

Trabajo Practico I, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

| | |
|--------------------------------|-----------|
| 1. TAD UNIVERSIDAD | 2 |
| 2. TAD AGENTE | 10 |
| 3. TAD POSICION | 11 |
| 4. TAD DIRECCION | 13 |
| 5. TAD SITUACIONESRODEO | 13 |
| 6. TAD TIPOELEM | 14 |

1. TAD UNIVERSIDAD

TAD UNIVERSIDAD

géneros uni

exporta uni, Generadores, Observadores Basicos

usa NAT, CONJU(α), BOOL, TUPLA($\alpha_1, \dots, \alpha_n$)

igualdad observacional

$$(\forall u, u' : \text{uni}) \left(u =_{\text{obs}} u' \iff \left(\begin{array}{l} \text{alto?}(u) =_{\text{obs}} \text{alto?}(u') \wedge \text{ancho?}(u) =_{\text{obs}} \text{ancho?}(u') \wedge \text{obstaculos?}(u) \\ =_{\text{obs}} \text{obstaculos?}(u') \wedge \text{agentes?}(u) =_{\text{obs}} \text{agentes?}(u') \wedge \text{estudian-} \\ \text{tes?}(u) =_{\text{obs}} \text{estudiantes?}(u') \wedge \text{hippies?}(u) =_{\text{obs}} \text{hippies?}(u') \end{array} \right) \right)$$

observadores básicos

alto? : uni \longrightarrow nat

ancho? : uni \longrightarrow nat

obstaculos? : uni \longrightarrow conj(*pos*)

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

hippies? : uni \longrightarrow conj($\langle \text{id}, \text{pos} \rangle$)

estudiantes? : uni \longrightarrow conj($\langle \text{id}, \text{est}, \text{pos} \rangle$)

generadores

TAD id ES Nat TAD hip ES Id TAD est ES Secu(Dir)

nuevaUni : conj($\langle \text{as} \times \text{pos} \rangle$) \times nat \times nat \times conj(*pos*) \longrightarrow uni

agregarE : uni \times id \times est \times pos \longrightarrow uni

agregarH : uni \times hip \times pos \longrightarrow uni

moverAS : uni \times id \longrightarrow uni

moverH : uni \times id \longrightarrow uni

moverE : uni \times id \longrightarrow uni

otras operaciones

capturar : pos \times conj($\langle \text{id} \times \text{agente} \times \text{pos} \rangle$) \longrightarrow conj($\langle \text{agente}, \text{pos} \rangle$)

sancionar : pos \times conj($\langle \text{id} \times \text{agente} \times \text{pos} \rangle$) \longrightarrow conj($\langle \text{agente}, \text{pos} \rangle$)

queTipoHay : pos \times $\langle \text{cAs} \times \text{cH} \times \text{cEst} \rangle$ \longrightarrow tipo

cuatroVecinosShort : pos \times $\langle \text{cAs} \times \text{cH} \times \text{cEst} \rangle \times$ uni \longrightarrow multiconj(tipo)

cuatroVecinos : pos \times $\langle \text{cAs} \times \text{cH} \times \text{cEst} \rangle \times$ uni \times conj(*pos*) \longrightarrow multiconj(tipo)

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

damePosicionesAs : conj($\langle \text{id} \times \text{agente} \times \text{pos} \rangle$) \longrightarrow conj(*pos*)

damePosicionesEst : conj($\langle \text{id} \times \text{agente} \times \text{pos} \rangle$) \longrightarrow conj(*pos*)

damePosicionesH : conj($\langle \text{id} \times \text{pos} \rangle$) \longrightarrow conj(*pos*)

moverEstudianteYChequearSituaciones : id *is* \times uni *u* \longrightarrow $\langle \text{cAs}, \text{cH}, \text{cEst} \rangle$
 $\{ \exists t \in \text{estudiantes?}(u) / \Pi_0(t) = is \}$

dameEstudiante : id *is* \times conj($\langle \text{id} \times \text{est} \times \text{pos} \rangle$) conjEst \longrightarrow est
 $\{ \exists t \in \text{conjEst} / \Pi_0(t) = is \}$

dameEstudiantePos : id *is* \times conj($\langle \text{id} \times \text{est} \times \text{pos} \rangle$) conjEst \longrightarrow pos
 $\{ \exists t \in \text{conjEst} / \Pi_0(t) = is \}$

sacarEstudianteId : id *is* \times conj($\langle \text{id} \times \text{est} \times \text{pos} \rangle$) conjEst \longrightarrow conj($\langle \text{id}, \text{est}, \text{pos} \rangle$)
 $\{ \exists t \in \text{conjEst} / \Pi_0(t) = is \}$

entradas? : nat \times nat \longrightarrow conj(*pos*)

dirLibres : pos \times uni \longrightarrow conj(*dir*)

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

axiomas $\forall \text{cAs}: \text{conj}(\langle \text{id}, as, pos \rangle) \forall \text{cH}: \text{conj}(\langle \text{id}, pos \rangle) \forall \text{cEst}: \text{conj}(\langle \text{id}, est, pos \rangle) \forall \text{cDirs}: \text{conj}(\text{dir})$
 $\forall \text{al}, \text{an}, \text{maxC}, \text{minN}: \text{nat} \forall \text{cObs}: \text{conj}(\text{pos}) \forall u: \text{uni} \forall i: \text{id} \forall e: \langle \text{id}, est, pos \rangle \forall \text{agente}: \langle \text{id}, as, pos \rangle \forall h: \langle \text{id}, pos \rangle \forall p: \text{pos} \forall p4v: \text{conj}(\text{pos}) \forall \text{tripla}, \text{triplaInfo}: \langle \text{conj}(\langle \text{id}, as, pos \rangle), \text{cH} \text{ conj}(\langle \text{id}, pos \rangle), \text{conj}(\langle \text{id}, est, pos \rangle) \rangle \forall \text{mcT}: \text{multiconj}(\text{tipo})$

Observadores Basicos

$\text{alto?}(\text{nuevaUni}(\text{cAs}, \text{al}, \text{an}, \text{cObs})) \equiv$
 al
 $\text{alto?}(\text{agregarE}(u, i, e, p)) \equiv$
 $\text{alto?}(u)$
 $\text{alto?}(\text{agregarH}(u, i, p)) \equiv$
 $\text{alto?}(u)$

```

    alto? (moverAs(u, i)) ≡
alto?(u)
    alto? (moverH(u, i)) ≡
alto?(u)
    alto? (moverE(u, i)) ≡
alto?(u)
    ancho? (nuevaUni(cAs, al, an, cObs)) ≡
an
    ancho? (agregarE(u, i, e, p)) ≡
ancho?(u)
    ancho? (agregarH(u, i, p)) ≡
ancho?(u)
    ancho? (moverAs(u, i)) ≡
ancho?(u)
    ancho? (moverH(u, i)) ≡
ancho?(u)
    ancho? (moverE(u, i)) ≡
ancho?(u)
    obstaculos? (nuevaUni(cAs, al, an, cObs)) ≡
cObs
    obstaculos? (agregarE(u, i, e, p)) ≡
obstaculos?(u)
    obstaculos? (agregarH(u, i, p)) ≡
obstaculos?(u)
    obstaculos? (moverAs(u, i)) ≡
obstaculos?(u)
    obstaculos? (moverH(u, i)) ≡
obstaculos?(u)
    obstaculos? (moverE(u, i)) ≡
obstaculos?(u)
    agentes? (nuevaUni(cAs, al, an, cObs)) ≡
cAs
    agentes? (agregarE(u, i, e, p)) ≡
Π1( chequearSituacionShort( agregarEstudianteTripla( ⟨ i, e, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    agentes? (agregarH(u, i, pos)) ≡
Π1( chequearSituacionShort( agregarHippieTripla( ⟨ i, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    agentes? (moverAs(u, i)) ≡
Π1( moverAgenteYSancionarYCapturar(i, u) )
    agentes? (moverE(u, i)) ≡
Π1( moverEstudiante(i, u) )
    agentes? (moverH(u, i)) ≡
Π1( moverHippie(i, u) )
    hippies? (nuevaUni(cAs, al, an, cObs)) ≡
∅
    hippies? (agregarE(u, i, e, p)) ≡
Π2( chequearSituacionShort( agregarEstudianteTripla( ⟨ i, e, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    hippies? (agregarH(u, i, p)) ≡

```

```

Π2( chequearSituacionShort( agregarHippieTripla( ⟨ i, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    hippies? (moverAs(u, i)) ≡
Π2( moverAgenteYSancionarYCapturar(i, u) )
    hippies? (moverE(u, i)) ≡
Π2( moverEstudiante(i, u) )
    hippies? (moverH(u, i)) ≡
Π2( moverHippie(i, u) )
    estudiantes? (nuevaUni(cAs, al, an, cObs)) ≡
∅
    estudiantes? (agregarE(u, i, e, p)) ≡
Π3( chequearSituacionShort( agregarEstudianteTripla( ⟨ i, e, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    estudiantes? (agregarH(u, i, p)) ≡
Π3( chequearSituacionShort( agregarHippieTripla( ⟨ i, p ⟩,
⟨ agentes?(u), hippies?(u), estudiantes?(u) ⟩ ) ) )
    estudiantes? (moverAs(u, i)) ≡
Π3( moverAgenteYSancionarYCapturar(i, u) )
    estudiantes? (moverE(u, i)) ≡
Π3( moverEstudiante(i, u) )
    estudiantes? (moverH(u, i)) ≡
Π3( moverHippie(i, u) )
    Otras Operaciones
    capturar(p, cAs) ≡
if ( p ∈ posiciones4Vecinas( Π3( dameUno( 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
    cuatroVecinosShort(p, tripla, u) ≡
cuatroVecinos( p, tripla, u, posiciones4Vecinas(p) )
    cuatroVecinos(p, tripla, u, p4v) ≡

```

```

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(cAs)  $\equiv$ 
if vacio?(cAs) then  $\emptyset$  else Ag(  $\Pi_2$ ( dameUno(cAs) ), damePosicionesAs( sinUno(cAs) ) ) 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
    { agentes?(u), hippies?(u), estudiantes?(u) }
else
if  $\neg$ (  $\Pi_1$ ( dameEstudiante(i, u) )  $\in$  dirLibres( dameEstudiantePos(i, estudiantes?(u) ), u ) ) then
    { agentes?(u), hippies?(u), estudiantes?(u) }
else
    chequearSituacionShort( agregarEstudianteTripla( { i, fin( dameEstudiante(i, u) ), mover(
    dameEstudiantePos(i, estudiantes?(u) ),  $\Pi_1$ ( dameEstudiante(i, u) ) } ), { agentes?(u), hippies?(u),
    sacarEstudianteId( i, estudiantes?(u) ) } ) )
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  $\emptyset$  fi ) - ( if col?(p) = an-1 then {e} else  $\emptyset$  fi ) - ( if fila?(p) = 0
then {n} else  $\emptyset$  fi ) - ( if fila?(p) = al-1 then {s} else  $\emptyset$  fi )

    dirNoOcupadas( cDirs, p, u )  $\equiv$ 
if ( vacio(cDirs) ) then
     $\emptyset$ 
else
if ( mover( p, dameUno(cDirs) )  $\in$  damePosicionesH( hippies?(u) )  $\vee$  mover( p, dameUno(cDirs) )
     $\in$  obstaculos?(u)  $\vee$  mover( p, dameUno(cDirs) )  $\in$  damePosicionesAs( agentes?(u) )  $\vee$  mover( p,
    dameUno(cDirs) )  $\in$  damePosicionesEst( estudiantes?(u) ) ) then
    dirNoOcupadas( sinUno(cDirs), p, u )
else
    Ag( dameUno(cDirs), dirNoOcupadas( sinUno(cDirs), p, u ) )
fi
fi

```

```

    posNoOcupadas(cPos, u)  $\equiv$ 
if ( vacio?(cPos) ) then
     $\emptyset$ 
else
    if  $\neg$  dameUno(cPos)  $\in$  ( damePosicionesAs( agentes?(u) )  $\cup$  damePosicionesEst( estudiantes?(u) )  $\cup$  dame-
    PosicionesH( hippies?(u) )  $\cup$  obstaculos?(u) ) then
        Ag( dameUno(cPos), posNoOcupadas(sinUno(cPos), u) )
    else
        posNoOcupadas(sinUno(cPos), u)
    fi
fi
    moverAgenteYChequearSituacion(i, u)  $\equiv$ 
if ( vacio?( hippies?(u) )  $\wedge$  ( dameAgentePos( i, agentes?(u) )  $\in$  entradas?( alto?(u), ancho?(u) ) )  $\vee$  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)  $\equiv$ 
if ( inactivo?(agente) ) then
     $\emptyset$ 
else
    if ( vacio?( hippies?(u) ) ) then
        if ( dirLibres(agente, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, posNoOcupadas(
        entradas(an, al), u) ) ) ) =  $\emptyset$  ) then
            dirLibres( p, u)
        else
            dirLibres( p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort(p, posNoOcupadas(
            entradas(an, al), u) ) ) )
        fi
    else
        if dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) ) =  $\emptyset$  then
            dirLibres(p, u)
        else
            dirLibres(p, u)  $\cap$  direccionesOptimas( dameUno( cPosMasCercanaShort( p, hippies?(u) ) ) )
        fi
    fi
fi
    dameAgente(i, cAs)  $\equiv$ 
if (  $\Pi_1$ ( dameUno(cAs) ) = i ) then  $\Pi_2$ ( dameUno(cAs) ) else dameAgente( i, sinUno(cAs) ) fi
    dameAgentePos(i, cAs)  $\equiv$ 
if (  $\Pi_1$ (dameUno(cAs)) = i ) then  $\Pi_3$ ( dameUno(cAs) ) else dameAgente( i, sinUno(cAs) ) fi
    sacarAgenteId(i, cAs)  $\equiv$ 
if ( vacio?(cAs) ) then
     $\emptyset$ 
else
    if (  $\Pi_1$ ( dameUno(cAs) ) = i ) then
        sinUno(cAs)
    else
        Ag( dameUno(cAs), sacarAgente( i, sinUno(cAs) ) )
    fi
fi
    moverHippieYChequearSituacion(i, u)  $\equiv$ 

```

```

if ( vacio?( posibleMovH( dameHippiePos( $i, u$ ),  $u$ ) ) ) then
   $\langle$  agentes?( $u$ ), hippies?( $u$ ), estudiantes?( $u$ )  $\rangle$ 
else
  chequearSituacionShort( agregarHippieTripla(  $\langle$   $i$ , mover( dameHippiePos( $i, u$ ), dameUno( posibleMovH(
    dameHippiePos( $i, u$ ),  $u$ ) ) )  $\rangle$ ,  $\langle$  agentes?( $u$ ), sacarHippieId(  $i$ , hippies?( $u$ ) ), estudiantes?( $u$ )  $\rangle$  ) )
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( $i$ , 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( $i$ , sinUno( $cH$ ))) fi
fi
  chequearSituacionShort(tripla,  $u$ )  $\equiv$ 
  chequearSituacion( tripla, tripla,  $u$ )
  agAgenteTripla( $a, \langle$   $cAs, cH, cEst$   $\rangle$ )  $\equiv$ 
   $\langle$  Ag( $a, cAs$ ),  $cH, cEst$   $\rangle$ 
  agHippieTripla( $h, \langle$   $cAs, cH, cEst$   $\rangle$ )  $\equiv$ 
   $\langle$   $cAs$ , Ag( $h, cH$ ),  $cEst$   $\rangle$ 
  agEstudianteTripla( $e, \langle$   $cAs, cH, cEst$   $\rangle$ )  $\equiv$ 
   $\langle$   $cAs, cH$ , 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
  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) ), triplaInfo, u ) )
      else
        agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle$  sancionar( dameEstudiantePos( dameUno(cEst) ), cAs), cH, sinUno(cEst) ), 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) ), triplaInfo, u ) )
      else
        agregarEstudianteTripla( dameUno(cEst), chequearSituacion(  $\langle$  cAs, cH, sinUno(cEst) ), 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 ), 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 ), sacarHippieTripla(dameUno(cH), triplaInfo, u) ) )
    else
      agregarHippieTripla( dameUno(cH), chequearSituacion(  $\langle$  cAs, sinUno(cH), cEst ), 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 )  $\equiv$ 
 $\langle cAs, sacarHippieId(\Pi_0(h), cH), cEst \rangle$ 

  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( $\Pi_1$ (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( $\Pi_1$ (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( $\Pi_1$ (dameUno(cAs))), maxCapturas(sinUno(cAs))) fi
  minPlaca(cAs)  $\equiv$ 
if vacio?(cAs) then 0 else min(numPlaca( $\Pi_1$ (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
   $(\forall a, a' : \text{as}) \left( a =_{\text{obs}} a' \iff \left( \text{numPlaca}(a) =_{\text{obs}} \text{numPlaca}(a') \wedge \text{hippiesAtrapados}(a) =_{\text{obs}} \text{hippie-} \right. \right. \right.$ 
   $\left. \left. \text{sAtrapados}(a') \wedge \text{numSanciones}(a) =_{\text{obs}} \text{numSanciones}(a') \right) \right)$ 

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  → as

otras operaciones
    inactivo? : as  → bool

axiomas      ∀ n: nat ∀ a: as
Observadores Basicos
    numPlaca(nuevoAs(n)) ≡
n
    numPlaca(darCaptura(a)) ≡
numPlaca(a)
    numPlaca(darSancion(a)) ≡
numPlaca(a)
    hippiesCapturados(nuevoAs(n)) ≡
0
    hippiesCapturados(darCaptura(a)) ≡
1+ hippiesCapturados(a)
    hippiesCapturados(darSancion(a)) ≡
hippiesCapturados(a)
    numSanciones(nuevoAs(n)) ≡
0
    numSanciones(darCaptura(a)) ≡
numSanciones(a)
    numSanciones(darSancion(a)) ≡
1+ numSanciones(a)

Otras Operaciones
    inactivo?(a) ≡
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
    (∀p, p' : pos) (p =obs p' ⇔ (fila?(p) =obs fila(p') ∧ col?(p) =obs col(p')))

observadores básicos
    fila? : pos  → nat
    col? : pos  → nat

generadores
    nuevaPos : nat f × nat c  → pos                                     {(f ≥ 0) ∧ (c ≥ 0)}

otras operaciones
    direccionesOptimas : pos × pos  → conj(dir)
    cPosMasCercana : pos × conj(pos) × nat  → conj(pos)

```

```

cPosMasCercanaShort : pos × conj(pos) → conj(pos)
menorDistancia : pos × conj(pos) cPos → nat {¬ Vacía?(cPos)}
dist : pos × pos → nat
posiciones4Vecinas : pos → conj(pos)
mover : pos p × dir → pos {¬(col?(p) = 0 ∧ (o) = dir) ∧ ¬(fila?(p) = 0 ∧ (n) = dir)}

axiomas    ∀ p, p': pos ∧ ∀ f, c, menorDist: nat ∀ d: dir
    fila?(nuevaPos(f, c)) ≡
f
    col?(nuevaPos(f, c)) ≡
c
    direccionesOptimas(p, p') ≡
{n, s, e, o} - (if fila?(p) <= fila?(p') then {n} else ∅ fi)
- (if fila?(p) >= fila?(p') then {s} else ∅ fi)
- (if col?(p) <= col?(p') then {o} else ∅ fi)
- (if col?(p) >= col?(p') then {e} else ∅ fi)
    PosMasCercanaShort(p, cPos) ≡
cPosMasCercana(p, cPos, menorDistancia(p, cPos))
    cPosMasCercana(p, cPos, menorDist) ≡
if vacío?(cPos) then
    ∅
else
    if dist(p, dameUno(cPos)) = menorDist then
        Ag(p, cPosMasCercana(p, sinUno(cPos), menorDist))
    else
        cPosMasCercana(p, sinUno(cPos), menorDist)
    fi
fi
    menorDistancia(p, cPos) ≡
if ¬ vacía?(cpos) then
    min(dist(p, dameUno(cPos)), menorDistancia(p, sinUno(cpos)))
else
    dist(p, dameUno(cPos))
fi
    dist(p, p') ≡
(max(fila?(p), fila?(p')) - min(fila?(p), fila?(p')) + (max(col?(p), col?(p')) - min(col?(p), col?(p'))
    mover(p, d) ≡
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
    posiciones4Vecinas(p) ≡

```

```

if (fila?(p) = 0 )  $\wedge$  (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
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

$n : \longrightarrow \text{dir}$

$s : \longrightarrow \text{dir}$

$e : \longrightarrow \text{dir}$

$o : \longrightarrow \text{dir}$

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