Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

Departamento de Computación Facultad de Ciencias Exactas y Naturales Universidad de Buenos Aires

El diseño contraataca

La cosa se pone compleja

Integrante	LU	Correo electrónico
Church, Alonso	1/20	alonso@iglesia.com
Lovelace, Ada	10/19	ada_de_los_dientes@tatooine.com
Null, Linda	100/18	null@null.null
Turing, Alan	314/16	halting@problem.com
asd	232	2323

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. Introducción

Esta es la introducción en LATEX.

2. Desarrollo

2.1. Parte 1

```
TAD CASILLERO es TUPLA(nat, nat)
   TAD FANTASMA es CASILLERO
   TAD PARED es CASILLERO
TAD DIRECCIÓN
     géneros
                    dirección
     generadores
       arriba : \longrightarrow dirección
       abajo : \longrightarrow dirección
       derecha : \longrightarrow dirección
       izquierda : \longrightarrow dirección
Fin TAD
TAD MAPA
     usa
                   Tipo1, Tipo2, Nat
                   observadores, operación adicional
     exporta
     géneros
                   mapa
     observadores básicos
       fantasmas : mapa \longrightarrow conj(Fantasma)
       paredes : mapa \longrightarrow conj(Pared)
       dimensiones : mapa \longrightarrow tupla(nat,nat)
       casillero<br/>Inicial : mapa \longrightarrow Casillero
       casillero
DeLlegada : mapa \longrightarrow Casillero
     generadores
       nuevo
Mapa : tupla(nat \times nat) d \times Casillero \times Casillero \times conj(Fantasma) fs \times conj(Pared) ps \longrightarrow mapa
                                         \{\forall (f \in fs)(\neg \exists (p \in ps)(f = p)) \rightarrow_L (manhattan(f, inicio) \geq 3 \land (ps \land fs) \in d)\}
     axiomas
       fantasmas(nuevoMapa(dimension, inicio, fin, fantasmas, paredes)) \equiv fantasmas
       paredes(nuevoMapa(dimension, inicio, fin, fantasmas, paredes)) \equiv paredes
       dimensiones(nuevoMapa(dimension, inicio, fin, fantasmas, paredes)) \equiv dimension
       casilleroInicial(nuevoMapa(dimension, inicio, fin, fantasmas, paredes)) \equiv inicio
```

 ${\it casilleroDeLlegada(nuevoMapa(dimension,\,inicio,\,fin,\,fantasmas,\,paredes))} \;\equiv\; casilleroDeLlegada$

Fin TAD

```
TAD JUGADOR
```

```
usa
                 Tipo1, Tipo2, Nat, Mapa
exporta
                observadores, mover, estaAsustado?, gano?
géneros
                jugador
observadores básicos
  posición : jugador --> tupla(nat, nat)
generadores
  nuevoJugador : tupla(nat \times nat) \longrightarrow jugador
otras operaciones
  mover : jugadorj \times mapa × dirección \longrightarrow jugador
                                                                                                 {\neg esta Asustado?(j) \land \neg gano?(j)}
  esta
Asustado? : jugador \times mapa \longrightarrow bool
  gano? : jugador \times mapa \longrightarrow bool
  excedeElMapa? : mapa \times tupla(nat \times nat) \longrightarrow bool
axiomas
  posición(nuevoJugador(\langle x,y \rangle)) \equiv \langle x,y \rangle
  mover(nuevoJugador(\langle x,y \rangle), mapa), arriba) \equiv \mathbf{if} (y=0?) \lor_L (\langle x,y-1 \rangle \in paredes(mapa)) then
                                                                    nuevoJugador(\langle x,y \rangle)
                                                                else
                                                                    nuevoJugador(\langle x, y - 1 \rangle)
                                                                \mathbf{fi}
  mover(nuevoJugador(\langle x,y \rangle), mapa), abajo) \equiv if (y + 1 < \pi_2(\text{dimensiones}(mapa))) \lor (\langle x,y + 1 \rangle \in
                                                               paredes(mapa)) then
                                                                   nuevoJugador(\langle x,y \rangle)
                                                                else
                                                                   nuevoJugador(\langle x,y+1\rangle)
  mover(nuevoJugador(\langle x,y \rangle), mapa), derecha) \equiv if (x + 1 < \pi_1(\text{dimensiones}(mapa))) <math>\vee (\langle x + 1,y \rangle \in
                                                                  paredes(mapa)) then
                                                                      nuevoJugador(\langle x,y \rangle)
                                                                  else
                                                                      nuevoJugador(\langle x+1,y\rangle)
  mover(nuevoJugador(\langle x,y \rangle), mapa), izquierda) \equiv if (x=0?) \lor_L (\langle x-1,y \rangle \in paredes(mapa)) then
                                                                        nuevoJugador(\langle x,y \rangle)
                                                                    else
                                                                        nuevoJugador(\langle x-1,y\rangle)
  esta Asustado? (nuevo Jugador (posicion Del Jugador), mapa)) \equiv \exists (f \in fantas mas(mapa)) (manhattan(f, fantas mas(mapa)))
                                                                                  posicionDelJugador) \leq 3
```

gano?(nuevoJugador(posicionDelJugador), mapa) \equiv casilleroDeLlegada(mapa) = posicionDelJugador

Fin TAD

TAD PARTIDA

```
Mapa, Jugador, Bool
usa
exporta
               observadores, operación adicional
géneros
               partida
observadores básicos
  mapa : \longrightarrow mapa
  jugador : \longrightarrow jugador
  finalizó? : \longrightarrow bool
generadores
  nueva
Partida : mapa \times jugador \longrightarrow partida
otras operaciones
  entrada : partida \times dirección \longrightarrow jugador
                                                                                                               \{\neg finalizó?\}
axiomas
  mapa(nuevaPartida(mapa, jugador)) \equiv mapa
  jugador(nuevaPartida(mapa, jugador)) \equiv jugador
  finalizó?(nuevaPartida(mapa, jugador)) \equiv estaAsustado?(jugador, mapa) \vee gano?(jugador, mapa)
  nueva
Partida<br/>(mapa,\,jugador)\,\equiv\, {\rm posici\acute{o}n}(jugador)={\rm casilleroInicial}(mapa)
  entrada(partida, jugador, direccin) \equiv mover(jugador, mapa(partida), direccin)
```

Fin TAD

3. Conclusiones

Esta cátedra es la mejor. Especialmente en humildad.