A definition of type containment in Haskell

- Type instantiation (with a concrete type)
  - $\text{Maybe a} \quad \succ \quad \text{Maybe Int}$
  - $\text{Show a} \Rightarrow \text{a} \to \text{String} \quad \succ \quad \text{Bool} \to \text{String}$
- Swapping argument order
  - $\text{A} \to \text{B} \to \text{C} \quad \cong \quad \text{B} \to \text{A} \to \text{C}$
- "Inlining" non-recursive types which have a single constructor
  - $\textbf{data}\ (,)\ \text{a b}\ \textbf{where}\ (,) :: \text{a} \to \text{b} \to (\text{a, b})$
  - $(\text{a, b}) \to \text{c} \quad \cong \quad \text{a} \to \text{b} \to \text{c}$