

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: 1.1 del 24 de agosto de 2017 (ver Changelog.txt)

Especificación

1. TAD DATO

TAD TIPO es BOOL

TAD DATO

| g neros | 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

$$\text{datoString} : \text{string} \longrightarrow \text{dato}$$
$$\text{datoNat} \quad : \quad \text{nat} \quad \longrightarrow \quad \text{dato}$$

observadores básicos

$$\text{tipo?} \quad : \text{ dato} \quad \longrightarrow \text{tipo}$$
$$\text{valorNat} : \text{dato } d \longrightarrow \text{nat} \quad \{\text{Nat?}(d)\}$$
$$\text{valorStr} : \text{dato } d \longrightarrow \text{string} \quad \{\text{String?}(d)\}$$

otras operaciones

$$\text{String?} \quad : \text{ dato} \quad \longrightarrow \text{ bool}$$
$$\text{Nat?} \quad : \text{dato} \quad \longrightarrow \text{bool}$$

axiomas

$$(\forall s: \text{string}, \forall n: \text{nat}, \forall d: \text{dato})$$
$$\text{tipo?}(\text{datoNat}(n)) \quad \equiv \quad \text{true}$$

```
tipo?(datoString(s))    ≡ false
```

$$\text{valorNat}(\text{datoNat}(n)) \quad \equiv \quad n$$

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

$$(\forall t_1, t_2 : \text{tabla}) \left(t_1 =_{\text{obs}} t_2 \iff \left(\begin{aligned} &\text{campos}(t_1) =_{\text{obs}} \text{campos}(t_2) \wedge_{\text{L}} (\forall c: \text{campo}) (c \in) \\ &\text{campos}(t_1) \Rightarrow_{\text{L}} \text{tipoCampo}(c, t_1) =_{\text{obs}} \text{tipoCampo}(c, t_2) \wedge \\ &\text{claves}(t_1) =_{\text{obs}} \text{claves}(t_2) \wedge \text{registros}(t_1) =_{\text{obs}} \text{registros}(t_2) \end{aligned} \right) \right)$$
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* ≠ ∅ ∧ *claves* ⊆ campos(*columnas*)}

agregarRegistro : registro *r* × tabla *t* → tabla {campos(*r*) =_{obs} campos(*t*) ∧ 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
 {(∀ *r'*: registro)(*r'* ∈ {*r*} ∪ *cr*) ⇒_L *claves* ⊆ campos(*r'*)}

filtrarRegistros : registro *r* × campo *clave* × conj(registro) *rs* → conj(registro)
 {(∀ *r'*: registro)(*r'* ∈ {*r*} ∪ *cr*) ⇒_L *clave* ∈ campos(*r'*)}

mismosTipos : conj(campo) *cs* × registro *r* × tabla *t* → bool
 {*cs* ⊆ campos(*r*) ∧ *cs* ⊆ campos(*t*)}

axiomas

(∀ *claves*: conj(campo), ∀ *t*: tabla, ∀ *r*: registro, ∀ *regs*: conj(registro) ∀ *nombre*: string)

claves(nuevaTabla(*claves*, *r*)) ≡ *claves*

claves(agregarRegistro(*r*, *t*)) ≡ claves(*t*)

campos(nuevaTabla(*claves*, *r*)) ≡ campos(*r*)

campos(agregarRegistro(*r*, *t*)) ≡ campos(*t*)

tipoCampo(*c*, nuevaTabla(*claves*, *r*)) ≡ tipo?(valor(*c*, *r*))

tipoCampo(*c*, agregarRegistro(*r*, *t*)) ≡ tipoCampo(*c*, *t*)

registros(nuevaTabla(*claves*, *r*)) ≡ ∅

registros(agregarRegistro(*r*, *t*)) ≡ Ag(*r*, registros(*t*))

puedoInsertar?(*r*, *t*) ≡ compatible(*r*, *t*) ∧ ¬ hayCoincidencia(*r*, claves(*t*), registros(*t*))

compatible(*r*, *t*) ≡ campos(*r*) = campos(*t*) ∧_L mismosTipos(campos(*t*), *r*, *t*)

hayCoincidencia(*r*, *claves*, *regs*) ≡ (∅?(*claves*) ∧ ¬∅?(*regs*)) ∨_L (¬∅?(*regs*) ∧_L hayCoincidencia(*r*, sinUno(*claves*), filtrarRegistros(*r*, dameUno(*c*), *regs*)))

filtrarRegistros(*r*, *c*, *regs*) ≡ **if** ∅?(*regs*) **then**
 regs
else
 if valor(*c*, *r*) == valor(*c*, dameUno(*regs*)) **then**
 Ag(dameUno(*regs*), filtrarRegistros(*r*, *c*, sinUno(*regs*)))
 else
 filtrarRegistros(*r*, *c*, sinUno(*regs*))
fi

mismosTipos(*cs*, *r*, *t*) ≡ ∅?(*cs*) ∨_L (mismosTipos(sinUno(*cs*), *r*, *t*) ∧ tipo?(valor(dameUno(*cs*), *r*)) ≡ tipoCampo(dameUno(*cs*), *t*))

Fin TAD