

#include "../include/ojota.h"

Atleta atletaProdigio (const JJOO &j){

// P: |competenciasFinalizadasConOroEnPodio(j)|>0

vector<Atleta> atletasCampeones;

//estado E0: vale |atletasCampeones|=0

unsigned int i = 0;

//estado E1: vale atletasCampeones=atletasCampeones@E0 && i==0

//implica |atletasCampeones|=0 (por E0)

**//Pc: |atletasCampeones|=0 && i==0 && |CompetenciasFinalizadasConOroEnPodio(j)|>0
(Por E1 y P)**

while (i<j.competenciasFinalizadasConOroEnPodio().size()){

**//l: 0<=i<=|competenciasFinalizadasConOroEnPodio(j)| && (∀a<-
atletasCampeones) (∃c <- competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a**

//v: |competenciasFinalizadasConOroEnPodio(j)| - i. c: 0

//estado C1: vale l && B

**atletasCampeones.push_back(j.competenciasFinalizadasConOroEnPodio()
[i].ranking()[0]);**

**//estado C2: vale atletasCampeones = atletasCampeones@C1 ++
[ranking(competenciasFinalizadasConOroEnPodio(j)[i])[0]] && i = [i@C1](#)**

i++;

//estado C3: vale atletasCampeones = atletasCampeones@C2 && i = i@C2 + 1

}

**//Qc: i == |competenciasFinalizadasConOroEnPodio(j)| && (∀a<-atletasCampeones) (∃c <-
competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a**

//estado E2: vale Qc

unsigned int k = 0;

//estado E3: vale k==0 && i == i@E2 && atletasCampeones == [atletasCampeones@E2](#)

Atleta prodigio = atletasCampeones[0];

**//estado E4: vale i == i@E3 && atletasCampeones == atletasCampeones@E3 &&
prodigio==atletasCampeones[0] && k==k@E3**

**//implica i == i@E2 && atletasCampeones == atletasCampeones@E2 &&
prodigio==atletasCampeones[0] && k==0 (por E3)**

**//implica i == |competenciasFinalizadasConOroEnPodio(j)| && atletasCampeones ==
[ranking(c)[0] | c<-competenciasFinalizadasConOroEnPodio(j)] && prodigio==atletasCampeones[0]
&& k==0 (por E2)**

**//Pc: atletasCampeones==atletasCampeones@E4 && i == |
competenciasFinalizadasConOroEnPodio(j)| && atletasCampeones == [ranking(c)[0] | c<-
competenciasFinalizadasConOroEnPodio(j)] && prodigio==atletasCampeones[0] && k==0**

```

while ( $k < \text{atletasCampeones.size}()$ ){

    //I:  $\text{atletasCampeones} == \text{atletasCampeones}@E4 \ \&\& \ 0 \leq k \leq |\text{atletasCampeones}| \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ (\forall a \leftarrow \text{atletasCampeones}[0..k]) \ \text{añoNacimiento}(a) \leq \text{añoNacimiento}(\text{prodigio})$ 
    //v:  $|\text{atletasCampeones}| - i. \ c: \ 0$ 
    //estado C4: vale I && B
    //Pif:  $\text{atletasCampeones} == \text{atletasCampeones}@E4 \ \&\& \ k < |\text{atletasCampeones}| \ \&\& \ \text{prodigio} == \text{prodigio}@E4 \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio}$ 

    if ( $\text{prodigio.añoNacimiento}() < \text{atletasCampeones}[k].\text{añoNacimiento}()$ ){

        //Eif1: vale Pif && B

        prodigio = atletasCampeones[k];

        //Eif2: vale  $\text{atletasCampeones} == \text{atletasCampeones}@Eif1 \ \&\& \ k = k@Eif1 \ \&\& \ \text{prodigio} == \text{atletasCampeones}[k]$ 

    }

    //Qif:  $\text{atletasCampeones} == \text{atletasCampeones}@C4 \ \&\& \ k == k@C4 \ \&\& \ k < |\text{atletasCampeones}| \ \&\& \ (\text{prodigio} == \text{prodigio}@C4 \ \&\& \ \text{añoNacimiento}(\text{prodigio}) \geq \text{añoNacimiento}(k) \ || \ \text{prodigio} == \text{atletasCampeones}[k]) \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio}$ 
    //estado C5: vale Qif
    //implica  $k < |\text{atletasCampeones}| \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ \text{añoNacimiento}(\text{prodigio}) \geq \text{añoNacimiento}(k) \ (A == B \rightarrow A \geq B)$ 

    k++;

    //estado C6: vale  $\text{atletasCampeones} == \text{atletasCampeones}@C5 \ \&\& \ k = k@C5 + 1 \ \&\& \ \text{prodigio} == \text{prodigio}@C5$ 

}

//Qc:  $\text{atletasCampeones} == \text{atletasCampeones}@E4 \ \&\& \ k == |\text{atletasCampeones}| \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ (\forall a \leftarrow \text{atletasCampeones}) \ \text{añoNacimiento}(a) \leq \text{añoNacimiento}(\text{prodigio})$ 
//estado E5: vale Qc

return prodigio;

//estado E6:  $\text{atletasCampeones} == \text{atletasCampeones}@E5 \ \&\& \ \text{vale} \ k == k@E5 \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ (\forall a \leftarrow \text{atletasCampeones}) \ \text{añoNacimiento}(a) \leq \text{añoNacimiento}(\text{prodigio})$ 
//implica  $k == |\text{atletasCampeones}| \ \&\& \ (\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ (\forall a \leftarrow \text{atletasCampeones}) \ \text{añoNacimiento}(a) \leq \text{añoNacimiento}(\text{prodigio})$  (por E5)
//implica  $(\exists a \leftarrow \text{atletasCampeones}) \ a == \text{prodigio} \ \&\& \ (\forall a \leftarrow \text{atletasCampeones}) \ \text{añoNacimiento}(a) \leq \text{añoNacimiento}(\text{prodigio}) \ (A \ \&\& \ B \rightarrow A)$ 
//implica  $(\exists c \leftarrow \text{competenciasFinalizadasConOroEnPodio}(j)) \ \text{ranking}(c)[0] == \text{prodigio} \ \&\& \ (\forall c \leftarrow \text{competenciasFinalizadasConOroEnPodio}(j)) \ \text{añoNacimiento}(\text{ranking}(c)[0]) \leq \text{añoNacimiento}(\text{prodigio})$  (por E2)

}

```

```
//Rama True
//Eif2
//implica atletasCampeones==atletasCampeones@Eif1 && k=k@Eif1 &&
prodigio==atletasCampeones[k] && (∃a<-atletasCampeones) a==prodigio (A==BS[n] -> (∃B<-BS)
A==B (en particular BS[n] es un valor de BS que cumple))
//implica atletasCampeones==atletasCampeones@C4 && k=k@C4 &&
prodigio==atletasCampeones[k] && (∃a<-atletasCampeones) a==prodigio
(atletasCampeones@Eif1==atletasCampeones@C4 && k@Eif1==k@C4 (Por Eif1 && Pif))
//implica atletasCampeones==atletasCampeones@C4 && k==k@C4 && k<|atletasCampeones|
&& (prodigio==prodigio@C4 && añoNacimiento(prodigio)>añoNacimiento(k) || prodigio ==
atletasCampeones[k]) && (∃a<-atletasCampeones) a==prodigio (A -> A || B)
//implica Qif (sintácticamente iguales)
```

```
//Rama False
//estado else0: vale Pif && ¬B: atletasCampeones==atletasCampeones@E4 && k<|
atletasCampeones| && (∃a<-atletasCampeones) a==prodigio && prodigio==prodigio@E4 &&
añoNacimiento(atletasCampeones[k])<añoNacimiento(prodigio)
//implica atletasCampeones==atletasCampeones@E4 && k==k@C4 && k<|atletasCampeones|
&& (∃a<-atletasCampeones) a==prodigio && prodigio==prodigio@E4 &&
añoNacimiento(atletasCampeones[k])<añoNacimiento(prodigio) (como no se realizó ninguna
operación se puede asumir que ninguna variable cambió su valor (en particular k))
//implica atletasCampeones==atletasCampeones@C4 && k==k@C4 && k<|atletasCampeones|
&& (prodigio==prodigio@C4 && añoNacimiento(prodigio)>añoNacimiento(k) || prodigio ==
atletasCampeones[k]) && (∃a<-atletasCampeones) a==prodigio (A -> A || B)
//implica Qif (sintácticamente iguales)
```

//Teorema de correctitud de ciclos

//ciclo 1

```
//Pc -> I
//i==0 && |competenciasFinalizadasConOroEnPodio(j)|>0 -> 0==i<|
competenciasFinalizadasConOroEnPodio(j)| -> 0<=i<|competenciasFinalizadasConOroEnPodio(j)|
//|atletasCampeones|=0 -> (∀a<-atletasCampeones) (∃c <-
competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a (todos los elementos de una lista
vacía cumplen cualquier condición)
```

```
//(I && ¬B) -> Qc
//I && ¬B: 0<=i<=|competenciasFinalizadasConOroEnPodio(j)| && (∀a<-atletasCampeones) (∃c <-
competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a && i>=|
competenciasFinalizadasConOroEnPodio|
//0<=i<=|competenciasFinalizadasConOroEnPodio(j)| && i>=|
competenciasFinalizadasConOroEnPodio| -> i == |competenciasFinalizadasConOroEnPodio(j)|
//(∀a<-atletasCampeones) (∃c <- competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a ->
(∀a<-atletasCampeones) (∃c <- competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a
```

//El cuerpo del ciclo preserva I

```
//Por I && B: 0<=i@C1<=|competenciasFinalizadasConOroEnPodio(j)| && i@C1<|
competenciasFinalizadasConOroEnPodio(j)|
//implica 0<=i@C1<|competenciasFinalizadasConOroEnPodio(j)|
//implica 0<=i@C2<|competenciasFinalizadasConOroEnPodio(j)| (i@C2==i@C1 (Por C2))
//implica 0<=i@C2+1<=|competenciasFinalizadasConOroEnPodio(j)|
//implica 0<=i@C3<=|competenciasFinalizadasConOroEnPodio(j)| (i@C3==i@C2+1 (por C3))
//Por I: (∀a<-atletasCampeones@C1) (∃c <- competenciasFinalizadasConOroEnPodio(j))
ranking(c)[0]=a
```

```
//implica (∀a<-atletasCampeones@C1) (∃c <- competenciasFinalizadasConOroEnPodio(j))
ranking(c)[0]=a && ∀a'<-[ranking(competenciasFinalizadasConOroEnPodio(j))[i]][0]] (∃c <-
competenciasFinalizadasConOroEnPodio(j)) ranking(c)[0]=a
//implica (∀a<-atletasCampeones@C2) (∃c <- competenciasFinalizadasConOroEnPodio(j))
ranking(c)[0]=a (atletasCampeones@C2==atletasCampeones@C1++
[ranking(competenciasFinalizadasConOroEnPodio(j))[i]][0])(Por C2))
//implica (∀a<-atletasCampeones@C3) (∃c <- competenciasFinalizadasConOroEnPodio(j))
ranking(c)[0]=a (atletasCampeones@C3==atletasCampeones@C2(Por C3))
```

```
//v@C3<v@C1
//v@C3 == |competenciasFinalizadasConOroEnPodio(j)| - i@C3 == |
competenciasFinalizadasConOroEnPodio(j)| - (i@C2 + 1) == |
competenciasFinalizadasConOroEnPodio(j)| - i@C1 - 1 < |
competenciasFinalizadasConOroEnPodio(j)| - i@C1 == v@C1
```

```
//v=c -> ¬B
//|competenciasFinalizadasConOroEnPodio(j)| - i = 0 -> |
competenciasFinalizadasConOroEnPodio(j)| == i -> ¬(i<|
competenciasFinalizadasConOroEnPodio(j)|)
```

//ciclo 2

```
//Pc -> I
//atletasCampeones==atletasCampeones@E4 -> atletasCampeones==atletasCampeones@E4
//k==0 -> 0<=k<=|atletasCampeones| && (∀a<-atletasCampeones[0..k)) añoNacimiento(a) <=
añoNacimiento(prodigio) (todos los elementos de una lista vacía cumplen cualquier condición)
//prodigio==atletasCampeones[0] ->(∃a<-atletasCampeones) a==prodigio
```

```
//(I && ¬B) -> Qc
//I && ¬B: atletasCampeones==atletasCampeones@E4 && 0<=k<=|atletasCampeones| && (∃a<-
atletasCampeones) a==prodigio && (∀a<-atletasCampeones[0..k)) añoNacimiento(a) <=
añoNacimiento(prodigio) &&k>=|atletasCampeones
//atletasCampeones==atletasCampeones@E4 -> atletasCampeones==atletasCampeones@E4
//0<=k<=|atletasCampeones| && k>=|atletasCampeones -> k==|atletasCampeones|
//(∃a<-atletasCampeones) a==prodigio -> (∃a<-atletasCampeones) a==prodigio
//(∀a<-atletasCampeones[0..k)) añoNacimiento(a) <= añoNacimiento(prodigio) && k==|
atletasCampeones| -> (∀a<-atletasCampeones) añoNacimiento(a) <= añoNacimiento(prodigio)
```

```
//El cuerpo del ciclo preserva I
//Por I: atletasCampeones@C4==atletasCampeones@E4
//implica atletasCampeones@C5==atletasCampeones@E4
(atletasCampeones@C5==atletasCampeones@C4(Por C5))
//implica atletasCampeones@C6==atletasCampeones@E4
(atletasCampeones@C6==atletasCampeones@C5(Por C6))
//Por I && B: 0<=k@C4<=|atletasCampeones| && k@C4<|atletasCampeones|
//implica 0<=k@C4<|atletasCampeones|
//implica 0<=k@C5<|atletasCampeones| (k@C5==k@C4(Por C5))
//implica 0<=k@C5+1<=|atletasCampeones|
//implica 0<=k@C6<=|atletasCampeones| (k@C6==k@C5+1(Por C6))
//Por I: (∃a<-atletasCampeones) a==prodigio@C4 && (∀a<-atletasCampeones[0..k))
añoNacimiento(a) <= añoNacimiento(prodigio@C4)
//(∃a<-atletasCampeones) a==prodigio@C5 && (∀a<-atletasCampeones[0..k)) añoNacimiento(a)
<= añoNacimiento(prodigio@C5) (prodigio@C5==prodigio@C4 &&
añoNacimiento(prodigio)>=añoNacimiento(k)|| prodigio == atletasCampeones[k](Por C5))
//(∃a<-atletasCampeones) a==prodigio@C6 && (∀a<-atletasCampeones[0..k)) añoNacimiento(a)
<= añoNacimiento(prodigio@C6) (prodigio@C6==prodigio@C5(Por C6))
```

//v@C3<v@C1

//v@C3 == |atletasCampeones| - i@c3 == |atletasCampeones| - (i@c2 + 1) == |
atletasCampeones| - i@c1 -1 < |atletasCampeones| - i@c1 = v@C1

//v=c -> ¬B

//|atletasCampeones| - i == 0 -> |atletasCampeones| == i -> ¬(i<|atletasCampeones|)