Resolución Ejercicio 3.c) Práctica 4

Guía práctica de correctitud de algoritmos

1^{er} cuatrimestre 2019

Ejercicio 1. Demostrar que cada uno de los siguientes algoritmos es correcto respecto de su especificación.

Encabezado: *shiftRight:* $A \in \mathbb{Z}[] \to \emptyset$ (Modifica a A)
Precondición: $\{A = A_0 \ \wedge \ |A| > 0\}$
Poscondición: $\{|A| > 0 \wedge |A| = |A_0| \wedge A[0] = A_0[|A| - 1] \wedge \forall i (1 \leq i < |A| \to A[i] = A_0[i-1])\}$
Variables aux.: $i, previousValue, tmp \in \mathbb{Z}$
Algoritmo:

$previousValue \leftarrow A[|A| - 1]$
$i \leftarrow 0$
while ($i < |A|$) {
  $tmp \leftarrow A[i]$
  $A[i] \leftarrow previousValue$
  $previousValue \leftarrow tmp$
  $i \leftarrow i + 1$
}

Algunas sugerencias:

- Tener en cuenta la idea que propone el siguiente ejemplo sobre la asignación en arreglos:

  Sea $P : \{A = A_0\}$
  Se tiene $S = A[k] \leftarrow e$ (para algún $0 \leq k < |A|$ y alguna expresión e)

  Se observa que:
  $P \equiv (\forall j)\ (0 \leq j < |A| \to A[j] = A_0[j])$
  $\equiv (\forall j)\ (0 \leq j < k \to A[j] = A_0[j])$
  $\wedge\ A[k] = A_0[k]$
  $\wedge\ (\forall j)(\ k < j < |A| \to A[j] = A_0[j])$

  Luego,
  $sp(A[k] \leftarrow e, P) \equiv \big(\exists x\big)\big(A[k] = e[A[k] : x]\ \wedge P[A[k] : x]\big)$
  $\equiv \Big(\exists x\Big)\Big(A[k] = e[A[k] : x]\ \wedge \big((\forall j)\ (0 \leq j < k \to A[j] = A_0[j])$
  $\wedge\ A[k] = A_0[k]\ \wedge\ (\forall j)\ (k < j < |A| \to A[j] = A_0[j])\big)\big[A[k] : x\big]\Big)$

  $\equiv \big(\exists x\big)\big(A[k] = e[A[k] : x]\ \wedge (\forall j)\ (0 \leq j < k \to A[j] = A_0[j])$
  $\wedge\ x = A_0[k]\ \wedge\ (\forall j)\ (k < j < |A| \to A[j] = A_0[j])\big)$

- En el invariante puede ser útil decir también las cosas que no cambian.

- Para el invariante, considerar $previousValue = A_0[(i-1)\ \%|A|]$.

Resolución. Iremos siguiendo las instrucciones del algoritmo y viendo los estados alcanzados. En los casos triviales, simplemente escribiremos los estados resultantes de la aplicación de una instrucción sin escribir explícitamente todo el desarrollo del cálculo de la SP. Al ejecutar la primera instrucción partiendo de la precondición del algoritmo (a la que nos referiremos como P_0) arribamos al siguiente estado:

$$SP(previousValue \leftarrow A[|A|-1], P_0) \equiv \{previousValue = A[|A|-1] \wedge A = A_0 \wedge |A| > 0\} \equiv P_1$$

Veamos ahora el estado alcanzado al hacer $i \leftarrow 0$ desde lo que llamamos P_1:

$$SP(i \leftarrow 0, P_1) \equiv \{i = 0 \wedge previousValue = A[|A|-1] \wedge A = A_0 \wedge |A| > 0\} \equiv P_2$$

Para probar que el ciclo es correcto usaremos el Teorema de Correctitud de Ciclos. Tomaremos como función variante y respectiva cota a

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

Tomaremos el siguiente invariante:

$$I \equiv \Big(0 \leq i \leq |A| \quad \wedge \quad |A| > 0 \quad \wedge \quad |A| = |A_0| \quad \wedge \quad previousValue = A_0[(i-1)\,\%|A|] \quad \wedge$$
$$\big(\forall j\big)\ \big(0 \leq j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\big) \quad \wedge \quad \big(\forall j\big)\ \big(i \leq j < |A| \rightarrow A[j] = A_0[j]\big)\Big)$$

Veamos que con I, fv, y c se verifican las hipótesis del Teorema de Correctitud de Ciclos.

1. "$P \Rightarrow I$"

   En este caso P corresponde a P_2.

   - $i = 0 \wedge |A| > 0 \quad \Rightarrow \quad 0 \leq i \leq |A| \quad \checkmark$
   - $|A| > 0 \quad \Rightarrow \quad |A| > 0 \quad \checkmark$
   - $A = A_0 \quad \Rightarrow \quad |A| = |A_0| \quad \checkmark$
   - $i = 0 \wedge |A| > 0 \quad \Rightarrow \quad (i-1)\,\%|A| = |A| - 1$
     $\therefore previousValue = A[|A|-1] \quad \Rightarrow \quad previousValue = A[(i-1)\,\%|A|] \quad \checkmark$
   - $i = 0 \quad \Rightarrow \quad \Big((\forall j)\ 0 \leq j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \quad \checkmark$
   - $i = 0 \wedge A = A_0 \quad \equiv \quad i = 0 \wedge (\forall j)\ (0 \leq j < |A| \rightarrow A[j] = A_0[j])$
     $\Rightarrow \Big((\forall j)\ i \leq j < |A| \rightarrow A[j] = A_0[j]\Big) \quad \checkmark$

2. *fv es estrictamente decreciente*

   Como i se incrementa en cada iteración del ciclo, $fv = |A| - i$ es estrictamente decreciente $\checkmark$

3. $(I \wedge fv \leq c) \Rightarrow \neg B$

   $|A| - i \leq 0 \Rightarrow |A| \leq i \Rightarrow \neg(i < |A|) \quad \checkmark$

4. $(I \wedge \neg B) \Rightarrow Q$

   En este caso Q corresponde a la poscondición del algoritmo.

   - $|A| = |A_0| \quad \Rightarrow \quad |A| = |A_0| \quad \checkmark$
   - $|A| > 0 \quad \Rightarrow \quad |A| > 0 \quad \checkmark$
   - $0 \leq i \leq |A| \wedge i \geq |A| \quad \Rightarrow \quad i = |A|$

     $i = |A| \wedge \Big((\forall j)\ 0 \leq j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big)$
     $\Rightarrow \quad \Big((\forall j)\ 0 \leq j < |A| \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big)$

$$\Rightarrow \quad A[0] = A_0[-1\,\%|A|] \quad \wedge \quad \Big((\forall j)\ 1 \le j < |A| \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big)$$
$$\Rightarrow \quad A[0] = A_0[|A|-1] \quad \wedge \quad \Big((\forall j)\ 1 \le j < |A| \rightarrow \quad A[j] = A_0[j-1]\Big) \ \checkmark$$

5. $\{I \wedge B\}\ S\ \{I\}$

   Comencemos por ver qué pasa con la instrucción $tmp \leftarrow A[i]$:

   $SP(tmp \leftarrow A[i], B \wedge I)$
   $\equiv tmp = A[i] \ \wedge B \wedge I$
   $\equiv P_3$

   Calculemos ahora el estado que alcanzamos desde P_3 con la instrucción $A[i] \leftarrow previousValue$:

   $SP(A[i] \leftarrow previousValue, P_3)$
   $\equiv \quad (\exists y) \quad A[i] = previousValue \ \wedge\ tmp = y \ \wedge\ i < |A| \ \wedge\ 0 \le i \le |A| \ \wedge\ |A| > 0 \ \wedge$
   $|A| = |A_0| \ \wedge\ \Big((\forall j)\ 0 \le j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ y = A_0[i] \ \wedge$
   $\Big((\forall j)\ i < j < |A| \rightarrow A[j] = A_0[j]\Big) \ \wedge\ previousValue = A_0[(i-1)\,\%|A|]$

   $\equiv \quad A[i] = A_0[(i-1)\,\%|A|] \ \wedge\ tmp = A_0[i] \ \wedge\ 0 \le i < |A| \ \wedge\ |A| > 0 \ \wedge\ |A| = |A_0| \ \wedge$
   $\Big((\forall j)\ 0 \le j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ \Big((\forall j)\ i < j < |A| \rightarrow A[j] = A_0[j]\Big) \ \wedge$
   $previousValue = A_0[(i-1)\,\%|A|]$

   $\equiv \quad tmp = A_0[i] \ \wedge\ 0 \le i < |A| \ \wedge\ |A| > 0 \ \wedge\ |A| = |A_0| \ \wedge$
   $\Big((\forall j)\ 0 \le j \le i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ \Big((\forall j)\ i < j < |A| \rightarrow A[j] = A_0[j]\Big) \ \wedge$
   $previousValue = A_0[(i-1)\,\%|A|]$

   $\equiv P_4$

   Calculemos ahora el estado que alcanzamos desde P_4 con la instrucción $previousValue \leftarrow tmp$:

   $SP(previousValue \leftarrow tmp, P_4)$
   $\equiv \quad (\exists y) \quad previousValue = tmp \ \wedge\ tmp = A_0[i] \ \wedge\ 0 \le i < |A| \ \wedge\ |A| > 0 \ \wedge\ |A| = |A_0| \wedge$
   $\Big((\forall j)\ 0 \le j \le i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ \Big((\forall j)\ i < j < |A| \rightarrow A[j] = A_0[j]\Big) \ \wedge$
   $y = A_0[(i-1)\,\%|A|]$

   $\Rightarrow \quad previousValue = A_0[i] \ \wedge\ 0 \le i < |A| \ \wedge\ |A| > 0 \ \wedge\ |A| = |A_0| \ \wedge$
   $\Big((\forall j)\ 0 \le j \le i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ \Big((\forall j)\ i < j < |A| \rightarrow A[j] = A_0[j]\Big)$

   $\equiv P_5$

   Calculemos ahora el estado que alcanzamos desde P_5 con la instrucción $i \leftarrow i + 1$:

   $SP(i \leftarrow i+1, P_5)$
   $\equiv \quad (\exists y) \quad i = y + 1 \ \wedge\ previousValue = A_0[y] \ \wedge\ 0 \le y < |A| \ \wedge\ |A| > 0 \ \wedge\ |A| = |A_0|$
   $\wedge\ \Big((\forall j)\ 0 \le j \le y \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge\ \Big((\forall j)\ y < j < |A| \rightarrow A[j] = A_0[j]\Big)$

$$\equiv \quad previousValue = A_0[i-1] \ \wedge \ 0 \leq i-1 < |A| \ \wedge \ |A| > 0 \ \wedge \ |A| = |A_0| \ \wedge$$
$$\Big((\forall j)\ 0 \leq j \leq i-1 \rightarrow A[j] = A_0[(j-1)\,\%|A|]\Big) \ \wedge \ \Big((\forall j)\ i-1 < j < |A| \rightarrow A[j] = A_0[j]\Big)$$

$$\equiv P_6$$

Veamos entonces por último que $P_6 \rightarrow I$:

- $0 \leq i-1 < |A| \quad \Rightarrow \quad 0 \leq i \leq |A| \quad \checkmark$
- $|A| = |A_0| \quad \Rightarrow \quad |A| = |A_0| \quad \checkmark$
- $|A| > 0 \quad \Rightarrow \quad |A| > 0 \quad \checkmark$
- $0 \leq i-1 < |A| \quad \Rightarrow \quad (i-1)\,\%|A| = i-1$
  $\therefore previousValue = A_0[i-1] \quad \Rightarrow \quad previousValue = A[(i-1)\,\%|A|] \quad \checkmark$
- $\big((\forall j)\ 0 \leq j \leq i-1 \rightarrow A[j] = A_0[(j-1)\,\%|A|]\big) \quad \Rightarrow$
  $\big((\forall j)\ 0 \leq j < i \rightarrow A[j] = A_0[(j-1)\,\%|A|]\big) \quad \checkmark$
- $\big((\forall j)\ i-1 < j < |A| \rightarrow A[j] = A_0[j]\big) \quad \Rightarrow$
  $\big((\forall j)\ i \leq j < |A| \rightarrow A[j] = A_0[j]\big) \quad \checkmark$

Por lo tanto, el algoritmo es correcto respecto a su especificación. □