

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Pacalgo2

Los inertes

Integrante	LU	Correo electrónico
Bruno Robbio	480/09	brobbio@hotmail.com
Nicolas Andres Kinaschuk	248/15	nicolaskinaschuk@gmail.com
Pedro Joel Burgos	804/18	facultadburgospedrojoel@hotmail.com
Valentina Madelaine Saravia Ruiz	257/18	valentina.saraviaruiz@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. Introducción

1.1. Consideraciones TP1

- No se puede arrancar el mapa ganando o perdiendo
- El mapa puede no tener un camino ganador o perdedor
- El vértice del mapa es la esquina inferior izquierda

1.2. Consideraciones TP2

- Puede haber más de un jugador en una posición del ranking
- Si el jugador actual está en el primer puesto, el contrincante a superar puede ser él mismo
- No todos los jugadores juegan el mismo mapa

2. Desarrollo

2.1. TP 1

TAD CASILLERO

extiende Tupla(int, int)

usa Int

géneros casillero

exporta casillero, +, -, aDistanciaMenosDeN

otras operaciones

• + • : casillero \times casillero \longrightarrow casillero

• - • : casillero \times casillero \longrightarrow casillero

aDistanciaMenosDeN : casillero \times nat \longrightarrow conj(casillero)

axiomas

$\pi_1(c1 + c2) \equiv \pi_1(c1) + \pi_1(c2)$

$\pi_2(c1 + c2) \equiv \pi_2(c1) + \pi_2(c2)$

$\pi_1(c1 - c2) \equiv \pi_1(c1) - \pi_1(c2)$

$\pi_2(c1 - c2) \equiv \pi_2(c1) - \pi_2(c2)$

$aDistanciaMenosDeN(c, n) \equiv$ **if** $n=0?$ **then**
 {c}
 else
 ($aDistanciaMenosDeN(c + \langle 1,0 \rangle, n - 1) \cup$
 $aDistanciaMenosDeN(c - \langle 1,0 \rangle, n - 1) \cup$
 $aDistanciaMenosDeN(c + \langle 0,1 \rangle, n - 1) \cup$
 $aDistanciaMenosDeN(c - \langle 0,1 \rangle, n - 1) \cup$
 $aDistanciaMenosDeN(c, n - 1)$)
 fi

Fin TAD

TAD MAPA

usa Casillero, Conjunto, Bool

géneros mapa

exporta mapa, observadores, generadores, casillerosLibres

igualdad observacional

$$(\forall m_1, m_2 : \text{mapa}) \left(m_1 =_{\text{obs}} m_2 \iff \begin{pmatrix} \text{conjFantasmas}(m_1) =_{\text{obs}} \text{conjFantasmas}(m_2) \wedge \\ \text{conjParedes}(m_1) =_{\text{obs}} \text{conjParedes}(m_2) \wedge \\ \text{conjChocolates}(m_1) =_{\text{obs}} \text{conjChocolates}(m_2) \wedge \\ \text{dimensiones}(m_1) =_{\text{obs}} \text{dimensiones}(m_2) \wedge \\ \text{casilleroInicial}(m_1) =_{\text{obs}} \text{casilleroInicial}(m_2) \wedge \\ \text{vértice}(m_1) =_{\text{obs}} \text{vértice}(m_2) \wedge \\ \text{casilleroDeLlegada}(m_1) =_{\text{obs}} \text{casilleroDeLlegada}(m_2) \end{pmatrix} \right)$$

observadores básicos

$\text{conjFantasmas} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$

$\text{conjParedes} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$

$\text{conjChocolates} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$

$\text{dimensiones} : \text{mapa} \rightarrow \text{tupla}(\text{nat}, \text{nat})$

$\text{vértice} : \text{mapa} \rightarrow \text{casillero}$

$\text{casilleroInicial} : \text{mapa} \rightarrow \text{casillero}$

$\text{casilleroDeLlegada} : \text{mapa} \rightarrow \text{casillero}$

generadores

$\text{nuevoMapa} : \text{tupla}(\text{nat}, \text{nat}) \text{ dimensión} \times \text{casillero vértice} \times \text{casillero inicio} \times \text{casillero fin} \times$
 $\text{conj}(\text{casillero}) \text{ fantasmas} \times \text{conj}(\text{casillero}) \text{ paredes} \times \text{conj}(\text{casillero}) \text{ chocolates} \rightarrow \text{mapa}$

$$\left\{ \begin{array}{l} \emptyset?(fantasmas \cap paredes) \wedge \emptyset?(chocolates \cap paredes) \wedge \\ \emptyset?(fantasmas \cap chocolates) \wedge (\text{inicio} \neq \text{fin}) \wedge \\ \text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{inicio}) \wedge \\ \text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{fin}) \wedge \\ (\forall f \in \text{fantasmas})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, f) \wedge \\ (f \notin \text{aDistanciaMenosDeN}(\text{inicio}, f, 3) \vee (\text{inicio} \in \text{chocolates}))) \wedge \\ (\forall c \in \text{chocolates})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, c)) \wedge \\ (\forall p \in \text{paredes})(\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, p)) \end{array} \right\}$$

otras operaciones

$\text{casillerosLibres} : \text{mapa} \rightarrow \text{conj}(\text{casillero})$

$\text{dentroDeLasDimensiones} : \text{tupla}(\text{int} \times \text{int}) \times \text{casillero} \times \text{casillero} \rightarrow \text{bool}$

$\text{casillerosADerecha} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$

$\text{casillerosIzquierda} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$

$\text{casillerosHorizontales} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$

$\text{casillerosArriba} : \text{casillero } c \times \text{mapa } m \rightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$

$\text{casillerosAbajo} : \text{casillero } c \times \text{mapa } m \longrightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$
 $\text{casillerosVerticales} : \text{casillero } c \times \text{mapa } m \longrightarrow \text{conj}(\text{casillero})$
 $\{\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c)\}$
 $\text{generarCasillerosHorizontales} : \text{mapa } m \times \text{conj}(\text{casillero}) \ C \longrightarrow \text{conj}(\text{casillero})$
 $\{(\forall c \in C)(\text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c))\}$

axiomas

$\text{conjFantasmas}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{fantasmas}$
 $\text{conjParedes}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{paredes}$
 $\text{conjChocolates}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{chocolates}$
 $\text{dimensiones}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{dimensión}$
 $\text{casilleroInicial}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{inicio}$
 $\text{casilleroDeLlegada}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{fin}$
 $\text{vértice}(\text{nuevoMapa}(\text{dimensión}, \text{vértice}, \text{inicio}, \text{fin}, \text{fantasmas}, \text{paredes}, \text{chocolates})) \equiv \text{vértice}$
 $\text{dentroDeLasDimensiones}(\text{dimensión}, \text{vértice}, \text{casilla}) \equiv (0 \leq \pi_1(\text{casilla}) - \pi_1(\text{vértice}) < \pi_1(\text{dimensión})) \wedge$
 $(0 \leq \pi_2(\text{casilla}) - \pi_2(\text{vértice}) < \pi_2(\text{dimensión}))$
 $\text{casillerosLibres}(m) \equiv \text{generarCasillerosHorizontales}(m, \text{casillerosVerticales}(\text{vértice}(m), m)) - (\text{conjFantasmas}(m)$
 $\cup \text{conjParedes}(m))$
 $\text{generarCasillerosHorizontales}(m, \text{casilleros}) \equiv \text{if } \text{casilleros} = \emptyset \text{ then}$
 \emptyset
 else
 $\text{casillerosHorizontales}(\text{dameUno}(\text{casilleros}, m)) \cup$
 $\text{generarCasillerosHorizontales}(m, \text{sinUno}(\text{casilleros}))$
 fi
 $\text{casillerosHorizontales}(c, m) \equiv \text{casillerosADerecha}(c, m) \cup \text{casillerosAIzquierda}(c, m)$
 $\text{casillerosADerecha}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c + \langle 1, 0 \rangle) \text{ then}$
 $\text{casillerosADerecha}(c + \langle 1, 0 \rangle, m)$
 else
 \emptyset
 $\text{fi}) \cup \{c\}$
 $\text{casillerosAIzquierda}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c - \langle 1, 0 \rangle) \text{ then}$
 $\text{casillerosAIzquierda}(c - \langle 1, 0 \rangle, m)$
 else
 \emptyset
 $\text{fi}) \cup \{c\}$
 $\text{casillerosVerticales}(c, m) \equiv \text{casillerosArriba}(c, m) \cup \text{casillerosAbajo}(c, m)$
 $\text{casillerosArriba}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c + \langle 0, 1 \rangle) \text{ then}$
 $\text{casillerosArriba}(c + \langle 0, 1 \rangle, m)$
 else
 \emptyset
 $\text{fi}) \cup \{c\}$
 $\text{casillerosAbajo}(c, m) \equiv (\text{if } \text{dentroDeLasDimensiones}(\text{dimensión}(m), \text{vértice}(m), c - \langle 0, 1 \rangle) \text{ then}$
 $\text{casillerosAbajo}(c - \langle 0, 1 \rangle, m)$
 else
 \emptyset
 $\text{fi}) \cup \{c\}$

Fin TAD

TAD PACALGO2**usa** Mapa**géneros** pacalgo2**exporta** pacalgo2, observadores, generadores, direccionesPosibles, perdió?, ganó?**igualdad observacional**

$$(\forall p_1, p_2 : \text{pacalgo2}) \left(p_1 =_{\text{obs}} p_2 \iff \begin{pmatrix} \text{verMapa}(p_1) =_{\text{obs}} \text{verMapa}(p_2) \wedge \\ \text{posiciónActual}(p_1) =_{\text{obs}} \text{posiciónActual}(p_2) \wedge \\ \text{nivelDeChocolate}(p_1) =_{\text{obs}} \text{nivelDeChocolate}(p_2) \wedge \\ \text{chocolatesRestantes}(p_1) =_{\text{obs}} \text{chocolatesRestantes}(p_2) \wedge \\ \text{ganó?}(p_1) =_{\text{obs}} \text{ganó?}(p_2) \wedge \\ (\text{ganó?}(p_1) \Rightarrow_L \text{puntaje}(p_1) =_{\text{obs}} \text{puntaje}(p_2)) \end{pmatrix} \right)$$

observadores básicosverMapa : pacalgo2 \longrightarrow mapaposiciónActual : pacalgo2 \longrightarrow casilleropuntaje : pacalgo2 \longrightarrow nat {ganó?(p)}nivelDeChocolate : pacalgo2 \longrightarrow natchocolatesRestantes : pacalgo2 \longrightarrow conj(casillero)**generadores**inicializarJuego : mapa \longrightarrow pacalgo2arriba : pacalgo2 $p \longrightarrow$ pacalgo2
 $\{(posiciónActual(p) + \langle 0, 1 \rangle) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p)\}$ abajo : pacalgo2 $p \longrightarrow$ pacalgo2
 $\{(posiciónActual(p) - \langle 0, 1 \rangle) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p)\}$ derecha : pacalgo2 $p \longrightarrow$ pacalgo2
 $\{(posiciónActual(p) + \langle 1, 0 \rangle) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p)\}$ izquierda : pacalgo2 $p \longrightarrow$ pacalgo2
 $\{(posiciónActual(p) - \langle 1, 0 \rangle) \in \text{direccionesPosibles}(p) \wedge \neg \text{ganó?}(p) \wedge \neg \text{perdió?}(p)\}$ **otras operaciones**direccionesPosibles : pacalgo2 \longrightarrow conj(casillero)perdió? : pacalgo2 \longrightarrow boolganó? : pacalgo2 \longrightarrow boolpasos : pacalgo2 \longrightarrow nat**axiomas**verMapa(inicializarJuego(m)) $\equiv m$ verMapa(arriba(p)) $\equiv \text{verMapa}(p)$ verMapa(abajo(p)) $\equiv \text{verMapa}(p)$ verMapa(izquierda(p)) $\equiv \text{verMapa}(p)$ verMapa(derecha(p)) $\equiv \text{verMapa}(p)$ posiciónActual(inicializarJuego m) $\equiv \text{casilleroInicial}(m)$ posiciónActual(arriba(p)) $\equiv \text{posiciónActual}(p) + \langle 0, 1 \rangle$ posiciónActual(abajo(p)) $\equiv \text{posiciónActual}(p) - \langle 0, 1 \rangle$

posiciónActual(izquierda(p))	\equiv posiciónActual(p) - $\langle 1, 0 \rangle$
posiciónActual(derecha(p))	\equiv posiciónActual(p) + $\langle 1, 0 \rangle$
pasos(inicializarJuego(m))	\equiv 0
pasos(arriba(p))	\equiv 1 + pasos(p)
pasos(abajo(p))	\equiv 1 + pasos(p)
pasos(izquierda(p))	\equiv 1 + pasos(p)
pasos(derecha(p))	\equiv 1 + pasos(p)
puntaje(p)	\equiv pasos(p)
nivelDeChocolate(inicializarJuego(m))	\equiv if casilleroInicial(m) \in conjChocolates(m) then 10 else 0 fi
nivelDeChocolate(arriba(p))	\equiv if (posiciónActual(p) + $\langle 0, 1 \rangle$) \in chocolatesRestantes(p) then 10 else if 0?(nivelDeChocolate(p)) then nivelDeChocolate(p) else nivelDeChocolate(p) - 1 fi fi
nivelDeChocolate(abajo(p))	\equiv if (posiciónActual(p) - $\langle 0, 1 \rangle$) \in chocolatesRestantes(p) then 10 else if 0?(nivelDeChocolate(p)) then nivelDeChocolate(p) else nivelDeChocolate(p) - 1 fi fi
nivelDeChocolate(izquierda(p))	\equiv if (posiciónActual(p) - $\langle 1, 0 \rangle$) \in chocolatesRestantes(p) then 10 else if 0?(nivelDeChocolate(p)) then nivelDeChocolate(p) else nivelDeChocolate(p) - 1 fi fi
nivelDeChocolate(derecha(p))	\equiv if (posiciónActual(p) + $\langle 1, 0 \rangle$) \in chocolatesRestantes(p) then 10 else if 0?(nivelDeChocolate(p)) then nivelDeChocolate(p) else nivelDeChocolate(p) - 1 fi fi
chocolatesRestantes(inicializarJuego(m))	\equiv if casilleroInicial(m) \in conjChocolates(m) then conjChocolates(m) - { casilleroInicial(m) } else conjChocolates(m) fi

<code>chocolatesRestantes(arriba(p))</code>	\equiv if (<code>posiciónActual(p) + $\langle 0, 1 \rangle$</code>) \in <code>chocolatesRestantes(p)</code> then <code>chocolatesRestantes(p) - { (posiciónActual(p) + $\langle 0, 1 \rangle$) }</code> else <code>chocolatesRestantes(p)</code> fi
<code>chocolatesRestantes(abajo(p))</code>	\equiv if (<code>posiciónActual(p) - $\langle 0, 1 \rangle$</code>) \in <code>chocolatesRestantes(p)</code> then <code>chocolatesRestantes(p) - { (posiciónActual(p) - $\langle 0, 1 \rangle$) }</code> else <code>chocolatesRestantes(p)</code> fi
<code>chocolatesRestantes(izquierda(p))</code>	\equiv if (<code>posiciónActual(p) - $\langle 1, 0 \rangle$</code>) \in <code>chocolatesRestantes(p)</code> then <code>chocolatesRestantes(p) - { (posiciónActual(p) - $\langle 1, 0 \rangle$) }</code> else <code>chocolatesRestantes(p)</code> fi
<code>chocolatesRestantes(derecha(p))</code>	\equiv if (<code>posiciónActual(p) + $\langle 1, 0 \rangle$</code>) \in <code>chocolatesRestantes(p)</code> then <code>chocolatesRestantes(p) - { (posiciónActual(p) + $\langle 1, 0 \rangle$) }</code> else <code>chocolatesRestantes(p)</code> fi
<code>perdió?(p)</code>	\equiv $\neg \emptyset?(\text{conjFantasmas}(\text{verMapa}(p)) \cap$ $\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 3)) \wedge$ $0?(\text{nivelDeChocolate}(p))$
<code>ganó?(p)</code>	\equiv <code>posiciónActual(p) = casilleroDeLlegada(verMapa(p))</code>
<code>direccionesPosibles(p)</code>	\equiv if $0?(\text{nivelDeChocolate}(p))$ then $(\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 1) - \text{posiciónActual}(p))$ \cap $\text{casillerosLibres}(\text{verMapa}(p))$ else $(\text{aDistanciaMenosDeN}(\text{posiciónActual}(p), 1) - \text{posiciónActual}(p))$ \cap $(\text{casillerosLibres}(\text{verMapa}(p)) \cup \text{conjFantasmas}(\text{verMapa}(p)))$ fi

Fin TAD

2.2. TP 2

TAD persona es String

TAD RANKING

usa Nat, persona

géneros ranking

exporta observadores, generadores, contrincante

igualdad observacional

$$(\forall r_1, r_2 : \text{ranking}) (r_1 =_{\text{obs}} r_2 \iff (\text{ver}(r_1) =_{\text{obs}} \text{ver}(r_2)))$$

observadores básicos

$\text{ver} : \text{ranking} \longrightarrow \text{dicc}(\text{persona}, \text{nat})$

generadores

$\text{iniciarRanking} : \longrightarrow \text{ranking}$

$\text{cargarPuntaje} : \text{ranking} \times \text{persona} \times \text{nat} \longrightarrow \text{ranking}$

otras operaciones

$\text{contrincante} : \text{ranking } r \times \text{persona } j \longrightarrow \text{tupla}(\text{persona}, \text{nat})$
 $\{\text{def?}(j, \text{ver}(r))\}$

$\text{superioresInmediatos} : \text{ranking } r \times \text{persona } j \longrightarrow \text{conj}(\text{persona})$
 $\{\text{def?}(j, \text{ver}(r))\}$

$\text{elMaximoEntreMenores}^1 : \text{conj}(\text{nat}) \text{ conjunto} \times \text{nat } p \longrightarrow \text{nat}$

$\text{todosLosPuntajes} : \text{ranking } r \times \text{conj}(\text{persona}) \text{ nombres} \longrightarrow \text{conj}(\text{nat})$
 $\{\text{nombres} \subseteq \text{claves}(\text{ver}(r))\}$

$\text{jugadoresConPuntajeIgual} : \text{ranking } r \times \text{conj}(\text{persona}) \text{ nombres} \times \text{nat} \longrightarrow \text{conj}(\text{persona})$
 $\{\text{nombres} \subseteq \text{claves}(\text{ver}(r))\}$

$\text{menoresQueP} : \text{conj}(\text{nat}) \times \text{nat} \longrightarrow \text{conj}(\text{nat})$

$\text{máximoDeConjunto} : \text{conj}(\text{nat}) \text{ } C \longrightarrow \text{nat} \quad \{\neg \emptyset?(C)\}$

axiomas

$\text{ver}(\text{iniciarRanking}()) \equiv \text{vacío}$

$\text{ver}(\text{cargarPuntaje}(\text{ranking}, \text{nombre}, \text{puntaje})) \equiv \text{if } \text{def?}(\text{nombre}, \text{ver}(\text{ranking})) \text{ then}$
 $\quad \text{if } \text{puntaje} < \text{obtener}(\text{ver}(r), \text{nombre}) \text{ then}$
 $\quad \quad \text{definir}(\text{ver}(\text{ranking}), (\text{nombre}, \text{puntaje}))$
 $\quad \text{else}$
 $\quad \quad \text{ver}(\text{ranking})$
 $\quad \text{fi}$
 else
 $\quad \text{definir}(\text{ver}(\text{ranking}), (\text{nombre}, \text{puntaje}))$
 fi

$\text{todosLosPuntajes}(r) \equiv \text{if } \emptyset?(\text{ver}(r)) \text{ then}$
 $\quad \emptyset$
 else
 $\quad \text{Ag}(\text{obtener}(\text{ver}(r), \text{dameUno}(\text{claves}(\text{ver}(r)))),$
 $\quad \text{todosLosPuntajes}(r))$
 fi

¹Si no hay un menor que p en el *conjunto*, devuelve p


```

jugadoresConPuntajeIgual(r, nombres, puntaje)  ≡  if  $\emptyset?(nombres) \vee \emptyset?(claves(ver(r)))$  then
     $\emptyset$ 
else
    if obtener(ver(r), dameUno(nombres)) = puntaje
    then
        Ag(dameUno(nombres),
            jugadorConPuntajeIgual(r,      sinUno(nombres),
            puntaje))
    else
        jugadorConPuntajeIgual(r,sinUno(nombres),puntaje)
    fi
fi

superioresInmediatos(r, j)  ≡  jugadoresConPuntajeIgual(r,claves(ver(r)),
    elMaximoEntreMenores(todosLosPuntajes(r),
    obtener(ver(r),j)))

elMaximoEntreMenores(puntajes, p)  ≡  if  $\emptyset?(menoresQueP(puntajes, p))$  then
    p
else
    máximoDeConjunto(menoresQueP(puntajes, p))
fi

menoresQueP(puntajes, p)  ≡  if  $\emptyset?(puntajes)$  then
     $\emptyset$ 
else
    if dameUno(puntajes)<p then
        Ag(dameUno(puntajes),
            menoresQueP(sinUno(puntajes), p))
    else
        menoresQueP(sinUno(puntajes), p)
    fi
fi

máximoDeConjunto(C)  ≡  if  $\emptyset?(sinUno(C))$  then
    dameUno(C)
else
    máximo(dameUno(C), máximoDeConjunto(sinUno(C)))
fi

contrincante(r,j)  ≡  (dameUno(superioresInmediatos(r,j)),  obtener(ver(r),
    dameUno(superioresInmediatos(r,j))))

```

Fin TAD

TAD FICHÍN**usa** Ranking, Pacalgo2**géneros** fichín**exporta** observadores, generadores, otras operaciones**igualdad observacional**

$$(\forall f_1, f_2 : \text{fichín}) \left(f_1 =_{\text{obs}} f_2 \iff \begin{pmatrix} \text{conocerRanking}(f_1) =_{\text{obs}} \text{conocerRanking}(f_2) \wedge \\ \text{estáLibre?}(f_1) =_{\text{obs}} \text{estáLibre?}(f_2) \wedge \\ (\neg \text{estáLibre?}(f_1) \Rightarrow_L \\ \text{verJuegoActual}(f_1) =_{\text{obs}} \text{verJuegoActual}(f_2) \wedge \\ \text{jugadorActual}(f_1) =_{\text{obs}} \text{jugadorActual}(f_2)) \end{pmatrix} \right)$$

observadores básicosconocerRanking : fichín \longrightarrow rankingverJuegoActual : fichín $f \longrightarrow$ pacalgo2 { \neg estáLibre?(f)}jugadorActual : fichín $f \longrightarrow$ persona { \neg estáLibre?(f)}**generadores**nuevoFichin : \longrightarrow fichínnuevaPartida : fichín $f \times$ pacalgo2 $p \times$ persona $j \longrightarrow$ fichín {estáLibre?(f)}

palancaArriba : fichín $f \longrightarrow$ fichín

$$\left\{ \neg \text{estáLibre?}(f) \wedge_L \right. \\ \left. (\text{posiciónActual}(\text{verJuegoActual}(f)) + \langle 0, 1 \rangle) \in \text{direccionesPosibles}(\text{verJuegoActual}(f)) \right\}$$

palancaAbajo : fichín $f \longrightarrow$ fichín

$$\left\{ \neg \text{estáLibre?}(f) \wedge_L \right. \\ \left. (\text{posiciónActual}(\text{verJuegoActual}(f)) + \langle 0, -1 \rangle) \in \text{direccionesPosibles}(\text{verJuegoActual}(f)) \right\}$$

palancaDerecha : fichín $f \longrightarrow$ fichín

$$\left\{ \neg \text{estáLibre?}(f) \wedge_L \right. \\ \left. (\text{posiciónActual}(\text{verJuegoActual}(f)) + \langle 1, 0 \rangle) \in \text{direccionesPosibles}(\text{verJuegoActual}(f)) \right\}$$

palancaIzquierda : fichín $f \longrightarrow$ fichín

$$\left\{ \neg \text{estáLibre?}(f) \wedge_L \right. \\ \left. (\text{posiciónActual}(\text{verJuegoActual}(f)) + \langle -1, 0 \rangle) \in \text{direccionesPosibles}(\text{verJuegoActual}(f)) \right\}$$
otras operaciones

puntajeDeJugadorActual : fichín $f \longrightarrow$ nat

$$\{ \neg \text{estáLibre?}(f) \wedge_L \text{jugadorActual}(f) \in \text{claves}(\text{ver}(\text{conocerRanking}(f))) \}$$

verContrincante : fichín $f \longrightarrow$ tupla(persona, nat)

$$\{ \neg \text{estáLibre?}(f) \wedge_L \text{jugadorActual}(f) \in \text{claves}(\text{ver}(\text{conocerRanking}(f))) \}$$
estáLibre? : fichín \longrightarrow Bool

puntajeParaMejorar : fichín $f \longrightarrow$ nat

$$\{ \neg \text{estáLibre?}(f) \wedge_L \text{jugadorActual}(f) \in \text{claves}(\text{ver}(\text{conocerRanking}(f))) \}$$
axiomasestáLibre?(nuevoFichin()) \equiv TrueestáLibre?(nuevaPartida(f, p, j)) \equiv FalseestáLibre?(palancaArriba(f)) \equiv ganó?(arriba(verJuegoActual(f))) \vee perdió?(arriba(verJuegoActual(f)))estáLibre?(palancaAbajo(f)) \equiv ganó?(abajo(verJuegoActual(f))) \vee perdió?(abajo(verJuegoActual(f)))estáLibre?(palancaDerecha(f)) \equiv ganó?(derecha(verJuegoActual(f))) \vee perdió?(derecha(verJuegoActual(f)))

<code>estáLibre?(palancaIzquierda(f))</code>	\equiv	<code>ganó?(izquierda(verJuegoActual(f)))\veeperdió?(izquierda(verJuegoActual(f)))</code>
<code>jugadorActual(nuevaPartida(f,p,j))</code>	\equiv	<code>j</code>
<code>jugadorActual(palancaArriba(f))</code>	\equiv	<code>jugadorActual(f)</code>
<code>jugadorActual(palancaAbajo(f))</code>	\equiv	<code>jugadorActual(f)</code>
<code>jugadorActual(palancaDerecha(f))</code>	\equiv	<code>jugadorActual(f)</code>
<code>jugadorActual(palancaIzquierda(f))</code>	\equiv	<code>jugadorActual(f)</code>
<code>conocerRanking(nuevoFichin())</code>	\equiv	<code>iniciarRanking()</code>
<code>conocerRanking(nuevaPartida(f,p,j))</code>	\equiv	<code>conocerRanking(f)</code>
<code>conocerRanking(palancaArriba(f))</code>	\equiv	if <code>ganó?(arriba(verJuegoActual(f)))</code> then <code>cargarPuntaje(conocerRanking(f),</code> <code>puntaje(verJuegoActual(palancaArriba(f))))</code> else <code>conocerRanking(f)</code> fi
<code>conocerRanking(palancaAbajo(f))</code>	\equiv	if <code>ganó?(abajo(verJuegoActual(f)))</code> then <code>cargarPuntaje(conocerRanking(f),</code> <code>puntaje(verJuegoActual(palancaAbajo(f))))</code> else <code>conocerRanking(f)</code> fi
<code>conocerRanking(palancaDerecha(f))</code>	\equiv	if <code>ganó?(derecha(verJuegoActual(f)))</code> then <code>cargarPuntaje(conocerRanking(f),</code> <code>puntaje(verJuegoActual(palancaDerecha(f))))</code> else <code>conocerRanking(f)</code> fi
<code>conocerRanking(palancaIzquierda(f))</code>	\equiv	if <code>ganó?(izquierda(verJuegoActual(f)))</code> then <code>cargarPuntaje(conocerRanking(f),</code> <code>puntaje(verJuegoActual(palancaIzquierda(f))))</code> else <code>conocerRanking(f)</code> fi
<code>verJuegoActual(nuevaPartida(f, p, j))</code>	\equiv	<code>p</code>
<code>verJuegoActual(palancaArriba(f))</code>	\equiv	<code>arriba(verJuegoActual(f))</code>
<code>verJuegoActual(palancaAbajo(f))</code>	\equiv	<code>abajo(verJuegoActual(f))</code>
<code>verJuegoActual(palancaDerecha(f))</code>	\equiv	<code>derecha(verJuegoActual(f))</code>
<code>verJuegoActual(palancaIzquierda(f))</code>	\equiv	<code>izquierda(verJuegoActual(f))</code>
<code>verContrincante(f)</code>	\equiv	<code>contrincante(conocerRanking(f),jugadorActual(f))</code>
<code>puntajeDeJugadorActual(f)</code>	\equiv	<code>obtener(ver(conocerRanking(f)),jugadorActual(f))</code>
<code>puntajeParaMejorar(f)</code>	\equiv	<code>pred(π_2(verContrincante(f)))</code>

Fin TAD