Bidirectional Typing

- *Type inference* is generally undecidable for any language that is complicated enough.
- Instead of giving up type inference or choosing a simpler language, **type annotations** can be added to programs whenever *necessary*.
- *Bidirectional type synthesis* switches between two modes when traversing a term:
  - synthesis $\Gamma \vdash t :^{\Rightarrow} A$
  - checking $\Gamma \vdash t :^{\Leftarrow} A$
- The technique does **scale up** and has been applied to many languages.

From Simply Typed λ-Calculus to a Bidirectional Variant

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A}\ \textsc{Var}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash (t \mathbin{⨾} A) : A}\ \textsc{Anno}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \supset B}\ \textsc{Abs}$$

$$\frac{\Gamma \vdash t : A \supset B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\ u : B}\ \textsc{App}$$