CECILIA (1)
AGUAYSOL (a)

Para definir una relación de fuerza entre predicados α y β decimos que α es más fuerte que β si $\alpha \rightarrow \beta$ es una tautología es decir, si es siempre verdadero.
Aclarado esto, definimos:

esElUnico → enPosicionesPares → pertenece ✓

Notemos que si hay un único elemento, entonces necesariamente va a estar en una posición par pues el 0 es un número par. Sin embargo, basta dar una secuencia con longitud mayor a uno para demostrar que enPosicionesPares → esElUnico es una contingencia pues sólo será verdadero cuando la entrada tenga una longitud igual a 1. ✓
Por otro lado, pertenece es más débil que enPosicionesPares ya que si un elemento pertenece a una secuencia no podemos asegurar que se encuentre en una posición par. Por el contrario, si sabemos que un elemento se encuentra en una posición par entonces dicho elemento pertenece a la secuencia. ✓

(b) El programa1 se encarga de devolvernos el entero 0 independientemente de las entradas así que viendo lo que nos pide la postcondición tenemos que $\{0 \leq 0 < |l| \wedge_L l[0] = e\}$.
Para este caso, nos alcanza pedir enPosicionesPares

✓ Bien!

CECILIA

AGUAYSOL

(2)

(a)

```
pred maximoDeSeq (S: seq<Z>) {
  (∃i: Z)(0 ≤ i < |S| ∧_L (∀j: Z)(0 ≤ j < |S| →_L S[j] ≤ S[i]))
}

pred esCreciente (S: seq<Z>) {
  (∀i: Z)(0 ≤ i < |S| - 1 →_L S[i] < S[i+1])
}

pred esDecreciente (S: seq<Z>) {
  (∀i: Z)(0 ≤ i < |S| - 1 →_L S[i] > S[i+1])
}

pred noTieneRepetidos (S: seq<Z>) {
  (∀i, j: Z)((0 ≤ i < |S| ∧ i ≠ j) →_L S[i] ≠ S[j]
                                      ∧ 0 ≤ j < |S|
  ~~sonDistintos(S[i], S[j])~~)
}

pred sonDistintos (S: seq<Z>, t: seq<Z>) {
  (|S| ≠ |t|) ∨ ((∃i: Z)(0 ≤ i < |t| ∧_L ~~S[i] ≠ t[i]~~ S[i] ≠ t[i]))
}

pred esSubeBaja (S: seq<Z>) {
  |S| ≥ 3 ∧ (maximoDeSeq ≠ S[0]) ∧ (maximoDeSeq ≠ S[|S|-1])
∧ noTieneRepetidos(S) ∧ ~~esCreciente(subseq(S, 0, maximoDeSeq(S)))~~
(∃i: Z)((0 ≤ i < |S| ∧ S[i] = maximoDeSeq(S)) ∧_L (esDecreciente(subseq(S, i, |S|))
∧ esCreciente(subseq(S, 0, i+1))))
}
```

LO USO EN (2b)

Muy bien!

(6) proc armarSecuenciasSubeBaja (in s: seq<ℤ>, out res: seq<seq<ℤ>>){

Pre {esSubeBaja(s)} No se pedía esto. El ej s=<4,4,5,3,1> no es ↑↓ pq tiene reps

Post {|res| ≥ 1 ∧ (∀i:ℤ)(0 ≤ i < |res| →ₗ esSubeBaja(res[i]) ∧ tieneElemDeLaSeqOriginal(s, res[i])) ∧ noExistenMasSubeBaja(s, res) ∧ noTieneSecuenciasRepetidas(res)}

}

pred tieneElemDeLaSeqOriginal(s: seq<ℤ>, t: seq<ℤ>){

(∀i:ℤ)(0 ≤ i < |t| →ₗ cantApariciones(t, t[i]) ≤ cantApariciones(s, t[i]))

}

aux cantApariciones(s: seq<ℤ>, e: ℤ): ℤ = $\sum_{i=0}^{|s|-1}$ (if s[i]=e then 1 else 0 fi);

pred noExistenMasSubeBaja(s: seq<ℤ>, r: seq<seq<ℤ>>){

¬((∃t: seq<ℤ>)((esSubeBaja(t) ∧ tieneElemDeLaSeqOriginal(s, t)) ∧ₗ (∀j:ℤ)(0 ≤ j < |r| →ₗ r[j] ≠ t)))

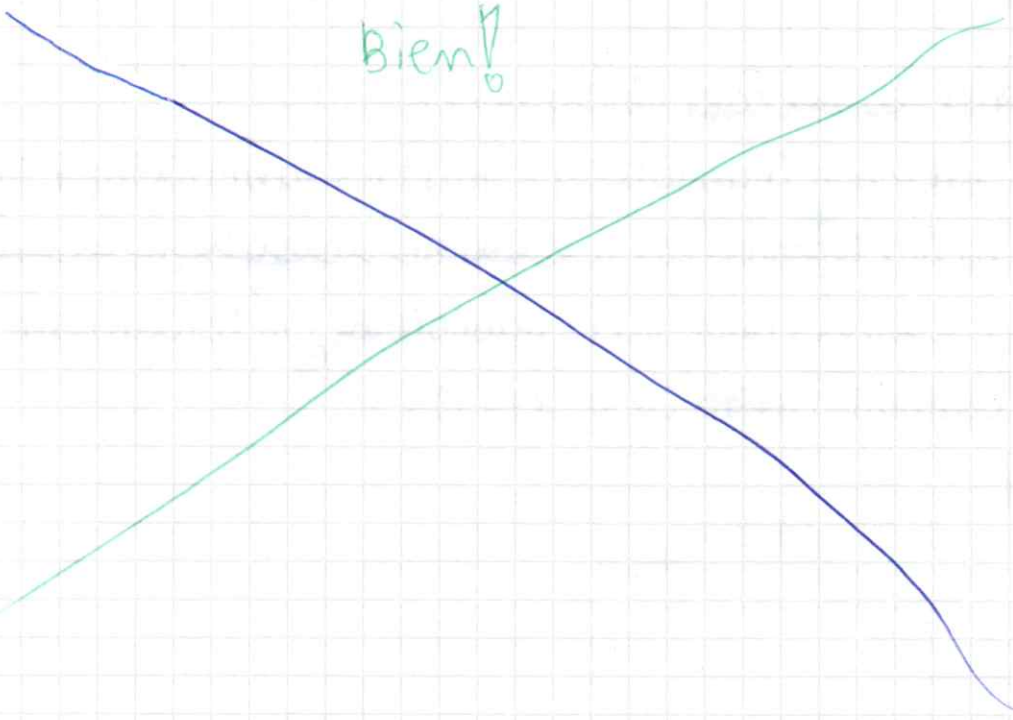
}

pred noTieneSecuenciasRepetidas(r: seq<seq<ℤ>>){

(∀i,j)((0 ≤ i < |r| ∧ 0 ≤ j < |r| ∧ i ≠ j) →ₗ sonDistintos(r[i], r[j]))

}

Bien!



CECILIA
AGUAYSOL

(3)

(a) pred esUnCorte (S: seq<ℤ>, C: (seq<ℤ> × seq<ℤ>)) {
(∀i: ℤ)(0 ≤ i < |S| →ₗ S[i] = (C[0] ++ C[1])[i])
}

✓

ojo es una tupla se accede C.y C.1

(b) proc corteMasParejo (in S: seq<ℤ>, out res: (seq<ℤ> × seq<ℤ>)) {
Pre {|S| > 0}

Post {esUnCorte(S, res) ∧ₗ ((suma(C[0]) - suma(C[1]) = 0) ∨ esLaDiferenciaMasChica(res, S))}
}

aux suma (C: seq<ℤ>): ℤ = $\sum_{i=0}^{|C|-1} C[i]$;

pred esLaDiferenciaMasChica (r: (seq<ℤ> × seq<ℤ>), S: seq<ℤ>) {

(∀t: (seq<ℤ> × seq<ℤ>)) (esUnCorte(S, t) →ₗ ~~suma~~ .
|suma(t[0]) - suma(t[1])| ≥ |suma(r[0]) - suma(r[1])|)
}

✓

Bien!