

Algoritmos y Estructuras de Datos II

Primer Cuatrimestre de 2015

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

Trabajo Práctico 1

Especificación

Integrante	LU	Correo electrónico
INTEGRANTE, 1	123/12	1@gmail.com
INTEGRANTE, 2	123/12	2@gmail.com
INTEGRANTE, 3	123/12	3@gmail.com
INTEGRANTE, 4	123/12	4@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

Índice

1. TAD DATO	3
2. TAD REGISTRO	3
3. TAD TABLA	4
4. TAD BASEDEDATOS	5

1. TAD DATO

TAD DATO

géneros dato

igualdad observacional

$$(\forall d, d' : \text{dato}) \left(d =_{\text{obs}} d' \iff \left((EsNat?(d) = 1 \wedge EsNat?(d) = EsNat?(d')) \Rightarrow_L DNat(d) = \right. \right. \\ \left. \left. DNat(d') \vee (EsNat?(d) = 0 \wedge EsNat?(d') = EsNat?(d')) \Rightarrow_L DString(d) = DString(d') \right) \right)$$

usa Bool, Nat, String

exporta

observadores básicos

EsNat? : dato \rightarrow bool

DNat : dato $d \rightarrow$ nat

$\{EsNat?(d)\}$

DString : dato $d \rightarrow$ string

$\{\neg EsNat?(d)\}$

generadores

NDat : bool \times nat \times string \rightarrow dato

axiomas

EsNat?(NDato(b,n,s)) $\equiv b$

DNat(NDato(b,n,s)) $\equiv n$

DString(NDato(b,n,s)) $\equiv s$

Fin TAD

2. TAD REGISTRO

TAD REGISTRO

géneros reg

usa Lista, Conjunto, Nat, Bool, DATO

exporta

igualdad observacional

$$(\forall r, r' : \text{reg}) \left(r =_{\text{obs}} r' \iff \left(Campos(r) = Campos(r') \wedge (\forall c : \text{String}) \text{tupla}(c, \text{bool}) \in \right. \right. \\ \left. \left. Campos(r) \Rightarrow_L Dato?(r) = Dato?(r') \right) \right)$$

observadores básicos

Campos : reg \rightarrow conj(tupla(string, bool))

Dato? : reg $r \times$ String $c \rightarrow$ Dato

$\{\text{tupla}(c, \text{True}) \in Campos(r) \vee \text{tupla}(c, \text{False}) \in Campos(r)\}$

generadores

NReg : \rightarrow reg

AgCampo : reg $r \times$ String $c \times$ Bool $b \times$ Dato $d \rightarrow$ reg

$\{EsNat?(d) \equiv b\}$

otras operaciones

JRegs : reg $r \times$ reg $s \rightarrow$ reg

$$\text{tgregs} : \text{conj}(\text{campo}) \ c \times \text{reg} \ r \times \text{dicc}(\text{campo} \times \text{dato}) \ d \longrightarrow \text{reg} \\ \{(\forall \ c_1:\text{campo}) \ (c_1 \in c) \Rightarrow (c_1 \in \text{Campos}(r) \vee c_1 \in \text{claves}(d))\}$$
axiomas

$$\text{Campos}(\text{NReg}) \equiv \emptyset$$

$$\text{Campos}(\text{AgCampo}(r, c, b, d)) \equiv \text{Ag}(\text{tupla}(c, b))$$

$$\text{Dato?}(\text{AgCampo}(r, c, b, d), c') \equiv \text{if } c = c' \text{ then } d \text{ else } \text{Dato?}(r) \text{ fi}$$

$$\text{Campos}(\text{JRegs}(r, s)) \equiv \text{Campos}(r) \cup \text{Campos}(s)$$

$$\text{Dato?}(\text{JRegs}(r, s), c) \equiv \text{if } c \in \text{Campos}(r) \text{ then } \text{Dato?}(r, c) \text{ else } \text{Dato?}(s, c) \text{ fi}$$

$$\text{tgregs}(c, r, d) \equiv \text{if } c = \emptyset \text{ then}$$

$$\text{Nreg}$$

$$\text{else}$$

$$\text{if } \text{DameUno}(c) \in \text{Campos}(r) \text{ then}$$

$$\text{AgCampo}(\text{tgregs}(\text{SinUno}(c), r, d), \text{DameUno}(c), \text{Dato?}(r, \text{Dameuno}(c)))$$

$$\text{else}$$

$$\text{AgCampo}(\text{tgregs}(\text{SinUno}(c), r, d), \text{DameUno}(c), \text{Obtener}(\text{DameUno}(c), d))$$

$$\text{fi}$$

$$\text{fi}$$
Fin TAD**3. TAD TABLA****TAD TABLA**

géneros tab

usa

exporta

igualdad observacional

$$(\forall t, t' : \text{tab}) \ (t =_{\text{obs}} t' \iff (\text{Campos}(t) = \text{Campos}(t') \wedge \text{Claves}(t) = \text{Claves}(t')))$$

observadores básicos

$$\text{CamposT} : \text{tab} \longrightarrow \text{conj}(\text{campo})$$

$$\text{Claves} : \text{tab} \longrightarrow \text{conj}(\text{campo})$$

generadores

$$\text{NTab} : \text{conj}(\text{campo}) \ cp \times \text{conj}(\text{campo}) \ cl \longrightarrow \text{tab}$$

$$\{(\forall \ c:\text{campo}) c \in cl \Rightarrow c \in cp\}$$
axiomas

$$\text{Campos}(\text{NTab}(cp, cl)) \equiv cp$$

$$\text{Claves}(\text{NTab}(cp, cl)) \equiv cl$$
Fin TAD

4. TAD BASEDEDATOS

TAD BASEDEDATOS

géneros bds

usa Bool, Nat, String, Conjunto, Dicc(clave,significado), Tupla(),Dato, Tabla, Registro

exporta

observadores básicos

Tablas : bds \longrightarrow conj(tab)

RegistrosT : bds $b \times$ tab $t \longrightarrow$ conj(reg) $\{t \in Tablas(b)\}$

Joins : bds \longrightarrow dicc(tupla(conj(tab),campo),tab))

RegistrosJ : bds $b \times$ tupla(conj(tab) \times campo) $tj \longrightarrow$ conj(reg) $\{tj \in Claves(Joins(b))\}$

Triggers : bds $b \times$ tab $t \longrightarrow$ dicc(tab/ t_2 , dicc(campo/ c , dato) $\{t \in Tablas(b)\}$

generadores

NuevaBase : conj(tab) \longrightarrow bds

AgTab : bds $b \times$ tab $t \longrightarrow$ bds $\{t \notin Tablas(bds)\}$

AgReg : bds $b \times$ tab $t \times$ reg $r \longrightarrow$ bds
 $\left\{ \begin{array}{l} CamposT(t) = Campos(r) \wedge ((\forall r':reg, \forall c:campo)(r' \in \Pi_2(t) \wedge c \in claves(t) \Rightarrow_L (Dato?(r, c) \neq \\ Dato?(r', c))) \end{array} \right\}$

ElimReg : bds $b \times$ tab $t \times$ campo $c \times$ dato $d \longrightarrow$ bds $\{c \in CamposT(t) \wedge EsNat?(d) = \Pi_2(c)\}$

CrearJoin : bds $b \times$ tab $t1 \times$ tab $t2 \times$ campo $cl \longrightarrow$ bds
 $\{t1, t2 \in Tablas(bds) \wedge cl \in campos(t1) \wedge cl \in campos(t2) \wedge tupla(\{t1, t2\}, cl) \notin Claves(Joins(b))\}$

EliminarJoin : bds $b \times$ tab $t1 \times$ tab $t2 \times$ campo $cl \longrightarrow$ bds $\{tupla(\{t1, t2\}, cl) \in Claves(Joins(b))\}$

AgTrigger : bds $b \times$ tab $t1 \times$ tab $t2 \times$ dicc(campo \times dato) *default* \longrightarrow bds
 $\{(claves(t2) \subseteq claves(t1)) \wedge (\forall c:campo(c \in claves(default) \Leftrightarrow (c \in campos(t2) \wedge c \notin campos(t1))))\}$

EliminarTrigger : bds $b \times$ tab $t1 \times$ tab $t2 \longrightarrow$ bds

otras operaciones

TopModificadasT : bds \longrightarrow conj(tupla(tabla,conj(reg),nat)

ERR : conj(reg) \times campo \times dato \longrightarrow conj(reg)

JoinRegistros : conj(reg) \times conj(reg) \times campo \longrightarrow conj(reg)

DameRegCon : conj(reg) \times dato \times campo \longrightarrow conj(reg)

DameRegSin : conj(reg) \times dato \times campo \longrightarrow conj(reg)

axiomas

Tablas(NuevaBase(ct)) $\equiv ct$

Tablas(Agtab(b, t)) $\equiv Ag(t, Tablas(b))$

Tablas(AgReg(b, t, r)) $\equiv Tablas(b)$

Tablas(ElimReg(b, t, r)) $\equiv Tablas(b)$

Tablas(CrearJoin(b, t1, t2, c)) $\equiv Tablas(b)$

Tablas(AgTrigger(b, t1, t2, d)) $\equiv Tablas(b)$

RegistrosT(AgTab(b,t1),t) \equiv **if** $t = t1$ **then** \emptyset **else** $RegistrosT(b,t)$ **fi**

```

RegistrosT(AgReg(b, t1, r), t)  $\equiv$  if  $t = t1$  then
     $Ag(r, RegistrosT(b, t))$ 
else
    if  $t \in claves(Triggers(b, t1))$  then
         $Ag(tgregs(campos(t), r, Obtener(t, Triggers(b, t1))),$ 
         $RegistrosT(b, t))$ 
    else
         $RegistrosT(b, t)$ 
    fi
fi

RegistrosT(ElimReg(b, t1, c, d), t)  $\equiv ERR(RegistrosT(b, t), c, d)$ 

RegistrosT(CrearJoin(b, t1, t2, c))  $\equiv RegistrosT(b, t)$ 

RegistrosT(AgTrigger(b, t1, t2, d))  $\equiv RegistrosT(b, t)$ 

ERR(rs, c, d)  $\equiv$  if  $rs = \emptyset$  then
     $\emptyset$ 
else
    if  $d = Dato?(DameUno(rs), c)$  then
         $ERR(SinUno(rs), c, d)$ 
    else
         $Ag(DameUno(rs), ERR(SinUno(rs), c, d))$ 
    fi
fi

Joins(NuevaBase(ct))  $\equiv vaco$ 

Joins(AgTab(bds, t))  $\equiv Joins(bds)$ 

Joins(CrearJoin(bds, t1, t2, c))  $\equiv Definir(tupla(t1, t2, c), Ntab((CamposT(t1) \cup CamposT(t2) - c),$ 
     $(ClavesT(t1) \cup ClavesT(t2) - c)), Joins(bds))$ 

Joins(EliminarJoin(bds, t1, t2, c))  $\equiv Borrar(tupla(t1, t2, c), Joins(bds))$ 

Joins(AgReg(bds, t, r))  $\equiv Joins(bds)$ 

Joins(ElimReg(bds, t, c, d))  $\equiv Joins(bds)$ 

Joins(AgTrigger(bds, t, c, d))  $\equiv Joins(bds)$ 

RegistrosJ(AgTab(bds, t), tj)  $\equiv RegistrosJ(bds, tj)$ 

RegistrosJ(CrearJoin(bds, t1, t2, c), tj)  $\equiv$  if  $t1 \in \pi_1(tj) \wedge t2 \in \pi_1(tj) \wedge c = \pi_2(tj)$  then
     $JoinRegistros(RegistrosT(t1), RegistrosT(t2), c)$ 
else
     $RegistrosJ(bds, tj)$ 
fi

RegistrosJ(EliminarJoin(bds, t1, t2, c), tj)  $\equiv RegistrosJ(bds, tj)$ 

RegistrosJ(AgReg(bds, t, r), tj)  $\equiv$  if  $t \notin \pi_1(tj)$  then
     $RegistrosJ(bds, tj)$ 
else
    if  $DameRegCon(RegistrosT(\pi_1(tj) - t), Dato?(r, \pi_2(tj)), \pi_2(tj)) \neq \emptyset$ 
    then
         $Ag(Jreg(DameUno(DameRegCon(RegistrosT(\pi_1(tj) - t),$ 
         $Dato?(r, \pi_2(tj)), \pi_2(tj))), r), RegistrosJ(bds, tj))$ 
    else
         $RegistrosJ(bds, tj)$ 
    fi
fi

RegistrosJ(ElimReg(bds, t, c, d), tj)  $\equiv$  if  $t \in \pi_1(tj)$  then
     $DameRegSin(RegistrosJ(bds, tj), d, c)$ 
else
     $RegistrosJ(bds, tj)$ 
fi

RegistrosJ(AgTrigger(bds, t1, t2, d), tj)  $\equiv RegistrosJ(bds, tj)$ 

```

```

Triggers(NuevaBase(c) ,t)  $\equiv \emptyset$ 
Triggers(AgTab(b,t1) ,t)  $\equiv Triggers(b,t)$ 
Triggers(AgReg(b,t1,r) ,t)  $\equiv Triggers(b,t)$ 
Triggers(EliminarReg(b,t1,c,d) ,t)  $\equiv Triggers(b,t)$ 
Triggers(CrearJoin(b,t1,t2,cl) ,t)  $\equiv Triggers(b,t)$ 
Triggers(EliminarJoin(b,t1,t2,cl) ,t)  $\equiv Triggers(b,t)$ 
Triggers(AgTrigger(b,t1,t2,d) ,t)  $\equiv$  if  $t = t1$  then definir(Triggers( $b,t$ ), $t2,d$ ) else Triggers( $b,t$ ) fi
Triggers(EliminarTrigger(b,t1,t2) ,t)  $\equiv$  if  $t = t1$  then borrar(Triggers( $b,t$ ), $t2$ ) else Triggers( $b,t$ ) fi
JoinRegistros(r1,r2,c)  $\equiv$  if  $r1 = \emptyset \vee r2 = \emptyset$  then
     $\emptyset$ 
else
    if DameRegCon( $r1,Dato?(DameUno(r2),c),c) \neq \emptyset$  then
        Ag(JRegs(DameUno( $r2$ ),
            DameUno(DameRegCon( $r1,Dato?(DameUno(r2),c))$ )),
            JoinRegistros( $r1,SinUno(r2),c$ ))
        else
            JoinRegistros( $r1,SinUno(r2),c$ )
        fi
    fi
fi
DameRegCon(r,d,c)  $\equiv$  if  $r = \emptyset$  then
     $\emptyset$ 
else
    if Dato?(DameUno( $r$ ), $c$ ) =  $d$  then
        DameUno( $r$ )
    else
        DameRegCon(SinUno( $r$ ), $d,c$ )
    fi
fi
DameRegSin(r,d,c)  $\equiv$  if  $r = \emptyset$  then
     $\emptyset$ 
else
    if Dato?(DameUno( $r$ ), $c$ ) =  $d$  then
        DameRegSin(SinUno( $r$ ), $d,c$ )
    else
        Ag(DameUno( $r$ ),DameRegSin(SinUno( $r$ ), $d,c$ ))
    fi
fi

```

Fin TAD