

Organisation

Programmation Fonctionnelle

Ralf Treinen



Université Paris Diderot
UFR Informatique
Laboratoire Preuves, Programmes et Systèmes

treinen@pps.univ-paris-diderot.fr

September 20, 2012

- ▶ Les TP commencent la semaine du 24 septembre
- ▶ Période des examens : en janvier
- ▶ Il y a un projet de programmation, mais pas de partiel
- ▶ Attention L2 Math-Info:
 - Groupe 1 : Mardi 14h30-16h30, salle 554C
 - Groupe 2 : Lundi 10h30-12h30, salle 548C

Contrôle de connaissances

Organisation

- ▶ Première session :

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

- ▶ Deuxième session :

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

- ▶ <http://www.pps.univ-paris-diderot.fr/~treinen/teaching/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 TP

Le projet

- ▶ Traitement d'image : réduction de la largeur, en conservant les proportions (voir la démo)
- ▶ À faire en binôme
- ▶ Plus sur l'organisation du projet : voir les TP

Ceci n'est pas seulement
un cours
de Programmation Fonctionnelle

Objective Caml

- ▶ Ceci est (plutôt) un cours de programmation en Objective Caml
- ▶ C'est un langage **multi-paradigme** : il permet
 - ┆ Programmation impérative
 - ┆ Programmation fonctionnelle
 - ┆ Programmation orientée objet
 - ┆ Graphisme, applications réseaux, ...

Programmation Fonctionnelle ?

- ▶ Sera expliquée plus tard
- ▶ Un paradigme de programmation différent
- ▶ Permet d'écrire des programmes qui sont plus proche au problème donné
- ▶ ... et cela sans perte d'efficacité (importante)

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)

Fonctionnels	Impératifs	Orientés Objet	Déclaratifs/Logiques	Concurrents
193x: λ -calcul				
1958 Lisp	1956 Fortran			
	1959 Cobol			
	1963 Basic			
	1970 C			
197x Scheme			1972 Prolog	
	197x Pascal			
	198x C++			
	1983 Ada			
1985 Miranda				
1987 Caml				
...1990 Haskell				1987 Erlang
	199x Java			
	1996 OCaml			
	2000 C#			
	200x F#			

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 ?

- ▶ Autres cours (compil...)
- ▶ 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://cml.inria.fr>
- ▶ Emmanuel Chailloux, Pascal Manoury et Bruno Pagano : *Développement d'Applications avec Objective Caml*
O'Reilly, 2000 disponible en ligne

Autres ouvrages

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

Pierre Weis et Xavier Leroy
Le langage Caml
 Dunod, 1999

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