

Especificacion de TADs auxiliares de Base de Datos

Trabajo Práctico 1 - 2^{do} cuatrimestre de 2017

Algoritmos y Estructuras de Datos II, DC, UBA.

Versión: 1.3 del 4 de septiembre de 2017 (ver CHANGELOG.md)

Especificación

1. TAD DATO

TAD TIPO es BOOL

TAD DATO

géneros dato

usa string, nat, tipo

exporta generadores, observadores básicos y otras operaciones

igualdad observacional

$$(\forall d_1, d_2 : \text{dato}) \left(d_1 =_{\text{obs}} d_2 \iff \left(\begin{array}{l} \text{tipo?}(d_1) \equiv \text{tipo?}(d_2) \wedge_{\text{L}} \\ ((\text{Nat?}(d_1) \Rightarrow_{\text{L}} \text{valorNat}(d_1) =_{\text{obs}} \text{valorNat}(d_2)) \wedge \\ (\text{String?}(d_1) \Rightarrow_{\text{L}} \text{valorStr}(d_1) =_{\text{obs}} \text{valorStr}(d_2))) \end{array} \right) \right)$$

generadores

datoString : string \longrightarrow dato

datoNat : nat \longrightarrow dato

observadores básicos

tipo? : dato \longrightarrow tipo

valorNat : dato $d \longrightarrow$ nat {Nat?(d)}

valorStr : dato $d \longrightarrow$ string {String?(d)}

otras operaciones

String? : dato \longrightarrow bool

Nat? : dato \longrightarrow bool

axiomas

$(\forall s : \text{string}, \forall n : \text{nat}, \forall d : \text{dato})$

tipo?(datoNat(n)) \equiv true

tipo?(datoString(s)) \equiv false

valorNat(datoNat(n)) \equiv n

$$\begin{aligned}\text{valorStr}(\text{datoString}(s)) &\equiv s \\ \text{Nat?}(d) &\equiv \text{tipo?}(d) \\ \text{String?}(d) &\equiv \neg \text{Nat?}(d)\end{aligned}$$
Fin TAD

2. TAD REGISTRO

TAD CAMPO es STRING

TAD REGISTRO

TAD REGISTRO extiende a DICCIONARIO(CAMPO, DATO)

géneros registro**usa** string, dato, campo, dice**exporta** otras operaciones**otras operaciones** $\text{campos} : \text{registro} \longrightarrow \text{conj}(\text{campo})$ $\text{valor} : \text{campo } c \times \text{registro } r \longrightarrow \text{dato} \quad \{c \in \text{campos}(r)\}$ **axiomas** $(\forall c: \text{campo}, \forall r: \text{registro})$ $\text{campos}(r) \equiv \text{claves}(r)$ $\text{valor}(c, r) \equiv \text{obtener}(c, r)$ **Fin TAD**

3. TAD TABLA

TAD TABLA**géneros** tabla**usa** campo, clave, registro, conj, string**exporta** Generadores, observadores, otras operaciones**igualdad observacional**

$$(\forall t_1, t_2 : \text{tabla}) \left(t_1 =_{\text{obs}} t_2 \iff \left(\begin{aligned} &\text{campos}(t_1) =_{\text{obs}} \text{campos}(t_2) \wedge_{\text{L}} (\forall c: \text{campo}) (c \in \text{campos}(t_1) \iff c \in \text{campos}(t_2)) \\ &\wedge \text{campos}(t_1) \Rightarrow_{\text{L}} \text{tipoCampo}(c, t_1) =_{\text{obs}} \text{tipoCampo}(c, t_2) \wedge \\ &\wedge \text{claves}(t_1) =_{\text{obs}} \text{claves}(t_2) \wedge \text{registros}(t_1) =_{\text{obs}} \text{registros}(t_2) \end{aligned} \right) \right)$$
observadores básicos $\text{claves} : \text{tabla} \longrightarrow \text{conj}(\text{campo})$ $\text{campos} : \text{tabla} \longrightarrow \text{conj}(\text{campo})$ $\text{tipoCampo} : \text{campo } c \times \text{tabla } t \longrightarrow \text{tipo} \quad \{c \in \text{campos}(t)\}$ $\text{registros} : \text{tabla} \longrightarrow \text{conj}(\text{registro})$ **generadores**

nuevaTabla : conj(campo) *claves* × registro *columnas* → tabla
 $\{claves \neq \emptyset \wedge claves \subseteq campos(columnas)\}$
 agregarRegistro : registro *r* × tabla *t* → tabla $\{campos(r) =_{obs} campos(t) \wedge puedoInsertar?(r, t)\}$

otras operaciones

puedoInsertar? : registro *r* × tabla *t* → bool
 compatible : registro × tabla → bool

hayCoincidencia : registro *r* × conj(campo) *claves* × conj(registro) *cr* → bool
 $\{(\forall r': registro)(r' \in \{r\} \cup cr) \Rightarrow_L claves \subseteq campos(r')\}$
 filtrarRegistros : registro *r* × campo *clave* × conj(registro) *rs* → conj(registro)
 $\{(\forall r': registro)(r' \in \{r\} \cup cr) \Rightarrow_L clave \in campos(r')\}$
 mismosTipos : conj(campo) *cs* × registro *r* × tabla *t* → bool
 $\{cs \subseteq campos(r) \wedge cs \subseteq campos(t)\}$

axiomas

($\forall claves: conj(campo), \forall t: tabla, \forall r: registro, \forall regs: conj(registro) \forall nombre: string$)

$claves(nuevaTabla(claves, r)) \equiv claves$

$claves(agregarRegistro(r, t)) \equiv claves(t)$

$campos(nuevaTabla(claves, r)) \equiv campos(r)$

$campos(agregarRegistro(r, t)) \equiv campos(t)$

$tipoCampo(c, nuevaTabla(claves, r)) \equiv tipo?(valor(c, r))$

$tipoCampo(c, agregarRegistro(r, t)) \equiv tipoCampo(c, t)$

$registros(nuevaTabla(claves, r)) \equiv \emptyset$

$registros(agregarRegistro(r, t)) \equiv Ag(r, registros(t))$

$puedoInsertar?(r, t) \equiv compatible(r, t) \wedge \neg hayCoincidencia(r, claves(t), registros(t))$

$compatible(r, t) \equiv campos(r) = campos(t) \wedge_L mismosTipos(campos(t), r, t)$

$hayCoincidencia(r, claves, regs) \equiv (\emptyset?(claves) \wedge \neg \emptyset?(regs)) \vee_L (\neg \emptyset?(regs) \wedge_L hayCoincidencia(r, sinUno(claves), filtrarRegistros(r, dameUno(c), regs)))$

$filtrarRegistros(r, c, regs) \equiv \text{if } \emptyset?(regs) \text{ then } regs$
 else
 $\text{if } valor(c, r) == valor(c, dameUno(regs)) \text{ then}$
 $Ag(dameUno(regs), filtrarRegistros(r, c, sinUno(regs)))$
 else
 $filtrarRegistros(r, c, sinUno(regs))$
 fi

$mismosTipos(cs, r, t) \equiv \emptyset?(cs) \vee_L (mismosTipos(sinUno(cs), r, t) \wedge tipo?(valor(dameUno(cs), r)) \equiv tipoCampo(dameUno(cs), t))$

Fin TAD