

Einführung zum Parsing – Q&As

(SEKS-BlockTag3, WS2009, 2010-01-07)

Was ist Syntax? – Die Syntax eines Programms beschreibt, welche Zeichenfolgen ein gültiges Programm darstellt. Zum Beispiel definiert die Java-Syntax, welche Folge von Zeichen ein gültiges Java-Programm darstellt.

Was ist Semantik? – Die Semantik legt die Bedeutung eines (syntaktisch korrekten) Programms fest. Die Zeichenfolge „2 + 3“ mag einer Syntax für arithmetische Ausdrücke folgen. Damit ist noch nicht gesagt, dass die Zeichen „2“ und „3“ die Bedeutung (Semantik) von Zahlen und das „+“ die Bedeutung einer Funktion hat, die die Zahlen addiert. Diese Interpretation wird über die Semantik definiert.

Was ist eine Grammatik? – Als Grammatik bezeichnet man eine Menge von Regeln zur Erzeugung (oder Überprüfung) eines syntaktisch gültigen Ausdrucks einer Sprache.

Was ist eine kontextfreie Grammatik? – Eine Grammatik, die ausschließlich durch Regeln der Form $N \rightarrow T$ beschrieben werden kann, heißt kontextfreie Grammatik (*context free grammar*). Die Regeln heißen Produktionen oder Produktionsregeln. Die linke Seite der Regeln sind Nicht-Terminalsymbole, die rechte Seite besteht aus einer definierten Abfolge von Terminalsymbole und/oder Nicht-Terminalsymbole. Terminalsymbole werden in der Regel auch als Token bezeichnet.

Was ist die EBNF? – Die BNF (Backus-Naur-Form) bzw. die EBNF (*Extended BNF*) ist eine Sprache zur Beschreibung der Produktionsregeln einer kontextfreien Sprache. Eine Sprache zur Beschreibung von Sprachen nennt man Metasprache.

Was ist ein Syntaxbaum? – Ein (konkreter) Syntaxbaum (Parsebaum) bildet die Struktur eines Programms gemäß den Produktionsregeln einer Grammatik als Baum ab. Die Blätter des Baums sind die Terminalsymbole der Grammatik, die restlichen Knoten sind die Nicht-Terminalsymbole der Grammatik. Anders als der Abstrakte Syntax Baum (AST, siehe unten) enthält der Parsebaum alle Zeichen des geparsen Programms sowie die Information, welche Regeln der Grammatik beim Parsen angewendet wurden.

Was ist ein AST (*Abstract Syntax Tree*)? – Die Ersetzung von Tokens im Syntaxbaum durch Werte, die von der konkreten Syntax absehen, sprich abstrahieren, führt zu einem AST. Ein Beispiel: Die Zeichenfolge „1“ wird durch den Wert 1 ersetzt. Ebenfalls werden alternative Zahlendarstellungen, wie „0x01“ (hexadezimale Kodierung) oder „0000001b“ (binäre Darstellung) durch den Wert 1 im AST ersetzt. Der AST macht unabhängig von der konkret verwendeten Syntax.

Was ist ein Tokenizer? – Der Tokenizer stellt Zeichenfolgen zu Einheiten zusammen. Die Zeichenfolge „2“, „3“, „+“, „1“ wird vom Tokenizer z.B. in die Abfolge „23“, „+“, und „1“ zerlegt. Der Tokenizer beschränkt sich auf die Zusammenfassung direkt aufeinander folgender Zeichen.

Was ist ein Parser? – Der Parser erzeugt aus einer Zeichenfolge einen konkreten oder abstrakten Syntaxbaum. Manche Definitionen beschränken die Arbeit des Parsers auf die Erstellung eines konkreten Syntaxbaums. Die Arbeit des Parsers kann, muss aber nicht, durch einen Tokenizer unterstützt werden.

Was ist ein Compiler? – Ein Compiler besteht aus einem Parser und einer Programmeinheit, die den AST in eine andere Sprache umwandelt.



Was ist ein Interpreter? – Ein Interpreter besteht aus einem Parser und einer Programmeinheit, die den AST „interpretiert“, sprich als Anweisung versteht und ausführt.