

# 形式语言与自动机理论

Formal Languages and Automata Theory

2023 年

# 1.1 课程介绍



## 课程性质：

- 专业基础；
- 计算理论：研究理论计算机的科学。理论计算机研究计算机的理论模型，研究计算机的本征，把计算机看成一个数学系统进行研究。

## 研究内容：

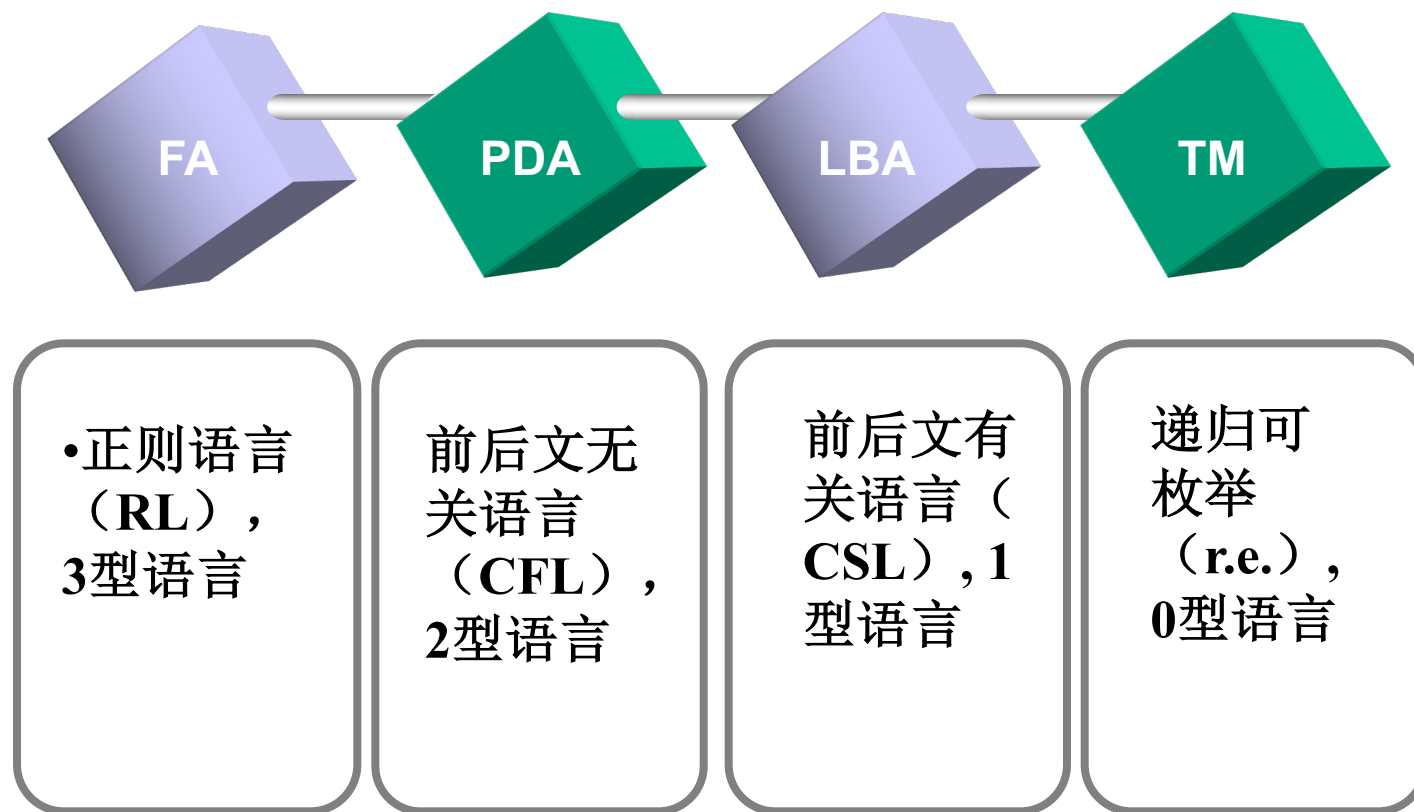
- 自动机与语言 (Automata Theory)
- 可计算性 (Computability Theory)
- 计算复杂性 (Complexity Theory)

# 计算理论的发展简史



- 1900, David Hilbert, 23个数学问题 数学完备吗？数学一致吗？数学可判定吗？
- 1931, 哥德尔不完备定理
- 1936, Alan Turing, On Computable Numbers, with an Application to the Entscheidungsproblem 中提出“图灵机”的设想，美国数学家 Alonzo Church 提出  $\lambda$  演算，Church-Turing 论题，通用图灵机
- 1937, Atanasoff-Berry Computer, 简称ABC计算机，不可编程，且非图灵完全。
- 1946, ENIAC (Electronic Numerical Integrator And Computer), 世界上第一台通用计算机，图灵完备的电子计算机，能够编程。
- 1951, EDVAC (Electronic Discrete Variable Automatic Computer), 与ENIAC不同，EDVAC采用二进制，而且是一台冯·诺伊曼结构的计算机。
- 1950, Chomsky 文法体系
- 1960s, 计算复杂性
- 1970s, NP完全问题（千禧年大奖难题，Millennium Prize Problems），P vs NP,

# 自动机理论 (Automata Theory)



FA: 有穷自动机, PDA: 下推自动机, LBA: 线性有界自动机, TM: 图灵机, 它们都是**计算模型**。

# 图灵与图灵机 (Turing Machine)

## □ On computable Number, 1936

- 这篇奠基之作其实是回答德国大数学家David Hilbert在世界数学家大会上提出的“23个数学难题”中的一个问题：“是否所有的数学问题在原则上都是可解的”
- 图灵认为“有些数学问题是不可解的”
- 图灵机只是在这篇论文的一个脚注中顺便提出的



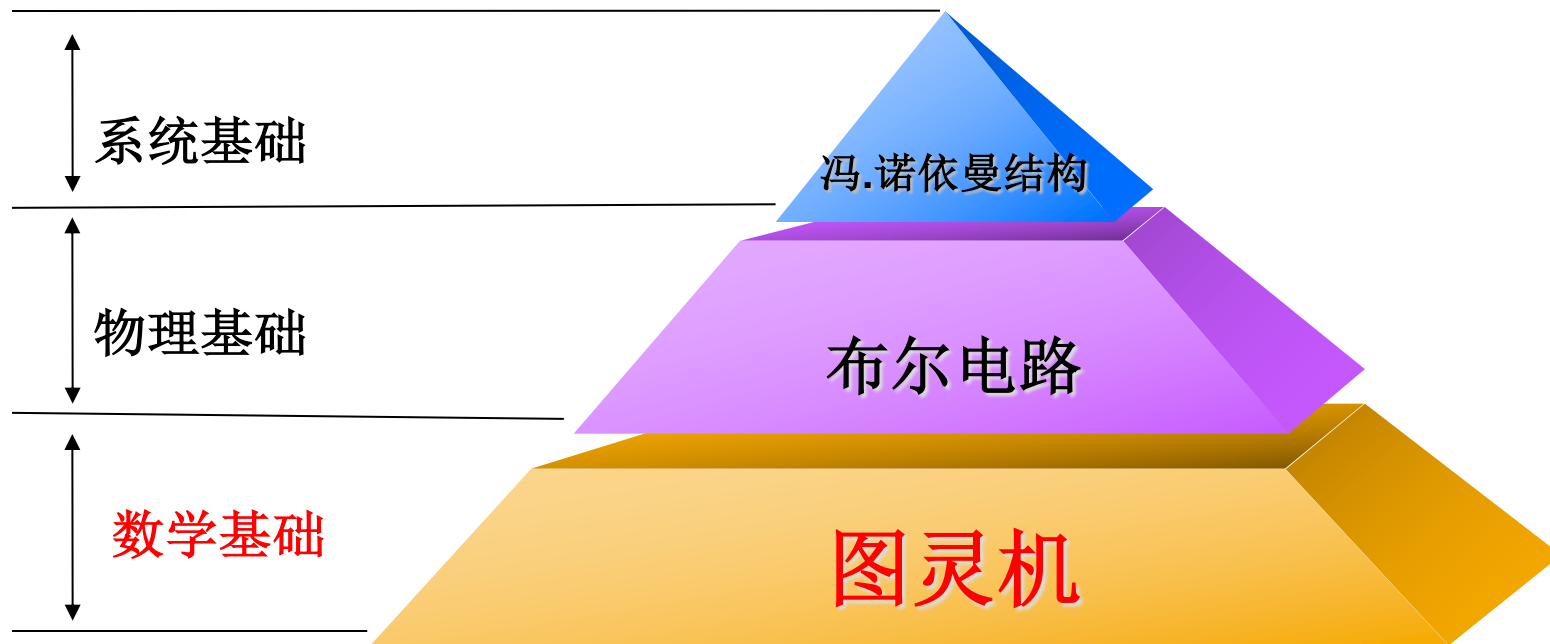
### Endnotes

8. It is most natural to construct first a choice machine (§2) to do this. But it then easy to construct the required automatic machine. We can suppose that the choices are always choices between two possibilities 0 and 1. Each proof will then be determined by a sequence of choices  $i_1, i_2, \dots, i_n$  ( $i_1 = 0$  or  $1, i_2 = 0$  or  $1, \dots, i_n = 0$  or  $1$ ), and hence the number  $2^{i_1} + 2^{i_2} + \dots + 2^{i_n}$ , completely determines the proof. The automatic machine carries out successively proof 1, proof 2, proof 3, ....

# 图灵机与计算机的关系



## ■ 图灵机概念的引入



## 电子计算机的三大基础

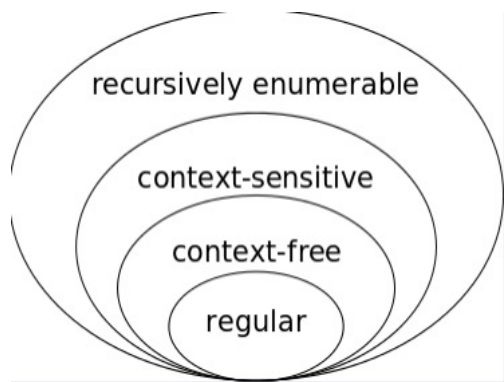
今天所有的计算机，都是图灵机的实例，都建立在冯.诺依曼结构之上，都由若干电子器件组合而成的。

# 形式语言 (Formal Language)



Avram Noam Chomsky (born December 7, 1928) is an American linguist, philosopher, cognitive scientist, historian, social critic, and political activist. Sometimes described as "**the father of modern linguistics**", Chomsky is also a major figure in analytic philosophy, and one of the founders of the field of cognitive science.

研究语言**语法**的数学和计算机科学分支叫做**形式语言理论**，不致力于语言的**语义**研究。



**乔姆斯基文法体系**

- 1956年，Chomsky，从**语言产生**的角度，定义了语言与文法；
- 1951-1956，Kleene提出了有穷状态自动机（FA），从**语言识别**的角度，定义了语言；
- 1959年，Chomsky证明了语言与自动机的等价性，**形式语言**从此诞生；



## 2. Computability Theory

In computability theory, the classification of problems is by those that are **solvable** and those that **are not**. (**solvable means computable**)

## 3. Complexity Theory

In complexity theory, the objective is to classify problems as **easy** and **hard** ones.



# 课程特点与应用



## 课程特点：

- 高度抽象和形式化---**计算思维能力**
- 不易理解，难于联系实际

## 典型应用：

- 程序语言与设计
- 编译理论与技术
- 模式识别（Pattern Recognition）
- 自然语言理解（Nature Language Process）
- 现在密码学

形式语言与自动机理论不仅是计算机学科重要的理论基础，有着广泛的应用，而且非常有利于培养计算机学科人员的**计算思维能力：问题的形式化和模型化描述、抽象思维能力、逻辑思维能力。**

# 课程考核与参考教材



## 课程考核：

1. 平时作业占30分，共10次作业；
2. 出勤占10分，随机点名5次或10次；
3. 期末考试占60分，采用**闭卷**形式；

## 设计软件：

- JFLAP: <http://www.jflap.org>
- 大部分作业，要求使用JFLAP完成

## 参考教材：

1. (美) 霍普克罗夫特等著, 孙家骅译, 自动机理论、语言和计算导论, 北京: 机械工业出版社, 2022
2. Michael Sipser, Introduction to the Theory of Computation(Third Edition), Cengage Learning, 2013

## 1.2 Mathematical Notions and Terminology



### 集合

- Union:  $A \cup B$
- Intersection:  $A \cap B$
- Complement:  $\bar{A}$
- Cartesian Product:  $A \times B$
- Power set:  $P(A)$  或  $2^A$  幂集、超集

# 1.2 Mathematical Notions and Terminology

$\Sigma = \{0,1\}$  字母表

$\Sigma \times \Sigma = \{(0,0), (0,1), (1,0), (1,1)\}$

Example:  $A = \{x,y\}$

$P(A) = \{S \mid S \subseteq A\}$

$= \{\{\}, \{x\}, \{y\}, \{x,y\}\}$

一切子集的集合

其中  $\{\} = \Phi$

设  $A = \{x_1, x_2, \dots, x_n\}$ ,

子集编码 0,0,0,0,0

0,0,0,0,1

0,0,0,1,0

1,1,1... 1 共  $2^n$  个

Note the different sizes:

$|P(A)| = 2^{|A|}$  个数是幂, 名称的来源

$|A \times A| = |A|^2$

## 1.3 Strings and Languages



定义 1. 字母表: 符号的**有穷非空**集合, 用  $\Sigma$  表示。

例 1.1  $\Sigma = \{a, b, \dots, z\}$ ,  $\Sigma = \{0, 1\}$ 。

定义 2. 字符串: 从某个字母表中选择的符号的**有穷序列**。

例 1.2 1101001 是从字母表  $\Sigma = \{0, 1\}$  中选出的串。

**注:**空串记为  $\varepsilon$ , 在软件JFLAG中, 记作 $\lambda$ 。

定义 3. 串的长度: 串中符号的位数。串  $w$  的长度记为  $|w|$ 。

例 1.3  $|010|=3$ ,  $|\varepsilon|=0$ .

## 1.3 Strings and Languages



**定义 4. 字母表的幂：** 如果  $\Sigma$  是一个字母表，则用指数记号来表示这个字母表某个长度的所有串的集合。即  $\Sigma^k$  是长度为  $k$  的串的集合，这些串的每个符号都属  $\Sigma$ 。

**例 1.4**  $\Sigma = \{0,1\}$ , 则

$$\Sigma^1 = \{0,1\},$$

$$\Sigma^2 = \{00,01,10,11\},$$

$$\Sigma^3 = \{000,001,010,011,100,101,110,111\}$$

**注意：**

$\Sigma$  和  $\Sigma^1$  的区别：  $\Sigma$  是字母表，其元素是**符号**；  $\Sigma^1$  是串的集合，其元素是**串** 0 和 1，每个串的长度为 1。

## 1.3 Strings and Languages



定义5. 克林闭包 (Kleene Closure)

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i$$

约定  $\Sigma^0 = \{\varepsilon\}$ ,  $\varepsilon$  是长度为 0 的唯一的串。  $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$ , 即字母表  $\Sigma$  上所有串的集合。

定义6. 正闭包 (Positive Closure)

$$\Sigma^+ = \bigcup_{i=1}^{\infty} \Sigma^i$$

显然:  $\Sigma^* = \Sigma^0 \cup \Sigma^+$

## 1.3 Strings and Languages



定义 7. 语言: 若  $\Sigma$  是一个字母表,  $L \subset \Sigma^*$ , 则  $L$  是  $\Sigma$  上的语言。  
(语言就是字符串的集合)

例1.5

$L = \{ x \mid x \text{ is a bit string with two zeros} \}$

$L = \{ a^n b^n \mid n \in \mathbb{N} \}$

$L = \{ 1^n \mid n \text{ is prime} \}$  每个字符串是素数个1连接而成的

语言连接  $\neq$  笛卡尔积

For example, let  $A = \{0,00\}$  then

$A \bullet A = \{ 00, 000, 0000 \}$  with  $|A \bullet A|=3$ , 连接 一维

$A \times A = \{ (0,0), (0,00), (00,0), (00,00) \}$  叉乘 二维  
with  $|A \times A|=4$



# 1.4 Types of Proof



- **Definitions, Theorems, and Proofs**
- **Type of Proof**
  - Proof by Construction（构造法）
  - Proof by contradiction（反证法）
  - Proof by Induction(归纳法： 整数归纳和结构归纳)
    - Basis:
    - Induction Step
    - Results
  - Deduction Proof（演绎法）

## 1.4 举例：正则语言的泵引论

Show  $E = \{0^i 1^j \mid i > j\}$  is not Regular Language. **证明思路总结**

Step 1: 选择反证法;

Step 2: 构造 string  $s = 0^{p+1} 1^p$ ; 利用泵长度  $p$

Step 3: 发现矛盾

$s = xyz$ , 由泵引论3)  $|xy| \leq p$  知:  $y = 0^k$ ,  $k > 0$ ; **不可能包含 1**  
 $xy^i z = 0^{p-k} 0^{k \cdot i} 1^p = 0^{p+k(i-1)} 1^p$ , 当  $i=2$  时, 显然  $xyyz \in E$ ; **可惜没矛盾, 只好换一条路走**

**Pumping Down**: The pumping lemma states that  $xy^i z \in E$  even if when  $i=0$ , so let's consider the string  $xy^0 z = xz$ .  
**结果怎么样呢?**  
 $xz = 0^p 1^p \notin E$ ; 矛盾出现

Step 4: 得出结论。

## 1.5 思考与总结



1. 集合  $\Phi^0$ 、 $\{\varepsilon\}^0$ 、 $\Phi^*$ 、 $\{\varepsilon\}^*$  分别等于什么？

2.  $\Sigma^*$  一定不等于  $\Sigma^+$  吗？

3.  $\varepsilon.A = A.$   $\varepsilon = A$  ？

4.  $\Phi.A = A.\Phi = \Phi$ ？

5.  $\Sigma^*1\Sigma^* = \{ \omega \mid \quad \quad \quad \}$ .

6.  $(\Sigma\Sigma)^* = \{ \omega \mid \quad \quad \quad \}$ .