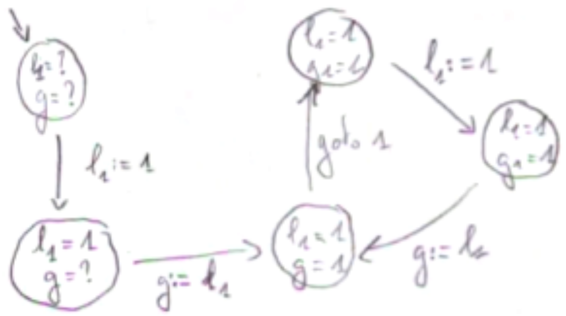
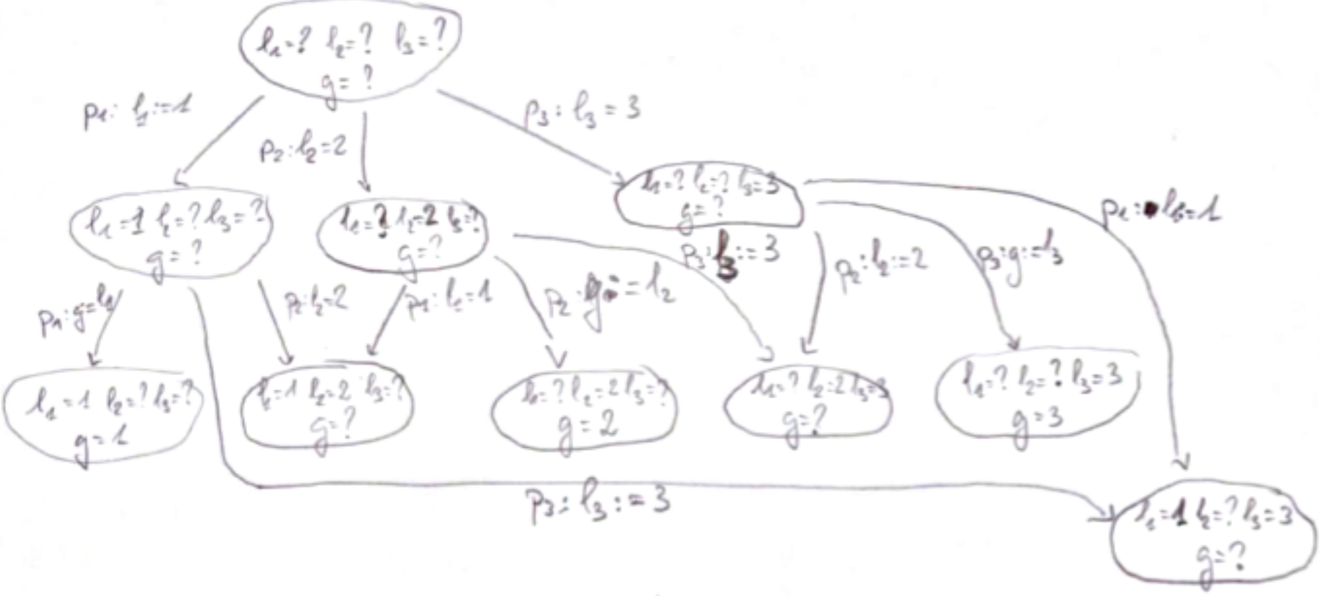
Ex 7.1

1]



2]



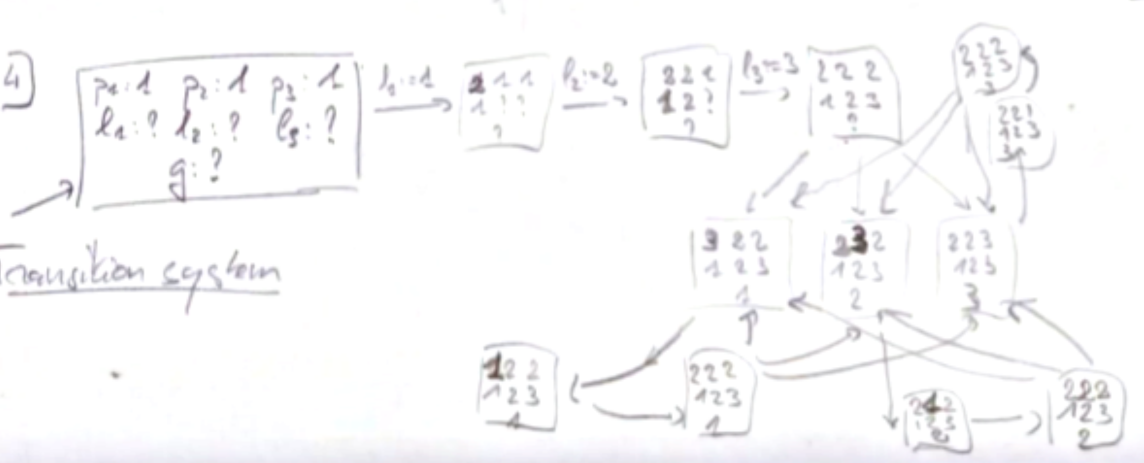
3] $V = \{g := l_1 ; g := l_2 ; g := l_3\}$ are visible actions

I =

	$l_1 \, l_2 \, l_3$ (:=1, :=2, :=3)	$g \, g \, g$ (:= l_1, := l_2, := l_3)	$p_1 \, p_2 \, p_3$ (goto 1)
$l_1 := 1$, $l_2 := 2$, $l_3 := 3$	(clearly) independent	Independent (α)	Independent (α)
$g := l_1$, $g := l_2$, $g := l_3$		Not independent	Independent (α)
p_1: goto 1, p_2: goto 1, p_3: goto 1			(Clearly) Independent

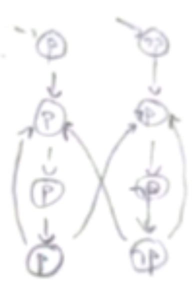
α: Why are $l_2 := 2$ and $g := l_2$ independent?
The control flow of process p_2 forbid any choice between the two: there are not diamond possible: Therefore all diamonds are closed

4]



Transition system

Kripke structure



5, 6. We recall the 4 conditions:

- c0: $red(s) = \emptyset$ iff $en(s) = \emptyset$

  This condition is clearly satisfied, as at least one action is kept.

- c1: Every path of K starting at a state s satisfies the following: no action that depends on some action in $red(s)$ occurs before an action from $red(s)$.

  Reduced states contain exactly one action, which is independant with all other, therefore the property c1 holds.

- c2: If $red(s) \neq en(s)$ then all actions in $red(s)$ are invisible.

  Reduced states contain exactly one action, which is invisible.

- c3: For all cycles in K' the following holds: if $a \in en(s)$ for some state s in the cycle, then $a \in red(s')$ for some (possibly other) state s' in the cycle.

  This condition holds for the concurrent execution of processes p_1, p_2, p_3. Provided there is at least one global assignment in any loop of any process, this condition hold.

7. If such instructions are introduced, we have to take extra care with condition $c1$ as such action would not be independant with the global assignments. Therefore, if we modify the previous partial order reduction by replacing the second bullet by "else if p_2 may execute an instruction other than a global variable assignment or a test, it does".

   If we allow (only) process p_2 to test the value of the global variable (through an action g==k), how to obtain a (non-trivial) partial order reduction ?

8. if we check the parity condition on variable l_1 it is no more a partial order reduction , because assignment to variable l_1 are no more invisible actions (and hence condition $c1$ is no more satisfied).

9. If we replace the first bullet by "if p_1 may execute an instruction other than an assignment (local or global), it does", we obtain a partial order reduction.