Program Optimization

Exercise Sheet 3
08.11., due 14.11. 12:30

Exercise 1: Monotonicity and Distributivity

Recall the lattice $\mathbb{Z}_\top = \mathbb{Z} \cup \{\bot, \top\}$ with the ordering "\=". Determine for each of the following functions $\mathbb{D} \rightarrow \mathbb{D}$, whether it is monotonic, distributive, and strict.

\[
\begin{align*}
\text{bot}(x) &= \bot \\
\text{top}(x) &= \top \\
\text{zero}(x) &= \begin{cases} 
\top & x = \top \\
1 & x = 0 \\
0 & \text{otherwise}
\end{cases} \\
\text{exp}(x) &= \begin{cases} 
\top & x = \top \\
2^x & x \geq 0 \\
\bot & \text{otherwise}
\end{cases} \\
\text{inv}(x) &= \begin{cases} 
\top & x = \top \\
-x & x \in \mathbb{Z} \\
\bot & x = \bot
\end{cases} \\
\text{sq}(x) &= \begin{cases} 
x^2 & x \in \mathbb{Z} \\
\bot & x = \bot
\end{cases}
\end{align*}
\]

Exercise 2: Liveness vs. True Liveness

Consider the following program fragment:

\[
\begin{align*}
x &= 42; \\
r &= x + 99; \\
i &= 0; \\
y &= 0; \\
r &= 1; \\
\text{while}(i < 3) \{ \\
&\quad r = r*n; \\
&\quad y = y + 5; \\
&\quad i = i + 1;
\}
\end{align*}
\]

The corresponding control flow graph looks as follows:
The program fragment computes \( n^3 \) for a given \( n \) and returns the result in the variable \( r \).

1. Give the constraint system for calculating the live variable sets. Assume the variable \( r \) to be live at the end of the program since it contains the result of the computation and may be read afterwards.

2. Give a solution to the constraint system. Use your solution to apply transformation 2 to the program and give either the resulting CFG or the source code.

3. Give the constraint system for calculating the truly live variable sets. Again assume \( r \) to be live at the end of the program.

4. Give a solution to the constraint system for true liveness. Again use your solution to apply transformation 2 to the program and give either the resulting CFG or the source code.

5. Compare the three programs by counting the number of operations executed. One operation is a binary operation \((+,-,\ast,/)\), an assignment to a variable, or an comparison \((=,\neq,\leq,\geq)\). The statement \( x = y+1 \) counts as two operations.

6. Instead of using the constraint system for true liveness, one could run the standard liveness analysis multiple times. Would that approach yield the same result as the true liveness for the given program? If not, which of the assignments would not be removed? Reason your answer.