

プログラム言語論

亀山幸義

筑波大学 情報科学類

No. 2b

目次

① OCaml の利用

OCaml の利用法あれこれ

OCaml が動くコンピュータ

- ▶ 情報科学類 計算機室の Linux マシンで ocaml コマンド
 - ▶ 3C113 計算機室に行って直接使う。
 - ▶ 自分の PC から ssh user@azalea01.coins.tsukuba.ac.jp でログイン
 - ▶ (01 の部分は、30までの異なる数にすると負荷分散できる)
- ▶ 自分の PC に OCaml を載せて使う
 - ▶ Linux: Debian 系なら sudo apt-get install ocaml
 - ▶ MacOS: brew install ocaml
 - ▶ Linux, MacOS でのより良い方法: 最初に opam を載せ、opam を使って OCaml を載せる。
 - ▶ Windows: 可能
- ▶ ウェブサービスを使う TryOCaml
 - ▶ try.ocamlpro.com

OCaml のファイルの作成

Emacs エディタを使う場合

- ▶ tuareg をいれると、使い勝手が格段に向上する。
- ▶ cat ~kam/ocaml/tuareg.el >> ~/.emacs とするとよい。

vscode でエディタを使う場合

- ▶ 拡張機能には例えば OCaml Platform がある
<https://github.com/ocaml-labs/vscode-ocaml-platform>
- ▶ 言語サーバのインストールなど依存解決が必要であり、詳しくは上記公式ページの Getting started の項を読むこと。
- ▶ OCaml ごと opam を使って実行環境を整える方が簡単かも。

OCaml 起動テスト

対話的に使えるかのテスト:

```
$ ocaml  
OCaml version 4.14.1
```

```
# 1 + 2 ;;  
- : int = 3  
#
```

(対話モードを終了したいときは、ctl-dなどを押す)。

OCaml ソースコードがはいったファイル test1.ml を作成:

```
let f x = x + 1 ;;
```

上記の test1.ml を対話的に使う:

```
# #use "test1.ml";;  
val f : int -> int = <fun>  
# f 3 ;;  
- : int = 4
```

OCaml ファイルのコンパイル

コンパイルして高速化したい場合、まずソースファイル test2.ml:

```
let f x = x + 1;;
let res = print_endline (string_of_int (f 5)) ;;
```

上記をコンパイルして実行 (OCaml の外でやる):

```
$ ocamlc test2.ml      (a.out ファイルができる)
$ ./a.out                (実行する)
6                        (print されたものが表示される)
```

上記をネイティブコンパイルして実行 (OCaml の外でやる):

```
$ ocamlopt test2.ml    (a.out ファイルができる)
$ ./a.out                (実行する)
6                        (print されたものが表示される)
```

インターネット上の記事 (非常に多数あり、良質な解説も多い):

日本語の本:

- ▶ 浅井健一, 「プログラミングの基礎」, サイエンス社, 2007.
- ▶ 五十嵐淳, 「プログラミング in OCaml」, 技術評論社, 2007.

最新の情報:

- ▶ 日本の OCaml ユーザ: <https://ocaml.jp>
- ▶ OCaml の本家: <https://ocaml.org>