

CP - Ficha 6

Exercício 1

O código que se segue, escrito em Haskell, implementa a noção de ciclo-for, onde b é o corpo (“body”) do ciclo e i é a sua inicialização:

$$\begin{cases} \text{for } b \text{ i } 0 = i \\ \text{for } b \text{ i } (n + 1) = b (\text{for } b \text{ i } n) \end{cases} \quad (\text{F1})$$

Mostre que

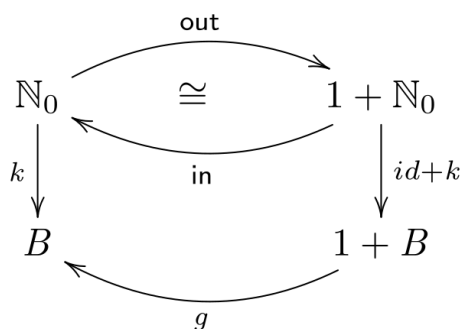
$$\text{for } b \text{ i } = \llbracket g \rrbracket \quad (\text{F2})$$

se verifica, para um dado g (descubra qual). Sugestão: recorra à lei universal

$$k = \llbracket g \rrbracket \Leftrightarrow k \cdot \text{in} = g \cdot (\text{id} + k)$$

abordada na aula teórica, onde

$$\begin{cases} \text{in} = [\text{zero}, \text{succ}] \\ \text{zero } _ = 0 \\ \text{succ } n = n + 1 \end{cases} \quad (\text{F3})$$



Resolução 1

Let $\langle g \rangle = \langle [g_1, g_2] \rangle$

for $b \ i = \langle [g_1, g_2] \rangle$

\equiv (46: Universal-cata)

$(\text{for } b \ i) \cdot \text{in} = [g_1, g_2] \cdot (\text{id} + (\text{for } b \ i))$

\equiv (Def. in, 22: Absorção-+, 1: Natural-id)

$(\text{for } b \ i) \cdot [\underline{0}, \text{succ}] = [g_1, g_2 \cdot (\text{for } b \ i)]$

\equiv (20: Fusão-+)

$[(\text{for } b \ i) \cdot \underline{0}, (\text{for } b \ i) \cdot \text{succ}] = [g_1, g_2 \cdot (\text{for } b \ i)]$

\equiv (27:Eq-+, 4: Absorção-const)

$$\begin{cases} \underline{\text{for } b \ i \ 0} = g_1 \\ (\text{for } b \ i) \cdot \text{succ} = g_2 \cdot (\text{for } b \ i) \end{cases}$$

\equiv (72: Ig. Ext., 73: Def-comp, Def. succ, 75: Def-const)

$$\begin{cases} \text{for } b \ i \ 0 = g_1 \ x \\ \text{for } b \ i \ (n + 1) = g_2 \ (\text{for } b \ i \ n) \end{cases}$$

\Rightarrow

$$\begin{cases} g_1 = \underline{i} \\ g_2 = b \end{cases}$$

Logo: $\text{for } b \ i = \langle [\underline{i}, b] \rangle$

Exercício 2

Na sequência da questão anterior, codifique

$$f = \pi_2 \cdot aux \quad \textbf{where } aux = \text{for } \langle \text{succ} \cdot \pi_1, \text{mul} \rangle (1, 1) \quad (\text{F4})$$

em Haskell e inspecione o seu comportamento. Que função f é essa?

Resolução 2

Pelo exercício anterior, temos:

$$aux = \llbracket \underline{(1, 1)}, \langle \text{succ} \cdot \pi_1, \text{mul} \rangle \rrbracket$$

$$\llbracket \underline{(1, 1)}, \langle \text{succ} \cdot \pi_1, \text{mul} \rangle \rrbracket : 1 + \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{N}_0 \times \mathbb{N}_0$$

```
aux 0 = (1, 1)
aux 1 = (2, 1)
aux 2 = (3, 2)
aux 3 = (4, 6)
aux 4 = (5, 24)
aux 5 = (6, 120)
```

```
aux 0 = (1, 1)
aux (n + 1) = (pi_1 (aux n) + 1, mul (aux n))
```

A função f n é o factorial de n .

Exercício 3

Mostre que $(a+)$ dada a seguir é um ciclo-for b i (F1) para um dado b e um dado i — descubra quais:

$$\begin{cases} a + 0 = a \\ a + (n + 1) = 1 + (a + n) \end{cases} \quad (\text{F5})$$

Resolução 3

$$\begin{aligned} & \begin{cases} a + 0 = a \\ a + (n + 1) = 1 + (a + n) \end{cases} \\ & \equiv \quad (72: \text{Ig. Ext.}, 75: \text{Def-const}, \text{Def. succ}) \\ & \begin{cases} (a+) (\underline{0} \ n) = \underline{a} \ n \\ (a+) (\text{succ } n) = \text{succ } ((a+) \ n) \end{cases} \\ & \equiv \quad (74: \text{Def-comp}, 72: \text{Ig. Ext.}) \\ & \begin{cases} (a+) \cdot \underline{0} = \underline{a} \\ (a+) \cdot \text{succ} = \text{succ} \cdot (a+) \end{cases} \\ & \equiv \quad (27: \text{Eq-+}) \\ & [(a+) \cdot \underline{0}, (a+) \cdot \text{succ}] = [\underline{a}, \text{succ} \cdot (a+)] \\ & \equiv \quad (20: \text{Fusão-+}, 1: \text{Natural-id}, 22: \text{Absorção-+}) \\ & (a+) \cdot [\underline{0}, \text{succ}] = [\underline{a}, \text{succ}] \cdot (id + (a+)) \\ & \equiv \quad (46: \text{Universal-cata}) \\ & (a+) = ([\underline{a}, \text{succ}]) \\ & \equiv \quad (\text{F2}) \\ & (a+) = \text{for succ } a \end{aligned}$$

Exercício 4

Recorde a lei de “fusão-cata”:

$$f \cdot \langle g \rangle = \langle h \rangle \Leftarrow f \cdot g = h \cdot (id + f) \quad (F6)$$

deduzida na aula teórica. Recorra a (F6) para demonstrar a propriedade:

$$f \cdot (\text{for } f \ i) = \text{for } f \ (f \ i)$$

sabendo que $\text{for } f \ i = \langle [i, f] \rangle$.

Resolução 4

$$f \cdot (\text{for } f \ i) = \text{for } f \ (f \ i)$$

\equiv

(F2)

$$f \cdot \langle [i, f] \rangle = \langle [f \ i, f] \rangle$$

\Leftarrow

(49: Fusão-cata (F6))

$$f \cdot [i, f] = [f \ i, f] \cdot (id + f)$$

\equiv

(20: Fusão-+, 22: Absorção-+)

$$[f \cdot i, f \cdot f] = [f \ i \cdot id, f \cdot f]$$

\equiv

(4: Absorção-const, 1: Natural-id)

$$[f \ i, f \cdot f] = [f \ i, f \cdot f]$$

Exercício 5

Mostre que as funções:

$$f = \text{for } id \ i$$

$$g = \text{for } \underline{i} \ i$$

são a mesma função. (Qual?)

Resolução 5

$$g = \text{for } \underline{i} \ i$$

$$\equiv$$

(F2)

$$g = ([\underline{i}, \underline{i}])$$

$$\equiv$$

(46: Universal-cata)

$$g \cdot \text{in} = [\underline{i}, \underline{i}] \cdot (id + g)$$

$$\equiv$$

(Isomorfismo in/out, 33: Shunt-left)

$$g = [\underline{i}, \underline{i}] \cdot (id + g) \cdot \text{out}$$

$$\equiv$$

(1: Natural-id, 20: Fusão-+)

$$g = \underline{i} \cdot [id, id] \cdot (id + g) \cdot \text{out}$$

$$\equiv$$

(3: Fusão-const)

$$g = \underline{i}$$

$$\underline{i} = \text{for } id \ i$$

$$\equiv$$

(F2)

$$\underline{i} = ([\underline{i}, id])$$

$$\equiv$$

(46: Universal-cata)

$$\underline{i} \cdot \text{in} = [\underline{i}, id] \cdot (id + \underline{i})$$

$$\equiv$$

(3: Fusão-const, 2w: Absorção-+)

$$\underline{i} = [\underline{i}, \underline{i}]$$

$$\equiv$$

(1: Natural-id, 20: Fusão-+, 3: Fusão-const)

$$\underline{i} = \underline{i}$$

Exercício 6

Considere o catamorfismo $rep\ f = ([id, (f \cdot)])$. Comece por fazer um diagrama do catamorfismo e responda:

Qual é o tipo de rep ? O que faz rep ?

Usando o combinador `cataNat g` da biblioteca `Nat.hs` para implementar (g) , avalie no GHCi expressões como, por exemplo `rep (2*) 0 3` e `rep ("a"++) 10 "b"` e veja se os resultados confirmam as suas respostas acima.

Resolução 6

TODO

Exercício 7

Qualquer função $k = \text{for } f \ i$ pode ser codificada em sintaxe C escrevendo:

```
int k(int n) {  
    int r = i;  
    int j;  
    for (j = 1; j < n + 1; j++) { r = f(r); }  
    return r;  
}
```

Escreva em sintaxe C as funções $(a*) = \text{for } (a+) \ 0$ e outros catamorfismos de naturais de que se tenha falado nas aulas da UC.

Resolução 7

TODO

Exercício 8

Questão prática

Problem requirements: The following function:

```
func :: Eq a => b -> [(a, b)] -> (a -> b)
func b = (maybe b id .) . flip lookup
```

“functionalizes” a finite list of (key, value) pairs by converting it to a function from keys to values. The first parameter provides a default value for keys that cannot be found in the list.

As example, let us have a list of people (where the key is some numeric id):

```
a = [(140999000, "Manuel"), (200100300, "Mary"), (000111222, "Teresa")]
```

their nationalities (if known):

```
b = [(140999000, "PT"), (200100300, "UK")]
```

and places of residence (if known):

```
c = [(140999000, "Braga"), (200100300, "Porto"), (151999000, "Lisbon")]
```

Using only *func*, $\langle f, g \rangle$, π_1 , π_2 , *map* and *nub*, write a Haskell expression representing the following data aggregation:

Id	Name	Country	Residence
140999000	Manuel	PT	Braga
200100300	Mary	UK	Porto
000111222	Teresa	?	-
151999000	(Unknown)	?	Lisbon

Resolução 8

TODO

