Introducción a la Computación (Matemática)

Primer Cuatrimestre de 2019

Verificación Formal de Algoritmos

Verificación formal de algoritmos

Terna de Hoare

{precondición}	$\{P\}$
algoritmo	S
{poscondición}	$\{Q\}$

Una forma de probar que S es correcto consiste en probar que:

$$sp(S, P) \Rightarrow Q$$

donde $sp(S, P)$ (*strongest postcondition*) es el predicado más fuerte que resulta de ejecutar S a partir del estado P.

Asignación

$\{P\}\ x \leftarrow E\ \{Q\}$

Queremos probar que $sp(x \leftarrow E,\ P) \Rightarrow Q$.

Definición:

$$sp(x \leftarrow E,\ P) \equiv (\exists v)\ x = E[x : v] \wedge P[x : v]$$

donde:

- v es una variable no usada;
- $H[x : E]$ es la sustitución de cada instancia en H de la variable x por la expresión E.
  Ejemplo: $(x + y)[y : z^2 + 1] = (x + z^2 + 1)$.

Secuencialización

$\{P\}\ S_1;S_2\ \{Q\}$

Queremos probar que $sp(S_1;S_2,\ P) \Rightarrow Q$.

Definición:

$$sp(S_1;S_2,\ P) \equiv sp(S_2, sp(S_1, P))$$

Condicional

$\{P\}$ if (B) S_1 else S_2 $\{Q\}$

Queremos probar que $sp(\text{if } (B)\ S_1 \text{ else } S_2,\ P) \Rightarrow Q$.

Definición:

$$sp(\text{if } (B)\ S_1 \text{ else } S_2,\ P) \equiv sp(S_1, B \land P) \lor sp(S_2, \neg B \land P)$$

Ciclo

$\{P\}$ while (B) S $\{Q\}$

Queremos probar que $sp(\text{while } (B)\ S,\ P) \Rightarrow Q$.

$$sp(\text{while } (B)\ S,\ P) \equiv \,?$$

Esta forma de encarar la demostración de que un ciclo es correcto es muy complicada. Mejor veamos otra forma.

Encabezado: $sumatoria : x \in \mathbb{Z} \rightarrow \mathbb{Z}$
Precondición: $\{x = x_0 \wedge x_0 \geq 0\}$
Poscondición: $\{RV = \sum_{1 \leq j \leq x_0} j\}$

$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ {
    // estado e1
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
    // estado e2
}
donde $i \in \mathbb{Z}$.

Estados para $x_0 = 6$:

e1		e2	
RV	i	RV	i
0	1	1	2
1	2	3	3
3	3	6	4
6	4	10	5
10	5	15	6
15	6	21	7

Obs#1: $1 \leq i \leq x + 1$ y $RV = \sum_{1 \leq j < i} j$ valen en cada paso en e1 y en e2, pero no en el medio.
Obs#2: Cuando $i = x + 1$, la ejecución del ciclo termina.

Invariante

Predicado que describe la *idea* de un ciclo.

- Vale justo antes del ciclo (antes de evaluar B por primera vez).
- Vale en cada iteración:
  - justo antes de ejecutar la primera instrucción de S, y
  - justo después de ejecutar la última instrucción de S;
  - pero no vale durante la ejecución de S.
- Vale justo después del ciclo (después de evaluar B por última vez).
- No se conoce una forma algorítmica de encontrarlo.

Ejemplo de Invariante

$RV \leftarrow 0$
$i \leftarrow 1$
$//$ vale I
while $(i \leq x)$ {
    $//$ vale $I \wedge B$
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
    $//$ vale I
}
$//$ vale $I \wedge \neg B$

$$
\begin{aligned}
I \;\equiv\; & 1 \leq i \leq x + 1 \;\wedge \\
& RV = \textstyle\sum_{1 \leq j < i} j \;\wedge \\
& x = x_0 \wedge x_0 \geq 0
\end{aligned}
$$

Otro ejemplo de Invariante

Enc: $creciente : A \in \mathbb{Z}[\,] \to bool$
Pre: $\{A = A_0 \wedge |A_0| > 0\}$
Pos: $\{RV = true \Leftrightarrow \big((\forall j)\ 0 \leq j < |A_0| - 1 \Rightarrow A_0[j] < A_0[j+1]\big)\}$

$i \leftarrow 0$
// vale I
while $(i < |A| - 1\ \wedge\ A[i] < A[i+1])$ {
    // vale $I \wedge B$
    $i \leftarrow i + 1$
    // vale I
}
// vale $I \wedge \neg B$
$RV \leftarrow (i = |A| - 1)$

$$I \equiv 0 \leq i \leq |A_0| - 1 \ \wedge\ \big((\forall j)\ 0 \leq j < i \Rightarrow A_0[j] < A_0[j+1]\big) \wedge A = A_0 \ \wedge\ |A_0| > 0$$

¿Cómo probamos que un ciclo termina?

Acotamos superiormente la cantidad de iteraciones del ciclo mediante una función variante (fv).

Es una función del lenguaje de especificación, de tipo $\mathbb{Z}$:

- definida a partir de las variables del programa;
- debe decrecer estrictamente en cada iteración del ciclo.

Damos una cota (c) (valor entero fijo) tal que cuando fv la alcanza, se niega la guarda y termina la ejecución del ciclo.

Ejemplo de función variante y cota

$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ $\{$
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
$\}$

$fv = x - i$
$c = -1$

En cada iteración del ciclo, i se incrementa y x no se modifica. Por lo tanto, $x - i$ es estrictamente decreciente.

Cuando $x - i \leq c$, se niega la guarda y termina el ciclo.

Otro ejemplo de función variante y cota

$i \leftarrow 0$
while $(i < |A| - 1 \ \wedge \ A[i] < A[i+1])$ {
    $i \leftarrow i + 1$
}
$RV \leftarrow (i = |A| - 1)$

$fv = |A| - 1 - i$
$c = 0$

En cada iteración del ciclo, i se incrementa y $|A|$ no se modifica.
Por lo tanto, $|A| - 1 - i$ es estrictamente decreciente.

Cuando $|A| - 1 - i \leq c$, se niega la guarda y termina el ciclo.

Obs: El ciclo quizá termine *antes*, cuando $A[i] \geq A[i+1]$.
Sólo nos interesa *acotar* la cantidad de iteraciones.

Teorema de terminación

Sean I el invariante de un ciclo con guarda B, fv una función entera estrictamente decreciente y $c \in \mathbb{Z}$, tales que $(I \wedge fv \leq c) \Rightarrow \neg B$. Entonces el ciclo termina.

Demostración
Sea fv_j el valor que toma fv luego de ejecutar el cuerpo del ciclo por j-ésima vez. Dado que $fv \in \mathbb{Z}$, $fv_j \in \mathbb{Z}$ para todo j.

Como fv es estrictamente decreciente, $fv_0 > fv_1 > fv_2 > \dots$, y necesariamente existe un k tal que $fv_k \leq c$.

Dado que $(I \wedge fv \leq c) \Rightarrow \neg B$, luego de k iteraciones vale $\neg B$. Por lo tanto el ciclo termina. $\square$

Teorema de correctitud de ciclos

Sea while (B) S un ciclo con precondición P, poscondición Q, invariante I, función variante fv y cota c. Si valen:

1. $P \Rightarrow I$
2. $\{I \wedge B\}\ S\ \{I\}$ es correcto
3. fv es estrictamente decreciente
4. $(I \wedge fv \leq c) \Rightarrow \neg B$
5. $(I \wedge \neg B) \Rightarrow Q$

entonces el ciclo es correcto para su especificación (P, Q).

Dem: Por hip. 1, si suponemos que antes del ciclo vale P, también vale I. Si la valuación de B es verdadera, se ejecuta S. Por hip. 2, al concluir S sigue valiendo I, y se vuelve a evaluar B. Por hip. 3 y 4 y el teorema de terminación, sabemos que este proceso ocurre un número finito de veces. Al final de cada ejecución de S sigue valiendo I. Cuando finalmente B se niega, tenemos $(I \wedge \neg B)$, lo cual por hip. 5 implica Q. $\square$

Ejemplo completo: 1. $P \Rightarrow I$

Pre: $\{x = x_0 \wedge x_0 \geq 0\}$
$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ $\{$
$\quad RV \leftarrow RV + i$
$\quad i \leftarrow i + 1$
$\}$
Pos: $\{RV = \sum_{1 \leq j \leq x_0} j\}$

$$I \equiv 1 \leq i \leq x + 1 \wedge RV = \sum_{1 \leq j < i} j \wedge x = x_0 \wedge x_0 \geq 0$$

Queremos ver que $sp(i \leftarrow 1, sp(RV \leftarrow 0, Pre)) \Rightarrow I$.

$sp(i \leftarrow 1, sp(RV \leftarrow 0, Pre)) \equiv \ldots$
$\equiv (RV = 0 \wedge i = 1 \wedge x = x_0 \wedge x_0 \geq 0)$

$i = 1 \wedge x = x_0 \wedge x_0 \geq 0$ implica $1 \leq i \leq x + 1$.

$RV = 0 \wedge i = 1$ implica $RV = \sum_{1 \leq j < i} j$. $\square$

Ejemplo completo: 2. $\{I \wedge B\}\ S\ \{I\}$

$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ {
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
}

$$I \equiv 1 \leq i \leq x + 1 \wedge RV = \sum_{1 \leq j < i} j \wedge x = x_0 \wedge x_0 \geq 0$$

$sp(RV \leftarrow RV + 1; i \leftarrow i + 1,\ I \wedge B) \equiv$
$\equiv sp(i \leftarrow i + 1, sp(RV \leftarrow RV + i,\ I \wedge B)) \equiv \ldots \equiv$
$\equiv (\exists b)\ i = b + 1\ \wedge\ (\exists a)\ RV = a + b\ \wedge\ 1 \leq b \leq x + 1\ \wedge$
$a = \sum_{1 \leq j < b} j\ \wedge\ x = x_0\ \wedge\ x_0 \geq 0\ \wedge\ b \leq x$

$i = b + 1 \wedge 1 \leq b \leq x + 1 \wedge b \leq x$ implica $1 \leq i \leq x + 1$.

$i = b + 1 \wedge RV = a + b \wedge a = \sum_{1 \leq j < b} j$ implica $RV = \sum_{1 \leq j < i} j$. $\square$

Ejemplo completo: 3. *fv* es estrict. decreciente

$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ $\{$
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
$\}$

$fv = x - i$

En cada iteración del ciclo, i se incrementa y x no se modifica.
Por lo tanto, $x - i$ es estrictamente decreciente. $\square$

Ejemplo completo: 4. $(I \wedge fv \leq c) \Rightarrow \neg B$

$$
\begin{array}{l}
RV \leftarrow 0 \\
i \leftarrow 1 \\
\text{while } (i \leq x) \ \{ \\
\qquad RV \leftarrow RV + i \\
\qquad i \leftarrow i + 1 \\
\}
\end{array}
$$

$fv = x - i$

$c = -1$

$x - i \leq -1$ implica $x \leq i - 1$, o sea $x < i$, lo cual equivale a $\neg B$. $\square$

Obs: En este caso I no fue necesaria para la demostración, pero en el caso general sí hace falta.

Ejemplo completo: 5. $(I \wedge \neg B) \Rightarrow Q$

Pre: $\{x = x_0 \wedge x_0 \geq 0\}$
$RV \leftarrow 0$
$i \leftarrow 1$
while $(i \leq x)$ {
    $RV \leftarrow RV + i$
    $i \leftarrow i + 1$
}
Pos: $\{RV = \sum_{1 \leq j \leq x_0} j\}$

$$I \equiv 1 \leq i \leq x + 1 \wedge RV = \sum_{1 \leq j < i} j \wedge x = x_0 \wedge x_0 \geq 0$$

Queremos ver que $I \wedge \neg B \Rightarrow$ Pos.

$I \wedge \neg B \equiv 1 \leq i \leq x + 1 \wedge RV = \sum_{1 \leq j < i} j \wedge x = x_0 \wedge x_0 \geq 0 \wedge i > x$

$1 \leq i \leq x + 1 \wedge i > x \wedge x = x_0$ implica $i = x_0 + 1$.

$i = x_0 + 1 \wedge RV = \sum_{1 \leq j < i} j$ implica $RV = \sum_{1 \leq j \leq x_0} j$. $\square$

Ejemplo completo: Grande Finale

Ya probamos los siguientes puntos:

1. $P \Rightarrow I$
2. $\{I \wedge B\}\ S\ \{I\}$
3. fv es estrictamente decreciente
4. $(I \wedge fv \leq c) \Rightarrow \neg B$
5. $(I \wedge \neg B) \Rightarrow Q$

Por lo tanto, según el teorema de correctitud de ciclos, el algoritmo dado es correcto respecto de la especificación. $\square$

Repaso de la clase de hoy

- Verificación formal de ciclos.
- Invariante, función variante.
- Teoremas de terminación y correctitud de ciclos.