



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

TPE - Ajedrez Lite

7 de mayo de 2021

Algoritmos y Estructuras de Datos I

Integrante	LU	Correo electrónico
Saravia, Ramiro Nicolás	641/19	ramirosaravia14@gmail.com
Ilundayn, Ivan Antiel	369/19	ilundayn29@gmail.com
Artigas Ciabattini, Gaston Ezequiel	57/21	gastonartcia@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Parte 1

1.1. Ejercicio 1

```

pred esPosicionValida (p:posicion) {
    esJugadorValido(p) ∧ esTableroValido((p)0)
}
pred esJugadorValido (p:posición) {
    (p)1 = 1 ∨ (p)1 = 2
}
pred esTableroVálido (t:tablero) {
    |t| = 8 ∧ (∀i : ℤ)((0 ≤ i < |t|) →L |t[i]| = 8 ∧ esCasillasCorrectas(t) ∧ haySoloDosReyes(t))
}
pred esCasillasCorrectas (t:tablero) {
    (∀i : ℤ)((0 ≤ i < |t|) →L (∀j : ℤ)(0 ≤ j < |t[i]| →L
        ((0 ≤ (t[i][j])0 ≤ 4) ∧ (0 ≤ (t[i][j])1 ≤ 2) ∧ ((t[i][j])0 = 0 ↔ (t[i][j])1 = 0)))
}
pred haySoloDosReyes (t:tablero) {
    #aparicion(t, (4, 1)) = #aparicion(t, (4, 2)) = 1
}
aux #aparicion (t:tablero,e: casilla) : ℤ = ∑i=0|t|-1 ∑j=0|t[i]|-1 if t[i][j] = e then 1 else 0 fi ;

```

1.2. Ejercicio 2

```

pred esPosiciónInicial (p:posición) {
    (p)1 = 1 ∧
    esFilas0y7((p)0) ∧
    esFilas1y6((p)0) ∧
    esFilasEntre2y5((p)0)
}
pred esFilas0y7 (t:tablero) {
    t[0] =< (3, 2), (0, 0), (2, 2), (0, 0), (4, 2), (2, 2), (0, 0), (3, 2) > ∧
    t[7] =< (3, 1), (0, 0), (2, 1), (0, 0), (4, 1), (2, 1), (0, 0), (3, 1) >
}
pred esFilas1y6 (t:tablero) {
    (∀j : ℤ)((0 ≤ j < |t[1]|) →L t[1][j] = (1, 2)) ∧
    (∀j : ℤ)((0 ≤ j < |t[6]|) →L t[6][j] = (1, 1))
}
pred esFilasEntre2y5 (t:tablero) {
    (∀i : ℤ)((2 ≤ i ≤ 5) →L (∀j : ℤ)((0 ≤ j < |t[i]|) →L t[i][j] = (0, 0)))
}

```

1.3. Ejercicio 3

```

pred esMovimientoValido (p:posición, o:coordenada, d:coordenada) {
    coordenadasvalidas(o, d, (p)0) ∧L
    (p)0[(o)0][(o)1] ≠ (0, 0) ∧
    (p)0[(d)0][(d)1] = (0, 0) ∧
    (((p)0[(o)0][(o)1])0 = 1 →L movPeonV(o, d, (p)0)) ∧
    (((p)0[(o)0][(o)1])0 = 2 →L movAlfilV(o, d, (p)0)) ∧
    (((p)0[(o)0][(o)1])0 = 3 →L movTorreV(o, d, (p)0)) ∧
    (((p)0[(o)0][(o)1])0 = 4 →L movReyV(o, d, (p)0))
}

```

```

pred coordenadasvalidas (o:coordenada, d:coordenada, t:tablero) {
   $0 \leq (o)_0 < |t| \wedge_L 0 \leq (o)_1 < |t[0]| \wedge_L$ 
   $0 \leq (d)_0 < |t| \wedge_L 0 \leq (d)_1 < |t[0]|$ 
}

pred movPeonV (o:coordenada, d:coordenada.t:tablero) {
   $((t[(o)_0][(o)_1])_1 = 1 \wedge d = ((o)_0 - 1, (o)_1) \vee$ 
   $((t[(o)_0][(o)_1])_1 = 2 \wedge d = ((o)_0 + 1, (o)_1)))$ 
}

pred movAlfilV (o:coordenada, d:coordenada.t:tablero) {
   $1 \leq (t[(o)_0][(o)_1])_1 \leq 2 \wedge$ 
   $(\exists n : \mathbb{Z})(n \geq 1 \wedge (((d)_0 = (o)_0 + n \vee (d)_0 = (d)_0 - n) \wedge$ 
   $((d)_1 = (o)_1 + n \vee (d)_1 = (o)_1 - n))) \wedge \text{vacioEntreMedioA}(o, d, t)$ 
}

pred vacioEntreMedioA (o:coordenada, d:coordenada.t:tablero) {
   $((((d)_0 < (o)_0 \wedge (d)_1 < (o)_1) \rightarrow_L (\sum_{i=1}^{(o)_0 - (d)_0 - 1} \text{if } t[(d)_0 + i][(d)_1 + i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_0 < (o)_0 \wedge (d)_1 > (o)_1) \rightarrow_L (\sum_{i=1}^{(o)_0 - (d)_0 - 1} \text{if } t[(d)_0 + i][(d)_1 - i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_0 > (o)_0 \wedge (d)_1 < (o)_1) \rightarrow_L (\sum_{i=1}^{(d)_0 - (o)_0 - 1} \text{if } t[(d)_0 - i][(d)_1 + i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_0 > (o)_0 \wedge (d)_1 > (o)_1) \rightarrow_L (\sum_{i=1}^{(d)_0 - (o)_0 - 1} \text{if } t[(d)_0 - i][(d)_1 - i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0)))$ 
}

pred movTorreV (o:coordenada, d:coordenada.t:tablero) {
   $1 \leq (t[(o)_0][(o)_1])_1 \leq 2 \wedge$ 
   $(\exists n : \mathbb{Z})(n \geq 1 \wedge (((d)_0 = (o)_0 + n \vee (d)_0 = (o)_0 - n) \wedge$ 
   $(d)_1 = (o)_1) \vee ((d)_1 = (o)_1 + n \vee (d)_1 = (o)_1 - n) \wedge (d)_0 = (o)_0)) \wedge \text{vacioEntreMedioT}(o, d, t)$ 
}

pred vacioEntreMedioT (o:coordenada, d:coordenada.t:tablero) {
   $((((d)_0 < (o)_0 \wedge (d)_1 = (o)_1) \rightarrow_L (\sum_{i=1}^{(o)_0 - (d)_0 - 1} \text{if } t[(d)_0 + i][(d)_1] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_0 > (o)_0 \wedge (d)_1 = (o)_1) \rightarrow_L (\sum_{i=1}^{(d)_0 - (o)_0 - 1} \text{if } t[(d)_0 - i][(d)_1] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_1 < (o)_1 \wedge (d)_0 = (o)_0) \rightarrow_L (\sum_{i=1}^{(d)_1 - (o)_1 - 1} \text{if } t[(d)_0][(d)_1 + i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0) \wedge$ 
   $((((d)_1 > (o)_1 \wedge (d)_0 = (o)_0) \rightarrow_L (\sum_{i=1}^{(o)_1 - (d)_1 - 1} \text{if } t[(d)_0][(d)_1 - i] = (0, 0) \text{ then } 0 \text{ else } 1 \text{ fi}) = 0))$ 
}

pred movReyV (o:coordenada, d:coordenada.t:tablero) {
   $1 \leq (t[(o)_0][(o)_1])_1 \leq 2 \wedge$ 
   $((o)_0 - 1 \leq (d)_0 \leq (o)_0 + 1 \wedge (o)_1 - 1 \leq (d)_1 \leq (o)_1 + 1)$ 
}

```

1.4. Ejercicio 4

```

pred esCapturaValida (p:posición, o:coordenada, d:coordenada) {
  coordenadasvalidas(o, d, (p)0) ∧L
  (p)0[(o)0][(o)1] ≠ (0, 0) ∧
  (p)0[(d)0][(d)1] ≠ (0, 0) ∧
  ((p)0[(o)0][(o)1])1 ≠ ((p)0[(d)0][(d)1])1 ∧
  (((p)0[(o)0][(o)1])0 = 1 →L ataquePeon(o, d, (p)0) ∧
  ((p)0[(o)0][(o)1])0 = 2 →L movAlfilV(o, d, (p)0) ∧
  ((p)0[(o)0][(o)1])0 = 3 →L movTorreV(o, d, (p)0) ∧
  ((p)0[(o)0][(o)1])0 = 4 →L movReyV(o, d, (p)0))
}

pred ataquePeon (o:coordenada, d:coordenada.t:tablero) {
  ((t[(o)0][(o)1])1 = 1 ∧ (d = ((o)0 - 1, (o)1 - 1) ∨ d = ((o)0 - 1, (o)1 + 1))) ∨
  ((t[(o)0][(o)1])1 = 2 ∧ (d = ((o)0 + 1, (o)1 - 1) ∨ d = ((o)0 + 1, (o)1 + 1)))
}

```

1.5. Ejercicio 5

```

proc casillasAtacadas (in p: posición, in j: jugador, out atacadas: seq⟨coordenada⟩) {
  Pre {esPosicionValida(p) ∧ (j = 1 ∨ j = 2)}
  Post {todasLasAtacadas(p, j, atacadas) ∧
  sinRepetidos(atacadas)}
}

pred todasLasAtacadas (p: posicion, j: jugador, atacadas: seq⟨coordenada⟩) {
  (∀d : coordenada)(0 ≤ (d)0 < |(p)0| ∧L 0 ≤ (d)1 < |(p)0[0]| ∧L
  (((p)0[(d)0][(d)1])1 ≠ j ∧ esteSiendoAtacada(d, p) ↔
  d ∈ atacadas)
}

pred esteSiendoAtacada (d: coordenada, p: posicion) {
  (∃o : coordenada)(0 ≤ (o)0 < |(p)0| ∧L 0 ≤ (o)1 < |(p)0[0]| ∧
  esCapturaValida(p, o, d))
}

pred sinRepetidos (atacadas: seq⟨coordenada⟩) {
  (∀i : ℤ)(0 ≤ i < |atacadas| →L
  ¬(∃j : ℤ)(0 ≤ j < |atacadas| ∧L i ≠ j ∧ atacadas[i] = atacadas[j]))
}

```

1.6. Ejercicio 6

```

proc dondeEstaElRey (in p: posición, in j:jugador, out c: coordenada) {
  Pre {esPosicionValida(p) ∧ (j = 1 ∨ j = 2)}
  Post {0 ≤ (c)0 < |(p)0| ∧L 0 ≤ (c)1 < |(p)0[(c)0|] ∧
  ((p)0[(c)0][(c)1] = (4, j))
}

```

1.7. Ejercicio 7

```

proc esPosicionSiguiente (in p1: posicion, in p2:posicion, in o: coordenada, in d: coordenada, out
res: Bool) {
  Pre {esPosicionValida(p1) ∧ esPosicionValida(p2) ∧ coordenadasvalidas(o, d, (p1)0)}
  Post {res = True ↔ (∃c1 : casilla)(∃c2 : casilla)(c1 ∈ (p1)0 ∧ c2 ∈ (p2)0 ∧
    ((p1)0[(o)0][(o)1] = c1 ∧
    ((p2)0[(d)0][(d)1] = c2 ∧
    tablerosIgualesMenosOyD((p1)0, (p2)0, o, d) ∧
    (esCapturaValida(p1, o, d) ∨ esMovimientoValido(p1, o, d)) ∧
    (¬esPeonTransformado(c1, d) → c1 = c2 ∧ (p2)0[(o)0][(o)1] = (0, 0) ∨
    (esPeonTransformado(c1, d) → (c2)0 = 3 ∧ (c2)1 = (c1)1 ∧ (p2)0[(o)0][(o)1] = (0, 0)))}
}

pred esPeonTransformado (c1: casilla , d: coordenada) {
  (c1)0 = 1 ∧
  (((d)0 = 0 ∧ (c1)1 = 1) ∨
  ((d)0 = 7 ∧ (c1)1 = 2))
}

pred tablerosIgualesMenosOyD (t1: tablero, t2: tablero , o:coordenada, d: coordenada) {
  (∀i :  $\mathbb{Z}$ )(∀j :  $\mathbb{Z}$ )(0 ≤ i < |t1| ∧L 0 ≤ j < |t1[i]| ∧L (i, j) ≠ o ∧ (i, j) ≠ d →L
  t1[i][j] = t2[i][j])
}

```

1.8. Ejercicio 8

```

proc estaOrdenado (in p: posición, out res: Bool) {
  Pre {esPosicionValida(p)}
  Post {res = True ↔ ((∀i :  $\mathbb{Z}$ )(0 ≤ i < |(p)0| →L
  ((∀j :  $\mathbb{Z}$ )(0 ≤ j < |((p)0)[i]| - 1 →L
  (((p)0)[i][j])0 ≤ (((p)0)[i][j+1])0 ∨ (((p)0)[i][j+1])0 = 0))))}
}

```

2. Parte 2

2.1. Ejercicio 9

```

proc ordenarTablero (inout p:posición) {
  Pre { $p = p_0 \wedge esPosicionValida(p)$ }
  Post { $|p_0| = |p| \wedge_L$ 
         $seMantienenLasCasillasVacias(p_0, (p_0)_0) \wedge$ 
         $estaOrdenado(p) \wedge$ 
         $aparicionesPorFilasIguales((p)_0, (p_0)_0)$ }
}

pred seMantienenLasCasillasVacias (t,t0:tablero) {
  ( $\forall i : \mathbb{Z})(0 \leq i < |t_0|) \wedge_L$ 
  ( $\forall j : \mathbb{Z})(0 \leq j < |t_0[i]| \wedge_L t_0[i][j] = (0,0) \leftrightarrow t[i][j] = (0,0))$ )
}

pred aparicionesPorFilasIguales (t,t0:tablero) {
  ( $\forall i : \mathbb{Z})(0 \leq i < |t_0|) \wedge_L (\forall c : casilla)(c \in t_0[i] \leftrightarrow c \in t[i])$ )
}

```

2.2. Ejercicio 10

```

proc esJaqueMate (in p: posicion, out res: Bool) {
  Pre { $esPosicionValida(p)$ }
  Post {( $esJaque(p, (p)_1) \wedge$ 
         $\neg((\exists o, d : coordenada)(\exists p2 : posicion)(esPosicionSiguiente(p, p2, o, d) \wedge \neg esJaque(p2, (p)_1)))$ )}}
}

pred esJaque (in p: posicion, in j: jugador) {
  ( $\exists o, d : coordenada)(esRey(p, d) \wedge (d)_1 = j \wedge esCapturaValida(p, o, d))$ )
}

pred esRey (in p:posicion, in d: coordenada) {
  ( $(p)_0[(d)_0][(d)_1] = (4,1) \vee (p)_0[(d)_0][(d)_1] = (4,2)$ )
}

```

2.3. Ejercicio 11

```

proc esEmpate (in p:posicion, out res: Bool) {
  Pre { $esPosicionValida(p)$ }
  Post { $reyAhogado(p) \vee soloQuedanAmbosReyes(p)$ }
}

pred soloQuedanAmbosReyes (p:posicion) {
  ( $\forall i : \mathbb{Z})(0 \leq i < |(p)_0| \longrightarrow_L$ 
  ( $\forall j : \mathbb{Z})(0 \leq j < |(p)_0[i]| \longrightarrow_L ((p)_0[i][j] = (0,0) \vee esRey(p, (i, j)))$ )
  )
}

pred reyAhogado (in p:posicion) {
   $\neg esJaque(p, (p)_1) \wedge \neg((\exists o, d : coordenada)(p2 : posicion)(esPosicionSiguienteLegal(p, p2, o, d)))$ 
}

```

```

pred esPosicionSiguienteLegal (p1: posicion, p2: posicion, o: coordenada, d: coordenada) {
  esPosicionValida(p1) ∧ esPosicionValida(p2) ∧ coordenadasvalidas(o, d, (p1)) ∧
  (∃c1 : casilla)(∃c2 : casilla)(c1 ∈ (p1)0 ∧ c2 ∈ (p2)0 ∧
  ((p1)0)[(o)0][(o)1] = c1 ∧
  ((p2)0)[(d)0][(d)1] = c2 ∧
  tablerosIgualesMenosOyD((p1)0, (p2)0, o, d) ∧
  esJugadaLegal(p1, o, d) ∧
  (¬esPeonTransformado(c1, d) → c1 = c2 ∧ (p2)0[(o)0][(o)1] = (0, 0) ∨
  (esPeonTransformado(c1, d) → (c2)0 = 3 ∧ (c2)1 = (c1)1 ∧ (p2)0[(o)0][(o)1] = (0, 0))
}

```

2.4. Ejercicio 12

```

pred esJugadaLegal (p: posición, o: coordenada, d: coordenada) {
  (esMovimientoValido(p, o, d) ∨ esCapturaValida(p, o, d)) ∧
  respetaLasReglas(p, o, d)
}

pred respetaLasReglas (p: posicion, o: coordenada, d: coordenada) {
  empiezanBlancasYTurnosAlternados(p, o, d) ∧
  autoJaqueInvalido(p, o, d) ∧
  jugadaLegalParaSalirDeJaque(p, o, d)
}

pred empiezanBlancasYTurnosAlternados (p: posicion, o: coordenada, d: coordenada) {
  (esPosicionInicial(p) → (p)1 = 1) ∨ ¬((∃p1 : posicion)(esPosicionSiguiente(p, p1, o, d) ∧
  (p = p1 ∨ (p)1 = (p1)1)))
}

pred autoJaqueInvalido (p: posicion, o: coordenada, d: coordenada) {
  ¬(∃p1 : posicion)(esPosicionSiguiente(p, p1, o, d) ∧ esJaque(p1, (p)1))
}

pred jugadaLegalParaSalirDeJaque (p: posicion, o: coordenada, d: coordenada) {
  esJaque(p, (p)1) ∧ (∃p1 : posicion)(esPosicionSiguiente(p, p1, o, d) ∧ ¬esJaque(p1, (p)1))
}

```

2.5. Ejercicio 13

```

proc piezasMovibles (in p: posicion, in movibles: seq<coordenada>, out res: Bool) {
  Pre {PosicionValida(p)}
  Post {res = True ↔ ((∀o, d : coordenada)(esJugadaLegal(p, o, d) ↔ o ∈ movibles)) ∧
  SinRepetidos(movibles) ∧
  (∀i : ℤ)(0 ≤ i < |movibles| →L (t[(i)0][(i)1])1 = (p)1)}
}

```

2.6. Ejercicio 14

```

proc esPosicionFutura (in p1: posicion, in p2: posicion, out res: Bool) {
  Pre {esPosicionValida(p)}
  Post {res = True ↔ secuenciaJugadasLegales(p1, p2)}
}

pred secuenciaJugadasLegales (p1: posicion, p2: posicion) {
  (∃s : seq<posicion>)(|s| > 1 ∧L s[0] = p1 ∧ s[|s| - 1] = p2 ∧
  (∀i : ℤ)(0 ≤ i < |s| - 1 →L esPosicionSiguienteLegal(s[i], s[i + 1])))
}

```

2.7. Ejercicio 15

```

proc hayJaqueDescubierto (in p: posicion, out res: Bool) {
  Pre {esPosicionValida(p)}
  Post {(∃p2 : posicion)(∃o, d : coordenada)
    (esPosicionSiguienteLegal(p, p2, o, d) ∧
    esJaque(p2, (p2)1))}
}

```

2.8. Ejercicio 16

```

proc hayMateEn1 (in p: posicion, out res: Bool) {
  Pre {esPosicionValida(p)}
  Post {res = True ↔
    (∃o, o2, d, d2 : coordenada)(∃p2, p3 : posicion)(esPosicionSiguienteLegal(p, p2, o, d) ∧
    esPosicionSiguienteLegal(p2, p3, o2, d2) ∧
    esJaqueMate(p3))}
}

```

2.9. Ejercicio 17

```

proc ejecutarSecuenciaForzada (inout p:posicion, in s:seq(cordenada × cordenada)) {
  Pre {esPosicionValida(p) ∧ p = p0}
  Post {(∃r : seq(posicion))(|r| = |s| * 2 + 1 ∧L p0 = r[0] ∧L p = r[|r| - 2] ∧
    ((∀i : ℤ)(0 ≤ i < |s| →L
    (∃j : ℤ)(0 ≤ j < |r| - 1 ∧L esPosicionSiguienteLegal(r[j], r[j + 1], s[(i)]0, s[(i)]1)) ∧
    (∀j : ℤ)(0 ≤ j < |r| - 1 ∧L (j)1 ≠ (p0)1 →L
    (∃o, d : coordenada)(esPosicionSiguienteLegal(j, j + 1, o, d) ∧
    ¬(∃o1, d1)(o ≠ o1 ∨ d ≠ d1 ∧ esPosicionSiguienteLegal(r[j], r[j + 1], o1, d1)))}
}

```