

Trabajo Practico I, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD UNIVERSIDAD	2
2. TAD AGENTE	14
3. TAD POSICION	15
4. TAD DIRECCION	16
5. TAD SITUACIONESRODEO	16
6. TAD TIPOELEM	17

TAD ID es NAT, TAD PLACA es NAT, TAD EST es SECU(DIR)

1. TAD UNIVERSIDAD

TAD UNIVERSIDAD

géneros uni

exporta uni, Generadores, Observadores Basicos, masVigilante, cuantosHippies, cuantosEstudiantes

usa NAT, CONJ(α), BOOL, TUPLA($\alpha_1, \dots, \alpha_n$), MULTICONJ(α), ID, PLACA, EST, CAMPUS, DIRECCION, SITUACIONESRODEO, TIPOELEM, DICCIONARIO(CLAVE, SIGNIFICADO)

igualdad observacional

$$(\forall u, u' : \text{uni}) \left(u =_{\text{obs}} u' \iff \begin{pmatrix} \text{campus?}(u) =_{\text{obs}} \text{campus?}(u') \wedge \\ \text{estudiantes?}(u) =_{\text{obs}} \text{estudiantes?}(u') \wedge \\ \text{hippies?}(u) =_{\text{obs}} \text{hippies?}(u') \wedge \\ \text{agentes?}(u) =_{\text{obs}} \text{agentes?}(u') \wedge_L \\ (\forall p : \text{placa}) \text{def?}(p, \text{agentes?}(u)) \Rightarrow_L \\ (\text{sanciones?}(u, p) =_{\text{obs}} \text{sanciones?}(u', p) \wedge \\ \text{capturas?}(u, p) =_{\text{obs}} \text{capturas?}(u', p)) \end{pmatrix} \right)$$

Los Agentes son manejados como triplas que tienen al Id, al Agente en cuestion, y su posicion. Los estudiantes son triplas con Id, Estudiante y Posicion. Los Hippies son duplas de Id y Posicion.

Los Estudiantes son representados como una secuencia de direcciones en las cuales moverse. Si en algun momento su camino esta bloqueado, no se mueven, ya que consideramos que lo interesante se encuentra en el camino que debe recorrer.

observadores básicos

campus? : uni \rightarrow campus

agentes? : uni \rightarrow dicc(placa, pos)

hippies? : uni \rightarrow dicc(id, pos)

estudiantes? : uni \rightarrow dicc(id, \langle est, pos \rangle)

sanciones? : placa $pl \times$ uni $u \rightarrow$ nat {def?(pl, agentes?(u))}

capturas? : placa $pl \times$ uni $u \rightarrow$ nat {def?(pl, agentes?(u))}

generadores

nuevaUni : campus $c \times$ dicc(placa, pos) $as \rightarrow$ uni {posicionesValidas(c, as)}

agregarE : uni $u \times$ id $i \times$ est $e \times$ pos $p \rightarrow$ uni
 $\{\neg(\text{def?}(i, \text{hippies?}(u))) \wedge \neg(\text{def?}(i, \text{estudiantes?}(u))) \wedge \text{posValida}(p, u) \wedge \text{entrada?}(p, \text{campus?}(u))\}$

agregarH : uni $u \times$ id $i \times$ pos $p \rightarrow$ uni
 $\{\neg(\text{def?}(i, \text{hippies?}(u))) \wedge \neg(\text{def?}(i, \text{estudiantes?}(u))) \wedge \text{posValida}(p, u) \wedge \text{entrada?}(p, \text{campus?}(u))\}$

moverAS : uni $u \times$ placa $p \rightarrow$ uni { (def?(p, agentes?(u))) \wedge_L (sanciones?(p, u) < 3)}

moverH : uni $u \times$ id $i \rightarrow$ uni {def?(i, hippies?(u))}

moverE : uni $u \times$ id $i \rightarrow$ uni {def?(i, estudiantes?(u))}

otras operaciones

capturar : pos \times conj(\langle id, agente, pos \rangle) \rightarrow conj(\langle agente, pos \rangle)

sancionar : pos \times conj(\langle id, agente, pos \rangle) \rightarrow conj(\langle agente, pos \rangle)

queTipoHay : pos \times \langle cAs, cH, cEst $\rangle \rightarrow$ tipo

cuatroVecinosShort : pos \times \langle cAs, cH, cEst $\rangle \times$ uni \rightarrow multiconj(tipo)

cuatroVecinos : pos \times \langle cAs, cH, cEst $\rangle \times$ uni \times conj(pos) \rightarrow multiconj(tipo)

queSituacion : multiconj(tipo) \rightarrow conj(situ)

damePosicionesAs : dicc(placa, pos) \rightarrow conj(pos)

$\text{damePosicionesEst} : \text{dicc}(\text{id}, \langle \text{agente}, \text{pos} \rangle) \longrightarrow \text{conj}(\text{pos})$
 $\text{damePosicionesH} : \text{dicc}(\text{id}, \text{pos}) \longrightarrow \text{conj}(\text{pos})$
 $\text{moverEstudianteYChequearSituaciones} : \text{id } is \times \text{uni } u \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle$
 $\{\exists t \in \text{estudiantes?}(u) / \Pi_0(t) = is\}$
 $\text{dameEstudiante} : \text{id } is \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \text{ conjEst} \longrightarrow \text{est}$
 $\{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$
 $\text{dameEstudiantePos} : \text{id } is \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \text{ conjEst} \longrightarrow \text{pos}$
 $\{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$
 $\text{sacarEstudianteId} : \text{id } is \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \text{ conjEst} \longrightarrow \text{conj}(\langle i, \text{est}, \text{pos} \rangle)$
 $\{\exists t \in \text{conjEst} / \Pi_0(t) = is\}$
 $\text{entradas?} : \text{nat} \times \text{nat} \longrightarrow \text{conj}(\text{pos})$
 $\text{dirLibres} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$
 $\text{dirValidas} : \text{pos} \times \text{nat} \times \text{nat} \longrightarrow \text{conj}(\text{dir})$
 $\text{dirNoOcupadas} : \text{conj}(\text{dir}) \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$
 $\{\neg(\text{col?}(\text{pos}) = 0 \wedge \{o\} \in \text{cDirs}) \wedge \neg(\text{fila?}(\text{pos}) = 0 \text{ y } \{n\} \in \text{cDirs})\}$
 $\text{posNoOcupadas} : \text{conj}(\text{pos}) \times \text{uni} \longrightarrow \text{conj}(\text{pos})$
 $\text{moverAgenteYChequearSituacion} : \text{id } is \times \text{uni } u \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle$
 $\{\exists t \in \text{agentes?}(u) / \Pi_0(t) = is\}$
 $\text{posibleMovAs} : \text{As} \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$
 $\text{dameAgente} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{as}$
 $\{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$
 $\text{dameAgentePos} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{pos}$
 $\{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$
 $\text{sacarAgenteId} : \text{id } is \times \text{conj}(\langle i, \text{as}, \text{pos} \rangle) \text{ conjAs} \longrightarrow \text{conj}(\langle i, \text{as}, \text{pos} \rangle)$
 $\{\exists t \in \text{conjAs} / \Pi_0(t) = is\}$
 $\text{moverHippieYChequearSituacion} : \text{id } is \times \text{uni} \times u \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle$
 $\{\exists t \in \text{hippies?}(u) / \Pi_0(t) = is\}$
 $\text{posibleMovH} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$
 $\text{dameHippiePos} : \text{id } is \times \text{conj}(\langle i, \text{pos} \rangle) \text{ conjHip} \longrightarrow \text{pos}$
 $\{\exists t \in \text{conjHip} / \Pi_0(t) = is\}$
 $\text{sacarHippieId} : \text{id } is \times \text{conj}(\langle i, \text{pos} \rangle) \text{ conjHip} \longrightarrow \text{conj}(\langle i, \text{pos} \rangle)$
 $\{\exists t \in \text{conjHip} / \Pi_0(t) = is\}$
 $\text{chequearSituacionShort} : \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \times \text{uni} \longrightarrow$
 $\langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{chequearSituacion} : \text{MACRO1} \times \text{MACRO1} \times \text{uni} \longrightarrow \text{MACRO1}$
 $\text{MACRO1} = \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{agAgenteTripla} : \langle i, \text{as}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$
 $\langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{agHippieTripla} : \langle i, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$
 $\langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{agEstudianteTripla} : \langle i, \text{est}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$
 $\langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{estudiantesAdyacentesPos} : \text{pos} \times \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \longrightarrow \text{conj}(\text{est})$
 $\text{convertirHippieAEst} : \langle \text{id}, \text{pos} \rangle \times \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \longrightarrow \langle \text{id}, \text{est}, \text{pos} \rangle$
 $\text{convertirEstAHippie} : \langle \text{id}, \text{est}, \text{pos} \rangle \longrightarrow \langle \text{id}, \text{pos} \rangle$
 $\text{sacarHippieTripla} : \langle \text{id}, \text{pos} \rangle \times \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle \longrightarrow$
 $\langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{conj}(\langle i, \text{pos} \rangle), \text{conj}(\langle i, \text{est}, \text{pos} \rangle) \rangle$
 $\text{dondeEstaAgente} : \text{id } is \times \text{uni } u \longrightarrow \text{pos}$
 $\{\exists t \in \text{agentes?}(u) / \Pi_0(t) = is\}$
 $\text{dondeEstaHippie} : \text{id } is \times \text{uni } u \longrightarrow \text{pos}$
 $\{\exists t \in \text{hippies?}(u) / \Pi_0(t) = is\}$
 $\text{dondeEstaEstudiante} : \text{id } is \times \text{uni } u \longrightarrow \text{pos}$
 $\{\exists t \in \text{estudiantes?}(u) / \Pi_0(t) = is\}$
 $\text{masVigilante} : \text{uni} \longrightarrow \langle \text{id}, \text{as}, \text{pos} \rangle$
 $\text{losMasVigilantes} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \times \text{nat} \longrightarrow \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle)$
 $\text{losMasVigilantesShort} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \longrightarrow \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle)$
 $\text{agenteConMenorPlaca} : \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \times \text{nat} \longrightarrow \langle \text{id}, \text{as}, \text{pos} \rangle$

agenteConMenorPlacaShort : conj($\langle \text{id}, \text{as}, \text{pos} \rangle$) \longrightarrow $\langle \text{id}, \text{as}, \text{pos} \rangle$

maxCapturas : conj($\langle \text{id}, \text{as}, \text{pos} \rangle$) \longrightarrow nat

minPlaca : conj($\langle \text{id}, \text{as}, \text{pos} \rangle$) \longrightarrow nat

cuantosHippies : uni \longrightarrow nat

cuantosEstudiantes : uni \longrightarrow nat

axiomas $\forall cAs: \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \forall cH: \text{conj}(\langle \text{id}, \text{pos} \rangle) \forall cEst: \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \forall cDirs: \text{conj}(\text{dir})$
 $\forall al, an, maxC, minN: \text{nat} \forall cObs: \text{conj}(\text{pos}) \forall u: \text{uni} \forall i: \text{id} \forall e: \langle \text{id}, \text{est}, \text{pos} \rangle \forall agente: \langle \text{id}, \text{as}, \text{pos} \rangle \forall h: \langle \text{id}, \text{pos} \rangle \forall p: \text{pos} \forall pAv: \text{conj}(\text{pos}) \forall tripla, triplaInfo: \langle \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle), cH \text{ conj}(\langle \text{id}, \text{pos} \rangle), \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \rangle \forall mcT: \text{multiconj}(\text{tipo})$

Observadores Basicos

campus? (nuevaUni(c, dAs)) \equiv

c

campus? (agregarE(u, i, e, p)) \equiv

campus?(u)

campus? (agregarH(u, i, p)) \equiv

campus?(u)

campus? (moverAs(u, p)) \equiv

campus?(u)

campus? (moverH(u, i)) \equiv

campus?(u)

campus? (moverE(u, i)) \equiv

campus?(u)

agentes? (nuevaUni(c, dAs)) \equiv

dAs

agentes? (agregarE(u, i, e, p)) \equiv

agentes?(u)

agentes? (agregarH(u, i, pos)) \equiv

agentes?(u)

agentes? (moverAs(u, p)) \equiv

moverAgente(p, u)

agentes? (moverE(u, i)) \equiv

agentes?(u)

agentes? (moverH(u, i)) \equiv

agentes?(u)

hippies? (nuevaUni(c, dAs)) \equiv

vacio

hippies? (agregarE(u, i, e, p)) \equiv

```

if hayDosHippies?(p, hippies?(u)) then
    definir(i, p, hippies?(u))
else
    if algunHippieEncerradoPorEstudiantes(hippies?(u), definir(i,  $\langle e, p \rangle$ , estudiantes?(u))) then
        borrar( hippieEncerradoPorEstudiantes(hippies?(u), definir(i,  $\langle e, p \rangle$ , estudiantes?(u))), hippies?(u) )
    else
        hippies?(u)
    fi
fi

    hippies? (agregarH(u, i, p))  $\equiv$ 
if murioAlgunHippie(campus?(u), agentes(u), definir(i, p, hippies?(u)), estudiantes?(u)) then
    borrarTodosHippiesMuertos( campus?(u), moverAgente(i, u), definir(i, p, hippies?(u)), estudiantes?(u) )
else
    definir(i, p, hippies?(u))
fi

    hippies? (moverAs(u, i))  $\equiv$ 
if murioAlgunHippie(campus?(u), moverAgente(i, u), hippies?(u), estudiantes?(u)) then
    borrarTodosHippiesMuertos( campus?(u), moverAgente(i, u), hippies?(u), estudiantes?(u) )
else
    hippies?(u)
fi

    hippies? (moverE(u, i))  $\equiv$ 
if murioAlgunHippie(campus?(u), agentes?(u), hippies?(u), moverEstudiante(i, u)) then
    if hayDosHippies?(obtener(i, moverEstudiante(i, u)), borrarTodosHippiesMuertos( campus?(u), agentes?(u),
    hippies?(u), moverEstudiante(i, u) )) then
        definir(i, obtener(i, moverEstudiante(i, u)), borrarTodosHippiesMuertos( campus?(u), agentes?(u),
        hippies?(u), moverEstudiante(i, u) ))
    else
        if algunHippieEncerradoPorEstudiantes(borrarTodosHippiesMuertos( campus?(u), agentes?(u),
        hippies?(u), moverEstudiante(i, u) ), moverEstudiante(i, u)) then
            borrarTodosHippiesEncerradoPorEstudiantes( borrarTodosHippiesMuertos( campus?(u), agentes?(u),
            hippies?(u), moverEstudiante(i, u) ), moverEstudiante(i, u) )
        else
            borrarTodosHippiesMuertos( campus?(u), agentes?(u), hippies?(u), moverEstudiante(i, u) )
        fi
    fi
else
    if hayDosHippies?(obtener(i, moverEstudiante(i, u)), hippies?(u)) then
        definir(i, obtener(i, moverEstudiante(i, u)), hippies?(u))
    else
        if algunHippieEncerradoPorEstudiantes(hippies?(u), moverEstudiante(i, u)) then
            borrarTodosHippiesEncerradoPorEstudiantes( hippies?(u), moverEstudiante(i, u) )
        else
            hippies?(u)
        fi
    fi
fi

    hippies? (moverH(u, i))  $\equiv$ 

```

```

if murioAlgunHippie(campus?(u), agentes?(u), moverHippie(i, u), estudiantes?(u)) then
  if conviertoAlgunEstudiante?( borrarTodosHippiesMuertos( campus?(u), agentes?(u), moverHippie(i, u),
    estudiantes?(u) ), estudiantes?(u) ) then
    borrarTodosHippiesMuertos( campus?(u), agentes?(u), agregoEstudiantesConvertidos(
      borrarTodosHippiesMuertos( campus?(u), agentes?(u), moverHippie(i, u), estudiantes?(u) ),
      estudiantes?(u) )
    else
      borrarTodosHippiesMuertos( campus?(u), agentes?(u), moverHippie(i, u), estudiantes?(u) )
    fi
  else
    if conviertoAlgunEstudiante?( moverHippie(i, u), estudiantes?(u) ) then
      borrarTodosHippiesMuertos( campus?(u), agentes?(u), agregoEstudiantesConvertidos( moverHippie(i, u),
        estudiantes?(u), estudiantes?(u) )
    else
      hippies?(u)
    fi
  fi

  estudiantes? (nuevaUni(c, dAs)) ≡
vacio

  estudiantes? (agregarE(u, i, e, p)) ≡
if hayDosHippies?(p, hippies?(u)) then estudiantes?(u) else definir(i, ⟨ e, p ⟩, estudiantes?(u)) fi

  estudiantes? (agregarH(u, i, p)) ≡
if conviertoAlgunEstudiante?( definir(i, p, hippies?(u)), estudiantes?(u) ) then
  borrarTodosEstudiantesConvertidos( definir(i, p, hippies?(u)), estudiantes?(u) )
else
  estudiantes?(u)
fi

  estudiantes? (moverAs(u, i)) ≡
estudiantes?(u)

  estudiantes? (moverE(u, i)) ≡
if murioAlgunHippie(campus?(u), agentes?(u), hippies?(u), moverEstudiante(i, u)) then
  if hayDosHippies?(obtener(i, moverEstudiante(i, u)), borrarTodosHippiesMuertos( campus?(u), agentes?(u),
    hippies?(u), moverEstudiante(i, u) )) then
    borrar(i, estudiantes?(u))
  else
    if algunHippieEncerradoPorEstudiantes(borrarTodosHippiesMuertos( campus?(u), agentes?(u),
      hippies?(u), moverEstudiante(i, u) ), moverEstudiante(i, u)) then
      convertirTodosHippies( borrarTodosHippiesMuertos( campus?(u), agentes?(u), hippies?(u),
        moverEstudiante(i, u) ), moverEstudiante(i, u) )
    else
      estudiantes?(u)
    fi
  fi
else
  if hayDosHippies?(obtener(i, moverEstudiante(i, u)), hippies?(u)) then
    borrar(i, estudiantes?(u))
  else
    if algunHippieEncerradoPorEstudiantes(hippies?(u), moverEstudiante(i, u)) then
      convertirTodosHippies( hippies?(u), moverEstudiante(i, u) )
    else
      estudiantes?(u)
    fi
  fi
fi

  estudiantes? (moverH(u, i)) ≡

```

```

if conviertoAlgunEstudiante?( moverHippie( $i, u$ ), estudiantes?( $u$ ) ) then
  borrarTodosEstudiantesConvertidos( moverHippie( $i, u$ ), estudiantes?( $u$ ) )
else
  estudiantes?( $u$ )
fi

sanciones? : uni  $u \times$  placa  $p \longrightarrow \text{nat}$  {def?( $p$ , agentes?( $u$ ))}
capturas? : uni  $u \times$  placa  $p \longrightarrow \text{nat}$  {def?( $p$ , agentes?( $u$ ))}
sanciones? ( $pl$ , nuevaUni( $c, dAs$ ))  $\equiv$ 
0

sanciones? ( $pl$ , agregarE( $u, i, e, p$ ))  $\equiv$ 
if hayUnAgenteAlrededor?( $p$ , agentes?( $u$ ))  $\wedge$ 
movRestringido(  $p$ , campus?( $u$ ), agentes?( $u$ ), hippies?( $u$ ), definir( $i, \langle e, p \rangle$ , estudiantes?( $u$ )) ) then
  if  $pl \in$  agentesAlrededor( $p$ , agentes?( $u$ )) then  $1+$  sanciones?( $pl, u$ ) else sanciones?( $pl, u$ ) fi
else
  sanciones?( $pl, u$ )
fi

sanciones? ( $pl$ , agregarH( $u, i, p$ ))  $\equiv$ 
if algunEstudianteEncerradoPorAgentes?( campus?( $u$ ), agentes?( $u$ ), definir( $i, p$ , hippies?( $u$ )), estudiantes?( $u$ ) )
then
  if  $pl \in$  agentesQueAtrapanEstudiantes( campus?( $u$ ), agentes?( $u$ ), definir( $i, p$ , hippies?( $u$ )), estudiantes?( $u$ ) )
  then
     $1+$  sanciones?( $pl, u$ )
  else
    sanciones?( $pl, u$ )
  fi
else
  sanciones?( $pl, u$ )
fi

sanciones? ( $pl$ , moverAs( $u, p$ ))  $\equiv$ 
if algunEstudianteEncerradoPorAgentes?( campus?( $u$ ), moverAgente( $pl, u$ ), hippies?( $u$ ), estudiantes?( $u$ ) ) then
  if  $pl \in$  agentesQueAtrapanEstudiantes( campus?( $u$ ), moverAgente( $pl, u$ ), hippies?( $u$ ), estudiantes?( $u$ ) ) then
     $1+$  sanciones?( $pl, u$ )
  else
    sanciones?( $pl, u$ )
  fi
else
  sanciones?( $pl, u$ )
fi

sanciones? ( $pl$ , moverH( $u, i$ ))  $\equiv$ 
if algunEstudianteEncerradoPorAgentes?( campus?( $u$ ), agentes?( $u$ ), moverHippie( $i, u$ ), estudiantes?( $u$ ) ) then
  if  $pl \in$  agentesQueAtrapanEstudiantes( campus?( $u$ ), agentes?( $u$ ), moverHippie( $i, u$ ), estudiantes?( $u$ ) ) then
     $1+$  sanciones?( $pl, u$ )
  else
    sanciones?( $pl, u$ )
  fi
else
  sanciones?( $pl, u$ )
fi

sanciones? ( $pl$ , moverE( $u, i$ ))  $\equiv$ 

```

```

if algunEstudianteEncerradoPorAgentes?( campus?(u), agentes?(u), hippies?(u), moverEstudiante(i,u) ) then
    if pl ∈ agentesQueAtrapanEstudiantes( campus?(u), agentes?(u), hippies?(u), moverEstudiante(i,u) ) then
        1+ sanciones?(pl,u)
    else
        sanciones?(pl,u)
    fi
else
    sanciones?(pl,u)
fi

```

Otras Operaciones

```

    capturar(p, cAs) ≡
if ( p ∈ posiciones4Vecinas( obtener( dameUno( claves(cAs) ) ) ) ) then
    Ag( ⟨ Π1(cAs), darCaptura( seg( dameUno(cAs) ) ), Π3( dameUno(cAs) ) ), capturar( p, sinUno(cAs) ) )
else
    Ag( dameUno(cAs), capturar( p, sinUno(cAs) ) )
fi

```

```

    sancionar(p, cAs) ≡
if ( p ∈ posiciones4Vecinas( Π3( dameUno ( cAs ) ) ) ) then
    Ag( ⟨ Π1(cAs), darSancion( seg( dameUno(cAs) ) ), Π3( dameUno(cAs) ) ), capturar( p, sinUno(cAs) ) )
else
    Ag( dameUno(cAs), sancionar( p, sinUno(cAs) ) )
fi

```

```

    queTipoHay(p, tripla, u) ≡
if ( fila?(p) ≥ alto?(u) ∨ col?(p) ≥ ancho?(u) ) then
    FueraDeRango
else
    if ( p ∈ obstaculos?(u) ) then
        Obstaculo
    else
        if ( p ∈ damePosicionesAs( Π1(tripla) ) ) then
            Agente
        else
            if ( p ∈ damePosicionesH( Π2(tripla) ) ) then
                Hippie
            else
                if ( p ∈ damePosicionesEst( Π3(tripla) ) ) then Estudiante else ∅ fi
            fi
        fi
    fi
fi

```

```

    cuatroVecinosShort(p, tripla, u) ≡
    cuatroVecinos( p, tripla, u, posiciones4Vecinas(p) )

```

```

    posiciones4Vecinas(p) ≡
if (fila?(p) = 0 ) ∧ (col?(p) = 0) then
    { nuevaPos(1,0), nuevaPos(0,1) }
else
    if fila?(p) = 0 then
        { nuevaPos(0, col?(p)-1), nuevaPos(0, col?(p)+1), pos(1, col?(p)) }
    else
        if col?(p) = 0 then
            { nuevaPos(fila?(p)-1, 0), nuevaPos(fila?(p)+1, 0), nuevaPos(fila?(p), 1) }
        else
            { nuevaPos(fila?(p), col?(p)-1), nuevaPos(fila?(p), col?(p)+1),
              nuevaPos(fila?(p)+1, col?(p)), nuevaPos(fila?(p)-1, col?(p) ) }
        fi
    fi
fi

```



```

    cuatroVecinos(p, tripla, u, p4v)  $\equiv$ 
if ( vacio?(p4v) ) then
    vacio
else
    Ag( queTipoHay( dameUno(p4v), tripla, u), cuatroVecinos( p, tripla, u, sinUno(p4v) ) )
fi

    queSituacion(mcT)  $\equiv$ 
    movRest?(mcT)  $\cup$  unAgente?(mcT)  $\cup$  dosHippies?(mcT)  $\cup$  cuatroEstudiantes?(mcT)

    damePosicionesAs(dAs)  $\equiv$ 
if vacio = dAs then
     $\emptyset$ 
else
    Ag( obtener( dameUno(claves(dAs)), dAs ), damePosicionesAs( borrar(dameUno(claves(dAs)), dAs) ) )
fi

    damePosicionesEst(cEst)  $\equiv$ 
if vacio?(cEst) then  $\emptyset$  else Ag(  $\Pi_3$ ( dameUno(cEst) ), damePosicionesEst( sinUno(cEst) ) ) fi

    damePosicionesH(cH)  $\equiv$ 
if vacio?(cH) then  $\emptyset$  else Ag(  $\Pi_2$ ( dameUno(cH) ), damePosicionesH( sinUno(cH) ) ) fi

    moverEstudianteYChequearSituaciones(i, u)  $\equiv$ 
if ( vacio?( dameEstudiante( i, estudiantes?(u) ) ) ) then
     $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$ 
else
    if  $\neg$ (  $\Pi_1$ ( dameEstudiante(i, u) )  $\in$  dirLibres( dameEstudiantePos(i, estudiantes?(u) ), u ) ) then
     $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$ 
    else
    chequearSituacionShort( agregarEstudianteTripla(  $\langle$  i, fin( dameEstudiante(i, u) ), mover(
    dameEstudiantePos(i, estudiantes?(u) ),  $\Pi_1$ ( dameEstudiante(i, u) )  $\rangle$  ),  $\langle$  agentes?(u), hippies?(u),
    sacarEstudianteId( i, estudiantes?(u) )  $\rangle$  ) )
    fi
fi

    dameEstudiante(i, cEst)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then  $\Pi_2$ ( dameUno(cEst) ) else dameEstudiante(i, sinUno(cEst) ) fi

    dameEstudiantePos(i, cEst)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then  $\Pi_3$ ( dameUno(cEst) ) else dameEstudiantePos( i, sinUno(cEst) ) fi

    sacarEstudianteId(i, cEst)  $\equiv$ 
if ( vacio?(cEst) ) then
     $\emptyset$ 
else
    if (  $\Pi_1$ ( dameUno(cEst) ) = i ) then
    sinUno(cEst)
    else
    Ag( dameUno(cEst), sacarEstudiante( i, sinUno(cEst) ) )
    fi
fi

    entradas?(an, al)  $\equiv$ 
if ( an = 0 ) then  $\emptyset$  else Ag( pos(0, an-1), Ag( pos(al-1, an-1), entradas(an-1, al) ) ) fi

    dirLibres(p, u)  $\equiv$ 
    dirNoOcupadas( dirValidas( p, ancho?(u), alto?(u) ), p, u )

    dirValidas(p, an, al)  $\equiv$ 

```

```
{n,s,e,o} - (if col?(p) = 0 then {o} else ∅ fi) - (if col?(p) = an-1 then {e} else ∅ fi) - (if fila?(p) = 0 then {n} else ∅ fi) - (if fila?(p) = al-1 then {s} else ∅ fi)
```

```
dirNoOcupadas( cDirs, p, u ) ≡
if (vacio(cDirs)) then
  ∅
else
  if ( mover( p, dameUno(cDirs) ) ∈ damePosicionesH( hippies?(u) ) ∨ mover( p, dameUno(cDirs) ) ∈ obstaculos?(u) ∨ mover( p, dameUno(cDirs) ) ∈ damePosicionesAs( agentes?(u) ) ∨ mover( p, dameUno(cDirs) ) ∈ damePosicionesEst( estudiantes?(u) ) ) then
    dirNoOcupadas( sinUno(cDirs), p, u )
  else
    Ag( dameUno(cDirs), dirNoOcupadas( sinUno(cDirs), p, u ) )
  fi
fi
```

```
posNoOcupadas(cPos, u) ≡
if ( vacio?(cPos) ) then
  ∅
else
  if ¬ dameUno(cPos) ∈ ( damePosicionesAs( agentes?(u) ) ∪ damePosicionesEst( estudiantes?(u) ) ∪ damePosicionesH( hippies?(u) ) ∪ obstaculos?(u) ) then
    Ag( dameUno(cPos), posNoOcupadas(sinUno(cPos), u) )
  else
    posNoOcupadas(sinUno(cPos), u)
  fi
fi
```

```
moverAgenteYChequearSituacion(i, u) ≡
if ( vacio?( hippies?(u) ) ∧ ( dameAgentePos( i, agentes?(u) ) ∈ entradas?( alto?(u), ancho?(u) ) ) ) ∨ inactivo?(dameAgente(i, u) ) then
  > agentes?(u), hippies?(u), estudiantes?(u) <
else
  if vacio?( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ), u ) ) then
    > agentes?(u), hippies?(u), estudiantes?(u) <
  else
    chequearSituacionShort( agregarAgenteTripla( ) i, dameAgente( i, agentes?(u) ), mover( dameAgentePos(i, agentes?(u) ), dameUno( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ), u ) ) ) <, > sacarAgenteId(i, agentes?(u) ), hippies?(u), estudiantes?(u) < )
  fi
fi
```

```
posibleMovAs( agente, p, u ) ≡
if ( inactivo?(agente) ) then
  ∅
else
  if ( vacio?( hippies?(u) ) ) then
    if ( dirLibres(agente, u) ∩ direccionesOptimas( dameUno( cPosMasCercanaShort( p, posNoOcupadas(entradas(an, al), u) ) ) ) = ∅ ) then
      dirLibres( p, u )
    else
      dirLibres( p, u ) ∩ direccionesOptimas( dameUno( cPosMasCercanaShort( p, posNoOcupadas(entradas(an, al), u) ) ) )
    fi
  else
    if dirLibres(p, u) ∩ direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) ) = ∅ then
      dirLibres(p, u)
    else
      dirLibres(p, u) ∩ direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) )
    fi
  fi
fi
```

```

    dameAgente(i, cAs)  $\equiv$ 
if (  $\Pi_1(\text{dameUno}(cAs)) = i$  ) then  $\Pi_2(\text{dameUno}(cAs))$  else dameAgente(  $i, \text{sinUno}(cAs)$  ) fi

    dameAgentePos(i, cAs)  $\equiv$ 
if (  $\Pi_1(\text{dameUno}(cAs)) = i$  ) then  $\Pi_3(\text{dameUno}(cAs))$  else dameAgente(  $i, \text{sinUno}(cAs)$  ) fi

    sacarAgenteId(i, cAs)  $\equiv$ 
if ( vacio?(cAs) ) then
     $\emptyset$ 
else
    if (  $\Pi_1(\text{dameUno}(cAs)) = i$  ) then
    sinUno(cAs)
    else
    Ag( dameUno(cAs), sacarAgente(  $i, \text{sinUno}(cAs)$  ) )
fi
fi

    moverHippieYChequearSituacion(i, u)  $\equiv$ 
if ( vacio?( posibleMovH( dameHippiePos( $i, u$ ),  $u$ ) ) ) then
     $\langle \text{agentes?}(u), \text{hippies?}(u), \text{estudiantes?}(u) \rangle$ 
else
    chequearSituacionShort( agregarHippieTripla(  $\langle i, \text{mover}(\text{dameHippiePos}(i, u), \text{dameUno}(\text{posibleMovH}(\text{dameHippiePos}(i, u), u)) \rangle, \langle \text{agentes?}(u), \text{sacarHippieId}(i, \text{hippies?}(u)), \text{estudiantes?}(u) \rangle \rangle$ 
fi

    posibleMovH (p, u)  $\equiv$ 
if vacio?( estudiantes?(u) ) then
     $\emptyset$ 
else
    if dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, damePosicionesEst( estudiantes?(u) ) ) ) ) =  $\emptyset$  then
    dirLibres(p, u)
    else
    dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, damePosicionesEst( estudiantes?(u) ) ) ) )
fi
fi

    dameHippiePos(i, cH)  $\equiv$ 
if prim(dameUno(cH)) =  $i$  then seg(dameUno(cH)) else dameHippiePos( $i, \text{sinUno}(cH)$ ) fi

    sacarHippieId(i, cH)  $\equiv$ 
if vacio(cH) then
     $\emptyset$ 
else
    if prim(dameUno(cH)) =  $i$  then sinUno(cH) else Ag(dameUno(cH), sacarHippie( $i, \text{sinUno}(cH)$ ))) fi
fi

    chequearSituacionShort(tripla, u)  $\equiv$ 
    chequearSituacion( tripla, tripla, u)

    agAgenteTripla(a,  $\langle cAs, cH, cEst \rangle$  )  $\equiv$ 
     $\langle \text{Ag}(a, cAs), cH, cEst \rangle$ 

    agHippieTripla(h,  $\langle cAs, cH, cEst \rangle$  )  $\equiv$ 
     $\langle cAs, \text{Ag}(h, cH), cEst \rangle$ 

    agEstudianteTripla(e,  $\langle cAs, cH, cEst \rangle$  )  $\equiv$ 
     $\langle cAs, cH, \text{Ag}(e, cEst) \rangle$ 

```

```

    estudiantesAdyacentesPos(p, cEst)  $\equiv$ 
if dameEstudiantePos( dameUno(cEst) )  $\in$  posiciones4Vecinas(p) then
    Ag(  $\Pi_2$ ( dameUno(cEst) ), estudiantesAdyacentesPos(p, sinUno(cEst) ) )
else
    estudiantesAdyacentesPos(p, sinUno(cEst))
fi

```

Cuando chequeamos los cambios de situaciones al moverse un agente, hippie o estudiante, primero vemos que pasa con los Agentes segun sus 4-vecinos, luego con los Hippies, y por ultimo con los Estudiantes. Cada uno de estos conjuntos no siguen ningun orden particular.

```

    chequearSituacion(  $\langle$  cAs, cH, cEst  $\rangle$ , triplaInfo, u )  $\equiv$ 
if vacio?(cH) then
    if vacio?(cEst) then
         $\langle$  cAs, cH, cEst  $\rangle$ 
    else
        if movRest, unAgente  $\subseteq$  queSituacion( cuatroVecinosShort( dameEstudiantePos( dameUno(cEst)), triplaInfo, u ) ) then
            if dosHippies  $\in$  queSituacion( cuatroVecinosShort( dameEstudiantePos( dameUno(cEst)), triplaInfo, u ) ) then
                agregarHippieTripla( convertirEstudianteAHippie(dameUno(cEst)) , chequearSituacion(  $\langle$  sancionar( dameEstudiantePos( dameUno(cEst) ), cAs), cH, sinUno(cEst)  $\rangle$ , triplaInfo, u ) )
            else
                agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle$  sancionar( dameEstudiantePos( dameUno(cEst) ), cAs), cH, sinUno(cEst)  $\rangle$ , triplaInfo, u ) )
            fi
        else
            if dosHippies  $\in$  queSituacion( cuatroVecinosShort(dameEstudiantePos(dameUno(cEst)), triplaInfo, u ) ) then
                agregarHippieTripla( convertirEstudianteAHippie(dameUno(cEst)), chequearSituacion(  $\langle$  cAs, cH, sinUno(cEst)  $\rangle$ , triplaInfo, u ) )
            else
                agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle$  cAs, cH, sinUno(cEst)  $\rangle$ , triplaInfo, u ) )
            fi
        fi
    fi
else
    if movRest, unAgente  $\subseteq$  queSituacion( cuatroVecinosShort( dameHippiePos( dameUno(cH) ), triplaInfo, u ) ) then
        chequearSituacion(  $\langle$  capturar(dameHippiePos(dameUno(cH)), cAs), sinUno(cH), cEst  $\rangle$ , sacarHippieTripla(dameUno(cH), triplaInfo, u ) )
    else
        if movRest, cuatroEstudiantes  $\subseteq$  queSituacion( cuatroVecinosShort( dameHippiePos( dameUno(cH) ), triplaInfo, u ) ) then
            agregarEstudianteTripla( convertirHippieAEst(dameUno(cH), cEst), chequearSituacion(  $\langle$  cAs, sinUno(cH), cEst  $\rangle$ , sacarHippieTripla(dameUno(cH), triplaInfo, u ) ) )
        else
            agregarHippieTripla( dameUno(cH), chequearSituacion(  $\langle$  cAs, sinUno(cH), cEst  $\rangle$ , triplaInfo, u ) )
        fi
    fi
fi

```

```

    convertirHippieAEst(h, cEst)  $\equiv$ 
 $\langle$   $\Pi_0$ (h), dameUno(estudiantesAdyacentesPos( $\Pi_1$ (h), cEst)),  $\Pi_1$ (h)  $\rangle$ 

```

```

    convertirEstAHippie(e)  $\equiv$ 
 $\langle$   $\Pi_0$ (e),  $\Pi_2$ (e)  $\rangle$ 

```

```

    sacarHippieTripla(h,  $\langle$  cAs, cH, cEst  $\rangle$ )  $\equiv$ 
 $\langle$  cAs, sacarHippieId( $\Pi_0$ (h), cH), cEst  $\rangle$ 

```

```

    dondeEstaAgente(i, u)  $\equiv$ 

```

dameAgentePos(i , agentes?(u))

 dondeEstaHippie(i , u) \equiv
dameHippiePos(i , hippies?(u))

 dondeEstaEstudiante(i , u) \equiv
dameEstudiantePos(i , estudiantes?(u))

 masVigilante(u) \equiv
agenteConMenorPlacaShort(losMasVigilantesShort(agentes?(u)))

 losMasVigilantes(cAs , $maxC$) \equiv
if vacio?(cAs) **then**
 \emptyset
else
 if hippiesCapturados(Π_1 (dameUno(cAs))) = $maxC$ **then**
 Ag(dameUno(cAs), losMasVigilantes(sinUno(cAs), $maxC$))
 else
 losMasVigilantes(sinUno(cAs), $maxC$)
 fi
fi

 losMasVigilantesShort(cAs) \equiv
losMasVigilantes(cAs , maxCapturas(cAs))

 agenteConMenorPlaca(cAs , $minN$) \equiv
if vacio?(cAs) **then**
 \emptyset
else
 if numPlaca(Π_1 (dameUno(cAs))) = $minN$ **then**
 Ag(dameUno(cAs), agenteConMenorPlaca(sinUno(cAs), $minN$))
 else
 agenteConMenorPlaca(sinUno(cAs), $minN$)
 fi
fi

 agenteConMenorPlacaShort(cAs) \equiv
agenteConMenorPlaca(cAs , minPlaca(cAs))

 maxCapturas(cAs) \equiv
if vacio?(cAs) **then** 0 **else** max(hippiesCapturados(Π_1 (dameUno(cAs))), maxCapturas(sinUno(cAs))) **fi**

 minPlaca(cAs) \equiv
if vacio?(cAs) **then** 0 **else** min(numPlaca(Π_1 (dameUno(cAs))), minPlaca(sinUno(cAs))) **fi**

 cuantosHippies(u) \equiv
hippies?(u)

 cuantosEstudiantes(u) \equiv
estudiantes?(u)

Fin TAD

2. TAD AGENTE

TAD AGENTE

géneros as

exporta as, Generadores, Observadores Basicos, inactivo?

usa NAT, BOOL

igualdad observacional

$$(\forall a, a' : \text{as}) \left(a =_{\text{obs}} a' \iff \left(\text{numPlaca}(a) =_{\text{obs}} \text{numPlaca}(a') \wedge \text{hippiesAtrapados}(a) =_{\text{obs}} \text{hippie-} \right. \right. \\ \left. \left. \text{sAtrapados}(a') \wedge \text{numSanciones}(a) =_{\text{obs}} \text{numSanciones}(a') \right) \right)$$

observadores básicos

numPlaca : as \longrightarrow nat

hippiesCapturados : as \longrightarrow nat

numSanciones : as \longrightarrow nat

generadores

nuevoAs : nat \longrightarrow as

darCaptura : as a \longrightarrow as

darSancion : as \longrightarrow as

otras operaciones

inactivo? : as \longrightarrow bool

axiomas $\forall n: \text{nat} \forall a: \text{as}$

Observadores Basicos

numPlaca(nuevoAs(n)) \equiv

n

numPlaca(darCaptura(a)) \equiv

numPlaca(a)

numPlaca(darSancion(a)) \equiv

numPlaca(a)

hippiesCapturados(nuevoAs(n)) \equiv

0

hippiesCapturados(darCaptura(a)) \equiv

1+ hippiesCapturados(a)

hippiesCapturados(darSancion(a)) \equiv

hippiesCapturados(a)

numSanciones(nuevoAs(n)) \equiv

0

numSanciones(darCaptura(a)) \equiv

numSanciones(a)

numSanciones(darSancion(a)) \equiv

1+ numSanciones(a)

Otras Operaciones

inactivo?(a) \equiv

if numSanciones(a) > 3 **then** true **else** false **fi**

Fin TAD

3. TAD POSICION

TAD POSICION

géneros pos

exporta pos, Generadores, Observadores Basicos, Otras Operaciones

usa NAT, BOOL, CONJUNTO(α), DIR

igualdad observacional

$(\forall p, p' : \text{pos}) (p =_{\text{obs}} p' \iff (\text{fila?}(p) =_{\text{obs}} \text{fila?}(p') \wedge \text{col?}(p) =_{\text{obs}} \text{col?}(p')))$

observadores básicos

$\text{fila?} : \text{pos} \rightarrow \text{nat}$

$\text{col?} : \text{pos} \rightarrow \text{nat}$

generadores

$\text{nuevaPos} : \text{nat } f \times \text{nat } c \rightarrow \text{pos} \quad \{(f \geq 0) \wedge (c \geq 0)\}$

otras operaciones

$\text{direccionesOptimas} : \text{pos} \times \text{pos} \rightarrow \text{conj}(\text{dir})$

$\text{cPosMasCercana} : \text{pos} \times \text{conj}(\text{pos}) \times \text{nat} \rightarrow \text{conj}(\text{pos})$

$\text{cPosMasCercanaShort} : \text{pos} \times \text{conj}(\text{pos}) \rightarrow \text{conj}(\text{pos})$

$\text{menorDistancia} : \text{pos} \times \text{conj}(\text{pos}) \text{ cPos} \rightarrow \text{nat} \quad \{\neg \text{Vacía?}(\text{cPos})\}$

$\text{dist} : \text{pos} \times \text{pos} \rightarrow \text{nat}$

$\text{posiciones4Vecinas} : \text{pos} \rightarrow \text{conj}(\text{pos})$

$\text{mover} : \text{pos } p \times \text{dir} \rightarrow \text{pos} \quad \{\neg(\text{col?}(p) = 0 \wedge (o) = \text{dir}) \wedge \neg(\text{fila?}(p) = 0 \wedge (n) = \text{dir})\}$

axiomas $\forall p, p' : \text{pos} \wedge \forall f, c, \text{menorDist} : \text{nat} \forall d : \text{dir}$

$\text{fila?}(\text{nuevaPos}(f, c)) \equiv$

f

$\text{col?}(\text{nuevaPos}(f, c)) \equiv$

c

$\text{direccionesOptimas}(p, p') \equiv$

$\{n, s, e, o\} - (\text{if } \text{fila?}(p) \leq \text{fila?}(p') \text{ then } \{n\} \text{ else } \emptyset \text{ fi})$

$- (\text{if } \text{fila?}(p) \geq \text{fila?}(p') \text{ then } \{s\} \text{ else } \emptyset \text{ fi})$

$- (\text{if } \text{col?}(p) \leq \text{col?}(p') \text{ then } \{o\} \text{ else } \emptyset \text{ fi})$

$- (\text{if } \text{col?}(p) \geq \text{col?}(p') \text{ then } \{e\} \text{ else } \emptyset \text{ fi})$

$\text{PosMasCercanaShort}(p, \text{cPos}) \equiv$

$\text{cPosMasCercana}(p, \text{cPos}, \text{menorDistancia}(p, \text{cPos}))$

$\text{cPosMasCercana}(p, \text{cPos}, \text{menorDist}) \equiv$

if $\text{vacío?}(\text{cPos})$ **then**

\emptyset

else

if $\text{dist}(p, \text{dameUno}(\text{cPos})) = \text{menorDist}$ **then**

$\text{Ag}(p, \text{cPosMasCercana}(p, \text{sinUno}(\text{cPos}), \text{menorDist}))$

else

$\text{cPosMasCercana}(p, \text{sinUno}(\text{cPos}), \text{menorDist})$

fi

fi

$\text{menorDistancia}(p, \text{cPos}) \equiv$

```

if  $\neg$  vacia?(cpos) then
    min(dist(p, dameUno(cPos)), menorDistancia(p, sinUno(cpos)))
else
    dist(p, dameUno(cPos))
fi

    dist(p,p')  $\equiv$ 
    (max(fila?(p), fila?(p')) - min(fila?(p), fila?(p')) + (max(col?(p), col?(p')) - min(col?(p), col?(p')))

    mover(p, d)  $\equiv$ 
if (d = n) then
    nuevaPos(fila?(p)-1, col?(p))
else
    if (d=s) then
        nuevaPos(fila?(p)+1, col?(p))
    else
        if ( dir = o ) then nuevaPos(fila?(p), col?(p)-1) else nuevaPos(fila?(p), col?(p)+1) fi
    fi
fi

Fin TAD

```

4. TAD DIRECCION

TAD DIRECCION

géneros dir

exporta dir, Generadores

usa

igualdad observacional

$$\left(\begin{array}{l} n =_{\text{obs}} n \wedge s =_{\text{obs}} s \wedge e =_{\text{obs}} e \wedge o =_{\text{obs}} o \wedge \neg(s =_{\text{obs}} n) \wedge \neg(s =_{\text{obs}} o) \wedge \neg(s =_{\text{obs}} e) \wedge \neg(n =_{\text{obs}} s) \wedge \neg(n \\ =_{\text{obs}} e) \wedge \neg(n =_{\text{obs}} o) \wedge \neg(e =_{\text{obs}} s) \wedge \neg(e =_{\text{obs}} n) \wedge \neg(e =_{\text{obs}} o) \wedge \neg(o =_{\text{obs}} s) \wedge \\ \neg(o =_{\text{obs}} n) \wedge \neg(o =_{\text{obs}} e) \end{array} \right)$$

observadores básicos

generadores

n : \longrightarrow dir

s : \longrightarrow dir

e : \longrightarrow dir

o : \longrightarrow dir

otras operaciones

Fin TAD

5. TAD SITUACIONESRODEO

TAD SITUACIONESRODEO

géneros situ

exporta situ, Generadores

usa

igualdad observacional

$$\left(\begin{aligned} & \text{movRest} =_{\text{obs}} \text{movRest} \wedge \text{unAgente} =_{\text{obs}} \text{unAgente} \wedge \text{dosHippies} =_{\text{obs}} \text{dosHippies} \wedge \text{cuatroEstudian-} \\ & \text{tes} =_{\text{obs}} \text{cuatroEstudiantes} \wedge \neg(\text{movRest} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{movRest} =_{\text{obs}} \text{dosHippies}) \wedge \neg(\text{movRest} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{dosHippies}) \wedge \neg(\text{unAgente} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{dosHippies} \\ & =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{unAgente}) \wedge \\ & \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{dosHippies}) \end{aligned} \right)$$

observadores básicos

generadores

movRest : \longrightarrow situ

unAgente : \longrightarrow situ

dosHippies : \longrightarrow situ

cuatroEstudiantes : \longrightarrow situ

otras operaciones

Fin TAD

6. TAD TIPOELEM

TAD TIPOELEM

géneros tipo

exporta tipo, Generadores

usa

igualdad observacional

$$\left(\begin{aligned} & \text{Vacio} =_{\text{obs}} \text{Vacio} \wedge \text{Obstaculo} =_{\text{obs}} \text{Obstaculo} \wedge \text{Agente} =_{\text{obs}} \text{Agente} \wedge \text{Hippie} =_{\text{obs}} \text{Hippie} \wedge \text{Estudiante} \\ & =_{\text{obs}} \text{Estudiante} \wedge \text{FueraDeRango} =_{\text{obs}} \text{FueraDeRango} \quad // \text{ y son diferentes de manera cruzada} \end{aligned} \right)$$

observadores básicos

generadores

Vacio : \longrightarrow tipo

Obstaculo : \longrightarrow tipo

Agente : \longrightarrow tipo

Hippie : \longrightarrow tipo

Estudiante : \longrightarrow tipo

FueraDeRango : \longrightarrow tipo

otras operaciones

Fin TAD