

## Dependability- TD 1- Threats

### Exercise 1 :

We consider a hot drinks dispenser which allows drinks to be delivered and change to be given.

1. Classify these observations:
  - a) Makoccino selection is no longer possible for all users
  - b) The dispenser is too slow
  - c) The dispenser doesn't give anyone change
  - d) The distributor is broken then yesterday
  - e) Some users consider Espresso to be a bit light.
  - f) The distribution of iced tea is not possible between 8 a.m. and 9 a.m.
  - g) 50DA coins are no longer accepted.
  - h) Change is not given.



2. Classify these Faults or Errors:
  - a) Resource exhausted (Goblet) without message.
  - b) Verbena button is clogged
  - c) A communication link loses the message
  - d) A cable is disconnected
  - e) The cable is worn.

### Exercise 2

What is the expected behavior of the following systems? Give examples of failures?

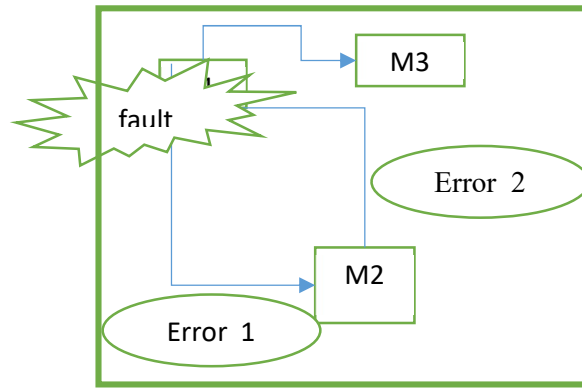
- a) Bulb
- b) Printer



### Exercise 3:

Consider the following structured system of 3 machines:

- a) Represent levels of error contamination and failure.
- b) calculate latency by sum of local latencies: fault at time 0, ; L1 latency in M1, propagation through M2, M3 L2 and L latency 3 and finally output assignment.



L1=10ms, L2=100ms, L3=30ms.

### Exercise 4:

Consider an asynchronous binary counter of 16 which counts pulses arriving at input E and displays the result on S in natural binary code on 4 bits a ,b,c,d . A lasting failure of sticking to 0 of the weight output a occurs while the counter is in the initial state 0000. We therefore evolve with setting 0 ,b,c,d .

Calculate the average latency of this fault knowing that the average time interval between two input events is 2 milliseconds.

### Exercise 5 (optional):

Or the incorrect program below. For  $x = [-4,2,0,2]$  , this program causes a failure (expected result: 2).

- a) Identify the fault.
- b) If possible, identify a test case that does not execute the fault.
- c) If possible, identify a test case that executes the fault, but does not result in an error state.
- (d) If possible, identify a test case that results in an error, but not a failure.
- e) For the given test case, identify the first error state and describe it
- f) Correct the error and verify that the given test now produces the expected result.

```
// Effects: if x==null throw NullPointerException
// else return the number of positive elements in x
public static int countPositive (int [] x) {
    int count = 0;
    for (int i=0; i < x.length; i++) {
        if (x[i] >= 0) {
            count++;
        }
    }
    return count;
}
```