

情報システム科学実習 II

第 1 回 授業概要

担当: 山口 和紀・五十嵐 淳

2001 年 10 月 10 日

次回以降は
情報教育南棟 4 階中演習室 1
にて実習を行います。

1 授業内容紹介より

OCaml 言語によるプログラミング演習を通じて種々の“プログラミング・パラダイム”や最新の“プログラミング・コンセプト”を学ぶ。特に、関数型プログラミング・オブジェクト指向プログラミングといったスタイルを通して、再帰・高階関数・多相型システム・抽象データ型・カプセル化・継承などの意味を知り、具体的なプログラムにおいてどのように使われるかを体験する。

2 各種事務連絡

居室 (五十嵐)	15 号館 605B
office hour	未定
e-mail	igarashi@graco.c.u-tokyo.ac.jp
レポート提出	fp-report@graco.c.u-tokyo.ac.jp
連絡用	fp-lab@graco.c.u-tokyo.ac.jp
WWW	http://www.graco.c.u-tokyo.ac.jp/~igarashi/class/syslabII01.html

3 成績評価

プログラミングの練習問題を主とするレポート (5-6 回予定) により評価する。レポートは e-mail による提出で、締め切りは基本的に出題 2 週間後の火曜日までとする。それまでにすべての問題を解き終わっていない場合でも、できた部分とできない部分の進捗状況を提出すること。単位を取るためには、

- レポートが全部出ている
- 最後の課題 (わりと大きなプログラムを書いてもらう予定) が、とてもとても素晴らしいできである

のいずれかを満たすこと。

4 授業スケジュール (予定)

日付	トピック	日付	トピック
10/10	事務連絡, 環境設定	11/28	モジュール, 抽象データ型
10/17	基本データ型, 宣言	12/5	オブジェクト指向プログラミング
10/24	関数宣言, 高階関数	12/12	ラベル
10/31	(出張のため休講予定)	12/19	GUI プログラミング
11/7	データ型, 多相型システム	1/9	(予備日)
11/14	多相型データ型	1/16	(出張のため休講予定)
11/21	例外, 配列, 参照, 入出力		

5 関数型言語 ML と Objective Caml について

プログラミング言語 ML は、元々は計算機による証明支援において、証明を記述する object language (対象言語) に対する、証明戦略の記述用の言語である、meta language (頭文字をとって ML) から発展してきた言語で、関数型プログラミングと呼ばれるプログラミングスタイルをサポートしている。ML は核となる部分が小さくシンプルであるため、プログラミング初心者向けの教育用に適した言語であると同時に、大規模なアプリケーション開発のためのサポート (モジュールシステム・ライブラリ) が充実している。ML の核言語は型付き入計算と呼ばれる、形式的な計算モデルに基づいている。このことは、言語仕様を形式的に (数学的な厳密な概念を用いて) 定義し、その性質を厳密に「証明」することを可能にしている。実際、Standard ML という標準化された言語仕様 [4] においては、(コンパイラの受理する) 正しいプログラムは決して未定義の動作をおこさない、といった性質が証明されている。

この演習で学ぶのは ML の方言である Objective Caml (OCaml) という言語である。OCaml は INRIA というフランスの国立の計算機科学の研究所でデザイン開発された言語で、Standard ML とは文法的には違った言語であるが、ほとんどの機能は共有している。また、OCaml

では Standard ML には見られない，独自の拡張が多く施されており，関数型プログラミングだけでなく，オブジェクト指向プログラミングもサポートしている．またコンパイラも効率のよいものが開発されている．

6 参考書，資料，マニュアル

OCaml のマニュアル [3] はオンライン利用が可能 (ECC 内では `file:/home/cigar/ocamlman/index.html`) である．(膨大な量であるのでむやみに印刷しないこと．) また `http://pauillac.inria.fr/caml/` から，FAQ などの文書，OCaml を使ったソフトウェアなどが利用できる．

(オブジェクト指向がサポートされていない本来の) Caml の教科書として [1] が出版されている．これが現在入手できる英語で書かれた唯一の Caml の教科書である．一方，Standard ML の教科書はそれに比べれば多数出版されている [5, 6, 7] が，OCaml とは文法を含め微妙に異なるので ML 入門者はかえって混乱するかもしれない．

[2] は，再帰/型の概念を対話形式で平易に解説している一風変わった本である．読みやすいので，この講義で興味を持ったら読んでみると面白いだろう．

参考文献

- [1] Guy Cousineau and Michel Mauny. *The Functional Approach to Programming*. Cambridge University Press, 1997. 現在，関数型プログラミングの教科書の中で Caml を直接対象にした (英語では) 唯一のもの．
- [2] Matthias Felleisen and Daniel P. Friedman. *The Little MLer*. The MIT Press, 1998. プログラミングにおける再帰・型の概念を Standard ML を使って解説．内容は OCaml にも，ほぼそのままあてはまる．
- [3] Xavier Leroy, Damien Doligez, Jacques Garrigue, Didier Rémy, and Jérôme Vouillon. *The Objective Caml system release 3.02: Documentation and user's manual*, 2001. `http://pauillac.inria.fr/caml/ocaml/htmlman/index.html`.
- [4] Robin Milner, Mads Tofte, Robert Harper, and David MacQueen. *The Definition of Standard ML (Revised)*. The MIT Press, Cambridge, MA, 1997. Standard ML の形式的定義．数学的な定義が並んでいるもので解説はないので読むのは困難．コンパイラ実装者など言語仕様を正確に知りたい人向け．
- [5] L. C. Paulson. *ML for the Working Programmer*. Cambridge University Press, second edition, 1996. Standard ML の教科書．
- [6] Jeffrey D. Ullman. *Elements of ML Programming*. Prentice Hall, ML97 edition, 1998. Standard ML の教科書．初版は和訳がある (駒場生協書籍部で目撃したことあり)．

- [7] 大堀 淳. プログラミング言語 *Standard ML*. 共立出版, 2001. 日本語で書かれた Standard ML の数少ない教科書 .

A 環境設定

環境設定は, OCaml システムのための実行パス設定, Emacs エディタでのプログラム編集/実行のための設定を行う .

実行パス設定 OCaml コンパイラなどは `~cigar/bin` にインストールしてあるので, 各自使用しているシェルに応じて設定を行うこと . `csh` または `tcsh` の場合, `~/.cshrc` もしくは `~/.tcshrc` の適当な場所 (最後でよい) に以下の行を加える .

```
set path = ($path ~cigar/bin)
```

`bash` の場合, `~/.profile` の適当な場所 (最後でよい) に以下の行を加える .

```
PATH=$PATH:~cigar/bin
export PATH
```

新しいシェル (端末ウィンドウ) を立ち上げるか, コマンドプロンプトで `source ~/.cshrc` (もしくは `~/.tcshrc`, `~/.profile`) を実行することで変更が有効になるので, `which ocaml` として, 実行パスが表示されることを確認すること .

```
cigar@as301> source ~/.cshrc
cigar@as301> which ocaml
/home/cigar/bin/ocaml
```

Emacs の設定 Emacs (Mule) 上で OCaml プログラムの編集を助けるプログラムが `~cigar/elisp` にインストールされている . 以下は `~/.emacs` に加える設定である .

```
;;; caml-mode
(setq load-path (cons "~cigar/elisp/" load-path))
(setq auto-mode-alist
  (cons '("\\.ml[iy]p?$" . caml-mode) auto-mode-alist))
(autoload 'caml-mode "caml" "Major mode for editing Caml code." t)
(autoload 'run-caml "inf-caml" "Run an inferior Caml process." t)
(if (and (string-match "^19" emacs-version) window-system)
    (require 'caml-hilit)) ;; for Mule
(if (and running-20 window-system) (require 'caml-font)) ;; for Emacs20
```

Emacs を起動し直して, `.ml` という拡張子を持つファイルを開いたときにモードラインに (`caml`) と表示されることを確認すること .

以上の設定は, 授業 WWW ページの授業スケジュール欄の環境設定 (<http://www.graco.c.u-tokyo.ac.jp/~igarashi/class/configure.txt>) よりオンラインで見えるので, カット&ペーストするとよいだろう .

B アンケート

B.1 これまでに学んだことのあるプログラミング言語と、それらをどのように学習しましたか (独学・授業で、など)

B.2 プログラミングに関連して、特に興味がある/この授業でとりあげてほしいトピックがあれば自由に回答してください

例: オブジェクト指向プログラミング, GUI プログラミング, システムソフトウェア, 計算モデル, アルゴリズムとデータ構造, コンパイラ・インタプリタ, 型システム

B.3 その他, 要望などあれば自由に書いてください

fp-lab に登録するので、名前とアドレスをお願いします。

学生証番号	名前	e-mail アドレス