Algorimos sobre secuencias

15 de Junio de 2018

Ejercicio 1a

Dada la siguiente especificación:

proc armarPiramide (in v: $\mathbb{Z}$, inout l $seq\langle\mathbb{Z}\rangle$) {
Pre $\{l = L_0\}$
Post $\{length(L_0) = length(l) \wedge esPiramide(l, v)\}$
pred esPiramide (l: $seq\langle\mathbb{Z}\rangle$, v: $\mathbb{Z}$) {
$(\forall j : \mathbb{Z})(0 \leq j < length(l)/2 \Rightarrow_L l[j] = v + j) \wedge$
$(\forall j : \mathbb{Z})(length(l)/2 \leq j < length(l) \Rightarrow_L$
$l[j] = v + length(l) - j - 1)\}$
}

Dar un programa que satisfaga la especificación y tenga un ciclo con el siguiente invariante:

1. $length(l) = length(L_0) \wedge length(l)/2 \leq i \leq length(l) \wedge_L$
$((i = length(l)/2 \wedge l = L_0) \vee_L$
$(\exists p : seq\langle\mathbb{Z}\rangle)(length(p) = length(s) \wedge esPiramide(p, v) \wedge$
$subseq(p, length(l) - i, i) = subseq(length(i) - i, i)))$

Ejercicio 1b

Dada la siguiente especificación:

proc armarPiramide (in v: $\mathbb{Z}$, inout l $seq\langle\mathbb{Z}\rangle$) {
Pre $\{l = L_0\}$
Post $\{length(L_0) = length(l) \wedge esPiramide(l, v)\}$
pred esPiramide (l: $seq\langle\mathbb{Z}\rangle$, v: $\mathbb{Z}$) {
$(\forall j : \mathbb{Z})(0 \leq j < length(l)/2 \Rightarrow_L l[j] = v + j) \wedge$
$(\forall j : \mathbb{Z})(length(l)/2 \leq j < length(l) \Rightarrow_L$
$l[j] = v + length(l) - j - 1)\}$
}

Dar un programa que satisfaga la especificación y tenga un ciclo con el siguiente invariante:

2. $length(l) = length(L_0) \wedge 0 \leq i \leq$
$length(l) \wedge_L piramideHastaI(l, i, v)$
pred piramideHastaI (l: $seq\langle\mathbb{Z}\rangle$, i:$\mathbb{Z}$, v: $\mathbb{Z}$)
$\{(\exists p : seq\langle\mathbb{Z}\rangle)length(p) =$
$length(l) \wedge esPiramide(p, v) \wedge subseq(p, 0, i) = subseq(l, 0, i)\}$

Ejercicio 2

Dar un programa que satisfaga la siguiente especificación:

proc partirEnTres (in l: $seq\langle\mathbb{Z}\rangle$, out res: $<\mathbb{Z}, \mathbb{Z}>$) {
Pre $\{True\}$
Post $\{seParteEnTres(l, res) \vee noSeParteEnTres(l, res)\}$
pred seParteEnTres (l: $seq\langle\mathbb{Z}\rangle$, res: ¡$\mathbb{Z}$, $\mathbb{Z}$¿)
$\{(\exists i, j : \mathbb{Z})0 \leq i < j < length(l) \wedge_L \sum_{x=0}^{i-1} l[x] = \sum_{x=i}^{j-1} l[x] =$
$\sum_{x=j}^{length(l)-1} l[x] \wedge res_0 = i \wedge res_1 = j\}$
pred noSeParteEnTres (l: $seq\langle\mathbb{Z}\rangle$, res: ¡$\mathbb{Z}$, $\mathbb{Z}$¿)
$\{\neg(\exists i, j : \mathbb{Z})()0 \leq i < j < length(l) \wedge_L \sum_{x=0}^{i-1} l[x] =$
$\sum_{x=i}^{j-1} l[x] = \sum_{x=j}^{length(l)-1} l[x]) \wedge res_0 = -1 \wedge res_1 = -1\}$

Ejercicio 2

Dar un programa que satisfaga la siguiente especificación:

`proc partirEnTres` (in l: $seq\langle\mathbb{Z}\rangle$, out res: $<\mathbb{Z}, \mathbb{Z}>$) {

`Pre` $\{True\}$

`Post` $\{separteEnTres(l, res) \lor noSeParteEnTres(l, res)\}$

`pred seParteEnTres` (l: $seq\langle\mathbb{Z}\rangle$, res: ¡$\mathbb{Z}$, $\mathbb{Z}$¿)

$\{(\exists i, j : \mathbb{Z}) 0 \leq i < j < length(l) \land_L \sum_{x=0}^{i-1} l[x] = \sum_{x=i}^{j-1} l[x] = \sum_{x=j}^{length(l)-1} l[x] \land res_0 = i \land res_1 = j\}$

`pred noSeParteEnTres` (l: $seq\langle\mathbb{Z}\rangle$, res: ¡$\mathbb{Z}$, $\mathbb{Z}$¿)

$\{\neg(\exists i, j : \mathbb{Z})()0 \leq i < j < length(l) \land_L \sum_{x=0}^{i-1} l[x] = \sum_{x=i}^{j-1} l[x] = \sum_{x=j}^{length(l)-1} l[x]) \land res_0 = -1 \land res_1 = -1\}$

Es posible programarlo sin ciclos anidados?

}

Ejercicio 3

Dada una secuencia, definimos una *infrasecuencia* como una secuencia cuyos elementos son algunos de los elementos de la secuencia original, y aparecen en el mismo orden que en dicha secuencia.
Por ejemplo:
$\langle 0, 2, 4\rangle$ es infrasecuencia de $\langle 0, 1, 2, 3, 4\rangle$
$\langle 0, 0, 0, 1, 1, 1\rangle$ es infrasecuencia de $\langle 0, 0, 0, 1, 1, 1\rangle$
$\langle 0, 0, 0, 0\rangle$ no es infrasecuencia de $\langle 0, 0, 0, 1, 1, 1\rangle$ porque la secuencia no tiene cuatro ceros
$\langle 1, 0\rangle$ no es infrasecuencia de $\langle 0, 1\rangle$ porque no hay un 1 antes de un 0 en la secuencia original.
Especificar el problema *infrasecuencia* y dar un programa que satisfaga la especificación dada. Dar el invariante del ciclo principal.

Ejercicio 4

Dados dos strings, escribir un algoritmo que encuentre la subsecuencia de strings común más larga, es decir, la secuencia que es subsecuencia de ambas y que es de mayor longitud.

Ejercicio 5

Dada una secuencia de pares ⟨ string, int ⟩, donde el primer componente del par representa el nombre de una ciudad, y el segundo componente representa el índice de otra ciudad con la cual se conecta, dar algoritmos para:

1. Dada la secuencia y el nombre de una ciudad c, listar todas las ciudades que no son alcanzables desde c.
2. Indicar cuál es la ciudad a la cual llegan la mayor cantidad de rutas, recorriendo la secuencia una única vez.
3. Dada la secuencia y el nombre de una ciudad c, dar los nombres de todas las ciudades por las cuales se pasaría más de una vez si se comienza el recorrido en c.