

# 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、ヴァリエーション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  $bt \rightarrow int$  となる。
3. `bt` 型の値  $t$  と  $int \rightarrow int$  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  $(int \rightarrow int) \rightarrow bt \rightarrow bt$  となる。

## OCaml のインストールと設定

本書の実装課題に取り組むための OCaml の設定方法については，講義ホームページに載せる予定である．