CS 516—Software Foundations via Formal Languages —Spring 2025

Instructor	Alley Stoughton
E-mail	`stough@bu.edu`
Personal Website	`alleystoughton.us`
Course Website	`alleystoughton.us/cs516`
Class Sessions	Tues/Thurs 11am–12:15pm KCB 201
Problem Solving Sessions	Tues 6:30–7:45pm CAS 203
Office Hours	Tues 3-4pm, Thurs 1-2pm CDS 1013
Course Piazza	`piazza.com/bu/spring2025/cs516`

Overview

- This is an advanced undergraduate/beginning graduate-level course in formal language (automata) theory.

Overview

- This is an advanced undergraduate/beginning graduate-level course in formal language (automata) theory.
- Such a course is typically taught as a paper-and-pencil mathematics course.

Overview

- This is an advanced undergraduate/beginning graduate-level course in formal language (automata) theory.
- Such a course is typically taught as a paper-and-pencil mathematics course.
- But our course will balance proof with *experimentation*, carried out using the Forlan toolset.
  - Implemented in the functional programming language Standard ML (SML).

Overview

- 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

- 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.
- *Formal prerequisites*:
  - CAS CS 320 (Concepts of Programming Languages); and
  - CAS CS 330 (Introduction to Analysis of Algorithms)

Overview

- 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.
- *Formal prerequisites*:
    - CAS CS 320 (Concepts of Programming Languages); and
    - CAS CS 330 (Introduction to Analysis of Algorithms)
- *Informal prerequisites*:
    - Enough familiarity with concepts of programming languages so learning basic functional programming won't be too much of a stretch.
    - Intermediate level of mathematical maturity, competency using proof techniques like mathematical induction, and proof by contradiction.

Textbook

- We'll be using Spring 2025 draft of my textbook *Formal Language Theory: Integrating Experimentation and Proof*:
  - See `alleystoughton.us/forlan/book.pdf`.

Forlan Toolset

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

Forlan Toolset

- 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

- 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

- 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 on macOS, Linux and Windows.
  - Instructions on Forlan website: `alleystoughton.us/forlan`.

Assessment

- Seven problem sets, each worth 100 points, plus a course project *or* final exam, worth 200 points.

Assessment

- Seven problem sets, each worth 100 points, plus a course project *or* final exam, worth 200 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 course project *or* final exam, worth 200 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 course project *or* final exam, worth 200 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 course project *or* final exam, worth 200 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.
- Possible extra credit for finding errors in, or suggesting improvements to, course materials.

Assessment

- Seven problem sets, each worth 100 points, plus a course project *or* final exam, worth 200 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.
- Possible 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 course project/final exam.

Assessment

- When working on problem set, may optionally work in a pair.
  - Will then submit single solution, receive joint grade.
  - You and partner must both understand all of solution, or must note discrepancy—in which case grade may be adjusted.

Assessment

- When working on problem set, may optionally work in a pair.
  - Will then submit single solution, receive joint grade.
  - You and partner must both understand all of solution, or must note discrepancy—in which case grade may be adjusted.
- Apart from pair work, work must be your own.
  - May discuss problem sets with others in general terms.
  - May *not* base work on other's work.
  - May *not* show draft work to others.
  - Must cite your sources.

Pedagogical Approach

- Course learning mostly happens while working on problem sets, supported by lectures and problem solving sessions.
- Resist temptation to look for shortcuts on problem sets.
- You are *not* competing against each other for grades; everyone can earn an “A” with excellent work.
- Please focus on and engage with presentations, not your computers or phones.

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`

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`
- Incidents of academic misconduct will be reported to the Academic Conduct Committee (ACC).
  - Sanctions: drop in final course grade... suspension... expulsion.

Problem Solving Sessions, Office Hours, Piazza, Gradescope and GitHub

- Weekly problem solving sessions:
    - Tuesdays from 6:30–7:45pm in CAS 203.

Problem Solving Sessions, Office Hours, Piazza, Gradescope and GitHub

- Weekly problem solving sessions:
  - Tuesdays from 6:30–7:45pm in CAS 203.
- Office hours: Tuesdays 3-4pm, Thursdays 1-2pm in CDS 1013.
  - Or by appointment, in person or on Zoom.

Problem Solving Sessions, Office Hours, Piazza, Gradescope and GitHub

- Weekly problem solving sessions:
  - Tuesdays from 6:30–7:45pm in CAS 203.
- Office hours: Tuesdays 3-4pm, Thursdays 1-2pm in CDS 1013.
  - Or by appointment, in person or on Zoom.
- Using Piazza for online discussions.
  - Visit `piazza.com/bu/spring2025/cs516` to join in.

Problem Solving Sessions, Office Hours, Piazza, Gradescope and GitHub

- Weekly problem solving sessions:
  - Tuesdays from 6:30–7:45pm in CAS 203.
- Office hours: Tuesdays 3-4pm, Thursdays 1-2pm in CDS 1013.
  - Or by appointment, in person or on Zoom.
- Using Piazza for online discussions.
  - Visit `piazza.com/bu/spring2025/cs516` to join in.
- Assignment submission via Gradescope (`www.gradescope.com`).
  - I will post the entry code on Piazza.

Problem Solving Sessions, Office Hours, Piazza, Gradescope and GitHub

- Weekly problem solving sessions:
  - Tuesdays from 6:30–7:45pm in CAS 203.
- Office hours: Tuesdays 3-4pm, Thursdays 1-2pm in CDS 1013.
  - Or by appointment, in person or on Zoom.
- Using Piazza for online discussions.
  - Visit `piazza.com/bu/spring2025/cs516` to join in.
- Assignment submission via Gradescope (`www.gradescope.com`).
  - I will post the entry code on Piazza.
- Create private GitHub (`github.com`) repository for sharing Forlan/sml code.
  - Grant me (`alleystoughton`) access.

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.