

Kurze Darstellung der Sprache PRO

Programme in der Sprache PRO bestehen aus Vereinbarungen und Anweisungen.

Eine **Vereinbarung** ordnet einem Bezeichner einen Typ zu. In PRO sind folgende Typen zugelassen:

Zahl,
Zahlenfolge,
Text.

Zu den Vereinbarungen zählen auch **Funktionen** (Funktionsprozeduren). Funktionen sind selbst wieder wie Programme aufgebaut; sie können zusätzlich in der definierenden ersten Zeile formale Parameter enthalten, die bei einem späteren Aufruf durch aktuelle Parameter ersetzt werden. Funktionen liefern stets einen Wert, der durch

ergebnis ...

in der Funktion kenntlich zu machen ist.

Anweisungen sind entweder elementar oder zusammengesetzt. Elementare Anweisungen sind die leere Anweisung oder von der Form

$x \leftarrow \alpha$ (**Zuweisung**);

hierbei ist x ein Bezeichner und α ein Ausdruck; in α dürfen auch Prozeduraufrufe enthalten sein, die jeweils aus dem Bezeichner für eine zuvor vereinbarte Prozedur und den entsprechenden aktuellen Parametern bestehen. Weiterhin zählen das Eingabekommando

lies(...)

und das Ausgabekommando

zeige(...)

zu den elementaren Anweisungen.

Zusammengesetzte Anweisungen entstehen in PRO

- durch das Verbinden von Anweisungen mithilfe des Konstruktors ";"
(Semikolon) (**Konkatenation**),
- durch die **bedingte Anweisung**

wenn *Bedingung* dann

A1

sonst

A2

ende,

wobei A1 und A2 wiederum beliebige Anweisungen sind,

- durch eine **Schleife**

solange *Bedingung* tue

A

ende,

wobei A eine beliebige Anweisung ist.

Bedingungen sind von der Form $\alpha = \beta$, $\alpha \neq \beta$ für zwei Ausdrücke α und β . Falls α und β als Ergebnis je eine Zahl liefern, sind auch $\alpha \leq \beta$, $\alpha < \beta$, $\alpha \geq \beta$ und $\alpha > \beta$ als Bedingungen erlaubt. Mehrere Bedingungen dieser Form können durch und und oder miteinander verknüpft werden oder mittels nicht negiert werden.

Schließlich muß man festlegen, wie **Ausdrücke** aufgebaut sind. Diese bestehen aus Konstanten (Zahlen, in Anführungsstriche gesetzte Texte, in eckige Klammern eingefaßte Liste von Zahlen), aus Bezeichnern und aus Operatoren. Operatoren sind Klammern, sowie

- die Zeichen +, -, *, / für Zahlen,
- die Zeichen •, erstes, rest für Zahlenfolgen.

Für Texte wurden bisher keine Operatoren eingeführt.

Zusätzlich müssen folgende Regeln gelten, die wir hier nicht formal definieren, sondern nur umgangssprachlich beschreiben:

- a) Jeder in einer Anweisung oder einer Bedingung auftretende Bezeichner muß zuvor deklariert worden sein. Tritt der Bezeichner in einem Funktionsrumpf auf, so muß er ein formaler Parameter dieser Funktion sein, oder er muß in dieser Funktion deklariert worden sein, oder er ist der Bezeichner für eine Funktion, die in einer umfassenden Funktion vereinbart wurde.
- b) Kein Bezeichner darf zweimal deklariert werden. (Diese Regel wird später aufgeweicht.)
- c) Bei jedem Funktionsaufruf müssen die aktuellen Parameter vom gleichen Typ (Zahl, Zahlenfolge, Text) wie der korrespondierende formale Parameter sein.
- d) Ausdrücke sind nur dann korrekt aufgebaut, wenn die Operatoren nur mit Operanden des entsprechenden (Ergebnis-)Typs verbunden werden. (Verboten ist also z.B. `rest(3+7)`, weil `3+7` eine Zahl ist, der Operator `rest` jedoch eine Zahlenfolge erwartet; ebenso sind `7•"ja"` oder `["nein"]` oder `"ich"+"du"` nicht erlaubt.) Auch müssen in Bedingungen $\alpha=\beta$ usw. die beiden Ausdrücke vom gleichen Ergebnis-Typ sein. (Verboten ist also z.B. `"ja"=3` oder `[4,2]≠6+23` oder `2=3` und `27`.)

Weiterhin verlangt man, daß in einem Programm stets zunächst alle Vereinbarungen und erst danach die Anweisungen stehen. Dadurch werden die Programme übersichtlicher.

Wir formalisieren die Syntax der Sprache PRO später exakt.