

Projet 1 : étape 1 spécification en FSP ( 10%) : *Synchronisation de feux de circulation*

À remettre avant 23h55, le mardi **12 mars 2019**. Travail en équipe de **2 à 4**.

Il s'agit de concevoir et développer un système de contrôle de lumière sur une intersection en T et sur une intersection ordinaire. Ce système doit permettre aux voitures et aux piétons de traverser. Évidemment, il faut éviter les collisions voiture-voiture et voiture-piéton. La figure 1 montre la configuration de ces intersections.

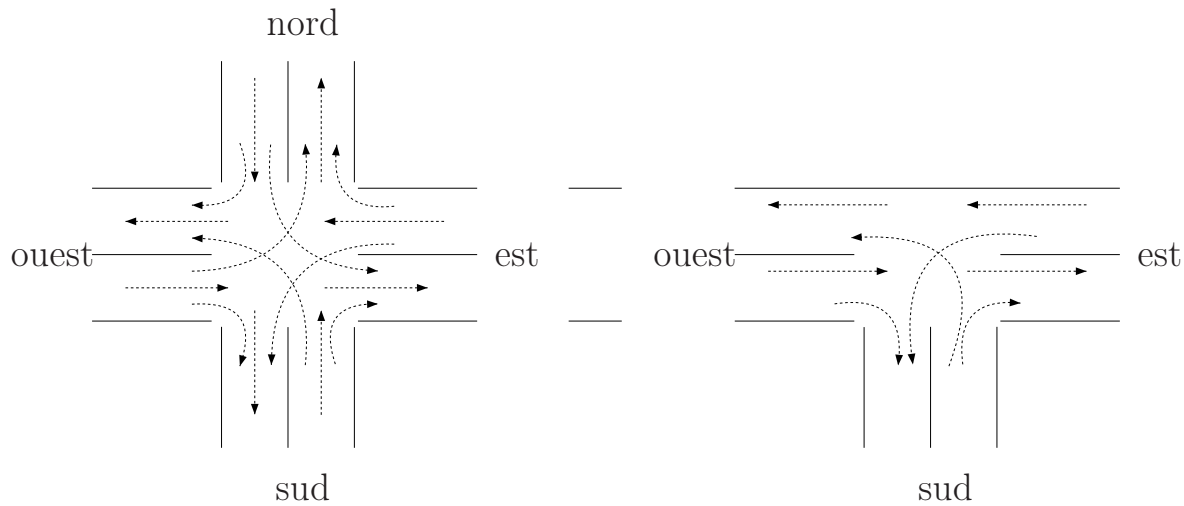


FIGURE 1 – Configuration de l'intersection

Les lignes pleines indiquent les voies et les lignes pointillées indiquent le trajet possible des voitures.

La première étape consiste à spécifier en FSP le système et la deuxième étape consiste à le développer en Java.

Ce cours va faire partie des cours éligibles au prix Pierre Ardouin. L'équipe sélectionnée dans ce cours gagne le prix de ce cours et fera partie d'une autre sélection avec les équipes gagnantes des autres cours. Un comité choisira alors le meilleur travail et c'est ce dernier qui gagnera le prix Pierre Ardouin.

Pour être sélectionnée comme équipe gagnante, l'équipe doit en premier lieu répondre à toutes les exigences demandées dans les 2 énoncés successifs mais aussi d'aller un peu plus loin dans la réalisation que ça soit en spécifiant et développant d'autres propriétés que celles demandées ou bien en réalisant un développement qui intègre une interface personne-machine, par exemple ou d'autres fonctionnalités.

Les équipes peuvent prendre 100% des notes sans toutefois être choisies. Il faut que l'équipe se démarque par une réalisation pour pouvoir être choisie.

## Travail demandé

Les questions Q1 jusqu'à Q7 concernent l'intersection en T.

**Q1 (5 points).** Donnez la spécification en FSP d'un processus **VOITURE** qui représente une voiture à l'intersection, peut continuer, tourner à droite ou tourner à gauche. Ce processus doit être cyclique permettant de modéliser plusieurs voitures, faire boucler le processus ne signifie pas que c'est la même voiture qui . Le processus **VOITURE** peut faire les actions suivantes, **continue**, **tourneDroite** ou **tourneGauche**. Ces actions signifient ce que leurs noms suggèrent. Après chaque action ce processus boucle sur lui même pour représenter le fait qu'une autre voiture pourrait arriver.

**Q2 (3 points).** Donnez la spécification en FSP d'un processus **VOITURES** qui représente la mise en parallèle de 3 processus **VOITURE** correspondant respectivement aux voitures venant de l'est, de l'ouest et du sud. Vous pouvez préfixer le processus **VOITURE** par **est**, **ouest** et **sud**.

**Q3 (2 points).** Spécifiez un processus **PIETONS** qui représente le comportement des piétons. Un piéton peut effectuer une seule action **traverse** puis boucle sur lui-même.

**Q4 (5 points).** Par souci de simplicité, nous supposons que les lumières peuvent être seulement rouge ou verte. Nous supposons que les lumières sont initialement toutes rouges. Chacune des lumières passe du rouge au vert et inversement.

Spécifiez un processus **LUMIERE** qui fait passer une lumière du vert au rouge.

Spécifiez un processus **LUMIERE** qui représente la mise en parallèle de 3 processus **LUMIERE** correspondant respectivement aux lumières face à l'est, à l'ouest et au sud. Vous pouvez préfixer le processus **LUMIERE** par **est**, **ouest** et **sud**.

**Q5 (30 points).** Spécifiez des processus contrôlant le passage des voitures, des piétons et le changement des lumières. Les règles suivantes permettent de concevoir ce processus. Vous pouvez l'appeler **CONTROLE\_INTERSECTION**

1. Les lumières sont toutes rouges au départ.
2. Les piétons ne peuvent traverser que quand toutes les lumières sont rouges.
3. Les voitures provenant du sud ne peuvent pas continuer mais seulement tourner à gauche ou tourner à droite.
4. Les voitures provenant de l'est ne peuvent que continuer ou tourner à gauche.
5. Les voitures provenant de l'ouest ne peuvent que continuer ou tourner à droite.
6. Quand la lumière face à une voiture est verte et que toutes les autres sont rouges, celle-ci peut continuer, tourner à droite ou tourner à gauche si la configuration le permet. item  
Quand la lumière face à une voiture est verte et que celle opposée l'est aussi celle-ci peut continuer ou tourner à droite si la configuration le permet.
7. On ne peut pas avoir deux lumières adjacentes, est et sud par exemple, vertes en même temps.
8. On peut avoir deux lumières opposées, est et ouest, par exemple vertes en même temps ou bien l'une verte et l'autre rouge.

Utilisez les processus déjà spécifiés comme ceci :

```
||INTERSECTION = (LUMIERES || PIETONS ||  
    VOITURES || CONTROLE_INTERSECTION) .
```

Vous n'êtes pas obligés de suivre l'indication suivante, si vous proposez une solution plus élégante, je saurai apprécier. Indication : inspirez-vous de l'exemple du pont et utilisez des indices à un processus local où la valeur d'un paramètre correspondrait à une lumière rouge ou verte. Regardez comment dans cet exemple on a défini une variable qui vaut True ou False. Notez que nous pouvons utiliser la conditionnelle **when** avec la même condition autant de fois qu'on veut. Par exemple, **when (v==0) a --> P1 when (v==0) b --> P2**

Utilisez le RUN pour vous assurer que tout fonctionne comme vous le voulez.

**Q5 (10 points).** Pour vous assurer que le processus **||INTERSECTION** ne met pas en danger les piétons, spécifiez une propriété que vous composerez avec **||INTERSECTION** pour vous assurer que l'action **traverse** ne sera jamais effectuée quand il y a au moins une lumière verte allumée. Spécifiez un processus **TEST1** pour faire la vérification au moyen de **Check** de LTSA.

**Q6 (15 points).** Spécifiez 3 propriétés une par côté de l'intersection servant à garantir l'absence de collision.

Utilisez un processus **TEST2** pour faire cette vérification.

**Q7 (5 points).** Spécifiez des ensembles d'actions permettant de vérifier que les piétons peuvent toujours traverser l'intersection et que les voitures peuvent faire les actions prévues, dans la composition parallèle de la question précédente, **TEST2**, représentant le système en entier. À vous de décider s'il faut spécifier des singletons ou non. Justifiez par une phrase de commentaire votre choix. Il ne doit pas y avoir violation de propriétés de progrès.

**Q8 (15 points).** Faites toutes les étapes précédentes Q1 à Q7 mais pour l'autre intersection. Inspirez-vous de vos solutions, évidemment. Vos processus correspondant à ceux des questions Q5 et Q6, s'appelleraient, pas obligatoirement, **TEST3** et **TEST5**.

Notez que pour tester les propriétés de progrès de ces derniers vous devez commenter les propriétés de progrès de l'intersection en T, et vice-versa.

**Q9 (10 points).**

Spécifiez un processus **TEST5** qui met en parallèle **TEST2** et **TEST4**. Vous pouvez préfixer les processus de l'intersection en T par [1] et ceux de l'autre par [2] pour maximiser la réutilisation. Vérifier les propriétés définies dans les questions précédentes ainsi que les propriétés de progrès. Utilisez le renommage de certaines actions pour synchroniser les lumières. Ainsi les lumières est et ouest des deux intersections doivent être, rouges ou vertes, en même temps. La même chose pour les lumières côté sud de l'intersection et sud en T et nord de l'autre. Renommez aussi les actions montrant la continuité des intersections ainsi une voiture venant du nord de la deuxième intersection et qui tournerait à gauche serait une voiture qui continue en provenance de l'ouest pour l'intersection en T.

Pour tester le progrès nous avons besoin de toutes les actions servant à définir les propriétés précédentes sauf celles renommées différemment. Il ne devrait y avoir ni blocage ni violation de propriétés ni de violation de progrès.

**Consignes à suivre obligatoirement**

- Vous devez fournir un fichier .lts dans lequel vous donnez les spécifications, **les commentaires aidant la compréhension.**

**Consignes à suivre obligatoirement pour la remise :** Vous devez remettre par intranet les documents suivants dans un .zip. votre fichier .lts .

**Le fichier .lts doit contenir les numéro de matricule et les noms de tous les membres de l'équipe. (perte de 5% des points sinon),.**

Ne m'envoyez pas vos fichiers par courriel. Sauf force majeure.

**Autres contraintes :** Vous devez respecter les contraintes suivantes.

- Ce travail se fait obligatoirement en équipe.
- Vous devez prendre toutes les mesures nécessaires pour éviter le plagiat. Si vous proposez des solutions trop semblables à celles d'une autre équipe ou à des solutions trouvées sur internet, des sanctions pourraient être prises conformément au règlement départemental et de l'université Laval.