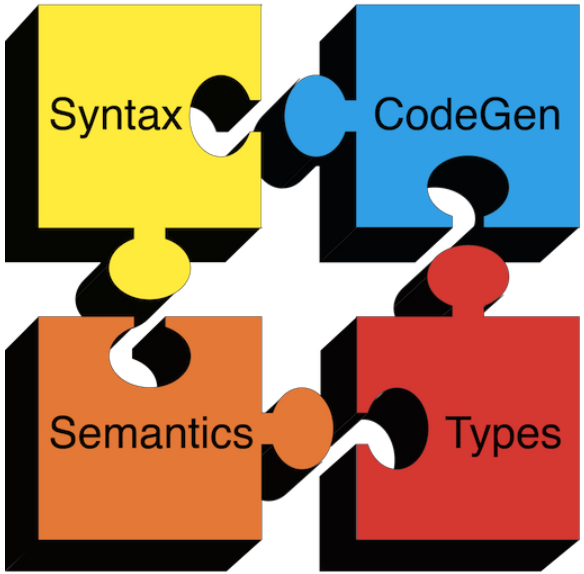

Compilers

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

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

O,M,C

```
TypeCheck(Environment, e1 + e2) = {
    T1 = TypeCheck(Environment, e1);
    T2 = TypeCheck(Environment, e2);
    Check T1 == T2 == Int;
    return Int; }
```

Implementing TC

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

```
TypeCheck(Environment, let x:T ←e0 in e1) = {
    T0 = TypeCheck(Environment, e0);
    T1 = TypeCheck(Environment.add(x:T), e1);
    Check  subtype(T0,T1);
    return T1}
```