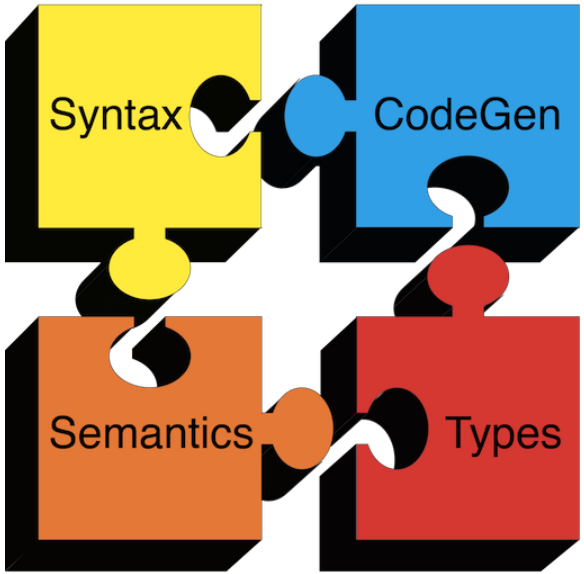

Compilers

Self Type Operations

Self Type Operations

```
class Count {
    Int i ← 0;
    inc(): SELF_TYPE { i ← i+1; self; };
};
```

- What can be the dynamic type of the object returned by inc?
  - Answer: whatever could be the dynamic type of "self"

Count or any subclass of Count

```
class A inherits Count { } ;
class B inherits Count { } ;
class C inherits Count { } ;
```

(inc could be invoked through any of these classes)

class C { ... E : SELF_TYPE ... }

Self Type Operations

- In general, if SELF_TYPE appears textually in the class C as the declared type of E then

$$\text{dynamic\_type}(E) \leq C$$

- Note: The meaning of SELF_TYPE depends on where it appears
  - SELF_TYPE_C to refers to an occurrence of SELF_TYPE in the body of C

- This suggests a typing rule:

$$\text{SELF\_TYPE}_C \leq C \qquad (*)$$

Self Type Operations

- Rule (*) has an important consequence:
  - In type checking it is always safe to replace $SELF\_TYPE_C$ by C

- This suggests one way to handle SELF_TYPE :
  - Replace all occurrences of $SELF\_TYPE_C$ by C

Self Type Operations

- Recall the operations on types
  - $T_1 \leq T_2$ T_1 is a subtype of T_2
  - $lub(T_1,T_2)$ the least-upper bound of T_1 and T_2

- Extend these operations to handle SELF_TYPE

Self Type Operations

Let T and T' be any types but SELF_TYPE

A ≤ A
C ≤ C

1. $SELF\_TYPE_C \leq SELF\_TYPE_C$
   - In Cool we never compare SELF_TYPEs coming from different classes

2. $SELF\_TYPE_C \leq T$ if $C \leq T$
   - $SELF\_TYPE_C$ can be any subtype of C
   - This includes C itself
   - Thus this is the most flexible rule we can allow

Self Type Operations

3. $T \leq SELF\_TYPE_C$ always false

   Note: $SELF\_TYPE_C$ can denote any subtype of C.

4. $T \leq T'$ (according to the rules from before)

Self Type Operations

Choose the subtype relations that are true for the class definitions given at right

- ☐ $Square \leq SELF\_TYPE_{Shape}$
- ☐ $SELF\_TYPE_{Circle} \leq Quad$
- ☐ $SELF\_TYPE_{Shape} \leq Shape$
- ☐ $SELF\_TYPE_{Rect} \leq Shape$

```
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
```

Self Type Operations

Let T and T' be any types but SELF_TYPE

1. $lub(SELF\_TYPE_C, SELF\_TYPE_C) = SELF\_TYPE_C$

2. $lub(SELF\_TYPE_C, T) = lub(C, T)$

   This is the best we can do because $SELF\_TYPE_C \leq C$

3. $lub(T, SELF\_TYPE_C) = lub(C, T)$

4. $lub(T, T')$ defined as before