

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

Reservado para la cátedra

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		

TAD DCNET**géneros** *dcnet***igualdad observacional**

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

generadores

<i>crearRed</i>	: <i>topo</i>	→ <i>dcnet</i>
<i>seg</i>	: <i>dcnet</i>	→ <i>dcnet</i>
<i>mandarPaquete</i>	: <i>dcnet dcn</i> × <i>pc p1</i> × <i>pc p2</i> × <i>paquete</i>	→ <i>dcnet</i> $\left\{ (p_1 \in \text{pcs}(\text{topo}(\text{dcn})) \wedge p_2 \in \text{pcs}(\text{topo}(\text{dcn}))) \wedge_{\text{L}} \right\}$ $\left\{ \text{conectadas?}(\text{topo}(\text{dcn}), p_1, p_2) \right\}$

observadores básicos

<i>recorridoPaquete</i>	: <i>dcnet dcn</i> × <i>paquete p</i>	→ <i>secu((ip, interface))</i> $\{(\exists c : \text{pc})(c \in \text{pcs}(\text{topo}(\text{dcn})) \wedge_{\text{L}} (p \in \text{buffer}(\text{dcn}, c)))\}$
<i>buffer</i>	: <i>dcnet dcn</i> × <i>pc p</i>	→ <i>conj(paquete)</i> $\{p \in \text{pcs}(\text{topo}(\text{dcn}))\}$
<i>paquetesMandados</i>	: <i>dcnet dcn</i> × <i>pc p</i>	→ <i>nat</i> $\{p \in \text{pcs}(\text{topo}(\text{dcn}))\}$
<i>topo</i>	: <i>dcnet</i>	→ <i>topologia</i>

otras operaciones

<i>paqueteEnTransito?</i>	: <i>dcnet</i> × <i>paquete</i>	→ <i>bool</i>
<i>maxPaquetesMandados</i>	: <i>dcnet</i>	→ <i>pc</i>

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

<i>topo(crearRed(t))</i>	≡ <i>t</i>
<i>topo(seg(dcn))</i>	≡ <i>topo(dcn)</i>
<i>topo(mandarPaquete(dcn, c, c', p))</i>	≡ <i>topo(dcn)</i>
<i>paquetesMandados(crearRed(t))</i>	≡ 0
<i>paquetesMandados(seg(dcn))</i>	≡ <i>paquetesMandados(dcn)</i>
<i>paquetesMandados(mandarPaquete(dcn, o, d, p), c)</i>	≡ if <i>c = o</i> then <i>paquetesMandados(dcn, c) + 1</i> else <i>paquetesMandados(dcn, c)</i> fi

Fin TAD

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

g neros topologia

generadores

$$\begin{array}{lll}
\text{NuevaTopo} & : & \longrightarrow \text{topologia} \\
\text{Compu} & : \text{topologia} \times \text{nat} \times \text{nat} & \longrightarrow \text{topologia} \\
\text{Cable} & : \text{topologia} \times \text{nat} \times \text{nat} \times \text{nat} \times \text{nat} & \longrightarrow \text{topologia}
\end{array}$$

observadores básicos

compus	: topologia	\longrightarrow	$\text{conj}(\text{nat})$	
vecinas	: topologia $t \times \text{nat}$	ip	\longrightarrow	$\text{conj}(\text{nat})$ $\{ip \in \text{compus}(t)\}$
cables	: topologia $t \times \text{nat}$	ip	\longrightarrow	$\text{conj}(\text{tupla}(\text{nat}, \text{nat}))$ $\{ip \in \text{compus}(t)\}$

otras operaciones

seAlcanzan?	: topologia $t \times \text{nat } ipA \times \text{nat } ipB$	$\longrightarrow \text{bool}$	$\{ipA \in \text{compus}(t) \wedge ipB \in \text{compus}(t)\}$
todasLasQueAlcanza	: topologia $t \times \text{nat } ip$	$\longrightarrow \text{conj}(\text{nat})$	$\{ip \in \text{compus}(t)\}$
expandirFull	: topologia $t \times \text{conj}(\text{nat}) \text{ cs}$	$\longrightarrow \text{conj}(\text{nat})$	$\{\text{cs} \subseteq \text{compus}(t)\}$
exp1	: topologia $t \times \text{conj}(\text{nat}) \text{ cs}$	$\longrightarrow \text{conj}(\text{nat})$	$\{\text{cs} \subseteq \text{compus}(t)\}$

axioms $\forall t: \text{topologia}, \forall ip, ipBus, ipA, ipB, ifA, ifB, numIfaces: \text{nat}$

compus(NuevaTopo)	$\equiv \emptyset$
compus(Compu($t, ip, numIfaces$))	$\equiv \text{Ag}(ip, \text{compus}(t))$
compus(Cable(t, ipA, ifA, ipB, ifB))	$\equiv \text{compus}(t)$
vecinas(NuevaTopo, $ipBus$)	$\equiv \emptyset$
vecinas(Compu($t, ip, numIfaces$), $ipBus$)	$\equiv \text{vecinas}(t, ipBus)$
vecinas(Cable(t, ipA, ifA, ipB, ifB), $ipBus$)	$\equiv \text{if } ipBus = ipA \text{ then } \text{Ag}(ipB, \emptyset) \text{ else } \emptyset \text{ fi } \cup$ $\text{if } ipBus = ipB \text{ then } \text{Ag}(ipA, \emptyset) \text{ else } \emptyset \text{ fi } \cup$ $\text{vecinas}(t, ipBus)$
cables(NuevaTopo, $ipBus$)	$\equiv \emptyset$
cables(Compu($t, ip, numIfaces$), $ipBus$)	$\equiv \text{cables}(t, ipBus)$
cables(Cable(t, ipA, ifA, ipB, ifB), $ipBus$)	$\equiv \text{if } ipBus = ipA \text{ then } \text{Ag}(\langle ifA, ipB \rangle, \emptyset) \text{ else } \emptyset \text{ fi } \cup$ $\text{if } ipBus = ipB \text{ then } \text{Ag}(\langle ifB, ipA \rangle, \emptyset) \text{ else } \emptyset \text{ fi } \cup$ $\text{cables}(t, ipBus)$
seAlcanzan?(t, ipA, ipB)	$\equiv ipA \in \text{todasLasQueAlcanza}(t, ipB)$
todasLasQueAlcanza(t, ip)	$\equiv \text{expandirFull}(t, \text{Ag}(ip, \emptyset))$
expandirFull(t, cs)	$\equiv \text{if } \text{exp1}(t, cs) \subseteq cs \text{ then}$ $\quad cs$ else $\quad \text{expandirFull}(t, \text{exp1}(t, cs))$ fi

```
exp1( $t$ ,  $cs$ )  $\equiv$  if  $\emptyset?(cs)$  then  
     $\emptyset$   
else  
    Ag(dameUno( $cs$ ), vecinas( $t$ , dameUno( $cs$ )))  $\cup$  exp1( $t$ ,  
    sinUno( $cs$ ))  
fi
```

Fin TAD