Introduction

Methods	Percentage
Assignments	20
Quizzes	20
Group project	20
Final Assessment	40
TOTAL	100

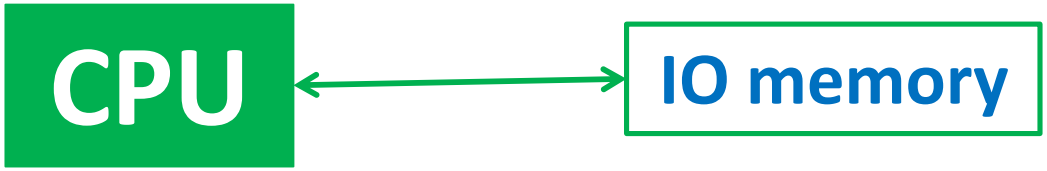
What Is the TA about?

- **Automata Theory** constitute the **theoretical foundation** of computer science (the study of abstract computing devices, or "machines") and covers several topics: **formal languages**, **grammars**, **automata**, **computability** and **complexity**.
- **Languages**, **grammars** and **automata** provide **abstract models** of computers and computation.
- **Computability** and **complexity** determine what can be computed in **principle** what can be computed in **practice**.

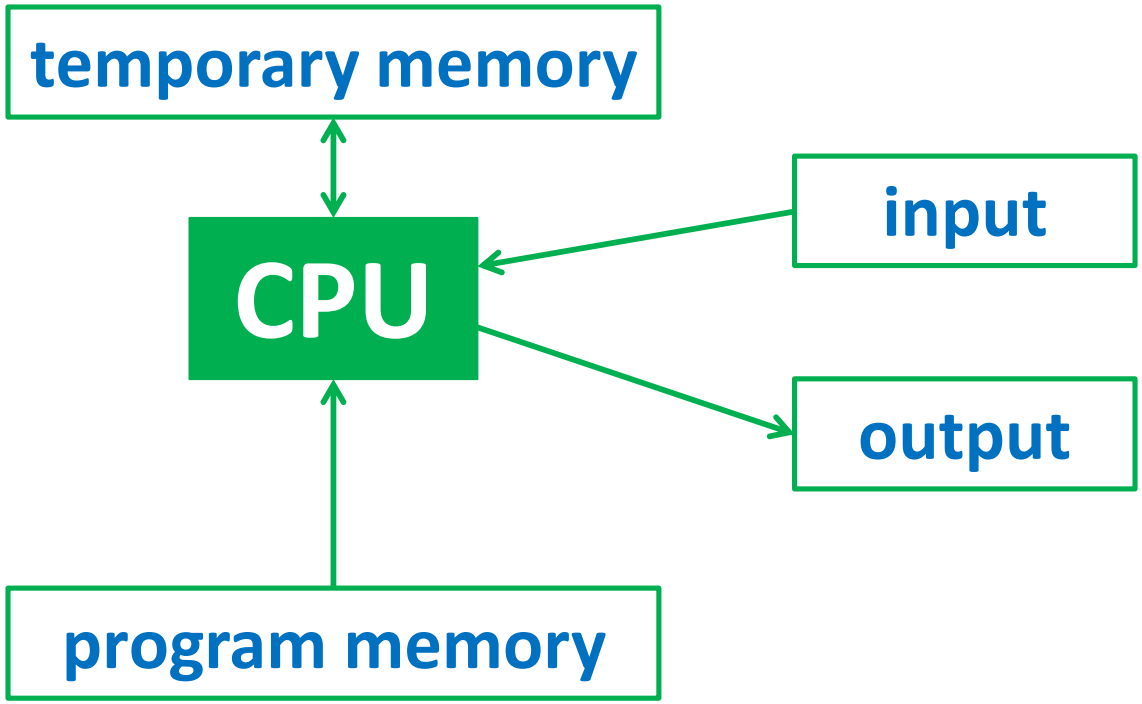
Why Is the TA Useful?

- Direct **application** to creating
  - **compilers, lexical analyzers, programming languages,**
  - **HTML, natural language processing, security,**
  - **designing applications, etc.**
- **Formal framework** to analyze new types of computing devices, e.g. **bio-computers** or **quantum computers**.
- Develop **mathematically mature** computer scientists capable of **precise** and **formal reasoning**!

Computation Model



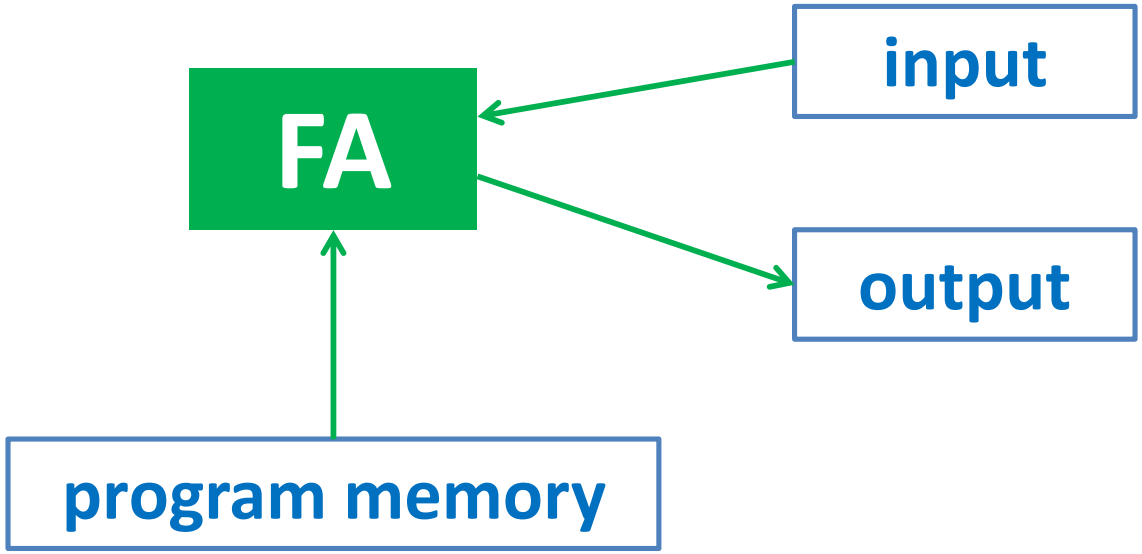
Computation Model



Automata

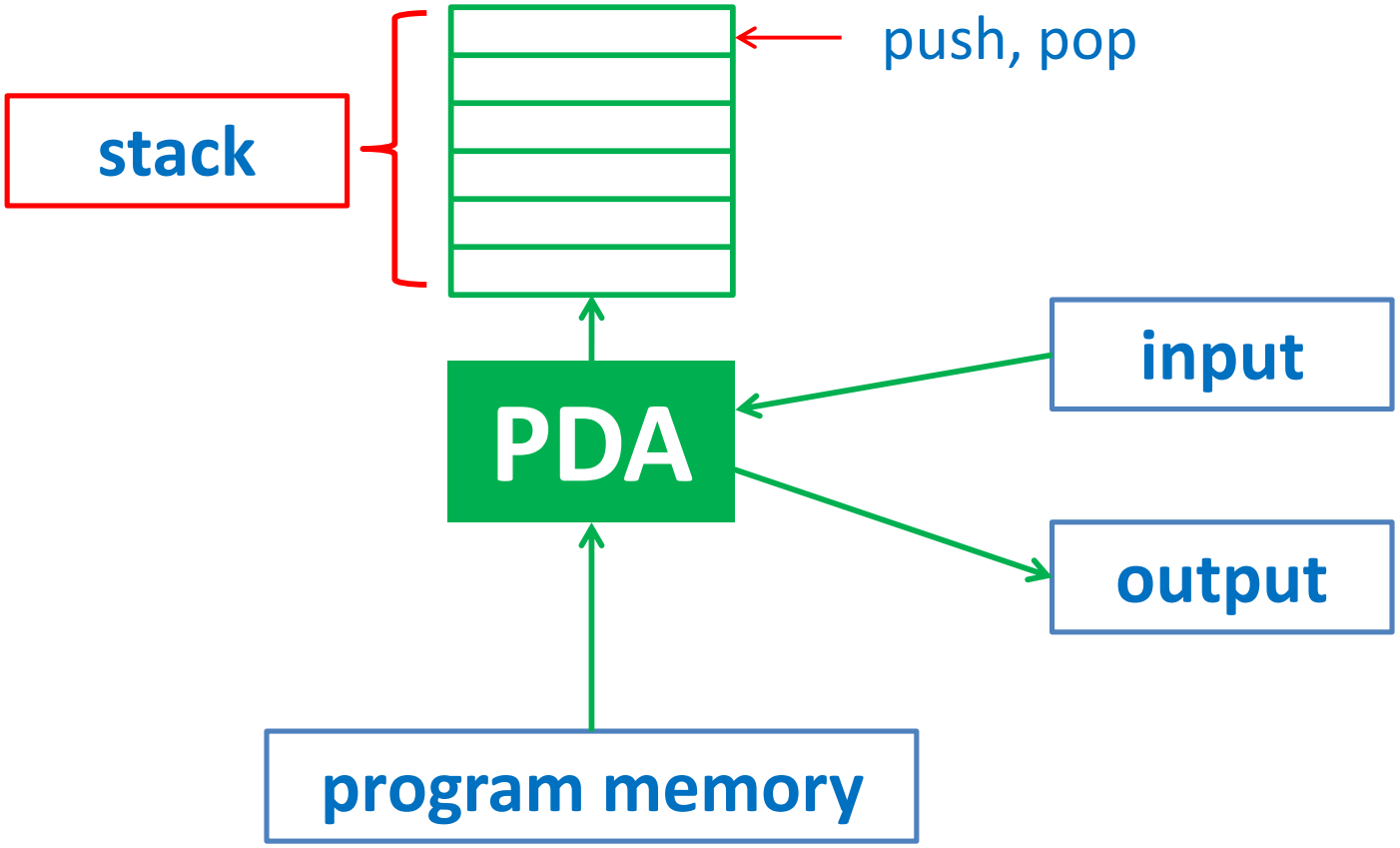
- **Automata** are distinguished by the **temporary memory**
  - **Finite Automata**: **no temporary memory**
  - **Pushdown Automata**: **stack**
  - **Turing Machines**: **random access memory**

Finite Automata



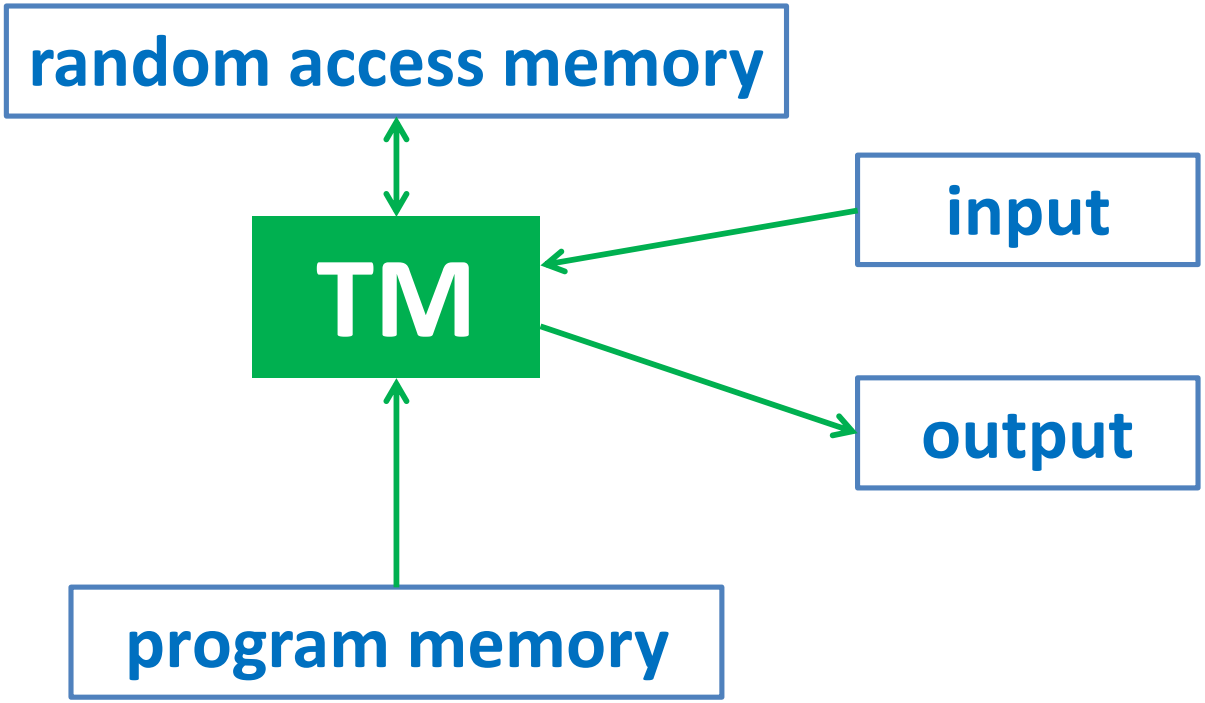
Example: **Vending Machines** (small computing power)

Pushdown Automata



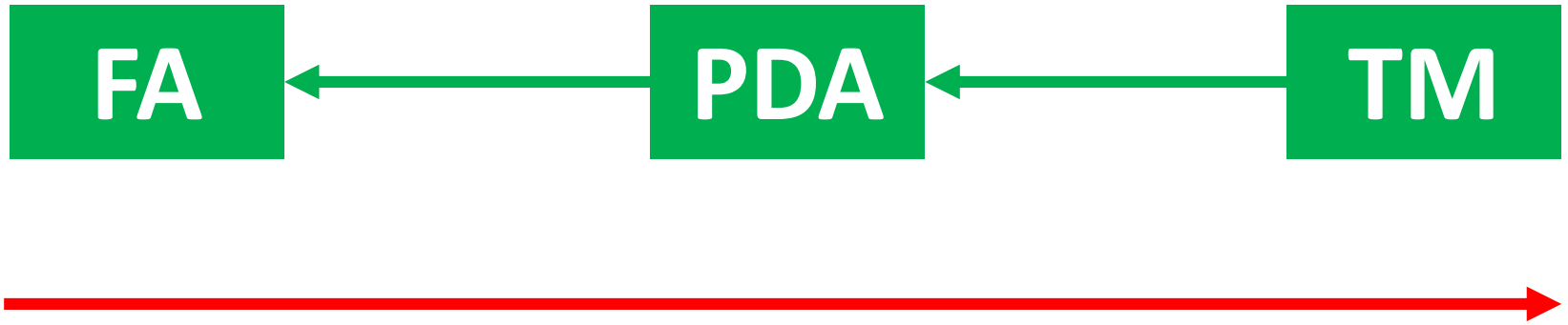
Example: **Compilers for Programming Languages**

Turing Machines



Example: **Any Algorithm** (highest computing power)

Computational Power



Solve more computational problems