# Algoritmos y Estrucutras de Datos

# Practica 2

## 2.1. Funciones Auxiliares

### Ejercicio 1

a) *pred* raizCuadrada $(x : \mathbb{Z})\{$

$(\exists c : \mathbb{Z})(c > 0 \ \wedge \ (c * c = x))$

$\}$

b) *pred* esPrimo $(x : \mathbb{Z})\{$

$(\forall n : \mathbb{Z})((1 < n < x) \rightarrow_L (x \mod n \ \neq 0))$

$\}$

### Ejercicio 2

a) *pred* sonCoprimos $(x, y : \mathbb{Z})\{$

$((\forall n : \mathbb{Z})(((n > 1) \wedge_L (x \mod n = 0)) \rightarrow_L (y \mod n \neq 0))) \wedge_L$

$((\forall m : \mathbb{Z})(((m > 1) \wedge_L (x \mod m = 0)) \rightarrow_L (y \mod m \neq 0)))$

$\}$

b) *pred* mayorPrimoQueDivide $(x, y : \mathbb{Z})\{$

$(esPrimo(y) \wedge_L (x \mod y = 0) \wedge (\forall c : \mathbb{Z})(((esPrimo(c)) \wedge_L (x \mod c = 0)) \rightarrow_L (c \leq y))$

$\}$

### Ejercicio 3

a) *pred* todosElementosPositivos $(s : seq\langle Z\rangle)\{$

$(\forall i : \mathbb{Z})((0 \leq i < |s|) \rightarrow_L \; s[i] \geq 0)$

}

b) *pred* todosDistintos $(s : seq\langle Z\rangle)\{$

$(\forall i : \mathbb{Z})((0 \leq i < |s|) \rightarrow_L (\forall j : \mathbb{Z})((0 \leq j < |s| \land i \neq j) \rightarrow_L (s[i] \neq s[j])))$

}

# Ejercicio 4

a) *pred* esPrefijo $(s, t : seq\langle Z\rangle)\{$
$(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L (s[i] = t[i]))$
}

Falta aclarar que $|s| \leq |t|$

b) *pred* estaOrdenada $(s : seq\langle Z\rangle)\{$

$(\forall i, j : \mathbb{Z})(i \leq j \rightarrow_L (s[i] \leq s[j]))$

}

c) *pred* hayUnoParQueDivideAlResto $(s : seq\langle Z\rangle)\{$

$(\exists i : \mathbb{Z})((0 \leq i < |s|) \land_L (s[i] \mod 2 = 0) \land_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L (s[j] \mod s[i] = 0)))$

}

d) *pred* igualesEnElRango $(t : seq\langle z\rangle, i, j, n : \mathbb{Z})\{$

$(\forall m : \mathbb{Z})(i \leq m \leq j \rightarrow_L (s[m] = n))$

*pred* enTresPartes $(s : seq\langle Z\rangle)\{$

$(\exists i, j : \mathbb{Z})(0 < i < j < |s| - 1) \land_L$

$igualesEnELRango(s, 0, i, 0) \land_L$

$igualesEnELRango(s, i + 1, j, 1) \land_L$

$igualesEnELRango(s, j + 1, |s| - 1, 2)$

}

**Extra:** Cambiaria los $\land_L$ por $\lor_L$

# Ejercicio 5

a) cantApariciones $(s : seq\langle\mathbb{Z}\rangle, e : \mathbb{Z}) = \sum_{i=0}^{|s|-1} IfThenElseFi(s[i] = e, 1, 0)$

b) *pred* esPar $(x : \mathbb{Z})\{(x \mod 2 = 0)\}$

posImp $(s : seq\langle\mathbb{Z}\rangle) = \sum_{i=0}^{|s|-1} IfThenElseFi(\neg esPar(i), s[i], 0)$

c) positivos $(s : seq\langle\mathbb{Z}\rangle) = \sum_{i=0}^{|s|-1} IfThenElseFi(s[i] > 0, s[i], 0)$

d) positivos $(s : seq\langle\mathbb{Z}\rangle) = \sum_{i=0}^{|s|-1} IfThenElseFi(s[i] \neq 0, \frac{1}{s[i]}, 0)$

## 2.2. Analisis de especificacion

### Ejercicio 6

a) No es correcta debido a que si se toma el caso i=0 se cumple la precondicion del implica $(0 \leq i < |l|)$ pero la postcondicion es falsa ya que $l[i-1] = i[0-1] = i[-1] = \perp$. Para que no se indetermine propongo la siguiente especificiacion:

*proc* progresionGeometricaFactor2 $(in\ l : seq\langle\mathbb{Z}\rangle) : Bool$

requiere {True}

asegura $\{res = True \leftrightarrow ((\forall i : \mathbb{Z})(0 \leq i < |l|-1 \rightarrow_L l[i+1] = 2*l[i]))\}$

b) No es correcta dado que no hay relacion entre antecedente y consecuente y la lista deberia tener al menos un elemento. Propongo:

$proc$ minimo $(in\ l : seq\langle\mathbb{Z}\rangle) : \mathbb{Z}$

requiere $\{|l| > 0\}$

asegura $\{res = y \leftrightarrow (\exists y : \mathbb{Z})(y \in l \wedge (\forall x : \mathbb{Z})(x \in l \rightarrow_L y \leq x))\}$

## Ejercicio 7

a) I) $l = \langle 1, 2, 3, 4\rangle \implies indiceDelMaximo(l) = 3$

II) $l = \langle 15.5, -18, 4.215, 15.5, -1\rangle \implies indiceDelMaximo(l) = 0 \vee indiceDelMaximo(l) = 3$

III) $l = \langle 0, 0, 0, 0, 0, 0\rangle \implies indiceDelMaximo(l) = 0 \vee 1 \vee 2 \vee 3 \vee 4 \vee 5$

b) I) $l = \langle 1, 2, 3, 4\rangle \implies indiceDelPrimerMaximo(l) = 3$

II) $l = \langle 15.5, -18, 4.215, 15.5, -1\rangle \implies indiceDelPrimerMaximo(l) = 0$

III) $l = \langle 0, 0, 0, 0, 0, 0\rangle \implies indiceDelPrimerMaximo(l) = 0$

c) indiceDelMaximo e indiceDelPrimerMaximo tienen necesariamente la misma salida para las listas que tengan un unico maximo, ya que en caso contrario la primer funcion podria tener mas de una respuesta posible

### Ejercicio 8

a) La especificacion no es correcta ya que es contradictorio que la port-condicion sea correcta cuando $a < 0$ y cuando $a \geq 0$

b) Correcta

c) La especificacion no es correcta ya que por tabla de verdad de la implicacion, podria darse que el antecedente es falso y el consecuente correcto. Esto daria lugar a absurdos tales como que si parto de a=1 ($a \geq 0$), b=2 entonces serian validos $f(a, b) = 1$ y $f(a, b) = 4$ al mismo tiempo, cosa que es absurda.

d) Correcta

### Ejercicio 9

a) unoMasGrande(3)=9 $\therefore$ se cumple la postcondicion

b) unoMasGrande(0.5)=0.25 pero 0.25 $\not> 0.5$
unoMasGrande(1)=1 pero 1 $\not> 1$
unoMasGrande(-0.2)=0.4 y 0.4 $> -0.2$
unoMasGrande(-7)=49 y 49 $> -7$

c) requiere $\{x < 0 \ \vee x > 1\}$

## 2.3. Relacion de fuerza

## Ejercicio 10

a) $P1 : \{x \leq 0\}$ $P2 : \{x \leq 10\}$ $P3 : \{x \leq -10\}$

$P1 \rightarrow P2$ **True** $\forall x$ por lo tanto $P1 > P2$

$P2 \rightarrow P1$ **False** por ejemplo con x=2 se cumple P2 pero no P1 por lo tanto $P2 \not> P1$

$P1 \rightarrow P3$ **False** por ejemplo con x=-2 se cumple P1 pero no P3 por lo tanto $P1 \not> P3$

$P3 \rightarrow P1$ **True** $\forall x$ por lo tanto $P3 > P1$

$P3 \rightarrow P2$ **True** $\forall x$ por lo tanto $P3 > P2$

$P2 \rightarrow P3$ **False** por ejemplo con x=-2 se cumple P2 pero no P3 por lo tanto $P1 \not> P3$

b) $Q1 : \{r \geq x^2\}$ $Q2 : \{r \geq 0\}$ $Q3 : \{r = x^2\}$

$Q1 \rightarrow Q2$ **True** ya que $x^2 \geq 0$ $\forall x$ por lo tanto $Q1 > Q2$

$Q2 \rightarrow Q1$ **False** depende del valor de $x^2$, por lo tanto no vale siempre la implicacion $Q1 \rightarrow Q3$ **False** la primer condicion permite que $r > x^2$ lo cual implicaria $r \neq x^2$ y $True \rightarrow False = False$ ; por lo tanto $P1 \not> P3$

$Q3 \rightarrow Q1$ **True** $\forall r, x$ por lo tanto $P3 > P1$

$Q3 \rightarrow Q2$ **True** $\forall r, x$ por lo tanto $P3 > P2$

$Q2 \rightarrow Q3$ **False** depende del valor de $x^2$, por lo tanto no vale siempre la implicacion

d) i) Cumple ambas condiciones

   ii) No cumple la **precondicion**

   iii) Cumple ambas condiciones

   iv) No cumple la **postcondicion**

   v) Cumple ambas condiciones

   vi) No asegura la **ninguna**

e) Las precondiciones y postcondiciones deben ser **mas fuertes** que las anteriores para poder reemplazar una especificacion de manera segura
*Esta mal el e), la postcondicion debe ser mas debil*

## Ejercicio 11

a) Como el requiere de p1 se cumple entonces vale que $x \neq 0$. Luego si $n \leq 0$ entonces ($n \leq 0 \ \rightarrow x \neq 0$)=True pues $True \rightarrow True = True$. En cambio si $n > 0$ entonces tambien ($n \leq 0 \ \rightarrow x \neq 0$)=True pues $False \rightarrow True = True$

b) $\lfloor x^n \rfloor$ implica que siendo a y b los enteros mas cercanos $x^n$ entonces $\lfloor x^n \rfloor = \ a$, es decir el 'piso', por lo tanto sera como mucho 0.999... menor a $x^n$. Esto implica que $x^n - 1 < \lfloor x^n \rfloor < x^n$ y por lo tanto el resultado cumple tanto p1 como p2.

c) $a$ no satisface p1 dado que por ejemplo podria satisfacer la precondicion de p2 con un $n > 0$ y un $x = 0$ ya que $False \rightarrow False \ = \ True$

pero no la de p1. Sin embargo como se vio en b) si satisface la postcondicion

## 2.4. Especificacion de problemas

### Ejercicio 12

a) $proc$ esPar $(in\ x : \mathbb{Z}) : Bool\{$

$requiere\{True\}$

$asegura\{res = True \leftrightarrow (x \mod 2) = 0\}$

b) $proc$ esMult $(in\ n, m : \mathbb{Z}) : Bool\{$

$requiere\{True\}$

$asegura\{res = True \leftrightarrow (\exists k : \mathbb{Z})(n * k = m)\}$

c) $proc$ listDiv $(n : \mathbb{Z}) : seq\langle\mathbb{Z}\rangle\{$

$requiere\{True\}$

$asegura\{(\forall i : \mathbb{Z})(0 \leq i < |res| \rightarrow_L ((n \mod res[i] = 0) \wedge res[i] > 0))\}$

d) $proc$ descomPrimos $(in\ x : \mathbb{Z}) : seq\langle(\mathbb{Z}, \mathbb{Z})\rangle\{$

$requiere\{x > 0\}$

$asegura\{(p, e) \in res \rightarrow_L (x \mod p\ = 0) \wedge_L esPrimo(p) \wedge_L p * e = x\}$

$asegura\{(\forall i, p : \mathbb{Z})(0 \leq i, p < |s| \wedge i \neq p \rightarrow_L res[i] \neq res[p])\}$

$asegura\{(\forall i, p : \mathbb{Z})(0 \leq i < p < |s| \rightarrow_L res[i][0] < res[p][0])\}$

# Ejercicio 13

a) $proc$ contenidoEn $(s, t : seq\langle\mathbb{Z}\rangle) : Bool\{$

$requiere\{True\}$

$asegura\{(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L (\exists j : \mathbb{Z})(0 \leq j < |t| \wedge_L s[i] = t[j]))\}$

$\}$

b) $proc$ interseccion $(s, t : seq\langle\mathbb{Z}\rangle) : seq\langle\mathbb{Z}\rangle\{$

$requiere\{True\}$

$asegura\{(\forall e : \mathbb{Z})((e \in s \wedge e \in t) \rightarrow_L$

$\#apariciones(e, res) = min(\#apariciones(e, s), \#apariciones(e, t)))\}$

$\}$

$aux$ #apariciones $(e : \mathbb{Z}, s : seq\langle\mathbb{Z}\rangle) = \sum_{i=0}^{|s|-1} IfThenElseFi(s[i] = e, 1, 0)$

c) $aux$ divideNElementos $(s : seq\langle\mathbb{Z}\rangle, n : \mathbb{Z}) = \sum_{i=0}^{|s|-1} IfThenElse(s[i] \mod n = \ 0, 1, 0)$

$proc$ divideMasElementos $(s : seq\langle\mathbb{Z}\rangle) : \mathbb{Z}\{$

$requiere\{True\}$

$asegura\{res \in s \wedge (\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L divideNElementos(s, res) \geq divideNElementos(s, s[i])\}$

d) $aux$ maxValor $(s : seq\langle\mathbb{Z}\rangle) : \mathbb{Z} = res \in s \wedge (\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L (res \geq s[i]))$

$proc$ seqMaxValor $(l : seq\langle seq\langle\mathbb{Z}\rangle\rangle) : seq\langle\mathbb{Z}\rangle\{$

$requiere\{True\}$

$asegura\{res \in l \wedge (\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L maxValor(res) \geq maxValor(l[i]))\}$

e) $pred$ noHayRepes $(s : seq\langle Z\rangle)\{$

$(\forall i : \mathbb{Z})((0 \leq i < |s|) \rightarrow_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L (s[i] \neq s[j])))$

$\}$

$aux$ cantSeqNelem $(t : seq\langle seq\langle\mathbb{Z}\rangle\rangle, n : \mathbb{Z}) : \mathbb{Z} = \sum_{i=0}^{|s|-1} IfThenElse(|s[i]| = n, 1, 0)$

$pred$ respetaOrden $(s,t : seq\langle Z\rangle)\{(\forall i,j:\mathbb{Z})(0 \leq i < j < |t| \rightarrow_L ((t[i] \in s \wedge_L t[j] \in s) \rightarrow_L \neg(\exists n,m:\mathbb{Z})(0 \leq n < m < |s| \rightarrow_L s[n] = s[i] \wedge_L s[m] = t[i]))\}$

$proc$ partes $(s : seq\langle \mathbb{Z}\rangle) : seq\langle seq\langle \mathbb{Z}\rangle\rangle\{$

$requiere\{True\}$

$asegura\{|res| = 2^{|s|} \wedge_L nohayRepes(res) \wedge_L (\forall i:\mathbb{Z})(0 \leq i \leq |s| \rightarrow_L cantSeqNelem(res,i) = \binom{|s|}{i} \wedge_L (\forall j:\mathbb{Z})(0 \leq j < |res| \rightarrow_L respetaOrden(res[i],s))\}$

**Nota**: luego de ver rtas agregue la cond. respeta orden

## 2.5. Especificacion de problemas usando *inout*

### Ejercicio 14

a) La especificacion esta mal dado que las variables $a$ y $b$ deben ser de tipo in, ya que no es necesario que se modifiquen para calcular su suma, solo se accede a su valor.

**Nota**: Falto mencionar estados previos, etc.

b) Correcta

c) Correcta. Cabe aclarar que no es necesario que $a$ y $b$ sean de tipo *inout*. Sin embargo al asegurar que no se modifica su valor gracias a la pre y postcondicion, no genera problemas

## Ejercicio 15

a) *Incorrecta.* No menciona los cambios de estado ni tampoco saca el primer elemento de l por lo tanto no cumple lo pedido.

b) *Incorrecta.* Habla de cambios de estado pero no saca el primer elemento de l, solo lo devuelve, por lo tanto no cumple lo pedido.

c) *Incorrecta.* En la precondicion no se menciona el estado de origen de l ($L_0$) y en la postcondicion no se elimina al primer elemento de l, solo se asegura que su largo es una unidad menor, por lo tanto, no cumple lo pedido.

d) *Correcta.* Respeta los cambios de estado y devuelve el primer elemento de l y a la lista l sin su cabeza.

## Ejercicio 16

a) La especificacion es incorrecta porque en una implicacion si el antedente es falso y el consecuente es verdadero, el predicado es verdadero. De esta manera si un numero esta en el rango de la secuencia, no es par y se duplica el valor de esa posicion en la lista, tendriamos un antecedente falso y un consecuente verdadero, por lo tanto la afirmacion seria verdadera. Sin embargo estariamos duplicando un elemento en una posicion impar, violando el enunciado.

b) La especificacion es incorrecta por la misma razon que en el punto a), para ambos disyuntos se puede plantear un contraejemplo que haga correcta la especificacion pero no cumpla la consigna.

**Nota**: Falto poner que no se asegura que se mantenga el tamaño de la secuencia

c) La especificacion no es correcta dado que no considera los cambios de estado de la secuencia.

- Propongo la siguiente especificacion:

$proc$ duplicarPares $(inout\ s : seq\langle\mathbb{Z}\rangle) : seq\langle\mathbb{Z}\rangle\{$

$requiere\{l = L_0\}$

$asegura\{|l| = |L_0|\}$

$asegura\{(\forall i : \mathbb{Z})((0 \leq i < |s| \wedge i \mod 2 = 0) \rightarrow_L l[i] = 2 * L_0)\}$

$asegura\{(\forall i : \mathbb{Z})((0 \leq i < |s| \wedge i \mod 2 \neq 0) \rightarrow_L l[i] = L_0)\}$

# Ejercicio 17

a) $proc$ primosHermanos $(inout\ l : seq\langle\mathbb{Z}\rangle)$

requiere$\{l = L_0\}$

asegura$\{(\forall i : \mathbb{Z})(0 \leq i < |L_0| \rightarrow_L$

$(l[i] = res[i] \wedge_L esPrimo(res[i]) \wedge_L$

$(\forall j : \mathbb{Z})((esPrimo(j) \wedge_L j \neq res[i]) \rightarrow_L dist(res[i], l[i]) \leq dist(j, l[i]) \wedge_L$

$res[i] < j))\}$

$aux$ dist $(a, b : \mathbb{Z}) : \mathbb{Z} = |a - b|$

b) $proc$ reemplazar $(inout\ l : seq\langle Char\rangle, in\ a, b : Char) : seq\langle Char\rangle\{$

requiere $\{l = L_0\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |L_0| \rightarrow_L (L_0[i] =' a' \wedge l[i] =' b'))\}$

asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |L_0| \rightarrow_L (L_0[i] \neq' a' \wedge l[i] = L_0[i]))\}$

asegura$\{|L_0| = |l|\}$

**Nota**: los '$\rightarrow_L$' estaban mal y los cambie por los '$\wedge$'

c) *proc* limpiarDuplicados $(s : seq\langle Char\rangle) : seq\langle Char\rangle$

requiere$\{l = L_0\}$

asegura$\{noHayRepes(l) \wedge\ (|l| \leq |L_0|) \wedge\ (|l| + |res| = |L_0|) \wedge\ (\forall i : \mathbb{Z})(0 \leq i < |l| \rightarrow_L \#apariciones(l[i], res) = \#apariciones(l[i], L_0) - 1\}$

**Nota**:Falto asegurar que contengan los mismos elementos (sin contar repes)

## 2.6. Ejercicios de parcial

### Ejercicio 18

a) *aux* sumaDivisores $(n : \mathbb{Z}) : \mathbb{Z} = \sum_{i=1}^{n-1} ifThenElse((n \mod i) = 0, i, 0)$

*proc* reemplazarNumerosPerfectos $(inout\ s : seq\langle \mathbb{Z}\rangle)$

requiere$\{s = S_0\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |S_0| \wedge_L sumaDivisores(S_0[i]) = S_0[i]\ \wedge s \neq 0) \rightarrow_L s[i] = i)\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |S_0| \land_L sumaDivisores(S_0[i]) \neq S_0[i]) \rightarrow_L s[i] = S_0[i])\}$

asegura$\{|s| = |S_0|\}$

b) *pred* esMayorATodos $(in\ s : seq\langle\mathbb{Z}\rangle, n : \mathbb{Z})\{(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L s[i] \leq n)\}$

*pred* estaOrdenada $(in\ s : seq\langle\mathbb{Z}\rangle)\{(\forall i : \mathbb{Z})(0 < i < |s| \rightarrow_L |s[i-1]| \leq |s[i]|)\}$

*proc* ordenarYBuscarMayor $(inout\ s : seq\langle\mathbb{Z}\rangle) : \mathbb{Z}$

requiere$\{s = S_0\}$

asegura$\{estaOrdenada(s)\}$

asegura$\{res \in S_0 \land esMayorATodos(S_0, res)\}$

asegura$\{|s| = |S_0|\}$

**Nota**: Falto asegurar que los elementos de s y $S_0$ sean los mismos

c) *pred* esPrimo $(n : \mathbb{Z})\{2 = \sum_{i=1}^{n} ifThenElse(n \mod i = 0, 1, 0)\}$

*proc* primosEnCero $(inout\ s : seq\langle\mathbb{Z}\rangle)$

requiere$\{s = S_0\}$

asegura$\{|s| = |S_0|\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |s| \land_L esPrimo(i) \rightarrow_L s[i] = 0)\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |s| \land_L \neg esPrimo(i) \rightarrow_L s[i] = S_0[i])\}$

d) *proc* positivosAumentados $(inout\ s : seq\langle\mathbb{Z}\rangle)$

requiere$\{s = S_0\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |s| \land_L S_0[i] \geq 0) \rightarrow_L s[i] = S_0[i] * i)\}$

asegura$\{(\forall i : \mathbb{Z})((0 \leq i < |s| \land_L S_0[i] < 0) \rightarrow_L s[i] = S_0[i])\}$

asegura$\{|s| = |S_0|\}$

e) *pred* palabraMasLarga $(s : seq\langle string\rangle, pal : string)\{(\forall i : \mathbb{Z})((0 \leq i < |s| \land_L pal \in s) \rightarrow_L |pal| \geq |s[i]|)\}$

*pred* esPrefijo $(pal, pre : string)\{(\forall i : \mathbb{Z})(0 \leq i < |pre| \rightarrow_L pre[i] = pal[i])\}$

$proc$ procesarPrefijos $(inout\ s : seq\langle string\rangle, in\ p : string) : \mathbb{Z}$

requiere$\{s = S_0\}$

asegura$\{(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L (s[i] \in S_0 \wedge esPrefijo(s[i], p)))\}$

asegura$\{(\exists j : \mathbb{Z})(0 \leq j < |s| \wedge_L res = |s[j]| \wedge_L palabraMasLarga(s, s[j]))\}$