Homework 2
Analysis of Software Artifacts
Due Date: Oct 18, 2002 (Friday)

Question 1 (Review 35 points): This question is a review of all the specifications we did in class. If you attended the class, this should be easy.
(Part A:) Explain the specification given below:

$$\mathbf{AG}\,[(\mathit{setup} \wedge \neg \mathit{started}) \rightarrow \mathbf{A}(\mathit{setup}\mathbf{U}\mathit{started})]$$

The proposition *setup* models the fact that we are trying to setup a transaction. Proposition *started* denotes the start of the transaction.

(Part B:) Explain the specifications given below:

$$\mathbf{AG}(\mathbf{AF}p)$$
$$\mathbf{AG}(\mathbf{EF}_{[\mathbf{0},\mathbf{10}]}p)$$

(Part C:) Show a model with *two states* such that $\mathbf{AG}(\mathbf{EF}p)$ is true and $\mathbf{AG}(\mathbf{AF}p)$ is false in the model.

(Part D:) Is there state-machine/model such that in the model $\mathbf{AG}(\mathbf{AF}p)$ is true and $\mathbf{AG}(\mathbf{EF}p)$ is false? Please justify your answer.

(Part E:) Write a CTL and $RTCTL$ specifications for the following property:

> If a transaction $T1$ is issued at a state, then within 10 cycles/time-steps transaction $T2$ is always issued. Assume that there are atomic propositions $T1.\mathit{issued}$ and $T2.\mathit{issued}$ corresponding to transaction $T1$ and $T2$ being issued.

Question 2 (Real-time scheduling 15 points): Assume that we have two periodic tasks τ_1 and τ_2. Task τ_1 has execution time 4 and period 10, and τ_2 has execution time 6 and period 14.

(Part A:) Determine whether τ_1 and τ_2 are schedulable using the rate monotonic scheduling or RMS.

(Part B:) Now assume that the execution time for τ_2 is increased from 6 to 8. How can you ensure that both the task's meet their deadlines without reducing execution time? (*Hint:* use period transformation.)

Question 2 (SMV 50 points): Modify the *vending machine* example provided in class in the following way:
Multiple Users: Model two users in the specification. Add an additional state `waiting` which signifies that the user is waiting for the vending machine. You should only have one `MODULE` description for the user. Two users are simply instantiations of this module. If two users are simultaneously waiting for the vending machine, the vending machine non-deterministically picks a user.

Indecisive user: A user can change his/her mind in the middle of the process (after depositing one or two coins). In this case the vending machine should return the coins and (of course) not dispense the drink.

Specification: Express the property given below in *CTL*
if a user is waiting, he/she always eventually gets the drink unless he/she changes their mind in the middle.

Checking: Using NuSMV check that the specification is true. If the specification is not true, explain the counter-example.

Logistics: Before you start changing the SMV model, write a *design document.* Outline your design strategy in the design document. Please be as detailed as possible. Show the design document to a fellow student, i.e., the *reviewer.* Mention the reviewer's name on the document. Keep the design document *consistent* with your specification. Please submit the *modified SMV specification AND the design document* with the homework. This will also help us grade your homework and will help you with the homework.