Problem 1  Prove that \((U : \mathcal{C} \to \mathcal{D}, F, \eta)\) is a right adjoint (resp. \((F : \mathcal{D} \to \mathcal{C}, \psi, \varepsilon)\) a right adjoint) iff there is a natural equivalence of functors
\[
\mathcal{C}(F(-), -) \cong \mathcal{D}(-, \psi(-)) : \mathcal{D} \times \mathcal{C} \to \text{Set}
\]

Problem 2  Prove that \(\mathcal{C}\) has coproducts (resp. products) iff there is a natural equivalence of functors
\[
\mathcal{C}(X \oplus Y, Z) \cong \mathcal{C}(X, Z) \times \mathcal{C}(Y, Z)
\]
(resp. \(\mathcal{C}(Z, X \times Y) \cong \mathcal{C}(Z, X) \times \mathcal{C}(Z, Y)\))

where \(X, Y, Z\) denote the 1st, 2nd, and 3rd projections.

For example:
\[
\begin{array}{ccc}
\mathcal{C} \times \mathcal{C} \times \mathcal{C} & \xrightarrow{\mathcal{C}(X \oplus Y, Z)} & \mathcal{C} \times \mathcal{C} \\
(\mathcal{C} \times \mathcal{C}) \circ \mathcal{C} & \xrightarrow{\mathcal{C}(-, -)} & \text{Set} \\
\mathcal{C}(X \oplus Y, Z) & \end{array}
\]
Problem 3. Prove that if \((F: D \to C, U, \varepsilon)\) is a left adjoint (resp. \((U: C \to D, F, \eta)\) is a right adjoint) then it preserves coproducts and initial object (resp. products and final object).

Problem 4. In any category \(C\), given morphisms

\[
A \xrightarrow{f} B, \text{ an object } I(f, g) \in C \text{ and morphism } I(f, g) \xrightarrow{\varepsilon} A
\]

(resp. object \(C(f, g) \in C\) and morphism \(B \xrightarrow{c} C(f, g)\)) is called an equalizer (resp. coequalizer) of \(f, g\) iff

- \((1)\) \(e; f = e; g\) (resp. \(f; e = f; c\)), and
- \((2)\) for each \(C \in C\) and \(h: C \to A\) (resp. \(C \in C\) and \(h: B \to C\)) such that \(h; f = h; g\) (resp. \(f; h = g; h\)) there is a unique \(\overline{h}: C \to I(f, g)\) (resp. \(\overline{h}: C(f, g) \to C\)) such that \(h = \overline{h}; e\) (resp. \(h = c; \overline{h}\)).

(a) Prove that the categories

- \(\text{Set}\)
- \(\text{Alg}\ \Sigma\)
- \(\text{Pfn}\)
- \(\text{Poset}\) have equalizers and coequalizers.
Problems 5  (Solving Systems of Equations in \( P_\omega (\omega) \))

Recall the category \( P_\omega (\omega) \) of \( \omega \)-cops, where we assume each \( A = (A, \leq_A) \in P_\omega (\omega) \) has a bottom element \( \bot_A \in A \) (\( P_\omega (\omega) \) can be defined without this requirement of \( \bot_A \in A \)).

(a) Let now \( A_1, \ldots, A_n \in P_\omega (\omega) \), and let
\[
\{ f_i : A_1 \times \cdots \times A_n \rightarrow A_i \mid 1 \leq i \leq n \}
\]
be a collection of \( \omega \)-continuous functions. Prove that we can solve the system of equations

\[
\begin{align*}
X_1 : A_1 & = f_1 (X_1 : A_1, \ldots, X_n : A_n) \\
\vdots \\
X_n : A_n & = f_n (X_1 : A_1, \ldots, X_n : A_n)
\end{align*}
\]

and that indeed there is a least \( \omega \)-possible such solution.

(b) Explain how (a) can be applied to give a fixpoint semantics to mutually recursive functions such as:

\[
\begin{align*}
\text{odd}(n) & = \text{if } n = 0 \text{ then false else even}(n) \text{ fi} \\
\text{even}(n) & = \text{if } n = 0 \text{ then true else odd}(n) \text{ fi}
\end{align*}
\]
Problem 6  (A Compiler from the λ-Calculus to $\lambda$CINNI)

Generalize the ideas in the compiler 1.2DB

in the module LAMBDA-2-DB-COMPILER to

define a similar compiler 1.2CINNI in your

module LAMBDA-2-LAMBDA-CINNI-COMPILER.