

# ML 演習 第 8 回

新井淳也、中村宇佑、前田俊行

2011/05/31

# 今回の内容

- 配列
- 最終課題
  - リバーシ
- リバーシ大会

Arrays

**配列**

# 配列

- 実は OCaml にも配列があります
- 配列の性質
  - 任意の要素に  $O(1)$  時間でアクセスできる
  - 要素を書き換えられる
  - 要素数を増やしたり減らしたりはできない

# 配列の操作

```
# let a = [|1; 2; 3|];;  
val a : int array = [|1; 2; 3|]  
# a.(1);;  
- : int = 2  
# a.(1) <- 0;;  
- : unit = ()  
# a;;  
- : int array = [|1; 0; 3|]
```

配列の生成

要素の参照

要素の書き換え

# 組み込みの Array モジュール

```
# Array.length [|1; 2; 3|];;
```

```
- : int = 3
```

```
# Array.make 5 0;;
```

```
- : int array = [|0; 0; 0; 0; 0|]
```

```
# Array.make_matrix 2 3 4;;
```

```
- : int array array = [| [|4; 4; 4|]; [|4; 4; 4|] |]
```

```
# Array.to_list [|1; 2; 3|];;
```

```
- : int list = [1; 2; 3]
```

- その他色々な関数があります
- 詳しくはマニュアルを参照

Reversi

リバーシ

# 最終課題

- リバーシの (強い) 思考ルーチンを作れ
  - いろいろなやり方があります
    - 探索手法
      - Minimax 法,  $\alpha$ - $\beta$  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 モジュールを使用

# 課題その 1 [締切: 6月21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 110xx」などとする

# 課題その 2 [締切: 8月31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
    - 実行方法は Rvclient を使った場合と互換させること
  - サンプルには勝てるようにしてほしい...
  - 注: 課題提出時のメールの Subject は「Report 9 110xx」などとする

# 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 11021 の人なら
  - 11021 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 11021 という実行ファイルができるようにしておく
  - 11021 というディレクトリごと圧縮する
    - tar czvf report9.tar.gz 11021

# 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
    - 課題締切後もコンテスト直前まで再提出可
  - 予選リーグ→決勝トーナメント  
で上位4名までを決定する予定

# 大会ルール (仮): 勝敗

- 1 試合 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)
    - 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)
    - 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): 反則

- 以下の場合そのセットは 0 対 64 で負けとなる
  - 置けないところに石を置いた
- 以下の場合それまでの結果によらずその試合は負けとなる
  - タイムアウトした
  - 外部ネットワークへ接続した
  - 審判サーバや対戦相手に対してアタックした
  - その他公正でない行為を行った場合
- ソースは全て OCaml で書くこと



# 大会ルール (仮): 実行環境

- 実行環境は csc クラスタ
  - 1 ルーチンにつき 1 台を割り当て
- 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