

# Trabajo Practico I, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

## Índice

<b>1. TAD UNIVERSIDAD</b>	<b>2</b>
<b>2. TAD AGENTE</b>	<b>9</b>
<b>3. TAD TABLERO</b>	<b>10</b>

# 1. TAD UNIVERSIDAD

## TAD UNIVERSIDAD

**géneros** uni

**exporta** uni, Generadores, Observadores Basicos

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

### igualdad observacional

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

### observadores básicos

alto? : uni  $\rightarrow$  nat

ancho? : uni  $\rightarrow$  nat

obstaculos? : uni  $\rightarrow$  conj(*pos*)

agentes? : uni  $\rightarrow$  conj( $\langle as, pos \rangle$ )

hippies? : uni  $\rightarrow$  conj(*pos*)

estudiantes? : uni  $\rightarrow$  conj( $\langle est, pos \rangle$ )

### generadores

//id es nat //hip es id

nuevaUni : conj( $\langle as \times pos \rangle$ )  $\times$  nat  $\times$  nat  $\times$  conj(*pos*)  $\rightarrow$  uni

agregarE : uni  $\times$  id  $\times$  est  $\times$  pos  $\rightarrow$  uni

agregarH : uni  $\times$  hip  $\times$  pos  $\rightarrow$  uni

moverAS : uni  $\times$  id  $\rightarrow$  uni

moverH : uni  $\times$  id  $\rightarrow$  uni

moverE : uni  $\times$  id  $\rightarrow$  uni

### otras operaciones

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

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

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

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

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

queSituacion : multiconj(tipoElem)  $\rightarrow$  conj(structSituacionesRodeo)

damePosicionesAs : conj( $\langle id \times agente \times pos \rangle$ )  $\rightarrow$  conj(*pos*)

damePosicionesEst : conj( $\langle id \times agente \times pos \rangle$ )  $\rightarrow$  conj(*pos*)

damePosicionesH : conj( $\langle id \times pos \rangle$ )  $\rightarrow$  conj(*pos*)

moverEstudianteYChequearSituaciones : id  $\times$  uni  $\rightarrow$   $\langle cAs, cH, cEst \rangle$

dameEstudiante : id  $\times$  conj( $\langle i \times est \times pos \rangle$ )  $\rightarrow$  est  $\{i \in \text{conj}\}$

dameEstudiantePos : id  $\times$  conj( $\langle i \times est \times pos \rangle$ )  $\rightarrow$  pos  $\{i \in \text{conj}\}$

sacarEstudianteId : id  $\times$  conj( $\langle i \times est \times pos \rangle$ )  $\rightarrow$  conj( $\langle i, est, pos \rangle$ )

entradas? : nat  $\times$  nat  $\rightarrow$  conj(*pos*)

$\text{dirLibres} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dirValidas} : \text{pos} \times \text{nat} \times \text{nat} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dirNoOcupadas} : \text{conj}(\text{dir}) \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\{ \neg(\text{col?}(\text{pos}) = 0 \wedge \{o\} \in \text{cDirs}) \wedge \neg(\text{fla?}(\text{pos}) = 0 \text{ y } \{n\} \in \text{cDirs}) \}$   
 $\text{posNoOcupadas} : \text{conj}(\text{pos}) \times \text{uni} \longrightarrow \text{conj}(\text{pos})$   
 $\text{moverAgenteYChequearSituacion} : \text{id} \times \text{uni} \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle$   
 $\text{posibleMovAs} : \text{As} \times \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dameAgente} : \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \longrightarrow \text{as} \quad \{i \in \text{conj}\}$   
 $\text{dameAgentePos} : \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \longrightarrow \text{pos} \quad \{i \in \text{conj}\}$   
 $\text{sacarAgenteId} : \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \longrightarrow \text{conj}(\langle i, \text{as}, \text{pos} \rangle)$   
 $\text{moverHippieYChequearSituacion} : \text{id} \times \text{uni} \longrightarrow \langle \text{cAs}, \text{cH}, \text{cEst} \rangle$   
 $\text{posibleMovH} : \text{pos} \times \text{uni} \longrightarrow \text{conj}(\text{dir})$   
 $\text{dameHippiePos} : \text{id} \times \text{conj}(\langle i \times \text{pos} \rangle) \longrightarrow \text{pos} \quad \{i \in \text{conj}\}$   
 $\text{sacarHippieId} : \text{id} \times \text{conj}(\langle i \times \text{pos} \rangle) \longrightarrow \text{conj}(\langle i, \text{pos} \rangle)$   
 $\text{chequearSituacionShort} : \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \times \text{uni} \longrightarrow \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{chequearSituacion} \rangle$   
 $\text{chequearSituacion} : \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \times \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \rangle$   
 $\text{agAgenteTripla} : \langle i \times \text{as} \times \text{pos} \rangle \times \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \longrightarrow \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{agAgenteTripla} \rangle$   
 $\text{agHippieTripla} : \langle i \times \text{pos} \rangle \times \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \longrightarrow \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{agHippieTripla} \rangle$   
 $\text{agEstudianteTripla} : \langle i \times \text{est} \times \text{pos} \rangle \times \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \longrightarrow \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{agEstudianteTripla} \rangle$   
 $\text{estudiantesAdyacentesPos} : \text{pos} \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \longrightarrow \text{conj}(\text{est})$   
 $\text{convertirHippieAEst} : \langle \text{id} \times \text{pos} \rangle \times \text{conj}(\langle \text{id} \times \text{est} \times \text{pos} \rangle) \longrightarrow \langle \text{id}, \text{est}, \text{pos} \rangle$   
 $\text{convertirEstAHippie} : \langle \text{id} \times \text{est} \times \text{pos} \rangle \longrightarrow \langle \text{id}, \text{pos} \rangle$   
 $\text{sacarHippieTripla} : \langle \text{id} \times \text{pos} \rangle \times \langle \text{conj}(\langle i \times \text{as} \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{pos} \rangle) \times \text{conj}(\langle i \times \text{est} \times \text{pos} \rangle) \rangle \longrightarrow \langle \text{conj}(\langle i, \text{as}, \text{pos} \rangle), \text{sacarHippieTripla} \rangle$   
**axiomas**  $\forall \text{cAs}: \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle) \forall \text{cH}: \text{conj}(\langle \text{id}, \text{pos} \rangle) \forall \text{cEst}: \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \forall \text{cDirs}: \text{conj}(\text{dir})$   
 $\forall \text{al}, \text{an}: \text{nat} \forall \text{cObs}: \text{conj}(\text{pos}) \forall u: \text{uni} \forall i: \text{id} \forall e: \langle \text{id}, \text{est}, \text{pos} \rangle \forall \text{agente}: \langle \text{id}, \text{as}, \text{pos} \rangle \forall h: \langle \text{id}, \text{pos} \rangle \forall p: \text{pos} \forall p4v: \text{conj}(\text{pos}) \forall \text{tripla}, \text{triplaInfo}: \langle \text{conj}(\langle \text{id}, \text{as}, \text{pos} \rangle), \text{cH} \text{ conj}(\langle \text{id}, \text{pos} \rangle), \text{conj}(\langle \text{id}, \text{est}, \text{pos} \rangle) \rangle \forall \text{mcT}: \text{multiconj}(\text{tipoElem})$

## Observadores Basicos

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

```

obstaculos? (agregarH(u, i, p))  $\equiv$  obstaculos?(u)
obstaculos? (moverAs(u, i))  $\equiv$  obstaculos?(u)
obstaculos? (moverH(u, i))  $\equiv$  obstaculos?(u)
obstaculos? (moverE(u, i))  $\equiv$  obstaculos?(u)
agentes? (nuevaUni(cAs, al, an, cObs))  $\equiv$  cAs
agentes? (agregarE(u, i, e, p))  $\equiv$   $\Pi_1$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
agentes? (agregarH(u, i, pos))  $\equiv$   $\Pi_1$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
agentes? (moverAs(u, i))  $\equiv$   $\Pi_1$ ( moverAgenteYSancionarYCapturar(i, u) )
agentes? (moverE(u, i))  $\equiv$   $\Pi_1$ ( moverEstudiante(i, u) )
agentes? (moverH(u, i))  $\equiv$   $\Pi_1$ ( moverHippie(i, u) )
hippies? (nuevaUni(cAs, al, an, cObs))  $\equiv$   $\emptyset$ 
hippies? (agregarE(u, i, e, p))  $\equiv$   $\Pi_2$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
hippies? (agregarH(u, i, p))  $\equiv$   $\Pi_2$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
hippies? (moverAs(u, i))  $\equiv$   $\Pi_2$ ( moverAgenteYSancionarYCapturar(i, u) )
hippies? (moverE(u, i))  $\equiv$   $\Pi_2$ ( moverEstudiante(i, u) )
hippies? (moverH(u, i))  $\equiv$   $\Pi_2$ ( moverHippie(i, u) )
estudiantes? (nuevaUni(cAs, al, an, cObs))  $\equiv$   $\emptyset$ 
estudiantes? (agregarE(u, i, e, p))  $\equiv$   $\Pi_3$ ( chequearSituacionShort( agregarEstudianteTripla(  $\langle i, e, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
estudiantes? (agregarH(u, i, p))  $\equiv$   $\Pi_3$ ( chequearSituacionShort( agregarHippieTripla(  $\langle i, p \rangle$ ,
 $\langle$  agentes?(u), hippies?(u), estudiantes?(u)  $\rangle$  ) ) )
estudiantes? (moverAs(u, i))  $\equiv$   $\Pi_3$ ( moverAgenteYSancionarYCapturar(i, u) )
estudiantes? (moverE(u, i))  $\equiv$   $\Pi_3$ ( moverEstudiante(i, u) )
estudiantes? (moverH(u, i))  $\equiv$   $\Pi_3$ ( moverHippie(i, u) )

```

#### Otras Operaciones

```

capturar(p, cAs)  $\equiv$  if ( p  $\in$  posiciones4Vecinas(  $\Pi_3$ ( dameUno( cAs ) ) ) )
then Ag(  $\langle$   $\Pi_1$ (cAs), darCaptura( seg( dameUno(cAs) ) ),  $\Pi_3$ ( dameUno(cAs) )  $\rangle$ , capturar(
p, sinUno(cAs) ) )
else Ag( dameUno(cAs), capturar( p, sinUno(cAs) ) )
fi
sancionar(p, cAs)  $\equiv$  if ( p  $\in$  posiciones4Vecinas(  $\Pi_3$ ( dameUno( cAs ) ) ) )
then Ag(  $\langle$   $\Pi_1$ (cAs), darSancion( seg( dameUno(cAs) ) ),  $\Pi_3$ ( dameUno(cAs) )  $\rangle$ , capturar(
p, sinUno(cAs) ) )
else Ag( dameUno(cAs), sancionar( p, sinUno(cAs) ) )
fi

```

```

queTipoHay(p, tripla, u)  $\equiv$  if ( fila?(p) = alto?(u)  $\vee$  col?(p) = ancho?(u) )
    then FueraDeRango
    else
    if ( p  $\in$  obstaculos?(u) )
    then Obstaculo
    else

    if ( p  $\in$  damePosicionesAs(  $\Pi_1$ (tripla) ) )
    then Agente
    else
    if ( p  $\in$  damePosicionesH(  $\Pi_2$ (tripla) ) )
    then Hippie
    else
    if ( p  $\in$  damePosicionesEst(  $\Pi_3$ (tripla) ) )
    then Estudiante
    else  $\emptyset$ 
    fi
    fi
    fi
    fi
    fi

cuatroVecinosShort(p, tripla, u)  $\equiv$  cuatroVecinos( p, tripla, u, posiciones4Vecinas(p) )

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(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  $\langle$  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) ) ) )
    ),  $\langle$  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  $\emptyset$  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) )  $\cup$  damePosicionesH( 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  $\rangle$  agentes?(u), hippies?(u), estudiantes?(u)  $\langle$ 
    else
        if vacio?( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ), u ) )
            then  $\rangle$  agentes?(u), hippies?(u), estudiantes?(u)  $\langle$ 
            else chequearSituacionShort( agregarAgenteTripla(  $\rangle$  i, dameAgente( i, agentes?(u) ), mover( dameAgentePos(i, agentes?(u) ), dameUno( posibleMovAs( dameAgente( i, agentes?(u) ), dameAgentePos( i, agentes?(u) ), u ) ) )  $\langle$ ,  $\rangle$  sacarAgenteId(i, agentes?(u) ), hippies?(u), estudiantes?(u)  $\langle$  )
        fi
    fi

```

```

posibleMovAs (agente, p, u)  $\equiv$  if ( inactivo?(agente) )
    then  $\emptyset$ 
    else
        if ( vacio?( hippes?(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 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  $\rangle$ 

```

**Fin TAD**



## 2. TAD AGENTE

### TAD AGENTE

**géneros** as

**exporta** as, Generadores, Observadores Basicos, inactivo?

**usa** NAT, BOOL

**igualdad observacional**

$(\forall a, a' : \text{as}) (a =_{\text{obs}} a' \iff ( ))$

**observadores básicos**

numPlaca : as  $\longrightarrow$  nat

hippiesAtrapados : 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**  $\forall :$

Observadores Basicos

numPlaca(nuevoAs(n))  $\equiv$  n

numPlaca(darCaptura(a))  $\equiv$  numPlaca(a)

numPlaca(darSancion(a))  $\equiv$  numPlaca(a)

hippiesAtrapados(nuevoAs(n))  $\equiv$  0

hippiesAtrapados(darCaptura(a))  $\equiv$  1+ hippiesAtrapados(a)

hippiesAtrapados(darSancion(a))  $\equiv$  hippiesAtrapados(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 TABLERO

#### TAD TABLERO

**géneros**       $\text{tab}$

**exporta**       $\text{tab}, \text{Generadores}, \text{Observadores Basicos}$

**usa**             $\text{NAT}, \text{POS}, \text{BOOL}, \text{CONJ}(\alpha)$

**igualdad observacional**  
 $(\forall t, t' : \text{tab}) \ (t =_{\text{obs}} t' \iff ( ))$

#### observadores básicos

$\text{alto?} : \text{tab} \longrightarrow \text{nat}$

$\text{ancho?} : \text{tab} \longrightarrow \text{nat}$

$\text{ocupadas?} : \text{tab} \longrightarrow \text{conj}(\text{pos})$

#### generadores

$\text{nuevoTablero} : \text{nat } an \times \text{nat } al \longrightarrow \text{tab } t$

$\text{agregarFicha} : \text{nat } id \times \text{pos } p \longrightarrow \text{tab } t$

**axiomas**       $\forall :$   
 $\equiv$

**Fin TAD**