*Technische Universität München*
*Fakultät für Informatik*
*Prof. Dr. H. Seidl*

*Winter Semester 10/11*
*Vesal Vojdani*
*Aleksandr Karbyshev*

# Program Optimization

*Exercise Sheet 10*

*Deadline: February 7, at the lecture!*

Exercise 1: *(H) Generating Loop Bounds* *8 Points*

Assume we want to reverse the loop-nesting orders in the following code:

```
for (i = 0; i <= 5; i++)
  for (j = i; j <= 7; j++)
    A[j][i] = 0;
```

We first write down all the constraints from the two loops:

$$0 \leq i \qquad\qquad i \leq j$$
$$i \leq 5 \qquad\qquad j \leq 7$$

In our new iteration order, the innermost variable is $i$, so we infer all the constraints on $i$:

$$0 \leq i$$
$$i \leq 5$$
$$i \leq j$$

This gives us the code for the innermost loop:

```
  for (i = 0; i <= min(5,j); i++)
```

Now, we perform Fourier-Motzkin elimination to get rid of the variable $i$. We obtain one additional constraint, $0 \leq j$, which replaces all inequalities involving $i$ in the original set. We end up with two constraints, $0 \leq j$ and $j \leq 7$, yielding the following code:

```
for (j = 0; j <= 7; j++)
  for (i = 0; i <= min(5,j); i++)
    A[j][i] = 0;
```

We're done, except *you* haven't done anything yet, so please perform the precise same transformation on the following loop:

```
for (i = 1; i <= 30; i++)
  for (j = i+2; j <= 40-i; j++)
    A[j][i] = 0;
```

Exercise 2: *(H) Deforestation* *5 Points*

Let comp, id, map, foldl, and tabulate be defined as in the lecture. Given the following code:

$$\begin{array}{lcl} \textbf{let}\ \mathsf{sq}\ x & = & x * x \\ \textbf{let}\ \mathsf{sumSquares} & = & \mathsf{comp}\ (\mathsf{foldl}\ (+)\ 0)\ (\mathsf{comp}\ (\mathsf{map}\ \mathsf{sq})\ (\mathsf{tabulate}\ \mathsf{id})) \end{array}$$

Simplify the function sumSquares as much as possible.