

Verificacion

Danny Rodriguez

March 2020

1 Especificación de un algoritmo

Estado: permite describir las condiciones que deben reunir los datos de entrada, los resultados, etc ($y_i=0$)

Estado= Aplicación de las variables en el conjunto de sus valores ($x=9$)

Aserto={ conjunto de estados } = $\{x = 9 \wedge y \leq 0 \wedge x \geq y\}$

Permiten especificar el comportamiento que se espera del algoritmo.

Permiten calcular su comportamiento real

Documentan el programa.

2 Notación de Hoare

{P} Precondiciones

S Sentencias

{Q} Postcondiciones

3 Verificación del algoritmo

Demostración formal de que el algoritmo cumple su especificación

Corrección Parcial: Si el algoritmo acaba entonces es correcto.

Base de comprobación

Demostrar que se cumple la postcondición:

Anotar al inicio y al final del programa las precondiciones y postcondiciones.

Incluir asertos en los puntos intermedios del programa que describan el estado en ese punto.

Demostrar que si se cumple un aserto en un punto del programa y se siguen cada una de las líneas de ejecución posible hasta llegar a otro aserto, dicho aserto ha de cumplirse (aplicando las leyes de la lógica y de acuerdo a las acciones realizadas por el programa)

Corrección Total: Para todo dato de entrada válido, el algoritmo acaba y el

resultado es correcto.

Base de comprobación

Corrección parcial + Demostración de que todos los bucles acaban en un número finito de pasos (función monótona con valor acotado)

4 Propiedades

$A_1 \Rightarrow A_2$ A_1 es un aserto más fuerte que A_2

A_1 es un subconjunto de A_2

Supongamos que esta especificación es correcta:

$\{x \leq 5\}$

S

$\{x \leq 10\}$

También será correcta si buscamos otro aserto inicial $R : \{x \leq 0\}$ que implique la precondition $R \Rightarrow P$

También será correcta si buscamos otro aserto final $T : \{x \leq 45\}$ que sea implicado por la postcondición $Q \Rightarrow T$

5 Reglas de Consecuencia

Primera Regla de Consecuencia (Reforzar la precondition)

$\{P\}S\{Q\} \wedge R \Rightarrow P$ entonces

$\{R\}S\{Q\}$

Segunda Regla de Consecuencia (Debilitar la postcondición)

$\{P\}S\{Q\} \wedge Q \Rightarrow R$ entonces

$\{P\}S\{R\}$

6 Axioma de la Asignación

$x \leftarrow E$

Q_E^x Aserto resultante de sustituir toda aparición de x por E

Axioma: La siguiente especificación es correcta

$\{Q_E^x\}$

$x \leftarrow E$

$\{Q\}$

7 Regla de inferencia de la Asignación

Axioma de asignación + Primera regla de consecuencia

Para que sea correcto:

$\{P\}$

S

$\{Q\}$

Se obtiene una especificación correcta aplicando el axioma de asignación

$\{Q_E^x\}$

$x \leftarrow E$

$\{Q\}$

Debe cumplirse que $P \Rightarrow Q_E^x$ aplicando la **Primera regla de consecuencia**

8 Composición Secuencial

La especificación

$\{P\}$

S_1

S_2

$\{Q\}$

Es correcta si encontramos un aserto R tal que se cumple que

a) $\{P\}S_1\{R\}$

b) $\{R\}S_2\{Q\}$

9 Esquema de selección: Selectiva Simple

La especificación

$\{P\}$

Si B entonces

S

fin_si

$\{Q\}$

Es correcta si satisface las condiciones de verificación

a.1) $P \wedge \neg B \Rightarrow Q$

a.2) $\{P \wedge B\}S\{Q\}$

10 Esquema de selección: Selectiva Doble

La especificación

$\{P\}$

Si B entonces

S_1

si_no
 S_2
 fin_si
 $\{Q\}$
 Es correcta si satisface las condiciones de verificación
 $a.1)\{P \wedge B\}S_1\{Q\}$
 $a.2)\{P \wedge \neg B\}S_2\{Q\}$

11 Esquema de selección múltiple

La especificación
 $\{P\}$
 Segun sea B hacer
 $B_1 : S_1$
 $B_2 : S_2$
 ...
 $B_n : S_n$
 fin_segun
 $\{Q\}$
 Es correcta si satisface las condiciones de verificación
 $a.1)P \Rightarrow B_1 \vee B_2 \vee \dots \vee B_n$
 $a.2)P \Rightarrow \forall i, j : 1 \leq i, j \leq n \wedge i \neq j \Rightarrow \neg(B_i \wedge B_j)$
 $a.3)\forall i : 1 \leq i \leq n : \{P \wedge B_i\}S_i\{Q\}$

12 Razonamiento sobre bucles

$\{P\}$
 Mientras B hacer $\{I\}$
 S
 fin_mientras
 $\{Q\}$
 INVARIANTE: Conjunto de condiciones lógicas que se cumplen antes, durante y después de la ejecución del bucle.
 La especificación será parcialmente correcta si:
 $a.1)P \Rightarrow I$
 $a.2)I \wedge \neg B \Rightarrow Q$
 $a.3)\{I \wedge B\}S\{I\}$
Corrección Total
 Encontrar una función monótona decreciente que garantice que el algoritmo acaba en un número finito de pasos, alcanzando la cota 0.
 $a.4)I \wedge B \Rightarrow t > 0$
 $a.5)\{I \wedge B \wedge t = T\}S\{t < T\}$

13 El Invariante

El invariante es un predicado que describe todos los estados por los que atraviesa el cómputo realizado por un bucle, observados justo antes de evaluar la condición de terminación.

El invariante se satisface antes de la primera iteración, después de cada una de ellas y, después de la última.

14 Búsqueda de Invariantes

1. Elaborar una traza del algoritmo.
2. Debe ser parecido a la postcondición del bucle, ya que el invariante debe cumplirse antes, durante y justo después del bucle.
3. En cada iteración el invariante debe acercarse a la postcondición del bucle, por tanto debe contener variables que se modifican dentro del bucle.
4. NUNCA usar expresiones del tipo $x=x+b$.