

Interprétation des programmes – TP 4 :

De Hobix à Retrolix, en sautant Fopix

Université Paris Diderot – Master 1

(2016-2017)

Cette séance de travaux pratiques a pour objectifs de :

- vous faire comprendre la sémantique de FOPIX ;
- vous faire implémenter de façon partielle la traduction de HOBIX vers FOPIX ;
- vous faire implémenter une traduction de FOPIX vers RETROLIX sans les conventions d'appel.

1 Le saut par dessus Fopix

Exercice 1 (Sémantique de Fopix)

1. Quelles sont les différences entre les langages HOBIX et FOPIX ?
2. Que contient l'environnement d'évaluation de l'interprète de FOPIX ?
3. Complétez les cas manquants de l'interprète de FOPIX.

□

Exercice 2 (Traduction partielle de Hobix vers Fopix)

Pour le moment, on se contente de traduire un sous-ensemble de HOBIX où :

- les programmes ne contiennent pas de fonctions anonymes ;
- les seules fonctions définissables sont les fonctions "toplevels" mutuellement récursives ;
- les appels de fonctions ne peuvent se faire sur des identificateurs de fonctions connues.

1. Complétez la fonction de traduction des expressions.
2. Complétez la fonction de traduction des fonctions mutuellement récursives.
3. Que se passe-t-il si le programme source contient plusieurs fonctions de même nom définies à toplevel ?

□

2 Traduction de Fopix vers Retrolix

Dans cette partie, on implémente une traduction de FOPIX vers RETROLIX dont les programmes générés ne respectent pour l'instant pas les conventions d'appel. Ce sujet fera l'objet de la séance suivante de travaux pratiques.

Exercice 3 (Sémantique de Retrolix)

1. Quelles sont les différences entre les langages FOPIX et RETROLIX ?
2. Que contient l'environnement d'évaluation de l'interprète de RETROLIX ?
3. Les fonctions de RETROLIX sont-elles mutuellement récursives ?
4. Comment sont (actuellement) implémentés les opérateurs de comparaison de RETROLIX ?
5. Quelles sont les valeurs initiales des variables locales ?
6. Comment l'interpréteur détermine quelles sont les variables locales d'une fonction ?

□

Exercice 4 (Compilation)

1. Implémenter `FopixToRetrolix.locals`.
2. Compléter les appels de fonction dans `FopixToRetrolix.expression`.
3. Écrire le code RETROLIX correspondant au code FOPIX suivant :

```
val x :=  
  val x := 5;  
  val y := 2 * x + x * x + 1;  
  val x := 3;  
  x + y
```

En déduire le rôle de la fonction `FopixToRetrolix.preprocess`.

4. Implémenter `FopixToRetrolix.preprocess`.
5. Écrire sur papier le code compilé RETROLIX correspondant au programme FOPIX suivant :

```
val x :=  
  if 1 <= 0 then  
    1  
  else  
    0
```

6. Compléter le cas des expressions conditionnelles et du Switch dans la fonction `FopixToRetrolix.expression`.

□