CS415 Compilers

Topics in the design of programming language translators, including scanning, parsing, error recovery, code generation, and code improvement.

Prerequisites:

- CS 211 (Computer Architecture)
- CS 314 (Principles of Programming Languages)
- for additional prerequisits, please see the 415 WWW home page

Important facts:

staff:	Prof. Barbara Ryder CoRE 311 (ryder@cs.rutgers.edu)
lectures:	Mo & Wed 6:10 – 7:30 p.m.
recitations:	Wed 7:40 – 8:35 p.m.
Location:	lecture+recitation: Science & Engineering Resource Center SEC202

CS415 Compilers

Basis for grades:

10%	homework
15%	mid-term exam (March 10)
25%	final exam (May 7)
50%	one project in four "pieces"
20%	late penalty for each day past due date

Participation in class and recitation is factored into your grade.

Special permission numbers

- put your information on sign-up sheet.
- if you want, send mail to ryder@cs.rutgers.edu to explain your situation.

Overview

This course will cover the following topics (incomplete list):

1. Lexical Analysis (*scanning*)
2. Syntax Analysis (*parsing*)
3. Context-sensitive Analysis
4. Intermediate Representations
5. Code Generation
6. Code Improvement Techniques

References

- The required textbook for this course is *Modern Compiler Implementation in Java, Cambridge University Press, 1998*, by A. Appel.
- Additonal (recommended) text: *The Java Programming Language, Second Edition,Addison Wesley, 1998*, by K. Arnold and J. Gosling.

Overview (cont.)

Course material is available on the World Wide Web at

http://www.cs.rutgers.edu/~ryder/cs415

In addition, there is a news group for this class:

ru.nb.dcs.class.415

YOU SHOULD READ THE NEWS GROUP AT LEAST EVERY OTHER DAY

Programming Project

Programming Project: write a compiler for a small imperative language in **Java**:

- A lexical analyzer using `JLex`.
- A basic parser using `CUP (yacc)`.
- Type checking for a block-structured language (???).
- Code generation for a RISC processor (???).

All programming will be done on the CS UNIX Cluster in Java. Familiarity with object-oriented programming is a prerequisite; if you do not already know the Java language, then you will have to learn it **ON YOUR OWN** as part of the course.

THIS IS A LARGE PROJECT with work throughout the semester.
YOU SHOULD THINK TWICE BEFORE TAKING ANOTHER PROJECT COURSE (e.g.416, 431) WITH THIS ONE.

Class-taking technique

- all transparencies are on the Web (PostSript)
- you should still take some notes

- *You* need to read the book
- between the two, you should get a lot of material
- Recommended book is particularly useful for projects.
  Copies of the books will be on reserve

Compilers

What is a compiler?

- a program that translates an *executable* program in one language into an *executable* program in another language
- the compiler typically *lowers* the level of abstraction of the program
- we expect the program produced by the compiler to be *better*, in some way, than the original

What is an interpreter?

- a program that maps an *executable* program in one language into the *answers* produced by the executable program
- compilers and interpreters share many components

Motivation

Why build compilers?

Why study compiler construction?

Why attend class?

Reasons:

- compilers provide an essential interface between applications and architectures
- compilers embody a wide range of theoretical techniques
- compiler construction teaches programming and software engineering skills

Role of Compilers

High-level programming languages

- increase programmer productivity
- better maintenance
- portable

Low-level machine details

- instruction selection
- addressing modes
- pipelines
- registers & cache
- instruction-level parallelism

Compilers are needed to efficiently bridge the gap!

Isn't it a solved problem?

> *"Optimization for scalar machines is a problem that was solved ten years ago"*
>
> David Kuck
> Fall 1990 at Rice University

Machines have changed since 1980

Changes in architecture $\Rightarrow$ changes in compilers

- new features present new problems
- changing costs lead to different concerns
- must re-engineer well-known solutions

Significant differences in performance

Interest

Compiler construction shows us a microcosmic view of computer science.

artificial intelligence	greedy algorithms learning algorithms
algorithms	graph algorithms union-find network flows dynamic programming
theory	*dfa*'s for scanning parser generators lattice theory for analysis
systems	allocation and naming locality synchronization
architecture	pipeline management memory hierarchy management instruction set use

Inside a compiler, all these things come together.

As a result, compiler construction is challenging and fun.

Compiler Construction

Compilers are large, complex pieces of software.

By working on compilers, you'll learn to use

- programming tools (compilers, debuggers)
- program-generation tools (lex, yacc)
- software libraries (sets, collections)

Hopefully you will also enhance your software engineering skills.

Experience

You have used several compilers.

What qualities do you want in a compiler?

Here is a list:

1. Correct code
2. Output runs fast
3. Compiler runs fast
4. Compile time proportional to program size
5. Support for separate compilation
6. Good diagnostics for syntax errors
7. Works well with the debugger
8. Good diagnostics for flow anomalies
9. Cross language calls
10. Consistent, predictable optimization

Some Advice

The emphasis in CS415 is on compiler front ends. There are graduate courses that mainly deal with the back end and compiler optimizations (e.g., CS516).

Compiler construction is one of the most time-consuming undergraduate level programming courses, along with operating systems and databases.

Be prepared to spend a lot of time on projects if you don't have much programming experience in UNIX and Java. Be prepared to lose a lot of sleep if you don't start projects early!

Next Class

Prof. Ryder will be back on Monday.

Things to do:

- read Chapter 1 in required text
- start learning Java; the first Java assignment will be next week.
- learn to use *nn* to read news groups and post messages