

# CS6013 - Modern Compilers: Theory and Practise

## Control Tree based Data flow analysis

**V. Krishna Nandivada**

IIT Madras

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 analysis, intra-procedural and inter-procedural constant propagation.

Today: Data flow analysis on control trees.



## Control-tree based DFA

- Disadv: Harder to implement than the iterative methods.
- Adv: easier to update the data flow information (say incremental update).
- Historically, known as elimination methods.

Makes two passes over the control tree for a procedure.

- Pass 1: A bottom-up pass – constructs a flow function that represents the effect of executing the portion of the procedure.
- Pass 2: A top-down pass – starts from the abstract node for the whole procedure + initial information (of which block?). Construct and evaluate the data-flow equations to propagate through each region.



## Structural analysis (recall)

- A more refined form of interval analysis.
- Differs from basic interval analysis in that it identifies many types of control structures than just loops.
- Each such structure is turned into a region and provides a basis for doing efficient data-flow analysis on each of the different regions.
- Output - a control tree.  
Typically larger than that we find for interval analysis.  
But the individual regions are simpler and simpler.
- Region – has exactly one entry point –  
How to include an irreducible or improper region? add the lowest common dominator of the set of entries – entry point for the multiple-entry cycle.



On the board (reaching definitions and constant propagation)



What have we done today?

- Data flow analysis on control trees.

To read

- Muchnick - Ch 8.7

Next:

- Points to analysis

