

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	4

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 \longrightarrow bool

DNat : dato $d \longrightarrow$ nat

$\{EsNat?(d)\}$

DString : dato $d \longrightarrow$ string

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

generadores

NDat : bool \times nat \times string \longrightarrow 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

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

observadores básicos

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

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

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

generadores

NReg : \longrightarrow reg

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

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

otras operaciones

JRegs : $\text{reg } r \times \text{reg } s \longrightarrow \text{reg}$

axiomas

Campos(NReg) $\equiv \emptyset$

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

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

Campos(JRegs(r, s)) $\equiv Campos(r) \cup Campos(s)$

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

Fin TAD

3. TAD TABLA

TAD TABLA

géneros tab

usa

exporta

igualdad observacional

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

observadores básicos

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

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

generadores

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

axiomas

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

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

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 : $\text{bds} \longrightarrow \text{conj}(\text{tab})$

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

Joins : $\text{bds} \longrightarrow \text{dicc}(tupla(\text{conj}(\text{tab}), \text{campo}), \text{tab})$

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

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

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

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

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

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

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

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

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

$RegistrosT(AgReg(b, t1, r), t) \equiv \text{if } t = t1 \text{ then } Ag(r, RegistrosT(b, t)) \text{ else } RegistrosT(b, t) \text{ fi}$

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

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

$ERR(rs, c, d) \equiv \text{if } rs = \emptyset \text{ then } \emptyset$
 else
 $\text{if } d = Dato?(DameUno(rs), c) \text{ then } ERR(SinUno(rs), c, d)$
 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)
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
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
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
DameRegCon(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), DameRegCon(SinUno(r), d, c))
    fi
fi

```

Fin TAD