

# Interprétation des programmes – TP 2 :

## Données du premier ordre

Université Paris Diderot – Master 1

(2014-2015)

Nous avons introduit un nouveau langage en cours. Il s'agit de DATIX, un langage qui étend FOPIX avec des types et des constructeurs/destructeurs de données.

Cette séance de travaux pratiques a pour objectifs :

- de vous faire lire et étendre le code source d'un vérificateur de types simples ;
- de continuer votre apprentissage de l'écriture d'un interprète à travers l'écriture de celui de DATIX ;
- de vous faire concevoir une passe de compilation qui fait un *choix de représentation* pour des données.

Le code source correspondant à ces travaux pratiques se trouve sur le GIT, dont on rappelle l'URL :

<http://moule.informatique.univ-paris-diderot.fr:8080/Yann/compilation-m1>

On rappelle que vous devez faire des *commits* réguliers (à chaque modification de votre code) pour que nous puissions suivre votre avancement.

### Exercice 1 DATIX est une extension de FOPIX

1. Quelles différences notez-vous entre FOPIX et DATIX ?
2. Dans l'interprète de DATIX, quels cas de l'interprète de FOPIX sont réutilisables ?

□

### Exercice 2 Vérificateur de types de DATIX

1. Pourquoi l'implémentation fournie du vérificateur de types de DATIX est-elle incorrecte pour le cas des fonctions mutuellement récursives ?
2. Dans quel cas peut-on se passer de spécifier le type de retour d'une fonction ? Écrivez plusieurs tests positifs et négatifs pour décrire cette caractérisation.<sup>1</sup>
3. Modifier le vérificateur de types de DATIX pour traiter les fonctions mutuellement récursives en ne demandant un type de retour que uniquement lorsque c'est nécessaire.

□

### Exercice 3 N-uplets en DATIX

1. Quel constructeur de l'arbre de syntaxe abstraite de DATIX permet de représenter un n-uplet ?
2. Quelles formes de motifs (patterns en anglais) permettent d'observer les composantes d'un n-uplet ? Écrivez plusieurs tests positifs qui utilisent ces constructions du langage.
3. Étendre l'interprète de DATIX pour traiter les cas de la construction et de l'observation d'un n-uplet. Quelles erreurs peuvent apparaître pendant l'exécution de ces cas ?
4. Étendre la vérification des types pour traiter les cas de la construction et de l'observation des n-uplets.
5. Comment les blocs de données de FOPIX peuvent-ils être utilisés pour représenter un n-uplet de DATIX ?

---

1. Dans cette question, on suppose bien sûr qu'on garde l'architecture actuelle du vérificateur de type sans se lancer dans l'écriture d'un moteur d'inférence de type.

6. À votre avis, comment les motifs d'observation des composants de  $n$ -uplets doivent-ils être traduits en FOPIX ? Appliquez votre idée manuellement sur l'exemple suivant :

```
val t = (1, 2, 3)
val (x, y, z) = t
```

7. Étendre la passe de compilation de DATIX vers FOPIX pour qu'elle prenne en compte les  $n$ -uplets.

□

#### Exercice 4 Enregistrements en DATIX

1. Quel constructeur de l'arbre de syntaxe abstraite de DATIX permet de représenter un enregistrement ?
2. Quel constructeur de l'arbre de syntaxe abstraite de DATIX permet de représenter l'observation de la valeur d'un champ d'un enregistrement ?
3. Étendre l'interprète de DATIX pour traiter les cas de la construction et de l'observation d'un enregistrement. Quelles erreurs peuvent apparaître pendant l'exécution de ces cas ?
4. Étendre la vérification des types pour traiter les cas de la construction et de l'observation des enregistrements.
5. Comment les blocs de données de FOPIX peuvent-ils être utilisés pour représenter un enregistrement de DATIX ? (Indication : utiliser un ordre sur les étiquettes des champs.)
6. Étendre la passe de compilation de DATIX vers FOPIX pour qu'elle prenne en compte les enregistrements.

□

#### Exercice 5 Données étiquetées DATIX

1. Quel constructeur de l'arbre de syntaxe abstraite de DATIX permet de représenter la construction d'une donnée étiquetée ?
2. Quel constructeur de l'arbre de syntaxe abstraite de DATIX permet de représenter l'observation de l'étiquette apposée sur une donnée ?
3. Étendre l'interprète de DATIX pour traiter les cas de la construction et de l'observation d'une donnée étiquetée. Quelles erreurs peuvent apparaître pendant l'exécution de ces cas ?
4. Étendre la vérification des types pour traiter les cas de la construction et de l'observation des données étiquetées.
5. Comment les blocs de données de FOPIX peuvent-ils être utilisés pour représenter une donnée étiquetée de DATIX ?
6. Étendre la passe de compilation de DATIX vers FOPIX pour qu'elle prenne en compte les données étiquetées.

□