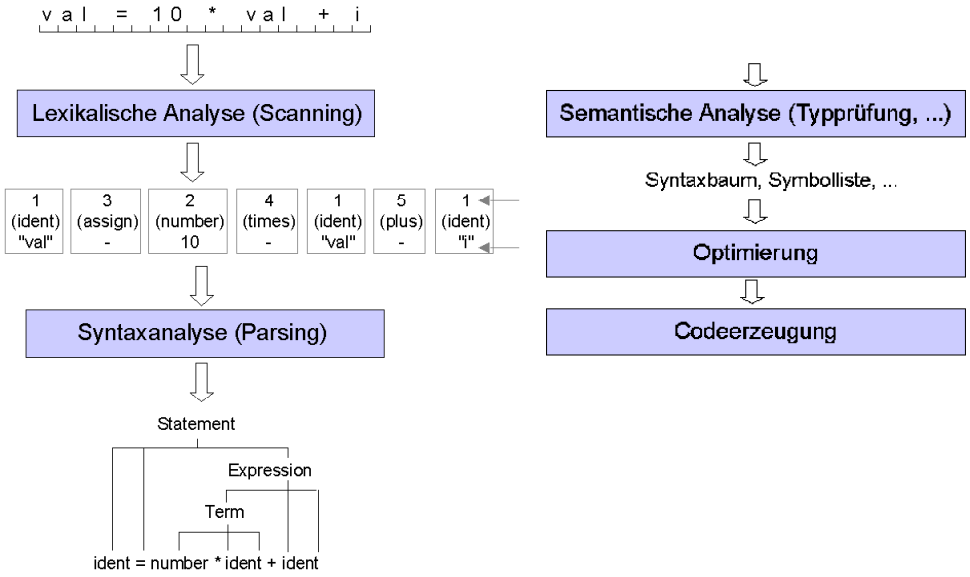
Zürcher Hochschule
für Angewandte Wissenschaften

Programmiersprachen - Compiler 1

Berechnung von arithmetischen Ausdrücke

In diesem Praktikum werden ein Scanner und ein Parser eingesetzt, um arithmetische Ausdrücke mit natürlichen Zahlen auszuwerten und zu berechnen, z.B.: (1 + 3) * 5 = 20 .

Dabei soll die Berechnung durchgeführt werden. Eine Grundstruktur von Scanner und Parser ist vorgegeben. Der Ablauf vom bei der Verarbeitung stellt folgende Grafik dar:



Der Scanner

Mit dem Scanner wird der arithmetische Ausdruck (Zeichenstrom) in folgende Teile (Tokens) zerlegt: Zahlen, Operatoren und Klammern. Dieser Tokenstrom wird dann dem Parser übergeben.

Der Parser

Der Parser in unserem Beispiel kann Ausdrücke aus der Sprache mit folgender Grammatik parsen:

```
Expression = Term {("+" | "-") Term}
Term = Faktor {("*" | "/") Faktor}
Factor = Zahl | "(Expression )"
```

Ausdruck, Term und Faktor sind sog. Produktionen. Für jede Produktion muss im Parser eine Methode gleichen Namens bestehen, welche gemäss Angaben in der Grammatik rekursiv die Methoden der anderen Produktionen aufruft, aus denen die

Produktion selbst besteht. So ruft z.B. die Methode Factor diejenige Methode der Produktion Ausdruck (expr) auf.

Aufgabe 1

Der Scanner soll so umgeschrieben werden, dass Werte vom Typ Double verarbeitet werden können. Dazu muss die Methode readNumber im Scanner so angepasst werden, dass Fliesskommazahlen erkannt werden können (ohne Exponenten). überlegen Sie sich den Automaten für Fliesskommazahlen mit folgender Grammatik Ziffer {Ziffer} ['.' {Ziffer}] und implementieren Sie diesen im Scanner.

Aufgabe 2

Der Parser ist noch unvollständig. Ihre Aufgabe ist es den Parser so zu ergänzen, dass er einfache arithmetische Ausdrücke direkt ausrechnen kann.

Die Berechnung soll als sog. Stackrechner implementiert werden: Bei einer Zahl wird diese einfach auf den Stack geschoben. Bei einem Operanden wird die Operation mit den beiden obersten Werten auf dem Stack ausgeführt und das Resultat wieder auf dem Stack abgelegt.

Aufgabe 3

Es sollen auch die beiden Konstanten "PI" und "E" in Ausdrücken verwendet werden können: 4*PI + E

Dazu muss die Factor Methode erweitert werden und im Scanner die Methode readName implementiert werden. Als *kind* soll IDENT zurückgegeben werden.

Implementieren Sie den Scanner so, dass der Scanner leicht neue Konstantennamen erweitert werden kann.