

Primer cuatrimestre 2012

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

**TP OJOTA (Organización de Juegos Olímpicos) v1.0**

**Grupo 4**

| Integrante            | LU     | Correo electrónico        |
|-----------------------|--------|---------------------------|
| Nicolas Lasso         | 763/10 | lasso.nico@gmail.com      |
| Guido Tripodi         | 843/10 | guido.tripodi@hotmail.com |
| Tomas Agustin Shaurli | 671/10 | zeratulzero@hotmail.com   |
| Patricio Inzaghi      | 255/11 | pinzaghi@dc.uba.ar        |

# 1. Resolucion

1. problema entrenarNuevoDeporte (a: Atleta, d: Deporte, c:  $\mathbb{Z}$ ) {  
  requiere  $d \notin \text{deportes}(\text{pre}(a))$ ;  
  requiere  $0 \leq c \leq 100$ ;  
  modifica  $a$ ;  
  asegura  $\text{nombre}(a) == \text{nombre}(\text{pre}(a))$ ;  
  asegura  $\text{sexo}(a) == \text{sexo}(\text{pre}(a))$ ;  
  asegura  $\text{añoNacimiento}(a) == \text{añoNacimiento}(\text{pre}(a))$ ;  
  asegura  $\text{naionalidad}(a) == \text{nacionalidad}(\text{pre}(a))$ ;  
  asegura  $\text{ciaNumber}(a) == \text{ciaNumber}(\text{pre}(a))$ ;  
  asegura  $\text{mismos}(\text{deportes}(a), \text{deportes}(\text{pre}(a)) : d)$ ;  
  asegura  $\text{ordenada}(\text{deportes}(a))$ ;  
  asegura  $\text{capacidad}(a, d) == c$ ;  
}
2. problema finalizarCompetencia (c: Competencia, posiciones: [Atleta], control: [(Atleta, Bool)]) {  
  modifica  $c$ ;  
  asegura  $\text{categoria}(c) == \text{categoria}(\text{pre}(c))$ ;  
  asegura  $\text{participantes}(c) == \text{participantes}(\text{pre}(c))$ ;  
  asegura  $\text{finalizada}(c)$ ;  
  asegura  $\text{ranking}(c) == \text{posiciones}$ ;  
  asegura  $\text{mismos}([\text{prm}(d) | d \leftarrow \text{control}], \text{lesTocoControlAntiDoping}(c))$ ;  
  asegura  $(\forall x \leftarrow \text{control}), \text{lesDioPositivo}(c, \text{prm}(x)) == \text{sgd}(x)$ ;  
}
3. problema linfordChristie (c: Competencia, a: Atleta) {  
  requiere  $\text{finalizada}(c) == \text{False}$ ;  
  requiere  $\text{en}(a, \text{participantes}(c))$ ;  
  modifica  $c$ ;  
  asegura  $\text{categoria}(c) == \text{categoria}(\text{pre}(c))$ ;  
  asegura  $\text{atletaNoPertenece}(a, \text{participantes}(c))$ ;  
  asegura  $\text{long}(\text{participantes}(c)) == \text{long}(\text{participantes}(\text{pre}(c)) - 1$ ;  
  asegura  $\text{mismos}(a : \text{participantes}(c), \text{participantes}(\text{pre}(c)))$ ;  
}
4. problema gananLosMasCapaces (c: Competencia) = result : Bool {  
  requiere  $\text{finalizada}(c) == \text{True}$ ;  
  requiere  $|\text{ranking}(c)| > 1$ ;  
  asegura  $\text{Result} == \text{masCapaces}(c)$ ;  
}
5. problema sancionarTramposos (c: Competencia) {  
  modifica  $c$ ;  
  asegura  $\text{categoria}(c) == \text{categoria}(\text{pre}(c))$ ;  
  asegura  $\text{finalizada}(c)$ ;  
  asegura  $\text{participantes}(c) == \text{participantes}(\text{pre}(c))$ ;  
  asegura  $\text{lesTocoControlAntiDoping}(c) == \text{lesTocoControlAntiDoping}(\text{pre}(c))$ ;  
  asegura  $\text{leDioPositivo}(c, a) == \text{leDioPositivo}(\text{pre}(c), \text{pre}(a))$ ;  
  asegura  $\text{ranking}(c) == \text{borrarPositivos}(\text{pre}(c))$ ;  
}

6. problema dePaseo (j: JJOO) = result : [Atleta] {  
 requiere  $\text{cantDias}(j) \geq 1$ ;  
 requiere  $|\text{atletas}(j)| > 0$ ;  
 asegura  $\text{result} == [x|y \leftarrow \text{cantDias}(j), k \leftarrow \text{cronograma}(j, y), x \leftarrow \text{atletas}(j), x \notin \text{participantes}(k)]$ ;  
}
7. problema medallero (j: JJOO) = result : [(País, [Z])] {  
 asegura  $(\forall p \leftarrow \text{result}, \text{algunaMedalla}(\text{sgd}(p)))$ ;  
 asegura  $(\forall p \leftarrow \text{result}, |\text{sgd}(p)| == 3)$ ;  
 asegura  $(\forall x \leftarrow [0..|\text{result}| - 1], \text{mejoresOIgualesMedallas}(\text{sgd}(\text{result}_x), \text{sgd}(\text{result}_{x+1})))$ ;  
}
8. problema boicotPorDisciplina (j: JJOO, cat: (Deporte, Sexo), p: País ) = result : Z {  
 requiere  $\text{long}(\text{atletas}(j)) > 0$ ;  
 modifica  $j$ ;  
 asegura  $\text{año}(j) == \text{año}(\text{pre}(j))$ ;  
 asegura  $\text{cantDias}(j) == \text{cantDias}(\text{pre}(j))$ ;  
 asegura  $\text{cronograma}(j) == \text{cronograma}(\text{pre}(j))$ ;  
 asegura  $\text{jornadaactual}(j) == \text{jornadaactual}(\text{pre}(j))$ ;  
 asegura  $(\forall a \in \text{atletasEnCategoria}(j, \text{cat}, p)) a \notin \text{atletas}(j)$ ;  
 asegura  $\text{result} == |\text{atletasEnCategoria}(j, \text{cat}, p)|$ ;  
}
9. problema losMasFracasados (j: JJOO, p: País ) = result : [Atleta] {  
 asegura  $(\forall a \in \text{result}) \text{noGanoMedallas}(j, a)$ ;  
 asegura  $(\forall x \in \text{result}, y \in \text{atletasSinMedallasPorPais}(j, p)) \text{competenciasDelAtleta}(j, x) > \text{competenciasDelAtleta}(j, y)$ ;  
 aux noGanoMedallas (j:JJOO,a:Atleta) : Bool =  $(\forall c \in \text{competencias}(j), \text{finalizada}(c)) a \neq \text{ranking}(c)_0 \wedge a \neq \text{ranking}(c)_1 \wedge a \neq \text{ranking}(c)_2$ ;  
 aux competenciasDelAtleta (j:JJOO,a:Atleta) : Bool =  $[c|c \in \text{competencias}(j), a \in \text{participantes}(c)]$ ;  
 aux atletasSinMedallasPorPais (j:JJOO,p:País) : Bool =  $[a|a \in \text{atletas}(j), \text{nacionalidad}(a) == p, \text{noGanoMedallas}(j, a)]$ ;  
}
10. problema liuSong (j: JJOO, a: Atleta, p: País ) {  
 requiere  $a \in \text{atletas}(\text{pre}(j))$ ;  
 modifica  $a, j$ ;  
 asegura  $\text{nombre}(a) == \text{nombre}(\text{pre}(a))$ ;  
 asegura  $\text{sexo}(a) == \text{sexo}(\text{pre}(a))$ ;  
 asegura  $\text{añoNacimiento}(a) == \text{añoNacimiento}(\text{pre}(a))$ ;  
 asegura  $\text{ciaNumber}(a) == \text{ciaNumber}(\text{pre}(a))$ ;  
 asegura  $\text{depores}(a) == \text{deportes}(\text{pre}(a))$ ;  
 asegura  $\text{nacionalidad}(a) == p$ ;  
 asegura  $\text{año}(j) == \text{año}(\text{pre}(j))$ ;  
 asegura  $\text{cantDias}(j) == \text{cantDias}(\text{pre}(j))$ ;  
 asegura  $\text{cronograma}(j) == \text{cronograma}(\text{pre}(j))$ ;  
 asegura  $\text{jornadaActual}(j) == \text{jornadaActual}(\text{pre}(j))$ ;  
 asegura  $|\text{atletas}(j)| == |\text{atletas}(\text{pre}(j))|$ ;  
 aux quitarAtleta (j:JJOO,a:Atleta) : [Atleta] =  $[x|x \leftarrow \text{atletas}(j), x \neq a]$ ;  
 asegura  $\text{atletas}(j) == (\text{quitarAtleta}(\text{pre}(j), \text{pre}(a)) : a)$ ;  
}

11. problema `stevenBradbury` (j: JJOO) = `result` : `Atleta` {  
`requiere` ( $\exists dia \in [1...cantDias(j)], cron \in cronograma(j, dia), comp \in cron) finalizada(comp)$  ;  
`asegura` ( $\exists dia \in [1...cantDias(j)], cron \in cronograma(j, dia), comp \in cron) result == ranking(comp)_0$  ;  
`asegura` ( $\forall a \in atletas(j), a \neq result) capacidad(result) \leq capacidad(a)$  ;  
}
12. problema `uyOrdenadoAsíHayUnPatrón` (j: JJOO) = `result` : `Bool` {  
`aux mejoresAtletas` (j:JJOO) : [`Atleta`] = [`ranking(c)`<sub>0</sub> |  $c \in competenciasFinalizadas(j)$ ] ;  
`aux paisesConRepetidos` (j: JJOO) : [`Pais`] = [`nacionalidad(a)` |  $a \in atletas(j)$ ] ;  
`aux orosPorPais` (l:[`Atleta`]) :  $\mathbb{Z}$  = [`l`] ;  
}
13. problema `sequíaOlimpica` (j: JJOO) = `result` : [`País`] {  
`aux todosLosPaises` (J:JJOO) : [`País`] = [`nacionalidad(x)` |  $x \leftarrow atletas(j)$ ] ;  
`asegura` *sinRepetidos*(`todosLosPaises(j)`) ;  
}
14. problema `transcurrirDia` (j: JJOO) {  
}

## 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) :  $\mathbb{Z}$  ;
  observador nacionalidad (a: Atleta) : Pais ;
  observador ciaNumber (a: Atleta) :  $\mathbb{Z}$  ;
  observador deportes (a: Atleta) : [Deporte] ;
  observador capacidad (a: Atleta, d: Deporte) :  $\mathbb{Z}$  ;
    requiere  $d \in \text{deportes}(a)$  ;

  invariante  $\text{sinRepetidos}(\text{deportes}(a))$  ;
  invariante  $\text{ordenada}(\text{deportes}(a))$  ;
  invariante  $\text{capacidadEnRango} : (\forall d \leftarrow \text{deportes}(a)) 0 \leq \text{capacidad}(a, d) \leq 100$  ;
}
```

## 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  $\text{finalizada}(c)$  ;
  observador lesTocoControlAntiDoping (c: Competencia) : [Atleta] ;
    requiere  $\text{finalizada}(c)$  ;
  observador leDioPositivo (c: Competencia, a: Atleta) : Bool ;
    requiere  $\text{finalizada}(c) \wedge a \in \text{lesTocoControlAntiDoping}(c)$  ;

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

## 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  $1 \leq \text{dia} \leq \text{cantDias}(j)$  ;
  observador jornadaActual (j: JJOO) :  $\mathbb{Z}$  ;

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

}

## 6. Auxiliares

```

aux algunaMedalla (m: [Z]) : Bool = (m0 ≠ 0 ∨ m1 ≠ 0 ∨ m2 ≠ 0);
aux atletaNoPertenece (a: Atleta, l: [Atletas]) : Bool = (∀x ← l, a ≠ x);
aux ciaNumbers (as: [Atleta]) : [Z] = [ciaNumber(a) | a ← as];
aux competencias (j: JJOO) : [Competencia] = [c | d ← [1..cantDias(j)], c ← cronograma(j, d)];
aux competenciasFinalizadas (j: JJOO) : [Competencia] = [c | C ← [1..cantDias(j)], c ← cronograma(j, d), finalizada(c)];
aux incluida (l1, l2: [T]) : Bool = (∀x ← l1) cuenta(x, l1) ≤ cuenta(x, l2);
aux lasPasadasFinalizaron (j: JJOO) : Bool = (∀d ← [1..jornadaActual(j)]) (∀c ← cronograma(j, d)) finalizada(c);
aux lasQueNoPasaronNoFinalizaron (j: JJOO) : Bool =
  (∀d ← (jornadaActual(j)..cantDias(j))) (∀c ← cronograma(j, d)) ¬finalizada(c);
aux ordenada (l: [T]) : Bool = (∀i ← [0..|l| - 1]) li ≤ li+1;
aux sinRepetidos (l: [T]) : Bool = (∀i, j ← [0..|l|], i ≠ j) li ≠ lj;
aux masCapaces (c: competencia) : Bool = (∀x ← [0..|ranking(c)| - 1], capacidad(ranking(c)x) > capacidad(ranking(c)x+1));
aux mejoresOIgualesMedallas (m: [Z], p: [Z]) : Bool = superaPorOro(m, p);
aux superaPorOro (m: [Z], p: [Z]) : Bool = ifThenElse(m0 > p0, True, ifThenElse(m0 == p0, superaPorPlata(m, p), False));
aux superaPorPlata (m: [Z], p: [Z]) : Bool = ifThenElse(m1 > p1, True, ifThenElse(m1 == p1, superaPorBronce(m, p), False));
aux superaPorBronce (m: [Z], p: [Z]) : Bool = ifThenElse(m2 > p2, True, ifThenElse(m2 == p2, True, False));
aux listaPositivos (c: competencia) : [Atleta] = [lesTocoControlAntiDoping(c), lesDioPositivo(c, x) == True];
aux borrarPositivos (c: competencia) : [Atleta] = [x | x ← ranking(c), (∀y ← listaPositivos(c)) y ≠ x];
aux atletasdelpaisout (cat: (Deporte, Sexo), p: Pais, j: JJOO) : Bool = (x ← [0..|atletas(j)| - 1], p(cat(atletas(j)x)) ≠
  atletas(j)x);
aux atletasEnCategoria (j: JJOO, cat: Categoria, p: País ) : [Atleta] = [a | a ∈ atletas(j), nacionalidad(a) == p ∧
  sexo(a) == sgd(cat) ∧ prm(cat) ∈ deportes(a)];

```

## 7. Aclaraciones

1. En el ejercicio 4 requerimos que el ranking sea mayor a uno ya que sino no quedaria sin efecto la funcion auxiliar ya que de esta manera no se podrian comparar los atletas porque habria uno solo.
2. En el ejercicio 7, en caso de empate en medallas de oro, se compara las medallas de plata, si siguen siendo las mismas, se compara las medallas de bronce. En caso de tener las 3 medallas iguales se considera mejor al primero que se comparo.