Chair for Software Modeling and Verification

apl. Prof. Dr. Thomas Noll   Christina Jansen, Christoph Matheja

General Remarks

- Please hand in your solutions in groups of 3. If you are still looking for a group or your group has less than 3 members, please use the L2P or contact us after the exercise class such that we can mediate.

- If you have questions regarding the exercises and/or lecture, feel free to write us an email or visit us at our office.

Exercise 1 (Available Expressions):   (4 Points)

Extend the WHILE programming language of the lecture by a *do-while*-construct.

a) Adapt the *init*- and *final*-mapping as well as the *flow*-relation to capture the newly introduced construct.

b) Additionally, adapt all concepts needed to perform an available expression analysis on programs using the *do-while*-construct.

c) Perform an available expression analysis on the following program:

```
y := x + 1;
x := x + y;
do
    y := x + 1;
while (y < x);
y := y * x;
```

Exercise 2 (Isolated Entries and Exits):   (3 Points)

Consider the Available Expressions Analysis and the Live Variable Analysis from the lecture. Which of the equations make sense for programs that do not have isolated entries/exists? Explain your answer. How can the equations that do not make sense be improved?

Exercise 3 (Live Variables):   (3 Points)

Consider the following program:

```
x := 42;
x := x - 23;
x := 17;
```

a) At which labels is x a live variable?

b) Is the result of a Live Variable Analysis reasonable for the program from above? Explain your answer.

c) Improve the Live Variable Analysis from the lecture to yield more reasonable results for programs as above.