Problème du Choix Distant en Asynchrone

Dans le cadre de ce TP, on veut étudier le problème de choix distant dans le cadre de communications asynchrones : `gauche` et `droit` vont communiquer par l' intermédiaire de files fifo.
Les sources pour ce TP sont à récupérer sous http://www.laas.fr/~francois/UPPAAL/ . Le système complet est décrit dans le fichier `choixasync.xta` . Les propriétés sont décrites dans le fichier `spec1.q`.

1 File avec et sans contrôle de flux

1. Simulez le système `choixasync.xta` .
2. Vérifiez les propriétés décrites dans le fichier `spec1.q`.
3. En quoi, le processus `fifo` assure-t-il un contrôle de flux ?
4. Modifiez le processus `fifo` de telle façon qu'il puisse - suivant son instantiation - assurer un contrôle de flux (même comportement que précédemment) ou non; dans ce cas vous ferez en sorte que tout ajoût dans une file ayant atteint sa capacité la met dans un état d'erreur.
5. Simulez le système avec le processus `fifo` ainsi modifié en considérant les deux cas: avec et sans contrôle de flux. Vérifiez dans les deux cas, les propriétés décrites dans le fichier `spec1.q`.
6. Expliquez l'état de blocage constaté sur la version sans contrôle de flux. Donnez une formule permettant de le caractériser.
7. En quoi serait-il illusoire de vouloir augmenter la capacité des files de communication ?

2 Contrôle de flux explicite entre les sites

Par la suite, on considère des communications asynchrones via des files de communication standard (sans contrôle de flux). Pour pouvoir remédier au problème constaté précédemment, les instances du processus `site` vont mettre en place elles mêmes un mécanisme de contrôle de flux via des acquittements basé sur la politique suivante : Après avoir émis, une entité ne retourne à son état initial qu'après avoir reçu l'acquittement correspondant au message envoyé.

1. Implantez cette solution et simulez la.
2. Vérifiez les propriétés décrites dans le fichier `spec1.q`.
3. Que peut-on dire pour les propriétés P1 et P3 ?
4. En quoi la politique d'acquittement adoptée introduit-elle forcément un interblocage ?

5. Un site est-il capable de détecter l'interblocage ?

6. Proposez une politique d'acquittement "plus souple" permettant d'éliminer l'interblocage et assurant que tout message émis est effectivement acquitté (Il vous faudra définir cette propriété et la vérifier). Que peut on adopter comme capacité minimale pour les files de communication?

3 Solution via un tirage au sort distribué

Tirage au sort distribué On considère le "jeu à deux joueurs" suivant: Chacun des deux joueurs - "left"et "right" - choisit une valeur particulière $\in \{0, 1\}$. Il la mémorise et l'envoi à son partenaire. Il récupère ensuite la valeur choisie par son partenaire. A l'issue de l'échange chaque joueur dispose de deux valeurs : celle qui l'a choisie et celle choisie par son partenaire. Left gagne si les valeurs échangées sont les mêmes, right gagne si les valeurs échangées sont distinctes.

1. On va utiliser ce service de "Tirage au sort" pour résoudre le problème de choix distant. Le fichier `cointossing.xta` offre un début de solution. Chaque site dispose d'un processus `player` dont le rôle est d'effectuer le tirage au sort (c'est à vous de l'implanter). La communication est synchrone entre un site et son joueur. Les joueurs communiquent de façon asynchrone en partageant les files de communication déjà existantes.

2. Ecrivez l'ensemble des propriétés à satisfaire par ce système. Assurez-vous que votre implantation les satisfait.

3. Le contrôle de flux est il nécessaire? Quelle dimension doit on prendre pour les files de communication?