Register Allocation

Register Allocation

- Register allocation improves code
  - accessing faster memory
  - fewer instructions
- But…
  - There are a limited number of machine registers
  - Some registers can only hold certain types of data
- So, which variables to we allocate to registers?
- Register allocation is extremely important
  - has huge impact on performance
  - it's NP-Complete (not solvable in polynomial time)
  - Use heuristics

Approaches to Register Allocation

- Global Register Allocation Using Usage Counts
  - Assume R registers are available. For each loop nest, allocate registers to the R variables which show the largest estimated benefit from being kept in a register. Little or no cross nest allocation is done.
- Register Allocation by Graph Coloring
  - currently the most common method
    - known about since 1971 but was impractical in early compilers
    - Chaitin came up with 1st implementation in 1981
    - Briggs proposed an optimistic extension to it around 1989
  - express overlap of the lifetimes of vars with an *interference* graph
  - try to *color* this graph with R colors
  - generate spill code when necessary to make the graph R-colorable