

Algoritmos y Estructura de Datos I

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 $\text{atletaPertenezca} : a \in \text{participantes}(c)$;
 modifica c ;
 asegura $\text{categoria}(c) == \text{categoria}(\text{pre}(c))$;
 asegura $\text{atletaNoPertenece} : a \notin \text{participantes}(c)$;
 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 $\text{paísesConMedallas} : (\forall p \leftarrow \text{result}, \text{algunaMedalla}(\text{sgd}(p)))$;
 asegura $\text{soloTresTiposDeMedallas} : (\forall p \leftarrow \text{result}, |\text{sgd}(p)| == 3)$;
 asegura $\text{mejoresMedallasQueSiguiente} : (\forall x \leftarrow [0..|\text{result}|-1], \text{mejoresOIgualesMedallas}(\text{sgd}(\text{result}_x), \text{sgd}(\text{result}_{x+1})))$;
 aux mejoresOIgualesMedallas (m: [Z], p: [Z]) : Bool = $\text{superaPorOro}(m, p)$;
 aux superaPorOro (m: [Z], p: [Z]) : Bool = $\text{ifThenElse}(m_0 > p_0, \text{True}, \text{ifThenElse}(m_0 == p_0, \text{superaPorPlata}(m, p), \text{False}))$;
 aux superaPorPlata (m: [Z], p: [Z]) : Bool = $\text{ifThenElse}(m_1 > p_1, \text{True}, \text{ifThenElse}(m_1 == p_1, \text{superaPorBronce}(m, p), \text{False}))$;
 aux superaPorBronce (m: [Z], p: [Z]) : Bool = $\text{ifThenElse}(m_2 > p_2, \text{True}, \text{ifThenElse}(m_2 == p_2, \text{True}, \text{False}))$;
 }
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] {
 aux atletasDelPais (j:JJOO, p: País) : [Atleta] = $[x|x \leftarrow \text{atletas}(j), \text{nacionalidad}(x) == p]$;
 aux competenciasAtleta (j:JJOO, p:País, a:Atleta) : [Competencia] = $[x|x \leftarrow \text{competencias}(j), \text{alguno}(\text{participantes}(x) == a) \wedge \text{en}(\text{prm}(\text{categoria}(x)), \text{deportes}(a))]$;
 aux atletasFracasados (j:JJOO, p:País) : [(atleta, Z)] = $[(y, |\text{competenciasAtleta}(j, p, y)|)| y \leftarrow \text{atletasDelPais}(j, p), c \leftarrow \text{competenciasAtleta}(j, y, p), \text{en}(y, \text{sub}(\text{ranking}(c), 3, |\text{ranking}(c) - 1|))]$;
 aux masFracasados (j:JJOO, p:País) : [Atleta] = $[\text{prm}(y)|y \leftarrow \text{atletasFracasados}(j, p), k \leftarrow \text{atletasFracasados}(j, p), \text{sgd}(y) \geq \text{sgd}(k)]$;
 asegura $\text{result} == \text{masFracasados}(j, p)$;
 }
10. problema liuSong (j: JJOO, a: Atleta, p: País) {
 requiere $a \in \text{atletas}(j)$;
 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 $\text{mismos}(\text{deportes}(a), \text{deportes}(\text{pre}(a)))$;
 asegura $\text{cambioNacionalidad} : \text{nacionalidad}(a) == p$;
 asegura $\text{año}(j) == \text{año}(\text{pre}(j))$;
 asegura $\text{cantDias}(j) == \text{cantDias}(\text{pre}(j))$;
 asegura $\text{mismos}(\text{cronograma}(j), \text{cronograma}(\text{pre}(j)))$;
 asegura $\text{jornadaActual}(j) == \text{jornadaActual}(\text{pre}(j))$;

```

asegura  $|atletas(j)| == |atletas(pre(j))|$ ;
asegura  $mismosAtletas : (\forall x \in atletas(pre(j)))ciaNumber(x) \in cianumbers(j)$ ;
}

11. problema stevenBradbury (j: JJO) = result : Atleta {
  requiere  $algunaCompetenciaFinalizada(j)$ ;
  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: JJO) = result : Bool {
  asegura  $result == verificaPaises(j)$ ;
  aux competenciasPorDia (j:JJO) :  $[(\mathbb{Z}, [Competencia])]$  =  $[(dia, c)|dia \leftarrow [1..jornadaActual(j)], c \leftarrow cronograma(j, dia)]$ ;
  aux oroPorDia (j:JJO) :  $[(\mathbb{Z}, [Atleta])]$  =  $[(dia, a)|h \leftarrow competenciasPorDia(j), dia \leftarrow prm(h), x \leftarrow sgd(h),$ 
     $i \leftarrow [0..|x| - 1], a \leftarrow ranking(x)_i]$ ;
  aux paisesPorDia (j:JJO) :  $[(\mathbb{Z}, [Pais])]$  =  $[(dia, p)|x \leftarrow oroPorDia(j), dia \leftarrow prm(x), h \leftarrow sgd(x),$ 
     $i \leftarrow [0..|h| - 1], p \leftarrow nacionalidad(h)_i]$ ;
  aux mejoresPaises (j:JJO) :  $[(\mathbb{Z}, [Pais])]$  =  $[(dia, p)|x \leftarrow paisesPorDia(j), dia \leftarrow prm(x), p \leftarrow sgd(x),$ 
     $i \leftarrow [0..|sgd(x)| - 2], (cuenta(p, sgd(x)) \geq cuenta(sgd(x)_{i+1}, sgd(x)))]$ ;
  aux paisesMasGrosos (j:JJO) : [Pais] =  $[p|x \leftarrow mejoresPaises(j), p \leftarrow sgd(x), i \leftarrow [0..|sgd(x)| - 2], p \leq sgd(x)_{i+1}]$ ;
  aux patronDePaises (j:JJO) : [Pais] =  $[sub(paisesMasGrosos(j), 0, i-2)|i \leftarrow [0..|paisesMasGrosos(j)| - 1], \neg sinRepetidos(su$ 
  aux verificaPaises (j:JJO) : Bool =  $\forall i \in [0..|paisesMasGrosos(j)| - 1],$ 
     $paisesMasGrosos(j)_i == paisesMasGrosos(j)_{i+|patronDePaises(j)|} \wedge$ 
     $((i + |patronDePaises(j)|) \leq |paisesMasGrosos(j)|)$ ;
}

13. problema sequíaOlimpica (j: JJO) = result : [País] {
  aux todosLosPaises (J:JJO) : [País] =  $[nacionalidad(x)|x \leftarrow atletas(j)]$ ;
  asegura  $sinRepetidos(todosLosPaises(j))$ ;
}

14. problema transcurrirDia (j: JJO) {
  requiere  $jornadaActual(j) < cantDias(j)$ ;
  modifica j;
  asegura  $cantDias(j) == cantDias(pre(j))$ ;
  asegura  $año(j) == año(pre(j))$ ;
  asegura  $mismos(atletas(j), atletas(pre(j)))$ ;
  asegura  $(\forall d \in [1...cantDias(pre(j))])cronograma(pre(j), d) == cronograma(j, d)$ ;
  asegura  $cambiaElDia : jornadaActual(j) == jornadaActual(pre(j)) + 1$ ;
  asegura  $rankingSegunMasCapaces : (\forall c \in cronograma(pre(j), jornadaActual(pre(j))))masCapaces(c)$ ;
  asegura  $unControlPorCompetencia : (\forall c \in cronograma(pre(j), jornadaActual(pre(j)))$ 
     $|lesTocaControlAntiDoping(c)| == 1)$ ;
  asegura  $dopingPositivoMax5Porciento : (\exists a \in atletasLesTocoAntiDoping(j, jornadaActual(pre(j))))$ 
     $0 \leq porcentajePositivos(j) \leq 5$ ;
  aux porcentajePositivos (j:JJO) :  $\mathbb{R} = |atletasLesDioPositivo(j, jornadaActual(pre(j)))| * 100$ 
     $/|atletasLesTocoAntiDoping(j, jornadaActual(pre(j)))|$ ;
  aux atletasLesTocoAntiDoping (j:JJO, d:Z) : [Atleta] =  $[lesTocoControlAntiDoping(c)_0|c \in cronograma(j, d)]$ ;
  aux atletasLesDioPositivo (j:JJO, d:Z) : [Atleta] =  $[lesTocoControlAntiDoping(c)_0|c \in cronograma(j, d),$ 
     $lesTocoControlAntiDoping(c)_0 \in leDioPositivo(c)]$ ;
}

```

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 atletasEnCategoria (j:JJO, cat: Categoria, p: País ) : [Atleta] = [a | a ∈ atletas(j), nacionalidad(a) == p ∧
    sexo(a) == sgd(cat) ∧ prm(cat) ∈ deportes(a)];
aux algunaCompetenciaFinalizada (j:JJO) : Bool = (∃ dia ∈ [1..cantDias(j)], cron ∈ cronograma(j, dia), comp ∈
    cron) finalizada(comp);
aux algunaMedalla (m: [Z]) : Bool = (m0 ≠ 0 ∨ m1 ≠ 0 ∨ m2 ≠ 0);
aux borrarPositivos (c:competencia) : [Atleta] = [x | x ← ranking(c), (∀ y ← listaPositivos(c)) y ≠ x];
aux ciaNumbers (as: [Atleta]) : [Z] = [ciaNumber(a) | a ← as];
aux competencias (j: JJO) : [Competencia] = [c | d ← [1..cantDias(j)], c ← cronograma(j, d)];
aux competenciasFinalizadas (j: JJO) : [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: JJO) : Bool = (∀ d ← [1..jornadaActual(j)]) (∀ c ← cronograma(j, d)) finalizada(c);
aux lasQueNoPasaronNoFinalizaron (j: JJO) : Bool =
    (∀ d ← (jornadaActual(j)..cantDias(j))) (∀ c ← cronograma(j, d)) ¬ finalizada(c);
aux listaPositivos (c:competencia) : [Atleta] = [lesTocoControlAntiDoping(c), lesDioPositivo(c, x) == True];
aux ordenada (l: [T]) : Bool = (∀ i, j ← [0..|l| - 1]) li ≤ lj+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 atletasdelpaisout (cat: (Deporte, Sexo), p: Pais, j: JJO) : Bool = (x ← [0..|atletas(j)| - 1], p(cat(atletas(j)x)) ≠
    atletas(j)x);

```

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.