Department of Computer Science

Computer Science 177: Modelling and Simulation

MIDTERM EXAMINATION — Spring 1998

1. Suppose you want to write a simulation model for an elevator using a general-purpose language like C or C++. For each of the following model components, state whether it is an event, entity, or attribute. Briefly explain how you would represent it in your program.

    a. the arrival of a customer
    An event. It would be represented by a function which gets called whenever a customer arrives.

    b. the floor k on which the customer arrived
    An attribute. It would be represented by a field in the struct that contains data about the customer.

    c. the elevator
    An entity. It would be represented by a struct that contains data about current location of the elevator, whether it is going up or down, how many people are on it, etc.

    d. the number of people in the elevator
    An attribute. It would be represented by a field in the struct that contains data about the elevator.

2. List at least 4 items that should be included in the system state for the elevator simulation in question 1. Briefly describe the sequence of changes to the system state that occur between the arrival of a customer on floor 1 at 2pm, and the departure of the same customer on floor 4, assuming the no other customers tried to use the elevator at the same time.
There is a long list of possible items, including number of customers waiting on each floor, number of customers on the elevator, time of next customer arrival, location of elevator (at a floor or traveling between floors), direction of travel (up/down) for elevator, time of arrival for elevator at next floor.

    For this example, we add a customer to the waiting line on floor 1 at 2pm, and call the elevator if it has not already been done by a previous customer. Elevator travels to floor 1 and opens door. Customer gets on the elevator. Door closes and elevator goes to floor 4. Customer gets off.

3. Consider an experiment where I roll a pair of dice. If the numbers on both dice are the same, I roll again, and so on until the numbers are different.

    a. Give a list of possible outcomes for this experiment.
    The result of rolling of pair of dice once can lead to 36 distinct values, corresponding to a different combination of numbers between 1 and 6 for the value of each die, i.e., (1,1), (1,2), ..., (6,6). In this experiment, however, the values are thrown away and the dice are rerolled as long as the result is one of the following 6 "forbidden" combinations: (1,1), (2,2), (3,3), ..., (6,6). Therefore, the outcome of this experiment must be one of the remaining 30 rolls of the dice: (1,2), (1,3), ..., (1,6), (2,1), (2,3), ...

    b. Find the probability for each outcome.
    Because of symmetry, there is no reason for one outcome to occur more frequently than another, so each has probability 1/30.

    c. An event in probability is a subset of the possible outcomes. For example, finding that the sum of the numbers on the two dice is even is an event. Briefly explain why the probability of rolling an even number is 1/2 if you roll a pair of dice once.

The easiest explanation is simply that half the possible values for a die are even and half are odd. Hence, no matter what value comes up for the first die, half the time it will be even and half the time it will be odd. Now if the second die is even, then their sum is even if and only if the first die is even; if the second die is odd, then their sum is even if and only if the first die is odd. Either way, half the outcomes are even and half are odd.

Another way to approach this is to count up the number of odd and even values among the 36 outcomes. Note that you can't just look at the 11 possible values of their sum, ranging from 2 to 12, which has more distinct even sums than odd sums, because the sums in the middle have higher probability than the sums at the top and bottom.

d. What is the probability of rolling an even number in this experiment, where you reroll as long as the two numbers are the same.
This one is a bit tricky! If you roll a pair of dice once, then there are 36 different outcomes, of which 18 are even and 18 are odd. If you now keep repeating the roll until the two dice come out different, then you have removed 6 of the outcomes from the experiment: (1,1), (2,2), ..., (6,6). Notice that all 6 of the missing outcomes are even! Therefore, we are left with only 12 outcomes for which the sum is even out of 30 possible outcomes where the dice have different values, or a probability of 12/30 or 2/5.

4. This question refers to the CSIM tanker simulation program, which we discussed in class. A copy of the program is attached to the end of this test. Please refer to the line numbers in this program in your answers, for example: "change lines 112–115 to the following."

a. Suppose the number of berths in the harbor is increased from 3 to 4. State the changes that must be made in the program to handle this change to the model.
First, you need to increase the capacity of the "berths" storage on line 47 (notice that on lines 93−95 the tanker allocates one unit of this storage before unloading). Then you need to change the initial value of the variable "NbBerth" from 3 to 4 on line 112 (this is used by the tug to decide if there is room in the berths for another tanker).

b. Now suppose we want to increase the number of tugs from one to two. What is the *minimum change* to the program that will add a second tug to the model. (Don't worry about making changes to the program to make the tugs coordinate their actions in a sensible way. Just add another tug.)
All you need to do is duplicate line 55, which is the function call that creates a tug.

c. Assume that all tugs have been created before the first tanker enters the port queue and calls a tug (line 87). How many tugs will respond? Justify your answer.
Only one tug will respond to the call. The critical statement is at line 122, which is the only place in the program that a tug (or tugs) is/are waiting for a tanker to call for service. Normally, all processes doing a "wait" will be awakened by the appearance of a signal, and you should do "queue(call)" if you want to awaken only one tug. However, notice that line 122 is inside a critical section because it is sandwiched between "reserve(tug_waiting)" and "release(tug_waiting)", so only one tug can ever execute the "wait".

d. Currently, lines 126–154 handle one trip by the tug from the berth area (where ships are unloaded) to the harbor area (where incoming ships enter the model) and back. Currently, it is split into two parts, where the first part (lines 126–145) handles the case of taking a ship from the berth out to the harbor and optionally bringing bringing in another ship, while the second part (lines 146–154) handles the case where the tug goes out to the harbor by itself and brings in a ship.

Combine these two cases into a single block of code that (i) checks to make sure there is at least one ship needing to be moved, (ii) brings one ship out from the berth to the harbor, if possible, or else goes out by itself, (iii) brings a ship into the berths if one is waiting and a berth is available and there is no storm, or else it returns empty.

```
if ( status(berth_queue)==BUSY ) /* if tank waiting at berths */
{
   set(tug_at_berth);      /* tell tanker the tug is there */
   wait(secure);           /* wait for tank to drop tow line */
   NbBerth++;              /* at least one berth is free */
   hold(1.0);              /* trip to port takes one hour */
   set(tug_arrived);       /* tell tanker that the trip is over */
}
else
   hold(0.75);  /* go out to harbor empty */
if ( status(port_queue)==BUSY && NbBerth > 0
       && !(status(NastyWeather)==BUSY))
   {              /* if tanks waiting at port and there is a free berth
      set(tug_at_port);    /* tell tanker that tug arrived at port */
      wait(secure);        /* wait for tank to drop tow line */
      hold(1.0);           /* trip to berths takes one hour */
      set(tug_arrived);    /* tell tank that the trip is over */
      NbBerth--;           /* one more berth is occupied */
   }
}
else
   hold(0.75);  /* return to berths empty */
```

5. Consider a "left triangle distribution" over the range from 0 to L, for which the probability density function (pdf) is shaped like a right-angle triangle with height H and length L, where the three corners located at the points $(0,0)$, $(0,H)$ and $(L,0)$.

   a. Solve for L as a function H. (Hint: Remember that the area of a right-angle triangle is half the length times the width.)

$$area = L * H/2 \equiv 1$$

$$L = 2/H$$

   b. Sketch the pdf for this distribution, and show that it can be written in the form:

$$f(x) = H\,(1 - x/L)$$

   This is pretty obvious if you draw the triangle, and recognize that $f(x)$ must drop linearly from H to 0, as x increases from 0 to $2/H$. In other words, we need $f(L) \equiv 0$, or $H\,(1 - L/L) = 0$ as required.

   c. Briefly explain what a maximum likelihood estimator (MLE) is for.
   A maximum likelihood estimator is the value of some parameter for which the probability of getting the given observations is higher than it would have been for any other parameter value.

   d. Find the MLE for H assuming you have only *one sample point* X_1 available.

First step is to write the pdf as a function of H only, by substituting the value of L:

$$f(x) = H\,(1 - x * H/2) = H - x * H^2/2$$

Next we need to take the derivative of f(x) with respect to H:

$$\frac{df}{dH} = 1 - x * H$$

Setting the derivative equal to zero at a maximum, we get $H = 1/x$ and thus $L = 2 * x$.