

Precondición más débil en SmallLang

Ejercicio 1. Calcular las siguientes expresiones, donde a, b son variables reales, i una variable entera y A es una secuencia de reales.

- a) $\text{def}(a + 1).$
- b) $\text{def}(a/b).$
- c) $\text{def}(\sqrt{a/b}).$
- d) $\text{def}(A[i] + 1).$
- e) $\text{def}(A[i + 2]).$
- f) $\text{def}(0 \leq i \leq |A| \wedge_L A[i] \geq 0).$

a) $\text{def}(a) \wedge \text{def}(1)$

$$\text{True} \wedge \text{True} = \text{True}$$

b) $\text{def}(a/b)$

$$\text{def}(a) \wedge (\text{def}(b) \wedge b \neq 0)$$

c) $\text{def}(\sqrt{a/b})$ ¿Qué operación primero, Div?

$$\text{def}(a) \wedge (\text{def}(b) \wedge b \neq 0) \wedge a/b \geq 0$$

d) $\text{def}(A[i]) + \text{def}(1)$

$$(\text{def}(A) \wedge \text{def}(i) \wedge 0 \leq i < |A|) \wedge \text{def}(1)$$

e) $(\text{def}(A) \wedge \text{def}(i) \wedge 2 \leq i < |A|)$

f) ¿Pueden los def ser nulos?

¿Cómo los terminar? Prefiero 6M (

Ejercicio 2. Calcular las siguientes precondiciones más débiles, donde a, b son variables reales, i una variable entera y A es una secuencia de reales.

- a) $wp(a := a+1; b := a/2, b \geq 0).$
 b) $wp(a := A[i] + 1; b := a*a, b \neq 2).$
 c) $wp(a := A[i] + 1; a := b*b, a \geq 0).$
 d) $wp(a := a-b; b := a+b, a \geq 0 \wedge b \geq 0).$

2) Al tener m醩 de un S usaremos AXIOMA 3.

$$a) \underbrace{WP(a := Q+1; b := a/2, b \geq 0)}_{WP(S_1; WP(S_2, Q))}$$

$$WP(S_1; WP(S_2, Q))$$

$$WP(S_2, Q) = \text{def}(a/2) \wedge_L Q_{a/2}^b$$

$$\begin{aligned} &= \text{True} \wedge_L a/2 > 0 \\ &= a > 0 \quad \exists E1 \end{aligned}$$

$$WP(S_1, E1) = WP(a := a+1, a \geq 0)$$

$$= \text{def}(a+1) \wedge_L Q_{a+1}^a$$

$$= \text{True} \wedge_L a+1 > 0$$

$$= a > -1 \rightarrow WP$$

VERIFI:

$$a = -1 \Rightarrow P$$

$$\{a = -1\} \models \{A_0 = -1\}$$

$$a = A_0 + 1$$

$$\{a = 0\} \models \{A_1 = 0\}$$

$$b = A_1/2$$

$$\{b = 0\} \models \{B_0 = 0\}$$

$$b > 0 \checkmark \Rightarrow Q$$

$$b) \text{wp}(S_1; \text{wp}(S_2, Q))$$

$$\text{wp}(S_2, Q) = \text{def}(a * a) \wedge_L Q_{a * a}^b$$

$$= \text{True} \wedge_L a * a \neq 2$$

* $\exists a * a \neq 2 \in E_1$

$\rightarrow A[i]$ la Pifid Cev. Nels heren op? Erste 2?

$$\text{wp}(S_1, E_1) = \text{def}(A[i] + 1) \wedge_L Q_{A[i] + 1}^a$$

$$= \text{def}(A[i]) \wedge \text{def}(1) \wedge_L A[i] + 1 * A[i] + 1 \neq 2$$

$$= \text{def}(i) \wedge \text{def}(A) \wedge_{0 \leq i < |A|} A[i] * A[i] \neq 0$$

$$= A[i] * A[i] \neq 0$$

i für multiplikation alle nur elementen 0 oder 1?

todo: Come mit?

$$c) \text{wp}(S_1; \text{wp}(S_2, Q))$$

$$\text{wp}(S_2, Q) = \text{def}(b * b) \wedge_i Q_{b * b}^a$$

$$= \text{True} \wedge_i b * b \geq 0$$

≥ 0

* = True $\in E_1$

$$WP(S_1, E_1) = \text{def}(A[i] + 1) \wedge_Q Q_{A[i]+1}$$

$$= (\text{def}(A) \wedge_L \text{def}(i) \wedge_L 0 \leq i < |A|) \wedge \text{def}(1) \wedge_L A[i+1]$$

$$= A[i+1]$$

OPTMA FORM

$$* b * b \geq 0 \equiv E 1$$

$$WP(S_1, E_1) = \text{def}(A[i] + 1) \wedge_L Q_{A[i]+1}^b$$

$$= (\text{def}(A) \wedge \text{def}(i) \wedge_L 0 \leq i < |A| \wedge \text{def}(1) \wedge_L (A[i] + 1) \geq 0)$$

$$= (A[i] + 1)^2 \geq 0$$

$$= \text{true} ?$$

$$\text{d)} WP(S_2, Q) = \text{def}(a+b) \wedge_L Q_{a+b}^b$$

$$= \text{true} \wedge_L a > 0 \wedge a+b \geq 0$$

$$= a > 0 \wedge a+b \geq 0 \equiv E 1$$

$$WP(S_1, E_1) = \text{def}(a-b) \wedge_L Q_{a>0 \wedge a+b \geq 0}^a$$

$$= \text{true} \wedge_L (a > 0 \wedge a+b \geq 0) - b$$

= j?

31900

4)

Ejercicio 4. Para los siguientes pares de programas S y postcondiciones Q

- Escribir la precondición más débil $P = wp(S, Q)$
- Mostrar formalmente que la P elegida es correcta

a) $S \equiv$

```
if( a < 0 )
    b := a
else
    b := -a
endif
```

$$Q \equiv (b = -|a|)$$

b) $S \equiv$

```
if( i > 0 )
    s[i] := 0
else
    s[0] := 0
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0)$$

c) $S \equiv$

```
if( i > 1 )
    s[i] := s[i-1]
else
    s[i] := 0
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L s[j] = s[j-1])$$

d) $S \equiv$

```
if( s[i] > 0 )
    s[i] := -s[i]
else
    skip
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0)$$

o) Al haber un if hay que usar el Axioma 5.

$$\begin{aligned} \text{si } a < 0 \Rightarrow b \leq 0 \\ 0 > 0 \Rightarrow b \leq 0 \end{aligned}$$

Para b ser siempre 0-positivo, y sin importar

que a sea negativo, b siempre será $-|a|$.

$$\text{Ej: } a = -2 \Rightarrow b = -2 \text{ pero } b = -|-2| = -2$$

$$a = 2 \Rightarrow b = -2 \text{ ferner } b = -|2| = -2.$$

$P = \text{True} \rightarrow \forall a$.

$$\text{def}(B) \wedge ((B \wedge w_P(s_1, Q)) \vee (\bar{B} \wedge w_P(s_2, Q)))$$

$$w_P(s_1, Q) = (b := a, b = -|a|)$$

$$= \text{def}(a) \wedge Q_a^b$$

$$= \text{True} \wedge a = -|a|$$

$$= a = -|a| \text{ (or } a < 0)$$

$$w_P(s_2, Q) = (b := -a, b = -|a|)$$

$$= \text{def}(-a) \wedge Q_{-a}^b$$

$$= \text{True} \wedge -a = -|a|$$

$$= a = |a| \text{ (or } a > 0)$$

$$\text{further, } \text{def}(a < 0) \wedge ((a < 0 \wedge a = -|a|) \vee (a > 0 \wedge a = |a|))$$

$$= \text{True} \wedge (a = -|a| \vee a = |a|)$$

$$= a = -|a| \vee a = |a| \Rightarrow \forall a$$

$$= \text{True}$$

¿Cómo manejar formalmente?

Refago

1) a) $\text{def}(a+1) = \text{def}(a) + \text{def}(1) = \text{True} \wedge \text{True} = \text{True}$ ✓

- ¿Cuál es el criterio para ir repartiendo en def ?
- ¿Por operación?

Ej.: $\text{def}\left(\sqrt{\frac{a}{b}} + \frac{c}{d}\right)$ viene considerar que los se obtienen sobre $a, b, c, d > 0$?

$$\text{def}(a) \wedge (\text{def}(b) \wedge b \neq 0) \wedge \text{def}(c) \wedge (\text{def}(d) \wedge d \neq 0) \\ \wedge ((a/b + c/d) > 0)$$

b) $\text{def}(a/b) = \text{def}(a) \wedge (\text{def}(b) \wedge b \neq 0)$

$$= \text{True} \wedge \text{True} \wedge b \neq 0$$

$$= b \neq 0$$
 ✓

c) $\text{def}(\sqrt{a/b}) = \text{def}(a) \wedge (\text{def}(b) \wedge b \neq 0)$

$$\wedge (a/b \geq 0)$$

$$= \text{True} \wedge \text{True} \wedge b \neq 0 \wedge a/b \geq 0$$

$$= b \neq 0 \wedge a/b > 0 \quad \checkmark$$

d) $\text{def}(a[i]+1) = (\text{def}(a) \wedge \text{def}(i)) \wedge 0 \leq i < |a| \wedge \text{def}(1)$

$$= \text{True} \wedge \text{True} \wedge 0 \leq i < |a| \wedge \text{True}$$
$$= 0 \leq i < |a| \quad \checkmark$$

e) $\text{def}(a[i+2]) = (\text{def}(a) \wedge \text{def}(i)) \wedge 0 \leq i < |a|$

$$= \text{True} \wedge \text{True} \wedge -2 \leq i < |a|-2$$

$$= -2 \leq i < |a|-2 \quad \checkmark$$

↓

MINVAL

PART A[0]

↓

MAXVAL

PART A.

f) $\text{def}(0 \leq i \leq |A| \wedge \underline{A[i] \geq 0})$

Liste der positiven
Werte der Positionen
i ist nur die index

$$= \text{def}(0 \leq i \leq |A|) \wedge \text{def}(A[i] \geq 0)$$

$$= i \neq |A| \text{ ?}$$

?) a) $w_{ab}(a:=a+1; b:=0/1; b>0)$

$$W_P(S_1, W_P(S_2, Q)) = \text{AXIOMA 3}$$

$$W_P(S_2, Q) = W_P(b := Q/2, b > 0)$$

PARA $a := Q/2$

$$= \text{def}(a/2) \wedge_Q^b Q_{a/2}$$

$$= \text{true} \wedge_Q^b Q/2 > 0$$

$$= a > 0 \models E_1 \checkmark$$

Luego, $W_P(S_1, E_1) = W_P(a := Q+1; Q > 0)$

$$= \text{def}(Q+1) \wedge_Q^a Q_{a+1}$$

$$= \text{true} \wedge_Q^a a+1 > 0$$

$$= a \geq -1 \checkmark$$

Por lo tanto,

$$W_P(S_1, E_1) \models (a \geq -1)$$

Verif x reciproca: $a < -1 \Rightarrow a = -2$.

$$a = -1; b = -\frac{1}{2}; b \neq 0$$

Luego, vale la W_P .

b) $W_P(a := a[i] + 1; b := a * a, b \neq 2)$

$$W_P(S_1, W_P(S_2, Q)) = AXIOMA \ 3$$

$$W_P(S_2, Q) = W_P(b := a * a; b \neq 2)$$

$$= \text{def}(a * a) \wedge_L Q_{a * a}^b$$

$$= \text{true} \wedge_L a * a \neq 2$$

$$= a^2 \neq 2$$

$$= |a| \neq \sqrt{2} \equiv E_1 \checkmark$$

$$\text{Luego, } W_P(S_1, E_1) = W_P(a : a[i] + 1; |a| \neq \sqrt{2})$$

$$= \text{def}(a[i] + 1) \wedge_L Q_{a[i] + 1}^a$$

$$= (\text{def}(a) \wedge \text{def}(i)) \wedge 0 \leq i < |a| \wedge \text{def}(1)$$

$$\wedge_L |a[i] + 1| \neq \sqrt{2}$$

$$= 0 \leq i < |a| \wedge_L |a[i]| + 1 \neq \sqrt{2}$$

$$= |a[i]| \wedge_L (0 \leq i < |a|) \neq \sqrt{2} - 1 \checkmark$$

¿Está bien la razonamiento?

$$\text{Por lo tanto, } W_P(S_1, E_1) = |a[i]| \wedge_L (0 \leq i < |a|) \neq \sqrt{2} - 1$$

$$\text{Verif x reciproca: } a = \sqrt{2} - 1 \Rightarrow a : \sqrt{2}; b : \sqrt{2} * \sqrt{2} - 2 \\ \not\Rightarrow b \neq 2$$

Luego, vale la W_P .

$$C) \text{wp}(a : a[i] + 1 ; a := b * b ; a \geq 0)$$

$$\text{wp}(S_1, \text{wp}(S_2, Q))$$

$$\text{wp}(S_2, Q) = \text{wp}(0 := b * b, a \geq 0)$$

$$= \text{def}(b * b) \wedge_L Q_{b * b}^a$$

$$= \text{True} \wedge_L b * b \geq 0$$

$$= b * b \geq 0 \in E_1 \quad \checkmark$$

$$\text{wp}(S_1, E_1) = \text{wp}(a : a[i] + 1 ; b * b \geq 0)$$

$$= \text{def}(a[i] + 1) \wedge_L Q_{a[i] + 1}^b$$

$$= \underbrace{a[i] + 1 + a[i] + 1}_{\text{*}} \geq 0$$

$$= \text{True} \quad \checkmark \quad *$$

* NEG + NEG = + ≥ 0

POS * POS = + ≥ 0

$$d) \text{wp}(a : a - b ; b := a + b ; a \geq 0 \wedge b \geq 0)$$

$$\text{wp}(S_1, \text{wp}(S_2, Q))$$

$$\text{wp}(S_2, Q) = \text{wp}(b := a + b ; a \geq 0 \wedge b \geq 0)$$

$$= \text{def}(a + b) \wedge_L Q_{a+b}^b$$

$$= \text{True} \wedge_L 0 \geq 0 \wedge 0 + b \geq 0$$

$$= 0 \geq 0 \wedge 0 + b \geq 0 \stackrel{E1}{=} \checkmark$$

$$Wp(S_1, E_1) = Wp(0 : a - b; 0 \geq 0 \wedge 0 + b \geq 0)$$

$$= \text{def}(a - b) \wedge_L Q_{a-b}^a$$

$$= \text{True} \wedge_L 0 - b \geq 0 \wedge (a - b) + b \geq 0$$

$$= a - b \geq 0 \wedge a \geq 0 \checkmark$$

VERIF X RECIP: $a = 6 \quad b = -7$

$$a : 13 ; b := -1 \vdash a \geq 0 \wedge b \geq 0$$

Luego, vale la Wp .

$$3) Q \equiv \left(\forall j : \mathbb{Z} \mid (0 \leq j < |A| \rightarrow_L A[j] \geq 0) \right) i : \mathbb{Z}, A : \text{list}(\mathbb{Z})$$

$$a) Wp(A[i] := 0, Q)$$

No puedo usar el Axioma de Inducción ya que así hablamos de Recursión.

Reescribo $b[i] := E$ (con SETAT(b,i,E))

$$= Wp(\text{SETAT}(A, i, E), Q)$$

TODO: no entiendo como reemplazar en un V.

Ejercicio 4. Para los siguientes pares de programas S y postcondiciones Q

- Escribir la precondition más débil $P = wp(S, Q)$
- Mostrar formalmente que la P elegida es correcta

a) $S \equiv$

```
if( a < 0 )
    b := a
else
    b := -a
endif
```

$$Q \equiv (b = -|a|)$$

c) $S \equiv$

```
if( i > 1 )
    s[i] := s[i-1]
else
    s[i] := 0
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L s[j] = s[j-1])$$

b) $S \equiv$

```
if( i > 0 )
    s[i] := 0
else
    s[0] := 0
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0)$$

d) $S \equiv$

```
if( s[i] > 0 )
    s[i] := -s[i]
else
    skip
endif
```

$$Q \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0)$$

Axioma 4: $\text{def}(B) \wedge_L ((B \wedge wp(s_1, Q)) \vee (\neg B \wedge wp(s_2, Q)))$

$$\text{a)} \quad wp(s_1, Q) = (b := a, b = -|a|)$$

$$= \text{def}(a) \wedge_L Q_a^b$$

$$= \text{true} \wedge_L a = -|a|$$

$$= a = -|a| \models E1$$

$$wp(s_2, Q) = (b := -a; b = -|a|)$$

$$= \text{def}(-a) \wedge_L Q_{-a}^b$$

$$= \text{true} \wedge_L -a = -|a|$$

$$= -a = -|a|$$

$$= a = |a|$$

$$wp(s, Q) = \text{True} \wedge_L ((a < 0 \wedge a = -|a|) \vee (a \geq 0 \wedge a = |a|))$$

if a is negative $\Rightarrow -a$

if a is positive $\Rightarrow +a$

Eins Copia $\forall a \left(\begin{array}{l} a > 0 \Rightarrow a \\ a \leq 0 \Rightarrow -a \end{array} \right)$

b es siempre a o $-a$ en forma negativa
Luego, $\text{True} \wedge_L \text{True} = \text{True}$ ✓

b) Y hace a los demás los mantiene al lo tiene 0 .

$$\text{def}(B) \wedge \left(\underbrace{(B \wedge wp(s_1, Q))}_{1} \vee \underbrace{(\neg B \wedge wp(s_2, Q))}_{2} \right)$$

$$1 \mid \left(i > 0 \wedge wp(s_1, Q) \right)$$

$$1.2 \mid wp(N[i] := 0, Q)$$

$$= \text{def}(\text{rectAn}(n, i, 0)) \wedge_L Q_{\text{rectAn}(n, i, 0)}$$

$$= (\text{def}(n) \wedge \text{def}(i) \wedge \text{def}(0)) \wedge_L 0 \leq i < |n| \wedge_L Q_{\text{rectAn}(n, i, 0)}$$

$$= ((\text{True} \wedge \text{True} \wedge \text{True}) \wedge_{\mathbb{L}} 0 \leq i < |N| \wedge_{\mathbb{L}} Q_{\text{retAt}}(n, i, 0))$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_{\mathbb{L}} \text{retAt}(n, i, 0)[j] \geq 0)$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_{\mathbb{L}} \text{retAt}(n, i, 0)[j] \geq 0)$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} ((\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_{\mathbb{L}} ((i = j \rightarrow_{\mathbb{L}} n[i] \geq 0) \vee_{\mathbb{L}} (i \neq j \rightarrow_{\mathbb{L}} n[j] \geq 0)))$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} ((\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i = j \rightarrow_{\mathbb{L}} (n[i] \geq 0)) \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i \neq j \rightarrow_{\mathbb{L}} (n[j] \geq 0)))$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} ((\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i = j \rightarrow_{\mathbb{L}} \text{True}) \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i \neq j \rightarrow_{\mathbb{L}} (n[j] \geq 0)))$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i \neq j \rightarrow_{\mathbb{L}} (n[j] \geq 0))$$

$$1) (\underbrace{i > 0 \wedge}_{\text{True}} (0 \leq i < |N| \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i \neq j \rightarrow_{\mathbb{L}} (n[j] \geq 0))))$$

$$= 0 \leq i < |N| \wedge_{\mathbb{L}} (\forall j : \mathbb{Z}) (0 \leq j < |N| \wedge i \neq j \rightarrow_{\mathbb{L}} (n[j] \geq 0))$$

$$2) (\underbrace{i \leq 0 \wedge}_{2.1} w_P(s_2, Q))$$

$$2.1) w_P(N[0]) := 0, Q$$

$$= \text{def}(\text{retAt}(n, 0, 0)) \wedge_{\mathbb{L}} \overset{\text{True}}{Q}_{\text{retAt}}(n, 0, 0)$$

$$= ((\text{def}(n) \wedge \text{def}(0) \wedge \text{def}(0)) \wedge_{\mathbb{L}} \underbrace{0 \leq 0 < |N| \wedge_{\mathbb{L}} \overset{|N| > 0}{Q}_{\text{retAt}}(n, 0, 0)}$$

$$= (|N| > 0 \wedge_{\mathbb{L}} \overset{|N| > 0}{Q}_{\text{retAt}}(n, 0, 0))$$

$$= (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_L \text{RETAIN}(n, 0, 0) \rightarrow_L \text{RETAIN}(n, 0, 0) [j] \geq 0))$$

$$= (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_L \text{RETAIN}(n, 0, 0) [j] \geq 0))$$

$$= (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) (0 \leq j < |N| \rightarrow_L ((\underbrace{j=0 \wedge n[0] \geq 0}_{C_1}) \vee (\underbrace{j \neq 0 \wedge n[j] \geq 0}_{C_2})))$$

$$= (|N| > 0 \wedge_L ((\forall j : \mathbb{Z}) ((0 \leq j < |N| \wedge j = 0) \rightarrow_L (n[0] \geq 0)) \wedge (\forall j : \mathbb{Z}) ((0 \leq j < |N| \wedge j \neq 0) \rightarrow_L (n[j] \geq 0)))$$

True $\Rightarrow V$
False $\Rightarrow F$

$$= (|N| > 0 \wedge_L ((\forall j : \mathbb{Z}) ((0 \leq j < |N| \wedge j = 0) \rightarrow_L \text{True}) \wedge (\forall j : \mathbb{Z}) ((0 \leq j < |N| \wedge j \neq 0) \rightarrow_L (n[j] \geq 0)))$$

$$= (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) ((0 \leq j < |N| \rightarrow_L (n[j] \geq 0)))$$

$$2) (i \leq 0 \wedge (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) ((0 \leq j < |N| \rightarrow_L (n[j] \geq 0))))$$

$$\text{RTA: } (0 \leq i < |N| \wedge_L (\forall j : \mathbb{Z}) ((0 \leq j < |N| \wedge i \neq j) \rightarrow_L (n[j] \geq 0)))$$

V

$$(i \leq 0 \wedge (|N| > 0 \wedge_L (\forall j : \mathbb{Z}) ((0 \leq j < |N| \rightarrow_L (n[j] \geq 0))))$$

C) d) TODO

Ejercicio 5. Para las siguientes especificaciones:

- Poner nombre al problema que resuelven
- Escribir un programa S sencillo en SmallLang, sin ciclos, que lo resuelva
- Dar la precondition más débil del programa escrito con respecto a la postcondición de su especificación

a) proc problema1 (in s: seq $\langle \mathbb{Z} \rangle$, in i: \mathbb{Z} , inout a: \mathbb{Z})

 requiere $\{0 \leq i < |s| \wedge_L a = \sum_{j=0}^{i-1} s[j]\}$
 asegura $\{a = \sum_{j=0}^i s[j]\}$

b) proc problema2 (in s: seq $\langle \mathbb{Z} \rangle$, in i: \mathbb{Z}) : Bool

 requiere $\{0 \leq i < |s| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] \geq 0)\}$
 asegura $\{res = \text{true} \leftrightarrow (\forall j : \mathbb{Z})(0 \leq j \leq i \rightarrow_L s[j] \geq 0)\}$

c) proc problema3 (inout s: seq $\langle \mathbb{Z} \rangle$, in i: \mathbb{Z})

 requiere $\{(0 \leq i < |s|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow s[j] = \text{fibonacci}(j))\}$
 asegura $\{(\forall j : \mathbb{Z})(0 \leq j \leq i \rightarrow s[j] = \text{fibonacci}(j))\}$

a). SUMARWASTAI

$$\cdot Q := N[i] + a$$

$$\cdot \text{WP}(a := N[i] + a, Q = \sum_{j=0}^i N[j])$$

$$= \text{def}(N[i] + a) \wedge Q_{N[i]+a}$$

$$= ((\text{def}(N) \wedge \text{def}(i) \wedge \text{def}(a)) \wedge 0 \leq i < |N| \wedge Q_{N[i]+a})$$

$$= (0 \leq i < |N| \wedge N[i] + a = \sum_{j=0}^i N[j])$$

$$= (0 \leq i < |N| \wedge a = \sum_{j=0}^i N[j] - N[i]) \quad \begin{matrix} \text{UUT TERM} \rightarrow N \text{ cancels..} \\ \text{Vorher i-1 now} \\ \text{entfernt} \end{matrix}$$

$$= (0 \leq i < |N| \wedge a = \sum_{j=0}^{i-1} N[j])$$

WP

b) • NUM IN SJCF E.S POSITIV

F2

• $\text{ver} = \text{false}$

IF ($N[i] \geq 0$) THEN

$\text{ver} := \text{true}$

ELSE

Skip

ENDIF

F1

• IF ($N[i] \geq 0$) THEN

$\text{ver} := \text{true}$

ELSE

$\text{ver} := \text{false}$

ENDIF.

• WANTS F1

$$\text{def}(N[i] \geq 0) \wedge \text{def}(N[i] \geq 0) \wedge \dots \wedge \text{def}(N[i] \geq 0)$$

$$\text{def}(N[i] \geq 0) \wedge ((N[i] \geq 0 \wedge \text{wp}(\text{new} := \text{True}, Q)) \vee (N[i] < 0 \wedge \text{wp}(\text{new} := \text{False}, Q)))$$

$$1) \text{def}(\text{True}) \wedge \text{Q}_{\text{true}}$$

$$= \text{True} \wedge (\text{True} = \text{True} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \rightarrow N[j] \geq 0))) \\ = (\text{True} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \rightarrow N[j] \geq 0)))$$

$$2) \text{def}(\text{false}) \wedge \text{Q}_{\text{false}}$$

$$= \text{True} \wedge (\text{False} = \text{True} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \Rightarrow N[j] \geq 0))) \\ = (\text{False} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \Rightarrow N[j] \geq 0)))$$

Recurse F1.

$$0 \leq i < |N| \wedge \left((N[i] \geq 0 \wedge (\text{True} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \rightarrow N[j] \geq 0)))) \vee \right. \\ \left. (N[i] < 0 \wedge (\text{False} \Leftrightarrow (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \Rightarrow N[j] \geq 0)))) \right)$$

$0 \leq i < |N|$ RESTRIKTE = $0 \leq j \leq i \leq 0 \leq i < |N|$

$0 \leq i < |N|$ REMINDERS NEGATIVES

$$\text{Ex: } N = [-1, 1, 2, 3] \quad i = 3 \quad N[0] = -1$$

$(N[i] < 0 \text{ zu } \vee \text{fuer } i=0 \text{ wahr } \Rightarrow \text{fals})$

$$(\forall j : \mathbb{Z} \mid (0 \leq j \leq 0 \rightarrow N[0] \geq 0)) \text{ fuer } N[0] \neq 0$$

$$= 0 \leq i < |N| \wedge \left((N[i] \geq 0 \wedge (\forall j : \mathbb{Z} \mid (0 \leq j \leq i \rightarrow N[j] \geq 0))) \vee N[i] < 0 \right)$$

• Umwandl. F2

1

$$\text{def}(N[i] \geq 0) \wedge \left((N[i] \geq 0 \wedge \text{wp}(\text{new} := \text{True}, Q)) \vee (N[i] < 0 \wedge \text{wp}(\text{new} := \text{False}, Q)) \right)$$

2
 $\text{wp}(\text{skip}, Q))$

1) $\text{def}(\text{true}) \wedge_L Q_{\text{true}}^{\text{new}}$

$$= \text{true} \wedge_L (\text{true} = \text{true} \Rightarrow (\forall j : \mathbb{Z} \mid 0 \leq j \leq i \rightarrow_L N[j] \geq 0))$$

$$= (\text{true} \Rightarrow (\forall j : \mathbb{Z} \mid 0 \leq j \leq i \rightarrow_L N[j] \geq 0))$$

2) $\exists j \text{ hay } m \in$

func,

$$(0 \leq i < |N| \wedge ((N[i] \geq 0 \wedge (\forall j : \mathbb{Z} \mid 0 \leq j \leq i \rightarrow_L N[j] \geq 0)) \vee (N[i] < 0)))$$

La RIA es igual.

C) TODO.

Ejercicio 6. Dado el siguiente código y postcondición

```

if (i mod 3 = 0)
    s[i] = s[i] + 6;
else
    s[i] = i;
endif

```

$Q \equiv \{(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$

Mostrar que las siguientes WP son incorrectas, dando un contraejemplo de ser posible

- a) $P \equiv \{0 \leq i \leq |s| \wedge i \bmod 3 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$
- b) $P \equiv \{0 \leq i < |s| \wedge i \bmod 3 \neq 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$
- c) $P \equiv \{0 \leq i < |s| \wedge (i \bmod 3 = 0 \vee i \bmod 2 = 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$
- d) $P \equiv \{i \bmod 3 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$
- e) $P \equiv \{0 \leq i < |s| \wedge i \bmod 3 \neq 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \bmod 2 = 0)\}$

A) ve se habrá. si $i = |N| \Rightarrow N[i]$ habrá

b) Cada par, indice q ms res mult de 3.

Los tngs q mstn ind no mult 3 q q en los pds
haga un par para depur por q impar para romper Q

COMO EJEMPLO

$$[0, 2, 4, 6] = \text{Ej: } i=1, \text{ ENT M A ELSE, S}[1]=1 \rightarrow S[1] \text{ NO Cumple Q} \\ \text{TAL QUE } S[1] \bmod 2 = 0.$$

c) Mismo criterio (b) pero filtra INDICE MOD 2 v INDICE MOD 3.

[0, 2, 4, 6, 8, 10]: Ej: $i=5$ rompe rmb xq tiene o else
 $N[5]=5$ q luego la linea q
no tiene todos pares.

d) Siempre tiene o IF q p me pde poner. PAR+PAR=PAR.

$i \neq N[3]$ oculto en P. si $N=[2, 4, 6]$ q mstn $i=3$
mst pver $3 \bmod 3 = 0$ q con todos pares pver $N[3]$ expd.

e) Cada q mstn todos xq limite range, i debe ser mult 3
q la linea q tiene todos pares.

OJO. Que la WP nra incongrua NO

significo q NO Cumple QC.

Algunos cumplen todos pver ms la WP
xq ms son los mst dleiles.

Especificación

```
proc sumar (in s: array < Z >): Z
    requiere {True}
    asegura {res =  $\sum_{j=0}^{|s|-1} s[j]$ }
```

Implementación en SmallLang

```
res := 0;
i := 0;
while (i < s.size()) do
    res := res + s[i];
    i := i + 1;
endwhile
```

Invariante de Ciclo

$$I \equiv 0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]$$

- a) Escribir la precondición y la postcondición del ciclo.
- b) ¿Qué punto falla en la demostración de corrección si el primer término de la sumatoria se reemplaza por $0 \leq i < |s|$?
- c) ¿Qué punto falla en la demostración de corrección si el límite superior de la sumatoria ($i - 1$) se reemplaza por i ?
- d) ¿Qué punto falla en la demostración de corrección si se invierte el orden de las dos instrucciones del cuerpo del ciclo?
- e) Mostrar la corrección parcial del ciclo, usando los primeros puntos del teorema del invariante.
- f) Proponer una función variante y mostrar la terminación del ciclo, utilizando la función variante.

a) $P_C : \text{PRECOND CICLO}$

$Q_C : \text{POSTCOND CICLO}$

$$P_C = \{ res = 0 \wedge i = 0 \}$$

$$Q_C = \{ i = |N| \wedge res = \sum_{j=0}^{|N|-1} N[j] \}$$

↓ Que debemos probar?

↓ Límite rango, no debemos ir más?

$i < |N| \Rightarrow i-1$ en i no llegará a ese índice.

$$b) I = 0 \leq i < |N| \wedge res = \sum_{j=0}^{i-1} N[j]$$

$$c) I = 0 \leq i \leq |N| \wedge res = \sum_{j=0}^{i-1} N[j]$$

d) Corrección parcial del ciclo:

- $P \Rightarrow w_P(s_1; s_2, P_C)$

1. $P_C \Rightarrow I$

2. $\{I \wedge B\} S \{I\}$

3. $\{I \wedge \neg B\} \Rightarrow Q_C$

$$P_C = \{ i=0 \wedge n[i]=0 \}$$

$$1. \{ i=0 \wedge n[i]=0 \} \Rightarrow \{ 0 \leq i \leq |N| \wedge n[i] = \sum_{j=0}^{i-1} n[j] \}$$

$$\Rightarrow \{ 0 \leq i \leq |N| \wedge 0 = \sum_{j=0}^{i-1} n[j] \}$$

$$\Rightarrow \{ |N| \geq 0 \wedge 0 = 0 \}$$

$\Rightarrow \{ |N| \geq 0 \}$ Luego esto es V puro & liso, la

longitud mínima es 1 y el requiere no tiene

restrictiones sobre él.

WP \Rightarrow w0 de S & I

2. $\{ I^A B \} (S) \{ I \}$ luego vemos que $I^A B \Rightarrow WP$

$$0 \leq i < |N|$$

$$\{ 0 \leq i \leq |N| \wedge \sum_{j=0}^{i-1} n[j] \leq i \leq |N| \} S \{ 0 \leq i \leq |N| \wedge \sum_{j=0}^{i-1} n[j] \}$$

Si $i < |N|$ entonces $0 \leq i \leq |N|$ puro $0 \leq i \leq |N|$ es más restrictivo que $i < |N|$.

Entonces concluimos que el invariante vale incluyendo después se entra al ciclo puro la memoria no borra $i-1$ ($i < |N|$).

$$2. \{ T \wedge T B \} \Rightarrow Q_C$$

$$\{0 \leq i \leq |N| \wedge \text{new} = \sum_{j=0}^{|N|-1} N[j] \wedge i \geq |N|\} \subseteq \{i = |N| \wedge \text{new} = \sum_{j=0}^{|N|-1} N[j]\}$$

Si $i \geq |N|$ no se cumple $i = |N|$ en Q_C , & no se cumple la invariante que es igual el range. ✓

R) $F_v = |N| - i \Rightarrow 0$ metidos que i crece, $|N| - i$ se acerca a 0.

$$1. \{I \wedge B \wedge v_0 = F_v\} \subseteq \{F_v < v_0\}$$

$$2. I \wedge F_v \leq 0 \Rightarrow \neg B$$

$$1. \{0 \leq i \leq |N| \wedge \text{new} = \sum_{j=0}^{i-1} N[j] \wedge v_0 = |N| - i\} \subseteq \{|N| - i < v_0\}$$

Logo $WP(\text{new} := \text{new} + N[i]; i := i + 1, |N| - i < v_0)$

$$WP(\text{new} := \text{new} + N[i]; WP(i := i + 1, |N| - i < v_0))$$

$$1. \text{def}(i+1) \wedge Q_{i+1}^i = |N| - i < v_0 + 1$$

$$= \text{True} \wedge |N| - (i+1) < v_0 = |N| - i \leq v_0$$

$$= |N| - i - 1 < v_0$$

$$2. WP(\text{new} := \text{new} + N[i], E1)$$

$$= \text{def}(\text{new} + \text{N}[i]) \wedge_L Q_{\text{new} + \text{N}[i]}^{\text{new}}$$

$$= \text{def}(\text{new}) \wedge (\text{def}(n) \wedge \text{def}(i)) \wedge_L 0 \leq i < n \wedge_L Q_{\text{new} + \text{N}[i]}^{\text{new}}$$

$$= 0 \leq i < n \wedge_L |n| - i \leq \text{no}_j ?$$

Ejercicio 2. Dadas la especificación y la implementación del problema `sumarParesHastaN`, escribir la precondición y la postcondición del ciclo, y mostrar su corrección a través del teorema del invariante.

Especificación

```
proc sumarParesHastaN (in n: Z) : Z
    requiere {n ≥ 0}
    asegura {res =  $\sum_{j=0}^{n-1}$  (if j mod 2 = 0 then j else 0 fi)}
```

Implementación en SmallLang

```
res := 0;
i := 0;
while (i < n) do
    res := res + i;
    i := i + 2
endwhile
```

Invariante de ciclo

$$I \equiv 0 \leq i \leq n+1 \wedge i \bmod 2 = 0 \wedge \text{res} = \sum_{j=0}^{i-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})$$

$$P_C = \{m \geq 0 \wedge \text{new} := 0 \wedge i := 0\}$$

$$Q_C = \{i \geq m \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{m-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$$

$$\cdot P \rightarrow P_C: \{m \geq 0\} \Rightarrow \{m \geq 0 \wedge \text{new} := 0 \wedge i := 0\}$$

$$\Rightarrow \{\text{new} := 0 \wedge i := 0\}$$

Problema: si no cumple el require al inicio de `new` e `i`.

$$\cdot P_C \Rightarrow I: \{m \geq 0 \wedge \text{new} := 0 \wedge i := 0\} \Rightarrow \{0 \leq i \leq m+1 \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{i-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$$

$$\Rightarrow \{0 \leq i \leq m+1 \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{i-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$$

$$\Rightarrow \{0 \leq i \leq m+1\}$$

$$\cdot \{I \wedge B\} \models \{I\}$$

$$\left\{ 0 \leq i \leq m+1 \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{i-1} \text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi} \wedge i < m \right\} S \left\{ 0 \leq i \leq m+1 \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{i-1} \text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi} \wedge i < m \right\}$$

$i \leq i < n \Rightarrow i \leq m+1$ pero $i \leq m+1$ es más restrictivo.

PREGUNTAR

- $\{ I \wedge \top_B \} \Rightarrow Q_c$ PREGUNTAR.

$$= \{ I \wedge i \geq m \} \Rightarrow \{ i \geq m \wedge i \bmod 2 = 0 \wedge \text{new} = \sum_{j=0}^{m-1} \text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi} \}$$

Ejercicio 3. Considere el problema `sumaDivisores`, dado por la siguiente especificación:

```
proc sumaDivisores (in n: Z) : Z
    requiere {n ≥ 1}
    asegura {res = ∑j=1n (if n mod j = 0 then j else 0 fi)}
```

a) Escribir un programa en SmallLang que satisfaga la especificación del problema y que contenga exactamente un ciclo.

1

b) Escribir la pre y post condición del ciclo y su invariante.

c) Considere el siguiente invariante para este problema

$$I \equiv 1 \leq i \leq n/2 \wedge \text{res} = \sum_{j=1}^i (\text{if } n \bmod j = 0 \text{ then } j \text{ else } 0 \text{ fi})$$

Si no coincide con el propuesto en el inciso anterior, ¿qué cambios se le deben hacer al programa para que lo represente este invariante? ¿Deben cambiar la pre y post condición?

Nuestro divisor tiene n iteraciones

$i := 1$

$\text{new} := 0$

while ($i \leq m$) do

 if ($m \bmod i = 0$) THEN

```

    rev := rev + 3
ELSE
    SKIP
    Fi
    i := i+1
END WHILE

```

$M = m = 4$ after 3(1,2,4)

$M = 4, i = 1, rev = 1 \Rightarrow i = 2$
 $i = 2, rev = 3 \Rightarrow i = 3$
 $i = 3, rev = 3 \Rightarrow i = 4$
 $i = 4, rev = 7 \Rightarrow i = 5$

while
ciels.

$$b). P_C = \{ M \geq 1 \wedge i := 1 \wedge rev := 0 \}$$

$M \rightarrow$ erreichbar?

$$\cdot Q_C = \{ i > M \wedge rev = \sum_{j=0}^M \text{IF}(M \text{ mod } j = 0 \text{ THEN } 1 \text{ ELSE } 0) \}$$

$$\cdot I = \{ 1 \leq i \leq M \wedge rev = \sum_{j=0}^{i-1} \text{IF}(M \text{ mod } j = 0 \text{ THEN } 1 \text{ ELSE } 0) \}$$

c) El problema con este algoritmo es que

la memoria no tiene la memoria de los divisores ninguno de ellos.

PC tiene que ser para cuando Q_C es el programa.

```
i := 1  
nev := 0  
while (i <=  $\frac{m}{2}$ ) do  
    if (m mod i = 0) then  
        nev := nev + 1  
    else  
        skip  
    fi  
    i := i + 1  
end while
```

$$Q_C = \left\{ i > \frac{m}{2} \wedge \sum_{j=1}^{\frac{m}{2}} \text{if } (m \bmod j = 0) \text{ then } 1 \text{ else } 0 \right\}$$

Entonces $i \leq m=4$, $1 \leq i \leq 2$

$i = 1$, $nev = 1$

$i = 2$, $nev = 3$

↓ Terminó

Ejercicio 4. Considere la siguiente especificación e implementación del problema copiarSecuencia, y la pre y post condiciones del ciclo

Especificación

```
proc copiarSecuencia (in s: array < Z >, inout r: array < Z >)  
    requiere { |s| = |r| }  
    asegura { |s| = |r| }  
    ( $\forall j : \mathbb{Z}$ ) ( $0 \leq j < |s| \rightarrow_L s[j] = r[j]$ )
```

Implementación en SmallLang

```
i := 0;  
while (i < s.size()) do  
    r[i] := s[i];  
    i := i + 1  
endwhile
```

$$P_c \equiv |s| = |r| \wedge i = 0$$

$$Q_c \equiv (\forall j : \mathbb{Z}) (0 \leq j < |r| \rightarrow_L s[j] = r[j])$$

- a) ¿Qué variables del programa deben aparecer en el invariante? CUAL PARA QUE INU PARA LAS DEMOSTRACIONES?
- b) Proponer un invariante e indicar qué cláusula del mismo es necesario para cada paso de la demostración.
- c) Proponer una función variante que permita demostrar que el ciclo termina.
- d) Comparar la solución propuesta con la que ofrecemos al final de la guía.

a) $i \in \mathbb{N} \wedge n$

b) $\{ |n| \leq i \wedge 0 \leq i \leq |n| \wedge (\forall j \in \mathbb{Z}) (0 \leq j < i \rightarrow n[j] = n[i]) \}$

c) $FV = |n| - i$
 esto muestra el teorema

lo demos por inducción:

Ejercicio 5. Sea el siguiente ciclo con su correspondiente precondición y postcondición:

```
while (i >= s.size() / 2) do
    suma := suma + s[s.size() - 1 - i];
    i := i - 1
endwhile
```

$$P_c : \{|s| \bmod 2 = 0 \wedge i = |s| - 1 \wedge suma = 0\}$$

$$Q_c : \{|s| \bmod 2 = 0 \wedge i = |s|/2 - 1 \wedge_L suma = \sum_{j=0}^{|s|/2-1} s[j]\}$$

- Proponer un invariante e indicar qué cláusula del mismo es necesario para cada paso de la demostración.
- Proponer una función variante que permita demostrar que el ciclo termina.
- Comparar la solución propuesta con la que ofrecemos al final de la guía.

{ ¿Qué hace el programa?

Va a ir restando i y refiere a i

$i > \frac{|n|}{2}$, suma los últimos números

Ej: $n = [1, 2, 3, 4]$ $i = 3$ se hace $i = 1$ para $\frac{4}{2} = 2$

$$\text{suma} := \text{suma} + \overbrace{n[4-1-3]}^{0}$$

$$i = 2$$

$$\text{num} := \text{num} + N[\overbrace{i-1-2}^1]$$

Otro término

$$2) I = \{ |N| \neq 2 : 0 \wedge 0 \leq i \leq \frac{|N|}{2} \wedge_{\sim} (H_i : \gamma_L) \cdot (0 \leq i < i \rightarrow \text{num} = \sum_{j=0}^{i-1} N(j)) \}$$

MAL

o

$$\leq \frac{|N|}{2} - 1 = < \frac{|N|}{2}$$

$$I = \{ |N| \neq 2 : 0 \wedge \frac{|N|}{2} \leq i < \frac{|N|}{2} \wedge_{\sim} (H_i : \gamma_L) \mid (0 \leq i < i \rightarrow \text{num} = \sum_{j=0}^{i-1} N(j)) \}$$

$ N = 4$	i	$N[i]$	$ N $
	3	0	4
	2	1	4

$$b) |N| + i - 1 = 4 + 3 - 1 = 6$$

$$= 4 + 2 - 1 = 5$$

$$i - \left(\frac{|N|}{2} - 1 \right)$$

Ejercicio 6. Dado el siguiente problema

```
proc sumarElementos (in s: array < Z >): Z
    requiere { |s| ≥ 1 }
    asegura { res =  $\sum_{j=0}^{|s|-1} s[j]$  }
```

a) Corregir las siguientes implementaciones

b) Dar un invariante y función variante para cada una de estas implementaciones

2

a) $\text{res} := 0$
 $i := 0$ ↗
while ($i > s.\text{size}()$) **do**
 ↳ $\text{res} := \text{res} + s[i];$
 $i := i + 1$
endwhile

c) $\text{res} := 0$
 $i := s.\text{size}() - 1$ ✓
while ($i \geq 0$) **do**
 ↳ $\text{res} := \text{res} + s[i];$
 $i := i - 1$
endwhile

b) $\text{res} := 0$
 $i := 0$ ↗
while ($i > s.\text{size}()$) **do** ↗ +1
 ↳ $\text{res} := \text{res} + s[s.\text{size}() - i];$ ↗ +1
 $i := i + 1$ ↗ -1
endwhile

d) $\text{res} := 0$
 $i := 0$ ↗
while ($i \geq s.\text{size}() / 2$) **do** ↗
 ↳ $\text{res} := \text{res} + s[i] + s[s.\text{size}() - i];$ ↗
 $i := i + 1$ ↗
endwhile

↑ i = 0 se incluye

$$b) a) I = \{ 0 \leq i \leq |n| \wedge \forall j: \pi \mid 0 \leq j < i \rightarrow n_j = \sum_{j=0}^{i-1} n[j] \}$$

$$\Gamma_v = |n| - i$$

$$P_C = \{ n_i = 0 \wedge i = 0 \wedge |n| \geq 1 \}$$

$$Q_C = \{ i \geq |n| \wedge n_i = \sum_{j=0}^{|n|-1} n[j] \}$$

Von $P \Rightarrow w_P(n := 0; i := 0, P_C)$

$$w_P(i := 0, P_C) \equiv \text{def}(0)^i, Q_0^i \equiv \text{True} \wedge$$

$$n := 0 \wedge 0 = 0 \wedge |N| \geq 1 \equiv n := 0 \wedge |N| \geq 1 \stackrel{?}{=} E1$$

folgt, $w_P(n := 0, E1) \equiv \text{def}(0)^0, Q_0^0 \equiv \text{True} \wedge$

$$0 = 0 \wedge |N| \geq 1 \equiv |N| \geq 1 \checkmark$$

folgt, $|N| \geq 1 \Rightarrow |N| \geq 1 \quad /$

Probenraum für $P \Rightarrow P_C$

$$\bullet P_C \Rightarrow I$$

$$\{n := 0 \wedge i := 0 \wedge |N| \geq 1\} \Rightarrow$$

$$0 \leq i \leq |N| \wedge \left(\forall j: \pi \mid 0 \leq j < i \rightarrow \sum_{j=0}^i n[j] \right) \quad \text{oder} \quad \sum_{j=0}^i n[j] = |N|$$

$$\Rightarrow \{|N| \geq 1\} \Rightarrow \left\{ 0 \leq 0 \leq |N| \wedge 0 = \sum_{j=0}^0 n[j] \right\}$$

$$\Rightarrow \{|N| \geq 1\} \Rightarrow \{|N| \geq 0 \wedge \text{True}\}$$

$$\Rightarrow \{|N| \geq 1\} \quad \text{per se mas restriktiv.} \quad \checkmark$$

$$|N| \geq 1 \subseteq |N| \geq 0$$

PREGUNTAR. ✓

$$\cdot \{ I \wedge \neg B \} \Rightarrow \{ Q_c \}$$

$$\{ 0 \leq i \leq |N| \wedge_L (\forall j: N) \underbrace{| 0 \leq j < i} \rightarrow_L \text{Ner} = \sum_{j=0}^{i-1} n[j] \} \uparrow i > |N|$$

$\curvearrowleft S \curvearrowright i = |N|$

$$\{ \underbrace{i = |N|}_{\text{S}} \wedge \text{Ner} = \sum_{j=0}^{|N|-1} n[j] \}$$

PREGUNTAR ✓

Lo visto en ejercicio de clase horas $\neg B$.

$$\cdot \{ I \wedge B \wedge F_v = v_0 \} \cup \{ F_v < v_0 \}$$

Ciclo.

$$= w_p \left(\underbrace{\text{Ner} := \text{Ner} + n[i];}_{\text{Ciclo}} \underbrace{i := i+1; |N|-i < v_0}_{\text{Ciclo}} \right)$$

$$= w_p (i := i+1, |N|-1 < v_0)$$

$$\equiv \text{Udef}(i+1) \wedge_L Q_{i+1}^i$$

$$\equiv (\exists r_{v0} \wedge_L |N| - (i+1) < v_0)$$

$$\equiv |N| - i - 1 < v_0$$

$$\equiv |N| - i \leq v_0 \equiv E1$$

$F_v < V_0$

Leyendo, $WP(n_r := n_r + n[i], |N|-i \leq V_0) \equiv |N|-i \leq V_0$

Leyendo, $\{I \wedge B \wedge F_v = V_0\} \vdash \{F_v < V_0\}$

$I \wedge B \wedge F_v = V_0 \Rightarrow |N| - i \leq V_0$

$\underbrace{\{0 \leq i \leq |N| \wedge \bigwedge_L (U_j : \pi) \mid 0 \leq j < i \rightarrow_r n_r = \sum_{j=0}^{i-1} n[j]\}}$

$\wedge (i < |N|) \wedge (|N| - i = V_0) \Rightarrow |N| - i \leq V_0$

$\Rightarrow |N| - i \leq |N| - i$

$\Rightarrow \text{True}$ ✓

$\cdot \{I \wedge F_v \leq 0\} \Rightarrow \neg B$ *

FORMA 1: (AMIGUANERA)

$0 \leq i \leq |N| \wedge \bigwedge_L (U_j : \pi) \mid 0 \leq j < i \rightarrow_r n_r = \sum_{j=0}^i n[j] \wedge |N| - i \leq 0$

$\Rightarrow i \geq |N|$

$i \geq |N|$ significa que todos los círculos para

$n_r = \sum_{j=0}^i \dots$ no tiene $i=0$ hasta $i=|N|-1$.

FORMA 2:

$$0 \leq i \leq |v| \wedge (\forall j : \gamma_L)(0 \leq j < i \rightarrow v_{\bar{v}j} = \sum_{j=0}^i N[j]) \wedge |v|-i \leq 0$$

$$i \leq |N| \wedge |N| - i \leq 0 \equiv i \leq |N| \wedge |N| \leq i \equiv \neg(|N| \leq i) \equiv |N| > i \quad \text{PNEG}$$

$$0 \leq i < 10 \Rightarrow 1 \leq i < 10 \Rightarrow F.$$

$$0 \leq i \leq |N| \wedge_L (\forall j : \gamma_L) \left(0 \leq j < i \rightarrow_L \text{rem} = \sum_{j=0}^i v[j] \wedge (|N|-i \leq 0) \right)$$

$$\Rightarrow i \geq |N|$$

$$* |n| - i \leq 0 \Rightarrow i \geq |n|$$

$$|n| \leq i \Rightarrow i > |n|$$

True ✓

Ejercicio 7. Considerando el siguiente Invariante:

$$I \equiv \{0 \leq i \leq |s| \wedge (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L (j \bmod 2 = 0 \wedge s[j] = 2 \times j) \vee (j \bmod 2 \neq 0 \wedge s[j] = 2 \times j + 1))\}$$

- Escribir un programa en SmallLang que se corresponda al invariante dado.
- Defina las P_c , B y Q_c que correspondan a su programa.
- Dar una función variante para que se pueda completar la demostración.

Pista los números se los lleva $|N| = \Theta(|n|)$

a)

$i := 0$

WHILE ($i < n.\text{size}()$)

IF ($i \bmod 2 == 0$) THEN

$N[i] := 2 * i$

ELSE

$N[i] := 2 * i + 1$

ENDIF

$i := i + 1$

λ := λ + 1

END WHILE

$$b) P_C = \{ |N| > 0 \wedge i = 0 \}$$

$$B = \{ i < |N| \}$$

$$Q_C = \{ i > |N| \wedge 0 \leq i \leq |N| \wedge \forall j : \exists l (0 \leq j < i - 1) \wedge$$

$$\left((j \bmod 2 = 0 \wedge N[j] = 2 * j) \vee (j \bmod 2 \neq 0 \wedge N[j] = 2 * j + 1) \right)$$

$$F_V = |N| - i$$

Ejercicio 9. Indique si el siguiente enunciado es verdadero o falso; fundamente:

Si dados B y I para un ciclo S existe una función f_v que cumple lo siguiente:

- $\{I \wedge B \wedge f_v = V_0\} S \{f_v > V_0\} \rightarrow \text{NUNCA PASA}$
- $\exists (k : \mathbb{Z}) (I \wedge f_v \geq k \rightarrow \neg B)$
 entonces el ciclo siempre termina.

F_V debe ser $\leq k$ y $k \leq 0$.

ES V, PORQUE VA AL RÉFÍS

Ejercicio 10. Considere la siguiente especificación y su implementación

Especificación

```
proc existeElemento (in s: array < Z >, in e: Z) : Bool
    requiere {True}
    asegura {res = true  $\leftrightarrow$ 
         $(\exists k : \mathbb{Z})(0 \leq k < |s| \wedge s[k] = e)$ }
```

Implementación en SmallLang

```
i := 0;
j := -1;
while (i < s.size()) do
    if (s[i] = e) then
        j := i
    else
        skip
    endif;
    i := i + 1
endwhile;
if (j != -1)
    res := true
else
    res := false
endif
```

Escribir los pasos necesarios para demostrar la correctitud de la implementación respecto a la especificación usando WP y el teorema del invariante

Demostrar invario del elem.

$P_C = \{ i := 0 \wedge j := -1 \} \rightarrow j \text{ dependet auf } i \text{ if, nach TMB}$

$Q_C = \{ i \geq |N| \wedge \text{new} = \text{True} \Leftrightarrow (\exists k \in \mathbb{Z}) (0 \leq k < i) \wedge N[k] = e \}$

$I = \{ 0 \leq i \leq |N| \wedge \text{new} = \text{True} \Leftrightarrow (\exists k \in \mathbb{Z}) (0 \leq k < i) \wedge N[k] = e \}$

\downarrow $\overset{\text{Const}}{\text{dPUE}} \text{ Stein abhängt von } i, j, \text{new?}$

• $P \Rightarrow \text{wp}(i := 0, j := -1, P_C)$

$\equiv \text{wp}(j := -1, i := 0 \wedge j := -1)$

$\equiv \text{def}(-1) \wedge Q_{-1}^j \equiv \text{True} \wedge i := 0 \wedge -1 = -1$

$\equiv i := 0 \equiv E1$

Folger, $\text{wp}(i := 0, i := 0) \equiv \text{def}(0) \wedge Q_0^i$

$\equiv 0 = 0$

$\equiv \text{True}$

Folger, $\text{True} \Rightarrow \text{True}$

• Dagegen wp ifc.

1

2

$\text{def}(B) \wedge_L ((B \wedge \text{wp}(\text{new} := \text{True}, Q_C)) \vee (\neg B \wedge \text{wp}(\text{new} := \text{False}, Q_C)))$

\downarrow
Wie kann das geladen werden?

1. $j \neq 1 \wedge \text{wp}(\text{new} := \text{True}, i \geq |N| \wedge \text{new} = \text{True} \Leftrightarrow ((\exists k : N)(0 \leq k < n) \wedge N[k] = e))$

$\equiv \text{def}(\text{True}) \wedge Q_{\text{True}}^{\text{new}}$

$\equiv \text{True} \wedge i \geq |N| \wedge \text{True} = \text{True} \Leftrightarrow \dots$

Como $j \neq 1$ no es ni dentro elem, x lo tanto

$j \neq 1 \wedge \text{True} \equiv j \neq 1 \wedge i \geq |N|$

2. $j = 1 \wedge \text{wp}(\text{new} := \text{False}, i \geq |N| \wedge \text{new} = \text{True} \Leftrightarrow ((\exists k : N)(0 \leq k < n) \wedge N[k] = e))$

$\equiv \text{def}(\text{False}) \wedge Q_{\text{False}}^{\text{new}}$

$\equiv \text{True} \wedge i \geq |N| \wedge \text{False} \Leftrightarrow \dots$

Como $j = 1$ NO es el elem x lo tanto el

es nulo.

Luego, $j = 1 \wedge i \geq |N|$

Por lo tanto, $\text{True} \wedge ((j \neq 1 \wedge i \geq |N|) \vee (j = 1 \wedge i \geq |N|))$

$\equiv j \neq 1 \vee j = 1$

$\equiv \text{True}$



