

Problem 3.1. (5 points) Which of the following statements are true?

1. 100% path coverage implies 100% branch coverage

True.

2. 100% statement coverage implies 100% branch coverage

False.

Example in lecture slides:

if (x != 0) x = x + 10;

y = y/x;

3. Suppose test suite T₁ gets 100% branch coverage and test suite T₂ gets 100% statement coverage. There can be an execution path explored in T₂ but not by T₁.

True.

4. There is a program such that no test suite can get 100% branch coverage.

True.

Any example with deadcode.

x = 10;

if (x != 10) y = x;

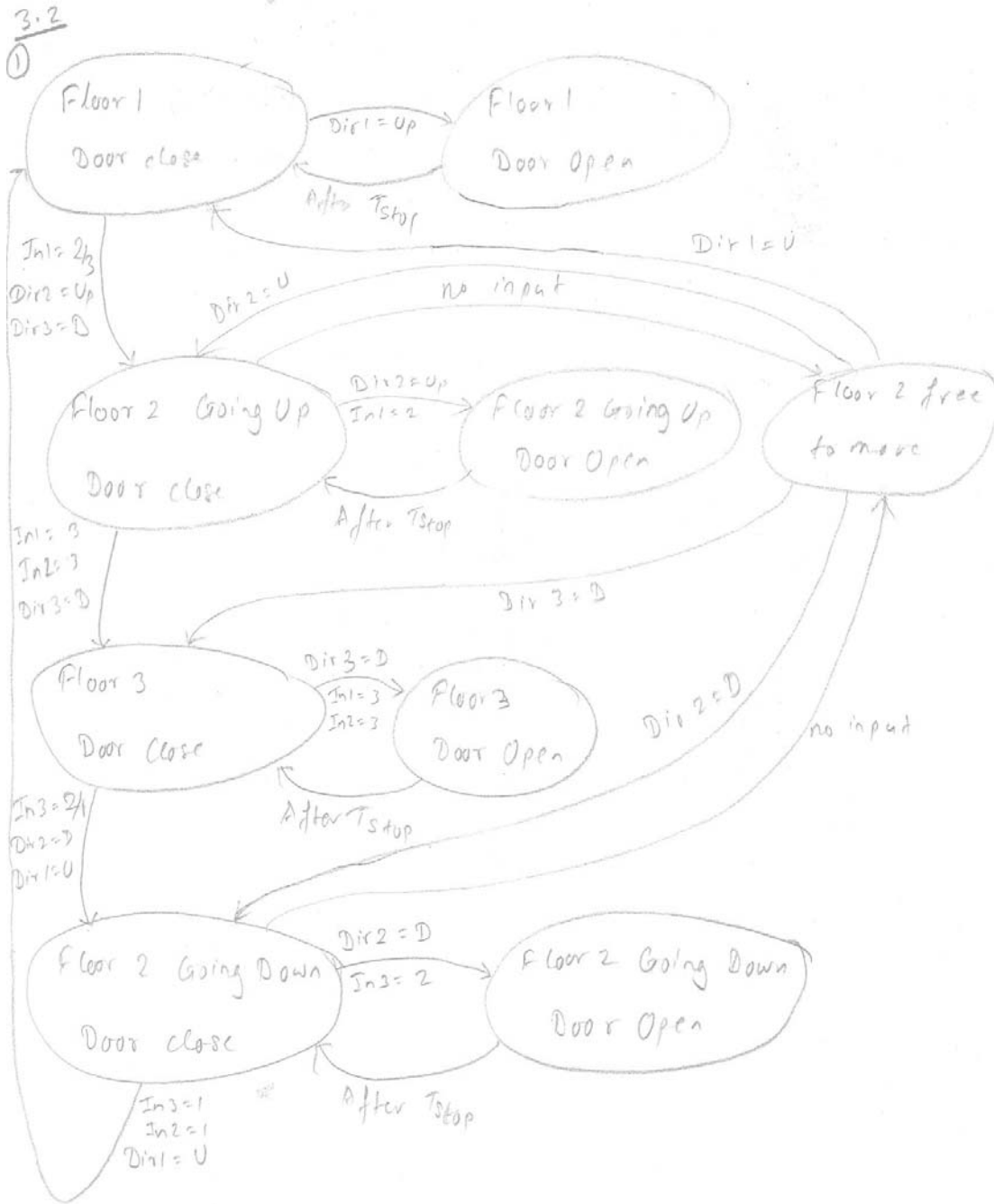
5. There are incomparable coverage metrics; i.e., there are two coverage metrics such that 100% coverage for metric 1 does not imply 100% coverage for metric 2 and vice versa.

True.

Dataflow coverage and branch coverage.

Problem 3.2.

1. Finite state machine ?



Dir1/2/3 → Request for Up or Down direction from Level 1, 2, 3 (Up or D)

In1/2/3 → Input given from level 1, 2, 3 (1, 2, 3)

2. Suggest some test coverage metrics for the finite state machine programming model. Can you get 100% coverage on these metrics for your elevator controller? **Two possible metrics are state coverage and path coverage. It is possible to get 100% coverage both the metrics for this example.**

Problem 3.3. (5 points) Here is the binary search code that was implemented in Java's JDK and used widely for many years:

```
1: public static int binarySearch(int[] a, int key) {
2: int low = 0;
3: int high = a.length - 1;
4:
5: while (low <= high) {
6: int mid = (low + high) / 2;
7: int midVal = a[mid];
8:
9: if (midVal < key)
10: low = mid + 1
11: else if (midVal > key)
12: high = mid - 1;
13: else
14: return mid; // key found
15: }
16: return -(low + 1); // key not found.
17: }
```

(a) Find tests to achieve 100% branch coverage.

a = {0, 1, 2, 3, 4}... key = 1

a = {0, 1, 2, 3, 4}... key = 5

(b) This code has a bug. Can you find a test that demonstrates the bug?
Case where array length is greater than integer bounds.