Problem Sheet 7: Program Slicing and Abstract Interpretation

Chris Poskitt*
ETH Zürich

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

1 Program Slicing

These exercises are based on the material from the "Program Slicing" section of this lecture:

http://se.inf.ethz.ch/courses/2014b_fall/sv/slides/09-ProgramAnalysis-Slicing.pdf

Consider the following program fragment:

```
x := 0;
y := 0;
i := n;
j := n;
while i > 0 do
     x := x + 1;
     i := i - 1;
     j := i;
     while j > 0 do
          y := y + 1;
          j := j - 1;
     end
end
print(x);
print(y);
```

i. Draw the *program dependence graph* for this fragment.

ii. Compute the backward slice of the program fragment for the *slicing criteria* `print(x)` and `print(y)`.

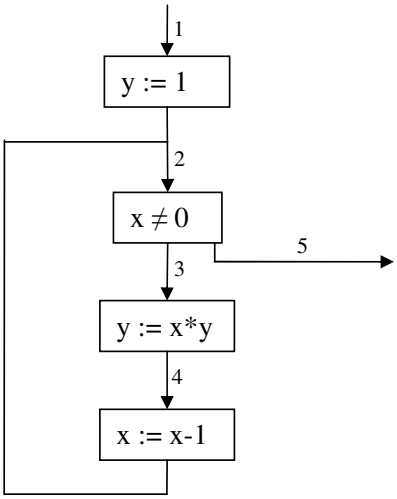
*These exercises are from previous iterations of the course when Stephan van Staden was the teaching assistant.

2 Abstract Interpretation

These exercises are based on the material from the "Abstract Interpretation" lecture:

http://se.inf.ethz.ch/courses/2014b_fall/sv/slides/10-AbstractInterpretation.pdf

Consider again the factorial algorithm from the lecture with sign analysis equations:



$$A_1 = [x \mapsto +, y \mapsto \top]$$
$$A_2 = A_1[y \mapsto +] \sqcup A_4[x \mapsto A_4(x) \ominus +]$$
$$A_3 = A_2$$
$$A_4 = A_3[y \mapsto A_3(x) \otimes A_3(y)]$$
$$A_5 = A_2 \sqcap [x \mapsto 0, y \mapsto \top]$$

i. Compute the analysis result by *chaotic iteration*.

ii. Is the analysis precise? What is it unable to prove about the program?

iii. Improve the precision by:

(a) Changing the program but not the analysis (i.e. compute the factorial in a way that is more "friendly" for the analysis).

(b) (∗) Changing the analysis but not the program.