Trabajo Practico I, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD Universidad

TAD Universidad

géneros uni

exporta uni, Generadores, Observadores Basicos

usa Nat, Conju(α), Bool, Tupla($\alpha_1, \ldots, \alpha_n$)

igualdad observacional

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

observadores básicos

alto? : uni $\longrightarrow$ nat

ancho? : uni $\longrightarrow$ nat

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

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

hippies? : uni $\longrightarrow$ conj(pos)

estudiantes? : uni $\longrightarrow$ 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) $\longrightarrow$ 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

chequearSituacionShort(tripla, u) : $\longrightarrow$

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

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

axiomas $\forall$:

Observadores Basicos

alto? (nuevaUni(ca, al, an, cobs)) $\equiv$ al

alto? (agregarE(u, i, e, pos)) $\equiv$ alto?(u)

alto? (agregarH(u, i, pos)) $\equiv$ 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(ca, al, an, cobs)) $\equiv$ an

ancho? (agregarE(u, i, e, pos)) $\equiv$ ancho?(u)

ancho? (agregarH(u, i, pos)) $\equiv$ 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(ca, al, an, cobs)) $\equiv$ cobs

obstaculos? (agregarE(u, i, e, pos)) $\equiv$ obstaculos?(u)

obstaculos? (agregarH(u, i, pos)) $\equiv$ 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(ca, al, an, cobs)) $\equiv$ ca

agentes? (agregarE(uni, i, e, pos)) $\equiv$ Π_1(chequearSituacionShort(agregarEstudianteTripla($\langle$ i, e, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

agentes? (agregarH(uni, i, pos)) $\equiv$ Π_1(chequearSituacionShort(agregarHippieTripla($\langle$ i, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

agentes? (moverAs(uni, i)) $\equiv$ Π_1(moverAgenteYSancionarYCapturar(i, uni))

agentes? (moverE(uni, i)) $\equiv$ Π_1(moverEstudiante(i, uni))

agentes? (moverH(uni, i)) $\equiv$ Π_1(moverHippie(i,uni))

hippies? (nuevaUni(ca, al, an, cobs)) $\equiv$ $\emptyset$

hippies? (agregarE(uni, i, e, pos)) $\equiv$ Π_2(chequearSituacionShort(agregarEstudianteTripla($\langle$ i, e, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

hippies? (agregarH(uni, i, pos)) $\equiv$ Π_2(chequearSituacionShort(agregarHippieTripla($\langle$ i, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

hippies? (moverAs(uni, i)) $\equiv$ Π_2(moverAgenteYSancionarYCapturar(i, uni))

hippies? (moverE(uni, i)) $\equiv$ Π_2(moverEstudiante(i, uni))

hippies? (moverH(uni, i)) $\equiv$ Π_2(moverHippie(i, uni))

estudiantes? (nuevaUni(ca, al, an, cobs)) $\equiv$ $\emptyset$

estudiantes? (agregarE(uni, i, e, pos)) $\equiv$ Π_3(chequearSituacionShort(agregarEstudianteTripla($\langle$ i, e, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

estudiantes? (agregarH(u, i, pos)) $\equiv$ Π_3(chequearSituacionShort(agregarHippieTripla($\langle$ i, pos $\rangle$, $\langle$ agentes?(u), hippies?(u), estudiantes?(u) $\rangle$)))

estudiantes? (moverAs(uni, i)) $\equiv$ Π_3(moverAgenteYSancionarYCapturar(i, uni))

estudiantes? (moverE(uni, i)) $\equiv$ Π_3(moverEstudiante(i, uni))

estudiantes? (moverH(uni, i)) $\equiv$ Π_3(moverHippie(i, uni))

Otras Operaciones

capturar(p, cAs) $\equiv$ if ($p \in$ posiciones4Vecinas(Π_3(dameUno(cAs))))
then Ag($\langle$ Π_1(cAs), darCaptura(seg(dameUno(cAs))), Π_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(Π_3(dameUno (cAs))))
then Ag($\langle$ Π_1(cAs), darSancion(seg(dameUno(cAs))), Π_3(dameUno(cAs)) $\rangle$, capturar(p, sinUno(cAs)))
else Ag(dameUno(cAs), sancionar(p, sinUno(cAs)))
fi

sancionar(p, cAs) $\equiv$ if ($p \in$ posiciones4Vecinas(Π_3(dameUno(cAs))))
then Ag($\langle$ Π_1(cAs), darSancion(seg(dameUno(cAs))), Π_3(dameUno(cAs)) $>$, 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(Π_1(tripla)))
then Agente
else
if ($p \in$ damePosicionesH(Π_2(tripla)))
then Hippie
else
if ($p \in$ damePosicionesEst(Π_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(ca) $\equiv$ if vacio?(ca)
then $\emptyset$
else Ag(Π_2(dameUno(ca)), damePosicionesAs(sinUno(ca)))
fi

damePosicionesEst(ce) $\equiv$ if vacio?(ce)
then $\emptyset$
else Ag(Π_3(dameUno(ce)), damePosicionesEst(sinUno(ce)))
fi

damePosicionesH(ch) $\equiv$ if vacio?(ch)
then $\emptyset$
else Ag(Π_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 $\neg$(Π_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)), Π_1(dameEstudiante(i, u))) $\rangle$, $\langle$ agentes?(u), hippies?(u), sacarEstudianteId(i, estudiantes?(u)) $\rangle$))
fi
fi

dameEstudiante(i, cEst) $\equiv$ if (Π_1(dameUno($cEst$)) = i)
then Π_2(dameUno($cEst$))
else dameEstudiante(i, sinUno($cEst$))
fi

dameEstudiantePos(i, cEst) $\equiv$ if (Π_1(dameUno($cEst$)) = i)
then Π_3(dameUno($cEst$))
else dameEstudiantePos(i, sinUno($cEst$))
fi

sacarEstudianteId(i, cEst) $\equiv$ if (vacio?($cEst$))
then $\emptyset$
else
if (Π_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(pos, u) $\equiv$ dirNoOcupadas(dirValidas(pos, ancho?(u), alto?(u)), pos, u)

dirValidas(pos, an, al) $\equiv$ {n,s,e,o} - (if col?(pos) = 0 then {o} else $\emptyset$ fi) - (if col?(pos) = an-1 then {e} else $\emptyset$ fi) - (if fila?(pos) = 0 then {n} else $\emptyset$ fi) - (if fila?(pos) = al-1 then {s} else $\emptyset$ fi)

dirNoOcupadas(cDirs, pos, u) $\equiv$ if (vacio(cDirs))
then $\emptyset$
else
if (mover(pos, dameUno($cDirs$)) $\in$ damePosicionesH(hippies?(u)) $\vee$ mover(pos, dameUno($cDirs$)) $\in$ obstaculos?(u) $\vee$ mover(pos, dameUno($cDirs$)) $\in$ damePosicionesAs(agentes?(u)) $\vee$ mover(pos, dameUno($cDirs$)) $\in$ damePosicionesEst(estudiantes?(u)))
then dirNoOcupadas(sinUno($cDirs$), pos, u)
else Ag(dameUno($cDirs$), dirNoOcupadas(sinUno($cDirs$), pos, 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, pos, u) $\equiv$ if (inactivo?(agente))
then $\emptyset$
else
if (vacio?(hippes?(u)))
then
if (dirLibres(agente, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, posNoOcupadas(entradas(an, al), u)))) = $\emptyset$)
then dirLibres(pos, u)
else dirLibres(pos, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, posNoOcupadas(entradas(an, al), u))))
fi
else
if dirLibres(pos, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, hippies?(u)))) = $\emptyset$
then dirLibres(pos, u)
else dirLibres(pos, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, hippies?(u))))
fi
fi
fi

dameAgente(i, cAs) $\equiv$ if (Π_1(dameUno(cAs)) = i)
then Π_2(dameUno(cAs))
else dameAgente(i, sinUno(cAs))
fi

dameAgentePos(i, cAs) $\equiv$ if (Π_1(dameUno(cAs)) = i) then Π_3(dameUno(cAs)) else dameAgente(i, sinUno(cAs)) fi

sacarAgenteId(i, cAs) $\equiv$ if (vacio?(cAs))
then $\emptyset$
else
if (Π_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 (pos, u) $\equiv$ **if** vacio?(estudiantes?(u)) **then**
$\emptyset$
else
if dirLibres(pos, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, damePosicionesEst(estudiantes?(u))))) $= \emptyset$ **then**
dirLibres(pos, u)
else
dirLibres(pos, u) $\cap$ direccionesOptimas(dameUno(cPosMasCercanaShort(pos, 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(Π_2(dameUno($cEst$)), estudiantesAdyacentesPos(p, sinUno($cEst$)))
else
estudiantesAdyacentesPos(p, sinUno($cEst$))
fi

chequearSituacion(⟨ cAs, cH, cEst ⟩ , triplaInfo, u) ≡ **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

Fin TAD

2. TAD Agente

TAD Agente

géneros as

exporta as, Generadores, Observadores Basicos, hippiesAtrapados

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

capturarH : as a $\longrightarrow$ as

sancionar : as $\longrightarrow$ as

otras operaciones

inactivo? : as $\longrightarrow$ bool

axiomas $\forall$:

Observadores Basicos

numPlaca(nuevoAs(n)) ≡ n

numPlaca(capturarH(a)) $\equiv$ numPlaca(a)

numPlaca(sancionar(a)) $\equiv$ numPlaca(a)

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

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

hippiesAtrapados(sancionar(a)) $\equiv$ hippiesAtrapados(a)

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

numSanciones(capturarH(a)) $\equiv$ numSanciones(a)

numSanciones(sancionar(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 tab

exporta tab, Generadores, Observadores Basicos

usa Nat, Pos, Bool, Conj(α)

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

observadores básicos

alto? : tab $\longrightarrow$ nat

ancho? : tab $\longrightarrow$ nat

ocupadas? : tab $\longrightarrow$ conj(*pos*)

generadores

nuevoTablero : nat *an* $\times$ nat *al* $\longrightarrow$ tab *t*

agregarFicha : nat *id* $\times$ pos *p* $\longrightarrow$ tab *t*

axiomas $\forall$:

$\equiv$

Fin TAD