1 Full Program Synthesis

Using the synthesis rules provided, synthesize a function $I$ that satisfies the following judgement:

$$
\vdash \mu \text{nat}. (\cdot + \text{nat} \rightarrow \text{nat}. (\cdot + \text{nat})) \triangleright \cdot \mapsto (ex_0 \mapsto ex'_0 \land ex_1 \mapsto ex'_1 \land ex_2 \mapsto ex'_2 \land ex_3 \mapsto ex'_3) \mapsto I
$$

where:

- $ex_0 := \text{inl}()$
- $ex'_0 := \text{inl}()$
- $ex_1 := \text{inr}((\text{inl}()))$
- $ex'_1 := \text{inr}((\text{inr}((\text{inl}()))))$
- $ex_2 := \text{inr}((\text{inr}((\text{inr}((\text{inl}()))))))$
- $ex'_2 := \text{inr}((\text{inr}((\text{inr}((\text{inr}((\text{inl}())))))))$
- $ex_3 := \text{inr}((\text{inr}((\text{inr}((\text{inr}((\text{inl}()))))))))$
- $ex'_3 := \text{inr}((\text{inr}((\text{inr}((\text{inr}((\text{inr}((\text{inl}())))))))))$

Show the full derivation (you can omit variable replacement derivations and you can omit $E \models \phi$ derivations and you do not need to show how you evaluated distribute or apply).

2 Completeness Counterexample

Come up with a function $I$, a type $\tau$, and a partial function example $ex = ex_1 \mapsto ex'_1 \land \ldots \land ex_n \mapsto ex'_n$ such that for all $i$, $lex_i \mapsto^* ex'_i$ and $\vdash \tau \triangleright \cdot \mapsto I$, but it is not the case that $\vdash \tau \triangleright \cdot \mapsto ex \mapsto I$. In other words, the function $I$ is well-typed and satisfies the examples, but there is no derivation showing this. You do not need to prove this, but argue in natural language why there is no derivation.