

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 **lli**

exporta generadores, categorias, links, categoriaLink, fechaActual, fechaUltimoAcceso, accesosRecientesDia, esReciente?, accesosRecientes, linksOrdenadosPorAccesos, cantLinks

usa BOOL, NAT, CONJUNTO, SECUENCIA, ARBOLCATEGORIAS

observadores básicos

| | | | |
|---------------------|--|------------------------------|--------------------------|
| categorias | : lli s | \longrightarrow acat | |
| links | : lli s | \longrightarrow conj(link) | |
| categoriaLink | : lli \times link | \longrightarrow categoria | |
| fechaActual | : lli | \longrightarrow fecha | |
| fechaUltimoAcceso | : lli $s \times$ link l | \longrightarrow fecha | $\{l \exists links(s)\}$ |
| accesosRecientesDia | : lli $s \times$ link $l \times$ fecha f | \longrightarrow nat | |

generadores

| | | | |
|-----------|--|-----------------------|---|
| iniciar | : acat ac | \longrightarrow lli | |
| nuevoLink | : lli $s \times$ link $l \times$ categoria c | \longrightarrow lli | $\{\neg(l \exists links(s)) \wedge esta?(c, categorias(s))\}$ |
| acceso | : lli $s \times$ link $l \times$ fecha f | \longrightarrow lli | $\{l \exists links(s) \wedge f \geq fechaActual(s)\}$ |

otras operaciones

| | | | |
|-----------------------------|---|-------------------------------|--|
| esReciente? | : lli $s \times$ link $l \times$ fecha f | \longrightarrow bool | $\{l \exists links(s)\}$ |
| accesosRecientes | : lli $s \times$ categoria $c \times$ link l | \longrightarrow nat | $\{esta?(c, categorias(s)) \wedge l \exists links(s) \wedge esSubCategoria(categorias(s), c, categoriaLink(s, l))\}$ |
| linksOrdenadosPorAccesos | : lli $s \times$ categoria c | \longrightarrow secu(link) | $\{esta?(c, categorias(s))\}$ |
| cantLinks | : lli $s \times$ categoria c | \longrightarrow nat | $\{esta?(c, categorias(s))\}$ |
| menorReciente | : lli $s \times$ link l | \longrightarrow fecha | $\{l \exists links(s)\}$ |
| diasRecientes | : lli $s \times$ link l | \longrightarrow fecha | $\{l \exists links(s)\}$ |
| diasRecientesDesde | : lli $s \times$ link l | \longrightarrow fecha | $\{l \exists links(s)\}$ |
| linksCategoriasOHijos | : lli $s \times$ categoria c | \longrightarrow conj(link) | $\{esta?(c, categorias(s))\}$ |
| filtrarLinksCategoriaOHijos | : lli $s \times$ categoria $c \times$ conj(link) ls | \longrightarrow conj(link) | $\{esta?(c, categorias(s)) \wedge ls \subseteq links(s)\}$ |
| diasRecientesParaCategoria | : lli $s \times$ categoria c | \longrightarrow conj(fecha) | $\{esta?(c, categorias(s))\}$ |
| linkConUltimoAcceso | : lli $s \times$ categoria $c \times$ conj(link) ls | \longrightarrow link | $\{esta?(c, categorias(s)) \wedge \neg \emptyset?(ls) \wedge ls \subseteq linksCategoriasOHijos(s, c)\}$ |
| sumarAccesosRecientes | : lli $s \times$ link $l \times$ conj(fecha) fs | \longrightarrow nat | $\{l \exists links(s) \wedge fs \subseteq diasRecientes(s, l)\}$ |
| linksOrdenadosPorAccesosAux | : lli $s \times$ categoria $c \times$ conj(link) ls | \longrightarrow secu(link) | $\{esta?(c, categorias(s)) \wedge ls \subseteq linksCategoriasOHijos(s, c)\}$ |
| linkConMasAccesos | : lli $s \times$ categoria $c \times$ conj(link) ls | \longrightarrow link | $\{esta?(c, categorias(s)) \wedge ls \subseteq linksCategoriasOHijos(s, c)\}$ |
| β | : bool b | \longrightarrow nat | |

axiomas $\forall it, it': linklinkIT$
 $\forall a: arbolDeCategorias$
 $\forall c: categoria$
 $\forall l: link$
 $\forall f: fecha$
 $\forall cc: conj(categoria)$

$\text{categorias}(\text{iniciar}(\text{ac})) \equiv \text{ac}$

$\text{categorias}(\text{nuevoLink}(s, l, c)) \equiv \text{categorias}(\text{ac})$

$\text{categorias}(\text{acceso}(s, l, f)) \equiv \text{categorias}(\text{ac})$

$\text{links}(\text{iniciar}(\text{ac})) \equiv \emptyset$

$\text{links}(\text{nuevoLink}(s, l, c)) \equiv \text{Ag}(l, \text{links}(s))$

$\text{links}(\text{acceso}(s, l, f)) \equiv \text{links}(s)$

$\text{categoriaLink}(\text{nuevoLink}(s, l, c), l') \equiv \text{if } l == l' \text{ then } c \text{ else } \text{categoriaLink}(s, l') \text{ fi}$

$\text{categoriaLink}(\text{acceso}(s, l, f), l') \equiv \text{categoriaLink}(s, l')$

$\text{fechaActual}(\text{iniciar}(\text{ac})) \equiv 0$

$\text{fechaActual}(\text{nuevoLink}(s, l, c)) \equiv \text{fechaActual}(s)$

$\text{fechaActual}(\text{acceso}(s, l, f)) \equiv f$

$\text{fechaUltimoAcceso}(\text{nuevoLink}(s, l, c), l') \equiv \text{if } l == l' \text{ then } \text{fechaActual}(s) \text{ else } \text{fechaUltimoAcceso}(s, l') \text{ fi}$

$\text{fechaUltimoAcceso}(\text{acceso}(s, l, f), l') \equiv \text{fechaUltimoAcceso}(s, l')$

$\text{menorReciente}(s, l) \equiv \max(\text{fechaUltimoAcceso}(s, l) + 1, \text{diasRecientes}) - \text{diasRecientes}$

$\text{esReciente?}(s, l, f) \equiv \text{menorReciente}(s, l) \leq f \wedge f \leq \text{fechaUltimoAcceso}(s, l)$

$\text{accesoRecienteDia}(\text{nuevoLink}(s, l, c), l', f) \equiv \text{if } l == l' \text{ then } 0 \text{ else } \text{accesoRecienteDia}(s, l', f) \text{ fi}$

$\text{accesoRecienteDia}(\text{acceso}(s, l, f), l', f') \equiv \beta(l == l' \wedge f == f') + \text{if } \text{esReciente?}(s, l, f') \text{ then } \text{accesoRecienteDia}(s, l', f') \text{ else } 0 \text{ fi}$

$\text{accesosRecientes}(s, c, l) \equiv \text{sumarAccesosRecientes}(s, l, \text{diasRecientesParaCategoria}(s, c) \cap \text{diasRecientes}(s, l))$

$\text{linksOrdenadosPorAccesos}(s, c) \equiv \text{linksOrdenadosPorAccesosAux}(s, c, \text{linksCategoriaOHijos}(s, c))$

$\text{linksOrdenadosPorAccesosAux}(s, c, ls) \equiv \text{if } \emptyset?(ls) \text{ then}$

\emptyset

else

$\text{linkConMasAccesos}(s, c, ls) \bullet \text{linksOrdenadosPorAccesosAux}(s, c, ls - \text{linkConMasAccesos}(s, c, ls))$

fi

$\text{linkConMasAccesos}(s, c, ls) \equiv \text{if } \#ls == 1 \text{ then}$

$\text{dameUno}(ls)$

else

$\text{if } \text{accesosRecientes}(s, c, \text{dameUno}(ls)) > \text{accesosRecientes}(s, c, \text{linkConMasAccesos}(s, c, \text{sinUno}(ls))) \text{ then}$
 $\text{dameUno}(ls)$

else

$\text{linkConMasAccesos}(s, c, \text{sinUno}(ls))$

fi

fi

$\text{cantLinks}(s, c) \equiv \#\text{linksCategoriaOHijos}(s, c)$

$\text{diasRecientes}(s, l) \equiv \text{diasRecientesDesde}(s, l, \text{menorReciente}(s, l))$

$\text{diasRecientesDesde}(s, l, f) \equiv \text{if } \text{esReciente?}(s, l, f) \text{ then } \text{Ag}(f, \text{diasRecientesDesde}(s, l, f+1)) \text{ else } \emptyset \text{ fi}$

```

linksCategoriaOHijos(s, c)  $\equiv$  filtrarLinksCategoriaOHijos(s, c, links(s))
filtrarLinksCategoriaOHijos(s, c, ls)  $\equiv$  if  $\emptyset?(ls)$  then
     $\emptyset$ 
else
    (if esSubCategoria(categorias(s),c,categoriaLink(s,dameUno(ls)))
    then
        dameUno(ls)
    else
         $\emptyset$ 
    fi)  $\cup$  filtrarLinksCategoriaOHijos(s, c, siunUno(ls))
fi
diasRecientesParaCategoria(s, c)  $\equiv$  if  $\emptyset?(linksCategoriaOHijos(s,c))$  then
     $\emptyset$ 
else
    diasRecientes(s, linkConUltimoAcceso(s, c, linksCategoriaOHijos(s,c)))
fi
sumarAccesosRecientes(s, l, fs)  $\equiv$  if  $\emptyset?(fs)$  then
    0
else
    accesosRecientesDia(s, l, dameUno(f)) + sumarAccesosRecientes(s, l,
    sinUno(fs))
fi
 $\beta(b) \equiv$  if b then 1 else 0 fi

```

Fin TAD

2. TAD ARBOLDECATEGORIAS

TAD ARBOLDECATEGORIAS

géneros acat

exporta generadores, categorias, raíz, padre, id, altura, esta?, esSubCategoria, alturaCategoria, hijos

usa BOOL, NAT, CONJUNTO

observadores básicos

categorias : acat $ac \rightarrow$ conj(categoria)

raíz : acat $ac \rightarrow$ categoria

padre : acat $ac \times$ categoria $h \rightarrow$ categoria $\{esta?(h, ac) \wedge raíz(ac) \neq h\}$

id : acat $ac \times$ categoria $c \rightarrow$ nat $\{esta?(c, ac)\}$

generadores

nuevo : categoria $c \rightarrow$ acat $\{\neg vacia?(c)\}$

agregar : acat $ac \times$ categoria $c \times$ categoria $h \rightarrow$ acat $\{esta?(c, ac) \wedge \neg vacia?(h) \wedge \neg esta?(h, ac)\}$

otras operaciones

altura : acat $ac \rightarrow$ nat

esta? : categoria $c \times$ acat $ac \rightarrow$ bool

esSubCategoria : acat $ac \times$ categoria $c \times$ categoria $h \rightarrow$ bool $\{esta?(c, ac) \wedge esta?(h, ac)\}$

alturaCategoria : acat $ac \times$ categoria $c \rightarrow$ nat $\{esta?(c, ac)\}$

hijos : acat $ac \times$ categoria $c \rightarrow$ conj(categoria) $\{esta?(c, ac)\}$

axiomas $\forall a$: arbolDeCategorias

$\forall c$: categoria

$\forall ca$: conj(arbolDeCategoria)

$\forall cc$: conj(categoria)

$\text{categorias}(\text{nuevo}(c)) \equiv c$

$\text{categorias}(\text{agregar}(ac, c, h)) \equiv \text{Ag}(h, \text{categorias}(ac))$

$\text{raiz}(\text{nuevo}(c)) \equiv c$

$\text{raiz}(\text{agregar}(ac, c, h)) \equiv \text{raiz}(ac)$

$\text{padre}(\text{agregar}(ac, c, h), h') \equiv \text{if } h == h' \text{ then } c \text{ else } \text{padre}(ac, c, h') \text{ fi}$

$\text{id}(\text{nuevo}(c), c') \equiv 1$

$\text{id}(\text{agregar}(ac, c, h), h') \equiv \text{if } h == h' \text{ then } \# \text{categorias}(ac) + 1 \text{ else } \text{id}(ac, h2) \text{ fi}$

$\text{altura}(\text{nuevo}(c)) \equiv \text{alturaCategoria}(\text{nuevo}(c), c)$

$\text{altura}(\text{agregar}(ac, c, h)) \equiv \max(\text{altura}(ac), \text{alturaCategoria}(\text{agregar}(ac, c, h), h))$

$\text{alturaCategoria}(ac, c) \equiv \text{if } c == \text{raiz}(ac) \text{ then } 1 \text{ else } 1 + \text{alturaCategoria}(ac, \text{padre}(ac, c)) \text{ fi}$

$\text{esta?}(c, ac) \equiv c \in \text{categorias}(ac)$

$\text{esSubCategoria}(ac, c, h) \equiv c == h \vee L(h = \text{raiz}(ac) \wedge L \text{ esSubCategoria}(ac, c, \text{padre}(ac, h)))$

$\text{hijos}(\text{nuevo}(c1), c2) \equiv \emptyset$

$\text{hijos}(\text{agregar}(ac, c, h), c') \equiv \text{if } h == c' \text{ then } \emptyset \text{ else } (\text{if } c == c' \text{ then } h \text{ else } \emptyset \text{ fi}) \cup \text{hijos}(ac, c, c') \text{ fi}$

Fin TAD

3. Renombres

TAD CATEGORIA

es String

Fin TAD

TAD LINK

es String

Fin TAD

TAD FECHA

es Nat

Fin TAD