Homework 01 Solutions

ECE 587, Spring 2026

1. *(2 points)* The tic-tac-toe game, as shown below, uses a 3-by-3 board. Two players, X and O, take turns to fill an empty board until one of them wins by placing the same three in a row (horizontal, vertical, or diagonal).

X	O	
	X	
		O

To build a FSM model for this game, we use the board itself as the state and the state transition happens when a player places an X or O in an empty box.

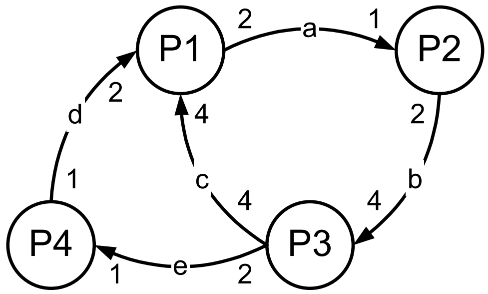
1) How many states are there in the FSM model? What are the initial states and the final states? (Hint: First decide how many ways you could fill a single box. For simplicity you could count a state even if it cannot appear in a valid game, e.g. a board filled with all X.)
2) What are the inputs? What is the next state function?

Answer: Note that there is no single "correct" solution to the homework questions. I would suggest you to compare your model with my model in order to understand if your model is reasonable or not.

For 1), the board has 9 boxes and there are three possibilities for each box (empty, X, and O). So there are 3^9 states. There is a single initial state where each box is empty. There are many final states where one of the players wins, or the game ties if all boxes are filled with O or X.

For 2), each input indicates what (X or O) should be played next and where it should be placed (among the 9 boxes). The next-state function may reject an invalid input by staying at the currect state, or move to the next state by adding the X or O to the box.

2. *(2 points)* Consider the computation specified by the following Synchronous Data Flow (SDF) model consisting of four processes `P1`,`P2`,`P3`,`P4` and five channels `a`,`b`,`c`,`d`,`e`.



1) Determine the relative execution rates of the four processes.
2) Schedule the processes according to the rates computed in 1) (there are many possible schedules and you are free to pick any one). Compute the number of initial tokens required on each channel and the queue size of each channel according to your scheduling.

Answer:

For 1), let p_1, p_2, p_3, p_4 be the firings per round for the four processes respectively. We have

$$\begin{aligned} 2p_1 &= p_2, \\ 2p_2 &= 4p_3, \\ 4p_3 &= 4p_1, \\ p_4 &= 2p_1, \\ 2p_3 &= p_4. \end{aligned}$$

Solve it and we obtain a solution

$$p_1 = 1, p_2 = 2, p_3 = 1, p_4 = 2.$$

Now for 2), let's consider the schedule P1 P2 P2 P3 P4 P4. Assuming that each channel has no token in the beginning, the number of tokens then change as follows.

	a	b	c	d	e
P1	2	0	-4	-2	0
P2	1	2	-4	-2	0
P2	0	4	-4	-2	0
P3	0	0	0	-2	2
P4	0	0	0	-1	1
P4	0	0	0	0	0

So initially there should be 4 tokens on c and 2 tokens on d, and the queue sizes are 2, 4, 4, 2, 2 for a, b, c, d, e respectively.

3. *(1 point)*

   Read the following sequential program that and model one loop iteration by a Data Flow Graph (DFG).

```
double a = ..., b = ..., c = ..., d = ...;
for (...) {
  double s=a+b;
  double t=a-b;
  a=s;
  b=t;
  s=c*d+a;
  t=c/d-b;
  c=s;
  d=t;
}
```

   Answer:

   The key points are to rename the temporary variables so that each of them is only written once, and to rename the inputs/outputs so that they are only updated at the end of the iteration.

   The DFG can be listed per level (or you can actually draw it):

Level 1: s1=a+b, t1=a-b

Level 2: s2=c*d, t2=c/d

Level 3: s3=s2+s1, t3=t2-t1

Level 4: a=s1, b=t1, c=s3, d=s4