

Assignment 5

Deadline for solutions: 11.07.2019, **12.15 a.m.**

Exercise 1 CCC (6 Points)

Use the definition of exponentials from the lecture and show that

- (a) the correspondence $(B, A) \mapsto A^B$ extends to a bi-functor $\mathcal{C}^{\text{op}} \times \mathcal{C} \rightarrow \mathcal{C}$ sending $f: A' \rightarrow A$ and $g: B \rightarrow B'$ to

$$\text{curry}(B^A \times A' \xrightarrow{\text{id} \times f} B^A \times A \xrightarrow{\text{ev}} B \xrightarrow{g} B'): B^A \rightarrow B'^{A'};$$

- (b) $\text{curry}: \text{Hom}(A \times B, C) \cong \text{Hom}(A, C^B)$ is natural both in B and in C .

Exercise 2 Commutative Monads (7 Points)

Let T be a commutative monad whose tensorial strength is $\tau_{A,B}: A \times TB \rightarrow T(A \times B)$ on a category \mathcal{C} . Let $X \otimes Y = X \times Y$ for $X, Y \in \text{Ob}(\mathcal{C})$ and let

$$f \otimes g = (X \times X' \xrightarrow{\text{id} \times g} X \times TY' \xrightarrow{\tau} T(X \times Y') \xrightarrow{f \times \text{id}} T(TY \times Y') \xrightarrow{\bar{\tau}^*} T(Y \times Y'))$$

for $f: X \rightarrow TY$, $g: X' \rightarrow TY'$ where $\bar{\tau} = T(\text{snd}, \text{fst}) \circ \tau \circ \langle \text{snd}, \text{fst} \rangle: TY \times Y' \rightarrow T(Y \times Y')$.

- (a) Prove that \otimes is a bifunctor $\mathcal{C}_T \times \mathcal{C}_T \rightarrow \mathcal{C}_T$;
 (b) Give an example of a commutative monad for which \otimes is not a Cartesian product (with a proof!).

Exercise 3 20 Questions in Haskell (7 Points)

Come up with an interactive binary search procedure in the spirit of the Twenty Questions game, as explained in Wikipedia: https://en.wikipedia.org/wiki/Twenty_Questions.

To that end you need to

1. Introduce a suitable data type for trees whose internal nodes are labeled with yes/no questions and whose terminal nodes are labeled with answers.
2. Introduce a monad obtained by applying the state monad transformer to the IO monad where the state contains the current subtree and the number of questions already asked:

```
type AskM = StateT (Int, QTree) IO
```

3. Implement a function for searching an element in a tree by asking the user the questions at the nodes one by one and updating the current tree according to the answer as a computation w.r.t. the monad from the previous clause.

4. Implement a test run of your program to guess an object from the domain of your choice. That could be a Star Wars character, country, food from the mensa menu, etc.

Example run:

```
*TwentyQuestions> twenty langTree
(0) Functional? y
Please answer "yes" or "no":
yes
(1) Statically typed? yes
(2) Dependantly typed? no
The answer is: Haskell
*TwentyQuestions>
```