

Algoritmos y Estructuras de Datos II

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

Trabajo Práctico I

Grupo: 12

Integrante	LU	Correo electrónico
Pondal, Iván	078/14	ivan.pondal@gmail.com
Paz, Maximiliano León	251/14	m4xileon@gmail.com
Mena, Manuel	313/14	manuelmena1993@gmail.com
Demartino, Francisco	348/14	demartino.francisco@gmail.com

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

1. TAD DCNET

TAD DCNET

géneros dcnnet

igualdad observacional

$$(\forall d, d' : \text{dcnnet}) \left(d =_{\text{obs}} d' \iff \left(\begin{array}{l} (\text{topo}(d) =_{\text{obs}} \text{topo}(d')) \wedge \\ ((\forall p : \text{pc})(p \in \text{pcs}(\text{topo}(d)) \wedge p \in \text{pcs}(\text{topo}(d'))) \Rightarrow_{\text{L}} \\ (\text{buffer}(d, p) =_{\text{obs}} \text{buffer}(d', p) \wedge \\ \# \text{paquetesEnviados}(d, p) \\ \# \text{paquetesEnviados}(d', p)) \wedge \\ ((\forall p : \text{paquetes})((\exists c : \text{pc})(c \in \text{pcs}(\text{topo}(d') \wedge \\ c \in \text{pcs}(\text{topo}(d'))) \wedge_{\text{L}} (p \in \text{buffer}(d, c) \wedge \\ p \in \text{buffer}(d', c))) \Rightarrow_{\text{L}} \\ (\text{recorridoPaquete}(d, p) =_{\text{obs}} \text{recorridoPaquete}(d', p))) \end{array} \right) =_{\text{obs}} \right)$$

generadores

CrearRed : topo \longrightarrow dcnnet
 Seg : dcnnet \longrightarrow dcnnet
 PaquetePendiente : dcnnet dcn \times pc p1 \times pc p2 \times paquete \longrightarrow dcnnet
 $\{(p_1 \in \text{pcs}(\text{topo}(\text{dcn})) \wedge p_2 \in \text{pcs}(\text{topo}(\text{dcn}))) \wedge_{\text{L}} \text{conectadas?}(\text{topo}(\text{dcn}), p_1, p_2)\}$

observadores básicos

recorridoPaquete : dcnnet dcn \times paquete p \longrightarrow secu((ip, interface))
 $\{(\exists c : \text{pc})(c \in \text{pcs}(\text{topo}(\text{dcn})) \wedge_{\text{L}} (p \in \text{buffer}(\text{dcn}, c)))\}$
 buffer : dcnnet dcn \times pc p \longrightarrow conj(paquete) $\{p \in \text{pcs}(\text{topo}(\text{dcn}))\}$
 #paquetesEnviados : dcnnet dcn \times pc p \longrightarrow nat $\{p \in \text{pcs}(\text{topo}(\text{dcn}))\}$
 topo : dcnnet \longrightarrow topologia

otras operaciones

paqueteEnTransito? : dcnnet \times paquete \longrightarrow bool
 perteneceBuffers? : paquete \times buffers \longrightarrow bool
 compuQueMasEnvio : dcnnet \longrightarrow pc
 auxMaxPaquetes : dcnnet \times conj(pc) \longrightarrow pc
 pasoSeg : topo \times buffers \times buffers \longrightarrow buffers
 regresion : topo \times buffers \times secu(buffers) \longrightarrow buffers
 generarHistoria : dcnnet \times diccionario(pc \times \longrightarrow secu(buffers)
 conj(paquete))
 auxDefinir : buffers \times pc \times conj(paquete) \times \longrightarrow buffers
 conj(paquete)
 auxBorrar : buffers \times pc \times conj(paquete) \times \longrightarrow buffers
 conj(paquete)
 transacion : topo \times buffers \times conj(pc) \longrightarrow buffers
 envio : topo \times buffers \times ip \times conj(paquete) \longrightarrow buffers
 nuevosPaquetes : buffers \times buffers \longrightarrow buffers
 damePaquete : conj(paquete) \longrightarrow paquete
 pasarA : topologia \times pc \times pc \longrightarrow pc

axiomas $\forall p, p' : \text{paquete}, \forall c, c' : \text{pc}, \forall \text{dcn} : \text{dcnnet}, \forall t : \text{topologia}$

topo(crearRed(t)) \equiv t

<code>topo(seg(dcn))</code>	\equiv <code>topo(dcn)</code>
<code>topo(paquetePendiente(dcn,c,c',p))</code>	\equiv <code>topo(dcn)</code>
<code>#paquetesEnviados(crearRed(t),c)</code>	\equiv 0
<code>#paquetesEnviados(seg(dcn),c)</code>	\equiv <code>#paquetesEnviados(dcn)</code>
<code>#paquetesEnviados(paquetePendiente(dcn,o,d,p),c)</code>	\equiv if $c = o$ then <code>#paquetesEnviados(dcn, c) + 1</code> else <code>#paquetesEnviados(dcn, c)</code> fi
<code>buffer(dcn,c)</code>	\equiv <code>obtener(c,regresion(topo(dcn),vacio,generarHistoria(dcn,vacio))</code>
<code>compuQueMasEnvio(dcn)</code>	\equiv <code>auxMaxPaquetes(dcn,pcs(topo(dcn)))</code>
<code>auxMaxPaquetes(dcn,cs)</code>	\equiv if $\emptyset?(sinUno(cs))$ then <code>dameUno(cs)</code> else if <code>#paquetesEnviados(dcn, dameUno(cs))</code> < <code>#paquetesEnviados(dcn, auxMaxPaquetes(dcn, sinUno(cs)))</code> then <code>auxMaxPaquetes(dcn, sinUno(cs))</code> else <code>dameUno(cs)</code> fi fi
<code>paqueteEnTransito?(dcn,p)</code>	\equiv <code>perteneceBuffers?(p,regresion(topo(dcn),vacio,generarHistoria(dcn,vacio)))</code>
<code>perteneceBuffers?(p,bs)</code>	\equiv if $\emptyset?(claves(bs))$ then <code>false</code> else if $p \in obtener(dameUno(claves(bs)), bs)$ then <code>true</code> else <code>perteneceBuffers?(p, borrar(dameUno(claves(bs)), bs))</code> fi fi
<code>generarHistoria(crearRed(t),bs)</code>	\equiv <code>bs • <></code>
<code>generarHistoria(seg(dcn),bs)</code>	\equiv <code>bs • generarHistoria(dcn,vacio)</code>
<code>generarHistoria(paquetePendiente(dcn,o,d,p),bs)</code>	\equiv if $def?(c, bs)$ then <code>generarHistoria(dcn, definir(c, n obtener(o, bs), bs))</code> \cup else <code>generarHistoria(dcn, definir(c, n))</code> fi
<code>auxBorrar(bs,c,b,p)</code>	\equiv if $\emptyset?(p - \{b\})$ then <code>borrar(c, n)</code> else <code>borrar(c, bs) definir(c, p - \{b\}, bs)</code> fi
<code>regresion(t,bs,cbs)</code>	\equiv if $vacia?(fin(cbs))$ then <code>pasoSeg(bs, t, prim(cbs))</code> else <code>regresion(t, pasoSeg(bs, t, prim(cbs)), fin(cbs))</code> fi
<code>pasoSeg(t,bs,nbs)</code>	\equiv <code>nuevosPaquetes(transacion(t,bs,claves(bs)) ,nbs)</code>

transacion(t,bs,cp)

```
≡ if ∅?(cp) then
    bs
    else
        transacion(t,envio(t,bs,dameUno(cp)),
            sinUno(cp))
    fi
```

pasarA(t,o,d)

```
≡ prim(caminoMin(t,o,d))
```

envio(t,bs,ip,cp)

```
≡ if ∅?(damePaquete(cp)) then
    bs
    else
        if pasarA(t,ip,destino(damePaquete(cp))) =
            destino(damePaquete(cp)) then
            envio(t,quitarPaquete(bs,ip),ip,cp) –
            damePaquete(cp))
        else
            envio(t,quitarPaquete(pasarPaquete(bs,ip,damePaquete(
                ,ip,cp – damePaquete(b)))
            fi
    fi
```

nuevosPaquetes(bs,nbs)

```
≡ if ∅?(claves(nbs)) then
    bs
    else
        nuevosPaquetes(auxDefinir(bs,dameUno(claves(nbs),obt
            (dameUno(claves(nbs),nbs),obtener(dameUno
            (claves(nbs),bs))),sinUno(nbs))
    fi
```

TAD buffers es diccionario(pc,conj(paquete))

Fin TAD

2. TAD TOPOLOGÍA

Este TAD modela cómo se conectan las computadoras. Las IP son únicas entre compus de la topología. Las compus tienen interfaces numeradas con los naturales de manera consecutiva (todas funcionan perfecto y todo eso, el DC las cuida y mantiene como corresponde).

TAD TOPOLOGÍA

géneros topologia

generadores

NuevaTopo	:		\longrightarrow topologia
Compu	:	topologia \times nat $ip \times$ nat	\longrightarrow topologia $\{ \neg(ip \in compus(t)) \}$
Cable	:	topologia \times nat $ipA \times$ nat $ifA \times$ nat $ipB \times$ nat ifB	\longrightarrow topologia $\left\{ \begin{array}{l} (ipA \in compus(t) \wedge ipB \in compus(t)) \wedge_L \\ (ifA < \#interfaces(t, ipA)) \wedge \\ (ifB < \#interfaces(t, ipB)) \wedge \\ \neg(ifA \in interfacesOcupadasDe(t, ipA)) \wedge \\ \neg(ifB \in interfacesOcupadasDe(t, ipB)) \wedge \\ \neg(ipA \in vecinas(t, ipB)) \end{array} \right\}$

observadores básicos

compus	:	topologia	\longrightarrow conj(nat)
cablesEn	:	topologia $t \times$ nat ip	\longrightarrow conj(tupla(nat, nat)) $\{ ip \in compus(t) \}$
$\#interfaces$:	topologia $t \times$ nat ip	\longrightarrow nat $\{ ip \in compus(t) \}$

otras operaciones

vecinas	:	topologia $t \times$ nat ip	\longrightarrow conj(nat) $\{ ip \in compus(t) \}$
interfacesOcupadasDe	:	topologia $t \times$ nat ip	\longrightarrow conj(nat) $\{ ip \in compus(t) \}$
conectados?	:	topologia $t \times$ nat $ipA \times$ nat ipB	\longrightarrow bool $\{ ipA \in compus(t) \wedge ipB \in compus(t) \}$
darCaminoMasCorto	:	topologia $t \times$ nat $a \times$ nat b	\longrightarrow secu(nat) $\{ conectados?(t, a, b) \}$
darRutas	:	topologia \times nat \times nat \times conj(nat) \times secu(nat)	\longrightarrow conj(secu(tupla(nat, nat)))
darRutasVecinas	:	topologia \times conj(nat) \times nat \times conj(nat) \times secu(nat)	\longrightarrow conj(secu(tupla(nat, nat)))
longMenorSec	:	conj(secu(α))	\longrightarrow nat
secusDeLongK	:	conj(secu(α)) \times nat	\longrightarrow conj(secu(α))
π_1 Conj	:	conj(tupla(nat, nat))	\longrightarrow conj(nat))
π_2 Conj	:	conj(tupla(nat, nat))	\longrightarrow conj(nat))

axiomas $\forall t$: topologia, $\forall ipNueva, ip, ipA, ipB, ifA, ifB, cantInterfaces, k$: nat, $\forall conjDuplas$: conj(tupla(nat, nat)), $\forall cs, rec, vecinas$: conj(nat), $\forall secus$: conj(secu(α)), $\forall sc$: conj(secu(α)), $\forall ruta$: secu(nat)

compus(NuevaTopo)	$\equiv \emptyset$
compus(Compu($t, ipNueva, cantInterfaces$))	$\equiv Ag(ipNueva, compus(t))$
compus(Cable(t, ipA, ifA, ipB, ifB))	$\equiv compus(t)$
cablesEn(NuevaTopo, ip)	$\equiv \emptyset$
cablesEn(Compu($t, ipNueva, cantInterfaces$), ip)	$\equiv cablesEn(t, ip)$

$\text{cablesEn}(\text{Cable}(t, ipA, ifA, ipB, ifB), ip)$	\equiv	if $ip = ipA$ then $\text{Ag}(\langle ifA, ipB \rangle, \emptyset)$ else \emptyset fi \cup if $ip = ipB$ then $\text{Ag}(\langle ifB, ipA \rangle, \emptyset)$ else \emptyset fi \cup $\text{cablesEn}(t, ip)$
$\#interfaces(\text{NuevaTopo}, ip)$	\equiv	0
$\#interfaces(\text{Compu}(t, ipNueva, cantInterfaces), ip)$	\equiv	if $ip = ipNueva$ then $cantInterfaces$ else $\#interfaces(t)$ fi
$\#interfaces(\text{Cable}(t, ipA, ifA, ipB, ifB), ip)$	\equiv	$\#interfaces(t)$
$\text{interfacesOcupadasDe}(t, ip)$	\equiv	$\pi_1 \text{Conj}(\text{cablesEn}(t, ip))$
$\text{vecinas}(t, ip)$	\equiv	$\pi_2 \text{Conj}(\text{cablesEn}(t, ip))$
$\text{conectados?}(t, ipA, ipB)$	\equiv	$\neg \emptyset?(\text{darRutas}(t, ipA, ipB, \emptyset, \langle \rangle))$
$\text{darRutas}(t, ipA, ipB, rec, ruta)$	\equiv	if $ipB \in \text{vecinas}(t, ipA)$ then $\text{Ag}(ruta \ \& \ (ipA \bullet ipB \bullet \langle \rangle), \emptyset)$ else if $\emptyset?(\text{vecinas}(t, ipA) - rec)$ then \emptyset else $\text{darRutas}(t, \text{dameUno}(\text{vecinas}(t, ipA) - rec), ipB,$ $\text{Ag}(ipA, rec), ruta \circ ipA) \cup$ $\text{darRutasVecinas}(t, \text{sinUno}(\text{vecinas}(t, ipA) - rec), ipB,$ $\text{Ag}(ipA, rec), ruta \circ ipA)$ fi fi
$\text{darRutasVecinas}(t, vecinas, ipB, rec, ruta)$	\equiv	if $\emptyset?(vecinas)$ then \emptyset else $\text{darRutas}(t, \text{dameUno}(vecinas), ipB, rec, ruta) \cup$ $\text{darRutasVecinas}(t, \text{sinUno}(vecinas), ipB, rec, ruta)$ fi
$\text{darCaminoMasCorto}(t, ipA, ipB)$	\equiv	$\text{dameUno}(\text{secusDeLongK}(\text{darRutas}(t, ipA, ipB, \emptyset, \langle \rangle),$ $\text{longMenorSec}(\text{darRutas}(t, ipA, ipB, \emptyset, \langle \rangle)))$
$\text{secusDeLongK}(secus, k)$	\equiv	if $\emptyset?(secus)$ then \emptyset else if $\text{long}(\text{dameUno}(secus)) = k$ then $\text{dameUno}(secus) \cup \text{secusDeLongK}(\text{sinUno}(secus), k)$ else $\text{secusDeLongK}(\text{sinUno}(secus), k)$ fi fi
$\text{longMenorSec}(secus)$	\equiv	if $\emptyset?(secus)$ then 0 else $\min(\text{long}(\text{dameUno}(secus)),$ $\text{longMenorSec}(\text{sinUno}(secus)))$ fi
$\pi_1 \text{Conj}(conjDuplas)$	\equiv	if $\emptyset?(conjDuplas)$ then \emptyset else $\text{Ag}(\pi_1(\text{dameUno}(conjDuplas)),$ $\pi_1 \text{Conj}(\text{sinUno}(conjDuplas)))$ fi

$\pi_2 \text{Conj}(\text{conjDuplas})$ \equiv **if** $\emptyset?(conjDuplas)$ **then**
 \emptyset
else
 $\text{Ag}(\pi_2(\text{dameUno}(conjDuplas)),$
 $\pi_2 \text{Conj}(\text{sinUno}(conjDuplas)))$
fi

Fin TAD