Licence informatique & vidéoludisme
Semestre 5

Interprétation et compilation

Chapitre 1
Introduction à OCaml

Pablo Rauzy <pr@up8.edu>
pablo.rauzy.name/teaching/ic

Introduction à OCaml

- Créé en 1996 faisant suite à Caml Light (début 90) et Caml (1985).
- Langage fonctionnel se voulant pragmatique (permet l'impératif et l'objet).
- Typage statique fort et inférence de type, types algébriques.
- Dispose d'un interpréteur (`ocaml`), d'un compilateur bytecode (`ocamlc`), et d'un compilateur natif (`ocamlopt`).

Jouons un peu avec le langage

- Vous savez déjà programmer et avez déjà eu un cours de programmation fonctionnelle.
- Voyons ensemble la syntaxe et les spécificités d'OCaml :
  - valeurs et types de bases ;
  - expressions, variables, et inférence de type ;
  - fonctions et curryfication ;
  - listes et fonctions récursives ;
  - filtrage (*pattern matching*) ;
  - types algébriques (ADT) et filtrage ;
  - types récursifs et paramétrés ;
  - filtrage avec garde ;
  - inférence de type et filtrages non-exhaustifs ;
  - enregistrements et déclarations directs dans les ADT ;
  - modules "fichiers", modules "déclarés", et foncteurs.

Outils de développement

- Site et documentation : `https://ocaml.org/`
  - bibliothèque standard (et les alternatives Batteries et Core).
- Gestionnaire de paquets `opam` : `https://opam.ocaml.org/`.
- Moteur de production `ocamlbuild` (alternative : `dune`).
- REPL `utop`.
- IDE dans Emacs avec `tuareg` et `merlin` (alternative : serveur LSP).
- Autres outils : `ocamldebug`, `ocamlprof`, `ocamldoc`, `ocamldep`, ...

Un “vrai” projet OCaml

- Étudions ensemble un projet petit mais complet écrit en OCaml.
- Il s’agit un compilateur pour un langage très simple d’expression booléenne.
- Il compile vers un petite machine virtuelle simple.
- Au passage, il s’agit d’un exemple simplifié du type de chose que l’on va apprendre à faire dans ce cours.
- `https://code.up8.edu/-/snippets/3`