Syntaxbeschreibung mit EBNF

Die Syntax einer Sprache legt die Regeln fest, nach denen Sätze gebildet werden können. Diese Syntax wird ihrerseits in einer "Metasprache" beschrieben. Bei Programmiersprachen haben sich zwei Syntaxbeschreibungsformate durchgesetzt:

- Syntaxdiagramme (grafische Syntaxbeschreibung mit Pfeilen und Kästchen)
- Backus-Naur-Notation (textliche Syntaxbeschreibung)

Die erweiterte Backus-Naur-Notation (englisch: extended Backus-Naur-Form - EBNF) wurde von Niklaus Wirth (ETH Zürich) aus der Backus-Naur-Notation (BNF) weiterentwickelt. Die hier verwendete EBNF-Notation folgt dem ISO-Standard ISO_IEC_14977_1996E.

Als Beispiel soll die Syntax einer vereinfachten Schachzugnotation mit den Mitteln der EBNF beschrieben werden (Spezialzüge wie das Schlagen en passant und Umwandlungen sind dabei weggelassen). Ein Ausschnitt aus einem Schachspiel könnte beispielsweise so aussehen:

e2 – e4	*Bauer zieht von e2 nach e4*
0-0-0	*große Rochade*
Sb1 x c3 +	*Springer zieht von b1 nach c3, schlägt gegnerische Figur, bietet Schach*
Ta1 – a7 ++	*Turm zieht von a1 nach a7, setzt gegnerischen König matt*

Die Syntax, also die Bildungsregeln dieser Schachzüge, können wir in EBNF wie folgt durch eine Reihe sogenannter Produktionsregeln beschreiben (die in diesen Regeln verwendeten Metasymbole wie senkrechter Strich oder eckige Klammern werden weiter unten erläutert):

Schachzug = (Figurzug | Rochade) , ["+" | "++"];

Diese Regel liest man wie folgt: ein Schachzug ist ein Figurzug oder eine Rochade, optional gefolgt von einem Schachgebot ("+") oder dem Schachmatt ("++").

Figurzug = [Spielfigur] , Feld , ("-" | "x") , Feld;
Feld = Linie , Reihe;

Ein Figurzug ist eine schlagende ("x") oder nichtschlagende ("-") Bewegung einer Figur von einem Ausgangsfeld zu einem Zielfeld. Die Spielfigur wird nur dann angegeben, wenn es sich nicht um einen Bauern handelt.

Spielfigur = "K" | "D" | "T" | "L" | "S";

König, Dame, Turm, Läufer und Springer werden durch ihren jeweiligen Anfangs-buchstaben bezeichnet (je nach Sprachraum verschieden).

Linie = "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h";
Reihe = "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8";

Das quadratische Spielfeld besteht aus acht Linien und acht Reihen.

Rochade = "0-0" | "0-0-0";

Bei der kleinen bzw. großen Rochade ziehen Turm und König gleichzeitig.

Diese Syntaxbeschreibung sagt allerdings nichts darüber aus, ob die Semantik, also der Bedeutungsgehalt, des beschriebenen Schachzuges korrekt ist; der Zug "Sb1 – b2" ist beispielsweise syntaktisch korrekt, semantisch aber inkorrekt, da der Springer nicht auf ein unmittelbar benachbartes Feld springen kann. Selbst der Zug "Sb1 x c3" kann semantisch inkorrekt sein, wenn z.B. in der konkreten Spielsituation der Springer nicht auf b1 steht, oder wenn auf c3 keine gegnerische Figur steht.

Allgemein gesprochen bestehen die "Sätze" einer Sprache – im obigen Beispiel die Menge aller möglichen Schachzüge - aus einer Folge von *Symbolen.* Bei den Symbolen unterscheiden wir zwei Arten:

- *terminale Symbole*
- *nichtterminale Symbole*

Terminale Symbole sind die Grundbausteine, die "Wörter" einer Sprache. Im Deutschen gehören z.B. die bestimmten Artikel "der", "die", "das" dazu, bei Programmiersprachen reservierte Wörter wie "while" oder "switch", beim genetischen Code die Basen "Adenin", "Cytosin", "Guanin" und "Thymin". Beachte, daß Wörter zwar in der Regel aus mehreren Zeichen (Buchstaben) bestehen, aber dennoch als nicht weiter zerlegbare Einheit aufgefaßt werden, wenn es um die Regeln der Satzbildung geht.

Zu den terminalen Symbolen gehören auch die Satz- und Sonderzeichen, die wir umgangssprachlich nicht zu den Wörtern rechnen, also z.B. Ausrufe- und Fragezeichen im Deutschen, die Rochade- und Schachgebotszeichen in der Schachnotation, oder die geschweiften, eckigen und runden Klammern in der Programmiersprache C.

In Syntaxdiagrammen werden terminale Symbole durch Kreise oder Kästchen mit runden Ecken gekennzeichnet, in der EBNF wahlweise durch Einschließen des Symbols in Anführungszeichen oder Apostrophe.

Nichtterminale Symbole sind aus anderen (terminalen oder nichtterminalen) Symbolen zusammengesetzt und werden durch möglichst sinnvoll gewählte Namen bezeichnet. In Syntaxdiagrammen werden sie durch Kästchen mit rechtwinkligen Ecken gekennzeichnet, in der EBNF durch den Symbolnamen ohne Anführungszeichen. Letztlich werden alle nichtterminalen Symbole durch entsprechende Produktionsregeln auf terminale zurückgeführt.

Bei der oben gezeigten Schachnotation ist z.B. das Symbol *0-0-0* für die große Rochade ein terminales Symbol, während das Symbol *Feld* ein nichtterminales Symbol ist (das seinerseits durch die nichtterminalen Symbole *Linie* und *Reihe* definiert wird).

Um die Syntax einer Sprache beschreiben zu können, muß die Syntaxbeschreibung *Metasymbole* verwenden, die in der Sprache selbst nicht vorkommen. Syntaxdiagramme verwenden grafische Metasymbole (Pfeile, Kreise, Kästchen), die textbasierte EBNF bestimmte Sonderzeichen (wenn die Sonderzeichen in der zu beschreibenden Sprache vorkommen, werden diese als terminale Symbole in Anführungszeichen eingeschlossen)

Die nachfolgende Aufzählung skizziert die Verwendung der Metasymbole:

- = *Produktionsregel (die linke Seite wird durch die rechte Seite definiert)*
- ; *Jede EBNF-Regel wird durch ein Semikolon beendet*
- | *Entweder die Definition links oder die Definition rechts des Striches*
- , *Das Komma trennt die einzelnen Terme einer Definition*
- \- *Das Minus wird zur Bildung der Mengendifferenz verwendet*
- * *Der Stern spezifiziert die Anzahl der Wiederholungen eines primären Faktors. In diesem Zusammenhang werden auch ganze nicht-negative Zahlen als Meta-Symbole verwendet*
- " " *Terminale Symbole werden durch Anführungszeichen eingeschlossen*
- ‘‘ *Terminale Symbole können auch durch Apostrophe eingeschlossen werden*
- ( ) *Runde Klammern gruppieren Definitionslisten, um den Vorrang klar zu machen*
- [ ] *Definitionslisten in eckigen Klammern sind optional (keinmal oder genau einmal)*
- { } *Definitionslisten in geschweiften Klammern kommen gar nicht oder beliebig oft vor*
- ?? *Zwischen zwei Fragezeichen kann man umgangssprachlich frei definieren. Die so definierte Sprache sollte aber offensichtlich regulär sein. Wird meistens verwendet, um mühsame Aufzählungen einzelner Zeichen zu vermeiden.*

Eine Produktionsregel definiert das nichtterminale Symbol auf der linken Seite des Metasymbols "=" durch eine Definitionsliste auf der rechten Seite. Eine Definitionsliste ist eine Folge von, durch das Zeichen "|" getrennten, Definitionen. Die durch das Metasymbol "|" getrennten Definitionen stellen Alternativen dar.
Eine Definition ist eine, durch Kommata getrennte, Folge von Termen. Im einfachsten Fall ist ein Term ein primärer Faktor. Er kann aber auch die Differenz zweier Sprachen spezifizieren oder die exakte Anzahl von Wiederholungen eines primären Faktors beschreiben.
Ein primärer Faktor kann ein terminales oder nichtterminales Symbol sein, er kann aber auch eine Gruppierung, ein optionales Vorkommen oder eine beliebige Wiederholung ganzer Definitionslisten sein. Er kann sogar die leere Zeichenkette spezifizieren.
Leerraum (Leerzeichen, Tabulatorzeichen, Zeilenvorschub) wird in den Produktionsregeln der EBNF stets ignoriert, soweit er nicht in einem terminalen Symbol vorkommt oder Bestandteil des Bezeichners für ein Nichtterminal ist (z.B. unten im Nichtterminal `primärer Faktor`).
Die Syntax der EBNF kann man[1] mit ihren eigenen Sprachmitteln rekursiv[2] beschreiben:

```
EBNF-Syntax            = Produktionsregel , { Produktionsregel };
Produktionsregel       = nichtterminales Symbol , "=" ,
                         Definitionsliste , ";" ;
Definitionsliste       = Definition ,  { "|" , Definition  };
Definition             = Term , { "," , Term };
Term                   = Faktor , [ "-" , syntaktische Ausnahme ];
Faktor                 = [ Integer Zahl , "*" ] ,  primärer Faktor;
primärer Faktor        = leere Sequenz |
                         nichtterminales Symbol |
                         terminales Symbol |
                         "(" , Definitionsliste , ")" |
                         "[" , Definitionsliste , "]" |
                         "{" , Definitionsliste , "}";
                         "?", ? Alle Zeichen außer Fragezeichen?  ,"?";
leere Sequenz          = ;
syntaktische Ausnahme  = ? Ein Faktor, der durch einen anderen Faktor
                           ersetzt werden könnte, in dem keine
                           nichtterminalen Symbole mehr vorkommen. ? ;
```

[1] Hier nur vereinfacht gezeigt. Der ISO-Standard enthält die vollständige Definition der EBNF in EBNF
[2] Das Nichtterminal `Definitionsliste` wird rekursiv verwendet

Nachfolgend einige weitere EBNF-Beispiele:

Ziffer = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9";

Eine Ziffer ist entweder eine "0", oder eine "1", oder eine "2", oder ... eine "9".

if-Anweisung = "if" , "(" , Bedingung , ")" , Anweisung , ["else" , Anweisung];

Die if-Anweisung hat einen optionalen else-Zweig.

Name = Buchstabe , {Buchstabe | Ziffer};

Ein Name besteht aus einem Buchstaben, gefolgt von beliebig vielen weiteren Buchstaben und/oder Ziffern. Daraus folgt, daß ein Name mindestens aus einem Buchstaben besteht, eine Obergrenze für die Namenslänge existiert dagegen nicht.

Telephonnummer = [(Ländervorwahl | "0") , Ortsvorwahl] , Nummer;

Eine Telephonnummer ist eine Nummer mit einer optionalen Ortsvorwahl, der entweder eine Null oder eine Ländervorwahl vorangestellt ist. Die runden Klammern fassen die alternativen Vorwahlarten zusammen um klarzustellen, dass beiden eine Ortsvorwahl folgen muss; Statt der Gruppierung durch ( ) hätte man auch, wie nachfolgende gezeigt, ein eigenes Nichtterminal einführen können.

Telephonnummer = [Vorwahl , Ortsvorwahl] , Nummer;
Vorwahl = Ländervorwahl | "0";

Beispiel: die "nackte" Nummer der FH Ingolstadt ist 93480. Die Ortsvorwahl von Ingolstadt ist 841. Die Ländervorwahl von Deutschland ist 0049. Erlaubt sind nach obiger Syntax die Nummern 93480, 0841 93480 und 0049 841 93480, aber nicht 0049 0841 93480 und auch nicht 0049 93480.

Buchstaben = "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J"|
"K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T"|
"U" | "V" | "W" | "X" | "Y" | "Z";
Vokale = "A" | "E" | "I" | "O" | "U";
Konsonanten = Buchstaben – Vokale;

Die Sprache der Konsonanten lässt sich elegant als die Differenz von Buchstaben und Vokalen spezifizieren; ein Konsonant ist ein Buchstabe, der kein Vokal ist. Die Sprache der Vokale ist offensichtlich eine reguläre Sprache, denn sie ist bereits ohne Verwendung von weiteren Nichtterminalen ausgedrückt. Daher ist in diesem Fall der Einsatz der Mengendifferenz erlaubt.

Buchstaben = ? Alle Großbuchstaben des lateinischen Alphabets ? ;

Anstatt wie oben die Buchstaben einzeln explizit aufzuzählen, definieren wir sie in Form einer speziellen Sequenz (special sequence). Das Ausdrucksmittel der speziellen Sequenz ?...? sollte man sparsam einsetzen, da man damit leicht den soliden Boden der formalen Sprachdefinition verlässt. Man setzt es i.a. nur ein, um wie oben triviale aber eben leider mühsame Aufzählungen von terminalen Zeichen zu vermeiden.