CSC 339 – Theory of Computation

Course Introduction Spring 2022

Contact

- Hessah Alraqibah
  - T136
  - halraqibah@ksu.edu.sa
- Office Hours
  - On LMS

  - Information and announcements will be communicated to you via email (university email).
  - All course materials and assignments will be uploaded on LMS. Make sure you have access to LMS.

Overview

- Course description:
  - The course introduces the foundations of automata theory, computability theory, and complexity theory. It shows relationship between automata and formal languages. Addresses the issue of which problems can be solved by computational means (decidability vs undecidability), and introduces concepts related to computational complexity of problems.
- Prerequisite:
  - CSC 281: Discrete Mathematics for Computer Science
- Prerequisite to
  - CSC 340: Programming Languages & Compilers

Textbooks

- **Introduction to the Theory of Computation**
  – International Edition, 3rd Edition
  – Author: Michael Sipser © 2013
  – ISBN-10: 1133187811, ISBN-13: 9781133187813

- https://www.youtube.com/watch?v=SV57Yv8BXBc

Course Objectives

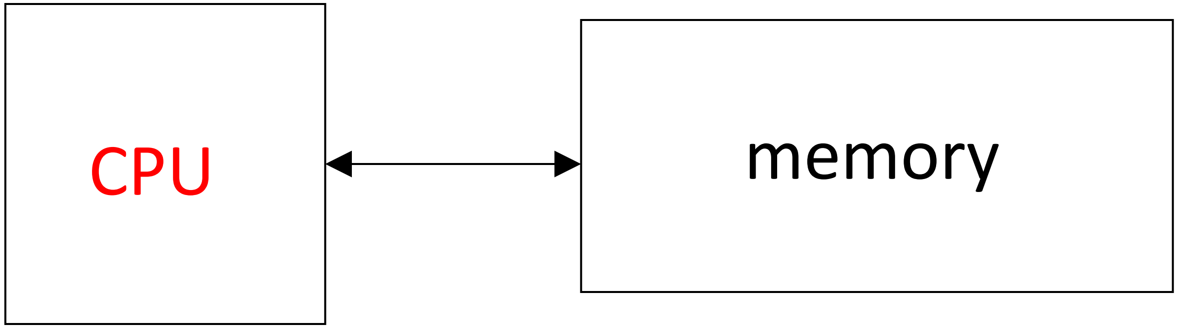
- The course aims at answering two questions:
  – what can be computed by a machine?
  – And how efficiently?
- It starts by presenting machines models, then addresses the computability problem, and then the complexity of algorithms and their classification according to it.

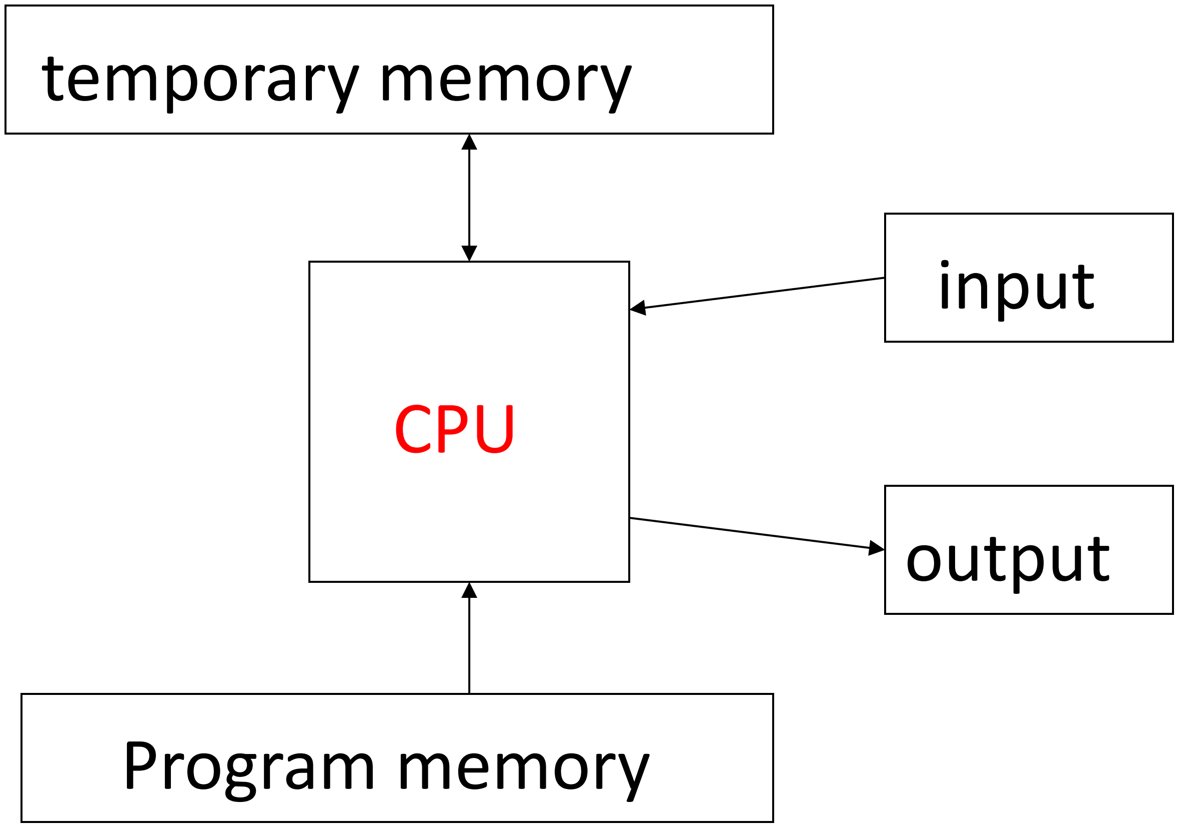
outlines

- Chapter 1
  - Finite state machines
  - Non-determinism
  - Regular expressions
- Chapter 2: context free language
- Chapter 3: Turing machine
- Chapter 4: decidability, Turing recognizability and the halting problem
- Chapter 5: undecidable problems
- Chapter 7: complexity

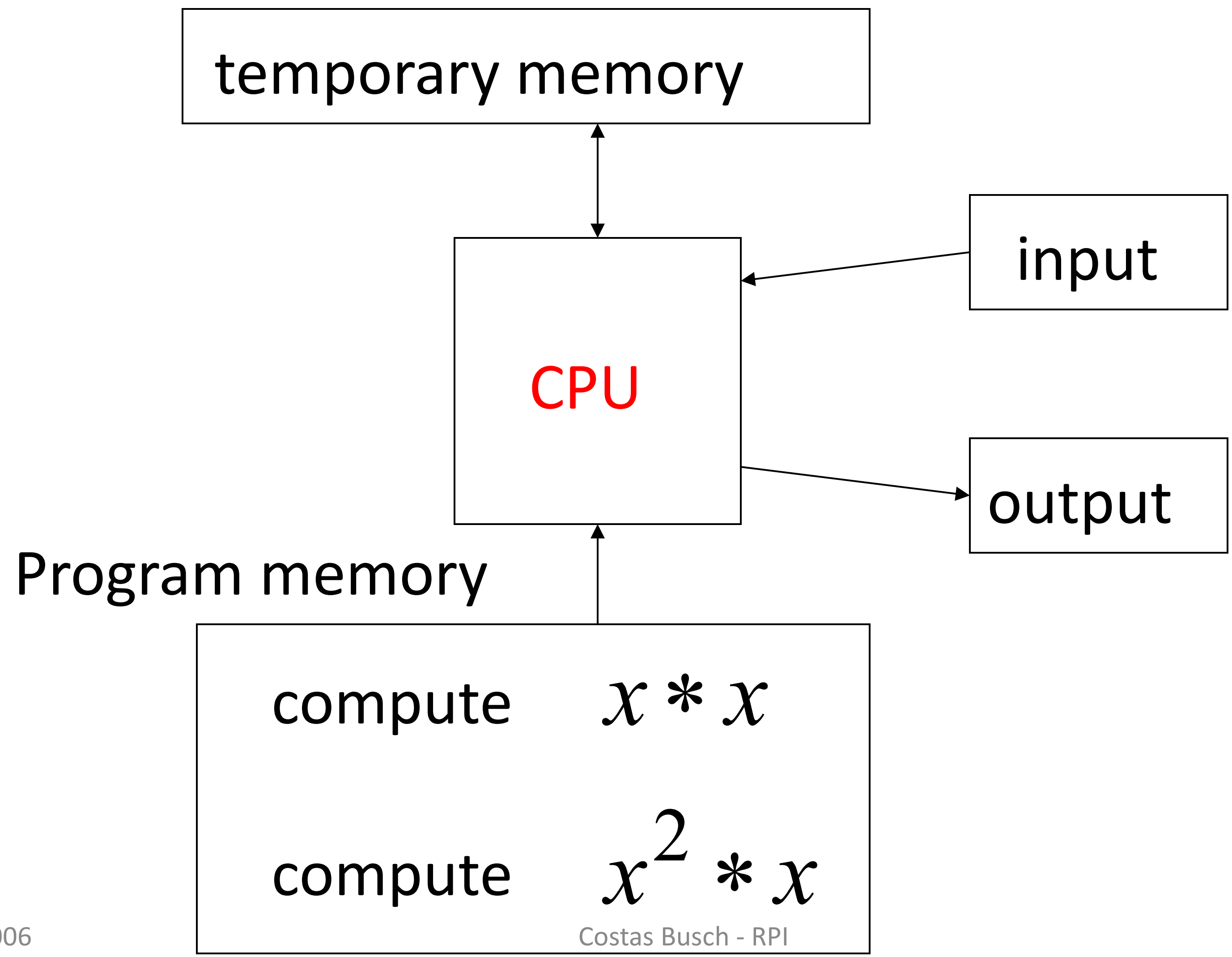
Outline of the course contents

Computation

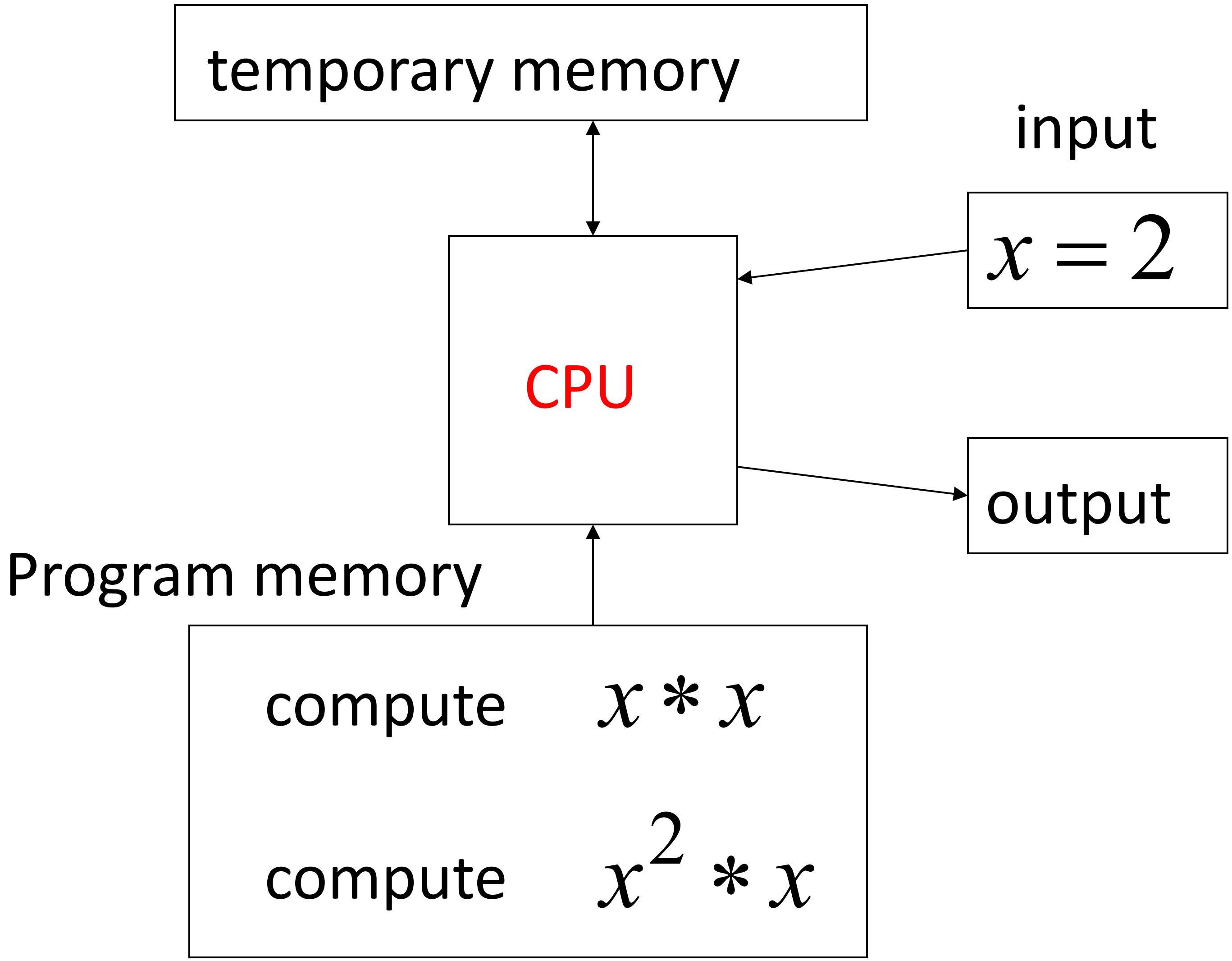



temporary memory
input
CPU
output
Program memory

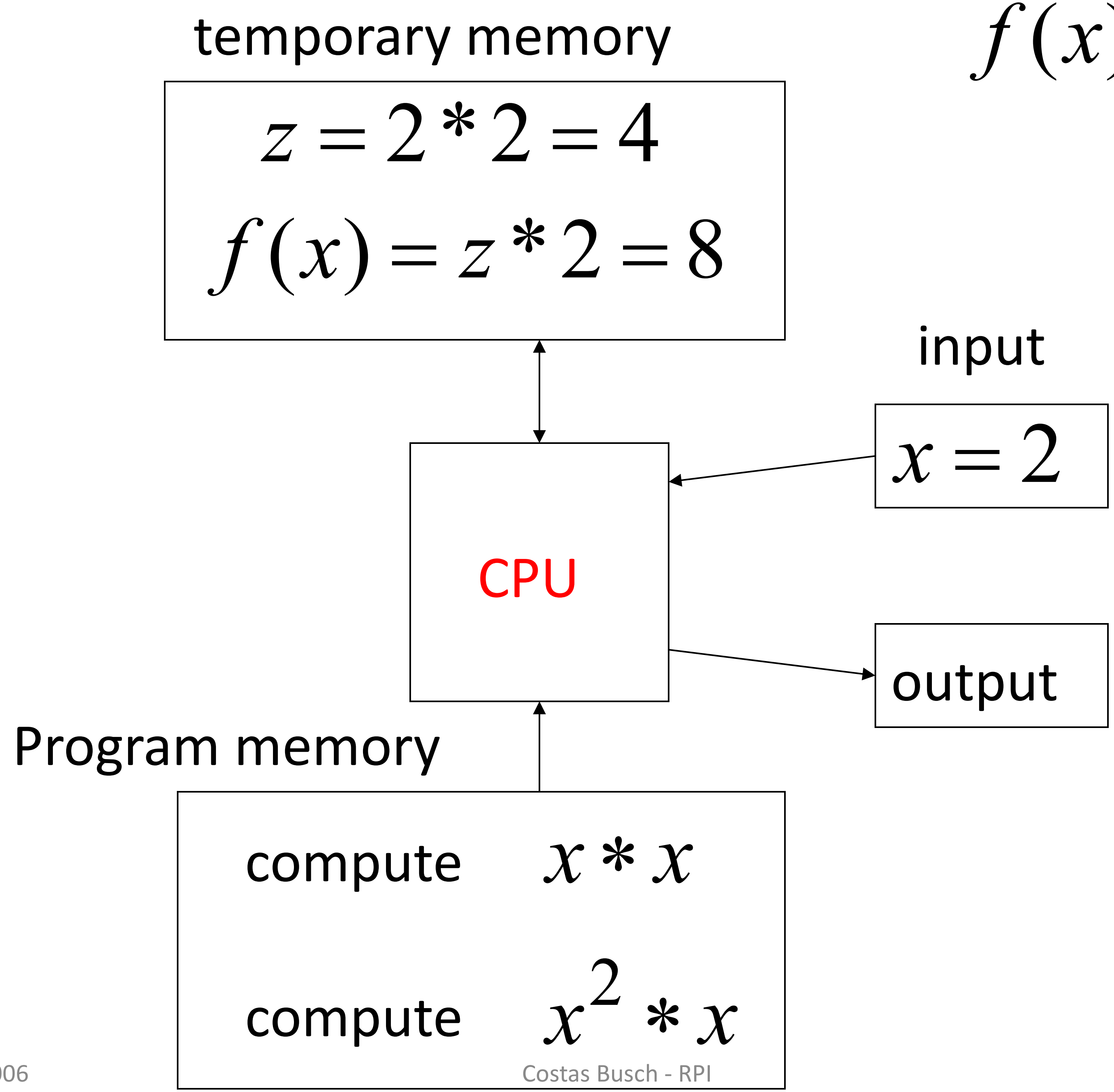
Example: $f(x) = x^3$



$$f(x) = x^3$$

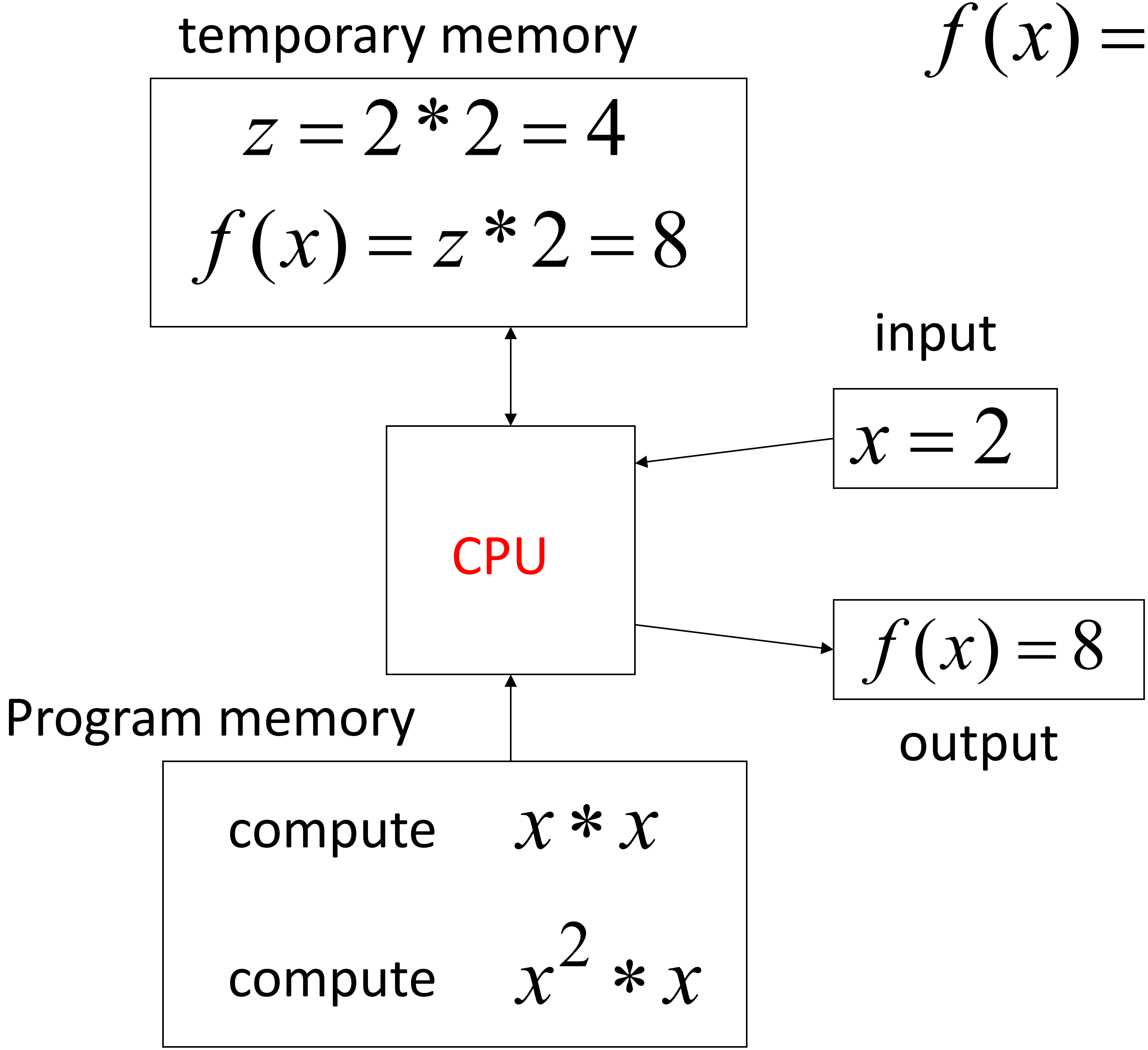

temporary memory
input
$x = 2$
CPU
output
Program memory
compute $x * x$
compute $x^2 * x$

$$f(x) = x^3$$

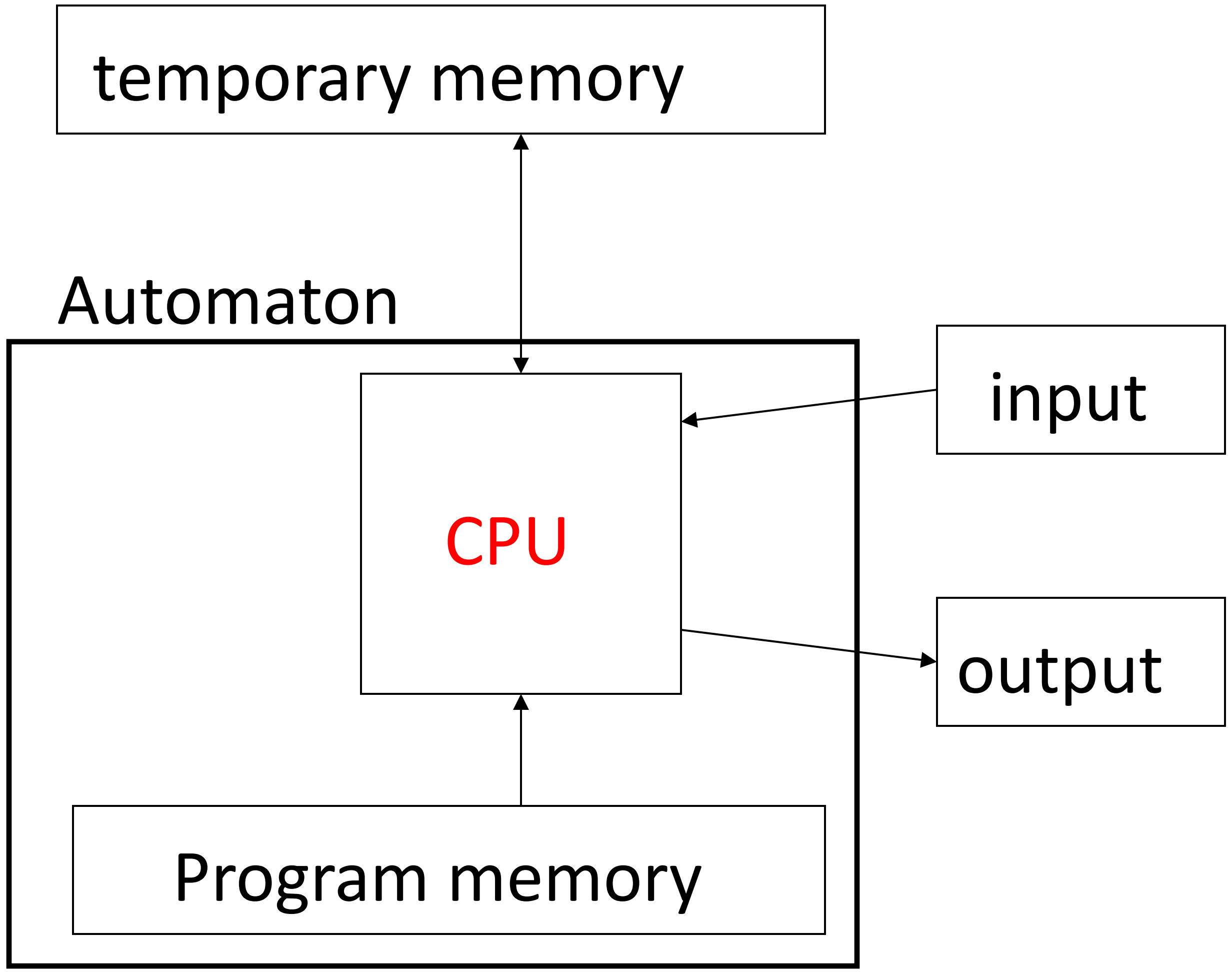


temporary memory
$z = 2 * 2 = 4$
$f(x) = z * 2 = 8$
input
$x = 2$
CPU
output
Program memory
compute $x * x$
compute $x^2 * x$

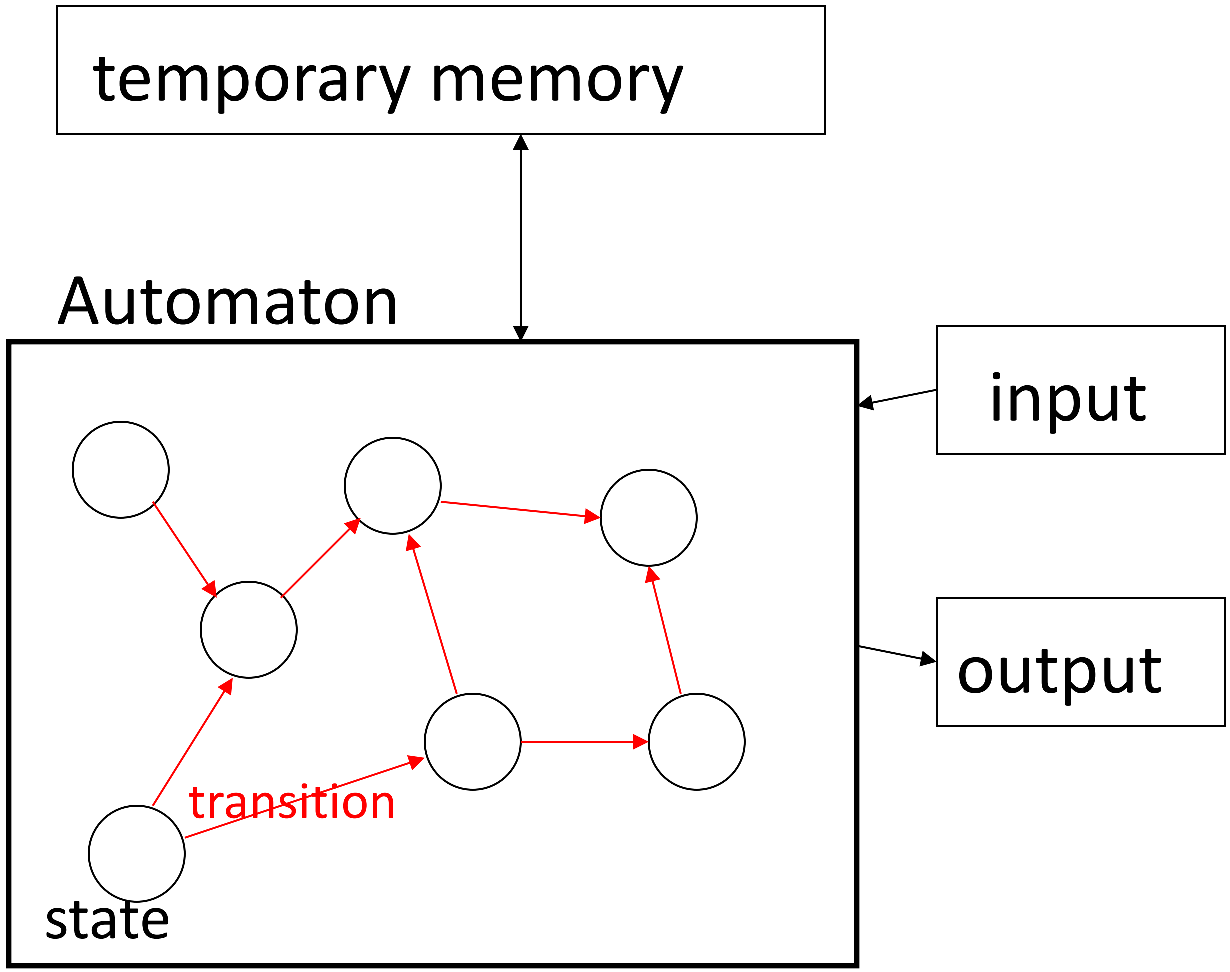
$$f(x) = x^3$$


temporary memory
$z = 2 * 2 = 4$
$f(x) = z * 2 = 8$
input
$x = 2$
CPU
$f(x) = 8$
output
Program memory
compute $x * x$
compute $x^2 * x$

Automaton


temporary memory
Automaton
CPU
input
output
Program memory

Automaton


temporary memory
Automaton
input
output
transition
state

Different Kinds of Automata

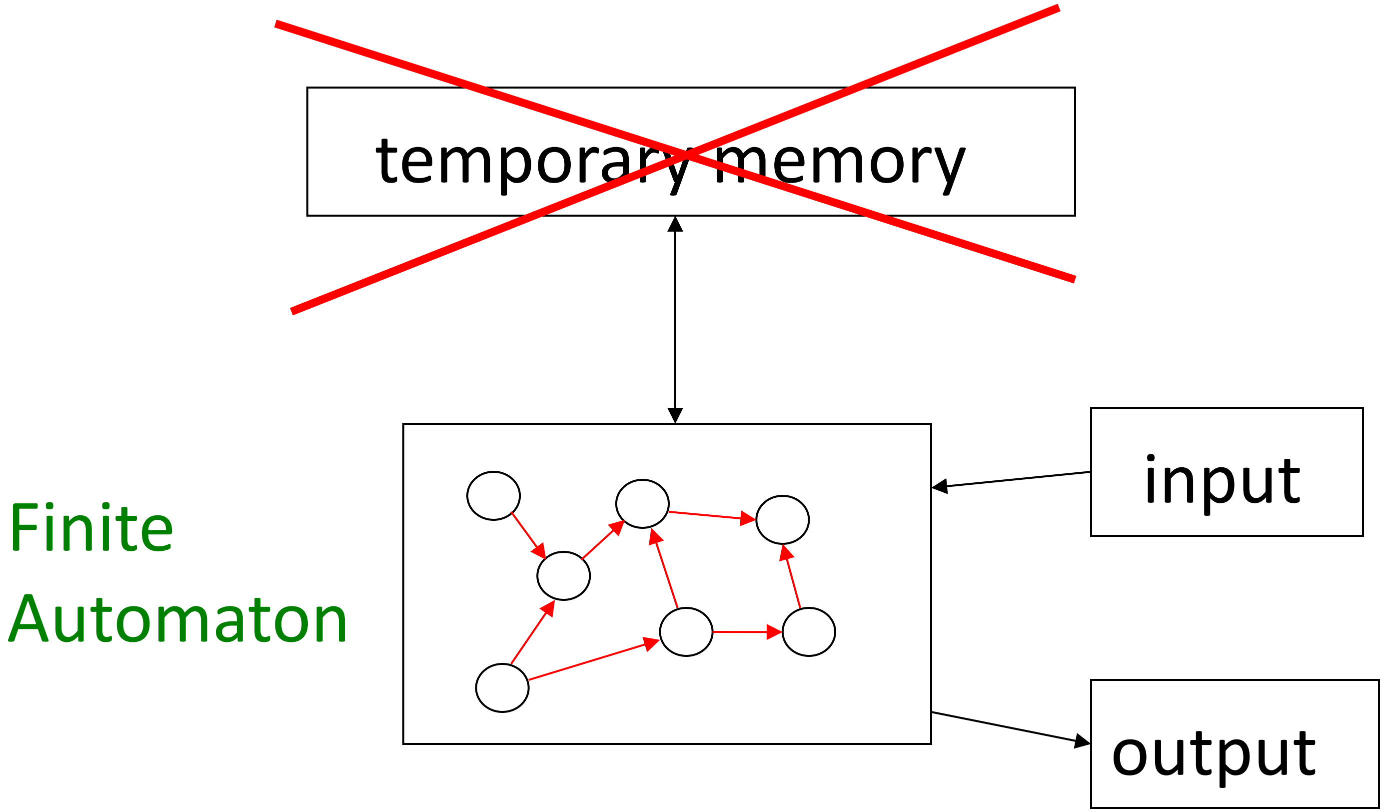
Automata are distinguished by the temporary memory

- **Finite Automata**: no temporary memory

- **Pushdown Automata**: stack

- **Turing Machines**: random access memory

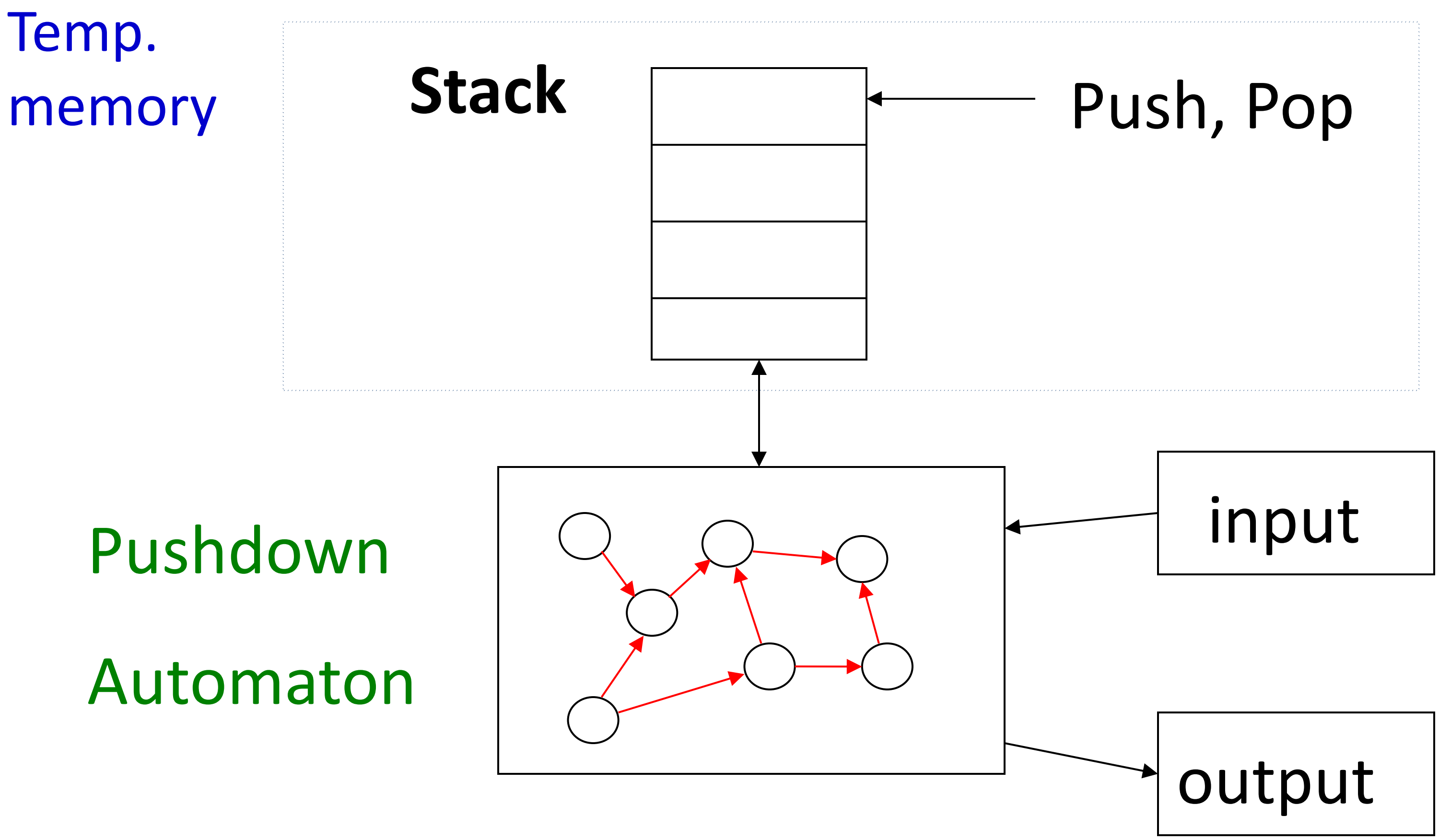
Finite Automaton



Example: Elevators, Vending Machines

(small computing power)

Pushdown Automaton



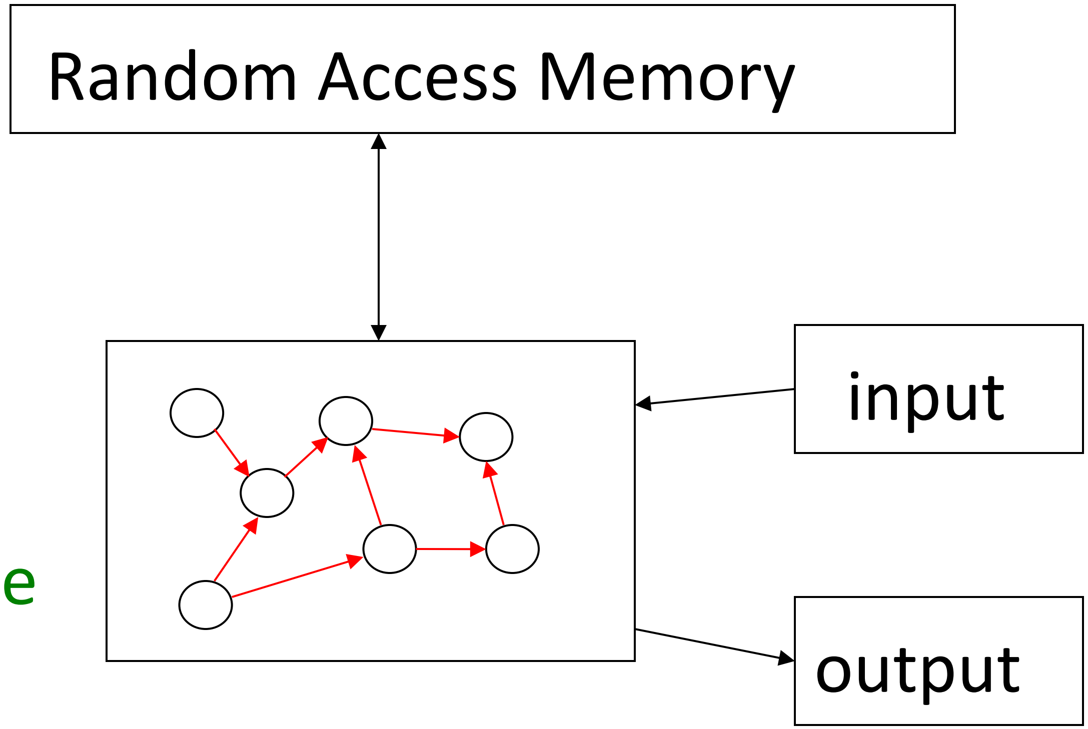
Example: Compilers for Programming Languages

(medium computing power)

Turing Machine

Temp. memory

Turing Machine



Examples: Any Algorithm

(highest computing power)

Power of Automata

Simple problems		More complex problems		Hardest problems
Finite Automata	$<$	Pushdown Automata	$<$	Turing Machine

Less power $\longrightarrow$ More power

Solve more computational problems

Power of Automata

- Turing Machine is the most powerful computational model known

- Question: Are there computational problems that a Turing Machine cannot solve?
  - Answer: Yes (unsolvable problems)

Time Complexity of Computational Problems

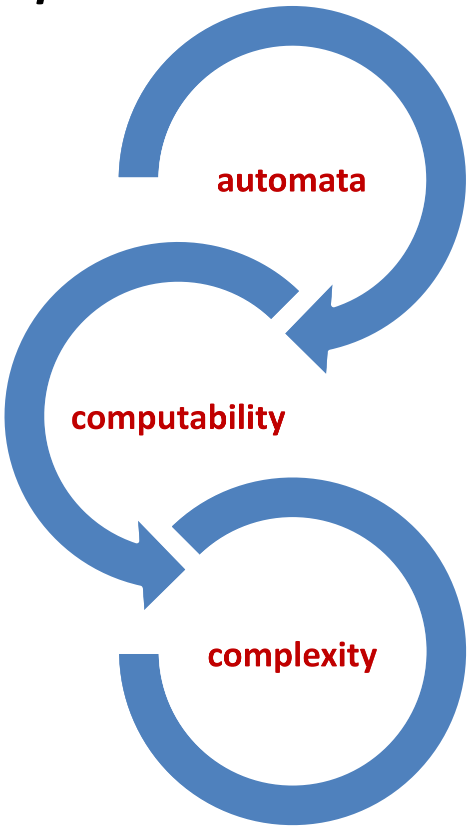
- NP-complete problems
  - <u>Believed</u> to take exponential time to be solved

- P problems
  - Solved in polynomial time

Focus

- *What are the fundamental capabilities and limitations of computers?*

Automata theory deals with the definitions and properties of mathematical models of computation.



the classification of problems is by those that are solvable and those that are not.

What makes some problems computationally hard and others easy?

Reading

- MATHEMATICAL NOTIONS AND TERMINOLOGY
  - Chapter 0 from Text Book