

Data Structure Homework #2

日本將棋對弈與棋譜瀏覽程式

Ping-Chen, Chung(鍾秉辰)

Student ID: 110503007

Department of Communication Engineering
National Central University

January 10, 2023

Contents

1	先備知識	2
1.1	棋駒	2
1.2	術語解釋	2
1.3	棋譜 / 遊戲紀錄格式	2
1.3.1	棋譜格式	2
1.3.2	KIF 記譜格式	2
1.3.3	CSA 記譜格式	3
1.3.4	SFEN 記譜格式[3]	5
1.4	初手局面	6
2	程式	7
2.1	實作功能	7
2.2	預定義常數	7
2.3	結構	7
2.4	函式	9
2.5	編譯方式	10
2.6	棋譜輸出格式	10
2.7	程式使用方式	11
2.7.1	開始新局面	11
2.7.2	單行sfen棋譜讀入	11
2.7.3	棋譜瀏覽	12

1 先備知識

由於本專案大量的使用日本將棋之日文術語用於命名變數，為求程式之可讀性，在此列出可能會使用到之術語：

1.1 棋駒

棋駒名稱	日文簡寫	羅馬拼音	羅馬簡寫
步兵	歩	fuhyou	fu
香車	香	kyousha	kyo
桂馬	桂	keima	kei
銀將	銀	ginshou	gin
金將	金	kinshou	kin
飛車	飛	hishya	hi
角行	角	kakugyo	kaku
玉將	玉	gyokushou	gyoku
と金	と	token	to
成香	杏	narikyo	narikyo
成桂	圭	narikei	narikei
成銀	全	narigin	narigin
竜王	竜	ryuuou	ryu
竜馬	馬	ryuuma	uma

Table 1: 棋駒名稱及常用羅馬拼音對照表

1.2 術語解釋

於此章節，主要介紹於此程式可能會使用到的術語。該術語將會長時間

1.3 棋譜 / 遊戲紀錄格式

In this section, we will introduce how the kifus (or game records) are stored in Japanese Shogi Association as well as Computer Shogi Association.

1.3.1 棋譜格式

1.3.2 KIF 記譜格式

KIF記譜格式為傳統記譜外最普及的將棋棋譜格式，因篇幅所限不仔細講解，並列格式如下：

術語	日文	羅馬拼音	定義
成變	なる	naru	友方棋駒到達敵陣三段目時之棋駒升級
持駒	もちこま	mochikoma	捕獲的對方持駒列表
盤面	ばんめん	bannmenn	將棋遊戲盤面
棋譜	きふ	kifu	紀錄對局紀錄之檔案
手番	てばん	teban	現在盤面輪到的對局者
先手	sente	sente	開局先走棋的對局者（奇數手）
後手	goて	gote	開局後走棋的對局者（偶數手）

Table 2: 術語名稱及常用羅馬拼音對照表

1	開始日時：2022/12/08 9:00:00
2	終了日時：2022/12/08 18:13:00
3	棋戦：棋王戦
4	場所：東京・将棋会館
5	持ち時間：4 時間
6	消費時間：128▲199△239
7	手合割：平手
8	先手：羽生善治九段
9	後手：藤井聡太竜王
10	戦型：角換わり腰掛け銀
11	手数----指手-----消費時間--
12	1 2 六歩(27) (00:00/00:00:00)
13	2 8 四歩(83) (00:00/00:00:00)
14	：
15	127 同 玉(59) (00:00/00:00:00)
16	128 5七銀打 (00:00/00:00:00)
17	129 投了 (00:00/00:00:00)
18	まで128手で後手の勝ち

1.3.3 CSA 記譜格式

根據計算機將棋協會(コンピュータ将棋協会, Computer Shogi Association)所制定之計算機通用棋譜紀錄格式[1]，在此列出CSA棋譜之記錄格式：

盤面 紀錄當時棋局的盤面。其要點如下所示：

- 各棋駒由下表賦予其英文代號：

歩	香	桂	銀	金	飛	角	玉/王
FU	KY	KE	GI	KI	HI	KA	OU
と	杏	圭	全		竜	馬	
TO	NY	NK	NG		RY	UM	

Table 3: KIF代號對照表

- 每個CSA檔案可包含一(可選的)盤面作為初始盤面。若無指定，將會預設為一般的初始盤面。
- 若要記錄持駒，可使用如 P[先後手]00[棋駒代號] 之方式。

棋譜 移動的棋譜將以以下物件生成：

- 手番：以+或是- 表示。
- 初始座標：將棋駒移動前的座標以[筋][段]之方式表示。
- 結束座標：將棋駒結束移動座標以[筋][段]之方式表示。

終局 終局時，可以 %TORYO 表記棋局結束。以下為標準的CSA 棋譜表示法：

```

1 V2.2
2 N+羽生善治九段
3 N-藤井聡太竜王
4 $EVENT:棋王戦
5 $SITE:東京・将棋会館
6 $START_TIME:2022/12/08 9:00:00
7 $END_TIME:2022/12/08 18:13:00
8 $OPENING:角換わり腰掛け銀
9 P1-KY-KE-GI-KI-OU-KI-GI-KE-KY
10 P2 * -HI * * * * * -KA *
11 P3-FU-FU-FU-FU-FU-FU-FU-FU-FU
12 P4 * * * * * * * * *
13 P5 * * * * * * * * *
14 P6 * * * * * * * * *
15 P7+FU+FU+FU+FU+FU+FU+FU+FU+FU
16 P8 * +KA * * * * * +HI *
17 P9+KY+KE+GI+KI+OU+KI+GI+KE+KY
18 +
19 +2726FU

```

```

20 -8384FU
21 +2625FU
22
23 :
24 +5958OU
25 -0057GI
26 %TORYO

```

1.3.4 SFEN 記譜格式[3]

SFEN 格式全稱為 *Shogi Forsyth-Edwards Notation*，係仿製西洋棋之 *Forsyth-Edwards Notation* [2] 所定義的盤面標示格式。該類型之棋譜可用於紀錄當前盤面，且無須紀錄該盤面自初手開始的棋局變化。所需之欄位如下所示：

盤面 紀錄當時棋局的盤面。其要點如下所示：

- 棋譜輸入自左到右，由上到下；對應於傳統將棋座標中，自 9 一開始到 1 九結束。
- 各棋駒由下表賦予其英文代號：

步	香	桂	銀	金	飛	角	玉/王
P	L	N	S	G	R	B	K

Table 4: SFEN代號對照表

- 若該棋駒為昇變棋駒，則於該代號前加上一個 + 號標記。
- 無棋駒的連續 n 個空格以數字 n 表示。
- 一行以九個棋駒為限，行與行間用 / 區隔。若令一行中有 m 個英文字元與 n 個數字，分別為 n_1, n_2, \dots ，則每一行有以下限制：

$$m + \sum n_i = 9$$

手番 記錄該盤面下一手的所屬。

持駒 記錄先後手的持駒。大寫的代號為先手持駒，小寫則為後手持駒。所有的持駒將會以無空白字串儲存。若一方持有 n 個相同持駒，則在對應的代號上加上數字 n 。

手數 紀錄當時的手數（自棋局開始後經過的棋步數量）。

1.4 初手局面

此為將棋棋局的最初局面，如下圖：該盤面可使用SFEN記譜表示為



Figure 1: 棋局初手局面

```
1 lns gkgsnl/1r5b1/ppppppppp/9/9/9/PPPPPPPPP/1B5R1/
  LNSGKGSNL b - 1
```

2 程式

於此章節，我們在此簡要敘述本程式使用到的函式及預定義的常數。

2.1 實作功能

- 棋駒移動
- 棋駒昇變
- 駒台與持駒打入
- 棋譜輸出
- 輸出一般網站可使用之棋譜 (sfen)

2.2 預定義常數

```
1 // 判斷是否為正確SFEN之正規表示式
2 #define SFEN_REGEX "[+plnsgkrbPLNSGKRB1
   -9]{1,18}\\/[+plnsgkrbPLNSGKRB1-9]{1,18}\\/[+
   plnsgkrbPLNSGKRB1-9]{1,18}\\/[+plnsgkrbPLNSGKRB1
   -9]{1,18}\\/[+plnsgkrbPLNSGKRB1-9]{1,18}\\/[+
   plnsgkrbPLNSGKRB1-9]{1,18}\\/[+plnsgkrbPLNSGKRB1
   -9]{1,18}\\/[+plnsgkrbPLNSGKRB1-9]{1,18}\\/[+
   plnsgkrbPLNSGKRB1-9]{1,18} [wb] [-
   plnsgkrbPLNSGKRB1-9]+ [1-9]+"
3
4 // 判斷是否為正確輸入之正規表示式
5 #define INPUT_REGEX "[0-9]{1,2}\\s[1-9]{1,2}"
6
7 // 重新命名(簡化)結構體
8 typedef struct Attr Attr;
9 typedef struct Komadai Komadai_t;
10 typedef struct UserInput UserInput;
11 typedef struct Location Location_t;
12 typedef struct Piece Piece_t;
13 typedef struct Board Board;
14 typedef struct SfenData SfenData_t;
```

2.3 結構

於此章節，我們定義此程式使用的結構體如下：

```

1 // 每種棋駒的各項性質，包含名稱、中文與棋譜輸出代號
2 struct Attr {
3     char sfen;
4     char *displayChar[2];
5     char *name[2];
6 };
7 // 主棋盤上的棋駒
8 struct Piece {
9     char type;
10    bool promoted;
11 };
12 // 駒台儲存該玩家持有的持駒，編號以 enum 表示
13 struct Komadai {
14     u_int8_t komaList[8];
15 };
16 // 完整遊戲所需的所有資料（主盤面、先後手駒台、手數、手番）
17 struct Board {
18     Piece_t pieces[9][9];
19     Komadai_t senteKomadai, goteKomadai;
20     int moveNumber;
21     bool turn;
22 };
23 // SFEN 棋譜處理後之資料儲存結構
24 struct SfenData {
25     char *matrix[9];
26     char turn; // turn
27     char *mochiKomaList;
28     int moveNumber;
29 };
30 // 棋駒之座標結構，可用於表示位置或向量
31 struct Location {
32     int X;
33     int Y;
34 };
35 // 使用者輸入之結構
36 struct UserInput {
37     char type[10];
38     Location_t init;
39     Location_t final;
40 };

```


2.4 函式

```
1 // 初始化本程式並讀入棋駒之各項資料
2 void initialize();
3 // 讀入SFEN棋譜
4 void readKifu(FILE *file);
5 // 將現有盤面轉為SFEN字串
6 void exportToSFEN(char *str);
7 // 將移動後的資訊輸出於螢幕上
8 void renderBoard();
9 // 瀏覽棋譜功能
10 void scrollKifu(bool function, Node *current);
11 // 處理使用者輸入
12 int userInput();
13 // 將輸入SFEN字串轉為本程式之資料結構
14 bool SFENParse(char *sfen);
15 // 將轉化後的 SFENData_t 結構化為 Board 結構
16 void SFENLoad(SfenData_t data);
17 // 將SFEN棋駒代號轉為本程式的編號 (enum PieceType)
18 int getPieceNumber(char c);
19 // 將一般座標轉化為矩陣座標
20 char coordTransfer(char axis, char input);
21 // 根據傳入之座標取得盤面上該座標之棋駒
22 Piece_t *getPieceBycoord(Location_t loc);
23 // 根據棋駒取得該棋駒之名稱
24 char *getPieceName(Piece_t piece);
25 // 移動棋駒(含昇變)
26 void makeMove(Location_t init, Location_t final,
    bool promote);
27 // 根據代號輸出該棋駒的擁有者
28 char owner(char pieceType);
29 // 確認移動是否合乎金將(或相同走法之棋駒)之移動方式
30 bool kinMove(Location_t loc, bool owner);
31 // 確認移動是否合乎各棋駒之移動方式
32 bool validMove(Location_t init, Location_t final);
33 // 確認移動是否合乎玉將之移動方式
34 bool gyokuMove(Location_t loc);
35 // 根據棋駒之羅馬拼音輸出該棋駒之編號
36 int getPieceNumByName(char *str);
```

```

37 // 根據棋駒與移動位置是否合乎金將(或相同走法之棋駒)之移動方式
38 bool kinDetection(char type, Location_t diff);
39 // 確認移動是否合乎角形之移動方式
40 bool kakuMove(Location_t diff, Location_t init);
41 // 確認移動是否合乎飛車之移動方式
42 bool hisyaMove(Location_t diff, Location_t init);
43 // 根據紀錄的鏈結串列生棋譜
44 void generateKifu(LinkedList list);
45 // 確認是否該棋駒可以進行昇變
46 bool canPromote();
47 // 悔棋
48 bool revert();
49 // 初始化盤面
50 void initializeBoard();
51 // 強制昇變
52 bool forcePromote();

```

2.5 編譯方式

請下載並開本專案資料夾，並輸入以下指令：

```

1 $ make
2 $ ./main

```

Listing 1: Compile Parameters

2.6 棋譜輸出格式

於本專案中，遊戲過程將於退出時紀錄於./sfenlist。該格式之讀法請參照subsubsection 1.3.4，以一行為單位為一手的盤面。範例輸出如下：

```

1 lnsbkgsnl/1r5b1/ppppppppp/9/9/9/PPPPPPPP/1B5R1/
   LNSBKGSNL b - 1
2 lnsbkgsnl/1r5b1/ppppppppp/9/9/2P6/PP1PPPPPP/1B5R1/
   LNSBKGSNL w - 2
3 lnsbkgsnl/1r5b1/pppppp1pp/6p2/9/2P6/PP1PPPPPP/1B5R1/
   LNSBKGSNL b - 3
4 lnsbkgsnl/1r5+B1/pppppp1pp/6p2/9/2P6/PP1PPPPPP/7R1/
   LNSBKGSNL w B 4
5 lnsbkgsnl/1r5s1/pppppp1pp/6p2/9/2P6/PP1PPPPPP/7R1/
   LNSBKGSNL b Bb 5
6 lnsbkgsnl/1r5s1/pppppp1pp/6p2/4B4/2P6/PP1PPPPPP/7R1/
   LNSBKGSNL w b 6

```

2.7 程式使用方式

2.7.1 開始新局面

請執行以下指令

```
1 $ ./main
```

Listing 2: Compile Parameters

以開新局面。進入後，系統將會提示輸入先手方之棋步，請依照以下規則輸入：[初始座標][結束座標][棋駒羅馬簡寫]，座標之(1,1)位於棋盤先手方之右上角。

範例如下：

```
1 77 76 fu
2 33 34 fu
3 88 22 kaku
4 31 22 gin
5 00 55 kaku //持駒打入，初始座標為00
```

Listing 3: Examples of Inputs

若棋駒可昇變，將會提示

```
1 <<成りますか？>>(naru, narazu)
```

請輸入naru以昇變或narazu以保持為原棋駒。

若要悔棋，可以輸入

```
1 revert
```

；若要退出，請輸入

```
1 quit
```

。退出之後，此程式會紀錄該次棋局之棋譜，存放於

```
1 ./kifu.sfenlist
```

存放，待回放使用。

2.7.2 單行sfen棋譜讀入

請執行以下指令

```
1 $ ./main --readsfen ${SFENFILE_LOCATION}
```

Listing 4: Compile Parameters

以使用sfen格式繼續中斷的棋局，並繼續遊玩。您可以使用<https://lishogi.org/analysis/> 以生成一個新盤面。

2.7.3 棋譜瀏覽

本棋譜瀏覽功能僅限讀入本程式之 `sfenlist` 檔案。請使用

```
1 $ ./main --replay ${SFENLIST_FILE_LOCATION}
```

並使用 `f`, `b` 前後瀏覽。退出時按下 `q` 即可。

References

- [1] http://www2.computer-shogi.org/protocol/record_v22.html
- [2] https://en.wikipedia.org/wiki/Forsyth%E2%80%93Edwards_Notation
- [3] https://en.wikipedia.org/wiki/Shogi_notationSFEN