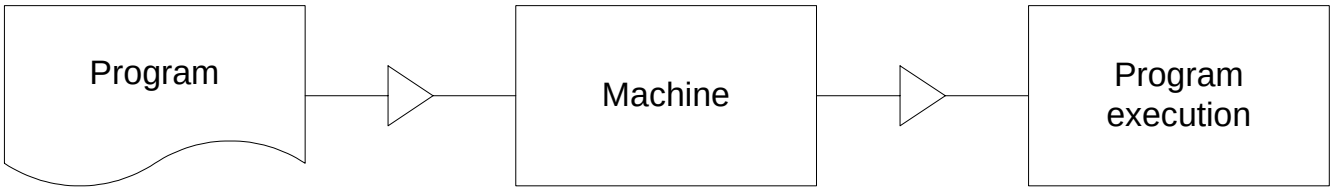
Syntax/semantics



Syntax

Semantics

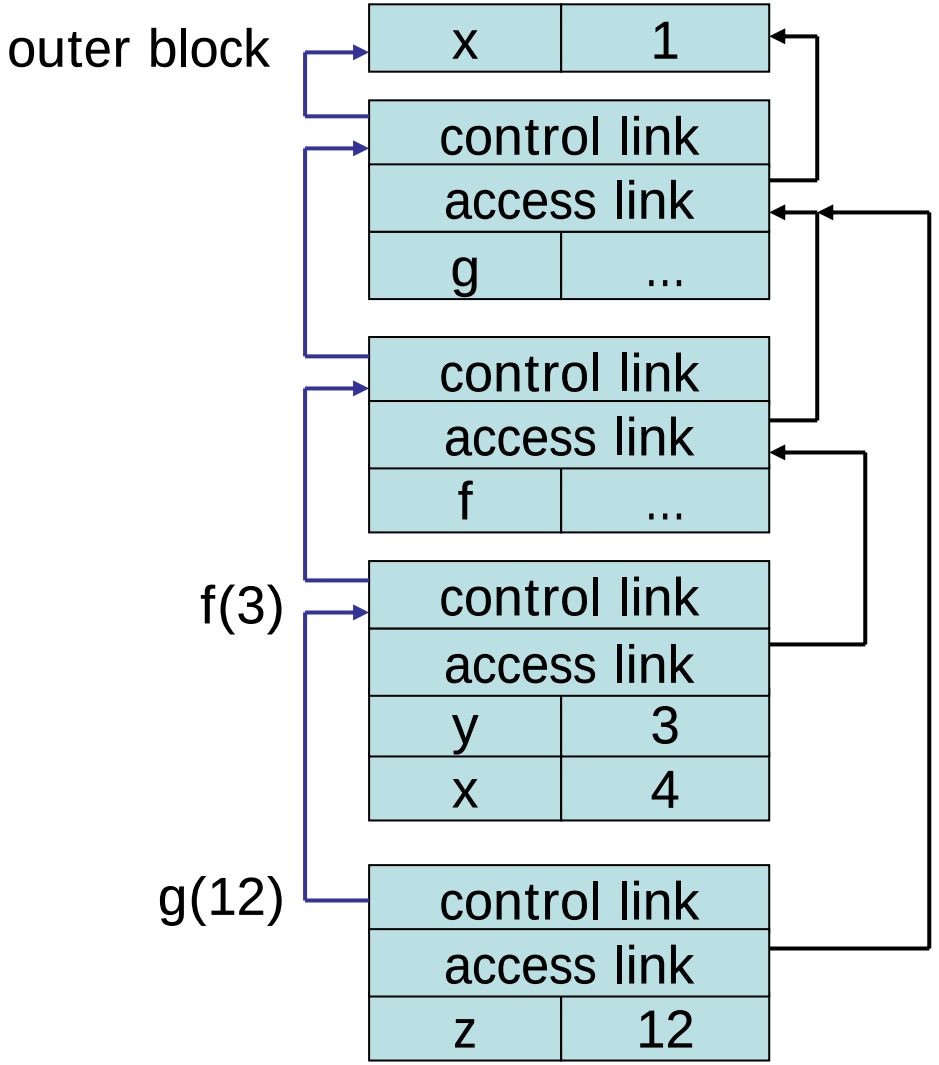
- Differences and similarities between
  - Parse trees
  - (Abstract) syntax trees
  - Meta models
- How language concepts are defined
  - Grammar
  - Static semantics
  - Dynamic semantics

Runtime organization

Static scope with access links – and closures!

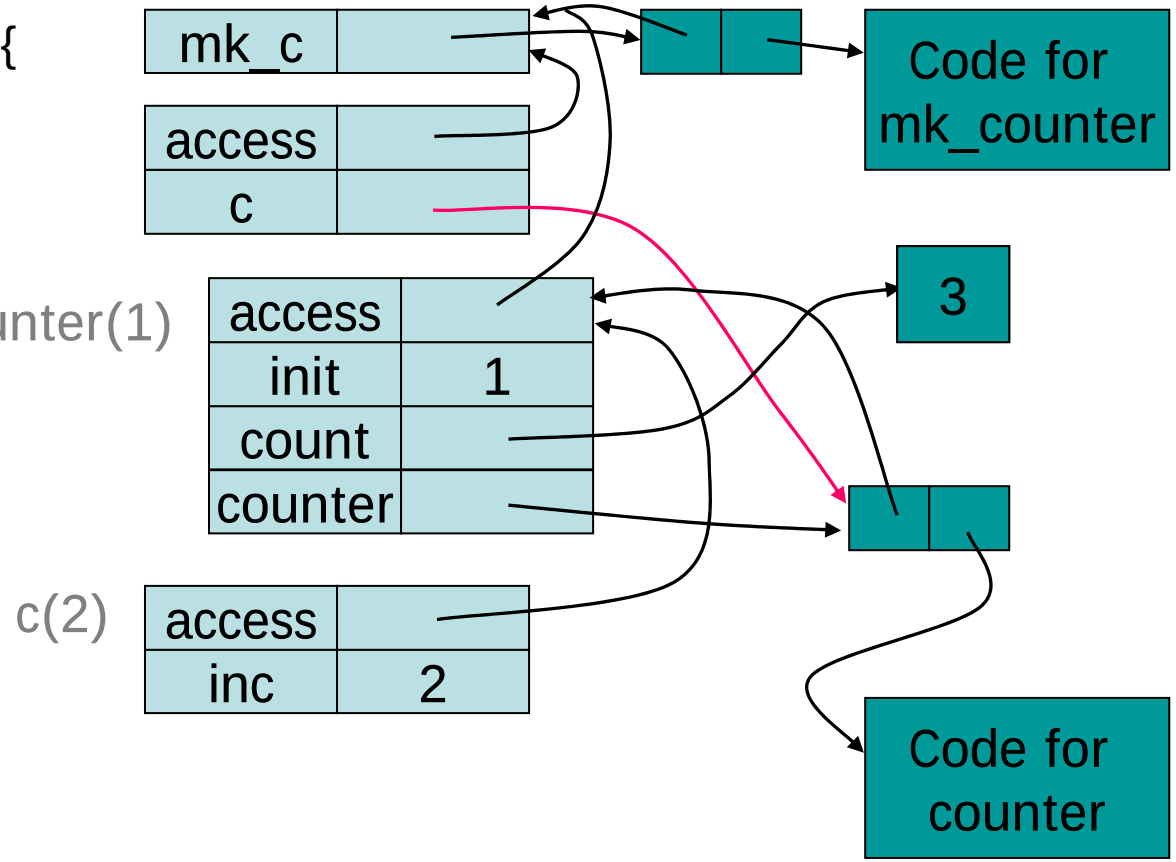
```
int x=1;
function g(z) = x+z;
function f(y) =
   { int x = y+1;
     return g(y*x) };
f(3);
```

- Stack of activation records
- Dynamic scope with control link
- Static scope with access links
- Closures!



Function Results and Closures

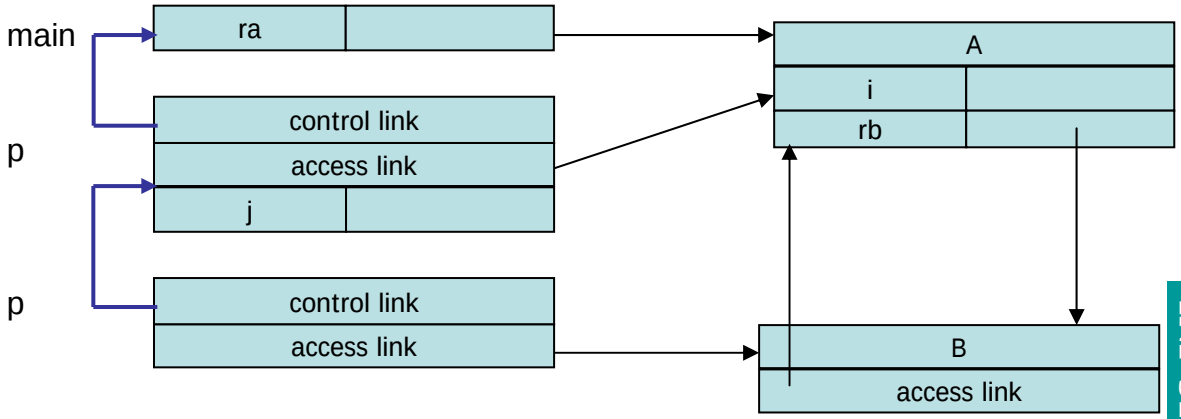
```
{int→int mk_counter (int init) {
      int count = init;
      int counter(int inc)
        { return count+=inc;}
      }
  int→int c = mk_counter(1);
  print c(2) + c(2);
}
```



Call changes cell value from 1 to 3

Access links for object oriented languages

```
class Program {
   public static void
   main(String[] args) {
      A ra=new A(); ra.p();
   };
};
class A{
   int i=1;
   class B {
      void p(){ i=2; };
   };
   B rb = new B();
   void p(){
      int j=i;
      rb.p();
   };
};
```



- Access links for method activations may be objects
- Objects may also have access links (classes in classes, or classes in methods)

Parameter passing

- Pass-by-value
- Pass-by-reference
- Pass-by-name

Type checking – type/subtype

- Static/compile-time
- Dynamic/run-time

- Overriding/redefinition of virtual methods versus overloading
- Covariance - contravariance

Inner classes - locally defined classes

```
class Apartment {
    Height height;
    Kitchen theKitchen = new Kitchen {… height  …}();
    class ApartmentBathroom extends Bathroom {… height …}
    ApartmentBathroom Bathroom_1 = new ApartmentBathroom ();
    ApartmentBathroom Bathroom_2 = new ApartmentBathroom ();
    Bedroom theBedroom = new Bedroom ();
    FamilyRoom theFamilyRoom = new FamilyRoom ();
    . . .
    Person Owner;
    Address theAddress = new Address()
};
```