

Outils formels de Modélisation

1^{er} Travail personnel

Dimitri Racordon, le 23.09.15

Date de rendu: Jeudi 12.10.17 à 23h55

Comme leur nom l'indique, ces travaux sont *personnels*. La copie est strictement interdite, et toutes similitudes entre deux rendus seront sanctionnées par la note de 0. Tout dépassement de la date et heure de rendu sera lourdement pénalisée. Date et heure de rendu sont toujours données en heure locale de Genève. Tout commit sur votre dépôt publié après la date de rendu sera ignoré.

Seule la branche **master** de votre dépôt sera prise en compte lors de la correction.

Votre code doit être compilable en Swift 4.0, avec un compilateur officiel. La note the 1 vous sera attribuée si votre code ne peut être compilé.

Dans ce TP, vous allez utiliser le formalisme des réseaux de Petri afin de corriger un comportement indésirable dans un système concurrent et non déterministe.

Prenez le temps d'étudier le système afin de bien comprendre le problème à corriger, ainsi que ses implications.

1 Spécification

Considérez le modèle d'un gestionnaire de tâches décrit par la spécification suivante:

1. Lorsqu'une tâche arrive, elle est placée dans ensemble des tâches à exécuter.
2. Lorsqu'un nouveau processus est prêt à exécuter une tâche, il est placé dans un autre ensemble.
3. Lorsque que c'est possible, le gestionnaire sélectionne une tâche et un processus pour l'exécuter. Le processus sélectionné est alors retiré de l'ensemble des processus disponibles.
4. Si la tâche réussit, elle est retirée de l'ensemble des tâches à exécuter et le processus est détruit.
5. Si la tâche échoue, elle reste dans l'ensembles de tâches à exécuter et le processus est détruit.

Une implémentation de cette spécification vous a été proposée en Fig. 1.1. Cependant lors des tests, il arrive occasionnellement qu'un processus ne puisse

jamais être détruit, malgré qu'il ait fini d'exécuter sa tâche. Votre mission est de corriger ce problème.

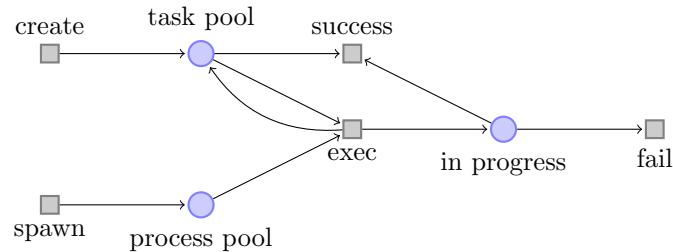


Figure 1.1: Gestionnaire de tâches

2 Mise en place du TP

Forkez le dépôt <https://github.com/cui-unige/outils-formels-modelisation>. Vous travaillerez sur votre propre version du dépôt, et effectuerez tous vos commit sur ce dépôt-ci.

Le projet pour ce TP se trouve dans le répertoire `tp-01`.

3 Analyse du problème

A l'aide du réseau de la Fig. 1.1, donnez un exemple d'exécution qui conduirait au problème décrit ci-dessus, sous la forme d'une suite d'exécution de la méthode `fire(from:)` sur les transitions appropriées.

4 Correction du problème

Modifiez le réseau de la Fig. 1.1 afin de corriger le problème identifié, en modifiant la définition du P/T-net construit par la fonction `createCorrectTaskManager`. Vous pouvez ajouter autant de places et de transitions que vous le souhaitez, mais vous ne pouvez rien retirer du réseau existant.

A l'aide de commentaires, expliquez en détail comment les modifications que vous avez apportées.

Enfin, montrez que la séquence de transition que vous aurez définie pour illustrer le problème ne conduit plus à un marquage problématique.