计算理论
Theory of Computation

https://courses.zju.edu.cn/course/63597

2023秋冬周一第3,4节/计算理论 内部

该群属于"浙江大学"内部群，仅组织内部成员可以加入，如果组织外部人员收到此分享，需要先申请加入该组织。

此二维码365天内有效（2024-09-15 前）

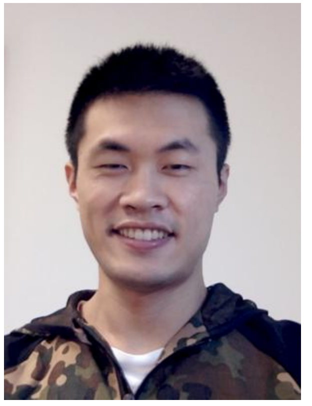
钉钉群号： 43215000056

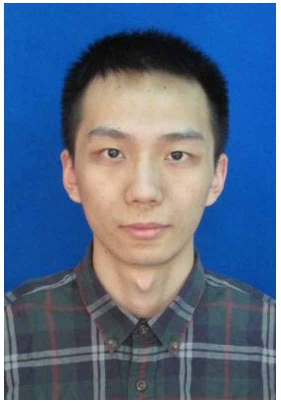
授课教师： 郑乾

qianzheng@zju.edu.cn

助教： 赵晨希

3190102973@zju.edu.cn



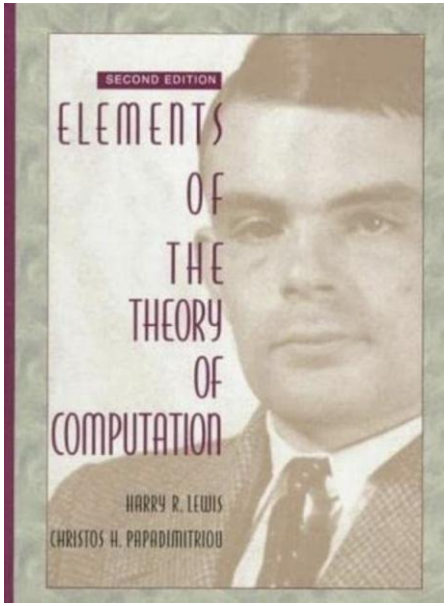


浙江大学玉泉校区教7-208

2023年9月25日

内容安排

– 3 classes
  Sets, Relations and Language（CH1）
– 3 classes
  Regular Language and Finite Automata (CH2)
– 3 classes
  Context-free Languages (CH3)
– 3 classes
  Turing machine (CH4)
– 2 classes
  Undecidability（CH5）
– 1 class
  – Review



Brief Review

1.4 Finite and Infinite sets

- Equinumerous
- Countable and Uncountable Infinite

1.5 Three Fundamental Proof Techniques

- The Principle of Mathematical Induction
- The Pigeonhole Principle
- The Diagonalization Principle

计算理论

第1章 集合、关系、语言

Ch1. Sets, Relations and Language

Closures - Intuitive

Idea

Natural numbers N are **closed** under $+$, i.e. for given two natural numbers n, m we always have that $n + m \in N$

Natural numbers N are **not closed** under subtraction $-$, i.e there are two natural numbers n, m such that $n - m \notin N$, for example $1, 2 \in N$ and $1 - 2 \notin N$

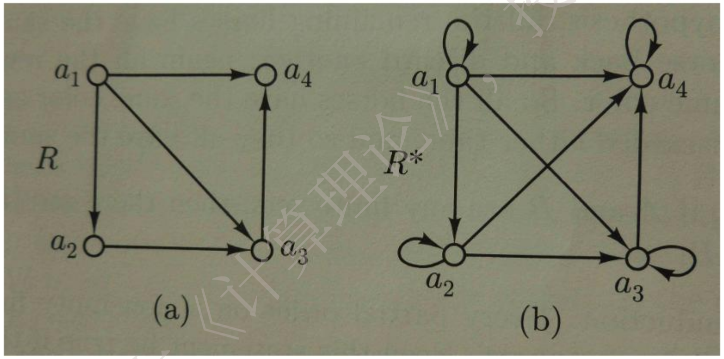
Integers Z are **closed** under $-$, moreover Z is the smallest set containing N and closed under subtraction $-$

The set Z is called a **closure** of N under subtraction $-$

1.6 Closures

Closures - Intuitive

Consider the two directed graphs R (a) and R^* (b) as shown below



Observe that $R^* = R \cup \{(a_i, a_i) : \ i = 1, 2, 3, 4\} \cup \{(a_2, a_4)\}$,

$R \subseteq R^*$ and is R^* is reflexive and transitive whereas R is neither, moreover R^* is also the smallest set containing R that is reflexive and transitive

We call such relation R^* the reflexive, transitive closure of R

1.6 Closures

□ The Transitive Closure

the "smallest" relation that includes R and is transitive (usually called R^+)

More formally:

R^+ is a relation such that

* $R \subseteq R^+$
* R^+ is transitive
* $\forall R', R \subseteq R'$ and R' is transitive, $\Rightarrow R^+ \subseteq R'$

1.6 Closures

□ Closures of Relations

Given any binary relation R, one can form closures with respect to any combinations of the properties:

- reflexive
- symmetric
- transitive

Note:

Reflexive, transitive closure of R is usually denoted R^*.

1.7 Alphabet and Language

Data are **encoded** in the computers' memory as strings of bits or other symbols appropriate for **manipulation**

The mathematical study of the **Theory of Computation** begins with understanding of mathematics of **manipulation** of strings of symbols

We first introduce two basic notions: Alphabet and Language

□ Alphabet

Definition

Any finite set is called an **alphabet**

Elements of the **alphabet** are called symbols of the alphabet

Alphabet Notation

We use a symbol Σ to denote the **alphabet**

Is ∅ an alphabet?

1.7 Alphabet and Language

□ Alphabet

E1 $\Sigma = \{\ddagger,\ \emptyset,\ \partial,\ \oint,\ \bigotimes,\ \vec{a},\ \nabla\}$

E2 $\Sigma = \{a,\ b,\ c\}$

E3 $\Sigma = \{n \in N :\ n \leq 10^5\}$

E4 $\Sigma = \{0, 1\}$ is called a **binary alphabet**

1.7 Alphabet and Language

□ Alphabet

A **finite sequence** of elements of a set A

length

Let Σ be an **alphabet**

We call finite sequences of the alphabet Σ **words** or **strings** over Σ

We denote by e the **empty word** over Σ

Some books use symbol λ for the **empty word**

Words over Σ

E5 Let $\Sigma = \{a, b\}$

We will write some words (strings) over Σ in a **shorthand** notaiton as for example

$$aaa,\ ab,\ bbb$$

Words in Σ^*

Let Σ be an **alphabet**. We denote by

$$\Sigma^*$$

the set of **all finite** sequences over Σ

Elements of Σ^* are called **words** over Σ

We write $w \in \Sigma^*$ to express that w is a **word** over Σ

Is $\emptyset = \Sigma^*$?

1.7 Alphabet and Language

□ **Operations of Strings:**

words are also named as strings

- **Concatenation:** $x \circ y$ or xy

  Substring, suffix, prefix

  Example: $\forall w, we = ew = w$

- **String exponentiation**

$w^0 = e$, the empty string
$w^{i+1} = w^i \circ w$, for each $i \geq 0$

—*definition by induction*

- **Reversal**

If w is a string of length 0, then $w^R = w = e$.
IF w is a string of length $n+1 > 0$, then $w = ua$ for some $a \in \sum$, and $w^R = au^R$.

1.7 Alphabet and Language

- **Language:** set of strings
  - $\sum$–alphabet, $\sum^*$–the set of all strings ($e \in \sum^*$)
  - Language $L \subseteq \sum^*$
  - $\emptyset$, $\sum$ and $\sum^*$ are languages.

  - Finite Language: by listing all the strings

    Infinite Language: specify by the following scheme

$$L = \{w \in \textstyle\sum^* : w \text{ has property } P\}$$

Example: $L = \{ab, aabb, aaabbb, \cdots\} = \{a^n b^n \mid n \geq 1\}$

1.7 Alphabet and Language

Theorem: If $\sum$ is a finite alphabet, then $\sum^*$ is countably infinite set.

Proof: Construct a bijection $f : \mathbb{N} \to \sum^*$.

Fix some ordering of the alphabet, say $\sum = \{a_1, a_2, \cdots, a_n\}$.

The member of $\sum^*$ can be enumerated in the following

For example, if $\Sigma = \{0, 1\}$, the order would be as follows:

$$e, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, \ldots$$

2) The n^k strings of length exactly k are enumerated lexicographically.

1.7 Alphabet and Language

How many words or strings over a non empty alphabet?

Alphabet is a finite set. All finite sequences of finite symbols is countably infinite.

Is a language countable over a non empty alphabet?

For any alphabet Σ, any language over Σ is **countable**

How many languages over a non empty alphabet?

For any alphabet $\Sigma \neq \emptyset$, there are exactly $C = |R|$ of **languages**

1.7 Alphabet and Language

□ Operations of Languages:

- **Union, Intersection, Difference, Complement**

$(\overline{A} = \Sigma^* - A)$

- **Concatenation:**

$$L^0 = \{e\}$$
$$L^{i+1} = LL^i, \text{ for each } i \geq 0$$

$$L_1L_2 = \{w_1w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$$

Example:

$L_1 = \{w \in \{0,1\}^* : w$ has an even number of 0's$\}$

$L_2 = \{w \in \{0,1\}^* : w$ starts with a 0, the rest symbol are 1's$\}$

$L_1L_2 = \{w \in \{0,1\}^* : w$ has an odd number of 0's$\}$.

1.7 Alphabet and Language

- **Kleene Star**

$$L^* = \{w \in \Sigma^* : w = w_1 \cdots w_k, k \geq 0, w_1, \cdots, w_k \in L\}$$

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

$$L^+ = L^1 \cup L^2 \cup L^3 \cup \cdots$$

Can e∈ L^+?

Example: $L = \{w \in \{0,1\}^* : w$ has an unequal number of 0's and 1's$\}$. Then $L^* = \{0,1\}^*$.

Hint: $L_1 \subseteq L_2 \Rightarrow L_1^* \subseteq L_2^*$ $\qquad$ $\{0,1\} \subseteq L$

$(L^*)^* = L^*$

1.7 Alphabet and Language

Remark:

1) The use of $\sum^*$ to denote the set of all strings over $\sum$ is consistent with the notation for the Kleene star of $\sum$.

2) $\emptyset^* = \{e\}$

3) $L^+ = LL^*$

4) For any language L, $(L^*)^* = L^*; L\emptyset = \emptyset L = \emptyset$

We write L^+ for the language LL^*. Equivalently, L^+ is the language

$$\{w \in \Sigma^* : w = w_1 \circ w_2 \circ \cdots \circ w_k \text{ for some } k \geq 1 \text{ and some } w_1, \ldots, w_k \in L\}.$$

Notice that L^+ can be considered as the *closure* of L under the function of concatenation. That is, L^+ is the smallest language that includes L and all strings that are concatenations of strings in L.

1.8 Finite Representations of Languages

We can represent a finite language by **finite means** for example listing all its elements

Languages are often infinite and so a natural question arises if a **finite representation** is possible and when it is possible when a language is infinite

The representation of languages by **finite specifications** is a central issue of the theory of computation

Of course we have to define first formally what do we mean by representation by finite specifications , or more precisely by a **finite representation**

1.8 Finite Representations of Languages

□ Finite Representations:

– must be a string

– different languages to have different representations

Example Let $L = \{w \in \{0,1\}^* : w$ has two or three occurrences of 1, the first and second of which are not consecutive$\}$.

- The language can be described using only singleton sets and the symbols $\cup$, $\circ$, and $*$ as

$$\{0\}^* \circ \{1\} \circ \{0\}^* \circ \{0\} \circ \{1\} \circ \{0\}^*((\{1\} \circ \{0\}^*) \cup \emptyset^*)$$

- The language can be written simply as

$$0^*10^*010^*(10^* \cup \emptyset^*).$$

– Regular Expressions

1.8 Finite Representations of Languages

Definition: The regular expressions are all strings over the alphabet $\sum \cup \{(,),\cup,\star\}$ that can be obtained as follows.

1) Θ and $\{x\}(\forall x \in \sum)$ is a regular expression.

2) If α and β are regular expressions, then so are $(\alpha\beta)$, $(\alpha \cup \beta)$, $\alpha^\star$.

3) Nothing is regular expression unless it follows from 1) through 2).

Example: a^*b^* $\quad a^* \cup b^*$ $\quad a(a^* \cup b^*)$

$(a^* \cup b^*)a(a^* \cup b^*)$ $\quad aaaaa^*$

1.8 Finite Representations of Languages

Definition

We define a $\mathcal{R}$ of **regular expressions** over an alphabet Σ as follows

$\mathcal{R} \subseteq (\Sigma \cup \{(,\ ),\ \emptyset,\ \cup,\ *\})^*$ and $\mathcal{R}$ is the smallest set such that

1. $\emptyset \in \mathcal{R}$ and $\Sigma \subseteq \mathcal{R}$, i.e. we have that

$$\emptyset \in \mathcal{R} \text{ and } \forall_{\sigma \in \Sigma}\ (\sigma \in \mathcal{R})$$

2. If $\alpha, \beta \in \mathcal{R}$, then

$(\alpha\beta) \in \mathcal{R}$ **concatenation**

$(\alpha \cup \beta) \in \mathcal{R}$ **union**

Example: $L = \{ab, aabb, aaabbb, \cdots\} = \{a^n b^n \mid n \geq 1\}$

$\alpha^* \in \mathcal{R}$ **Kleene's Star**

We use the regular expressions from the set $\mathcal{R}$ as a

Question

Can we **finitely represent** all languages over an alphabet $\Sigma \neq \emptyset$?

Representations
(Σ^*, countable)

Languages
(2^{Σ^*}, uncountable)

$\Rightarrow \exists$ undescribable languages!

Regular expressions & languages.

Definition

The **representation relation** between regular expressions and languages they **represent** is establish by a **function** $\mathcal{L}$ such that

if $\alpha \in \mathcal{R}$ is any regular expression, then $\mathcal{L}(\alpha)$ is the **language represented** by α

Regular expressions & languages.

Formal Definition

The function $\mathcal{L} : \mathcal{R} \longrightarrow 2^{\Sigma^*}$ is defined recursively as follows

1. $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\sigma) = \{\sigma\}$ for all $\sigma \in \Sigma$
2. If $\alpha, \beta \in \mathcal{R}$, then

$$\mathcal{L}(\alpha\beta) = \mathcal{L}(\alpha) \circ \mathcal{L}(\beta)$$ **concatenation**

$$\mathcal{L}(\alpha \cup \beta) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$$ **union**

$$\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$$ **Kleene's Star**

1.8 Finite Representations of Languages

Example: What is $\mathscr{L}(((a \cup b)^\star a))$? ?

$$\mathscr{L}(((a \cup b)^\star a)) = \mathscr{L}(((a \cup b)^\star))\mathscr{L}(a)$$
$$= \mathscr{L}(((a \cup b)^\star))\{a\}$$
$$= \mathscr{L}(((a \cup b))^*\{a\}$$
$$= (\mathscr{L}(a) \cup \mathscr{L}(b))^*\{a\}$$

$$= (\{a, b\})^*\{a\}$$
$$= \{w \in \{a, b\}^* : w \text{ ends with an } a\}$$

Regular Expression Identities

- $SR \neq RS$
- $S \cup R = R \cup S$
- $R(ST) = (RS)T$
- $R(S \cup T) = RS \cup RT$, $(R \cup S)T = RT \cup ST$
- $\emptyset^* = \{e\}$
- $(R^*)^* = R^*$
- $(R^*S^*)^* = (R \cup S)^*$
- $(\{e\} \cup R)^* = R^*$

(a) $(x+y)^*$
(b) $(x^*+y)^*$
(c) $x^*(x+y)^*$
(d) $(x+yx^*)^*$
(e) $(x^*y^*)^*$
(f) $x^*(yx^*)^*$
(g) $(x^*y)^*x^*$

Example: What language is represented by $(c^\star(a \cup (bc^\star))^\star)$?

1.8 Finite Representations of Languages

Example: What language is represented by $(c^\star(a\cup(bc^\star))^\star)$?

$L = \{w \in \{a,b,c\}^* :$ not have the substring $ac\}$.

$\boxed{\ldots\ldots}a\boxed{\ldots\ldots}a\ldots\ldots a\boxed{\ldots\ldots}$

c* or **(bc*)*** **={b,c}***

(bc*)* **e or b+ or bc* or b{b,c}***

1.8 Finite Representations of Languages

Remark:

1) Every language that can be represented by a regular expression can be represented by infinitely many of them.

2) The class of regular languages over an alphabet $\sum$ is defined to consist of all languages L such that $L = L(a)$ for some regular expression a over $\sum$. i.e. the class of regular languages over an alphabet $\sum$ is precisely the closure of the set of languages

$$\{\{\sigma\} : \sigma \in \sum\} \cup \{\emptyset\}$$

1.8 Finite Representations of Languages

3) The regular expressions are an inadequate specification method in general.
For example, $\{0^n 1^n : n \geq 0\}$ cannot be described by regular expressions.

4) Two important and useful means of representing languages:
– language recognition device

to answer questions of the form "Is string w a member of L?

– language generators

1.8 Finite Representations of Languages

language recognition device.

$$L = \{w \in \{0,1\}^* : w \text{ does not have } 111 \text{ as a substring}\}.$$

by reading strings, a symbol at a time, from left to right, might operate like this:

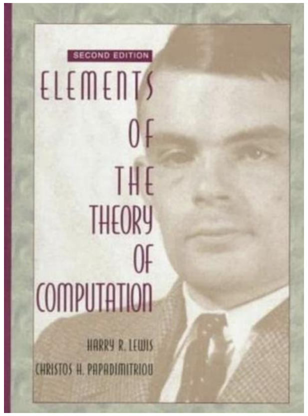
Keep a count, which starts at zero and is set back to zero every time a 0 is encountered in the input; add one every time a 1 is encountered in the input; stop with a No answer if the count ever reaches three, and stop with a Yes answer if the whole string is read without the count reaching three.

language generators

An alternative and somewhat orthogonal method for specifying a language is to describe how a generic specimen in the language is produced. For example, a regular expression such as $(e \cup b \cup bb)(a \cup ab \cup abb)^*$ may be viewed as a way of *generating* members of a language:

To produce a member of L, first write down either nothing, or b, or bb; then write down a or ab, or abb, and do this any number of times, including zero; all and only members of L can be produced in this way.

Homework 1:	
P46	
	1.7.4 (c)(d)
	1.7.6
P51	
	1.8.3(c)
	1.8.5



1.7.4. Show each of the following.

(a) $\{e\}^* = \{e\}$

(b) For any alphabet Σ and any $L \subseteq \Sigma^*$, $(L^*)^* = L^*$.

(c) If a and b are distinct symbols, then $\{a, b\}^* = \{a\}^*(\{b\}\{a\}^*)^*$.

(d) If Σ is any alphabet, $e \in L_1 \subseteq \Sigma^*$ and $e \in L_2 \subseteq \Sigma^*$, then $(L_1 \Sigma^* L_2)^* = \Sigma^*$.

(e) For any language L, $\emptyset L = L \emptyset = \emptyset$.

1.7.6. Under what circumstances is $L^+ = L^* - \{e\}$?

1.8.3. Let $\Sigma = \{a, b\}$. Write regular expressions for the following sets:

(a) All strings in Σ^* with no more than three a's.
(b) All strings in Σ^* with a number of a's divisible by three.
(c) All strings in Σ^* with exactly one occurrence of the substring aaa.

1.8.5. Which of the following are true? Explain.

(a) $baa \in a^*b^*a^*b^*$
(b) $b^*a^* \cap a^*b^* = a^* \cup b^*$
(c) $a^*b^* \cap b^*c^* = \emptyset$
(d) $abcd \in (a(cd)^*b)^*$