

Algoritmos y Estructuras de Datos II

Trabajo Práctico 1

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

Segundo Cuatrimestre de 2012

Grupo 9

Apellido y Nombre	LU	E-mail
María Candela Capra Coarasa	234/11	canduh_27@hotmail.com
Leandro Lovisolo	645/11	leandro@leandro.me
Gastón de Orta	244/11	gaston.deorta@hotmail.com
Lautaro José Petaccio	443/11	lausuper@gmail.com

Reservado para la cátedra

Instancia	Docente que corrigió	Calificación
Primera Entrega		
Recuperatorio		

Índice

1. TAD CATEGORÍA	2
2. TAD LINK	2
3. TAD DÍA	2
4. TAD ÁRBOLDECATEGORÍAS	2
5. TAD LINKLINKIT	3

1. TAD CATEGORÍA

TAD CATEGORÍA es STRING

2. TAD LINK

TAD LINK es STRING

3. TAD DÍA

TAD DÍA es NAT

4. TAD ÁRBOLDECATEGORÍAS

TAD ÁRBOLDECATEGORÍAS

géneros árbolDeCategorías

exporta árbolDeCategorías, generadores, observadores básicos, contiene?, buscarNodo, aplanar

usa BOOL, NAT, CATEGORÍA, CONJUNTO(CATEGORÍA), CONJUNTO(ÁRBOLDECATEGORÍAS),
SECUENCIA(CATEGORÍA), SECUENCIA(ÁRBOLDECATEGORÍAS)

igualdad observacional

$$(\forall a, a' : \text{árbolDeCategorías}) \quad (a =_{\text{obs}} a' \iff (\text{raíz}(a) =_{\text{obs}} \text{raíz}(a') \wedge \text{hijos}(a) =_{\text{obs}} \text{hijos}(a')))$$

generadores

$$\begin{aligned} \text{árbol} & : \text{categoría } c \times \text{conj}(\text{árbolDeCategorías}) \text{ } as \longrightarrow \text{árbolDeCategorías} \\ & \{ (\forall c' : \text{categoría}) (\# \text{Ocurrencias}(c', c \bullet \text{aplanarSecuencia}(\text{transformarEnSecuencia}(as))) \leq 1) \} \end{aligned}$$

observadores básicos

$$\begin{aligned} \text{raíz} & : \text{árbolDeCategorías} \longrightarrow \text{categoría} \\ \text{hijos} & : \text{árbolDeCategorías} \longrightarrow \text{conj}(\text{árbolDeCategorías}) \end{aligned}$$

otras operaciones

$$\begin{aligned} \text{contiene?} & : \text{categoría} \times \text{árbolDeCategorías} \longrightarrow \text{bool} \\ \text{buscarNodo} & : \text{categoría } c \times \text{árbolDeCategorías } a \longrightarrow \text{árbolDeCategorías} \\ & \{ \text{contiene?}(c, a) \} \\ \text{buscarNodoEnConjunto} & : \text{categoría} \times \text{conj}(\text{árbolDeCategorías}) \longrightarrow \text{árbolDeCategorías} \\ \text{aplanar} & : \text{árbolDeCategorías} \longrightarrow \text{conj}(\text{categoría}) \\ \text{aplanarConjunto} & : \text{conj}(\text{árbolDeCategorías}) \longrightarrow \text{conj}(\text{categoría}) \\ \text{transformarEnSecuencia} & : \text{conj}(\text{árbolDeCategorías}) \longrightarrow \text{secu}(\text{árbolDeCategorías}) \\ \text{aplanarSecuencia} & : \text{secu}(\text{árbolDeCategorías}) \longrightarrow \text{secu}(\text{categoría}) \\ \# \text{Ocurrencias} & : \text{categoría} \times \text{secu}(\text{categoría}) \longrightarrow \text{nat} \end{aligned}$$

axiomas $\forall c: \text{categoría}, \forall a: \text{árbolDeCategorías}, \forall as: \text{conj}(\text{árbolDeCategorías}),$
 $\forall scs: \text{secu}(\text{categoría}), \forall sas: \text{secu}(\text{árbolDeCategorías})$

$$\begin{aligned} \text{raíz}(\text{árbol}(c, as)) & \equiv c \\ \text{hijos}(\text{árbol}(c, as)) & \equiv as \\ \text{contiene?}(c, a) & \equiv c \in \text{aplanar}(a) \\ \text{buscarNodo}(c, a) & \equiv \text{buscarNodoEnConjunto}(c, \text{Ag}(a, \emptyset)) \end{aligned}$$

registrarAcceso : link $l \times$ día $d \times$ linkLinkIt lli \longrightarrow linkLinkIt
 $\{l \in \text{links}(lli) \wedge d \geq \text{díaÚltimoAcceso}(lli)\}$

observadores básicos

darÁrbolDeCategorías : linkLinkIt \longrightarrow árbolDeCategorías
 links : linkLinkIt \longrightarrow conj(link)
 categoríaLink : link $l \times$ linkLinkIt lli \longrightarrow categoría $\{l \in \text{links}(lli)\}$
 #AccesosPorDía : link $l \times$ día $d \times$ linkLinkIt lli \longrightarrow nat
 $\{l \in \text{links}(lli) \wedge d \in \text{díasAccesosRecientes}(lli)\}$

otras operaciones

díaÚltimoAcceso : linkLinkIt \longrightarrow día
 díasAccesosRecientes : linkLinkIt \longrightarrow conj(día)
 #LinksEnCategoría : categoría $c \times$ linkLinkIt lli \longrightarrow nat
 $\{\text{contiene?}(c, \text{darÁrbolDeCategorías}(lli))\}$
 filtrarYContarLinksPorCategoría : conj(links) $ls \times$ conj(categoría) $cs \times$ linkLinkIt lli \longrightarrow bool
 $\{ls \subseteq \text{links}(lli)\}$
 #AccesosMásRecientes : link $l \times$ linkLinkIt lli \longrightarrow nat
 $\{l \in \text{links}(lli)\}$
 #AccesosEnConjuntoDeDías : link $l \times$ conj(día) \times linkLinkIt lli \longrightarrow nat
 $\{l \in \text{links}(lli)\}$

axiomas $\forall c$: categoría, $\forall cs$: conj(categoría), $\forall l, l'$: link, $\forall ls$: conj(link), $\forall d, d'$: día, $\forall ds$: conj(día),
 $\forall a$: árbolDeCategorías, $\forall lli$: linkLinkIt

darÁrbolDeCategorías(iniciar(a)) $\equiv a$
 darÁrbolDeCategorías(registrarLink(l, c, lli)) $\equiv \text{darÁrbolDeCategorías}(lli)$
 darÁrbolDeCategorías(registrarAcceso(l, d, lli)) $\equiv \text{darÁrbolDeCategorías}(lli)$
 links(iniciar(a)) $\equiv \emptyset$
 links(registrarLink(l, c, lli)) $\equiv \text{Ag}(l, \text{links}(lli))$
 links(registrarAcceso(l, d, lli)) $\equiv \text{links}(lli)$
 categoríaLink($l, \text{registrarLink}(l', c, lli)$) $\equiv \text{if } l = l' \text{ then } c \text{ else } \text{categoríaLink}(l, lli) \text{ fi}$
 categoríaLink($l, \text{registrarAcceso}(l', d, lli)$) $\equiv \text{categoríaLink}(l, lli)$
 #AccesosPorDía($l, d, \text{iniciar}(a)$) $\equiv 0$
 #AccesosPorDía($l, d, \text{registrarLink}(l', c, lli)$) $\equiv \text{\#AccesosPorDía}(l, d, lli)$
 #AccesosPorDía($l, d, \text{registrarAcceso}(l', d', lli)$) $\equiv (\text{if } l = l' \wedge d = d' \text{ then } 1 \text{ else } 0 \text{ fi}) + \text{\#AccesosPorDía}(l, d, lli)$
 díaÚltimoAcceso(iniciar(a)) $\equiv 0$
 díaÚltimoAcceso(registrarLink(l, c, lli)) $\equiv \text{díaÚltimoAcceso}(lli)$
 díaÚltimoAcceso(registrarAcceso(l, d, lli)) $\equiv d$
 díasAccesosRecientes(iniciar(a)) $\equiv \emptyset$
 díasAccesosRecientes(registrarLink(l, c, lli)) $\equiv \text{díasAccesosRecientes}(lli)$

```

díasAccesosRecientes(registrarAcceso(l, d, lli))  ≡  if d = 0 then
    Ag(0, ∅)
else
    (if d = 1 then
        Ag(1, Ag(0, ∅))
    else
        Ag(d, Ag(d - 1, Ag(d - 2, ∅)))
    fi)
fi

#LinksEnCategoría(c, lli)  ≡  filtrarYContarLinksPorCategoría(
    links(lli),
    aplanar(buscarNodo(c, darÁrbolDeCategorías(lli))),
    lli)

filtrarYContarLinksPorCategoría(ls, cs, lli)  ≡  if ∅?(ls) then
    0
else
    (if categoríaLink(dameUno(ls), lli) ∈ cs then
        1
    else
        0
    fi) +
    filtrarYContarLinksPorCategoría(sinUno(ls), cs, lli)
fi

#AccesosMásRecientes(l, lli)  ≡  #AccesosEnConjuntoDeDías(
    l, díasAccesosRecientes(lli), lli)

#AccesosEnConjuntoDeDías(l, ds, lli)  ≡  if ∅?(ds) then
    0
else
    #AccesosPorDía(l, dameUno(ds), lli) +
    #AccesosEnConjuntoDeDías(l, sinUno(ds), lli)
fi

```

Fin TAD