1. (60 points)
Solve Exercise 1 of Chapter 14 of the textbook *Introduction to Embedded Systems* by Lee and Seshia. The book is available in PDF from here: [http://leeseshia.org/](http://leeseshia.org/). The exercise is repeated here, but you may want to consult the textbook as well to appreciate the context.

Consider the system $M$ modeled by the hierarchical state machine below which models an interrupt-driven program.

variables: $\text{timerCount: uint}$
input: $\text{assert: pure, return: pure}$
output: $\text{return: pure}$

Model $M$ in the modeling language of the verification tool SPIN (see installation instructions below). You will have to construct an environment model that asserts the interrupt. Express the following property as an LTL formula $\phi$:

$\phi$: The main program eventually reaches program location $C$.

Use SPIN to check whether $M$ satisfies $\phi$. Explain the output you obtain from the verification tool.
2. (40 points)

The following is a DE (discrete-event) model.

![Diagram of DE model]

The behavior of the actors in the model is as follows (ignore all ports which are not connected):

- **DiscreteClock** produces at its output port a sequence of periodic events with value 1 and period 1, starting at time 0.

- **Ramp** produces at its output port a sequence of events with values 0, 1, 2, ..., at times \( t_0, t_1, t_2, \ldots \), where \( t_i \) is the time that Ramp received an input event at its trigger input port. That is, for every input event it receives, Ramps produces an output event with the same timestamp. The value of the \( n \)-th output event of Ramp is \( n - 1 \). The value of its first output event is 0.

- **AddSubtract** produces at its output an event \((v, t)\), where \( v \) is the value and \( t \) the timestamp, as follows:
  - If AddSubtract receives two simultaneous input events \((v_1, t), (v_2, t)\), at ports + (plus) and − (minus) respectively, where \( v_1, v_2 \) are the values and \( t \) is the timestamp, then it produces an event \((v_1 - v_2, t)\) at its output.
  - If input event \((v, t)\) is received only at port + any there is no simultaneous event at port −, then event \((v, t)\) is produced at the output.
  - If input event \((v, t)\) is received only at port − any there is no simultaneous event at port +, then event \( (-v, t)\) is produced at the output.

- **TimeDelay** produces an event \((v, t + 1)\) at its output for every event \((v, t)\) that it receives at its (connected) input.

There are four channels in the model, denoted a,b,c,d in the figure.

(a) **Write the actor dependencies for this DE model.** Write a dependency as \( A \rightarrow B \) if actor \( B \) depends on \( A \). You may assume that for the source actor DiscreteClock we have \( \text{DiscreteClock} \rightarrow \text{Ramp} \).

(b) **Perform discrete-event simulation on this model up to (and including) time 6, and write down the set of events produced on each of the four channels.** Assume that all output events of DiscreteClock up to time 6 are generated upon initialization. Draw a horizontal axis for time, and a vertical “axis” for the four channels, \( a, b, c, d \). Write an event with
value \( v \) occurring in channel \( x \) at time \( t \) as a point \( v \) with coordinate \( (t, x) \). You don’t need to show the steps of the discrete-event simulation. Write only the events. Write only events with \( t \leq 6 \).

(c) For all events from item (b) that occur at some time \( t \leq 1 \), draw the dependencies between simultaneous events. Draw a dependency \( e_1 \rightarrow e_2 \) between events \( e_1 \) and \( e_2 \) if \( e_1 \) and \( e_2 \) are simultaneous (i.e., happen at the same time \( t \)) and \( e_1 < e_2 \) according to the precedence relation on events defined in the lecture. You do not need to draw dependencies that can be derived by transitive closure.

(d) Write the order in which the events from item (c) are processed by the discrete-event simulation algorithm. Are there simultaneous events which are processed together as a set and if so which ones?

3. (30 points)

Reduce the question

Given timed automaton with actions labeled with events \( a, b, \) or nothing, check whether in all behaviors of the automaton, every \( a \) is followed by \( b \) within at most 5 but no earlier than 2 time units (a single \( b \) can “cancel” many previous \( a \)'s).

to the basic control-state reachability question for timed automata.

4. (30 points)

Consider a different definition of the region graph, based on an equivalence with “squares” rather than “triangles”, as shown in the figure below:

For instance, the point marked with “\( \times \)” and the point marked with a dot in the above figure are considered equivalent.

Show that the region graph based on this “square equivalence” does not preserve control-state reachability.

This means that you should find a timed automaton which has a control state \( q \) which is unreachable, yet in the square-region graph of that automaton there is a reachable state \( (q, r) \).
5. (30 points)
Consider a modification of “Master Algorithm A” of slide 78 of the lecture slides http://embedded.eecs.berkeley.edu/eecsx44/lectures/continuous.pdf, where Step 4.1 is replaced by

\[ h' := \text{doStep}_c(h); \]

Does the modified version of the algorithm satisfy the determinacy property of slide 80 under assumptions (A0)–(A3) of slide 79? If not, provide a counter-example.

Does the modified version satisfy determinacy if, in addition to (A0)–(A3), we also assume (P1) of slide 80?

6. (30 points)
Summarize progress on your course project (at most 1 page). Your report should contain at least the following points:

- Results obtained so far (including partial theoretical/experimental results, benchmarks gathered, etc.);
- Changes in direction, if any;
- A week-by-week plan for the rest of the project.

IMPORTANT INSTRUCTIONS:
To install Spin, follow installation instructions found at:
http://spinroot.com/spin/Man/README.html
To become familiar with the Promela syntax and the usage of Spin, it is recommended to first read:
http://spinroot.com/spin/Man/GettingStarted.html
Additional information can be found from various links on the homepage:
http://spinroot.com/spin/whatispin.html