

Examen final d'ILP

1ère session

Christian Queinnec

19 décembre 2008

Conditions générales

Cet examen est formé d'un unique problème en plusieurs questions auxquelles vous pouvez répondre dans l'ordre qui vous plaît.

Le barème est fixé à 20 ; la durée de l'épreuve est de 3 heures. Tous les documents sont autorisés et notamment ceux du cours.

Votre copie sera formée de fichiers textuels que vous laisserez aux endroits spécifiés dans votre espace de travail pour Eclipse. L'espace de travail pour Eclipse sera obligatoirement nommé `workspace` et devra être un sous-répertoire direct de votre répertoire personnel.

À l'exception des clés USB en lecture seule, tous les appareils électroniques sont prohibés (y compris les téléphones portables, les assistants numériques personnels et les agendas électroniques).

L'examen sera corrigé à la main, il est donc absolument inutile de s'acharner sur un problème de compilation ou sur des méthodes à contenu informatif faible. Il est beaucoup plus important de rendre aisé, voire plaisant, le travail du correcteur et de lui indiquer, par tout moyen à votre convenance, de manière claire, compréhensible et terminologiquement précise, comment vous surmontez cette épreuve. À ce sujet, vos fichiers n'auront que des lignes de moins de 80 caractères, n'utiliseront que le codage ASCII ou UTF-8 enfin, s'abstiendront de tout caractère de tabulation.

Le langage à étendre est ILP4. Le paquetage Java correspondant à cet examen sera donc nommé `fr.upmc.ilp.ilp4array`. Sera ramassé, à partir de votre *workspace*, tout répertoire ou fichier ayant le fragment `4array` dans son nom.

1 Introduction de tableaux

On souhaite étendre ILP4 avec des tableaux unidimensionnels (des vecteurs) statiquement nommés. Le programme (inepte) suivant illustre les opérations (déclarations ou expressions) possibles sur les tableaux :

```
dimension tab[40]           // définition du tableau tab
dimension vec[1]            // définition du tableau vec
function foo (i, j) {
    i + vec[0]*tab[j]       // lecture
}
let k = 0
while ( k < tab'size ) {    // taille
    tab[k+1] = foo(k, k-1)  // écriture
    k = k+1
}
```

Les tableaux doivent être définis (ici avec une notation à la Fortran) avant les fonctions, avec une taille qui ne peut être qu'une constante. Les positions dans le tableau peuvent être lues ou écrites, elles sont indexées à partir de zéro (comme en C). La taille du tableau peut être obtenue (ici avec une notation à la Ada). Cette extension doit être sûre : toute lecture ou écriture en dehors du tableau signalera une exception.

Question 1 – Grammaire (2 points)

Étendre la grammaire d'ILP4 pour inclure les nouvelles déclarations et expressions autour des tableaux. Vous pouvez abuser de commentaires pour y insérer les points les plus pertinents de vos réflexions.

Livraison

- un fichier `grammar4array.rnc` placé dans le répertoire `Grammars`.

Question 2 – Programme (2 points)

Écrire un programme en syntaxe XML conforme à la grammaire précédente et utilisant toutes ces nouvelles extensions.

Livraison

- un fichier `a-4array.xml` placé dans le répertoire `Grammars/Samples`.

Question 3 – Interprétation (5 points)

Écrire les méthodes d'évaluation des expressions ajoutées. Vous préciserez en commentaire dans ces classes comment vous représentez les tableaux, où vous les créez et où vous les stockez.

Livraison

- les fichiers correspondant à ces classes placés dans le paquetage `fr.upmc.ilp.ilp4array`.

Question 4 – Suppression de tableaux inutiles (3 points)

Lorsqu'un tableau est déclaré mais qu'il n'est pas utilisé c'est-à-dire qu'il n'est ni lu ni écrit (mais que peut-être sa taille est consultée), on peut transformer le programme en un nouveau programme débarrassé de ce tableau. Vous préciserez, en commentaire de la classe `DeadArrayRemoval`, plus finement cette transformation (et notamment les schémas de transformations tels qu'utilisés dans le cours), quand elle doit être effectuée ; vous indiquerez aussi comment cette transformation pourrait être implantée.

Livraison

- une unique classe `DeadArrayRemoval` placé dans le paquetage `fr.upmc.ilp.ilp4array`.

Question 5 – Suppression de petits tableaux (2 points)

Lorsqu'un tableau est déclaré avec une taille de 1, une variable globale peut aisément le remplacer. Vous préciserez, en commentaire de la classe `SmallArrayRemoval`, plus finement cette transformation (et notamment les schémas de transformations tels qu'utilisés dans le cours), quand elle doit être effectuée ; vous indiquerez aussi comment cette transformation pourrait être implantée.

Livraison

- une unique classe `SmallArrayRemoval` placé dans le paquetage `fr.upmc.ilp.ilp4array`.

Question 6 – Compilation (6 points)

Écrire les schémas de compilation (tels qu'utilisés dans le cours) correspondant aux expressions ajoutées et au nouveau patron de génération du fichier C. Vous les préciserez dans les commentaires des classes concernées.

Livraison

- les fichiers correspondant à ces classes placés dans le paquetage `fr.upmc.ilp.ilp4array`.