Homework 2: Kindom of Kittens
15-819 Homotopy Type Theory

Out: 5/Oct/13
Due: 19/Oct/13

Hey this is the second homework assignment! This time we will play a little bit with category theory. Although not the absolutely essential part of this course, it provides a nice way to think about various properties diagrammatically.

1   Categorical Considerations

Similar to Heyting algebras, *bicartesian closed categories* also model the IPL. Intuitively, a bicartesian closed category is a category equipped with (binary) products, (binary) coproducts, an inital object, a terminal object, and exponentials as described in class.

> ERRATUM (2013/10/10 4PM): I forgot inital and terminal objects.

Actually, a Heyting algebra is a bicartesian closed category where there is at most one morphism for any two objects such that there is a morphism from A to B iff $A \leq B$. An immediate observation is that a Heyting Algebra can only keep track of *provability* that is represented by this sole morphism. Here we are considering a more general case where one can have multiple morphisms between objects which correspond to different *proofs* of the same proposition.

The theorem about the relationship between the logic and the categories involves two directions. One direction is to build a *syntactic category*, a category directly out of the syntactical rules, to show that IPL is complete. This plays a similar role as the Lindenbaum algebra for Heyting

algebras but we will not cover it in this assignment. For the other direction which is about the soundness of the IPL, the core part is a systematic translation of a judgment

$$\Gamma \vdash M : P$$

into a morphism

$$[\![\Gamma \vdash M : P]\!] : \Gamma^+ \to P^*$$

where, similar to what we did for Heyting algebras, $(-)^*$ is the (lifted) translation function from propositions to objects, and $(-)^+$ is the comprehension of this function for Γ.

Remark. *To match the diagram of exponentials in class, in Homework 2 the function $(-)^+$ does not swap the order of propositions in Γ, which is to say that $(\Gamma, x{:}A)^+ = \Gamma^+ \times A^*$.*

1.1 Structural Safety

As mentioned above, the argument critically depends on a systematic translation of judgments $[\![-]\!]$. However, we will not go through the construction here; instead, you only have to show that the category justifies various structural rules.

Task 1. *Write down a suitable morphism in terms of the primitive constructs and the morphisms immediately available in each subtask. The primitive constructs include id, $f \circ g$, $\langle f, g \rangle$, fst, snd, inl, inr, $\{f, g\}$, $\lambda(f)$ and ap.*

- *Reflexivity.*

  Write down a morphism from $\Gamma^+ \times P^*$ to P^*.

- *Contraction.*

  Write down a morphism from $\Gamma^+ \times P^*$ to Q^* in terms of a morphism $f : (\Gamma^+ \times P^*) \times P^* \to Q^*$.

- *Weakening.*

  Write down a morphism from $\Gamma^+ \times P^*$ to Q^* in terms of a morphism $f : \Gamma^+ \to Q^*$.

- *Exchange.*

  Write down a morphism from $(\Gamma^+ \times Q^*) \times P^*$ to R^* in terms of a morphism $f : (\Gamma^+ \times P^*) \times Q^* \to R^*$.

- *Substitution*.

  Write down a morphism from Γ^+ to Q^* in terms of two morphisms $f : \Gamma^+ \to P^*$ and $g : \Gamma^+ \times P^* \to Q^*$.

1.2 Utility of Universality

Beside the structural rules, another interesting fact is that β and η rules correspond to universal properties. This somewhat makes the seemingly arbitrary syntactical characterization canonical. In this subsection we will explore universal properties a little bit.

Task 2. *Show these rules via universal properties:*

- $\langle f, g \rangle \circ h = \langle f \circ h, g \circ h \rangle$

- $h \circ \{f, g\} = \{h \circ f, h \circ g\}$

Task 3. *Show that β and η rules of implications (function types) are justified by the universal property of exponentials. You may assume the translation of judgments $[\![-]\!]$ satisfies the following properties:*

- $[\![\Gamma \vdash \lambda x.M_1 : P \to Q]\!] = \lambda([\![\Gamma, x{:}P \vdash M_1 : Q]\!])$

- $[\![\Gamma \vdash M_1\,M_2 : Q]\!] = \mathtt{ap} \circ \langle [\![\Gamma \vdash M_1 : P \to Q]\!], [\![\Gamma \vdash M_2 : P]\!]\rangle$

- $[\![\Gamma, x{:}P \vdash x : P]\!]$ *gives the same morphism as you did in Task 1 for reflexivity.*

- $[\![\Gamma, x{:}P \vdash M : Q]\!]$ *also gives the same morphism in terms of $[\![\Gamma \vdash M : Q]\!]$ as you did in Task 1 for weakening.*

- *Fortunately,[1] $[\![\Gamma \vdash [M_2/x]M_1 : Q]\!]$ gives the same morphism in terms of $[\![\Gamma, x{:}P \vdash M_1 : Q]\!]$ and $[\![\Gamma \vdash M_2 : P]\!]$ as you did in Task 1 for substitution.*

You may freely assume previous tasks, as usual.

BUGFIX (2013/10/17 6PM): There was a bug in the assumption about reflexivity. $x{:}P$ was missing in the context. The fix is marked

---

[1]These are all provable theorems for a suitable, systematic translation.

2   Commuting Conversion

Now we are going back to the syntactical world temporarily.

Bonus Task 1. *There is a short proof of $[M_1/x]N \equiv [M_2/x]N$ when $M_1 \equiv M_2$ directly from suitable $\beta\eta$ rules and congruence. Try to find it!*

Task 4. *Check that*

$$M \equiv \texttt{case}(z; x.[\texttt{inl}(x)/z]M; y.[\texttt{inr}(y)/z]M)$$

is derivable from

- $z \equiv \texttt{case}(z; x.\texttt{inl}(x); y.\texttt{inr}(y))$ *and*

- $[\texttt{case}(z; x.\texttt{inl}(x); y.\texttt{inr}(y))/z]M \equiv \texttt{case}(z; x.[\texttt{inl}(x)/z]M; y.[\texttt{inr}(y)/z]M)$.
  (This is called commuting conversion.)

You may assume $[z/z]M \equiv M$ and previous tasks.

3   Narratives of Natural Numbers

3.1   Summary in Sum

Recall the alternative definition of natural number objects through this diagram:

$$
\begin{array}{ccc}
1 + \mathbb{N} & \xdashrightarrow{\ \texttt{id}+\texttt{rec}_{f,g}\ } & 1 + A \\
{\scriptstyle \{\texttt{zero},\texttt{suc}\}}\Big\downarrow & & \Big\downarrow{\scriptstyle \{f,g\}} \\
\mathbb{N} & \xdashrightarrow{\ !\,\texttt{rec}_{f,g}\ } & A
\end{array}
$$

Task 5. *Check that this diagram indeed defines a natural number object (in the sense described in class).*

3.2 Recursors Redefined

CLARIFICATION (2013/10/15 9AM): For the purpose of doing this homework, please only refer to lecture notes of first three weeks and ignore anything beyound those. In particular, the `rec` here is not the `rec` in dependent type theories.

Now go back to the syntactical world. The $\mathtt{rec}(P, x.Q)$ presented in class is sometimes called *iterator*, in contrast to an alternative version $\widetilde{\mathtt{rec}}(P, x.y.Q)$ where Q not only has access to the result of recursion as y, but also the number as x. This alternative version not only facilitates practical programming, but is also more suitable for the dependently-typed setting (which will be clear in the future lectures). For example, a straightforward definition of the factorial function computes the multiplication of the result from iteration and the number.

Fortunately, the seemingly more powerful $\widetilde{\mathtt{rec}}$ is definable in terms of the `rec` mentioned in class in the presence of products. Here is the formal specification:

- $\widetilde{\mathtt{rec}}(P; x.y.Q)(\mathtt{zero}) \equiv P$

- For any closed term $\emptyset \vdash M : \mathtt{nat}$,
  $\widetilde{\mathtt{rec}}(P; x.y.Q)(\mathtt{suc}(M)) \equiv [M, \widetilde{\mathtt{rec}}(P; x.y.Q)(M)/x, y]Q$

Note that Q has access to the *predecessor* of the argument, not the argument itself.

CHANGE (2013/10/15 9AM): The specification has been rewritten so that it is entirely within Gödel's T with products. Its meaning is still the same and this should not affect your answer to the following task.

Task 6. *Define $\widetilde{\mathtt{rec}}$ in terms of* `rec`. *You do not have to show the correctness.*