CS6013 - Modern Compilers: Theory and Practise Alias analysis

V. Krishna Nandivada

IIT Madras

Alias/Points to analysis

- May and must analysis.
- Flow sensitive and insensitive analysis.
- Abstract representation of Stack (ρ) and Heap (σ) .
- \bullet Rules for updating ρ and σ for each statement in minilR.
- Lattice (subset based).
- Meet operations for may and must analysis.



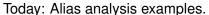
Opening remarks

What have we done so far?

- Compiler overview.
- Scanning and parsing.
- JavaCC, visitors and JTB
- Semantic Analysis specification, execution, attribute grammars.
- Type checking, Intermediate Representation, Intermediate code generation.
- Control flow analysis, interval analysis, structural analysis
- Data flow analaysis, intra-procedural and inter-procedural constant propagation.
- Loop Optimizations.
- Call Graphs, Inter procedural constant propagation.

Announcement:

Assignment 4 – one week to go.



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020



Example 1

```
i: m = new X(); // Ri
j: n = new X(); // Rj
k: p = m;
l: p = n;
a: q = p;
b: n = m;

Flow insensitive:
p -> {Ri, Rj}
q -> {Ri, Rj}
n -> {Ri, Rj}
```



Example 2



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020

.

o / 11

Example 4

```
insert (int m) {
   prev = lst
   while (lst != null) {
      prev = lst;
      lst = lst.next;
   }
   prev.next = new node(m);
   lst = prev;
}
```



Example 3

```
p = new A(); // R1
p.f = new Y(); // R2
if (cond) {
   q = new X(); // R3
   q.f = new Z(); // R4
   r1 = q;
} else {
   q = new X(); // R5
   q.f = new Z(); // R6
   r2 = q;
}
p.f = new Y(); // R7
q.f = new Z(); // R8
```



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020

6/1

Example 5

```
node insert (int [] a) {
   if (a.length == 0) return null;
   if (a.length == 1) return new node(a[0]);
   List lst = new node(a[0]); // R1
   for (int i=1;i <a.length;++i) {
        lst.next = new node(a[i]); // R2
        lst = lst.next;
   }
  return lst;
}</pre>
```



Example 6

```
node insert(node p, int m)
{
  node q;
  q = p;
  while (q != null) {
    if (q.val == m)
      return q;
    if (q.next == null) {
      q.next = new node(m); // R2
      }
      q = q.next
  }
  q = new node(m); // R1
  return q;
}
```



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020

9/11

Closing remarks

What have we done today?

• Flow sensitive intra-procedural alias analysis

To read

Muchnick - Ch 10, Dragon book - 12.4

Next:

Register Allocation



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020

11 / 11

Example 7

```
node find(node p, int m)
{
  node q;
  q = p;
  while (q != null) {
    if (q.val == m)
      return q;
    q = q.next
  }
  return null;
}
```



V.Krishna Nandivada (IIT Madras)

CS6013 - Jan 2020

10/11