8.2

a The codomain of *length* is $\mathbb{N}$ and by definition the map must be surjective, so if $\leq$ on $\mathbb{N}$ is a cpo, so then this relation is a cpo. However, $\mathbb{N}$ is not a cpo since it has no maximal element and so the relation is not a cpo.

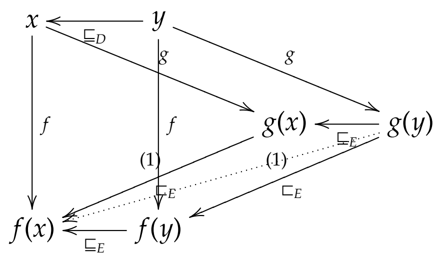
b The codomain of *Letters* is the powerset lattice 2^X, which is finite, and so the relation is a partial order.

c For the first question, the answer is clearly no. Pick X to be a unary alphabet, then the relation reduces to that in part a.

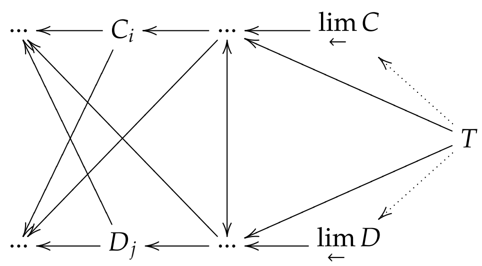
For the second question, let $\bar{X} = X^* \cup X^\omega$ and observe that all prefixes are of at most countable length (since they are elements of $\bar{X}$) and thus any chain has a some upper bound in X. By Zorn's Lemma, this implies that the chain must also have a maximal element and so the order is complete.

8.3 $f \subseteq g \implies f \leq g$: Since $x \sqsubseteq_D x$, the condition is vacuously true.

$f \leq g \implies f \subseteq g$: Pick a pair $x \sqsubseteq_D y$. Then the following diagram commutes:



8.12 Consider the following commutative diagram for a cone T where each arrow denotes a $\sqsubseteq$ relation:



So, any cone over C is also a cone over D and thus the limits are isomorphic.

9.3

a Consider the program:

```
void f() { f(); }
```

Which has the equation:

$$V_f = \{f\} \cup V_f$$

And so any set S such that $f \in S$ is a solution for V_f.

b For a given program, P, with function names, $S_P = \{f_0, f_1, ...f_n\}$, define a *dependence relation over P* to be a transitive relation, R_P on S. Call f_i dependent on f_j subject to R_P if $R_P(f_i, f_j) = 1$, or $R_P(f_i, f_j) = 0$ otherwise. Now define a category, D_P whose objects are dependence relations over P and whose morphisms are relation homomorphisms, ie:

$$\mathrm{Obj}(D_P) = \{R_P | R_P \text{ is a dependence relation over } P\}$$

$$\mathrm{Hom}_{D_P}(X, Y) = \{X \xrightarrow{f} Y | \forall f_i, f_j \in S : X(f_i, f_j) \implies Y(f_i, f_j)\}$$

This category has both initial and terminal objects, $\bot, \top$:

$$\forall f_i, f_j : \bot(f_i, f_j) = 0$$

$$\forall f_i, f_j : \top(f_i, f_j) = 1$$

And so (co)limits exist in D_P.

Next, define a natural transformation $\eta : D_P \to (D_P \to D_P)$ such that for relations $X, Y \in \mathrm{Obj}(D_P)$ and functions $f_i, f_j \in S$:

$$(\eta_X(Y))(f_i, f_j) = X(f_i, f_j) \vee \bigvee_{f_k \in S} X(f_i, f_k) \wedge Y(f_k, f_j)$$

Observe that the definition for a program P gives a particular relation $F_P \in \mathrm{Obj}(D_P)$ such that:

$$F_P\left(f_i, f_j\right) = \begin{cases} 1 & \text{if declaration of } f_i \text{ references } f_j \\ 0 & \text{otherwise} \end{cases}$$

And that since the Hom-sets in D_P are singletons, it is also a partial order.

Now there is an isomorphism between the sets V_{f_i} and the objects of D_P such that for each collection such set there is an unique element, $U \in \mathrm{Obj}(D_P)$, if and only if $f_j \in V_i$ then $U(f_i, f_j) = 1$. And moreover, the solution to the above system of equations is equivalent to the following:

$$V = \eta_{F_P}(V)$$

And since D_P has limits and η is a natural transformation, there is a unique universal object $\bar{X}$ such that:

$$\bar{X} = \bigsqcup_{k=0}^{\infty} \eta_{F_P}^k(\bot)$$

And moreover:

$$\bar{X} = \eta_{F_P}(\bar{X})$$

And so $\bar{X}$ is isomorphic to the least fixed point of ηF_P.

c For the sake of convenience, we shall denote the objects of D_P using a table, where the i, j^{th} entry for the description of the object X corresponds to the value X_{f_i, f_j}. Since we only have one f in problem A, this matrix can be written as just a single element and could be further reduced to just a single element. Thus:

$$\bot = 0$$

$$F_p = 1$$

$$\top = 1$$

And so we iterate to find the fixed poind and get:

$$F^0 = 0$$

$$F^1 = F_p \vee F^0 = 1$$

$$F^2 = F_p \vee F^1 = 1$$

d i. A function is recursive if its set of called functions contains itself. Thus, if the least upper bound of $F_P = X$, then f_i is recursive if and only if $X(f_i, f_i) = 1$.

ii. There is only one function, f and it is recursive.