

MODÉLISATION ET VÉRIFICATION DE  
SYSTÈMES RÉACTIFS. MANIPULATIONS SOUS  
CAAL

2019–2020

Voici quelques exercices en support au cours de vérification des systèmes réactifs. Ces exercices pourront être utilisés pour aborder certaines questions en examen le 17 janvier 2020.

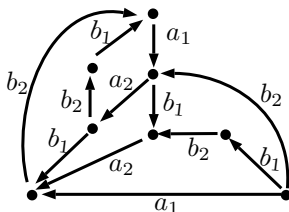
## 1 Ordonnanceur

## 1.1 Spécification

Considérons un ensemble de tâches  $T_1, \dots, T_n$ . On ignore le contenu de chaque tâche. En revanche, les tâches ne peuvent être lancées dans n'importe quel ordre. Supposons qu'une tâche  $T_i$  soit lancée par l'action  $a_i$  et qu'elle se termine par l'action  $b_i$ . Les différentes tâches peuvent se superposer : par exemple,  $T_2$  peut commencer avant que  $T_1$  ne se termine. Mais un ordonnanceur doit s'assurer qu'une tâche donnée se termine avant de la relancer. La spécification informelle de l'ordonnanceur est la suivante :

1. Les actions  $a_1, \dots, a_n$  doivent être activées cycliquement, en commençant par  $a_1$ .
2. Pour chaque  $i$ , les actions  $a_i$  et  $b_i$  doivent être exécutées alternativement.

L'ordonnanceur doit être aussi large que possible : il doit autoriser toutes les séquences d'actions de lancement et de terminaison compatibles avec les deux exigences précédentes. Par exemple, pour deux tâches seulement, les ordres possibles sont :



Les points sont des états, les flèches des transitions entre états et les étiquettes  $a_1, \dots, a_n$  sont les noms des tâches. Dans l'exemple, le système est initialement dans l'état du haut (la racine du graphe est en haut). Le système ne peut que passer dans l'état suivant en exécutant l'action  $a_1$ .

Question 1:

Donner un premier modèle CCS ou LOTOS décrivant cette spécification, pour

$n = 1, 2, 3, 4$ . En générant et en simulant les LTS obtenus, valider informellement cette spécification.  $\diamond$

**Question 2 :**

Pour  $n = 1, 2$ , vérifier formellement cette spécification en la comparant aux ordonnancements explicites construits manuellement. Utiliser les équivalences fortes et observationnelles.  $\diamond$

## 1.2 Modèle d'implantation de l'ordonnanceur

On propose de construire un modèle CCS ou LOTOS d'implantation de cet ordonnanceur, comme un anneau de  $n$  cellules  $C_1, \dots, C_n$ . Chaque cellule  $C_i$  possède quatre portes,  $a_i, b_i, c_i$  et  $\overline{c_{i-1}}$ . Le principe est le suivant :

- L'action  $a_i$  est utilisée pour enregistrer le lancement de la tâche contrôlée par la  $i^e$  cellule ;
- L'action  $b_i$  est utilisée pour enregistrer la fin de la tâche contrôlée par la  $i^e$  cellule ;
- L'action  $c_i$  est utilisée pour connecter la cellule  $C_i$  à la suivante dans l'anneau ;
- L'action  $c_{i-1}$  est utilisée pour connecter la cellule  $C_i$  à la précédente dans l'anneau.

La cellule qui suit  $C_n$  est  $C_1$  et la cellule précédant  $C_1$  est  $C_n$ . Ainsi,  $i - 1$  vaut en réalité  $n$  quand  $i = 1$ .

**Question 3 :**

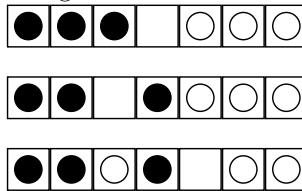
Donner une description CCS ou LOTOS de l'implantation de l'ordonnanceur vu comme un anneau de 4 cellules. Simuler les LTS obtenus.  $\diamond$

**Question 4 :**

Comparer par équivalence observationnelle, le modèle d'implantation avec le modèle de spécification de la question 1. Utiliser CAAL ou CADP pour vérifier ces équivalences pour  $n = 3, 4, 5$ .  $\diamond$

## 2 Solitaire

Un jeu de solitaire comporte sept cases et six pions, trois noirs et trois blancs. Les règles sont les suivantes : les pions noirs ne peuvent aller que sur la droite, soit sur une case vide, soit par dessus un pion blanc si la case de droite est vide. Les pions blancs ne peuvent aller que sur la gauche, soit sur une case vide, soit par dessus un pion noir si la case de gauche est vide.



...

Plusieurs scénarios conduisent à des blocages. Il s'agit de montrer que le jeu admet au moins une solution.

Pour cela, faire un modèle CCS comme sept cases adjacentes, vue chacune par un processus communiquant avec ses voisins de droite et de gauche. Expliquer les principes de votre modèle et les questions que l'on peut poser aux outils (CAAL) pour vérifier que le jeu peut être résolu.

## 3 Impressions 3D

Soit une entreprise de maquettage 3D.

### 3.1 Énoncé

L'entreprise fonctionne dans un premier temps avec un seul ingénieur.

- il reçoit des commandes correspondant à des cahiers des charges, à partir desquels il doit réaliser les plans détaillés informatisés en trois dimensions du futur bâtiment ;
- Si le plan répond à certains critères, l'ingénieur peut directement réaliser la maquette avec une imprimante 3D. Dans le cas contraire, il doit réaliser la maquette à la main. En pratique, il réutilise dans ce second cas plusieurs éléments déjà construits ou imprimables, mais nous considérerons qu'il réalise l'ensemble manuellement.
- Une fois la maquette réalisée, l'ingénieur va la présenter avant de commencer la commande suivante.

Cette entreprise fonctionne bien, mais l'ingénieur cherche à augmenter sa productivité. Il envisage de recruter un associé, ainsi que d'acheter une nouvelle imprimante, plus performante.

Cette imprimante plus performante permet de réaliser des maquettes plus complexes, mais certaines doivent encore être réalisées à la main. Les deux associés ont exactement les mêmes compétences et ne veulent pas se spécialiser, l'un dans les travaux faciles et imprimables, l'autre dans les maquettes délicates. Le temps et le coût d'une réalisation à la main, même pour des plans simples, est toujours très supérieur au coût d'une impression. Les ingénieurs privilégieront dans tous les cas les réalisations automatisées.

Les imprimantes 3D sont encore très complexes d'utilisation et doivent être paramétrées manuellement pour chaque réalisation. Elles n'ont donc pas de file d'attente d'impressions contrairement à une imprimante 2D habituelle.

Le but de l'exercice est de simuler le fonctionnement du bureau de maquettage, composé de deux ingénieurs et de deux imprimantes 3D.

### 3.2 Objectif

Montrer formellement que les processus entreprise ne comporte pas de blocage, pas de famine, et pas de livelock.