

Exercice 1 (*Ascenseur*)

Rendez l'exercice du TP UppAal sur l'ascenseur.

Exercice 2 (*Rush Hour*)

Il s'agit de modéliser le jeu "Rush Hour". Nous disposons d'un tableau 6×6 comme sur la figure 1 ci-dessous. Sur ce tableau sont placés des véhicules de différentes couleurs et de différentes tailles : les véhicules personnels occupent deux cases alors que les camions en prennent 3. Le tableau dispose également d'une sortie située sur la droite à mi-hauteur. Il y a en particulier une seule voiture individuelle rouge, qui est proche de la sortie et sur la ligne de la sortie. L'objectif est de déplacer un par un les véhicules, de case en case, sans les tourner ni les décaler (c'est-à-dire que les véhicules ne peuvent qu'avancer ou reculer), jusqu'à ce que le véhicule rouge quitte le tableau par la sortie.

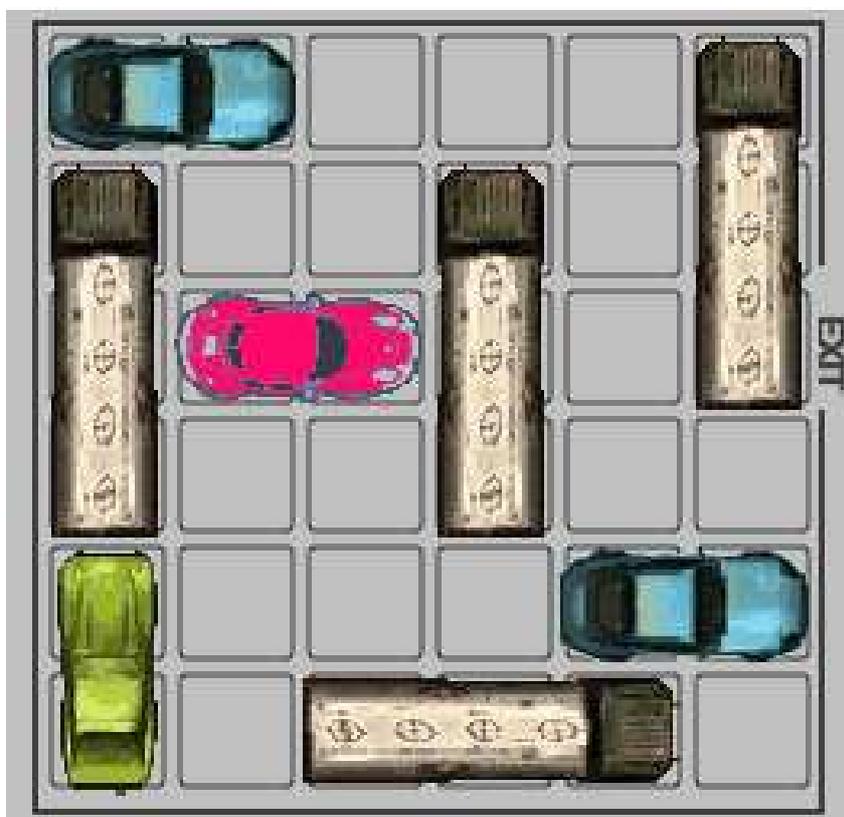


FIGURE 1 – Le tableau du jeu RUSH HOUR

Lorsqu'un camion effectue une suite de n déplacements consécutifs sur sa ligne (ou sur sa colonne) sans s'arrêter, cela nécessite $3 + 2 \times n$ unités de temps : il lui faut en effet 3 unités de temps pour démarrer puis, si les cases prévues sont encore vides, 2 unités de temps par case. Pour une voiture en revanche ce même déplacement nécessite seulement $1 + 1 \times n$ unités de temps, car une voiture démarre trois fois plus vite et avance deux fois plus vite qu'un camion.

Question 1. Donnez en uppaal une modélisation la plus simple possible du jeu décrit sur la figure 1 sous la forme d'un système d'automates temporisés. Combien y a-t-il de types de processus différents ?

Question 2. Comment peut-on utiliser `uppaal` afin de déterminer s’il existe une solution au problème ainsi formalisé ?

Question 3. Quel est le temps minimal pour faire sortir la voiture rouge ?

Exercice 3 (*Le jongleur*)

Modélisez un jongleur qui doit faire tourner des assiettes en équilibre au bout d’une tige pour les empêcher de tomber. En effet, les assiettes tombent et se cassent au bout d’un moment si elles ne sont pas à nouveau “agitées” par le jongleur. Le jongleur peut choisir de faire tourner une assiette pour un temps court, auquel cas cela empêche l’assiette de tomber pour un certain temps. Il peut aussi la faire tourner plus longtemps, ce qui la maintient en équilibre plus longtemps. Naturellement, le jongleur ne peut agiter qu’une seule assiette à la fois. Lorsque le jongleur commence à agiter une assiette, il le fait pendant au moins une unité de temps.

Malheureusement, il y a dans la pièce un moustique qui peut faire perdre l’équilibre aux assiettes. Lorsque le moustique touche une assiette, il réduit la stabilité de l’assiette, et son temps restant de stabilité décroît d’une unité de temps. Lorsque le moustique touche une assiette, il est effrayé, et il ne reviendra plus toucher une assiette avant D unités de temps.

Notre objectif est de trouver une stratégie pour le jongleur pour maintenir les assiettes en équilibre.

Vous pourrez assigner aux assiettes des durées de stabilité différentes (il peut y avoir des assiettes plus ou moins grandes). Les valeurs suivantes seront des paramètres définis de manière globale de sorte à pouvoir facilement changer leur valeur :

- `StabShort` : durée d’agitation “courte” du jongleur
- `StabTimeShort` : durée de stabilité de l’assiette pour une agitation courte
- `StabLong` : durée d’agitation “longue” du jongleur
- `StabTimeLong` : durée de stabilité de l’assiette pour une agitation longue
- `D` : durée de la mémoire du moustique

Question 1. Proposez une modélisation de ce système dans UpAal TiGa, avec deux assiettes. Quelles sont les transitions contrôlables ?

Question 2. En testant différentes valeurs numériques pour les assiettes, et le moustique, trouvez des situations dans lesquelles il existe une stratégie permettant au jongleur de maintenir les assiettes.

Question 3. Augmentez le nombre d’assiettes, et adaptez les valeurs numériques pour trouver une solution.