

TAD característica ES string

TAD Restriccion

generos restriccion

exporta AND, OR, NOT, VAR, cumple, FALSE, TRUE

igualdad observacional

$(\forall r_1, r_2 : \text{restriccion}) (r_1 =_{\text{obs}} r_2 \iff$

$(\forall c : \text{conj}(\text{caracteristica})) (\text{cumple}(c, r_1) =_{\text{obs}} \text{cumple}(c, r_2)))$

observadores basicos

$\text{cumple} : \text{conj}(\text{caracteristica}) \times \text{restriccion} \rightarrow \text{bool}$

generadores

AND: $\text{restriccion} \times \text{restriccion} \rightarrow \text{restriccion}$

OR: $\text{restriccion} \times \text{restriccion} \rightarrow \text{restriccion}$

NOT: $\text{restriccion} \rightarrow \text{restriccion}$

VAR: $\text{caracteristica} \rightarrow \text{restriccion}$

otras operaciones

TRUE: $\rightarrow \text{restriccion}$

FALSE: $\rightarrow \text{restriccion}$

axiomas $(\forall cs : \text{conj}(\text{caracteristica})) (\forall c_1, c_2 : \text{restriccion}) (\forall c : \text{caracteristica})$

$\text{cumple}(cs, \text{VAR}(c)) \equiv c \in cs$

$\text{cumple}(cs, \text{NOT}(c_1)) \equiv \neg(\text{cumple}(cs, c_1))$

$\text{cumple}(cs, \text{AND}(c_1, c_2)) \equiv \text{cumple}(cs, c_1) \wedge \text{cumple}(cs, c_2)$

$\text{cumple}(cs, \text{OR}(c_1, c_2)) \equiv \text{cumple}(cs, c_1) \vee \text{cumple}(cs, c_2)$

$\text{TRUE} \equiv \text{OR}(\text{VAR}(\text{"dummy"}), \text{NOT}(\text{VAR}(\text{"dummy"})))$

$\text{FALSE} \equiv \text{AND}(\text{VAR}(\text{"dummy"}), \text{NOT}(\text{VAR}(\text{"dummy"})))$

Fin TAD

TAD estacion ES string

TAD conexion ES (estacion, restriccion)

TAD conexiones ES conj(conexion)

TAD Mapa

generos mapa

exporta estaciones, conexiones, nuevo, crearEst, conectar,

esBloqueante, conectadas, camino

igualdad observacional

$(\forall m_1, m_2 : \text{mapa}) (m_1 =_{\text{obs}} m_2 \iff$

$\text{estaciones}(m_1) =_{\text{obs}} \text{estaciones}(m_2) \wedge$

$(\forall e \in \text{estaciones}(m_1)) (\text{conexiones}(m_1, e) =_{\text{obs}} \text{conexiones}(m_2, e)))$

observadores basicos

$\text{estaciones} : \text{mapa} \rightarrow \text{conj}(\text{estacion})$

$\text{conexiones} : \text{mapa } m \times \text{estacion } e \rightarrow \text{conj}(\text{conexion}) \{e \in \text{estaciones}(m)\}$

generadores

nuevo : $\rightarrow \text{mapa}$

crearEst : $\text{mapa } m \times \text{estacion } a \rightarrow \text{mapa } \{\neg(a \in \text{estaciones}(m))\}$

conectar : $\text{mapa } m \times \text{estacion } a \times \text{estacion } b \times \text{restriccion} \rightarrow \text{mapa}$
 $\{a, b \in \text{estaciones}(m) \wedge \neg \text{conectadas}(m, a, b)\}$

otras operaciones

$\text{esBloqueante} : \text{mapa } m \times \text{conj}(\text{caracteristica}) \times \text{estacion } e \rightarrow \text{bool } \{e \in \text{estaciones}(m)\}$

$\text{esBloqueante}' : \text{conj}(\text{conexion}) \times \text{conj}(\text{caracteristica}) \rightarrow \text{bool}$

$\text{conectadas} : \text{mapa } m \times \text{estacion } a \times \text{estacion } b \rightarrow \text{bool } \{a, b \in \text{estaciones}(m)\}$

$\text{conectadas}' : \text{estacion } \times \text{conj}(\text{conexion}) \rightarrow \text{bool}$

$\text{camino} : \text{mapa } m \times \text{estacion } a \times \text{estacion } b \rightarrow \text{restriccion}$

$\{a, b \in \text{estaciones}(m) \wedge \text{conectadas}(m, a, b)\}$

$\text{camino}' : \text{estacion } e \times \text{conj}(\text{conexion}) \text{ cc} \rightarrow \text{restriccion}$

$\{(\exists x \in \text{cc}) (\pi_1(x) =_{\text{obs}} e)\}$

axiomas $(\forall m : \text{mapa}) (\forall e, a, b : \text{estacion}) (\forall r : \text{restriccion}) (\forall cs : \text{conj}(\text{caracteristica}))$

$(\forall cc : \text{conj}(\text{conexion}))$

$\text{conexiones}(\text{crearEst}(m, e), k) \equiv \emptyset$

$\text{conexiones}(\text{conectar}(m, a, b, r), e) \equiv$

$(\text{if } e =_{\text{obs}} a \text{ then } \langle b, r \rangle$

$\text{else if } e =_{\text{obs}} b \text{ then } \langle a, r \rangle \text{ else } \emptyset \text{ fi}$

$\text{fi}) \cup \text{conexiones}(m, e)$

$\text{estaciones}(\text{nuevo}) \equiv \emptyset$

$\text{estaciones}(\text{crearEst}(m, e)) \equiv \{e\} \cup \text{estaciones}(m)$

```

estaciones(conectar(m, a, b))  $\equiv$  estaciones(m)

esBloqueante(m, cs, e)  $\equiv$ 
  if  $\phi?$ (conexiones(m, e)) then False
  else esBloqueante'(conexiones(m, e), cs) fi

esBloqueante'(cc, cs)  $\equiv$ 
  if  $\phi?$ (cc) then True
  else  $\neg$ (cumple(r,  $\pi_2$ (dameUno(cc))))  $\wedge$  esBloqueante'(sinUno(cc), cs) fi

conectadas(m, a, b)  $\equiv$  conectadas'(b, conexiones(m, a))

conectadas'(a, cc)  $\equiv$ 
  if  $\phi?$ (cc) then False
  else ( $\pi_1$ (dameUno(cc)) =obs a)  $\vee$  conectadas'(a, sinUno(cc)) fi

camino(m, a, b)  $\equiv$  camino'(b, conexiones(m, a))

camino'(a, cc)  $\equiv$ 
  if  $\pi_1$ (dameUno(cc)) =obs a then  $\pi_2$ (dameUno(cc))
  else caminos'(a, sinUno(cc)) fi
Fin TAD

TAD rur ES nat
TAD robot ES (rur, conj(caracteristica))

TAD Ciudad
  generos ciudad
  exporta mapeo, robots, posicion, #infracciones, #inspecciones, nueva, agregar,
    mover, inspeccion, caracteristicaMasInfractora

igualdad observacional
( $\forall c_1, c_2 : \text{ciudad}$ ) ( $c_1 = \text{obs } c_2 \iff$ 
  mapeo( $c_1$ ) =obs mapeo( $c_2$ )  $\wedge$  robots( $c_1$ ) =obs robots( $c_2$ )  $\wedge$ 
  (( $\forall e \in \text{estaciones}(\text{mapeo}(c))$ ) (#inspecciones( $c_1$ , e) =obs #inspecciones( $c_2$ , e)))  $\wedge$ 
  (( $\forall r \in \text{robots}(c_1)$ ) (
    #infracciones( $c_1$ ,  $\pi_1(r)$ ) =obs #infracciones( $c_2$ ,  $\pi_1(r)$ )  $\wedge$ 
    posicion( $c_1$ ,  $\pi_1(r)$ ) =obs posicion( $c_2$ ,  $\pi_1(r)$ ))))))

observadores basicos
  mapeo: ciudad  $\rightarrow$  mapa
  robots: ciudad  $\rightarrow$  conj(robot)
  posicion: ciudad c  $\times$  rur i  $\rightarrow$  estacion {( $\exists r \in \text{robots}(c)$ )  $\pi_1(r)$  =obs i}
  #infracciones: ciudad c  $\times$  rur i  $\rightarrow$  nat {( $\exists r \in \text{robots}(c)$ )  $\pi_1(r)$  =obs i}
  #inspecciones: ciudad c  $\times$  estacion e  $\rightarrow$  nat {e  $\in$  estaciones(mapeo(c))}
generadores
  nueva: mapa  $\rightarrow$  ciudad

  agregar: ciudad c  $\times$  rur i  $\times$  conj(caracteristica) cs  $\times$  estacion e  $\rightarrow$  ciudad
    { $\neg \phi?(cs) \wedge \neg((\exists r \in \text{robots}(c)) \pi_1(r) = \text{obs } i) \wedge e \in \text{estaciones}(\text{mapeo}(c)) \wedge$ 
     $\neg \text{esBloqueante}(\text{mapeo}(c), cs, e)$ }

  mover: ciudad c  $\times$  rur i  $\times$  estacion e  $\rightarrow$  ciudad
    {e  $\in$  estaciones(mapeo(c))  $\wedge ((\exists r \in \text{robots}(c)) \pi_1(r) = \text{obs } i) \wedge$ 
    conectadas(mapeo(c), posicion(c, i), e)}

  inspeccion: ciudad c  $\times$  estacion e  $\rightarrow$  ciudad {e  $\in$  estaciones(mapeo(c))}
otras operaciones

masInfracciones: ciudad c  $\times$  estacion e  $\rightarrow$  conj(robot) {e  $\in$  estaciones(mapeo(c))}
robotsEnEstacion: ciudad c  $\times$  conj(robot) cr  $\times$  estacion  $\rightarrow$  conj(robot) {cr  $\subseteq$  robots(c)}
maxInfractor: ciudad c  $\times$  conj(robot) cr  $\rightarrow$  robot
  { cr  $\subseteq$  robots(c)  $\wedge \neg \phi?(cr)$ }

caracteristicaMasInfractora: ciudad c  $\rightarrow$  caracteristica
  {( $\exists r \in \text{robots}(c)$ ) (#infracciones( $\pi_1(r)$ ) > 0)}
infractoras: ciudad c  $\times$  conj(robot) cr  $\rightarrow$  multiconj(caracteristica) {cr  $\subseteq$  robots(c)}
aMulti: conj(caracteristica)  $\rightarrow$  multiconj(caracteristica)
maximizar: multiconj(caracteristica) mc  $\rightarrow$  caracteristica { $\neg \phi?(mc)$ }
agregarN: nat  $\times$  conj(caracteristica)  $\rightarrow$  multiconj(caracteristica)

```

```

axiomas (Vm : mapa) (Vc : ciudad) (Vi : rur) (Vcc : conj(caracteristicas))
  (Ve, e' : estacion) (Vcr : conj(robot)) (Vmc : multiconj(caracteristica))
mapeo(nueva(m)) ≡ m
mapeo(agregar(c, i, cc, e)) ≡ mapeo(c)
mapeo(mover(c, i, e)) ≡ mapeo(c)
mapeo(inspeccion(c, e)) ≡ mapeo(c)

robots(nueva) ≡ ∅
robots(agregar(c, i, cc, e)) ≡ Ag({i, cc}, robots(c))
robots(mover(c, i, e)) ≡ robots(c)
robots(inspeccion(c, e)) ≡ robots(c) - masInfracciones(c, e)

posicion(agregar(c, i, cc, e), r) ≡
  if i =obs r then e
  else posicion(c, r) fi
posicion(mover(c, i, e), r) ≡
  if i =obs r then e
  else posicion(c, r) fi
posicion(inspeccion(c, e), r) ≡ posicion(c, r)

#infracciones(agregar(c, i, cc, e), r) ≡
  if i =obs r then 0
  else #infracciones(c) fi

#infracciones(inspeccion(c, e), r) ≡ #infracciones(c, r)

#infracciones(mover(c, i, e), r) ≡
  if r =obs i then
    #infracciones(c, r) + β(¬cumple(camino(mapeo(c), posicion(c, r), e)))
  else #infracciones(c, r) fi

#inspecciones(nueva, e) ≡ 0
#inspecciones(agregar(c, i, cc, e'), e) ≡ #inspecciones(c)
#inspecciones(mover(c, i, e'), e) ≡ #inspecciones(c)
#inspecciones(inspeccion(c, e'), e) ≡ β(e' =obs e) + #inspecciones(c, e)

masInfracciones(c, e) ≡
  if ¬φ?(robotsEnEstacion(c, robots(c), e)) ∧1
    #infracciones(c, π1(maxInfractor(c, robotsEnEstacion(c, robots(c), e)))) > 0
  then Ag(maxInfractor(c, robotsEnEstacion(c, robots(c), e)), ∅)
  else ∅ fi

maxInfractor(c, cr) ≡
  if φ?(sinUno(cr)) ∨1 #infracciones(c, π1(dameUno(cr))) >
    #infracciones(c, π1(maxInfractor(c, sinUno(cr))))
  then dameUno(cr)
  else maxInfractor(c, sinUno(cr)) fi

robotsEnEstacion(c, cr, e) ≡
  if φ?(cr) then
    ∅
  else
    if posicion(c, π1(dameUno(cr))) =obs e then
      Ag(dameUno(cr), ∅)
    else
      ∅
    fi ∪ robotsEnEstacion(c, sinUno(cr), e)
  fi

caracteristicaMasInfractora ≡ maximizar(infractoras(c, robots(c)))

infractoras(c, cr) ≡
  if φ?(cr) then ∅
  else
    infractoras(c, sinUno(cr)) ∪
    if #infracciones(c, dameUno(cr)) > 0 then
      agregarN(#infracciones(c, dameUno(cr)), π2(dameUno(cr)))
    else ∅ fi
  fi

```

```

aMulti(cs) ≡ if  $\phi?(cs)$  then  $\phi$  else Ag(dameUno(cs), aMulti(sinUno(cs))) fi
agregarN(i, cs) ≡ if i =obs 0 then  $\phi$  else aMulti(cs) u agregarN(i-1, cs) fi
maximizar(mc) ≡
  if  $\phi?(sinUno(mc)) \vee \#(dameUno(mc), mc) > \#(maximizar(sinUno(mc)), mc)$  then
    dameUno(mc)
  else maximizar(sinUno(mc)) fi
Fin TAD

```