



Éléments de correction Examen final réparti d'ILP

Christian Queinnec

5 janvier 2012

Voici quelques éléments de correction de l'examen de novembre 2013, les codes correspondant sont dans Mercurial sous l'étiquette `partiel2013nov`.

La fonction `allouer` dont l'emploi était suggéré pour l'écriture des tests des fonctions `memoryGet` et `memoryReset` devait bien sûr être écrite pour ILP1 même si l'on utilisait du pseudo-code. Tout appel au `malloc` de C était donc inapproprié. On demandait des tests c'est-à-dire des programmes qui vérifient tout seuls et sans l'intervention d'un humain qu'ils fonctionnent correctement ! Beaucoup de ces tests faisaient une hypothèse implicite sur le type de machine qui les exécutaient ce qui nécessite donc un jeu de tests différent par machine : pourquoi ne pas écrire des tests indépendants de la machine ?

Encore et toujours des copier-coller dangereux : des `CEATprogram` héritant d'ILP2 et non d'ILP3 ! Quelques rares mélanges de temps entre compilation et exécution du genre : la variable comptant les allocations étant une variable Java et non C !

L'implantation des fonctions `memoryGet` et `memoryReset` devait être conformes au protocole d'ILP à savoir rendre des objets ILP et non des `int` C. La taille de la pile C n'avait aucun rapport avec le problème posé. Définir `ILP_malloc` dans le fichier C engendré par `CEATprogram` permettait de masquer son homonyme dans `ilp.c`.

Si la question 3 présentait un exemple, la question ne se limitait pas à l'amélioration de ce programme. Le remplacement des constantes par des références à des variables globales ne peut se faire uniquement dans les boucles car le corps des fonctions peut aussi en profiter.

La pré-allocation des constantes 42 et 43 ne sert à rien si la fonction `ILP_make_integer` n'est pas modifiée pour en tenir compte.