Problem 1. For a detailed proof see Mac lane's book, IV.4, Thm 1.

Problem 2. Let us do the case for coproducts. The case for products then follows directly by duality. By the answer to Problem 1 and $\Sigma_g : \mathcal{C} \to \mathcal{C}^2$ right adjoint to $\Phi : \mathcal{C}^2 \to \mathcal{C}$ we have a natural iso:

$$\mathcal{C}(X \oplus Y, Z) \cong \mathcal{C}^2((X, Y), (Z, Z))$$

We now obtain our desired result by composing with the natural isomorphism

$$\mathcal{C}^2((X, Y), (X', Y')) = \mathcal{C}(X, X') \times \mathcal{C}(Y, Y')$$

directly based on the fact that $\mathcal{C}^2 = \mathcal{C} \times \mathcal{C}$ as a product in $\text{Cat}$, which gives us a natural equivalence:

$$\mathcal{C}(X \oplus Y, Z) \cong \mathcal{C}^2((X, Y), (Z, Z)) \cong \mathcal{C}(X, Z) \times \mathcal{C}(Y, Z)$$

as desired. \(\Box\)

Problem 3. Again, it is enough to do the case for left adjoint, since the right adjoint case then follows by duality.

a. Let $0$ be the initial object in $\mathcal{D}$. Since we have $\mathcal{D}(0, F(A)) \cong \mathcal{C}(F(0), A)$ for any $A \in \mathcal{C}$, we then have

$$1 = |\mathcal{D}(0, F(A))| = |\mathcal{C}(F(0), A)|$$

Therefore, $F(0)$ is initial in $\mathcal{C}$. 
6. We have, by the answers to Problems 1-2:

\[ \mathfrak{C}(F(X \oplus Y), Z) \cong \mathfrak{D}(X \oplus Y, U(Z)) \cong \mathfrak{D}(X, U(Z)) \times \mathfrak{D}(Y, U(Z)) \cong \mathfrak{C}(F(X), Z) \times \mathfrak{C}(F(Y), Z). \]

Therefore, \( F(X \oplus Y) = F(X) \oplus F(Y) \), as desired.

**Problem 4**

- Let \( A \xrightarrow{f} B \), \( I(f, g) = \{ a \in A \mid f(a) = g(a) \} \).

The inclusion \( I(f, g) \rightarrow A \) is then obviously an equalizer.

Likewise, let \( R_{f, g} \subseteq B^2 \) be the set \( R_{f, g} = \{ (f(a), g(a)) \mid a \in A \} \).

Then define \( C(f, g) = B / \overline{R_{f, g}} \), where \( \overline{R_{f, g}} \) is the equivalence relation \( \overline{R_{f, g}} = (R_{f, g} \cup R_{f, g}^{-1})^* \). That \( C(f, g) \) is a coequalizer follows then trivially from Lemma 7, p. 61, § 7.6 of \textsc{Stacs}.

- \( \textbf{Alg}_\Sigma \) Let \( A \xrightarrow{f} B \) be \( \Sigma \)-homomorphism and \( I(f, g) \) its equalizer in \( \textbf{Set} \). Note that \( I(f, g) \) is a \( \Sigma \)-subalgebra of \( A \), this is easy to see because: (i) for \( c \) a constant, \( f(c_A) = c_B = g(c_A) \), and for \( f \in \Sigma_n \), if \( f(a_1) = g(a_1), \ldots, f(a_n) = g(a_n) \),
then $h(f_A(a_1, ..., a_n)) = f_B(h(a_1), ..., h(a_n)) = f_B(g(a_1), ..., g(a_n)) = g(f_A(a_1, ..., a_n)).$

Suppose now that $q : C \rightarrow A$ in a $\Sigma$-hom. such that $q; h = q; g$. Since $I(f, g)$ is an equivalence set, there is a unique function $\bar{q} : C \rightarrow I(f, g)$ such that $q = \bar{q}; f$. But since $q$ is a $\Sigma$-homomorphism and $I(f, g)$ is a $\Sigma$-subalgebra of $A$, $\bar{q}$ is obviously a $\Sigma$-homom.

Recall that a $\Sigma$-congruence on a $\Sigma$-algebra $B$ is an equivalence relation $\equiv \subseteq B^2$ such that for each $f \in \Sigma m$, $n \in D$, if $b_1 \equiv b_1', ... , b_n \equiv b_n'$, then $f_B(b_1, ..., b_n) \equiv f_B(b_1', ..., b_n')$. Since it is easy to check that for an arbitrary family $\{ \equiv_i \}_{i \in I}$ of $\Sigma$-congruences, its intersection $\bigcap_i \equiv_i$ is also a $\Sigma$-congruence, this shows that the set $\Sigma$-Cong $(B) \subseteq P(B^2)$ of $\Sigma$-congruences for $B$ is a complete inf semi-lattice, and, therefore (see Exercise 5.7, 8, 7.5 in STACS), that it is a complete lattice.

Let $\equiv_{\Sigma} = \bigwedge \{ R : \Sigma$-Cong $(B) | R \supseteq R_{f, g} \}$

be the smallest $\Sigma$-congruence generated by $R_{f, g}$ (it can
of course be generated in a more constructive way from the
bottom up, as opposed to the above, easier but less constructive,
top-down fashion by a "congruence closure" algorithm
which you may like to write down, taking the word
"algorithm" with plenty of salt, since B need not be
a computable algebra at all).
Then define \( G(f,g) = B/\equiv_{fg} \) the quotient algebra.

Now consider \( \varphi : B \to C \) such that \( h; \varphi = g; \varphi \)
and note that \( \equiv_\varphi = \{(b,b') \in B^2 \mid \varphi(b) = \varphi(b')\} \) is
a \( \Sigma \)-congruence and that, obviously, \( R_{f,g} \subseteq \equiv_\varphi \).
Therefore, \( \equiv_{f,g} \subseteq \equiv_\varphi \). Therefore, again by Lemma 7,
18.64, 8.7.6 of STACS, there is a unique function
\( \overline{\varphi} : B/\equiv_{f,g} \to C \) such that \( \varphi_f; \overline{\varphi} = \varphi \),
where \( \varphi_f : B \to B/\equiv_{f,g} : b \mapsto [b] \) is the
quotient \( \Sigma \)-homomorphism each \( b \in B \) to its
\( \equiv_{f,g} \)-equivalence class. All we need to show is
that \( \overline{\varphi} \) is a \( \Sigma \)-homomorphism. But this is
trivial, since for each \( c \in \Sigma \) we have
\( G(B/\equiv_{f,g}) = [c_B] \) and therefore, \( \overline{\varphi}([c_B]) = \varphi(c_B) = c \).
Likewise, given \( f \in \Sigma_m \) and \( \{b_1, \ldots, b_n\} \in B/\equiv_f \),
then \( f \mid_{B/\equiv_f} (\{b_1, \ldots, b_n\}) = [f_B(b_1, \ldots, b_n)] \)
and then \( \bar{f}(\bar{f_B(b_1, \ldots, b_n)}) = f(\bar{f_B(b_1, \ldots, b_n)}) = \)
\( = (by \text{ \( \Sigma \)-hom.}) \ f_C(\bar{f}(b_1), \ldots, \bar{f}(b_n)) = \)
\( = f_C(\bar{f}(\{b_1\}), \ldots, \bar{f}(\{b_n\})) \), as desired.

**Proof.** Recall that we have an equivalence of categories
\( \text{Pfn} \cong \text{Set}_1 \), where \( \text{Set}_1 = \text{Alg } \Sigma_1 \) with \( \Sigma_1, 0 = 1 \}
and \( \Sigma_1, n = \emptyset, n > 0 \). So you do not have to prove
anything for \( \text{Pfn} \); just move the constructions over
from \( \text{Set}_1 \) to \( \text{Pfn} \) using the equivalence of categories.
Nevertheless, you should check what exactly do you get
by going back and forth from \( \text{Pfn} \) to \( \text{Set}_1 \) (to
confirm both \( I(f,g) \) and \( C(f,g) \)) and back to \( \text{Pfn} \).

**Problem 5.** Then \( A_1 \times \ldots \times A_n \) has a bottom element
\( (\perp A_1, \ldots, \perp A_n) \), and therefore the function
\[
\begin{array}{c}
A_1 \times \ldots \times A_n \\
\xrightarrow{(f_1, \ldots, f_n)} \\
A_1 \times \ldots \times A_n
\end{array}
\]
has a minimal fixpoint, which is a minimal solution for the
system of equations. The application to fixpoint semantics
of mutually recursive definitions is then immediate.