



CI2691: Laboratorio de Algoritmos y Estructuras I

TALLER 05: Verificación de las aserciones correspondientes a los invariantes y función de cota de una iteración, mediante acciones insertadas en el código..

La verificación de la corrección de un ciclo requiere conocer el invariante que se satisface durante toda la ejecución del mismo, así como la función de cota que garantiza que dicho ciclo termina. Las pruebas formales que garantizan que un ciclo es correcto se estudian en el curso teórico “Algoritmos y Estructuras 1”. Para verificar de manera práctica que el invariante se satisface generalmente requiere obtener el valor de un cuantificador o de una función de agregación. Este cálculo corresponde a volver a ejecutar el ciclo que se está verificando.

Para chequear que efectivamente el ciclo satisface el invariante y que la función de cota termina, se usará como estrategia la visualización, en cada iteración, de la variables que intervienen en el cuerpo del ciclo. Asimismo, se chequeará a través de una instrucción condicional (if) si la función de cota es mayor o igual que cero y si es estrictamente decreciente.

El propósito del presente taller es aprender a verificar que las instrucciones iterativas son robustas mediante la inclusión en el código de acciones que permiten visualizar los valores de las variables y el chequeo de la finitud del ciclo.

En caso de violación de alguno de los requisitos de la función de cota, la acción a tomar es dar un mensaje informativo al usuario del error ocurrido y parar la ejecución del programa (mediante la instrucción “halt”), de manera tal que se garantice que no se cae en un ciclo infinito. Note que esto garantiza que las variables se observan sólo hasta la finalización correcta del ciclo o hasta que ocurra un halt.

Ejercicios: Para cada uno de los siguientes problemas proponga un programa que lo resuelva, escribiendo el invariante y la función de cota en cada ciclo utilizado. Coloque un WRITELN antes de entrar al ciclo para observar los valores de las variables que intervienen en el cuerpo del mismo. Dentro del ciclo, antes de finalizar, inserte acciones para verificar que la función de cota es mayor o igual que cero y que decrece, usando instrucciones condicionales (if). Además debe colocar una instrucción WRITELN que muestre el valor de las variables antes de terminar el ciclo.

1. Calcular la suma de los N primeros enteros. N es un valor de entrada al programa que es positivo. Para ello utilice un ciclo WHILE.
 - a. Obtenga la forma cerrada de la sumatoria para verificar que el invariante se satisface, usando un if. Este if debe colocarse antes de entrar al ciclo y antes de salir. No coloque en este caso las instrucciones WRITELN.
 - b. Escriba el mismo programa, pero esta vez eliminando el chequeo de invariante con la instrucción condicional y dejando sólo la visualización de las variables con el WRITELN.
2. Verificar si una secuencia de enteros no negativos provisto por el teclado está ordenada en forma creciente. La secuencia de entrada termina cuando se recibe el número 0 (que no es parte de la secuencia, sino una marca de fin). No se admite la secuencia vacía ni unitaria.

Observe que en este caso no se conoce el tamaño de la secuencia, por lo que para establecer una función de cota debe asumir un número máximo de iteraciones. Este número se asume muy grande para dar la posibilidad que el usuario introduzca una entrada lo suficientemente grande sin producir que el programa aborte
3. Dado un número natural n , calcular el número de Fibonacci f_n . Donde, se define que: $f_0=0, f_1=0$ y para $i > 1$ $f_i = f_{i-2} + f_{i-1}$. Utilice un ciclo FOR.

Observe que en este caso no se necesita determinar una función de cota pues el ciclo FOR de Pascal siempre termina. Aquí basta con visualizar las variables del ciclo para verificar que el ciclo funciona bien.
4. Determinar si un número entero positivo dado es primo.
5. Calcular el factorial de N .
6. Calculo del valor aproximado de e , usando el método del Quiz 3.

Realicen los siguientes experimentos, para observar lo que ocurre:

1. ejecuten sus programas con datos de entrada incorrectos,
2. modifiquen ligeramente algunas de las acciones (introduzcan errores adrede dentro del código), como por ejemplo, no incrementar la variable de control del ciclo, cambiar la condición de parada del ciclo, cambiar el cálculo del resultado principal del ciclo.
3. corran sus programas con datos correctos.