Cálculo de Programas
Algebra of Programming

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

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

   Mostre que

   Show that

$$\mathsf{for}\ b\ i = (\!|\ g\ |\!) \tag{F2}$$

   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*

$$k = (\!|\ g\ |\!) \quad \Leftrightarrow \quad k \cdot \mathsf{in} = g \cdot (id + k)$$

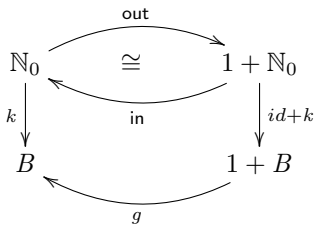
   abordada na aula teórica, onde

   approached in the theory class, where

$$\begin{cases} \mathsf{in} = [\mathsf{zero}\ , \mathsf{succ}] \\ \mathsf{zero}\ \_ = 0 \\ \mathsf{succ}\ n = n+1 \end{cases} \tag{F3}$$

   cf. o diagrama seguinte:

   cf. the following diagram:



2. Na sequência da questão anterior, codifique

As follow up of the previous question, encode

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

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:*

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

---

4. Recorde a lei de "fusão-cata"

Recall the "fusion-law"

$$f \cdot (\!|\, g \,|\!) = (\!|\, h \,|\!) \quad \Leftarrow \quad f \cdot g = h \cdot (id + f) \tag{F6}$$

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

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

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

sabendo que for $f\ i$ is $(\!|\, [\underline{i}\ , f] \,|\!)$.

knowing: for $f\ i = (\!|\, [\underline{i}\ , f] \,|\!)$.

---

5. Mostre que as funções

Show that functions

$$f = \mathsf{for}\ id\ i$$
$$g = \mathsf{for}\ \underline{i}\ i$$

são a mesma função. (Qual?)

are the same function. (Which one?)

---

6. Considere o catamorfismo $rep\ f = (\!|\, [\underline{id}\ , (f \cdot)] \,|\!)$. 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 $rep\ (2*)\ 0\ 3$, $rep\ (\texttt{"a"}+\!\!+)\ 10\ \texttt{"b"}$ e veja se os resultados confirmam as suas respostas acima.

Consider catamorphism $rep\ f = (\!|\, [\underline{id}\ , (f \cdot)] \,|\!)$. Draw a diagram of this catamorphism and answer: What is the type of rep? What does rep do?
Using `cataNat g` from library `Nat.hs` to implement $(\!|\, g \,|\!)$, evaluate in GHCi expressions like $rep\ (2*)\ 0\ 3$, $rep\ (\texttt{"a"}+\!\!+)\ 10\ \texttt{"b"}$ and check if the results confirm your answers above.

---

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

Any function $k = \mathsf{for}\ f\ i$ 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 $(a*) = \mathsf{for}\ (a+)\ 0$ e outros catamorfismos de naturais de que se tenha falado nas aulas da UC.

Encode function $(a*) = \mathsf{for}\ (a+)\ 0$ 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, whishing 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*

   $$func :: Eq\ a \Rightarrow b \rightarrow [(a,b)] \rightarrow (a \rightarrow b)$$
   $$func\ b = (maybe\ b\ id \cdot) \cdot \mathsf{flip}\ lookup$$

   "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),

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

   their nationalities (if known),

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

   and places of residence (if known):

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

   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*

   □