

# 第1章 はじめに

**重要：**このテキストは必要に応じてアップデートされる。アップデートしたテキストは *GitHub* 本講義のリポジトリにアップロードする。

本書は工学部専門科目「プログラミング言語処理系」と「計算機科学実験及演習3（ソフトウェア）」のテキストである。プログラミング言語の設計と実装に関わるトピックをカバーしている。

本書では (1) 工学部専門科目「プログラミング言語」と (2) 「言語・オートマトン」の内容を既知とする。実装課題に取り組む場合は特に (3) git の基本的な操作法と (4) OCaml の知識とある程度の実装力が必要である。本講義でも OCaml の復習を少しやる予定であるが、あまり時間をかけることはできないので、以下の問題が解ける程度になるまで各自自習されたい。五十嵐淳による OCaml 入門テキスト（本科目のリポジトリの doc ディレクトリ内の `mltext.pdf`）が参考になるであろう。

## OCaml 力をチェックするための問題

本書中の実装課題に取り組む場合、以下の問題を解ける程度の OCaml 力が必要である。

**Exercise 1.0.1** OCaml インタプリタに以下の入力を与えたところ、

```
# let rec f x = if x = 0 then x else false;;
```

Error: This expression has type bool but an expression was expected of type int.  
という応答が返ってきた。この応答の意味するところを、エラーメッセージ中の下線部が付された `This` が何を指すかを明らかにしつつ、説明せよ。

**Exercise 1.0.2** 1. 各ノードに `int` 型の値を保持する二分木を表すユーザ定義型 `bt` を、ヴァリエント型を用いて定義せよ。

2. `bt` 型の値  $t$  を受け取り、 $t$  中に現れるすべての値の和を求める関数 `sumtree` を書け。`sumtree` の型は `bt → int` となる。
3. `bt` 型の値  $t$  と `int → int` 型の関数  $f$  とを受け取り、 $t$  中に現れるすべての値に  $f$  を適用して得られる木を求める関数 `mapTree` を書け。`mapTree` の型は `(int → int) → bt → bt` となる。

## OCaml のインストールと設定

本書の演習問題に取り組むためには、OCaml の処理系と、元になるソースコードが必要となる。<sup>1</sup>OCaml の処理系は、OCaml のパッケージマネージャである `opam` を用いて導入するのをおすすめする。テキスト執筆日現在、`opam` のインストール方法は `ocaml.org` の <http://www.ocaml.org/docs/install.html> や、`opam` のウェブサイト中の <https://opam.ocaml.org/doc/Install.html> に書いてあるので、各自読みながらインストールされたい。<sup>2</sup>インストール後に `opam init` を実行するのを忘れないこと。また、`opam` が環境変数を操作した後は、シェルを再起動するか、`opam config env` を実行して環境変数を再度シェルに読み込ませる必要がある。<sup>3</sup>

`opam` を用いて様々な OCaml のライブラリが利用できるのだが、本書の演習問題で使うのは `menhir` のみである。シェルで `opam install menhir` とタイプしてインストールしよう。<sup>4</sup>

OCaml には広く使われている統合開発環境は無いので、好みのエディタでプログラムを書いて、それをビルドして実行する、という形で開発が進むことになる。どのような言語であっても、快適に開発をするためには、キーワードをハイライトしたりインデントを自動調整したりできるようにエディタを設定することが重要である。Emacs の場合は `tuareg-mode` というモードが `opam install tuareg && opam user-setup install` で入るはずである。

また、OCaml ソースコード中の変数や式の型を簡単なショートカットで表示するための `merlin` というライブラリを入れるのがとてもおすすめである。Merlin のリポジトリは <https://github.com/ocaml/merlin> にあり、README にインストール方法が載っている。Emacs と Vim については公式でサポートされており、その他のいくつかのエディタについては README に記述がある。`opam install merlin && opam user-setup install` でインストールされるはずである。

---

<sup>1</sup>計算機科学コースの演習室の計算機環境には、すでに必要な環境設定が施されているので、この節はスキップしてもよい。

<sup>2</sup>Windows 環境で `opam` が公式にサポートされているかどうか、テキスト執筆時点ではよく分からなかった。できれば Windows 以外の環境で環境構築するのがよいであろう。

<sup>3</sup>ライブラリが見つからない系のエラーが出たら、bash ではとりあえず `eval 'opam config env'` を実行する、とおぼえておくとよい。

<sup>4</sup>他の言語処理系同様、`opam` も各自のホームディレクトリにライブラリをインストールする。`opam` はデフォルトで `~/.opam/system/lib/` 以下にライブラリをインストールする。また、`opam` には幾つかのバージョンのコンパイラをスイッチする `opam switch` というコマンドがあるのだが、これを用いてデフォルト以外のバージョンのコンパイラをインストールした場合、このパスは `~/.opam/<version of OCaml>/lib/` になる。