Algoritmos y Estructuras de Datos

Guía Práctica 3
Verificación de programas (Parte 2)
Segundo Cuatrimestre 2024

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

3.2. Demostración de corrección de ciclos en SmallLang

3.2.1. Teorema del invariante: corrección de ciclos

Ejercicio 1. Consideremos el problema de sumar los elementos de un arreglo y la siguiente implementación en SmallLang, con el invariante del ciclo.

Especificación

proc sumar (in s: $array<\mathbb{Z}>$) : $\mathbb{Z}$ {
  requiere $\{True\}$
  asegura $\{res = \sum_{j=0}^{|s|-1} s[j]\}$
}

Implementación en SmallLang

```
res := 0;
i := 0;
while (i < s.size()) do
  res := res + s[i];
  i := i + 1
endwhile
```

Invariante de Ciclo

$$I \equiv 0 \leq i \leq |s| \wedge_L res = \sum_{j=0}^{i-1} s[j]$$

a) Escribir la precondición y la poscondición del ciclo.

b) ¿Qué punto falla en la demostración de corrección si el primer término del invariante se reemplaza por $0 \leq i < |s|$?

c) ¿Qué punto falla en la demostración de corrección si el límite superior de la sumatoria $(i-1)$ se reemplaza por i?

d) ¿Qué punto falla en la demostración de corrección si se invierte el orden de las dos instrucciones del cuerpo del ciclo?

e) Mostrar la corrección parcial del ciclo, usando los primeros puntos del teorema del invariante.

f) Proponer una función variante y mostrar la terminación del ciclo, utilizando la función variante.

Solución

a) $P_c \equiv \{res = 0 \wedge i = 0\}$ $Q_c \equiv \{i = |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

b) Falla la demostración de $I \wedge \neg B \rightarrow Q_c$ ya que no podemos restringir el valor de i para que ocurra $i = |s|$.

c) Falla la demostración de $P_c \rightarrow I$ porque antes de ingresar al ciclo se pide que en res esté almacenado el valor del primer elemento, pero no se cumple siempre porque al comenzar tenemos que $res = 0$. Además, en cada iteración se va a estar esperando que un elemento más en la suma resultante. Por otro lado, al ingresar a la última iteración además, se intentará acceder a una posición de la secuencia que no existe y se indefiniría la suma.

d) Falla la demostración de $\{I \wedge B\}$ **ciclo** $\{I\}$.

e)
1. $P_c \rightarrow I$. (Demostrar que vale I sabiendo que vale P_c).
   - $i = 0 \Rightarrow 0 \leq 0 \leq |s|$ ✓
   - $i = 0 \rightarrow res = \sum_{j=0}^{i-1} s[j] = \sum_{j=0}^{-1} s[j] = 0$ por tener rango vacío. ✓
2. $\{I \wedge B\}$ **ciclo** $\{I\}$. Para ver que la tripla de Hoare es válida, hay que probar $I \wedge B \rightarrow wp(ciclo, I)$

-
$$\begin{aligned}
wp(\textbf{ciclo}, I) &\equiv wp(\textbf{S1;S2}, I)\\
&\equiv wp(\textbf{S1}, wp(\textbf{S2}, I))\\
&\equiv wp(\textbf{S1}, (0 \leq i+1 \leq |s| \wedge_L res = \sum_{j=0}^{i} s[j]))\\
&\equiv 0 \leq i+1 \leq |s| \wedge_L res + s[i] = \sum_{j=0}^{i} s[j]\\
&\equiv \boxed{0 \leq i+1 \leq |s| \wedge_L res = \textstyle\sum_{j=0}^{i-1} s[j]}
\end{aligned}$$

- $I \wedge B \to wp(ciclo, I)$
  - $I \wedge B \Rightarrow 0 \leq i \leq |s| \wedge i < |s|$. Si $0 \leq i$ entonces $0 \leq i+1$. Por el otro lado, $i < |s|$ es lo mismo que decir $i+1 < |s|+1 \equiv i+1 \leq |s|$. Juntando todo llegamos a $0 \leq i+1 \leq |s|$ como dice la WP ✓
  - $res = \sum_{j=0}^{i-1} s[j]$ está idéntico en I y en la WP ✓

3. $(I \wedge \neg B) \to Q_c$
   - $I \wedge \neg B \to 0 \leq i \leq |s| \wedge i \geq |s|$. Entonces, sabemos que $i = |s|$ ✓
   - Usando eso en el otro término de I tenemos $res = \sum_{j=0}^{i-1} s[j] = \sum_{j=0}^{|s|-1} s[j]$ ✓

f) Función variante propuesta $f_v = |s| - i$

4. $\{I \wedge B \wedge v_0 = f_v\}$ **ciclo** $\{f_v < v_0\}$ válida si $(I \wedge B \wedge v_0 = f_v) \to wp(\textbf{ciclo}, f_v < v_0)$.
   -
$$\begin{aligned}
wp(\textbf{ciclo}, f_v < v_0) &\equiv wp(\textbf{S1;S2}, f_v < v_0)\\
&\equiv wp(\textbf{res := res + s[i]}, wp(\textbf{i := i + 1}, |s| - i < v_0))\\
&\equiv wp(\textbf{res := res + s[i]}, (|s| - i - 1 < v_0))\\
&\equiv \text{def}(s[i]) \wedge_L |s| - i - 1 < v_0\\
&\equiv 0 \leq i < |s| \wedge_L |s| - i - 1 < v_0\\
&\equiv \boxed{|s| - i - 1 < v_0}
\end{aligned}$$
   - $(I \wedge B \wedge v_0 = f_v) \to wp(\textbf{ciclo}, f_v < v_0)$: Por $v_0 = f_v$, $|s| - i - 1 < |s| - i \leftrightarrow -1 < 0$ ✓
5. $I \wedge f_v \leq 0 \to \neg B$: $I \wedge f_v \leq 0 \to 0 \leq i \leq |s| \wedge |s| - i \leq 0 \leftrightarrow$
   $0 \leq i \leq |s| \wedge |s| \leq i \leftrightarrow$
   $i = |s| \Rightarrow \neg(i < |s|) \equiv \neg B$ ✓

Entonces el ciclo termina y, por consiguiente, es correcto.

Ejercicio 2. Dadas la especificación y la implementación del problema `sumarParesHastaN`

Especificación

proc `sumarParesHastaN` (in n: $\mathbb{Z}$) : $\mathbb{Z}$ {
  `requiere` $\{n \geq 0\}$
  `asegura` $\{res = \sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j \ mod \ 2 = 0, j, 0))\}$
}

Implementación en SmallLang

```
res := 0;
i := 0;
while (i < n) do
  res := res + i;
  i := i + 2
endwhile
```

Invariante de ciclo

$$I \equiv 0 \leq i \leq n+1 \wedge i \ mod \ 2 \ = \ 0 \wedge res = \sum_{j=0}^{i-1}(\mathsf{IfThenElseFi}(j \ mod \ 2 = 0, j, 0))$$

a) Escribir la precondición y la poscondición del ciclo.

b) Mostrar la corrección parcial del ciclo, usando los primeros puntos del teorema del invariante.

c) Proponer una función variante y mostrar la terminación del ciclo, utilizando la función variante.

Solución

a) $P_c \equiv \{n \geq 0 \wedge res = 0 \wedge i = 0\}$
$Q_c \equiv \{i \geq n \wedge res = \sum_{j=0}^{i-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)\}$

b) 1. $P_c \rightarrow I$.
   - $i = 0 \Rightarrow 0 \leq 0 \leq n+1$ ✓
   - $i = 0 \Rightarrow 0 \bmod 2 = 0$ ✓
   - $res = 0 \wedge i = 0 \Rightarrow res = \sum_{j=0}^{0-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)) = 0$ porque el rango es vacío. ✓

   2. $\{I \wedge B\}$ **ciclo** $\{I\}$. $\leftrightarrow I \wedge B \rightarrow wp(ciclo, I)$
   -
$$\begin{aligned} wp(\textbf{ciclo}, I) &\equiv wp(\textbf{S1;S2}, I) \\ &\equiv wp\big(\textbf{res := res + i}, wp(\textbf{i := i + 2}, I)\big) \\ &\equiv wp\Big(\textbf{res := res + i}, 0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge \\ &\qquad\qquad \wedge\, res = \sum_{j=0}^{i+1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big)\Big) \\ &\equiv wp\Big(\textbf{res := res + i}, 0 \leq i+2 \leq n+1 \wedge i \bmod 2 = 0 \wedge \\ &\qquad\qquad \wedge\, res = \sum_{j=0}^{i+1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big)\Big) \\ &\equiv 0 \leq i+2 \leq n+1 \wedge i \bmod 2 = 0 \wedge res + i = \sum_{j=0}^{i+1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big) \\ &\equiv 0 \leq i+2 \leq n+1 \wedge i \bmod 2 = 0 \wedge res = \sum_{j=0}^{i+1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big) - i \\ &\equiv \boxed{0 \leq i+2 \leq n+1 \wedge i \bmod 2 = 0 \wedge res = \sum_{j=0}^{i-1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big)} \end{aligned}$$

     Para que $i+2$ sea par, tiene que ocurrir que i también lo sea. Como i es par, $i+1$ es impar por lo que se puede sacar de la sumatoria sin afectar el resultado.
   - $I \wedge B \rightarrow wp(\textbf{ciclo}, I)$.
     - $I \wedge B \Rightarrow 0 \leq i \leq n+1 \wedge i < n$. Si $0 \leq i$ entonces $0 \leq i+2$. Por el otro lado, $i < n$ que es lo mismo que decir $i+2 < n+2 \equiv i+2 \leq n+1$. Juntando todo llegamos a $0 \leq i+2 \leq n+1$ como dice la WP ✓
     - $i \bmod 2 = 0$. está idéntico en I y en la WP ✓
     - $res = \sum_{j=0}^{i-1}\big(\mathsf{IfThenElseFi}(j \bmod 2 = 0, j, 0)\big)$ está idéntico en I y en la WP ✓

   3. $(I \wedge \neg B) \rightarrow Q_c$. Hay que demostrar que vale Q_c sabiendo que vale $(I \wedge \neg B)$. Donde $\neg B \equiv i \geq n$.
   - $(n \bmod 2 = 0 \rightarrow i = n) \wedge (n \bmod 2 = 1 \rightarrow i = n+1)$. Analicemos los dos casos por separado.
     - Par. $n \bmod 2 = 0$. Quiero ver que $i = n$.
       - Sabemos por I que $i \leq n+1$ y que $i \bmod 2 = 0$

- ◦ Además, por $\neg B$, sabemos que $i \geq n$, por lo que $i = n \lor i = n+1$
  - ◦ Pero como i y n son pares, tenemos que $i = n$ ✓
- Impar. $n \bmod 2 = 1$. Quiero ver que $i = n+1$.
  - ◦ Sabemos por I que $i \leq n+1$ y que $i \bmod 2 = 0$
  - ◦ Además por $\neg B$ sabemos que $i \geq n$, por lo que $i = n \lor i = n+1$
  - ◦ Pero como i es par y n es impar, tenemos que $i = n+1$ ✓

- $i \bmod 2 = 0$. Trivialmente cierto por I ✓
- $res = \sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))$.
  - Por I sabemos que $res = \sum_{j=0}^{i-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))$
  - Además sabemos que $(n \bmod 2 = 0 \to i = n) \land (n \bmod 2 = 1 \to i = n+1)$
  - Veamos que ocurre en ambos casos donde quiero ver que $res = \sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))$
    - ◦ Par. $n \bmod 2 = 0$
      - ⋄ Sabemos que $i = n$
      - ⋄ Por I tenemos que $res = \sum_{j=0}^{i-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)) \Rightarrow res = \sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))$
    - ◦ Impar. $n \bmod 2 = 1$.
      - ⋄ Sabemos que $i = n+1$
      - ⋄ Por I tenemos que $res = \sum_{j=0}^{i-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))$
      - ⋄ Reemplazando nos queda que

$$\begin{aligned}
res = &\sum_{j=0}^{n}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)) = \\
= &\sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)) + (\mathsf{IfThenElseFi}(n\ mod\ 2 = 0, n, 0)) = \\
= &\sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0)) + 0 = \\
= &\boxed{\sum_{j=0}^{n-1}(\mathsf{IfThenElseFi}(j\ mod\ 2 = 0, j, 0))}
\end{aligned}$$

      - ⋄ Al extraer el último elemento de la sumatoria, como n es impar, el resultado del condicional es 0.

c) Función variante propuesta: $f_v = n + 2 - i$

4. $\{I \land B \land v_0 = f_v\}$ **ciclo** $\{f_v < v_0\}$. Queremos nuevamente ver que la tripla de Hoare planteada es válida. Es decir, que $(I \land B \land v_0 = f_v) \to wp(ciclo, f_v < v_0)$.

$$\begin{aligned}
wp(ciclo, f_v < v_0) &\equiv wp(S1; S2, f_v < v_0) \\
&\equiv wp(S1, wp(S2, n+1-i < v_0)) \\
&\equiv wp(S1, (n-i-1 < v_0)) \\
&\equiv n - i - 1 < v_0
\end{aligned}$$

- $(I \land B \land v_0 = f_v) \to wp(ciclo, f_v < v_0)$. Vemos que por $v_0 = f_v$ tenemos que $n - i - 1 < n - i \leftrightarrow -1 < 0$ que es trivialmente cierto.

5. $I \land f_v \leq 0 \to \neg B$. Donde $\neg B \equiv i \geq n$. De la función variante que $n + 1 - i \leq 0 \leftrightarrow n + 1 \leq i$. Si $n + 1 \leq i \leq n + 1$ entonces tenemos que $i = n + 1$. Se puede ver que $i = n + 1 \to i \geq n$.

Entonces el ciclo termina y, por consiguiente, es correcto.

Ejercicio 3. Considere el problema `sumaDivisores`, dado por la siguiente especificación:

proc `sumaDivisores` (in n: $\mathbb{Z}$) : $\mathbb{Z}$ {
  `requiere` $\{n \geq 1\}$
  `asegura` $\{res = \sum_{j=1}^{n}(\text{IfThenElseFi}(n\ mod\ j = 0, j, 0))\}$
}

a) Escribir un programa en SmallLang que satisfaga la especificación del problema y que contenga exactamente un ciclo.

b) Escribir la pre y post condición del ciclo y su invariante.

c) Considere el siguiente invariante para este problema

$$I \equiv 1 \leq i \leq n/2 \wedge res = \sum_{j=1}^{i}(\text{IfThenElseFi}(n\ mod\ j = 0, j, 0))$$

Si no coincide con el propuesto en el inciso anterior, ¿qué cambios se le deben hacer al programa para que lo represente este invariante? ¿Deben cambiar la pre y post condición?

Solución

a)
```
i := 1;
res := 0;
while (i < n + 1) do
    if (n mod i = 0) then
        res := res + i;
    else
        skip;
    endif;
    i := i + 1;
endwhile;
```

b) $P_c \equiv \{res = 0 \wedge i = 1\}$
$Q_c \equiv res = \sum_{j=1}^{n}(\text{IfThenElseFi}(n\ mod\ j = 0, j, 0))$
$I \equiv 1 \leq i \leq n+1 \wedge res = \sum_{j=1}^{i-1}(\text{IfThenElseFi}(n\ mod\ j = 0, j, 0)))$

c) Se puede cambiar la guarda del ciclo reemplazando `i < n + 1` por `i < n/2`. No hay que cambiar la pre y post condiciones dado que los divisores de un número entero están todos entre 1 y $n/2$

Ejercicio 4. Considere la siguiente especificación e implementación del problema `copiarSecuencia`, y la pre y post condiciones del ciclo.

Especificación

proc `copiarSecuencia` (in s: $array < \mathbb{Z} >$, inout r: $array < \mathbb{Z} >$) {
  `requiere` $\{|s| = |r| \wedge r = R_0\}$
  `asegura` $\{|s| = |r| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])\}$
}

Implementación en SmallLang

```
i := 0;
while (i < s.size()) do
  r[i]:=s[i];
  i:=i+1
endwhile
```

$$P_c \equiv |s| = |r| \wedge i = 0$$

$$Q_c \equiv (\forall j : \mathbb{Z})(0 \leq j < |r| \implies {}_L s[j] = r[j])$$

a) ¿Qué variables del programa deben aparecer en el invariante?

b) Proponer un invariante e indicar qué cláusula del mismo es necesario para cada paso de la demostración.

c) Proponer una función variante y demostrar que el ciclo termina.

Solución

a) Las variables que deben aparecer son i , r y s

b)
- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |r| \wedge (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = r[j])\}$
- Para poder demostrar $P_c \implies I$ solo necesito $0 \leq i \leq |s|$
- Para poder demostrar $\{I \wedge B\}S\{I\}$ solo necesito $0 \leq i \leq |s|$
- Ahora para demostrar $(I \wedge \neg B) \implies Q_c$ se necesita el invariante completo

c) Defino $f_v = |s| - i$ Para probar finalización del ciclo tengo que probar:

- $(I \wedge f_v \leq 0) \implies \neg B$
  $(I \wedge f_v \leq 0) \implies |s| - i \leq 0 \implies i \geq |s| \equiv \neg B$
- $\{I \wedge B \wedge f_v = v_0\}S\{f_v < v_0\}$
  $\{I \wedge B \wedge f_v = v_0\}S\{f_v < v_0\} \iff (I \wedge B \wedge f_v = v_0) \implies wp(S, f_v < v_0)$

$$\begin{aligned} wp(S, I) &\equiv wp(r[i] = s[i], wp(i = i + 1, |s| - i < v_0)) \\ &\equiv wp(r[i] = s[i], |s| - (i + 1) < v_0) \\ &\equiv |s| - (i + 1) < v_0 \end{aligned}$$

Pero $f_v = v_0 \implies v_0 - 1 < v_0 \iff -1 < 0$

Ejercicio 5. Sea el siguiente ciclo con su correspondiente precondición y postcondición:

```
while (i >= s.size() / 2) do
  suma := suma + s[s.size()-1-i];
  i := i - 1
endwhile
```

$$P_c : \{|s| \ mod \ 2 = 0 \wedge i = |s| - 1 \wedge suma = 0\}$$

$$Q_c : \{|s| \ mod \ 2 = 0 \wedge i = |s|/2 - 1 \ \wedge_L \ suma = \sum_{j=0}^{|s|/2-1} s[j]\}$$

a) Proponer un invariante e indicar qué clausula del mismo es necesaria para cada paso de la demostración.

b) Proponer una función variante que permita demostrar que el ciclo termina.

c) Demostrar la terminación del ciclo utilizando la función variante.

Solución

a) $I \equiv \{|s|/2 - 1 \leq i \leq |s| - 1 \wedge |s| \text{ mód } 2 = 0 \wedge suma = \sum_{j=0}^{|s|-2-i}\}$

- Para poder demostrar $P_c \implies I$ solo necesito $|s|/2 - 1 \leq i \leq |s| - 1$

- Para poder demostrar $\{I \wedge B\}S\{I\}$ solo necesito $|s|/2 - 1 \leq i \leq |s| - 1$
- Ahora para demostrar $(I \wedge \neg B) \implies Q_c$ se necesita el invariante completo

b) $f_v = i - |s|/2 - 1$ Para probar finalización del ciclo tengo que probar:

- $(I \wedge f_v \leq 0) \implies \neg B$
  $|s|/2 - 1 \leq i \leq |s| \wedge i - (|s|/2 - 1) \leq 0 \rightarrow i = |s|/2 - 1$
  $i = |s|/2 - 1 \rightarrow i < |s|/2 \equiv \neg B$
- $\{I \wedge B \wedge f_v = v_0\}S\{f_v < v_0\}$

Ejercicio 6. Dado el siguiente problema

`proc sumarElementos` (in s: $array < \mathbb{Z} >$) : $\mathbb{Z}$ {
  `requiere` $\{|s| \geq 1 \wedge |s| \textbf{ mod } 2 = 0\}$
  `asegura` $\{res = \sum_{j=0}^{|s|-1} s[j]\}$
}

Dar un invariante y función variante para cada una de estas posibles implementaciones

a)
```
res := 0
i := 0
while (i < s.size()) do
    res := res + s[i];
    i := i + 1
endwhile
```

b)
```
res := 0
i := 0
while (i < s.size()) do
    res := res + s[s.size() - 1 - i];
    i := i + 1
endwhile
```

c)
```
res := 0
i := s.size() - 1
while (i >= 0) do
    res := res + s[i];
    i := i - 1
endwhile
```

d)
```
res := 0
i := 0
while (i < s.size() / 2) do
    res := res + s[i] + s[s.size() - 1 - i];
    i := i + 1
endwhile
```

Solución

a)
- $I \equiv \{0 \leq i \leq |s| \wedge_L res = \sum_{j=0}^{i-1} s[i]\}$
- $f_v = |s| - i$

b)
- $I \equiv \{0 \leq i \leq |s| \wedge_L res = \sum_{j=0}^{i-1} s[|s| - 1 - i]\}$
- $f_v = |s| - i$

c)
- $I \equiv \{-1 \leq i \leq |s| - 1 \wedge_L res = \sum_{j=i+1}^{|s|-1} s[i]\}$
- $f_v = i + 1$

d)
- $I \equiv \{0 \leq i \leq |s|/2 \wedge_L res = \sum_{j=0}^{i-1} (s[i] + s[|s| - 1 - i])\}$
- $f_v = |s|/2 - i$

Ejercicio 7. Considerando el siguiente Invariante:

$$I \equiv \{0 \leq i \leq |s| \land (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L ((j \ mod \ 2 = 0 \land s[j] = 2 \times j) \lor (j \ mod \ 2 \neq 0 \land s[j] = 2 \times j + 1)))\}$$

- Escribir un programa en SmallLang que se corresponda al invariante dado.
- Defina las P_c, B y Q_c que correspondan a su programa.
- Dar una función variante para que se pueda completar la demostración.

Solución

a)
```
i := 0;
while (i < s.size()) do
    if (i mod 2 = 0) then
        s[i] := 2*i;
    else
        s[i] := 2*i +1;
    endif;
    i := i + 1;
endwhile;
```

b) $P_c \equiv i = 0$
$B \equiv i < |s|$
$Q_c \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L ((j \ mod \ 2 = 0 \land s[j] = 2 \times j) \lor (j \ mod \ 2 \neq 0 \land s[j] = 2 \times j + 1)))$

c) $f_v = |s| - i$

Ejercicio 8. Considerando el siguiente Invariante:

$$I \equiv \{0 \leq i \leq |s|/2 \land (\forall j : \mathbb{Z})(0 \leq j < i) \rightarrow_L (s[j] = 0 \land s[|s| - j - 1] = 0)\}$$

- Escribir un programa en SmallLang que se corresponda al invariante dado.
- Defina las P_c, B y Q_c que correspondan a su programa.
- Dar una función variante para que se pueda completar la demostración.

Solución

a)
```
i := 0;
while (i < s.size()/2) do
        s[i] := 0;
        s[s.size()-1-j] := 0;
    i := i + 1;
endwhile;
```

b)
- $P_c \equiv \{i = 0 \| |s| \text{mod} 2 = 0\}$
- $B \equiv \{i < |s|/2\}$
- $Q_c \equiv \{(\forall j : \mathbb{Z})(0 \leq j < |s|) \rightarrow_L (s[j] = 0\}$

c) $f_v = |s|/2 - i$

Ejercicio 9. Indique si el siguiente enunciado es verdadero o falso; fundamente:

Si dados B y I para un ciclo S existe una función f_v que cumple lo siguiente:

- $\{I \wedge B \wedge f_v = V_0\}S\{f_v > V_0\}$
- $\exists(k : \mathbb{Z})(I \wedge f_v \geq k \rightarrow \neg B)$

entonces el ciclo siempre termina.

Solución

Verdadero, si tomamos a $f_v = k - f_v$ podemos ver que se sigue cumpliendo el teorema de Terminacion

Ejercicio 10. Considere la especificación de la función `existeElemento` y su implementación

Especificación

`proc existeElemento` (in s: $array < \mathbb{Z} >$, in e: $\mathbb{Z}$) : Bool {
  `requiere` $\{True\}$
  `asegura` $\{res = True \leftrightarrow$
  $((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k] = e)\}$
}

Implementación en SmallLang

```
i := 0;
j := −1;
while (i < s.size()) do
   if (s[i] = e) then
     j := i
  else
     skip
  endif;
  i := i + 1
endwhile;
if (j != −1)
   res := true
else
   res := false
endif
```

Escribir los pasos necesarios para demostrar la correctitud de la implementación respecto a la especificación usando WP y el teorema del invariante

Solución

a) $P \implies wp(S, Q)$

b) Teorema del Invariante
$P_c \equiv i = 0 \wedge j = -1$
$Q_c \equiv res = True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k] = e)$
$I \equiv 0 \leq i \leq |s| \wedge (j \neq -1 \implies (\exists k : \mathbb{Z})(0 \leq k < i \wedge_L s[k] = e)) \wedge (j = -1 \implies (\exists k : \mathbb{Z})(0 \leq k < i \wedge_L s[k] \neq e))$
$B \equiv i \geq |s|$

- $P_c \implies I$
  Se que $i = 0 \wedge j = -1$ luego $0 \leq 0 \leq |s|$ es verdadero, la implicación de $j \neq -1$ es verdadera, la implicación de $j = -1$ es verdadera pues el antecedente del $\exists$ es falso.
- $(I \wedge \neg B) \implies Q_c$
  $(I \wedge \neg B) \implies i = |s|$ y por lo tanto para este valor de i, las dos ramas del invariante son iguales a las de Q_c
- $\{I \wedge B\}S\{I\}$
  $\{I \wedge B\}S\{I\} \iff (I \wedge B) \implies wp(S, I)$

c) $Q_c \implies wp(C, Post)$

$$\begin{aligned} wp(if(...), Post) &\equiv def(j \neq -1) \wedge_L (j \neq -1 \wedge wp(r = True, Post)) \vee \\ &\qquad (j = -1 \wedge wp(r = false, Post)) \\ &\equiv (j \neq -1 \wedge wp(r = True, Post)) \vee \\ &\quad (j = -1 \wedge wp(r = false, Post)) \\ &\equiv (j \neq -1 \wedge true = true \iff ((\exists k : \mathbb{Z})(0 \leq k < |s| \wedge_L s[k] = e))) \vee \\ &\quad (j = -1 \wedge false = true \iff ((\exists k : \mathbb{Z})(0 \leq k < |s| \wedge_L s[k] \neq e))) \end{aligned}$$

3.2.2. Ejercicios de parcial

Ejercicio 11. Dados los siguientes ciclos y sus respectivas precondición (P_c) y poscondición (Q_c).

1. Proponer un invariante (I) y una función variante (f_v) para el ciclo
2. Demostrar los siguientes pasos de la demostración de correctitud del ciclo
   I) $P_c \rightarrow I$
   II) $(I \wedge \neg B) \rightarrow Q_c$
   III) $(I \wedge f_v \leq 0) \rightarrow \neg B$

a) $P_c \equiv \{s = S_0 \wedge i = 0 \wedge 0 \leq d < |s|\}$

```
while i < d do
    s[i] := e;
    i := i + 1;
end
```

$Q_c \equiv \{(\forall j : \mathbb{Z})(0 \leq j < d \rightarrow_L s[j] = e) \wedge (\forall j : \mathbb{Z})(d \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$

b) $P_c \equiv \{s = S_0 \wedge 0 \leq d < |s| \wedge i = d\}$

```
while i < |s| do
    s[i] := e
    i := i + 1
end
```

$Q_c \equiv \{(\forall (j : \mathbb{Z})(0 \leq j < d \rightarrow_L s[i] = S_0[i])) \wedge (\forall (j : \mathbb{Z})(d \leq j < |s| \rightarrow_L s[i] = e))\}$

c) $P_c \equiv \{i = |s| - 1 \wedge res = 0\}$

```
while i ≥ 0 do
    res := res + s[i] + 1;
    i := i − 1;
end
```

$Q_c \equiv \{res = |s| + \sum_{j=0}^{|s|-1} s[j]\}$

Solución

a) 1.
- $I \equiv \{0 \leq i \leq d \wedge_L \big((\forall j : \mathbb{Z})(0 \leq j < i) \rightarrow_L (s[j] = e) \wedge (\forall j : \mathbb{Z})(i \leq j < |s|) \rightarrow_L (s[j] = S_0[j])\big)\}$
- $f_v = d - i$

2. I) $P_c \rightarrow I$:

- $i = 0 \rightarrow 0 \leq 0 \leq d$ ✓
- $i = 0 \rightarrow 0 \leq j < 0 \equiv False \Rightarrow (\forall j : \mathbb{Z})(\overbrace{0 \leq j < 0}^{False}) \rightarrow_L (s[j] = e) \equiv True$ ✓
- $i = 0 \land s = S_0 \rightarrow (\forall j : \mathbb{Z})(0 \leq j < |s|) \rightarrow_L (s[j] = S_0[j]) \equiv True$ ✓

II) $(I \land \neg B) \Rightarrow Q_c$:

- $(I \land \neg B) \Rightarrow (0 \leq i \leq d \land i \geq d) \equiv i = d$
- $i = d \land (\forall j : \mathbb{Z})(0 \leq j < i) \rightarrow_L (s[j] = e) \Rightarrow (\forall j : \mathbb{Z})(0 \leq j < d) \rightarrow_L (s[j] = e)$ ✓
- $i = d \land (\forall j : \mathbb{Z})(i \leq j < |s|) \rightarrow_L (s[j] = S_0[j]) \Rightarrow (\forall j : \mathbb{Z})(d \leq j < |s|) \rightarrow_L (s[j] = S_0[j])$ ✓

III) $(I \land f_v \leq 0) \rightarrow \neg B$

- $(I \land f_v \leq 0) \Rightarrow (0 \leq i \leq d \land d - i \leq 0) \equiv (0 \leq i \leq d \land d \leq i) \Rightarrow i = d \Rightarrow \neg(i < d) \equiv \neg B$

b) 1.
- $I \equiv \{d \leq i \leq |s| \land_L \big((\forall j : \mathbb{Z})(0 \leq j < d) \rightarrow_L (s[j] = S_0[j]) \land (\forall j : \mathbb{Z})(d \leq j < i) \rightarrow_L (s[j] = e)\big)\}$
- $f_v = |s| - i$

2. I) $P_c \rightarrow I$:

- $i = d \rightarrow d \leq d \leq |s|$ ✓
- $i = d \land s = S_0 \Rightarrow (\forall j : \mathbb{Z})(0 \leq j < d) \rightarrow_L (s[j] = S_0[j]) \equiv True$ ✓
- $i = d \rightarrow d \leq j < d \equiv False \rightarrow (\forall j : \mathbb{Z})(\overbrace{d \leq j < d}^{False}) \rightarrow_L (s[j] = S_0[j]) \equiv True$ ✓

II) $(I \land \neg B) \Rightarrow Q_c$:

- $(I \land \neg B) \Rightarrow (d \leq i \leq |s| \land i \geq |s|) \equiv i = |s|$
- $i = |s| \land (\forall j : \mathbb{Z})(d \leq j < i) \rightarrow_L (s[j] = e) \Rightarrow (\forall j : \mathbb{Z})(d \leq j < |s|) \rightarrow_L (s[j] = e)$ ✓
- $(\forall j : \mathbb{Z})(0 \leq j < d) \rightarrow_L (s[j] = S_0[j])$ es una cláusula del Invariante ✓

III) $(I \land f_v \leq 0) \rightarrow \neg B$

- $(I \land f_v \leq 0) \Rightarrow (d \leq i \leq |s| \land |s| - i \leq 0) \equiv (d \leq i \leq |s| \land |s| \leq i) \Rightarrow i = |s| \Rightarrow \neg(i < |s|) \equiv \neg B$

c) 1.
- $I \equiv \{-1 \leq i \leq |s| - 1 \land_L res = \sum_{j=i+1}^{|s|-1} (s[i] + 1)\}$
- $f_v = i + 1$

2. I) $P_c \rightarrow I$:

- $i = |s| - 1 \rightarrow res = \sum_{j=|s|}^{|s|-1} (s[i] + 1) = 0$ ✓

II) $(I \land \neg B) \Rightarrow Q_c$:

- $(I \land \neg B) \Rightarrow (-1 \leq i \leq |s| - 1 \land i < 0) \equiv i = -1 \Rightarrow$
  $res = \sum_{j=0}^{|s|-1} (s[i] + 1) = \sum_{j=0}^{|s|-1} s[i] + \sum_{j=0}^{|s|-1} 1 = \sum_{j=0}^{|s|-1} s[i] + |s|$ ✓

III) $(I \land f_v \leq 0) \rightarrow \neg B$

- $(I \land f_v \leq 0) \Rightarrow (-1 \leq i \leq |s| - 1 \land i + 1 \leq 0) \equiv (-1 \leq i \leq |s| - 1 \land i \leq -1) \Rightarrow i = -1 \Rightarrow \neg(i \geq 0) \equiv \neg B$