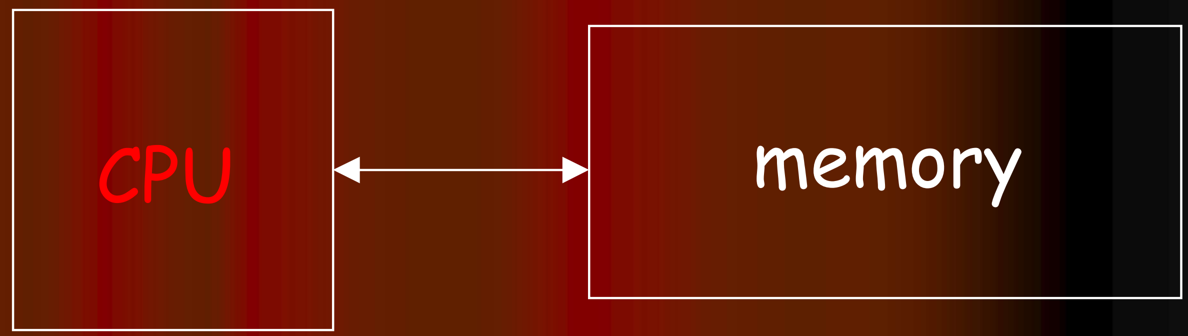
Theory of Automata & Formal Languages

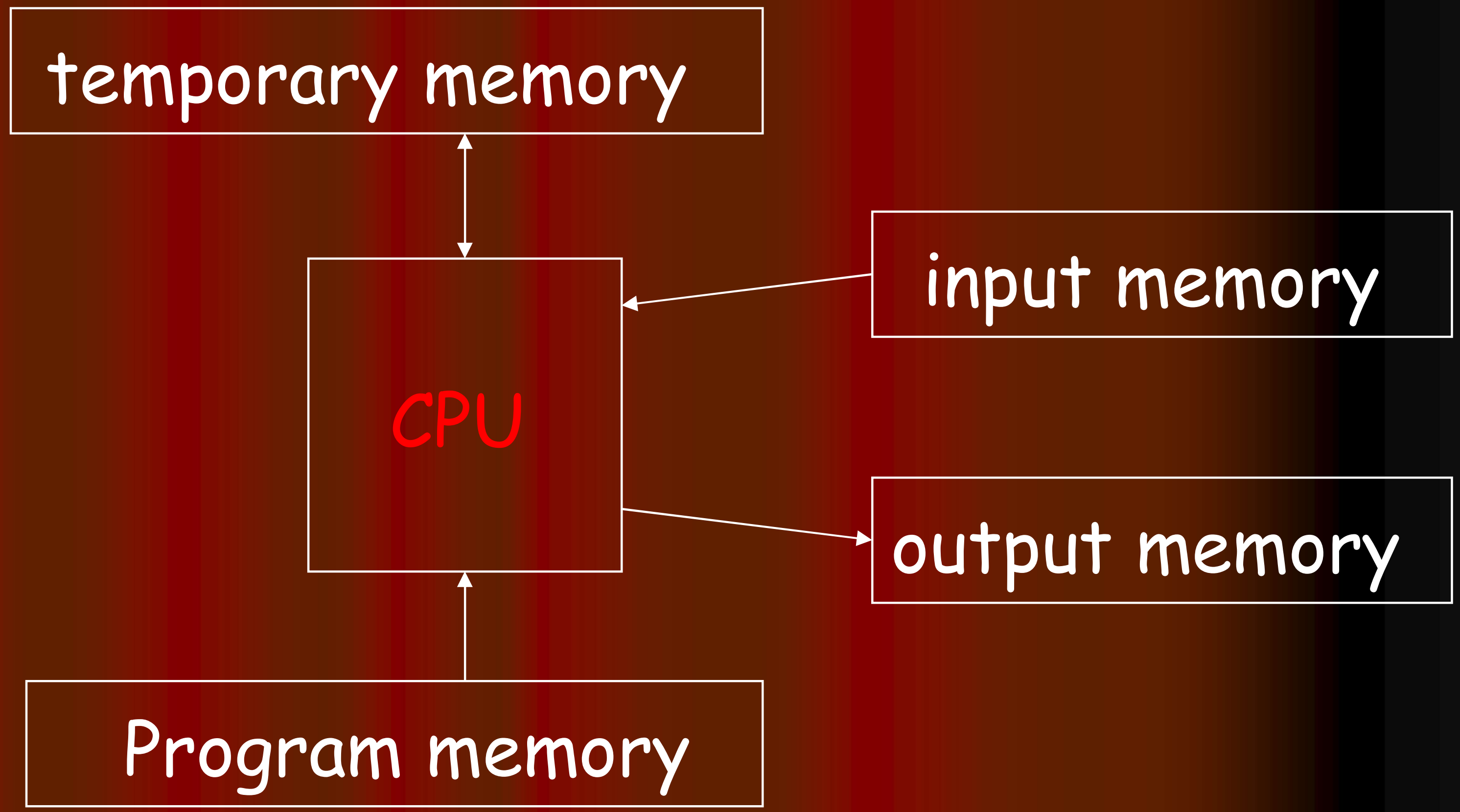
(Theory of Computation)

Compiled By

Prof. M. S. Bhatt

Computation




temporary memory
input memory
CPU
output memory
Program memory

Theory of Computation

Computability	- *What can be computed?* - *Can a computer solve any problem, given enough time and disk-space?*
Complexity	- *How fast can we solve a problem?* - *How little disk-space can we use to solve a problem*
Automata	-*What problems can we solve given really very little space? (constant space)*

Theory of Computation

Computability

Complexity

Automata

What problems can a computer solve?

Not all problems!!!

Eg. Given a C-program, we cannot check if it will not crash!

Verification of correctness of programs is hence impossible!

(The woe of Microsoft!)

Theory of Computation

What problems can a computer solve?

Computability

Complexity

Automata

Even checking whether a C-program will halt/terminate is not possible!

```
input n;
assume n>1;
while (n !=1) {
   if (n is even)
     n := n/2;
  else
     n := 3*n+1;
}
```

No one knows whether this terminates on on all inputs!

17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1.

Theory of Computation

Computability

Complexity

Automata

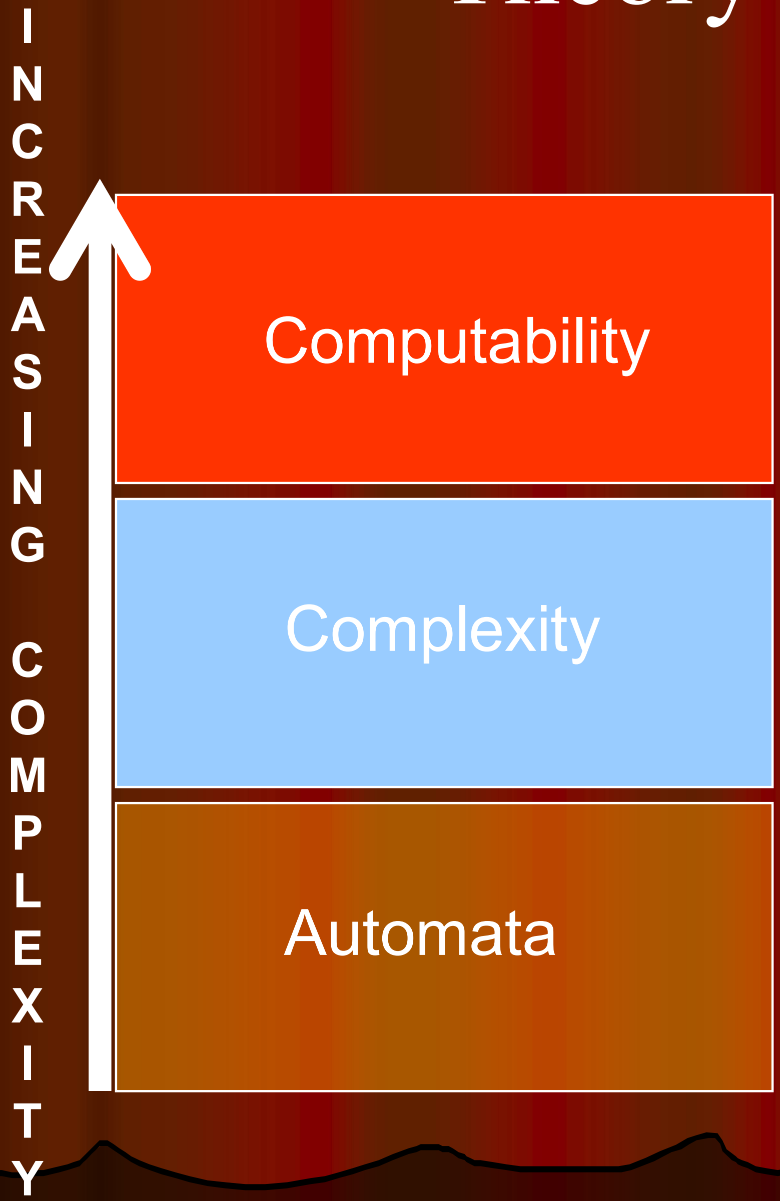
How fast can we compute a function?
How much space do we require?

- **Polynomial time computable**
- **Non-det Poly Time (NP)**
- **Approximation, Randomization**

Functions that cannot be computed fast:

- **Applications to security**
    - **Encrypt fast,**
    - **Decryption cannot be done fast**
- **RSA cryptography, web applications**

Theory of Computation

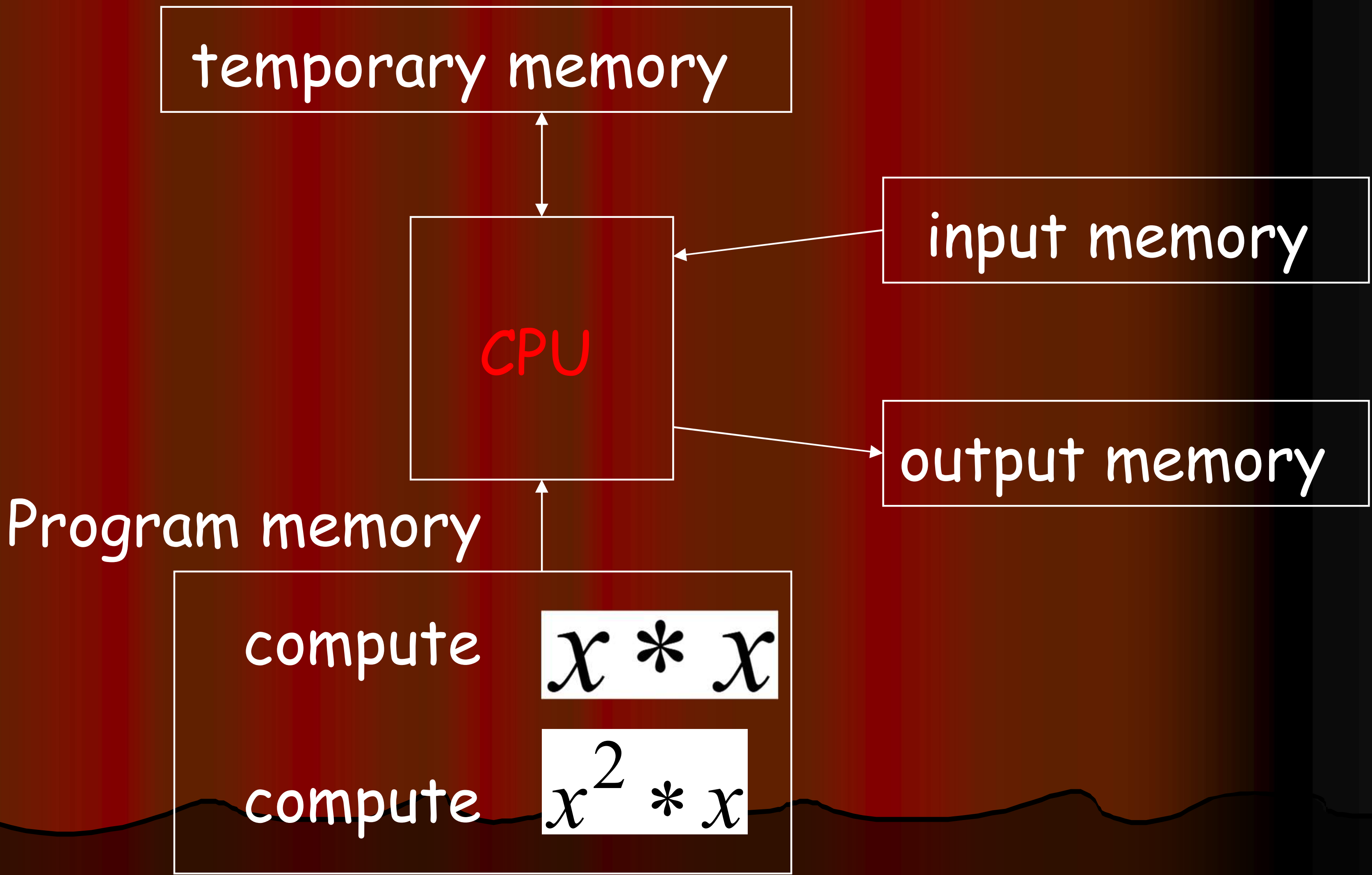


What can we compute?
-- Most general notions of computability
-- Uncomputable functions

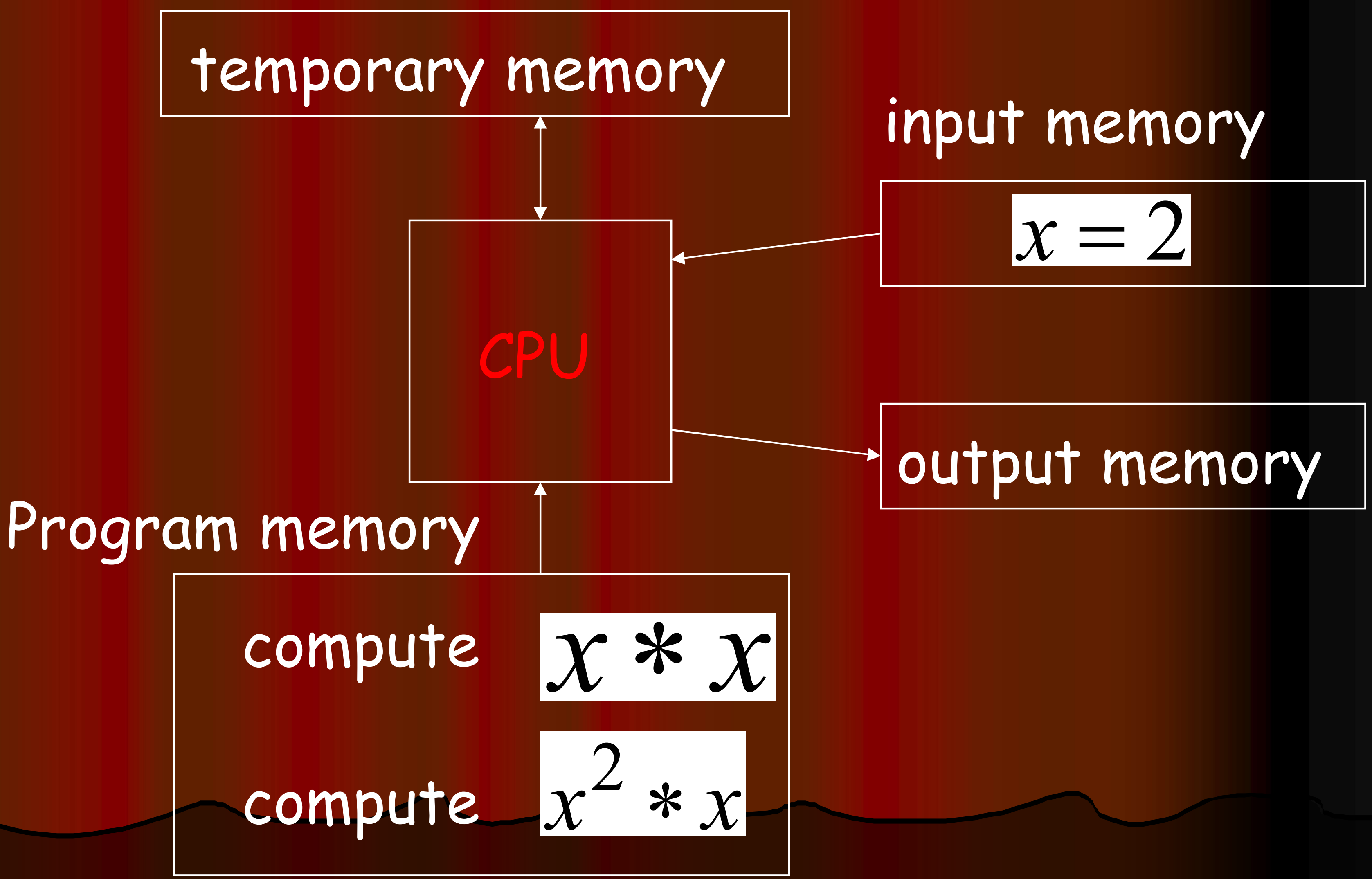
What can we compute fast?
-- Faster algorithms, polynomial time
-- Problems that cannot be solved fast:
* Cryptography

What can we compute with very little space?
-- Constant space (+stack)
* String searching, language parsing, hardware verification, etc.

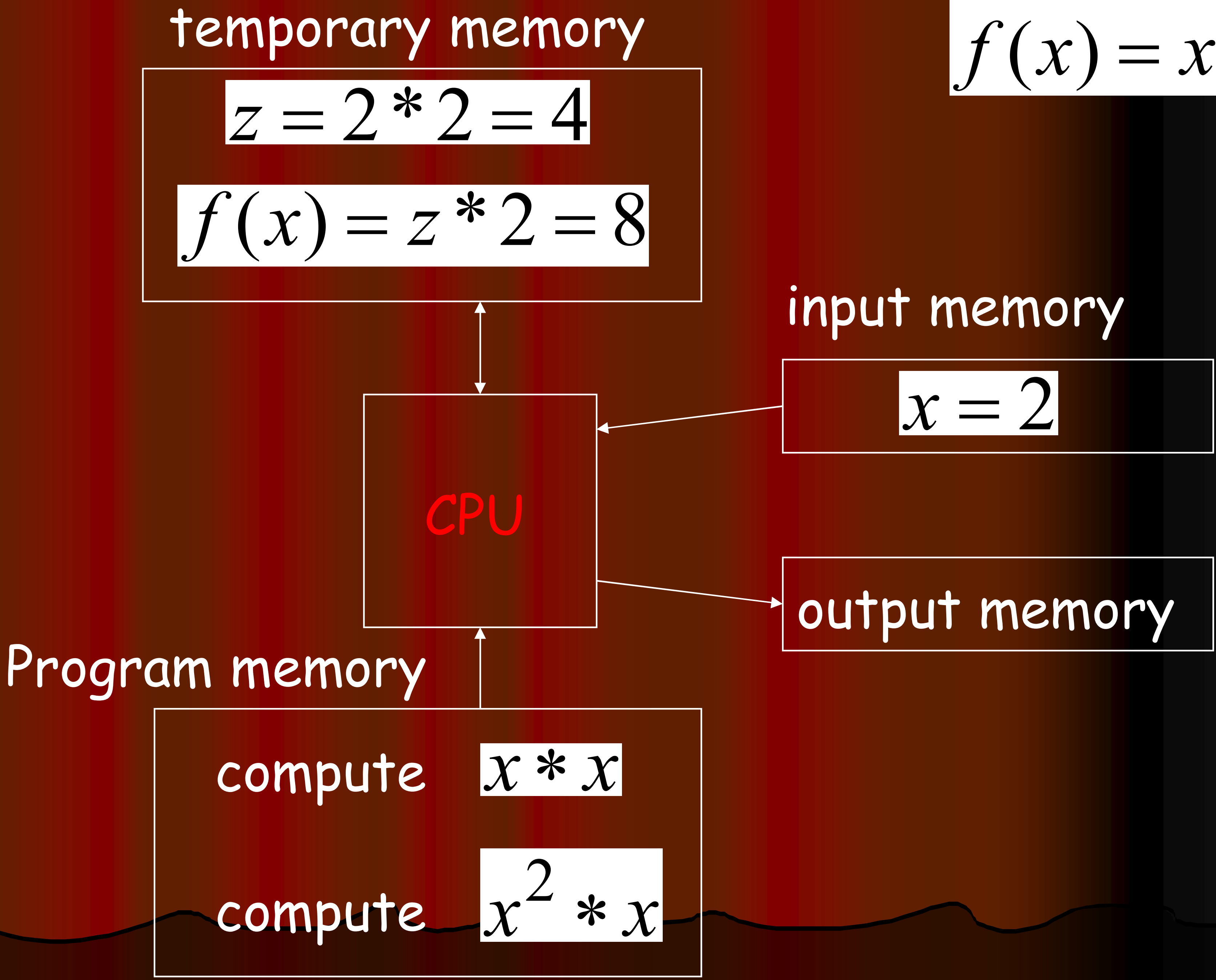
Example: $f(x) = x^3$



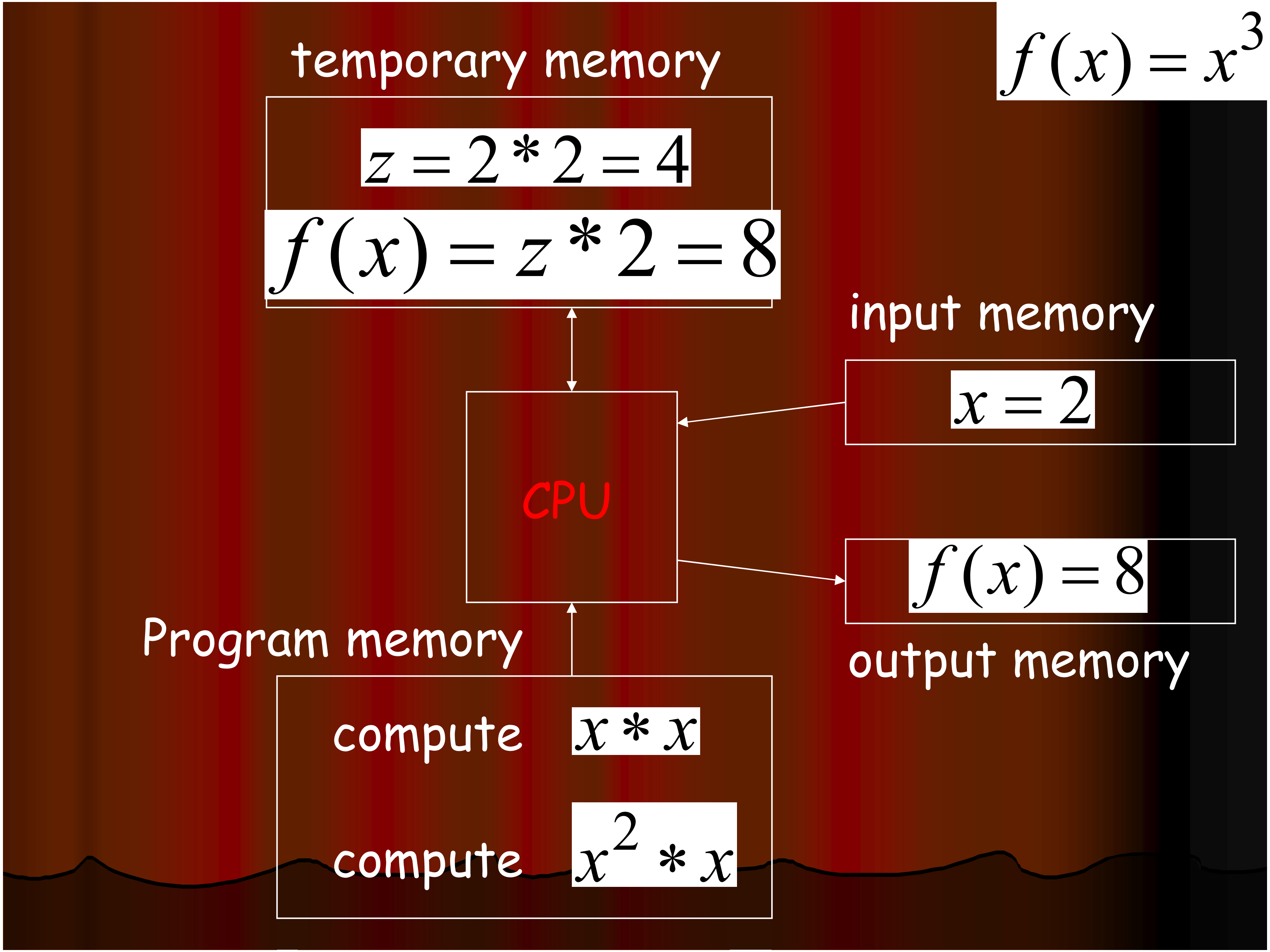
Example: $f(x) = x^3$



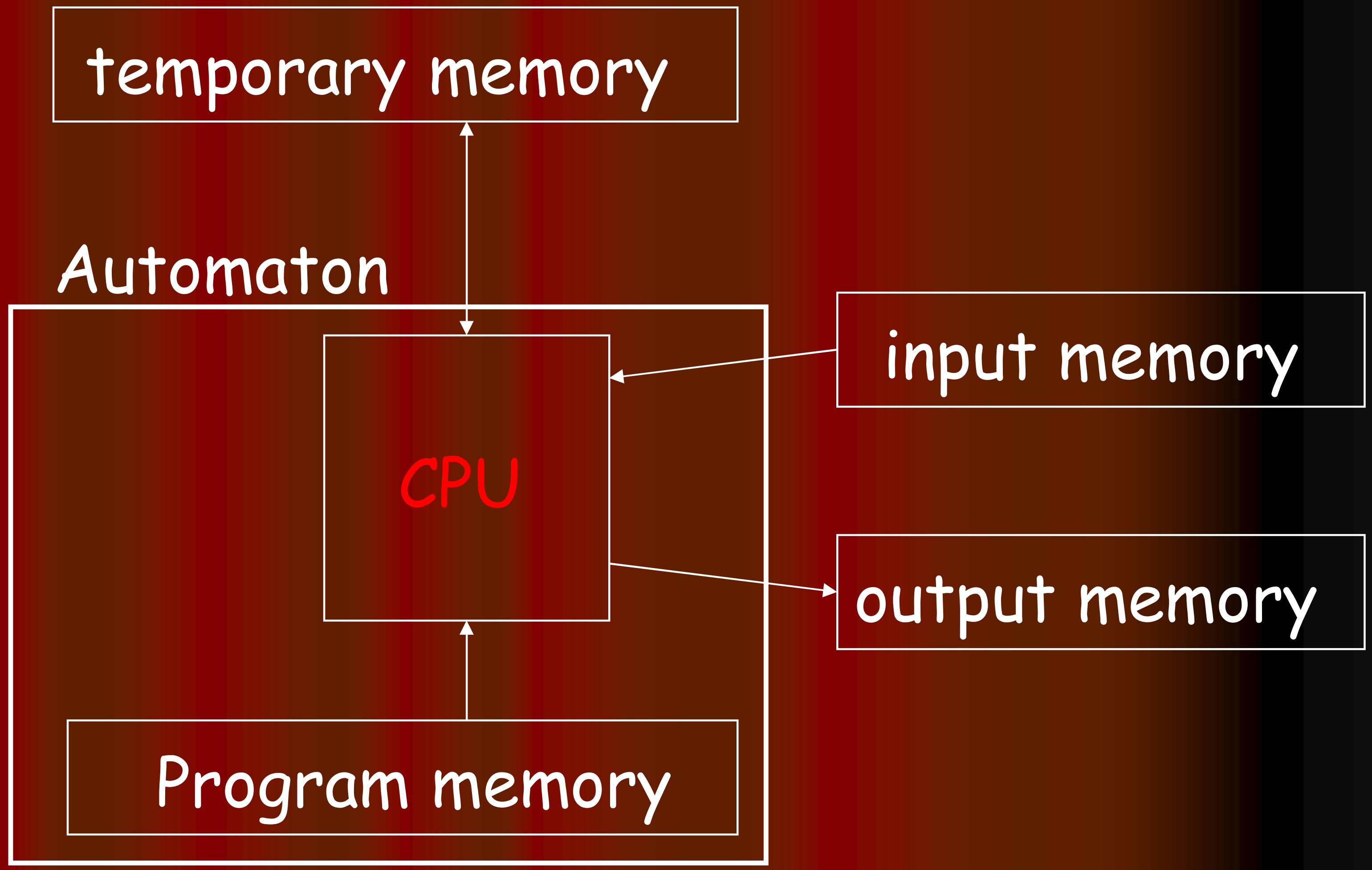
$$f(x) = x^3$$



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

Automaton (Robot/Machine)

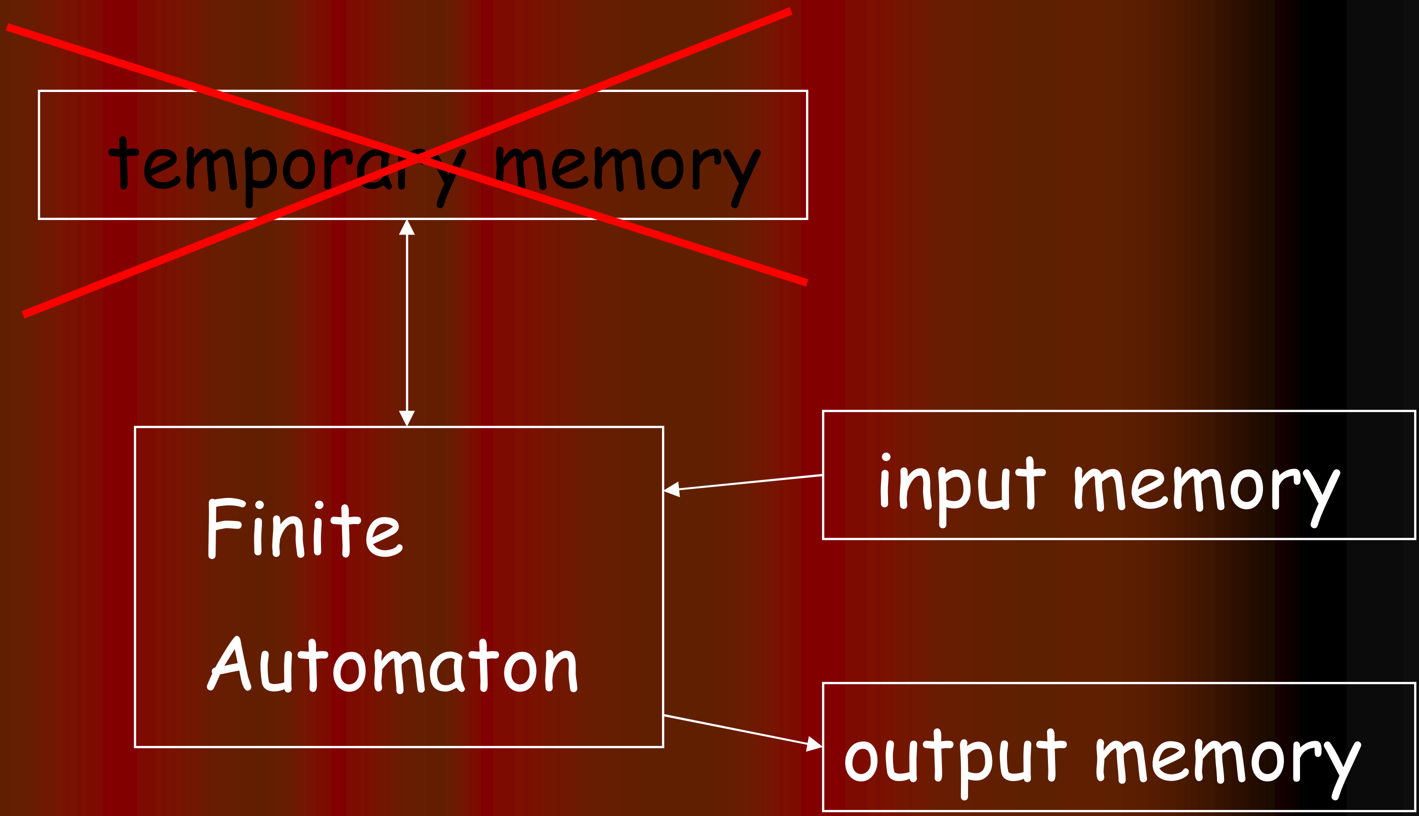


Different Kinds of Automata

Automata are distinguished by the temporary memory

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

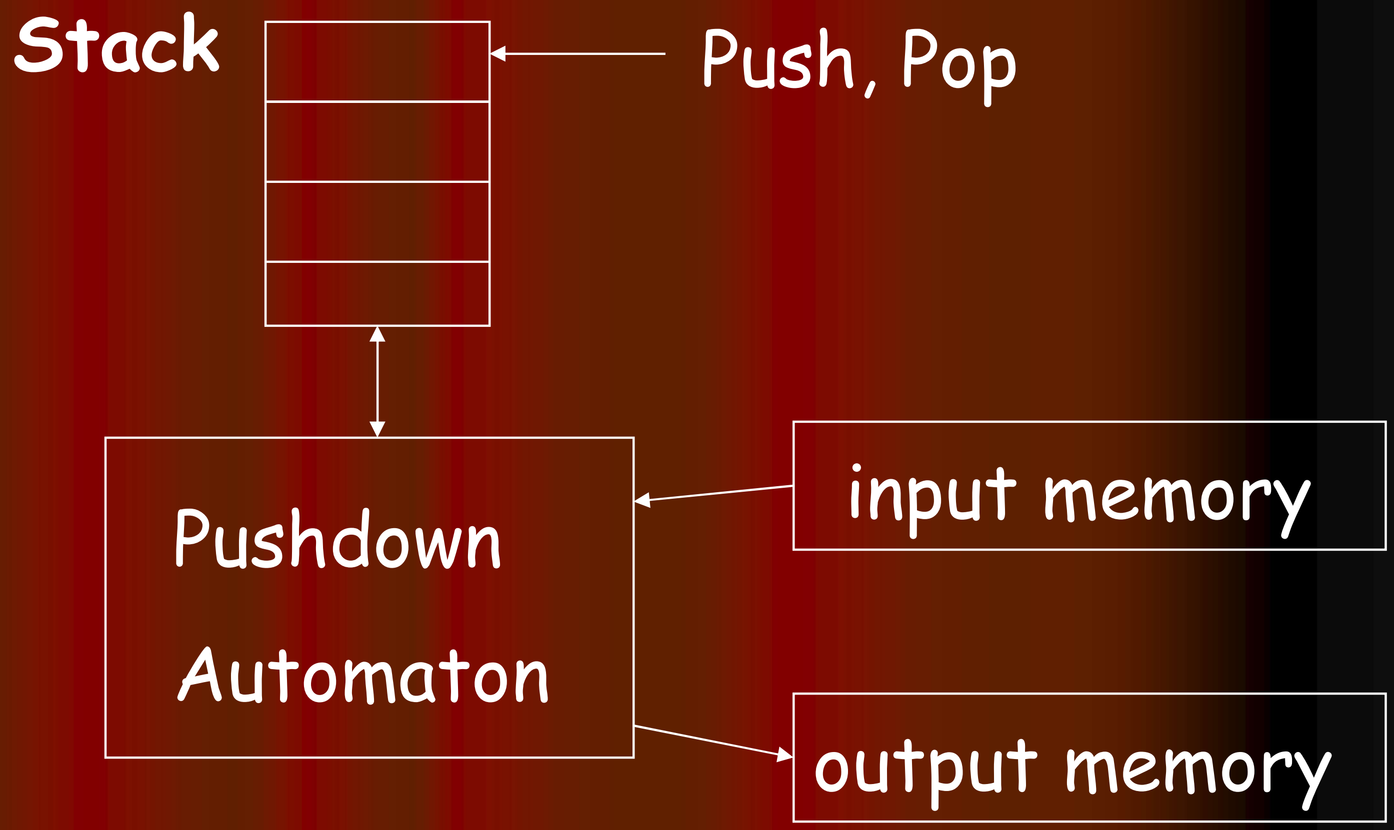
Finite Automaton



Example: Vending Machines

(small computing power)

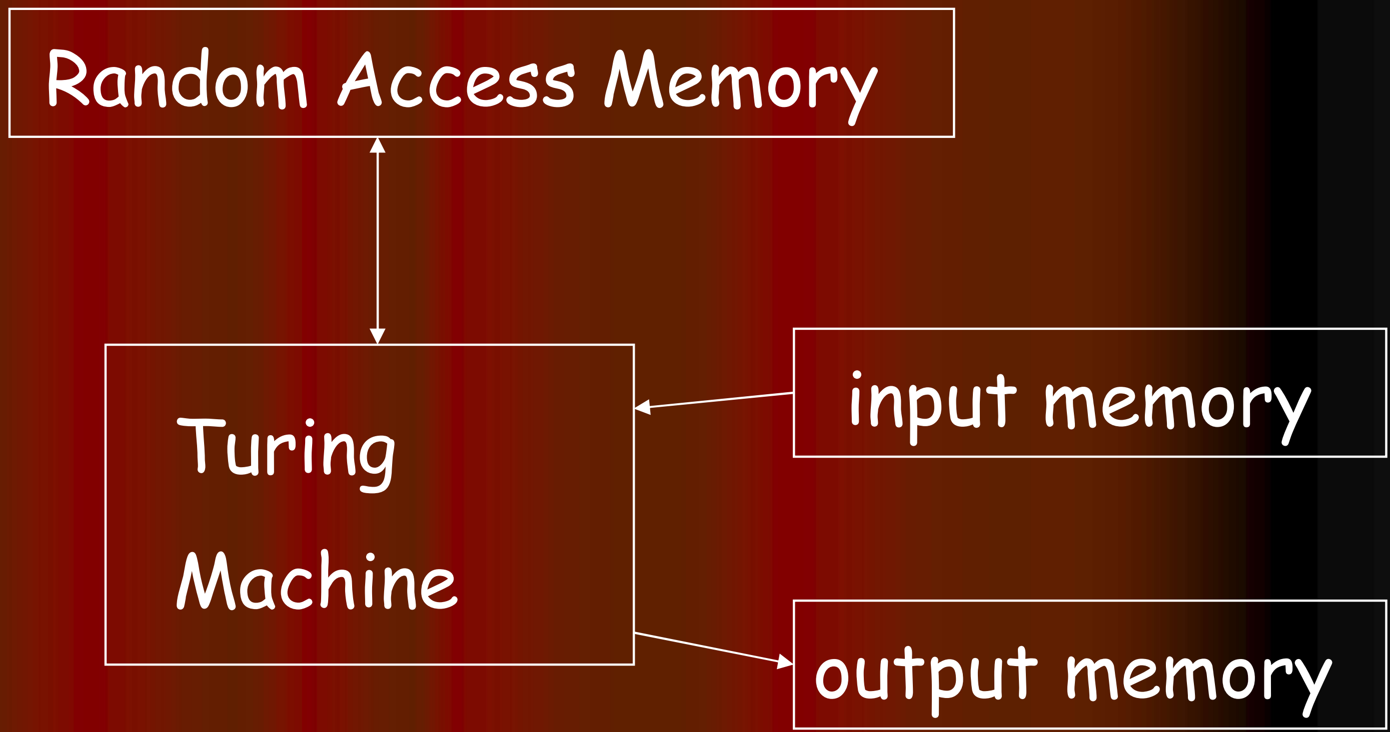
Pushdown Automaton



Example: Compilers for Programming Languages

(medium computing power)

Turing Machine



Examples: Any Algorithm

(highest computing power)

Power of Automata

Finite Automata < Pushdown Automata < Turing Machine

Less power → More power

Solve more computational problems

Formal Language

It is a restricted language with limited features in terms of :

- Input Alphabet
- Operations
- Memory

Formal Language Examples

- Regular Language
- Context- Free Language
- Context- Sensitive Language
- Phase Structure Language

- A language is a set of strings
- String: A sequence of letters
  - Examples: "cat", "dog", "house", …
  - Defined over an alphabet:

$$\Sigma = \{a, b, c, \ldots, z\}$$

Alphabets and Strings

- We will use small alphabets: $\Sigma = \{a, b\}$

- Strings

$$a$$
$$ab$$
$$abba$$
$$baba$$
$$aaabbbaabab$$

$$u = ab$$
$$v = bbbaaa$$
$$w = abba$$

Mathematical Preliminaries

- Sets
- Logic
- Functions
- Relations
- Proof Techniques (Mathematical Induction etc.)