CS132: Software Engineering

Midterm Exam

13:00-14:40, May 12th, 2021

There are 5 problem sets and the total points are 80 points. Your score will be divided by 4 to reflect the 20% constitution for the midterm. Each problem set includes a few questions. For each question, the maximum possible points are stated.

Please write your answers **legibly on the answer booklet** so that we can read and understand your answers. If a problem seems ambiguous, please feel free to state your assumption explicitly and solve the problem. Obviously, your assumption should be reasonable and should not trivialize the problem.

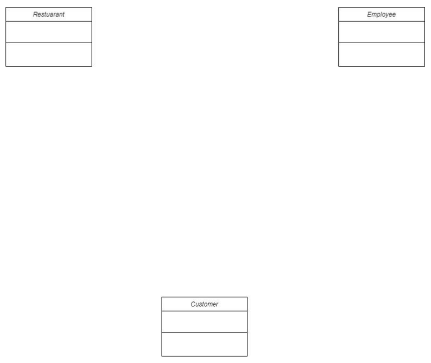
Pledge. Copy the following pledge and sign your name in your answer booklet:

I neither cheated myself nor helped anyone cheat on this exam.

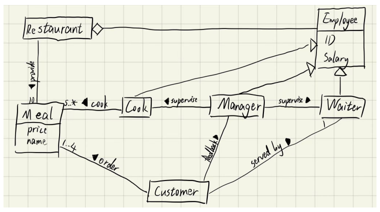
Problem 1. (15 pts)

Suppose there is a new restaurant going to be open in the Greenland Plaza. The customers are served by one waiter. There are 10 items on the menu, and each item has a name and a price. Each customer orders 1 to 4 items, which are cooked by several cooks. Each cook should be able to cook at least 5 items in the menu. The customer can provide feedback to a manager, who's supervising the waiters and cooks. Managers, waiters and cooks are employees of the restaurant. Each employee has an ID and a salary.

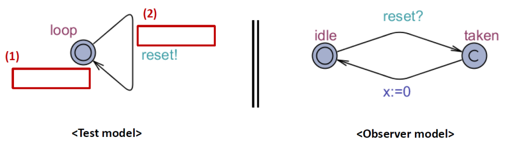
A partial class diagram is given to you below. Please complete the class diagram. Make sure to show attributes, multiplicities and aggregation associations, where appropriate. There is no need to show any operations.



Solution:



Problem 2. (10 pts)



The above Test and Observer models are composed for parallel execution. The channel, `reset`, is used for synchronization between the two models. One global clock, `x`, is declared.

Fill in (1) the **invariant** of loop location, (2) the **guard condition** of the transition in Test model so that the following two UPPAAL queries are satisfied:

(a) A□ `observer.idle imply` $(x \leq 10)$

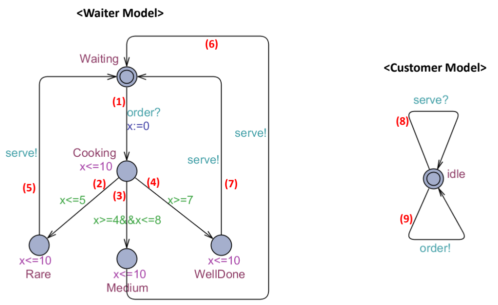
(b) A□ `observer.taken imply` $(x \geq 5$ `and` $x \leq 10)$

If no condition is required in (1) or (2), write "none."

Answer:
(1) $x \leq 10$, (2) $x \geq 5$

Grading - one correct answer gets 5pt; if students misundertood the problem and reasoning is partly correct give up to 2pt.

Problem 3. (25 pts)



A steak restaurant is a good place for customers to enjoy cooked steak. A steak restaurant has a waiter who can serve the customers following a certain workflow: the waiter accepts an order of steak from a customer by asking the ways to cook the steak (e.g., rare or medium or well-done); and cook the steak for an appropriate amount of time as ordered; and serves the ordered steak (either rare or medium or well-done steak) to the customers.

The left-side UPPAAL model in the figure abstracts the workflow of the waiter; the right-side UPPAAL model abstracts the customer behavior.

Here is a brief description of the model:

The waiter model interacts with the customer model using synchronization [order] and [serve].

The waiter model has one clock variable [x].

The waiter model has five locations:

- **Waiting**: the waiter waits for an synchronization [order], and makes a transition to Cooking location by resetting the clock [x] to 0.

- **Cooking**: Cooking location is associated with an invariant [x≤10]; the waiter makes a transition to **Rare** location if the clock guard condition satisfies [x≤5]; makes a transition to **Medium** location if the clock guard condition satisfies [x≥4 and x≤8]; makes a transition to **WellDone** location if the clock guard condition satisfies [x≥7].
- **Rare**, **Medium**, **WellDone**: the three locations are associated with an invariant [x≤10]; the waiter makes a transition to **Waiting** location upon an synchronization [serve].

Answer the following questions:

1. **(5 points)** Does the following property hold? If no, explain the reason with a counter example.

   A□ not deadlock

   Satisfied.

2. **(5 points)**Does the following property hold? If no, explain the reason with a counter example.

   A□ not (waiter.Rare and x == 4)

   Not satisfied; it is possible to have a clock value 4 in Rare state

3. **(5 points)**Does the following property hold? If no, explain the reason with a counter example.

   A□ not (waiter.Rare and x == 6)

   Not satisfied; it is possible to have a clock value in between [0, 10] in Rare state

4. **(5 points)**Does the following property hold? If no, explain the reason with a counter example.

   A□ not (waiter.Medium and x == 3)

   Satisfied.

5. **(5 points)**Does the following property hold? If no, explain the reason with a counter example.

   E◇ (waiter.WellDone and x == 10)

   Satisfied.

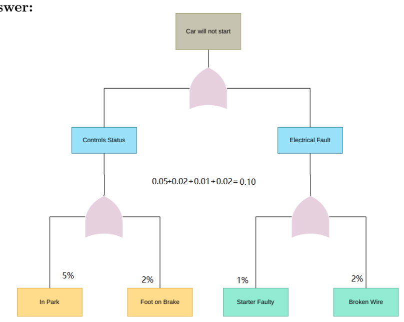
Problem 4. (10 pts)

There are many reasons when a car fail to start. There are operational reasons. The car should be in "Park" gear and the driver should press the brake pedal for the car to start. Electronics failures can be the reason as well. A broken starter or broken wire can both cause electronics failures.

Please answer the following questions:

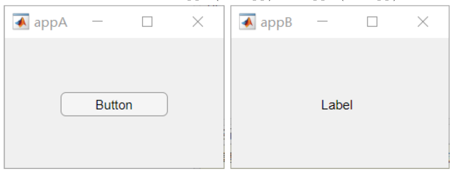
1. **(6 points)** Please draw the fault tree for the fault "Car fail to start".
2. **(4 points)** Assume the probability of the driver forgetting to put the car in "Park" gear is 5%, the probability of the driver forgetting to press the brake pedal is 2%, the probability of a starter failure is 1% and the probability of a broken wire is 2%, calculate the probability of a car failing to start. (You get the full credit if you provide the correct equation and you don't need to calculate the end result)

Answer:



Problem 5. (20 pts)

There are two Matlab APPs: appA(A.mapp) and appB(B.mapp):



When the button in appA is pressed, a function in a Processor object is called, and the Text of the label (app.Label) in appB is changed to "Yes!".

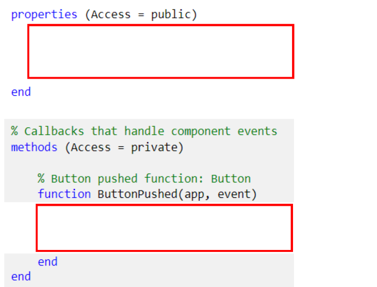
A script initializing the system is defined as follows:

```
close all
clear

pro = Processor;
appA = A;
appB = B;

pro.appB = appB;
appA.processor = pro;
```

Please complete the code in A.mlapp and Processor.m so that the functionality above can be achieved.



```
properties (Access = public)


end

% Callbacks that handle component events
methods (Access = private)

    % Button pushed function: Button
    function ButtonPushed(app, event)


    end
end
```

```
classdef Processor
    properties


    end

    methods
        function setValue(obj)


        end
    end
end
```

Answer:

```
    properties (Access = public)
        processor;
    end

    % Callbacks that handle component events
    methods (Access = private)

        % Button pushed function: Button
        function ButtonPushed(app, event)
            app.processor.setValue();
        end
    end
```

```
classdef Processor
    properties
        appB
    end

    methods
        function setValue(obj)
            obj.appB.Label.Text = "Yes!";
        end
    end
end
```