

Introducción a la programación

Práctica 2: Especificación de problemas

Ejercicio 1

Escribir semiformalmente los siguientes predicados.

- c) esMinimo: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el mínimo.

Ejercicio 1

Escribir semiformalmente los siguientes predicados.

- c) esMinimo: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el mínimo.

```
pred esMinimo( $s : seq\langle \mathbb{Z} \rangle$ ,  $elem : \mathbb{Z}$ ) {  
    elem pertenece a  $s$ , y no hay ningún elemento de  $s$  que sea  
    estrictamente menor a  $elem$   
}
```

Ejercicio 1

Escribir semiformalmente los siguientes predicados.

- c) esMinimo: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el mínimo.

```
pred esMinimo( $s : seq\langle \mathbb{Z} \rangle$ ,  $elem : \mathbb{Z}$ ) {  
    elem pertenece a  $s$ , y no hay ningún elemento de  $s$  que sea  
    estrictamente menor a  $elem$   
}
```

Un poco más formal...

Ejercicio 1

Escribir semiformalmente los siguientes predicados.

- c) esMinimo: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el mínimo.

```
pred esMinimo( $s : seq\langle \mathbb{Z} \rangle$ ,  $elem : \mathbb{Z}$ ) {  
     $elem$  pertenece a  $s$ , y no hay ningún elemento de  $s$  que sea  
    estrictamente menor a  $elem$   
}
```

Un poco más formal...

```
pred esMinimo ( $s : seq\langle \mathbb{Z} \rangle$ ,  $elem : \mathbb{Z}$ ) {  
     $elem \in s \wedge \neg(\exists x : \mathbb{Z})(x \in s \wedge x < elem)$   
}
```

Ejercicio 2

- e) buscar: que dado un entero *elem* y una secuencia de enteros *l* que incluye a *elem*, devuelva una posición donde esté *elem*.

Ejercicio 2

- e) buscar: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

problema buscar (l : $seq\langle\mathbb{Z}\rangle$, $elem$: \mathbb{Z}) : \mathbb{Z} {
 requiere: { $elem$ pertenece a l }

Ejercicio 2

- e) buscar: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

problema buscar ($l: seq\langle \mathbb{Z} \rangle$, $elem: \mathbb{Z}$) : \mathbb{Z} {
 requiere: { $elem$ pertenece a l }
 asegura OpcionA: { $resultado$ es una posición de l donde está
 el elemento $elem$ }

Ejercicio 2

- e) buscar: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

problema buscar ($l: seq\langle \mathbb{Z} \rangle$, $elem: \mathbb{Z}$) : \mathbb{Z} {
 requiere: { $elem$ pertenece a l }
 asegura OpcionA: { $resultado$ es una posición de l donde está el elemento $elem$ }
 asegura OpcionB: { $resultado$ es una posición válida de l donde está el elemento $elem$ }

Ejercicio 2

- e) buscar: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

problema buscar ($l: seq\langle \mathbb{Z} \rangle$, $elem: \mathbb{Z}$) : \mathbb{Z} {
 requiere: { $elem$ pertenece a l }
 asegura OpcionA: { $resultado$ es una posición de l donde está el elemento $elem$ }
 asegura OpcionB: { $resultado$ es una posición válida de l donde está el elemento $elem$ }
 asegura OpcionC: {se cumple $esPosicionValida(resultado, l)$ y además $l[resultado]$ es $elem$ }

Ejercicio 2

- e) buscar: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

```
problema buscar ( $l$ : seq( $\mathbb{Z}$ ),  $elem$ :  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
  requiere: { $elem$  pertenece a  $l$ }  
  asegura OpcionA: { $resultado$  es una posición de  $l$  donde está  
    el elemento  $elem$ }  
  asegura OpcionB: { $resultado$  es una posición válida de  $l$   
    donde está el elemento  $elem$ }  
  asegura OpcionC: {se cumple esPosicionValida( $resultado$ ,  $l$ ) y  
    además  $l[resultado]$  es  $elem$ }  
}
```

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

`problema ordenadaCrecientemente (l: $seq\langle\mathbb{Z}\rangle$) : Bool {`
 `requiere: {para todo elemento x de s , se cumple`
 `#apariciones(s, x)=1}`

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

`problema ordenadaCrecientemente (l: $seq\langle\mathbb{Z}\rangle$) : Bool {`
 `requiere: {para todo elemento x de s , se cumple`
 `#apariciones(s, x)=1}`
 `asegura OpcionA: {todo elemento de l es más chico que`
 `cualquier otro elemento más a la derecha}`

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

`problema ordenadaCrecientemente (l: $seq\langle\mathbb{Z}\rangle$) : Bool {`
 `requiere: {para todo elemento x de s , se cumple`
 `#apariciones(s, x)=1}`
 `asegura OpcionA: {todo elemento de l es más chico que`
 `cualquier otro elemento más a la derecha}`
 `asegura OpcionB: {todos los elementos de l cumplen que no`
 `tienen otro elemento más chico que esté a la derecha}`

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

`problema ordenadaCrecientemente (l: $seq\langle\mathbb{Z}\rangle$) : Bool {`
 `requiere: {para todo elemento x de s , se cumple`
 `#apariciones(s, x)=1}`
 `asegura OpcionA: {todo elemento de l es más chico que`
 `cualquier otro elemento más a la derecha}`
 `asegura OpcionB: {todos los elementos de l cumplen que no`
 `tienen otro elemento más chico que esté a la derecha}`
 `asegura OpcionC: {para cualquier par de elementos a y b en l ,`
 `si $a < b$ entonces la posición de b es mayor a la de a }`

Ejercicio 2

- e) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

```
problema ordenadaCrecientemente (l: seq<ℤ>) : Bool {  
  requiere: {para todo elemento x de s, se cumple  
    #apariciones(s,x)=1}  
  asegura OpcionA: {todo elemento de l es más chico que  
    cualquier otro elemento más a la derecha}  
  asegura OpcionB: {todos los elementos de l cumplen que no  
    tienen otro elemento más chico que esté a la derecha}  
  asegura OpcionC: {para cualquier par de elementos a y b en l,  
    si  $a < b$  entonces la posición de b es mayor a la de a}  
  asegura OpcionD: {para cualquier par de elementos a y b en l,  
    si vale  $\text{buscar}(a, l) < \text{buscar}(b, l)$  entonces  $a < b$ }  
}
```