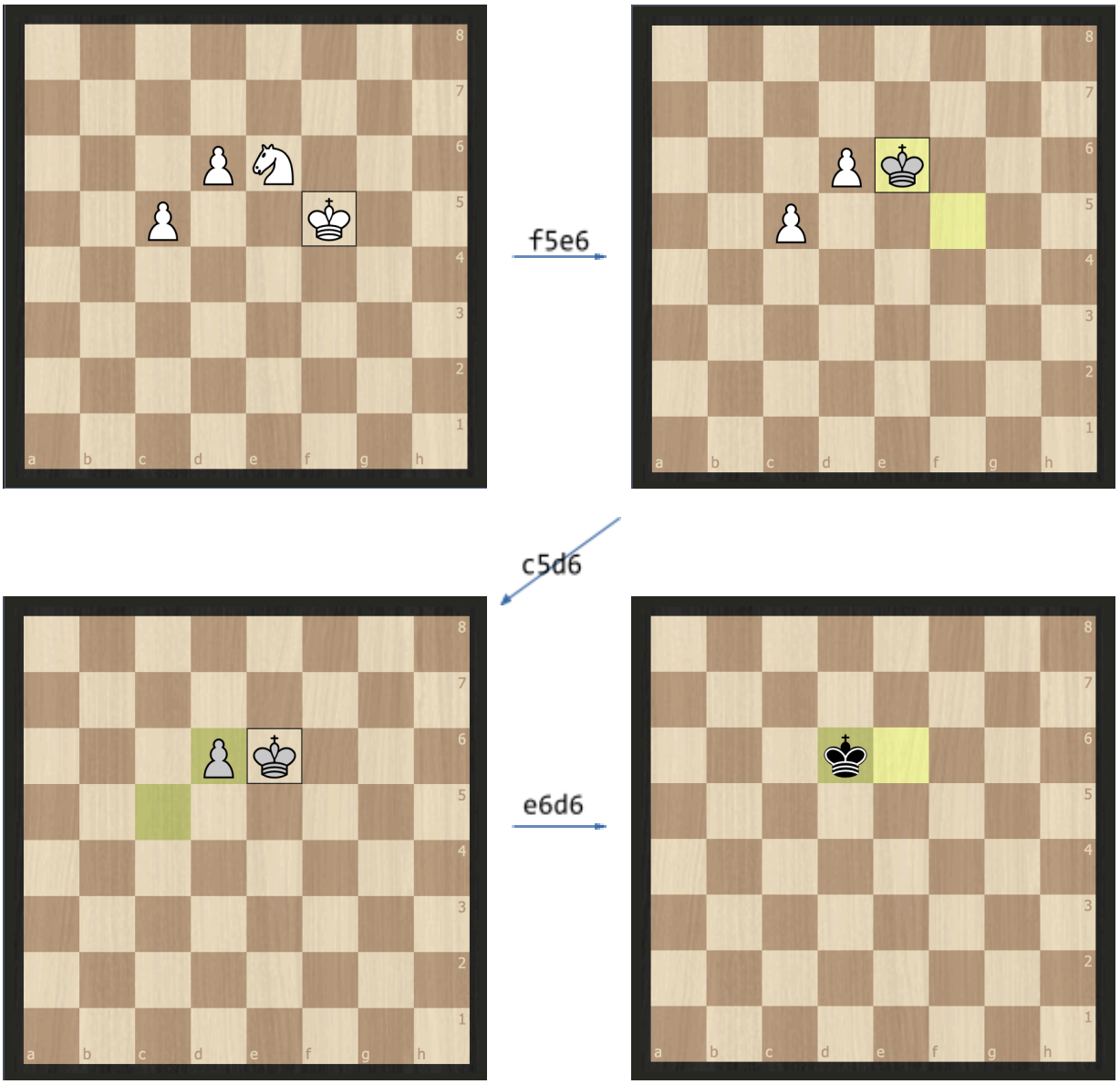
Solo Chess

“Solo Chess” es un puzzle lógico con las siguientes reglas:[1]

- Se colocan una serie de piezas blancas sobre un tablero de ajedrez de 8 x 8 casillas.[2]
- Las piezas se mueven igual que en el juego de ajedrez convencional.
- En particular, recuerde que el peón sólo puede avanzar, nunca retroceder[3].
- Sólo se pueden realizar movimientos de captura.
- Cada pieza sólo se puede mover 2 veces.
- No se puede capturar al rey.
- El objetivo es que quede sólo una pieza en el tablero.

La siguientes figuras ilustran una partida exitosa de un caso concreto de este puzzle:



[1] Existen recursos en Internet donde puede practicar este solitario. Por ejemplo, https://www.puzzle-chess.com/solo-chess-4/
(Tenga en cuenta que en ese mismo sitio hay otras variantes del juego)

[2] Se dispone de un stock suficiente de piezas blancas, de modo que no se está limitado por el número habitual del juego clásico de ajedrez.

[3] Al ser blancos, hacia arriba en las imágenes mostradas

El tablero inicial de esa partida podría representarse mediante la siguiente lista de *strings* de longitud 3:

```
["pc5"; "pd6"; "ne6"; "kf5"]
```

El primer carácter de cada *string* es el nombre de la pieza. Los nombres están inspirados en la terminología inglesa del ajedrez: para el rey **k,** de *king;* para la dama **q,** de *queen*; para el alfil **b,** de *bishop*; para el caballo **n,** de *knight*; para la torre **r,** de *rook*; y para el peón **p,** de *pawn*. El segundo y tercer caracteres corresponden a las coordenadas (columna y fila, respectivamente) de la casilla en la que está situada la pieza.

Se pide implementar una función

```
solo_chess : string list -> string list
```

que, dada una representación correcta con la posición inicial de las piezas, devuelva una lista de movimientos que conduzcan a la solución, o bien lance la excepción *Not_found* si no existe ninguna.

En el caso del tablero de la figura, esta función podría comportarse así:

```
# solo_chess ["pc5"; "pd6"; "ne6"; "kf5"];;
- : string list = ["f5e6"; "c5d6"; "e6d6"]
```

Pero también así:[4]

```
# solo_chess ["pc5"; "pd6"; "ne6"; "kf5"];;
- : string list = ["c5d6"; "f5e6"; "e6d6"]
```

Como se puede observar, cada movimiento se representa mediante un *string* de 4 caracteres, donde los dos primeros indican la casilla en la que estaba la pieza que captura y los dos últimos indican la casilla en la que estaba la pieza capturada.

Otros ejemplos de ejecución:

```
# solo_chess ["pc5"; "pd6"];;
- : string list = ["c5d6"]

# solo_chess ["kd4"; "pc4"; "bd5"];;
- : string list = ["d4c4"; "c4d5"]
```
[5]

```
# solo_chess ["pc5"; "pd6"; "ne6"; "kf5"; "ke7"];;
Exception: Not_found.

# solo_chess ["rf6"; "ne5"; "bd5"; "ke3"; "nf3"];;
```

[4] En este caso sólo varía el orden en que se realizan los movimientos; pero en otros podrán variar incluso los propios movimientos.

[5] En este caso hay otras 3 soluciones (y ninguna es una permutación de la primera).

```
- : string list = ["f6f3"; "e5f3"; "d5f3"; "e3f3"][6]

# solo_chess ["pd6"; "bc5"; "kd5"; "qe5"; "pf5"; "re4"; "nd3"; "nf3"];;
- : string list = ["c5d6"; "d6e5"; "e4e5"; "d3e5"; "f3e5"; "d5e5"; "e5f5"][7]

# solo_chess ["kg6"; "bg3"; "ra1"; "pg5"; "qa6"; "pd2"; "rf4"; "qg7";
  "rh8"; "rh6"; "re3"];;
- : string list =
["g6g7"; "a1a6"; "a6h6"; "g5h6"; "d2e3"; "e3f4"; "g3f4"; "f4h6"; "h8h6";
 "g7h6"][8]

# solo_chess ["ba3"; "pa4"; "pb2"; "pb4"; "pb5"; "rc5"; "bd3"; "pd4";
  "pd5"; "kd6"; "nf5"; "qh5"];;
- : string list =
["a3b2"; "b2d4"; "a4b5"; "b4c5"; "d3b5"; "f5d4"; "d4b5"; "h5d5"; "d5c5";
 "d6c5"; "c5b5"][9]
```

La definición de la función debe incluirse en un archivo con nombre “solo.ml”, que debe compilar correctamente (con el archivo de interfaz suministrado) con el comando

```
ocamlc -c solo.mli solo.ml
```

En esta ocasión no es necesario restringirse al paradigma funcional. Puede utilizar todos los recursos de la librería *Standard* de *OCaml*, con el fin de lograr un mayor rendimiento.

---

[6] En este caso hay otras 5 soluciones posibles; pero son todas permutaciones de la que se muestra aquí.

[7] En este caso hay otras 59 soluciones (todas ellas permutaciones de la mostrada)

[8] En este caso hay otras 7559 soluciones. Pero, del total, solo hay 5 diferentes módulo las permutaciones.

[9] En este caso hay otras 2429 soluciones (todas ellas permutaciones de la mostrada)