VAIBHAV BHAGEE
2014CS50297

COL - 830
Homework 1

Ans. 1 To show,

$$H \neq H' \implies TS_1 \|_H (TS_2 \|_{H'} TS_3) \neq (TS_1 \|_H TS_2) \|_{H'} TS_3$$

We will prove the contrapositive of this statement

$$TS_1 \|_H (TS_2 \|_{H'} TS_3) = (TS_1 \|_H TS_2) \|_{H'} TS_3$$

$$\implies H = H'$$

Let $Act(TS)$ represent the action set of a transition system TS.

Let $Act(TS_i) = A_i \quad \forall i \in \{1, 2, 3\}$

From the LHS for H' we get
$H' = A_2 \cap A_3$.
From the RHS, we get
$H' = Act(TS_1 \|_H TS_2) \cap A_3 = (A_1 \cup A_2) \cap A_3$
$= (A_1 \cap A_3) \cup (A_2 \cap A_3)$

$\Rightarrow (A_2 \cap A_3) = (A_2 \cap A_3) \cup (A_1 \cap A_3)$
$\Rightarrow A_1 \cap A_3 \subseteq A_2 \cap A_3$ — (1)

Similarly from LHS & RHS,

for H, we get

$$A_1 \cap (A_2 \cup A_3) = A_1 \cap A_2$$

$$\Rightarrow (A_1 \cap A_2) \cup (A_1 \cap A_3) = (A_1 \cap A_2)$$

$$\Rightarrow A_1 \cap A_3 \subseteq A_1 \cap A_2 \quad — (5)$$

Also, we know that TS composition is commutative.

$$\Rightarrow TS_1 \,\|_H\, (TS_2 \,\|_{H'}\, TS_3) = TS_1 \,\|_H\, (TS_3 \,\|_{H'}\, TS_2) = (TS_1 \,\|_H\, TS_3) \,\|_{H'}\, TS_2$$

By associativity assumptions

$$\Rightarrow A_1 \cap A_2 \subseteq A_2 \cap A_3 \quad — (3) \quad \&$$

$$A_1 \cap A_2 \subseteq A_1 \cap A_3 \quad — (4)$$

$$\Rightarrow A_1 \cap A_3 = A_1 \cap A_2 \quad (\text{By } (5) \text{ \& } (4))$$

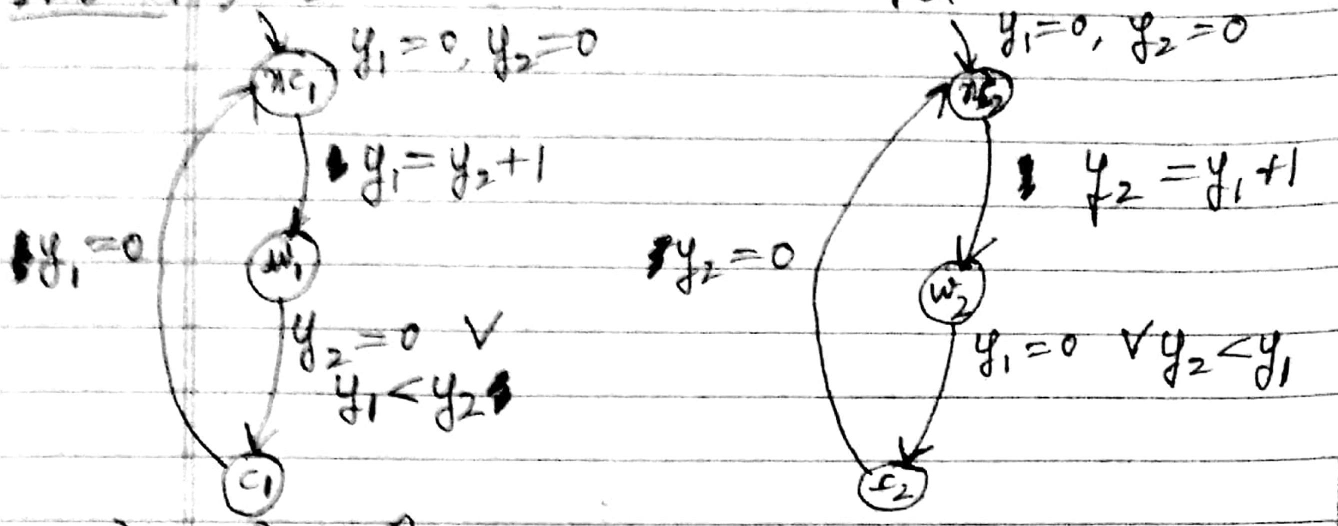
Similarly $A_1 \cap A_3 = A_2 \cap A_3$

$$\Rightarrow A_1 \cap A_2 = A_2 \cap A_3$$

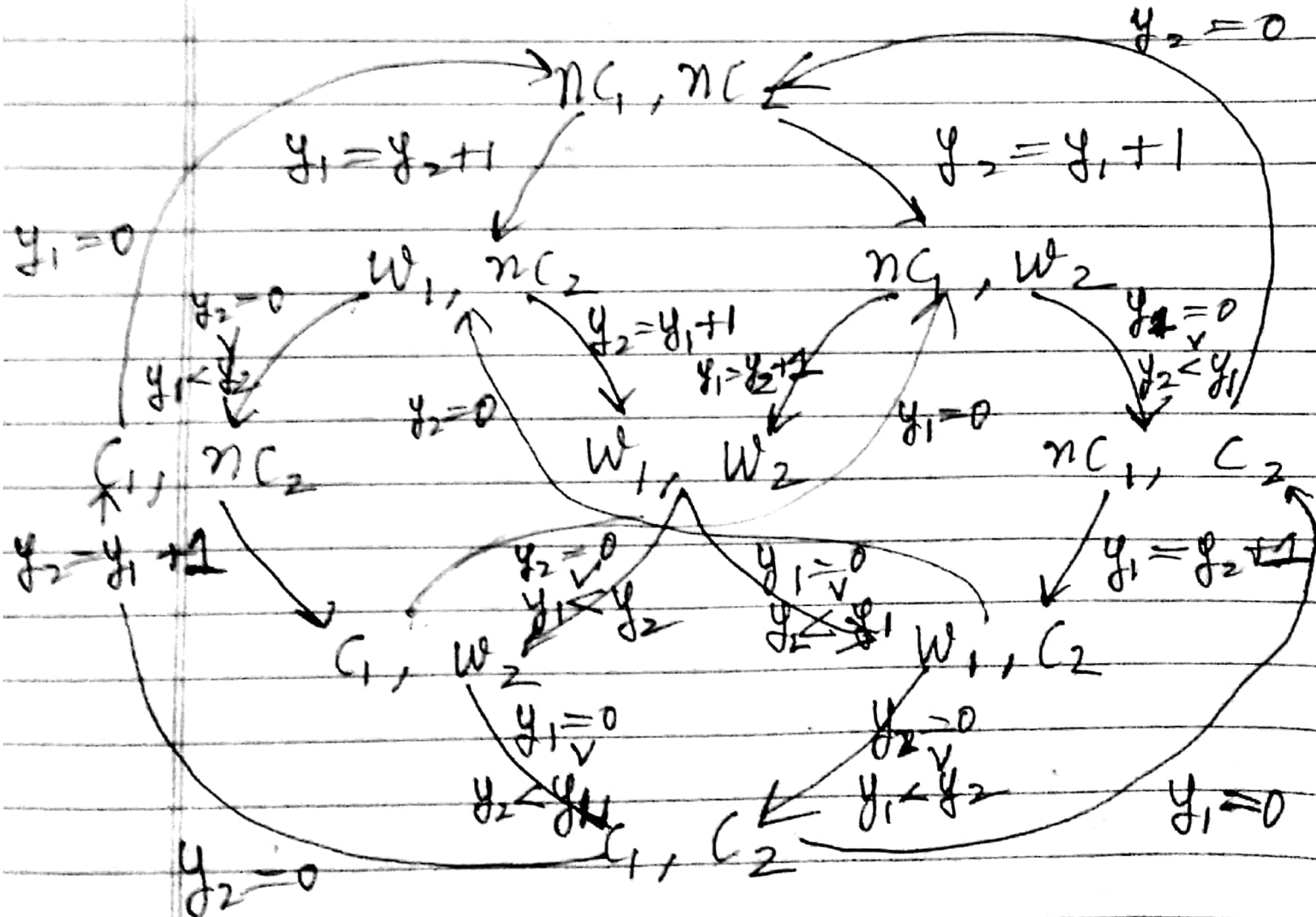
$$\Rightarrow H = H'$$

Hence, Proved.

Ans 2 a) P1 P2



b) $P_1 \parallel P_2$



The reachable states in $TS(P_1 \parallel P_2)$ for $y_1 \leq 2$ & $y_2 \leq 2$ are:

$(nc_1, nc_2, y_1=0, y_2=0)$, $(nc_1, w_2, y_1=0, y_2=1)$,
$(w_1, nc_2, y_1=1, y_2=0)$, $(w_1, w_2, y_1=1, y_2=2)$,
$(w_1, w_2, y_1=2, y_2=1)$, $(c_1, nc_2, y_1=1, y_2=0)$,
$(nc_1, c_2, y_1=0, y_2=1)$, $(c_1, w_2, y_1=1, y_2=2)$,
$(w_1, c_2, y_1=2, y_2=1)$ ~~$(c_1, w_2, y_1=1, y_2=2)$,~~
~~$(w_1, c_2, y_1=2, y_2=1)$,~~ $(nc_1, w_2, y_1=0, y_2=2)$,
$(w_1, nc_2, y_1=2, y_2=0)$, $(nc_1, c_2, y_1=0, y_2=0)$,
$(c_1, nc_2, y_1=2, y_2=0)$

c) To ensure mutual exclusion, the TS should not reach a state $(c_1, c_2, y_1=?, y_2=?)$.

Consider $PG1 \| PG2$.
We can reach (c_1, c_2) from (c_1, w_2) or (w_1, c_2).
We will argue for (c_1, w_2). The argument for (w_1, c_2) will be symmetrical.

To reach (c_1, w_2), we would have executed both $y_1 = y_2 + 1$ & $y_2 = y_1 + 1$, either taking the path $(nc_1, nc_2) \rightarrow (w_1, nc_2) \rightarrow (w_1, w_2) \rightarrow (c_1, w_2)$ or $(nc_1, nc_2) \rightarrow (w_1, nc_2) \rightarrow (c_1, nc_2) \rightarrow (c_1, w_2)$.

$\Rightarrow$ both y_1 & $y_2 > 0$ at (c_1, w_2).

Now, if we reach (c_1, w_2) from (w_1, w_2) then $\because$, $y_2 \neq 0$ & $y_1 \neq 0$ $\Rightarrow y_1 < y_2 \Rightarrow y_2 \not< y_1 \Rightarrow$ we can't reach (c_1, c_2).

If we reach from (c_1, nc_2) then $y_2 = y_1 + 1 \Rightarrow y_2 > y_1 \Rightarrow y_2 \not< y_1$ & $y_1 \neq 0 \Rightarrow$ we can't reach (c_1, c_2).

Hence, mutual exclusion is ensured.

d) Two processes will mutually wait for each other if they reach (w_1, w_2) in $PG_1 \| PG_2$ & can't progress further. At, (w_1, w_2) both $y_1 = y_2 + 1$ & $y_2 = y_1 + 1$ would have been executed & so $y_1 \neq 0$ & $y_2 \neq 0$ & either $y_1 > y_2$ or $y_2 > y_1$. $\Rightarrow$ Either P_1 or P_2 will make progress to c_1 or c_2 respectively.

Scanned by CamScanner

e) This happens on the path

$$\begin{tikzcd} (nc_1, nc_2) \arrow[r] & (w_1, nc_2) \arrow[r] & (c_1, nc_2) \arrow[ll, bend left] \end{tikzcd}$$

Here, P2 never gets a chance to enter its critical section & P1 always executes its critical section.

Logical time & Clocks

Ans. 1 Let $VC[1 \ldots n]$ be a vector timestamp at event e.

```
def convert (VC[1....n]):
    sum <- 0
    for i in 1 to n:
        sum += VC[i]
    return sum
```

We need a logical clock timestamp to satisfy
$\forall e_i, e_j \in H, e_i \rightarrow e_j \Rightarrow T(e_i) < T(e_j)$

where $T(e_i)$ = logical clock timestamp at e_i.

- We know that for Vector clock timestamps
$\forall e_i, e_j \in H, e_i \rightarrow e_j \Rightarrow VC_{e_i} < VC_{e_j}$

$VC_{e_i} < VC_{e_j} \Rightarrow \sum_k VC_{e_i}[k] < \sum_k VC_{e_j}[k]$

$\Rightarrow T(e_i) < T(e_j)$

Hence, Proved.

Page No.
Date:

Ans. 2 Strong clock consistency for vector clocks.

$\forall e_i, e_j \in H, i \neq j : e_i \rightarrow e_j \Leftrightarrow VC_{e_j} < VC_{e_j}$

We will prove this in two parts

a) $e_i \rightarrow e_j \Rightarrow VC_{e_i} < VC_{e_j}$

b) $e_i \not\rightarrow e_j \Rightarrow VC_{e_i} \not< VC_{e_j}$

Proof a):

Consider the following relation:

$R = \{ (a, b) \mid$ a just precedes b in a process execution or a is a msg send from a process i to a process j with b as the corresponding receive $\}$

We use the fact that the "happens before is the smallest relation which contains R & is irreflexive & transitive.

⇒ "happens before" is a transitive closure of R, & irreflexive.

⇒ $\rightarrow = R^+ = \bigcup_{i \in \mathbb{N}} R^i$

Now, we will prove a) on each R^i & show that R^+ satisfies a)

<u>Base Case:</u> Let $(e_i, e_j) \in R$. ⇒

$VC_{e_i} < VC_{e_j}$ by definition of vector clocks.

If e_i & e_j $\in$ same process, say P, then

$$VC_{e_j}[P] = VC_{e_i}[P] + d \quad (d > 0)$$

$$\& \; VC_{e_j}[k] = VC_{e_i}[k] \quad \forall k \neq P.$$

If $e_i \in P$ & $e_j \in P'$ & e_i & e_j are send & receive respectively then

$$VC_{e_j}[P'] > VC_{e_i}[P'] + d \quad (d > 0)$$

$$\& \; VC_{e_j}[k] > VC_{e_i}[k] \quad \forall k \neq P'$$

Page No.
Date.

$\Rightarrow$ a) Holds true for R.

Induction Hypothesis: Consider that a) holds true for R^N.

Induction Step: Consider R^{N+1}.

$R^{N+1} = R \circ R^N$

Let $(l_i, l_j) \in R^{N+1}$

$\Rightarrow \exists\, l_k$ s.t. $(l_i, l_k) \in R$ & $(l_k, l_j) \in R^N$. (by definition of R^{N+1})

$\because$ a) holds true for R & R^N

$\Rightarrow VC_{l_i} < VC_{l_k}$ & $VC_{l_k} < VC_{l_j}$

$\Rightarrow VC_{l_i} < VC_{l_j}$

$\Rightarrow \forall\, (l_i, l_j) \in R^{N+1}, VC_{l_i} < VC_{l_j}$

$\Rightarrow$ a) holds for R^{N+1}

$\Rightarrow$ a) holds for R^k s.t. $k \in N$

$\Rightarrow$ a) holds for R^+.

$\Rightarrow$ a) holds for $\rightarrow$.

Proof b)

$e_i \not\rightarrow e_j$

$\Rightarrow e_j \rightarrow e_i$ or $e_j \not\rightarrow e_i$

Case 1 $e_j \rightarrow e_i \Rightarrow VC_{e_j} < VC_{e_i}$
$\Rightarrow VC_{e_i} \not< VC_{e_j}$

Case 2 $e_i \not\rightarrow e_j$ & $e_j \not\rightarrow e_i$

$\Rightarrow e_i$ & e_j are concurrent.

$\Rightarrow e_i \in$ ~~process~~ P & $e_j \in P'$ s.t. $P \neq P'$.

Consider $VC_{e_i}[P'] \;\&\; VC_{e_j}[P']$.

If $VC_{e_i}[P'] \geq VC_{e_j}[P']$ then

there has to exist a trail of messages from P' to P which conveys the latest information about P' to P as otherwise P could not have known about the latest information of P'. $\Rightarrow$ that $\exists$ a chain of events with e_j & e_i & $e_j \rightarrow e_i$.

Similarly, if $VC_{e_j}[P] \geq VC_{e_i}[P]$

Then $\exists$ a chain of events which contain a series of messages from P to P' s.t. $e_i \rightarrow e_j$.

$\therefore$, $e_i \not\rightarrow e_j$ & $e_j \not\rightarrow e_i \Rightarrow$

$VC_{e_j}[P'] > VC_{e_i}[P']$ &

$VC_{e_i}[P] > VC_{e_j}[P]$.

$\Rightarrow VC_{e_i} \not< VC_{e_j}$

$\Rightarrow e_i \not\rightarrow e_j \Rightarrow VC_{e_i} \not< VC_{e_j}$.

$\Rightarrow VC_{e_i} < VC_{e_j} \Rightarrow e_i \rightarrow e_j$.

Hence, b) Proved.

Combining a) & b) $\Rightarrow$

$\forall e_i, e_j \in H$, $e_i \rightarrow e_j \Longleftrightarrow$
$VC_{e_i} < VC_{e_j}$.