

# Interprétation des programmes – TP 4 :

## Fonctions mutuellement récursives

Université Paris Diderot – Master 1

(2014-2015)

Cette dernière feuille de TP vous donne les étapes à suivre pour traiter les fonctions mutuellement récursives de HOPIX.

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 *N-uplets modifiables dans HOPIX*

1. Observez la nouvelle construction `MUTATETUPLE` de HOPIX, et DATIX. Elle sert à modifier la valeur d'un composant d'un *n-uplet*. `MutateTuple (e1, i, e2)` signifie que le composant numéro *i* du *n-uplet* produit par l'évaluation de *e1* doit prendre la valeur de *e2*.
2. Dans les interprètes de HOPIX et DATIX, écrire le cas de `MUTATETUPLE`.
3. Étendre les passes de compilation (de HOPIX vers DATIX et de DATIX vers FOPIX) pour prendre en compte `MUTATETUPLE`.

□

### Exercice 2 *Les fonctions mutuellement récursives de HOPIX*

1. Pourquoi la construction `MUTATETUPLE` est-elle utile pour construire des fermetures récursives ?
2. Dans le compilateur de HOPIX vers DATIX, complétez les fonctions `closure_conversion` et `hoist` pour qu'elles traitent les fonctions mutuellement récursives. (Le commentaire fourni dans le code devrait vous donner toutes les indications nécessaires.)

□