Algoritmos y Estructuras de Datos I
Primer Cuatrimestre 2020

Guía Práctica 4
Resolución de los Ejercicios Entregables

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

Integrantes: Andrés M. Hense, Victoria Espil

Ejercicio 12 Para probar que el programa es correcto respecto a la especificacón, vamos a probar estas implicaciones por separado, y por monotonia llegaremos a que el programa es correcto.

- $Pre \rightarrow wp(\textbf{codigo previo al ciclo}, P_c)$
- $P_c \rightarrow wp(\textbf{ciclo}, Q_c)$
- $Q_c \rightarrow wp(\textbf{codigo posterior al ciclo}, Post)$

Especificacion del ciclo:

- $Pre : True$
- $Post : r = True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])$
- $P_c : i = 0 \wedge j = -1$
- $Q_c : i = |s| \wedge (j! = -1) \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])$
- $B : i < |s|$
- $I : 0 \leq i < |s| \wedge (j! = -1) \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i) \wedge_L s[k = e])$
- $f_v : |s| - i$

Empecemos probando la primer implicación

$Pre \rightarrow wp(\textbf{codigo previo al ciclo}, P_c)$

$$
\begin{aligned}
wp(i := 0; j := -1, P_c) &\equiv wp(i := 0, wp(j := -1, P_c)) \\
&\equiv wp(i := 0, (P_c)^j_{-1}) \\
&\equiv (i = 0 \wedge -1 = -1)^i_0 \\
&\equiv 0 = 0 \wedge True \\
&\equiv True
\end{aligned}
$$

Luego $True \rightarrow True$, es tautologia.

$P_c \rightarrow wp(\textbf{ciclo}, Q_c)$

Demostraremos que el ciclo es correcto respecto a su especificación y ademas termina.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $0 \leq i < |s|$
  Por P_c sabemos que $i = 0$, entonces $0 \leq i$ vale. Además, $|s| \geq 0$ (porque las listas no pueden tener una cantidad negativa de elementos), luegos $|s| \geq i$
- $(j! = -1) \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < i) \wedge_L s[k = e])$
  P_c indica que $i = 0$ y $j = -1$. Luego como no existe un k entre $0 \leq k < 0$, la doble implicacion no se cumple, por lo que tiene que pasar que $j = -1$.

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- Por I sabemos que $i < |s|$, y por $\neg B$ sabemos que $i \geq |s|$. Entonces i debe ser igual a $|s|$

- Es trivial ver que vale $(j! = -1) \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])$ ya que, al reemplazar $i = |s|$ en el invariante llego a eso.

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $r[i] := s[i]$;
S2: $i := i + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{1}$$

$\{(I \wedge B \wedge v_0 = f_v)\}$ **ciclo** $\{(f_v < v_0)\}$

Dado que queremos demostrar que vale una tripla de Hoare, comenzaremos calculando la precondición más débil $wp(ciclo, f_v < v_0)$.

$$\begin{aligned} wp(S1; S2, f_v < v_0) &\overset{Ax3}{\equiv} wp(S1, wp(S2, |s| - i < v_0)) \\ &\overset{Ax1}{\equiv} wp(S1, true \wedge_L |s| - (i+1) < v_0) \\ &\overset{Ax3}{\equiv} true \wedge_L (true \wedge_L |s| - (i+1) < v_0) \\ &\equiv |s| - i - 1 < v_0 \end{aligned}$$

Es decir, $wp(S1; S2, f_v < v_0) = |s| - i - 1 < v_0$. Ahora debemos ver que $(I \wedge B \wedge v_0 = f_v)$ implican dicha WP. Parte de la hipótesis es que $v_0 = f_v$, es decir $v_0 = |s| - i$. Restando 1 a ambos lados se prueba, $|s| - i - 1 = v_0 - 1 < v_0$.

$(I \wedge f_v \leq 0) \Rightarrow \neg B$

Debemos mostrar que vale $\neg B$, es decir $i \geq |s|$.
Sabemos que $f_v \leq 0$, es decir $|s| - i \leq 0$, luego $|s| \leq i$, como queriamos demostrar.

$Q_c \rightarrow wp($**codigo posterior al ciclo**$, Post)$

S: if $(j! = -1)$ **then** $r = True$ **else** $r = False$ **endif**

$$\begin{aligned} wp(\mathbf{S}, Post) &\equiv \text{def}(j! = -1) \wedge_L \Bigg(\Big((j! = -1) \wedge wp(r = True, Post)) \Big) \vee \Big(\neg(j! = -1) \wedge wp(j = False, Post) \Big) \Bigg) \\ &\equiv True \wedge \Bigg(\Big((j! = -1) \wedge Post^r_{True}) \Big) \vee \Big(j = -1 \wedge Post^r_{False}) \Big) \Bigg) \\ &\equiv \Big(j! = -1 \wedge True = True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e]) \Big) \vee \\ &\quad \Big(j = -1 \wedge False = True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])) \Big) \\ &\equiv \Big(j! = -1 \wedge True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e]) \Big) \vee \\ &\quad \Big(j = -1 \wedge False \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])) \Big) \\ &\equiv \Big(j! = -1 \wedge ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e]) \Big) \vee \\ &\quad \Big(j = -1 \wedge \neg((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e])) \Big) \\ &\equiv (j! = -1) \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k = e]) \end{aligned}$$

Como llegue a lo mismo que Q_c, entonces vale.