

# Type systems

What is the purpose of a type system?

- ① Describes the possible programs/values and how to construct them.
- ② Picks out "good" programs from the space of all possible programs.

What does "good" mean? It depends! e.g.:

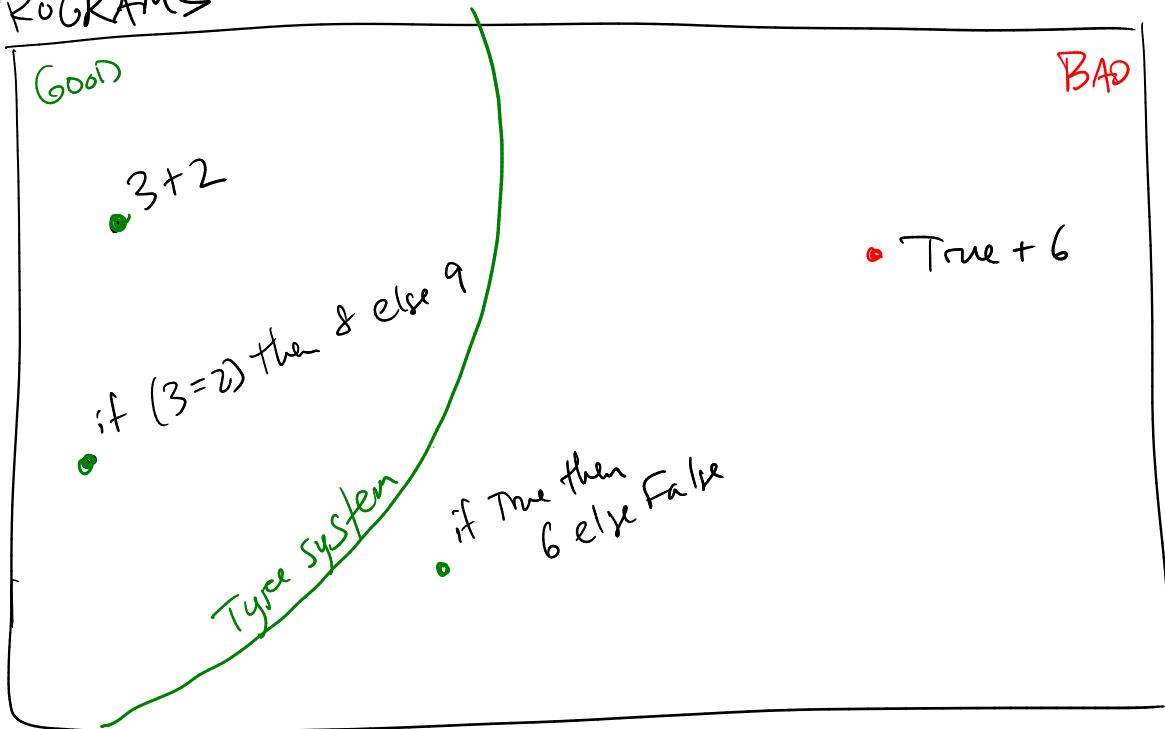
- Doesn't do nonsense operations.
- Doesn't crash. —
- Doesn't leak secret information. —
- Uses units correctly.
- ⋮

Consider example:

$(\text{if } (3 == 5) \text{ then } \underline{\text{True}} \text{ else } \underline{6}) + 5$

Actually is a "good" program — doesn't crash or do bad operations. But a static type system would reject it.

# PROGRAMS



Type = guarantee about the behavior of a program at runtime.

Static type system = make a guarantee about a program without running it.

Static type system for Arith + let + variables + bools + if.

$e ::= \underbrace{n}_{\substack{\uparrow \\ \text{integer} \\ \text{literal}}} \mid \text{True} \mid \text{False} \mid \underbrace{x}_{\substack{\uparrow \\ \text{variable}}} \mid e_1 \text{ op } e_2 \mid \text{let } x = e_1 \text{ in } e_2 \mid \text{if } e_1 \text{ then } e_2 \text{ else } e_3.$

$t ::= \text{Int} \mid \text{Bool}.$

$op ::= + \mid - \mid * \mid / \mid < \mid =$

"Inference rules": 
$$\frac{A \quad B \quad C \quad D}{E}$$

← if the things on top are all true THEN the thing on the bottom.

$$\boxed{e : t}$$

"e has type t"

$$\frac{}{n : \text{Int}}$$

$$\frac{}{\text{True} : \text{Bool}}$$

$$\frac{}{\text{False} : \text{Bool}}$$

$$\frac{e_1 : \text{Int} \quad e_2 : \text{Int}}{e_1 + e_2 : \text{Int}}$$

Similar rules for  $-$ ,  $*$ ,  $/$

$$\frac{e_1 : \text{Int} \quad e_2 : \text{Int}}{e_1 < e_2 : \text{Bool}}$$

$$\left( \frac{e_1 : \text{Bool} \quad e_2 : \text{Bool}}{e_1 < e_2 : \text{Bool}} \right)$$

$$\frac{e_1 : \text{Bool} \quad e_2 : t \quad e_3 : t}{\text{if } e_1 \text{ then } e_2 \text{ else } e_3 : t}$$