

Trabajo Practico II, Alta Seguridad nos cuida

Algoritmos y Estructuras de Datos II, DC, UBA.

Índice

1. TAD AGENTES	2
----------------	---

1. TAD AGENTES

TAD POSICION ES $\langle \text{NAT}, \text{NAT} \rangle$

TAD AGENTES

géneros agentes

exporta agentes, Generadores, Observadores Basicos, masVigilante, conMismasSanciones, conKSanciones, estaAgente

usa NAT, CONJ(α), BOOL, TUPLA($\alpha_1, \dots, \alpha_n$), PLACA, DICCIONARIO(CLAVE, SIGNIFICADO)

observadores básicos

agentes? : agentes \longrightarrow conj(placa)

posicionAgente : placa $a \times$ agentes $as \longrightarrow$ posicion $\{a \in \text{agentes?}(as)\}$

sancionesAgente : placa $a \times$ agentes $as \longrightarrow$ nat $\{a \in \text{agentes?}(as)\}$

capturasAgente : placa $a \times$ agentes $as \longrightarrow$ nat $\{a \in \text{agentes?}(as)\}$

generadores

nuevoAgentes : dicc(placa \times posicion) $d \longrightarrow$ agentes $\{\neg \emptyset?(\text{claves}(d))\}$

agregarSancion : placa $a \times$ agentes $as \longrightarrow$ agentes $\{a \in \text{agentes?}(as)\}$

cambiarPos : placa $a \times$ posicion $p \times$ agentes $as \longrightarrow$ agentes $\{a \in \text{agentes?}(as)\}$

agregarCaptura : placa $a \times$ agentes $as \longrightarrow$ agentes $\{a \in \text{agentes?}(as)\}$

otras operaciones

masVigilante : agentes \longrightarrow placa

conMismasSanciones : placa $a \times$ agentes $as \longrightarrow$ conj(placa) $\{a \in \text{agentes?}(as)\}$

conKSanciones : nat \times agentes \longrightarrow conj(placa)

estaAgente : placa $a \times$ agentes $as \longrightarrow$ bool

axiomas $\forall as: \text{agentes} \forall a: \text{placa} \forall k: \text{nat} \forall d: \text{diccionario}(\text{placa}, \text{posicion}) \forall p: \text{posicion} \forall cAs: \text{conj}(\text{placa})$

agentes?(nuevoAgentes(d)) \equiv claves(d)

agentes?(agregarSancion(a, as)) \equiv agentes?(as)

agentes?(cambiarPos(a, p, as)) \equiv agentes?(as)

agentes?(agregarCaptura(a, as)) \equiv agentes?(as)

posicionAgente($a, \text{nuevoAgentes}(d)$) \equiv obtener(a, d)

posicionAgente($a, \text{agregarSancion}(a, as)$) \equiv posicionAgente(a, as)

posicionAgente($a, \text{cambiarPos}(a, p, as)$) \equiv p

posicionAgente($a, \text{agregarCaptura}(a, as)$) \equiv posicionAgente(a, as)

sancionesAgente($a, \text{nuevoAgentes}(d)$) \equiv 0

sancionesAgente($a, \text{agregarSancion}(a, as)$) \equiv 1 + sancionesAgente(a, as)

sancionesAgente($a, \text{cambiarPos}(a, p, as)$) \equiv sancionesAgente(a, as)

sancionesAgente($a, \text{agregarCaptura}(a, as)$) \equiv sancionesAgente(a, as)

capturasAgente($a, \text{nuevoAgentes}(d)$) \equiv 0

capturasAgente($a, \text{agregarSancion}(a, as)$) \equiv capturasAgente(a, as)

capturasAgente($a, \text{cambiarPos}(a, p, as)$) \equiv capturasAgente(a, as)

capturasAgente($a, \text{agregarCaptura}(a, as)$) \equiv 1 + capturasAgente(a, as)

masVigilante(as) \equiv conMenorPlaca(conKCapturas(maxCapturas(agentes?(as), as), agentes?(as), as), as)

conMismasSanciones(a, as) \equiv agentesConKSanciones(sancionesAgente(a, as), agentes?(as), as)

```

conKSanciones( $k, as$ )  $\equiv$  agentesConKSanciones( $k, agentes?(as), as$ )
estaAgente( $a, as$ )  $\equiv a \in agentes?(as)$ 
conMenorPlaca( $cAs, as$ )  $\equiv$  if  $\#(cAs) = 1$  then
    dameUno( $cAs$ )
else
    min(dameUno( $cAs$ ), conMenorPlaca(sinUno( $cAs$ ),  $as$ ))
fi
maxCapturas( $cAs, as$ )  $\equiv$  if  $\emptyset?(as)$  then
    0
else
    max(capturasAgente(dameUno( $cAs$ ),  $as$ ), maxCapturas(sinUno( $cAs$ ),  $as$ ))
fi
conKCapturas( $k, cAs, as$ )  $\equiv$  if  $\emptyset?(as)$  then
     $\emptyset$ 
else
    if  $capturasAgente(dameUno(cAs), as) = k$  then
        Ag(dameUno( $as$ ), conKCapturas( $k, sinUno(cAs), as$ ))
    else
        conKCapturas( $k, sinUno(cAs), as$ )
    fi
fi
agentesConKSanciones( $k, cAs, as$ )  $\equiv$  if  $\emptyset?(as)$  then
     $\emptyset$ 
else
    if  $sancionesAgente(dameUno(cAs), as) = k$  then
        Ag(dameUno( $as$ ), agentesConKSanciones( $k, sinUno(cAs), as$ ))
    else
        agentesConKSanciones( $k, sinUno(cAs), as$ )
    fi
fi

```

Fin TAD

TAD NOMBRE ES STRING

TAD HIPPIES Y ESTUDIANTES**géneros** hipiest**exporta** hipiest, Generadores, Observadores Basicos, esta?, esHip?, esEst?, borrar**usa** NAT, CONJ(α), BOOL, TUPLA($\alpha_1, \dots, \alpha_n$), NOMBRE, DICCIONARIO(CLAVE, SIGNIFICADO)**observadores básicos**diccPos : hipiest \longrightarrow dicc(nombre, posicion)esHip? : nombre $n \times$ hipiest $he \longrightarrow$ bool {esta?(n, he)}**generadores**nuevoHippiesYEstudiantes : \longrightarrow hipiestdefHippie : nombre $n \times$ posicion $p \times$ hipiest $he \longrightarrow$ hipiestdefEstudiante : nombre $n \times$ posicion $p \times$ hipiest $he \longrightarrow$ hipiest**otras operaciones**posHippieYEst : nombre $n \times$ hipiest $he \longrightarrow$ posicion { $n \in hippies?(as) \vee n \in estudiantes?(as)$ }hippies? : hipiest \longrightarrow conj(nombre)estudiantes? : hipiest \longrightarrow conj(nombre)esta? : nombre \times hipiest \longrightarrow boolesEst? : nombre $n \times$ hipiest $he \longrightarrow$ bool {esta?(n, he)}

borrar : nombre $n \times$ hipiest $he \rightarrow$ hipiest {esta?(n, he)}
 soloHippies : conj(nombre) $cn \times$ hipiest $he \rightarrow$ conj(nombre)
{ $(\forall n : \text{nombre}) n \in cn \Rightarrow n \in \text{claves}(\text{diccPos}(he))$ }
 soloEstudiantes : conj(nombre) $cn \times$ hipiest $he \rightarrow$ conj(nombre)
{ $(\forall n : \text{nombre}) n \in cn \Rightarrow n \in \text{claves}(\text{diccPos}(he))$ }

axiomas $\forall he: \text{hipiest } \forall n: \text{nombre } \forall b: \text{nombre } \forall d: \text{diccionario}(\text{nombre}, \text{posicion}) \forall p: \text{posicion } \forall cN: \text{conj}(\text{nombre})$

diccPos(nuevoHippiesYEstudiantes()) \equiv vacio
 diccPos(defHippie(n, p, he)) \equiv definir($n, p, \text{diccPos}(he)$)
 diccPos(defEstudiante(n, p, he)) \equiv definir($n, p, \text{diccPos}(he)$)
 esHip?($b, \text{defHippie}(n, p, he)$) \equiv **if** $b = n$ **then** true **else** esHip?(b, he) **fi**
 esHip?($b, \text{defEstudiante}(n, p, he)$) \equiv **if** $b = n$ **then** false **else** esHip?(b, he) **fi**
 posHippieYEst(n, he) \equiv obtener($n, \text{diccPos}(he)$)
 esta?(n, he) $\equiv n \in \text{claves}(\text{diccPos}(he))$
 esEst?(n, he) $\equiv \neg \text{esHip}(n, he)$
 hippies?(he) \equiv soloHippies($\text{claves}(\text{diccPos}(he)), he$)
 estudiantes?(he) \equiv soloEstudiantes($\text{claves}(\text{diccPos}(he)), he$)
 soloHippies(cn, he) \equiv **if** $\emptyset?(cn)$ **then**
 \emptyset
 else
 if esHip?($\text{dameUno}(cn), he$) **then**
 Ag($cn, \text{soloHippies}(\text{sinUno}(cn, he))$)
 else
 soloHippies($\text{sinUno}(cn, he)$)
 fi
 fi
 soloEstudiantes(cn, he) \equiv **if** $\emptyset?(cn)$ **then**
 \emptyset
 else
 if $\neg \text{esHip}?(dameUno(cn), he)$ **then**
 Ag($cn, \text{soloEstudiantes}(\text{sinUno}(cn, he))$)
 else
 soloEstudiantes($\text{sinUno}(cn, he)$)
 fi
 fi

Fin TAD