

CS3300 Quiz II

Dept of CSE, IIT Madras

Total marks = 24+3

19 Oct 2016

Read 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 (8 marks each) + two bonus questions (3 marks). You will need approximately 15 minutes for answering an 8 marks question (plan your time accordingly). For questions with sub-parts, the division for the sub-parts are given in square brackets. Attempt the bonus question (marks disproportional to the time needed) at the end.

You will get an answer sheet with 8 pages (if you get a answer sheet with fewer pages then ask for a replacement sheet). Leave the first page empty and start from Page#2. Start each question on a new page. Think about the question before you start writing and write briefly. **For any question, the answer (including the answers for all the sub-parts) should NOT cross more than two pages.** If the answer for any question is spanning more than two 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. You mostly would NOT need any additional sheets.

1. [8] **Syntax Directed Translation:** Briefly state the difference between SDD and SDT scheme [1]. Give a grammar to recognize binary numbers [1]. Give a SDD to check if a given binary number is even [2]. Give a SDD to check if a given binary number is divisible by three [4]. Do so (div by 2 check, div by 3 check), without converting the binary to decimal.

2. [8+2] **IR Generation:**

Give a list of instructions of 3-address-codes (TAC), with one line description for each [3]. Note: Use 'bOP' to denote generic binary operator and 'uOP' to denote generic unary operator. Translate the following C code to TAC [5].

```
int foo(int n){
    do {
        switch (n%3) {
            case 0:
                a[i][j++] = n++; break;
            case 1:
                a[i++][j] = ++n; break;
        }
    } while ( n < 100);
    return n;
}
```

Bonus [1] : Assume my TAC does not admit pointers or structs, but admits array notation: write the TAC translation for the instructions in the function `bar`.

```
struct A {
    int i;
    int j;
};
void bar(int *q){
    struct A x;
    x.i = 5;
    *q = x.j;
}
```

3. [8] **Control Flow Analysis** Define a control-flow-graph (CFG) [1]. What is the minimum number of nodes in a any CFG? [1] Draw the CFG for the following piece of C code:

```
void fubar(){
    int i = 0;
    do {
        for (;i<n;++i){
            if (i > 3) if (i < 8) n=n*2; else n=n/2;
            n+=2;
        }
    } while (n > i);
}
```

4. [2] **Bonus: Liveness analysis:** How to extend the liveness analysis algorithm discussed in the class to use basic-blocks (not just one instruction per basic-bloc, but maximal sequence of single-entry-single-exit blocks).