Ankush Phulia
2014CS50279
23/08/2018

Homework 1

1. Consider 3 transition systems TS_1, TS_2 & TS_3

→ Firstly, we show that $TS_1 \|_H TS_2 = TS_2 \|_H TS_1$

This is quite obvious from definition of handshaking

- For LHS & RHS, set of actions $= A_1 \cup A_2 = A_2 \cup A_1$ where A_1 & A_2 are action sets of TS_1 & TS_2 respectively
- Further for LHS, states are $S_1 \times S_2$ & for RHS, $S_2 \times S_1$

we can define an isomorphism(ϕ) b/w these two as follows $\phi((s_1, s_2)) = (s_2, s_1)$ for $s_1 \in S_1$ & $s_2 \in S_2$

- This is clearly a 1-1, onto mapping

$\therefore$ $TS_1 \|_H TS_2 = TS_2 \|_H TS_1$

→ Now, to show that $H \neq H' \longrightarrow TS_1 \|_H (TS_2 \|_{H'} TS_3) \neq (TS_1 \|_H TS_2) \|_{H'} TS_3$

we consider the contrapositive, ie.

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

TP: $H = H'$

Proof: Let $A_i \equiv$ Action set of TS_i as before

- From defmition of handshaking, $H' = A_2 \cap A_3$ (LHS) (I)

Similarly, from RHS, $H = A_1 \cap A_2$ — (II)

- But from LHS' composition of 3 transition systems

$H = A_1 \cap (A_2 \cup A_3)$ as action set $(TS_2 \|_{H'} TS_3) = A_2 \cup A_3$

~~[illegible]~~ $\therefore$ $H = (A_1 \cap A_2) \cup (A_1 \cap A_3)$ (III)

Similarly for H', $H' = (A_1 \cap A_3) \cup (A_2 \cap A_3)$ (IV)

Combining (II) & (III) and (I) & (IV)

For H, $(A_1 \cap A_2) \cup (A_1 \cap A_3) = A_1 \cap A_2$

$\Rightarrow A_1 \cap A_3 \subseteq A_1 \cap A_2$

For H', $(A_1 \cap A_3) \cup (A_2 \cap A_3) = (A_2 \cap A_3)$ — I

$\Rightarrow A_1 \cap A_3 \subseteq A_2 \cap A_3$

23/08/2018

Further, since $\|_H$ is commutative

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

& by the (given), $TS = (TS_1 \|_H TS_3) \|_{H'} TS_2$

So, similar to before

$$A_1 \cap A_2 \subseteq A_2 \cap A_3 \quad \& \quad A_1 \cap A_3 \subseteq A_2 \cap A_3 \quad (\text{II})$$

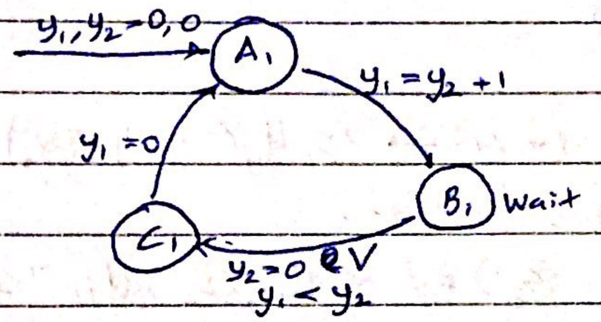
From (I) & (II)

$$A_1 \cap A_2 = A_1 \cap A_3 \quad \& \quad A_2 \cap A_3 = A_1 \cap A_3$$

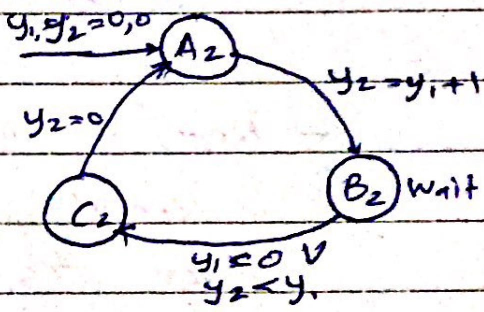
$$\Rightarrow A_1 \cap A_2 = A_2 \cap A_3$$

$$\Rightarrow H = H' \qquad \text{(QED)}$$

2 a)

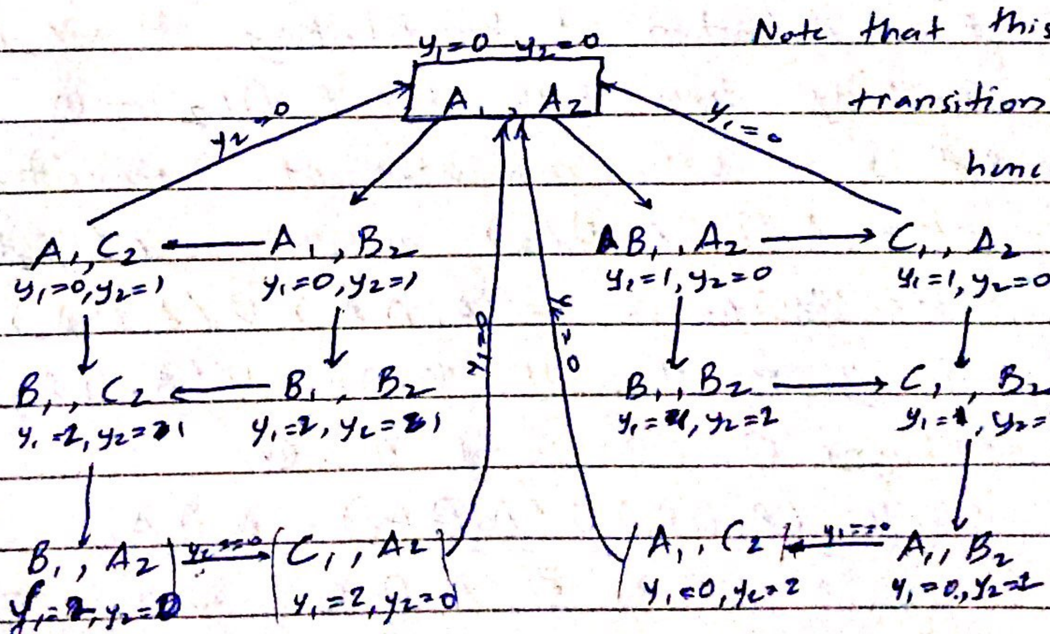


Process P_1



Process P_2

b) $P_1 \| P_2$ & $y_1 \leq 2$ & $y_2 \leq 2$



Note that this is the transition system, hence all states have assoc. var. values

Clearly, this entire TS is reachable (unreachable states are not drawn) – 13 states

c) To ensure mutual exclusion, in any reachable state, the "id's" at the state cannot ~~be~~ be (C_1, C_2). That is,
No state of the form $(C_1, C_2, y_1 = \_\_, y_2 = \_\_)$ is reached.
It is quite clear from (b) that such a thing isn't possible.
→ To further support the claim, note that to reach (C_1, C_2) it must be through (B_1, C_2) or (C_1, B_2). To get to (B_1, C_2), we came from (A_1, C_2) or (B_1, B_2), and so at this point, both $y_1 > 0$ & $y_2 > 0$ & $y_1 > y_2 \Rightarrow y_1 \neq y_2$
so the condition for transition $B_1 \rightarrow C_1$ doesn't hold ~~[illegible]~~ so
it must stay in that state & wait, while C_1 can adv. to A_1
→ Similar argument for $(C_1, B_2) \not\rightarrow (C_1, C_2)$
∴ Mutual exclusion is ensured

d) P_1 & P_2 will wait for each other (dead-locked), if in the system $P_1 \| P_2$, (B_1, B_2) is achieved, i.e. a state of the form $(B_1, B_2, y_1 = \_\_, y_2 = \_\_)$ in TS given (b)
AND cannot progress further.
From the TS (b), its clear that at both states (B_1, B_2)
$y_1 > 0$ & $y_2 > 0$ $(y_1 = 2, y_1 = 1$ ~~or~~ $y_1 = 1, y_2 = 2)$
Also, $y_1 \neq y_2$ either. Thus one of the conditions for $B_1 \rightarrow C_1$ or $B_2 \rightarrow C_2$ must be true (exactly one will be true)
∴ Depending on condition, P_1 or (exclusive) P_2 progress.

e) Consider the path (Top right portion in (b))

$A_1, A_2 \rightarrow B_1, A_2 \rightarrow C_1, A_2 \rightarrow A_1, A_2$

Quite obvious that even though P_1 executes critical section repeatedly, P_2 does not.
Thus P_2 is never able to enter critical section and must wait. However, it doesn't "wait" just before the section

23/08/2018

1. Consider a vector clock timestamp provided by as $VC_i^{(t)}$, where i is i^{th} process, t is ~~time~~ internal (process) event no.

We write a convert function = convert : $VC(e_i) \rightarrow$ int
↑ vector timestamp for event e_i

→ func convert $(VC_i(e_i))$ =
  $t = 0$
  for all indices j in $VC_i(e_i)$
    $t \leftarrow t + VC_i(e_i)[j]$
  ~~output~~ ~~output~~ return t

→ Claim: Output timestamp 't' satisfies weak consistency
i.e. $\forall e_i, e_j \in H \quad e_i \rightarrow e_j \Rightarrow t(e_i) < t(e_j)$
  where $t(e_i) = \text{convert}(VC(e_i))$

Proof: $e_i \rightarrow e_j \Rightarrow VC(e_i) < VC(e_j)$
  $\Rightarrow \forall k \; VC(e_i)[k] \leq VC(e_j)[k]$ &
  $\exists k' \ni VC(e_i)[k'] < VC(e_j)[k']$ — ①
  $\Rightarrow \sum_k VC(e_i)[k] \leq \sum_k VC(e_j)[k]$ (or $<$ due to ①)
  $\Rightarrow$ convert $(VC(e_i)) \leq$ convert $(VC(e_j))$
  $\Rightarrow t(e_i) \leq t(e_j)$ (QED)

2. Given: vector clock timestamps for $e_i \in H \;\forall i$, $\rightarrow$ rel.
TP: ~~$e_i \rightarrow e_j$~~ $\rightarrow$ being causal precedence relaⁿ
  $\forall i,j \quad e_i, e_j \in H, \; e_i \rightarrow e_j \leftrightarrow VC(e_i) < VC(e_j)$

Proof:
(1) $(\Longrightarrow)$ Referring to definition given, let us define
$\rightarrow' = \{(e_i, e_j) \mid (e_i \text{ preceed } e_j \text{ in same process}) \vee$
  $(e_i \text{ is SEND } \& \; e_j \text{ is matching RECV})\}$
$\rightarrow^k = \{(e_i, e_j) \mid \exists e_{k'} \ni e_i \rightarrow^{k-1} e_{k'} \;\&\; e_{k'} \rightarrow e_j\}$
Now define $\rightarrow^+ = \bigcup_{k \in \mathbb{N}} \rightarrow^k$ i.e. a transitive closure
(THIS is EXACTLY the causal precedence relation / happens before, as discussed in class & defines a P.O. over events)

23/08/2018

i.e. $\longrightarrow^+$ is irreflexive, transitive closure, of MINIMAL SIZE, which we may now call $\longrightarrow$ (defined in class)

→ Now ~~can~~ we proceed via induction on $\longrightarrow^k$

— Basis: consider $\longrightarrow^1$, applied ~~length~~ 2, $\{e_1, e_2\}$

- let $e_1 \to e_2 \Rightarrow$ if e_1 & e_2 in same process (say x)

$VC(e_1)[x] < VC(e_2)[x] = VC(e_1)[x] + d$ (from VC rules)

- Similarly, if e_1 = send & e_2 = Recv (from P_x to P_y)

$VC_y(e_2)[x] = \max(VC_y(e_2)[x], VC_x(e_1)[x])$
$\geq VC_x(e_1)[x]$

& $VC_y(e_2)[y] = VC_y(e_2)[y] + d$ & $d > 0$] → $VC(e_2) > VC_x(e_1)$

∴ Clearly ($\Rightarrow$) holds in base case

— Induction hypothesis: ($\Rightarrow$) holds for $\longrightarrow^n$

— Inductive step: Consider $\longrightarrow^{n+1}$

Now $e_i \longrightarrow^{n+1} e_j \Rightarrow \exists\, k \ni e_i \longrightarrow^{n} e_k$ & $e_k \longrightarrow^{1} e_j$

(from definition of $\longrightarrow^{n+1}$)

By using induction hypothesis $VC(e_i) < VC(e_k)$ &

$VC(e_k) < VC(e_j) \Rightarrow VC(e_i) < VC(e_j)$

∴ By PMI, same hypothesis holds for $(n+1)$ when it holds for n, hypothesis ($\Rightarrow$) is true $\forall\, n \in \mathbb{N}$

∴ ($\Rightarrow$) holds $\forall \longrightarrow^k\ k \in \mathbb{N} \Rightarrow$ ($\Rightarrow$) holds for $\longrightarrow^+ \equiv \longrightarrow$

(2) ($\Leftarrow$) Given: $e_i \not\to e_j$

TP: $VC(e_i) \not< VC(e_j)$

Proof: i) If $e_i \not\to e_j$ & $e_j \to e_i$, we are done as from prev. part $VC(e_j) < VC(e_i) \Rightarrow VC(e_i) \not< VC(e_j)$

Else $e_i \parallel e_j$ (concurrent) i.e. $e_j \not\to e_i$ either

(ii) e_i & e_j do not belong to the same process, otherwise $e_i \not\to e_j \Leftrightarrow e_j \to e_i$ & this is covered in (i)

23/08/2018

④ By transitivity
$e_i \rightarrow e_j \rightarrow$ a contradiction $\rightarrow VC(e_i)[x] > VC(e_j)[x]$ — ①

(iii) Thus e_i & e_j are on diff. processes, say P_x & P_y

Now $e_i \not\rightarrow e_j$, consider ~~e~~$_i$ $VC(e_i)[x]$ & $VC(e_j)[x]$

1) If $VC(e_i)[x] \leq VC(e_j)[x]$, then either e_i is a SEND & e_j its RECV, OR $\exists$ events $\{e_{k_1}, e_{k_2}, \ldots\}$

- Such that $e_i \rightarrow e_{k_1}$ i.e. e_i is SEND & e_{k_1} its RECV, or $\exists e_i' (\in P_x) \ni e_i \rightarrow e_i' \rightarrow e_{k_1} \rightarrow e_{k_n}$

- Similarly e_{k_n} sends info. to P_y & is received at or before e_j. Otherwise, no way for P_x's info. to reach $P_y \Rightarrow e_{k_n} \rightarrow e_j$ ⊛

- The crux is that a process has its own, most updated info.

2) Similarly, if $VC(e_i)[y] \geq VC(e_j)[y]$, same as above we conclude that $\exists$ a chain of events & $e_j \rightarrow e_i \Rightarrow$ ⨯

$\therefore VC(e_i)[y] < VC(e_j)[y]$ — ②

Clearly from ① & ②

$VC(e_i) \not< VC(e_j)$ (index x is more, index y is less),

Thus $e_i \not\rightarrow e_j \longrightarrow VC(e_i) \not< VC(e_j)$] contra-positive

$\Rightarrow VC(e_i) < VC(e_j) \rightarrow (e_i \rightarrow e_j)$

Thus ($\Leftarrow$) holds true

So ($\Rightarrow$) & ($\Leftarrow$) holding means, equivalence holds good.