Examen partiel 2012-2013

Modèles des langages de programmation
Master Parisien de Recherche en Informatique

Le mercredi 5 décembre 2012 de 8h45 à 11h45

L'examen peut être rédigé en français ou en anglais selon la préférence.

**Exercise.** We would like to construct the set-theoretic model of the simply-typed $\lambda$-calculus by picking an appropriate exponential modality $K$ inside the category of coherence spaces. To that purpose, we start from a construction seen during the third lecture of the course, which associates to every coherence space $A$ a domain $D(A)$ whose elements are the cliques of $A$ ordered by inclusion.

§1. Show that

$$D \quad : \quad Coh \quad \longrightarrow \quad Set$$

defines a functor from the category of coherence spaces and cliques to the category of sets and functions.

§2. Show that this functor is right adjoint to the functor

$$F \quad : \quad Set \quad \longrightarrow \quad Coh$$

which transports every set $X$ to the discrete coherence space $F(X)$ with web $X$ and pairwise incoherent elements.

§3. Show that $F$ transports the cartesian product $X \times Y$ of two sets to the tensor product $FX \otimes FY$ of their image, and the terminal set to the tensorial unit $\mathbf{1}$.

Note : in that case, recall from the course that the adjunction $F \dashv D$ is symmetric monoidal (in the lax sense) and thus defines a model of linear logic, with exponential modality defined as

$$K(A) \quad = \quad F \circ D(A)$$

for every coherence space $A$

§4. Describe the coherence space $K(FX)$ associated to a set $X$.

§5. Construct a clique

$$\underline{\varepsilon}_A \quad : \quad K(A) \quad \longrightarrow \quad A$$

for every coherence space $A$ and check that it is natural in $A$.

§6. Construct two cliques

$$\underline{d}_A \quad : \quad K(A) \quad \longrightarrow \quad K(A) \otimes K(A)$$

$$\underline{e}_A \quad : \quad K(A) \quad \longrightarrow \quad \mathbf{1}$$

which define a commutative comonoid structure on $K(A)$.

§7. Construct a clique

$$\varphi_A \quad : \quad K(A) \quad \longrightarrow \quad !A$$

and show that the two diagrams

$$\begin{array}{ccc} K(A) & \xrightarrow{\varphi_A} & !A \\ {\scriptstyle \underline{d}_A}\downarrow & & \downarrow{\scriptstyle d_A} \\ K(A) \otimes K(A) & \longrightarrow & !A \otimes !A \end{array} \qquad\qquad \begin{array}{ccc} K(A) & \xrightarrow{\varphi_A} & !A \\ & {\scriptstyle \underline{e}_A}\searrow \quad \swarrow{\scriptstyle e_A} & \\ & \mathbf{1} & \end{array}$$

commute for every coherence space $A$.

§8. Given a coherence space $A$ and a set $X$, establish the bijection between the two sets :

$$D(FX \multimap A) \quad \cong \quad X \Rightarrow DA.$$

§9. Deduce the bijection

$$D(KA \multimap B) \quad \cong \quad DA \Rightarrow DB.$$

for every two coherence spaces $A$ and $B$. Conclude that the cliques of

$$KA \multimap B$$

are in bijection with the functions from the set of cliques of $A$ to the set of cliques of $B$.

**Problem.** We suppose given a finite set $S$. Consider the functor

$$T \quad : \quad A \quad \mapsto \quad S \Rightarrow (S \times A) \quad : \quad Set \quad \longrightarrow \quad Set$$

which transports a set $A$ to the set $S \Rightarrow (S \times A)$.

§1. Show that $T$ defines a functor.

§2. Define a function

$$\eta_A \quad : \quad A \quad \longrightarrow \quad TA$$

for every set $A$ and show that $\eta$ defines a natural transformation.

§3. Define a function

$$\mu_A \quad : \quad TTA \quad \longrightarrow \quad TA$$

for every set $A$ and show that $\mu$ defines a natural transformation.

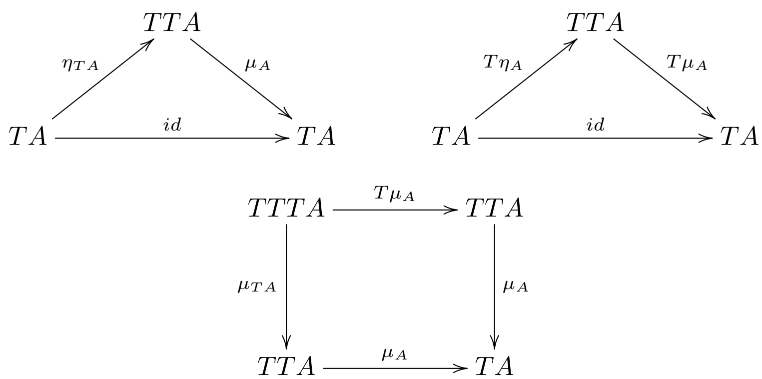
§4. Construct the derivation trees of the following sequents :

$$a : A \quad \vdash \quad \lambda s.\langle a, s\rangle \quad : \quad TA$$

$$\varphi : TTA \quad \vdash \quad \lambda s.(\pi_2(\varphi s))(\pi_1(\varphi s)) \quad : \quad TA$$

and show that the simply typed $\lambda$-terms are interpreted as the functions $\eta_A$ and $\mu_A$ in the category *Set* of sets and functions.

§5. Check using a computation in the $\lambda$-calculus that the diagrams below commute for every set $A$ :

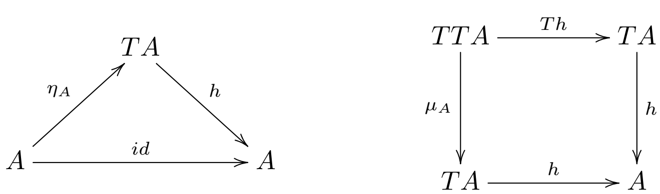



Note : this establishes that $T$ defines a monad with multiplication $\mu$ and unit $\eta$. This monad is called the *state monad* induced by the set $S$ of states.

§6. A $T$-algebra $(A, h)$ is defined as a set $A$ equipped with a function

$$h \quad : \quad TA \quad \longrightarrow \quad A$$

making the diagrams commute :




Show that every set $TA$ equipped with the function

$$\mu_A \quad : \quad TTA \quad \longrightarrow \quad TA$$

defines such a $T$-algebra. Hint : the property follows from the fact that $T$ is a monad, and not from the definition of $T$ as the state monad.

§7. Construct a function

$$h \quad : \quad S \Rightarrow A \quad \longrightarrow \quad A$$

defining a $T$-algebra structure on the set $S \Rightarrow A$.

§8. For every natural number, we write

$$[n] \quad = \quad \{1, \ldots, n\}$$

for the set consisting of $n$ elements. Show that every element

$$f \quad : \quad [1] \quad \longrightarrow \quad T([n])$$

defines a function

$$\langle f \rangle \quad : \quad A^n \quad \longrightarrow \quad A$$

on any $T$-algebra $A$. Hint : once again, one may only use the fact that $T$ is a monad, and observe that a word of length $n$ and with letters in $A$ is the same thing as a function

$$[n] \quad \longrightarrow \quad A.$$

For that reason, we call $n$-*ary operation* of the state monad $T$ any such element $f$ of $T([n])$.

§9. More generally, show that a function

$$g \quad : \quad [m] \quad \longrightarrow \quad T([n])$$

corresponds to a family of $n$-ary operations

$$g_1, \ldots, g_m \quad : \quad [1] \quad \longrightarrow \quad T([n])$$

and thus defines a function

$$\langle g \rangle \quad : \quad A^n \quad \longrightarrow \quad A^m$$

on any $T$-algebra $A$.

§10. Given a $m$-ary operation

$$f \quad : \quad [1] \quad \longrightarrow \quad T([m])$$

and a family of $n$-ary operations

$$g \quad : \quad [m] \quad \longrightarrow \quad T([n])$$

define the $n$-ary operation

$$g \bullet f \quad : \quad 1 \quad \longrightarrow \quad T([n])$$

as the composite

$$1 \xrightarrow{\ f\ } T([m]) \xrightarrow{\ Tg\ } TT([n]) \xrightarrow{\ \mu_{[n]}\ } T([n])$$

Show that the associated function

$$A^n \xrightarrow{\ \langle f \bullet g \rangle\ } A$$

coincides with the composite function

$$A^n \xrightarrow{\ \langle g \rangle\ } A^m \xrightarrow{\ \langle f \rangle\ } A$$

Note : this justifies to introduce the notation

$$f(x_1, \ldots, x_n) \quad : \quad [1] \quad \longrightarrow \quad T([n])$$

for an $n$-ary operation $f$ and to write

$$f(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)) \quad := \quad g \bullet f \quad : \quad [1] \quad \longrightarrow \quad T([n]).$$

We write $x = x_1$ and $y = x_2$ when there is no risk of confusion.

We would like to use this result in order to understand better the state monad. From now on, and in order to make our life a bit simpler, we consider a set $S = [2] = \{\mathbf{false}, \mathbf{true}\}$ of states corresponding to a unique register containing either $0 = \mathbf{false}$ or $1 = \mathbf{true}$. We also consider the binary operation

$$\mathbf{lookup}(x, y) \quad : \quad [1] \quad \longrightarrow \quad S \Rightarrow S \times [2]$$

defined as the currification of the diagonal

$$\Delta \quad : \quad S \quad \longrightarrow \quad S \times S$$

and the two unary operations

$$\mathbf{update}[\mathbf{false}](x) \quad : \quad [1] \quad \longrightarrow \quad S \Rightarrow S \times [1]$$

$$\mathbf{update}[\mathbf{true}](x) \quad : \quad [1] \quad \longrightarrow \quad S \Rightarrow S \times [1]$$

defined as the currification of the constant functions

$$\mathbf{false} \quad : \quad S \quad \longrightarrow \quad S$$

$$\mathbf{true} \quad : \quad S \quad \longrightarrow \quad S$$

respectively.

§11. Show that the unary operation

$$\mathbf{update}[\mathbf{false}](\mathbf{update}[\mathbf{true}](x)) \quad : \quad [1] \quad \longrightarrow \quad T([1])$$

coincides with the unary operation

$$\mathbf{update}[\mathbf{true}](x) \quad : \quad [1] \quad \longrightarrow \quad T([1])$$

Justify why this equation makes sense.

§12. Show that the binary operation

$$\mathbf{update}[\mathbf{false}](\mathbf{lookup}(x, y)) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

coincides with the binary operation

$$\mathbf{update}[\mathbf{false}](x) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

defined as the composite

$$[1] \xrightarrow{\mathbf{update}[\mathbf{false}]} T([1]) \xrightarrow{T(\mathbf{inl})} T([2])$$

where

$$\mathbf{inl} \quad : \quad [1] \quad \longrightarrow \quad [2]$$

is the function with transports 0 to 0 = **false**. Justify why this equation makes sense.

§13. Similarly, show that the binary operation

$$\mathbf{update}[\mathbf{true}](\mathbf{lookup}(x, y)) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

coincides with the binary operation

$$\mathbf{update}[\mathbf{true}](y) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

defined as the composite

$$[1] \xrightarrow{\mathbf{update}[\mathbf{true}]} T([1]) \xrightarrow{T(\mathbf{inr})} T([2])$$

where

$$\mathbf{inr} \quad : \quad [1] \quad \longrightarrow \quad [2]$$

is the function with transports 0 to 1 = **true**. Justify why this equation makes sense.

§14. Show that the binary operation

$$\mathbf{lookup}(x, y) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

coincides with

$$\mathbf{lookup}(\mathbf{update}[\mathbf{false}](x), \mathbf{update}[\mathbf{true}](y)) \quad : \quad [1] \quad \longrightarrow \quad T([2])$$

Justify why this equation makes sense.

§15. Define the three functions

$$\begin{array}{rcccl} \langle\mathbf{lookup}\rangle & : & (S \Rightarrow A)^2 & \longrightarrow & S \Rightarrow A \\ \langle\mathbf{update}[\mathbf{false}]\rangle & : & S \Rightarrow A & \longrightarrow & S \Rightarrow A \\ \langle\mathbf{update}[\mathbf{true}]\rangle & : & S \Rightarrow A & \longrightarrow & S \Rightarrow A \end{array}$$

associated to the three operations defined above, in the specific case of the $T$-algebra $S \Rightarrow A$ defined in §7.

§16. Check the equations established in §11, §12, §13, §14 in the particular case of such a $T$-algebra $S \Rightarrow A$.

Note : it is possible to establish that a $T$-algebra is the same thing as a set $A$ equipped with a binary operation **lookup** and two unary operations **update**[**false**] and **update**[**true**] satisfying (a slightly completed version of) the series of equations encountered during the problem.