

TPE – “Ajedrez Lite”

Fecha de entrega – Primera parte: miércoles 14 de abril

1. INTRODUCCIÓN

La consigna consiste en especificar el juego de *Ajedrez Lite*, que se parece al Ajedrez tradicional pero con bastantes simplificaciones¹. Para evitar malentendidos, se recomienda olvidar cualquier preconceito sobre el juego real y tomar esta descripción al pie de la letra.

Tablero. El *Ajedrez Lite* se juega en un tablero de 8×8 casilleros. Las coordenadas de los casilleros son tales que la coordenada $(0, 0)$ corresponde a la casilla superior izquierda, y la coordenada $(7, 7)$ corresponde a la casilla inferior derecha:

$$\begin{array}{ccc} (0, 0) & \cdots & (0, 7) \\ \vdots & \ddots & \vdots \\ (7, 0) & \cdots & (7, 7) \end{array}$$

Sobre el tablero se distribuyen las **piezas** de dos contrincantes. Uno de los jugadores utilizará **piezas blancas** y el otro **piezas negras**. Inicialmente, las piezas del **jugador blanco** se ubican en las 2 filas inferiores del tablero, y las piezas del **jugador negro** en las dos filas superiores del tablero. Representaremos el tablero como una matriz de **casillas**. Cada casilla es un par de enteros, donde la primera componente valdrá 0 si esa celda del tablero está vacía o un número del 1 al 4 indicando el tipo de pieza (ver abajo), mientras que la segunda componente tendrá un valor 0 si la casilla es vacía, un valor 1 si la pieza es blanca y un valor 2 si la pieza es negra.

Posición. Llamamos **posición** a una determinada distribución de piezas sobre el tablero, junto con la información de qué jugador es el que debe mover a continuación.

Piezas. En el *Ajedrez Lite* se usan las siguientes piezas, representadas con números enteros, y con las siguientes reglas de movimiento y captura:

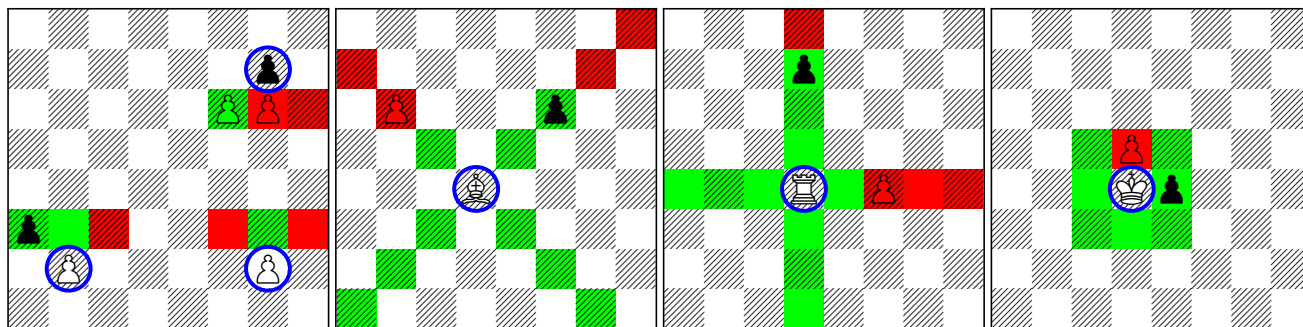




Figura 1: Ejemplos de movimientos válidos para las piezas marcadas con círculo azul para los 4 tipos de piezas del Ajedrez Lite. Las casillas verdes son movimientos/capturas legales y las casillas rojas son el resto de las casillas en el rango de movimiento/captura de la pieza indicada (que no se encuentran alcanzables para el ejemplo particular).

 **Peón** (1): Mueve solamente de a una posición hacia *adelante*² si la casilla destino está vacía. Captura piezas enemigas vecinas solamente en las dos casillas en diagonal hacia adelante (o sea, no puede capturar piezas directamente delante suyo).

 **Alfil** (2): Mueve y captura solamente en sentido diagonal, y en cualquier dirección y cantidad de casilleros sobre la diagonal elegida. Sólo puede moverse/capturar de la casilla A a la casilla B si todas las casillas intermedias se encuentran vacías, y si además la casilla B está vacía o tiene una pieza enemiga.

¹Para quienes sepan ajedrez: no existe el enroque y tampoco el movimiento “doble” del peón cuando está en su posición inicial. Además, varias reglas de juego relacionadas con empates han sido eliminadas.

²En este contexto, *adelante* significa en dirección hacia arriba para el jugador blanco y hacia abajo para el jugador negro.



Torre (3): Mueve y captura solamente en sentido horizontal/vertical, y en cualquier dirección sobre la fila/columna elegida. Sólo puede moverse/capturar de la casilla A a la casilla B si todas las casillas intermedias se encuentran vacías y si además la casilla B está vacía o tiene una pieza enemiga.



Rey (4): Sólo puede mover/capturar hacia cualquier casilla directamente adyacente (horizontal/vertical/diagonal).

Posición inicial. En el comienzo de la partida, la posición inicial, ilustrada en la Figura 2, consta de un tablero que contiene las siguientes piezas:

- Filas 0 y 7 (negras en la 0, blancas en la 7): Torre - Vacío - Alfil - Vacío - Rey - Alfil - Vacío - Torre
- Filas 1 y 6 (negras en la 1, blancas en la 6): 8 Peones

Casillas atacadas. Se dice que, en una posición determinada, una casilla (vacía o no) está siendo atacada por uno de los jugadores si dicha casilla se encuentra en el rango alcanzable de captura de alguna de sus piezas. Por ejemplo, un Peón “ataca” a las dos casillas que tiene delante suyo en diagonal (pero no ataca a la que tiene directamente delante suyo, pues no puede capturar en esa dirección), y un Rey “ataca” a cualquiera de sus casillas vecinas. Notar que, como muestra la Figura 3, una casilla atacada por un jugador puede estar tanto vacía como ocupada por una pieza tanto propia como ajena (si es propia, podemos pensar que está “defendida”, aunque la contaremos dentro del conjunto de casillas “atacadas”).
Nota: para determinar si una casilla está atacada por una pieza, no se chequea si el movimiento de captura de dicha pieza está impedido por las reglas de jaque mencionadas más adelante.

Conversiones de peones. Si al mover/capturar un peón blanco llega a la fila 0 o un peón negro llega a la fila 7, éste se convierte (inmediatamente) en una Torre del color correspondiente.

Jaque. Se dice que el rey de uno de los dos jugadores está en jaque si su casilla actual está siendo atacada por alguna pieza del contrincante.

Jaque mate. Se dice que ocurre “jaque mate” cuando el rey de uno de los jugadores está en jaque y ninguno de sus movimientos posibles (del rey u otra pieza) lo puede sacar de jaque.

Secuencia forzada. Llamamos *secuencia forzada* a una secuencia de movimientos de ambos jugadores comenzada por el jugador A en la cual todos los movimientos del jugador B están *forzados* por los del jugador A (es decir, tiene una única opción de movimiento posible), mientras que A puede tener múltiples opciones. Notar que, en este caso, los movimientos de B quedan determinados por la elección de movimientos de A.

Reglas generales de juego

- El primer movimiento de la partida es por las blancas, y se juega de manera alternada hasta que termina la partida. No es legal “pasar” el turno sin realizar un movimiento legal con alguna pieza: el jugador actual debe mover una pieza, ya sea a una posición libre o capturando una pieza enemiga.
- En su turno, un jugador no puede realizar un movimiento que lo ponga en jaque:
 - Mover/capturar con su rey a una casilla atacada por el contrincante.
 - Mover/capturar con otra pieza causando que su rey pase a ser atacado por alguna pieza del contrincante.
- En su turno, un jugador en jaque sólo puede realizar algún movimiento que lo saque de jaque:
 - Mover/capturar con su rey a una casilla no atacada por el contrincante.
 - Mover otra de sus piezas para “bloquear” el jaque.
 - Capturar (con el rey u otra pieza) la pieza que lo está poniendo en jaque (**si es una sola**).

Finalización de la partida. La partida se da por finalizada si se da cualquiera de las siguientes condiciones:

- Ocurre jaque mate (gana el contrincante)
- El jugador que debe mover no está en jaque y tampoco tiene movimientos legales para hacer (“rey ahogado”): (empate).
- Sólo quedan ambos reyes sobre el tablero (empate).

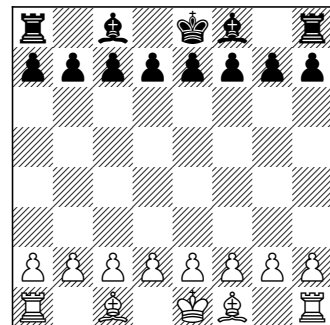


Figura 2: Posición inicial de una partida de Ajedrez Lite.

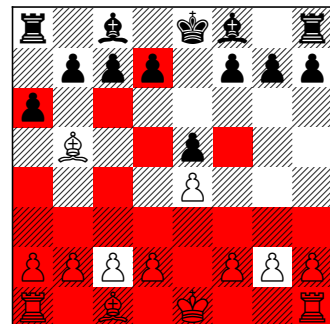
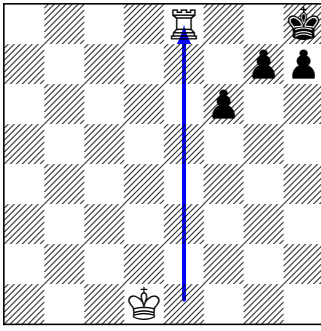
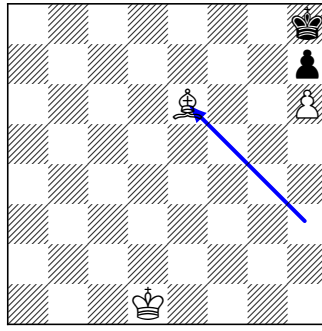


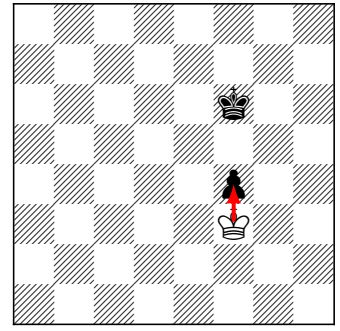
Figura 3: Los casilleros rojos son los actualmente atacados por las piezas del jugador blanco.



(a) Esta partida finalizó en jaque mate efectuado por la torre blanca.



(b) Esta partida finalizó en empate pues las negras no están en jaque pero tampoco tienen movidas posibles.



(c) En esta posición, en que mueven blancas, se finalizará en empate si el rey blanco come al peón.

Figura 4: Posibles finalizaciones de partida

2. Ejercicios – Primera parte

Especificar los siguientes predicados y procedimientos dado el **renombre** de tipos:

```
type jugador = ℤ
type casilla = ℤ × ℤ
type tablero = seq⟨seq⟨casilla⟩⟩
type posición = tablero × jugador
type coordenada = ℤ × ℤ
```

Ejercicio 1. `pred esPosiciónVálida(p: posición)`

Verdadero sii el jugador de posición es blanco o negro y la secuencia de secuencias contiene un tablero válido de ajedrez. Diremos que un tablero es válido si tiene las dimensiones correctas (8×8), las casillas están representadas de acuerdo con lo anteriormente estipulado (recordando que cada casilla puede estar vacía o contener una pieza válida de un jugador válido), y además hay exactamente un rey de cada jugador en el tablero.

Ejercicio 2. `pred esPosiciónInicial(p: posición)`

Verdadero sii la secuencia de secuencias dada contiene un tablero válido de ajedrez con las piezas correctamente ubicadas para dar comienzo a una partida, teniendo en cuenta que las blancas mueven primero.

Ejercicio 3. `pred esMovimientoVálido(p: posición, o: coordenada, d: coordenada)`

Verdadero sii dada una posición p que se asume válida, la casilla o (origen) no es vacía y la pieza de esa casilla puede mover a la casilla d (destino), teniendo **únicamente** en cuenta las reglas de movimiento de la pieza a mover. Dado que únicamente se tienen en cuenta las reglas de movimiento, vale aclarar que para este ejercicio:

- Las capturas no se consideran movimientos válidos.
- El jugador a quien le toca el turno es irrelevante para este ejercicio.
- La restricción que impide a un jugador realizar un movimiento que lo ponga en jaque no aplica en este ejercicio.

Ejercicio 4. `pred esCapturaVálida(p: posición, o: coordenada, d: coordenada)`

Verdadero sii dada una posición p que se asume válida, la casilla o (origen) no es vacía y la pieza de esa casilla puede capturar a la casilla d (destino), teniendo **únicamente** en cuenta las reglas de captura de la pieza a mover. Igual que en el ejercicio anterior, el jugador a quien le toca el turno y las restricciones relativas al jaque son irrelevantes en este ejercicio.

Ejercicio 5. `proc casillasAtacadas(in p: posición, in j: jugador, out atacadas: seq⟨coordenada⟩)`

Devuelve en `atacadas` todas las coordenadas que el jugador j está atacando actualmente (y ninguna otra).

Ejercicio 6. `proc dóndeEstaElRey(in p: posición, in j: jugador, out c: coordenada)`

Devuelve la coordenada en la que se encuentra el rey del jugador j .

Devuelve verdadero en **res** sii **p2** es la posición que resulta de realizar el movimiento/captura desde **o** (origen) a **d** (destino) en **p1**. **Nota:** se deben tener en cuenta las conversiones de peón a torre de cada jugador al llegar al extremo opuesto del tablero.

Devuelve verdadero en **res** sii el tablero correspondiente a la posición **p** está *ordenado*. Diremos que un tablero está ordenado si, en cada fila, las piezas que aparecen lo hacen en orden no decreciente (de acuerdo con el valor numérico que las identifica).

Ejercicio 9. `proc ordenarTablero(inout p: posición)` Dada una posición, la modifica de tal modo que, en cada fila, las piezas queden ordenadas por filas según su valor. Las casillas vacías del tablero deben respetarse.

Devuelve verdadero en res sii la partida terminó en jaque mate.

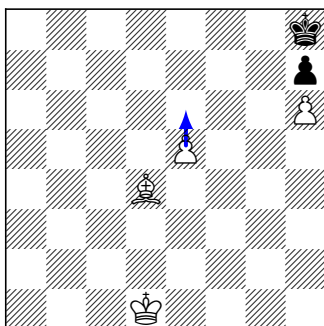
Devuelve verdadero en **res** sii la partida terminó en empate. Recordar que hay dos condiciones de empate (mate ahogado y ambos reyes solos).

Develve verdadero sii la casilla o (origen) no es vacía y la pieza de esa casilla puede mover/capturar legalmente a la casilla d (destino), teniendo en cuenta las reglas completas del juego. Recordar que, para que una jugada sea legal, es importante considerar quién es el jugador a quien le toca el turno actualmente, así como todas las restricciones relativas a movimientos que puedan poner en jaque a un jugador.

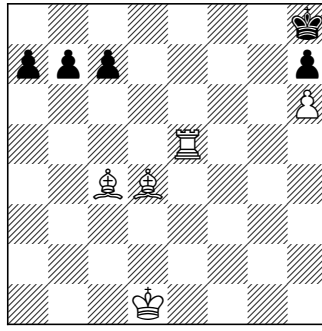
Devuelve verdadero en res sii las coordenadas de movibles corresponden a todas las piezas que el jugador actual podría mover legalmente (y ninguna otra).

Devuelve verdadero en **res** sii es posible llegar de la posición p1 a la posición p2 mediante alguna secuencia de jugadas legales.

Devuelve verdadero en **res** sii el jugador que mueve a continuación puede mover una pieza y poner en jaque al contrincante con alguna pieza diferente a la que es movida (la pieza movida podría o no dar jaque también).



Devuelve verdadero en **res** si el contrincante puede hacer jaque mate en la movida inmediatamente posterior sin importar qué movimiento haga el jugador de la movida actual.

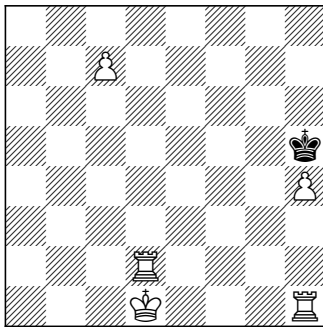


En este tablero hay mate en 1. Mueven las negras, y sólo pueden mover alguno de los 3 peones de la izquierda; y a continuación las blancas pueden realizar jaque mate moviendo la torre de (3,4) a (0,4).

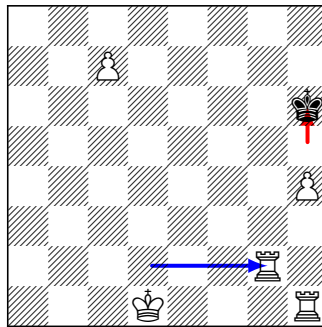
Ejercicio 17. : `proc ejecutarSecuenciaForzada(inout p: posición, in s: seq<coordenada × coordenada>)`

Dado un tablero inicial, `ejecutarSecuenciaForzada` lleva a cabo la secuencia provista de una o más movidas consecutivas que el jugador actual quiere hacer (no se informa las movidas del oponente, porque están determinadas por la secuencia forzada). Para cada movida de la secuencia provista, su oponente sólo puede tener una única respuesta legal posible, y la movida siguiente de la secuencia debe ser legal. La ejecución finaliza en la respuesta del contrincante al último movimiento informado, es decir, se ejecutan $2 \times |s|$ movimientos.

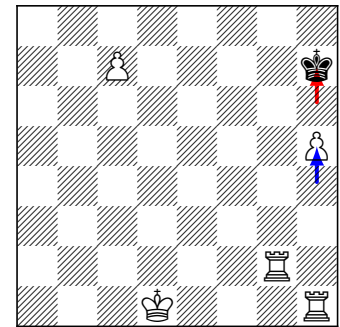
En la Figura 5 se muestra una posición inicial (jugable acá https://lichess.org/analysis/8/1P6/8/7k/7P/8/3R4/3K3R_w_-_0_1) y las posiciones visitadas al ejecutar la siguiente secuencia de movimientos del jugador blanco: $\langle ((6, 3), (6, 6)), ((4, 7), (3, 7)), ((3, 7), (2, 7)), ((1, 2), (0, 2)) \rangle$, con sus correspondientes respuestas del jugador negro.



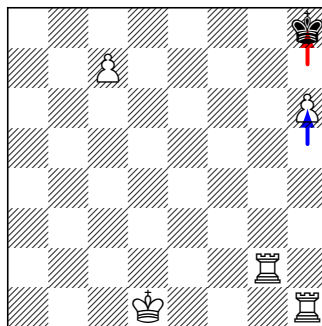
Posición inicial



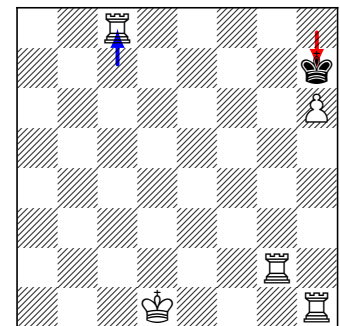
1. $((6,3),(6,6)), ((3,7),(2,7))$



2. $((4,7),(3,7)), ((2,7),(1,7))$



3. $((3,7),(2,7)), ((1,7),(0,7))$



4. $((1,2),(0,2)), ((0,7),(1,7))$

Figura 5: Ejemplo de una secuencia forzada de movimientos

Términos y condiciones

El trabajo práctico debe realizarse de manera grupal y todos los integrantes del grupo deben conocer como se resuelven los ejercicios. Para aprobar el trabajo se necesita:

- Que todos los ejercicios estén resueltos.
- Que las soluciones sean correctas.
- Que el lenguaje de especificación esté bien utilizado.

- Que las soluciones sean prolijas: evitar repetir especificaciones innecesariamente y usar adecuadamente las funciones y predicados auxiliares.
- Que no haya casos de sub-especificación ni sobre-especificación.

Pautas de Entrega

El trabajo debe ser subido al campus en la sección Trabajos Prácticos en la fecha estipulada

Importante: se admitirá un único envío por grupo, sin excepción alguna. Por favor planifiquen el trabajo para llegar a tiempo con la entrega.