[Ex. 1] (1st mid-term / regular exam)
Suppose one wants to insert some measure of efficiency in the operational semantics of IMP.

1. Redefine the operational semantics of IMP commands in such a way that the transition predicate takes the form

\[ \langle c, \sigma \rangle \xrightarrow{n} \sigma' \]

with the meaning that “the command \( c \), when executed in the state \( \sigma \) converges to the state \( \sigma' \) by evaluating exactly \( n \) boolean guards.”

2. Prove by rule induction that for all \( c, \sigma, \sigma' \):

\[ \langle c, \sigma \rangle \rightarrow \sigma' \Rightarrow \exists n \in \mathbb{N}. \langle c, \sigma \rangle \xrightarrow{n} \sigma'. \]

[Ex. 2] (1st mid-term / regular exam)
Consider the CPO \( (\wp(\mathbb{N}), \subseteq) \) and the function \( f : \wp(\mathbb{N}) \rightarrow \wp(\mathbb{N}) \) defined by:

\[ f(X) \overset{\text{def}}{=} \{ y \in \mathbb{N} | \exists a, b \in X. a \leq y \leq b \} \]

1. Prove that \( f \) is monotone.

2. Prove that \( f \) is continuous.

[Ex. 3] (1st mid-term)
Let us call a repetition any list where the same value occurs in all positions of the list. Write a Haskell function \texttt{decompose} that takes a list \( \texttt{xs} \) and returns the list of repetitions in \( \texttt{xs} \). For example, \texttt{decompose [1,1,1,2,2,2,1,1,3]} must return the list \([ [1,1,1], [2,2,2], [1,1], [3] ]\).

[Ex. 4] (1st mid-term)
Consider the HOFL terms

\[ t \overset{\text{def}}{=} \text{rec } f. \lambda x. \text{if } x \text{ then } (x - 1, f (x - 1)) \text{ else } (x + 1, f (x + 1)) \]
\[ s \overset{\text{def}}{=} \text{rec } g. \lambda y. \text{if } y \text{ then } g (y - 1) \text{ else } (y + 1, \text{fst}(g (y + 1))) \]

1. Find the principal type of \( t \), if it exists.

2. Find the principal type of \( s \), if it exists.