

Problème du Choix Distant

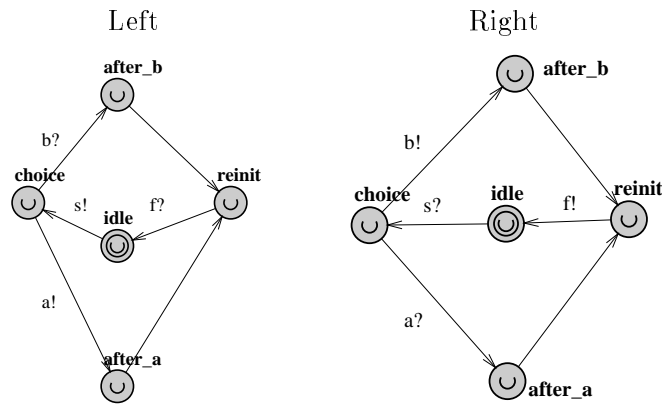
Préliminaires : Les sources du premier TP sont accessibles via

<http://www.laas.fr/~francois/UPPAAL/TP1>

Pour invoquer l'atelier UPPAAL, utilisez la commande uppaal.

1 Choix résolu par la synchronisation

On considère un système - décrit par le fichier `rdv.xta` - constitué par deux entités, **Left** et **Right** dont le comportement est donné ci-dessous;



Les deux composants se synchronisent sur **s**, puis ont le choix soit:

- d'une communication de gauche à droite (**gauche** envoie sur **a** tandis que **droit** reçoit sur **a**)
- d'une communication de droite à gauche (**droit** envoie sur **b** tandis que **gauche** reçoit sur **b**)

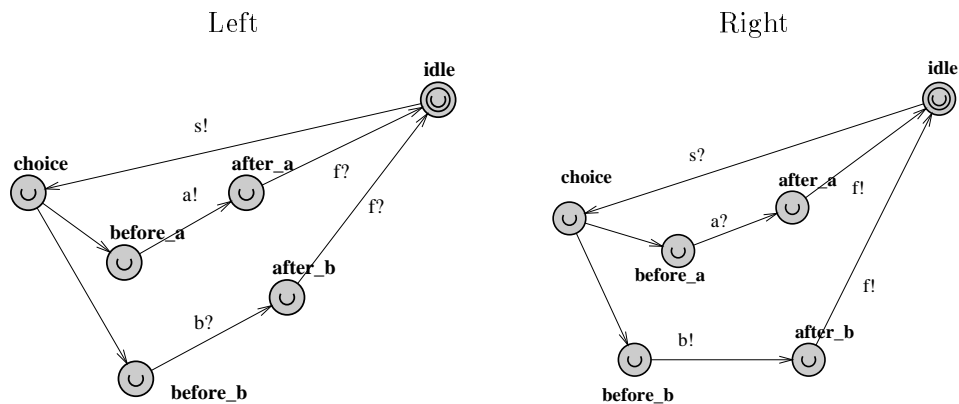
Après l'une ou l'autre de ces communications, chaque composant va de façon indépendante dans l'état **reinit**. Lorsqu'ils sont tous les deux dans l'état **reinit**, une synchronisation sur **f** a lieu et les deux composants retrouvent leur état initial.

Simulation

1. Exhibez une exécution où la communication a lieu de gauche à droite
2. Exhibez une exécution où la communication a lieu de droite à gauche
3. Persuadez-vous qu'il existe une exécution infinie comportant uniquement des communications de gauche à droite
4. Persuadez-vous qu'il existe une exécution infinie comportant uniquement des communications de droite à gauche.
5. Persuadez-vous que le système ne bloque pas.

2 Choix non déterministe

On considère maintenant le système décrit par le fichier `nondet.xta`. Le système est toujours constitué de deux entités `Left` et `Right` dont le comportement est donné ci-dessous;



2.1 Edition

Dessinez les automates associés à `gauche` et à `droite`

2.2 Simulation

1. Exhibez une exécution où la communication a lieu de gauche à droite
2. Exhibez une exécution où la communication a lieu de droite à gauche
3. Exhibez une exécution où le système est bloqué

3 Choix autoritaire

Pour résoudre le problème de choix, on admet que l'un des processus (`left`) est maître - décide du sens de communication - et que l'autre obéit. Vous chargerez le modèle simplifié `nondet.xta` de `Questions3_4`.

1. Implantez cette solution, vous utiliserez la synchronisation sur le canal `s` et une variable partagée pour transmettre le choix du maître à l'esclave.
2. Simulez le système et persuadez vous que tous les scénarios décrits dans la section 1 sont valides.
3. Pour vérifier - et **non plus se persuader** - que votre système est correct, allez sous le vérificateur. Chargez (menu `query`) le fichier de propriétés `specifs.q`. Assurez vous que chaque propriété est bien satisfaite.
4. Reprenez la solution précédente en utilisant un seul processus `agent` paramétré qui vous permettra - après instantiation - d'obtenir une instance du processus `left` et une instance du processus `right`.

4 Solution à autorité tournante

Pour obtenir une solution un peu plus symétrique, on veut maintenant que le statut de "maître" échoit à tour de rôle à chacun des deux processus. Implantez cette solution.

Simulez le système pour le "debugger". Assurez-vous qu'il vérifie les spécifications désirées.