UBA – Facultad de Ciencias Exactas y Naturales – Departamento de Computación

Algoritmos y Estructura de Datos I

Segundo cuatrimestre de 2016

5 de septiembre de 2016

TPE OJOTA (Organización de Juegos Olímpicos Tp de Algoritmos 1) v1.0

1. Auxiliares definidas por nosotros

Las siguientes funciones auxiliares serán usadas para varios problemas a lo largo del TP. Si la definición de alguna función no aparece en la especificación de un problema, está en esta sección.

`aux participa` (j:JJOO,a:Atleta) : Bool $= (\exists c \leftarrow competencias(j))a \in participantes(c)$;
`aux paises` (j:JJOO) : [Pais] $= [\, nacionalidad(a) \,|\, a \leftarrow atletas(j) \,]$;
`aux ganoOroPais` (c:Competencia,p:Pais) : Bool $= |ranking(c)| \geq 1 \wedge nacionalidad(ranking(c)_0) == pais$;
`aux ganoPlataPais` (c:Competencia,p:Pais) : Bool $= |ranking(c)| \geq 2 \wedge nacionalidad(ranking(c)_1) == pais$;
`aux ganoBroncePais` (c:Competencia,p:Pais) : Bool $= |ranking(c)| \geq 3 \wedge nacionalidad(ranking(c)_2) == pais$;
`aux mismaCompetencia` (c_1,c_2:Competencia) : Bool $= (categoria(c_1) == categoria(c_2))$
$\wedge\, (mismos(participantes(c_2), participantes(c2))) \wedge (finalizada(c_1) == finalizada(c_2))$
$\wedge\, (finalizada(c_1) \rightarrow ((ranking(c_1) == ranking(c_2))$
$\wedge\, (mismos(lesTocoControlAntiDoping(c_1), lesTocoControlAntiDoping(c_2))$
$\wedge\, ((\forall a \leftarrow lesTocoControlAntiDoping(c_1))leDioPositivo(c_1, a) == leDioPositivo(c_2, a))))$;

2. Tipos

`tipo` Deporte = String ;
`tipo` Pais = String ;
`tipo` Sexo = Femenino, Masculino ;

3. Atleta

```
tipo Atleta {
        observador nombre (a: Atleta) : String ;
        observador sexo (a: Atleta) : Sexo ;
        observador añoNacimiento (a: Atleta) : ℤ ;
        observador nacionalidad (a: Atleta) : Pais ;
        observador ciaNumber (a: Atleta) : ℤ ;
        observador deportes (a: Atleta) : [Deporte] ;
        observador capacidad (a: Atleta, d: Deporte) : ℤ ;
             requiere d ∈ deportes(a) ;

        invariante |deportes(a)| > 0 ;
        invariante sinRepetidos(deportes(a)) ;
        invariante ordenada(deportes(a)) ;
        invariante capacidadEnRango : (∀d ← deportes(a))0 ≤ capacidad(a, d) ≤ 100 ;
}

problema especialidad (a: Atleta) = res : Deporte {
        asegura result ∈ deportes(a) ;
        asegura (∀d ← deportes(a))capacidad(a, d) ≤ capacidad(a, res) ;
}

problema entrenarNuevoDeporte (a: Atleta, d: Deporte, c: ℤ) {
        requiere ¬(d ∈ deportes(a)) ;
        requiere 0 ≤ c ≤ 100 ;

        modifica a ;

        asegura nombre(a) == nombre(pre(a)) ;
        asegura sexo(a) == sexo(pre(a)) ;
        asegura añoNacimiento(a) == añoNacimiento(pre(a)) ;
```

asegura $nacionalidad(a) == nacionalidad(\mathsf{pre}(a))$;
asegura $ciaNumber(a) == ciaNumber(\mathsf{pre}(a))$;
asegura SeAgregoDeporte : $mismos(deportes(a), d : deportes(\mathsf{pre}(a))) \land ordenada(deportes(a))$;
asegura LosOtrosNoCambiaron : $(\forall x \leftarrow deportes(a), x \neq d) capacidad(a, x) == capacidad(\mathsf{pre}(a), x)$;
asegura CapacidadCorrecta : $capacidad(a, d) == c$;
}

4. Competencia

tipo Competencia {
observador categoria (c: Competencia) : (Deporte, Sexo) ;
observador participantes (c: Competencia) : [Atleta] ;
observador finalizada (c: Competencia) : Bool ;
observador ranking (c: Competencia) : [Atleta] ;
requiere $finalizada(c)$;
observador lesTocoControlAntiDoping (c: Competencia) : [Atleta] ;
requiere $finalizada(c)$;
observador leDioPositivo (c: Competencia, a: Atleta) : Bool ;
requiere $finalizada(c) \land a \in lesTocoControlAntiDoping(c)$;

invariante participaUnaSolaVez : $sinRepetidos(ciaNumbers(participantes(c)))$;
invariante participantesPertenecenACat :
$(\forall p \leftarrow participantes(c)) prm(categoria(c)) \in deportes(p) \land sgd(categoria(c)) == sexo(p)$;
invariante elRankingEsDeParticipantesYNoHayRepetidos :
$finalizada(c) \Rightarrow incluida(ranking(c), participantes(c))$;
invariante seControlanParticipantesYNoHayRepetidos :
$finalizada(c) \Rightarrow incluida(lesTocoControlAntiDoping(c), participantes(c))$;
}

problema finalizarCompetencia (c: Competencia, posiciones: [Atleta], control: [(Atleta, Bool)]) {
requiere $\neg finalizada(c)$;
requiere PosicionesCumplenInvariante : $incluida(posiciones, participantes(c))$;
requiere ControladosCumplenInvariante : $incluida(atletasControlados(control), participantes(c))$;

modifica c ;

asegura $(categoria(c) == categoria(\mathsf{pre}(c))) \land mismos(participantes(c), participantes(\mathsf{pre}(c)))$;
asegura $finalizada(c)$;
asegura RankingCorrecto : $ranking(c) == posiciones$;
asegura ControlCorrecto : $mismos(lesTocoControlAntiDoping(c), atletasControlados(control))$;
asegura ResultadoAntiDopingCorrecto : $(\forall a \leftarrow control)\mathsf{sgd}(a) == leDioPositivo(c, \mathsf{prm}(a))$;

aux atletasControlados (c:[(Atleta,Bool)]) : [Atleta] $= [\, \mathsf{prm}(a) \mid a \leftarrow c \,]$;
}

problema linfordChristie (c: Competencia, a: Atleta) {
requiere $\neg finalizada(c)$;
requiere $a \in participantes(c)$;

modifica c ;

asegura $\neg finalizada(c) \land categoria(c) == categoria(\mathsf{pre}(c))$;
asegura AtletaDescalificado : $mismos(participantes(\mathsf{pre}(c)), a : participantes(c))$;
}

problema gananLosMasCapaces (c: Competencia) = res : Bool {
requiere $finalizada(c)$;

asegura $res == (\forall i \leftarrow [1..|ranking(c)| - 1))$
$capacidadCompetencia(c, ranking(c)_i) \geq capacidadCompetencia(c, ranking(c)_{i+1})$;

aux capacidadCompetencia (c:Competencia,a:Atleta) : $\mathbb{Z} = capacidad(a, \mathsf{prm}(categoria(c)))$;
}

problema sancionarTramposos (c: Competencia) {
requiere $finalizada(c)$;

modifica c ;

asegura $finalizada(c) \wedge categoria(c) == categoria(\mathsf{pre}(c)) \wedge mismos(participantes(c), participantes(\mathsf{pre}(c)))$;
asegura $ranking(c) == rankingSinTramposos(\mathsf{pre}(c))$;
asegura $mismos(lesTocoControlAntiDoping(c), lesTocoControlAntiDoping(\mathsf{pre}(c)))$;
asegura $(\forall a \leftarrow lesTocoControlAntiDoping(c))leDioPositivo(c, a) == leDioPositivo(\mathsf{pre}(c), a)$;

aux rankingSinTramposos (c:Competencia) : [Atleta] $= [\, a \mid a \leftarrow ranking(c),$
$a \in lesTocoControlAntiDoping(c) \rightarrow leDioPositivo(c, a) ==$ False $]$;
}

5. JJOO

tipo JJOO {
observador año (j: JJOO) : $\mathbb{Z}$;
observador atletas (j: JJOO) : [Atleta] ;
observador cantDias (j: JJOO) : $\mathbb{Z}$;
observador cronograma (j: JJOO, dia: $\mathbb{Z}$) : [Competencia] ;
requiere $1 \leq dia \leq cantDias(j)$;
observador jornadaActual (j: JJOO) : $\mathbb{Z}$;

invariante atletasUnicos : $sinRepetidos(ciaNumbers(atletas(j)))$;
invariante unaDeCadaCategoria : $(\forall i, k \leftarrow [0..|competencias(j)|), i \neq k)$
$categoria(competencias(j)_i) \neq categoria(competencias(j)_k)$;
invariante competidoresInscriptos : $(\forall c \leftarrow competencias(j))incluida(participantes(c), atletas(j))$;
invariante jornadaValida : $1 \leq jornadaActual(j) \leq cantDias(j)$;
invariante finalizadasSiiYaPasoElDia : $lasPasadasFinalizaron(j) \wedge lasQueNoPasaronNoFinalizaron(j)$;
}

problema dePaseo (j: JJOO) = res : [Atleta] {
asegura $mismos(res, noParticipan(j))$;
aux noParticipan (j:JJOO) : [Atleta] $= [\, a \mid a \leftarrow atletas(j), \neg participa(j, a) \,]$;
}

problema medallero (j: JJOO) = res : [(Pais, [$\mathbb{Z}$])] {
asegura OrdenadoPorOro : $(\forall i \leftarrow [0..|res| - 1))\mathsf{sgd}(res_i)_0 \geq \mathsf{sgd}(res_{i+1})_0$;
asegura OrdenadoPorPlata : $(\forall i \leftarrow [0..|res| - 1), \mathsf{sgd}(res_i)_0 == \mathsf{sgd}(res_{i+1})_0)\mathsf{sgd}(res_i)_1 \geq \mathsf{sgd}(res_{i+1})_1$;
asegura OrdenadoPorBronce : $(\forall i \leftarrow [0..|res| - 1), \mathsf{sgd}(res_i)_1 == \mathsf{sgd}(res_{i+1})_1)\mathsf{sgd}(res_i)_2 \geq \mathsf{sgd}(res_{i+1})_2$;
asegura NoHayPaisesRepetidos : $sinRepetidos([\, \mathsf{prm}(p) \mid p \leftarrow res \,])$;
asegura TodosGanaronMedallas : $(\forall p \leftarrow res) \sum \mathsf{sgd}(p) > 0$;
asegura EstanTodosLosQueGanaron : $(\forall p \leftarrow paises(j), totalMedallasPais(j, p) > 0)((\exists q \leftarrow res)\mathsf{prm}(q) == p)$;
asegura CantidadCorrectaDeMedallas : $(\forall p \leftarrow res)$
$((totalOroPais(j, \mathsf{prm}(p)) == \mathsf{sgd}(p)_0) \wedge (totalPlataPais(j, \mathsf{prm}(p)) == \mathsf{sgd}(p)_1)$
$\wedge (totalBroncePais(j, \mathsf{prm}(p)) == \mathsf{sgd}(p)_2))$;

aux totalOroPais (j:JJOO,p:Pais) : $\mathbb{Z} = |[\, c \mid c \leftarrow competencias(j), finalizada(c) \wedge ganoOroPais(c, p) \,]|$;
aux totalPlataPais (j:JJOO,p:Pais) : $\mathbb{Z} = |[\, c \mid c \leftarrow competencias(j), finalizada(c) \wedge ganoPlataPais(c, p) \,]|$;
aux totalBroncePais (j:JJOO,p:Pais) : $\mathbb{Z} = |[\, c \mid c \leftarrow competencias(j), finalizada(c) \wedge ganoBroncePais(c, p) \,]|$;
aux totalMedallasPais (j:JJOO,p:Pais) : $\mathbb{Z} = totalOroPais(j, p) + totalPlataPais(j, p) + totalBroncePais(j, p)$;
}

problema boicotPorDisciplina (j: JJOO, cat: (Deporte, Sexo), p: Pais) = res : $\mathbb{Z}$ {
requiere CategoriaValida : $(\exists c \leftarrow competencias(j))categoria(c) == cat$;

modifica j ;

asegura $año(j) == año(\mathsf{pre}(j)) \wedge cantDias(j) == cantDias(\mathsf{pre}(j)) \wedge jornadaActual(j) == jornadaActual(\mathsf{pre}(j)) \wedge$
$mismos(atletas(j), atletas(\mathsf{pre}(j)))$;
asegura NoModificaOtrasCompetencias : $(\forall d \leftarrow [1..cantDias(j)])$
$((\forall c_1 \leftarrow cronograma(\mathsf{pre}(j), d), categoria(c_1) \neq cat)((\exists c_2 \leftarrow cronograma(j, d))mismaCompetencia(c_1, c_2)))$;
asegura NoCreaCompetenciasNuevas : $(\forall d \leftarrow [1..cantDias(j)])$
$((\forall c_1 \leftarrow cronograma(j, d), categoria(c_1) \neq cat)((\exists c_2 \leftarrow cronograma(\mathsf{pre}(j), d))mismaCompetencia(c_1, c_2)))$;
asegura $(\forall d \leftarrow [1..cantDias(j)])\, |cronograma(j, d)| == |cronograma(\mathsf{pre}(j), d)|$;
asegura CatFueModificada : $(\exists c_1 \leftarrow competencias(j), categoria(c_1) == cat)$
$((\exists c_2 \leftarrow competencias(\mathsf{pre}(j)), categoria(c_2) == cat)$
$(mismos(participantes(c_1), sacarPais(p, participantes(c_2))))$
$\wedge (finalizada(c_1) == finalizada(c_2))$

$\wedge\ (finalizada(c_1) \rightarrow ((ranking(c_1) == sacarPais(p, ranking(c_2)))$
$\wedge\ (mismos(lesTocoControlAntiDoping(c_1), sacarPais(p, lesTocoControlAntiDoping(c_2))))$
$\wedge\ ((\forall a\ \leftarrow\ lesTocoControlAntiDoping(c_1))leDioPositivo(c_1, a) == leDioPositivo(c_2, a)))))$;

`asegura ResultadoCorrecto` : $res == |[\, a \,|\, a \leftarrow atletas(j), nacionalidad(a) == p \wedge participa(j, a)\,]|$;

`aux sacarPais` (p:Pais,l:[Atleta]) : [Atleta] $= [\, a \,|\, a \leftarrow l, nacionalidad(a) \neq p\,]$;

}

`problema losMasFracasados` (j: JJOO, p: Pais) = `res` : [Atleta] {

`asegura Participaron` : $(\forall a\ \leftarrow\ res)a \in losQueMasParticiparon(j, p)$;
`asegura NoGanaronNada` : $(\forall a\ \leftarrow\ res)totalMedallasAtleta(j, a) == 0$;
`asegura EstanTodos` : $(\forall a\ \leftarrow\ losQueMasParticiparon(j, p), totalMedallasAtleta(j, a) == 0)a \in res$;

`aux losQueMasParticiparon` (j:JJOO,p:Pais) : [Atleta] $= [\, a \,|\, a \leftarrow atletas(j), nacionalidad(a) == p$
$\wedge\ ((\forall b\ \leftarrow\ atletas(j), nacionalidad(b) == p)cantCompetencias(j, a) \geq cantCompetencias(j, b))\,]$;
`aux cantCompetencias` (j:JJOO,a:Atleta) : $\mathbb{Z} = |[\, c \,|\, c \leftarrow competencias(j), a \in participantes(c)\,]|$;
`aux totalMedallasAtleta` (j:JJOO,a:Atleta) : $\mathbb{Z} =$
$|[\, c \,|\, c \leftarrow competencias(j), finalizada(c) \wedge ganoMedallaAtleta(c, a)\,]|$;
`aux ganoMedallaAtleta` (c:Competencia,a:Atleta) : Bool $= (|ranking(c)| \geq 1 \wedge ranking(c)_0 == a)$
$\vee\ (|ranking(c)| \geq 2 \wedge ranking(c)_1 == a) \vee (|ranking(c)| \geq 3 \wedge ranking(c)_2 == a)$;

}

`problema liuSong` (j: JJOO, a: Atleta, p: País) {

`requiere` $a \in atletas(j)$;

`modifica` j ;

`asegura` $año(j) == año(\mathsf{pre}(j)) \wedge cantDias(j) == cantDias(\mathsf{pre}(j)) \wedge jornadaActual(j) == jornadaActual(\mathsf{pre}(j))$;
`asegura SoloCambioElAtletaDeseado` : $(\forall d\ \leftarrow\ [1..cantDias(j)])$
$((\forall c_1\ \leftarrow\ cronograma(\mathsf{pre}(j), d))((\exists c_2\ \leftarrow\ cronograma(j, d))mismaCompNacCamb(c_1, c_2, a, p))$
$\wedge\ |cronograma(j, d)| == |cronograma(\mathsf{pre}(j), d)|)$;

`aux mismoNacCamb` (a_1,a_2:Atleta,p:Pais) : Bool $= nombre(a_1) == nombre(a_2) \wedge sexo(a_1) == sexo(a_2)$
$\wedge\ añoNacimiento(a_1) == añoNacimiento(a_2) \wedge nacionalidad(a_2) == p \wedge ciaNumber(a_1) == ciaNumber(a_2)$
$\wedge\ deportes(a_1) == deportes(a_2) \wedge ((\forall d\ \leftarrow\ deportes(a_1))capacidad(a_1, d) == capacidad(a_2, d))$;
`aux mismosAtlNacCamb` (l_1,l_2:[Atleta],a:Atleta,p:Pais) : Bool $=$
$((\forall a_1\ \leftarrow\ l_1, ciaNumber(a_1) \neq ciaNumber(a))((\exists a_2\ \leftarrow\ l_2)a_1 == a_2))$
$\wedge\ ((\exists a_1\ \leftarrow\ l_1, ciaNumber(a_1) == ciaNumber(a_2)) \rightarrow ((\exists a_2\ \leftarrow\ l_2)mismosNacCamb(a_1, a_2, p)))$
$\wedge\ |l_1| == |l_2|$;
`aux mismaCompNacCamb` (c_1,c_2:Competencia,a:Atleta,p:Pais) : Bool $=$
$categoria(c_1) == categoria(c_2) \wedge mismosAtlNacCamb(participantes(c_1), participantes(c_2), a, p)$
$\wedge\ finalizada(c_1) == finalizada(c_2)$
$\wedge\ (finalizada(c_1) \rightarrow (mismosAtlNacCamb(ranking(c_1), ranking(c_2), a, p)$
$\wedge\ mismoOrden(ranking(c_1), ranking(c_2))$
$\wedge\ mismosAtlNacCamb(lesTocoControlAntiDoping(c_1), lesTocoControlAntiDoping(c_2), a, p)$
$\wedge\ ((\forall b\ \leftarrow\ lesTocoControlAntiDoping(c_1), ciaNumber(b) \neq ciaNumber(a))$
$leDioPositivo(c_1, b) == leDioPositivo(c_2, b))$
$\wedge\ ((\exists a_1\ \leftarrow\ lesTocoControlAntiDoping(c_1), a_2\ \leftarrow\ lesTocoControlAntiDoping(c_2))mismoNacCamb(a_1, a_2, p))$
$\rightarrow leDioPositivo(c_1, a_1) == leDioPositivo(c_2, a_2)))$;
`aux mismoOrden` (l,s:[Atleta]) : Bool $= (\forall i\ \leftarrow\ [0..|l|))ciaNumber(l_i) == ciaNumber(s_i)$;

}

`problema stevenBradbury` (j: JJOO) = `res` : Atleta {

`asegura` $res \in ganaronOro(j)$;
`asegura MenosCapaz` : $(\forall a\ \leftarrow\ ganaronOro(j))menorCapOro(j, a) \geq menorCapOro(j, res)$;

`aux menorCapOro` (j:JJOO, a:Atleta) : $\mathbb{Z} = minimo(capacidadOro(j, a))$;
`aux capacidadOro` (j:JJOO, a:Atleta) : $[\mathbb{Z}] = [\, capacidad(a, d) \,|\, d \leftarrow deportesOro(j, a)\,]$;
`aux deportesOro` (j:JJOO, a:Atleta) : [Deporte] $=$
$[\, prm(categoria(c)) \,|\, c \leftarrow competencias(j), finalizada(c) \wedge ganoOroAtleta(c, a)\,]$;
`aux ganoOroAtleta` (c:Competencia, a:Atleta) : Bool $= |ranking(c)| \geq 1 \wedge ranking(c)_0 == a$;

}

`problema uyOrdenadoAsíHayUnPatrón` (j: JJOO) = `res` : Bool {

`asegura` $res == hayPatron(mejoresPaises(j))$;

`aux hayPatron` (ps:[Pais]) : Bool $= (\forall i, j\ \leftarrow\ [0..|ps|), i \equiv j(|sacarRepetidos(ps)|))ps_i == ps_j$;
`aux mejoresPaises` (j:JJOO) : [Pais] $= [\, paisDelDia(cronograma(j, d)) \,|\, d \leftarrow [1..jornadaActual(j)]\,]$;

aux paisDelDia (cs:[Competencia]) : Pais $= cab([p|p \leftarrow paisesCompetidores(cs),$
$((\forall q \leftarrow paisesCompetidores(cs))totalOroComp(cs,p) \geq totalOroComp(cs,q))$
$\wedge ((\forall q \leftarrow paisesCompetidores(cs), totalOroComp(cs,p) == totalOroComp(cs,q))p \leq q)])$;
aux paisesCompetidores (cs:[Competencia]) : [Pais] $= [\, p \,|\, p \leftarrow paises(c), c \leftarrow (cs) \,]$;
aux totalOroComp (cs:[Competencia], p:Pais) : $\mathbb{Z} = |[\, c \,|\, c \leftarrow cs, finalizada(c) \wedge ganoOroPais(c,p) \,]|$;
aux sacarRepetidos (l:[T]) : [T] $= [\, l_i \,|\, i \leftarrow [0..|l|), l_i \notin l_{[0..i)} \,]$;
}

problema sequíaOlímpica (j: JJOO) = res : [País] {
asegura SonLosDeMayorSequia : $(\forall p \leftarrow res)((\forall q \leftarrow paises(j))mayorSequia(j,p) \geq mayorSequia(j,q))$;
asegura EstanTodos : $(\forall p \leftarrow paises(j), ((\forall q \leftarrow paises(j))mayorSequia(j,p) \geq mayorSequia(j,q)))p \in res$;

aux mayorSequia (j:JJOO, p:Pais) : $\mathbb{Z} =$ cab$([\, k-i \,|\, i,k \leftarrow [1..jornadaActual(j)], i < k \wedge haySequia(j,p,i,k) \wedge$
$((\forall l,m \leftarrow [1..jornadaActual(j)], l < m \wedge haySequia(j,p,l,m))k-i \geq m-l) \,])$;
aux haySequia (j:JJOO, p:Pais, i,k:$\mathbb{Z}$) : Bool $= (\forall d \leftarrow [i..k])((\forall c \leftarrow cronograma(j,d))\neg ganoMedallaPais(c,p))$;
aux ganoMedallaPais (c:Competencia, p:Pais) : Bool $=$
$ganoOroPais(c,p) \vee ganoPlataPais(c,p) \vee ganoBroncePais(c,p)$;
}

problema transcurrirDia (j: JJOO) {
requiere $jornadaActual(j) \leq cantDias(j)$;

modifica j ;

asegura $jornadaActual(j) == jornadaActual(\text{pre}(j)) + 1$;
asegura $mismos(atletas(j), atletas(\text{pre}(j)))$;
asegura NoCambioCompetenciasAnteriores : $(\forall d \leftarrow [1..jornadaActual(\text{pre}(()j)) - 1])$
$mismoCronograma(cronograma(j,d), cronograma(\text{pre}(j),d))$;
asegura NoCambioCompetenciasSiguientes : $(\forall d \leftarrow [jornadaActual(\text{pre}(j)) + 1..cantDias(j)])$
$mismoCronograma(cronograma(j,d), cronograma(\text{pre}(j),d))$;
asegura NoCambioFinalizadas : $(\forall c_1 \leftarrow cronograma(\text{pre}(j), jornadaActual(\text{pre}(j))), finalizada(c_1))$
$((\exists c_2 \leftarrow cronograma(j, jornadaActual(\text{pre}(j))))mismaCompetencia(c_1,c_2))$;
asegura LasNoFinalizadasTerminaronBien : $(\forall c_1 \leftarrow cronograma(\text{pre}(j), jornadaActual(\text{pre}(j))), \neg finalizada(c_1))$
$(\exists c_2 \leftarrow cronograma(j, jornadaActual(\text{pre}(j))))categoria(c_1) == categoria(c_2) \wedge$
$mismos(participantes(c_1), participantes(c_2)) \wedge finalizada(c_2)$
$\wedge ganaronCapaces(c_2) \wedge controlAntiDopingCorrecto(c_2))$;
asegura $|cronograma(\text{pre}(j), jornadaActual(\text{pre}(j)))| == |cronograma(j, jornadaActual(\text{pre}(j)))|$;
asegura PorcentajeDopingCorrecto : $|atletasDopados(cronograma(j, jornadaActual(\text{pre}(j))))| /$
$|cronograma(j, jornadaActual(\text{pre}(j)))| \leq 0{,}05$;
(Justificación: como en cada competencia se le realiza el control antidoping a un único atleta, la cantidad de atletas controlados es igual a la cantidad de competencias del día.)

aux mismoCronograma (cs_1, cs_2:[Competencia]) : Bool $= ((\forall c_1 \leftarrow cs_1)((\exists c_2 \leftarrow cs_2)mismaCompetencia(c_1,c_2))$
$\wedge ((\forall c_2 \leftarrow cs_2)((\exists c_1 \leftarrow cs_1)mismaCompetencia(c_1,c_2)) \wedge |cs_1| == |cs_2|)$;
aux ganaronCapaces (c:Competencia) : Bool $= incluida(ranking(c), participantes(c))$
$\wedge ((\forall i \leftarrow [0..|ranking(c)| - 1])$
$capacidad(ranking(c)_i, \text{prm}(categoria(c))) \geq capacidad(ranking(c)_{i+1}, \text{prm}(categoria(c))))$;
aux controlAntiDopingCorrecto (c:Competencia) : Bool $=$
$incluida(lesTocoControlAntiDoping(c), participantes(c))$
$\wedge |lesTocoControlAntiDoping(c)| == 1$;
aux atletasDopados (cs:[Competencia]) : [Atleta] $= [a|a \leftarrow lesTocoControlAntiDoping(c),$
$c \leftarrow cs, finalizada(c), leDioPositivo(c,a)]$;
}

6. Auxiliares

aux ciaNumbers (as: [Atleta]) : [$\mathbb{Z}$] $= [\, ciaNumber(a) \,|\, a \leftarrow as \,]$;
aux competencias (j: JJOO) : [Competencia] $= [\, c \,|\, d \leftarrow [1..cantDias(j)], c \leftarrow cronograma(j,d) \,]$;
aux incluida (l_1, l_2:[T]) : Bool $= (\forall x \leftarrow l_1)cuenta(x,l_1) \leq cuenta(x,l_2)$;
aux lasPasadasFinalizaron (j: JJOO) : Bool $= (\forall d \leftarrow [1..jornadaActual(j)))(\forall c \leftarrow cronograma(j,d))finalizada(c)$;
aux lasQueNoPasaronNoFinalizaron (j: JJOO) : Bool $=$
$(\forall d \leftarrow (jornadaActual(j)..cantDias(j)])(\forall c \leftarrow cronograma(j,d))\neg finalizada(c)$;
aux ordenada (l:[T]) : Bool $= (\forall i \leftarrow [0..|l| - 1))l_i \leq l_{i+1}$;
aux sinRepetidos (l: [T]) : Bool $= (\forall i,j \leftarrow [0..|l|), i \neq j)l_i \neq l_j$;