22-1/ Type Systems

syntactic	"return 1+ ;"	
logic	"return 42;"	
partial funs	1/0	first (empty)
type errors	"four" + 1	

∴ it is possible to automatically determine if they are <u>not</u> present

22-2/ Type Systems predict if your program won't have an error

Theory — predictions

Model — real phenomenon

"$X \vdash P$" the theory X predicts P

"$Y \models Q$" the model Y performs Q

Newton's
Gravity $\vdash$ My pen will drop. ? Yes

Universe $\models$ my pen will drop ? Yes

22-3/

$\forall P.$ Theory $\vdash P \Rightarrow$ Model $\models P.$ Soundness

$\forall P.$ Model $\models P \Rightarrow$ Theory $\vdash P.$ Completeness

22-4/ Our program is a model.

We want to predict what it does.

.... build a theory

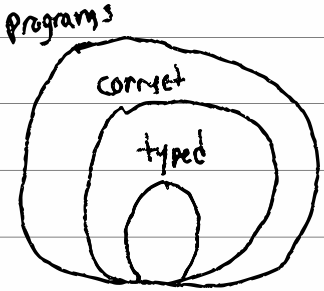
useful if prediction is faster than model

and sound

okay if not complete

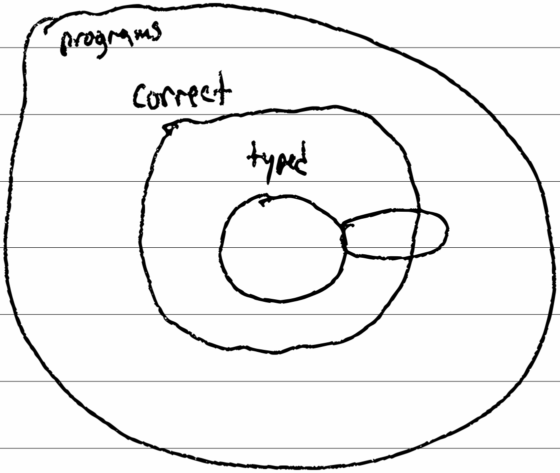
both are not possible

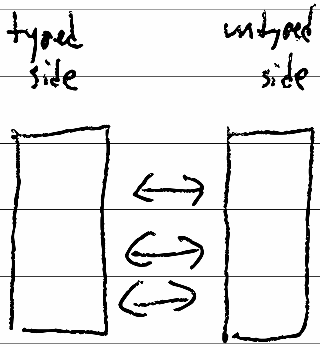
Gödel Incompleteness Theorem



typed = our sound theory says they are okay

22-5/ Gradual Typing





Typed → Untyped — errors are untype's fault

Untyped → Typed — errors are untype's fault

22-6/ Typed Racket
Type Script
Hack
Reticulated Python
Coffee Script

(typed e) => e after checking w/ type checker

(untyped T e) => (protect/contract e T
'untyped 'typed)

22-7 / $\forall p.\quad \text{Theory} \vdash P \Rightarrow \text{Model} \vDash P.$

/ \   / \

e I predict it will be in same set

$e \to e' \to e'' \to e''' \to \Rightarrow v$ where v is in same set

$e = v \mid (+\ e\ e) \mid (<\ e\ e) \mid (\text{not}\ e)$

$v = \text{true} \mid \text{false} \mid n$

$T = \text{Bool} \mid \text{Num}$

$\vDash \text{true} : \text{Bool} \qquad \vDash \text{false} : \text{Bool}$

$\vDash n : \text{Num}$

$E = \text{hole} \mid (+\ E\ e)$
$\mid (+\ v\ E) \mid (<\ E\ e)$
$\mid (<\ v\ E) \mid (\text{not}\ E)$

22-8/ inference rules

$$\frac{\text{conditions}}{\text{conclusion}}$$

$$\frac{\text{is raining} \qquad \text{am outside}}{\text{am wet}} \qquad \forall x, n.\ \frac{x = 0}{x \cdot n = 0}$$

$$\frac{}{\vdash \text{true} : \text{Bool}}\ A \qquad \frac{}{\vdash \text{false} : \text{Bool}}\ B \qquad \frac{}{\vdash n : \text{Num}}\ C$$

$$\frac{\vdash e_1 : \text{Num} \qquad \vdash e_2 : \text{Num}}{\vdash (+\ e_1\ e_2) : \text{Num}}\ D \qquad \frac{\vdash e_1 : \text{Num} \qquad \vdash e_2 : \text{Num}}{\vdash (<\ e_1\ e_2) : \text{Bool}}\ E$$

$$\frac{\vdash e_1 : \text{Bool}}{\vdash (\text{not}\ e_1) : \text{Bool}}\ F$$

not an algorithm

22-9/ $\underline{\vdash (\text{not } (< 1\ \ (+\ 2\ \ 3))) : \text{Bool}}\ \ F$

$$\underline{\vdash (< 1\ \ (+\ 2\ \ 3)) : \text{Bool}}\ \ E$$

$$\underline{\vdash 1 : \text{Num}}\ \ C \qquad \underline{\vdash (+\ 2\ 3) : \text{Num}}\ \ D$$

$$\underline{\vdash 2 : \text{Num}}\ \ C \qquad \underline{\vdash 3 : \text{Num}}\ \ C$$

(not (< 1 (+ 2 3))

(not (< 1 5))

(not true)

false $\vDash$ false : Bool? ✓

22-10

```
typeof : e => T
typeof bool(b) = Bool
typeof num(n) = Num
typeof not(e) =
    case typeof e of
        Bool -> Bool
        Num -> error "no type"
typeof add(e1, e2) =
    case typeof e1 of
        Num -> case typeof e2 of
                   Num -> Num
                   Bool -> error
        Bool -> error
```

22-11/ $e = \ldots \mid$ (read)

$\vdash$ (read) : num $\models$ (read) : Num

typeof read() = Num

not (< (read) (+ 2 3)) : Bool

22-12/

$$
\begin{aligned}
&\forall e, T. \\
&\quad \vdash e : T \\
&\Rightarrow \exists v, \quad e \to\to\to\to v \\
&\qquad \land \quad \vdash v : T
\end{aligned}
$$

an example soundness theorem

Progress

$$
\begin{aligned}
&\forall e, T. \\
&\quad \vdash e : T \\
&\Rightarrow (\exists e'. \; e \to e') \\
&\quad \lor (e \in V)
\end{aligned}
$$

Preservation

$$
\begin{aligned}
&\forall e, T, e'. \\
&\quad \vdash e : T \\
&\land \quad e \to e' \\
&\Rightarrow \quad \vdash e' : T
\end{aligned}
$$