EJERCICIO 1

JONATHAN BEKENSTEIN

348/11

2021-10-15

$S_1 \equiv s := s - t$

$S_2 \equiv t := t - 1$

$S \equiv S_1 ; S_2$

$B \equiv s > 0$

$\neg B \equiv s \leq 0$

$P_c : \{ n \geq 0 \wedge s = (n(n+1)) \text{ div } 2 \wedge t = n \}$

$Q_c : \{ s = 0 \wedge t = 0 \}$

$I : \{ 0 \leq t \leq n \wedge s = (t(t+1)) \text{ div } 2 \wedge s \geq 0 \}$

PREGUNTA A: $I \wedge \neg B \Rightarrow Q_c$

$I \wedge \neg B \equiv 0 \leq t \leq n \wedge s = (t(t+1)) \text{ div } 2 \wedge s \geq 0 \wedge s \leq 0$

- $s \geq 0 \wedge s \leq 0 \Rightarrow s = 0$
- $s = 0 \wedge s = (t(t+1)) \text{ div } 2$

$\Rightarrow s = 0 \wedge s = t(t+1)/2$ PUES LA SUMA DE DOS NÚMEROS $\mathbb{Z}$ CONSECUTIVOS SIEMPRE ES PAR.

3 + 4 = 7....
De hecho, la suma de dos enteros consecutivos siempre es impar.

$\Rightarrow 0 = t(t+1)/2$

$\Rightarrow t = 0 \vee t = -1$ PERO POR EL I: $0 \leq t \leq n \Rightarrow 0 \leq t$

$\Rightarrow t = 0$

$I \wedge \neg B \Rightarrow s = 0 \wedge t = 0 \equiv Q_c$

PREGUNTA B : $\{I \wedge B\}\ S\ \{I\} \iff I \wedge B \Rightarrow wp(S, I)$

$$wp(S, I) \equiv wp(S_1; S_2, I) \overset{Ax3}{\equiv} wp(S_1, wp(S_2, I))$$

$wp(S_2, I)$

$\equiv wp(t := t-1, I)$

$\overset{Ax1}{\equiv} def(t) \wedge_L I^{t}_{t-1}$

$\equiv True \wedge I^{t}_{t-1}$ ASUMIMOS QUE TODAS LAS VARIABLES ESTÁN DEFINIDAS

$\equiv 0 \leq t-1 \leq n \wedge s = ((t-1)(t-1+1)) \text{ div } 2 \wedge s \geq 0$

$\equiv 0 \leq t-1 \leq n \wedge s = (t(t-1)) \text{ div } 2 \wedge s \geq 0$

$\equiv E_2$

$wp(S_1, wp(S_2, I)) \equiv wp(S_1, E_2)$

$\equiv wp(s := s-t, E_2)$

$\overset{Ax1}{\equiv} def(s) \wedge def(t) \wedge_L {E_2}^{s}_{s-t}$

$\equiv 0 \leq t-1 \leq n \wedge \underbrace{s-t = (t(t-1)) \text{ div } 2}_{\equiv} \wedge s-t \geq 0$

$s-t = t(t-1)/2 \equiv s = t(t-1)/2 + t$

$\equiv s = t(t-1)/2 + 2t/2 \equiv s = (t(t-1) + 2t)/2$

$\equiv s = t(t-1+2)/2 \equiv s = t(t+1)/2$

$\equiv s = (t(t+1)) \text{ div } 2$

$\equiv 0 \leq t-1 \leq n \wedge s = (t(t+1)) \text{ div } 2 \wedge s-t \geq 0$

$\equiv E_1$

NOTA

EJERCICIO 1

JONATHAN BEKENSTEIN

348/11

2021-10-15

QVQ: $I \wedge B \Rightarrow wp(S, I) \equiv E_1$

$I \wedge B \equiv 0 \leq t \leq n \wedge s = (t(t+1)) \operatorname{div} 2 \wedge s \geq 0 \wedge s > 0$

$E_1 \equiv 0 \leq t-1 \leq n \wedge s = (t(t+1)) \operatorname{div} 2 \wedge s-t \geq 0$

$s = (t(t+1)) \operatorname{div} 2 \Rightarrow s = (t(t+1)) \operatorname{div} 2$

EN $I \wedge B$, $s > 0 \Rightarrow (t(t+1)) \operatorname{div} 2 > 0 \Rightarrow t > 0$

COMBINANDO $t > 0 \wedge 0 \leq t \leq n \Rightarrow 0 < t \leq n$

$\Rightarrow -1 < t-1 \leq n-1$

$\Rightarrow 0 \leq t-1 \leq n$

EN $I \wedge B$, $s = (t(t+1)) \operatorname{div} 2$, ES DECIR QUE s ES LA SUMA DE GAUSS DE LOS ENTEROS ENTRE 0 Y t. POR LO TANTO ES EVIDENTE QUE VALE $s-t \geq 0 \equiv s \geq t$ EN LA WP.

EJERCICIO 2 | JONATHAN BEKENSTEIN | HOJA N

348/11 | FECHA 2021-10-15

pred esDescompresión(cod: Código, pal: Palabra) {

$(\forall i : \mathbb{Z})(0 \leq i < |cod| \Rightarrow_L$

esFragmentoVálido(

$(cod[i])_0$,

$(cod[i])_1$,

sumarLongitudDeFragmentos(subseq(cod, 0, i)),

pal

)

)

}

pred esFragmentoVálido(frag: Palabra, rep: $\mathbb{Z}$, offset: $\mathbb{Z}$, pal: Palabra) {

$(\forall r : \mathbb{Z})(0 \leq r < rep \Rightarrow_L$

$offset + |Frag| \cdot r \leq offset + |Frag| \cdot (r+1) \leq |pal|$

$\wedge_L$ subseq(

pal,

$offset + |Frag| \cdot r$,

$offset + |Frag| \cdot (r+1)$

) = Frag

aux sumarLongitudDeFragmentos(cod: Código) =

$$\sum_{i=0}^{|cod|-1} |(cod[i])_0| \cdot (cod[i])_1$$

NOTA

```
proc comprimir (in pal: Palabra, out cod: Código) {
    Pre {
        |pal| > 0
    }
    Post {
        esCódigoVálido(cod) ∧ esDescompresión(cod, pal)
    }
    pred esCódigoVálido(cod: Código) {
        (∀i: ℤ)(0 ≤ i < |cod| ⟹L |(cod[i])₀| > 0 ∧ (cod[i])₁ > 0 )
    }
}

proc optimizarCódigo (inout cod: Código) {
    Pre {
        cod = C₀
    }
    Post {
        (∃pal: Palabra)(
            esDescompresión(C₀, pal)
            ∧ esDescompresión(cod, pal)
            ∧ esElCódigoMásBarato(cod, pal)
        )
    }
    pred esElCódigoMásBarato(c₁: Código, pal: Palabra) {
        (∀c₂: Código)(esDescompresión(c₂, pal) ⟹L costo(c₂) ≥ costo(c₁))
    }
}
```

NOTA

EJERCICIO 3

JONATHAN BEKENSTEIN

348/11

2021-10-15

$P : \{ k+1 < |s| \wedge_L \text{posicionesCorrespondientes}(s, k, i, j) \}$

$Q : \{ \text{posicionesCorrespondientes}(s, k+1, i, j) \}$

$B \equiv s[k] = 0$

$\neg B \equiv s[k] \neq 0$

$S_1 \equiv i := i+1$

$S_2 \equiv j := 0$

$S_3 \equiv j := j+1$

$S \equiv$ if B then $S_1; S_2$ else S_3 endif

PREGUNTA A

$$wp(S, Q) \overset{Ax4}{\equiv} def(B) \wedge_L ((B \wedge \underbrace{wp(S_1, wp(S_2, Q))}^{Ax3\ wp(S_1;S_2, Q) \equiv wp(S_1, wp(S_2, Q))}) \vee (\neg B \wedge wp(S_3, Q)))$$

$def(B) \equiv 0 \leq k < |s|$

$wp(S_2, Q) \overset{Ax1}{\equiv} Q^j_0 \equiv \text{posicionesCorrespondientes}(s, k+1, i, 0) \equiv E_2$

$wp(S_1, wp(S_2, Q)) \equiv wp(S_1, E_2) \overset{Ax1}{\equiv} E_2{}^{i}_{i+1} \equiv \text{posicionesCorrespondientes}(s, k+1, i+1, 0) \equiv E_1$

$wp(S_3, Q) \overset{Ax1}{\equiv} Q^j_{j+1} \equiv \text{posicionesCorrespondientes}(s, k+1, i, j+1) \equiv E_3$

$wp(S, Q) \equiv def(B) \wedge_L ((B \wedge E_1) \vee (\neg B \wedge E_3))$

$\equiv 0 \leq k < |s| \wedge_L ($

$(s[k] = 0 \wedge \text{posicionesCorrespondientes}(s, k+1, i+1, 0))$

$\vee$

$(s[k] \neq 0 \wedge \text{posicionesCorrespondientes}(s, k+1, i, j+1))$

$)$

PREGUNTA B

QVQ: $\{P\}\ S\ \{Q\} \iff P \Rightarrow wp(S, Q)$

$posicionesCorrespondientes(s, k, i, j) \Rightarrow 0 \leq k < |s|$

CASO $s[k]=0$: QVQ: $P \Rightarrow posicionesCorrespondientes(s, k+1, i+1, 0)$

$posicionesCorrespondientes(s, k+1, i+1, 0)$

$\equiv (0 \leq k+1 < |s| \wedge 0 \leq 0 \leq k+1)$ ①

$\wedge_L\ cantApariciones(subseq(s, 0, k+1-0), 0) = i+1$ ②

$\wedge\ (k+1-0=0 \vee_L s[k+1-0-1]=0)$ ③

$\wedge\ cantApariciones(subseq(s, k+1-0, k+1), 0) = 0$ ④

① POR P: $k+1<|s| \wedge 0 \leq k < |s| \Rightarrow 0 \leq k+1 < |s| \wedge 0 \leq k+1$

② POR P: $cantApariciones(subseq(s, 0, k-j), 0) = i$

ES DECIR HAY i CANTIDAD DE 0 EN LA SUBSEQ ENTRE $[0, k-j)$

ADEMÁS POR P: $cantApariciones(subseq(s, k-j, k), 0) = 0$

NOS DICE QUE NO HAY NINGÚN 0 EN LA SUBSEQ ENTRE $[k-j, k)$

$\Rightarrow cantApariciones(subseq(s, 0, k), 0) = i$

ES DECIR HAY i CANTIDAD DE CEROS EN LA SUBSEQ ENTRE $[0, k)$

COMBINANDO ESTO CON EL CASO $s[k]=0$, PODEMOS IMPLICAR ②

$\Rightarrow cantApariciones(subseq(s, 0, k+1), 0) = i+1$

ES DECIR AHORA HAY $i+1$ CANTIDAD DE 0 EN LA SUBSEQ ENTRE $[0, k+1)$

PUES INCLUIMOS $s[k]$ EN LA SUBSEQ QUE SABEMOS QUE VALE 0.

NOTA

EJERCICIO 3

JONATHAN BEKENSTEIN

348/11

2021-10-15

③ SIMPLIFIQUEMOS LO QUE QUEREMOS IMPLICAR

$(k+1-0=0 \vee_L s[k+1-0-1]=0) \equiv k+1=0 \vee_L s[k]=0$

COMO ESTAMOS EN EL CASO $s[k]=0$, ESTE PREDICADO SE CUMPLE.

④ SIMPLIFIQUEMOS LO QUE QUEREMOS IMPLICAR

$cantApariciones(\underbrace{subseq(s, k+1-0, k+1)}_{\text{DEVUELVE UNA LISTA VACÍA}}, 0) = 0$

$\equiv cantApariciones(\langle\rangle, 0) = 0 \equiv True$ ES UNA TAUTOLOGÍA

CASO $s[k] \neq 0$: QVQ: $P \Rightarrow posicionesCorrespondientes(s, k+1, i, j+1)$

$posicionesCorrespondientes(s, k+1, i, j+1)$

$\equiv (0 \leq k+1 < |s| \wedge 0 \leq j+1 \leq k+1)$

$\wedge_L cantApariciones(subseq(s, 0, k+1-(j+1)), 0) = i$

$\wedge (k+1-(j+1)=0 \vee_L s[k+1-(j+1)-1]=0)$

$\wedge cantApariciones(subseq(s, k+1-(j+1), k+1), 0) = 0$

$\equiv (0 \leq k+1 < |s| \wedge 0 \leq j+1 \leq k+1)$ ①

$\wedge_L cantApariciones(subseq(s, 0, k-j), 0) = i$ ②

$\wedge (k-j=0 \vee_L s[k-j-1]=0)$ ③

$\wedge cantApariciones(subseq(s, k-j, k+1), 0) = 0$ ④

② Y ③ SON IMPLICADOS TRIVIALMENTE POR P PUES SON EXACTAMENTE LOS MISMOS PREDICADOS.

① POR P: $k+1<|s| \wedge 0 \leq k < |s| \Rightarrow 0 \leq k+1 < |s|$

POR P: $0 \leq j \leq k \Rightarrow 1 \leq j+1 \leq k+1 \Rightarrow 0 \leq j+1 \leq k+1$

④ POR P: $cantApariciones(subseq(s, k-j, k), 0) = 0$

ADEMÁS ESTAMOS EN EL CASO $s[k] \neq 0$, ENTONCES SI INCLUÍMOS LA

POSICIÓN K EN LA SUBSEQ, VAMOS A SUMAR 0 APARICIONES DEL 0

PUES $S[K] \neq 0$

$\Rightarrow$ cantApariciones(subseq(s, k-1, k+1), 0) = 0 + 0 = 0

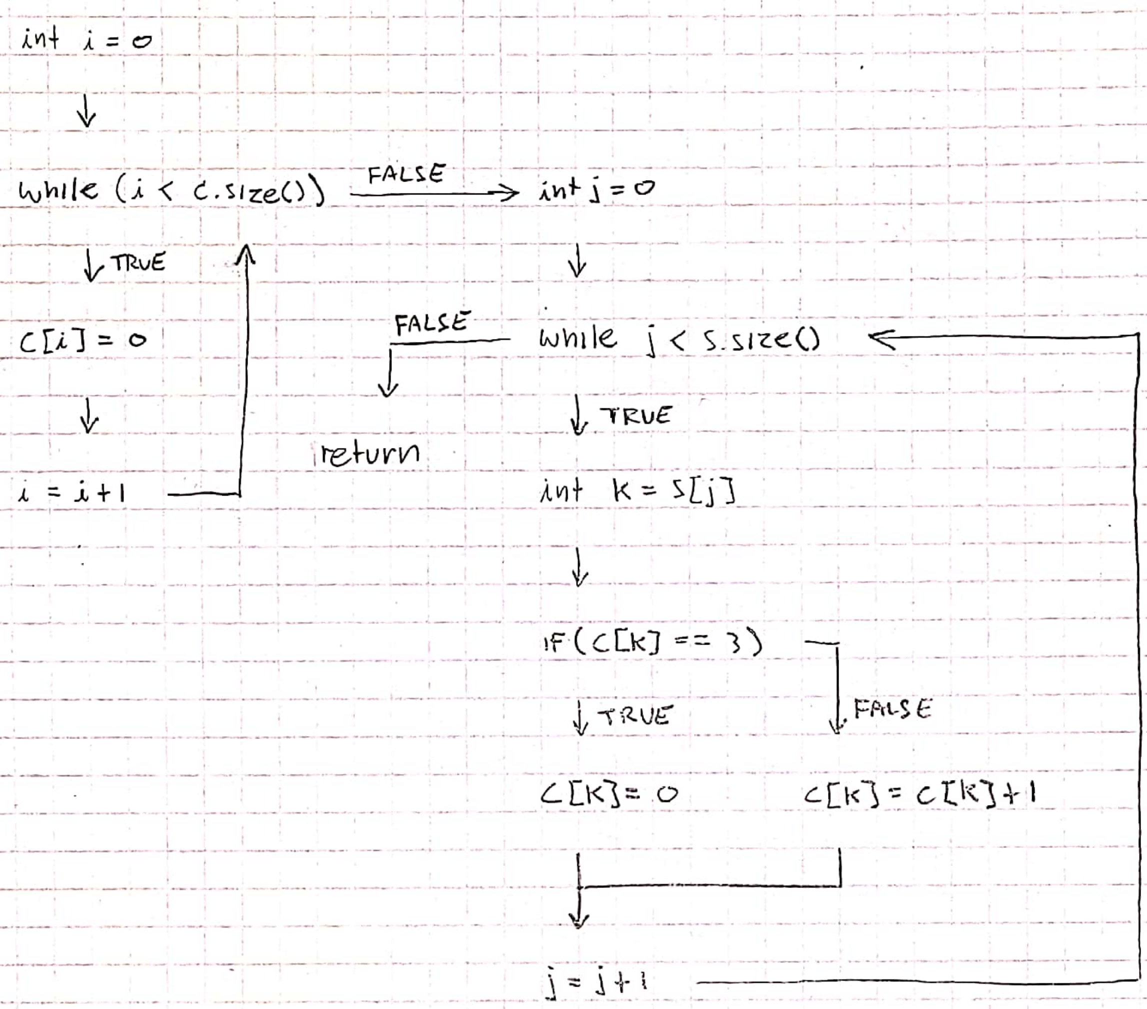
NOTA

EJERCICIO 4

JONATHAN BEKENSTEIN

348/11

2021-10-15



PREGUNTAS B C

test suite : Test 1 (Y ÚNICO)

ENTRADA: S = {0,0,0,0}

C = {0,0,0,0}

SALIDA : C = {1,0,0,0}

LÍNEAS CUBIERTAS : L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12

DECISIONES CUBIERTAS : L2-TRUE , L2-FALSE, L6-TRUE , L6-FALSE,

L8-TRUE , L8-FALSE

ESTE TEST CASE CAPTURA EL BUG. EL PROGRAMA DEVUELVE

C= {0,0,0,0} ≠ {1,0,0,0}

PREGUNTA D

SÍ, SE PUEDE

test:

ENTRADA : S = {0,0,0,0,0,0,0,0,0,0,0,0}

C = {0,0,0,0,0,0,0,0,0,0,0,0}

SALIDA : C = {0,0,0,0,0,0,0,0,0,0,0,0}

NOTA