# Compilerbau WiSe 2002/03

# Hinweise

## Sequentielle semantische Analyse

Für die Erstellung der Identifikatortabelle und der Zuordnung und Prüfung der Variablendeklarationen und Bindungen wird im Skript im Kapitel 5.2 ein sequentieller Algorithmus grob skizziert. Wir geben hier eine Verfeinerung de Algorithmus an. Dabei verwenden wir für den echten Blockzähler einen Keller `BK`, mit folgenden Funktionen:

`push(N,BK)`
Lege das Element N oben auf auf dem Keller ab. Bereits abgelegte Elemente bleiben erhalten.

`R = pop(BK)`
Entferne das oberste Element vom Keller BK und gebe es als Wert zurück.

`R = top(BK)`
Gebe das oberste Element vom Keller BK als Wert zurück.

`R = read(N,BK)`
Gebe das N-te Element von oben vom Keller BK als Wert zurück, mit 0 beginnend. Ist N+1 höher als die Zahl der Kellerelemente, wird `ERROR` zurückgegeben. (Also gilt `top(BK) == read(0,BK)`)

Entsprechende Funktionen verwenden wir auch für den Identifikatorenkeller `IK`.
Einträge im Identifikatorkeller sind von der Form
`Name:Typ:Dimension:Relativadresse:Blockniveau:Blockzähler`, wobei wir Dimension und Relativadresse hier der Klarheit halber ignorieren.
Nun zum Algorithmus. Wir lesen für jede Phase den MMS-Quellcode eines Programms von links nach rechts sequentiell durch.

### 1. Phase: Erkennen de Deklarationen, Aufbauen des Identikatorkellers

Da Variablen- und Prozedurdeklarationen gemischt auftreten können, müssen wir auch hier bereits im einen Blockkeller die aktuelle Blockschachtelung mitführen, um beim Verlassen eines Blocks noch zu wissen, welches der aktuelle Block für die Deklaration ist.

```
BZ := 0;   // globaler Blockzähler
BN := -1;  // Aktuelles Blockniveau
push(BZ,BK);

while (noch Wörter auf der Eingabe) {
 Lese Eingabe.
 Falls (treffe auf 'begin') {
  BZ := BZ + 1;
  BN := BN + 1;
  push(BZ,BK);
 }
 Falls (treffe auf 'end') {
  BN := BN - 1;
  pop(BK);
 }
```

```
 Falls (treffe auf Identifikatordeklaration 'var <name> : <typ>') {
   Falls (es gibt einen Eintrag in der ID-Tab mit Name==<name> und Blockzähler==top(BK))
      Fehler("Variable <name>       // einzelne Feldwerte mit ':' getrennt:
doppelt deklariert");
   sonst
      // lege die Deklaration im Identifikatorenkeller ab.
      push(<name>:<type>:?:?:BN:top(BK),IK)
 }
```

## 2. Phase: Prüfen der Bindungen

```
BZ := 0;   // globaler Blockzähler
BN := -1;  // Aktuelles Blockniveau
push(BZ,BK);

while (noch Wörter auf der Eingabe) {
 Lese Eingabe.
 Falls (treffe auf 'begin') {
  BZ := BZ + 1;
  BN := BN + 1;
  push(BZ,BK);
 }
 Falls (treffe auf 'end') {
  BN := BN - 1;
  pop(BK);
 }
 Falls (treffe auf angewandtes Auftreten des Identifikators <name>) {
   N := 0;
   while ( read(N,BK) != ERROR  &&
    es gibt keinen Eintrag in der ID-Tab mit name==<name>
        und Blockzähler==read(N,BK) ) N := N + 1;
   Falls read(N,BK) == ERROR
      Fehler("Variable <name> nicht gebunden");
 }
```

---

*Dietmar Lammers*

Last modified: Sun Jan 4 12:53:08 CET 2004