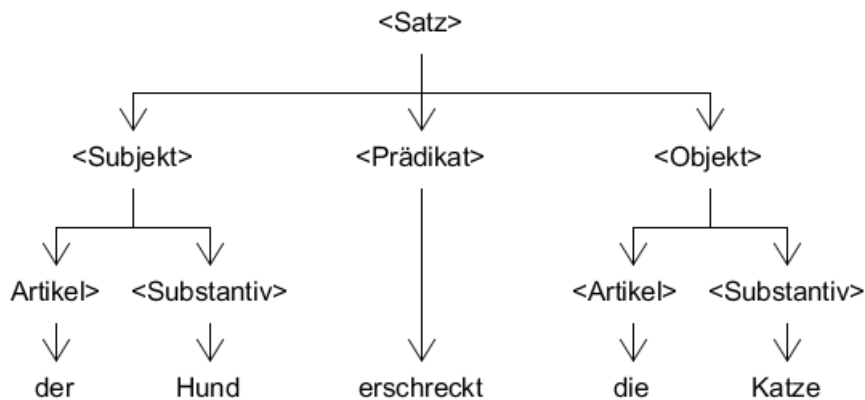
Ableitung eines Wortes

Durch Ableitung lassen sich Sätze bilden, deren Syntax korrekt ist (auch wenn die Semantik nicht immer sinnvoll ist). Dabei wird in jedem Ableitungsschritt das am weitesten links stehende **Nichtterminalsymbol** entsprechend der Regeln **ersetzt**.

- <Satz> → <Subjekt> <Prädikat> <Objekt>
- <Artikel> <Substantiv> <Prädikat> <Objekt>
- der <Substantiv> <Prädikat> <Objekt>
- der Hund <Prädikat> <Objekt>
- der Hund erschreckt <Objekt>
- der Hund erschreckt <Artikel> <Substantiv>
- der Hund erschreckt die <Substantiv>
- der Hund erschreckt die Katze

Ableitungsbaum

Die gleiche Ableitung lässt sich auch durch eine Baumstruktur darstellen:



Aufgabe

Gegeben sie die folgende, vereinfachte Grammatik der Programmiersprache Java.

Hinweis: ϵ („Epsilon“) bedeutet „nichts“, man kann also z.B. \langle Variablenliste \rangle durch nichts ersetzen.

- | | | | |
|------|------------------------------------|---------------|--|
| (1) | \langle Klasse \rangle | \rightarrow | class \langle Bezeichner \rangle { \langle Variablenliste \rangle \langle Methodenliste \rangle } |
| (2) | \langle Variablenliste \rangle | \rightarrow | \langle Variable \rangle ; \langle Variablenliste \rangle ϵ |
| (3) | \langle Methodenliste \rangle | \rightarrow | \langle Methode \rangle \langle Methodenliste \rangle ϵ |
| (4) | \langle Variable \rangle | \rightarrow | \langle Datentyp \rangle \langle Bezeichner \rangle |
| (5) | \langle Datentyp \rangle | \rightarrow | int double char boolean |
| (5) | \langle Methode \rangle | \rightarrow | void \langle Bezeichner \rangle () { \langle Variablenliste \rangle \langle Befehlsliste \rangle } |
| (6) | \langle Befehlsliste \rangle | \rightarrow | \langle Befehl \rangle \langle Befehlsliste \rangle ϵ |
| (7) | \langle Befehl \rangle | \rightarrow | \langle Bezeichner \rangle = \langle Wert \rangle ; \langle Bezeichner \rangle () ; |
| (8) | \langle Bezeichner \rangle | \rightarrow | \langle Buchstabe \rangle \langle Zeichenkette \rangle |
| (9) | \langle Wert \rangle | \rightarrow | \langle Ziffer \rangle \langle Zahl \rangle |
| (9) | \langle Zeichenkette \rangle | \rightarrow | \langle Zeichen \rangle \langle Zeichenkette \rangle ϵ |
| (10) | \langle Zahl \rangle | \rightarrow | \langle Ziffer \rangle \langle Zahl \rangle ϵ |
| (11) | \langle Buchstabe \rangle | \rightarrow | _ a ... z A ... Z |
| (12) | \langle Ziffer \rangle | \rightarrow | 0 ... 9 |
| (13) | \langle Zeichen \rangle | \rightarrow | \langle Buchstabe \rangle \langle Ziffer \rangle |

- a) Bilde die Ableitungen für die folgenden „Wörter“ (d.h. Programme) und prüfe so, ob das jeweilige „Wort“ der Grammatik entspricht. Stelle als Ableitung oder als Ableitungsbaum dar.

```

class Mover
{
    int x; int y;
    void move()
    {
        x = 10;
        y = 20;
    }
}
  
```

```

class 123Mover
{
    int x, y;
    void move()
    {
        x = x + 10;
        y = y + 20;
    }
}
  
```

- b) Zuweisungen mit Rechnungen, z.B. $x = x + 10$; oder $a = 10 * b$; lassen sich offensichtlich nicht mit den gegebenen Regeln erzeugen. Ergänze die Grammatik um entsprechende Regeln.