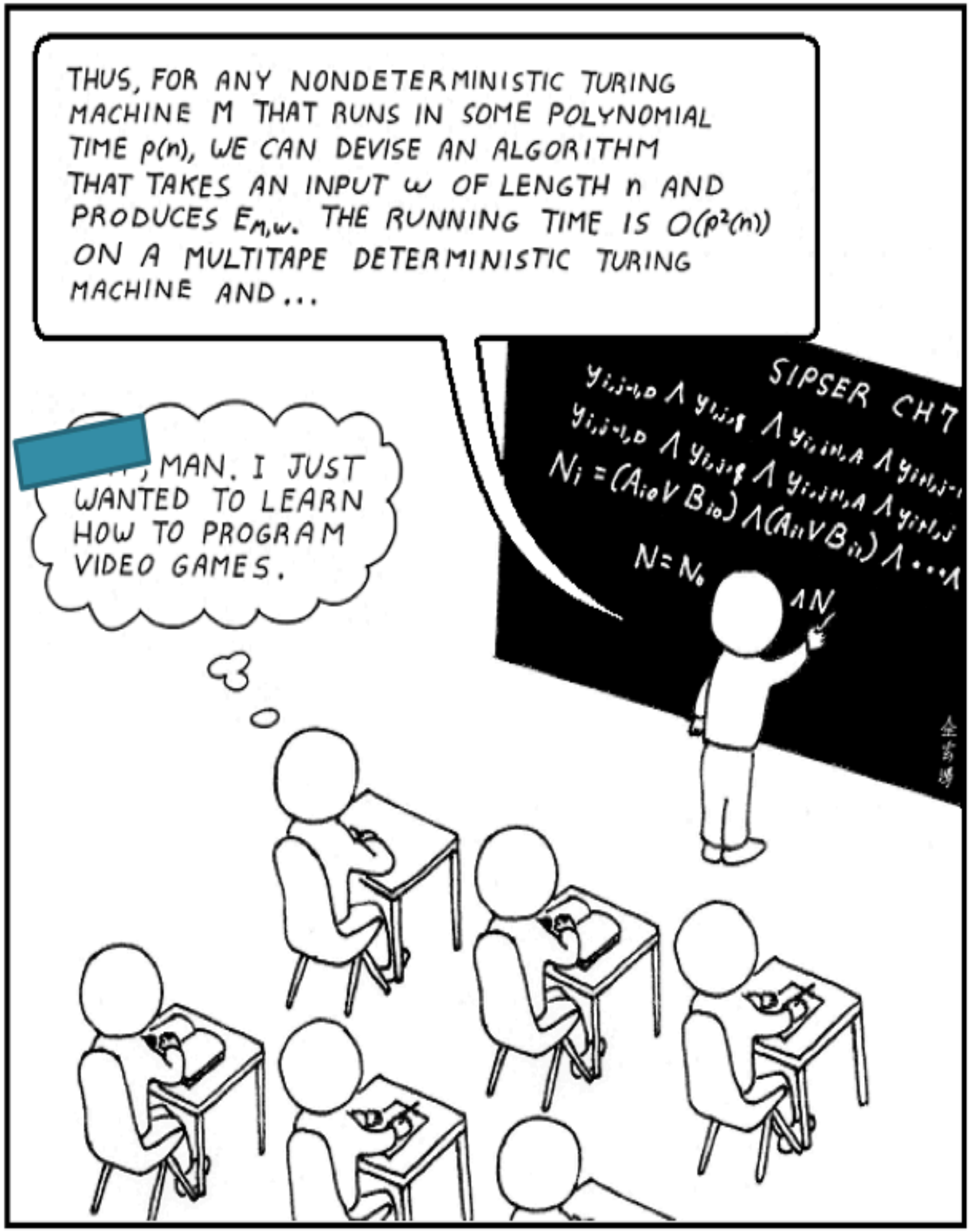
CSE 355: Introduction to Theoretical Computer Science

Instructor:
Dr. Yu ("Tony") Zhang

Lecture:
WGHL101, Tue/Thu, 3:00—4:15 PM

Office Hours:
BYENG 594, Tue/Thu, 5:00—6:00PM

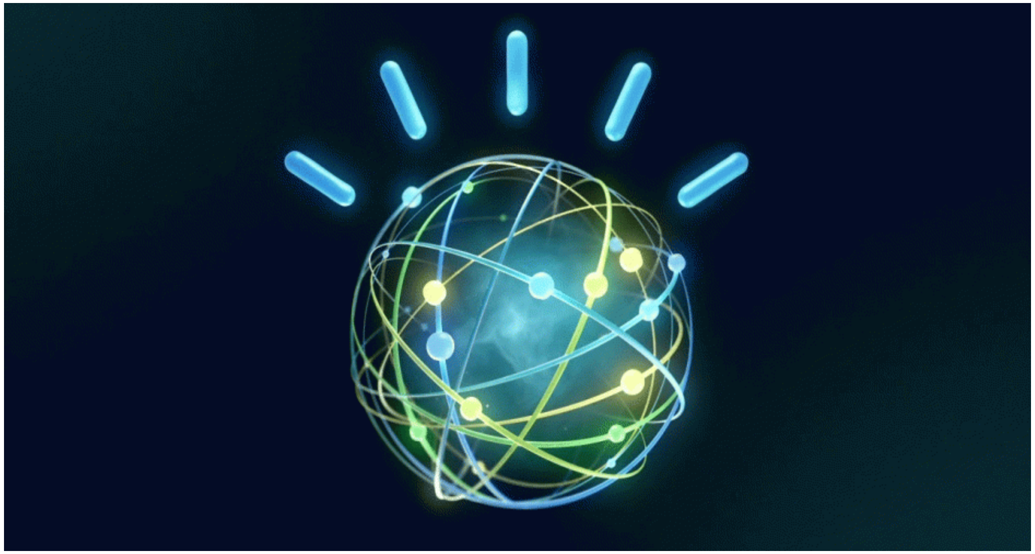
Ira A. Fulton
Schools of Engineering
ARIZONA STATE UNIVERSITY

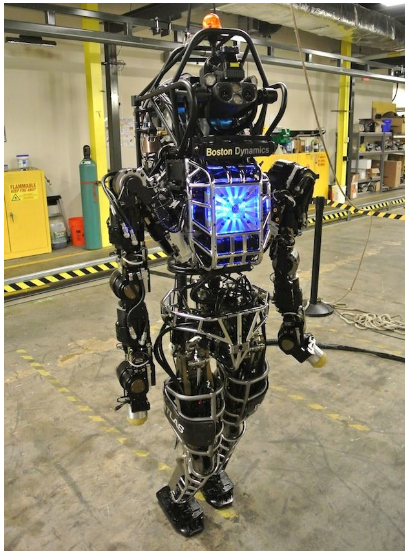

THUS, FOR ANY NONDETERMINISTIC TURING MACHINE M THAT RUNS IN SOME POLYNOMIAL TIME p(n), WE CAN DEVISE AN ALGORITHM THAT TAKES AN INPUT ω OF LENGTH n AND PRODUCES E_{M,ω}. THE RUNNING TIME IS O(p²(n)) ON A MULTITAPE DETERMINISTIC TURING MACHINE AND...
[redacted], MAN. I JUST WANTED TO LEARN HOW TO PROGRAM VIDEO GAMES.
SIPSER CH7

http://abstrusegoose.com/

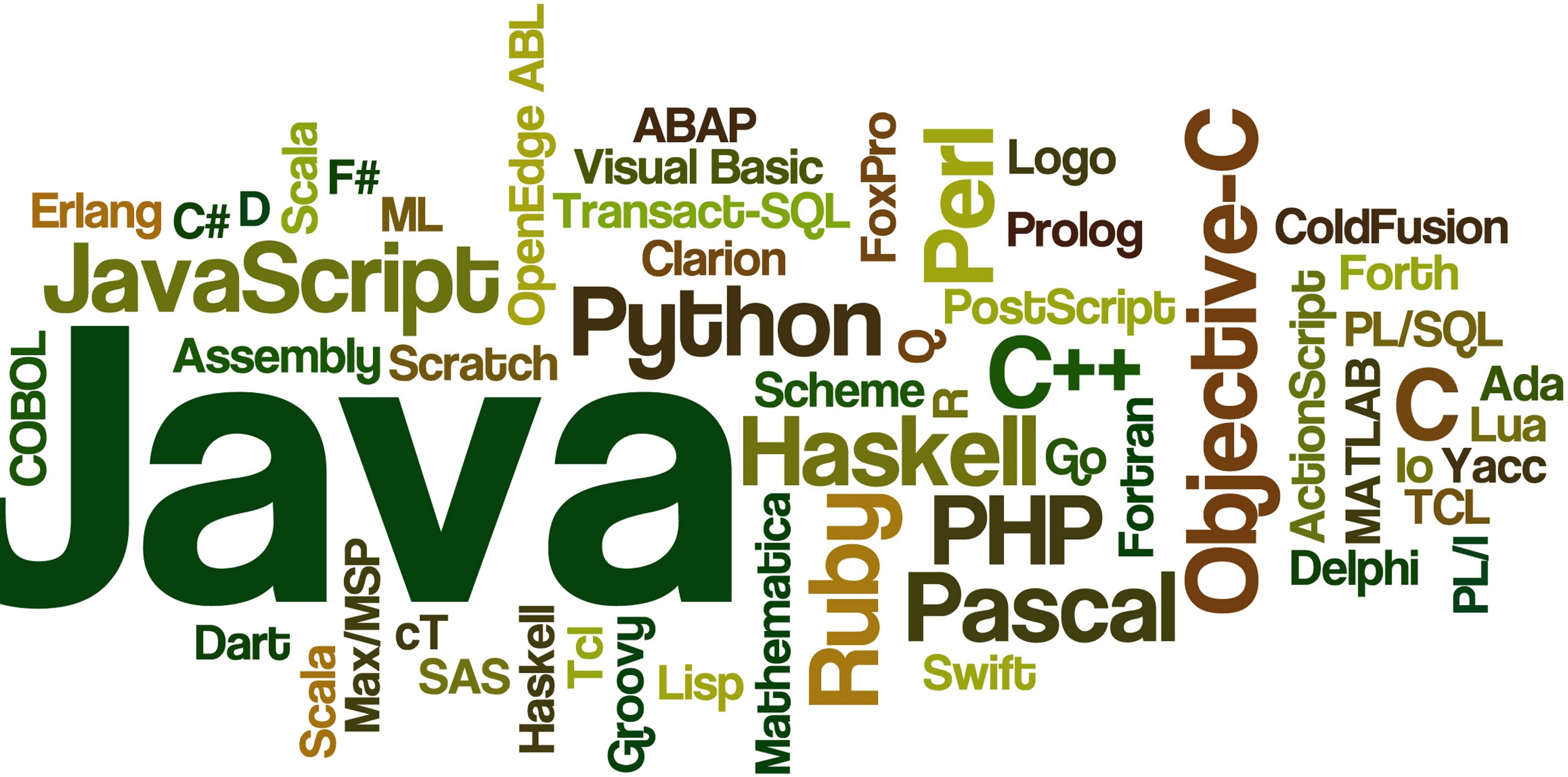
Subject of interest?

Subject of interest?

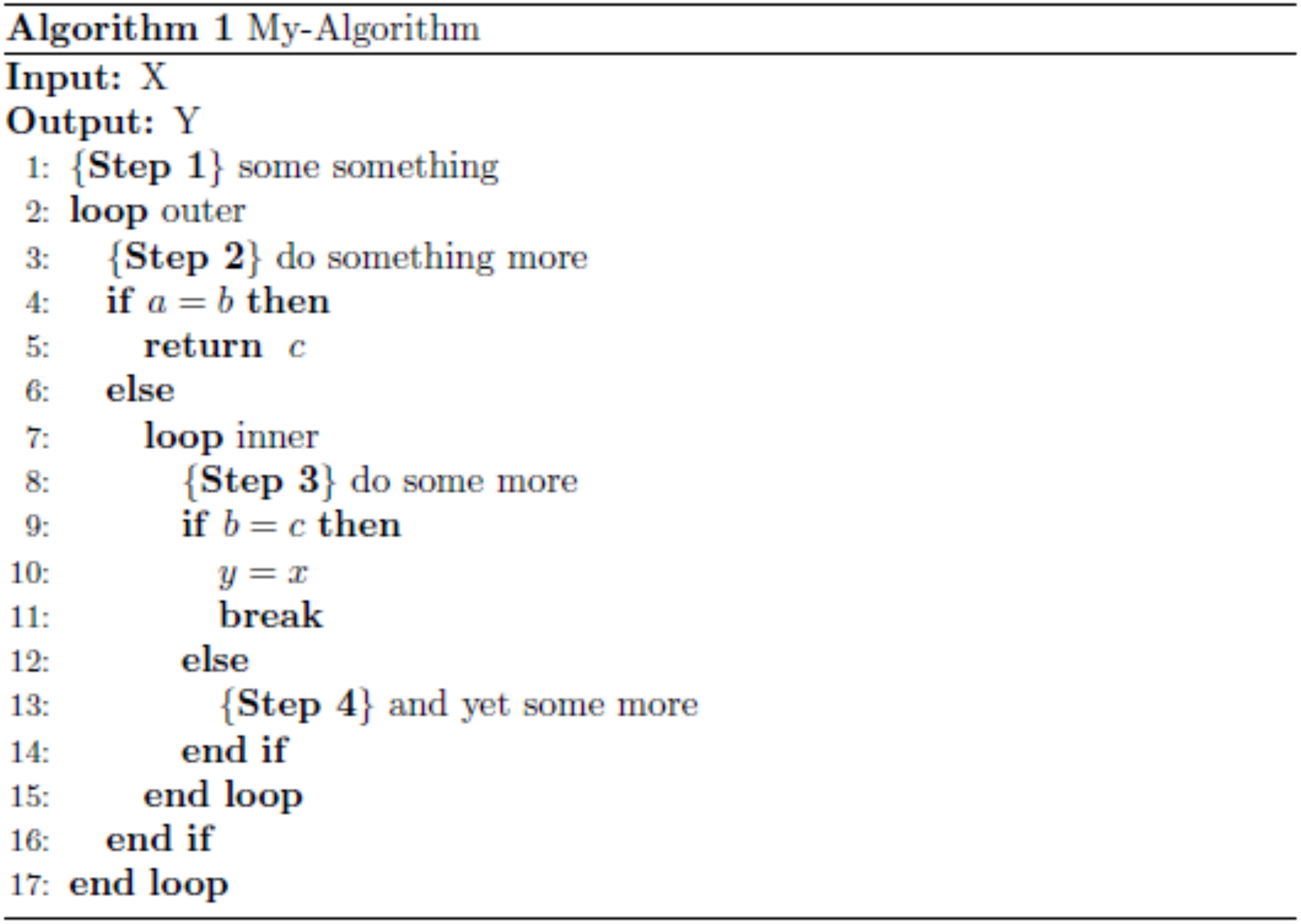




Subject of interest?



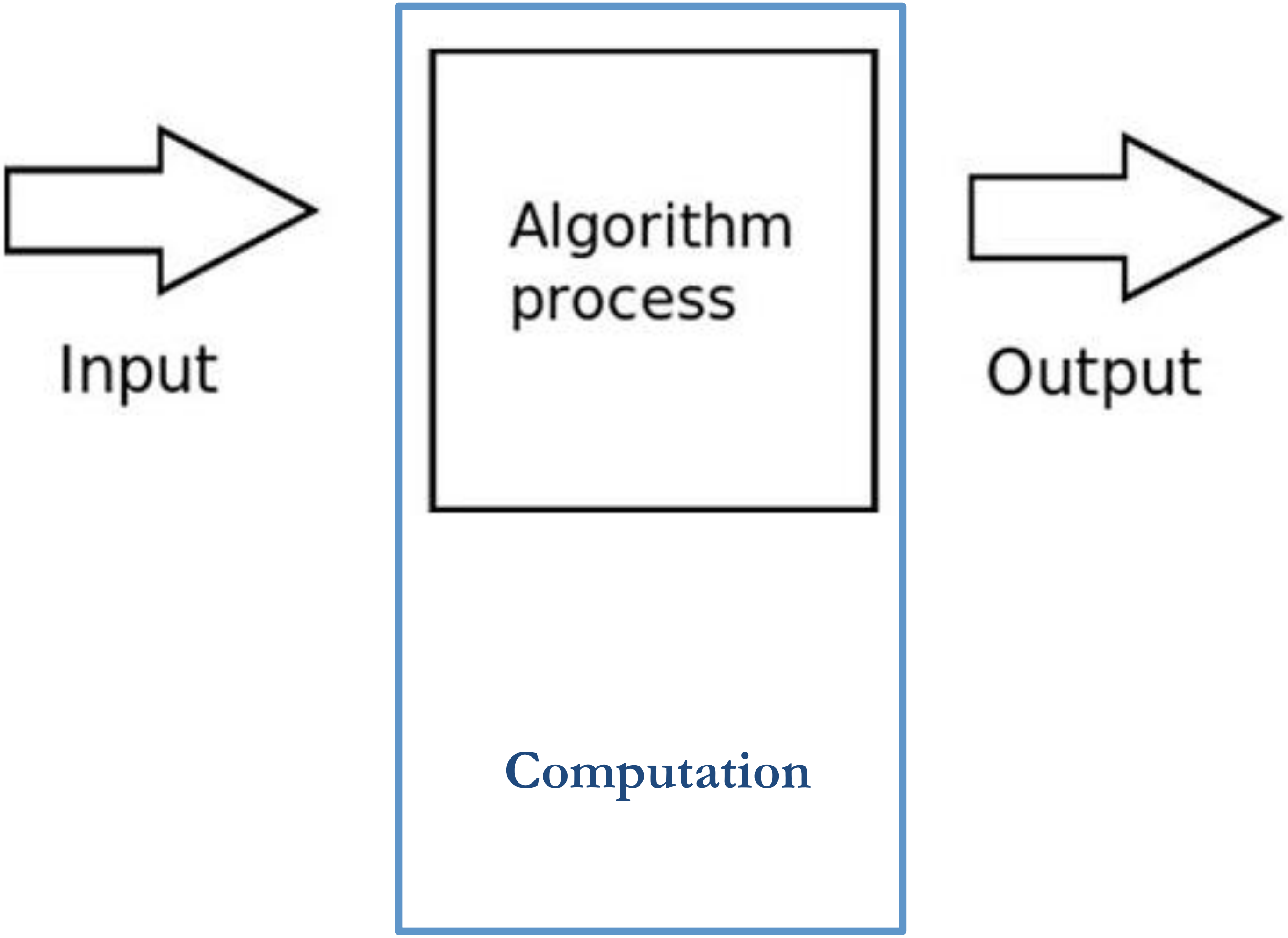
Subject of interest?



Algorithm 1 My-Algorithm

```
Input: X
Output: Y
{Step 1} some something
loop outer
    {Step 2} do something more
    if $a = b$ then
        return $c$
    else
        loop inner
            {Step 3} do some more
            if $b = c$ then
                $y = x$
                break
            else
                {Step 4} and yet some more
            end if
        end loop
    end if
end loop
```

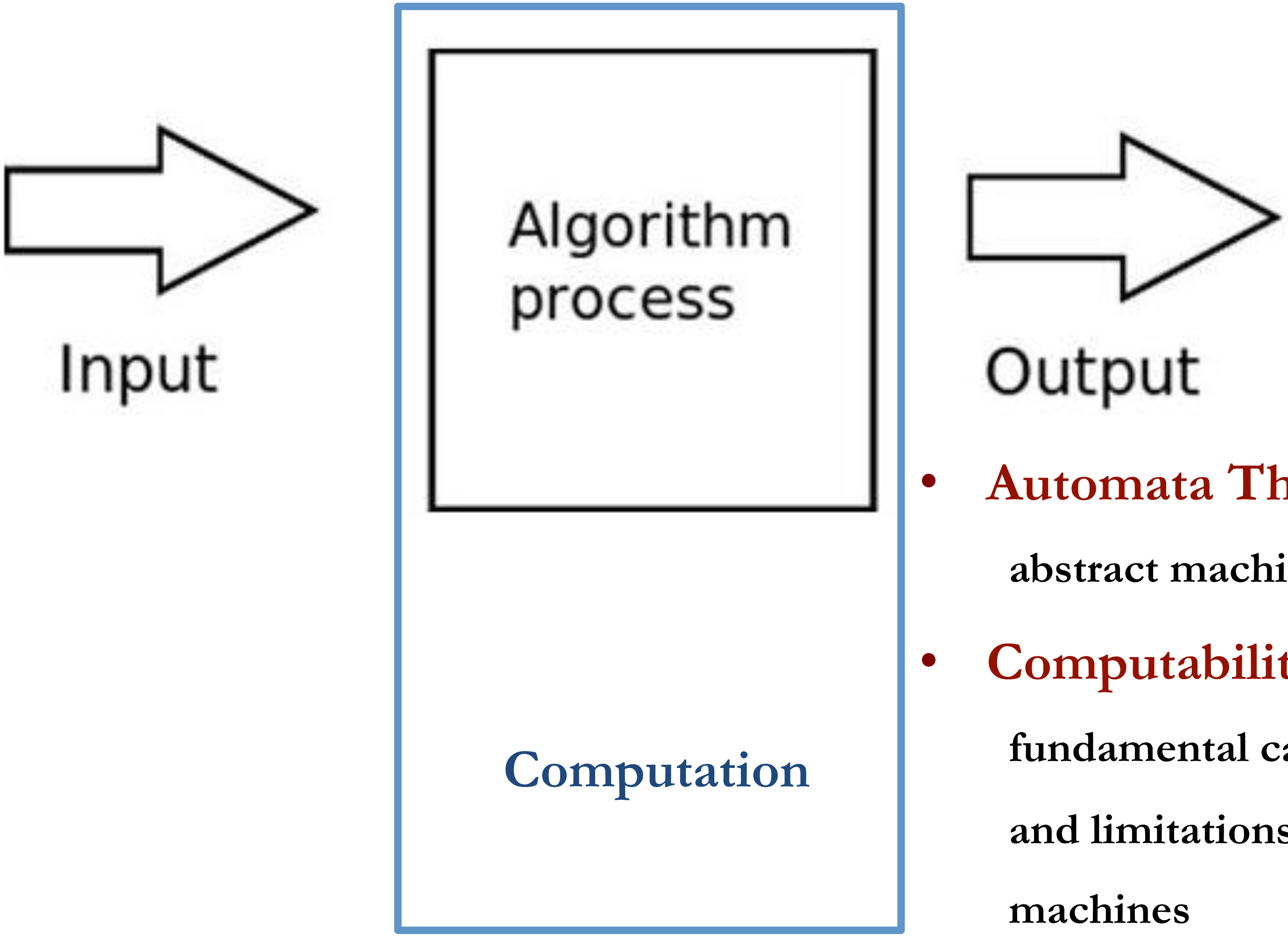
Subject of interest?



Outline for today

- **Theory of computation**
- Why it is important
- Discussion of Syllabus
- Questions and Answers

Theory of computation



- **Automata Theory**

  abstract machines

- **Computability Theory**

  fundamental capabilities and limitations of abstract machines

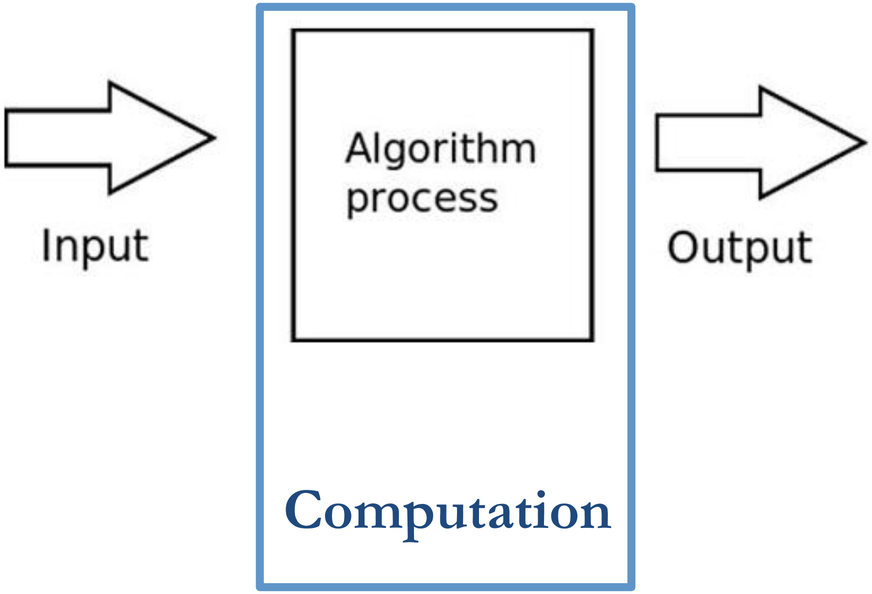
- **Complexity Theory**

  why certain problems are harder than others

Outline for today

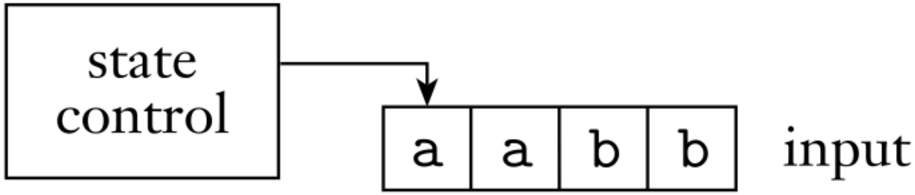
- Theory of computation
- **Why it is important**
- Discussion of Syllabus
- Questions and Answers

- **Automata Theory**
- Computability Theory
- Complexity Theory



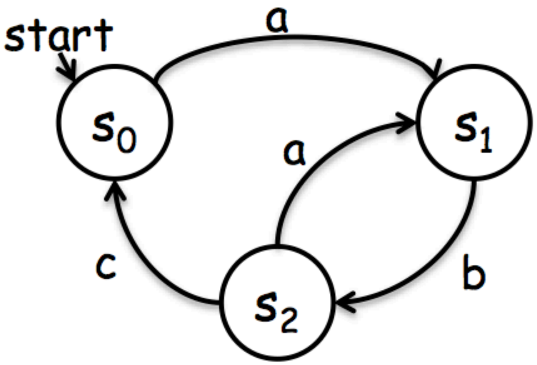
Automata Theory – Finite Automata

- An abstract machine (or mathematical model of computation) that can capture systems with a finite number of states
    - Automation applications where simple tasks need to be repeated
    - Easy to implement with limited resources (HW/SW)
    - Easy to design and visualize
    - Easy to verify correctness



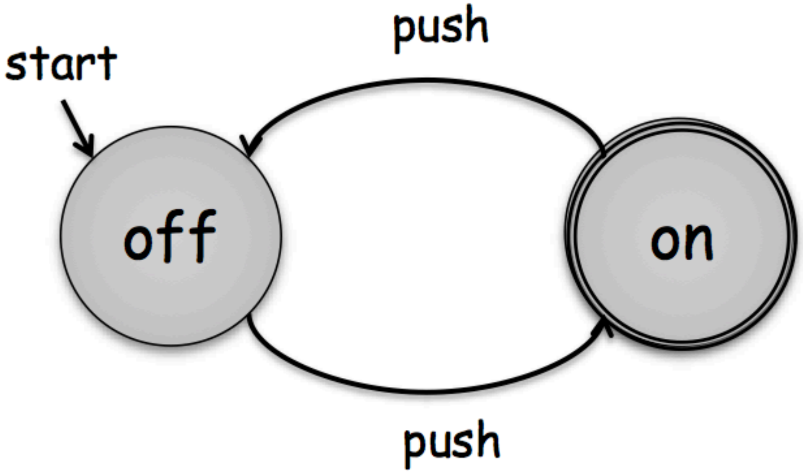
state control

a a b b input



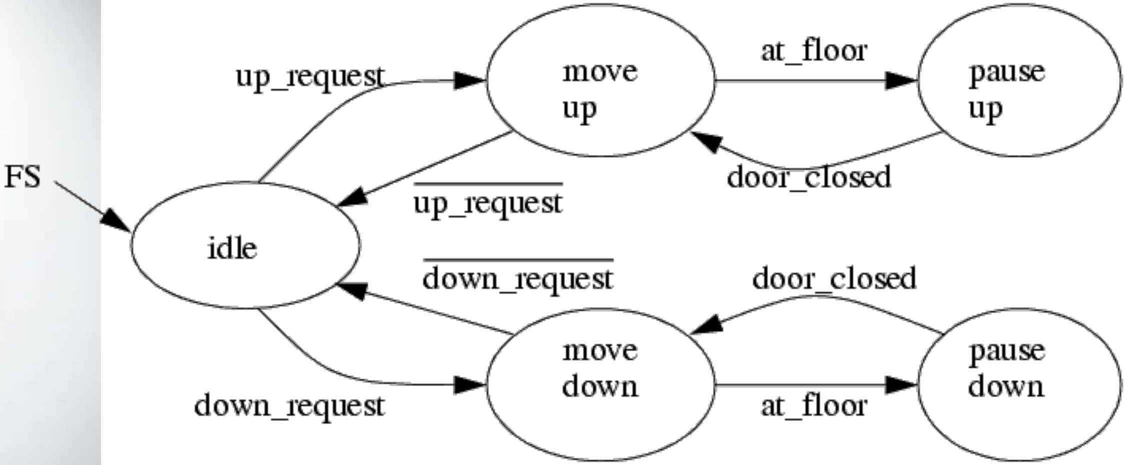
Automata Theory – Finite Automata

- Exampe



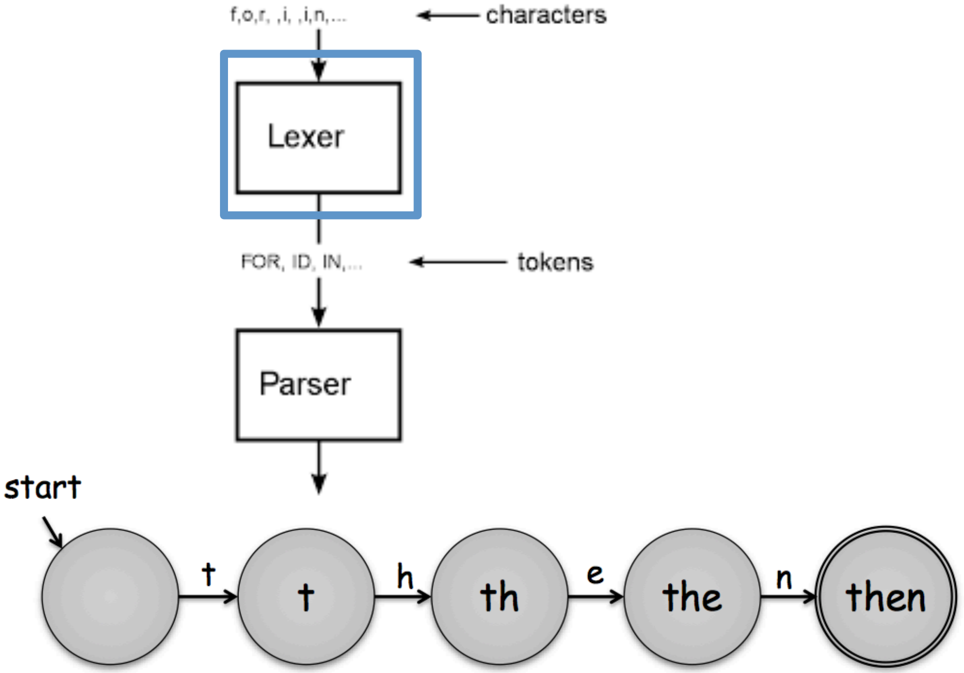
Automata Theory – Finite Automata

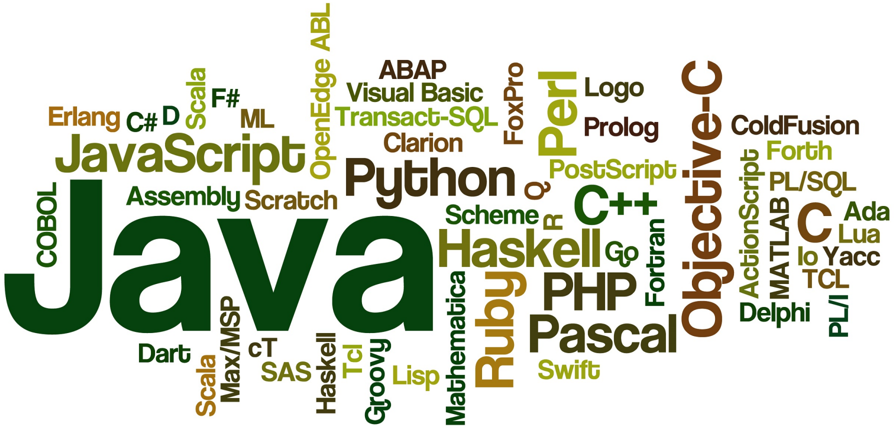
- Exampe



Automata Theory – Finite Automata

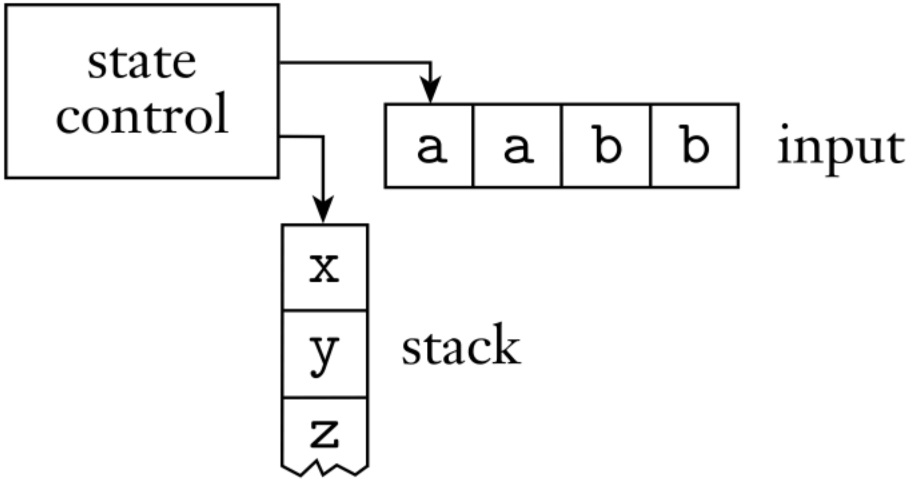
- Exampe





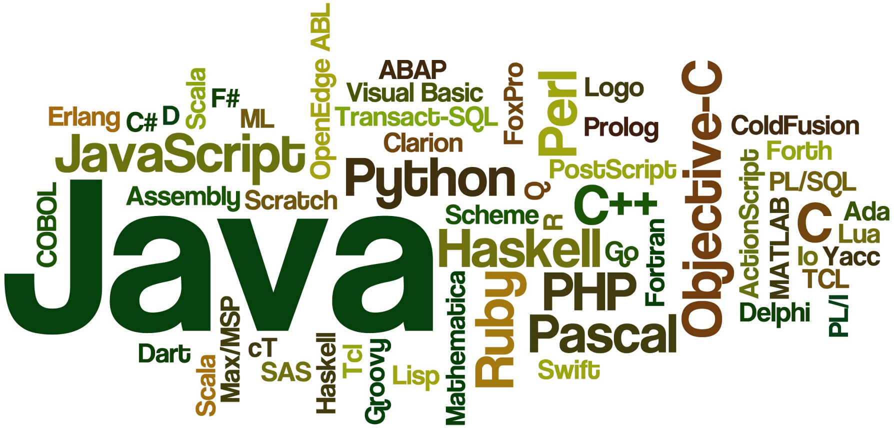
Automata Theory – Pushdown Automata

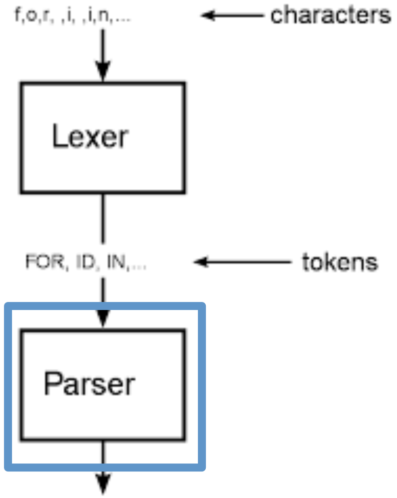
- An abstract machine (or mathematical model of computation) that can capture systems with a finite number of states **and a stack**
  - A PDA can write to a stack
  - At each step, the read–write head can only access the top symbol in the stack, which can be poped or kept; a new symbol may be pushed onto the stack



Automata Theory – Pushdown Automata

- Exampe





f,o,r, ,i, ,i,n,...

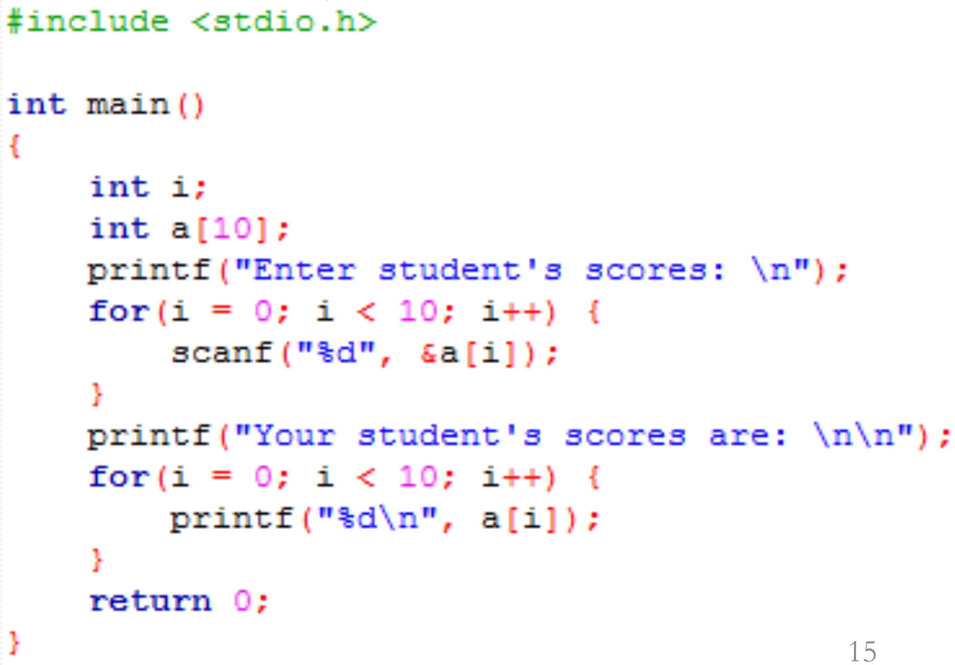
characters

Lexer

FOR, ID, IN,...

tokens

Parser

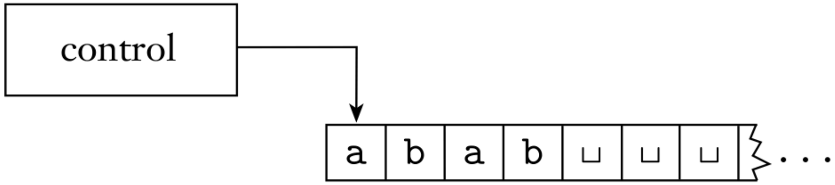


```
#include <stdio.h>

int main()
{
    int i;
    int a[10];
    printf("Enter student's scores: \n");
    for(i = 0; i < 10; i++) {
        scanf("%d", &a[i]);
    }
    printf("Your student's scores are: \n\n");
    for(i = 0; i < 10; i++) {
        printf("%d\n", a[i]);
    }
    return 0;
}
```

Automata Theory – Turing Machines

- An abstract machine (or mathematical model of computation) that can capture systems that can manipulate symbols on a strip of tape
  - ➢ Despite the model's simplicity, given any computer algorithm, a Turing machine can be constructed that is capable of simulating that algorithm's logic.
  - ○ A Turing machine can both write on the tape and read from it.
  - ○ The read–write head can move both to the left and to the right.
  - ○ The tape is infinite.
  - ○ The special states for rejecting and accepting take effect immediately.

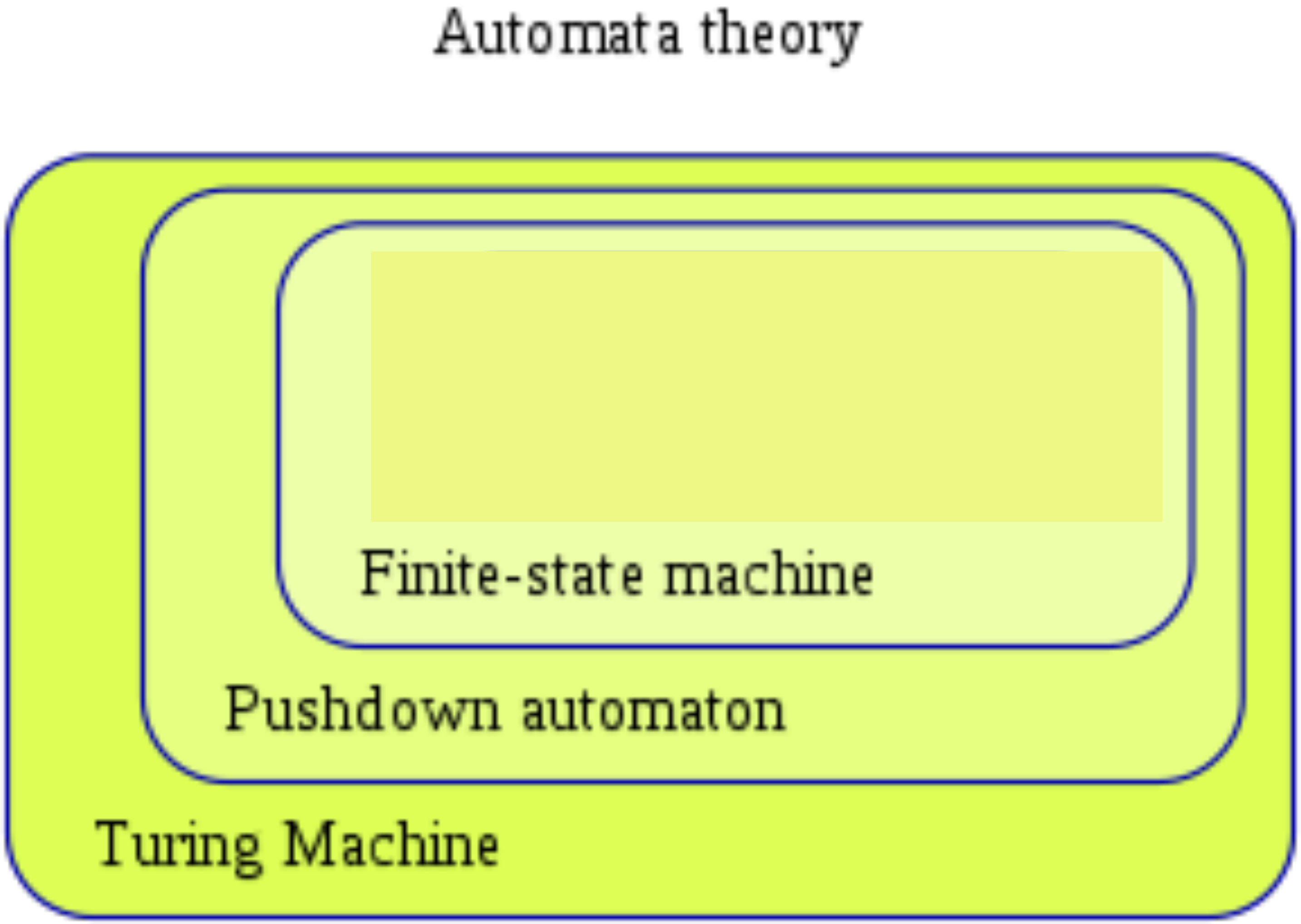


Automata Theory – Turing Machine

- Exampe

➢ Turing completeness is the ability for a system of instructions to simulate a Turing Machine. A programming language that is Turing complete is theoretically **capable of expressing all tasks accomplishable by computers**; nearly all programming languages are Turing complete if the limitations of finite memory are ignored.

Automata Theory



Outline for today

- Theory of computation
- **Why it is important**
- Discussion of Syllabus
- Questions and Answers

- Automata Theory
- **Computability Theory**
- Complexity Theory

Computability Theory

- What are the fundamental capabilities and limitations of computers?
  - Can we design a turing machine (or program) that could examine another turing machine (or program) M with input I, and decide whether M on input I will terminate?
  - Can we design a turing machine (or program) that could determine whether a mathematical statement is true of false?

➢ Can be used to identify other unsolvable problems via **reducibility**

Q: If "classical" computers cannot solve a problem, can quantum computers solve it?

- Can quantum computing solve classically unsolvable problems, A. Hodges, arXiv preprint quant-ph/0512248, 2005

Outline for today

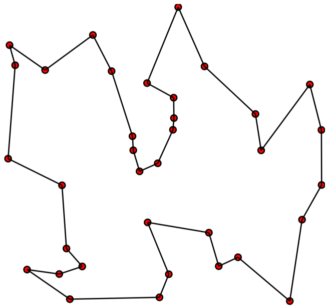
- Theory of computation
- **Why it is important**
- Discussion of Syllabus
- Questions and Answers

- Automata Theory
- Computability Theory
- **Complexity Theory**

Complexity Theory

- What makes some problems computationally hard and others easy?
  - Can we predict how fast a program will run?

- Sorting problem
- Scheduling problem
- TSP
- …



➢ In this course, we will learn how to distinguish between problems that can be solve efficiently (computationally easy problems) and problems that can take long time to be solved (computationally hard problems).

Learning goals

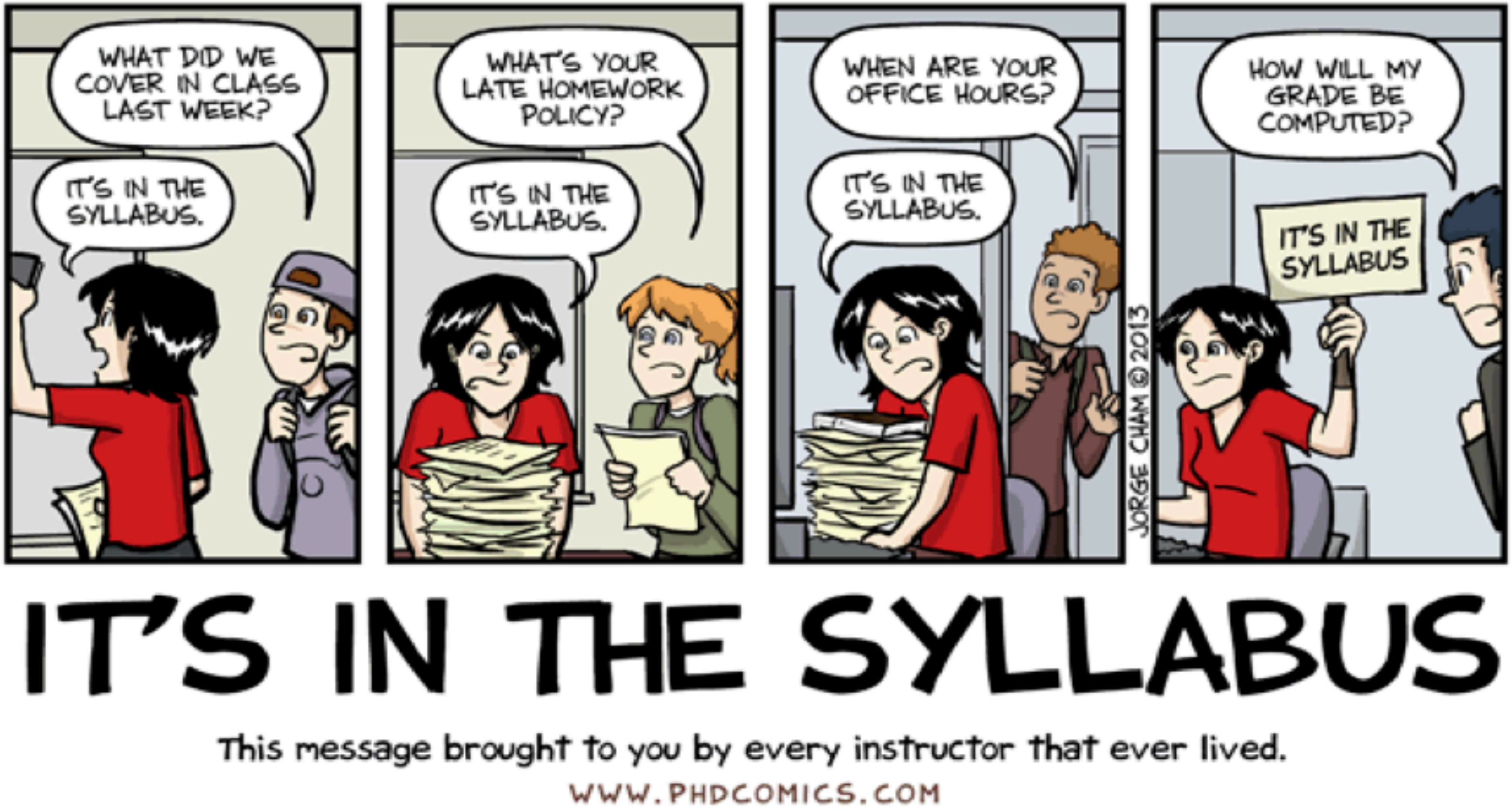
- We will learn how to abstract (i.e., abstract machines)!
- Familiarize ourselves with the mathematical formalisms and build strong foundations
  - You will understand the hardness results of various problems and rationals behind their solutions
  - You will be able to access the related literature
  - You will be able to analyze a new problem and design solutions for it
  - …

Outline for today

- Theory of computation
- Why it is important
- **Discussion of Syllabus**
- Questions and Answers

- Automata Theory
- Computability Theory
- Complexity Theory

Syllabus


WHAT DID WE COVER IN CLASS LAST WEEK?
IT'S IN THE SYLLABUS.
WHAT'S YOUR LATE HOMEWORK POLICY?
IT'S IN THE SYLLABUS.
WHEN ARE YOUR OFFICE HOURS?
IT'S IN THE SYLLABUS.
JORGE CHAM © 2013
HOW WILL MY GRADE BE COMPUTED?
IT'S IN THE SYLLABUS
IT'S IN THE SYLLABUS
This message brought to you by every instructor that ever lived.
WWW.PHDCOMICS.COM

Outline for today

- Theory of computation
  - Automata Theory
  - Computability Theory
  - Complexity Theory
- Why it is important
- Discussion of Syllabus
- **Questions and Answers**

- Reading assignment for the next class:
  - **Sipser Sec. 0.1, 0.2 and 1.1 – Quiz link will be sent out; due date is before the beginning of the next class**