CS 410/510 Compiling Advanced Languages – Spring 2000

Instructor:
Andrew Tolmach
PSU PCAT 128A
725-5492
email: `apt@cs.pdx.edu`
Office Hours: MW 11-12 or by appt.

Course meeting: TuFri 5:45-7pm at OGI

Course home page: `http://www.cs.pdx.edu/~apt/510cal`

Description

We will study compilers for a modern higher-order, typed ("HOT") programming languages such as Java and ML. The compiler will be constructed by building a series of progressively more complex interpreters, culminating in an interpreter that can be viewed as a code generator. Our implementation language for these interpreters will be OCAML. The emphasis of the course will be on understanding the distinctive challenges of compiling HOT languages. The main topics will include intermediate langugage design, type systems, dynamic function dispatch, whole-program optimization, storage management and garbage collection. We will not worry at all about "front-end" issues like parsing.

PSU Graduate students: Note that this course will substitute for CS577 in the "Languages" track.

Prerequisites

For everyone: Familiarity with object-oriented programming in Java or C++ and willingness to learn some ML.

For graduate students: PSU CS558 (or equivalent); a previous compiler course is desirable.

For undergraduates: Permission of the instructor. At a minimum, you need to have completed CS301/302 with good grades (and enthusiasm!).

Format

Two weekly lectures; bi-weekly homeworks; take-home final exam.

Tentative Schedule

Mar 28/31	Intro; Lambda Calculus
Apr 4/7	Core intermediate language; Data Representation
Apr 11/14	Objects
Apr 18/21	Type Systems; Polymorphism
Apr 25/28	Interpreters
May 2/5	Abstract and concrete machines
May 9/12	Dynamic function dispatch
May 16/19	Optimization
May 23/26	Code generation
May 30/Jun 2	Garbage collection

Programming

The homeworks and final will involve programming exercises in OCAML, the French dialect of ML. You will want to download the most recent executable and documentation for the system for use on your own machine (linux or windows) from

`http://pauillac.inria.fr/ocaml/`

CAML differs from Standard ML in superficial syntax and (just a few) semantic features, but you should have no trouble transfering whatever knowledge you have about SML to CAML.

Supporting materials for exercises will be made available on the course web page.

Texts

There is no required text.

Some readings may be handed out (or pointed at on the web) as the course progresses.

You will probably want access to a reference book on ML, in addition to the reference manual available with the OCAML system. Unfortunately, the only textbook specifically about the CAML dialect is an introduction to programming (rather than an introduction to ML) – but it is good as far as it goes:

Guy Cousineau and Michel Mauny, *The Functional Approach to Programming*, Cambridge University Press, 1998.

There are quite a number of books on Standard ML; I'd suggest

Larry Paulson, *ML for the Working Programmer*, Cambridge University Press, 2nd, ed., 1996.

but lots of others are ok. See

`http://cm.bell-labs.com/cm/cs/what/smlnj/doc/literature.html`

for more resources (including web tutorials) on SML. Again, most of what you read will transfer to CAML with some minor syntatic adjustments.