CSE 401
Intro Compilers

Final Review
(mostly post-Midterm)

Larry Ruzzo
Spring 1998

Compile- vs Run-Time

- procedures vs activation record/stack frame
- scope/symbol table vs environment/stack frame
- variable vs memory/stack/register location
- lexically enclosing scope vs static link
- caller vs dynamic link

Run Time Storage

- layout of data structures
- memory areas: static, stack (lifo), heap
- layout of stack frame: formals, locals, links, etc.
- calling conventions
- parameter passing modes:
  call-by-value vs call-by-reference vs ...

Intermediate Code Gen

- Why? How different from target? (temps, machine (in)dependence, ...)
- 3-address code
- gen IR from AST:
  l- vs r-value, exprs, assign, arrays, ...
- Short circuit code

Target Code Gen

- Instruction selection (RISC/CISC)
- Register allocation

Optimization

- Deduce as much as possible at compile time about run time bindings, values, control flow, ...
- Use it to
  - Specialize unnecessarily general code
  - Reorder code
  - Exploit target machine
- Scope:
  - Peephole
  - Local
  - Global (intra-procedural)
  - Inter-procedural

harder ↓ better