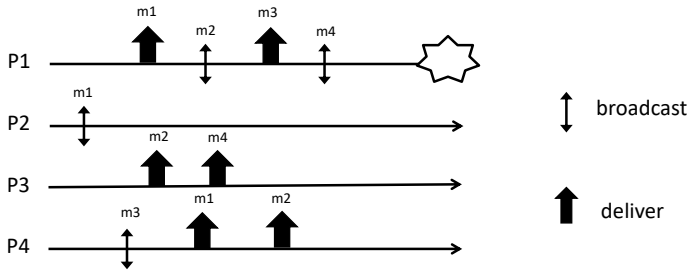
Dependable Distributed Systems
Master of Science in Engineering in Computer Science

AA 2023/2024

Week 6 – Exercises
November 1st, 2023

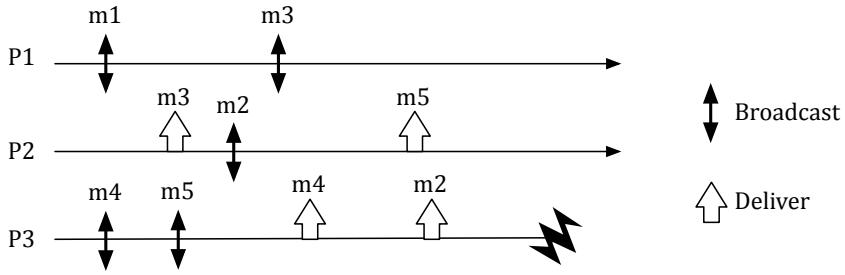
Ex 1: Consider the execution depicted in the Figure



Answer to the following questions:

1. Provide all the delivery sequences that satisfy both causal order and total order;
2. Complete the execution in order to have a run satisfying TO(UA, WNUTO), FIFO order but not causal order.

Ex 2: Consider the partial execution shown in the Figure and answer to the following questions:



1. Complete the execution to have a run satisfying the Regular Reliable Broadcast specification but not Uniform Reliable Broadcast one.

2. For each process, provide <u>ALL</u> the delivery sequences satisfying FIFO Reliable Broadcast but not satisfying causal order.

3. For each process, provide <u>ALL</u> the delivery sequences satisfying total order and causal order.

Ex 3: Consider a distributed system composed of N processes p_1, p_2, … p_N, each having a unique identifier myID. Initially, all processes are correct (i.e. correct={ p_1, p_2, … p_N }). Consider the following algorithm:

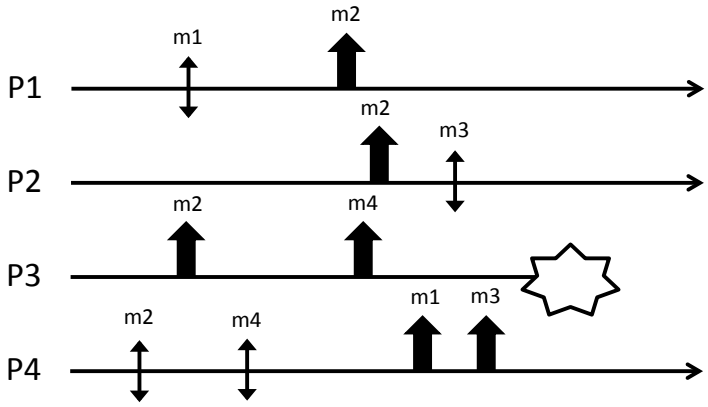
```
upon event Xbroadcast(m)
        mysn = mysn+1;
        ∀ p ∈ correct
                pp2pSend ("MSG", m, mysn, myId);

upon event pp2pReceive ("MSG", m, sn, i)
       mysn= mysn+1;
       if (m ∉ delivered)
          trigger XDeliver (m);
          delivered = delivered ∪ {m};

upon event crash (pi)
        correct = correct / {pi}
```

Let us assume that: (i) links are perfect, (ii) the failure detector is perfect and (iii) initially local variables are initialized as follows mysn=0 and delivered =∅.
Answer to the following questions:

1. Does the `Xbroadcast()` primitive implement a Reliable Broadcast, a Best Effort Broadcast or none of the two?
2. Considering only the ordering property of broadcast communication primitives discussed during the lectures (FIFO, Causal, Total), explain which ones can be satisfied by the `Xbroadcast()` implementation.

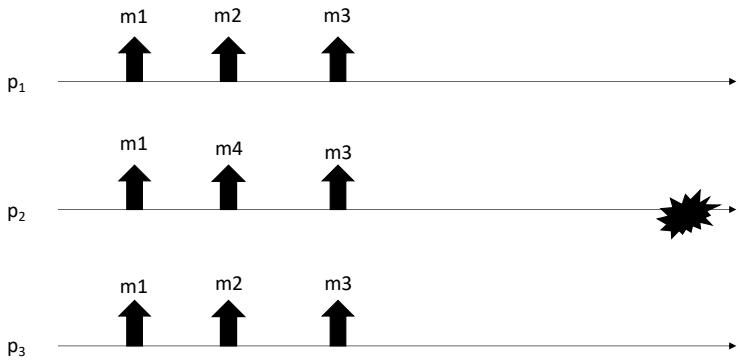
Provide examples to justify your answers.

Ex 4: Consider the message pattern shown in the Figure below and answer to the following questions:



1. Complete the execution in order to have a run satisfying Reliable Broadcast but not Uniform Reliable Broadcast.
2. Provide all the delivery sequences satisfying causal order and total order.
3. Provide all the delivery sequences violating causal order and satisfying TO(UA, WNUTO) but not satisfying TO(UA, SUTO)

Ex 5: Consider the partial execution in the following figure



Given the run depicted in the figure state the truthfulness of the following sentences:			
a	The strongest agreement property satisfied is UA	T	F
b	The NUA agreement property is violated	T	F
c	The strongest ordering property satisfied is SUTO	T	F
d	The WUTO ordering property is satisfied	T	F
e	The SNUTO ordering property is violated	T	F
f	Let us assume we can add only one more delivery to p1 and p3, it is not possible to get a run satisfying TO(NUA, SUTO)	T	F
g	If p2 is not going to deliver m4 then the strongest specification satisfied by the resulting execution is TO(UA, SUTO)	T	F
h	Let us assume we can add only one more delivery to p1 and p3, it is possible to get a run satisfying TO(UA, WNUTO) but not satisfying TO(UA, WUTO)	T	F
i	If p2 is not faulty, the NUA agreement property is satisfied	T	F
l	If p2 is not faulty, the SUTO ordering property is satisfied	T	F

For each point, provide a justification for your answer

Ex 6: Consider a distributed systems composed by a set of *n* processes p_1, p_2,… p_n. Processes have a unique identifier and are structured as a binary tree topology. Messages are exchanged between processes over the edges of the tree which act like perfect point-to-point links. Each process p_i has stored the identifiers of its neighbors into the local variables FATHER, R_CHILD e L_CHILD representing respectively the father of p_i, the right child and the left child (if they exists).
Assuming that processes are not going to fail, write the pseudo-code of an algorithm satisfying the following specification:

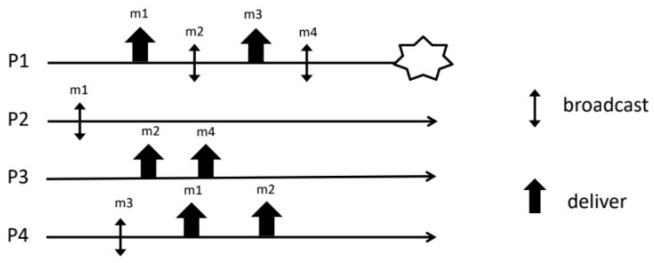
Events:

- **Request:** ⟨ *tob*, Broadcast | m ⟩: Broadcasts a message m to all processes.
- **Indication:** ⟨ *tob*, Deliver | p, m ⟩: Delivers a message m broadcast by process p.

Properties:

- *Validity:* If a correct process *p* broadcasts a message *m*, then *p* eventually delivers *m*.
- *No duplication:* No message is delivered more than once.
- *No creation:* If a process delivers a message *m* with sender *s*, then *m* was previously broadcast by process *s*.
- *Agreement:* If a message *m* is delivered by some correct process, then *m* is eventually delivered by every correct process.
- *Total order:* Let m_1 and m_2 be any two messages and suppose *p* and *q* are any two correct processes that deliver m_1 and m_2. If *p* delivers m_1 before m_2, then *q* delivers m_1 before m_2.

Ex 1: Consider the execution depicted in the Figure



Answer to the following questions:

1. Provide all the delivery sequences that satisfy both causal order and total order;
2. Complete the execution in order to have a run satisfying TO(UA, WNUTO), FIFO order but not causal order.

1) Casual and total

$m_1 \rightarrow m_3$

$m_2 \rightarrow m_4$

FIFO + local : $m_2 \rightarrow m_4$

$m_3 \rightarrow m_4$

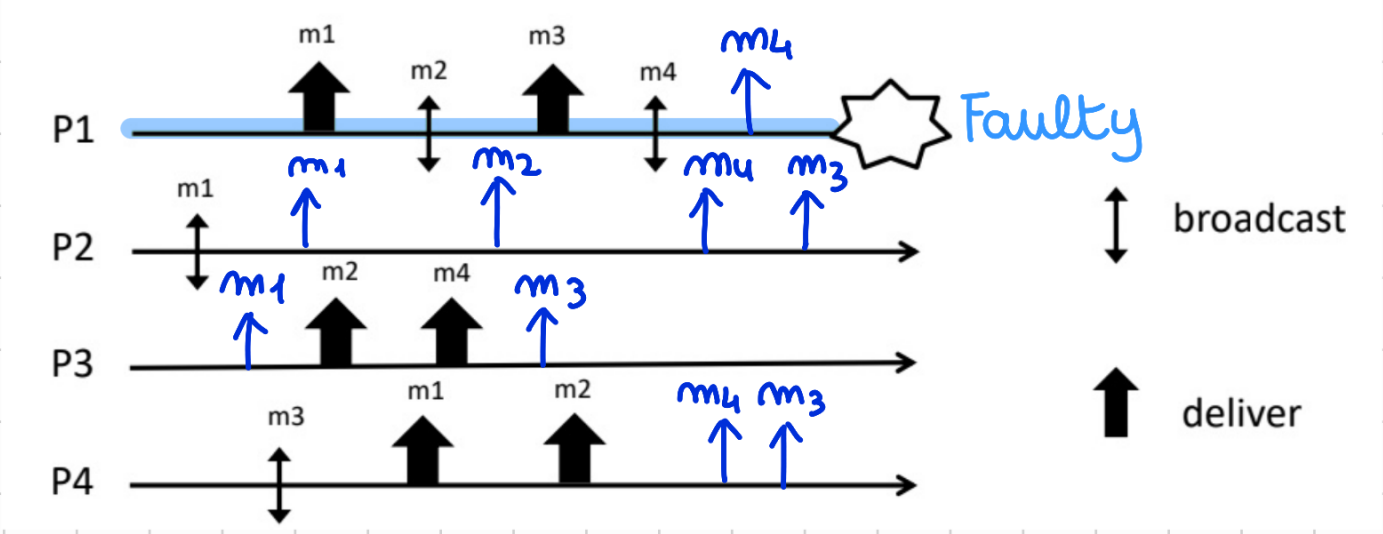
$m_1 \rightarrow m_2$

TOTAL: same $m_1 \rightarrow m_2$ sequence for correct processes

Sequences: $m_1\ m_2\ m_3\ m_4$

$m_1\ m_3\ m_2\ m_4$

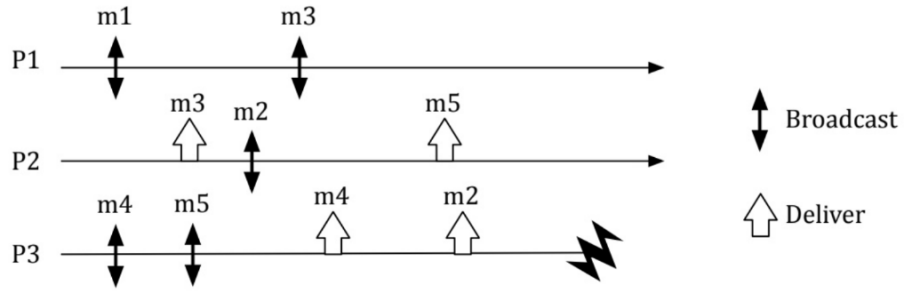
2) TO(UA, WNUTO), FIFO order and NOT CASUAL



FIFO: $m_2 \rightarrow m_4$

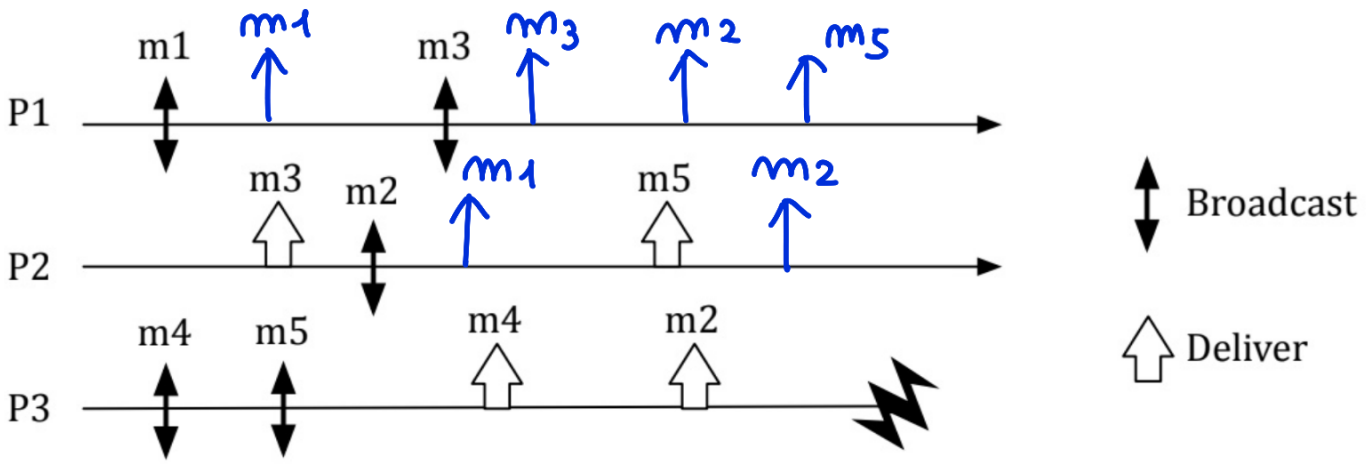
In the faulty we can tolerate 1 missing: in p_1 no m_2

Ex 2: Consider the partial execution shown in the Figure and answer to the following questions:



1. Complete the execution to have a run satisfying the Regular Reliable Broadcast specification but not Uniform Reliable Broadcast one.
2. For each process, provide ALL the delivery sequences satisfying FIFO Reliable Broadcast but not satisfying causal order.
3. For each process, provide ALL the delivery sequences satisfying total order and causal order.

1) RB not URB

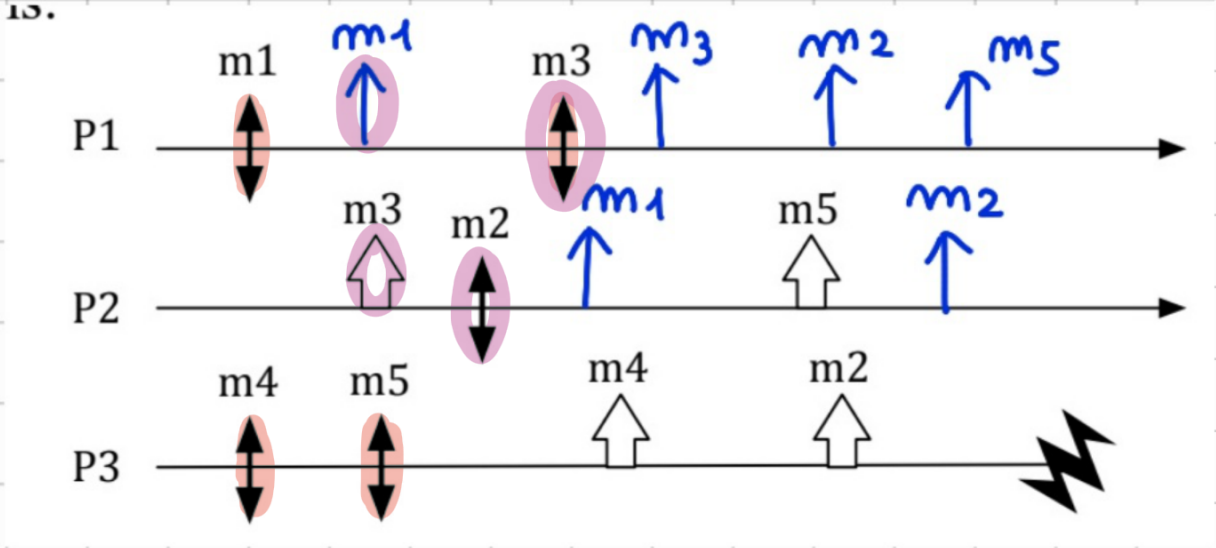


RB: correct same set ; not URB: faulty not subset of correct

$\boxed{m_1\ m_2\ m_3}\ m_5$ correct

$m_4\ m_2$ faulty

2) FIFO, not causal



Based on this execution we have:

FIFO: $m_1 \rightarrow m_3$ CAUSAL: $m_3 \rightarrow m_2$

$m_4 \rightarrow m_5$

Now for each process we have:

- p_1 have to deliver also m_4

Sequences:

1 2 3 4 5	2 1 4 3 5	4 1 5 2 3
2 1 3 4 5	2 4 5 1 3	4 1 2 5 3
4 5 2 1 3	2 4 1 5 3	4 2 5 1 3
4 1 2 3 5	2 1 4 5 3	4 2 1 3 5
1 2 4 3 5	1 4 5 2 3	4 2 1 5 3
1 2 4 5 3	1 4 2 5 3	1 4 2 3 5

- p_2 $\boxed{m_3 \rightarrow m_2}$ satisfy the causal order

?

- p_3 $\boxed{m_4 \rightarrow m_5}$ satisfy the FIFO

?

3) Total and causal

FIFO: $m_1 \rightarrow m_3$
$m_4 \rightarrow m_5$

CAUSAL: $m_3 \rightarrow m_2$

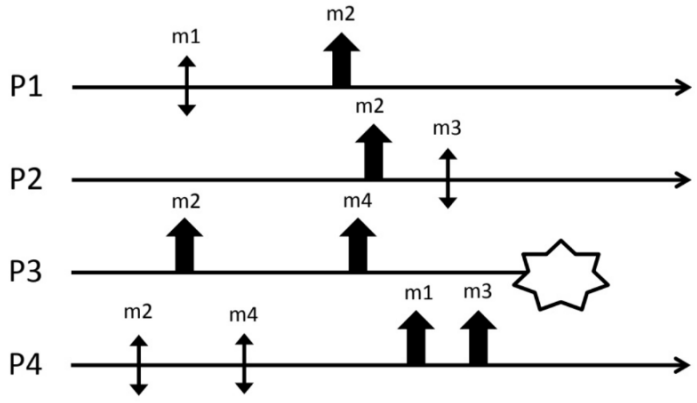
TOTAL: same sequence $m_4 \rightarrow m_2$ (p_3)
$m_3 \rightarrow m_5$ (p_2)

$m_1 \rightarrow m_3 \rightarrow m_2$
$m_4 \rightarrow m_5$
$m_4 \rightarrow m_2$
$m_3 \rightarrow m_5$

Sequences:

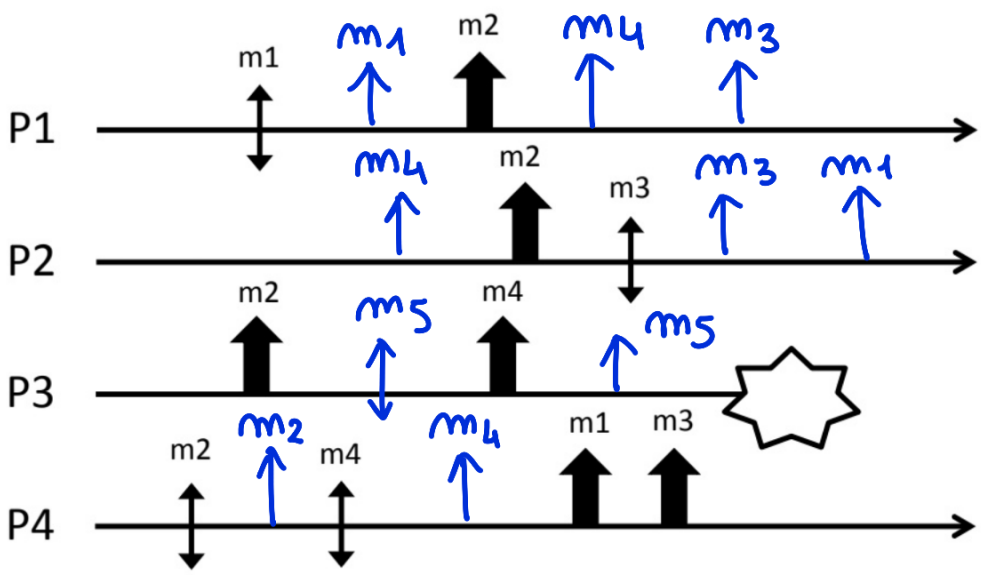
1 3 4 2 5
4 1 3 2 5
4 1 3 5 2
1 3 4 5 2
1 4 3 2 5

Ex 4: Consider the message pattern shown in the Figure below and answer to the following questions:



1. Complete the execution in order to have a run satisfying Reliable Broadcast but not Uniform Reliable Broadcast.
2. Provide all the delivery sequences satisfying causal order and total order.
3. Provide all the delivery sequences violating causal order and satisfying TO(UA, WNUTO) but not satisfying TO(UA, SUTO)

1) RB not URB



1243

4231

245

2413

m1 m3 m2 m4	correct

2) Causal and Total order

FIFO + local : $m_2 \to m_4$, $m_2 \to m_3$

TOTAL: $m_2 \to m_4$, $m_1 \to m_3$

Sequence:
$m_1\ m_2\ m_3\ m_4$
$m_1\ m_2\ m_4\ m_3$
$m_2\ m_1\ m_3\ m_4$
$m_2\ m_1\ m_4\ m_3$
$m_2\ m_4\ m_1\ m_3$

3) TO(UA, WNUTO) not CAUSAL, not TO(UA, SUTO)

FIFO + local : $m_2 \to m_3$, $m_2 \to m_4$, $m_1 \to m_3$

Sequences:
~~$m_1\ m_3\ m_2\ m_4$~~
~~$m_1\ m_3\ m_4\ m_2$~~
$m_4\ m_1\ m_2\ m_3$
$m_4\ m_2\ m_1\ m_3$
~~$m_4\ m_1\ m_3\ m_2$~~
$m_1\ m_4\ m_2\ m_3$

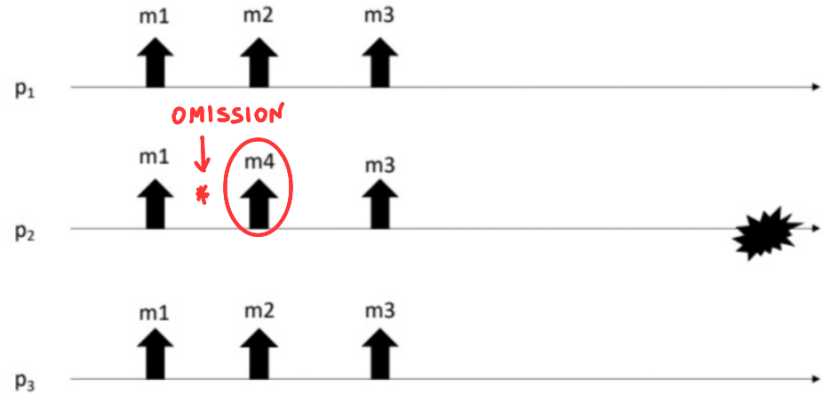
→ only in correct processes

Correzione Farina:
4.3) bisogna distinguere cosa succede sui processi corretti e faulty altrimenti potrebbero essere TO(UA, SUTO)

↓

I don't add a delivery in p_3 and without inserting $m_2 \to m_4$ in the correct

Ex 5: Consider the partial execution in the following figure



Given the run depicted in the figure state the truthfulness of the following sentences:			
a	The strongest agreement property satisfied is UA	T	F
b	The NUA agreement property is violated	T	F
c	The strongest ordering property satisfied is SUTO	T	F
d	The WUTO ordering property is satisfied	T	F
e	The SNUTO ordering property is violated	T	F
f	Let us assume we can add only one more delivery to p1 and p3, it is not possible to get a run satisfying TO(NUA, SUTO)	T	F
g	If p2 is not going to deliver m4 then the strongest specification satisfied by the resulting execution is TO(UA, SUTO)	T	F
h	Let us assume we can add only one more delivery to p1 and p3, it is possible to get a run satisfying TO(UA, WNUTO) but not satisfying TO(UA, WUTO)	T	F
i	If p2 is not faulty, the NUA agreement property is satisfied	T	F
l	If p2 is not faulty, the SUTO ordering property is satisfied	T	F

For each point, provide a justification for your answer

a) F, because in the faulty process there is a new delivery m_4 that it is not delivered by the correct and correct deliver the same set

b) F, because the faulty process can deliver something (m_4) that the correct have not delivered; order not respected and correct deliver the same set

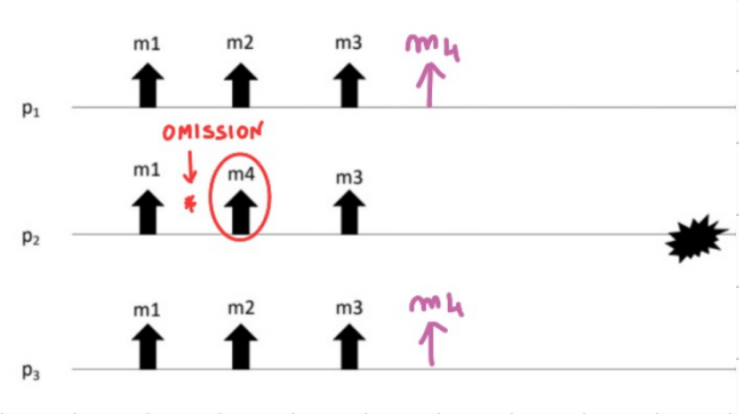
c) F, because m_4 is delivered before m_3

d) T, because the conditions that have to be

satisfied are $m_1 \to m_2$, $m_1 \to m_3$ and in p_2 are satisfied ($m_1 \to m_3$)

e) F, because faulty can do everything it wants and correct deliver the same sequence

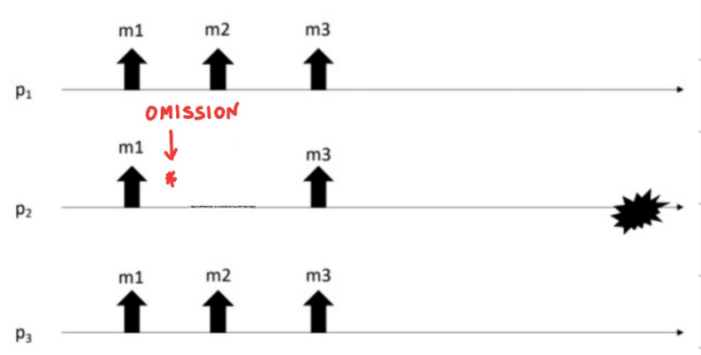
f) T, because if I add m_4, as in the figure, I have TO(UA, WNUTO)



$\Rightarrow m_3 \to m_4$ correct

$\Rightarrow m_4 \to m_3$ faulty

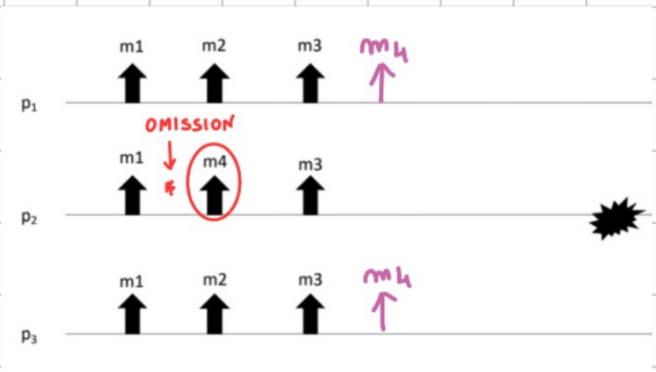
g) F, because p_2 have to deliver m_2 to satisfy SUTO



h) T, because it satisfies TO(UA, WNUTO) but not TO(UA, WUTO)

↓

p_2: we need to have $m_3 \to m_4$



i) F, because p_2 is correct and deliver m_4 that the other correct don't deliver

j) F, because m_2 is not delivered by p_2 (is a correct p.) and $m_4 \not< m_3$ in p_2 is a wrong deliver in terms of ordering.