Quiz 1, CS6013

Maximum marks = 50, Time: 50 min

26-Sep-2022

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 six questions; answer any five. Max marks = 50. 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 (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. [10 marks, 1 page] **Reaching Definitions:**

   (a) Assume that for a given procedure, we have computed the reaching definitions for all the variables. How to use this information to identify useless assignments? [5 marks]
   For example, in the C code shown, the first assignment of `x` in the line `L1` is useless and can be removed.

```
x = 5;// L1
y = 6;
if (cond) {x = 5; y = 7;}
else x = 6;
z = x + y; ...
```

   (b) In the code shown, for each definition, identify the statements where the definition reaches. [5 marks]
   State your answer as a table of the form:

Definition at Line	Reaches statement
...	...
...	...
...	...

```
L0: x = 3;
L1: y = 4;
L2: if (cond) {
L3:   x = 5 + x;
L4:   y = x - 3;
L5: } else {
L6:   x = 6 + y;
L7: }
L8: z = x + y;
L9: printf (z);
```

2. [10 marks, 1 page] **IR generation:**
   Consider the three-address code discussed in the class. Give a scheme to translate the do-while statement derived by the production rules shown on the right. Assume that the rules for assignment statements, sequence etc. are already available as discussed in the class.

```
S ::= do  {S} while (expr);

S ::= break;
```

3. [10 marks, 1 page] **Three Address Code**:

   For the C code shown, write the equivalent three-address-code that can be generated. Use the three-address-code instructions discussed in the class.

```
if (x + y < z){
  while (x++ < 100) {
    x += y*z;
  } }
p = x * y * z;
```

4. [10 marks, 1 pages] **Directed Acyclic Graph**

   Draw the DAG for the code shown on the right.

```
x = 5;
y = x + 6;
y = y + 1;
z = x + y;
x = 1;
y = x + z;
```

5. [10 marks, 1 page] **Type checking:**

   Give the rules to type check the Java statements of the form shown:

   - Copy statements of the form x = y;
   - Function calls of the form x = a.foo(b);

6. [10 marks, 1 page] **Syntax Tree:**
   Consider the CFG shown on the right. Draw the syntax-tree for the expression shown below, when parsed using the same grammar.

```
if (x) {
  z = y;
} else {
  if (z) p = q;
}
```

```
S ::= IfStmt | IfElseStmt | AssignStmt | {S}
IfStmt ::= If (Id) S
IfElseStmt ::= If (Id) S else S
AssignStmt ::= Id = Id;
```