AALTO UNIVERSITY

FIRST SEMESTER, 2019/2020

DISTRIBUTED INTELLIGENT AUTOMATION SYSTEMS
ELEC 8102
(Time Allowed: THREE hours)

NOTE:

- You can write the solution on this form or on separate sheets, but you need to submit it **as one pdf file**.
- Please keep your solutions just like you made and submitted them, in case something went wrong with the submission or need to be checked afterwards.
- The exam is "open book" so you can have the course material (lectures, exercises, homework) at your disposal. But you must not take help from any other person while taking the exam.
- Electronic calculators and computer software can be used.
- With your signature (on the answer sheet you will submit) you assert that you have followed the above regulations. Good luck!
- Fill your name and student ID on the first page of your solution.
- You may attempt all questions.
- **Threshold to pass the course is 50 marks and more.**

SURNAME:		SIGNATURE:
FORENAME(S):		
ID:		

For office use only

MARKS:		
1	Out of 20	
2	Out of 10	
3	Out of 10	
4	Out of 15	
5	Out of 15	
6	Out of 15	
7	Out of 20	
TOTAL: (Out of 105)		

CONTINUED

1. **State-based design of automation control logic.** 20 marks (10/10)

Consider the finite-state machine in Figure 1. It is controlling the automatic elevator in a two-story building. State S1 is the initial state. The state machine operates with the following Boolean variables:
- Up, Dn – commands to move the lift up and down accordingly;
- Close, Open – commands to close and open doors of the cabin.
- Bt1, Bt2 – lift calling buttons at the floor 1 and 2 accordingly;
- CabBt1, CabBt2 – buttons in the cabin for selecting the desired destination floor;
- DoorsClosed – sensor signal indicating that the doors of the lift are closed
- AtFlr1, AtFlr2 – Sensor signals indicating that the lift cabin has reached the Floor 1, and Floor 2, respectively.
- RESTART – button at the central control room to handle faults.
- ALARM – sound and light warning at the central control room.

Let us assume that the cabin is fully automatic: it opens its doors once it arrives at the desired floor and closes the doors automatically after 60 seconds, so you don't need to implement the door control, except for the case in #5 below.

Complete the following two tasks a) and b).
 a) **Add the following features to this state machine:**
 1. If the lift remains motionless on the second floor for more than 1 minute, it should automatically return to the first floor.
 2. If moving from the first to the second floor takes more than 2 minutes or moving from the second to the first floor takes more than 1 minute, the ALARM should be raised. The state machine should transition to the alarm state and the corresponding output variable should be set to TRUE.
 3. Once ALARM is raised, the elevator should be stopped and not react to any further commands from buttons until the RESTART button is pressed and held for 5 seconds.
 4. Once the lift exits the alarm state, it should move to the first floor. The ALARM variable should be reset to FALSE.
 5. If the lift is called from another floor when its doors are still not closed (i.e. before the 60 seconds timeout expires), motion should start after 15 seconds from the moment of the call.
You can modify the state machine directly on Figure 1 and explain in the box below.
For starting timer in a state, use the following notation:
➔T1.IN(10s) means: start timer T1 for 10 seconds when the state is entered.
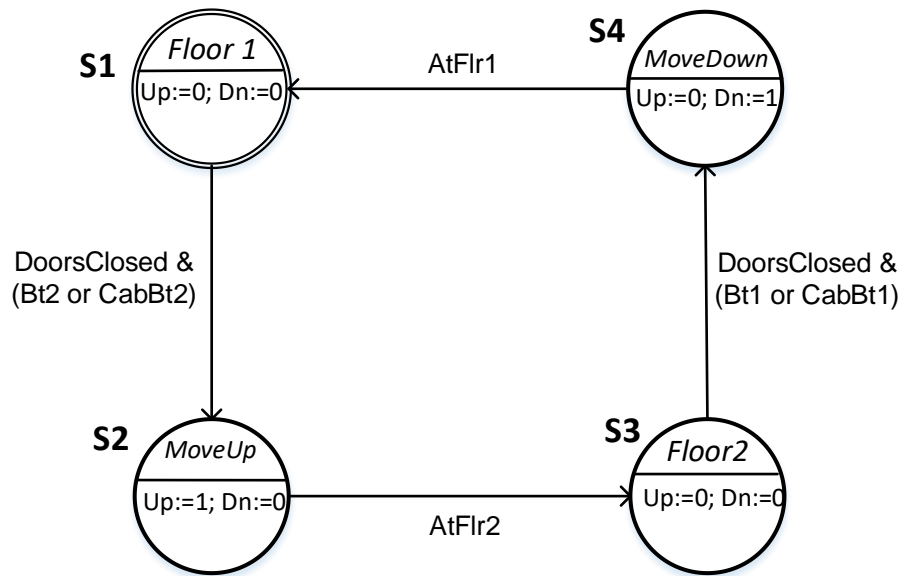
CONTINUED

S1 Floor 1 / Up:=0; Dn:=0

S4 MoveDown / Up:=0; Dn:=1

AtFlr1

DoorsClosed &
(Bt2 or CabBt2)

DoorsClosed &
(Bt1 or CabBt1)

S2 MoveUp / Up:=1; Dn:=0

S3 Floor2 / Up:=0; Dn:=0

AtFlr2

Figure 1.

b) **Implement the modified state machine in Boolean logic and ladder diagrams.**

First, present the Boolean equations implementing the state machine and after that the ladder diagram based on the Boolean equations.

Notation and declarations:

Timers are declared as follows
T1: TON;

T1.IN(10s) means start timer T1 preset to 10 seconds at the arrival to the state (rising edge).

For Boolean operations, please use the following notation:
A+B is A or B;
A B is A and B;
$\bar{A}$ is the negation of A.

Put the Boolean equations in this box. See the reminder below.

The state transition function is the logical sum of conditions on all incoming arcs to the state including the virtual loopback arc condition.

Draw the ladder diagram in this box using the ladder guidelines. Extend the provided grid, if needed.

2. Happened before ordering of events and Lamport clocks   10 marks (3/2/3/2)

Consider the process diagram in Figure 2.

The strict partial order relation "happened before" (as defined in the lectures) is denoted by the arrow sign, e.g. $a \to b$. The word "strict" means that $a \to a$ is false for any event a.
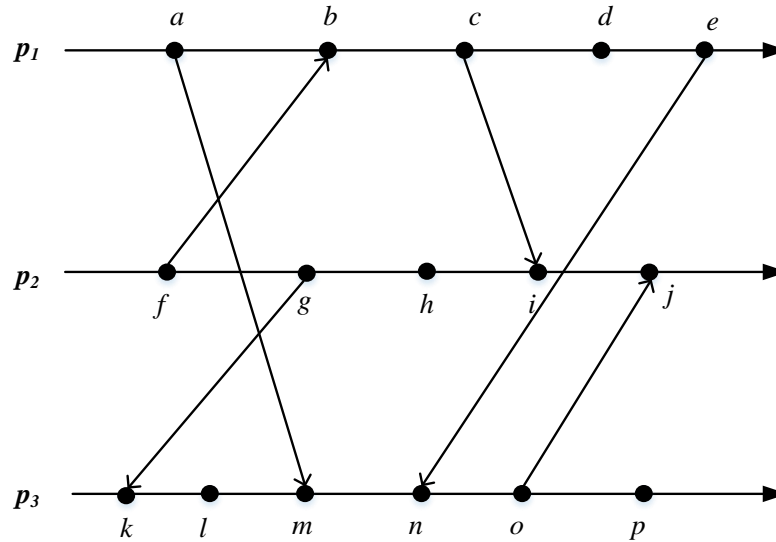

Figure 2.

a) Based on the definition of the "happened before" order, prove or disprove that $h \to p$.

b) List all events that happened-before event "m".

c) How could the diagram be changed so that the relation between h and p becomes opposite? Describe a minimal change.

d) Assign Lamport clocks to the events in the diagram in Figure 2.

3. Clock synchronization. 10 marks (3/3/4)

A client attempts to synchronize with time server. It sends *simultaneously* 4 request messages and records the round-trip times and timestamps returned by the server in the table below.

	Round-trip (ms)	Time (hr:min:sec.msec)
1	15	17:12:34.750
2	19	17:12:34.755
3	17	17:12:34.745
4	12	17:12:34.752

Consider the table above to answer the questions a) and b):

a) Which one of these round-trip measurements should it use to set its clock and why?

b) To what time will the client set its clock in this synchronization effort? Note that the setting is executed after all replies are received.

c) If clocks C1...C4 are internally synchronized with bound D, does it mean that they internally synchronized with bound 2D? Why?

CONTINUED

4. Engineering with IEC 61499 Function Blocks   15 marks (3/12)

Consider a liquid chemical tank plant as illustrated in Figure 3 below. The plant has the following sensors and actuators:

- MaxLevel, HighLevel, LowLevel, MinLevel, corresponding to four level sensors in descending order from the highest to the lowest level. Outputs high if they get submerged in the liquid, otherwise it outputs low.
- Three temperature sensors MaxTemp, MidTemp, and MinTemp in descending order from highest to lowest temperature level.
- A chemical gas sensor GasSensor (with binary outputs) outside of the tank. It outputs high if it detects the presence of the gas, otherwise it outputs low.
- One heater Heat (binary output).
- Two liquid inlets FastInlet and SlowInlet, and one outlet Outlet. The outlet is connected to the next processing station to supply liquid from this tank.
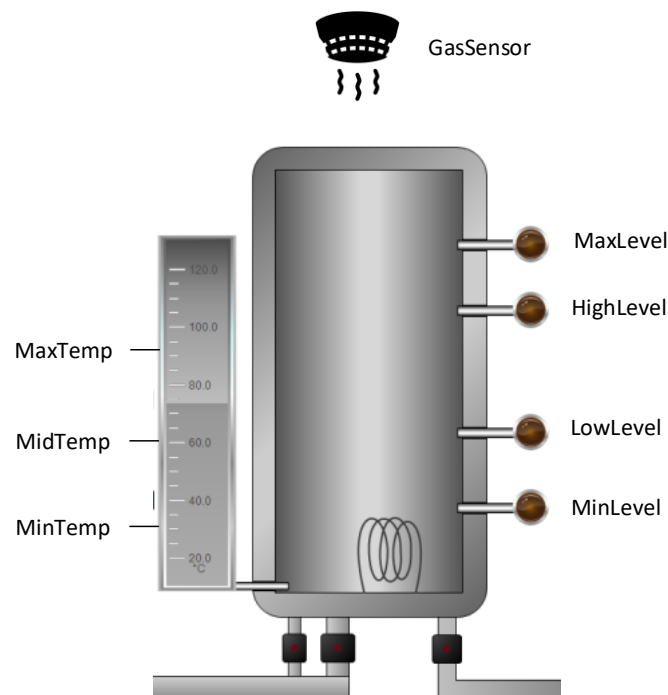
Figure 1. Illustration of the Plant.

The production should satisfy the following requirements below:

1. The temperature of the liquid must be kept between MidTemp and MinTemp level (it should alternate and reach those two levels).
2. The liquid in the tank should be transferred to the next processing station when the liquid's temperature is above MinTemp.
3. The liquid level should be maintained in between MaxLevel and MinLevel (it should alternate and reach those two levels).

4. When the liquid level is between MaxLevel and HighLevel, the rate of change of liquid level (when the level is increasing or decreasing) should be higher than when the liquid level is not between MaxLevel and HighLevel.

5. The liquid emits a dangerous gas which can be detected by the GasSensor. If the tank is somehow leaking such gas at any point in time, all liquid should be taken out from the tank and no more heating process should occur. Upon no detection of the gas by GasSensor, the plant should operate back normally.

Some important points that **must be taken into account**:
- Use the state-based approach.
- The initial condition: Some liquid is present in the tank. The liquid level is at the MinLevel and the temperature is below MinTemp.
- Some of the plant's physical characteristics:
  - There is heat dissipation that if the heater is off, the liquid's temperature will drop. The heater's heating capability is higher than the heat dissipation.
  - Heater, inlets, and outlet have mechanical components, such that these actuators must not be alternatingly turned on and off very quickly to avoid mechanical wear.
  - The flow rate of the FastInlet is higher than the SlowInlet. The flow rate of the Outlet is higher than the SlowInlet, but lower than the FastInlet. The sum of the flow rate of both FastInlet and SlowInlet is higher than the Outlet's flow rate.

Based on the specifications:

a) Develop control application as a network of IEC 61499 basic function blocks with inputs and outputs based on the plant's specifications above.

    A template of the application composed of two controller function blocks is provided in Figure 4. The template is incomplete, so you need to connect the input and output interfaces of the function blocks to the plant connectors ("FROM THE PLANT" and "TO THE PLANT"). Also, note that both controller function blocks may have missing input or output interface(s). If you think this is the case, then modify the controller function blocks drawing to add these interface(s) and also connect them to the corresponding plant connectors. You may put your modifications directly on the provided drawings, or you may draw the function blocks on the answer text box. **SnsEvt** corresponds to the event coming from the plant, which is generated in association to any changes of the value of any sensors, while **ControlEvt** refers to the event going to the plant. **E_MERGE** refers to a function block that merges two events. **PLCInit** generates the event which triggers the initialization of the function blocks.
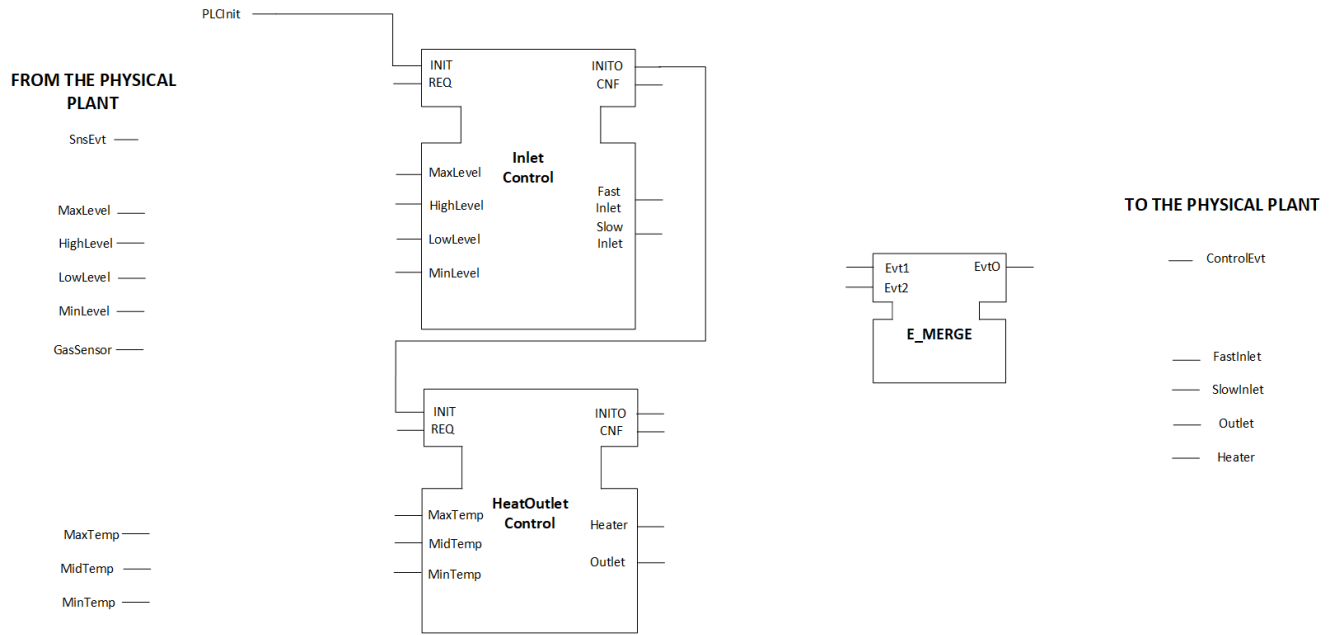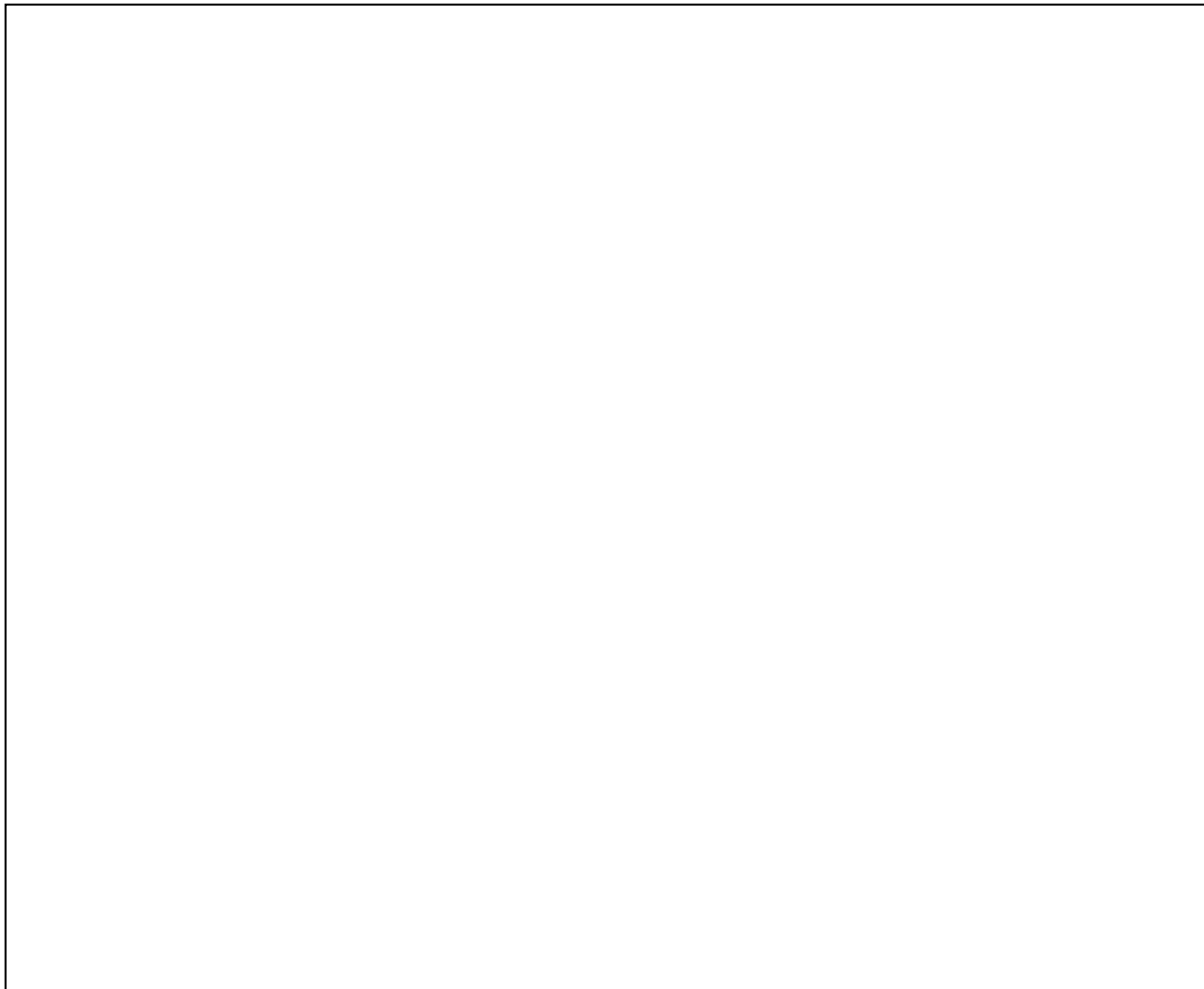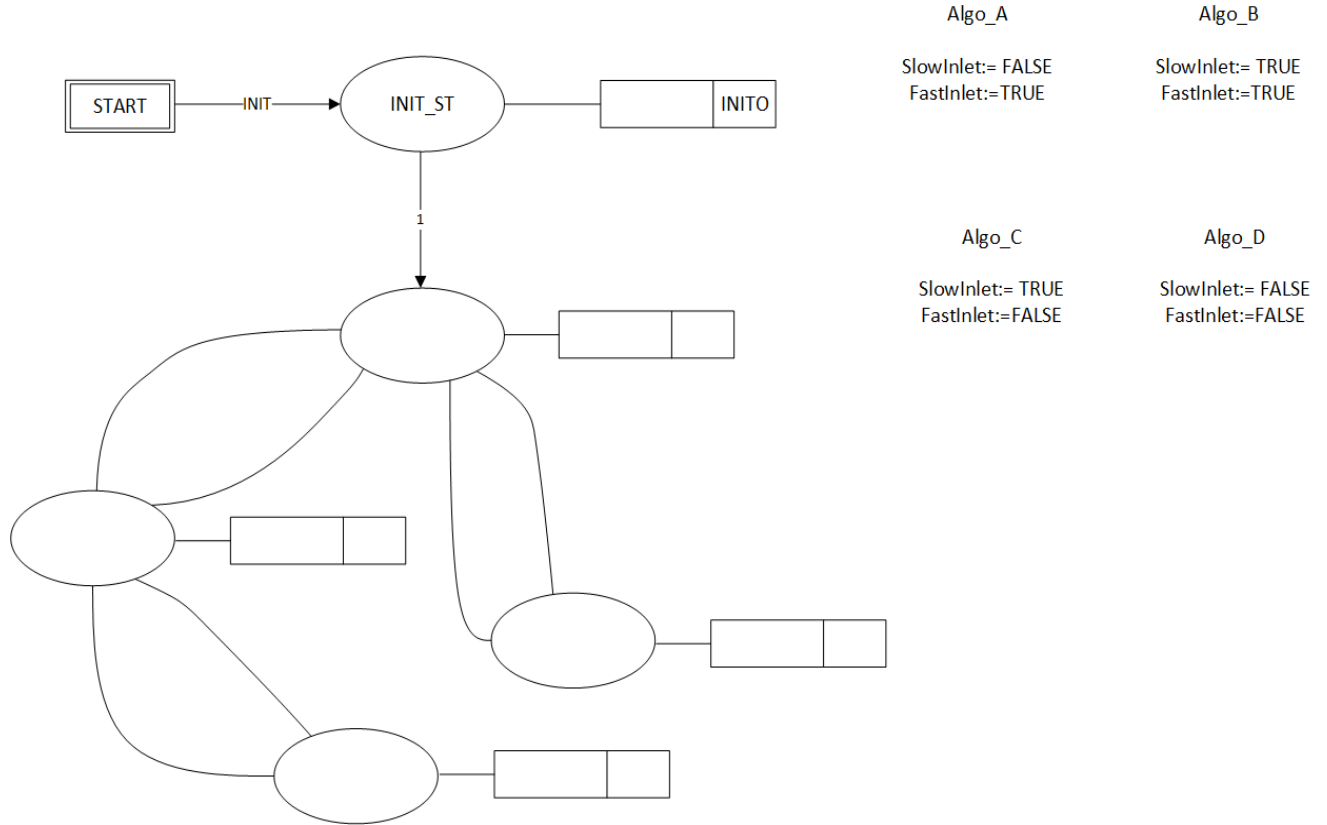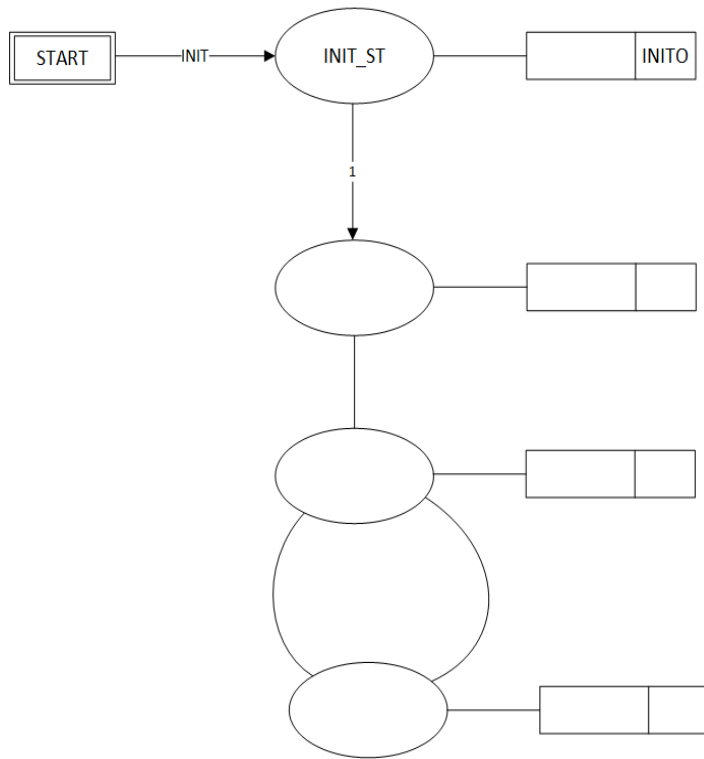
Figure 2. Controller Function Blocks.

b) Develop the **COMPLETE** Execution Control Chart / ECC of the controller IEC 61499 function block(s). Templates of the ECC of both controller function blocks are provided in Figures 5 and 6, along with several algorithms. Note that the templates are incomplete, lacking state transition conditions, names of the states and their algorithms, and the event output. Some state(s) may also be missing, so you may need to add those if you think this is the case. Then, **describe** the states that you defined in the ECC!
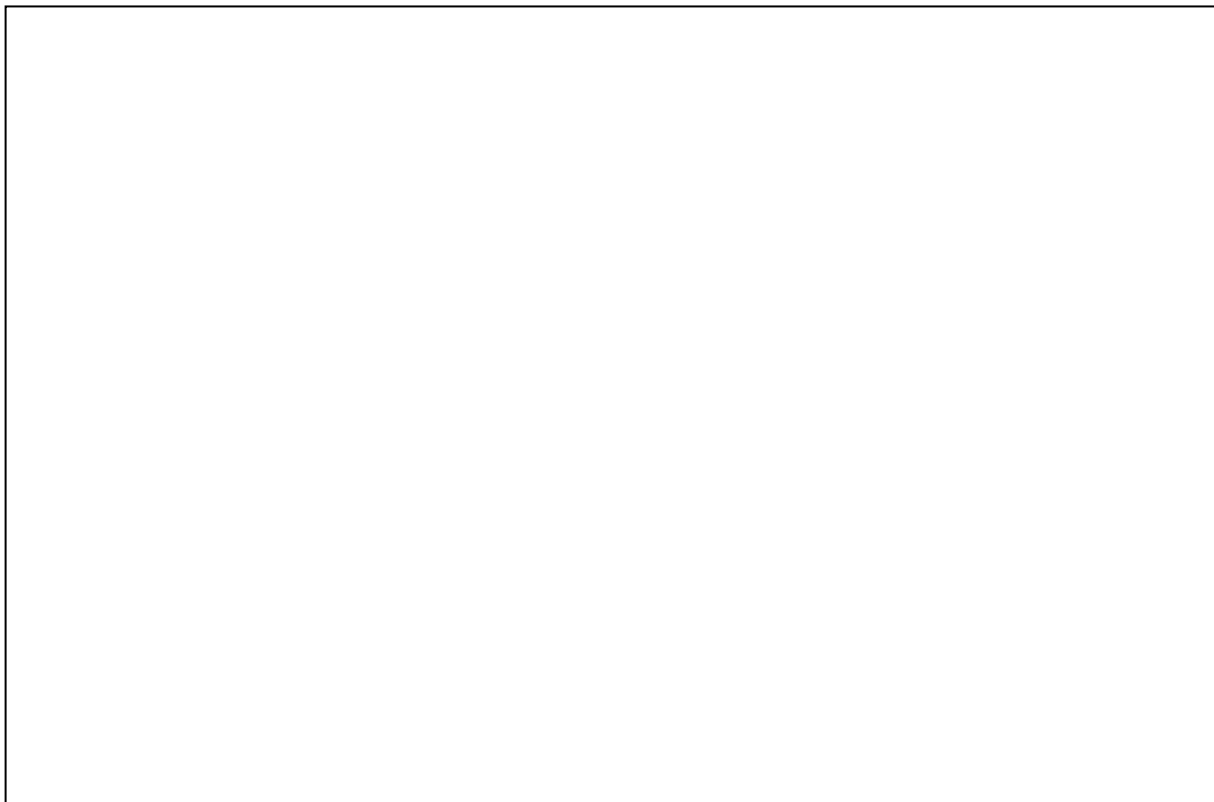

Figure 3. ECC of InletControl Function Block.

Figure 4. ECC of HeatOutletControl Function Block.

5. Mutual exclusion. 15 marks (5/5/5)

Recall the token ring mutual exclusion algorithm: a token is passed along a circle of nodes, only the node that has a token may enter the critical section, and the token is passed to the next node once the critical section is completed or the node does not need to enter it at the moment. Now consider the *ridiculous token ring algorithm*, which is the same as the token ring algorithm, except that: (1) there are two tokens passed in opposite directions, (2) a process can enter the critical section only if it has both tokens, and upon exit, it will be send the tokens to both neighbors, (3) if a process is not interested in a critical section and receives a token, it will immediately pass the token forward, (4) if a process is interested in a critical section and receives a token, it will wait until it gets the second token.

a) Check whether the ridiculous token ring algorithm satisfies the ME1 (safety) property. Explain your conclusions.

b) Show that the ridiculous token ring algorithm does not satisfy the ME2 (liveness) property by providing an example of a deadlock: a situation leading to the inability of any of the processes to acquire the access to the critical section.

CONTINUED

c) For a case of three processes, provide an example demonstrating that the ridiculous token ring algorithm does not satisfy the ME3 (happened-before ordering) property. You can draw on the timelines below. Also provide a short explanation. (Hint: neither does the regular token ring algorithm.)

p_1 ——————————————————————————————————————

p_2 ——————————————————————————————————————

p_3 ——————————————————————————————————————

6. Declarative programming languages
Below, you are given a Prolog program with some syntax errors.

```
some_relation([X], X) {}
some_relation([X|Tail], X) {
    some_relation(Tail, Y) && Y >= X;
}
some_relation([X|Tail], Y) {
    some_relation(Tail, Y) && Y < X;
}
```

Answer the following three questions.   **15 marks (5/6/4)**

a) Correct all syntax errors in this program. Write directly on the code above.

b) The name of this relation has been intentionally obfuscated. What does it do / what is its meaning?

c) Provide a concrete meaningful example of executing this program in Prolog command line, that is, fix some parameter(s) and leave other(s) as free variables. What will be the result of this execution?

7. Fuzzy inference

A simple fuzzy controller controls the water level of a fuzzy washing machine. It infers the required water level (output control value) (scale 0-10 liter) based on the load (scale 0-10 kg) and the amount of detergent (scale 0-10 cl) measurements. It applies *mamdani*-style fuzzy inference rules to infer the output water level. The scales of the input and output variables are partitioned into three fuzzy subsets as depicted in Figures. The preliminary test version of the rule base of the fuzzy inference system (FIS) contains only two rules:

1. IF (Load is '*low*')   AND   (Detergent is '*high*')   THEN Water is '*high*'
2. IF (Load is '*medium*') AND   (Detergent is '*medium*') THEN Water is '*medium*'

Question   **20 marks (5/5/5/5)**

Carry out the steps of fuzzy inference process applying these two rules graphically over the following two sketch drawings. The reasoning should be done for the following measured crisp values of the input variables: *Load* = 3kg; *Detergent* = 8cl

A) Represent the *fuzzification* operations graphically for both of these rules into the two sketch drawings below. Draw the necessary lines to indicate the fuzzification operations and highlight the results, i.e. the fuzzified values of the input variables

B) Represent the inference operations graphically and draw the inferred result output fuzzy sets into the OUTPUT diagrams on the right side.
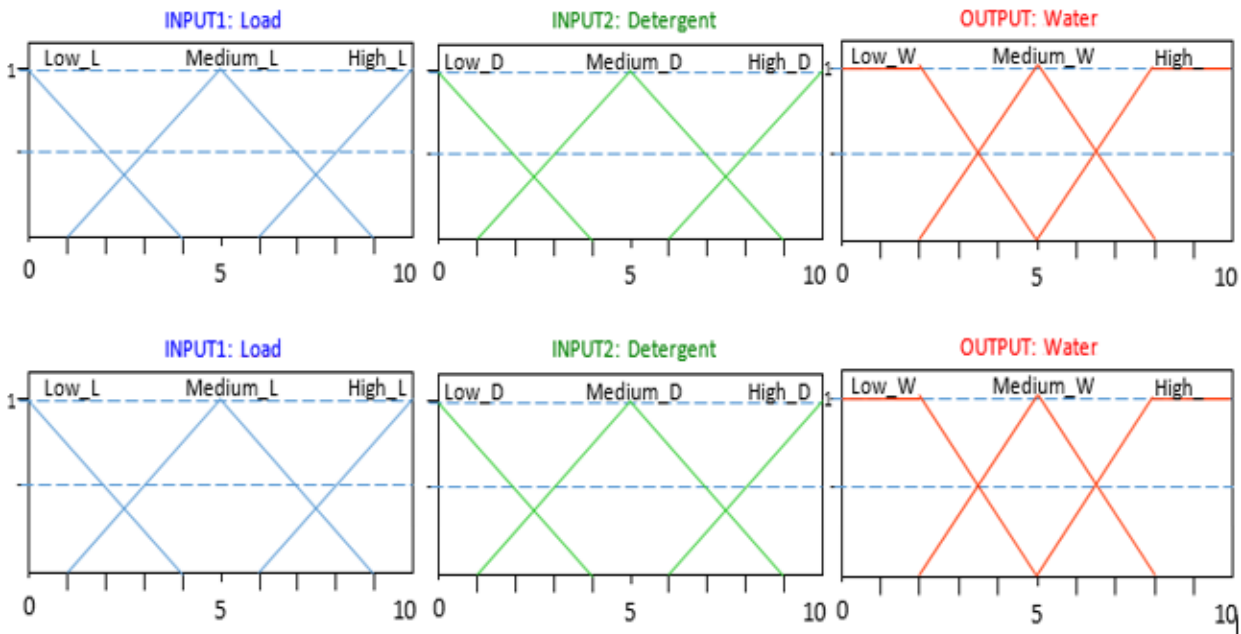
C) Draw a sketch of the *aggregated* fuzzy output set   in the Figure on the right hand side.
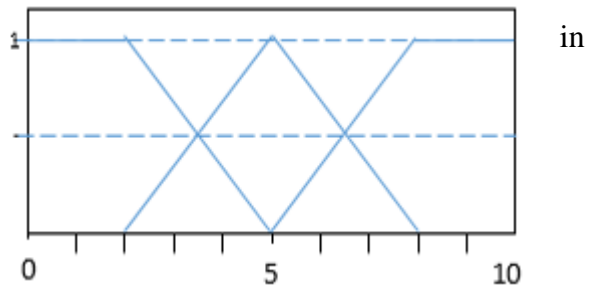
D) Calculate or approximate the crisp output value (with precision ± 0.5) and write down a formula how it could be calculated (The *defuzzification* method applied is *Center-Of-Gravity* (centroid)).