



Tema 8. Evaluación perezosa.

Supongamos la *definición* de una función:

$$\begin{array}{c} f \ x \ y = \frac{x*2+y}{\text{cuerpo}} \\ \text{parámetros} \nearrow \nearrow \end{array}$$

Pasos de evaluación de la expresión ? f 3 4

$\nearrow \nearrow$ argumentos

$$f \ 3 \ 4 \Rightarrow 3*2 + 4 \Rightarrow 6+4 \Rightarrow 10$$

- Definición usada como regla de reducción (izda -> dcha)
- Sustitución de parámetros por argumentos en el cuerpo.



Ev. Impaciente versus Perezosa

$$f\ x\ y = x*2+y$$

$$g\ x = x+10$$

Impaciente

? $f\ 3\ (g\ 5)$



$f\ 3\ (5+10)$



$f\ 3\ 15$



$3*2+ 15$



$6+15$



21

Perezosa

? $f\ 3\ (g\ 5)$



$3*2+(g\ 5)$



$6 + (g\ 5)$



$6 + (5+10)$



$6+15$



21



Estrategias de reducción: Teorema 1

Evaluación Impaciente:

- Elección del “redex” más interno (más izq.)
- Evaluar argumentos y aplicar después la función

Evaluación Perezosa:

- Elección del “redex” más externo (más izq.)
- Aplicar la función a los argumentos sin evaluar éstos (necesariamente)



Teorema 1: “El resultado (ó forma normal) de una expresión es único (si existe).”

Evaluación Perezosa (1)

$$h \ x \ y = x+12$$

$$g \ x = x+10$$

Impaciente

? $h \ 6 \ (g \ 5)$

↓
 $h \ 6 \ (5+10)$

↓
 $h \ 6 \ 15$

↓
 $6+12$

↓
18

Perezosa

? $h \ 6 \ (g \ 5)$

↓
 $6+12$

↓
18

- Argumentos se evalúan sólo si es necesario
- Funciones no-estrictas

Evaluación Perezosa (2)

$f [] = 0$

$f (x:s) = x+3$

Impaciente

? $f (\text{filter } (/=0) [6,0,1,2])$

↓
 $f (6: \text{filter } (/=0) [0,1,2])$

↓
 $f (6: \text{filter } (/=0) [1,2])$

⋮
↓
 $f (6:1:2:[])$
⋮
↓
9

Perezosa

? $f (\text{filter } (/=0) [6,0,1,2])$

↓
 $f (6: \text{filter } (/=0) [0,1,2])$

↓
 $6+3$
↓
9

➤ Argumentos se evalúan parcialmente (demanda)

Evaluación Perezosa (3)

$$h\ x = x + x$$

Impaciente

$$? \ h\ (3*2+8)$$

$$\downarrow$$
$$h\ (6+8)$$

$$\downarrow$$
$$h\ 14$$

$$\downarrow$$
$$14 + 14$$

$$\downarrow$$
$$28$$

Perezosa

$$? \ h\ (3*2+8)$$

$$\downarrow$$
$$\underline{x} + x \text{ where } x = 3*2+8$$

$$\downarrow$$
$$\underline{x} + x \text{ where } x = 6+8$$

$$\downarrow$$
$$14 + 14$$

$$\downarrow$$
$$28$$

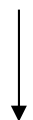
- Argumentos se evalúan como máximo una vez
Implementación: mediante “Reducción de grafos”



Evaluación Perezosa (4)

f x y	x > y	= a
	otherwise	= b
where a = x*6+y ; b = (x+y)*3		

? f 5 2



?? 5 > 2 → True

a where a = 5*6+2



a where a = 30+2



32

- La evaluación perezosa se aplica también a las definiciones locales

Estrategias de reducción: Teorema 2

$f \ x \ y = y+10$

$loop = loop$

Impaciente

? $f \ \underline{loop} \ 3$



$f \ \underline{loop} \ 3$



$f \ \underline{loop} \ 3$



(bucle)

Perezosa

? $f \ \underline{loop} \ 3$



$\underline{3 + 10}$



13



Teorema 2: “Si una expresión tiene forma normal, la evaluación perezosa la encuentra”

Reglas de cálculo de la evaluación (1)

La *definición general* de una función f es:

$f \ p_1 \ p_2 \ \dots \ p_k$ (p_i son patrones)

	g_{11}	$= e_{11}$
	g_{12}	$= e_{12}$
.....		
	g_{1n}	$= e_{1n}$
where defs-locales1		

$f \ q_1 \ q_2 \ \dots \ q_k$ (q_i son patrones)

	g_{21}	$= e_{21}$
	
where defs-locales2		

etc....



Reglas de cálculo de la evaluación (2)

Al evaluar $? f a_1 a_2 \dots a_k$ (a_i son argumentos) se realizan los siguientes cálculos:

- *Ajuste de patrones:* determina qué ecuación usar de la definición de f
 $\{a_1 a_2 \dots a_k$ ajustan con $p_1 p_2 \dots p_k$ resp?
y sustituye parámetros por argumentos en la parte derecha correspondiente
- *Guardas:* se evalúan de arriba abajo y la primera cierta determina qué expresión devuelve f
- *Definiciones locales:* se calculan por demanda (sólo cuando se necesitan)



Ejercicios de evaluación (1)

- *Ajuste de patrones*: Dar los pasos al evaluar $f [1..3] [5..8]$ siendo f la función definida:

$$f [] t = 0$$

$$f (x:s) [] = 0$$

$$f (x:s) (y:t) = x+y$$

- *Guardas*: Dar los pasos al evaluar $h (2+3) (4-1) (3+9)$ siendo h la función definida:

$$h m n p \mid m \geq n \ \&\& \ m \geq p \quad = m$$

$$\mid n \geq m \ \&\& \ n \geq p \quad = n$$

$$\mid \text{otherwise} \quad = p$$



Ejercicios de evaluación (2)

- *Definiciones locales*: Dar los pasos al evaluar **f 3 5** siendo **f**, **frente** y **notNil** las funciones definidas:

```
f m n
  | notNil lis = frente lis
  | otherwise = n
  where lis = [m..n]
```

```
frente (x:y:s) = x+y
```

```
frente [x]     = x
```

```
notNil []      = False
```

```
notNil (x:s)   = True
```

Evaluación perezosa y listas infinitas

El programa `suma2` se puede aplicar a listas infinitas:

```
suma2 (x:y:s) = x+y  
suma2 _ = 0
```

Impaciente

? suma2 [3..]
↓
suma2 (3:[4..])
⋮
(bucle)

Perezosa

? suma2 [3..]
↓
suma2 (3:[4..])
↓
suma2 (3:4:[5..])
↓
3+4
↓
7



Función predefinida “iterate”

➤ Función “iterate”

Idea: $\text{iterate } f \ x = [x, f \ x, f(f \ x), f(f(f \ x)), \dots]$

Definición: $\text{iterate } f \ x = x : \text{iterate } f \ (f \ x)$

➤ Ejemplo

Queremos definir $\text{digitos} :: \text{Int} \rightarrow [\text{Int}]$ tal que

? $\text{digitos } 3925$

$[3, 9, 2, 5]$

Una solución:

```
digitos = reverse • map (`mod` 10) •  
          takeWhile (/=0) • iterate (`div` 10)
```



Ejemplo: criba de Eratóstenes

Algoritmo para obtener los números primos.

Idea:

[2,3,4,5,6,7,8,9,10,11,12,13,.....]

↓ *criba (borramos los múltiplos de la cabeza)*

[3,5,7,9,11,13,15,17,19,]

↓ *criba (borramos los múltiplos de la cabeza)*

[5,7,11,13,17,19,23,25,]

↓
Programa:

```
primos = map head (iterate criba [2..])
```

```
  where criba (x:s) = [y | y <- s, y `mod` x /= 0]
```



Ejercicios: evaluación con listas infinitas

Dar los pasos de evaluación de las siguientes expresiones:

- `head (map (^2) [3..])`
- `filter (<10) [8..]`
- `takeWhile (<10) [8..]`
- `dropWhile (<10) [8..]`
- `takeWhile (/=0) (map (+3) [-5..])`
- `take 3 [2^ i | i <- [1..]]`
- `digitos 3925`
- `take 10 primos`