Points-to analysis for Java

Daqing Hou

Outline

- Context and context-sensitive
- Field-sensitive and object-sensitive
- Examples

Context-sensitive

M. Sharir, A. Pneuli. Two approaches to inter-procedural dataflow analysis.

- Keeping calling contexts distinct during analysis
- Two approaches to distinguishing analysis result:
  (1) **call string**: by call stack on which it is obtained (k-CFA)
  (2) **functional**: by program state at call (eg receiver identity, argument types, or combination)

Example 1

```
static void main() {
  B b1 = new B(); // OB
  A a1 = new A(); // OA
  A a2, a3;
C1: a2 = f(b1);
C2: a2.foo();
C3: a3 = f(a1);
C4: a3.foo();
}

static A f(A a4) { return a4;}
```

- **A is a supertype of B.**
- **Both A and B define foo().**

Q:

Which foo() is invoked at C2 and C4?

A:

1. *A programmer's answer*
   C2: B.foo(); C4: A.foo()
2. *Anderson's algorithm*
   C2, C4 : A.foo() and B.foo()
3. *1-CFA solution*
4. *Discussion: limitations?*

Example 1'

```
static void main() {
  B b1 = new B(); // O_B
  A a1 = new A(); // O_A
  A a;
C1: a = f(b1);
C2: a.foo();
C3: a = f(a1);
C4: a.foo();
}

static A f(A a4) { return a4; }
```

- A is a supertype of B.
- Both A and B define foo().

Q:

Which foo() is invoked at C2 and C4?

A:

1. 1-CFA solution

Example 2

```
static void main() {
  D d1 = new D(); // O_D
  if (…)
    C1: d1.f(new B()).g(); // O_B
  else
    C2: d1.f(new C()).g(); // O_C
}

class D {
A f(A a1) { return a1;}
}
```

- **B, C, D are subclasses of A.**
- **A, B, C define g().**

Q:

Which g() is invoked at C1 and C2?

A:

1. *A programmer's answer*
2. *1-CFA solution*
3. *2-CFA?*

Example 3

```
class A {
    X xx;
   A(X xa) { this.xx = xa;}     }
class B extends A {
   B(X xb) { super(xb);  }
    X f() { return this.xx; }    }
void main() {
  X x1, x2;
C1: B b1 = new B(new Y()); // O_B1, O_Y
C2: B b2 = new B(new Z()); // O_B2, O_Z
x1 = b1.f(); C3: x1.g();
x2 = b2.f(); C4: x2.g();}
```

- **Y & Z are subclasses of X.**
- **X, Y, Z define g().**

Q:

Which g() is invoked at C3 and C4?

A:

1. *A programmer's answer*
2. *1-CFA solution*