

# Projet Système de feux tricolores

Adrien Garandel      Bao Ran      Franck Boncler  
Jérémy Bardon

November 3, 2015

## Contents

<b>1</b>	<b>Partie 1 - Deux feux synchronisés</b>	<b>2</b>
1.1	Questions 1 et 2 . . . . .	2
1.2	Question 3 . . . . .	2
1.3	Question 4 . . . . .	3
1.4	Question 5 et 6 . . . . .	4
<b>2</b>	<b>Partie 2 - Deux feux temporisés</b>	<b>5</b>
2.1	Question 7 . . . . .	5
2.2	Question 8 . . . . .	6
<b>3</b>	<b>Partie 3 - Carrefour en T</b>	<b>9</b>
3.1	Question 9 . . . . .	9
3.2	Question 10 . . . . .	11
3.3	Question 11 . . . . .	12
3.3.1	Validations question 9 . . . . .	12
3.3.2	Validations question 10 . . . . .	13

## 1 Partie 1 - Deux feux synchronisés

Dans cette partie, on s'intéresse à la synchronisation de deux feux tricolores. On utilise pour cela des réseaux de Pétri et le logiciel Roméo.

### 1.1 Questions 1 et 2

Feux non synchronisés et non temporisés

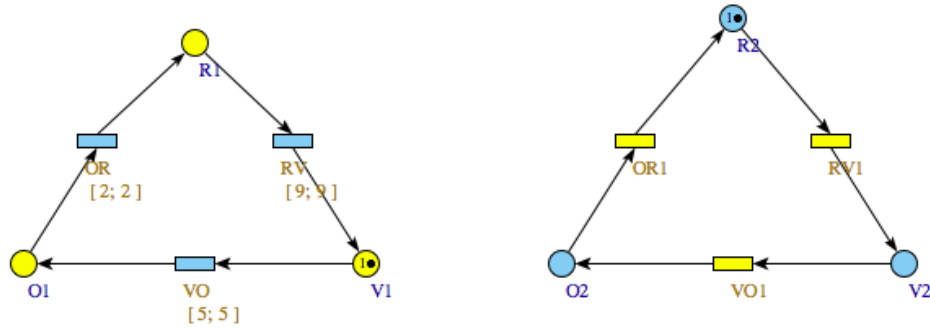


Figure 1: Réseau de Pétri, feux non synchronisés

Chaque feu est identique et dispose de trois places suivant les trois couleurs qu'un feu de circulation tricolore peut prendre: Rouge, Orange et Vert.

### 1.2 Question 3

Feux synchronisés et non temporisés



Pour respecter ce cahier des charges, nous avons ajouté deux places entre les deux feux. Alors, lorsqu'un feu devient Vert, il empêche l'autre de devenir Vert puis le libère une fois qu'il est passé à l'Orange puis au Rouge.

Un graphe de marquage permet de montrer les différentes exécutions possibles de notre système de réseaux de Pétri.

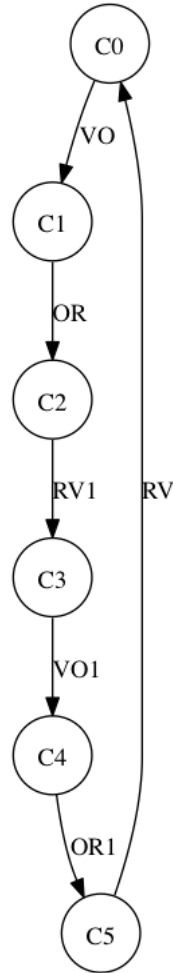


Figure 3: Graphe de marquage des feux synchronisés

On constate que l'exécution de notre système est linéaire. Cela signifie qu'il existe une seule exécution, ainsi prouver que celle-ci est correcte revient à prouver que le système entier est correct.

#### 1.4 Question 5 et 6

Vérification des propriétés de sûreté de vivacité du système de feux synchronisés.

**Pas 2 feux rouges en même temps (sûreté)**

$AG[0, \text{inf}] (M(R1) + M(R2) \geq 1)$

Un feu passe au moins une fois au vert (non bloqué)

$AG[0, \text{inf}](M(V2) = 1)$

Les feux ne se bloquent pas entre eux

$AG[0, \text{inf}](M(R1) + M(P7)) \# P7 = \text{place intermédiaire}$

## 2 Partie 2 - Deux feux temporisés

La partie précédente expose un système de feux synchronisés mais non temporisés. En effet, dans la réalité les feux restent temps donné dans leurs états. Nous proposons alors un système de feux temporisés sous la forme d'automates à états finis modélisés avec le logiciel Uppaal.

### 2.1 Question 7

Feux temporisés sans synchronisation

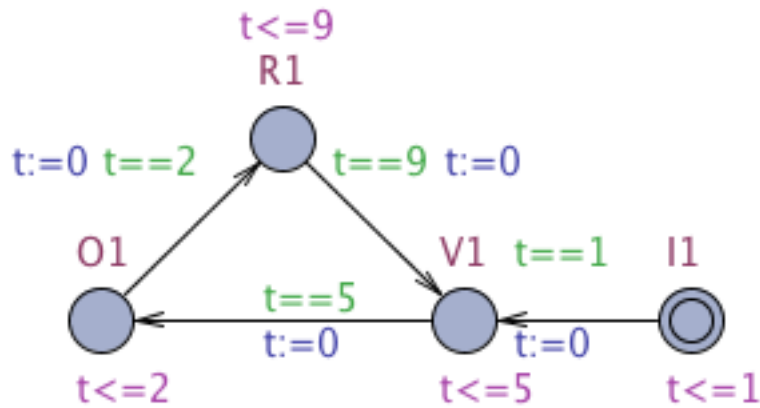


Figure 4: Automate du feu commençant en Vert

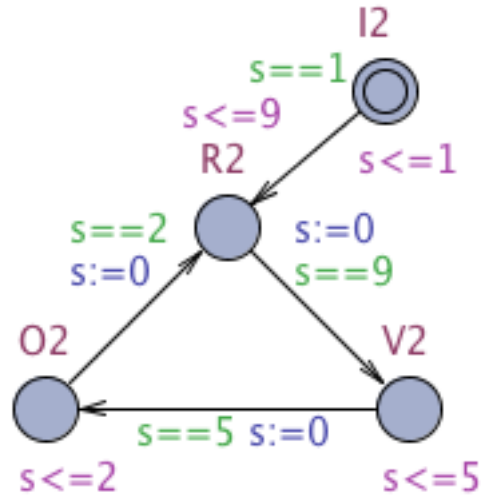


Figure 5: Automate du feu commençant en Rouge

Pour les deux feux, il est nécessaire d'ajouter une place initiale pour indiquer l'état initial de chacun des feux. Ce premier état attend 1 seconde car sinon les automates on observerait une désynchronisation des feux.

Validation des propriétés de sureté de vivacité du système de feux temporisées.

**A n'importe quel moment il n'y a pas deux feux rouges en même temps**

`A[] (feu1.R1 or feu2.R2 or feu1.I1 or feu2.I2)`

**Pas de blocage**

`E<> deadlock`

## 2.2 Question 8

Afin de contrôler au mieux le système, cette solution propose l'utilisation d'un contrôleur. Cet élément est chargé de faire évoluer les états des feux.

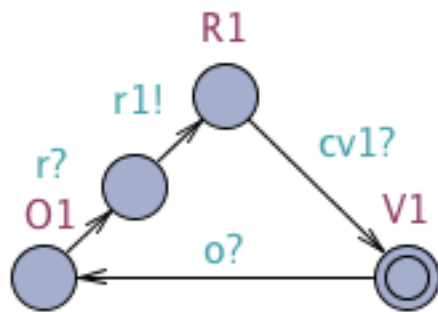


Figure 6: Automate du feu commençant en Vert

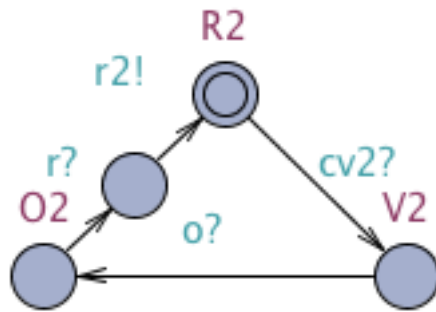


Figure 7: Automate du feu commençant en Rouge

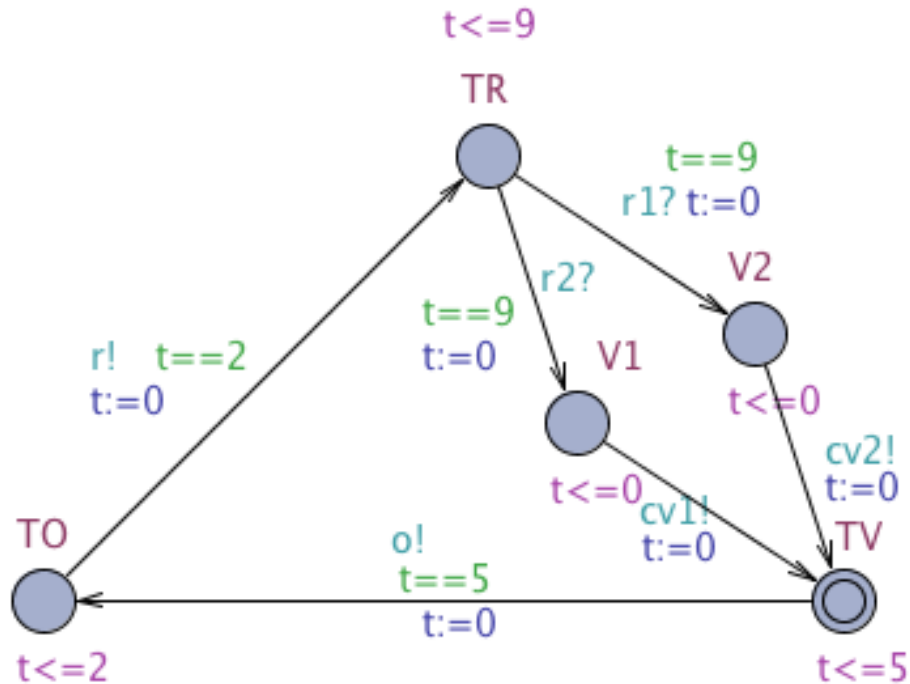


Figure 8: Contrôleur des feux

Les feux ne sont pas temporisés mais il y a toujours un feu Rouge et un feu Vert au démarrage du système. En effet, c'est le rôle du contrôleur qui temporise et commande les feux à travers une synchronisation par signaux.

Le contrôleur allume alors alternativement le premier et le second feu tout en prenant en compte le temps pendant lequel un feu doit rester dans un état donné.

Vérification des propriétés de sûreté de vivacité du système de feux synchronisés.

**A n'importe quel moment il n'y a pas deux feux rouges en même temps**

A[] (feu1.R1 or feu2.R2)

??????????

E<>(feu1.R1 and feu2.R2)

**Pas de deadlock**

E<> deadlock



### 3 Partie 3 - Carrefour en T

Après avoir étudié des systèmes représentant la synchronisation de deux de circulation sur une route, nous allons nous intéresser à un carrefour en T.

Comme dans le premier cas, il y a une route principale avec deux feux tricolores. On ajoute ici une route mineure sur le côté. Dans un cas normal, le feu de la route mineure est Rouge mais lorsque des voitures sont captées il passe au Vert.

#### 3.1 Question 9

Dans cette première version de notre système de feux, il n'y a pas de contraintes de temps.

- Les 2 feux de la grande route sont considérés comme un seul
- Processus qui régule l'arrivée des voitures dans la petite rue

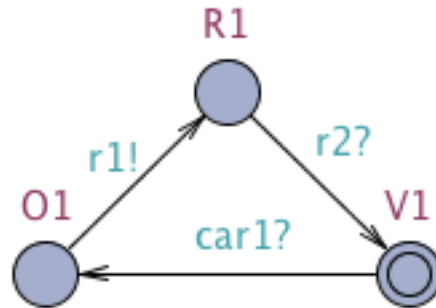


Figure 9: Automate du feu de la route majeure

Le feu de la route principale passe du Vert au Orange puis au Rouge lorsqu'un véhicule arrive sur la route mineure. Il attend alors qu'il n'y ai plus de voiture sur la route mineure pour repasser au Vert.

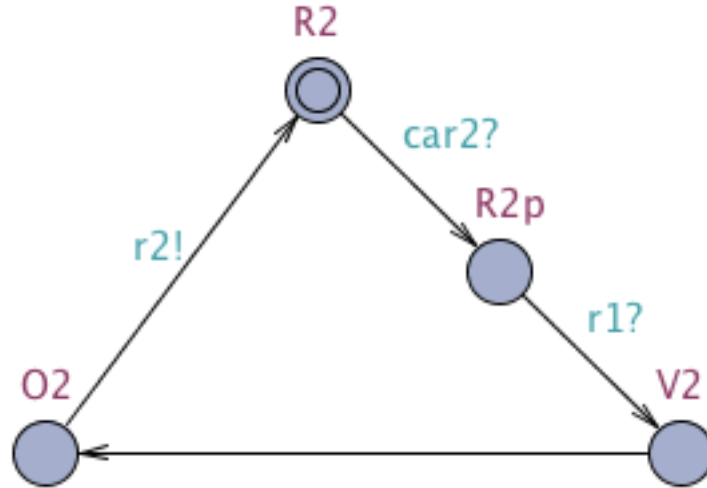


Figure 10: Automate du feu de la route mineure

Par défaut Rouge, le feu de la route mineure passe au Vert si une voiture est capté et une fois que les feux de la route principale sont Rouges.

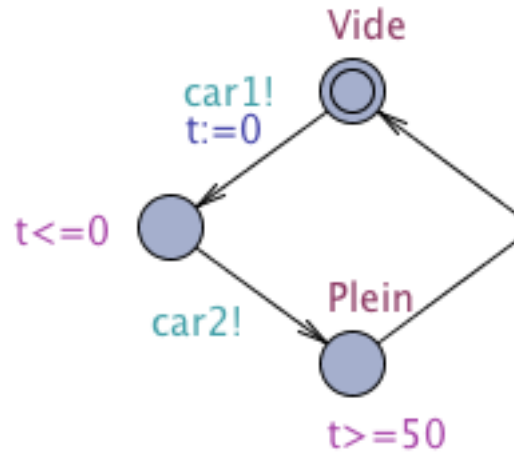


Figure 11: Automate de l'arrivée des véhicules sur la route mineure (capteur)

Cet automate régule l'arrivée de voiture sur la route mineure. Il détecte alors une voiture toutes les 50 tocs d'horloges.

### 3.2 Question 10

A notre système de feux sur un carrefour en T, nous ajoutons ici des contraintes de temps notamment sur le temps pendant lequel chaque feu doit rester Vert.

- Petite rue verte 30 secondes
- Dans un cycle, la grande route est Verte au moins 30 secondes
- Delai de 1 seconde entre chaque changement de couleur
- Chaque feu reste orange pendant 5 secondes

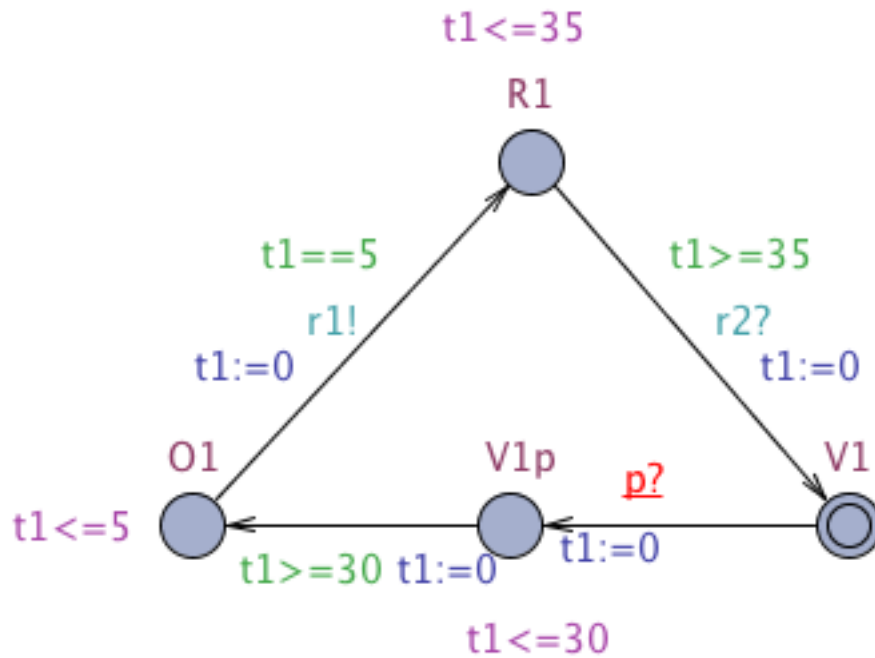


Figure 12: Automate du feu de la route majeure

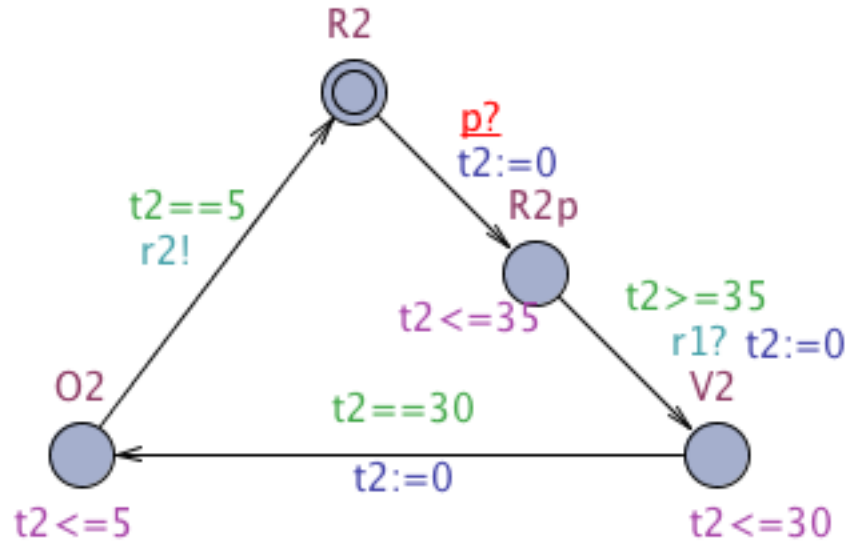


Figure 13: Automate du feu de la route mineure

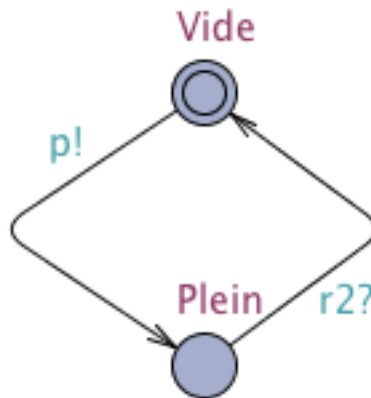


Figure 14: Automate de l'arrivée des véhicules sur la route mineure (capteur)

### 3.3 Question 11

#### 3.3.1 Validations question 9

A n'importe quel moment il n'y a pas deux feux rouges en même temps

A[] (Major.R1 or Minor.R2 or Minor.R2p)

**Pas de deadlock**

E<> deadlock

### 3.3.2 Validations question 10

**A n'importe quel moment il n'y a pas deux feux rouges en même temps**

A[] (Major.R1 or Minor.R2 or Minor.R2p)

**Pas de deadlock**

E<> deadlock