Guillaume RIPOLL
L3 -> L3

Compte-rendu de réunion : 17/07/2014

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

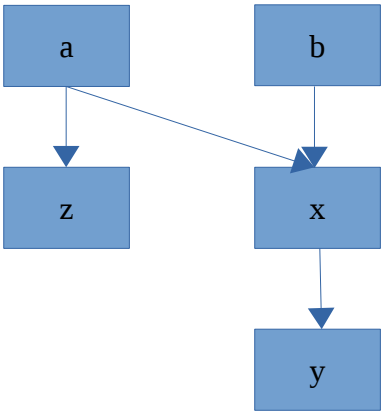
Présentation du travail effectué :

Présentation des différentes solutions Java implémentées :
-> gestion du deadlock non thread safe (cas de thread qui garde la main de l'instruction notifyAll() jusqu'au d.inc() sans que le deadlock ait le temps de se décrémenté)
-> solution avec booléens pas assez performante
-> utiliser l'implémentation avec un Executor :

Soit le code Signal :

```
process P = (? integer a; boolean b; ! integer x, y, z;)(
|x := a when b
|y := x +1
|z := a +3
|)end;
```

On peut alors générer, sous Ocaml, le graphe :



On associe chaque tâche à
- un nombre de dépendances
- une liste de tâches

Quand une tâche se termine bien
- elle décrémente de 1 le nombre de dépendances de chacune des tâches de sa liste
- à chaque fois qu'un nombre de dépendances devient égal à 0, on exécute la tâche

Quand une tâche se termine mal
- on ignore les tâches restantes

Résultats : dans une variable globale