

Programmanalysen und Compilerbau

Wintersemester 2007/8
Prof. Plödereder, Stefan Staiger

4. Übungsblatt

Die Übung findet am 11.12. statt.

Aufgabe 4.1

Gegeben sei das folgende Programm:

```
procedure Berechnung is
  X      : Integer;
  A,B,C,D : Integer;

  M : constant := 3;

  procedure P() is
  begin
    A := 2 * M;
    B := A;
  end P();

begin
  X := Input();

  P();

  if B > 2 then
    C := 1;
  else
    C := 2;
  end if;

  D := C * X;

  Print(D);
end Berechnung;
```

Optimieren Sie das Programm für eine möglichst effiziente Ausführung (Laufzeit). Welche Optimierungen haben Sie dabei angewendet, und wie hängen diese voneinander ab?

Aufgabe 4.2

Was versteht man im Zusammenhang mit Codeoptimierung und Datenflussanalyse unter „konservativen“ Entscheidungen oder Abschätzungen? Begründen Sie, ob die konservativen Lösungen der Datenflussprobleme „verfügbare Ausdrücke“ und „aktive Variablen“ Teil- oder Obermengen der tatsächlichen Lösungen sind.

Aufgabe 4.3

Gegeben sei folgendes Programmfragment:

```
...
a := 1;
b := 2;

REPEAT
  c := a + b;
  d := c - a;

  IF ... THEN
    <label>:
      d := b * d;

      IF ... THEN
        d := a + b;
        e := e + 1;
        GOTO label;
      END IF;
    END IF;
  UNTIL ...;

a := b + d;
b := a - d;
...
```

Die Verzweigungsbedingungen sollen hier ignoriert werden.

- a. Erstellen Sie den zugehörigen Kontrollflussgraphen.
- b. Lösen Sie die Datenflussprobleme
 - i) „verfügbare Ausdrücke“ (ohne Konstanten!) und
 - ii) „aktive Variablen“für die einzelnen Anweisungen mit Hilfe des iterativen Verfahrens über die Grundblöcke und mit nachträglichem Propagieren der Ergebnisse in die Grundblöcke.
- c. Welche Auswirkungen auf die Lösungen erwarten Sie, wenn die Verzweigungsbedingungen ergänzt werden und diese einige der vorkommenden Variablen enthalten?

Aufgabe 4.4

Welche Auswirkungen hat die Verwendung des GOTO-Befehls auf die Datenflussanalyse, wenn

- a. Sprungziele durch symbolische Namen („Label“) gekennzeichnet sind?
- b. Sprungziele durch Zeilennummern gekennzeichnet sind und das Argument des Sprungbefehls ein arithmetischer Ausdruck sein darf („computed Goto“)?
- c. Sprungziele durch symbolische Namen gekennzeichnet sind, diese aber zusätzlich Variablen zugewiesen und für Sprünge diese Variablen verwendet werden dürfen? (Dieses Sprachkonstrukt gibt es z.B. in Gnu C.)