Quiz 1, CS6013

Maximum marks = 80, Time: 80 min

23-Feb-2024

Read all the instructions and questions carefully. You can make any reasonable assumptions that you think are necessary; but state them clearly. There are total five questions, totalling 80 marks. You will need approximately 10 minutes for answering a 10 marks question (plan your time accordingly). For questions with sub-parts, the division for the sub-parts are given in square brackets. Start each question on a new page. The maximum space allowed for any question is specified at the beginning of the question.

1. [20 marks, 2 pages] **Control Flow Analysis.** Explain Visleshan (an imaginary student of CS6013) control-flow analysis by considering the following C code and drawing the CFG [10 marks] and control-tree [10 marks] for the code.

```
int foo(){
  int i, j, t, cnt;
  cnt = 0;
  for (i=0;i<n;++i) {
    for (j=0;j<n;++j) {
      if (A[i] < A[j]) {
        t = A[i];
        A[i] = A[j];
        A[j] = t;
        cnt++; } } }
  return cnt;
}
```

2. [20 marks, 2 pages] **Dominator Analysis.** Saralrekha (an imaginary student of CS6013) used the following changes in the dominator computation algorithm:

   - The repeat-until loop is executed exactly once.
   - The basic-blocks in the CFG are processed in the topological sort order; the blocks in a cycle (if present in the CFG) are processed in any order.

   Draw an example CFG, where the above scheme will compute the *right set* of dominators for each node (show the computed dominators). [10 marks]

   Draw an example CFG, where the above scheme will compute *an incorrect* set of dominators for each node (show the computed dominators). [10 marks]

3. [20 marks, 2 pages] **Control-flow optimizations.** Explain Sudharak (an imaginary student of CS6013) control-flow optimizations by writing example C codes, where the following control-flow optimizations may be applied; also show the optimized code: (i) unreachable code elimination, (ii) Loop inversion, (iii) If simplification, (iv) code straightening.

4. [10 marks, 1 page] **Lattices and Monotonic functions.** Bhramita (another imaginary student of CS6013) is confused about monotonic functions. Can you help her? Consider only functions of the form $f : L \to L$, where L is the three-bit-vector lattice.

   - Draw the three-bit-vector lattice [1 mark] and give the definition of a monotonic function [1 mark].
   - Give two functions that are monotonic. [2 x 2 marks]
   - Give two functions that are not monotonic. [2 x 2 marks]

5. [10 marks, 1 page] **Potpourri.** Can you help Khichdi (an imaginary student of CS6013) by answering these following true/false questions. Wrong answers have negative marks (-0.5 marks each).

   (a) Every lattice must have at least two elements.
   (b) Each lattice must have finite elements.
   (c) While computing reaching definitions, the minimum size of a KILL set = 0.
   (d) The number of nodes of a control tree can be greater than the number of nodes of the CFG.
   (e) In a CFG it is possible to have an unreachable node but having both predecessors and successors.
   (f) An extended basic block can have multiple exit points.
   (g) Every loop in a CFG must have a single back edge.
   (h) There is a unique IR for each high level language.
   (i) In C language, the multidimensional arrays are internally allocated as a single dimensional array.
   (j) In Java assignment statements, the LHS and RHS must have the same static type.