Recursión

Reglas de Correctitud

Jesús Ravelo
Kelwin Fernández
Universidad Simón Bolívar
Dpto. de Computación y Tecnología de la Información

**** **Observación:** esta versión es un borrador y posiblemente puede contener errores. Debe ser utilizada prudentemente. ****

Las siguientes notas se basan en la idea de que el lector ya conoce el comportamiento de funciones y procedimientos recursivos, además de tener una base acerca de la correctitud de algoritmos mediante el manejo de especificaciones. El alcance de estas notas consiste únicamente en brindar una plataforma acerca de la aplicación de reglas de correctitud en funciones y procedimientos de naturaleza recursiva.

Definición de Números de Fibonacci

La función de Fibonacci es una de las más conocidas funciones recursivas y es de gran importancia en muchos problemas de computación. Su definición es la siguiente:

$$fib : \mathbb{N} \longrightarrow \mathbb{N}$$

$$fib(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ fib(n-1) + fib(n-2) & \text{si } n \geq 2 \end{cases}$$

Implementación Recursiva

proc fibonacciRecursivo(**in** n: *int*; **out** f: *int*)
$\{$ **Pre:** $n \geq 0$ $\}$
$\{$ **Post:** $f = fib(n)$ $\}$
$\{$ **Cota:** n $\}$
|[
if $n = 0 \longrightarrow f := 0$
[] $n = 1 \longrightarrow f := 1$
[] $n > 1 \longrightarrow$ **var** f_1, f_2: *int*
; $fibonacciRecursivo(n-1, f_1)$
; $fibonacciRecursivo(n-2, f_2)$
; $f := f_1 + f_2$
fi
]|

Notación

Para lo subsiguiente utilizaremos la notación presentada a continuación: Sea un procedimiento recursivo llamado P:

- Denotaremos por $PreDef$ a su Precondición y $PostDef$ a su postcondición, por motivos de que estas "definen" al procedimiento.
- Denotaremos como $FCota$ a su Función de Cota.
- Supongamos que se da una llamada cualquiera al procedimiento, denotaremos por $PreLlamada$ a la precondición justo antes de la llamada a P y $PostLlamada$ a la postcondición justo después.

Esquema para la Demostración de Correctitud del Cuerpo de $fibonacciRecursivo$

Para esto, veamos el procedimiento insertando a su comienzo y fin la pre y postcondición correspondiente:

$\{ \ n \geq 0 \ \wedge \ n = X \ \}$

$$
\begin{array}{ll}
\underline{\textbf{if}} \ n = 0 \longrightarrow & f := 0 \\
[] \ n = 1 \longrightarrow & f := 1 \\
[] \ n > 1 \longrightarrow & \textbf{var} \ \ f_1, \ f_2\text{: } int \\
 & ; fibonacciRecursivo(n-1, f_1) \\
 & ; fibonacciRecursivo(n-2, f_2) \\
 & ; f := f_1 + f_2 \\
\underline{\textbf{fi}} &
\end{array}
$$

$\{ \ f = fib(n) \ \}$

Nótese que hemos agregado a la precondición que $n = X$, más adelante se explicará el motivo de esto.

La demostración correspondiente para el condicional sigue siendo igual que en los casos en donde no se involucra una llamada recursiva. Por lo cual no centraremos nuestra atención en este punto.

Lo primero que haremos será colocar aserciones intermedias antes y después de cada bloque del condicional y entre las distintas llamadas recursivas para reducir nuestro problema a demostrar tripletas mucho más sencillas. Veamos como queda:

$\{\ n \geq 0 \ \wedge \ n = X \ \}$

$\underline{\mathbf{if}}\ n = 0 \longrightarrow \quad \{n = 0\}$
$\qquad f := 0$
$\qquad \{f = fib(n)\}$

$[]\ n = 1 \longrightarrow \quad \{n = 1\}$
$\qquad f := 1$
$\qquad \{f = fib(n)\}$

$[]\ n > 1 \longrightarrow \quad \mathbf{var}\ \ f_1,\ f_2$: *int*
$\qquad \{\underline{R_0} :\ n = X \ \wedge \ n > 1\}$
$\qquad ;\ fibonacciRecursivo(n-1, f_1)$
$\qquad \{\underline{R_1} : f_1 = fib(n-1) \ \wedge \ R_0\}$
$\qquad ;\ fibonacciRecursivo(n-2, f_2)$
$\qquad \{\underline{R_2} : f_2 = fib(n-2) \ \wedge \ R_1\}$
$\qquad ;\ f := f_1 + f_2$
$\qquad \{f = fib(n)\}$

$\underline{\mathbf{fi}}$

$\{\ f = fib(n) \ \}$

Hemos reducido el problema, a demostrar distintas cuestiones que para los momentos ya sabemos demostrar. El punto nuevo, y crucial, es la demostración de correctitud de las llamadas recursivas.

Al igual que para la iteración debemos demostrar terminación, mediante el decremento de la cota en cada vuelta, para la recursión debemos garantizar que esta debe concluir en algún momento, es decir, que la función de cota del procedimiento recursivo decrece en la llamada. Por lo demás, la correctitud de la llamada al procedimiento se realizará como de costumbre, esto es:

$$Prellamada \Rightarrow PreDef(\textit{...sustitución entradas...})$$

Se debe probar:

$$Prellamada \Rightarrow (PreDef \ \wedge \ FCota < X)(\textit{...sustitución entradas...})$$

En un contexto en el que el cuerpo del procedimiento se demuestra agregando $FCota = X$ a su precondición.

Esto último quiere decir que el cuerpo debe demostrarse, no con la tripleta de Hoare

$$\{PreDef\}\, Cuerpo\, \{PostDef\}$$

Sino con la tripleta:

$$\{PreDef\ \wedge\ FCota = X\}\, Cuerpo\, \{PostDef\}$$

Debido a esto fue que agregamos a la precondición del cuerpo anteriormente que $n = X$.

Adicionalmente debemos demostrar no sólo que la cota decrece, sino que tiene un límite inferior (regularmente considerado en cero), para esto demostraremos:

$$PreDef \Rightarrow FCota \geq 0$$

Demostremos finalmente el ejemplo de $fibonacciRecursivo$:

- Cota:
  $(PreDef \Rightarrow FCota \geq 0)$

$$\begin{aligned} & n \geq 0 \Rightarrow n \geq 0 \\ \equiv \quad & \quad < p \Rightarrow p \equiv true > \\ & true \end{aligned}$$

- Primera llamada

Primero probaremos que ambas precondiciones son consistentes y que la cota decrece con la llamada.

$$((Prellamada \Rightarrow (PreDef\ \wedge\ FCota < X)(...sustitución...))$$

Queremos probar $R_0 \Rightarrow (n \geq 0\ \wedge\ n < X)[n := n - 1]$, utilizando el método de suposición del antecedente:

Supongo $R_0 : (n = X \ \wedge \ n > 1) \equiv true$

$$
\begin{array}{ll}
 & (n \geq 0 \ \wedge \ n < X)[n := n-1] \\
\equiv & \langle \text{ Sustitución Textual } \rangle \\
 & n-1 \geq 0 \ \wedge \ n-1 < X \\
\equiv & \langle \text{ Aritmética } \rangle \\
 & n \geq 1 \ \wedge \ n < X+1 \\
\equiv & \langle \text{ Propiedades de } \leq \ \rangle \\
 & n \geq 1 \ \wedge \ n \leq X \\
\equiv & \langle \text{ Dado } R_0, \ n = X \ \rangle \\
 & true \ \wedge \ X \leq X \\
\equiv & \langle \text{ Reflexividad de } \leq \text{, Neutro de } \wedge \ \rangle \\
 & true
\end{array}
$$

$\therefore \ R_0 \Rightarrow (n \geq 0 \ \wedge \ n < X)$

Al igual que con las precondiciones, debemos demostrar que las postcondiciones de la llamada y la de definición del procedimiento son consistentes.

Queremos probar entonces que:

$$(PostDef(\ldots sustitución \ldots) \Rightarrow PostLlamada)$$

Consideraremos inicialmente que $PostLlamada$ es $f_1 = fib(n-1)$.

$$
\begin{array}{ll}
 & (f = fib(n))[n, f := n-1, f_1] \Rightarrow f_1 = fib(n-1) \\
\equiv & \quad < \text{Sustitución Textual} > \\
 & f_1 = fib(n-1) \Rightarrow f_1 = fib(n-1)
\end{array}
$$

Luego por regla de "fortalecimiento por no-afectación", [1] podemos agregar R_0 en $PostLlamada$.

- Segunda llamada

[1] La regla de "fortalecimiento por no-afectación" dice que si en la tripleta $\{P\}\, I\, \{Q\}$, I no afecta a las variables de un cierto predicado R, podemos fortalecer el predicado Q con R.

El procedimiento será análogo al de la primera llamada. Inicialmente probaremos que se cumplen las condiciones con las precondiciones y seguidamente con las postcondiciones.

Para las precondiciones tenemos:

$$((Prellamada \Rightarrow (PreDef \ \wedge \ FCota < X)(...sustitución...))$$

Por claridad, supondremos adicionalmente R_0 como una hipótesis adicional. Nótese que esto no era necesario dado que al suponer R_1 estamos suponiendo al mismo tiempo a R_0.

Supongo $R_0 : (n = X \ \wedge \ n > 1) \equiv true$
$R_1 : f_1 = fib(n-1) \ \wedge \ R_0 \equiv true$

$$
\begin{array}{ll}
 & (n \geq 0 \ \wedge \ n < X)[n := n-2] \\
\equiv & \langle \text{ Sustitución Textual } \rangle \\
 & n-2 \geq 0 \ \wedge \ n-2 < X \\
\equiv & \langle \text{ Aritmética } \rangle \\
 & n \geq 2 \ \wedge \ n-2 < X \\
\equiv & \langle \text{ Dado } R_0,\, n > 1 \ \rangle \\
 & true \ \wedge \ n-2 \leq X \\
\equiv & \langle \text{ Neutro de } \wedge \text{, dado } R_0,\, n = X \ \rangle \\
 & X-2 \leq X \\
\equiv & \langle \text{ Propiedades de } \leq \ \rangle \\
 & true
\end{array}
$$

$\therefore \ R_1 \Rightarrow (n \geq 0 \ \wedge \ n < X)$

Para las postcondiciones tenemos:

$$(PostDef(\ldots sustitución \ldots) \Rightarrow PostLlamada)$$

$$
\begin{array}{ll}
 & (f = fib(n))[n, f := n-2, f_2] \Rightarrow f_2 = fib(n-2) \\
\equiv & < \text{Sustitución Textual} > \\
 & f_2 = fib(n-2) \Rightarrow f_2 = fib(n-2)
\end{array}
$$

Igualmente que con el caso de la Postcondición para la primera llamada, podemos agregar R_1 a esta.

Recuérdese que hemos probado sólo la correctitud de las llamadas recursivas, sin embargo, para demostrar a cabalidad la correctitud del procedimiento, es necesario demostrar el condicional por completo. Se deja la conclusión de esta demostración como ejercicio para el lector.

Caso General

Luego, dado un procedimiento P con al menos una llamada recursiva, como el siguiente:

proc $P(\dots \mathit{ParámetrosFormales} \dots)$
$\{$ **Pre:** $\mathit{PredDef}$ $\}$
$\{$ **Post:** $\mathit{PostDef}$ $\}$
$\{$ **Cota:** FCota $\}$
$|[$
Cuerpo
$]|$

Se debe demostrar:

(i) La correctitud del cuerpo mediante la tripleta

$$\{\mathit{PredDef} \ \wedge \ \mathit{FCota} = X\} \ \mathit{Cuerpo} \ \{\mathit{PostDef}\}$$

(ii) Garantizar cota de terminación demostrando

$$\mathit{PreDef} \Rightarrow (\mathit{FCota} \geq 0)$$

(iii) Para cada llamada recursiva, adicionalmente de probar lo usual:

$$\mathit{Prellamada} \Rightarrow \mathit{PreDef}(\text{...}\mathit{sustitución\ entradas}\text{...})$$

$$\mathit{PostDef}(\dots \mathit{sustitución} \dots) \Rightarrow \mathit{PostLlamada}$$

Debemos probar que:

$$(\mathit{PreLlamada} \Rightarrow (\mathit{PredDef} \ \wedge \ \mathit{FCota} < X)(\dots \mathit{sustitución} \dots))$$

Ejercicios

Dadas cada una de las siguientes funciones, implemente procedimientos recursivos que computen los resultados de aplicar dichas funciones y demuestre en cada caso su correctitud:

(i) Factorial:

$$fact : \mathbb{N} \longrightarrow \mathbb{N} \qquad fact(n) = \begin{cases} 1 & \text{si } n = 0 \\ n * fact(n-1) & \text{si } n \geq 1 \end{cases}$$

(ii) Factorial Doble:

$$fact2 : \mathbb{N} \longrightarrow \mathbb{N} \qquad fact2(n) = \begin{cases} 1 & \text{si } 0 \leq n \leq 1 \\ n * fact2(n-2) & \text{si } n \geq 2 \end{cases}$$

(iii) Potencia, que dado un valor entero de n y un natural m, devuelve el valor de n^m:

$$pot : \mathbb{Z} \times \mathbb{N} \longrightarrow \mathbb{Z}$$
$$pot(n,m) = \begin{cases} 1 & \text{si } m = 0 \\ pot(n*m, m\ div\ 2) & \text{si } m\ mod\ 2 = 0 \\ n * pot(n, m-1) & \text{si } m\ mod\ 2 = 1 \end{cases}$$

(iv) Máximo Común Divisor

$$mcd : \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+ \qquad mcd(n,m) = \begin{cases} n & \text{si } n = m \\ mcd(n-m, m) & \text{si } n > m \\ mcd(n, m-n) & \text{si } n < m \end{cases}$$

(v) Combinatorio: Cuenta las formas de escoger los subconjuntos de tamaño k de un conjunto con n elementos.

$$comb : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$
$$comb(n,k) = \begin{cases} 1 & \text{si } k = 0 \\ 0 & \text{si } k > n \\ comb(n-1, k-1) + comb(n, k-1) & \text{si } k \leq n \end{cases}$$

(vi) Stirling de Segunda Especie: Cuenta el número de particiones de tamaño k de un conjunto de tamaño n.

$$S : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$

$$S(k,n) = \begin{cases} 1 & \text{si } k = 1 \vee k = n \\ 0 & \text{si } k > n \\ comb(n,2) & \text{si } k = n-1 \\ 2^{n-1} - 1 & \text{si } k = 2 \\ S(k-1,n-1) + k \cdot S(k,n-1) & \text{en otro caso} \end{cases}$$

(vii) Ackerman (proponer la función de cota para esta función está fuera del alcance de este curso, sin embargo, implemente un procedimiento que compute esta función):

$$A : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$$

$$A(m,n) = \begin{cases} n+1 & \text{si } m = 0 \\ A(m-1,1) & \text{si } n = 0 \\ A(m-1, A(m,n-1)) & \text{en otro caso} \end{cases}$$