



DISEÑO DE ALGORITMOS ITERATIVOS

POR NEREA JIMÉNEZ GONZÁLEZ

VERIFICACIÓN

Semántica: encontrar predicados $\{P\}A\{Q\}$.

Verificar: $\{P=R_0\}A\{R_1\};\dots\{R_{n-1}\}A\{R_n=Q\}$ Si satisface $\{R_{k-1}\}A\{R_k\}$ para todo $k ==$ satisface $\{P\}A\{Q\}$

Si $\{P\}A\{Q\}$ y $P' \Rightarrow P$ entonces $\{P'\}A\{Q\}$. Fortalecimiento de la precondition.

Si $Q=Q' \Rightarrow \{P\}A\{Q'\}$. Debilitamiento de la postcondición.

Si $\{P\}A\{Q_1\}$ y $\{P\}A\{Q_2\} \Rightarrow \{P\}A\{Q_1 \text{ y } Q_2\}$. Conjunción en la postcondición.

Si $\{P_1\}A\{Q\}$ y $\{P_2\}A\{Q\} \Rightarrow \{P_1 \text{ o } P_2\}A\{Q\}$. Conjunción en la postcondición.

Como la precondition de una verificación correcta se puede fortalecer

\Rightarrow para demostrar la corrección de $\{P\}A\{Q\}$ es suficiente con

demostrar $\{R\}A\{Q\}$ donde R es la condición más débil de A con

respecto a Q , $\text{pmd}(A,Q)$ y ver que $P \Rightarrow R$

INVARIANTES

Para la verificación del bucle necesitamos un invariante I , que describe los estados por los que pasa el bucle.

I se satisface antes de ejecutarse el bucle ($P \Rightarrow I$), en cada iteración ($\{I \wedge B\}A\{I\}$) y cuando finaliza este ($I \wedge \neg B \Rightarrow Q$).

En cada iteración el valor de variables se modifica pero no su relación.

Cota: estado $\Rightarrow Z$. Depende de las variables del cuerpo del bucle. Es mayor o igual que 0 cuando B se cumple: $I \wedge B \Rightarrow \text{cota} \geq 0$.

Decrece al ejecutarse el bucle: $\{I \wedge B \wedge \text{cota} = T\}A\{\text{cota} < T\}$ donde T es una constante.

DERIVACIÓN

Derivar es construir las instrucciones a partir de su especificación asegurando su corrección.

La postcondición dirige el proceso de verificación.

Si la precondition es más fuerte que el predicado más débil y la postcondición, \Rightarrow Proceso finalizado

En caso contrario, instrucciones iterativas:

```

{P}
A0(inicialización)
{I,Cota}
while(B){
  {I ∧ B}
  A1(Restablecer)
  {R}
  A2 (avanzar)
  {I}
}
{Q}

```

donde A_0 instrucción que hace que el invariante se cumpla inicialmente,

A_1 mantiene el invariante a cierto, A_2 hace que la cota decrezca.

REGLAS PRÁCTICAS

Instrucciones de asignación expresiones aritméticas y lógicas, etc... coste constante, $O(1)$.
Return no se cuenta.

$S_1; S_2$ su coste es
coste S_1 + coste S_2 .
 $O(f_1(n)) + O(f_2(n)) =$
 $O(f_1(n) + f_2(n)) =$
 $O(\max(f_1(n), f_2(n)))$

Instrucción condicional:
 $\text{if}(B)\{S_1\}\text{else}\{S_2\}$
caso peor:
 $O(\max(f_B(n), f_1(n), f_2(n)))$

$\text{While}(B)\{S\}$
Calcular coste de cada vuelta, sumar los costes de todas las vueltas. El número de iteraciones depende de B .

Pasos para construir un **ALGORITMO CON BUCLE**

1

Diseñas el invariante y la condición del bucle sabiendo que se tiene que cumplir: $I \wedge \neg B \implies Q$

2

Diseñar A_0 para hacer el invariante cierto: $\{P\}A_0\{I\}$

3

Diseñar la función cota, C , de tal forma que: $I \wedge B \implies C > 0$

4

Diseñar A_2 y el predicado $R = \text{pmd}(A_2, I)$

5

Diseñar A_1 para que se cumpla $\{I \wedge B\}A_1\{R\}$

6

Comprobar que la cota realmente decrece: $\{I \wedge B \wedge C = T\}A_1; A_2\{C < T\}$