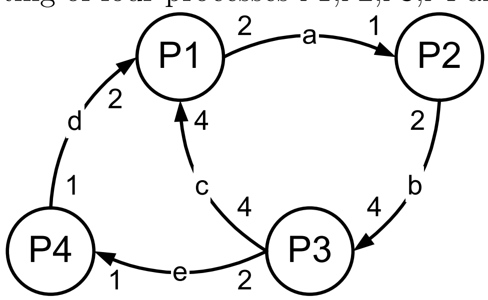
# Homework 02 Solutions

ECE 587, Spring 2025

1. *(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.

2. *(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

3. *(2 points)*

   Once upon a time a farmer went to a market and purchased a fox, a goose, and a bag of beans. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the fox, the goose, or the bag of beans.

   If left unattended together, the fox would eat the goose, or the goose would eat the beans. Will the farmer be able to carry himself and his purchases to the far bank of the river, leaving each purchase intact?

   While you may easily find the answer to this problem online, we are interested in a method that solves this and similar river crossing puzzles. Show that you can model the problem with a FSM and then apply model checking concepts to solve it.

   *Answer:*

   First of all, the states of the problem can be encoded by 4 bits, one each for the farmer, the fox, the goose, and the bag of beans. For each bit, 0 represents one bank and 1 represents the other.

   Therefore, the initial state is 0000 where all are at one bank and the final state is 1111 where all are at the other bank.

   The state transitions represent the action of the farmer crossing the river with at most one of his purchases. So if the farmer first crosses the river with the beans, the state transition is from 0000 to 1001.

   However, the state 1001 is not safe because the fox will eat the goose. Therefore, our goal is to find a path from 0000 to 1111 where all states along the path are safe, or to prove such a path does not exist. This is where the model checking concepts apply.

   To find the path, we annotate the states as safe or not, and start a breath-first search from 0000. The safe path to 1111 then can be found as 0000, 1010, 0010, 1110, 0100, 1101, 0101, 1111.