Course: Secure and Dependable Systems
Assignment 3

Fezile Manana

Problem 3.1

a) (i) False

The definition of a logical clock is a function such that Θ

$$a \prec b \Rightarrow \Theta(a) < \Theta(b)$$

where a and b have a casual relation and a *happened-before* b. However, the direction of implication of this statement does not necessarily work in the opposite direction, therefore

$$\Theta(a) < \Theta(b) \Rightarrow a \prec b$$

is not necessarily true.

(ii) False

The same reasoning in (i) can be applied in this situation as the implication is in wrong direction

b) (i) True

Vector clocks are defined as such:

$$\Theta_v(a) < \Theta_v(b) \iff \forall i(1 \leqslant i \leqslant b) : a_i \leqslant b_i \land \exists i' : a_{i'} < b_{i'}$$

meaning that $\Theta_v(a) < \Theta_v(b)$ *iff* every value for all process indices in $\Theta_v(a)$ is less than or equal the corresponding values in vector $\Theta_v(b)$ **and** there exists an entry where $\Theta_v(a)_i$ is strictly smaller than $\Theta_v(b)_i$.

Therefore if $a \prec b$, the following characteristic emerges:

$$\Theta_v(b)_{min} = \Theta_v(a) - I_i$$

where I_i is the initial vector of some process i (i.e. 1 in one entry and zero everywhere else)

$$\therefore \Theta_v(b) < \Theta_v(a)$$

(ii) False

This is a contradiction to the previous statement.

c) If the value of $\Theta_v(e)$ increases for process index other than itself, then it is a concurrent process.

Problem 3.2

a)

a = 1	g = 1	n = 1
b = 2	h = 2	o = 3
c = 3	i = 3	p = 4
d = 4	j = 4	q = 5
e = 7	k = 5	r = 6
f = 10	l = 8	s = 7
	m = 9	t = 8

b)

a = (1, 0, 0)	g = (0, 1, 0)	n = (0, 0, 1)
b = (2, 1, 0)	h = (0, 2, 0)	o = (0, 2, 2)
c = (3, 1, 0)	i = (0, 3, 0)	p = (0, 2, 3)
d = (4, 3, 0)	j = (3, 4, 0)	q = (0, 2, 4)
e = (5, 3, 5)	k = (3, 5, 0)	r = (0, 2, 5)
f = (6, 7, 6)	l = (3, 6, 6)	s = (0, 2, 6)
	m = (3, 7, 6)	t = (3, 5, 7)

c) Cut C_1 is consistent as the cut C_1 has the property that if $f \in C_1$ and $e \prec f$ then $e \in C_1$ holds. C_2, however, is inconsistent as event l is in the cut while event s is not and it also holds that $s \prec l$, giving an inconsistent cut.

Problem 3.3

Building on the primitives in the lecture notes:

```
upon initialisation do
    delivered = ∅
    past = ∅

upon event ⟨BROADCAST(R,m)⟩ do
    trigger ⟨BROADCAST(R, (DATA, past, m)⟩
    past = past.append((p_i,m))     \\ append to history

upon event ⟨DELIVER(R,p_i[DATA,past_m,m)⟩ do
    if(m ∈ delivered then
        for all(s_n, n) ∈ past_m do     \\ in ascending order
            if(n ∈ delivered) then
                trigger⟨ DELIVER(R,(s_n,n))⟩     \\deliver preceding messages
                delivered = delivered.append(n)
                past = past.append((s_n,n))     \\ append to history
            trigger⟨ DELIVER(R,p_i,m))⟩     \\ deliver current message
            delivered = delivered.append(m)
            past = past.append((p_i, m))    \\append to history
```

Reference for source https://link.springer.com/content/pdf/10.1007%2F3-540-28846-5_3.pdf