CSE 504: Compiler Design

Register Allocation

Pradipta De
pradipta.de@sunykorea.ac.kr
Current Topic

- Set of hardware registers is a constrained resource
- Compilers must allocate and assign hardware registers to program values

Goal: Use registers effectively to reduce number of load/stores to memory
Local Register Allocation

- Two approaches to replace virtual registers to physical registers

- Frequency counting approach
  - Count the number of references to a value in a block and use the frequency to rank which value stays in the register

- Count the spills
  - Walk over code block to determine if a spill is needed ➔ use the information for register allocation
Top-down frequency based Register Allocation

• Computer a priority for each virtual register
  – Count the number of times a VR occurs
• Sort the virtual registers in priority order
  – Priority is based on frequency ➔ can vary between 2 and the length of the block
• Assign the registers in priority order
  – Assign the first k registers to physical registers, where k is the number of physical registers
• Rewrite the code
  – Replace VR with physical registers
Bottom up Local Register Allocation

- Assume physical registers are initially empty
  - Create an empty list of physical registers
- Satisfy the demand for registers until the list is used up
  - Spill values into memory and reuse the register
  - Spill the value whose next use is farthest in future
- Issues:
  - What about a value that was created by loading from memory and does not need to be stored back
    • This is called “clean”, while a value that needs write back to memory is called “dirty”
  - What if the register contains a known constant value
Summary

• Local register allocation technique
• Different techniques needed for register allocation beyond basic blocks