

CS 6041

Theory of Computation

Overview

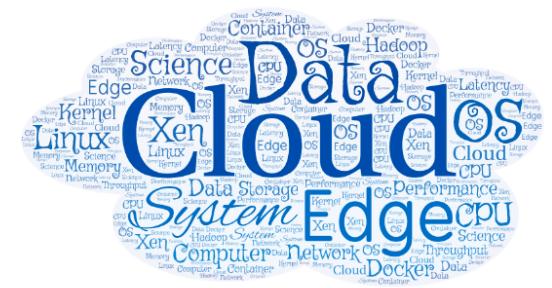
Kun Suo

Computer Science, Kennesaw State University

<https://kevinsuo.github.io/>

Self Introduction

- Kun Suo, Ph.D.
 - Homepage, <https://kevinsuo.github.io/>
 - Research interests:
 - Cloud computing and virtualization;
 - Operating systems, containers and kubernetes;
 - Software defined network (SDN) and network function virtualization (NFV)
 - Big data systems and machine learning systems
 - Projects you may be interested in:
 - Several projects in Cloud & Data & Edge
 - <https://kevinsuo.github.io/code-lab.html>



Now it's your turn

- Name, program/year, where from
- Your interests in Computer Science
- What do you expect in the Theory of Computation?

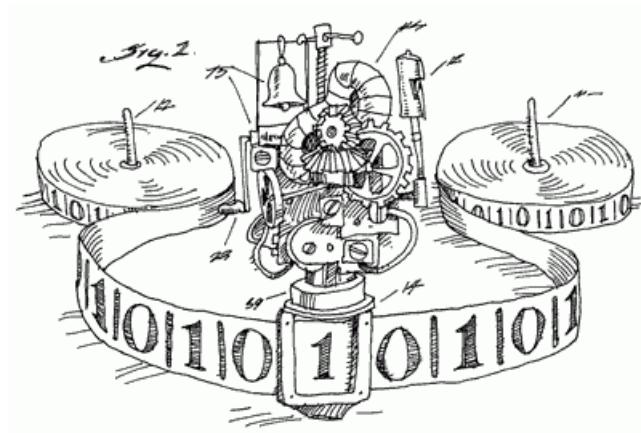
<https://www2.eecs.berkeley.edu/Research/Areas/CS/>

If you are in the online course, introduce yourself in D2L,
Discussions → Self-Introduction



What does Theory of Computation include?

- A study of topics from theoretical computer science that includes
 - Automata and languages
 - Computability theory
 - Complexity theory
- How efficiently (how fast, how much space, etc.) problems can be solved on a model of computation, using an algorithm



What does Theory of Computation include?

- Example 1: $L = \{s \mid \text{Binary strings end in } 1s\}$.
 - Is it computable?
 - How can human beings solve it?
 - Can you build a machine to solve it?

0101010100011

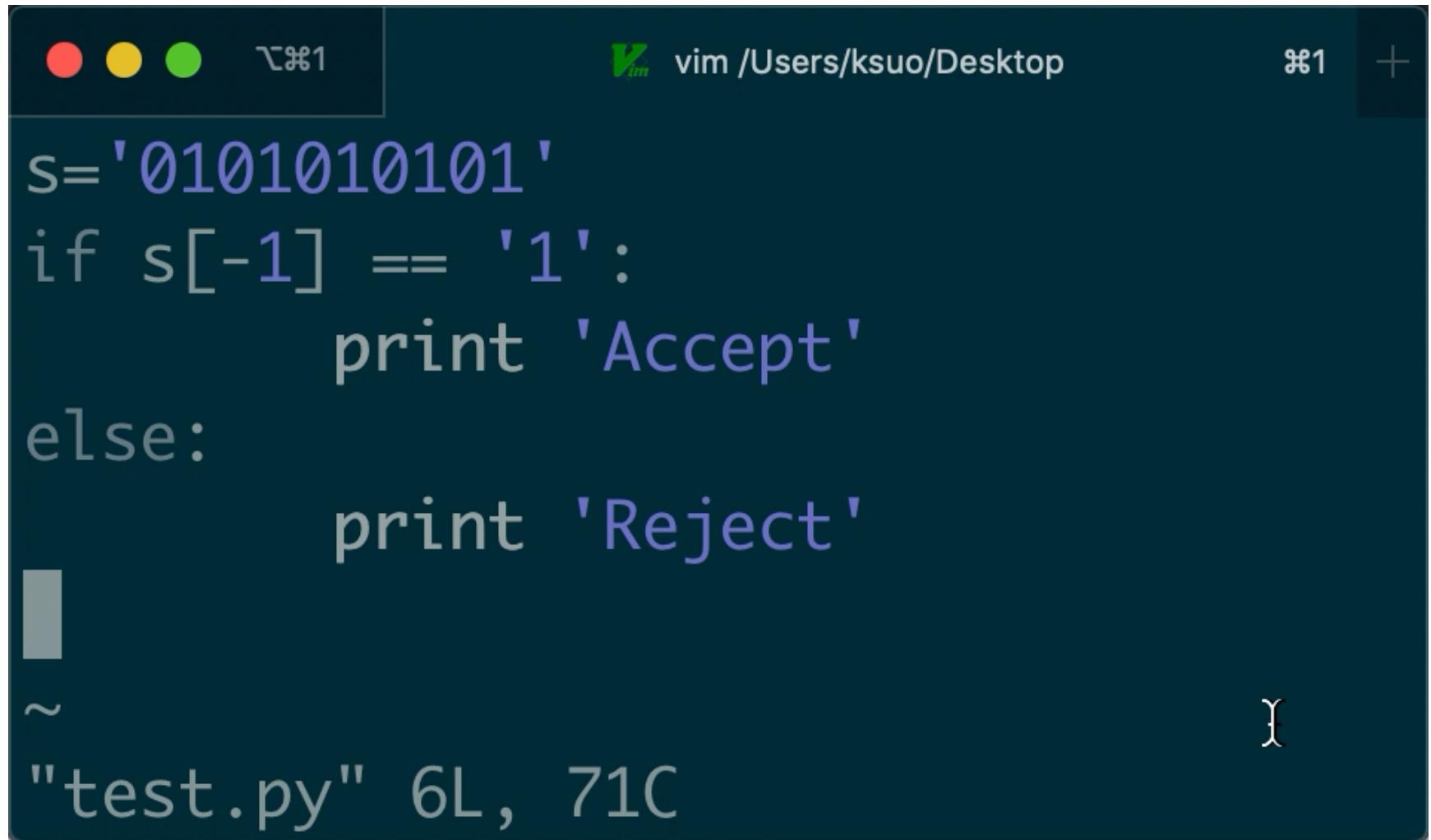
Accept

010100010

Reject



What does Theory of Computation include?



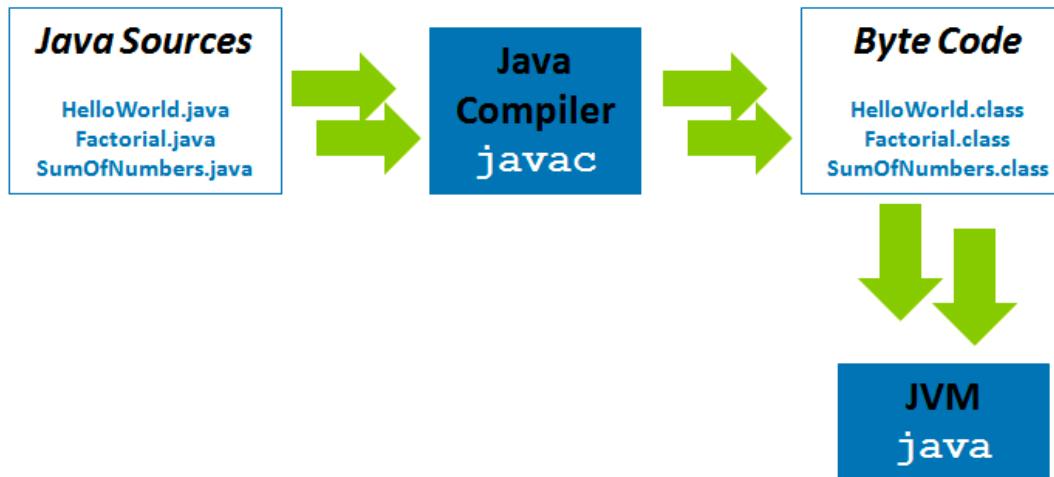
A screenshot of a terminal window titled 'vim /Users/ksuo/Desktop'. The code inside the terminal is as follows:

```
s='0101010101'
if s[-1] == '1':
    print 'Accept'
else:
    print 'Reject'
```

The terminal shows the file path '/Users/ksuo/Desktop' and the status 'vim'. The bottom of the terminal displays the file name "test.py" and its status "6L, 71C".

What does Theory of Computation include?

- Example 2: Accepts all valid Java code.
 - Is it computable?
 - Can you build a machine to solve it?
 - How can human beings solve it?



```
Output - compiler (run) X
run:
Success: true
java.lang.ClassNotFoundException: HelloWorld
    at java.net.URLClassLoader$1.run(URLClassLoader.java:372)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
    at java.security.AccessController.doPrivileged(Native Method)
    at java.net.URLClassLoader.findClass(URLClassLoader.java:360)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
    at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
    at java.lang.Class.forName0(Native Method)
    at java.lang.Class.forName(Class.java:260)
    at CompileSourceInMemory.main(CompileSourceInMemory.java:50)
BUILD SUCCESSFUL (total time: 2 seconds)
```



What does Theory of Computation include?

- Example 3 (halting problem): for input w , determine whether w will end in finite time or infinite loop.
 - Is it computable?
 - Can you build a machine to solve it?
 - How can human beings solve it?

There is no program that solves the halting problem.

This proof was found by [Alan Turing](#) in 1936.



https://scholar.google.com/scholar?hl=en&as_sdt=0%2C11&q=ON+COMPUTABLE+NUMBERS%2C+WITH+AN+APPLICATION+TO+THE+ENTSCHEIDUNGSPROBLEM&btnG=



What does Theory of Computation include?

- Example 1: Binary strings end in 0s.
- Example 2: Accepts all valid Java code.
- Example 3 (halting problem): for input w , determine whether w will end in finite time or infinite loop.

Easy vs Hard vs Not computable

Simple machine vs Complicated machine (with knowledge)

Fast vs Slow vs Not answer (proof)

1KB vs 500MB vs ...



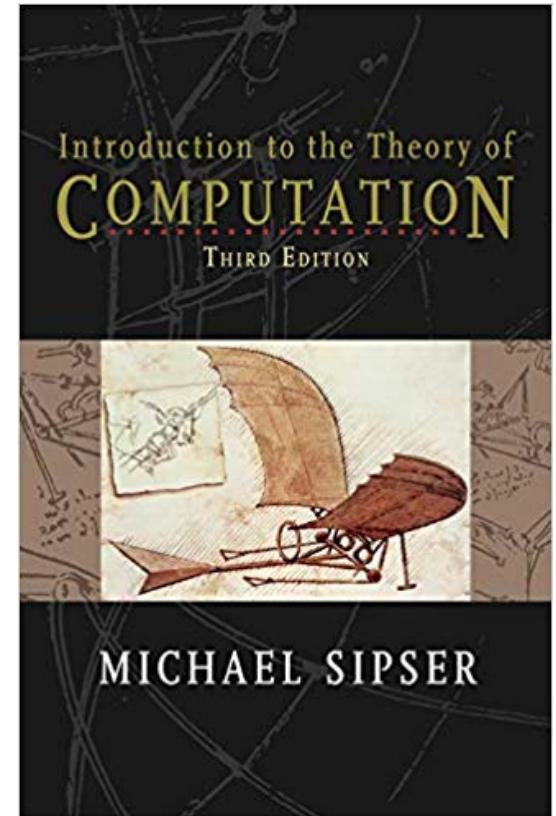
Course Information

- Instructor: Dr. Kun Suo
- Office: J-318
- Email: ksuo@kennesaw.edu
 - Only reply to e-mails that are sent from KSU student email accounts and title the course number [CS6041]
- Class Hours: T/R 5:00PM-6:15PM, J-217
- Office Hours:
 - Tuesday/Thursday, 2:00pm-3:00pm
 - By appointment
- Course Materials
 - Homework assignments, lecture slides, and other materials will be posted in the webpage (<https://kevinsuo.github.io/teaching/6041/class.html>) and D2L.
 - All lectures will be recorded.



Reference Book

- Professor Michael Sipser (MIT)
- Introduction to the Theory of Computation
- 3rd edition, Cengage Learning
ISBN 13-978-1-133-18779-0
- Buy/Rent/eBook/Library



Grading Policy

- Grading percentage
 - In-class discussion and attendance: 5%
 - Homework (7x): 35% (please submit **pdf file** on D2L)
 - 1st Midterm: 15%
 - 2nd Midterm: 15%
 - 3rd Final exam: 30%

*Late submission is **not accepted**.*



Course Policy

- Grading scale

Percentage	Grade
90 - 100	A
80 - 89	B
70 - 79	C
60 - 69	D
Below 60	F



Academic Integrity

- Academic dishonesty
 - Cheating
 - Plagiarism
 - Collusion
 - The submission for credit of any work or materials that are attributable in whole or in part to another person
 - Taking an examination for another person
 - Any act designed to give unfair advantage to a student or the attempt to commit



How to succeed this class

- THINK hard, not WORK hard
- Scientific Thinking
- Passion to learn something NEW
- ASK ME (office hours) / Friends / Classmates questions
- Begin homework assignments EARLY



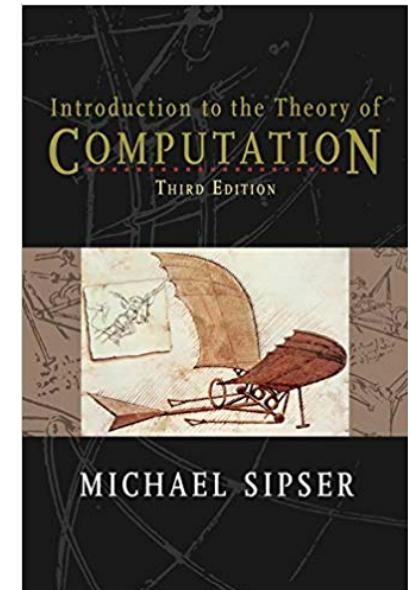
Where to go for help ?

- Ask questions in class
- Ask questions outside class
 - Classmates and friends
- Attend office hours
 - Dr. Kun Suo: Tuesday/Thursday 4:00PM – 5:00PM, J-318
- Search on the web
 - Stand on the shoulder of giants



Course Content Overview

- 0. Introduction
- 1. Regular language
- 2. Context-free language
- 3. Turing machine
- 4. Decidability
- 5. Reducibility
- 6. Advanced Topics in Computability Theory
- 7. Time complexity
- 8. Space complexity
- 9. Intractability
- 10. Advanced topics in complexity theory



Course Content Overview

- Chapter 0 Introduction
- Chapter 1 Regular Languages
- Chapter 2 Context-Free Languages
- Chapter 3 The Church-Turing Thesis
- Chapter 4 Decidability
- Chapter 5 Reducibility
- Chapter 7 Time Complexity



Course Content Overview

- Chapter 0 Introduction
 - Chapter 1 Regular Languages
 - Chapter 2 Context-Free Languages
 - Chapter 3 The Church-Turing Thesis
 - Chapter 4 Decidability
 - Chapter 5 Reducibility
 - Chapter 7 Time Complexity
-
- The diagram illustrates the course content overview. A vertical list of chapters is on the left, and three blue curly braces on the right group them into three sections corresponding to Exam 1, Exam 2, and Exam 3. The first two chapters are grouped under Exam 1, the next three chapters under Exam 2, and the last two chapters under Exam 3.

Course Content Overview

Week/Date	Topic	Chapters	Assignment
1	Introduction and overview	0	HW1
2	Deterministic finite automata	1.1	
3	Nondeterministic finite automata	1.2	HW2
4	Regular expression and Regular language	1.3	
5	Non-regular language	1.4	HW3
6	Exam 1		
7	Context free language	2.1	HW4
8	Pushdown Automata	2.2	
9	Non-Context free language	2.3	
10	Exam 2		
11	Turing machine	3	HW5
12	Decidability	4	
13	Reducibility	5	HW 6
14	Complexity and NP-completeness	7	HW 7
15	Conclusion		
16	Exam 3		



Language is the foundation of computation

```
package rentalStore;
import java.util.Enumeration;
import java.util.Vector;

class Customer {
    private String _name;
    private Vector<Rental> _rentals = new Vector<Rental>();

    public Customer(String name) {
        _name = name;
    }
    public String getMovie(Movie movie) {
        Rental rental = new Rental(new Movie("", Movie.NEW_RELEASE), 10);
        Movie m = rental._movie;
        return movie.getTitle();
    }
    public void addRental(Rental arg) {
        _rentals.addElement(arg);
    }
    public String getName() {
        return _name;
    }
}
```

Java source code

Read by people

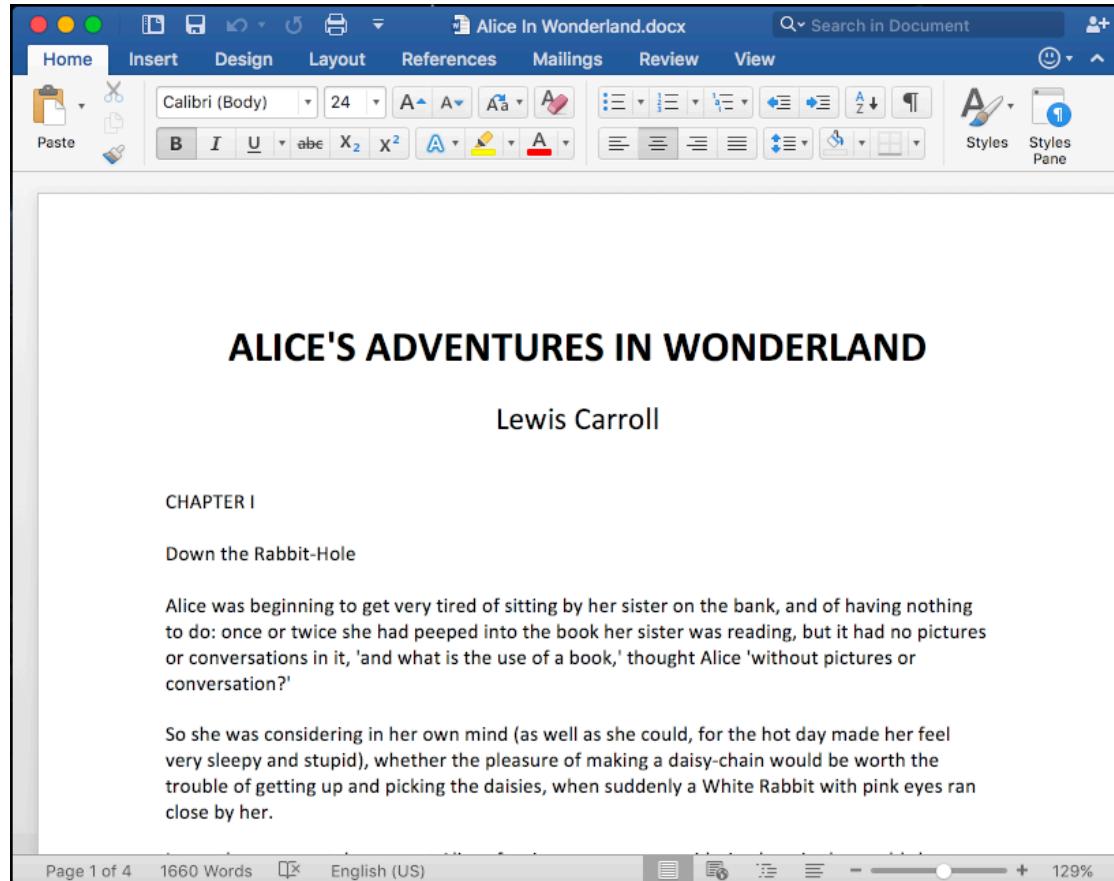


Binary code

Read by machines



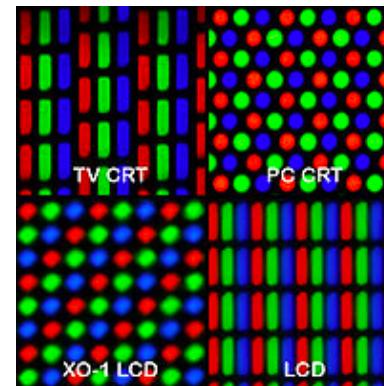
Language is the foundation of computation



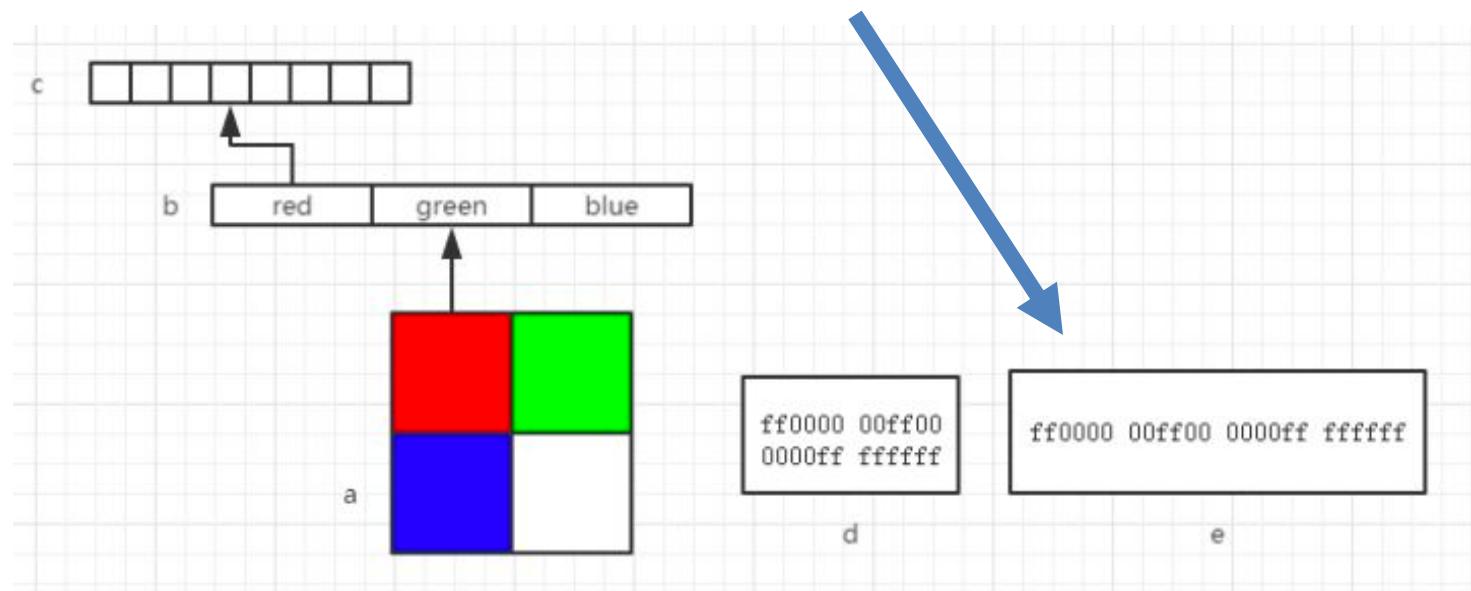
A word document



Language is the foundation of computation



A figure



Language universe in a big picture

- Are the languages all the same?



Language universe in a big picture

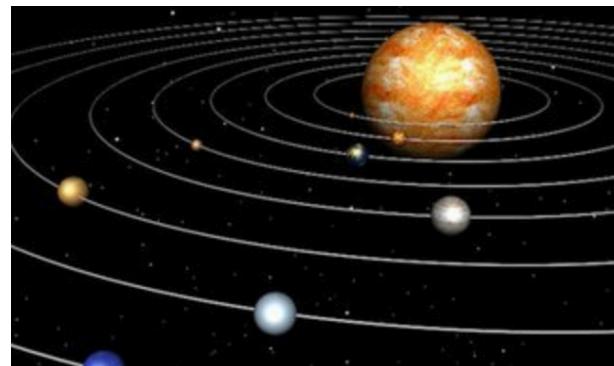
- Are the languages all the same?



Satellite << Planet << Star << ...



Earth

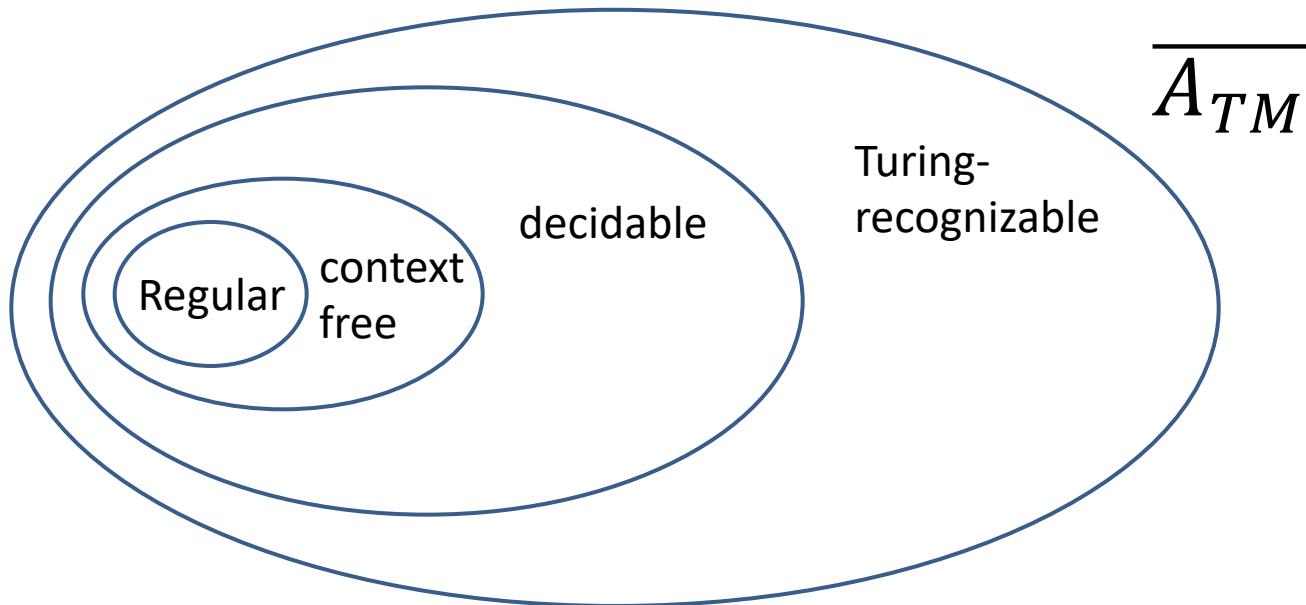


Solar system



Galaxy

Language universe in a big picture



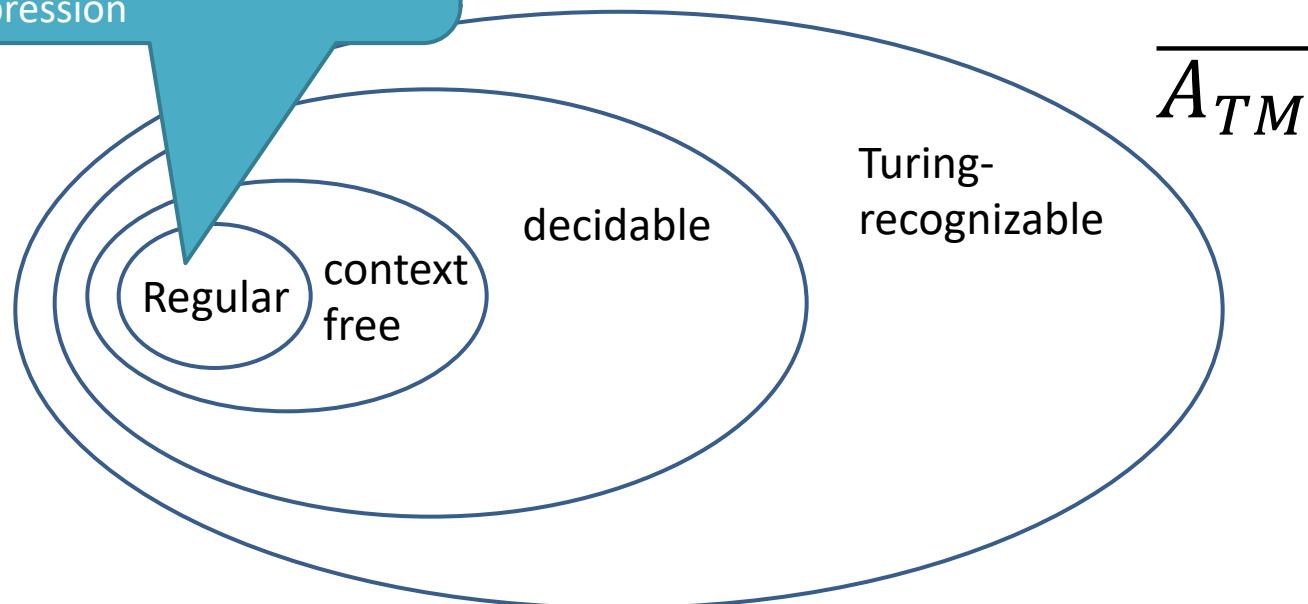
Language universe in a big picture

Regular language \Leftrightarrow

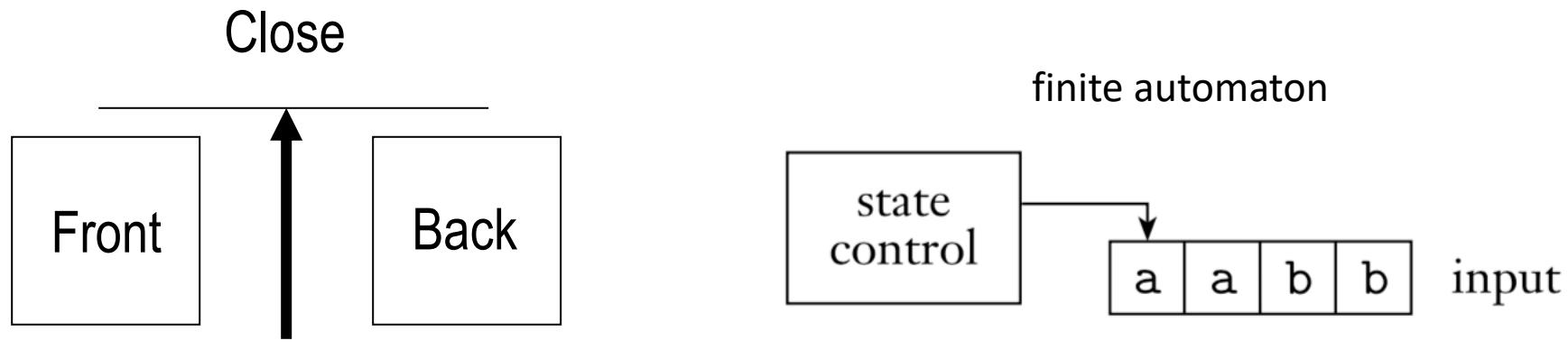
DFA: deterministic finite automata

NFA: non-deterministic finite automata

RE: regular expression



Language universe in a big picture



DFA example:
Automatic Door



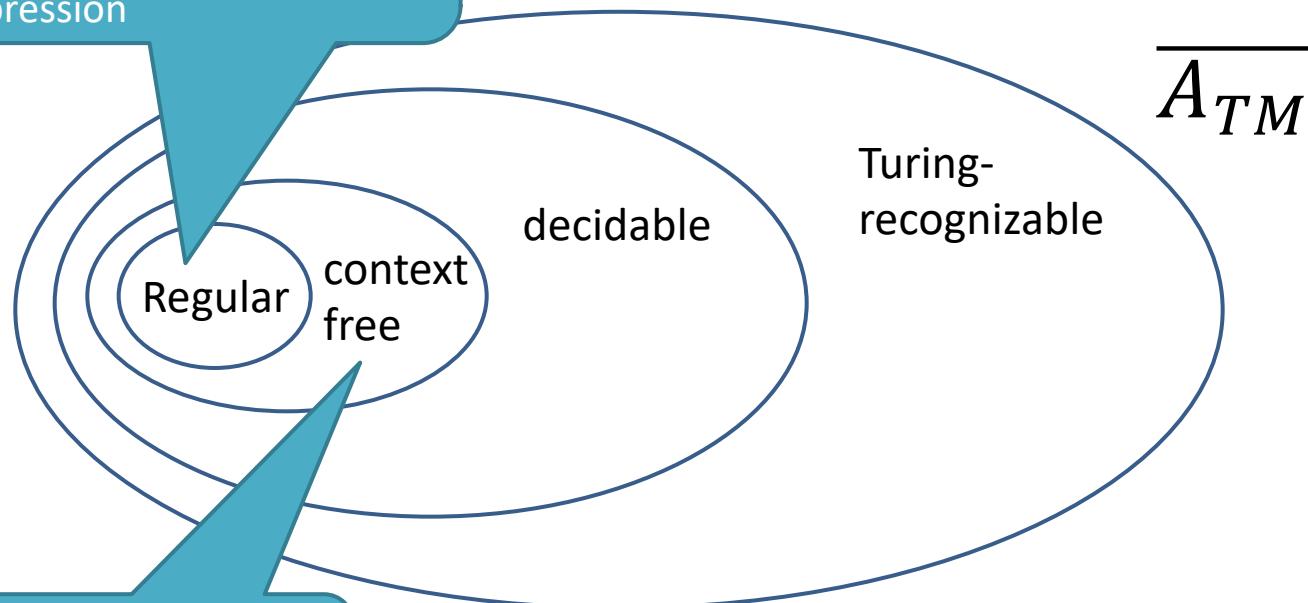
Language universe in a big picture

Regular language \Leftrightarrow

DFA: deterministic finite automata

NFA: non-deterministic finite automata

RE: regular expression



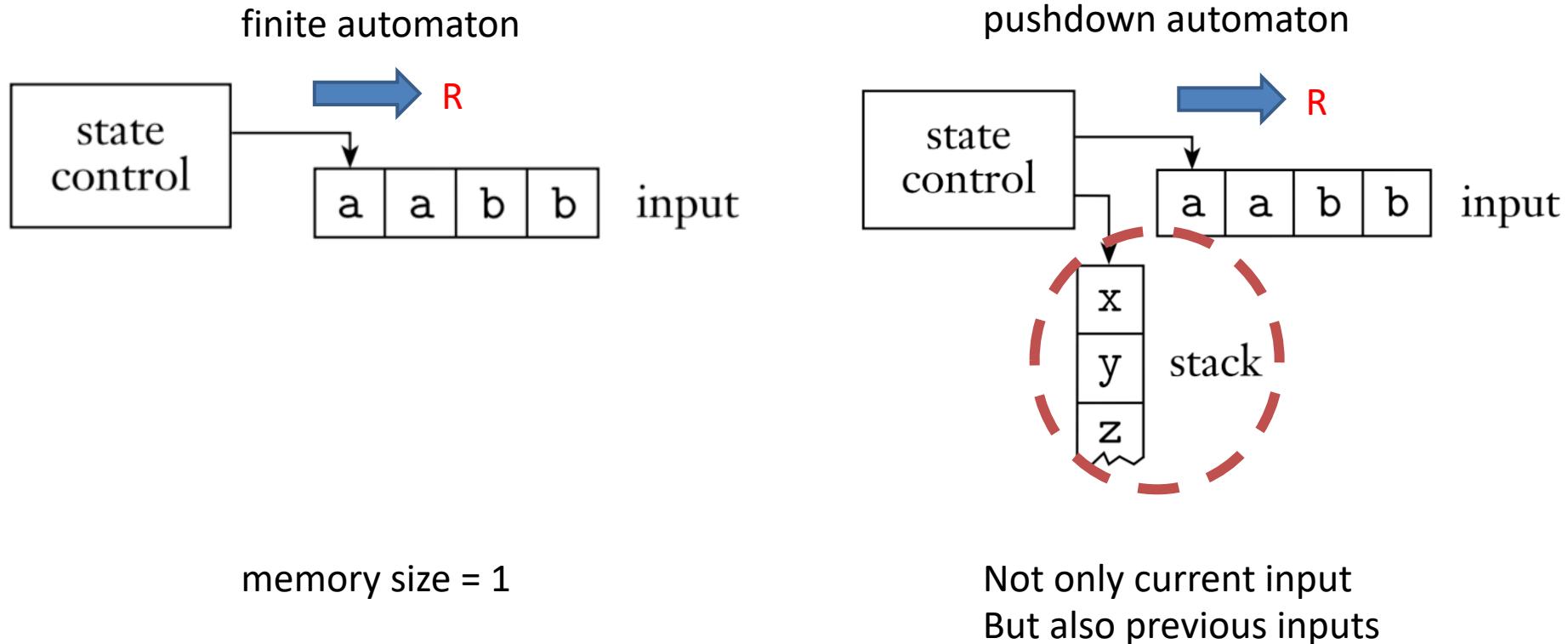
CFL: context free language

CFG: context free grammar

PDA: push down automata



Language universe in a big picture



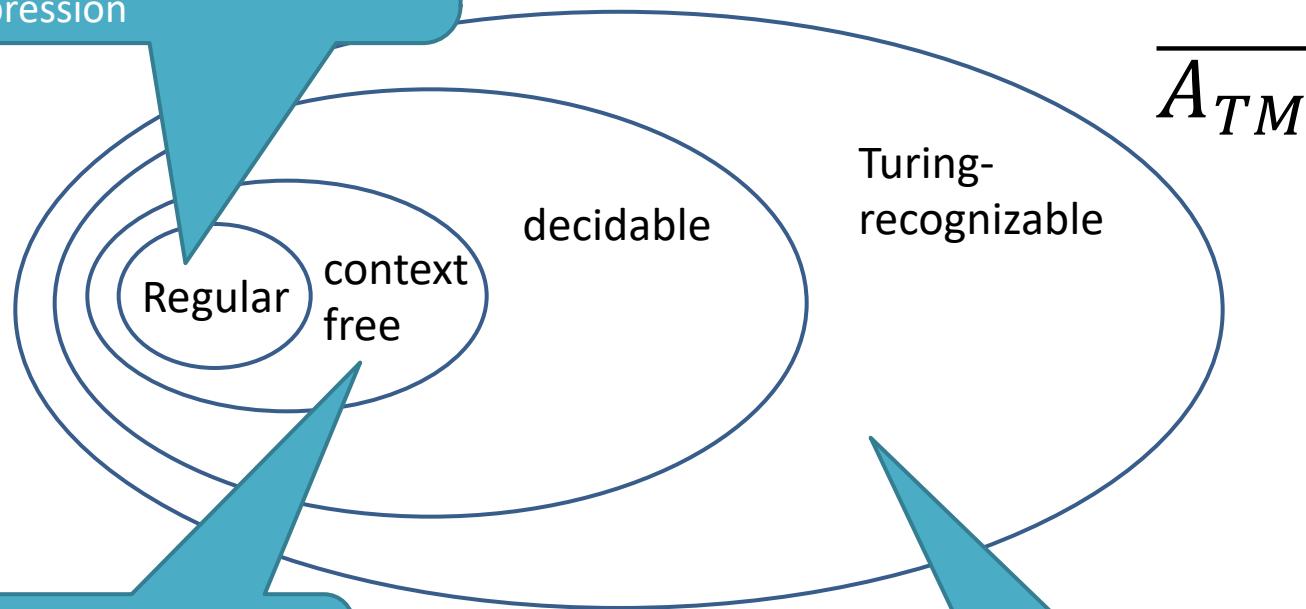
Language universe in a big picture

Regular language \Leftrightarrow

DFA: deterministic finite automata

NFA: non-deterministic finite automata

RE: regular expression



CFL: context free language

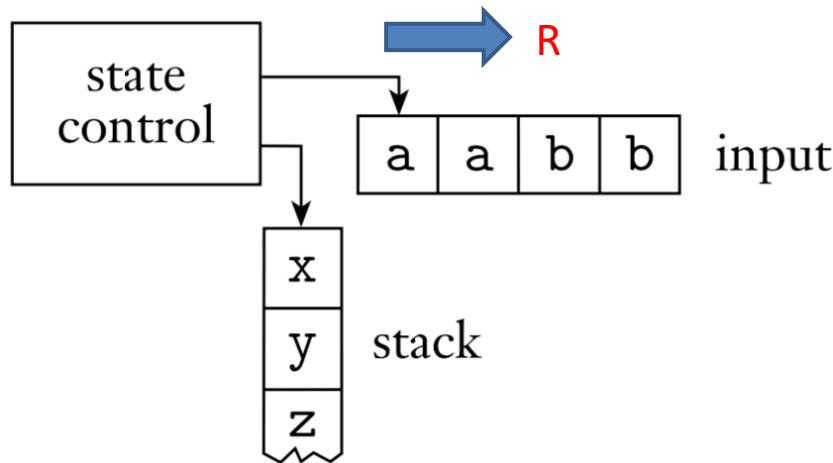
CFG: context free grammar

PDA: push down automata

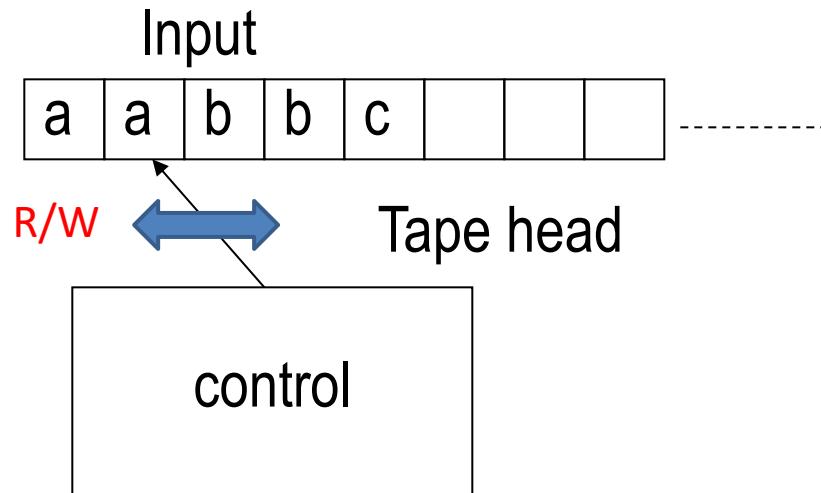
TM: turing machine

Language universe in a big picture

pushdown automaton

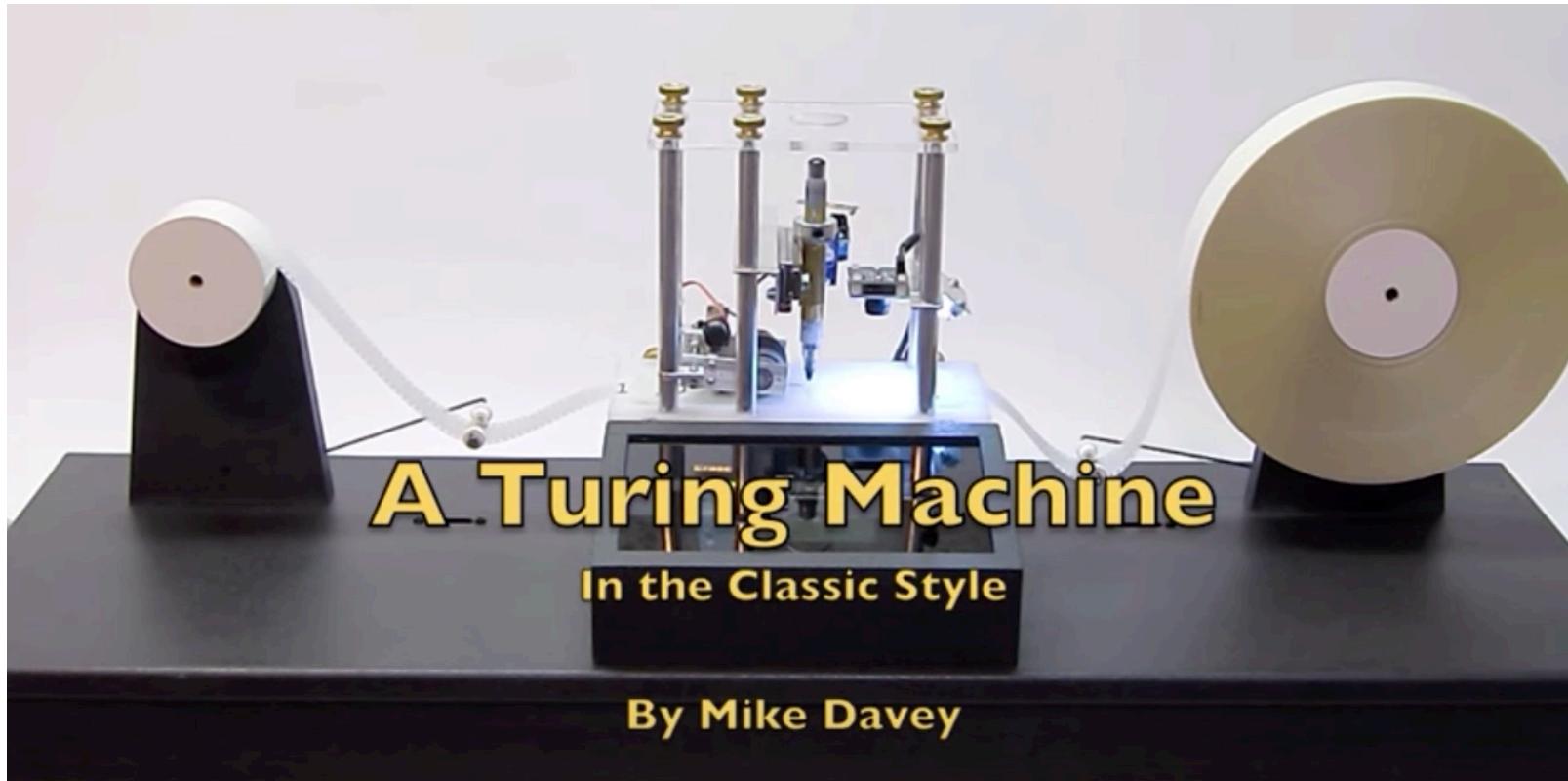


Turing machine



Turing machines are equivalent to modern electronic computers at a certain theoretical level, but differ in details.

Language universe in a big picture



<https://www.youtube.com/watch?v=E3keLeMwfHY>

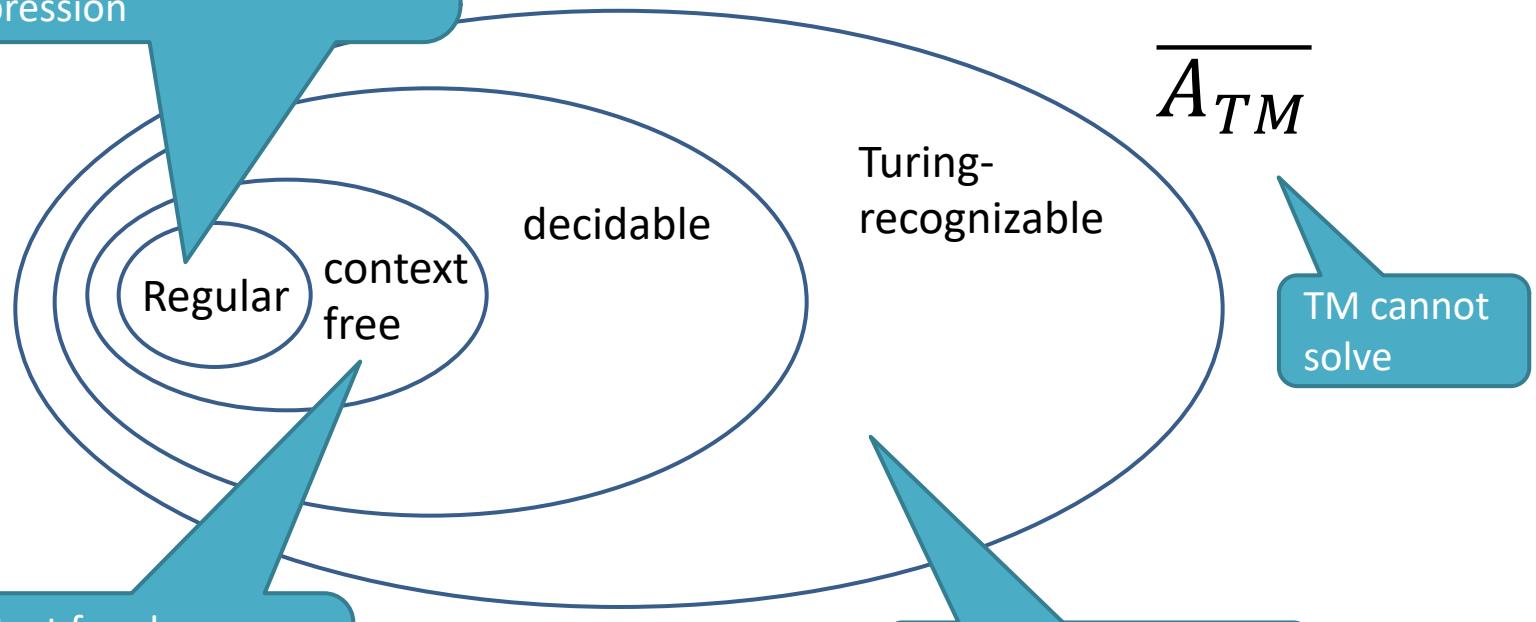
Language universe in a big picture

Regular language

DFA: deterministic finite automata

NFA: non-deterministic finite automata

RE: regular expression



CFL: context free language

CFG: context free grammar

PDA: push down automata

Language universe in a big picture

- P problem:
 - The general class of questions for which some algorithm can provide an answer in polynomial time
 - Give the answer!
- NP problem:
 - The class of questions for which an answer can be verified in polynomial time
 - Test the answer!



Language universe in a big picture

- P problem:
 - Give the answer!
 - E.g., Sorting for n numbers, no longer than $O(n^2)$

- NP problem:
 - Test the answer!
 - E.g., whether a given set has a subset that the sum of all its element is 0.
 - $s = \{-1, 3, 2, -5, 6\}$. subset = {3, 2, -5}, test in $O(n)$ time



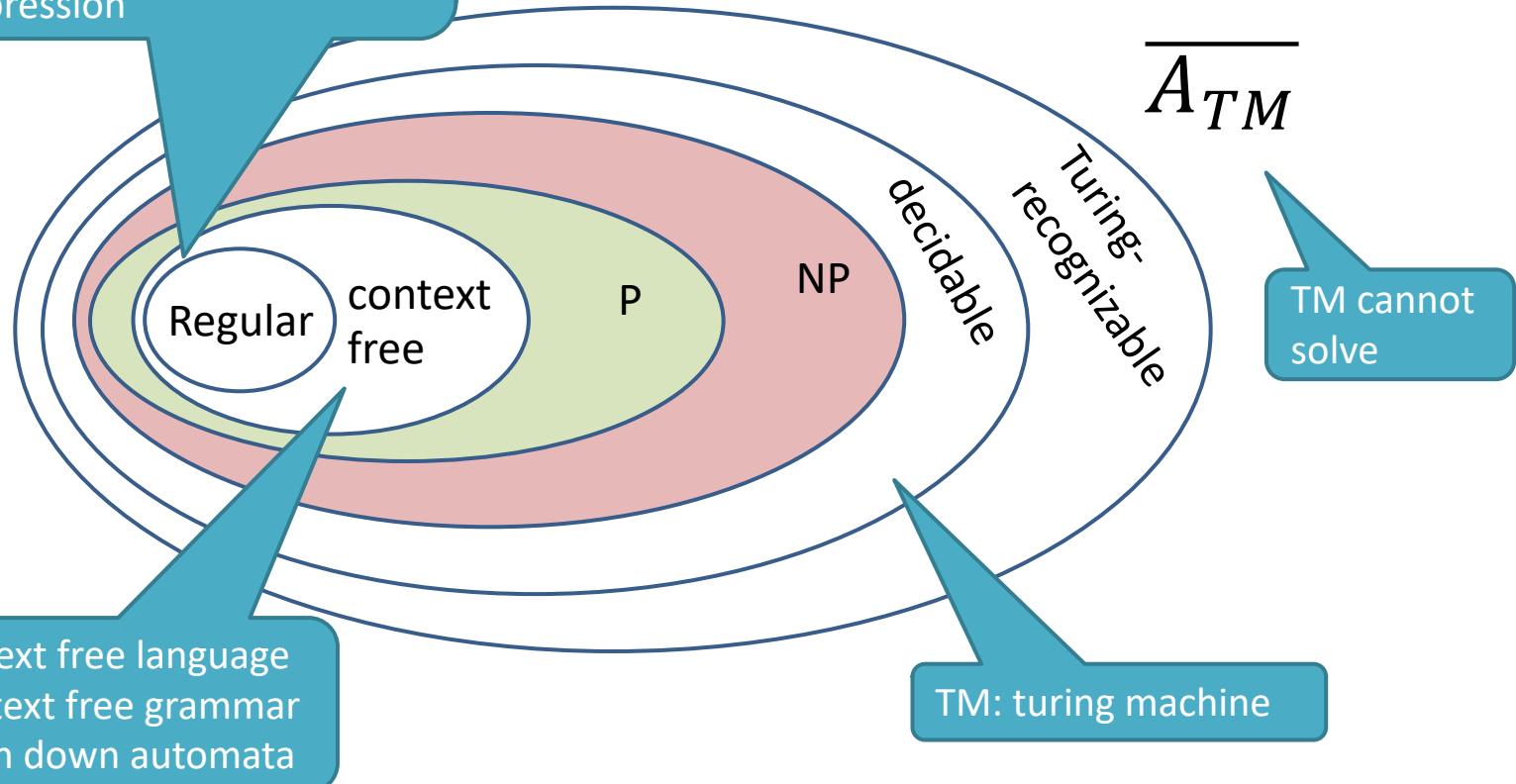
Language universe in a big picture

Regular language

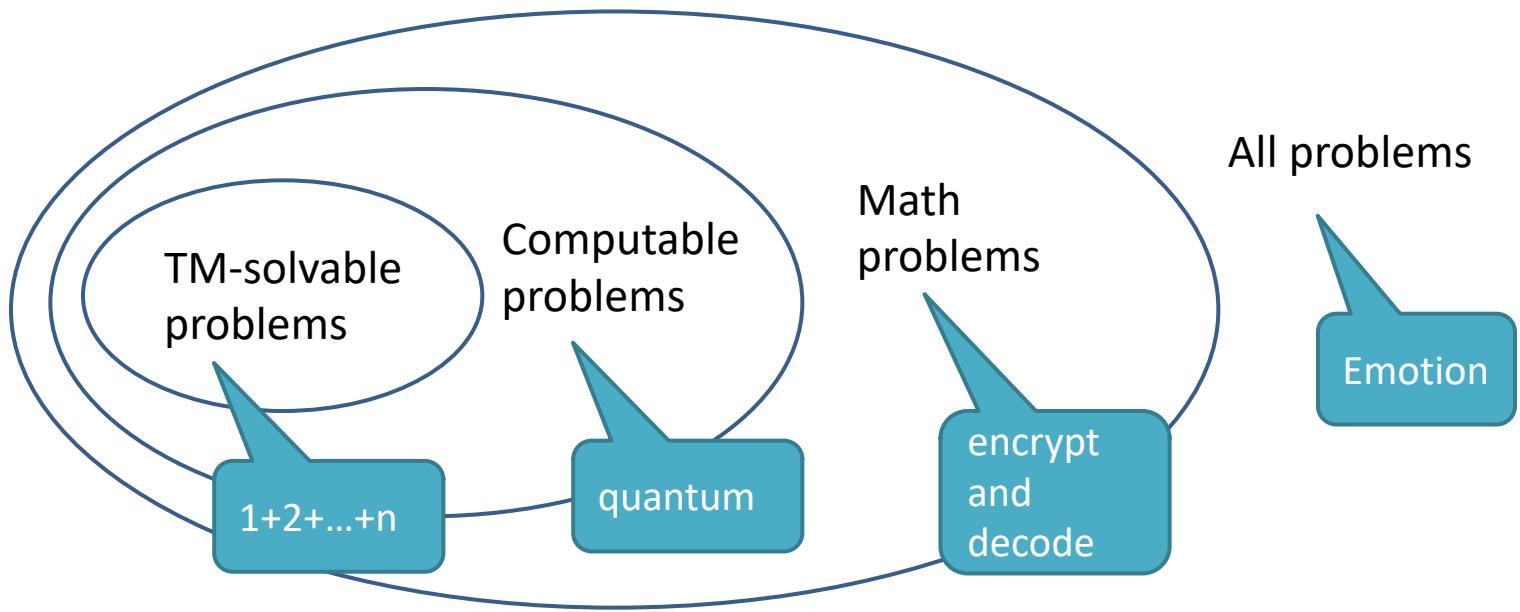
DFA: deterministic finite automata

NFA: non-deterministic finite automata

RE: regular expression



Problems



- Only a small proportion of problems can be solved by Turing machines in real life



Conclusion

- Introduction of course
 - Information, book, grading, ...
- Course content overview
 - Calendar, content, schedule, ...
- Language universe in a big picture
 - Regular language
 - CFL
 - Turing machine
 - P vs. NP

