

# Especificación Referencia TP2

Algoritmos y Estructuras de Datos II, DC, UBA.

1er cuatrimestre, 2020

## Índice

<b>1. Renombres de TADs</b>	<b>2</b>
<b>2. TAD MAPA</b>	<b>2</b>
<b>3. TAD SIMCITY</b>	<b>3</b>

## 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  $\text{Nat}$

## 2. TAD MAPA

### TAD MAPA

#### igualdad observacional

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

**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

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

$\text{agregarRios} : \text{Mapa } m \times \text{Conj}(\text{Nat}) \text{ } vs \times \text{Conj}(\text{Nat}) \text{ } hs \longrightarrow \text{Mapa}$

**axiomas**       $\forall m, m1, m2 : \text{Mapa}, \forall c : \text{Casilla}, \forall d : \text{Direccion}, \forall p : \text{Posicion}, \forall hs, vs : \text{Conj}(\text{Nat})$

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

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

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

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

$\text{agregarRios}(m, vs, hs) \equiv$  **if**  $\emptyset?(hs \cup vs)$  **then**  
     $m$   
    **else**  
    **if**  $\emptyset?(hs)$  **then**  
     $\text{agregarRios}(\text{agregarRio}(m, \text{Vertical}, \text{dameUno}(vs)), \text{sinUno}(vs), hs)$   
    **else**  
     $\text{agregarRios}(\text{agregarRio}(m, \text{Horizontal}, \text{dameUno}(hs)), vs, \text{sinUno}(hs))$   
    **fi**  
    **fi**

**Fin TAD**

### 3. TAD SIMCITY

#### igualdad observacional

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

#### TAD SIMCITY

**géneros**      SimCity

**exporta**      SimCity, generadores, observadores, otras operaciones

#### observadores básicos

mapa : SimCity  $\rightarrow$  Mapa

casas : SimCity  $\rightarrow$  Conj(Casilla)

comercios : SimCity  $\rightarrow$  Conj(Casilla)

nivel : SimCity  $s \times \text{Casilla } c \rightarrow \text{Nat}$  {hayConstruccion(s, c)}

huboConstruccion : SimCity  $s \rightarrow \text{bool}$

popularidad : SimCity  $\rightarrow \text{Nat}$

#### generadores

empezarPartida : Mapa  $\rightarrow$  SimCity

agregarCasa : SimCity  $s \times \text{Casilla } c \rightarrow \text{sePuedeConstruir}(s, c)$

agregarComercio : SimCity  $s \times \text{Casilla } c \rightarrow \text{sePuedeConstruir}(s, c)$

avanzarTurno : SimCity  $s \rightarrow \text{huboConstruccion}(s)$

unir : 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

antigüedad : SimCity  $\rightarrow \text{Nat}$

construcciones : SimCity  $\rightarrow \text{Conj}(\text{Casilla})$

hayConstruccion : SimCity  $\times \text{Casilla} \rightarrow \text{Bool}$

sePuedeConstruir : SimCity  $\times \text{Casilla} \rightarrow \text{Bool}$

esCasillaDeMaximoNivel : SimCity  $s \times \text{Casilla } c \rightarrow \text{Bool}$  {hayConstruccion(s, c)}

unirCasasPriorizandoNivelMaximo : SimCity  $s1 \times \text{SimCity } s2 \rightarrow \text{Conj}(\text{Casilla})$

unirComerciosPriorizandoNivelMaximo : SimCity  $s1 \times \text{SimCity } s2 \rightarrow \text{Conj}(\text{Casilla})$

casasADistancia3Manhattan : Conj(Casilla)  $\times \text{Casilla} \rightarrow \text{Conj}(\text{Casilla})$

maximoNivel : SimCity  $s \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Nat}$  {c  $\in$  construcciones(s)}

unirCasas : SimCity  $s1 \times \text{SimCity } s2 \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Conj}(\text{Casilla})$  {c = casas(s1)}

unirComercios : SimCity  $s1 \times \text{SimCity } s2 \times \text{Conj}(\text{Casilla}) c \rightarrow \text{Conj}(\text{Casilla})$  {c = comercios(s1)}

darNivelAComercio : SimCity  $s1 \times \text{SimCity } s2 \times \text{Casilla } c \rightarrow \text{Nat}$  {c  $\in$  comercios(s1)  $\cup$  comercios(s2)}

darMaximoNivelComercio : SimCity  $s1 \times \text{SimCity } s2 \times \text{Casilla } c \rightarrow \text{Nat}$  {c  $\in$  comercios(s1)  $\cup$  comercios(s2)}

maximoNivelComercio : SimCity s1 × SimCity s2 × Casilla c → Nat  
 $\{c \in \text{comercios}(s1) \cup \text{comercios}(s2)\}$

distanciaManhattan : Casilla x × Casilla y → Nat

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

mapa(empezarPartida(mapa)) ≡ mapa

mapa(agregarCasa(s, c)) ≡ mapa(s)

mapa(agregarComercio(s, c)) ≡ mapa(s)

mapa(avanzarTurno(s)) ≡ mapa(s)

mapa(unir(s1, s2)) ≡ unir(mapa(s1), mapa(s2))

casas(empezarPartida(mapa)) ≡ ∅

casas(agregarCasa(s, c)) ≡ Ag(c, casas(s))

casas(agregarComercio(s, c)) ≡ casas(s)

casas(avanzarTurno(s)) ≡ casas(s)

casas(unir(s1, s2)) ≡ unirCasasPriorizandoNivelMaximo(s1, s2)

comercios(empezarPartida(mapa)) ≡ ∅

comercios(agregarCasa(s, c)) ≡ Ag(c, comercios(s))

comercios(agregarComercio(s, c)) ≡ comercios(s)

comercios(avanzarTurno(s)) ≡ comercios(s)

comercios(unir(s1, s2)) ≡ unirComerciosPriorizandoNivelMaximo(s1, s2)

nivel(agregarCasa(s, c), c') ≡ **if** c' = c **then** 0 **else** nivel(s, c') **fi**

nivel(agregarComercio(s, c), c') ≡ **if** c' = c **then**  
     **if** ∅?(casasADistancia3Manhattan(casas(s), c)) **then**  
         0  
     **else**  
         maximoNivel(s, casasADistancia3Manhattan(casas(s), c))  
     **fi**  
**else**  
     nivel(s, c)  
**fi**

nivel(avanzarTurno(s), c) ≡ 1 + nivel(s, c)

nivel(unir(s1, s2), c) ≡ **if** c ∈ unirCasas(s1, s2, casas(s1)) **then**  
     nivel(s1, c)  
**else**  
     **if** c ∈ unirCasas(s2, s1, casas(s2)) **then**  
         nivel(s2, c)  
     **else**  
         darNivelAComercio(s1, s2, c)  
     **fi**  
**fi**

huboConstruccion(empezarPartida(mapa)) ≡ false

huboConstruccion(agregarCasa(s, c)) ≡ true

huboConstruccion(agregarComercio(s, c)) ≡ true

huboConstruccion(avanzarTurno(s)) ≡ false

huboConstruccion(unir(s1, s2)) ≡ huboConstruccion(s1) ∨ huboConstruccion(s2)

popularidad(empezarPartida) ≡ 0

popularidad(agregarCasa(s, c)) ≡ true

```

popularidad(agregarComercio(s, c))  $\equiv$  true
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)  $\notin$  ca-
    sas(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  $c \in \text{unirComercios}(s1, s2, \text{comercios}(s1))$  then
    if  $\emptyset?(\text{casasADistancia3Manhattan}(\text{unirCasasPriorizandoNivelMaximo}(s1, s2), c))$  then
        nivel(s1, c)
    else
        darMaximoNivelComercio(s1,s2,c)
    fi
else
    if  $\emptyset?(\text{casasADistancia3Manhattan}(\text{unirCasasPriorizandoNivelMaximo}(s1, s2), c))$  then
        nivel(s2, c)
    else
        darMaximoNivelComercio(s1, s2, c)
    fi
fi

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

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

hayConstruccion(s, c)  $\equiv c \in \text{construcciones}(s)$ 

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

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

antiguedad(s)  $\equiv \text{maximoNivel}(s, \text{construcciones}(s))$ 

construcciones(s)  $\equiv \text{casas}(s) \cup \text{comercios}(s)$ 

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

```

**Fin TAD**