Course Mechanics

Outline

1. Course Logistics

2. Course Overview

Course Logistics

Course website: http://www.swamiiyer.net/cs451 or .../cs651

Goal: write parsers, analyze and generate code for various programming constructs, and allocate physical registers to a program expressed in terms of virtual registers

Prerequisites: CS310 and CS420 or CS622

Instructor: Swami Iyer

Thrice a week classes (required)

Text: Introduction to Compiler Construction in a Java World by Bill Campbell, Swami Iyer, and Bahar Akbal-Delibaș

Course Logistics

Grading

- Project assignments (best 5 of 6): 45%
- Exams (best 2 of 3): 50%
- Attendance: 5%

CS account

We'll use the following online tools

- Piazza for Q&A
- iClicker for attendance
- Gradescope for grading

Policies

- Classroom etiquette
- Piazza etiquette
- Collaboration
- Code of conduct
- Accommodations for students with disabilities

Tips to succeed

Course Logistics

Items on the course website

- Announcements (landing page)
- Course Info
- Calendar
- Slides
- Assignments
- Resources

Things to do immediately

- Sign up for Piazza
- Sign up for iClicker
- Sign up for Gradescope
- Setup the programming environment
- Apply for a CS account

Course Overview

Compilation (Chapter 1)

- What are compilers and why study them?
- How does a compiler work? The phases of compilation
- An overview of the *j*-- to JVM compiler
- *j*-- compiler source tree

Lexical Analysis (Chapter 2)

- Tokens in a programming language and scanning them
- Regular expressions
- Finite state automata
- Non-deterministic finite-state automata (NFA) versus deterministic finite-state automata (DFA)
- Regular expressions to NFA
- NFA to DFA
- Minimal DFA
- Generating scanners using JavaCC

Course Overview

Parsing (Chapter 3)

- Context-free grammars and languages
- Top-down deterministic parsing
- Bottom-up deterministic parsing
- Generating parsers using JavaCC

Type Checking (Chapter 4)

- *j*-- types
- *j*-- symbol tables
- Pre-analysis of *j*-- programs
- Analysis of *j*-- programs

Course Overview

JVM Code Generation (Chapter 5)

- Classes and their members
- Control and logical expressions
- Message expressions, field selection, and array access expressions
- Assignment and similar operations
- String concatenation
- Casts

Translating JVM Code to MIPS Code (Chapter 6)

- What happens to JVM code?
- SPIM and MIPS architecture
- Our translator from JVM to MIPS

Register Allocation (Chapter 7)

- What is register allocation?
- Naïve register allocation
- Register allocation by graph coloring