Quiz 1, CS6013

Maximum marks = 45, Time: 50 min

06-Sep-2021

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 three questions, totalling 45 marks (+ 5 bonus marks). You will need approximately 15 minutes for answering a 15 marks question (plan your time accordingly). The bonus part may take more time. For questions with sub-parts, the division for the sub-parts are given in square brackets.

Start each question on a new page (and write your roll number on each page – both sides of the sheet, that is). Think about the question before you start writing and write briefly. Each question also specifies the maximum number of allowed pages (A4 size) for the question. If the answer for any question is spanning more than specified number of pages, we will strictly ignore the spill-over text. If you scratch/cross some part of the answer, you can use space from the next page.

1. [15 marks, 2 pages] **IR generation:**
   Consider the three-address code discussed in the class. Give a scheme to translate the statements derived by the production rules shown below:

   ```
   S ::= switch (expr) {CaseStatements}

   CaseStatements ::= case <INTEGER>: S; CaseStatements
                    | ; // empty statement

   S ::= id = expr;

   S ::= {S S}
   ```

   Can you propose a scheme to generate efficient code for the switch statement? [Bonus 5 marks].

2. [15 marks, 2 pages] **Uninitialized variables**:
   In languages like `bash` shell scripting language, we don't have to declare a variable. But the variable must be initialized before use. Assume that only the following types of statements are present: i) copy statement (for example, x = integer-constant, x = string-literal, or x = y, or x = y == z), ii) if-statement (of the form `if (x) Statement`), iii) loops (of the form `while (x) Statement`). In addition, say we want to ensure that if a variable is used to hold a value of certain type, it is not used to hold a value of different type.

   Write an algorithm to i) identify uses of uninitialized variables [7.5], ii) type check the assignment statements [7.5].

3. [15 marks, 2 pages] **Data Flow Analysis:** Answer any three of the four questions.

   (a) [5 marks] Chandrakant (an imaginary student of CS6013) used the following equations for computing reaching definitions:
   $IN(n) = \cup_{x \in pred(n)} OUT(x)$
   $OUT(n) = (GEN(n) \cup IN(n)) - KILL(n)$

   Would he lead to correct reaching definitions? If not - give an example program where the reaching definitions would be incorrect. If yes - give a proof of equivalence to the equations discussed in the class.

   (b) [5 marks] Bharati (another imaginary student of CS6013) used the constant propagation algorithm to identify constants in a programming language that supports:

   - Classes and Objects like Java, but they cannot contain any methods.
   - All fields are public scoped and can be only of integer type.
   - No conditional or looping constructs available.
   - A single main function to contain all the code.
   - No recursion allowed.
   - The fields are initialized to zero when the object is allocated.

   Bharati has a simple trick: she proposes to replace (internally in the compiler) each field access of the form `x.f` to a variable of the form `x_f` and run the constant propagation algorithm studied in the class. Would she obtain the constants as expected or not? If not - give a counter example. If yes - give a proof/argument supporting your answer.

   (c) [5] Draw the constant propagation lattice to identify string-constants.

   (d) [5] Say, we have run the data-flow analysis and obtained the IN and OUT sets for each basic-block b. How to compute the same for each statement inside a given block b?