

CSE401: Final review

David Notkin
Autumn 2000

Scope & ground rules

- Everything we covered on in class, with a much stronger focus on backend issues
- Ground rules are the same as the midterm: open book, open note, closed neighbor
- 8:30-10:20AM Wednesday December 13, 2000
 - In this very room
- The feel will be much like the midterm
- I will aim at a test that is around an hour long, but you have the full time to complete it

Backend overview

- Given a well-formed AST + symbol table, what can you do?
 - Analyses
 - Interpretation
 - Compilation
 - ...

Interpret vs. compile

- Tradeoffs
- Run-time and compile-time
- Advantages of one over the other
- Basic structure of an interpreter

Jobs of a compiler

- Representation and placement of run-time values
- Generate machine code
- Optimization

Activation records

- Distinguish from symbol tables
- What goes in them
- Static/dynamic links
 - What they are, why they are, and how they are managed

Procedure call protocols

- Related to activation records
- Handling registers
- Return locations
- Return values
- ...

Storage layout

- Representations of data
 - Atomic values, aggregates (arrays, records), etc.
- Location of data
 - Stack, global, heap, etc.

Parameter passing

- Call-by-value, call-by-reference, etc.
- The mechanisms
- The consequences of the mechanisms on programming language design and on programs

Code generation

- Structure of code generation, and benefits of that structure
- Intermediate vs. target code generation
- Three address code: what and why

Code generation

- Instruction selection
- Register allocation
- Impact of basic architectural features

Optimization

- The goal
- Different scopes
 - peephole, basic block, intraprocedural, interprocedural
- Examples of these various approaches

Implementation of optimization

- Analyses
 - live variable analysis
- Control and data flow graph representations
 - What and why
- Iterative dataflow analysis