

CS 591 S2—Formal Language Theory: Integrating Experimentation and Proof

Instructor	Alley Stoughton
E-mail	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

Overview

- Formal language (automata) theory traditionally taught as paper-and-pencil mathematics course.
 - Largely concerned with algorithms on subject's formalisms: regular expressions, automata, grammars and Turing machines.

Overview

- Formal language (automata) theory traditionally taught as paper-and-pencil mathematics course.
 - Largely concerned with algorithms on subject's formalisms: regular expressions, automata, grammars and Turing machines.
- Course balances proof with *experimentation*, carried out using Forlan toolset:
 - Implements algorithms of formal language theory;
 - Embedded in functional programming language Standard ML.

Overview

- Formal language (automata) theory traditionally taught as paper-and-pencil mathematics course.
 - Largely concerned with algorithms on subject's formalisms: regular expressions, automata, grammars and Turing machines.
- Course balances proof with *experimentation*, carried out using Forlan toolset:
 - Implements algorithms of formal language theory;
 - Embedded in functional programming language Standard ML.
- Course features rigorous approach to carrying out mathematical proofs:
 - Basic set theory;
 - Definitional techniques including various forms of recursion;
 - Proof techniques including different kinds of induction.

Overview

- Formal language (automata) theory traditionally taught as paper-and-pencil mathematics course.
 - Largely concerned with algorithms on subject's formalisms: regular expressions, automata, grammars and Turing machines.
- Course balances proof with *experimentation*, carried out using Forlan toolset:
 - Implements algorithms of formal language theory;
 - Embedded in functional programming language Standard ML.
- Course features rigorous approach to carrying out mathematical proofs:
 - Basic set theory;
 - Definitional techniques including various forms of recursion;
 - Proof techniques including different kinds of induction.
- Prerequisites: CS 112 (Introduction to Computer Science 2) and CS 131 (Combinatoric Structures).

Textbook

- Using Fall 2018 draft of my textbook *Formal Language Theory: Integrating Experimentation and Proof*:
 - Book will be posted in installments, as the semester progresses;
 - If you want to read ahead, you can also consult September 2012 draft.

Textbook

- Using Fall 2018 draft of my textbook *Formal Language Theory: Integrating Experimentation and Proof*:
 - Book will be posted in installments, as the semester progresses;
 - If you want to read ahead, you can also consult September 2012 draft.
- You may find it useful to study related sections of another book on formal language (automata) theory.
 - Notation and definitions used by any particular book won't be identical to ones we'll be using.

Forlan Toolset

- Under development since 2001.

Forlan Toolset

- Under development since 2001.
- Implemented as set of Standard ML (SML) modules.
 - SML is strongly typed, functional language.

Forlan Toolset

- Under development since 2001.
- Implemented as set of Standard ML (SML) modules.
 - SML is strongly typed, functional language.
- Used interactively:
 - Forlan session is SML session with Forlan modules.

Forlan Toolset

- Under development since 2001.
- Implemented as set of Standard ML (SML) modules.
 - SML is strongly typed, functional language.
- Used interactively:
 - Forlan session is SML session with Forlan modules.
- As course progresses, you'll learn how to use Forlan:
 - You'll use it when solving problem sets;
 - Forlan can be run as a sub-process of Emacs text editor, using SML mode for Emacs;
 - We'll also be using JForlan, a Java program for creating and editing Forlan automata and trees.

Forlan Toolset

- Under development since 2001.
- Implemented as set of Standard ML (SML) modules.
 - SML is strongly typed, functional language.
- Used interactively:
 - Forlan session is SML session with Forlan modules.
- As course progresses, you'll learn how to use Forlan:
 - You'll use it when solving problem sets;
 - Forlan can be run as a sub-process of Emacs text editor, using SML mode for Emacs;
 - We'll also be using JForlan, a Java program for creating and editing Forlan automata and trees.
- Forlan can be installed in personal computers;
 - Instructions on Forlan website.

Forlan Toolset

- Under development since 2001.
- Implemented as set of Standard ML (SML) modules.
 - SML is strongly typed, functional language.
- Used interactively:
 - Forlan session is SML session with Forlan modules.
- As course progresses, you'll learn how to use Forlan:
 - You'll use it when solving problem sets;
 - Forlan can be run as a sub-process of Emacs text editor, using SML mode for Emacs;
 - We'll also be using JForlan, a Java program for creating and editing Forlan automata and trees.
- Forlan can be installed in personal computers;
 - Instructions on Forlan website.
- Forlan is installed on CS Department's Linux Servers;
 - See course website for more information.

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.
- Each assessment unit 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).

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.
- Each assessment unit 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).
- Late work assessed 20% penalty during first twenty-four hours;
 - Work more than twenty-four hours late not graded.

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.
- Each assessment unit 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).
- Late work assessed 20% penalty during first twenty-four hours;
 - Work more than twenty-four hours late not graded.
- Elegance and simplicity of work taken into account.

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.
- Each assessment unit 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).
- Late work assessed 20% penalty during first twenty-four hours;
 - Work more than twenty-four hours late not graded.
- Elegance and simplicity of work taken into account.
- Extra credit for finding errors in, or suggesting improvements to, course materials.

Assessment

- Seven problem sets, each worth 100 points, plus a final exam, worth 150 points.
- Each assessment unit 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).
- Late work assessed 20% penalty during first twenty-four hours;
 - Work more than twenty-four hours late not graded.
- Elegance and simplicity of work taken into account.
- Extra credit for finding errors in, or suggesting improvements to, course materials.
- Final grade 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.

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](http://www.bu.edu/cas/files/2017/02/GRS-Academic-Conduct-Code-Final.pdf)

Academic Integrity

- You must do your own work:
 - You may discuss problem sets with others in general terms;
 - But may *not* base your work on other people's work;
 - And may *not* share your work with others;
 - You must cite your sources.

Academic Integrity

- You must do your own work:
 - You may discuss problem sets with others in general terms;
 - But may *not* base your work on other people's work;
 - And may *not* share your work with others;
 - You must cite your sources.
- Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC);
 - Sanctions: drop in final course grade. . . suspension. . . expulsion.

Email Communication, Problem Solving Sessions, Office Hours and Piazza

- I will often communicate with the class by email.
- Offering weekly optional help sessions:
 - Tuesdays from 6:30–7:30pm in MCS 144.
- Office hours: Tuesdays and Thursdays from 2:30–4pm in MCS 134.
- Using Piazza for online discussions.

LaTeX Document Preparation System

- LaTeX document preparation system useful when preparing solutions to problem sets.
- LaTeX handles mathematics very well.
- Widely used by academic computer scientists.
- More information on course website.