THE UNIVERSITY OF MELBOURNE

Semester 1 Assessment, 2022

Department of Electrical and Electronic Engineering

ELEN90066 Embedded System Design

Reading/ printing time: 30 minutes, Writing time: 180 minutes

Scanning and submission time: 30 minutes

This examination paper has 11 pages

Authorised materials:

This is a Zoom-supervised, hand-written exam where only the following materials are permitted:

- Printed copy of this exam paper and/or an offline electronic PDF reader.
- Any material available in hardcopy or stored electronically on a device disconnected from communication networks.
- A4 paper, pens, pencils, ruler. Electronic devices may not be used for writing your answers.
- Any calculator model.

Instructions to students:

- Attempt **ALL** questions.
- The questions carry weight in proportion to the marks stated for each question number. These marks total 100 marks.
- During Reading Time, you may print out a hardcopy of the exam. Alternatively, you may download the exam to an electronic PDF reading device, which must then be disconnected from the internet.
- During Writing Time you may only interact with the device running the Zoom session with supervisor permission. The printed test paper (or offline device containing the test paper) and any other working sheets must be visible to Zoom invigilators.
- Write your exam answers using pens/pencils and A4 paper. Start each question on a new page and write down the question numbers.
- During Scan/Upload Time, assemble your pages in question number order, and use a mobile phone or tablet to scan and combine them into a single PDF file. Check that all pages are included and clearly readable.
- Submit your PDF file to the Gradescope Assignment corresponding to this exam. Confirm with your Zoom supervisor that you have received confirmation of your submission.
- Collusion is **not allowed under any circumstances**. Collusion includes, but is not limited to, talking to, phoning, emailing, texting or using the internet to communicate with others.
- Plagiarism, through the use of sources without proper acknowledgement or referencing, **is not permitted** and can attract serious penalties. Plagiarism includes copying and pasting from the Internet without clear acknowledgement and paraphrasing or presenting someone else's work as your own.

Question 1 (13 marks)

Consider the representation of a synchronous scheduler interacting with a set of tasks shown by the diagram in Figure 1, where $i \in \{1, N\}$ and N is the number of tasks in the system.

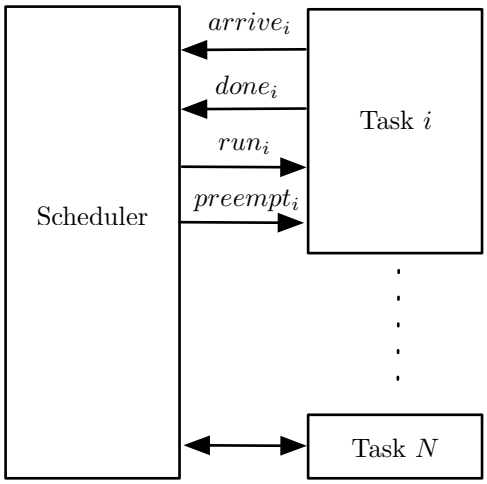


Figure 1: Interaction between scheduler and a number of tasks

Each task is an independent process that communicates with the scheduler process responsible for allocation of processor time via events on its output lines. Assume that the scheduler is scheduling tasks for a uniprocessor system, and that only one task can be running at any one time.

Initially a task is idle. When task i needs processing time, it communicates with the scheduler using the pure signal $\mathbf{arrive}_i$ and will then wait until the scheduler decides to allocate the processor to the task. The scheduler notifies the task of this using the pure signal $\mathbf{run}_i$ and task i proceeds to run.

When the current instance of the task finishes its execution, it communicates with the scheduler using the pure signal $\mathbf{done}_i$ and returns to being idle.

The scheduler is able to preempt task i before it is complete via the pure signal $\mathbf{preempt}_i$ and then allocate the processor to another task j that had already arrived using the $\mathbf{run}_j$ signal.

(a) **[4 marks]** Draw a three-state Finite State Machine representing the operation of a task under this model with the following inputs and outputs:

input: run_i, $preempt_i$: pure
outputs: $arrive_i$, $done_i$: pure

(b) **[6 marks]** Consider the case where there are only two tasks and Task 1 has higher priority; that is, if Task 2 is running and Task 1 needs processing time, Task 2 will be preempted and Task 1 will be given processor time. Draw a Finite State Machine representing the operation of the scheduler under this model with the following inputs and outputs:

inputs: $arrive_1$, $done_1$, $arrive_2$, $done_2$: pure
output: run_1, $preempt_1$, run_2, $preempt_2$: pure

(c) **[3 marks]** Consider the LTL expression $\mathbf{G}\neg(run_1 \wedge run_2)$. Is this an invariant, safety and/or liveness property? Does it hold for the FSM you designed in part (b)? Explain your answer.

Question 2 (16 marks)

A basic obstacle avoidance control algorithm for a robot is modelled by the Finite State Machine (FSM), S, given below

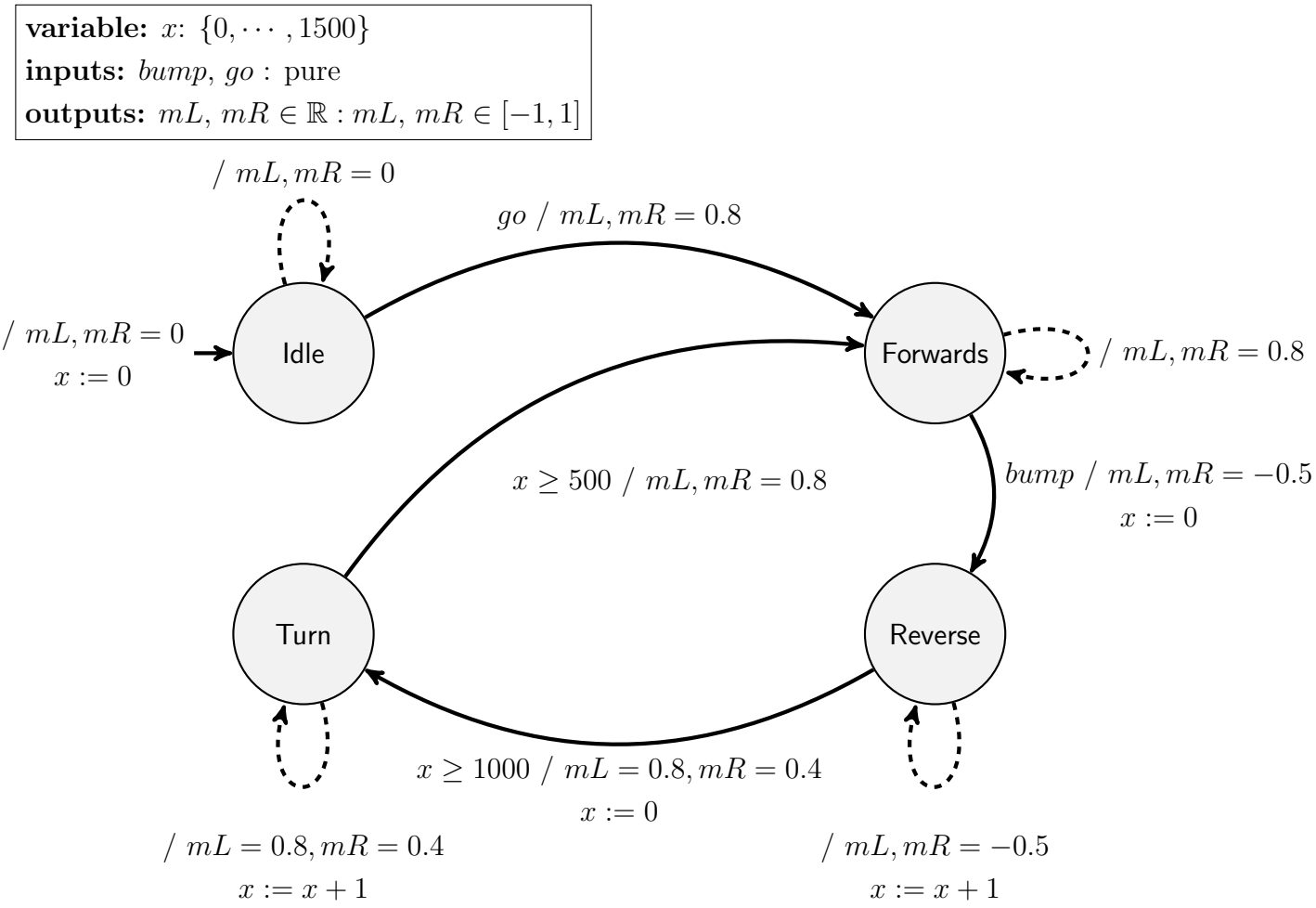


Figure 2: Robot FSM for Question 2

The input *bump* models a momentary contact switch by a pure signal that is only present when the front of the robot has bumped into an object. The input *go* is a pure signal that results only when a 'go' button is pushed on the robot's body in order to initiate its obstacle avoidance manoeuvrers.

The motor outputs mL and mR are real number valued in the range $[-1, 1]$ that represent a fraction of full speed for the left and right motors, respectively, with a negative number indicating the motor is to run in reverse. For example, $(mL, mR) = (1, 1)$ would drive the robot straight ahead at full speed, $(mL, mR) = (1, 0.5)$ would cause the robot to make an arc turn to the right, and $(mL, mR) = (-0.5, -0.5)$ would cause the robot to reverse at 50% speed.

The model is *time triggered* by a signal that causes the machine to react once every millisecond.

Question 2 continues over the next page.

Question 2 (continued)

(a) **[2 marks]** How many states does this state machine have?

(b) **[2 marks]** Give an *execution trace* of the Finite State Machine for the input sequence for $(go, bump)$ given by:

$((present, absent), (absent, absent), (absent, absent), (absent, present), (present, absent))$.

(c) **[8 marks]** For each of the following LTL formulae, determine whether it is TRUE or FALSE for the FSM given in Figure 2. You **MUST** give an explanation for your answer. If FALSE, give a counterexample.

(i) $\mathbf{G}(mL < 1)$

(ii) $\mathbf{F}(mR \leq 0)$

(iii) $\mathbf{FG}(mL > mR)$

(iv) $\mathbf{GF}(\neg bump) \implies (\neg \mathbf{F}\,\mathsf{Reverse})$

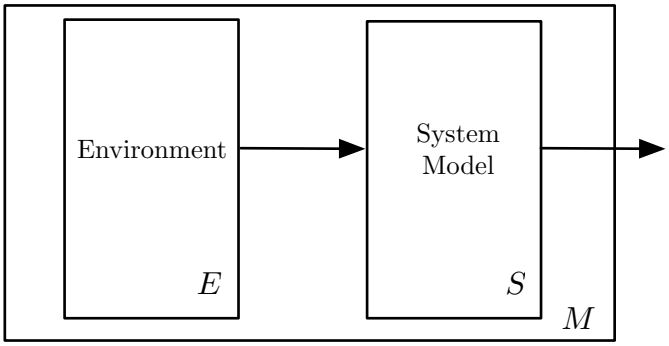
(v) $(\mathbf{G}\,bump) \implies \mathbf{GF}(x > 0)$

(vi) $(m_L \geq m_R)$

(vii) $(m_L = m_R)\mathbf{U}(bump)$

(viii) $\mathbf{G}(\mathsf{Forwards} \implies \mathbf{X^2}\mathsf{Turn})$

(c) **[4 marks]** In order to perform model verification, a model of the environment, E, needs to be constructed so that composing it with the robot control FSM, S, from Figure 2, yields a closed system M. This can be represented by the following block diagram:



Devise an FSM model of the environment, E, that provides inputs to the FSM, S, that would then be suitable for model verification against some specification. Be sure to specify the outputs of the model and their types.

Question 3 (18 marks)

Consider a model of a *synchronous buffer* that accepts integers in the range $[0, 127]$, via an input port `in`, and outputs them via an output port `out` after a certain delay. A block diagram of the input and output of the buffer is shown in Figure 3.

input: `in` $\in \{0, \ldots, 127\}$
output: `out` $\in \{0, \ldots, 127\}$

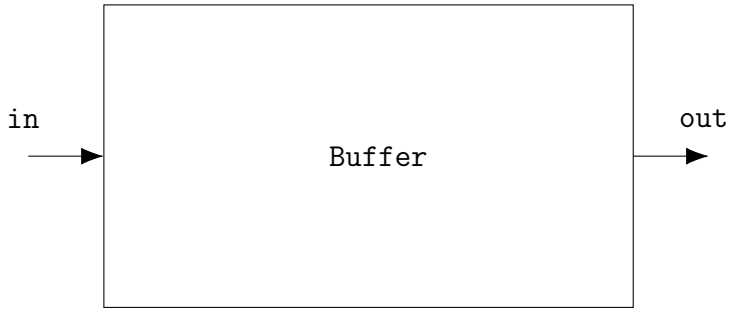


Figure 3: Synchronous buffer for Question 3

The buffer is *time triggered* – that is, it reacts *once* on every positive edge of a periodic clock signal (not shown on the block diagram). The buffer has a capacity of one integer, so that whenever an input value is supplied to the empty buffer it is stored in an internal memory, causing the buffer to be full. When the buffer is full, it simply ignores (and therefore loses) further inputs until it gets a chance to output the stored value to the output `out`.

The delay between the reception of an input value and the transmission of the corresponding output value is at least `LB` and at most `UB` time units, respectively. In other words, the buffer can produce the output only after the *lower bound* `LB` (time units) on the delay and is guaranteed to produce the output within the *upper bound* `UB` (time units) of receiving an input. It is assumed that each time unit is equal to one clock cycle.

(a) **[5 marks]** Draw an Extended Finite State Machine modelling the behaviour of the synchronous buffer as described above. You can consider the buffer to have two main states, `Empty` and `Full`. Be sure to list any variables defined as part of the machine.

How many states would the Extended Finite State Machine have if it were converted to a regular Finite State Machine, i.e. one with no variables in use? Explain your answer.

Question 3 continues over the next page.

Question 3 (continued)

(b) **[10 marks]** Consider two instances of the synchronous buffer connected in series as shown in Figure 4.

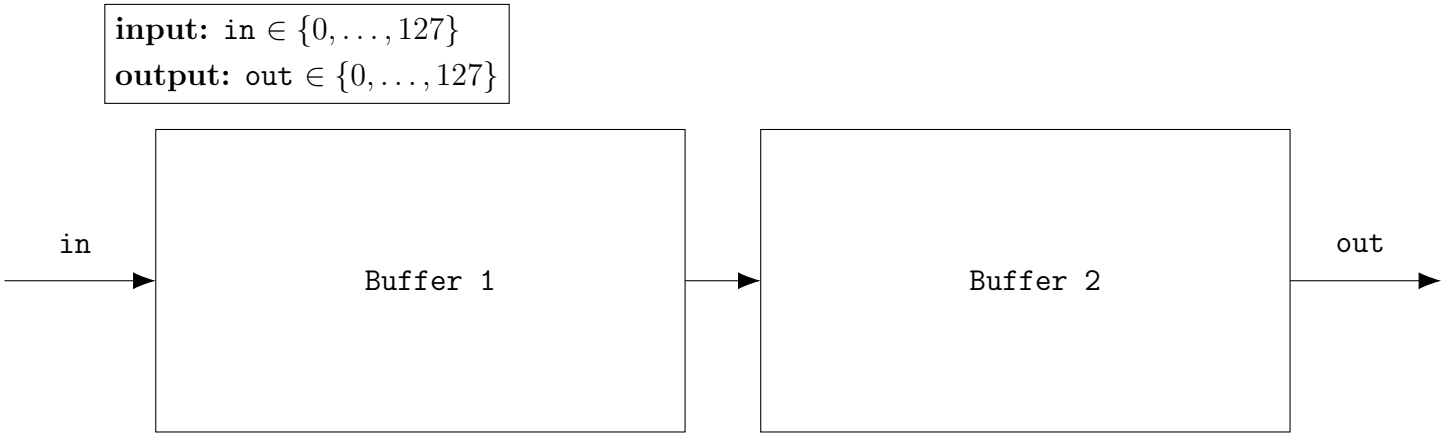


Figure 4: Synchronous buffers in series

The delay between the reception of an input value and the transmission of the corresponding output value is at least $\texttt{LB}_\texttt{i}$ and at most $\texttt{UB}_\texttt{i}$ time units, respectively, for each buffer, where $i \in \{1, 2\}$ refers to the buffer number.

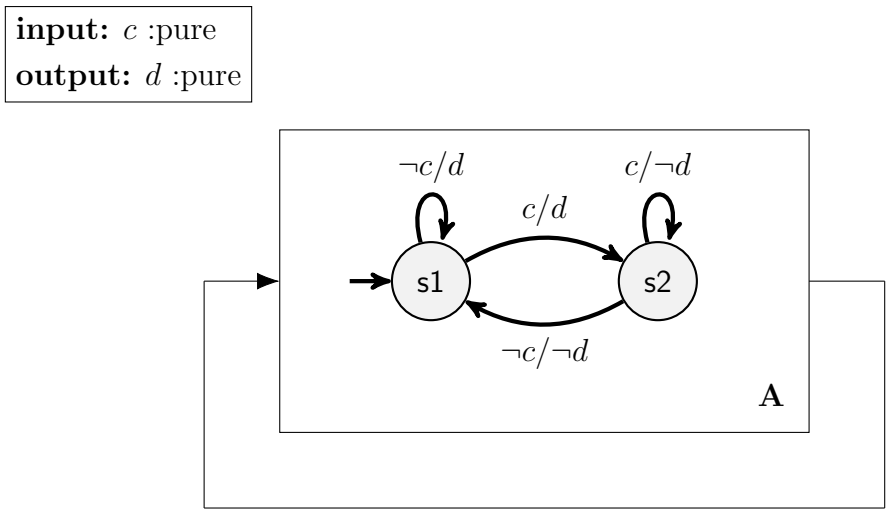
Draw the flattened, Extended Finite State machine resulting from this composition. Be sure to list any variables defined as part of the composed machine.

(c) **[3 marks]** Assume that $\texttt{UB}_1 < \texttt{LB}_2$ for the composition in Figure 4. Does this assumption always guarantee that no information is lost in between the buffers (i.e. Buffer 2 not being ready to accept Buffer 1's output)? Explain your answer, either by providing a proof or a counterexample.

Question 4 (10 marks)

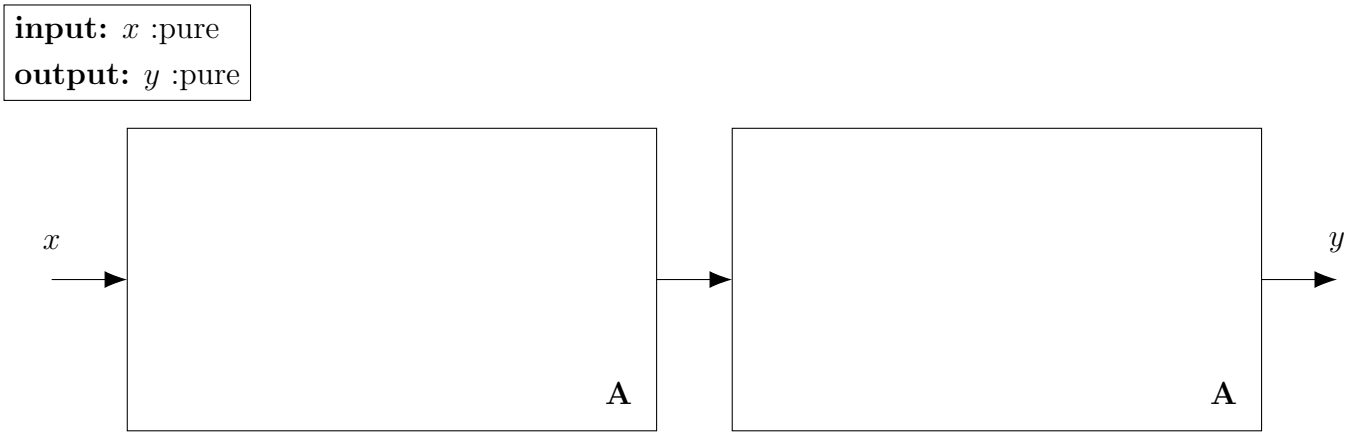
For all compositions in this question, you are to assume a Synchronous Reactive Model of Computation.

(a) **[2 marks]** Consider a system **A** modelled by the Finite State Machine (FSM) inside the rectangle, connected with the output being fed back into the input:



Is the feedback composition is well-formed? Explain your answer.

(b) **[4 marks]** Consider the cascade composition of two instances of system **A**. Let the input and output of the cascaded system be x and y, respectively, as shown in the block diagram below:



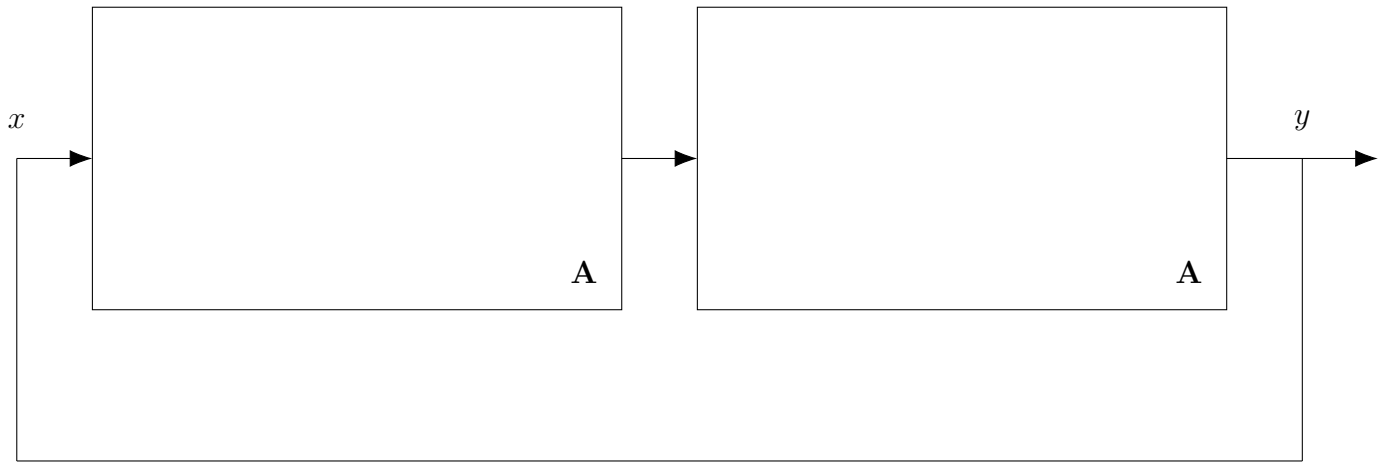
Construct a single, flattened FSM representing the composition of the cascade interconnection.

Question 4 continues over the next page.

Question 4 (continued)

(c) **[4 marks]** Consider now the the feedback loop $x = y$ around the cascade composition from part (b), as shown in the block diagram below:

input: x :pure
output: y :pure



Show that the feedback composition above is well-formed and construct the flattened Finite State Machine model for the composite system.

Question 5 (16 marks)

Consider six tasks $\tau_1, \ ..., \tau_6$ with execution times, and deadlines presented below. Assume that all tasks are released at time $t = 0$.

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
execution time, e_i	2	3	1	3	3	2
deadline, d_i	6	5	8	10	12	14

(a) **[2 marks]** Does a scheduling approach based on Earliest Deadline First (EDF) yield a feasible schedule? Explain why if the answer is 'no', or construct the schedule if the answer is 'yes'.

(b) **[4 marks]** We say task τ_i precedes task τ_j if the predecessor must complete the execution of τ_i before τ_j can execute. The precedence relationships between tasks τ_1, ...,τ_6 are as the following

- τ_1 precedes τ_2;
- τ_2 precedes τ_4;
- τ_2 precedes τ_3;
- τ_3 precedes τ_5;
- τ_4 precedes τ_5;
- τ_5 precedes τ_6.

Draw the directed acyclic graph representing the precedence of the tasks where a directed edge marks a precedence.

(c) **[8 marks]** Now consider the same list of tasks $\tau_1, \ ..., \tau_6$, but now with release times, execution times, and deadlines as given in the table below:

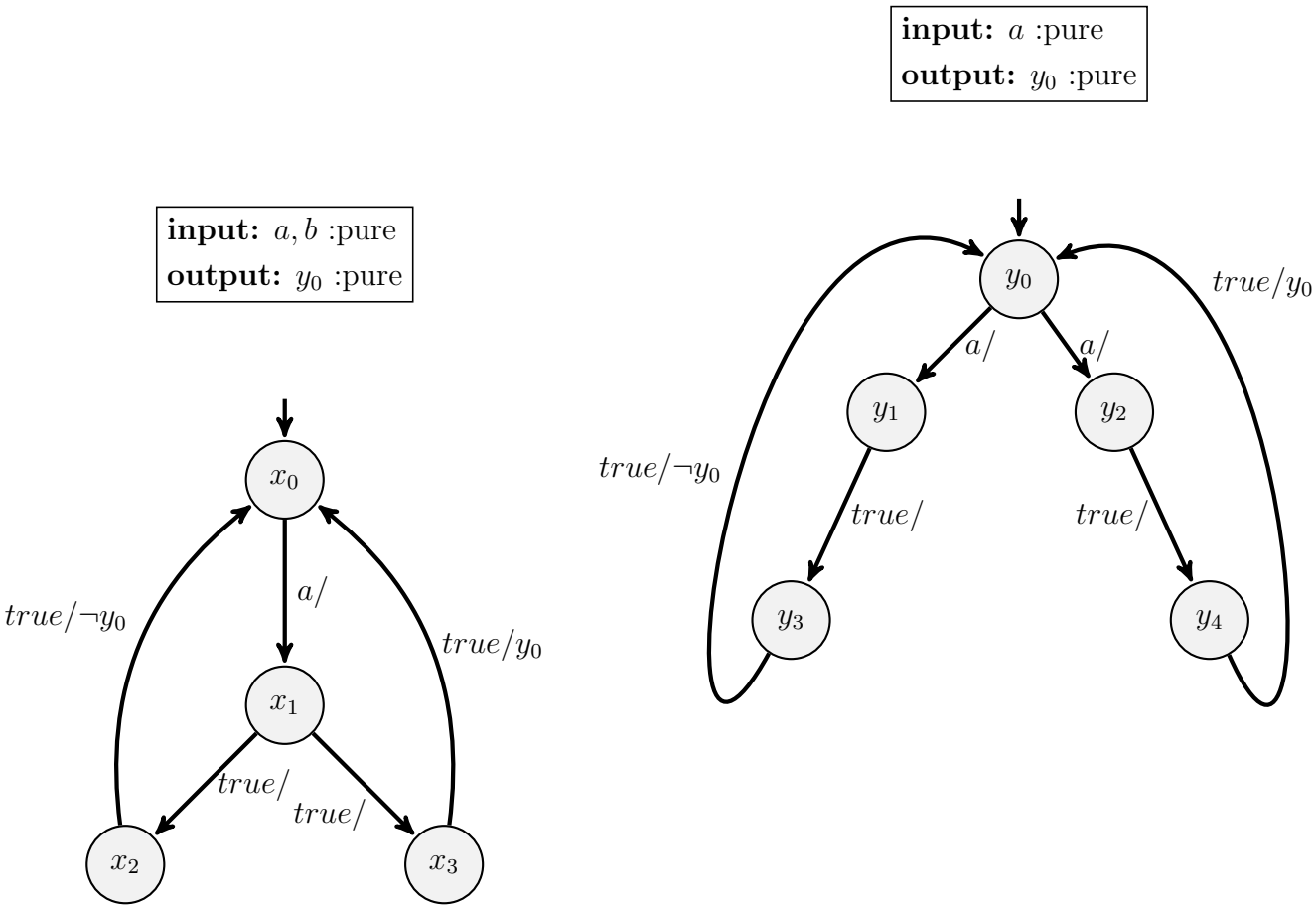
	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6
release time, r_i	0	2	7	3	4	2
execution time, e_i	2	3	1	3	3	2
deadline, d_i	6	5	8	10	12	14

Does Earliest Deadline First with Precedence (EDF*) yield a feasible schedule for the precedences given in (b)? Explain why if the answer is 'no', and construct the schedule if the answer is 'yes'. You must provide sufficient reasoning for your answer.

(d) **[2 marks]** What is the maximum lateness for the scheduling scheme implemented in (c)? Is this value minimised? Explain your answer.

Question 6 (12 marks)

Consider Finite State Machines Σ_1 and Σ_2 depicted in Figure 5.



(a) System Σ_1 (b) System Σ_2

Figure 5: Systems Σ_1 and Σ_2 for Question 6.

(a) **[5 marks]** Is Σ_1 simulated by Σ_2? Prove your answer.

(b) **[5 marks]** Is Σ_2 simulated by Σ_1? Prove your answer.

(c) **[2 marks]** Are Σ_1 and Σ_2 bisimilar? Prove your answer.

Question 7 (15 marks)

Figure 6 depicts an above view of the area of operation of a robot, where at any one time, the robot is in one of the square regions R_i, where $i \in \{1, 2, \ ..., 6\}$.

R_2	R_4	R_6
R_1	R_3	R_5

Figure 6: Area of operation of a robot for Question 7.

The state of the robot can be considered to be the region it is currently in. On every reaction, the robot can either remain in its current region or move only to another region that shares a common edge with its current region. For example, it would NOT be permitted to make the diagonal move $R_1 \rightarrow R_4$ in a single reaction as R_1 and R_4 are not adjacent and do not share a common edge. Let the logical proposition r_i correspond to the robot being in region R_i.

(a) [**4 marks**] Draw a non-deterministic Finite State Machine (FSM), with no inputs or outputs, that models the robot's behaviour. Assume that the robot starts in R_1.

(b) [**4 marks**] What does the LTL formula ϕ as defined below mean? Provide both a natural language translation, and an interpretation, of the specification.

$$\phi := r_i \wedge \mathbf{F} r_2 \wedge \mathbf{F} r_3 \wedge \mathbf{G} \neg r_6$$

(c) [**3 marks**] Is it possible for the robot to meet specification ϕ? Explain.

(d) [**4 marks**] Consider now the case where there is a single obstacle. The obstacle is initially located in R_3, and, on every reaction, it can either remain in its current region or move only to another region that is *adjacent* to the region it is currently in (i.e. it moves similarly to the robot). The obstacle is shown as a shaded square in R_3 in Figure 7.

R_2	R_4	R_6
R_1	R_3	R_5

Figure 7: Indicating initial obstacle position for Question 7.

The robot and the obstacle cannot co-exist in the same region at the same time and therefore neither can move into the other's region on a particular reaction.

If the starting region of the obstacle is R_3, and the robot starts in R_1, can the LTL specification $\mathbf{GF} r_5$ be satisfied by the robot? You must explain your answer to receive marks.

END OF EXAMINATION