

Especificación Referencia TP2

Algoritmos y Estructuras de Datos II, DC, UBA.

1er cuatrimestre, 2020

Índice

1. Renombres de TADs	2
2. TAD MAPA	2
3. TAD SIMCITY	2

1. Renombres de TADs

- **TAD** Casilla es de tipo $\text{tupla}\langle \text{Nat}, \text{Nat} \rangle$
- **TAD** Direccion es de tipo $\text{Enum}\{\text{Horizontal}, \text{Vertical}\}$
- **TAD** Posicion es de tipo Nat

2. TAD MAPA

TAD MAPA

igualdad observacional

$$(\forall m, m' : \text{Mapa}) (m =_{\text{obs}} m' \iff ((\forall c : \text{Casilla})(\text{hayRio}(m, c) =_{\text{obs}} \text{hayRio}(m', c))))$$

géneros mapa

exporta mapa, generadores, observadores

observadores básicos

$\text{hayRio} : \text{Mapa} \times \text{Casilla} \longrightarrow \text{Bool}$

generadores

$\text{nuevoMapa} : \longrightarrow \text{Mapa}$

$\text{agregarRio} : \text{Mapa } m \times \text{Direccion } d \times \text{Posicion } p \longrightarrow \text{Mapa}$

otras operaciones

$\text{unirMapa} : \text{Mapa } m1 \times \text{Mapa } m2 \longrightarrow \text{Mapa}$

axiomas $\forall m, m1, m2 : \text{Mapa}, \forall c : \text{Casilla}, \forall d : \text{Direccion}, \forall p : \text{Posicion}$

$\text{hayRio}(\text{nuevoMapa}, c) \equiv \text{false}$

$\text{hayRio}(\text{agregarRio}(m, d, p), c) \equiv \text{if } d = \text{Vertical} \text{ then}$
 $\quad p = \pi_1(c) \vee \text{hayRio}(m, c)$
 $\quad \text{else}$
 $\quad p = \pi_2(c) \vee \text{hayRio}(m, c)$
 $\quad \text{fi}$

$\text{unirMapa}(\text{nuevoMapa}, m2) \equiv m2$

$\text{unirMapa}(\text{agregarRio}(m, d, p), m2) \equiv \text{unirMapa}(m, \text{agregarRio}(m2, d, p))$

Fin TAD

3. TAD SIMCITY

igualdad observacional

$$(\forall s, s' : \text{SimCity}) \left(s =_{\text{obs}} s' \iff \left(\begin{array}{l} \text{mapa}(s) =_{\text{obs}} \text{mapa}(s') \wedge \text{casas}(s) =_{\text{obs}} \text{casas}(s') \wedge \\ \text{comercios}(s) =_{\text{obs}} \text{comercios}(s') \wedge \text{popularidad}(s) =_{\text{obs}} \\ \text{popularidad}(s') \wedge_L (\forall c : \text{Casilla}) c \in \text{construcciones}(s) \Rightarrow \\ \text{nivel}(s, c) = \text{nivel}(s', c) \end{array} \right) \right)$$

TAD SIMCITY

géneros SimCity

exporta SimCity, generadores, observadores, otras operaciones

observadores básicos

$\text{mapa} : \text{SimCity} \rightarrow \text{Mapa}$
 $\text{casas} : \text{SimCity} \rightarrow \text{Conj}(\text{Casilla})$
 $\text{comercios} : \text{SimCity} \rightarrow \text{Conj}(\text{Casilla})$
 $\text{nivel} : \text{SimCity } s \times \text{Casilla } c \rightarrow \text{Nat} \quad \{\text{hayConstruccion}(s, c)\}$
 $\text{huboConstruccion} : \text{SimCity } s \rightarrow \text{bool}$
 $\text{popularidad} : \text{SimCity} \rightarrow \text{Nat}$

generadores

$\text{empezarPartida} : \text{Mapa} \rightarrow \text{SimCity}$
 $\text{agregarCasa} : \text{SimCity } s \times \text{Casilla } c \rightarrow \text{SimCity} \quad \{\text{sePuedeConstruir}(s, c)\}$
 $\text{agregarComercio} : \text{SimCity } s \times \text{Casilla } c \rightarrow \text{SimCity} \quad \{\text{sePuedeConstruir}(s, c)\}$
 $\text{avanzarTurno} : \text{SimCity } s \rightarrow \text{SimCity} \quad \{\text{huboConstruccion}(s)\}$
 $\text{unir} : \text{SimCity } s1 \times \text{SimCity } s2 \rightarrow \text{SimCity}$

$$\left\{ \begin{array}{l} (\forall c : \text{Casilla}) c \in \text{construcciones}(s1) \Rightarrow \neg \text{hayRio}(\text{mapa}(s2), c) \wedge_L \\ (\forall c : \text{Casilla}) c \in \text{construcciones}(s2) \Rightarrow \neg \text{hayRio}(\text{mapa}(s1), c) \wedge_L \\ (\forall c1, c2 : \text{Casilla}) (c1 \in \text{construcciones}(s1) \wedge c2 \in \text{construcciones}(s2)) \Rightarrow \\ (\text{esCasillaDeMaximoNivel}(s1, c1) \wedge \text{esCasillaDeMaximoNivel}(s2, c2) \Rightarrow c1 \neq c2) \end{array} \right\}$$

otras operaciones

$\text{antiguedad} : \text{SimCity} \rightarrow \text{Nat}$
 $\text{construcciones} : \text{SimCity} \rightarrow \text{Conj}(\text{Casilla})$
 $\text{hayConstruccion} : \text{SimCity} \times \text{Casilla} \rightarrow \text{Bool}$
 $\text{sePuedeConstruir} : \text{SimCity} \times \text{Casilla} \rightarrow \text{Bool}$
 $\text{esCasillaDeMaximoNivel} : \text{SimCity } s \times \text{Casilla } c \rightarrow \text{Bool} \quad \{\text{hayConstruccion}(s, c)\}$
 $\text{unirCasasPriorizandoNivelMaximo} : \text{SimCity } s1 \times \text{SimCity } s2 \rightarrow \text{Conj}(\text{Casilla})$
 $\text{unirComerciosPriorizandoNivelMaximo} : \text{SimCity } s1 \times \text{SimCity } s2 \rightarrow \text{Conj}(\text{Casilla})$
 $\text{casasADistancia3Manhattan} : \text{Conj}(\text{Casilla}) \times \text{Casilla} \rightarrow \text{Conj}(\text{Casilla})$
 $\text{maximoNivel} : \text{SimCity } s \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Nat} \quad \{c \in \text{construcciones}(s)\}$
 $\text{unirCasas} : \text{SimCity } s1 \times \text{SimCity } s2 \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Conj}(\text{Casilla}) \quad \{c = \text{casas}(s1)\}$
 $\text{unirComercios} : \text{SimCity } s1 \times \text{SimCity } s2 \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Conj}(\text{Casilla}) \quad \{c = \text{comercios}(s1)\}$
 $\text{darNivelAComercio} : \text{SimCity } s1 \times \text{SimCity } s2 \times \text{Casilla } c \rightarrow \text{Nat}$

$$\{c \in \text{comercios}(s1) \cup \text{comercios}(s2)\}$$

 $\text{darMaximoNivelComercio} : \text{SimCity } s1 \times \text{SimCity } s2 \times \text{Casilla } c \rightarrow \text{Nat}$

$$\{c \in \text{comercios}(s1) \cup \text{comercios}(s2)\}$$

 $\text{maximoNivelComercio} : \text{SimCity } s1 \times \text{SimCity } s2 \times \text{Casilla } c \rightarrow \text{Nat}$

$$\{c \in \text{comercios}(s1) \cup \text{comercios}(s2)\}$$

 $\text{distanciaManhattan} : \text{Casilla } x \times \text{Casilla } y \rightarrow \text{Nat}$

axiomas $\forall s, s1, s2: \text{SimCity}, \forall \text{mapa}: \text{Mapa}, \forall \text{casas}, \text{comercios}: \text{Conj}(\text{Casilla}), \forall c, x, y: \text{Casilla}$

$\text{mapa}(\text{empezarPartida}(\text{mapa})) \equiv \text{mapa}$
 $\text{mapa}(\text{agregarCasa}(s, c)) \equiv \text{mapa}(s)$
 $\text{mapa}(\text{agregarComercio}(s, c)) \equiv \text{mapa}(s)$
 $\text{mapa}(\text{avanzarTurno}(s)) \equiv \text{mapa}(s)$
 $\text{mapa}(\text{unir}(s1, s2)) \equiv \text{unirMapa}(\text{mapa}(s1), \text{mapa}(s2))$
 $\text{casas}(\text{empezarPartida}(\text{mapa})) \equiv \emptyset$
 $\text{casas}(\text{agregarCasa}(s, c)) \equiv \text{Ag}(c, \text{casas}(s))$

```

casas(agregarComercio(s, c))  $\equiv$  casas(s)
casas(avanzarTurno(s))  $\equiv$  casas(s)
casas(unir(s1, s2))  $\equiv$  unirCasasPriorizandoNivelMaximo(s1, s2)

comercios(empezarPartida(mapa))  $\equiv$   $\emptyset$ 
comercios(agregarCasa(s, c))  $\equiv$  Ag(c, comercios(s))
comercios(agregarComercio(s, c))  $\equiv$  comercios(s)
comercios(avanzarTurno(s))  $\equiv$  comercios(s)
comercios(unir(s1, s2))  $\equiv$  unirComerciosPriorizandoNivelMaximo(s1, s2)

nivel(agregarCasa(s, c), c')  $\equiv$  if c' = c then 0 else nivel(s, c') fi
nivel(agregarComercio(s, c), c')  $\equiv$  if c' = c then
    if  $\emptyset?$ (casasADistancia3Manhattan(casas(s), c)) then
        0
    else
        maximoNivel(s, casasADistancia3Manhattan(casas(s), c))
    fi
else
    nivel(s, c)
fi
nivel(avanzarTurno(s), c)  $\equiv$  1 + nivel(s, c)
nivel(unir(s1,s2), c)  $\equiv$  if c  $\in$  unirCasas(s1, s2, casas(s1)) then
    nivel(s1, c)
else
    if c  $\in$  unirCasas(s2, s1, casas(s2)) then
        nivel(s2, c)
    else
        darNivelAComercio(s1, s2, c)
    fi
fi

huboConstruccion(empezarPartida(mapa))  $\equiv$  false
huboConstruccion(agregarCasa(s, c))  $\equiv$  true
huboConstruccion(agregarComercio(s, c))  $\equiv$  true
huboConstruccion(avanzarTurno(s))  $\equiv$  false
huboConstruccion(unir(s1, s2))  $\equiv$  huboConstruccion(s1)  $\vee$  huboConstruccion(s2)

popularidad(empezarPartida)  $\equiv$  0
popularidad(agregarCasa(s, c))  $\equiv$  popularidad(s)
popularidad(agregarComercio(s, c))  $\equiv$  popularidad(s)
popularidad(avanzarTurno(s))  $\equiv$  popularidad(s)
popularidad(unir(s1, s2))  $\equiv$  popularidad(s1) + popularidad(s2) + 1

unirCasasPriorizandoNivelMaximo(s1, s2)  $\equiv$  unirCasas(s1, s2, casas(s1))  $\cup$  unirCasas(s2, s1, casas(s2))

```

```

unirCasas(s1, s2, casas)  $\equiv$  if  $\emptyset?(casas)$  then
     $\emptyset$ 
else
    if ( $dameUno(casas) \notin casas(s2)$ )  $\vee$ 
        ( $dameUno(casas) \in casas(s2) \wedge_L nivel(s1, dameUno(casas)) \geq nivel(s2, dameUno(casas))$ ) then
        Ag( $dameUno(casas)$ , unirCasas(s1, s2, sinUno(casas)))
    else
        unirCasas(s1, s2, sinUno(casas))
    fi
fi

unirComerciosPriorizandoNivelMaximo(s1, s2)  $\equiv$  unirComercios(s1, s2, comercios(s1))  $\cup$  unirComercios(s2, s1, comercios(s2))

unirComercios(s1, s2, comercios)  $\equiv$  if  $\emptyset?(comercios)$  then
     $\emptyset$ 
else
    if ( $dameUno(comercios) \notin comercios(s2) \wedge dameUno(comercios) \notin casas(s2)$ )  $\vee$ 
        ( $dameUno(comercios) \in comercios(s2) \wedge_L nivel(s1, dameUno(comercios)) \geq nivel(s2, dameUno(comercios))$ ) then
        Ag( $dameUno(comercios)$ , unirComercios(s1, s2, sinUno(comercios)))
    else
        unirComercios(s1, s2, sinUno(comercios))
    fi
fi

casasADistancia3Manhattan(casas, c)  $\equiv$  if  $\emptyset?(casas)$  then
     $\emptyset$ 
else
    if  $distanciaManhattan(dameUno(casas), c) \leq 3$  then
        Ag( $dameUno(casas)$ , casasADistancia3Manhattan(sinUno(casas), c))
    else
        casasADistancia3Manhattan(sinUno(casas), c)
    fi
fi

maximoNivel(s, casas)  $\equiv$  if  $\#(casas) = 1$  then
    nivel(s, dameUno(casas))
else
    max{nivel(s, dameUno(casas)), maximoNivel(s, sinUno(casas))}
fi

darNivelAComercio(s1, s2, c)  $\equiv$  if  $c \in unirComercios(s1, s2, comercios(s1))$  then
    if  $\emptyset?(casasADistancia3Manhattan(unirCasasPriorizandoNivelMaximo(s1, s2), c))$  then
        nivel(s1, c)
    else
        darMaximoNivelComercio(s1, s2, c)
    fi
else
    if  $\emptyset?(casasADistancia3Manhattan(unirCasasPriorizandoNivelMaximo(s1, s2), c))$  then
        nivel(s2, c)
    else
        darMaximoNivelComercio(s1, s2, c)
    fi
fi

```

```

darMaximoNivelComercio(s1, s2, c)  $\equiv$  if  $\emptyset?$ (casasADistancia3Manhattan(unirCasas(s1, s2, casas(s1)), c))
then
    maximoNivel(s2, casasADistancia3Manhattan(unirCasas(s2, s1, casas(s2)), c))
else
    if  $\emptyset?$ (casasADistancia3Manhattan(unirCasas(s2, s1, casas(s2)), c))
    then
        maximoNivel(s1, casasADistancia3Manhattan(unirCasas(s1, s2, casas(s1)), c))
    else
        maximoNivelComercio(s1, s2, c)
    fi
fi

maximoNivelComercio(s1, s2, c)  $\equiv$  max{maximoNivel(s1, casasADistancia3Manhattan(unirCasas(s1, s2, casas(s1)), c)),
    maximoNivel(s2, casasADistancia3Manhattan(unirCasas(s2, s1, casas(s2)), c))}

hayConstruccion(s, c)  $\equiv$   $c \in$  construcciones(s)

sePuedeConstruir(s, c)  $\equiv$   $\neg$ hayConstruccion(s, c)  $\wedge$   $\neg$ hayRio(mapa(s), c)

distanciaManhattan(x, y)  $\equiv$   $|\pi_1(x) - \pi_1(y)| + |\pi_2(x) - \pi_2(y)|$ 

antiguedad(s)  $\equiv$  maximoNivel(s, construcciones(s))

construcciones(s)  $\equiv$  casas(s)  $\cup$  comercios(s)

esCasillaDeMaximoNivel(s, c)  $\equiv$  nivel(s, c) = maximoNivel(s, construcciones(s))

```

Fin TAD