CSI3344 Distributed Systems

Workshop Solution 07

Q1. The statement of causal consistency is:

Writes that are potentially casually related must be seen by all processes ***in the same order****. Concurrent writes may be seen in a different order on different machines.*

Given four processes P1, P2, P3 and P4, and their following 5 execution sequences, **explain** which sequence is causally consistent, and which one is not.

(a)

P1:	W(x)a			W(x)c		
P2:		R(x)a	W(x)b			
P3:		R(x)a			R(x)c	R(x)b
P4:		R(x)a			R(x)b	R(x)c

A: Casually consistent: There are two casual links a → b and a → c in P1 and/or P2. P3 and P4 read them by the same order, i.e., a before c and a before b.

(b)

P1:	W(x)a		W(x)c		
P2:		W(x)b			
P3:			R(x)a	R(x)c	R(x)b
P4:			R(x)b	R(x)a	R(x)c

A: Casually consistent: There is one casual link a → c in P1 only. P3 and P4 read them by the same order, i.e., a before c.

(c)

P1:	W(x)a					
P2:		R(x)a	W(x)b			
P3:				R(x)b	R(x)a	
P4:					R(x)a	R(x)b

A: Casually inconsistent: There is one casual link a $\rightarrow$ b in P1/P2 only. P4 reads them by the same order, ie, a before b, but P3 breaks this order.

(d)

P1:	W(x)a						
P2:		R(x)a	W(x)b				
P3:				R(x)a			R(x)b
P4:					R(x)a	R(x)b	

A: Casually consistent: There is one casual link a $\rightarrow$ b in P1/P2 only. P3 and P4 read them by the same order, i.e., a before b.

(e)

P1:	W(x)a					
P2:		W(x)b				
P3:			R(x)b	R(x)a		
P4:				R(x)a	R(x)b	

A: Casually consistent: There is no any casual link. P3 and P4 can read them in any order.

Q2. Consider the example given in lecture slide No.15: Although slide No.16 listed 4 possible execution sequences, there exist many other possible valid execution sequences for this example.
List 4 other valid execution sequences, and their corresponding sequences of Prints and Signatures for the example.

A: For each process, the first statement (e.g., x $\leftarrow$ 1; or y$\leftarrow$ 1; or z$\leftarrow$ 1;) must be executed before the 2nd statement. Based on this, we can list all possible execution sequences. The following four are among them (and they are different from those listed in Slide No.16).

x ← 1;
y ← 1;
z ← 1;
print(x, z);
print(x, y);
print(y, z);
 	
Prints: 111111
Signature:111111
	x ← 1;
	y ← 1;
	z ← 1;
	print(x, y);
	print(x, z);
	print(y, z);
	
	Prints: 111111
	Signature:111111
		x ← 1;
		y ← 1;
		z ← 1;
		print(x, y);
		print(y, z);
		print(x, z);
		
		Prints: 111111
		Signature:111111
			x ← 1;
			y ← 1;
			z ← 1;
			print(y, z);
			print(x, y);
			print(x, z);
			
			Prints: 111111
			Signature:111111

The following questions are *Optional*:

Q3. Continue with Q2, work out all other valid execution sequences for this example, and list all other possible Prints and Signatures.

Does the number of the Prints equal to the number of the Sequences in this case?

A: (As listed during the lecture session, there are altogether 90 valid execution sequences, and this is left for your exercise)

<u>END OF THE WORKSHOP SOLUTION</u>