

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

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

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

observadores básicos

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

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

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

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

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

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

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

Otras Operaciones

```

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

sancionar(p, cAs)  $\equiv$  if (  $p \in \text{posiciones4Vecinas}(\Pi_3(\text{dameUno}(cAs)))$  )
    then Ag(  $\langle \Pi_1(cAs), \text{darSancion}(\text{seg}(\text{dameUno}(cAs))), \Pi_3(\text{dameUno}(cAs)) \rangle$ , captu-
        rar( p, sinUno(cAs) ) )
    else Ag( dameUno(cAs), sancionar( p, sinUno(cAs) ) )
    fi

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

            if (  $p \in \text{damePosicionesAs}(\Pi_1(\text{tripla}))$  )
                then Agente
            else
                if (  $p \in \text{damePosicionesH}(\Pi_2(\text{tripla}))$  )
                    then Hippie
                else
                    if (  $p \in \text{damePosicionesEst}(\Pi_3(\text{tripla}))$  )
                        then Estudiante
                    else  $\emptyset$ 
                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(  $\Pi_2(\text{dameUno}(cAs))$ , damePosicionesAs( sinUno(cAs) ) )
    fi

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

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

```

```

moverEstudianteYChequearSituaciones(i, u) ≡ if ( vacio?( dameEstudiante( i, estudiantes?(u) ) ) )
    then < agentes?(u), hippies?(u), estudiantes?(u) >
    else
    if ¬( Π1( dameEstudiante(i, u) ) ∈ 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), Π1( dameEstudiante(i, u) ) )
    ), < agentes?(u), hippies?(u), sacarEstudianteId( i,
    estudiantes?(u) ) > ) )
    fi
    fi

dameEstudiante(i, cEst) ≡ if ( Π1( dameUno(cEst) ) = i )
    then Π2( dameUno(cEst) )
    else dameEstudiante(i, sinUno(cEst) )
    fi

dameEstudiantePos(i, cEst) ≡ if ( Π1( dameUno(cEst) ) = i )
    then Π3( dameUno(cEst) )
    else dameEstudiantePos( i, sinUno(cEst) )
    fi

sacarEstudianteId(i, cEst) ≡ if ( vacio?(cEst) )
    then ∅
    else
    if ( Π1( dameUno(cEst) ) = i )
    then sinUno(cEst)
    else Ag( dameUno(cEst), sacarEstudiante( i, sinUno(cEst) ) )
    fi
    fi

entradas?(an, al) ≡ if ( an = 0 )
    then ∅
    else Ag( pos(0, an-1), Ag( pos(al-1, an-1), entradas(an-1, al) ) )
    fi

dirLibres(p, u) ≡ dirNoOcupadas( dirValidas( p, ancho?(u), alto?(u) ), p, u )

dirValidas(p, an, al) ≡ {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) ) ∨ mo-
    ver( p, dameUno(cDirs) ) ∈ obstaculos?(u) ∨ mover( p, dameUno(cDirs) )
    ∈ damePosicionesAs( agentes?(u) ) ∨ mover( p, dameUno(cDirs) ) ∈ dame-
    PosicionesEst( 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)  $\equiv$  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 < agentes?(u), hippies?(u), estudiantes?(u) >
else chequearSituacionShort( agregarHippieTripla( < i, mover( dameHippiePos(i,u), dameUno( posibleMovH( dameHippiePos(i, u), u ) ) ) >, < agentes?(u), sacarHippieId( i, hippies?(u) ), estudiantes?(u) > ) )
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( ⟨ cAs, cH, cEst ⟩ , triplaInfo, u) ≡ if vacio?(cH) then
    if vacio?(cEst) then
        ⟨ cAs, cH, cEst ⟩
    else
        if movRest, unAgente ⊆ queSituacion(
            cuatroVecinosShort( dameEstudiantePos(
                dameUno(cEst)), triplaInfo, u) ) then
            if dosHippies ∈ queSituacion(
                cuatroVecinosShort( dameEstudiantePos(
                    dameUno(cEst)), triplaInfo, u) ) then
                agregarHippieTripla(
                    convertirEstudianteAHippie(dameUno(cEst))
                    , chequearSituacion( ⟨ sancionar(
                        dameEstudiantePos( dameUno(cEst) ),
                        cAs), cH, sinUno(cEst) ⟩ , triplaInfo, u )
                    )
            else
                agregarEstudianteTripla(
                    dameUno(cEst), chequearSituacion(
                        ⟨ sancionar( dameEstudiantePos(
                            dameUno(cEst) ), cAs), cH,
                            sinUno(cEst) ⟩ , triplaInfo, u ) )
            fi
        else
            if dosHippies ∈ queSituacion(
                cuatroVecinosShort(dameEstudiantePos(dameUno(cEst)),
                triplaInfo, u) ) then
                agregarHippieTripla(
                    convertirEstudianteAHippie(dameUno(cEst)),
                    chequearSituacion( ⟨ cAs, cH,
                        sinUno(cEst) ⟩ , triplaInfo, u ) )
            else
                agregarEstudianteTripla(
                    dameUno(cEst), chequearSituacion(
                        ⟨ cAs, cH, sinUno(cEst) ⟩ , triplaInfo, u
                        ) )
            fi
        fi
    fi
else
    if movRest, unAgente ⊆ queSituacion( cuatroVecinosShort(
        dameHippiePos( dameUno(cH) ), triplaInfo, u) ) then
        chequearSituacion(
            capturar(dameHippiePos(dameUno(cH)),
            cAs), sinUno(cH), cEst ⟩ ,
            sacarHippieTripla(dameUno(cH), triplaInfo, u )
        )
    else
        if movRest, cuatroEstudiantes ⊆ queSituacion(
            cuatroVecinosShort( dameHippiePos(
                dameUno(cH) ), triplaInfo, u) ) then
            agregarEstudianteTripla(
                convertirHippieAEst(dameUno(cH), cEst),
                chequearSituacion( ⟨ cAs, sinUno(cH),
                    cEst ⟩ , sacarHippieTripla(dameUno(cH),
                    triplaInfo), u ) )
            else
                agregarHippieTripla( dameUno(cH), chequearSituacion(
                    ⟨ cAs, sinUno(cH), cEst ⟩ , triplaInfo, u ) )
            fi
        fi
    fi

```

```

convertirHippieAEst(h, cEst)  $\equiv$   $\langle \Pi_0(h), \text{dameUno}(\text{estudiantesAdyacentesPos}(\Pi_1(h), cEst)), \Pi_1(h) \rangle$ 
convertirEstAHippie(e)  $\equiv$   $\langle \Pi_0(e), \Pi_2(e) \rangle$ 
sacarHippieTripla(h,  $\langle cAs, cH, cEst \rangle$ )  $\equiv$   $\langle cAs, \text{sacarHippieId}(\Pi_0(h), cH), cEst \rangle$ 
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( $\Pi_1(\text{dameUno}(cAs))$ ) = 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( $\Pi_1(\text{dameUno}(cAs))$ ) = 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( $\Pi_1(\text{dameUno}(cAs))$ ), maxCapturas(sinUno( $cAs$ )))
fi
minPlaca(cAs)  $\equiv$  if vacio?( $cAs$ ) then
    0
else
    min(numPlaca( $\Pi_1(\text{dameUno}(cAs))$ ), 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

$$(\forall a, a' : \text{as}) \left(a =_{\text{obs}} a' \iff \left(\begin{array}{l} \text{numPlaca}(a) =_{\text{obs}} \text{numPlaca}(a') \wedge \text{hippiesAtrapados}(a) =_{\text{obs}} \text{hippiesAtrapados}(a') \\ \wedge \text{numSanciones}(a) =_{\text{obs}} \text{numSanciones}(a') \end{array} \right) \right)$$

observadores básicos

$\text{numPlaca} : \text{as} \longrightarrow \text{nat}$
 $\text{hippiesCapturados} : \text{as} \longrightarrow \text{nat}$
 $\text{numSanciones} : \text{as} \longrightarrow \text{nat}$

generadores

$\text{nuevoAs} : \text{nat} \longrightarrow \text{as}$
 $\text{darCaptura} : \text{as} \longrightarrow \text{as}$
 $\text{darSancion} : \text{as} \longrightarrow \text{as}$

otras operaciones

$\text{inactivo?} : \text{as} \longrightarrow \text{bool}$

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

$\text{numPlaca}(\text{nuevoAs}(n)) \equiv n$
 $\text{numPlaca}(\text{darCaptura}(a)) \equiv \text{numPlaca}(a)$
 $\text{numPlaca}(\text{darSancion}(a)) \equiv \text{numPlaca}(a)$
 $\text{hippiesCapturados}(\text{nuevoAs}(n)) \equiv 0$
 $\text{hippiesCapturados}(\text{darCaptura}(a)) \equiv 1 + \text{hippiesCapturados}(a)$
 $\text{hippiesCapturados}(\text{darSancion}(a)) \equiv \text{hippiesCapturados}(a)$
 $\text{numSanciones}(\text{nuevoAs}(n)) \equiv 0$
 $\text{numSanciones}(\text{darCaptura}(a)) \equiv \text{numSanciones}(a)$
 $\text{numSanciones}(\text{darSancion}(a)) \equiv 1 + \text{numSanciones}(a)$

Otras Operaciones

$\text{inactivo?}(a) \equiv \text{if } \text{numSanciones}(a) > 3 \text{ then true else false fi}$

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

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

$col? : pos \rightarrow nat$

generadores

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

otras operaciones

$direccionesOptimas : pos \times pos \rightarrow conj(dir)$

$cPosMasCercana : pos \times conj(pos) \times nat \rightarrow conj(pos)$

$cPosMasCercanaShort : pos \times conj(pos) \rightarrow conj(pos)$

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

$dist : pos \times pos \rightarrow nat$

$posiciones4Vecinas : pos \rightarrow conj(pos)$

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

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

$fila?(nuevaPos(f,c)) \equiv f$

$col?(nuevaPos(f,c)) \equiv c$

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

$PosMasCercanaShort(p, cPos) \equiv cPosMasCercana(p, cPos, menorDistancia(p,cPos))$

$cPosMasCercana(p, cPos, menorDist) \equiv \text{if } vacio?(cPos) \text{ then}$
 $\quad \emptyset$
 $\quad \text{else}$
 $\quad \quad \text{if } dist(p, dameUno(cPos)) = menorDist \text{ then}$
 $\quad \quad \quad Ag(p, cPosMasCercana(p, sinUno(cPos), menorDist))$
 $\quad \quad \text{else}$
 $\quad \quad \quad cPosMasCercana(p, sinUno(cPos), menorDist)$
 $\quad \text{fi}$
 fi

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

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

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

```

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

```

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

$$\left(\begin{aligned} & \text{movRest} =_{\text{obs}} \text{movRest} \wedge \text{unAgente} =_{\text{obs}} \text{unAgente} \wedge \text{dosHippies} =_{\text{obs}} \text{dosHippies} \wedge \text{cuatroEs-} \\ & \text{tudiantes} =_{\text{obs}} \text{cuatroEstudiantes} \wedge \neg(\text{movRest} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{movRest} =_{\text{obs}} \text{dosHippies}) \\ & \wedge \neg(\text{movRest} =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{dos-} \\ & \text{Hippies}) \wedge \neg(\text{unAgente} =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{movRest}) \wedge \neg(\text{dosHippies} \\ & =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{dosHippies} =_{\text{obs}} \text{cuatroEstudiantes}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{movRest}) \wedge \\ & \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{unAgente}) \wedge \neg(\text{cuatroEstudiantes} =_{\text{obs}} \text{dosHippies}) \end{aligned} \right)$$

observadores básicos

generadores

movRest : \longrightarrow situ

unAgente : \longrightarrow situ

dosHippies : \longrightarrow situ

cuatroEstudiantes : \longrightarrow situ

otras operaciones

Fin TAD