CS 4271 Critical Systems and their Verification

AY 2010-11 Midterm Examination, Dated 2nd March 2011.

Time: 1 hr 30 minutes

Instructions to Candidates

1. **This is an open-book examination.**
2. **Answer all questions.**
3. **Answer to any question should be written in the space allotted below the question. It is alright to use the blank pages on the other side.**
4. **It is perfectly alright to use pencils, as long as I can read what you wrote.**
5. **Please write only your matriculation number below.**

MATRICULATION NUMBER:

Question A: 4 marks

Question B: 4 marks

Question C: 6 marks

Question D: 4 marks

Question E: 2 marks

Question F: 5 marks

TOTAL: 25 marks

Question A [4 marks]

Consider a situation that occurs in a programmed IO environment. The CPU issues a WRITE command to the I/O module which then checks if the peripheral is free. The request may either be accepted or rejected depending on whether the peripheral is free or not. If the request is accepted, the CPU sends the data to the I/O Module to write to the peripheral. If the peripheral is not free, the I/O module makes the CPU wait till the peripheral is free.

Show sample interactions of the above system using Sequence Diagrams.

Answer:

Question B. [4 marks]

Present an intra-component view of CPU and I/O module as a UML State Diagram.

Answer:

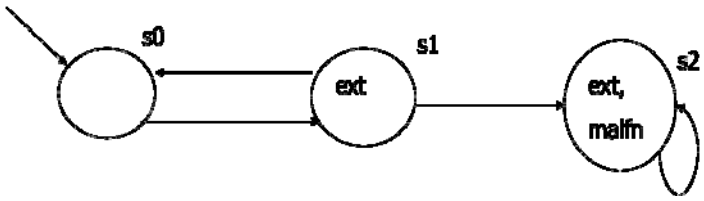
Question C. [6 marks]

Show that the following pairs of temporal logic formula are not equivalent. You should construct an example system model which satisfies one of them but not the other. You may assume that p,q are atomic propositions.

- i) FG p and GF p
- ii) (true U p) and (false U p)
- Iii) (p U q) and (q R p)

Answer:

Question D [4 marks]



Consider the above state machine which represents a simplified model of a spring controller. There are two atomic propositions *ext* and *malfn*. The proposition *ext* is true whenever the spring is extended, and the proposition *malfn* is true when the spring is malfunctioning. The true/false valuations of the propositions in the individual states of the state machine are shown.

i) Formally state the property "Whenever a spring is extended, it eventually malfunctions".

ii) Is the property true for the spring model shown above. Explain your answer.

Answer:

Question E [2 marks]

Consider a traffic light controller which initially shows green light. It senses traffic data every second. Thus, starting from time=0, the traffic is sensed at time = 1 sec, 2 sec, and so on. If there is no traffic movement, the light turns from green to yellow. If there is traffic movement, the light stays green, but it can stay green for a maximum of 3 seconds at a stretch. The light always stays yellow for exactly one second after which it turns red. Once the light is red, again traffic is sensed every second. If there is traffic, then the light becomes green after staying red for a minimum of 2 seconds. If there is no traffic, the light eventually becomes green, after staying red for 3 seconds.

Suppose you were trying to model the above controller as a state machine. Is it possible for you to model the exact requirements as stated above as a state machine? *What aspects of the requirements can you model and which aspects you cannot? Explain your answer.*

Answer:

Question F [5 marks]

Consider the following Promela specification of a simple producer and consumer. Assume that both of them access a buffer which serves as a shared data structure.

```
toktype = {P, C};
toktype turn = P;

active proctype producer()
do
:: (turn == P) -> /*produce 1 item*/; turn = C;
od
}

active proctype consumer() {
do
    :: (turn == C) -> /* consume 1 item */; turn = P;
od
}
```

If there is one producer process and one consumer process, will an interleaved execution violate mutual exclusion of access to the buffer? To answer this question, *construct the global state transition system after assigning labels to the control locations of the processes.*

Answer:

-END OF PAPER-