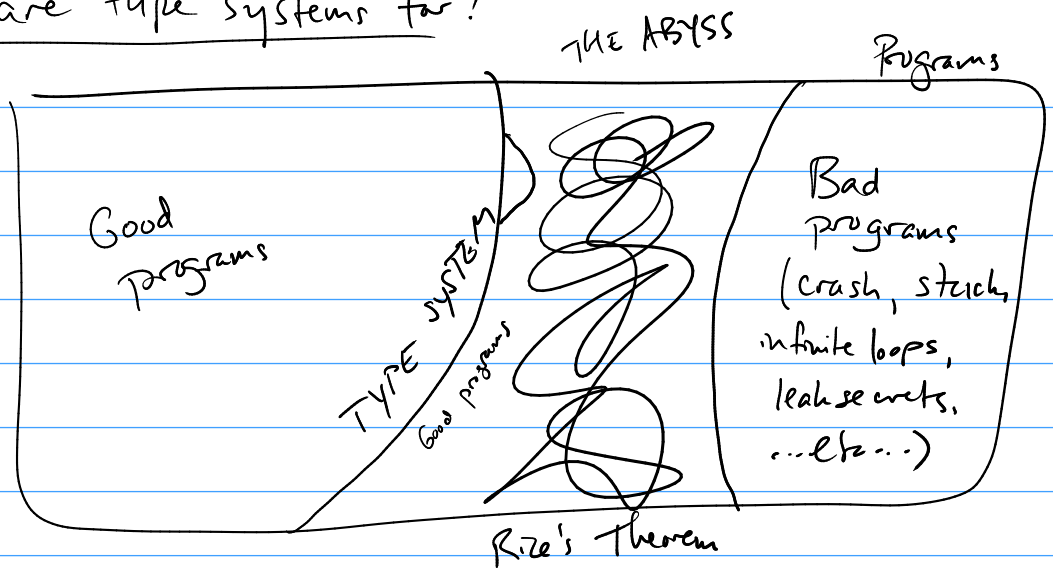
What are type systems for?



Simply Typed λ-calculus (STLC)

Goal: type system for λ-calculus that rules out nonterminating (aka infinite recursion).

$$t ::= x \mid \lambda x:\alpha.\, t \mid t_1\ t_2$$

$$\alpha, \beta ::= A, B, C, \ldots \mid \alpha \to \beta$$

(base types) (function type)

$$\boxed{\Gamma \vdash t : \alpha}$$

Γ = context of variable types (ie Map Var Type)

$$\frac{\Gamma \vdash t_1 : \alpha \to \beta \quad \Gamma \vdash t_2 : \alpha}{\Gamma \vdash t_1\ t_2 : \beta} \qquad \frac{x:\alpha \in \Gamma}{\Gamma \vdash x : \alpha}$$

$$\frac{\Gamma, x:\alpha \vdash t : \beta}{\Gamma \vdash \lambda x:\alpha.\, t : \alpha \to \beta}$$

$$\boxed{\Gamma \vdash t : \alpha}$$

t encodes a proof of the proposition α.

$$\frac{\Gamma \vdash t_1 : \alpha \to \beta \quad \Gamma \vdash t_2 : \alpha}{\Gamma \vdash t_1\ t_2 : \beta}\ \to\text{-elim} \qquad \frac{x:\alpha \in \Gamma}{\Gamma \vdash x : \alpha}\ Ax$$

$$\frac{\Gamma, x:\alpha \vdash t : \beta}{\Gamma \vdash \lambda x:\alpha.\, t : \alpha \to \beta}\ \to\text{-intro}$$