

Práctica 4. Verificación en Dafny.

EDA. Grupos E y F.
Profesor: Isabel Pita.

Continuamos practicando la verificación de programas en Dafny. Dada una especificación en lenguaje natural o formal, el alumno debe realizar la especificación en Dafny si no está dada e implementar un método para resolver el problema. Luego verificará el programa escribiendo el invariante del bucle. Para terminar, escribirá el programa en C++, copiando el programa Dafny en un fichero `.cpp`, haciendo las siguientes modificaciones y lo subirá al juez en el problema que se indique en cada apartado:

- Incluye las librerías que vayan a hacer falta. Por lo menos `<iostream>`, `<fstream>` y `<vector>`
- Escribe la cabecera del programa en C++, poniendo en primer lugar el tipo de retorno de la función, a continuación el nombre y entre paréntesis los parámetros de entrada, poniendo el tipo antes que el identificador y cambiando `array` por `vector`. Si el método devuelve un vector, para evitar realizar la copia de todos los elementos se declara la función `void` y se devuelve el vector por referencia. Si el método devuelve mas de un valor, se declara un `struct` con todos los valores y se devuelve el tipo de este `struct`. No olvidar declarar el vector `const&` cuando sea de entrada y `&` cuando sea de salida.
- Comentar la precondition, postcondition e invariante del algoritmo.
- Cambiar el operador de asignación poniendo `=` en lugar de `:=`. Modificar la declaración de variables, poniendo el tipo en lugar de la palabra reservada `var`.
- Modificar el método que define la longitud del vector, poniendo `size()` en lugar de `Length`.
- Descomenta la parte correspondiente a la función `resuelveCaso` y `main` de la plantilla, donde se encuentra implementada la entrada de datos del programa.

Utiliza la plantilla `PlantillaInvariantes2.dfy` que se encuentra en el campus.

Problemas:

1. Implementa un programa que dado un vector `v` calcule sobre un vector diferente `sol` la suma acumulada de los elementos. Esto es `sol[n] == v[n]`, `sol[n-1] == v[n] + v[n-1]`, `sol[n-2] == v[n] + v[n-1] + v[n-2]`, y así sucesivamente, siendo `n` la última posición del vector.
 - a) Especifica el problema en Dafny
 - b) Resuelve el problema suponiendo que el vector de entrada tiene al menos un valor
 - c) Resuelve el problema suponiendo que el vector de entrada puede ser vacío.
 - d) Verifica cada uno de los problemas y súbelos al juez en el problema 04 (primero uno y cuando de correcto, el otro)
2. Implementa con un algoritmo iterativo y verifica una función que cumpla la siguiente especificación. Modificala y súbela al juez (todavía no está abierto el problema).

```
function Fib (n : int) : int
{  if (n < 2) then n else Fib(n-2) + Fib(n-1) }

method Fibonacci (n : int) returns (f : int)
requires n >= 0
ensures f == Fib(n)
```