COMP 3152 Homework 11

This homework centres around Peterson's famous mutual exclusion algorithm as running example. It is an improvement of the brilliant original algorithm of Dekker.

The algorithm deals with two concurrent processes A and B that want to alternate critical and noncritical sections. Each of these processes will stay only a finite amount of time in its critical section, although it is allowed to stay forever in its noncritical section. The purpose of the algorithm is to ensure that they are never simultaneously in the critical section, and to guarantee that both processes keep making progress.

The processes use three variables. The boolean variable $readyA$ can be written by process A and read by process B, whereas $readyB$ can be written by B and read by A. By setting $readyA$ to $true$, process A signals to process B that it wants to enter the critical section. The variable $turn$ is a shared variable: it can be written and read by both processes. Its use is the brilliant part of the algorithm. Initially $readyA$ and $readyB$ are both $false$ and $turn = A$.

Process A

```
repeat forever
  ℓ1  noncritical section
  ℓ2  $readyA := true$
  ℓ3  $turn := B$
  ℓ4  await ($readyB = false \vee turn = A$)
  ℓ5  critical section
  ℓ6  $readyA := false$
```

Process B

```
repeat forever
  m1  noncritical section
  m2  $readyB := true$
  m3  $turn := A$
  m4  await ($readyA = false \vee turn = B$)
  m5  critical section
  m6  $readyB := false$
```

1. What would be wrong with this protocol if we omitted the variable $turn$?

2. Express **Process A** as a CCS, CSP or ACP expression, featuring six atomic actions $l_1, \ldots, l_6$ and a recursive equation with variable X.

3. Represent **Process A** as a process graph, by using $l_1, \ldots, l_6$ as transition labels. Also attach names to the states by calling each state after the transition that is enabled there. Thus transition ℓ_3 goes from state ℓ_3 to state ℓ_4.

4. Correct the answers to questions 2 and 3 by replacing the action l_4 by the two actions "*B not ready*" and "$turn = A$". "*B not ready*" denotes the action of reading the value of $readyB$ and finding that it is $false$. Likewise, "$turn = A$" denotes the action of reading the value of $turn$ and finding that it is A. [You may skip the original answers to 2 and 3.]

   It would be possible to model instruction ℓ_4 assuming *busy waiting*, by drawing self-loops in state ℓ_4 labelled "*B ready*" and "$turn = B$". However, I want to abstract from these unsuccessful read actions from the onset by not including them in our formal specification. Thus, the intuition of the **await** statement is that the process A patiently sits in state ℓ_4 until one of the transitions "*B not ready*" or "$turn = A$" is enabled.

5. Represent Peterson's algorithm as Petri net with 18 places: 6 places for each of the 6 internal states of the processes A and B and 2 places for each of the 3 variables. You need 16 transitions: 2 copies of ℓ_3 and m_3, and 1 copy of each of the other transitions.

6. Represent the behaviour of variable *readyA* as a process graph. Its transition labels are the write actions l_2 and l_6 that can be performed by processes **A**, and the read action "*A not ready*" that can be performed by process B.

7. Give a process algebraic expression of this behaviour.

8. Represent the behaviour of variable *turn* as a process graph. Its transition labels are the write actions l_3 and m_3 that can be performed by processes **A** and **B**, and the read actions $turn = A$ and $turn = B$. Also give a process algebraic expression of this behaviour.

9. Now give a process algebraic expression of the entire protocol, involving the parallel composition of 5 processes. All actions except the critical and noncritical sections (ℓ_1, ℓ_5, m_1 and m_5) are internal (only needed to make the protocol work) and should be abstracted away. (You may choose whether to use CCS, CSP or ACP, and feel free to rename for instance ℓ_2 into $\overline{\ell_2}$ if this suits you.)

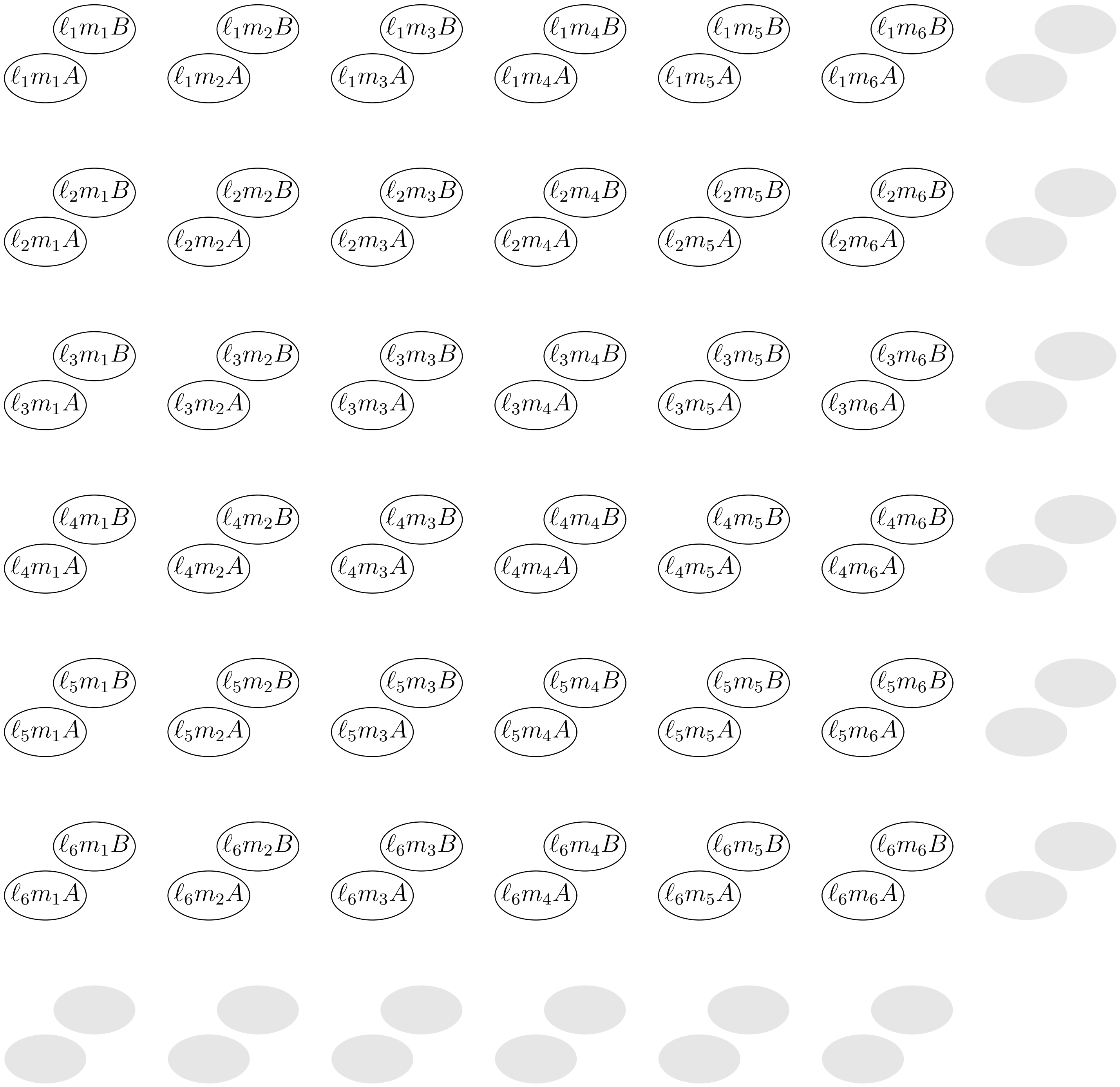
10. On the next page you see the potential states of a process graph representation of the entire algorithm. A state of the algorithm is completely determined by a state of process A, a state of process B and a state of *turn*. For the states of *readyA* and *readyB* are completely determined by the states of A and B. This observation yields $6 \times 6 \times 2 = 72$ potential states.

    Complete the given drawing into a process graph by supplying the transitions. Don't bother labelling them. Also don't draw loops backwards to the left or top rows; instead use the gray shadows, which represent copies of the transitions at the opposite end of the diagram. How many states are reachable from the initial state?

11. In this question we have an atomic proposition **IntentA**, saying that Process A intends to enter the critical section, but has not done so yet. It holds whenever Process A is in states ℓ_2 ℓ_3, ℓ_4 or ℓ_5. We also have an atomic proposition **CritA**, sating that Process A is about to enter its critical section; it holds in state ℓ_5 only. Likewise we have atomic propositions **IntentB** and **CritB**.

    Formulate an LTL property in terms of these four atomic propositions expressing the correctness of Peterson's algorithm. (See the beginning of this homework for the informal description of the two properties the protocol should have in order to be correct.)

Process Graph of Peterson's Mutual Exclusion Algorithm


ℓ_1m_1B
ℓ_1m_1A
ℓ_1m_2B
ℓ_1m_2A
ℓ_1m_3B
ℓ_1m_3A
ℓ_1m_4B
ℓ_1m_4A
ℓ_1m_5B
ℓ_1m_5A
ℓ_1m_6B
ℓ_1m_6A
ℓ_2m_1B
ℓ_2m_1A
ℓ_2m_2B
ℓ_2m_2A
ℓ_2m_3B
ℓ_2m_3A
ℓ_2m_4B
ℓ_2m_4A
ℓ_2m_5B
ℓ_2m_5A
ℓ_2m_6B
ℓ_2m_6A
ℓ_3m_1B
ℓ_3m_1A
ℓ_3m_2B
ℓ_3m_2A
ℓ_3m_3B
ℓ_3m_3A
ℓ_3m_4B
ℓ_3m_4A
ℓ_3m_5B
ℓ_3m_5A
ℓ_3m_6B
ℓ_3m_6A
ℓ_4m_1B
ℓ_4m_1A
ℓ_4m_2B
ℓ_4m_2A
ℓ_4m_3B
ℓ_4m_3A
ℓ_4m_4B
ℓ_4m_4A
ℓ_4m_5B
ℓ_4m_5A
ℓ_4m_6B
ℓ_4m_6A
ℓ_5m_1B
ℓ_5m_1A
ℓ_5m_2B
ℓ_5m_2A
ℓ_5m_3B
ℓ_5m_3A
ℓ_5m_4B
ℓ_5m_4A
ℓ_5m_5B
ℓ_5m_5A
ℓ_5m_6B
ℓ_5m_6A
ℓ_6m_1B
ℓ_6m_1A
ℓ_6m_2B
ℓ_6m_2A
ℓ_6m_3B
ℓ_6m_3A
ℓ_6m_4B
ℓ_6m_4A
ℓ_6m_5B
ℓ_6m_5A
ℓ_6m_6B
ℓ_6m_6A