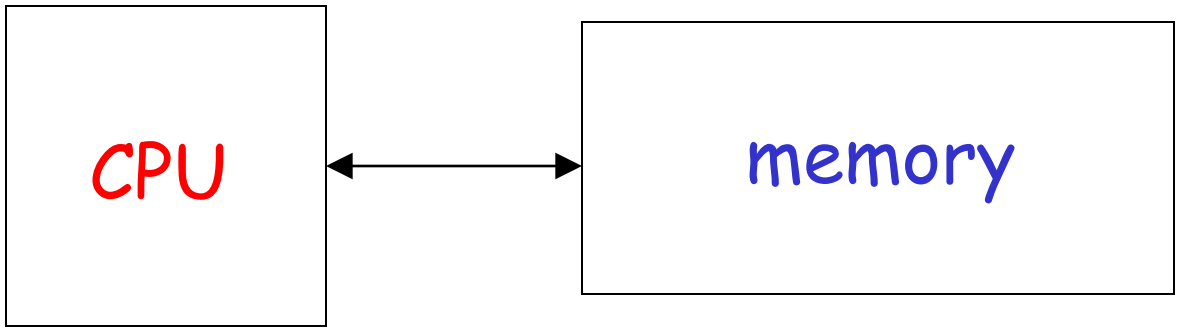
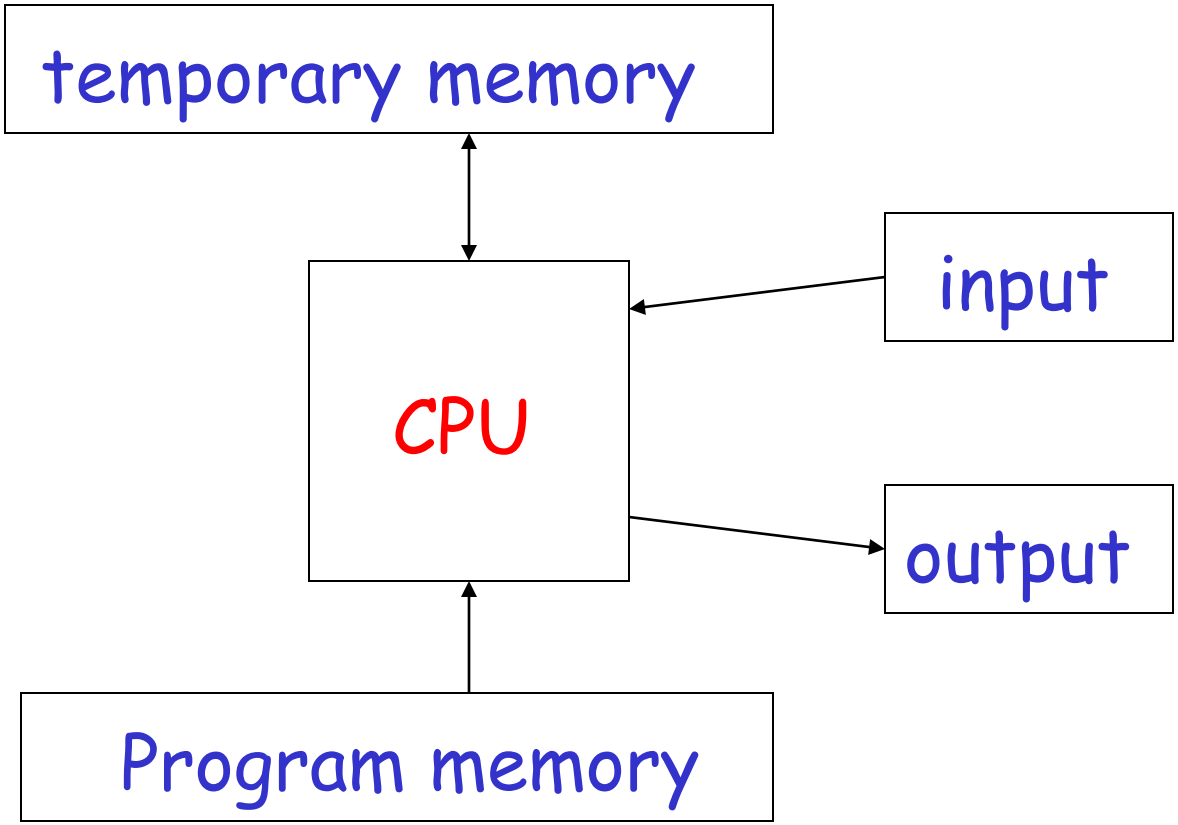
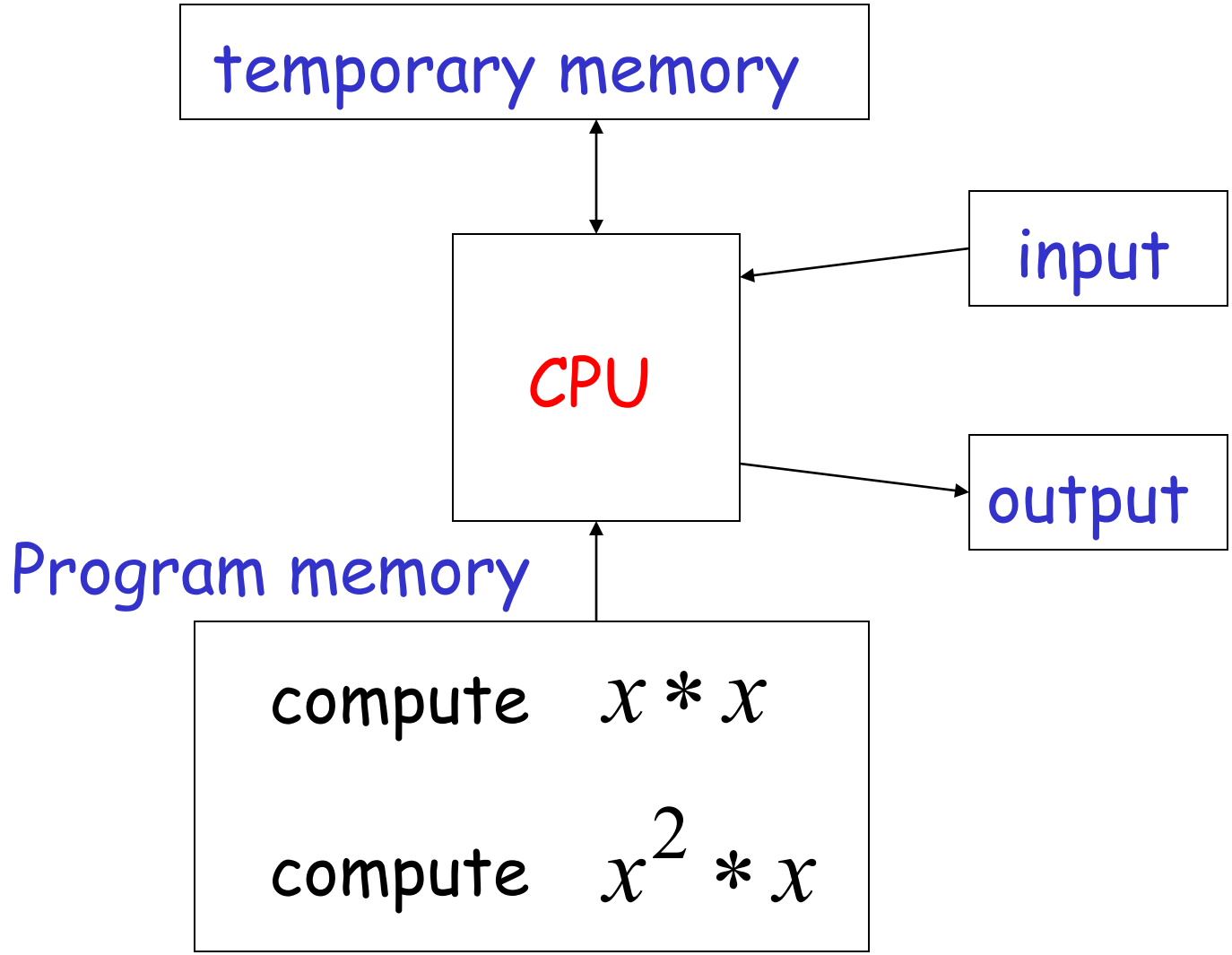
Outline of the course contents

Computation

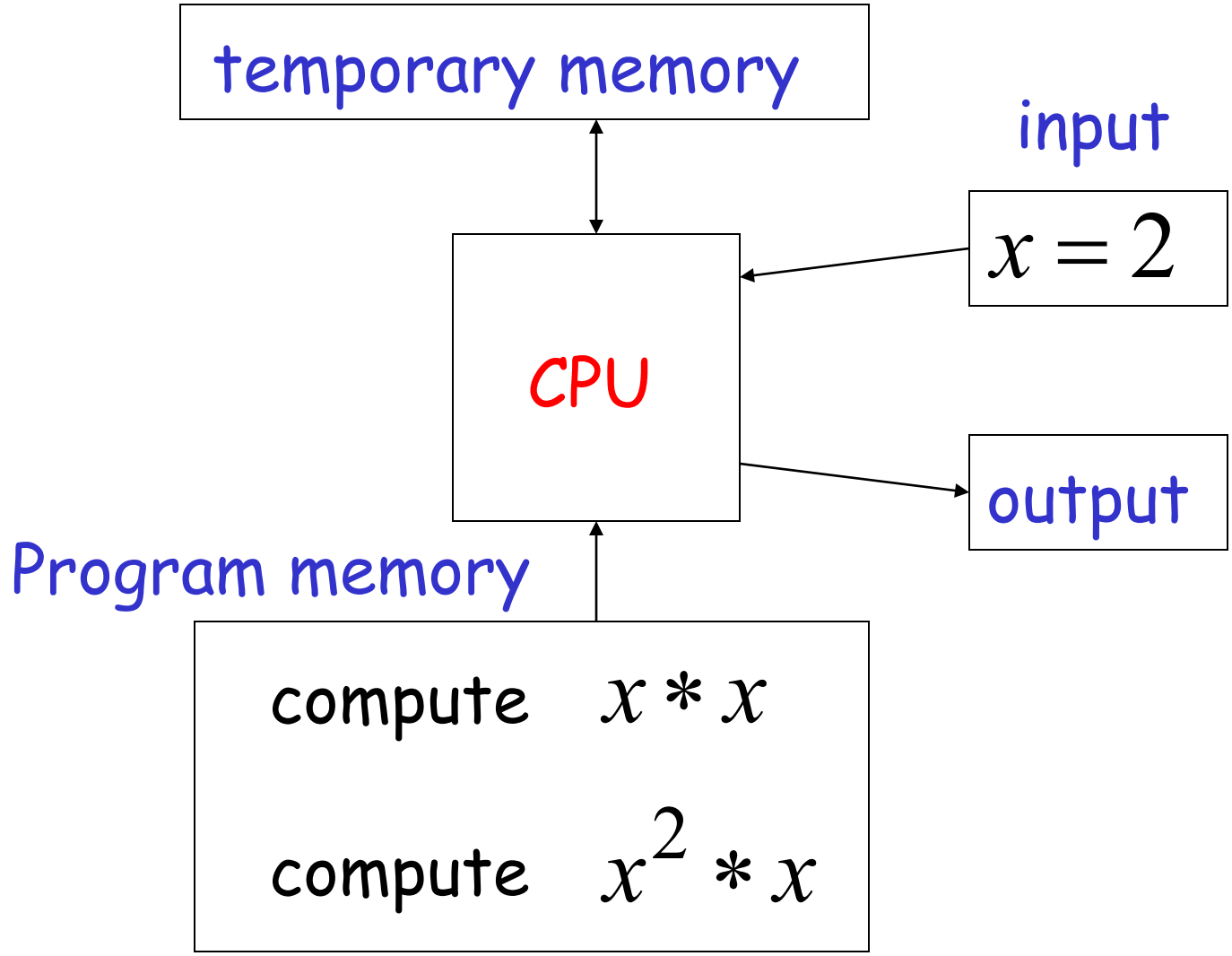



temporary memory
input
CPU
output
Program memory

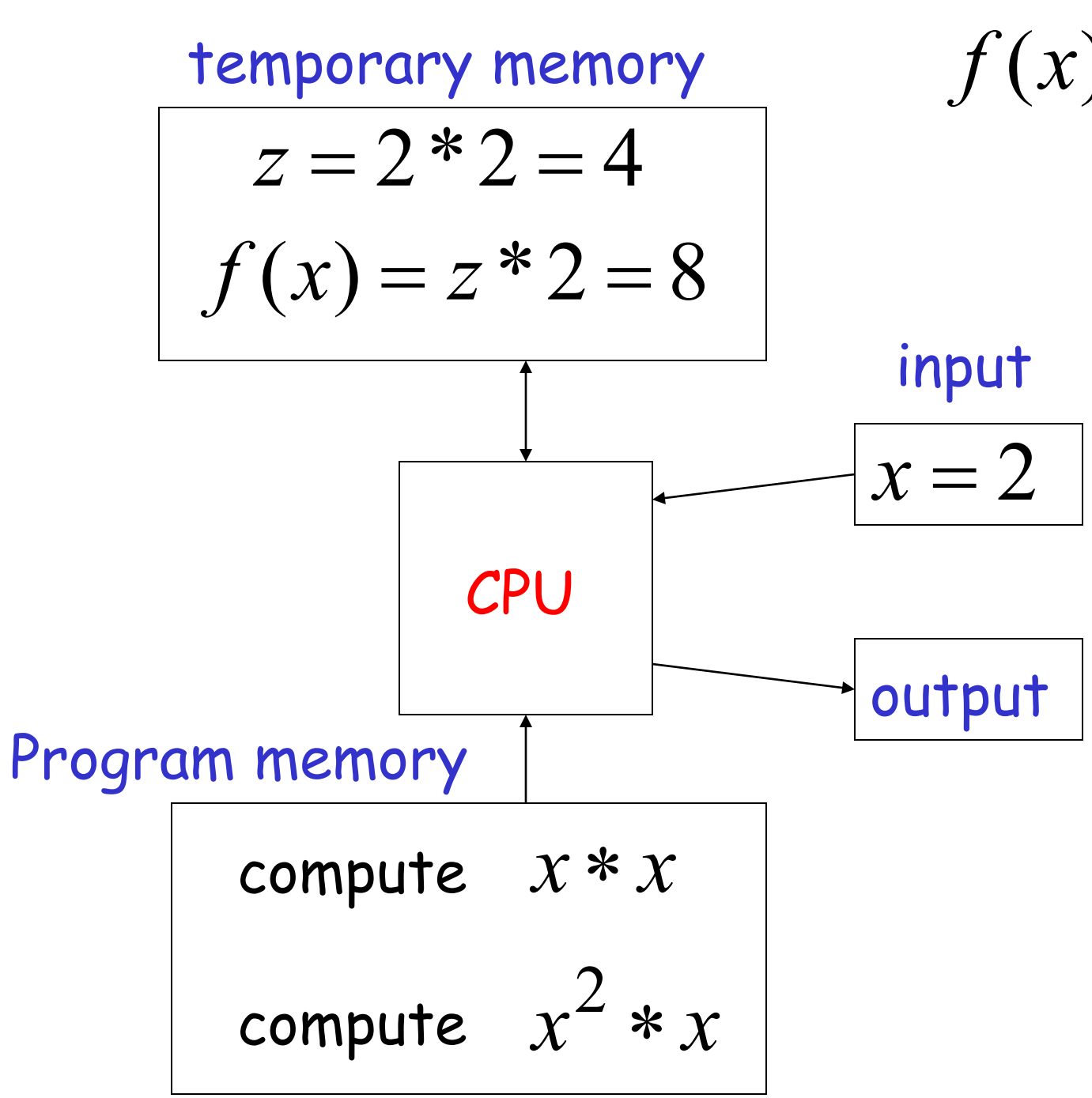
Example: $f(x) = x^3$



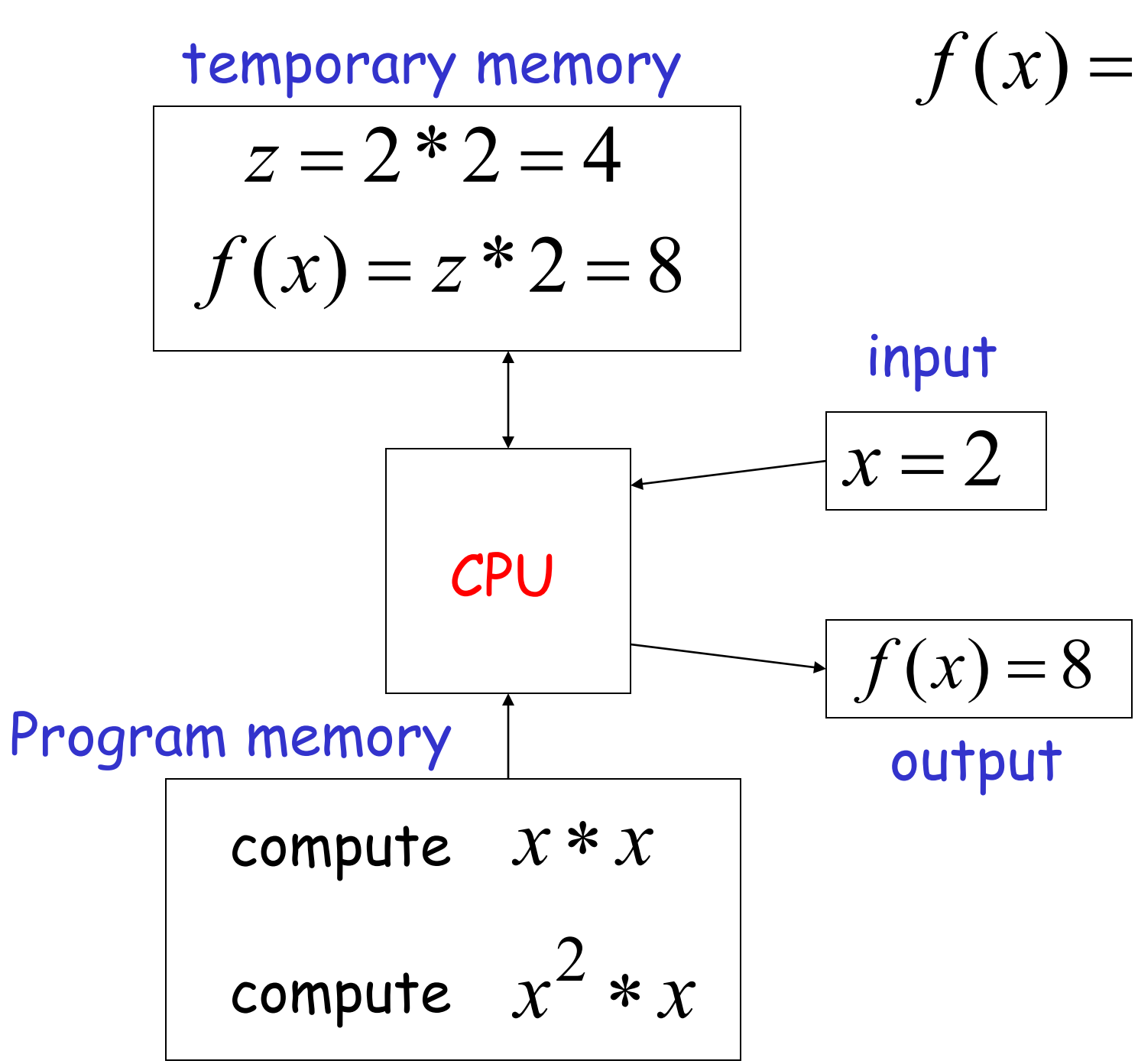
$$f(x) = x^3$$


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

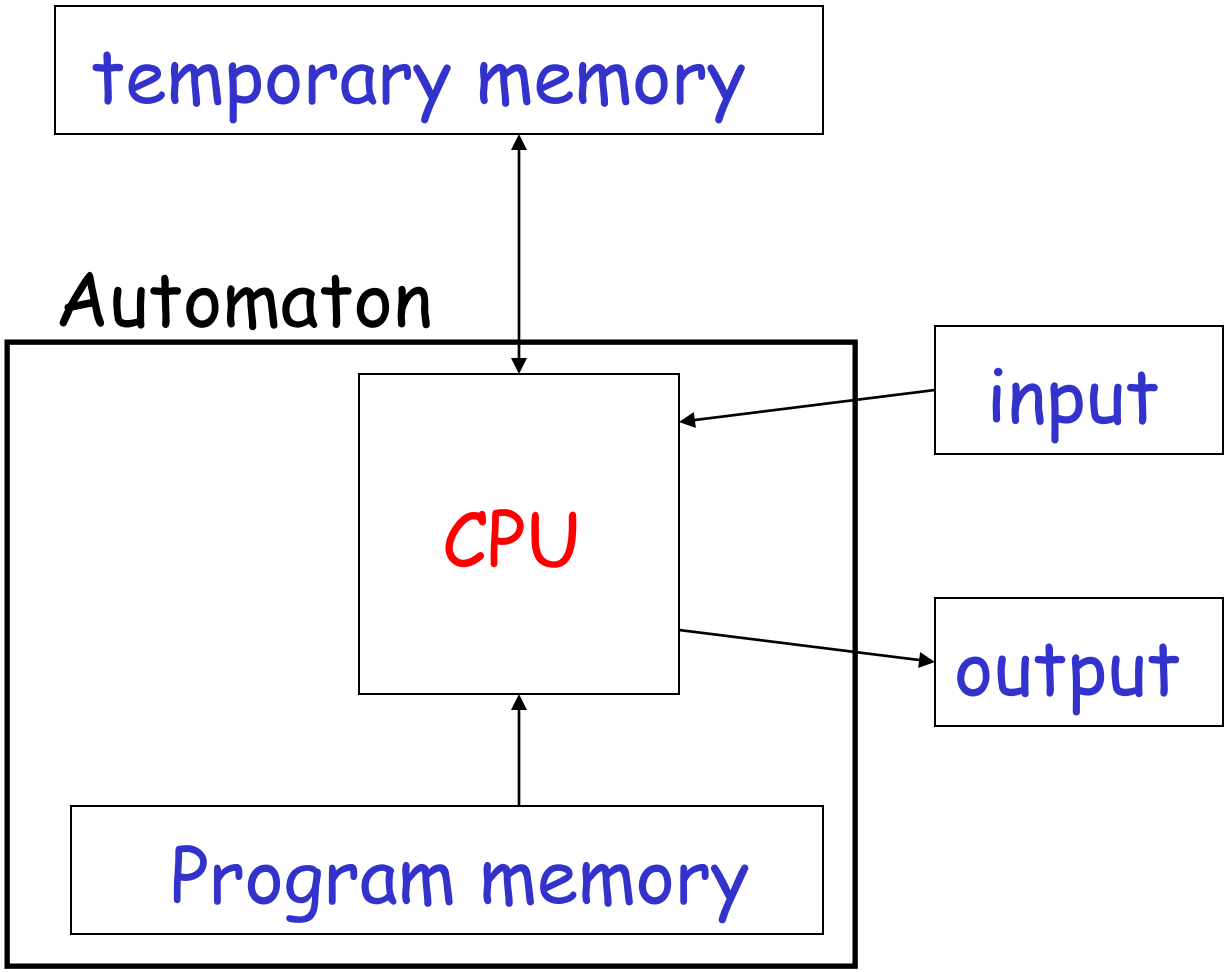
$f(x) = x^3$


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

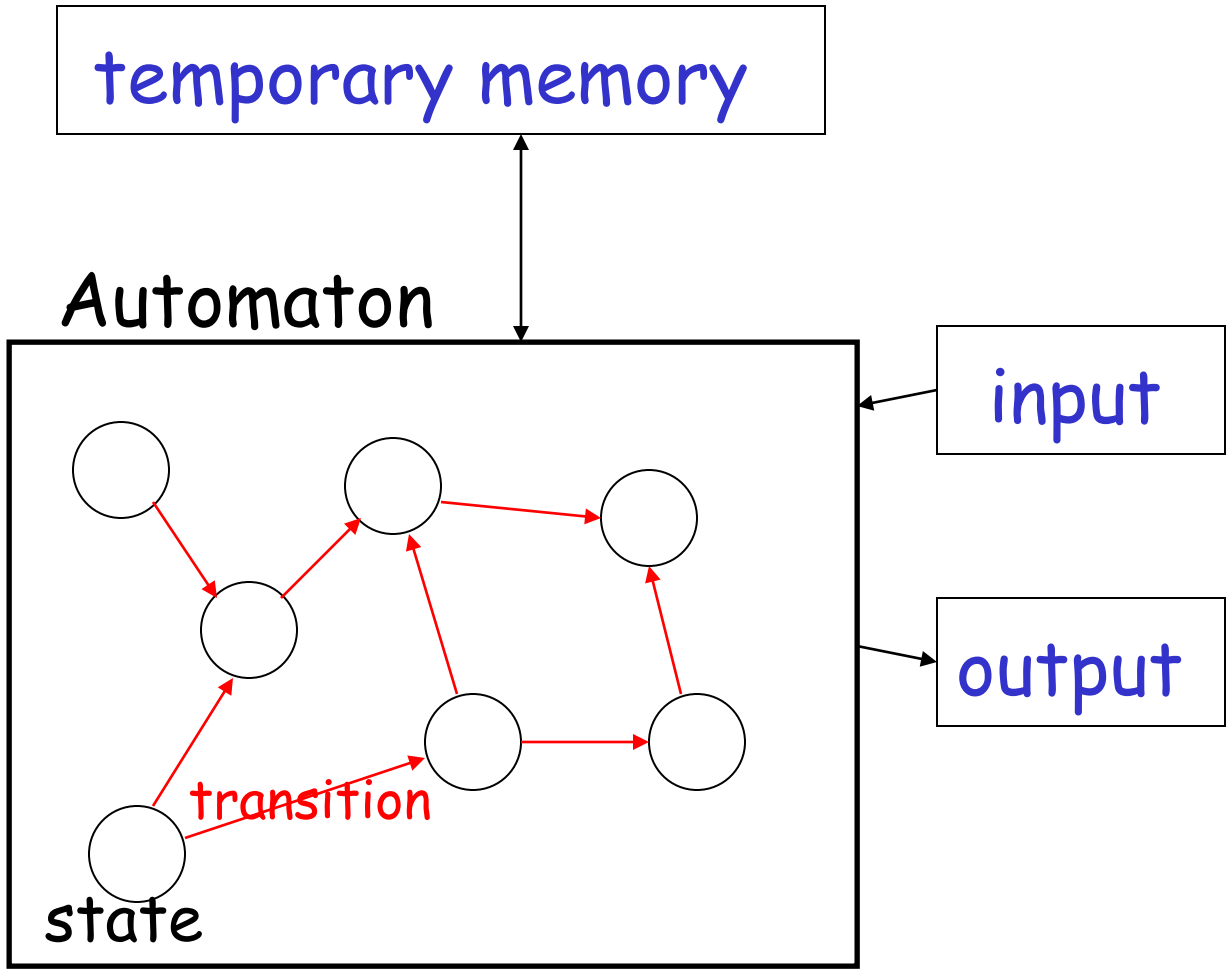
$f(x) = x^3$


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
input
CPU
output
Program memory

Automaton

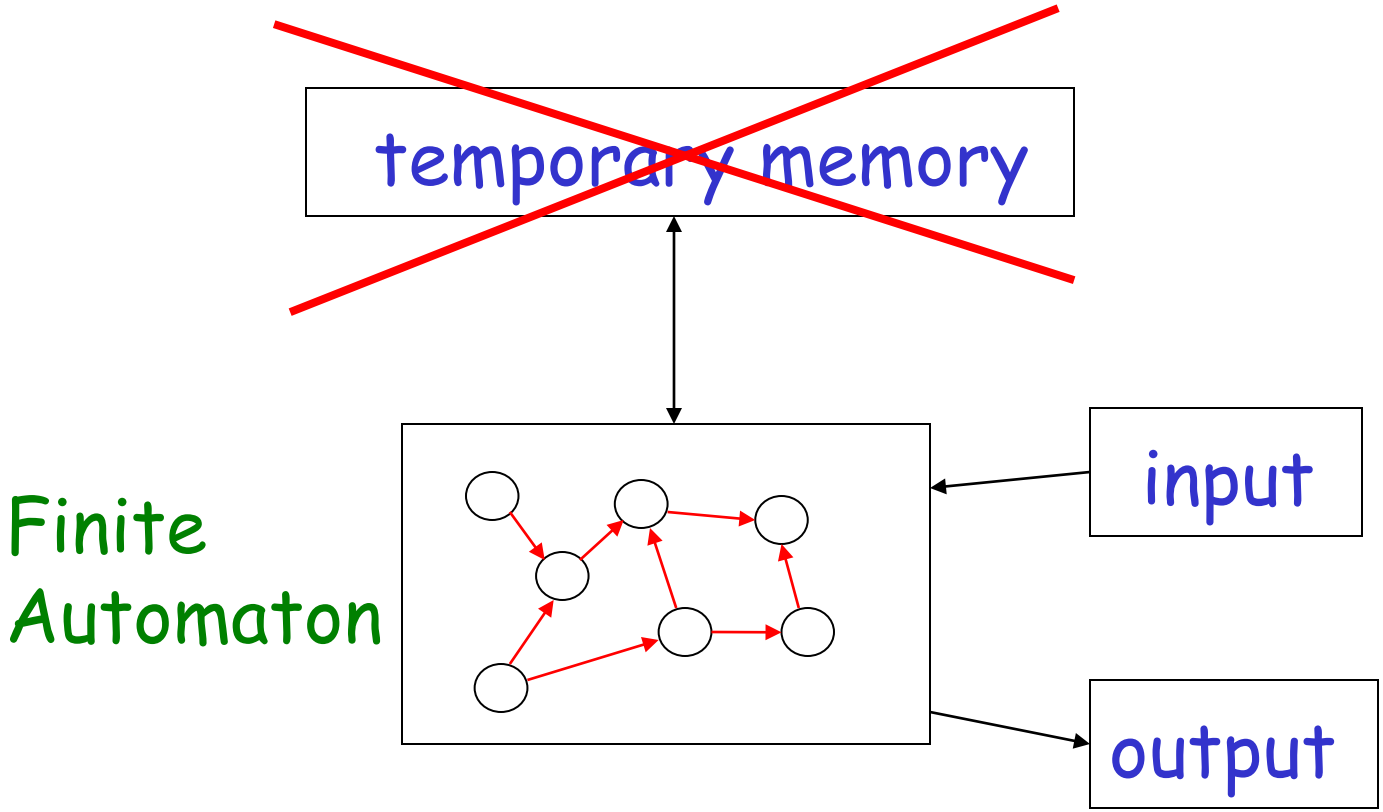

temporary memory
Automaton
input
output
transition
state

Different Kinds of Automata

Automata are distinguished by the temporary memory

- **Finite Automata:** no temporary memory
- **Pushdown Automata:** stack
- **Turing Machines:** random access memory

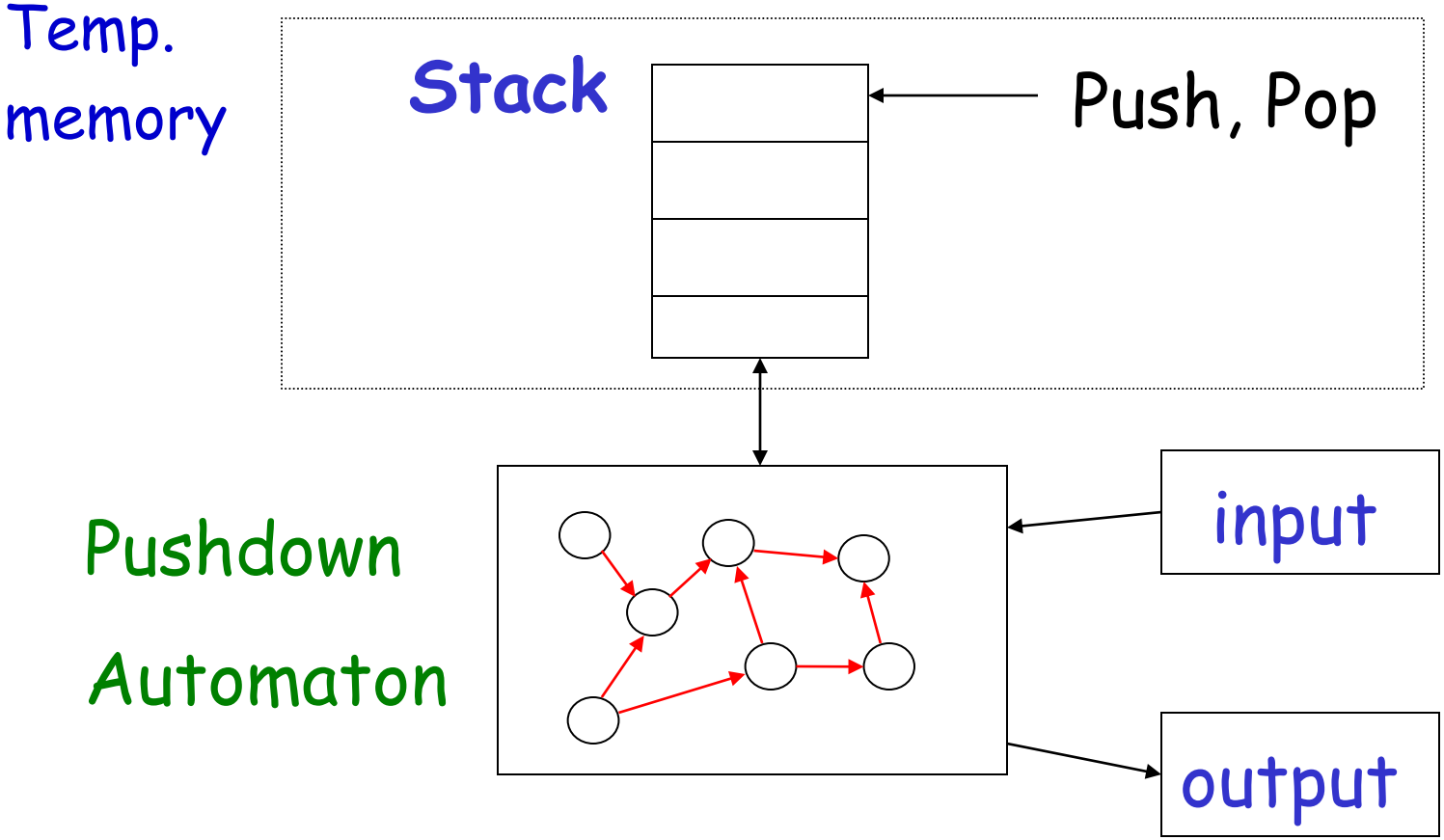
Finite Automaton



Example: Elevators, Vending Machines

(small computing power)

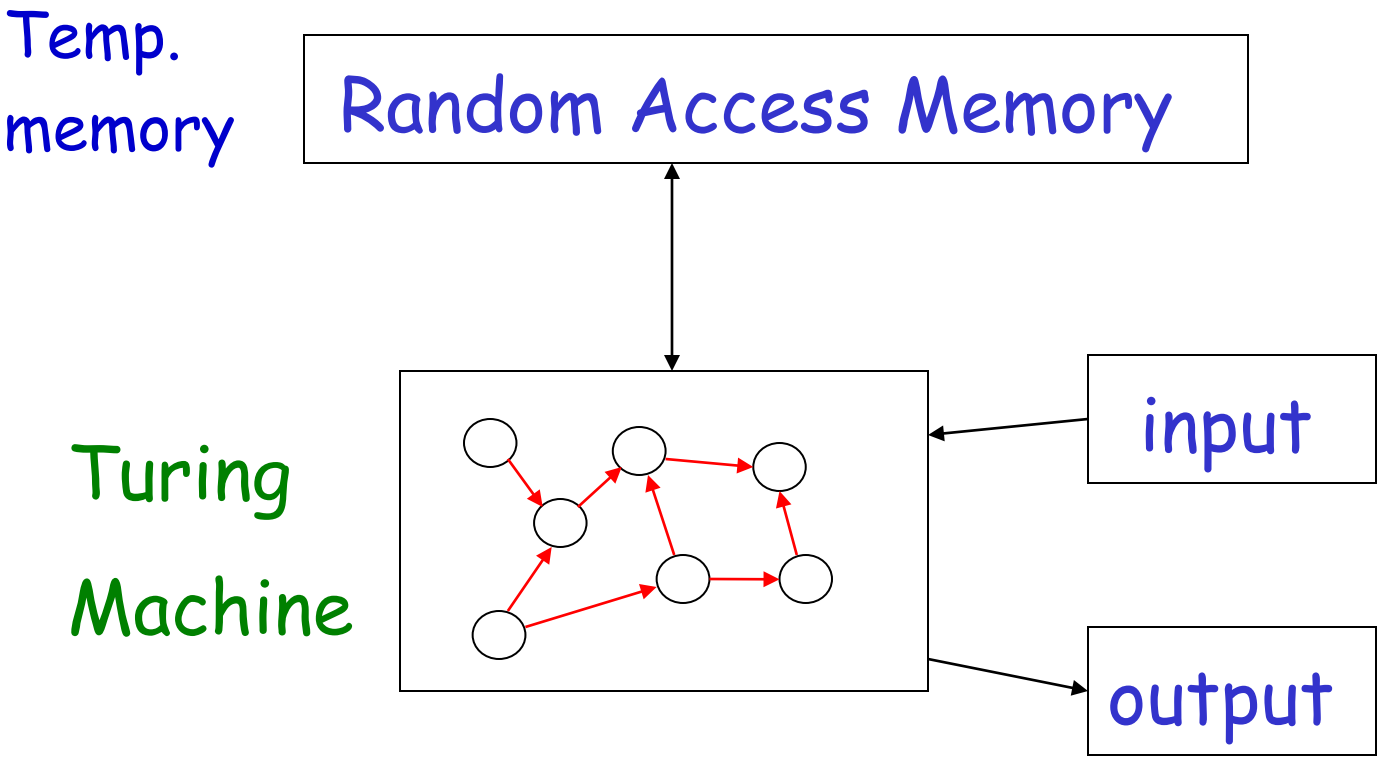
Pushdown Automaton



Example: Compilers for Programming Languages

(medium computing power)

Turing Machine



Examples: Any Algorithm

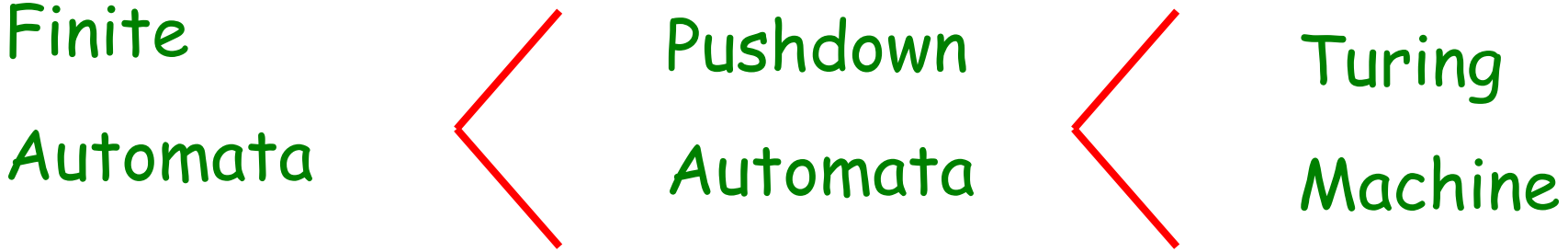
(highest computing power)

Power of Automata

Simple problems

More complex problems

Hardest problems



Less power

More power

Solve more computational problems

Turing Machine is the most powerful computational model known

Question: Are there computational problems that a Turing Machine cannot solve?

Answer: Yes (unsolvable problems)