

# TP - automates probabilistes

December 9, 2015

Dans ce TP, on revisitera la programmation événementielle, en formalisant nos algorithmes à l'aide d'automates.

## 1 Rappel concernant l'utilisation de la boucle événementielle

Voici, pour vous aider, un exemple de mini-programme qui utilise la boucle événementielle, et qui observe les boutons et les capteurs de proximité. Vous pouvez aussi regarder les sujets de TP à partir du TP 7.

```
import pythymio as pt
import time

custom = pt.customEvents('colors', 'sound')

with pt.thymio(['buttons', 'prox'], custom) as Thym:

    def progression(id,evt,data):

        if evt == "fwd.buttons":
            if data[0] == 1:
                Thym.set('motor.left.target', 500)
                Thym.set('motor.right.target', 500)
                Thym.send_event('become.blue')
            else:
                Thym.set('motor.left.target', 0)
                Thym.set('motor.right.target', 0)
                Thym.send_event('become.green')

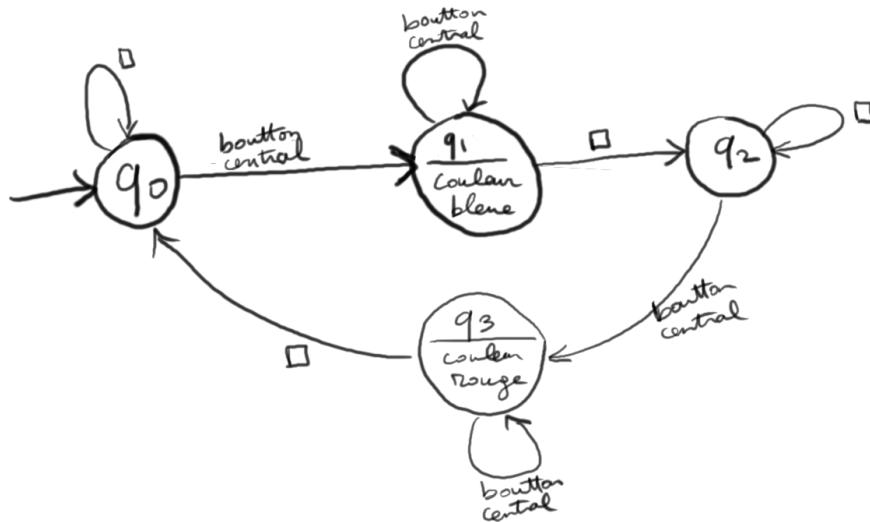
        print('le tableau de boutons est ',list(data))

        if evt == "fwd.prox":
            print('le tableau des telemetres est ',list(data))
```

1. Question: Dites *exactement* ce que fait ce programme

## 2 Implémentation d'un automate

Soit l'automate suivant:



*Rappels concernant les notations utilisées:*

- le symbole carré désigne un événement vide, c'est à dire un cas où il ne se passe rien.
- l'événement 'bouton central' désigne le fait que l'utilisateur appuie sur le bouton au centre du Thymio. Cet événement sera déclenché 20 fois par secondes jusqu'à ce que l'utilisateur relache le bouton.
- chaque état est représenté par un cercle. Dans ce cercle se trouvent le nom de l'état (en haut) et l'action associée à cet état (en bas).

Questions:

1. Dites **exactement** ce que fait cet automate
2. Implémentez le dans une boucle progression

## 3 The only limit is your imagination...

Parmi les propositions de sujets ci-dessous, vous en choisirez un. Ensuite, vous **écrirez sur papier un automate qui pourrait résoudre ce problème**, et enfin vous l'implémenterez dans la boucle progression.

Ces problèmes sont laissés volontairement vagues, pour que vous imaginiez vous-même comment les définir précisément, comment les modéliser avec un automate, et comment les coder.

### 1. Evitement d'obstacle

- (a) le robot devra suivre les murs s'il y a un mur sur le coté
- (b) s'il fait face à un obstacle, il devra faire demi-tour, par la droite ou par la gauche (ce sera choisi aléatoirement).  
*Remarque:* parce qu'il y a de l'aléatoire, il y aura des probabilités inscrites dans l'automate

### 2. Suivi (et attaque) d'autres robots

- (a) Le robot repère l'objet le plus proche de lui (si ses capteurs de proximité détectent quelque chose), et fait ceci:
  - i. il s'oriente pour lui faire face
  - ii. il s'en approche s'il en est éloigné
  - iii. il s'en éloigne s'il est trop procheEn bref, il doit s'en tenir à distance constante: avancer si l'objet avance, reculer s'il recule.
- (b) De temps en temps, avec une faible probabilité, le robot passe en mode "attaque". Il fonce dans l'objet qu'il suit et tente de le pousser. Ca doit durer au plus 2 secondes.

- 3. **Super Simon.** Le jeu Simon propose une séquence de notes et de couleurs que l'utilisateur doit mémoriser et reproduire. Proposez une extension de ce jeu, modélisez la et implémentez la.