

Transformación de subalgoritmos recursivos

10.-

entero función $\text{fun}(E \text{ vect: } x, E \text{ entero: } n, E \text{ entero: } i)$

$\{x = A[1..n] \wedge n \geq 0 \wedge 0 \leq i \leq n\}$

inicio

Si $i=0$ entonces

devolver 0

Si-no

devolver $x[i] + \text{fun}(x, n, i-1)$

fin-si

¿devuelve la suma de los i primeros elementos del vector?

fin-función

Se trata de una función recursiva no final

tupla de parámetros formales $\bar{x} \equiv x, n, i$ $\text{succ}(x, n, i) = x, n, i-1$

Caso-base? $(x, n, i) \equiv i=0$

$\text{comb}(x, n, i, \text{fun}) = x[i]$

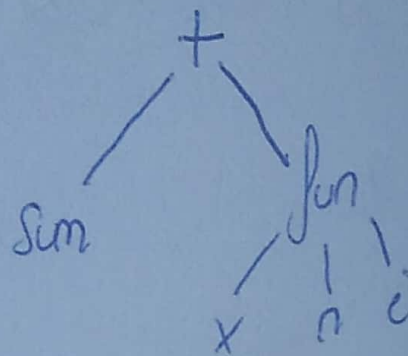
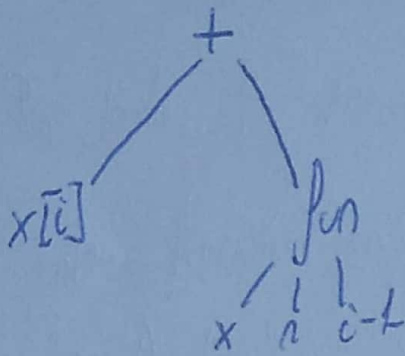
$\text{Sol}(x, n, i) = 0$

a) Generalización

Función sumergida: $f_{-rec}(x, n, i) = x[i] + f_{un}(x, n, i-1)$

Árbol sintáctica (función sumergida)

Árbol sintáctico con los parámetros de inmersión



- Añadir los parámetros de inmersión como argumentos de la Función Final.

Función inmersora: ~~sum + fun~~

$$f_{un-final}(x, n, i, sum) = sum + f_{un}(x, n, i)$$

Elemento neutro de la suma: 0

$$f_{un-final}(x, n, i, 0) = 0 + f_{un}(x, n, i)$$

Por tanto $f_{un-final}$ es una generalización de f_{un} , que se comporta de forma similar para los valores ^{iniciales} de los parámetros de inmersión

Y la función que realiza la llamada inicial a la función recursiva final:

entero función llamada(E Vect: x, E entero: n, E entero: i)

{ $x = A[1..n]$ $\wedge n \geq 0$ $\wedge 0 \leq i \leq n$ }

inicio

devolver fun-funl(x, n, i, 0)

fin-función

b) Desplegado.

entero función fun-funl(E Vect: x, E entero: n, E entero: i, E entero: sum)

{ $x = A[1..n]$ $\wedge n \geq 0$ $\wedge 0 \leq i \leq n$ }

inicio

si $i = 0$ entonces

devolver sum + 0

Si-no

devolver sum + $x[i]$ + fun($x, n, i-1$)

fin-si

{ devuelve la suma de los i primeros elementos del vector }

fin-función

c) Llegado

Los sucesores del parámetro de inversión utilizado en la invocación a la llamada recursiva son

$$suc(sum) = sum + x[i]$$

Por tanto, la versión Recursiva Final sería

entero función fun_fnd (Vect: x, E entero: n, E entero: i, E entero: sum)

{ $x = A[1..n]$ \wedge $n \geq 0$ \wedge $0 \leq i \leq n$ }

inicio

si $i = 0$ entonces

devolver sum + 0

Si no
devolver fun_fnd(x, n, i-1, sum + x[i])

fin-si

devolver la suma de los i primeros elementos del vector

fun-función.

Transformación de subalgoritmos recursivos

10.

Recursiva No Final a Iterativa

entero función fun-iterativa(E Vect: x, E entero n, E entero i)

{ $x = A[L..n]$, $n \geq 0$, $0 \leq i \leq n$ }

inicio

Var
entero res, c

inicio

$c \leftarrow 0$

mientras $\neg (i=0)$

$c \leftarrow c+1$

$i \leftarrow i-1$

fin-mientras

$res \leftarrow 0$

mientras $c \neq 0$ hacer

$c \leftarrow c-1$

$i \leftarrow i+1$

$res \leftarrow x[i] + res$

fin-mientras

devolver res

devuelve la suma de los i primeros elementos del vector x

fin-función

~~Optimización del primer bode muertos~~

Optimización de la función iterativa:

Para eliminar el primer bode muertos:

- El número de veces que se realiza la llamada recursiva será i veces
- El parámetro i , el último valor que toma en el caso base es 0, es posible realizar directamente esta inicialización

entero función fun(E Vect: x , E entero: n , E entero: i)

$\{x = A[L..n] \wedge n \geq 0 \wedge 0 \leq i \leq n\}$

iniciar

entero res, c

inicio

$c \leftarrow i$

$i \leftarrow 0$

$res \leftarrow 0$

mientras $c \neq 0$ hacer

$c \leftarrow c - 1$

$i \leftarrow i + L$

$res \leftarrow x[i] * res$

fin mientras

devolver res

¿devuelve la suma de los i primeros elementos del vector?

fin función

Fin a Iterativa

entonces función $fn_iterativa(E\text{ Vect: } x, E\text{ enteros: } n, E\text{ enteros: } i, E\text{ enteros: } sum)$
 $\{ x = A[1..n] \wedge n \geq 0 \wedge 0 \leq i \leq n \}$

inicio

mientras $\neg(i=0)$

$sum \leftarrow sum + x[i]$

$i \leftarrow i - 1$

fin mientras

devolver $sum + 0$

$\{ \text{devolver } 0 \text{ suma de } 0 \text{ e' primeros elementos del vector } \}$

fin función

entonces función llamada $iterativa(E\text{ Vect: } x, E\text{ enteros: } n, E\text{ enteros: } i)$

inicio

devolver $fn_iterativa(x, n, i, 0)$

fin función

Optimización de la función iterativa

- Los parámetros de inserción pueden ser eliminados como parámetros formales y ser declarados como variables locales en la función van e inicialmente al comienzo del cuerpo de la función
- La función que realiza la llamada recursiva a la función iterativa incrementando los parámetros de inserción ya no es necesaria y puede ser, por tanto, eliminada.

entonces función iterativa($\text{Vect } x$, $\text{Entero } E$, $\text{Entero } i$, $\text{Entero } \text{sum}$)
de $x = A[1:n]$ $1 \leq i \leq n$ $0 \leq \text{sum} \leq n$

var

entero sum

onico

sum $\leftarrow 0$

mientras $i > 0$ hacer

sum \leftarrow sum + $x[i]$

$i \leftarrow i - 1$

fin mientras

devolver sum + 1

devolver la suma de los i primeros elementos del vector

fin función