**School of Innovation,
Design and
Engineering**

Module 4: Assignments on Static Program Analysis

(Revised: 2015-11-05)

This document contains assignments on static analysis. Solutions to these assignments are to be handed in (in pdf format) to bjorn.lisper@mdh.se. These assignments provide part of the examination for Module 4.

ASSIGNMENTS

1. An example program

We will use the following simple program throughout:

```
x = 1;
y = 3;
while y > 0 do
  x = x*y;
```

2. Control-Flow Graph

Draw the Control-Flow Graph (CFG) for the example program. Put a unique number on each node in the CFG, just like it is done in the lecture on data flow analysis.

3. Reaching Definitions Analysis

Set up the data flow equations for doing a Reaching Definitions analysis of the example program. Then solve them using least fixed-point iteration. Include the equations, the fixed-point iterations, and the final solution in your solution to be handed in.

4. Even-odd Value Analysis

In this assignment you will analyse the given example program with a value analysis that can detect whether the value of a variable is always even or always odd in a given program point. The analysis is very similar to interval analysis, but rather than intervals the analysis has only four abstract values: EMPTY (the empty set), {O} (odd), {E} (even), and {O,E} (odd or even, i.e., no information).

The abstract states are tables where each table entry is some of the abstract values above.

We must interpret arithmetic constants and operations in terms of abstract values:

- A constant *c* is interpreted as {O} if odd, and {E} if even,
- The arithmetic operation * is interpreted over the abstract values as given by the following table:

*	EMPTY	{O}	{E}	{O,E}
EMPTY	EMPTY	EMPTY	EMPTY	EMPTY
{O}	EMPTY	{O}	{E}	{O,E}
{E}	EMPTY	{E}	{E}	{O,E}
{O,E}	EMPTY	{O,E}	{O,E}	{O,E}

This table represents rules such as {E}*{E} = {E} (even times even is even), {O}* {E} = {E} (odd times even is even), etc.

- The “join” (or “merge”) operation on abstract values is set union.

The transfer functions are similar to the transfer functions for the interval analysis:

- For **assignments** the transfer function updates the table entry for the assigned variable, using the interpretation of the right-hand side in terms of abstract values.
- For **join nodes** the transfer function is a merge of tables similar to the interval analysis, where the set union is applied elementwise to the table entries.
- For **test nodes**, both transfer functions are the identity function.

Set up the equations for doing an even-odd value analysis of the example program. Then solve them using least fixed-point iteration. Include the equations, the fixed-point iterations, and the final solution in your solution to be handed in. At entry, the program variables can hold any (integer) value: your analysis must handle this in a correct way.