

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

<code>crearRed</code>	: <code>topo</code>	\longrightarrow <code>dcnet</code>
<code>seg</code>	: <code>dcnet</code>	\longrightarrow <code>dcnet</code>
<code>paquetePendiente</code>	: <code>dcnet dcn</code> \times <code>pc p1</code> \times <code>pc p2</code> \times <code>paquete</code>	\longrightarrow <code>dcnet</code> $\{(p_1 \in \text{pcs}(\text{topo}(\text{dcn})) \wedge p_2 \in \text{pcs}(\text{topo}(\text{dcn}))) \wedge_{\text{L}} \text{conectadas?}(\text{topo}(\text{dcn}), p_1, p_2)\}$

observadores básicos

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

otras operaciones

<code>paqueteEnTransito?</code>	: <code>dcnet</code> \times <code>paquete</code>	\longrightarrow <code>bool</code>
<code>maxPaquetesMandados</code>	: <code>dcnet</code>	\longrightarrow <code>pc</code>
<code>pasoSeg</code>	: <code>topo</code> \times <code>buffers</code> \times <code>buffers</code>	\longrightarrow <code>buffers</code>
<code>regresion</code>	: <code>topo</code> \times <code>buffers</code> \times <code>secu(buffers)</code>	\longrightarrow <code>buffers</code>
<code>cronoPaquetes</code>	: <code>dcnet</code> \times <code>diccionario(pc</code> \times \longrightarrow <code>secu(buffers)</code> <code>conj(paquete))</code>	
<code>auxDefinir</code>	: <code>buffers</code> \times <code>pc</code> \times <code>conj(paquete)</code> \times \longrightarrow <code>buffers</code> <code>conj(paquete)</code>	
<code>auxBorrar</code>	: <code>buffers</code> \times <code>pc</code> \times <code>conj(paquete)</code> \times \longrightarrow <code>buffers</code> <code>conj(paquete)</code>	
<code>envioYReciboPaquetes</code>	: <code>topo</code> \times <code>buffers</code> \times <code>conj(pc)</code>	\longrightarrow <code>buffers</code>
<code>envio</code>	: <code>topo</code> \times <code>buffers</code> \times <code>buffer</code>	\longrightarrow <code>buffers</code>
<code>nuevosPaquetes</code>	: <code>buffers</code> \times <code>buffers</code>	\longrightarrow <code>buffers</code>
<code>pasarA</code>	: <code>topologia</code> \times <code>pc</code> \times <code>pc</code>	\longrightarrow <code>pc</code>

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

<code>topo(crearRed(t))</code>	\equiv <code>t</code>
<code>topo(seg(dcn))</code>	\equiv <code>topo(dcn)</code>
<code>topo(paquetePendiente(dcn, c, c', p))</code>	\equiv <code>topo(dcn)</code>
<code>paquetesMandados(crearRed(t))</code>	\equiv <code>0</code>
<code>paquetesMandados(seg(dcn))</code>	\equiv <code>paquetesMandados(dcn)</code>

```

paquetesMandados(paquetePendiente(dcn,o,d,p),c) ≡ if c = o then
    paquetesMandados(dcn,c) + 1
else
    paquetesMandados(dcn,c)
fi

dcNetBuffer(dcn,c) ≡ obtener(c,regresion(topo(dcn),vacio,cronoPaquetes(dcn,vacio)))

cronoPaquetes(crearRed(t),bs) ≡ <>

cronoPaquetes(seg(dcn),bs) ≡ bs • cronoPaquetes(dcn,∅)

cronoPaquetes(paquetePendiente(dcn,o,d,p),bs) ≡ auxDefinir(dp,o,Ag(p,∅),obtener(o,bs))
    cronoPaquetes(dcn,bs)

auxDefinir(bs,c,n,v) ≡ if def?(c,bs) then
    borrar(c,bs) definir(c,n ∪ v,bs)
else
    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) ≡ envioYReciboPaquetes(t,bs,claves(bs)) nuevosPaquetes(bs,nbs)

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

pasarA(t,o,d) ≡ prim(caminoMin(t,o,d))

envio(t,bs,b) ≡ auxDefinir(bs,pasarA(t,Π1(b),dest(Π2(b))),
    Ag(damePaquete(b),∅),obtener
    (pasarA(t,Π1(b),dest(Π2(b))),bs)
    auxBorrar(bs,Π1(b),damePaquete(b),
    obtener(bs,Π1(b))

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

```

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

TAD buffer es tupla(pc,conj(paquete))

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

$$\text{NuevaTopo} \quad : \quad \longrightarrow \text{topologia}$$
$$\text{Compu} \quad : \text{topologia} \times \text{nat} \times \text{nat} \quad \longrightarrow \text{topologia}$$
$$\text{Cable} : \text{topologia} \times \text{nat} \times \text{nat} \times \text{nat} \times \text{nat} \longrightarrow \text{topologia}$$

observadores básicos

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

otras operaciones

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

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

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

```
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