Prueba de Consistencia
de la Operación esVacía del TAD Cola
con implementación de cola circular en arreglos

Prof. Rosseline Rodríguez

Dada la siguiente especificación abstracta del TAD Cola:

Especificación A de TAD Cola(T)

Modelo de Representación

const MAX : int

var contenido : seq T

Invariante de Representación

$MAX > 0 \wedge \#\, contenido \leq MAX$

Operaciones

proc crear (**in** m : int; **out** c : Cola)

{ Pre: $m > 0$ **}**

{ Post: $c.MAX = m \wedge c.contenido = \langle\, \rangle$ **}**

proc encolar (**in-out** c : Cola; **in** e : T)

{ Pre: $\#\, c.contenido < c.MAX$ **}**

{ Post: $c.contenido = c_0.contenido ++ \langle\, e \,\rangle$ **}**

proc desencolar (**in-out** c : Cola)

{ Pre: $\#\, c.contenido > 0$ **}**

{ Post: $c.contenido = tl(c_0.contenido)$ **}**

proc primero (**in** c : Cola; **out** p : T)

{ Pre: $\#\, c.contenido > 0$ **}**

{ Post: $p = hd(c.contenido)$ **}**

proc esVacia (**in** c : Cola; **out** ev : boolean)

{ Pre: true }

{ Post: $ev \equiv \#\, c.contenido = 0$ **}**

Fin TAD

Se propone usar como modelo concreto de representación un arreglo de elementos de tipo T, que sea manejado de forma circular; para ello se incluye en el modelo de representación dos enteros *ini* y *fin*, que indican desde que posición hasta que posición (sin incluir ésta última) están almacenados los elementos de la cola; adicionalmente y para distinguir el caso cola vacía del caso cola llena se incluye un booleano llamado *vacia*. Veamos la re-especificación parcial del TAD:

Especificación B de TAD Cola(T), refinamiento de A

Modelo de Representación

const MAX : int

var ini, fin : int

var vacia : boolean

var elementos : array[0..MAX) of T

Invariante de Representación

$MAX > 0 \wedge 0 \leq ini, fin < MAX \wedge (vacia \Rightarrow ini = fin)$

Relación de Acoplamiento

$(ini < fin \vee vacia \Rightarrow contenido = \langle\, i : ini \rightarrow fin : elementos[i] \,\rangle) \wedge$

$(ini \geq fin \wedge \neg vacia \Rightarrow$

$contenido = \langle\, i : ini \rightarrow MAX : elementos[i] \,\rangle +\!\!+ \langle\, i : 0 \rightarrow fin : elementos[i] \,\rangle)$

Operaciones

proc crear (**in** m : int; **out** c : Cola)

{ Pre: $m > 0$ **}**

{ Post: $c.MAX = m \wedge \ldots$ **}**

proc encolar (**in-out** c : Cola; **in** e : T)

{ Pre: … **}**

{ Post: … **}**

proc desencolar (**in-out** c : Cola)

{ Pre: … **}**

{ Post: … **}**

proc primero (**in** c : Cola; **out** p : T)

{ Pre: … **}**

{ Post: … **}**

proc esVacia (**in** c : Cola; **out** ev : boolean)

{ Pre: true }

{ Post: … **}**

Fin TAD

Se quiere demostrar que las postcondiciones de la operación "esVacia" en las dos especificaciones corresponde adecuadamente, según la noción de refinamiento de datos.

Por regla de consistencia para operaciones debo probar

$[\ (\forall x$: x parámetro in(out o in-out) de tipo T: $x.Ac \wedge x.InvA \wedge x.InvC) \wedge$

$(\forall x$: x parámetro in-out de tipo T: $x_0.Ac \wedge x_0.InvA \wedge x_0.InvC) \Rightarrow (PostC \Rightarrow PostA)]$

Supongo como hipótesis (dado que sólo hay parámetros de entrada de tipo Cola):

H_0: $(c.ini < c.fin \vee c.vacia \Rightarrow c.contenido = \langle\, i : c.ini \rightarrow c.fin : c.elementos[i] \,\rangle) \wedge$

$(c.ini \geq c.fin \wedge \neg c.vacia \Rightarrow$

$c.contenido = \langle\, i: c.ini \rightarrow c.MAX : c.elementos[i] \,\rangle +\!\!+ \langle\, i: 0 \rightarrow c.fin : c.elementos[i] \,\rangle)$

H_1: $c.MAX > 0 \wedge \#\, c.contenido \leq c.MAX$

H_2: $c.MAX > 0 \wedge 0 \leq c.ini, c.fin < c.MAX \wedge (c.vacia \Rightarrow c.ini = c.fin)$

Pruebo: $[(ev \equiv c.vacia) \Rightarrow (ev \equiv (\#\, c.contenido = 0))]$

Por relación de acoplamiento (H_0) tengo dos casos:

Caso 1: c.ini < c.fin $\vee$ c.vacia

Caso 1.1: c.ini < c.fin

Partiendo del antecedente

$ev \equiv c.vacia$

$\equiv$ { H_0, caso 1, definición de c.contenido }

$(ev \equiv c.vacia) \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { definición de cardinalidad (#) en secuencias, con c.ini <c.fin }

$(ev \equiv c.vacia) \wedge \#c.contenido = (c.fin - c.ini)$

$\equiv$ { $c.fin - c.ini > 0$ }

$(ev \equiv c.vacia) \wedge \#c.contenido > 0$

$\Rightarrow$ { H_2, contrapositivo: $c.ini \neq c.fin \Rightarrow \neg c.vacia$, como c.ini<c.fin entonces c.vacia$\equiv$falso}

$(ev \equiv falso) \wedge \#c.contenido > 0$

$\equiv$ { $(\#c.contenido = 0) \equiv$falso pues $\#c.contenido > 0$ }

$(ev \equiv (\#c.contenido = 0)) \wedge \#c.contenido > 0$

$\Rightarrow$ { debilitamiento }

$ev \equiv (\#c.contenido = 0)$

Caso 1.2: c.vacia

Partiendo del antecedente

$ev \equiv c.vacia$

$\equiv$ { H_0, caso 1, definición de c.contenido }

$(ev \equiv c.vacia) \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { $c.vacia \equiv true$ }

$(ev \equiv true) \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { $(p \equiv true) \equiv p$}

$ev \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { $c.vacia \equiv true$ }

$ev \wedge c.vacia \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\Rightarrow$ { H_2: $c.vacia \Rightarrow (c.ini = c.fin)$ }

$ev \wedge (c.ini = c.fin) \wedge c.contenido = \langle\ i : c.ini \rightarrow c.fin : c.elementos[i]\ \rangle$

$\Rightarrow$ { sustitución, debilitamiento }

$ev \wedge c.contenido = \langle\ i : c.ini \rightarrow c.ini : c.elementos[i]\ \rangle$

$\Rightarrow$ { rango vacío en secuencias }

$ev \wedge c.contenido = \langle\ \rangle$

$\equiv$ { $(p \equiv q) \equiv p \wedge (q \equiv p)$ 3.50}

$ev \wedge (ev \equiv c.contenido = \langle\ \rangle)$

$\Rightarrow$ { debilitamiento }

$ev \equiv c.contenido = \langle\ \rangle$

$\equiv$ { definición de cardinalidad }

$ev \equiv (\#c.contenido = 0)$

Caso 2: $c.ini \geq c.fin \wedge \neg c.vacia$

Partiendo del antecedente

$ev \equiv c.vacia$

$\equiv$ { H0, caso 2, definición de c.contenido }

$(ev \equiv c.vacia)$

$\wedge\ c.contenido = \langle\ i: c.ini \rightarrow c.MAX : c.elementos[i]\ \rangle +\!\!+ \langle\ i: 0 \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { hipótesis : $\neg$c.vacia, definición de cardinalidad en concatenación }

$(ev \equiv falso)$

$\wedge\ \#c.contenido = \#\langle\ i: c.ini \rightarrow c.MAX : c.elementos[i]\ \rangle + \#\langle\ i: 0 \rightarrow c.fin : c.elementos[i]\ \rangle$

$\equiv$ { definición de cardinalidad en secuencias }

$(ev \equiv falso) \wedge \#c.contenido = (c.MAX - c.ini) + c.fin$

$\Rightarrow$ {H_2: $(c.ini < c.MAX) \equiv (c.MAX\text{-}c.ini > 0)$ y $0 \leq c.fin$ entonces $(c.MAX - c.ini) + c.fin > 0$ }

$(ev \equiv falso) \wedge \#c.contenido > 0$

$\equiv$ { $(\#c.contenido = 0) \equiv$falso pues $\#c.contenido > 0$ }

$ev \equiv (\#c.contenido = 0) \wedge \#c.contenido > 0$

$\equiv$ { debilitamiento }

$ev \equiv (\#c.contenido = 0)$

Por casos 1 y 2 concluimos $(ev \equiv c.vacia) \Rightarrow (ev \equiv (\#\ c.contenido = 0))$