# TP Algoritmo y Estructura de Datos I

Ignacio F. Oromendia, Boris Batlle, Lucas Merzbacher

Abril 2021

## Ejercicio 1

aux ReyBlanco : $\mathbb{Z}$x $\mathbb{Z}$ $= (4, 1)$;
aux ReyNegro : $\mathbb{Z}$x $\mathbb{Z}$ $= (4, 2)$;
pred esPosicionValida (p: posicion) {
$(\forall i : \mathbb{Z})((0 \leq i < 8) \longrightarrow_L (|p_0|[i] = 8)) \wedge$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \leq i, j < 8) \longrightarrow_L (|p_0|[i][j] = 8)) \wedge$
$(\exists i : \mathbb{Z})(\exists j : \mathbb{Z})((0 \leq i, j < 8) \wedge_L (p_0[i][j] = ReyBlanco)) \wedge$
$(\exists n : \mathbb{Z})(\exists k : \mathbb{Z})((0 \leq n, k < 8) \wedge_L (p_0[n][k] = ReyNegro))$
}

## Ejercicio 2

aux casilleroVacio : $\mathbb{Z}$x $\mathbb{Z}$ $= (0, 0)$;
aux jugadorBlanco : $\mathbb{Z}$ $= 1$;
aux jugadorNegro : $\mathbb{Z}$ $= 2$;
aux AlfilBlanco : $\mathbb{Z}$x $\mathbb{Z}$ $= (2, jugadorBlanco)$;
aux AlfilNegro : $\mathbb{Z}$x $\mathbb{Z}$ $= (2, jugadorNegro)$;
aux TorreBlanca : $\mathbb{Z}$x $\mathbb{Z}$ $= (3, jugadorBlanco)$;
aux TorreNegra : $\mathbb{Z}$x $\mathbb{Z}$ $= (3, jugadorNegro)$;
aux PeonBlanco : $\mathbb{Z}$x $\mathbb{Z}$ $= (1, jugadorBlanco)$;
aux PeonNegro : $\mathbb{Z}$x $\mathbb{Z}$ $= (1, jugadorNegro)$;
pred esPosicionInical (p: posicion) {
$(p_0[0][1] = p_0[0][3] = p_0[0][3] = p_0[0][6] = p_0[7][1]) = p_0[7][3] = p_0[7][6] = casilleroVacio)) \wedge$
$(\forall i : \mathbb{Z})((0 \leq i < 8) \longrightarrow_L (p_0[2][i] = p_0[3][i] = p_0[4][i] = p_0[5][i] = casilleroVacio)) \wedge$
$(\mathrm{p}_0[0][0] = p_0[0][7] = TorreNegra) \wedge$
$(\mathrm{p}_0[0][2] = p_0[0][5] = AlfilNegro) \wedge$
$(\mathrm{p}_0[0][4] = ReyNegro) \wedge$
$(\forall i : \mathbb{Z})((0 \leq i < 8) \longrightarrow_L (p_0[1][i] = PeonNegro)) \wedge$
$(\mathrm{p}_0[7][0] = p_0[7][7] = TorreBlanca) \wedge$
$(\mathrm{p}_0[7][2] = p_0[7][5] = AlfilBlanco) \wedge$
$(\mathrm{p}_0[7][4] = ReyBlanco) \wedge$
$(\forall i : \mathbb{Z})((0 \leq i < 8) \longrightarrow_L (p_0[6][i] = PeonBlanco))$

}

## Ejercicio 3

pred casillaVacia (c: coordenada) {
$c = (0, 0)$

}
pred esPeon (c: coordenada) {
$c_0 = 1$
}
pred esAlfil (c: coordenada) {
$c_0 = 2$
}
pred esTorre (c: coordenada) {
$c_0 = 3$
}
pred esRey (c: coordenada) {
$c_0 = 4$
}
pred caminoHorizontalVacio (t: tablero, o : coordenada, d: coordenada ) {
$((o_0 \neq d_0 \wedge o_1 = d_1) \wedge_L$
$((\mathrm{d}_0 < o_0) \wedge_L (\forall i : \mathbb{Z})((d_0 < i < o_0) \longrightarrow_L casillaVacia(t[i][o_1]))) \vee$
$((\mathrm{d}_0 > o_0) \wedge_L (\forall i : \mathbb{Z})((o_0 < i < d_0) \longrightarrow_L casillaVacia(t[i][o_1])))\vee$
}
pred caminoVerticalVacio (t: tablero, o : coordenada, d: coordenada ) {
$((o_0 = d_0 \wedge o_1 \neq d_1) \wedge_L$
$((\mathrm{o}_1 > d_1) \wedge_L (\forall i : \mathbb{Z})((d_1 < i < o_1) \longrightarrow_L casillaVacia(t[o_0][i]))) \vee$
$((\mathrm{d}_1 > o_1) \wedge_L (\forall i : \mathbb{Z})((o_1 < i < d_1) \longrightarrow_L casillaVacia(t[d_0][i])))\vee$
}
pred caminoDiagonalVacio (t: tablero, o : coordenada, d: coordenada ) {
$((o_0 \neq d_0 \wedge o_1 \neq d_1) \wedge_L$
$((\mathrm{o}_0 < d_0 \wedge o_1 < d_1) \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(((o_0 < i < d_0) \wedge (o_1 < j < d_1)) \longrightarrow_L casillaVacia(t[i][j])))$
$((\mathrm{o}_0 > d_0 \wedge o_1 > d_1) \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(((d_0 < i < o_0) \wedge (d_1 < j < o_1)) \longrightarrow_L casillaVacia(t[i][j])))$
$((\mathrm{o}_0 < d_0 \wedge o_1 > d_1) \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(((o_0 < i < d_0) \wedge (d_1 < j < o_1)) \longrightarrow_L casillaVacia(t[i][j])))$
$((\mathrm{o}_0 > d_0 \wedge o_1 < d_1) \wedge_L (\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(((d_0 < i < o_0) \wedge (o_1 < j < d_1)) \longrightarrow_L casillaVacia(t[i][j])))$
}
pred movimientoAlfil (t: tablero, o : coordenada, d: coordenada ) {
$caminoDiagonalVacio(t, o, d) \wedge$
$(\mathrm{o}_0 < d_0 \wedge o_1 < d_1) \wedge (\exists n : \mathbb{Z})((0 < n < (8 - o_0))) \wedge_L ((d_0 = o_0 + n) \wedge (d_1 = o_1 + n)) \vee$
$(\mathrm{o}_0 > d_0 \wedge o_1 > d_1) \wedge (\exists n : \mathbb{Z})((0 < n < (8 - o_0))) \wedge_L ((d_0 = o_0 - n) \wedge (d_1 = o_1 - n)) \vee$
$(\mathrm{o}_0 > d_0 \wedge o_1 < d_1) \wedge (\exists n : \mathbb{Z})((0 < n < (8 - o_0))) \wedge_L ((d_0 = o_0 - n) \wedge (d_1 = o_1 + n)) \vee$
$(\mathrm{o}_0 < d_0 \wedge o_1 > d_1) \wedge (\exists n : \mathbb{Z})((0 < n < (8 - o_0))) \wedge_L ((d_0 = o_0 + n) \wedge (d_1 = o_1 - n))$
}
pred mueveElAlfil (t: tablero, o : coordenada, d: coordenada ) {
$(esAlfil(t[o_0][o_1]) \wedge movimientoAlfil(t, o, d))$
}
pred mueveElPeon (j: jugador, o: coordenada, d: coordenada) {
$(esPeon(c)) \wedge$
$(\mathrm{j}_1 = 1) \wedge (d = (o - (1, 0))) \vee$
$(\mathrm{j}_1 = 2) \wedge (d = (o + (1, 0)))$
}
pred mueveLaTorre (t: tablero, o: coordenada, d: coordenada) {
$(esTorre(t[o_0][o_1])) \wedge$
$(\mathrm{o}_0 \neq d_0 \wedge o_1 = d_1) \wedge (caminoHorizontalVacio(t, o, d)) \vee$
$(\mathrm{o}_0 = d_0 \wedge o_1 \neq d_1) \wedge (caminoVerticalVacio(t, o, d))$
}
pred movimientoRey ((o: ($\mathbb{Z}$x $\mathbb{Z}$), d:($\mathbb{Z}$x $\mathbb{Z}$))) {

$|o_0 - d_0| \leq 1 \land |o_1 - d_1| \leq 1$
}

pred esMovimientoValido (p: posicion, o: coordenada, d: coordenada) {
$\neg(casillaVacia(p_0[o_0][o_1]) \land \neg(casillaVacia(p_0[d_0][d_1])) \land (0 \leq d_0, d_1, o_0, o_1 < 8)) \land$
(mueveElPeon($p_0[o_0][o_1], o, d$) $\lor$ $mueveElAlfil(p_0[o_0][o_1], o, d) \lor$
mueveLaTorre($p_0[o_0][o_1], o, d$) $\lor$ $mueveElRey(p_0[o_0][o_1], o, d)$)
}

## Ejercicio 4

pred esCapturaValida (p: posicion, o: coordenada, d: coordenada) {
$(\neg casillaVacia(p_0[o_0][o_1]) \land \neg casillaVacia(p_0[d_0][d_1])) \land$
(esPeon($p_0[o_0][o_1]$) $\land (p_1 = 1)) \land$
$((d_1 > o_1) \land (d = (o + (-1, 1))) \lor (d_1 < o_1) \land (d = (o + (-1, -1))) \lor$
$(p_1 = 2) \land$
$(d_1 > o_1) \land (d = (o + (1, 1))) \lor (d_1 < o_1) \land (d = (o + (1, -1)))) \lor$
(mueveElAlfil($p_0[o_0][o_1], o, d$)) $\lor$ $(mueveLaTorre(p_0[o_0][o_1], o, d)) \lor (mueveElRey(p_0[o_0][o_1], o, d))$
}

## Ejercicio 5

pred hayPeonQueAtaca (p: posicion, j: jugador, c: coordenada) {
$((j = 1) \land (p_0[c_0][c_1]_1 = 2) \land$
$((c_0 < 7 \land c_1 > 0) \land_L (p_0[c_0 + 1][c_1 - 1] = (1, 1))) \lor ((c_0 < 7 \land c_1 < 7) \land_L (p_0[c_0 + 1][c_1 + 1] = (1, 1)))) \lor$
$((j=2) \land (p_0[c_0][c_1]_1 = 1) \land$
$((c_0 > 0 \land c_1 > 0) \land_L (p_0[c_0 - 1][c_1 - 1] = (1, 2))) \lor ((c_0 > 0 \land c_1 < 7) \land_L (p_0[c_0 - 1][c_1 + 1] = (1, 1))))$
}

pred hayReyQueAtaca (p: posicion, j: jugador, c: coordenada) {
$(\exists cOrigen : coordenada)((cOrigen = ReyBlanco \lor cOrigen = ReyNegro) \land movimientoRey(cOrgien, c))$
}

pred hayUnAlfilQueAtaca (p: posicion, j: jugador, c: coordenada) {
$(\exists i : \mathbb{Z})(\exists k : \mathbb{Z})((0 \leq i < 8) \land (0 \leq k < 8)) \land_L$
$p_0[c_0][c_1]_1 \neq j \land movimientoAlfil(p_0, c, (i, k))$

}

pred perteneceASecuencia (c: casilla, s: $seq\langle casilla \rangle$) {
$(\exists i : \mathbb{Z})(0 \leq i < 8 \land_L s[i] = casilla)$
}

pred hayUnaTorreQueAtaca (p: posicion, j: jugador, c: coordenada) {
$(\forall i : \mathbb{Z})((c_0 < i < 8)) \longrightarrow_L (p_0[c_0][c_1]_1 \neq j) \land_L$
$((p_0[i][c_1] = (3, j) \land caminoVacio(t, o, (i, c_1))) \lor$
$(p_0[c_0][i] = (3, j) \land caminoVacio(t, o, (c_0, i))))$

}

pred esAtacada (p: posicion, j: jugador, c: coordenada) {
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(((0 \leq i < 8) \land (0 \leq j < 8)) \longrightarrow_L$
hayUnAlfilQueAtaca (p,j,c)$\lor$
hayUnaTorreQueAtaca(p,j,c)$\lor$
hayPeonQueAtaca(p,j,c)$\lor$
hayReyQueAtaca(p,j,c))
}

proc casillasAtacadas (in p:posicion,in j: jugador, out atacadas: seq(coordenadas)) {
Pre $\{esPosicionValida(p) \wedge (j = 1) \vee (j = 2)\}$
Post $\{(\forall i : \mathbb{Z})((0 \leq i < |atacadas|) \longrightarrow_L esAtacada(p, j, atacadas[i])) \wedge$
$(\neg\exists cAtacada : casilla)(esAtacada(p, j, cAtacada) \wedge perteneceASecuencia(cAtacada, atacadas))\}$
}

## Ejercicio 6

proc dondeEstaElRey (in p: posicion, in j: jugador, out c: coordenada) {
Pre $\{esPosicionValida(p) \wedge (j = 1 \vee j = 2)\}$
Post $\{(\exists i : \mathbb{Z})(\exists k : \mathbb{Z})(0 \leq i < 8)(0 \leq k < 8) \wedge_L (p_0[i][k] = (4, j) \wedge c = (i, k))\}$
}

## Ejercicio 7

pred peonATorre (p1: posicion, p2: posicion, d: coordenada) {
$(\forall i : \mathbb{Z})((0 \leq i < 8) \longrightarrow_L (d = (0, i) \vee d = (7, i))) \wedge$
$(\text{p1}_0[o_0][o_1] = (1, p_1) \wedge p2_0[d_0][d_1] = (3, p_1))$
}
pred seMovio (p1: posicion, p2: posicion, d: coordenada, o: coordenada) {
$esPeon(o) \wedge peonATorre(p1, p2, d) \vee (p1[d_0][d_{d1}] = p2[o_0[o_1]])$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \leq i, j < 8) \longrightarrow_L ((i, j) = o \vee (i, j) = d) \vee (p_1[i][j] = p_2[i][j]))$
}
proc esPosicionSiguiente (in p1: posicion, in p2: posicion,in o: coordenada,in d:coordenada, out result: Bool) {
Pre $\{esPosicionValida(p : p1) \wedge esPosicionValida(p : p2)\}$
Post $\{res = True \iff casillaVacia(o) \wedge esMovimientoValido(p_1, o, d) \wedge seMovio(p_1, p_2, o, d)) \vee (esCapturaValida($
$seMovio(p_1, p_2, o, d))\}$
}

## Ejercicio 8

pred decrece (p: posicion) {
$(\forall k : \mathbb{Z})(0 \leq k < 8) \longrightarrow_L (\exists i : \mathbb{Z})(\exists j : \mathbb{Z})(0 \leq i < 8) \wedge (0 \leq j < i) \wedge \neg casillaVacia(p_0[k][j]) \wedge \neg casillaVacia(p_0[k][i]) \wedge_L$
$\neg(p_0[k][j]_0 > p_0[k][i]_0)$
}
proc estaOrdenado (in p: posicion, out result: Bool) {
Pre $\{esPosicionValida(p : p)\}$
Post $\{result = decrece(p)\}$
}

## Ejercicio 9

aux cantApariciones (p:posicion, fila: $\mathbb{Z}$, casilla: casilla) : $\mathbb{Z} = \sum_{i=0}^{7}$ if $p[fila][i] = casilla$ then 1 else 0 fi ;
pred esTableroOrdenado (p1:posicion, p2:posicion) {
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(\exists k : \mathbb{Z})((0 \leq i, j, k < 8) \longrightarrow_L (p1[i][j] = p2[i][k]) \wedge cantApariciones(p1, i, p1[i][j]) =$
$cantApariciones(p2, i, p2[i][j]) \wedge decreceFila(p2, i))$

}

proc ordenarTablero (inout p:posicion) {
Pre $\{p = P_0 \wedge esPosicionValida(P_0)\}$
Post $\{esTableroOrdenado(p, P_0 \wedge p = P_0)\}$
}

## Ejercicio 10

aux jugadorOponente (j:jugador) : $\mathbb{Z}$ = if $j = 1$ then 2 else 1 fi;
pred posicionDelRey (p:posicion, j:jugador, c:coordenada ) {
$(\exists i : \mathbb{Z})(\exists k : \mathbb{Z})((0 \leq i < 8) \wedge (0 \leq k < 8)) \longrightarrow_L (p_0[i][k] = (4, j) \wedge c = (i, k))$
}
pred esJaque (p:posicion, j:jugador) {
$(\exists c : coordenada)(posicionDelRey(p, j, c) \wedge esAtacada(p, jugadorOponente(j), c))$
}
pred capturaValidaEvitandoJaque (p:posicion, o:coordenada, d:coordenada) {
$(\exists pos : posicion)(seMovio(p, pos, o, d) \wedge \neg(esJaque(pos, j))) \wedge capturaValida(p, o, d)$
}
pred movimientoValidoEvitandoJaque (p:posicion, o:coordenada, d:coordenada) {
$(\exists pos : posicion)(seMovio(p, pos, o, d) \wedge \neg(esJaque(pos, j))) \wedge movimientoValido(p, o, d)$
}
pred hayJugadaValida (p:posicion) {
$(\exists cOrigen : coordenada)(\exists cDestino : coordenada)(movimientoValidoEvitandoJaque(p, cOrigen, cDestino) \vee$
$capturaValidaEvitandoJaque(p, cOrigen, cDestino))$
}

proc esJaqueMate (in p:posicion, out res: Bool) {
Pre $\{esPosicionValida(p)\}$
Post $\{res = True \iff hayJugadaValida(p) \wedge esJaque(p, p1)\}$
}

## Ejercicio 11

pred soloHayDosReyes (p:posicion) {
$(\exists c_1 : coordenada)(\exists c_2 : coordenada)((posicionDelRey(p, 1, c)) \wedge (posicionDelRey(p, 2, c))) \wedge$
$(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})((0 \leq i, j < 8) \longrightarrow_L ((i, j) = c_0 \vee (i, j) = c_1) \vee casillaVacia(p_0[i][j]))$
}
pred mateAhogado (p: posicion) {
$(\exists cO : coordenada)(\exists cD : coordenada)((0 \leq cO_0, cD_0 < 8) \wedge (0 \leq cO_1, cD_1 < 8)) \wedge_L$
$\neg esMovimientoValido(p, cO, cD) \wedge \neg esCapturaValido(p, cO, cD) \wedge \neg esJaque(p, p_1)$
}

proc esEmpate (in p:posicion, out result: Bool) {
Pre $\{esPosicionValida(p)\}$
Post $\{result =$ if $mateAhogado(p) \vee soloHayDosReyes(p)$ then $True$ else $False$ fi$\}$
}

## Ejercicio 12

pred esJugadaLegal (p: posicion, o: coordenada, d: coordenada) {
$movimentoValidoEvitandoJque(p, o, d) \vee capturaValidaEvitandoJaque(p, o, d)$

}

## Ejercicio 13

pred estaEnSecuencia (movibles: $seq\langle coordenada\rangle$, c: coordenada) {
$(\exists i : \mathbb{Z})((0 \leq i < 8) \wedge_L (movibles[i] = c))$
}
pred esMovimietnoYNoEstaEnSecuencia (p: posicion, movibles: $seq\langle coordenada\rangle$, o: coordenada) {
$(\exists cDestino : coordenada)(((0 \leq cDestino_0 < 8) \wedge (0 \leq cDestino_1 < 8)) \wedge_L$
(esJugadaLegal(p,o,cDestino) $\wedge \neg estaEnSecuencia(movibles, cDestino)$))
}

proc piezasMovibles (in p: posicion, in movibles: $seq\langle coordenada\rangle$, out res: Bool) {
Pre $\{esPosicionValida(p)\}$
Post $\{res = true \iff (\forall i : \mathbb{Z})(\forall cDestino : coordenado)((0 \leq i < |movibles|) \longrightarrow_L \neg esMovimientoYNoestaEnsec$
}

## Ejercicio 14

pred posicionesFuturas (p1: posicion, p2: posicion) {
$(\forall i : \mathbb{Z})((0 \leq i < |sPos| - 1) \longrightarrow_L (\exists sPos : seq\langle posicion\rangle)(\exists cOrigen : coordenada)(\exists cDestino : coordenada)$
(sPos[0] = p1 $\wedge sPos[|sPos| - 1] = p2) \wedge$
(esPosicionValida(sPos[i]) $\wedge esPosicionValida(sPos[i + 1]) \wedge seMovio(sPos[i], sPos[i + 1], cOrigen, cDestino) \wedge$
esJugadaLegal(sPos[i],cOrigen,cDestino)))
}

proc esPosicionFutura (in p1: posicion, in p2:posicion , out res: Bool) {
Pre $\{esPosicionValida(p1) \wedge esPosicionValida(p2)\}$
Post $\{res =$ if $posicionesFuturas(p1, p2)$ then $True$ else $False$ fi$\}$
}

## Ejercicio 15

pred chequeoDeJaqueDescubierto (p: posicion) {
$(\exists pos : posicion)(\exists cOrigen : coordenada)(\exists cDestino : coordenada)(\exists cAtacante : coordenada)(\exists cRey : coordenada)$
((esPosicionValida(pos) $\wedge$
posicionDelRey(p,p1,cRey) $\wedge cAtacante \neq cOrigen \wedge cAtacante \neq cDestino) \wedge_L$
(esAtacada(p,p1,cOrigen) $\wedge seMovio(p, pos, cOrigen, cDestino) \wedge esJugadaLegal(p, cOrigen, cDestino) \wedge$
esJaque(pos,p1) $\wedge \neg capturaValida(p, cAtacante, cRey) \wedge$
capturaValida(pos,cAtacante,cRey))) }

proc hayJaqueDescubierto (in p: posicion, out res: Bool) {
Pre $\{esPosicionValida(p)\}$
Post $\{res =$ if $chequeoDeJaqueDescubierto(p)$ then $True$ else $False$ fi$\}$
}

## Ejercicio 16

pred mateEn1 (p : posicion) {

$(\exists pos : posicion)(\exists cOrigen : coordenada)(\exists cDestino : coordenada)((esPosicionValida(pos) \wedge$
$(0 \leq cDestino_0, cOrigen_0 < 8) \wedge (0 \leq cDestino_1, cOrigen_1 < 8)) \wedge_L$
$((\text{esJaque(pos,jugadorOponenete(p}_1\text{))}) \wedge \neg hayJugadaValida(pos)) \vee chequeoDeJaqueDescubierto(p))$
$\wedge seMovio(p, pos, cOrigen, cDestino) \wedge esJugadaLegal(p, cOrigen, cDestino))$

}

`proc hayMateEn1` (in p: posicion, out res: Bool) {
`Pre` $\{esPosicionValida(p)\}$
`Post` $\{res =$ if $mateEn1(p)$ then $True$ else $False$ fi$\}$
}

## Ejercicio 17

`pred unicoMovimiento` (p:posicion, c1: coordenada, c2: coordenada) {
$esJugadaLegal(p, c1, c2) \wedge (\neg\exists c3 : coordenada)(\neg\exists c4 : coordenada)((c3 \neq c1 \wedge c4 \neq c2) \wedge_L esJugadaLegal(p, c3, c4))$
}
`pred chequeoPosicionesValidas` (s:seq¡posicion x posicion¿) {
$(\forall i : \mathbb{Z})(0 \leq i < |s| \longrightarrow_L posicionValida(s[i]))$
}
`pred secuenciaForzadaTotal` (p:$seq\langle posicion\rangle$, s:$seq\langle coordenadaxcoordenada\rangle$, s':$seq\langle coordenadaxcoordenada\rangle$) {
$(\forall i : \mathbb{Z})((0 \leq i < |s|) \longrightarrow_L s'[2i] = s[i] \wedge unicoMovimiento(p[2i+1], s'[2i+1]_0, s'[2i+1]_1)) \wedge$
$(\forall i : \mathbb{Z})((0 \leq i < |s'| - 1) \longrightarrow_L seMovio(p[i], p[i+1], s'[i+1]_0, s'[i+1]_1))$
}
`pred ejecutarSeceucnia` (p1: $seq\langle posicion\rangle$, p2: $seq\langle posicon\rangle$, s: $seq\langle coordenadaxcoordenada\rangle$) {
$(\exists posSeq : seq\langle posicion\rangle)(\exists s' : seq\langle coordenadaxcoordenada\rangle)(|posSeq| = 2 * |s| \wedge$
$(\text{posSeq[0]} = \text{p1} \wedge posSeq[|posSeq| - 1] = p2) \wedge secuenciaForzadaTotal(posSeq, s, s'))$
}
`pred esSecuenciaForzada` (p: posicion, s:seq¡coordenada x coordenada¿) {
$(\exists posSeq : pos < posicion >)(\exists s' : seq < coordenadaxcoordenada >)(|posSeq| = 2 * |s| \wedge |s'| = 2 * |s| \wedge$
$\text{posSeq[0]} = \text{p}) \wedge_L secuenciaForzadaTotal(posSeq, s, s'))$
}
`proc ejecutarSecuenciaForzada` (inout p: posicion,in s: seq¡coordenada x coordenada¿) {
`Pre` $\{p = P_0 \wedge |s| > 0 \wedge esPosicionValida(P_0) \wedge esSecuenciaForzada(p, s)\}$
`Post` $\{ejecutarSecuencia(P_0, p, s)\}$
}