

ML 演習 第 8 回

2007/07/17

飯塚 大輔, 後藤 哲志, 前田 俊行

<http://www.yl.is.s.u-tokyo.ac.jp/~sgotou/lecture/caml-enshu>

今回の内容

- 配列
- 最終課題
 - リバーシ
 - リバーシ大会

配列の作成

```
# let a1 = [|1; 2; 3|];;
val a1 : int array = [|1; 2; 3|]
# Array.make 5 0;;
- : int array = [|0; 0; 0; 0; 0|]
# Array.init 5 (fun i -> i + 1);;
- : int array = [|1; 2; 3; 4; 5|]
# let a2 = Array.make_matrix 2 3 0;;
val a2 : int array array =
  [| [|0; 0; 0|]; [|0; 0; 0|] |]
```

配列の参照・代入

```
# let a1 = [|1; 2; 3|];;  
val a1 : int array = [|1; 2; 3|]  
# a1.(0);; (* 参照 *)  
- : int = 1  
# let a2 = Array.make_matrix 2 3 0;;  
# a2.(1).(2);;  
- : int = 0  
# a2.(1).(2) <- 3;; (* 代入 *)  
- : unit = ()  
# a2.(1).(2);;  
- : int = 3
```

その他の配列操作

- モジュール Array に以下のような関数あり
 - 結合、部分列抽出、コピー
 - 新規配列を返す (いちいちメモリを確保する) ので遅い
 - 配列とリストの間の変換
 - iter, map, fold
 - などなど
- 詳しくはマニュアル参照
 - <http://caml.inria.fr/pub/docs/manual-ocaml/manual034.html> など

最終課題

- リバーシの (強い) 思考ルーチンを書け
 - 多くの研究があるので調べて下さい
 - 探索手法
 - min-max法, α - β 法, negamax 法, etc...
 - 評価関数のチューニング
 - 機械学習, etc...
 - その他
 - 定石, etc...

こちらから提供するコード

- reversi.mli : 盤面抽象データのシグネチャ
- rvclient.ml, mli : 通信クライアントモジュール
 - Reversi モジュールを使用
- rvserver.ml, mli : 審判サーバ
 - Reversi モジュールを使用
- mak.ml : サンプル思考ルーチン
 - Reversi, Rvclient を使用

課題 (必須) [締切: 8月31日]

- reversi.mli に従って
Reversiモジュールの実装 (reversi.ml)
を作れ
 - 石が置けるかどうかチェックしたり
石をひっくり返したりする関数を書いてください
 - 作らないとサンプルもサーバも動きません
- 注: 課題提出時のメールの Subject は
Report 8 710xx などとすること

課題 (必須) [締切: 9月21日]

- リバーシの思考ルーチンを作れ
 - 基本的には Reversi モジュールや Rvclient モジュールを使って実装するのが良い
 - 効率を追い求める or イバラの道を行きたい人はこれらのモジュールを使わずに実装しても良い
 - 審判との通信プロトコルは守ること
 - プロトコルは rvclient.ml のコメントを参照
 - プログラムの実行方法は Rvclient を使った場合と互換させること
 - 注: 課題提出時のメールの Subject は Report 9 710xx などとすること

リバーシ大会

- 提出された思考ルーチン同士で対戦
 - 10月上旬開催予定

大会ルール (仮)

- 1 ゲーム 4 セット
 - 1, 2 セット目: 早指し勝負
 - 持ち時間: 1 セット 30 秒 (一手 30 秒ではないので注意)
 - 1 セットごとに先手後手交代
 - 3, 4 セット目: 読み勝負
 - 持ち時間: 1 セット 300 秒 (一手 300 秒ではないので注意)
 - 1 セットごとに先手後手交代
- 勝ったセット数の多い方が勝者
 - 勝敗同数の場合は石差
 - それでも決まらないときは, そのときに考える

大会ルール (仮)

- 実行環境は csc クラスタ
 - 1 ルーチンにつき 1 台で対戦
- 外部ファイルは 10 MB まで
 - 定石テーブルなどに利用可
- その他のルールは標準的なリバーシのルールを準用
 - 盤面サイズ 8×8
 - 初期盤面も標準的な配置

大会ルール (仮)

- ソースは全て O'Caml で書くこと
 - 一部を C で書いてラッパーをかぶせて使ったりするのは禁止
- 外部ネットワークへの接続は禁止
- あまり怪しいこと
いんちきなどはしない
 - 審判サーバに対するアタック, etc...

レポート提出上の注意 (1/2)

- .zip または .tar.gz で固めたものをメールに添付して提出
 - アーカイブの内容
 - 自分の学生証番号名のディレクトリを作って
その中に以下のファイルを入れたものを圧縮すること
 - report.txt or report.pdf
 - テキストか pdf のレポート
 - Makefile
 - makeするとアカウント名の実行ファイルを生成
 - ソースファイル
 - 使用する外部ファイル

レポート提出上の注意 (2/2)

- 例えば学生証番号 01021 の人なら
 - 01021 というディレクトリを作り
 - その中に以下のファイルを入れて
 - report.txt (or report.pdf)
 - Makefile
 - ソース
 - 外部ファイル
 - その中で make を実行すると 01021 という実行ファイルができるようにしておく
 - 01021 というディレクトリごと圧縮する
 - `tar czvf report8.tar.gz 01021`