M2/ALMA Formal Software Engineering

Projet du cours de Claude Jard et Benoît Delahaye, Octobre 2015

Objectif : apprendre à modéliser en utilisant des spécifications formelles, trouver des erreurs, prouver des propriétés, produire une implémentation, expérimenter. L'accent est mis sur la conception de systèmes parallèles asynchrones et temporisés. Les validations formelles sont conduites avec les réseaux de Petri à l'aide de l'outil Romeo pour l'aspect non temporisé, et avec les automates temporisés pour prendre en compte le temps à l'aide de l'outil Uppaal. La partie implémentation peut être menée en parallèle avec les activités d'analyse. Le projet sera conduit collectivement par des groupes d'environ 4 personnes et sera évalué sur la base d'un rapport d'une dizaine de pages maximum.

1 Un système à deux feux, sans temps

On considère une portion de route à protéger par deux feux de circulation. Considérant qu'un automobiliste arrivant sur le feu s'arrête si celui-ci est rouge, l'objectif est d'empêcher deux voitures d'être simultanément sur la portion protégée (propritété de sûreté). On souhaite aussi qu'aucun des feux ne reste rouge (ou vert) infiniment longtemps (propriété de vivacité).

Question 1 : modéliser à l'aide d'un réseau de Petri le comportement d'un feu de circulation.

Question 2 : considérer deux feux semblables fonctionnant en parallèle et sans synchronisation entre eux. On pourra initialiser le premier à « vert » et le second à « rouge ». Dessiner le graphe des marquages correspondant.

Question 3 : en l'absence de contraintes temporelles dans le modèle, on propose de synchroniser l'activité des feux en leur permettant de s'échanger des messages de façon asynchrone (le milieu de communication pourra être représenté par des places supplémentaires partagées par les feux). Proposer un tel modèle de réseau de Petri.

Question 4 : Construire le graphe de marquage avec Romeo. Dessiner le.

Question 5 : écrire en LTL les propriétés de sûreté et de vivacité attendues.

Question 6 : utiliser le « model-checker » de Romeo pour valider ces propriétés. La sûreté est-elle assurée ? La vivacité est-elle assurée ?

2 Un système à deux feux avec temps

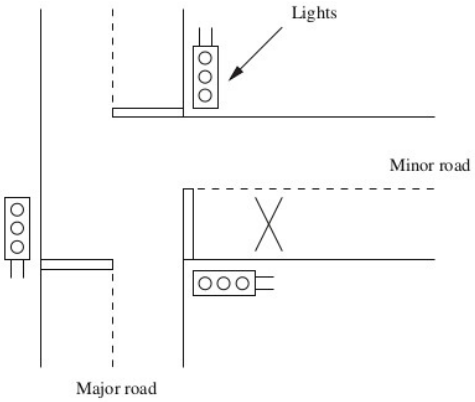
Question 7 : on reprend le modèle de la question 2, en munissant les feux d'horloges. Proposer des durées pendant lesquelles chacun des feux doit rester dans un état donné (vert, orange ou rouge). Proposer un modèle avec deux automates temporisés respectant les durées énoncées précédemment. Dans ce modèle, on ne souhaite pas avoir de synchronisation entre les feux. Essayer de valider les propriétés en utilisant le « model-checler » de l'outil UPPAAL.

Question 8 : on souhaite maintenant ajouter de la synchronisation entre les feux en utilisant un « controleur ». Ce contrôleur est muni d'une horloge et donne l'ordre aux feux de changer de couleur quand c'est nécessaire. Les feux ne sont alors plus temporisés. Proposer des modèles d'automates temporisés pour le contrôleur et les deux feux dans ce cas de figure. Essayer à nouveau de valider les propriétés en utilisant le « model checker » de l'outil UPPAAL. Que pouvez-vous en conclure ?

3 Modélisation d'un carrefour en T

On considère la spécification informelle suivante, tirée d'une application réelle.

A control system must ensure the safe and correct functioning of a set of traffic lights at a T-junction between a major and a minor road. The lights will be set on green on the major road and red on the minor road unless a vehicle is detected by a sensor in the road just before the lights on the minor road. In this case the lights will be switchable in the standard manner and allow traffic to leave the minor road. After a suitable interval the lights will revert to their default position to allow traffic to flow on the major road again. Once a vehicle is detected the sensor will be disabled until the minor-road lights are set to red again. A sketch of the T-junction is provided below.



Question 9 : First we ignore all timing issues involved and concentrate on the qualitative aspects of the behavior of the traffic lights. Model the above system as a network of (timed) automata. For convenience, you may assume that the two major-road lights are fully synchronized and can be modeled as a single light. Complement your system model by adding a process that regulates the arrival of cars in the minor road.

Question 10 : Adapt your model so as to incorporate the following timing constraints. Deal with each timing constraint separately so as reduce the complexity. Indicate for each timing constraint the necessary adaptations to your un-timed model :

— a minor-road light stays on green for 30 seconds,
— all interim lights stay on for 5 seconds,
— there is a 1 second delay between switching one light off and another on (e.g., switching from green to orange),
— the major-road lights must be on green for at least 30 seconds in each cycle,
— (more involved) but must respond to the sensor immediately after that.

Question 11 : modéliser les situations décrites en questions 9 et 10 en UPPAAL. Essayer de valider les propriétés de vivacité et sûreté avec le « model-checker » d'UPPAAL en les adaptant à votre modèle.

4 Implémentation

Question 12 : il s'agit maintenant de programmer une démonstration du système. Liberté vous est donnée pour le choix de l'environnement de programmation et la façon dont vous pilotez le système.

Question 13 : donner des éléments de conclusion sur l'ensemble du projet.