

Roll No: _____

CS6843 Program Analysis at IIT Madras
EndSem July 4, 2014

Total Marks: 25

Number of questions: 9 **compulsory** questions

Duration: 120 minutes

Marking: Q1 carries 1 mark, all others carry 3 marks each

1. Write a suggestion to improve one aspect in the department.

2. For the following code, write down the equations in $Ax \leq B$ matrix format for achieving parallelization, that is, satisfying true dependence.

```
for (i = 4; i <= N; ++i)
    a[i - 3] = a[2 * i + 3] - 4;
```

3. A barrier ensures that all the participating threads reach a program point before any one of them can execute the next instruction. Can you think of a reason to break this definition in a sound way? That is, even though some threads may not have reached a barrier, some threads that have reached the barrier are allowed to execute their post-barrier instructions, without changing the functionality of the original program with barriers.

4.

(a) Write the weakest pre-condition for $x = y * 2$ where the post-condition is $\{x < 10\}$. [1]

(b) Write the strongest post-condition for `if (y == x) then y = ++x; else x = y;` where the pre-condition is $\{y > 2 \text{ and } x > 5\}$. [2]

5. Design an analysis to check for dangling pointers for C programs. A pointer is dangling if it points to a freed or unallocated memory. Make suitable assumptions (and mention those).

6. Compute a backward slice of the following program for criteria <11, sum> where 11 is the line number. Don't write how you got the slice. Write/mark only the statement numbers.

```
1: void main() {
2:     int sum = 0;
3:     int i = 0;
4:     int j = 1;
5:     while (i < N) {
6:         i = ++j;
7:         sum += i;
8:         i = sum / 2;
9:     }
10:    if (sum >= 100)
11:        printf(“%d\n”, sum);
12:    else
13:        printf(“sum is less than hundred.\n”);
14: }
```

7. Develop an analysis to detect out-of-bounds array access.

