CS 591 S2–Formal Language Theory: Integrating Experimentation and Proof–Fall 2018

Instructor	Alley Stoughton
Email	`stough@bu.edu`
Personal Website	`alleystoughton.us`
Course Website	`alleystoughton.us/591-s2`
Class Sessions	TR 12:30-1:45pm CAS 324
Problem Solving Sessions	T 6:30-7:30pm MCS 144
Office Hours	TR 2:30-4pm, MCS 134
Course Piazza	`piazza.com/bu/fall2018/cs591s2`

1 Overview

The subject of formal language (automata) theory is traditionally taught as a paper-and-pencil mathematics course, even though it is largely concerned with algorithms on the subject's formalisms: regular expressions, automata, grammars and Turing machines. In contrast, this course approaches the subject in a way that balances proof with *experimentation*, carried out using the instructor's Forlan toolset, which implements the algorithms of formal language theory. Forlan is embedded in the functional programming language Standard ML (SML), a language whose notation and concepts are similar to those of mathematics. SML is strongly typed and interactive, properties that make sophisticated experimentation robust and enjoyable. The course features a rigorous approach to carrying out mathematical proofs, focusing on basic set theory, definitional techniques including various forms of recursion, and proof techniques including different kinds of induction.

Prerequisites: CS 112 (Introduction to Computer Science 2) and CS 131 (Combinatoric Structures), or equivalent.

2 Textbook

We will be using the Fall 2018 draft of my textbook on formal language theory, entitled *Formal Language Theory: Integrating Experimentation and Proof*. The book will be posted in installments, as the semester progresses (see `alleystoughton.us/forlan/book.pdf`). But, if you want to read ahead, you can also consult the September 2012 draft (see `alleystoughton.us/forlan/book2012.pdf`).

You may find it useful to also study the related sections of another book on formal language (automata) theory. Note, however, that the notation and definitions used by any particular book won't be identical to the ones we'll be using.

3   Forlan Toolset

Many of the algorithms that we will study are implemented as part of the Forlan toolset—a collection of tools for experimenting with formal languages that I've been developing since 2001. Forlan is implemented as a set of Standard ML (a functional programming language) modules. It's used interactively. In fact, a Forlan session is nothing more than an SML session in which the Forlan modules are available.

As the course progresses, you'll learn how to use Forlan, and you'll be using it when solving some of the problem sets. The best way to run Forlan or SML is as a sub-process of the Emacs text editor, using the SML mode for Emacs. We'll also be using JForlan, a Java program for creating and editing Forlan automata and trees.

See the Forlan website (alleystoughton.us/forlan/) for instructions for installing Forlan on personal computers. Forlan is also installed on the CS Department's Linux servers `csa1.bu.edu`, `csa2.bu.edu` and `csa3.bu.edu`, which may be accessed either remotely or from the Linux workstations in the CS Department's Computing Lab (Room 302 of 730 Commonwealth Avenue). See the course website for instructions.

4   Assessment

There will be seven problem sets, each worth 100 points, plus a final exam, worth 150 points. Problem set solutions will be submitted as hardcopies, either in class, at office hours, or via the course's CS Department drop box. But when a problem set involves writing Forlan code, that code must also be submitted via email to me (`stough@bu.edu`).

Each assessment unit will be graded using the following 100 point scale: A+ (100), A (92), A- (84), B+ (76), B (68), B- (60), C+ (52), C (44), C- (36), D+ (28), D (20), D- (12), F (0). (E.g., a grade of 80 points should be thought of as being halfway between a B+ and an A-.)

Late work will be assessed a penalty of 20% during the first twenty-four hours. Work that is more than twenty-four hours late will not be graded.

The elegance and simplicity of your work will be taken into account when it is assessed. I reserve the right to award extra credit to students who find errors in, or suggest improvements to, course materials.

A student's grade for the course will be the letter (possibly followed by a + or -) whose value is nearest to the weighted average of the grades of the student's problem sets and final exam. Ties will be resolved by selecting the better of the two grades.

5 Academic Integrity

You are responsible for reading and understanding BU's Academic Conduct Code (for undergraduates)

www.bu.edu/academics/policies/academic-conduct-code

or the GRS Academic and Professional Conduct Code (for graduate students)

www.bu.edu/cas/files/2017/02/GRS-Academic-Conduct-Code-Final.pdf

The work you hand in must be your own. You may discuss problem sets with others in general terms, but you may not base your work on other people's work, and you may not show your draft work to others. If you use resources other than course materials when working on the problem sets, you must cite your sources. (You may, of course, share your draft work with myself and any other course staff.)

Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC). If the ACC finds a student guilty, punishment could range from a minimum of a grading sanction (e.g., dropping your final grade by one letter grade) all the way up to suspension or expulsion from the University.

6 Email Communication, Problem Solving Sessions, Office Hours and Piazza

I will often communicate with the class by email, and so it is important that you read your email frequently. I'll be offering weekly optional help sessions, in which I'll solve problems on the blackboard and answer questions. They will be on Tuesdays from 6:30–7:30pm in MCS 144. My office hours will be on Tuesdays and Thursdays from 2:30–4pm in MCS 134. You can also make an appointment with me at an alternative time. We'll be using Piazza for online discussions. Visit piazza.com/bu/fall2018/cs591s2 to join in.

7 LaTeX Document Preparation System

When preparing solutions to assignments, you may wish to make use of the LaTeX document preparation system. LaTeX handles mathematics very well, and is widely used by academic computer scientists. See alleystoughton.us/latex for more information.