SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Jan Reineke
Sebastian Hahn, M.Sc.

Verification of Real-Time Systems SS 2015

Assignment 4

Deadline: May 28, 2015, before the lecture

Exercise 4.1: Intervals, Variable Copies, and Cooperation (20 = 3+1+4+1+7+4(+4) Points)

Consider the following C function.

```
int f(int z)
{
  int x = 1;
  int y = z;
  while (complex expression) {
    if (y != z)
      x = 2;
    x = 2 - x;
    if (x != 1)
      y = 2;
  }
  return x;
}
```

(a) Construct the control-flow graph for this program. Nodes in the graph represent program points, edges represent statements to execute. (See pages 6 and 7, slide set of 30th April)

(b) Mark all statements reachable during an actual program execution. Can you make a statement about `x` at the end of the program?

(c) Perform an interval analysis as presented in the lecture. First, give the set of equalities of the abstract reachability semantics. Second, determine the least fixed point of the equations (e.g. by Kleene iteration) and give your intermediate steps.

(d) Is interval analysis sufficient to prove the properties identified in (b)?

(e) The first `if` compares two variables. To make a statement about the outcome of this comparison, it is important to know the relation between them. Therefore, design a *copy propagation* analysis. The analysis should determine whether one variable is a copy of another variable. Give the lattice you operate on (domain, partial order, join, bottom and top element) and all transfer functions. Handle the condition evaluation in a smart way.

(f) Perform the above defined copy propagation analysis. Are the results sufficient to prove the properties identified in (b)?

(g) *Optional:* Can you combine the interval and the copy propagation analysis in a way that is sufficient to determine the properties in (b)? Briefly describe your approach on a high-level. What is the resulting lattice you operate on? Also give the transfer functions, if changed.

Exercise 4.2: Modulo p Analysis (15 = 3+3+3+2+4 Points)

Design an analysis that tracks for each variable its value *modulo* p, where $p \geq 1$ is an arbitrary but fixed natural number.

(a) Define the lattice you operate on (domain, partial order, join, bottom and top element).

(b) Give all transfer functions. Handle the condition evaluation in a smart way.

(c) Give the abstract operator evaluations (in form of tables) for operators $+, -, *$.
Hint: Be specific about the case $0 \ mod \ p$.

Consider the following C function.

```
void f(unsigned z)
{
  unsigned x = 5;
  unsigned y = 2 * z;
  while (complex expression) {
    x = x + 2;
    if (y == 5) {
      x = x + 1;
      y = y - 1;
    }
  }
  assert(x is odd);
}
```

Construct the control-flow graph for this function and perform your new $mod \ p$ analysis for a suitable p. Are you able to prove the assertion in the end? If not, why?