[Ex. 1] Suppose we add to IMP the command `repeat c until b`, whose denotational semantics is defined recursively as:

\[ C[\text{repeat } c \text{ until } b] \sigma = (\lambda \sigma'. B[b] \sigma' \rightarrow \sigma', C[\text{repeat } c \text{ until } b] \sigma')^*(C[c] \sigma) \]

1. Define the operational semantics of the new construct.
2. Extend the proof of determinacy of the operational semantics taking into account the new construct.
3. Define the function $\Gamma_{c,b}$ such that $C[\text{repeat } c \text{ until } b] = \text{fix } \Gamma_{c,b}$.
4. Compute the denotational semantics of `repeat x := x + 1 until true`.

[Ex. 2] Consider the CPO $D \defeq (\wp(\mathbb{N}), \subseteq)$ and the function $f : \wp(\mathbb{N}) \rightarrow \wp(\mathbb{N})$ such that $f(X) \defeq \{ y \mid \exists x \in X. y \leq x \}$, where $\leq$ is the usual total order on $\mathbb{N}$.

1. Is $f$ monotone?
2. Is $f$ continuous?
3. What is the least fixpoint of $f$? Does $f$ have other fixpoints?

[Ex. 3] Write a Haskell function that takes a list `xs` and returns the list of all pairs $(x, n)$ such that $x$ occurs $n$ times in `xs`, preserving the order of appearance. For example, given the input "hello" the function must return the list

\[
[(\text{’h’}, 1), (\text{’e’}, 1), (\text{’l’}, 2), (\text{’o’}, 1)]
\]

[Ex. 4] Consider the HOFL terms

\[
t_0 \defeq \text{rec } f. \lambda x. \text{if } x \text{ then } (x, f x) \text{ else } (f x, x)
\]

\[
t_1 \defeq \text{rec } f. \lambda x. \text{if } x \text{ then } (x, \text{snd}(f x)) \text{ else } (x, x)
\]

Which term is well-typed? What is its principal type?