Costos de funciones p/arreglos

COMO DATO, TENEMOS LOS COSTOS DE LAS SIGUIENTES FUNCIONES: 1

FUNCIÓN	W	S
nth	$O(1)$	$O(1)$
length	$O(1)$	$O(1)$
fromList xs	$O(\lvert xs \rvert)$	$O(1)$
tabulate f n	$O\left(\sum_{i=0}^{n-1} W(f_i)\right)$	$O\left(\max_{i=0}^{n-1} S(f_i)\right)$
joinS s	$O(\lvert s \rvert) + \sum_{i=0}^{\lvert s \rvert -1} O(\lvert s_i \rvert)$	$O(\lg \lvert s \rvert)$

- map

map f ar = tabulate (\i → f (nth ar i)) (length ar)

SEA $g = \lambda i \rightarrow f\,(nth\ ar\ i)$

$W_g(i) = W_f(i) + O(1)$

$= \langle \cancel{W_f(i) +} O(1)$ ACOTADO POR $W_f(i) \rangle$

$W_f(i)$ $\quad$ W_g-map

$$W_{map}(f, ar) \overset{\circledast}{=} W_{tab}(g, |ar|) + O(1)$$

$$= \langle \text{def. tabulate} \rangle \; (1)$$

$$O\left(\sum_{i=0}^{|ar|-1} W_g(i)\right) + O(1)$$

$$= \langle W_g - map \rangle$$

$$O\left(\sum_{i=0}^{|ar|-1} W_f(i)\right) + O(1) = O\left(\sum_{i=0}^{|ar|-1} W_f(i)\right)$$

$$S_g(i) = S_f(i) + O(1) = S_f(i) \quad (S_g - map)$$

$$S_{map}(f, ar) \overset{\circledast}{=} S_{tab}(g, |ar|) + O(1)$$

$$= \langle \text{def. tabulate} \rangle \; (1)$$

$$O\left(\max_{i=0}^{|ar|-1} S_g(i)\right) + O(1)$$

$$= \langle S_g - map \rangle$$

$$O\left(\max_{i=0}^{|ar|-1} S_f(i)\right) + O(1) = O\left(\max_{i=0}^{|ar|-1} S_f(i)\right)$$

$\circledast$ def. map, (1)

Escaneado con CamScanner

- append

append s t = joinS (fromList [s,t])

SEA ar = fromList [s,t], |ar| CONSTANTE (|ar| = 2)

$W_{app}(s,t)$

= < def. append >

$W_{join}(ar) + W_{fromL}([s,t])$

= < def. joinS, fromList >

$$O(|ar|) + \sum_{i=0}^{|ar|-1} O(|ar_i|) + O(|ar|)$$

=

$$O(|ar|) + O(|s|) + O(t)$$

= < dist. O, O(|ar|) ACOTADO POR O(|s|+|t|) >

$$O(|s| + |t|)$$

$S_{app}(s,t)$

= < def. append >

$S_{join}(ar) + S_{fromL}([s,t])$

= < def joinS, fromL >

$O(\lg |ar|) + O(|ar|)$

$= \langle \lg |ar| = 1, |ar|$ CONSTANTE $\rangle$

$O(1)$

- reduce S

POR HIPÓTESIS, LA FUNCIÓN $\oplus \in O(1)$.

PRIMERO, SE ANALIZAN LOS COSTOS DE LA FUNCIÓN contract. (L1)

```
contract f ar n = tabulate (\i ->
    if i == div n 2
       then nth ar (i*2)
       else ⊕ (nth ar (i*2)) (nth ar (i*2 +1)))
                    (div (n+1) 2)
```

SEA g LA FUNCIÓN ARGUMENTO DE tabulate.

$W_g(i) =$

$\langle$ def g $\rangle$

$W_{\oplus}(\overbrace{nth\ ar\ (i*2),\ nth\ ar\ (i*2+1)}^{nth}) + O(1)$

$\langle$ HIPÓTESIS $\rangle$

$O(1)$ $\quad (W_g$- contract$)$

Escaneado con CamScanner

$W_{cont}(\theta, n)$

$= \langle$ def. contract $\rangle$

$W_{tab}\left(g, \lfloor \frac{n+1}{2} \rfloor\right)$

$= \langle$ def. ① tabulate, div ~~W_g - contract~~ $\rangle$

$O\left(\sum_{i=0}^{\frac{n+1}{2}-1} W_g(i)\right) + O(1)$

$= \langle W_g$ - contract $\rangle$

$O\left(\sum_{i=0}^{\frac{n+1}{2}-1} O(1)\right) + O(1)$

$= \langle$ AGRUPAR ~~ASOCIATIVA~~ $\rangle$

$O\left(\sum_{i=0}^{\frac{n+1}{2}} O(1)\right)$

$= \langle O(1) \times \left(\frac{n+1}{2}\right) = O\left(\frac{n+1}{2}\right), \frac{n+1}{2}$ ES $O(n) \rangle$

$O(n)$

LA DEMOSTRACIÓN DE $S_g(i)$ ES ANÁLOGA A W_g. POR LO TANTO, $S_g = O(1)$ (S_g - contract)

$S_{cont}(\oplus, n)$

$= \langle$ def. contract $\rangle$

$S_{tab}(g, \lfloor \frac{n+1}{2} \rfloor)$

$= \langle$ def ① tabulate, div $\rangle$

$\max_{i=0}^{\lfloor \frac{n+1}{2} \rfloor - 1} S_g(i) + O(1)$

$= \langle S_g$ - contract $\rangle$

$\max_{i=0}^{\lfloor \frac{n+1}{2} \rfloor - 1} O(1) + O(1)$

$\langle$ def. max $\rangle$

$O(1)$

```
reduce ⊕ e ar
  | isEmpty ar = e
  | isSingleton ar = ⊕ e (first ar)
  | otherwise = let arCont = contract ⊕ ar (length ar)
                in reduce ⊕ e arCont
```

LAS DOS PRIMERAS CONDICIONES DE reduce DESCRIBEN APLICACIONES A TRABAJO Y PROFUNDIDAD CONSTANTE.

SE DEMUESTRA ~~ANÁLISIS~~ EL COSTO RESPECTO A LA ~~FUNCIÓN~~ RESTANTE.

CONDICIÓN

Escaneado con CamScanner

$W_{red}(\oplus, ar)$

$= \langle$ def. reduce $\rangle$

$W_{cont}(\oplus, |ar|) + W_{red}(\oplus, \frac{|ar|}{2})$

$= \langle$ ~~def. contract~~ L1 $\rangle$

$O(n) + W_{red}(\oplus, \frac{|ar|}{2})$

$= \langle$ TEOREMA MAESTRO $\rangle$

$O(n)$

$S_{red}(\oplus, ar)$

$= \langle$ def. reduce $\rangle$

$S_{cont}(\oplus, ar) + S_{red}(\oplus, \frac{|ar|}{2})$

$= \langle$ ~~def. contract~~ $\rangle$ $\langle$ L1 $\rangle$

$O(1) + S_{red}(\oplus, \frac{|ar|}{2})$

$=$

$O(\lg |ar|)$

. scanS

POR HIPÓTESIS, LA FUNCIÓN $\oplus \in O(1)$.

PRIMERO, SE ANALIZAN LOS COSTOS DE LA FUNCIÓN expand (L2)

Escaneado con CamScanner

```
expand ⊕ arCont arOr n =
  tabulate (λ i →
    if mod i 2 == 0
      then nth arCont (div i 2)
      else let a = nth arCont (div i 2)
               b = nth arOr (i-1)
           in ⊕ a b) n
```

SEA g LA FUNCIÓN ARGUMENTO DE tabulate.

POR HIPÓTESIS, PODEMOS AFIRMAR QUE $g \in O(1)$ (g-scanS)

$W_{exp}(\oplus, n)$

= < def. expand >

$W_{tab}(g, n)$

= < def. ① tabulate >

$O\left(\sum_{i=0}^{n-1} W_g(i)\right)$

= < g-scanS >

$O\left(\sum_{i=0}^{n-1} O(1)\right)$

= < $O(1) \times n$ ES $O(n)$ >

$O(n)$

$S_{exp}(\oplus, n)$

= < def. expand >

$S_{exp}(g, n)$

= < def. ① tabulate >

$O\left(\max_{i=0}^{n-1} S_g(i)\right)$

= < g-scanS >

$O\left(\max_{i=0}^{n-1} O(1)\right)$

= < def. max >

$O(1)$

Escaneado con CamScanner

```
scanS ⊕ e ar
  | isEmpty ar = (ar, e)
  | isSingleton ar = let el = first ar
                     in (singleton e, ⊕ e el)

  | otherwise = let nElem = length ar
                    arCont = contract f ar nElem
                    (r, t) = scan ⊕ e arCont
                in (expand ⊕ r ar nElem, t)
```

AL IGUAL QUE EN reduceS, SE DEMUESTRA EL COSTO PARA LA ÚLTIMA CONDICIÓN DE LA FUNCIÓN scanS.

$W_{scan}(\oplus, n)$

$= \langle$ def. scanS $\rangle$

$W_{len}(ar) + W_{cont}(\oplus, n) + W_{scan}(\oplus, \frac{n}{2}) + W_{exp}(\oplus, n)$

$= \langle$ def ① length, L1, L2 $\rangle$

$O(1) + O(n) + W_{scan}(\oplus, \frac{n}{2})$

$= \langle O(1)$ ACOTADO POR $O(n) \rangle$

$O(n) + W_{scan}(\oplus, \frac{n}{2})$

$= \langle$ TEOREMA MAESTRO $\rangle$

$O(n)$

Escaneado con CamScanner

$S_{scan}(\oplus, n)$

$= \langle \text{def. scan}S \rangle$

$S_{len}(ar) + S_{cont}(\oplus, n) + S_{scan}(\oplus, \frac{n}{2}) + S_{exp}(\oplus, n)$

$= \langle \text{def. } ① \text{ length, } L1, L2 \rangle$

$O(1) + S_{scan}(\oplus, \frac{n}{2})$

$=$

$O(\lg n)$

SEBASTIÁN MOLINÉ

SEBASTIÁN MORALES

Escaneado con CamScanner