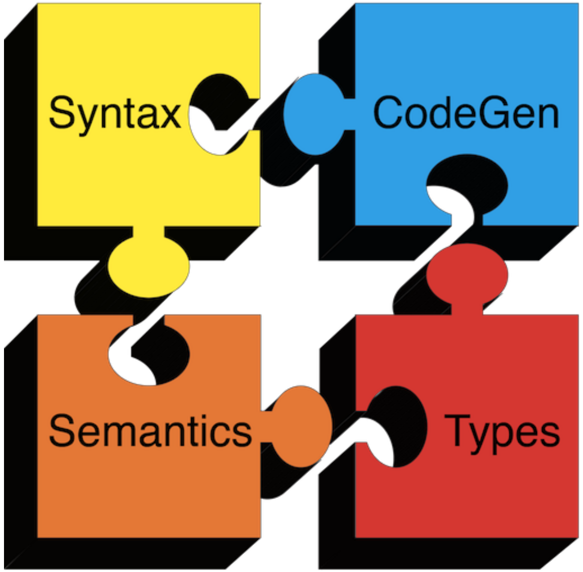

Compilers

Subtyping

Alex Aiken

Consider let with initialization:

$$\frac{\begin{array}{c} O \vdash e_0 : T_0 \\ O[T_0/x] \vdash e_1 : T_1 \end{array}}{O \vdash \text{let } x : T_0 \leftarrow e_0 \text{ in } e_1 : T_1} \quad \text{[Let-Init]}$$

Subtyping

- Define a relation $\leq$ on classes
  - $X \leq X$
  - $X \leq Y$ if X inherits from Y
  - $X \leq Z$ if $X \leq Y$ and $Y \leq Z$

$$\frac{\begin{array}{c} O \vdash e_0 : T_0 \\ O[T/x] \vdash e_1 : T_1 \\ T_0 \leq T \end{array}}{O \vdash \text{let } x{:}T \leftarrow e_0 \text{ in } e_1 : T_1} \quad \text{[Let-Init]}$$

Subtyping

$$\frac{\begin{array}{c} O(x) = T_0 \\ O \vdash e_1 : T_1 \\ T_1 \leq T_0 \end{array}}{O \vdash x \leftarrow e_1 : T_1} \quad \text{[Assign]}$$

Subtyping

- Let $O_C(x) = T$ for all attributes $x{:}T$ in class C

$$\frac{\begin{array}{c} O_C(x) = T_0 \\ O_C \vdash e_1 : T_1 \\ T_1 \leq T_0 \end{array}}{O_C \vdash x{:}T_0 \leftarrow e_1;} \quad \text{[Attr-Init]}$$

Subtyping

- Consider:
  if e_0 then e_1 else e_2 fi

- The result can be either e_1 or e_2

- The type is either e_1's type of e_2's type

- The best we can do is the smallest supertype larger than the type of e_1 or e_2

Subtyping

- lub(X,Y), the *least upper bound* of X and Y, is Z if
  - $X \leq Z \wedge Y \leq Z$

    Z is an upper bound

  - $X \leq Z' \wedge Y \leq Z' \Rightarrow Z \leq Z'$

    Z is least among upper bounds

- In COOL, the least upper bound of two types is their least common ancestor in the inheritance tree

Subtyping

Given the class definitions at right, which of the following least upper bound statements are true?

- ☐ lub(Point, Quad) = Object
- ☐ lub(Square, Rect) = Quad
- ☐ lub(Square, Rect) = Rect
- ☐ lub(Square, Circle) = Object

Class Object
Class Bool inherits Object
Class Point inherits Object
Class Line inherits Object
Class Shape inherits Object
Class Quad inherits Shape
Class Circle inherits Shape
Class Rect inherits Quad
Class Square inherits Rect

Subtyping

$$\frac{\begin{array}{c} O \vdash e_0 : \text{Bool} \\ O \vdash e_1 : T_1 \\ O \vdash e_2 : T_2 \end{array}}{O \vdash \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \text{ fi} : \text{lub}(T_1, T_2)} \quad \text{[If-Then-Else]}$$

Subtyping

- The rule for case expressions takes a lub over all branches

$$\frac{\begin{array}{c} O \vdash e_0 : T_0 \\ O[T_1/x_1] \vdash e_1 : T_{1'} \\ \ldots \\ O[T_n/x_n] \vdash e_n : T_{n'} \end{array}}{O \vdash \text{case } e_0 \text{ of } x_1 : T_1 \rightarrow e_1; \ldots; x_n : T_n \rightarrow e_n; \text{ esac} : \text{lub}(T_{1'}, \ldots, T_{n'})} \quad \text{[Case]}$$