

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}) {
}
2. problema finalizarCompetencia (c: Competencia, posiciones: [Atleta], control: [(Atleta, Bool)]) {
}
3. problema linfordChristie (c: Competencia, a: Atleta) {
 requiere *finalizada*(c) == False;
 requiere *en*(a, *participantes*(c));
 modifica c;
 asegura *categoria*(c) == *categoria*(*pre*(c));
 asegura *atletaNoPertenece*(a, *participantes*(c));
 asegura *long*(*participantes*(c)) == *long*(*participantes*(*pre*(c)) - 1;
 asegura *mismos*(a : *participantes*(c), *participantes*(*pre*(c)));
}
4. problema gananLosMasCapaces (c: Competencia) = result : Bool {
 requiere *finalizada*(c) == True;
 requiere | *ranking*(c) | > 1;
 asegura *Result* == *masCapaces*(c);
}
5. problema sancionarTramposos (c: Competencia) {
}
6. problema dePaseo (j: JJOO) = result : [Atleta] {
 requiere *cantDias*(j) ≥ 1;
 requiere | *atletas*(j) | > 0;
 asegura *result* == [x|y ← *cantDias*(j) ∧ k ← *cronograma*(j, y) ∧ x ← *atletas*(j), x ∉ *participantes*(k)];
}
7. problema medallero (j: JJOO) = result : [(País, \mathbb{Z})] {
}
8. problema boicotPorDisciplina (j: JJOO, cat: (Deporte, Sexo), p: País) = result : \mathbb{Z} {
}
9. problema losMasFracasados (j: JJOO, p: País) = result : [Atleta] {
}
10. problema liuSong (j: JJOO, a: Atleta, p: País) {
}
11. problema stevenBradbury (j: JJOO) = result : Atleta {
}
12. problema uyOrdenadoAsiHayUnPatrón (j: JJOO) = result : Bool {
}
13. problema sequiaOlimpica (j: JJOO) = result : [País] {
}
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 ciaNumbers (as: [Atleta]) : [Z] = [ciaNumber(a) | a ← as];
aux competencias (j: JJOO) : [Competencia] = [c | d ← [1..cantDias(j)], c ← cronograma(j, d)];
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 atletaNoPertenece (a: Atleta, l: [Atletas]) : Bool = (∀x ← l, a ≠ x);
```

7. Aclaraciones

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.