

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 `dcnet`

igualdad observacional

$$(\forall d, d' : \text{dcnet}) \quad d =_{\text{obs}} d' \iff \left(\begin{array}{l} (topo(d) =_{\text{obs}} topo(d')) \wedge \\ ((\forall p : pc)(p \in compus(topo(d)) \wedge p \in compus(topo(d'))) \Rightarrow_L \\ (buffer(d, p) =_{\text{obs}} buffer(d', p) \wedge \\ \#paquetesEnviados(d, p) \\ \#paquetesEnviados(d', p)) \wedge \\ ((\forall p : paquetes)((\exists c : pc)(c \in compus(topo(d')) \wedge \\ c \in compus(topo(d')) \wedge_L (p \in buffer(d, c) \wedge \\ p \in buffer(d', c))) \Rightarrow_L \\ (recorridoPaquete(d, p) =_{\text{obs}} recorridoPaquete(d', p))) \end{array} \right) =_{\text{obs}}$$

generadores

`CrearRed` : `topo` \longrightarrow `dcnet`
`Seg` : `dcnet` \longrightarrow `dcnet`
`CrearPaquete` : `dcnet dcn` \times `pc p1` \times `pc p2` \times `paquete` \longrightarrow `dcnet`
 $\{(p_1 \in compus(topo(dcn)) \wedge p_2 \in compus(topo(dcn))) \wedge_L conectadas?(topo(dcn), p_1, p_2)\}$

observadores básicos

`topo` : `dcnet` \longrightarrow `topologia`
`#paquetesEnviados` : `dcnet dcn` \times `pc p` \longrightarrow `nat` $\{p \in compus(topo(dcn))\}$
`buffer` : `dcnet dcn` \times `pc p` \longrightarrow `conj(paquete)`
 $\{p \in compus(topo(dcn))\}$

otras operaciones

`recorridoPaquete` : `dcnet dcn` \times `nat id` \longrightarrow `secu(tupla(nat, nat, nat, nat))`
 $\{(paqueteEnTransito?(dcn, id)\}$
`cortarRecHasta` : `sec(tupla(nat \times nat \times nat \times nat)) \times nat \longrightarrow sec(tupla(nat, nat, nat, nat))
buscarPaquete : dcnet dcn \times conj(nat) pcs \times nat id \longrightarrow nat
 $\{pcs = compus(topo(dcn)) \wedge (\exists ip : nat)(ip \in pcs \wedge id \in buffer(dcn, ip))\}$
 π_1 Conj : conj(tupla(nat, nat, nat, nat)) \longrightarrow conj(nat)
paqueteEnTransito? : dcnet \times nat \longrightarrow bool
existePqEnBuffers? : dcnet dcn \times conj(nat) pcs \times nat id \longrightarrow bool $\{pcs = compus(topo(dcn))\}$
perteneceBuffers? : paquete \times buffers \longrightarrow bool
darPaqueteEnviado : conj(paquete) \longrightarrow paquete
darPrioridad : dcnet dcn \times nat id \longrightarrow nat $\{id \in paquetesEnLaRed(dcn)\}$
buscarPrioridad : nat \times conj(paquetes) \longrightarrow nat
maxPrioridad : dcnet \times conj(pc) \longrightarrow nat
PaquetesConPrioridadKdcnet \times conj(pc) \times nat \longrightarrow paquete
paquetesEnLaRed : dcnet \longrightarrow conj(paquete)
buscarPaquetesEnLaRed : dcnet \times conj(pc) \longrightarrow conj(paquete)
compuQueMasEnvio : dcnet \longrightarrow pc
laQueMasEnvio : dcnet \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 conj(paquete))	\longrightarrow secu(buffers)
auxDefinir	: buffers \times pc \times conj(paquete) \times conj(paquete)	\longrightarrow buffers
auxBorrar	: buffers \times pc \times conj(paquete) \times conj(paquete)	\longrightarrow buffers
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
pasarA	: topologia \times pc \times pc	\longrightarrow pc

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

topo(crearRed(t))	$\equiv t$
topo(seg(dcn))	$\equiv \text{topo(dcn)}$
topo(CrearPaquete(dcn, c, c', p))	$\equiv \text{topo(dcn)}$
#paquetesEnviados(crearRed(t), c)	$\equiv 0$
#paquetesEnviados(seg(dcn), c)	$\equiv \#paquetesEnviados(dcn)$
#paquetesEnviados(CrearPaquete(dcn, o, d, p), c)	$\equiv \text{if } c = o \text{ then } \#paquetesEnviados(dcn, c) + 1$ $\text{else } \#paquetesEnviados(dcn, c)$ fi
buffer(dcn, c)	$\equiv \text{obtener}(c, \text{regresion}(\text{topo(dcn)}, \text{vacio}, \text{generarHistoria(dcn, vacio)}))$
recorridoPaquete(dcn, p)	$\equiv \text{cortarRecHasta}(\text{darCaminoMasCorto}(\text{topo(dcn)}, \text{origen(p)}, \text{destino(p)}), \text{buscar}(\text{compus}(\text{topo(dcn)}), p))$
cortarRecHasta(s, ip)	$\equiv \text{if } \text{vacía?}(s) \vee_L \text{ip} = \text{ipOrigen}(\text{prim}(s)) \text{ then } \langle \rangle$ $\text{else } \text{prim}(s) \bullet \text{cortarRecHasta}(\text{fin}(s), \text{ip})$ fi
buscarPaquete(dcn, compus, id)	$\equiv \text{if } \text{id} \in \pi_1 \text{Conj}(\text{buffer(dcn, dameUno(compus))}) \text{ then } \text{dameUno(compus)}$ $\text{else } \text{buscarPaquete}(\text{sinUno(compus)}, \text{id})$ fi
$\pi_1 \text{Conj}(\text{conjTuplas})$	$\equiv \text{if } \emptyset?(\text{conjTuplas}) \text{ then } \emptyset$ $\text{else } \text{Ag}(\pi_1(\text{dameUno}(\text{conjTuplas})), \pi_1 \text{Conj}(\text{sinUno}(\text{conjTuplas})))$ fi
paqueteEnTransito?(dcn, id)	$\equiv \text{existePqEnBuffers?}(dcn, \text{compus}(\text{topo(dcn)}), \text{id})$
existePqEnBuffers?(dcn, pcs, id)	$\equiv \text{if } \emptyset?(pcs) \text{ then } \text{false}$ $\text{else } \text{if } \text{id} \in \pi_1 \text{Conj}(\text{buffer(dcn, dameUno(pcs))}) \text{ then } \text{true}$ $\text{else } \text{existePqEnBuffers?}(dcn, \text{sinUno(pcs)}, \text{id})$ fi

buscarpaquetesEnLaRed(dcn,cc)	≡ if $\emptyset?(cc)$ then \emptyset else buffer(dcn,dameUno(cc)) buscarpaquetesEnLaRed(dcn,sinUno(cc)) fi	U
paquetesEnLaRed(dcn)	≡ buscarpaquetesEnLaRed(dcn,compus(topo(dcn)))	
buscarPrioridad(id,cp)	≡ if $i = \Pi_1(dameUno(cp))$ then $\Pi_4(dameUno(cp))$ else darPrioridad(id,sinUno(cp)) fi	
darPrioridad(dcn,id)	≡ buscarPrioridad(id,compus(dcn))	
darPaqueteEnviado(dcn,cp)	≡ dameUno(PaquetesConPrioridadK(dcn,cp,maxPrioridad(dcn,cp)))	
maxPrioridad(dcn,cp)	≡ if $\emptyset?(sinUno(cp))$ then darPrioridad(dcn,dameUno(cp)) else max(darPrioridad(dcn,dameUno(cp)), maxPrioridad(dcn,sinUno(cp))) fi	
PaquetesConPrioridadK(dcn,cp,k)	≡ if $\emptyset?(cp)$ then \emptyset else if darPrioridad(dcn,dameUno(cp)) = k then Ag(dameUno(cp),PaquetesConPrioridadK(dcn,sinUno(cp),k)) else PaquetesConPrioridadK(dcn,sinUno(cp),k) fi fi	
compuQueMasEnvio(dcn)	≡ laQueMasEnvio(dcn,compus(topo(dcn)))	
laQueMasEnvio(dcn,cs)	≡ if $\emptyset?(sinUno(cs))$ then dameUno(cs) else if #paquetesEnviados(dcn,dameUno(cs)) < #paquetesEnviados(dcn,laQueMasEnvio(dcn,sinUno(cs))) then laQueMasEnvio(dcn,sinUno(cs)) else dameUno(cs) fi fi	<
perteneceBuffers?(p,bs)	≡ if $\emptyset?(claves(bs))$ then false else if $p \in obtener(dameUno(claves(bs)),bs)$ then true else perteneceBuffers?(p,borrar(dameUno(claves(bs)),bs)) fi fi	
generarHistoria(crearRed(t),bs)	≡ bs • <>	
generarHistoria(seg(dcn),bs)	≡ bs • generarHistoria(dcn,vaco)	

```

generarHistoria(CrearPaquete(dcn,o,d,p),bs)  ≡ if def?(c,bs) then
                                             generarHistoria(dcn,definir(c,n
                                             obtener(o,bs),bs))
                                             else
                                             generarHistoria(dcn,definir(c,n)
                                             fi
auxBorrar(bs,c,b,p)  ≡ if ∅?(p - {b}) then
                     borrar(c,n)
                     else
                     borrar(c,bs) definir(c,p - {b},bs)
                     fi
regresion(t,bs,cbs)  ≡ if vacia?(fin(cbs)) then
                     pasoSeg(bs,t,prim(cbs))
                     else
                     regresion(t,pasoSeg(bs,t,prim(cbs)),fin(cbs))
                     fi
pasoSeg(t,bs,nbs)    ≡ nuevosPaquetes(transacion(t,bs,claves(bs)) ,nbs)
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 ∅?(darPaqueteEnviado(cp)) then
                     bs
                     else
                     if pasarA(t,ip,destino(darPaqueteEnviado(cp))) =
                     destino(darPaqueteEnviado(cp)) then
                         envio(t,quitarPaquete(bs,ip),ip,cp
                         darPaqueteEnviado(cp))
                     else
                         envio(t,quitarPaquete(pasarPaquete(bs,ip,darPaqueteEnviado(
                         ,ip,cp - darPaqueteEnviado(b)))
                     fi
                     fi
nuevosPaquetes(bs,nbs) ≡ if ∅?(claves(nbs)) then
                     bs
                     else
                     nuevosPaquetes(auxDefinir(bs,dameUno(claves(nbs)),obtener
                     (dameUno(claves(nbs)),nbs),obtener(dameUno
                     (claves(nbs),bs))),sinUno(nbs))
                     fi

```

TAD paquete ES tupla(nat id, nat ipOrigen, nat ipDestino, nat prioridad)

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

otras operaciones

vecinas	:	\longrightarrow	conj(nat) $\{ ip \in compus(t) \}$
interfacesOcupadasDe	:	\longrightarrow	conj(nat) $\{ ip \in compus(t) \}$
conectados?	:	\longrightarrow	bool $\{ ipA \in compus(t) \wedge ipB \in compus(t) \}$
darInterfazConectada	:	\longrightarrow	nat $\{ ipB \in \pi_2 Conj(cablesA) \}$
darSegmento	:	\longrightarrow	segmento $\{ ipA \in compus(t) \wedge_L ipB \in vecinas(t, ipA) \}$
darCaminoMasCorto	:	\longrightarrow	secu(segmento) $\{ ipA \in compus(t) \wedge ipB \in compus(t) \wedge_L conectados?(t, ipA, ipB) \}$
darRutas	:	\longrightarrow	conj(secu(segmento)) $\{ ipA \in compus(t) \wedge ipB \in compus(t) \}$
darRutasVecinas	:	\longrightarrow	conj(secu(segmento)) $\{ ip \in compus(t) \}$
longMenorSec	:	\longrightarrow	nat
secusDeLongK	:	\longrightarrow	conj(secu(α))
π_1 Conj	:	\longrightarrow	conj(nat))

$\pi_2 \text{Conj}$	$: \text{conj}(\text{tupla}(\text{nat}, \text{nat})) \longrightarrow \text{conj}(\text{nat})$
axiomas	$\forall t: \text{topologia}, \forall ipNueva, ip, ipA, ipB, ifA, ifB, cantIfaces, k: \text{nat}, \forall conjDuplas: \text{conj}(\text{tupla}(\text{nat}, \text{nat})), \forall conjCablesIpA: \text{conj}(\text{tupla}(\text{nat}, \text{nat})), \forall cs, rec, vecinas: \text{conj}(\text{nat}), \forall secus: \text{conj}(\text{secu}(\alpha)), \forall sc: \text{conj}(\text{secu}(\alpha)), \forall ruta: \text{secu}(\text{segmento})$
$\text{compus}(\text{NuevaTopo})$	$\equiv \emptyset$
$\text{compus}(\text{Compu}(t, ipNueva, cantIfaces))$	$\equiv \text{Ag}(ipNueva, \text{compus}(t))$
$\text{compus}(\text{Cable}(t, ipA, ifA, ipB, ifB))$	$\equiv \text{compus}(t)$
$\text{cablesEn}(\text{NuevaTopo}, ip)$	$\equiv \emptyset$
$\text{cablesEn}(\text{Compu}(t, ipNueva, cantIfaces), ip)$	$\equiv \text{cablesEn}(t, ip)$
$\text{cablesEn}(\text{Cable}(t, ipA, ifA, ipB, ifB), ip)$	$\equiv \text{if } ip = ipA \text{ then } \text{Ag}(\langle ifA, ipB \rangle, \emptyset) \text{ else } \emptyset \text{ fi} \cup \text{if } ip = ipB \text{ then } \text{Ag}(\langle ifB, ipA \rangle, \emptyset) \text{ else } \emptyset \text{ fi} \cup \text{cablesEn}(t, ip)$
$\#interfaces(\text{NuevaTopo}, ip)$	$\equiv 0$
$\#interfaces(\text{Compu}(t, ipNueva, cantIfaces), ip)$	$\equiv \text{if } ip = ipNueva \text{ then } cantIfaces \text{ else } \#interfaces(t) \text{ fi}$
$\#interfaces(\text{Cable}(t, ipA, ifA, ipB, ifB), ip)$	$\equiv \#interfaces(t)$
$\text{interfacesOcupadasDe}(t, ip)$	$\equiv \pi_1 \text{Conj}(\text{cablesEn}(t, ip))$
$\text{vecinas}(t, ip)$	$\equiv \pi_2 \text{Conj}(\text{cablesEn}(t, ip))$
$\text{conectados?}(t, ipA, ipB)$	$\equiv \neg \emptyset?(\text{darRutas}(t, ipA, ipB, \emptyset, <>))$
$\text{darInterfazConectada}(conjCablesIpA, ipB)$	$\equiv \text{if } ipB = \pi_2(\text{dameUno}(conjCablesIpA)) \text{ then } \pi_1(\text{dameUno}(conjCablesIpA)) \text{ else } \text{darInterfazConectada}(\text{sinUno}(conjCablesIpA), ipB) \text{ fi}$
$\text{darSegmento}(t, ipA, ipB)$	$\equiv \langle ipA, \text{darInterfazConectada}(\text{cablesEn}(t, ipA), ipB), ipB, \text{darInterfazConectada}(\text{cablesEn}(t, ipB), ipA) \rangle$
$\text{darCaminoMasCorto}(t, ipA, ipB)$	$\equiv \text{dameUno}(\text{secusDeLongK}(\text{darRutas}(t, ipA, ipB, \emptyset, <>), \text{longMenorSec}(\text{darRutas}(t, ipA, ipB, \emptyset, <>)))$
$\text{darRutas}(t, ipA, ipB, rec, ruta)$	$\equiv \text{if } ipB \in \text{vecinas}(t, ipA) \text{ then } \text{Ag}(ruta \circ \text{darSegmento}(t, ipA, ipB), \emptyset) \text{ else } \text{if } \emptyset?(\text{vecinas}(t, ipA) - rec) \text{ then } \emptyset \text{ else } \text{darRutas}(t, \text{dameUno}(\text{vecinas}(t, ipA) - rec), ipB, \text{Ag}(ipA, rec), ruta \circ \text{darSegmento}(t, ipA, \text{dameUno}(\text{vecinas}(t, ipA) - rec))) \cup \text{darRutasVecinas}(t, \text{sinUno}(\text{vecinas}(t, ipA) - rec), ipB, \text{Ag}(ipA, rec), ruta \circ \text{darSegmento}(t, ipA, \text{dameUno}(\text{vecinas}(t, ipA) - rec))) \text{ fi}$
$\text{darRutasVecinas}(t, vecinas, ipB, rec, ruta)$	$\equiv \text{if } \emptyset?(vecinas) \text{ then } \emptyset \text{ else } \text{darRutas}(t, \text{dameUno}(vecinas), ipB, rec, ruta) \cup \text{darRutasVecinas}(t, \text{sinUno}(vecinas), ipB, rec, ruta) \text{ fi}$

$\text{darCaminoMasCorto}(t, ipA, ipB)$ $\text{secusDeLongK}(\text{secus}, k)$ $\text{longMenorSec}(\text{secus})$ $\pi_1 \text{Conj}(\text{conjDuplas})$ $\pi_2 \text{Conj}(\text{conjDuplas})$	$\equiv \text{dameUno}(\text{secusDeLongK}(\text{darRutas}(t, ipA, ipB, \emptyset, <>),$ $\text{longMenorSec}(\text{darRutas}(t, ipA, ipB, \emptyset, <>)))$ $\equiv \text{if } \emptyset?(\text{secus}) \text{ then}$ $\quad \emptyset$ else $\quad \text{if } \text{long}(\text{dameUno}(\text{secus})) = k \text{ then}$ $\quad \quad \text{dameUno}(\text{secus}) \cup \text{secusDeLongK}(\text{sinUno}(\text{secus}), k)$ $\quad \text{else}$ $\quad \quad \text{secusDeLongK}(\text{sinUno}(\text{secus}), k)$ $\quad \text{fi}$ fi $\equiv \text{if } \emptyset?(\text{secus}) \text{ then}$ $\quad 0$ else $\quad \min(\text{long}(\text{dameUno}(\text{secus})),$ $\quad \quad \text{longMenorSec}(\text{sinUno}(\text{secus})))$ fi $\equiv \text{if } \emptyset?(\text{conjDuplas}) \text{ then}$ $\quad \emptyset$ else $\quad \text{Ag}(\pi_1(\text{dameUno}(\text{conjDuplas})),$ $\quad \quad \pi_1 \text{Conj}(\text{sinUno}(\text{conjDuplas})))$ fi $\equiv \text{if } \emptyset?(\text{conjDuplas}) \text{ then}$ $\quad \emptyset$ else $\quad \text{Ag}(\pi_2(\text{dameUno}(\text{conjDuplas})),$ $\quad \quad \pi_2 \text{Conj}(\text{sinUno}(\text{conjDuplas})))$ fi
--	--

TAD segmento es tupla(nat, nat, nat ,nat)

Fin TAD