プログラミングAIII

2025年度講義資料 (1)

新潟大学工学部工学科知能情報システムプログラム

青戸等人

目次

目次

科目の概要

- オブジェクト指向型のプログラミングと並んで，関数型のプログラミングは，抽象度が高く，高信頼なプログラムを構築するアプローチとして，近年，著しく発展している．
- また，関数型のプログラミングに用いられる諸要素は，さまざまな近代的なプログラミング言語の設計に取り入れられており，これらの諸要素に親しんでおくことは，より普遍的なプログラミング能力を身に付ける上で重要である．
- このような観点から，本講義では，代表的な関数型プログラミング言語の1つであるStandard ML (SML)を用いたプログラミングを学ぶ．

科目のねらい

- ML系言語(SML，OCaml)やHaskell等の関数型言語におけるプログラミング手法，また様々な実用プログラミング言語における関数型的なプログラミング手法に通じる， モダンな関数型プログラミングの基本概念とプログラミング技術を学習する．
- プログラミングにおいて普遍的に必要となる諸態度の確認
  1. どのようなアルゴリズムを実現し，プログラムがどのように動くのかについて，明確なアイデアを持つことの必要性
  2. プログラムがどうしてエラーになるのかを論理的に考え，デバッグすることの必要性
  3. より見通しのよい，整理された，わかりやすいコードを書くことの必要性
  4. 問題をどのように分割して，求めるプログラムを実現するのか

学習の到達目標

1. SMLの構文とその意味を理解し，SMLでプログラムが書ける．
2. 計算の再帰構造を見出し，再帰を用いて関数が書ける．
3. パラメトリックな多相性を理解し，多相関数の利用・定義ができる．
4. 組，レコード，リスト，オプションなどの代表的なデータ構造を扱える．
5. データ型を定義してプログラムが書ける．
6. 分割コンパイルシステムを利用してプログラムを開発できる．
7. 高階関数を利用したプログラムが書ける．
8. 関数型プログラミングの考え方を通じて問題を分析しプログラムとして実現できる．

◎以下の学習・教育目標に該当する
[1]知識・理解(d)「コンピュータのソフトウェアに関する基礎知識を修得する」
[2]当該分野固有の能力(a)「情報の構造を設計する能力及び計算を設計し表現する能力」

履修登録の条件(注意)

- 以下のような情報分野の基本的な素養は仮定します．
  - プログラミング経験
  - エディタ(Emacs)の基本的な扱い方(Emacs以外でもよいが，プログラミング用途にも想定されているもの)
  - シェルの基本的な扱い方
  - アルゴリズムとデータ構造の知識
- 以下の講義は履修していることが望ましい．
  - 「離散数学」
  - 「数理論理学」
  - 「データ構造とアルゴリズム」

教科書について

- 教科書
  「プログラミング言語Standard ML入門改訂版」
  大堀淳著
  共立出版，2021

!!!注意!!!
使用するのは，「改訂版」です．前版は，扱っているシステムが違うので，使えません!!

この教科書は，最初から発展的な内容や難易度の高い内容も頻繁に出てきます．わからない箇所はあまり悩まず，最初は飛ばすと良いでしょう．

授業計画(第1〜4回)

1. 講義情報
   授業時間外の学修 シラバスを確認しておく．計算機室の計算機環境に慣れておく．
2. 式の評価と変数の束縛
   授業時間外の学修 教科書1.1〜1.6節を読み，SML#の対話型システムの利用法，式の評価，変数の束縛について予習する．
3. 関数の束縛と組型
   授業時間外の学修 教科書2.1節を読み，関数の束縛，組型について予習する．教科書2.3節を読み，再帰的定義について予習する．
4. 局所定義，関数を返す関数
   授業時間外の学修 教科書2.4，2.8節を読み，局所定義と変数のスコープについて予習する．教科書2.6節の前半を読み，関数を返す関数について予習する．

授業計画(第5〜8回)

5. 多相性，リスト型
   授業時間外の学修 教科書3.2節を読み，多相性について予習する．教科書6.1,6.2節を読み，リスト型について予習する．
6. パターンマッチを用いた定義とリスト処理関数
   授業時間外の学修 教科書6.3,6.4節を読み，パターンマッチを用いた定義とリスト処理関数について予習する．
7. case式とオプション型
   授業時間外の学修 教科書6.3節の前半を改めて読み，case式によるパターンマッチについて予習する．教科書7.4節を読み，オプション型について予習する．
8. データ型
   授業時間外の学修 教科書7章を読み，データ型について予習する．

授業計画(第9〜12回)

9. 関数適用の評価，副作用
   授業時間外の学修 教科書2.2，2.3節を読み，関数の評価について予習する．また，教科書8.1節を読み，評価戦略について予習する．
10. 実行形式ファイルへのコンパイル
    授業時間外の学修 教科書18章を読み，実行形式ファイルの作り方を予習する．
11. 高階関数，レコード型
    授業時間外の学修 教科書2章の2.4節以降を改めて読み，高階関数について予習する．教科書5章を読み，レコード型について予習する．
12. グラフィックスプログラミング
    授業時間外の学修 講義資料を読み，使用するライブラリについて予習する．

授業計画(第13〜16回)

13. リストの高階関数
    授業時間外の学修 教科書6.5節を中心に6章を改めて読み，リストに関する高階関数について予習する．
14. データのフォーマッティング
    授業時間外の学修 講義資料および教科書16章を読み，文字列解析について予習する．
15. システム設計・開発技法
    授業時間外の学修 講義資料を読み，関数型のシステム設計・開発技法について予習する．参考文献「SML#で始める実践MLプログラミング」の6章も読んでおくとなお良い．
16. 期末試験
    授業時間外の学修 これまでの学習内容を復習する．

参考書について

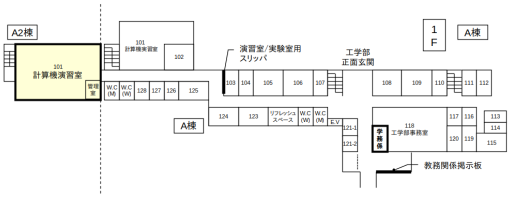
- 「SML#で始める実践MLプログラミング」
  大堀淳，上野雄大著
  共立出版
  より進んだ技術を身に付けたい方に．

- 「ML for the Working Programmer」
  L. C. Paulson 著
  Cambridge University Press, 2nd ed.
  対話的証明システム Isabelle の開発でも有名な Paulson 先生による教科書．英語で勉強したい方に．

目次

講義の情報

- 第1回〜第8回：青戸が担当します．
- 第9回〜第15回：上野が担当します．
- 講義時限：月曜3,4限(全16回)．
- 講義室：204室講義と計算機演習室(工学部A2棟1F)
- 講義室から計算機演習室へは， 1Fの工学部事務室のところを左に曲がって，ずっと奥のつきあたりまで進んでください． 途中で，靴をスリッパにはきかえて下さい．



期末試験と成績

- 期末試験
  資料，参考書等持ち込み可．
- 成績
  レポート・発表(50%)および期末試験(50%)により，学習の到達目標に達しているかに照らして評価する．

計算機演習室の利用

- 計算機演習室には，個々のPCは用意されていません．自分が所有するノートPC等を持ってきて使用することが想定されています．
- 自分のPCから，計算機演習室のサーバにリモートログインすることで，計算機演習室の環境を使って演習を行います．
  1. 計算機演習室のWiFi「ce-wifi」に接続．（学務情報システムのアカウントで接続します(2025.8より変更)．）
  2. 「x-host.ce.ie.niigata-u.ac.jp」にVNC接続をし，ログイン．
  3. ログインに成功すると，環境選択メニューが出るので，「programming-a-iii」の環境を選択．

VNC接続について

- 計算機演習室のページ（学内接続限定）：http://www.ce.ie.niigata-u.ac.jp の「VNC接続について」を参照．
- TigerVNC https://tigervnc.org を自分のPCにインストール．（ページ中程の「Downloads」のところからダウンロードページに行けます．各自の OS に合うものをダウンロードしインストールしてください．）
- TigerVNC を起動し，接続先サーバに「x-host.ce.ie.niigata-u.ac.jp」を指定して「接続(connect)」を開始．
- 計算機演習室サーバのログインページが開く．
- 使用後は必ずログアウト操作をしてから TigerVNC を終了．

SML#処理系の利用について

- programming-a-iiiの環境には，SML#の処理系を用意してあります．（端末等で，smlsharpを起動．）
- また，自分のPCなどに，SML#の処理系をインストールして，利用しても結構です．

  SML#のインストールについては，教科書18.1節もしくは下記のSML#プロジェクトのウェブページ(「ダウンロード」)を参照．

  https://smlsharp.github.io/ja/
- 計算機演習室の環境は，自宅等からもリモート接続可能です．（環境によってはうまくいかないときあり．）詳細は，計算機演習室の「演習室VPN接続で学外からログイン」参照．

計算機演習室利用のルール

- 計算機演習室のページ（学内接続限定）：http://www.ce.ie.niigata-u.ac.jp を参照
- 利用時間：平日8:00〜17:00（土日祝日は利用不可）． 授業時間中は受講者以外は利用不可．
- 土足禁止です．スリッパに履き替えてください．
- 飲食禁止です．ペットボトルを置くなどもしないでください．
- 散らかしたり汚したりしないでください．講義室とは異なり，精密機器の保全などのため，計算機演習室は頻繁に掃除が行われません．
- 関係者以外は立入禁止です．演習室利用講義の受講者以外は入室禁止です．
- 不明な点は，技術職員の宇川さんに相談してください．

講義(青戸担当分)の情報 (1)

- 講義ホームページ

  http://www.nue.ie.niigata-u.ac.jp/~aoto/lecture/ProgrammingAIII/
- スライド資料(講義ホームページに掲載)を利用．
- 講義の進め方

  3限〜4限の初め： 204室で2回分の講義内容をスライドを使って解説をします．

  説明終了後：休憩 + 各自での計算機演習室で演習およびレポート作成．計算機演習室で質問の受け付け．

  レポート：当該週の金曜23:59〆切

講義(青戸担当分)の情報 (2)

- 予習

  シラバスの 授業時間外の学修 で指定された教科書の部分を読み，予習しておくこと．
- 復習とレポート

  レポートは，テキストファイルもしくはPDFファイルで，学務情報システムから提出をしてください．
- レポート課題は，講義スライドの実習課題や実行例，教科書の対応箇所の問や実行例です． どの課題や実行例，問を取りあげるかは各自の自由です．（頑張った人の評価を高くします．）
- レポートには，どの課題なのかの情報，プログラムと最低限のコメント/実行例をつけてください．レポートとしての体裁や完成度は問いません．実質的にやったことがわかればOKです．

講義(上野担当分)の情報

- 後半開始が近くなったら，上野先生から，学務情報システムから連絡があると思いますので，アナウンスに注意しておいてください．

目次

SML♯の起動と終了(教科書1.1節，1.6節)

- 起動：ターミナル上で，smlsharp コマンドで起動する
- 終了：^D で終了(^D は Controlキーを押しながらDキーを押すことを表す).

```
1 $ smlsharp
2 SML# unknown for x86_64-pc-linux-gnu with LLVM
      13.0.1
3 # 1 + 1;
4 val it = 2 : int
5 #                    (* ^D で終了 *)
6 $
```

- $ はシェルプロンプト
- # は， SML#インタープリタのプロンプト
- (* と *) の間はコメント

インタープリタの動作

```
1 $ smlsharp
2 ...
3 # 1 + 1;
4 val it = 2
5 # 1; 1+1; 1+2;
6 val it = 1 : int
7 val it = 2 : int
8 val it = 3 : int
9 #
```

- インタプリタの動作：
  - 行末で Enterキーを押すと，その行がインタープリタに渡される.
  - インタープリタは入力を解釈して，結果を返す.
  - セミコロン(;)までが，一つの式として，解釈される.
  - プロンプトを出して次の入力を待つ.


```
1 $ smlsharp
2 ...
3 # 1 + 1
4 > ^CInterrupt    (*  ^C を入力 *)
5 # 1 + 1
6 > ;
7 val it = 2 : int
8 #
```

- インタプリタの中断：^C を入力することで，(たいがい)通常のプロンプトのところ(トップレベル)に戻れる. トップレベルで^C を入力すればSMLが中断終了する.
- emacsの shellモードでは，^C を2回入力.
- セミコロン(;)が行末にないと，式の入力途中と解釈して，プロンプトが大なり記号(>)に変わる. 大なり記号(>)のあとに，式の続きを入力すれば最初の行に続けたのと同じこと.

復習(1)：ターミナル(端末)とシェル

- ターミナル：仮想デスクトップ環境では，疑似端末.「端末」アイコン等から起動.
- ターミナル上では，シェルという対話プログラムが動いている. Linuxの標準では bash というシェルが使われている. シェルは，ユーザーが入力したコマンドを解釈して，プログラムを呼び出す.

```
1 $ ls
2 .....
3   # カレントディレクトリのファイル一覧を表示
4 $ which ls
5 /usr/bin/ls
6   # lsコマンドで呼び出しされたプログラム
7 $
```


復習(2)：emacs上でのシェル環境

SMLのインタープリンタを使う場合は， emacs 上でのシェル環境を使うのが便利.

- emacs を起動する「Emacs」アイコン等から起動もしくはターミナル上で「emacs &」を入力して起動
- 「M-x shell」で，シェル環境を開く.
- 「M-x shell」は，以下の操作を表わす： (1) Esc キーを押して(離して)から， X キーを押す. (2) emacs の下の方のコマンド入力バッファに "M-x" と表示が出るので，その後ろに shell と入力する.

sml-mode in emacs

- emacs で，拡張子が sml のファイルを読み込むと(設定されていれば) SML プログラムを書くようのモード `sml-mode` になる.
- emacs のバッファが sml プログラム用のモードになっていないときは， `M-x sml-mode` を実行するとなる.
- (自分のPCなどで) sml-mode がインストールされていないときは， (emacs 上で) `M-x package-install` から `sml-mode` と打ち込んだり， (端末上で) `sudo apt install sml-mode` としたりして，インストールする.
- TABキーで，自動的にインデントが揃う.

復習(3)：emacs 上での操作

```
C-b    左にカーソルを移動する   (backward)
C-f    右にカーソルを移動する   (forward)
C-d    カーソルのある文字を消す (delete)
C-a    行の先頭に移動する
C-e    行の末尾に移動する   (end)
C-p    上の行に移動する   (previous)
C-n    下の行に移動する   (next)
C-k    カーソルから右をカットする
C-y    カットした文字列をペーストする
```

(C-k を連続して叩くと，複数行のカット，C-yで複数行のペーストになる)

```
C-g    コマンドの中止
M-p    前のコマンドへ遡る（シェル環境のみ）
M-n    後のコマンドへ遡る（シェル環境のみ）
```


Control キーと Caps Lock キーの交換

前ページで見た通り，emacsによるファイルの編集では，Control キーを押しながら何かのキーを押す という操作が多用されます. このため，高速なファイル編集には，キーボードで，Control キーが， (左手小指ですぐ届く)アルファベットのAのキーのすぐ横にあるのは，とても重要です.
しかし，ほとんどのキーボードでは， Caps Lock キーが，Aのキーのすぐ横にあり， Control キーはAのはるか下の方に追いやられています. このため，慣れた人は，Caps Lock キーとControl キーを交換する設定をしておきます. そこで，まだ設定していない人は， Caps Lock キーとControl キーの交換の設定をしておくとよいと思います. 計算機演習室の環境では，

```
setxkbmap -option ctrl:swapcaps
```

という行を「.xsessionrc」ファイルに書いておけばよいかと思います.