Mads Tofte, March 1, 1989
Laboratory for Foundations of Computer Science
Department of Computer Science
Edinburgh University

Four Lectures on Standard ML

The following notes give an overview of Standard ML with emphasis placed on the Modules part of the language.

The notes are, to the best of my knowledge, faithful to "The Definition of Standard ML, Version 2"[1], as regards syntax, semantics and terminology. They have been written so as to be independent of any particular implementation. The exercises in the first 3 lectures can be tackled without the use of a machine, although having access to an implementation will no doubt be beneficial. The project in Lecture 4 presupposes access to an implementation of the full language, including modules. (At present, the Edinburgh compiler does not fall into this category; the author used the New Jersey Standard ML compiler.)

Lecture 1 gives an introduction to ML aimed at the reader who is familiar with some programming language but does not know ML. Both the Core Language and the Modules are covered by way of example.

Lecture 2 discusses the use of ML modules in the development of large programs. A useful methodology for programming with functors, signatures and structures is presented.

Lecture 3 gives a fairly detailed account of the static semantics of ML modules, for those who really want to understand the crucial notions of sharing and signature matching.

Lecture 4 presents a one day project intended to give the student an opportunity of modifying a non-trivial piece of software using functors, signatures and structures.

[1] R. Harper, R. Milner and M. Tofte: "The Definition of Standard ML, Version 2", (ECS–LFCS–88–62) Labroatory for Foundations of Computer Science, Dept. of Computer Science, University of Edinburgh.