Problem Sheet 6: Data Flow Analysis

Chris Poskitt*
ETH Zürich

Starred exercises (∗) are more challenging than the others.

1 Reaching Definitions Analysis

These exercises are based on the material from the "Reaching Definitions Analysis" section of this lecture:

http://se.inf.ethz.ch/courses/2013b_fall/sv/slides/08-ProgramAnalysis.pdf

Consider the following program fragment:

```
x := 10;
while x > 0 do
     y := 2 * y;
     if y > 10 do
          x := x - 1;
     else
          y := x + 2;
     end
     x := x - 1;
end
x := x - 1;
```

i. Draw the *control flow graph* of the program fragment.

ii. Annotate the control flow graph with the results of a *reaching definitions analysis*.

iii. Provide (or draw) the *use-definition* information for program variables `x` and `y`.

*These exercises are from previous iterations of the course when Stephan van Staden was the teaching assistant.

2 Live Variables Analysis

These exercises are based on the material from the "Live Variables Analysis" and "Equation Solving" sections of this lecture:

http://se.inf.ethz.ch/courses/2013b_fall/sv/slides/08-ProgramAnalysis.pdf

Consider the following program fragment:

```
x := y;
x := x - 1;
x := 4;
while y < x do
     y := y + x;
end
y := 0;
```

i. Identify the elementary blocks of the program and label them.

ii. Write down the *equations* for a live variables analysis of the program.

iii. Solve the data flow equations using *chaotic iteration*.

iv. Using the result obtained in (iii), perform *dead code elimination* on the program fragment.

v. (∗) Is the program resulting in step (iv) free of dead variables? If not, explain why and modify the live variables analysis so that it can be used to produce a program free of dead variables.