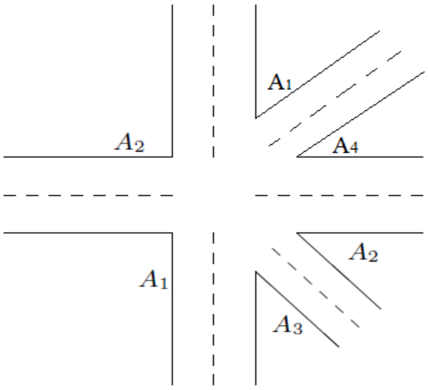
Mirpur University of Science and Technology (MUST), Mirpur-10250 (AJK), Pakistan

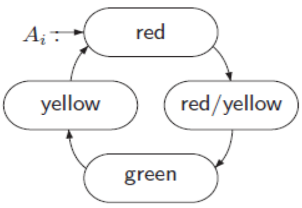
Department of Software Engineering

Course Title:	**Formal Methods in SE**	**Session:**	2015-2019
Course Code:	SE-364	**Semester:**	Spring-2018
Instructor:	Engr. Afzal Ahmed	**Deadline:**	18/04/2018
Max Marks:	**50**	**CLO:**	CLO-1

Assignment 01

Q1. Please consider the following street junction with the specification of a traffic light as outlined on the right.





a) Choose appropriate actions and label the transitions of the traffic light transition system accordingly.

b) Give the transition system representation of a (reasonable) controller C that switches the green signal lamps in the following order: A_1, A_2, A_3, A_4, A_1, A_2, A_3, A_4 . . . (*Hint: Choose an appropriate communication mechanism*.)

c) Outline the transition system $A_1||A_2||A_3||A_4||C$.

Q2. The following program is a mutual exclusion protocol for two processes due to Pnueli [1]. There is a single shared variable s which is either 0 or 1, and initially 1. Besides, each process has a local Boolean variable y that initially equals 0. The program text for process P_i (i = 0, 1) is as follows:

loop forever do

  begin

  l1: Noncritical section

  l2: $(y_i, s) := (1, i)$;

  l3: **wait until** $((y_{1-i} = 0) \vee (s \neq i))$;

  l4: Critical section

  l5: $y_i := 0$

end.

Here, the statement $(y_i, s) := (1, i)$; is a *multiple assignment* in which variable $y_i := 1$ and $s := I$ is a single, atomic step.

Please answer following questions:

a) Define the program graph of a process in Pnueli's algorithm.

b) Determine the transition system for each process.

c) Construct their parallel composition.

d) Check whether the algorithm ensures mutual exclusion.
e) Check whether the algorithm ensures starvation freedom.
f) The last two questions may be answered by inspecting the transition system.

Q3. Please consider the following mutual exclusion algorithm that uses the shared variables y_1 and y_2 (initially both 0).

Process P_1:
while true **do**
. . . noncritical section . . .
$y_1 := y_2 + 1$;
wait until $(y_2 = 0) \vee (y_1 < y_2)$
. . . critical section . . .
$y_1 := 0$;
od

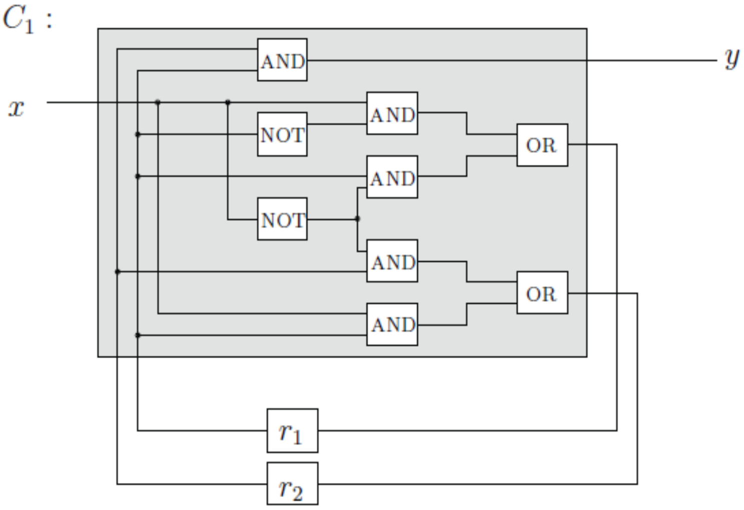
Process P_2:
while true **do**
. . . noncritical section . . .
$y_2 := y_1 + 1$;
wait until $(y_1 = 0) \vee (y_2 < y_1)$
. . . critical section . . .
$y_2 := 0$;
od

Please answer following questions:

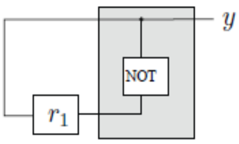
a) Give the program graph representations of both processes. (A pictorial representation suffices.)
b) Give the reachable part of the transition system of $P1 \,||\, P2$ where $y1 \leq 2$ and $y2 \leq 2$.
c) Describe an execution that shows that the entire transition system is infinite.
d) Check whether the algorithm indeed ensures mutual exclusion.
e) Check whether the algorithm never reaches a state in which both processes are mutually waiting for each other.
f) Is it possible that a process that wants to enter the critical section has to wait ad infinitum?

Q4. Please consider the following two sequential hardware circuits C_1 and C_2:

C_1 :



C_2 :



a) Give the transition system representation $TS(C_1)$ of the circuit C_1.
b) Let $TS(C_2)$ be the transition system of the circuit C_2. Outline the transition system $TS(C_1) \otimes TS(C_2)$.

Q5. Please consider the following leader election algorithm: For $n \in N$, n processes $P_1, \ldots, P_n$ are located in a ring topology where each process is connected by an unidirectional channel to its neighbor in a clockwise manner.

To distinguish the processes, each process is assigned a unique identifier $id \in \{1, \ldots, n\}$. The aim is to elect the process with the highest identifier as the leader within the ring. Therefore each process executes the following algorithm:

send (*id*);	*initially set to process' id*
while (true) **do**	
receive (*m*);	
if (*m* = *id*) **then stop**;	*process is the leader*
if (*m* > *id*) **then send** (*m*);	*forward identifier*
od	

a) Model the leader election protocol for n processes as a channel system.

b) Give an initial execution fragment of $TS([P_1|P_2|P_3|P_4])$ such that at least one process has executed the send statement within the body of the whileloop. Assume for $0 < i \leq 3$, that process P_i has identifier $id_i = i$.

Good Luck ☺

Reference(s)

[1] W.-P. de Roever and F. S. de Boer and U. Hannemann and J. Hooman and Y. Lakhnech and M. Poel and J. Zwiers. *Concurrency Verification: Introduction to Compositional and Noncompositional Methods*. Number 54 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.