Programmation Fonctionnelle

Ralf Treinen

Université Paris Diderot
UFR Informatique
Laboratoire Preuves, Programmes et Systèmes

treinen@pps.univ-paris-diderot.fr

September 18, 2013

Organisation

- Les TP commencent la semaine du 23 septembre
- Période des examens : en janvier
- Il y a un projet de programmation, mais pas de partiel

Contrôle de connaissances

- Première session :

$$\frac{1}{3} * \mathrm{proj}\ \mathrm{t} + \frac{2}{3} * \ \mathrm{xam}$$

- Deuxième session :

$$\max(\frac{1}{3} * \mathrm{proj}\ \mathrm{t} + \frac{2}{3} * \ \mathrm{xam2},\ \mathrm{xam2})$$

Organisation

- `http://www.pps.univ-p ris-diderot.fr/~treinen/ te ching/pf5/`
- Support : copies des transparents
- Il y a des ressources en ligne (voir la page web du cours)
- Il est indispensable d'assister au cours et au TP, et de faire le projet.
- Inscrivez-vous à la liste de diffusion du cours (voir le lien sur la page du cours) !

Le projet

Ceci n'est pas

- Générateur de structures récursives (voir la démo)
- À faire en binôme
- Plus sur l'organisation du projet : voir les TP

Hello, World en Java

```
class Helloworld {

  public static void main(String []args) {
    System.out.println("Bonjour");
  }

}
```


Hello World en Objective Caml

```
print_string "Hello, World!\n"
```


La classe `Formule` de OL3

Définition de type en Objective Caml :

```
type formule =
  Var of string
| Negation of formule
| And of formule * formule
| Or of formule * formule
```

Fonctions pour l'évaluation, mise en forme normale, etc. s'écrivent (presque) comme les définitions recursives vues en Cours OL3.

Programmation fonctionnelle

Definition : Langage fonctionnel

Langage qui permet de manipuler des fonctions comme des valeurs de première classe (entiers, chaînes de caractères).

En particulier, on peut écrire :

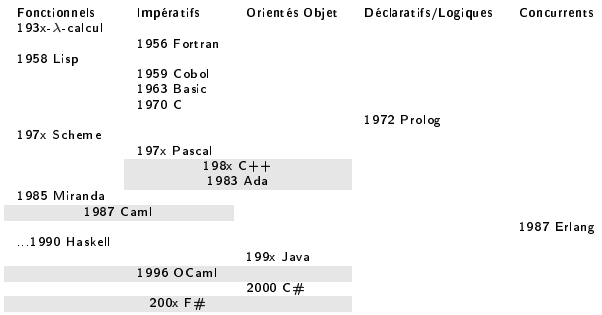
- des fonctions qui prennent des fonctions en argument ;
- des fonctions qui renvoient des fonctions créées dynamiquement.

Pourquoi est-ce utile ?

- Normalisation des formules vues en OL3 :
- On avait vu plusieurs formes normales (NNF, CNF, DNF)
- Pour chacune, on écrit une fonction qui exécute une étape de normalisation (par exemple : `nnf-step`)
- On écrit une fonction `fixpoint` qui prend en argument une fonction f et une formule p, et qui applique f sur p jusqu'à rien ne change plus (point fixe)
- $nnf(p) = fixpoint(nnf - step, p)$
  (la syntaxe va être légèrement différente)

On distingue :

- les langages fonctionnels *purs* (Haskell...)
- les langages qui combinent fonctionnel et impératif (OCaml)



Applications en OCaml

- Langage généraliste
- Exemples d'applications en OCaml : Unison, Coq, MLDonkey, Ocsigen, ...
- Nombreuses utilisations industrielles (Dassault, Lexifi, Jane Street Capital, Baretta, Merjis, RedHat, Mandriva, ...)

Histoire

- 1973 ML Milner (tactiques de preuves pour le prouveur LCF)
- 1980 Projet Formel à l'INRIA (Gérard Huet), Categorical Abstract Machine (Pierre-Louis Curien)
- 1984-1990 Définition de SML (Milner)
- 1987 *Caml* (implémenté en Lisp) Guy Cousineau, Ascander Suarez, (avec Pierre Weis et Michel Mauny)
- 1990-1991 *Caml Light* par Xavier Leroy (et Damien Doligez pour la gestion de la mémoire)
- 1995 *Caml Special Light* puis 1996 *OCaml* (Xavier Leroy, Jérôme Vouillon, Didier Rémy, Michel Mauny)

Principaux traits

- typage statique avec inférence de types ;
- fonctions comme objets de première classe (fonctionnel) ;
- polymorphisme paramétrique ;
- types sommes et *pattern matching* (filtrage) ;
- gestion d'exceptions ;
- gestion de mémoire automatisée (*Garbage Collector*) ;
- modules paramétrables (et récursifs) ;
- système de classes évolué (objets) avec inférence de type et classes paramétriques ;
- compilateur natif (pour de nombreuses architectures) et bytecode ;
- préprocesseur expressif et sûr (Camlp4) syntaxe modifiable

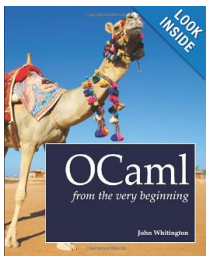
Pourquoi apprendre OCaml ?

- Nécessaire pour d'autres cours à Paris-Diderot
- Richesse conceptuelle (apprentissage d'autres langages)
- Préfigure les langages du futur (Introduction des clôtures dans C#)
- De plus en plus d'entreprises comprennent l'intérêt de programmer plus proprement pour réduire le temps de débogage
- Parcours Langages et Programmation du Master Pro

Bibliographie

- Xavier Leroy et al : *The Objective Caml system : documentation and user's manual*
`http://c ml.inri .fr`
- Emmanuel Chailloux, Pascal Manoury et Bruno Pagano : *Développement d'Applications avec Objective Caml*
O'Reilly, 2000 disponible en ligne

Vient de paraître



John Whitington : *OCaml from the Very Beginning*
S'adresse plutôt à des débutants.
(a été commandé pour la bibliothèque centrale)

Programmation Fonctionnelle
└Introduction
└Bibliographie

Autres ouvrages

Guy Cousineau et Michel Mauny
Approche fonctionnelle de la programmation
Dunod, 1995

Pierre Weis et Xavier Leroy
Le langage Caml
Dunod, 1999

Programmation Fonctionnelle
└Introduction
└Bibliographie

Bibliographie

Catherine Dubois et Valérie Ménissier-Morain
Apprentissage de la programmation avec OCaml. Hermès, 2004

Louis Gacogne
Programmation par l'exemple en Caml
Ellipse, 2004

Philippe Nardel
Programmation fonctionnelle, générique et objet : Une introduction avec le langage OCaml
Vuibert, 2005