

# 形式语言与自动机

罗贵明，高跃  
2022年春

2/22/2022

School of Software



## 课程信息

- ◇ 名称 形式语言与自动机
- ◇ 时间 每周三上午 9:50-12:15
- ◇ 教室 六教 6A214
- ◇ 班级 软件01、02、03班  
及其他同学
- ◇ 时数 48

2/22/2022

School of Software



## 课程信息

### ◇ 计算机、软件相关专业基础课

Computer science as an academic discipline began in the 1960's:

- programming languages, compilers, operating systems, and the mathematical theory

Theoretical computer science covered:

- finite automata, regular expressions, context-free languages, and computability.

-----John Hopcroft, 2020

### ◇ 相关领域及应用

- 语言的形式化表示、软件建模、模型检测
- 任何信息处理和计算的问题都可对应于语言处理问题

2/22/2022

School of Software



## 相关课程

### ◇ 先修课程

- 《离散数学》（数理逻辑，集合论）

### ◇ 后续课程

- 《编译原理》，《嵌入式系统建模与分析》，《软件系统建模与验证》，《模型检测技术》...

### ◇ 其它相关课程

- 《程序设计语言》、《算法分析》、《计算语言学》、《计算复杂性理论》...

2/22/2022

School of Software



## 教师信息

- ◇ 姓名 罗贵明
- ◇ 单位 软件学院
- ◇ 电话 62795440 (O)
- ◇ 办公室 东主楼11-312
- ◇ 电子信箱 gluo@tsinghua.edu.cn

2/22/2022

School of Software



## 教师信息

- ◇ 姓名 高跃
- ◇ 单位 软件学院
- ◇ 电话 62793049 (O)
- ◇ 办公室 东主楼11-319
- ◇ 电子信箱 gaoyue@tsinghua.edu.cn

2/22/2022

School of Software



## 助教信息

- ◇ 姓名 储超群、陈志扬、李傲然、唐诗
- ◇ 单位 软件学院
- ◇ 电话 18810305352 (储)、13161313655 (李)  
15588770792 (陈)、13980055122 (唐)
- ◇ 答疑时间 每周 ?
- ◇ 答疑地点 ?
- ◇ 电子信箱 ccq1993@gmail.com (储)  
1792266893@qq.com (陈)  
lar20@mails.tsinghua.edu.cn (李)  
dujy21@mails.tsinghua.edu.cn (唐)

2/22/2022

School of Software



## 教材信息

- ◇ 书名 Introduction to Automata Theory, Languages, and Computation (Third Edition)
- ◇ 作者 John E. Hopcroft (Cornell)  
Rajeev Motwani (Stanford)  
Jefferey D. Ullman (Stanford)
- ◇ 出版社 Addison Wesley (2007)  
China Machine Press (影印版)



John E. Hopcroft  
Turing Award  
winner in 1986.

2/22/2022

School of Software



John E. Hopcroft is the IBM Professor of Engineering and Applied Mathematics in Computer Science at Cornell University. From January 1994 until June 2001, he was the Joseph Silbert Dean of Engineering. After receiving both his M.S. (1962) and Ph.D. (1964) in electrical engineering from Stanford University. He served as chairman of the Department of Computer. Hopcroft's research centers on theoretical aspects of computing, especially analysis of algorithms, automata theory, and graph algorithms.

He has coauthored four books on formal languages and algorithms with Jeffrey D. Ullman and Alfred V. Aho. His most recent work is on the study of information capture and access.

He was honored with the A. M. Turing Award in 1986. He is a member of the National Academy of Sciences (NAS), the National Academy of Engineering (NAE) and a fellow of the American Academy of Arts and Sciences (AAAS), the American Association for the Advancement of Science, the Institute of Electrical and Electronics Engineers (IEEE), and the Association of Computing Machinery (ACM).

2/22/2022

School of Software

The computer science theory of the last thirty years needs to be extended to cover the next thirty years.

-- John E. Hopcroft

2/22/2022

School of Software

软件理论基础

罗贵明 著

科学出版社

北京

2/22/2022

School of Software

## MOOC课程

学堂在线 《软件理论基础》



2/22/2022

School of Software



## 参考书目

- 教材**Third Edition** 中译本。孙家骢等译 机械工业出版社，北京，2008
- *An Introduction to Formal Languages and Automata* . P. Linz著，机械工业出版社影印，2004
- 《形式语言与自动机导论》(中译本). 孙家骢 等译，机械工业出版社，2005
- 《形式语言与自动机理论》，蒋宗礼，姜守旭 编著，清华大学出版社，北京，2003
- 《形式语言与自动机》，陈有祺编著，机械工业出版社，2008.

2/22/2022

School of Software



## 课程网页

- ✧ 清华大学网络学堂  
<http://learn.tsinghua.edu.cn/>
- ✧ 习题选解等

2/22/2022

School of Software



## 课程计划与进度

### 进度安排（粗略）

- 课程概况及预备知识 3 学时
- 有限状态自动机，正则语言，正则表达式  
第 2, 3, 4 章，约 15 学时
- 上下文无关文法，上下文无关语言，下推自动机  
第 5, 6, 7 章，约 15 学时
- 图灵机，计算问题分类  
第 8 章，约 6 学时；  
第 9, 10, 11 章，约 6 学时
- 机动 3 学时

2/22/2022

School of Software



## 课后练习

- ✧ 随堂布置
  - 以课本中的练习为主  
标记：\*，！，！！
  - 思考题

2/22/2022

School of Software



## 课程方式

### 总评成绩 (100%)

#### 1. 课堂成绩：

- 独立作业 2 次 (20%)  
(大约安排在第六次课、第十一次课)
- 平时纪录 (25%)，  
(其中平时作业15%，课堂10%)
- 期末考试 (55%)

#### 2. 混合成绩

慕课成绩 20%，课堂成绩 80%

2/22/2022

School of Software



## 课程答疑与交流

- ✧ 通过网络
  - 清华网络学堂（师生讨论版）
  - 电子邮件
- ✧ 面对面
  - 时间预约  
第 2 - 16 周上班时间（节假日除外）
  - 地点

2/22/2022

School of Software



## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- 语言基础
- 语言运算

2/22/2022

School of Software

## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- 语言基础
- 语言运算

2/22/2022

School of Software

软件无处不在



软件可以定义世界

2/22/2022

## 概要

软件是一个新型学科，然而其发展史可以追溯到200年前的数学家阿达。



奥古斯塔·阿达·拜伦。计算机程序创始人，建立了循环和子程序概念。为计算程序拟定“算法”，写作的第一份“程序设计流程图”，被珍视为“第一位给计算机写程序的人”。

## 概要

软件是一个新型学科，然而其发展史可以追溯到200年前的数学家阿达。

艾伦·麦席森·图灵 (Alan Mathison Turing)，英国数学家、逻辑学家，被称为计算机科学之父，人工智能之父。



2/22/2022

## 概要

1. 软件的发展飞速，需要什么？
  2. 软件科学与理论是什么？
- 架构  
规范  
计算的判定  
计算的复杂性

2/22/2022

School of Software

## 概要 – 课程内容

正则语言	上下文无关语言	递归可枚举语言	非递归可枚举语言
自动机 DFA NFA 关系 优化	自动机 PDA DPDA 关系	自动机 Turing 扩展 计算	
正则表示 正则文法 语言的性质	上下文无关文法 文法的规范 语言的性质	递归语言 判定性问题 算法的复杂性	
语言的描述	文法和自动机的设计和应用		

2/22/2022

School of Software

19

## 概要

### 参考书

- 罗贵明, 软件理论基础, 科学出版社
- John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman, Introduction to Automata Theory, Languages, and Computation, (Third Edition), Addison Wesley (2007), 机械工业出版社 (影印版)
- Peter Linz, An Introduction to Formal Languages and Automata, 机械工业出版社影印, 2004

2/22/2022

School of Software

20

## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- 语言基础
- 语言运算

2/22/2022

School of Software

21

## 幂集合

幂集是集合所有子集的集合。

例:  $S = \{a, b, c\}$

S的幂集是S所有子集的集合:

$2^S = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

性质

$$|2^S| = 2^{|S|} \quad (8 = 2^3)$$

2/22/2022

School of Software

22

## 笛卡尔积

$A = \{2, 4\}, \quad B = \{2, 3, 5\}$

$A \times B = \{(2, 2), (2, 3), (2, 5), (4, 2), (4, 3), (4, 5)\}$

显然成立:

$$|A \times B| = |A| * |B|$$

类似扩展到多个集合的情况:

$$A \times B \times \dots \times Z$$

2/22/2022

School of Software

23

## 关系

集合A,B的关系R为:

$$R \subseteq A \times B$$

$$R = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots\}$$

$$x_i R y_i$$

$$x_i \in A, y_i \in B.$$

例: R ">"是实数上的一个关系。

2/22/2022

School of Software

24

### 等价关系

等价关系:

- 自反:  $x R x$
- 对称:  $x R y \iff y R x$
- 传递:  $x R y \text{ 且 } y R z \implies x R z$

例如:  $R = "="$

$$\begin{aligned} x &= x \\ x &= y \iff y = x \\ x &= y \text{ 且 } y = z \implies x = z \end{aligned}$$

2/22/2022

School of Software

19

### 等价关系

对集合A上等价关系  $R$ , 定义元素  $x$  的等价类:

$$[x]_R = \{y \mid x R y\}$$

例如:  $A = \{1, 2, 3, 4\}$

$$R = \{(1, 1), (2, 2), (1, 2), (2, 1), (3, 3), (4, 4), (3, 4), (4, 3)\}$$

$$[1]_R = \{1, 2\}; \quad [3]_R = \{3, 4\}$$

集合上的等价关系可以对集合进行划分。

2/22/2022

School of Software

20

### 半群

半群 (Semigroups)

- 定义: 非空集合S上满足结合律的二元运算 $*$ 构成的数学结构 $(S, *)$ 称为半群。若二元运算 $*$ 很清楚时, 半群可简记为S;  $a*b$ 称为元素 $a$ 和 $b$ 的积。
- 若半群的二元运算 $*$ 是可交换的, 则称 $(S, *)$ 为可交换半群。

例:  $(2^S, \cup)$  和  $(2^S, \cap)$  都是半群。

集合 $Z$ 和减法不构成半群。

2/22/2022

School of Software

21

### 同构

同构 (Isomorphism)

- 定义: 设  $(S, *)$  和  $(T, \circ)$  是半群。若函数  $f: S \rightarrow T$  是由S到T的一一对应, 且对S中所有的 $a$ 和 $b$ , 都有:

$$f(a*b) = f(a) \circ f(b)$$

则称半群 $(S, *)$ 与半群 $(T, \circ)$ 是同构的, 并称函数 $f$ 为半群 $(S, *)$ 到半群 $(T, \circ)$ 的同构映射。

- 性质: 若  $f: S \rightarrow T$  是由半群 $(S, *)$ 到半群 $(T, \circ)$ 的同构映射, 则  $f^{-1}$ 存在, 且  $f^{-1}$ 是由 $(T, \circ)$ 到 $(S, *)$ 的同构映射。

2/22/2022

School of Software

22

### 同构

证明两个半群同构的步骤:

- Step 1 定义一个函数  $f: S \rightarrow T$ , 使得  $\text{Dom}(f) = S$ ;
- Step 2 证明  $f$  是单射;
- Step 3 证明  $f$  是满射;
- Step 4 证明  $f(a*b) = f(a) \circ f(b)$

2/22/2022

School of Software

23

### 同构

例:  $S = \{a, b, c\}$ ,  $T = \{x, y, z\}$ 。

*	a	b	c
a	a	b	c
b	b	c	a
c	c	a	b

o	x	y	z
x	x	x	y
y	x	y	z
z	y	z	x

可以证明上面的运算表给出半群结构 $(S, *)$ 和 $(T, \circ)$ 是同构的。(作业)

2/22/2022

School of Software

24

## 同态

### 同态 (Homomorphism)

定义：设  $(S, *)$  和  $(T, \circ)$  是半群。处处有定义的函数  $f: S \rightarrow T$  称为由  $(S, *)$  到  $(T, \circ)$  的同态，若对  $S$  中所有的  $a$  和  $b$

$$f(a * b) = f(a) \circ f(b)$$

称半群  $(S, *)$  和  $(T, \circ)$  同态。

若  $f$  也是映上的，则  $T$  是  $S$  的同态映像。

2/22/2022

School of Software

19

## 同态

例：设  $A = \{0, 1\}$ ，半群  $(A^*, \cdot)$  和  $(A, +)$ ， $\cdot$  是连接运算， $+$  由下表定义

$+$	0	1
0	0	1
1	1	0

定义函数  $f: A^* \rightarrow A$  为：

$$f(\alpha) = \begin{cases} 1, & \text{若 } \alpha \text{ 含奇数个 } 1 \\ 0, & \text{若 } \alpha \text{ 含偶数个 } 1 \end{cases}$$

可以证明， $f$  是同态映射，但  $f$  不是同构。  
(作业)

2/22/2022

School of Software

20

## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- 语言基础
- 语言运算

2/22/2022

School of Software

21

## 图

有向图  $G = \langle V, E \rangle$

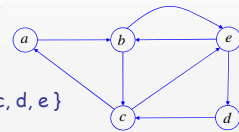
结点 (顶点)

$$V = \{a, b, c, d, e\}$$

边

$$E = \{(a,b), (b,c), (b,e), (c,a), (c,e), (d,c), (e,b), (e,d)\}$$

可见，边是顶点集合上的一个关系



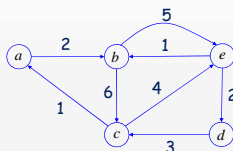
2/22/2022

School of Software

22

## 图

### 标记图



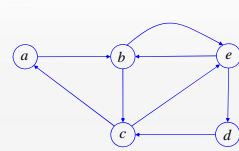
2/22/2022

School of Software

23

## 图

### 通路 (walk)



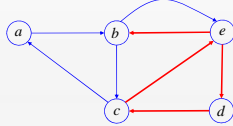
2/22/2022

School of Software

24

图

## 通路 (walk)



相邻的边尾首连接的序列, 如:

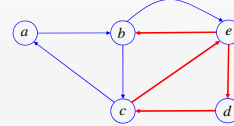
(c,e), (e,d), (d,c), (c,e), (e,b)

2/22/2022

School of Software

图

## 路径 (path)



没有重复边的通路, 如:

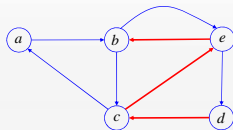
(e,d), (d,c), (c,e), (e,b)

2/22/2022

School of Software

图

## 简单路径



没有重复顶点的路径, 如:

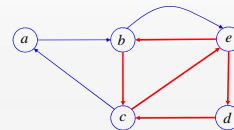
(d,c), (c,e), (e,b)

2/22/2022

School of Software

图

## 回路 (环cycle)



从某顶点到自身的一条路径, 如:

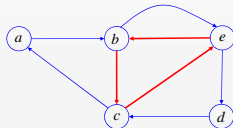
(e,d), (d,c), (c,e), (e,b), (b,c), (c,e)

2/22/2022

School of Software

图

## 简单回路



只有开始顶点重复的回路, 如:

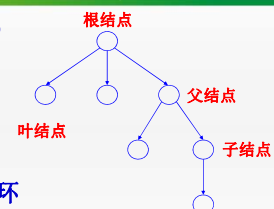
(c,e), (e,b), (b,c)

2/22/2022

School of Software

树

## 树 (tree)

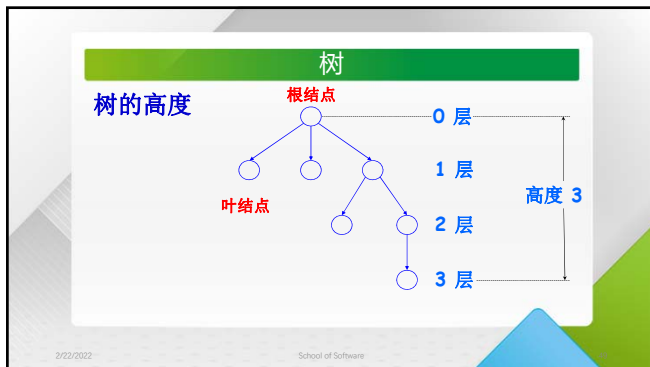


树中没有环

2/22/2022

School of Software





**基础知识**

- 概要
- 数学基础
- 图
- **证明方法**
- 语言基础
- 语言运算

2/22/2022 School of Software

**证明方法**

1. 演绎证明
2. 数学归纳法
3. 反证法
4. 鸽巢原理

2/22/2022 School of Software

**演绎证明**

演绎法：前提  $\Rightarrow$  结论

- 已知的命题集合称为假设 (hypothesis) 或前提 (premise)，推导的命题称为该前提的结论 (conclusion)。
- 一个证明是命题推理的序列，其中的每个命题或者是已知、或者是由前面证明的命题，使用公理和逻辑规则推得。

2/22/2022 School of Software

**演绎证明**

**证明方法：If-Then**  
If 部分作为已知的命题，Then 部分作为结论。

例： 如果  $x+y=1$ ，那么  $x^2-y^2=x-y$ 。

证明：

- 1  $x^2-y^2 = (x+y)(x-y)$  (已知结论)
- 2  $(x+y) = 1$  (前提)
- 3  $x^2-y^2 = x-y$  (根据 1、2 和运算性质)

2/22/2022 School of Software

**演绎证明**

**证明方法：If-and-only-if (当且仅当)**

欲证：

**$A$  if and only if  $B$**

可分别证明如下两个命题：

1. 若  $A$  则  $B$
2. 若  $B$  则  $A$

2/22/2022 School of Software

### 演绎证明

例：设  $R, S$  为集合。

欲证  $R \subseteq S$ , 可证明如下命题：  
若  $x \in R$ , 则  $x \in S$

欲证  $R = S$ , 可分别证明如下两个命题：  
1. 若  $x \in R$ , 则  $x \in S$   
2. 若  $x \in S$ , 则  $x \in R$

2/22/2022

School of Software

### 数学归纳法

数学归纳法是从有限到有限或无限的证明方法，包括归纳基础、归纳假设、归纳推导三个步骤。

设有命题  $P_1, P_2, P_3, \dots$

基础：  $P_1$  为真

假设：  $n \geq 1, P_n$  为真，

推导：  $P_{n+1}$  为真

则对  $\forall i, P_i$  均为真

2/22/2022

School of Software

### 数学归纳法

数学归纳法另一种形式

证明对任意自然数  $n, P(n)$  成立，

- (1) 证明  $P(0)$  成立；
- (2) 若对任意  $k < n, P(k)$  成立，证明  $P(n)$  成立

2/22/2022

School of Software

### 数学归纳法

互归纳法 (Mutual Induction)

设有性质  $P_1(n), P_2(n), \dots, P_m(n)$ , 要证  $P_1(n)$ 。

若证明性质

$$H(n) = P_1(n) \wedge P_2(n) \wedge \dots \wedge P_m(n)$$

比证明  $P_1(n)$  容易。

则可证明  $H(n)$ , 即同时证明  $P_1(n), P_2(n), \dots, P_m(n)$ , 即证明了性质  $P_1(n)$ 。

2/22/2022

School of Software

### 数学归纳法

结构归纳法

由归纳定义的具有结构  $f$  的集合  $S$ , 欲证：

对  $\forall x \in S$ , 满足性质  $P(x)$ 。

基础：证明存在  $a \in S, P(a)$

归纳：若  $a_1, a_2, \dots, a_n \in S, P(a_1), P(a_2), \dots, P(a_n)$ ,

证明  $f(a_1, a_2, \dots, a_n) \in S, P(f(a_1, a_2, \dots, a_n))$

2/22/2022

School of Software

### 数学归纳法

例：定义算术运算

基础：任意数字或字母（变量）都是表示式；

归纳：如果  $P$  和  $R$  是表示式，则

$$P+R, P*R, (P)$$

也是表示式。

例如：1、2、 $x$ 、 $y$  都是表示式

$1+x, (y+2), 2*(1+x)$  都是表示式

2/22/2022

School of Software

### 数学归纳法

例：S为前面定义算术表示式的集合。则每个算术表示式“(”与“)”数目相等。

证明：

基础：空串  $\varepsilon$  的“(”与“)”数目相等，都为0；

归纳：设  $x, y$  的“(”与“)”数目相等，分别为  $m$  和  $n$ ；

$(x)$  的“(”与“)”数目都为  $m+1$ ；

$xy$  的“(”与“)”数目都为  $m+n$ 。

### 反证法

原命题与逆否命题等价。

欲证 若  $A$ ，则  $B$ ，

可证明如下命题：

若非  $B$ ，则非  $A$

### 反证法

例： $\sqrt{2}$  是无理数

证明：假设它是有理数

$$\sqrt{2} = n/m$$

$n$  和  $m$  没有公因子的整数

将证明这是不可能的

### 反证法

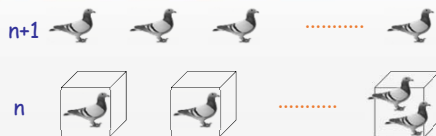
$$\sqrt{2} = n/m \implies 2m^2 = n^2$$

则  $n^2$  是偶数  $\implies n$  是偶数， $n = 2k$

$$2m^2 = 4k^2 \implies m^2 = 2k^2 \implies \begin{matrix} m \text{ 为偶数} \\ m = 2p \end{matrix}$$

因此， $m$  和  $n$  有公因子 2。矛盾

### 鸽巢原理



如果  $n+1$  个物品被放入  $n$  个盒子中，那么至少一个盒子中放有两个或者更多物品。

### 鸽巢原理

例：将 5 个点放在边长 2cm 的正方形中，则至少一对结点之间的距离  $\leq \sqrt{2}$

根据鸽巢原理，如果将正方形分为4个方块，至少有2个点落在同一个方块中。但这些方块的对角线是  $\sqrt{2}$

因此，有两个点之间的距离不超过  $\sqrt{2}$

## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- **语言基础**
- 语言运算

2/22/2022

School of Software

## 字母表

字母表(Alphabet). 例

1. 英文字母表  $\{a, b, \dots, z, A, B, \dots, Z\}$
2. 英文标点符号表  
 $\{, ; : . ? ! ' ' " " ( ) [ ] - - \dots\}$
3. 汉字表  $\{\dots, \text{自}, \dots, \text{动}, \dots, \text{机}, \dots\}$
4. 化学元素表  $\{H, He, Li, \dots, \text{Une}\}$
5.  $\Sigma = \{a, n, y, \text{任意}\}$

2/22/2022

School of Software

## 字符串

字母表:  $\Sigma = \{a, b\}$

串:  $a$

$ab$

$abba$

$baba$

$aaabbbbaabab$

$u = ab$

$v = bbaaaa$

$w = abba$

2/22/2022

School of Software

## 字符串的运算

给定字符串

$w = a_1 a_2 \dots a_n$

$abba$

$v = b_1 b_2 \dots b_m$

$bbbaaaa$

连接 (Concatenation)

$wv = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$

$abbabbbbaaaa$

2/22/2022

School of Software

## 字符串的运算

给定字符串

$w = a_1 a_2 \dots a_n$

$ababaaabbb$

反转 (Reverse)

$w^R = a_n \dots a_2 a_1$

$bbbaaababa$

2/22/2022

School of Software

## 串的长度

字符串的长度:

$w = a_1 a_2 \dots a_n \quad |w| = n$

例:

$|abba| = 4$

$|aa| = 2$

$|a| = 1$

2/22/2022

School of Software

### 串的长度

连接运算串的长度计算:

$$|uv| = |u| + |v|$$

例:  $u = aab, |u| = 3$

$v = abaab, |v| = 5$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

### 空串

空串  $\varepsilon$

定义: 空串  $\varepsilon$  是不含任何字符的串

$$|\varepsilon| = 0$$

空串的连接:

$$\varepsilon w = w \varepsilon = w$$

$$\varepsilon abba = abba \varepsilon = abba$$

### 子串

字符串的子串:

字符串中部分连续字符构成的字符串

	String	Substring
字符串	<u>ab</u> bab	ab
<b>abbab</b>	<u>abb</u> a	abba
	a <u>bb</u> a	b
	ab <u>ba</u>	bbab

### 子串

$$w = uv$$

前缀 后缀

字符串  
**abbab**

前缀

$\varepsilon$   
a  
ab  
abb  
abba  
abbab

后缀

abbab  
bbab  
bab  
ab  
b  
 $\varepsilon$

### 字符串的幂运算

$$w^n = \underbrace{ww \cdots w}_n$$

例:  $(abba)^2 = abbaabba$

规定:  $w^0 = \varepsilon$

例:  $(abba)^0 = \varepsilon$

### \* 运算

字母表的 \* (闭包) 运算

$\Sigma$  : 字母表

$\Sigma^*$  : 字母表中所有有限字符构成串的集合

例:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

## + 运算

字母表的 + (闭包) 运算

 $\Sigma^+$ :  $\Sigma^*$  中除去空串的集合

例:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

$$\Sigma^+ = \Sigma^* - \{\varepsilon\}$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

2/22/2022

School of Software

## 基础知识

- 概要
- 数学基础
- 图
- 证明方法
- 语言基础
- 语言运算

2/22/2022

School of Software

## 语言

字母表为  $\Sigma$ 。  $L \subseteq \Sigma^*$  称为  $\Sigma$  上的一个语言。

例:

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$$

语言:

$$\{\varepsilon\}$$

$$\{a, aa, aab\}$$

$$\{\varepsilon, abba, baba, aa, ab, aaaaaa\}$$

2/22/2022

School of Software

## 语言

空集与空串:

空集为一个语言, 空串是语言的一个元素

$$\emptyset = \{ \} \neq \{\varepsilon\}$$

集合大小

$$|\{ \}| = |\emptyset| = 0$$

$$|\{\varepsilon\}| = 1$$

空串长度

$$|\varepsilon| = 0$$

2/22/2022

School of Software

## 语言

例:  $L = \{a^n b^n \mid n \geq 0\}$ 是  $\Sigma = \{a, b\}$  上一个无限的语言。 $\varepsilon$  $ab$  $aabb$  $aaaaabbbbb$  $\left. \begin{array}{l} \varepsilon \\ ab \\ aabb \\ aaaaabbbbb \end{array} \right\} \in L$  $abb \notin L$ 

2/22/2022

School of Software

## 语言的运算

集合运算:

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

补运算:

$$\bar{L} = \Sigma^* - L$$

$$\overline{\{a, ba\}} = \{\varepsilon, b, aa, ab, bb, aaa, \dots\}$$

2/22/2022

School of Software

## 语言的运算

语言的反转：

$$L^R = \{w^R \mid w \in L\}$$

例：

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

$$L = \{a^n b^n \mid n \geq 0\}$$

$$L^R = \{b^n a^n \mid n \geq 0\}$$

2/22/2022

School of Software

## 语言的运算

语言的连接

设语言 $L_1, L_2$ , 其连接为：

$$L_1 L_2 = \{xy \mid x \in L_1, y \in L_2\}$$

例：

$$\{a, ab, ba\} \{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

2/22/2022

School of Software

## 语言的运算

语言 $L$ 的幂：

$$L^n = \underbrace{LL \cdots L}_n$$

例：

$$\{a, b\}^3 = \{a, b\} \{a, b\} \{a, b\} =$$

$$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

规定：

$$L^0 = \{\varepsilon\}$$

$$\{a, bba, aaa\}^0 = \{\varepsilon\}$$

2/22/2022

School of Software

## 语言的运算

语言 $L$ 的幂：

$$L^n = \underbrace{LL \cdots L}_n$$

例：

$$L = \{a^n b^n \mid n \geq 0\}$$

$$L^2 = \{a^n b^n a^m b^m \mid n, m \geq 0\}$$

$$aabbbaaabb \in L^2$$

2/22/2022

School of Software

## 语言的闭包运算

语言 $L$  - “\*” 闭包 (Kleene闭包)：

$$L^* = L^0 \cup L^1 \cup L^2 \cdots$$

例：

$$\{a, bb\}^* = \left\{ \begin{array}{l} \varepsilon, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

2/22/2022

School of Software

## 语言的闭包运算

语言 $L$  - “+” 闭包：

$$L^+ = L^1 \cup L^2 \cup \cdots$$

$$= L^* - \{\varepsilon\}$$

例：

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

2/22/2022

School of Software

## 课后练习

✧ Let  $L = \{ab, aa, baa\}$ . Which of the following strings are in  $L^*$ :

*abaabaaabaa*

*aaaabaaaa*

*baaaaabaaaaab*

*baaaaabaa ?*

✧ PPT中第35页、37页的作业题

2022/2/22

School of Software



Thank you

2/22/2022

School of Software