

## Exercise Sheet 9

### Exercise 9.1 Static Single Assignment Form

- 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 9.2 SSA Form and Flow-Sensitivity

A commonly encountered argument for using the SSA form for program analysis is that analyses on SSA programs need not be flow-sensitive (i.e. they do not need to have a mapping from program locations to some abstract states) to achieve the same precision as flow-sensitive analyses.

Show that this argument is wrong by giving an example program in SSA form and a program analysis that gives better results on this input if performed flow-sensitively.

*Hint: Which operations can influence the analysis information for a variable  $x$  other than an assignment to  $x$ ?*

### Exercise 9.3 Interference Graphs

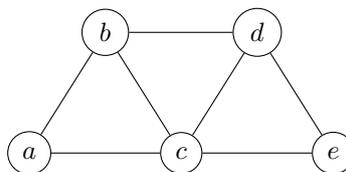
Consider the following code snippet.

```
s1 = 47;
s2 = 42;
s3 = s1 + s2;
do {
    s3 = s3 - s1;
    s4 = s3 + 2;
} while (s3 > s2);
s5 = s2 * s4;
s6 = s3 / s2;
print(s6 - s5);
```

1. Draw the interference graph.
2. Assign actual registers to the symbolic registers by coloring the interference graph using Chaitin's local-colorability criterion. Assume an overall number of 4 registers to be available.

### Exercise 9.4 More Interference Graphs

Consider the following interference graph.



Give an SSA form program without control flow that has the above graph as an interference graph. You may use assignments of constants to variables, assignments of results of binary arithmetic operations to variables, and stores of variable values to some (fixed) memory address in your program.

### Exercise 9.5 Chordal Graphs

Are the following graphs chordal or not? Justify your claims!

