

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

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

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> $\{(p_1 \in pcs(topo(dcn)) \wedge p_2 \in pcs(topo(dcn))) \wedge_L conectadas?(topo(dcn), p_1, p_2)\}$

observadores básicos

<code>recorridoPaquete</code>	: <code>dcnet dcn</code> \times <code>paquete p</code>	\longrightarrow <code>secu((ip, interface))</code> $\{(\exists c : pc)(c \in pcs(topo(dcn)) \wedge_L (p \in buffer(dcn, c)))\}$
<code>dcNetBuffer</code>	: <code>dcnet dcn</code> \times <code>pc p</code>	\longrightarrow <code>conj(paquete)</code> $\{p \in pcs(topo(dcn))\}$
<code>paquetesMandados</code>	: <code>dcnet dcn</code> \times <code>pc p</code>	\longrightarrow <code>nat</code> $\{p \in pcs(topo(dcn))\}$
<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>perteneceBuffers?</code>	: <code>paquete</code> \times <code>buffers</code>	\longrightarrow <code>bool</code>
<code>maxPaquetesMandados</code>	: <code>dcnet</code>	\longrightarrow <code>pc</code>
<code>auxMaxPaquetes</code>	: <code>dcnet</code> \times <code>conj(pc)</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>damePaquete</code>	: <code>buffer</code>	\longrightarrow <code>paquete</code>
<code>pasarA</code>	: <code>topologia</code> \times <code>pc</code> \times <code>pc</code>	\longrightarrow <code>pc</code>

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

<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),c)</code>	\equiv 0
<code>paquetesMandados(seg(dcn),c)</code>	\equiv <code>paquetesMandados(dcn)</code>
<code>paquetesMandados(paquetePendiente(dcn,o,d,p),c)</code>	\equiv if $c = o$ then \quad <code>paquetesMandados(dcn, c) + 1</code> else \quad <code>paquetesMandados(dcn, c)</code> fi
<code>dcNetBuffer(dcn,c)</code>	\equiv <code>obtener(c,regresion(topo(dcn),vacio,cronoPaquetes(dcn,vacio)))</code>
<code>maxPaquetesMandados(dcn)</code>	\equiv <code>auxMaxPaquetes(dcn,pcs(topo(dcn)))</code>
<code>auxMaxPaquetes(dcn,cs)</code>	\equiv if $\emptyset?(sinUno(cs))$ then \quad <code>dameUno(cs)</code> else \quad if <code>paquetesMandados(dcn, dameUno(cs))</code> < <code>paquetesMandados(dcn, auxMaxPaquetes(dcn, sinUno(cs)))</code> then $\quad\quad$ <code>auxMaxPaquetes(dcn, sinUno(cs))</code> \quad else $\quad\quad$ <code>dameUno(cs)</code> \quad fi fi
<code>paqueteEnTransito?(dcn,p)</code>	\equiv <code>perteneceBuffers?(p,regresion(topo(dcn),vacio,cronoPaquetes(dcn,vacio)))</code>
<code>perteneceBuffers?(p,bs)</code>	\equiv if $\emptyset?(claves(bs))$ then \quad <code>false</code> else \quad if $p \in obtener(dameUno(claves(bs)), bs)$ then $\quad\quad$ <code>true</code> \quad else $\quad\quad$ <code>perteneceBuffers?(p, borrar(dameUno(claves(bs)), bs))</code> \quad fi fi
<code>cronoPaquetes(crearRed(t),bs)</code>	\equiv <code><></code>
<code>cronoPaquetes(seg(dcn),bs)</code>	\equiv <code>bs • cronoPaquetes(dcn,∅)</code>
<code>cronoPaquetes(paquetePendiente(dcn,o,d,p),bs)</code>	\equiv <code>auxDefinir(dp, o, Ag(p, ∅), obtener(o, bs))</code> <code>cronoPaquetes(dcn, bs)</code>
<code>auxDefinir(bs,c,n,v)</code>	\equiv if $def?(c, bs)$ then \quad <code>borrar(c, bs) definir(c, n ∪ v, bs)</code> else \quad <code>definir(c, n)</code> fi
<code>auxBorrar(bs,c,b,p)</code>	\equiv if $\emptyset?(p - \{b\})$ then \quad <code>borrar(c, n)</code> else \quad <code>borrar(c, bs) definir(c, p - \{b\}, bs)</code> fi
<code>regresion(t,bs,cbs)</code>	\equiv if $vacia?(fin(cbs))$ then \quad <code>pasoSeg(bs, t, prim(cbs))</code> else \quad <code>regresion(t, pasoSeg(bs, t, prim(cbs)), fin(cbs))</code> fi
<code>pasoSeg(t,bs,nbs)</code>	\equiv <code>envioYReciboPaquetes(t,bs,claves(bs))</code> <code>nuevosPaquetes(bs,nbs)</code>

```

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

pasarA(t,o,d)
envio(t,bs,b)
≡ prim(caminoMin(t,o,d))
≡ auxDefinir(bs,pasarA(t, $\Pi_1(b)$ ,dest( $\Pi_2(b)$ )),
  Ag(damePaquete(b), $\emptyset$ ),obtener
  (pasarA(t, $\Pi_1(b)$ ,dest( $\Pi_2(b)$ )),bs)
  auxBorrar(bs, $\Pi_1(b)$ ,damePaquete(b),
  obtener(bs, $\Pi_1(b)$ ))

nuevosPaquetes(bs,nbs)
≡ if  $\emptyset?(claves(nbs))$  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

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

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

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