Algoritmos y Estructuras de Datos I

Primer Cuatrimestre 2020

Guía Práctica 5

Ejercicios entregables

Integrantes:

Risaro Daniela Belén LU: 666/09

Sturmer Eva Sylvia Juliet LU: 606/19

Ejercicio 12 Demostrar que el siguiente programa es correcto respecto a la especificación dada.

Especificación

proc existeElemento (in s: seq$\langle\mathbb{Z}\rangle$, in e: $\mathbb{Z}$, out r: Bool) {

Pre $\{\mathit{True}\}$

Post $\{r = \text{True} \leftrightarrow$

$((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k] = e)\}$

}

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)
    r := true
else
    r := false
endif
```

Respuesta:

Para probar que el programa es correcto respecto a su especificacion vamos a probar que:

- Pre $\rightarrow$ wp(codigo previo al ciclo, Pc)
- Pc $\rightarrow$ wp(ciclo, Qc)
- Qc $\rightarrow$ wp(codigo posterior al ciclo, Post)

Si probamos estas tres cosas, por monotonía sabemos que Pre $\rightarrow$ wp(programa completo, Post) y por lo tanto el programa es correcto con respecto a la especificación.

Además para probar Pc $\rightarrow$ wp(ciclo, Qc) utilizaremos el Teorema del invariante:

- Pc $\rightarrow$ I
- $(I \wedge \neg B) \rightarrow Qc$
- $\{I \wedge B\}$ ciclo $\{I\}$
- $\{(I \wedge B \wedge v0 = fv)\}$ ciclo $\{fv < v0\}$
- $(I \wedge fv \leq 0) \rightarrow \neg B$

Eleccion de Pc, Qc, B, I, fv:

- $\text{Pc} \equiv \text{i} = 0 \land \text{j} = -1$
- $\text{Qc} \equiv \text{i} = |\text{s}| \land (\text{j} \mathrel{!}= -1 \leftrightarrow ((\exists \text{k} :\mathbb{Z})(0 \leq \text{k} < |\text{s}|) \land_L \text{s[k]} = \text{e}))$
- $\text{B} \equiv \text{i} < |\text{s}|$
- $\text{I} \equiv 0 \leq \text{i} \leq |\text{s}| \land_L (\text{ j} \mathrel{!}= -1 \leftrightarrow ((\exists \text{k} :\mathbb{Z})(0 \leq \text{k} < \text{i}) \land_L \text{s[k]} = \text{e}))$
- $\text{fv} = |\text{s}| - \text{i}$

COMENZAMOS PROBANDO Pre → wp(codigo previo al ciclo, Pc):

Queremos probar **Pre →L wp(i := 0; j := −1, Pc)**:

1. Para esto calculamos esta wp:

   $\text{wp(i := 0; j := −1, Pc)} \equiv \text{wp(i := 0, wp(j := −1, Pc)}$

2. Calculamos **wp(j := −1, Pc)**:

$$\begin{aligned} \text{wp}(\text{j} := -1, \{\text{i} = 0 \land \text{j} = -1\}) &\equiv \text{def}(-1) \land \text{i} = 0 \land -1 = -1 \\ &\equiv \text{True} \land \text{i} = 0 \land \text{True} \\ &\equiv \mathbf{i = 0 \equiv E1} \end{aligned}$$

3. Calculamos **wp(i := 0, E1)**:

$$\begin{aligned} \text{wp}(\text{i} := 0, \text{E1}) &\equiv \text{wp}(\text{i} := 0, \{\text{i} = 0\}) \\ &\equiv \text{def}(0) \land 0 = 0 \\ &\equiv \text{True} \land \text{True} \\ &\equiv \mathbf{True \equiv E2} \end{aligned}$$

Y como Pre ≡ E2, Pre → E2

AHORA QUEREMOS PROBAR Pc → wp(ciclo, Qc):

Como mencionamos anteriormente para ello utilizatemos el teorema del invariante:

1. $\text{Pc} \rightarrow \text{I}$
2. $(\text{I} \land \neg \text{B}) \rightarrow \text{Qc}$
3. $\{\text{I} \land \text{B}\}$ ciclo $\{\text{I}\}$
4. $\{(\text{I} \land \text{B} \land \text{v0} = \text{fv})\}$ ciclo $\{\text{fv} < \text{v0}\}$
5. $(\text{I} \land \text{fv} \leq 0) \rightarrow \neg \text{B}$

Empezamos por ver 1. Pc → I:

Si el antecedente es Falso la implicación es True. Ahora si el antecedente es verdadero hay que probar que la implicación vale.

$\mathbf{Pc \rightarrow I \equiv i = 0 \land j = -1 \rightarrow 0 \leq i \leq |s| \land_L (j \mathrel{!}= -1 \leftrightarrow ((\exists k :\mathbb{Z})(0 \leq k < i) \land_L s[k] = e))}$

$0 \leq \text{i} \leq |\text{s}|$ vale, ya que i=0 → $0 \leq \text{i}$.

Por otro lado, la listas no tienen número negativo de elementos, entonces $|\text{s}| \geq 0$.
Y como $\text{i} = 0$, entonces $|\text{s}| \geq \text{i}$.

Dado que $\text{i} = 0 \nexists$ un k que cumpla $0 \leq \text{k} < \text{i}$

Entonces, $(\exists \text{k} :\mathbb{Z})(0 \leq \text{k} < \text{i}) \land_L \text{s[k]} = \text{e})$ es falso.

Y para que se cumpla: $j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i) \wedge_L s[k] = e)$ el antecedente debe también ser falso. Por lo tanto: $j = -1$

Ahora probaremos 2. (I ∧ ¬ B) → Qc:

(I ∧ ¬ B):
- $0 \leq i \leq |s|$
- $j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i) \wedge_L s[k] = e)$
- $i \geq |s|$

Qc:
- $i = |s|$
- $j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k] = e)$

Dado que $0 \leq i \leq |s|$ sabemos que $i \leq |s|$, y por otro lado $\neg B$ nos dice que $i \geq |s|$, entonces $i = |s|$

Al reemplazar $i = |s|$ en el invariante pruebo el resto de Qc.

Ahora tenemos que probar que la tripla de Hoare: 3. {I ∧ B} ciclo {I} es valida y para eso tenemos que demostrar $(I \wedge B) \rightarrow wp(ciclo, I)$ donde el ciclo contiene un If y una instrucción i:= i + 1.

Entonces vamos a calcular: wp(if...then...else...fi; i:= i + 1, I)

```
while (i < s.size()) do
S1:  if (s[ i ] = e) then
          j := i
     else
          skip
     endif;
S2:  i := i + 1
endwhile;
```

Entonces wp(if...then...else...fi; i:= i + 1, I) ≡ wp(S1; S2, I)

Aplicanco el axioma 3 queda: wp(S1, wp(S2,I))

Vamos a analizar primero wp(S2,I). Por el Axioma 1:

$$
\begin{aligned}
\mathbf{E1} &\equiv wp(i := i + 1, I) \equiv def(i + 1) \wedge_L 0 \leq i+1 < |s| \wedge_L (j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i+1) \wedge_L s[k] = e)) \\
&\equiv true \wedge_L 0 \leq i+1 < |s| \wedge_L (j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i+1) \wedge_L s[k] = e)) \\
&\equiv 0 \leq i+1 < |s| \wedge_L (j \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i+1) \wedge_L s[k] = e))
\end{aligned}
$$

Ahora podemos calcular wp(S1, E1):

Por medio del axioma 4 calculamos:

$$
\begin{aligned}
wp(S1, E1) &\equiv def(s[\,i\,] = e) \wedge_L ((s[\,i\,] = e \wedge wp(j := i, E1)) \vee (s[\,i\,] \mathrel{!}= e \wedge wp(skip, E1))) \\
&\equiv 0 \leq i < |s| \wedge_L ((s[\,i\,] = e \wedge wp(j := i, E1)) \vee (s[\,i\,] \mathrel{!}= e \wedge E1))
\end{aligned}
$$

Desglosamos los terminos para que quede mas claro:

$0 \leq i < |s| \wedge_L (\text{Parte-A} \vee \text{Parte-B})$

Parte-A:

$s[\,i\,] = e \wedge wp(j{:=}\ i, E1) \equiv s[\,i\,] = e \wedge 0 \leq i+1 < |s| \wedge_L (i \mathrel{!}= -1 \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i+1) \wedge_L s[k] = e))$

Parte-B:

s[i] != e $\wedge$ E1) $\equiv$ s[i] != e $\wedge$ $0 \leq$ i+1 $<$ |s| $\wedge$L (j != $-1 \leftrightarrow$ (($\exists$k :$\mathbb{Z}$)($0 \leq$ k $<$ i+1) $\wedge$L s[k] = e))

Falta una DEMOSTRACION ACA!!!!

Ahora vamos a probar 4. {(I $\wedge$ B $\wedge$ v0 = fv)} ciclo {fv $<$ v0}

Calculamos wp(S1; S2, fv $<$ v0)

E0 $\equiv$ wp(S1, wp(S2, |s| - i $<$ v0)) $\equiv$ wp(S1, (i := i+1, |s| - i $<$ v0))

$\equiv$ wp(S1, (true $\wedge$ |s| - (i+1) $<$ v0))

Como S1 no involucra a i, entonces: E0 $\equiv$ |s| - i - 1 $<$ v0

Queremos ver que usando {(I $\wedge$ B $\wedge$ v0 = fv)} llegamos a E0.

Como v0 = fv entonces v0 = |s| - i, luego v0 -1 = |s| - i - 1 $<$ v0

Finalmente, 5. (I $\wedge$ fv $\leq$ 0) $\rightarrow \neg$ B

Dado que fv $\leq$ 0 $\equiv$ |s| - i $\leq$ 0

$\equiv$ |s| $\leq$ i $\equiv \neg$ B

Por lo tanto como cumple:

1. Pc $\rightarrow$ I ✓
2. (I $\wedge \neg$ B) $\rightarrow$ Qc ✓
3. {I $\wedge$ B} ciclo {I} ✓
4. {(I $\wedge$ B $\wedge$ v0 = fv)} ciclo {fv $<$ v0} ✓
5. (I $\wedge$ fv $\leq$ 0) $\rightarrow \neg$ B ✓

Entonces Pc $\rightarrow$ wp(ciclo, Qc)

FINALMENTE PROBAMOS Qc $\rightarrow$ wp(codigo posterior al ciclo, Post):

Queremos probar que **Qc $\rightarrow$ wp(if... then... else... fi, Post)**

Para ello usamos el Axioma 4, que nos dice que si S = if B then S1 else S2 endif, entonces wp(S, Post) $\equiv$ def(B) $\wedge$L ((B $\wedge$ wp(S, Post)) $\vee$ ($\neg$B $\wedge$ wp(S2, Post)))

1. Calculamos la wp:

wp(if (j != -1) then r:= true else r:=false fi, Post)

$\equiv$ def(j != −1) $\wedge$L (((j != −1) $\wedge$ wp(r:=true, Post)) $\vee$ ($\neg$(j != −1) $\wedge$ wp(r:=false, Post)))

$\equiv$ True $\wedge$L (((j != −1) $\wedge$ wp(r := true, Post)) $\vee$ ($\neg$(j != −1) $\wedge$ wp(r := false, Post)))

Lo dividimos en 2:

- (j != −1) $\wedge$ wp(r := true, Post)
- $\neg$(j != −1) $\wedge$ wp(r := false, Post)

(j != −1) $\wedge$ wp(r := true, Post)

$\equiv$ (j != −1) $\wedge$ wp(r := true, r = True $\leftrightarrow$ (($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e))

$\equiv$ (j != −1) $\wedge$ True $\wedge$L true = true $\leftrightarrow$ (($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e)

$\equiv$ (j != −1) $\wedge$ ($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e

$\neg$(j != −1) $\wedge$ wp(r := false, Post)

$\equiv$ $\neg$(j != −1) $\wedge$ wp(r := false, r = True $\leftrightarrow$ (($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e))

$\equiv$ (j = −1) $\wedge$L false = true $\leftrightarrow$ (($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e)

$\equiv$ (j = −1) $\wedge$ $\neg$($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e

Por lo que wp es:

E3 $\equiv$ ((j != −1) $\wedge$ ($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e) $\vee$ ((j = −1) $\wedge$ $\neg$($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e)

Aplico (p $\wedge$ q) $\vee$ ($\neg$p $\wedge$ $\neg$q) $\equiv$ p $\leftrightarrow$ q

E3 $\equiv$(j != −1) $\leftrightarrow$ ($\exists$k :$\mathbb{Z}$)(0 $\leq$ k < |s|) $\wedge$L s[k] = e

Y como Qc $\equiv$ E3, Qc $\rightarrow$ E3