

Compte-rendu de réunion : 09/07/2014

Stage facultatif sur la génération de code parallèle pour les langages synchrones à l'IRIT avec l'équipe ACADIE

Présentation du travail effectué :

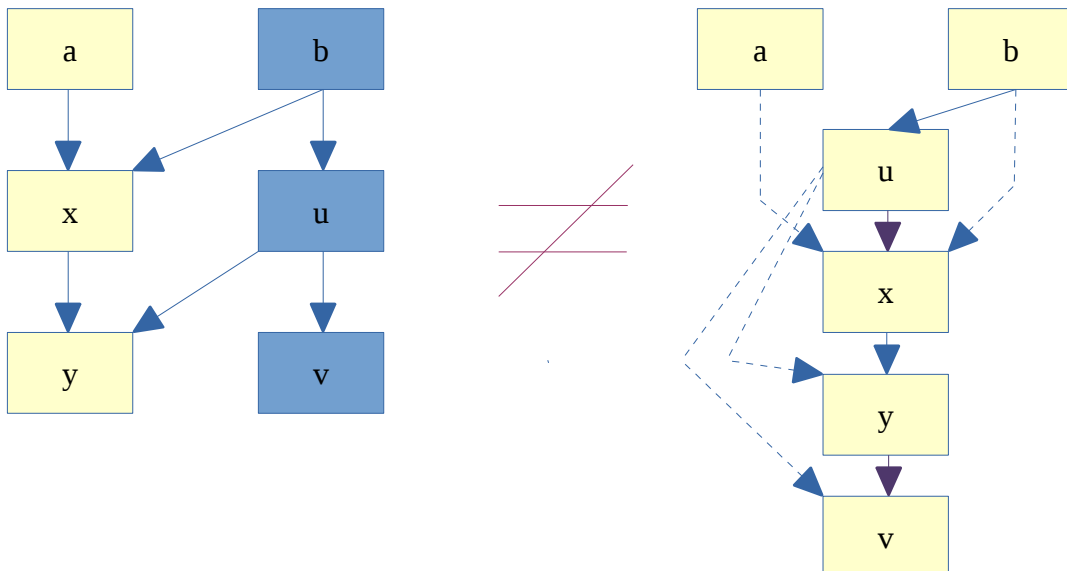
Présentation du template Java imaginé :

1. dynamique de méthodes set et get simples pour les cas généraux et de syn_set et syn_get pour des cas de variables synchronisées

2. solution du calcul des présences de variables trop lourde :

-> vérifier s'il est possible d'utiliser les propriétés du scheduler pour permettre à un thread qui aurait une faible priorité, de se lancer seulement si tout autre thread est bloqué. L'usage d'un boolean de présence ne serait donc plus nécessaire.

Remarque : côté Ocaml, si un thread peut être bloqué lors du calcul d'une variable synchronisée, le nombre de thread ne pourra pas toujours être diminué :



Dans le cas ci-dessus, on voit que si <a> non présent, un thread chargé du calcul de <x> bloque. Ainsi le graphe de droite ne calcul pas <v> qui pouvait être calculé sur le graphe de gauche.

3. Présentation de la structure Ocaml de graphes imaginée

Définition des nouveaux objectifs :

-> Mettre en place un template qui prévoit un environnement avec en entrée un fichier qui définit les variables d'entrées à chaque cycle et en sortie un second fichier qui définit l'ensemble des variables à chaque cycle.

-> Se documenter sur le fonctionnement des schedulers Java et Posix (Ocaml), de sorte à savoir s'il est possible de détecter le blocage de l'ensemble des threads au moment où ils sont tous bloqués.