

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.

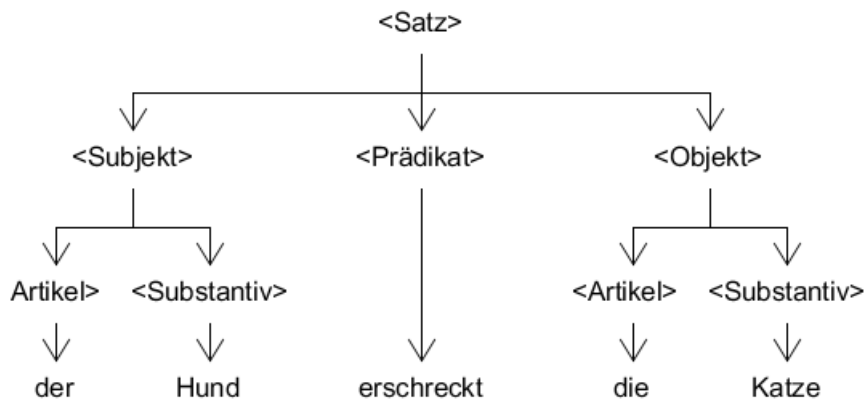
Ableitung eines Wortes

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 sei die folgende, vereinfachte Grammatik der Programmiersprache Java.

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

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

- 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 nicht mit den gegebenen Regeln erzeugen. Ergänze die Grammatik um entsprechende Regeln.