Christoph Kessler

DF00100
Advanced Compiler Construction
Spring 2023

Exercise Set 1

Exercise 1.1

Given the following C module:

```
#define NLIMIT 128
int A[NLIMIT];
int ntop = 0;  // largest tabulated A entry so far
extern void errormsg( void );

int f( int n )
{
 int i;
 if (n<1 || n>=NLIMIT) {
    errormsg();
    return 0;
 }
 if (n>ntop) {
    for (i=ntop+1; i<=n; i++)
       A[i] = A[i-1] + i;
    ntop = n;
 }
 return A[n];
}
```

1. Give the symbol table of the program and the abstract syntax tree of function `f`.

   Use the notation of the slides and complement with your own notation where necessary.

2. Give a HIR representation of function `f`.

   (Hint: You can use short-circuit evaluation of the `||` operator).

3. Give a MIR representation of function `f`.

4. Give a LIR representation of function `f`.

5. Identify the basic block leaders in the MIR representation of `f` and draw the basic block graph and the flow graph of `f`. Assume here for simplicity that function calls do not delimit a basic block.

6. Compute the dominators and immediate dominators for all nodes in the flow graph of `f`, using the iterative algorithm. Draw the dominator tree of function `f`.

Exercise 1.2

Given the following basic block in MIR code:

```
c ← a + 1
b ← 2 * a
b ← b + c
d ← c
a ← b / d
d ← b / c
c ← a - b
a ← 2 * a
```

Perform local common subexpression elimination to obtain a DAG representation of the code.

Exercise 1.3

We visualized the *dominance* relation on the slides with the light flow analogy. What would be the corresponding setup for the *postdominance* relation?

Recall that the postdominance relation is defined as follows:
p *postdominates* b (p pdom b) if every possible execution path $b \rightarrow^* \boxed{\text{exit}}$ includes p.

Exercise 1.4

Given the following MIR code

$i \leftarrow$ From
$s \leftarrow 0$
$t_0 \leftarrow$ (From $\leq$ To)
if not t_0 **goto** $L0$
L1: $t_1 = i \times Int\_Size$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow [t_2]$ (memory read)
$t_4 \leftarrow (t_3 < 0)$
if t_4 **goto** $L2$
$t_5 \leftarrow a \times 0.8$
$t_6 \leftarrow s + t_5$
$s \leftarrow t_6$
goto $L3$
L2: $t_7 \leftarrow a \times 0.8$
$t_8 \leftarrow s - t_7$
$s \leftarrow t_8$
L3: $t_9 \leftarrow i \times Int\_Size$
$t_{10} \leftarrow E + t_9$
$[t_{10}] \leftarrow s$ (memory write)
$t_{11} \leftarrow i + 1$
$i \leftarrow t_{11}$
$t_{12} \leftarrow (i \leq$ To)
if t_{12} **goto** $L1$
L0: **return** s

1. Draw the control flow graph, find the basic block leaders, and draw the basic block graph and the flow graph.

2. Compute the dominators of all nodes in the flow graph using the iterative algorithm.

Compute the immediate dominators and draw the dominator tree.

3. Do a DFS traversal of the flow graph, starting with the entry node, and label the edges with T, F, B and C.

4. Identify strongly connected components and natural loops in the code. Is the flow graph reducible? If yes, draw the region hierarchy graph, following the structural analysis transformations suggested in the lecture on control flow analysis.

   Give names to the variable definitions and expressions that occur in the code [Example: d_1 denotes the first assignment to i, and e_1 denotes the comparison (From$\leq$To)].

5. Formulate the flow functions and the dataflow equations for the Available Expressions problem for each basic block.

6. Give a trivial, but correct solution to the Available Expressions problem.

7. Determine a fixed-point solution of the Available Expressions problem with the worklist algorithm. Explain whether your solution is a maximal safe solution or not.

8. Formulate the flow functions and the dataflow equations for the Reaching Definitions (May-Reach) problem for each basic block.

9. Give a trivial, but correct solution to the Reaching Definitions problem.

10. Determine a fixed-point solution of the Reaching Definitions problem with the worklist algorithm. Explain whether your solution is a maximal (in the information-theoretic, not set-theoretic sense) safe solution or not.

Please prepare exercises 1.1–1.4.4 for the first lesson, exercises 1.4.4–1.4.10 for the second lesson.