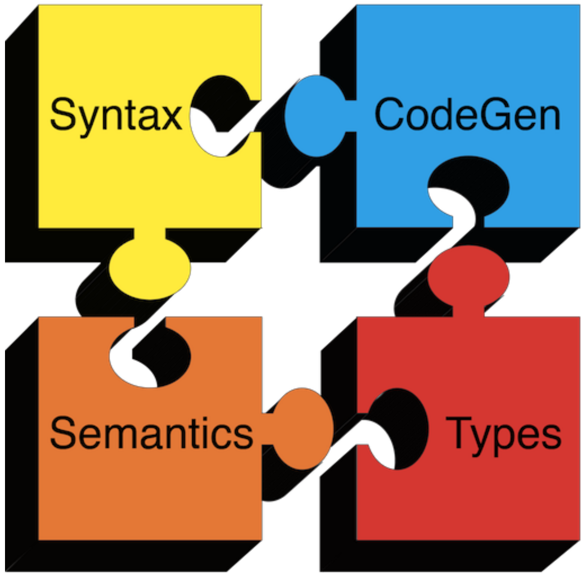

Compilers

Implementing Type Checking

Alex Aiken

Implementing TC

- COOL type checking can be implemented in a single traversal over the AST
- Type environment is passed down the tree
  - From parent to child
- Types are passed up the tree
  - From child to parent

Implementing TC

$$\frac{O,M,C \vdash e_1 : \text{Int} \quad O,M,C \vdash e_2 : \text{Int}}{O,M,C \vdash e_1 + e_2 : \text{Int}} \text{[Add]}$$

```
TypeCheck(Environment, $e_1$ + $e_2$) = {
    $T_1$ = TypeCheck(Environment, $e_1$);
    $T_2$ = TypeCheck(Environment, $e_2$);
    Check $T_1$ == $T_2$ == Int;
    return Int; }
```

Implementing TC

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

```
TypeCheck(Environment, let x:T ← $e_0$ in $e_1$) = {
    $T_0$ = TypeCheck(Environment, $e_0$);
    $T_1$ = TypeCheck(Environment.add(x:T), $e_1$);
    Check  subtype($T_0$,$T_1$);
    return ; $T_1$}
```