

# Compilerbau & Abstrakte Maschinen

*Sommersemester 2006*

*3. Übungsblatt*      *Abgabetermin: Mo, 22. Mai 2005 in der Vorlesung*

Aufgabe 1: Initialisierung von Variablen

*4 Punkte*

Modifizieren Sie die Übersetzungsfunktion `code` so, dass Variablen initialisiert werden können. Beispielsweise soll die Initialisierung `int a = 0;` übersetzt werden können.

Aufgabe 2: Referenz-Parameter

*8 Punkte*

C++ bietet neben Parameter-Übergabe “by-value” auch Referenz-Parameter an. Wird z.B. im Rumpf der Funktion  $f$  der formale Parameter  $x$  verwendet, ist die Variable gemeint, mit der  $f$  als aktuellem Parameter aufgerufen wurde.

```
int a;  
void f(int &x) {  
    x=7;  
}  
main() {  
    f(a);  
}
```

Nach Ausführung von  $f(a)$  sollte also die Variable  $a$  den Wert  $7$  enthalten.

- Modifizieren Sie das Übersetzungsschema für Parameter-Übergabe
- und für Variablenzugriffe so, dass Referenz-Parameter korrekt behandelt werden!
- Übersetzen Sie das Programm und überprüfen Sie die Korrektheit mit dem VAM-System!

### Aufgabe 3: Variable Argumentlisten

8 Punkte

Im nächsten Schritt erweitern wir C um Funktionen mit variablen Argumentlisten. Bei solchen Funktionen steht am Ende der Parameter-Liste “...” (davor können normale Parameter stehen). Innerhalb des Rumpfs der Funktion erhält man das jeweils nächste (variable) Argument von Typ  $T$  mit einem Aufruf  $next(T)$ . Beispiel:

```
int sum(int n, ...) {
    // Summe von n Zahlen
    int result = 0;
    while(n-->0) result += next(int);
    return result;
}
```

Mögliche Aufrufe wären z.B. `sum(5,a,b,c,d,e)` oder `sum(3,1,2,3)`.

- a) Wie muss die Übersetzung eines solchen Funktionsaufrufs modifiziert werden ?
- b) Überlegen Sie sich eine Übersetzung für  $next(T)$ !