

EVALUACIÓN DE EXPRESIONES
AJUSTE DE PATRONES
FUNCIONES ESTRUCTURADAS

Nociones básicas

Curso 2021/22

Susana Nieva Soto

Transparencia referencial

- En los lenguajes funcionales puros como Haskell, la evaluación de una expresión **nunca** produce un **efecto colateral**.
- El resultado de evaluar una expresión e , es **independiente del contexto**
- La evaluación de e produce un valor
- Valor =
 - Constante (constructora de datos de aridad 0) :
`True, False, -2147483648, . . . , -1, 0, 1, . . . , 2147483647, [], ()`
 - Aplicación de una constructora de datos de aridad n a n valores :
`(True, []) (0, 5, 7) [-214, 74] [[('a',1), ('c',1)]]`

Evaluación de expresiones (lo básico)

- Todo tipo T denota un conjunto de valores \mathcal{T} .
 $Bool = \{True, False\}$ ¿Qué denota $[Int]$?
- Toda expresión $e :: T$ sintácticamente correcta tiene un valor v , dentro del conjunto \mathcal{T} , denotado por su tipo. Notación $[[e]] = v$

$[[3 + 1]] = 4$ $[[(2 < 3, [])]]$ = (True, []) $[[suc 'a']] = 'b'$

$[[if 2 < 3 then [3 + 1] else []]]$ = [4]

$[[let x = 2 in x * 3]] = [[(x * 3)[x/2]]]$ = 6

¿Cómo se evalúan las aplicaciones $(e1\ e2)$? Depende de la definición de la función $e1$

Tipo de las aplicaciones

$[[(e1\ e2)]] = v$ ¿cómo se obtiene v ?

$e1 :: T \rightarrow T' \quad e2 :: T \quad (e1\ e2) :: T'$

El valor v está en \mathcal{T}' (conjunto denotado por el tipo T')

Además la función puede tener varios argumentos

Función *currifcada*

$f :: T1 \rightarrow T2 \rightarrow \dots \rightarrow Tn \rightarrow T = (T1 \rightarrow (T2 \rightarrow \dots \rightarrow (Tn \rightarrow T) \dots))$

$f\ e1\ e2\ \dots\ en = (\dots((f\ e1)\ e2)\dots en)$

$[[f\ e1]] : \mathcal{T}'2 \rightarrow \dots \rightarrow \mathcal{T}'n \rightarrow \mathcal{T}' \quad e1 :: T1, e2 :: ?$

...

$\dots en-1 :: Tn-1$

$[[f\ e1\ e2\ \dots\ en-1]] : \mathcal{T}'n \rightarrow \mathcal{T}'$

$[[f\ e1\ \dots\ en-1\ en]]$ es un valor de \mathcal{T}' siempre que $en :: Tn$

Este valor depende de la definición de f

Definición ecuacional de una función

- Sucesión de ecuaciones con guardas o no:
 $f :: T1 \rightarrow \dots \rightarrow Tn \rightarrow T$ (declaración de tipo aconsejable)
 $f \ p_1 \dots p_n = e_1$ $p_1 \dots p_n$ patrones lineales
... (sin variables en común)
 $f \ p'_1 \dots p'_n$
| $b_1 = e_{x1}$
| $b_2 = e_{x2}$
...
| $b_k = e_{xk}$
...

Patrones Haskell

Un patrón p puede tener las siguientes formas:

- x identificador de variable
- $_$ variable anónima
- C constructora de aridad 0 (constante)
- $p1 \dots pn$ sucesión de patrones
- $C p1 \dots pn$ constructora de aridad n aplicada a n patrones

$(x:xs) \cong (\colon) x xs$ (\colon) constructora de listas, x , xs variables

Evaluación de una expresión funcional

$f :: T1 \rightarrow T2 \rightarrow \dots \rightarrow Tn \rightarrow T$

$f\ e1\ e2\ \dots\ en = (\dots((f\ e1)\ e2)\dots en)$ (asocia por la izquierda)

$[[f\ e1\ e2\ \dots\ en]] = \text{valor perteneciente a } \mathcal{T}$

1. Se busca la primera ecuación de la def. de f cuyo lado izquierdo $f\ p1\dots pn$ sea tal que los parámetros actuales $e1\ \dots\ en$ *ajusten* con los parámetros formales $p1\dots pn$: *Ajuste de patrones*
2. Se busca la primera guarda (si las hay) para el caso $f\ p1\dots pn$ que se evalúe a True, teniendo en cuenta la *sustitución de ajuste*.
3. Se evalúa la expresión de la derecha de la ecuación que cumple las condiciones anteriores, teniendo en cuenta la *sustitución de ajuste*.

Ajuste de una expresión a un patrón

e se ajusta al patrón p si *tiene la forma* de p al sustituir adecuadamente las variables de p por otras expresiones :

- x cualquier expresión e ajusta con x . Sustitución de ajuste $[x/e]$
- $_$ cualquier expresión e ajusta con $_$ No produce sustitución de ajuste
- C el valor de e tiene que ser igual a C . No produce sustitución de ajuste
- $p_1 \dots p_n$ ajustan con él las expresiones de la forma $e_1 \dots e_n$ si las expresiones e_i ajustan con los patrones p_i ($1 \leq i \leq n$).
Sustitución de ajuste = reunión de las sustituciones de los n ajustes
- $C p_1 \dots p_n$ ajustan con él las expresiones de la forma $C e_1 \dots e_n$ si cada e_i ajusta con p_i ($1 \leq i \leq n$).
Sustitución de ajuste = reunión de las sustituciones de los n ajustes

$[1,2,3]$ ajusta con $(x:xs)$ y con $(_:xs)$, pero no con $(_:[])$

Ejemplo: evaluación de $(f\ e1\ \dots\ en)$

$f :: \text{Int} \rightarrow \text{Int} \rightarrow \text{Int}$

$f\ 0\ 1 = 2$

$f\ x\ y$

| $x > 0 = y$

| otherwise = x

— ¿Cuánto vale $f\ (1-1)\ 1$? A partir de la definición de f :

El valor de $(1-1)$ ajusta con 0 y 1 ajusta con 1 (1ª ecuación)

Se evalúa la parte derecha de esta ecuación que da 2

— ¿Cuánto vale $f\ (1-1)\ 3$?

El valor de $(1-1)$ ajusta con 0 pero 3 no ajusta con 1

Hay ajuste de patrones con la 2ª ecuación

La sustitución de ajuste es $[x/(1-1), y/3]$

La primera guarda de la 2ª ecuación que se hace cierta es **otherwise**

Se evalúa su parte derecha con la sustitución $[[x[x/(1-1)]]] = 0$

— ¿Cuánto vale $f\ (1-2)\ (3-2)$? ¿Se evalúa $3-2$, y $1-2$?

Ejemplo: evaluación de $(f' \ e1 \ \dots \ en)$

$f' :: \text{Int} \rightarrow \text{Int} \rightarrow \text{Int}$

$f' \ x \ y$

| $x > 0 = y$

| otherwise = x

$f' \ 0 \ 1 = 2$

— ¿Cuánto vale $f' \ (1-1) \ (4-1)$?

$(1-1)$ ajusta con x , $(4-1)$ ajusta con y sin necesidad de evaluar (1ª ecuación). Sustitución de ajuste $\sigma = [x/(1-1), y/(4-1)]$

Se evalúa la guarda con $\sigma \ [x[x/(1-1)] > 0] = \text{False}$

Se evalúa la siguiente guarda **otherwise** que da **True**

El resultado es $[x[x/(1-1)]] = 0$

No se ha evaluado el segundo parámetro.

— ¿Cuánto vale $f' \ (1-1) \ 1$?

— ¿Cuánto vale $f' \ (2-1) \ (4-1)$?

El valor indefinido

- Cuando una expresión no puede evaluarse porque da un error de ejecución o es infinita se dice que está indefinida, $[[e]] = \perp$ (*bottom*).
- A las expresiones e tales que $[[e]] = \perp$, las identificamos con *bottom*.
- Todas las siguientes expresiones son *bottom*:
 - `div 1 0`
 - `undefined`
 - `head []`
 - `error "mensaje de error"`
 - `c = c`
 -
- Admitimos \perp como elemento de cualquier \mathcal{T} .
- ¿Cuál es el tipo de la expresión error?

Funciones estrictas

- $f :: T \rightarrow T'$ es estricta cuando:

$$[[f \text{ bottom}]] = \perp$$

- $f :: T_1 \rightarrow \dots \rightarrow T_n \rightarrow T$ es estricta en el i -ésimo argumento cuando:

$$[[f \ e_1 \ \dots \ e_{i-1} \ \text{bottom} \ e_{i+1} \ \dots \ e_n]] = \perp$$

para cualquier valor de los restantes argumentos.

Con las definiciones anteriores de f y f' :

¿Cuál es el valor de $(f \ 0 \ \text{undefined})$ y de $(f' \ 0 \ \text{undefined})$?

¿Es f estricta en alguno de sus argumentos?

¿Es f' estricta en alguno de sus argumentos?