



UFR 919 Informatique – Master Informatique

Spécialité STL – UE 5I553 – PPC

Paradigmes de programmation concurrente

TME 5 — (2 h)

Uppaal

Carlos Agon

Question 1 : Prendre le temps pour dîner

Vous trouverez sur la page de l'UE le fichier `tme4Q2.xml` contenant une solution possible à la question 2 du dernier TME. Après l'avoir testé et compris, nous allons appliquer certaines modifications afin d'introduire la notion de temps.

Nous supposons qu'un client a besoin de minimum 5 secondes pour boire du champagne; 10 secondes pour le foie gras s'il prend la cuillère ou 15 secondes s'il prend la fourchette et 20 secondes pour manger la soupe de tortue.

Vous pouvez définir des horloges en UPPAAL de la même façon qu'on a défini des variables, mais cette fois en utilisant le mot `clock`. Ainsi `clock x` définit une horloge `x` initialisée à 0.

1. Définissez une horloge locale pour le dîner de chaque client.
2. A l'aide des champs `guard` et `update` sur les transitions, gérez les différents temps pour chaque mets.
3. Pour mesurer le temps total du repas, définissez une horloge globale.
4. Vérifiez s'il existe un état où le dîner est fini.
5. Si un tel état existe, utilisez l'option `Fastest` du menu `Diagnostic Trace` et analysez la trace proposée

Pour que le repas ne s'éternise pas, nous imposons au client de prendre maximum 7 secondes pour boire du champagne; 12 secondes pour le foie gras s'il prend la cuillère ou 17 s'il prend la fourchette et 22 secondes pour manger la soupe. Pour imposer ce type d'invariantes à un état, vous pouvez double-cliquer sur l'état désiré et éditer le champ `invariant`. En rentrant `x <= 7` dans l'état `DRINK`, nous spécifions qu'on peut rester dans cet état au maximum 7 unités de temps.

En ajoutant les contraintes précédentes au système, calculez le repas le plus long possible.

Question 2 : Un café-calva

Dans cet exercice, vous devez créer un système pour contrôler la bonne distribution du digestif emblématique du restaurant, à savoir, le café-calva. Le problème principal réside dans le fait qu'il n'y a qu'un seul verre pour servir le calvados et donc les clients doivent rivaliser pour l'accès à cette précieuse ressource. Une fois le repas fini, le service du digestif se passe de la manière suivante :

- le client demande au barman s'il peut boire un verre de calvados, 10 secondes après si le barman ne dit rien il doit traverser un couloir et ainsi boire en toute sécurité. La traversée lui prendra 10 secondes. Après avoir bu son verre (il faut entre 3 et 5 secondes pour boire un calvados), le client dit au barman qu'il a fini et retourne à table.
- après avoir reçu la commande, le barman a 10 secondes pour prévenir le client que le verre est occupé. Le client doit alors prendre un café et attendre que le barman lui donne l'instruction pour y aller. Une fois l'autorisation reçue, le client prendra entre 7 et 15 secondes pour rejoindre l'endroit où se trouve le verre de calvados.

Ecrivez en Uppaal un système qui modélise la bonne distribution du digestif et vérifiez que pour 5 clients :

1. Le barman peut recevoir des commandes à tout moment.
2. Le client 0 pourra boire son calvados.
3. Le client 2 peut boire pendant que le client 3 boit un café.
4. Le client 0 peut boire pendant que les autres boivent du café.
5. Un seul client à la fois peut boire du calvados.
6. Les 5 clients ne boivent jamais du café en même temps.
7. Chaque fois qu'un client passe commande, il peut éventuellement boire son calvados.