

5 Semantische Analyse

5.1 Überblick

Ein Compiler muß prüfen, ob die Quellprogramme sowohl den syntaktischen als auch den semantischen Konventionen genügen. Wir unterscheiden:

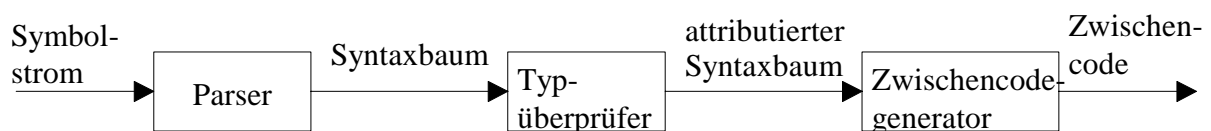
- statische Überprüfungen (während der Kompilation)
- dynamische Überprüfungen (zur Laufzeit)

Beispiele für **statische** Überprüfungen:

1. Typüberprüfungen
2. Überprüfungen des Kontrollflusses (goto, break)
3. Eindeutigkeitsprüfungen (z.B. switch -Anweisung)
4. Gültigkeitsprüfungen (scope)
5. Auf Namen bezogene Überprüfungen (PROCEDURE a; ... END a; in Modula-2)

Die meisten dieser Überprüfungen können in andere Aktivitäten einbezogen werden, z.B. können Informationen über Namen in die Symboltabelle eingetragen werden, was zur Prüfung auf eindeutige Deklaration genutzt werden kann. Pascal-Compiler kombinieren die statische Überprüfung und die Zwischencode-Erzeugung mit der Syntaxanalyse.

Bei komplexeren Sprachen, wie z.B. Ada, wird zwischen der Syntaxanalyse und der Zwischencode-Erzeugung eine Phase zur Typprüfung eingesetzt.



5.2 Typ-Überprüfung

Ein Typüberprüfer verifiziert, daß der Typ eines Konstrukts zu dem Typ paßt, der aufgrund des Kontextes des Konstruktes erwartet wird.

Bsp:

```

a) VAR  i : integer;   (Pascal)
      a : real;

```

⋮

```

  i := a+1;
integer   Ausdruck vom Typ real

```

Fehler!


```

procedure whilestatement;

var    x: item;
       lc1,lc2: integer;
begin
  insymbol;
  lc1 := lc;
  expression(fsys+dosy], x);
  if not (x.temp in [bools,notyp]) then error(17);
  lc2 := lc; emit(11);

  if sy = dosy then insymbol else error(54);
  statement(fsys);
  emit(10,lc1);
  code[lc2],y := lc
end { whilestatement } ;

```

b) Es folgen Beispiele für die Implementation von Typ-Überprüfern im PL0-Compiler in C:

```

void callstmnt(){
  int i;

  getsym();
  if (sym == ident) {
    i = position(id, txcurr);
    if (i == 0) error(11);
    if (objkind(i) != procobj) error(15);
    emit(CAL, levcurr-objlevel(i), objaddr(i));
    getsym();
  } else error(14);
}

void readstmnt() {
  int i, k;

  getsym();
  if (sym == lparen) {
    do {
      getsym();
      if (sym == ident) {
        i = position(id, txcurr);
        if (i == 0) error(11);
        k = objkind(i);
        if (k == constobj || k == procobj)
          error(12);
        else if (k == varobj) {
          emit(INP, 0, 0);
          emit(STO, levcurr-objlevel(i), objaddr(i));
          getsym();
        }
      } else error(33);
    } while (sym == comma);
    match(rparen, 22);
  } else error(43);
}

```