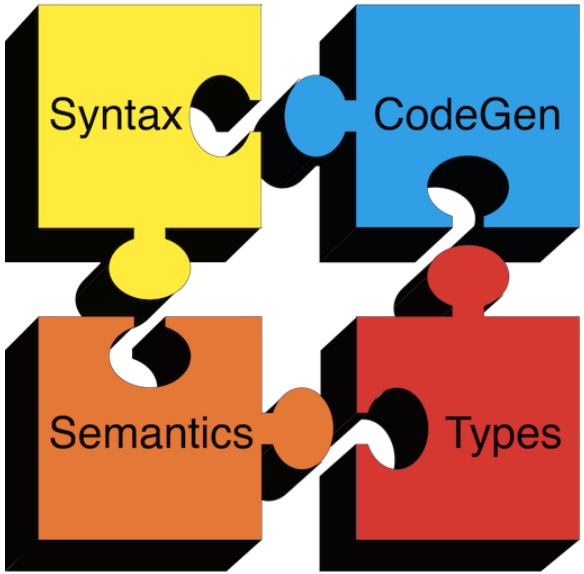

Compilers

Self Type Usage

Self Type Usage

- The parser checks SELF_TYPE appears only where a type is expected

- But SELF_TYPE is not allowed everywhere a type can appear:

1. class T inherits T' {…}
   - T, T' cannot be SELF_TYPE
2. x : T
   - Attribute x
   - T can be SELF_TYPE

Self Type Usage

3. let x : T in E
   - T can be SELF_TYPE

4. new T
   - T can be SELF_TYPE
   - Creates an object of the same type as self

dynamic

5. m@T(E_1,…,E_n)
   - T cannot be SELF_TYPE

Self Type Usage

$e.m(e')$ $e' : T_0$ $T_0 \leq SELF\_TYPE_C$

6. m(x : T) : T' { … }
   - Only T' can be SELF_TYPE !

What could go wrong if T were SELF_TYPE?

```
class A {  comp(x : SELF_TYPE) : Bool  {…};  };
class B inherits A {
     b : int;
   comp(x : SELF_TYPE) : Bool { … x.b …};  };
…
  let x : A ← new B in  … x.comp(new A); …
…
```

Self Type Usage

Which of the following usages of SELF_TYPE is incorrect?

- ...
  let x: SELF_TYPE <-
    (new SELF_TYPE) in
      { ... };
  ...

- class Animal {
    addFriend(friend: Animal): SELF_TYPE
      { ... }
    ...
  }

- ...
  (new Animal)@SELF_TYPE.bark();
  ...

- ...
  (new SELF_TYPE).foo();
  ...