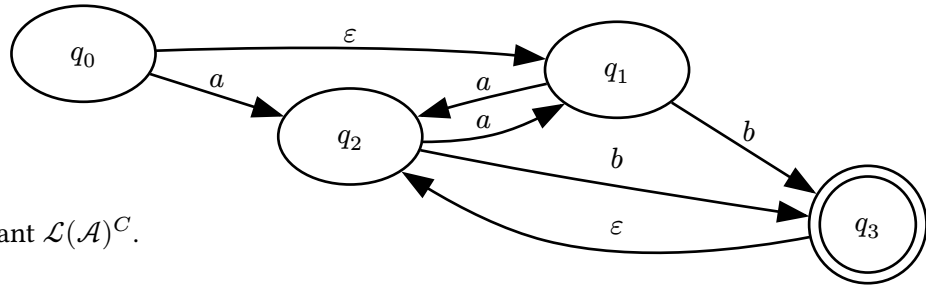


## Sujet IMT-2

### I - Automates

On considère l'automate  $\mathcal{A}$  ci-contre.

1. Déterminer et compléter l'automate.
2. Donner une expression régulière dénotant  $\mathcal{L}(\mathcal{A})^C$ .



### II - Verrous à n fils d'exécution

#### II.a - Un premier verrou

On considère le pseudo-code suivant (avec `get_thread_id()` -> l'identifiant du fil courant).

```

1: function CREATELOCK()
2:   return {turn = 0, busy = false}
3: end
  
```

```

1: function UNLOCK(l)
2:   l.busy ← false
3: end
  
```

```

1: function LOCK(m, t)
2:   me ← get_thread_id()
3:   repeat
4:     l.turn ← me
5:   until not l.busy
6:   l.busy ← true
7: end
  
```

1. Ce protocole garantit-il l'exclusion mutuelle ?

On propose une autre version de la fonction Lock :

```

1: function LOCK(m, t)
2:   me ← get_thread_id ()
3:   repeat
4:     repeat
5:       l.turn ← me
6:     until not l.busy
7:     l.busy ← true
8:   until l.turn = me
9: end
  
```

2. Ce nouveau protocole garantit-il l'absence d'interblocage ?
3. Que dire de l'exclusion mutuelle ?

#### II.b - Boulangerie de Lamport

4. Expliquer précisément le fonctionnement de l'algorithme de la boulangerie de Lamport.
5. Compléter (au tableau) le pseudo-code ci-dessous.

```

1: function CREATELOCK(want ←
   false,...,false, ticket ←
   0,...,0, return {want,ticket})
2: end
  
```

```

1: function UNLOCK(m.want[i] ← ...)
2: end
  
```

```

1: function LOCK(m, i)
2:   m.want[i] ← ...
3:   m.ticket[i] ← ...
4:   while ... do
5:     end
6: end
  
```

6. Montrer que l'algorithme de la boulangerie de Lamport satisfait l'absence d'interblocage.
7. ... respecte l'ordre "FIFO". Qu'en déduire sur l'absence de famine ?
8. ... satisfait l'exclusion mutuelle.