25-1/ $e ::= \ldots \mid \{x : e, \ldots\} \mid e.x$

$v ::= \ldots \mid \{x : v, \ldots\}$

$T ::= \ldots \mid \{x : T, \ldots\}$

distFromOrigin ($\{x : Num, y : Num\}$ p) {

return $\sqrt{p.x^2 + p.y^2}$;

distFromOrigin ($\{x : 1+2, y : 3+4\}$)

X distFromOrigin (3)

X distFromOrigin ($\{x:3, y:7, z:8\}$)

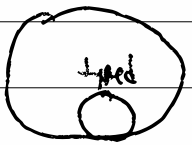
$$\frac{\Gamma \vdash e : \{x_1 : T_1, \ldots, x_i : T_i, \ldots, x_n : T_n\}}{\Gamma \vdash e.x_i : T_i}$$

$$\frac{\Gamma \vdash e_1 : T_1 \quad \ldots \quad \Gamma \vdash e_n : T_n}{\Gamma \vdash \{x_1 : e_1, \ldots, x_n : e_n\} : \{x_1 : T_1, \ldots, x_n : T_n\}}$$

25-2/ Types make predictions

.... typed programs don't crash

connect



OLD: $\dfrac{\Gamma \vdash f : D \Rightarrow R \quad \Gamma \vdash a : D}{\Gamma \vdash f\ a : R}$

NEW: "X and Y are compatible"

$\dfrac{\Gamma \vdash f : X \Rightarrow R \quad \Gamma \vdash a : Y}{\Gamma \vdash f\ a : R}$

25-3/ X and Y are compatible

... the Liskov Substitution Principle

$Y <: X$ — subtyping relation

Y is compatible with X if

any programs expecting an X

will not crash if given a Y.

Rotweiller <: Dog Dog ≮: Rotweiller Dog <: Animal

Animal ≮: Dog

$$\frac{}{T <: T} \qquad \frac{F <: A \quad G <: B}{F \times G <: A \times B} \qquad \frac{F <: A \quad G <: B}{F + G <: A + B}$$

$$\{(f_0, T_0), \ldots (f_n, T_n)\} \supseteq \{(g_0, T_0), \ldots, (g_m, T_m)\}$$

$$\{f_0 : T_0, \ldots, f_n : T_n\} <: \{g_0 : T_0, \ldots, g_m : T_m\}$$

25-4)

$$\frac{\{x, y, z\} \geq \{x, y\} \checkmark}{\{x:N,\ y:N,\ z:N\} <: \{x:N,\ y:N\}}$$

compare two fuctions

$$\frac{X_d <: Y_d \qquad Y_r <: X_r}{Y_d \to Y_r \quad <: \quad X_d \to X_r}$$

Animal → Pos <: Animal → Num

Cat → Num <: Animal → Num ? X

f (weight : Animal → Num) {

weight (~~Dog~~ whale) + weight (Cat)

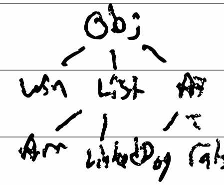
25-5

Structural subtyping	Nominal subtypn "by name"
US / Theory / J-lang	C++ Java

Haskell
ML
Coq

Python
JS
Racket

"duck typing"



$$\frac{X = Y}{Y <: X}$$

$$\frac{Y \text{ inherits from } X}{Y <: X}$$

class Posn { int x, y; }   class Posn3d { int x, y, z; }