- Parse a type signature and additional definitions in any programming language as long as parametric polymorphism is expressible

\[
\text{Any } a \Rightarrow \text{ total } \text{id} : a \rightarrow a
\]

- Interpret the type signature as a proposition and attempt to automatically find a constructive proof

\[
\frac{a_1 : a, \Gamma \vdash}{\Gamma \vdash \text{intro}\langle a_1 \rangle}
\]

\[
\frac{a_1 : a, \Gamma \vdash a}{\Gamma \vdash \text{trivial}\langle a_1 \rangle}
\]

\[
\frac{a_1 : a, \Gamma \vdash a}{\Gamma \vdash \text{clear}\langle a_1 \rangle}
\]

- Translate the proof into any programming language

\[
id = \text{intro}\langle a_1 \rangle \text{ trivial}\langle a_1 \rangle \text{ clear}\langle a_1 \rangle
\]

\[
id = \lambda a_1 \rightarrow a_1
\]

\[
id = \lambda a_1 \rightarrow a 1 \quad \text{-- Valid Haskell}
\]