



Examen réparti d'ILP

Christian Queinnec

2 février 2012

Conditions générales

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

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 communicants sont prohibés (et donc notamment les téléphones portables). Vos oreilles ne doivent pas être reliées à ces appareils.

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 sera nommé `fr.upmc.ilp.ilp4ret`. Sera ramassé, à partir de votre `workspace` (situé sous ce nom directement dans votre HOME), le seul répertoire `ILP/Java/src/fr/upmc/ilp/ilp4ret/` et tout ce qu'il contient.

Pour vous éviter de la taper à nouveau, voici l'url du site du master (mais il faut, pour y accéder, ne plus passer par le proxy) et celle de l'ARI où se trouvent de nombreuses documentations dont celle de Java :

```
http://www-master.ufr-info-p6.jussieu.fr/site-annuel-courant/  
http://www-ari.ufr-info-p6.jussieu.fr/
```

Introduction

En Pascal, la valeur que renvoie une fonction f est spécifiée par une sorte d'affectation où f est le nom de la pseudo-variable affectée. Ainsi, peut-on écrire, en pseudo-Pascal :

```
function moinsun (x) {
  if ( x > 0 ) {
    moinsun := x-1
  }
}

function fact (n) {
  if n < 2
  then
    fact := one()
    print("OK")
  else
    fact := n * fact(moinsun(n))
    fact(one())
  end
}

function quoi (x, y) {
  x + y
}

function one () {
  one := 1
  print("coucou")
}

// Ainsi fact(3) imprime coucou et OK trois fois de suite et renvoie 6
```

On souhaite ajouter cette caractéristique à ILP4. On devra aussi assurer, lorsqu'une fonction est invoquée, que la valeur de retour est bien spécifiée et qu'elle ne l'est qu'une unique fois.

Voici la grammaire `grammar4ret.rnc` spécifiant cette sorte d'affectation permettant de spécifier la valeur de retour de la fonction englobante.

```
include "grammar4.rnc"
expression |= finalValue

finalValue = element finalValue {
  element valeur {
    expression
  }
}
```

Voici un fragment XML correspondant à la fonction moinsun :

```
<programme4>
<definitionFonction nom='moinsun'>
<variables><variable nom='x' /></variables>
<corps>
<alternative>
<condition><operationBinaire operateur='>'>
<operandeGauche><variable nom='x' /></operandeGauche>
<operandeDroit><entier valeur='0' /></operandeDroit>
</operationBinaire></condition>
<consequence><finalValue><valeur>
<operationBinaire operateur='- '>
<operandeGauche><variable nom='x' /></operandeGauche>
<operandeDroit><entier valeur='1' /></operandeDroit>
</operationBinaire></valeur></finalValue></consequence>
</alternative></corps></definitionFonction>
...
```

Remarque : pour cet examen, vous vous interdirez d'utiliser le DOM mais utiliserez directement l'AST.

Question 1 – Stratégie (7 points)

Vous explicitez les grandes lignes de votre solution dans le fichier `strategie.txt` en particulier comment vous traitez l'élément `finalValue` et l'intégration (*inlining*).

Livraison

- le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp4ret/strategie.txt`

Question 2 – Interprétation (7 points)

Vous rédigez la classe `CEASTfinalValue`. Vous indiquerez en particulier le code de la méthode `eval` en mentionnant en commentaire les hypothèses sur lesquelles il repose.

Livraison

- le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp4ret/CEASTfinalValue.java` et les classes associées nécessaires.

Question 3 – Compilation (6 points)

Vous indiquerez le schéma de compilation de l'élément `finalValue` ainsi que les autres schémas de compilation ou de transformation associés.

Livraison

- le fichier `workspace/ILP/Java/src/fr/upmc/ilp/ilp4ret/CEASTfinalValue.java` et les fichiers associés nécessaires.