

Cálculo de Programas

Algebra of Programming

UNIVERSIDADE DO MINHO
Lic. Ciências da Computação (2º ano)
Lic. em Engenharia Informática (3º ano)

2024/25 - Ficha (*Exercise sheet*) nr. 6

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:

The following Haskell code implements a for-loop where b is the loop-body and i is its initialization:

$$\left\{ \begin{array}{l} \text{for } b \ i \ 0 = i \\ \text{for } b \ i \ (n + 1) = b \ (\text{for } b \ i \ n) \end{array} \right. \quad (\text{F1})$$

Mostre que

Show that

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

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

*holds for some g (find which). **Hint:** resort to the universal law*

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

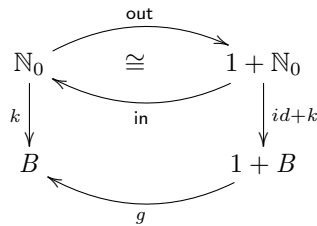
abordada na aula teórica, onde

approached in the theory class, where

$$\left\{ \begin{array}{l} \text{in} = [\text{zero}, \text{succ}] \\ \text{zero } _ = 0 \\ \text{succ } n = n + 1 \end{array} \right. \quad (\text{F3})$$

cf. o diagrama seguinte:

cf. the following diagram:



2. Na sequência da questão anterior, codifique

As follow up of the previous question, encode

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

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

in Haskell and inspect its behavior. Which function is f ?

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

Show that $(a+)$ given next is a for-loop for b i (F1) for b and i to be calculated:

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

4. Recorde a lei de “fusão-cata”

Recall the “fusion-law”

$$f \cdot \langle g \rangle = \langle h \rangle \iff f \cdot g = h \cdot (id + f) \quad (\text{F6})$$

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

proved in the theory class. Use (F6) to prove the property:

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

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

knowing: for $f \ i$ is $\langle [i, f] \rangle$.

5. Mostre que as funções

Show that functions

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

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

são a mesma função. (Qual?)

are the same function. (Which one?)

6. Considere o catamorfismo $rep \ f = \langle [id, (f \cdot)] \rangle$. Comece por fazer um diagrama do catamorfismo e responda: Qual é o tipo de rep ? O que faz rep ?

Consider catamorphism $rep \ f = \langle [id, (f \cdot)] \rangle$. Draw a diagram of this catamorphism and answer: What is the type of rep ? What does rep do?

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

Using `cataNat g` from library `Nat.hs` to implement $\langle g \rangle$, evaluate in GHCi expressions like $rep \ (2) \ 0 \ 3$, $rep \ ("a"++) \ 10 \ "b"$ and check if the results confirm your answers above.*

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

Any function $k = \text{for } f \ i$ can be encoded in the syntax of C by writing:

```
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.

Encode function $(a*) = \text{for } (a+) 0$ in C and other catamorphisms that have been discussed in the previous classes.

8. **Questão prática** — Este problema não irá ser abordado em sala de aula. Os alunos devem tentar resolvê-lo em casa e, querendo, publicarem a sua solução no canal **#geral** do Slack, com vista à sua discussão com colegas. Dão-se a seguir os requisitos do problema.

Open assignment — This assignment will not be addressed in class. Students should try to solve it at home and, wishing so, publish their solutions in the **#geral** Slack channel, so as to trigger discussion among other colleagues. The requirements of the problem are given below.

Problem requirements: The following function

$$\begin{aligned} \text{func} &:: \text{Eq } a \Rightarrow b \rightarrow [(a, b)] \rightarrow (a \rightarrow b) \\ \text{func } b &= (\text{maybe } b \text{ id}) \cdot \text{flip lookup} \end{aligned}$$

“functionalizes” a finite list of $(\text{key}, \text{value})$ 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),

$$a = [(140999000, \text{"Manuel"}), (200100300, \text{"Mary"}), (000111222, \text{"Teresa"})]$$

their nationalities (if known),

$$b = [(140999000, \text{"PT"}), (200100300, \text{"UK"})]$$

and places of residence (if known):

$$c = [(140999000, \text{"Braga"}), (200100300, \text{"Porto"}), (151999000, \text{"Lisbon"})]$$

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

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

□