

Introducción a la programación

Repaso

Especificación

Completar las siguientes especificaciones:

problema a ($b: seq\langle \mathbb{Z} \rangle, c: \mathbb{Z} \rangle : \mathbb{Z} \{$
 requiere $x: \{c \in b\}$
 asegura $y: \{(\exists i: \mathbb{Z})(0 \leq i < |b| \wedge b[i] = c \wedge resultado = i)\}$
}

Especificación

Completar las siguientes especificaciones:

problema a ($b: seq\langle \mathbb{Z} \rangle, c: \mathbb{Z} \rangle : \mathbb{Z} \{$
 requiere $x: \{c \in b\}$
 asegura $y: \{(\exists i: \mathbb{Z})(0 \leq i < |b| \wedge b[i] = c \wedge resultado = i)\}$
}

Primero miremos las expresiones y tratemos de escribirlas en lenguaje natural

Especificación

problema a ($b: seq\langle \mathbb{Z} \rangle, c: \mathbb{Z} \rangle : \mathbb{Z} \{$
 requiere $x: \{c \in b \leadsto c \text{ pertenece a } b\}$
 asegura $y: \{(\exists i: \mathbb{Z})(0 \leq i < |b| \wedge b[i] = c \wedge resultado = i) \leadsto$
 resultado es una posición de b donde está el elemento $c\}$
 $\}$

Especificación

problema a ($b: seq\langle \mathbb{Z} \rangle, c: \mathbb{Z} \rangle : \mathbb{Z} \{$
 requiere $x: \{c \in b \leadsto c \text{ pertenece a } b\}$
 asegura $y: \{(\exists i: \mathbb{Z})(0 \leq i < |b| \wedge b[i] = c \wedge resultado = i) \leadsto$
 resultado es una posición de b donde está el elemento $c\}$
 $\}$

Ahora podemos ponerle nombres a los requiere y asegura

Especificación

```
problema a (b: seq<ℤ>, c: ℤ) : ℤ {  
  requiere pertenece: {c ∈ b}  
  asegura estaElementoEnPosicion:  
    {(∃ i : ℤ)(0 ≤ i < |b| ∧ b[i] = c ∧ resultado = i)}  
}
```

Especificación

problema a ($b: seq\langle \mathbb{Z} \rangle, c: \mathbb{Z}$) : \mathbb{Z} {
 requiere pertenece: $\{c \in b\}$
 asegura estaElementoEnPosicion:
 $\{(\exists i : \mathbb{Z})(0 \leq i < |b| \wedge b[i] = c \wedge resultado = i)\}$
}

Y ahora podemos ponerle nombre al problema

Especificación

```
problema buscarPosicion (b: seq<ℤ>, c: ℤ) : ℤ {  
  requiere pertenece: {c ∈ b}  
  asegura estaElementoEnPosicion:  
    {(∃i : ℤ)(0 ≤ i < |b| ∧ b[i] = c ∧ resultado = i)}  
}
```


Especificación

```
problema buscarPosicion (b: seq<ℤ>, c: ℤ) : ℤ {  
  requiere pertenece: {c ∈ b}  
  asegura estaElementoEnPosicion:  
    {(∃i : ℤ)(0 ≤ i < |b| ∧ b[i] = c ∧ resultado = i)}  
}
```

Y finalmente reemplazamos las variables en los predicados

Especificación

Finalmente nos queda:

```
problema buscarPosicion ( $l$ :  $seq\langle\mathbb{Z}\rangle$ , elemento:  $\mathbb{Z}$ ) :  $\mathbb{Z}$  {  
  requiere pertenece:  $\{elemento \in l\}$   
  asegura estaElementoEnPosicion:  
     $\{(\exists i : \mathbb{Z})(0 \leq i < |l| \wedge l[i] = elemento \wedge resultado = i)\}$   
}
```

Especificación (más difícil)

```
problema a ( $b: seq\langle \mathbb{Z} \rangle$ ) :  $seq\langle \mathbb{Z} \rangle$  {  
  requiere: {True}  
  asegura w:  $\{|resultado| \leq |b|\}$   
  asegura x:  $\{(\forall i : \mathbb{Z})(0 \leq i < |resultado| \rightarrow$   
     $(\exists j : \mathbb{Z})(0 \leq j \leq |b| \wedge resultado[i] = b[j]))\}$   
  asegura y:  $\{(\forall i : \mathbb{Z})(0 \leq i < |b| \rightarrow$   
     $(\exists j : \mathbb{Z})(0 \leq j \leq |resultado| \wedge b[i] = resultado[j]))\}$   
  asegura z:  $\{(\forall i, j : \mathbb{Z})((0 \leq i < |resultado| \wedge$   
     $resultado[i] = resultado[j]) \rightarrow i = j)\}$   
}
```

Especificación (más difícil)

problema a ($b: seq\langle \mathbb{Z} \rangle$) : $seq\langle \mathbb{Z} \rangle$ {
 requiere: {True}
 asegura w: $\{|resultado| \leq |b| \leadsto$
 la long de resultado es a lo sumo la de la lista original}
 asegura x: $\{(\forall i : \mathbb{Z})(0 \leq i < |resultado| \rightarrow$
 $(\exists j : \mathbb{Z})(0 \leq j \leq |b| \wedge resultado[i] = b[j])) \leadsto$
 todos los elementos de resultado están en b}
 asegura y: $\{(\forall i : \mathbb{Z})(0 \leq i < |b| \rightarrow$
 $(\exists j : \mathbb{Z})(0 \leq j \leq |resultado| \wedge b[i] = resultado[j])) \leadsto$
 todos los elementos de b están en resultado}
 asegura z: $\{(\forall i, j : \mathbb{Z})((0 \leq i < |resultado| \wedge$
 $resultado[i] = resultado[j]) \rightarrow i = j) \leadsto$
 resultado no tiene elementos repetidos}
}

Especificación (más difícil)

```
problema a (b: seq<ℤ>) : seq<ℤ> {  
  requiere: {True}  
  
  asegura tieneLongMenorOIgual: {|resultado| ≤ |b|}  
  asegura estanTodosEnB: {(∀i : ℤ)(0 ≤ i < |resultado| →  
    (∃j : ℤ)(0 ≤ j ≤ |b| ∧ resultado[i] = b[j]))}  
  
  asegura estanTodosEnResultado: {(∀i : ℤ)(0 ≤ i < |b| →  
    (∃j : ℤ)(0 ≤ j ≤ |resultado| ∧ b[i] = resultado[j]))}  
  
  asegura sinRepetidos: {(∀i, j : ℤ)((0 ≤ i < |resultado| ∧  
    resultado[i] = resultado[j]) → i = j)}  
}
```

Especificación (más difícil)

```
problema eliminarRepetidos (l: seq<ℤ>) : seq<ℤ> {  
  requiere: {True}  
  
  asegura tieneLongMenorOIgual: {|resultado| ≤ |l|}  
  asegura estanTodosEnL: {(∀i : ℤ)(0 ≤ i < |resultado| →  
    (∃j : ℤ)(0 ≤ j ≤ |l| ∧ resultado[i] = l[j]))}  
  
  asegura estanTodosEnResultado: {(∀i : ℤ)(0 ≤ i < |l| →  
    (∃j : ℤ)(0 ≤ j ≤ |resultado| ∧ l[i] = resultado[j]))}  
  
  asegura sinRepetidos: {(∀i, j : ℤ)((0 ≤ i < |resultado| ∧  
    resultado[i] = resultado[j]) → i = j)}  
}
```