Exercise 1 (Reduction Relation with Closures)

For the evaluation of lambda terms that is closer to evaluation of programs in functional programming languages, one usually replaces textual substitution \( t[v/x] \) with a more lazy approach that records the binding \( x \mapsto v \) in an environment. These bindings are used whenever we need the value of a variable \( v \).

In this approach abstractions \( \lambda x.t \) do not evaluate to themselves, but to a pair \( (\lambda x.t)[e] \), where \( e \) is the current environment. We call such pairs function closures.

a) Define a big-step reduction relation for the lambda calculus with function closures and environments.

b) Add explicit error handling for the case where the binding of a free variable \( v \) cannot be found in the environment. Introduce an explicit value \texttt{abort} to indicate such an exception in the relation.

Exercise 2 (Reduction Relation with Pattern Matching)

In this exercise, we consider a \( \lambda \)-calculus extended with a special set of constructor values and pattern matching. Constructor values are constructed according to the following grammar:

\[
c ::= C(c_1, \ldots, c_n) \text{ for } n \geq 0
\]

where \( C \) is one from a distinguished set of constructor symbols.

We illustrate pattern matching by example. The expression

\[
\text{match } C_1 \text{ (false) with } C_2 \text{ () } \rightarrow \text{true} | \ C_1 \text{ (x) } \rightarrow \text{x}
\]

should evaluate to \texttt{false}, while

\[
\text{match } C_2 \text{ (false) with } C_2 \text{ () } \rightarrow \text{true} | \ C_1 \text{ (x) } \rightarrow \text{x}
\]

should evaluate to \texttt{abort}.

a) Define a big-step reduction relation for this language.

b) Prove that the two derivations stated informally above are indeed possible in the relation.
Homework 3 (Normal Forms)
Recall the inductive definition of the set $\text{NF}$ of normal forms:

\[
\begin{align*}
  t & \in \text{NF} \\
  \lambda x.t & \in \text{NF} \\
  n \geq 0 & \Rightarrow t_1 \in \text{NF} \\
  t_2 \in \text{NF} & \ldots t_n \in \text{NF} \\
  x t_1 t_2 \ldots t_n & \in \text{NF}
\end{align*}
\]

Show that this set precisely captures all normal forms, i.e.:

\[ t \in \text{NF} \iff \nexists t'. t \rightarrow_\beta t' \]

Homework 4 (Normal Forms & Big Step)
Show:

\[ t \in \text{NF} \land t \Rightarrow_n u \Rightarrow u = t \]

Homework 5 (Proofs with Small-steps and Big-steps)
Let $\omega := \lambda x.xx$ and $t := (\lambda x. (\lambda y x. z) y) \omega (\omega ((\lambda x y. z x y)))$.

Prove the following:

a) \[ t \Rightarrow_n z\]

b) \[ t \rightarrow_{cbv}^3 t\]

c) \[ t \not\rightarrow_{cbn}^+ t\]