Introduction

Lecture 1

Robb T. Koether

Hampden-Sydney College

Wed, Aug 24, 2016

1. The Syllabus
    - The Instructor
    - Introduction
    - Grading
    - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Outline

1. The Syllabus
    - The Instructor
    - Introduction
    - Grading
    - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

The Instructor

- Office: Bagby 114
- Office phone: 223-6207
- Home phone: 392-8604
- Office hours: 1:30 - 3:20 MTW, 1:30 - 3:20 R (usually); other hours by appointment.
- E-mail: rkoether@hsc.edu
- Web page: http://people.hsc.edu/faculty-staff/robbk

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Introduction

- The class meets in B 020 at 12:30 MWF.
- The text for the course is An Introduction to Formal Languages and Automata, 6th ed., by Peter Linz.

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Grading

- There will be homework assignments, several computer programs, three tests, and a final exam.
- In the final average, these will have the following weights:

Category	Weight
Computer programs	20%
Exercise sets	20%
Test average	40%
Final exam	20%

Homework Assignments

- Homework will be assigned regularly, beginning this Friday, and then selected problems will be collected approximately every two weeks.
- Most of the problems will be taken from the book, but some may be from other sources.

Computer Programs

- We will use the software JFLAP to build models of simple machines.
- This program accepts descriptions of
  - Determininistic Finite Automata
  - Non-determininistic Finite Automata
  - Pushdown Automata
  - Deterministic Pushdown Automata
  - Turing Machines

  and then they simulate that machine on the given input.

Tests

- There will be three tests, given on the following dates:

Test	Date
#1	Fri, Sep 23
#2	Fri, Oct 21
#3	Fri, Nov 18

Final Exam

- The final exam will be cumulative.
- It will be given on Friday, December 9 at 2:00 pm in Bagby 020,
- Or it will a take-home exam, to be turned in by 5:00 pm Friday, December 9.

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Attendance

- Attendance will be checked at the beginning of each class. If you arrive late, you will be counted absent. If that happens and you would like to be marked present, see me after class. Otherwise, late arrivals and absences will all count as absences. When assigning final grades, attendance will be taken into account.

Absences	Action
0 - 2	Grade bonus (1 “part” of a grade)
3 - 4	Neutral
5 - 6	Grade penalty (1 “part” of a grade)
> 6	Withdrawal (WF)

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Course Overview

- We will first study very simple machines called finite automata.
- These are basically pattern recognizers.
- An FA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.

Course Overview

- We will first study very simple machines called finite automata.
- These are basically pattern recognizers.
- An FA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
- For example, “**a**”, “**aaa**”, and “**abc**”.

Course Overview

- We will first study very simple machines called finite automata.
- These are basically pattern recognizers.
- An FA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
- For example, “**a**”, “**aaa**”, and “**abc**”.
- Also, “**aaabbcccc**” and “**abcabcabc**”.

Course Overview

- We will first study very simple machines called finite automata.
- These are basically pattern recognizers.
- An FA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
- For example, “**a**”, “**aaa**”, and “**abc**”.
- Also, “**aaabbcccc**” and “**abcabcabc**”.
- However, an FA cannot recognize patterns such as $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}$.

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. **Pushdown Automata**
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Course Overview

- Then we will first study somewhat more sophisticated machines called pushdown automata.
- These can recognize more complicated patterns.
- A PDA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
  - Nested recognizable patterns.

Course Overview

- Then we will first study somewhat more sophisticated machines called pushdown automata.
- These can recognize more complicated patterns.
- A PDA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
  - Nested recognizable patterns.
- Including everything that a FA can recognize.

Course Overview

- Then we will first study somewhat more sophisticated machines called pushdown automata.
- These can recognize more complicated patterns.
- A PDA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
  - Nested recognizable patterns.
- Including everything that a FA can recognize.
- Also, patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}$

Course Overview

- Then we will first study somewhat more sophisticated machines called pushdown automata.
- These can recognize more complicated patterns.
- A PDA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
  - Nested recognizable patterns.
- Including everything that a FA can recognize.
- Also, patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}$
  - And $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{m}\underbrace{\mathbf{dd}\cdots\mathbf{d}}_{m}$

Course Overview

- Then we will first study somewhat more sophisticated machines called pushdown automata.
- These can recognize more complicated patterns.
- A PDA can recognize
  - Single letters.
  - Repetitions of recognizable patterns.
  - Sequential blocks of recognizable patterns.
  - Nested recognizable patterns.
- Including everything that a FA can recognize.
- Also, patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}$
  - And $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{m}\underbrace{\mathbf{dd}\cdots\mathbf{d}}_{m}$
  - And $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{m}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{m}\underbrace{\mathbf{dd}\cdots\mathbf{d}}_{n}$

Course Overview

- However, an PDA cannot recognize patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{n}$

Course Overview

- However, an PDA cannot recognize patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{n}$
  - Or $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{m}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{n}\underbrace{\mathbf{dd}\cdots\mathbf{d}}_{m}$

Course Overview

- However, an PDA cannot recognize patterns such as
  - $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{n}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{n}$
  - Or $\underbrace{\mathbf{aa}\cdots\mathbf{a}}_{n}\underbrace{\mathbf{bb}\cdots\mathbf{b}}_{m}\underbrace{\mathbf{cc}\cdots\mathbf{c}}_{n}\underbrace{\mathbf{dd}\cdots\mathbf{d}}_{m}$
  - Or $\underbrace{\mathbf{abccac}}_{\text{string}}\underbrace{\mathbf{abccac}}_{\text{copy}}$

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Course Overview

- The third (and last) type of machine we study is called a Turing machine.
- A Turing machine can do more than simply accept or reject its input.
- A Turing machine can compute anything that is “computable” by any computer.
- Indeed, a Turing machine is equivalent to a computer program.

Course Overview

- The third (and last) type of machine we study is called a Turing machine.
- A Turing machine can do more than simply accept or reject its input.
- A Turing machine can compute anything that is “computable” by any computer.
- Indeed, a Turing machine is equivalent to a computer program.
- Is there anything that a Turing machine cannot compute?

Outline

1. The Syllabus
    - The Instructor
    - Introduction
    - Grading
    - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

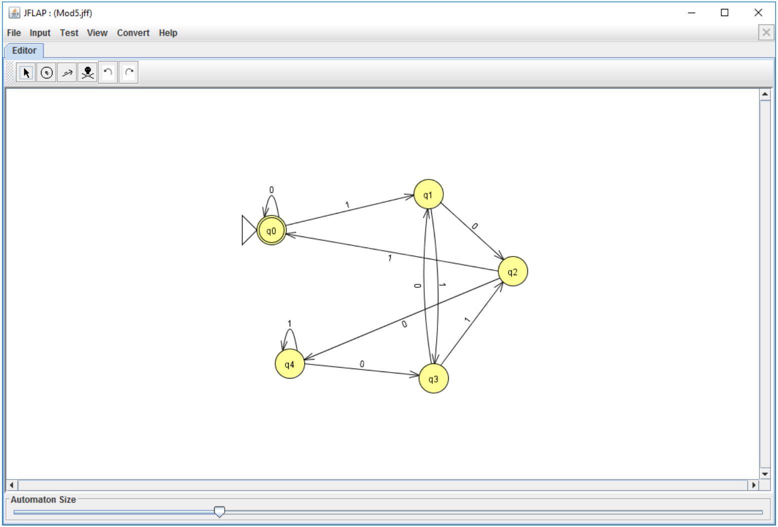
Course Overview

- Finally, we distinguish those problems that can be solved efficiently by a Turing machine from those that cannot be solved efficiently.
- For example,
  - Sorting a list of 1,000,000,000,000 names is feasible.
  - Finding the shortest circuit (round trip) through 25 cities is not feasible. (Aproximately 620,448,401,700,000,000,000,000 possibilities.)

Outline

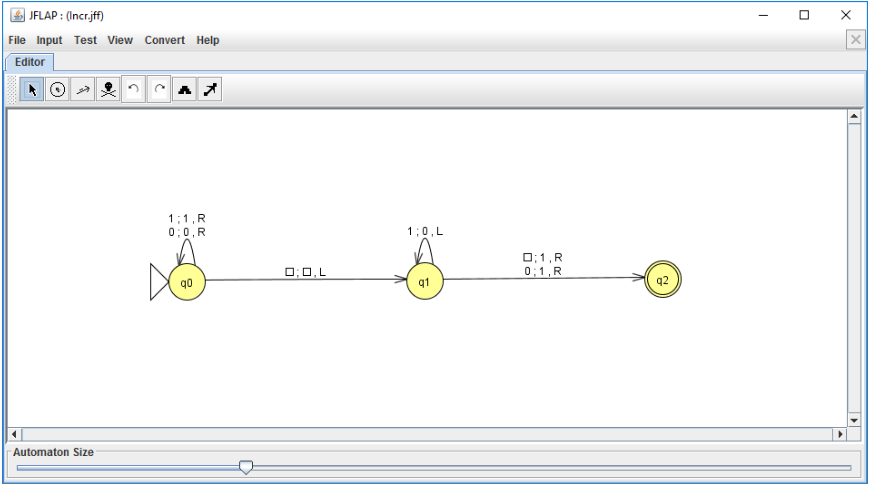
1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Example of a DFA



A DFA that will determine whether a binary number is a multiple of 5

Example of a Turing Machine



A Turing machine that will increment a binary number

Outline

1. The Syllabus
   - The Instructor
   - Introduction
   - Grading
   - Attendance
2. Finite Automata
3. Pushdown Automata
4. Turing Machines
5. Efficiency
6. Examples
7. Assignment

Assignment

- Assignment
  - Read Section 1.1.