Exercise 3.1: Live Variables vs. Truly Live Variables (Points: 6)

Perform an analysis for live variables as well as an analysis for truly live variables on the following program. Use Round-Robin Iteration to solve your systems of inequations. Please write down all iteration steps. Assume $X$, the set of variables live at the end of the program, to be the empty set.

```
x = 5;
y = 6;
c:
if(y > 0) {
x = x * x;
x = x - 1;
y = y - 1;
goto c;
}
y = y * 1;
stopAt = y;
M[RESULT] = x;
```

Exercise 3.2: Truly Live Variables (Points: 5)

Implement a truly live variables analysis in PAG/WWW.

Exercise 3.3: Dead Variables (Points: 4)

In the lecture we discussed two analyses that enable the removal of unnecessary assignments: live and truly live variable analysis. In this exercise you should design a third analysis enabling such an optimization: an analysis that determines the set of (definitely) dead variables at each program point.

Exercise 3.4: Parameters (Points: 5)

Design an analysis that determines which variables can be classified as parameters. A parameter in this sense is simply a variable that is read before it is written by the program.

Exercise 3.5: Forward or Backward? (Bonus Points: 5)

In the first lecture, you learned that each analysis can be implemented as a forward or a backward analysis. However, one direction may be more complicated and less intuitive than the other. An analysis to compute available expressions as designed in Exercise 1.2 is typically realized as a forward analysis. In this exercise you are to redesign your available expressions analysis as a backward analysis. I.e., construct a backward analysis that computes what expressions are available at the different program points.