

Übungen zur Vorlesung: Struktur und Implementierung von Programmiersprachen I Blatt 10 (Programmanalyse)

Aufgabe 13 (Symbolische Interpretation und Datenflussanalyse)

Nachfolgend finden Sie Pascal-Programme, von denen jedes, wie in der Vorlesung behandelt, nach folgenden Eigenschaften zu analysieren ist:

- (a) Einfache symbolische Interpretation des Initialisierungszustands der Variablen (**U**ninit, **M**aybeinit, **I**nit).
- (b) Last-def-Analyse durch volle symbolische Interpretation.
- (c) Liveness von Variablen durch Rückwärtsanalyse mit Hilfe von Datenflussgleichungen.

Verwenden Sie bitte die in Kommentarklammern gegebenen Nummern, um auf das dahinter stehende Statement (falls vorhanden) oder den Knoten im Flussgraphen Bezug zu nehmen.

```
program ex1;
var a, b, c: integer;
begin
  (*1*) read a;
  (*2*) b := a+1;
  (*3*) b := a+b;
  (*4*) if a>0
    then begin
      (*5*) c := a*b;
      (*6*) b := c-1
      (*7*)
    end;
  (*8*) c := b+1;
  (*9*) write c
end.
```

```

program ex2;
var a, b, c: integer;
begin
  (*1*) read a;
  (*2*) if a<0 then begin
    (*3*) write c
    (*4*)
  end;
  (*5*) for b := 0 to a do begin
    (*6*) c := b
    (*7*)
  end;
  (*8*) write c
end.

```

```

program ex3;
var a, b, c, d: integer;
begin
  (*1*) read a;
  (*2*) for b := 0 to a do begin
    (*3*) if b<3 then begin
      (*4*) c := a+1;
      (*5*) d := a*c
      (*6*)
    end else begin
      (*7*) c := a+1;
      (*8*) d := c+d
      (*9*)
    end;
    (*10*) a := d+1
    (*11*)
  end;
  (*12*) write a;
  (*13*) while c>a*a do begin
    (*14*) c := c-1
    (*15*)
  end;
  (*16*) write c
end.

```