SAARLAND UNIVERSITY
COMPUTER SCIENCE

Prof. Dr. Reinhard Wilhelm
Prof. Dr. Sebastian Hack
Dr. Daniel Grund
Michael Jacobs, M.Sc.

Compiler Construction WS11/12

Exercise Sheet 10

Please hand in the solutions to the theoretical exercises until the beginning of the lecture next Friday 2012-01-27, 12:00. Please write the number of your tutorial group or the name of your tutor on the first sheet of your solution.

Exercise 10.1 Static Single Assignment Form (Points: 2+3)

- Which two criteria have to be fulfilled to assume that a given program is in SSA form?
- Transform the following program to SSA form.

```
x1 = 10;
x2 = 1;
while (x1 > 0) {
  x2 = x2 * 2;
  x1 = x1 - 1;
}
x2 = x2 + 3;
```

Exercise 10.2 Global Value Numbering (Points: 2+6+1)

Reconsider Kildall's Algorithm and the AWZ Algorithm for global value numbering as introduced in the lecture. They describe two different approaches to detect equivalence of program expressions.

- Give an example for a loop-free program for which Kildall's approach can detect more equivalences than the AWZ algorithm can. The plus in detected equivalences should allow to better optimize the program.
- Perform both analyses on your example program to show that Kildall's approach really results in more detected equivalences. Make the intermediate steps of your fixed point iteration explicit for both analyses.
- Describe at which point of the AWZ Algorithm the imprecision arises and explain why this happens.

Exercise 10.3 Partial Redundancy Elimination (Points: 6+3)

Consider Partial Redundancy Elimination as introduced in the lecture. The following program P contains some partially redundant computations.

```
b = 0;
while (i > 0) {
  if (i > 10) {
    t = a + 1;
    b = b + t;
  }
  s = a + 1;
  b = b + s;
  i = i - 1;
}
b = b + 1;
```

- Perform Partial Redundancy Elimination on P following the algorithm described in the lecture. This means you have to analyze for Anticipability, analyze for Earliestness and then apply the program transformation. You only have to consider non-trivial calculations in conditions and on right sides of assignments. Trivial calculations are a read from a single variable or the use of an integer constant.
- Which partial redundancy could not be removed from program P and which part of the Partial Redundancy Elimination approach is responsible for this? Transform the control flow of program P in a semantics preserving way that does not directly remove the partial redundancies but that allows a run of our Partial Redundancy Elimination approach to remove them.