

Algoritmos y Estructuras de Datos II

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

Trabajo Práctico de Especificación

Grupo 1

Integrante	LU	Correo electrónico
Bálsamo, Facundo	874/10	facundobalsamo@gmail.com
Lasso, Nicolás	892/10	lasso.nico@gmail.com
Rodríguez, Agustín	120/10	agustinrodriguez90@hotmail.com
Tripodi, Guido	843/10	guido.tripodi@hotmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. TAD LINKLINKIT

TAD LINKLINKIT

géneros linkLinkIT

exporta linkLinkIT, generadores, observadores

usa ARBOLCATEGORIAS, LINK, CATEGORIA, FECHA

igualdad observacional

$$(\forall l, l' : \text{linkLinkIT}) \left(l =_{\text{obs}} l' \iff \left(\begin{array}{l} \text{categorias}(l) =_{\text{obs}} \text{categorias}(l') \wedge \\ \text{accesos}(l) =_{\text{obs}} \text{accesos}(l') \wedge \\ \text{linksXCategoria}(l) =_{\text{obs}} \text{linksXCategoria}(l') \end{array} \right) \right)$$

observadores básicos

categorias : linkLinkIT \longrightarrow arbolDeCategorias
accesos : linkLinkIT $it \times \text{link } l \longrightarrow \text{secu}(\text{tupla}(\text{fecha}, \text{nat})) \quad \{l \in \text{LinkLinkIT}(it)\}$
linksXCategoria : linkLinkIT $it \times \text{categoria } c \longrightarrow \text{conj}(\text{link}) \quad \{c \in \text{dameTodos}(\text{categorias}(it))\}$

generadores

iniciar : arbolCategorias \longrightarrow linkLinkIT
agregarLink : linkLinkIT $it \times \text{link } l \times \text{categoria } c \longrightarrow$ linkLinkIT
 $\{ \neg(l \in \text{links}(it)) \ \&\& \ c \in \text{dameTodos}(\text{categorias}(it)) \}$
accederLink : linkLinkIT $it \times \text{link } l \times \text{fecha } f \longrightarrow$ linkLinkIT
 $\{ l \in \text{links}(it) \ \&\& \ f \geq \text{ultimoAcceso}(it) \}$

otras operaciones

links : linkLinkIT $\longrightarrow \text{conj}(\text{link})$
#accesos3Dias : linkLinkIT $it \times \text{link } l \longrightarrow \text{nat} \quad \{l \in \text{links}(it)\}$
#LinksXCategoria : linkLinkIT $it \times \text{categoria } c \longrightarrow \text{nat} \quad \{c \in \text{dameTodos}(\text{categorias}(it))\}$
ultimoAcceso : linkLinkIT $\longrightarrow \text{fecha}$

axiomas $\forall it, it' : \text{linklinkIT}$
 $\forall a : \text{arbolDeCategorias}$
 $\forall c : \text{categoria}$
 $\forall l : \text{link}$
 $\forall f : \text{fecha}$

categorias(iniciar(a)) $\equiv a$
categorias(agregarLink(it, l, c)) $\equiv \text{categorias}(it)$
categorias(accederLink(it, l, f)) $\equiv \text{categorias}(it)$

accesos(agregarLink(it, l, c), l') $\equiv \emptyset$
accesos(accederLink(it, l, f), l') \equiv **if** $l = l'$ **then** agregarAcceso(accesos(it, l'), f)
else accesos(it, l')
fi

linksXCategoria(iniciar(a), c) $\equiv \emptyset$

```

linksXCategoria(agregarLink(it, l, c),c')    ≡ if c = c' then
                                             Ag(l, linksXCategoria(it,c))
                                             else
                                             linksXCategoria(it,c')
                                             fi
linksXCategoria(accederLink(it, l, f),c')    ≡ linksXCategoria(it,c')

links(it)                                     ≡ todosLosLinks((dameTodos(categorias(it))),it)
todosLosLinks(c,it)                         ≡ if  $\emptyset?(c)$  then
                                              $\emptyset$ 
                                             else
                                             linksXCategoria(it,dameUno(c))       $\cup$       linksXCategoria(it,sinUno(c))
                                             fi

#LinksXCategoria(it,c)                     ≡ #LinksXCategoria(it,c)

#accesos3Dias(it,l)                         ≡ contarAccesos3Dias(accesos(it,l),ultimoAcceso(it))

contarAccesos3Dias(ls, f)                   ≡ if vacia?(ls) then
                                             0
                                             else
                                             if  $f - \prod_1(\text{prim}(ls)) \leq 3$  then
                                              $\prod_2(\text{prim}(ls)) + \text{contarAccesos3Dias}(\text{fin}(ls), f)$ 
                                             else
                                             0
                                             fi
                                             fi

ultimoAcceso(it)                             ≡ accesoFinal(links(it))
accesoFinal(ls)                             ≡ if  $\prod_1(\text{prim}(\text{accesos}(\text{dameUno}(ls)))) \leq \text{accesoFinal}(\text{sinUno}(ls))$ 
                                             then
                                              $\prod_1(\text{prim}(\text{accesos}(\text{dameUno}(ls))))$ 
                                             else
                                             accesoFinal(sinUno(ls))
                                             fi

```

Fin TAD

2. TAD ARBOLDECATEGORIAS

TAD ARBOLDECATEGORIAS

géneros arbolDeCategorias

exporta arbolDeCategorias, generadores, observadores

usa CATEGORIA

igualdad observacional

$$(\forall ac, ac' : \text{arbolDeCategorias}) \left(ac =_{\text{obs}} ac' \iff \begin{pmatrix} \text{raiz}(ac) =_{\text{obs}} \text{raiz}(ac') \wedge \\ \text{hijos}(ac) =_{\text{obs}} \text{hijos}(ac') \end{pmatrix} \right)$$

observadores básicos

$$\begin{array}{lll} \text{raiz} & : \text{arbolDeCategorias} & \longrightarrow \text{categoria} \\ \text{hijos} & : \text{arbolDeCategorias} & \longrightarrow \text{conj}(\text{arbolDeCategorias}) \end{array}$$

generadores

$$\text{AC} : \text{conj}(\text{arbolDeCategorias}) \quad a \times \longrightarrow \text{arbolDeCategorias} \quad \{\neg \text{estaEnAlguno}(c, a)\}$$

otras operaciones

dameTodos	:	arbolDeCategorias	\longrightarrow	conj(categoria)	
losFamiliares	:	categoria $c \times$ arbolDeCategorias ac	\longrightarrow	conj(categoria)	$\{c \in dameTodos(ac)\}$
estaEnAlguno	:	arbolDeCategorias \times categoria	\longrightarrow	bool	

axiomas $\forall a$: arbolDeCategorias
 $\forall c$: categoria

```

raiz(AC(a,c))           ≡ c
hijos(AC(a,c))          ≡ a
estaEnAlguno(AC(a,c),c') ≡ if c = c' || c' ∈ a then true else false fi
losFamiliares(c,ac)      ≡ if c = raiz(ac) then dameTodos(ac) else losParientes(c,hijos(ac)) fi
losParientes(c,conj)     ≡ if ∅(conj) then
                           ∅
                           else
                           if estaEnAlguno(dameUno(conj),c) then
                               losFamiliares(c,dameUno(conj))
                           else
                               losParientes(c,sinUno(conj))
                           fi
                           fi
dameTodos(a)             ≡ Ag(raiz(ac),todosLosDeAbajo(hijos(a)))
todosLosDeAbajo(ac)      ≡ if ∅(ac) then
                           ∅
                           else
                               dameTodos(dameUno(ac)) ∪ todosLosDeAbajo(sinUno(ac))
                           fi
estaEnAlguno(ac,c)       ≡ if c ∈ dameTodos(ac) then true else false fi

```

Fin TAD

3. Renombres

TAD CATEGORIA

es String

Fin TAD

TAD LINK

es String

Fin TAD

TAD FECHA

es Nat

Fin TAD