Note of Discrete Mathematics

Yzy

2020 年 3 月 19 日

目录

1 Propositions 命题

Theorem 1 *A Propositions is a declarative sentence that is either true or false, but not both.* 命题就是陈述性的语句，只有“对”或者“错”。

conjunction 合取—$\wedge$ disjunction 析取—$\vee$ exclusive or—$\oplus$
conditional statements—$p \to q$: (等价于 $\neg p \wedge q$) [1]
if p, then q(若 p, 则 q) 等价于 **p only if q(p 仅当 q)**
converse(逆命题) of $p \to q$: $q \to p$
contrapositive(逆否命题) of $p \to q$: $\neg q \to \neg p$ [2]
inverse(反命题) of $p \to q$: $\neg p \to \neg q$ [3]
当两个 propositions 具有相同真值时，称为等价(equivalent).
$p \leftrightarrow q$(p 当且仅当 q) 与 $(p \to q) \wedge (q \to p)$ 等价

2 命题等价式 Propositional Equvialent

- 永真式 (tautology 重言式): 永远为真。
- 矛盾式 (contradiction): 永远为假。
- 可能式 (contingency): 既不是永真式又不是矛盾式。

[1] 条件语句作为一个数学概念不依赖于假设和结论之间的因果关系，而是用定义规定了它的真值！
[2] 逆否命题与原命题具有相同的真值！
[3] 逆命题和反命题是等价的！

2.1 逻辑等价式 Logical Equvialent

逻辑等价：在所有情况下都有相同真值的两个复合命题，称为逻辑等价。记为: $p \equiv q$. (注意：$\equiv$ 并不是逻辑连接词)

德摩根律 DeMorgan's Laws : $\dfrac{\neg(p \wedge q) \equiv \neg p \vee \neg q}{\neg(p \vee q) \equiv \neg p \wedge \neg q}$

构建复合命题的真值表时，有几个命题变元，真值表就有 $\mathbf{2^n}$ 行。

27，28 页：表 6,7,8 有用的逻辑等价式

有如下事实：若 $p \equiv q$ 且 $q \equiv r$, 则 $p \equiv r$.

2.2 命题的可满足性 Propositional Satisfiability

一个命题称为是: $\begin{cases} \text{可满足的 (satisfiable)}-\text{如果}\textbf{存在一个赋值}\text{使其真值为真} \\ \text{不可满足的 (unsatisfiable)}-\text{对}\textbf{所有变元的真值赋值}\text{都是假的} \end{cases}$

- 与非 (NAND) $|$, 顾名思义　$p|q \equiv \neg(p \wedge q)$
- 或非 (NOR) $\downarrow$, 同理　$p \downarrow q \equiv \neg(p \vee q)$

$\{\neg \vee \wedge\}$,$\{\neg\vee\}$, $\{\neg\wedge\}$, $\{\downarrow\}$ 均是功能完备集 functional collection。

$A_1, ..., A_n$ 均为合取式, 则 $A_1 \vee A_2 \vee ... \vee A_n$ 称为析取范式，合取范式反之亦然。

2.3 谓词和量词 Predicates and Quantifiers

n 个变量 $x_1, x_2, \ldots, x_n$ 的语句表示成：$P(x_1, x_2, \ldots, x_n)$. P(...) 就是命题函数 P 在 n 元组 $(x_1, x_2, \ldots, x_n)$ 的真值。P 也称为 **n 元谓词**。

全称量词 universal quantifiers 数学命题断言某一性质对于变量在某一特定域内的所有值为真，这一特定域成为变量的**论域** (domain of discourse) 或全体域 (universe of discourse) 或域 (domain)

全称量词 (universal quantifiers)—$\forall$ (注：使用量词时必须先指定论域)

$\forall x P(x)$ 表示 $P(x)$ 的全称量化，“$P(x)$ 对 x 在其论域的所有值为真，”也读作：“对所有 x，P(x)”。

存在量词 (existential quantifiers)—$\exists$

$\exists x P(x)$, “论域中存在一个个体 x 满足 $P(x)$。”

唯一性量词 (uniqueness quantifiers)—$\exists!$
$\exists! x P(x)$ 表示：存在一个唯一的 x 使得 P(x) 为真。（有且只有一个）
约束论域: 变量必须满足的条件**直接放在量词的后面**
如：
$\forall x < 0 (x^2 > 0)$ 表示“对每一个满足 x>0 的实数 x 有 $x^2 > 0$”—等价于：
$\forall x (x < 0 \to x^2 > 0)$
$\exists z > 0 (z^2 = 0)$ 等价于 $\exists z (z > 0 \wedge z^2 = 0)$
由上可知，

- **全称量化的约束**和一个**条件语句的全称量化**等价:
  $\forall x$(约束条件)$P(x) \equiv \forall x$(约束条件 $\to P(x)$)
- 存在量化的约束和一个**合取式的存在量化**等价

量词有**最高的优先级** (precedence)
涉及**量词的逻辑等价式**:
例：$\forall x (P(x) \wedge Q(x)) \equiv \forall x P(x) \wedge \forall x Q(x)$
$\exists x (P(x) \vee Q(x)) \equiv \exists x P(x) \vee \exists x Q(x)$

- 全称量词对于一个**合取式**是可分配的, 但对析取式则不可分配。
- 存在量词对于一个**析取式**是可分配的, 但对合取式则不可分配。

量化表达式的否定
量词否定的规则成为**量词的德摩根律**:
$\neg \forall x P(x) \equiv \exists x \neg P(x)$
$\neg \exists x Q(x) \equiv \forall x \neg Q(x)$
空量化 null quantification—用于将量词**左移**
(1.4 练习 46 49) 假设 x 在 A 中均不作为自由变元出现，且论域非空：
全称量词：

- $(\forall x P(x)) \vee A \equiv \forall (x P(x) \vee A)$
- $(\forall x P(x)) \wedge A \equiv \forall x (P(x) \wedge A)$
- $A \to \forall x P(x) \equiv \forall x (A \to P(x))$
- $\exists x P(x) \to A \equiv \forall x (P(x) \to A)$

存在量词：

- $(\exists x P(x)) \vee A \equiv \exists x(P(x) \vee A)$
- $(\exists x P(x)) \wedge A \equiv \exists x(P(x) \wedge A)$
- $A \to \exists x P(x) \equiv \exists x(A \to P(x))$
- $\forall x P(x) \to A \equiv \exists x(P(x) \to A)$

(以上逻辑等价式均可用分情况证明！)

2.4 嵌套量词 Nested Quantifiers

嵌套量词，即一个量词出现在另一个量词的作用域内，如：$\forall x \exists y(x+y) = 0$。注意，**量词范围内的一切都可以认为是一个命题函数**。

$\forall x \exists y(x+y) = 0$ 可以描述为 $\forall x Q(x)$，其中 $Q(x) \equiv \exists y P(x,y)$，$P(x,y) \equiv x + y = 0$.

将量化当做循环

如要判定 $\forall x \exists y P(x,y)$ 是否为真，对 x 的所有值循环。对每个 x 值，再对 y 值循环直到找到一个 y 使 P(x,y) 为真。如果对所有的 x 值都能找到一个 (或以上)y 值使 P(x,y) 为真，那么该表达式为真。

$\exists x \forall y P(x,y), \exists x \exists y P(x,y)$ 同理。

量词顺序

在没有其他量词的语句中，在不改变量化式意义的前提下嵌套全称量词的顺序是可以改变的。

嵌套量词的否定

方法：通过连续地应用**量词的德摩根律**。

例：$\neg(\forall x \exists y(xy = 1)) = \exists x \neg \exists y(xy = 1) = \exists x \forall y \neg(xy = 1) = \exists x \forall y(xy \neq 1)$

注意，把 $\neg$ 放量词后面时，不改变变元的论域，如：$\neg \forall x > 0 \exists y < 0 P(x,y) = \exists x > 0 \neg \exists y < 0 P(x,y)$

前束范式 prenex normal form,PNF

当且仅当表达形式为：$Q_1 x_1 Q_2 x_2 \ldots Q_k x_k P(x_1, x_2, \ldots, x_k)$，其中每个 Q_i 都是全称量词或者存在量词，并且 $P(x_1, x_2, \ldots, x_k)$ 是不含量词的谓词。

1.5 48 $\forall x P(x) \vee \forall x Q(x) \equiv \forall x \forall y(P(x) \vee Q(x))$

1.5 49 $\forall x P(x) \wedge \exists x Q(x) \equiv \forall x \exists y(P(x) \wedge Q(x))$

1.5 49 $\forall x P(x) \vee \exists x Q(x) \equiv \forall x \exists y(P(x) \vee Q(x))$

对**唯一性量词**有等价式：$\exists! x P(x) \equiv \exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x)) \equiv \exists x \forall y(P(x) \leftrightarrow y = x)$

2.5 推理规则 Rules of Inference

论证 (argument): 一连串的命题并以结论作为最后的命题；
有效性 (valid): 结论或论证的最后一个命题必须根据论证过程前面的命题或前提 (premise) 的真实性推出;
一个论证是有效的**当且仅当**不可能出现所有前提为真而结论为假的情况，即它的所有前提为真蕴含着结论为真。
谬误 (fallacy): 错误推理。
如果前提均为真时结论为真，那么说这个论证形式是**有效的**, 也就是说这个推理规则是**可以用的！**
永真式 $(p \wedge (p \rightarrow q)) \rightarrow q$ 是**假言推理 (modus ponens)** 或分离规则 (law of detachment) 的推理规则的基础。它说明：**如果一个条件语句以及它的前提都为真，那么结论肯定为真！**
章节 1.6.3 常用推理规则表

- 化简律 $(p \wedge q) \rightarrow p$
- 附加律 $p \rightarrow (p \vee q)$
- 取拒式 $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$

消解律 $((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$, 其中 $(q \vee r)$ 称为消解式
常把假设和结论表示为子句（变量或其否定的一个析取式），再用消解律构造证明。**量化命题的推理规则**
1.6.7—全称实例，全称引入，存在实例，存在引入
1.6.8—命题和量化命题推理规则的组合使用：全称假言推理，全称取拒式

2.6 证明导论

定理 theorem: 一个能够被证明是真的语句。
证明定理的方法：

p	q	$value$
T	T	T
T	F	F
F	T	T
F	F	T

- 直接证明法，利用条件语句 $p \to q$ 来构造
- 间接证明法 (反证法 proof by contrapositive), 利用 $p \to q \equiv \neg q \to \neg p$ 这一事实来构造
- 空证明 (vacuous proof) 即若前提 p 为假，则 $p \to q$ 一定为真
- 平凡证明 (trivial proof) 即若结论 q 为真，则 $p \to q$ 一定为真
- 归谬证明法 (proof by contradiction) 即找到一个矛盾式 $r \wedge \neg r$ 使得 $\neg p \to (r \wedge \neg r)$ 为真，则 p 为真
- 等价证明法为了证明 $p \leftrightarrow q$, 即要证明 $p \to q$ 和 $q \to p$ 都是真的
- 反例证明法要证明形如 $\forall x P(x)$ 的语句为假，则只需要找到一个反例即可
- 分情形证明法 $[(p_1 \vee p_2 \vee \cdots \vee p_n) \to q] \leftrightarrow [(p_1 \to q) \wedge (p_2 \to q) \wedge \cdots \wedge (p_n \to q)]$
- 穷举证明法 (proof by exhaustion) 通过检验相对少量的例子来证明

不失一般性 without loss of generality(WLOG)
一般是为了缩短篇幅，即后面的情况可以采用同样的 (或有明显改变) 论证来完成证明。

3 基本结构：集合，函数，序列，求和与矩阵

3.1 集合 Sets

集合是对象的一个无序聚集 (collection), 对象也称为元素 (element) 或成员 (member)
通常大写表示集合，小写表示集合中的元素。 $a \in A, a \notin A$

可用集合构造器 (set builder) 来描述集合：$O = \{x \mid x$是小于 10 的正奇数$\}$
$O = \{x \in Z^+ \mid x$为奇数$, x < 10\}$
集合相等　$\forall x(x \in a \leftrightarrow x \in B)$记为$(A = B)$，只要**元素相同**，就算是**顺序不同**或者**元素出现次数不同**都是相等！
空集 (empty set or null set): $\varnothing$　*or*　$\{\}$
单元素集 (singleton set): 只有一个元素的集合。如: $\{\varnothing\}$，**但是它不是空集！**
可以类比为**:**$\varnothing \rightarrow$ 一个空文件夹　　$\{\varnothing\} \rightarrow$ 一个文件夹中只有一个空文件夹
子集 (subsets): $A \subseteq B$当且仅当$\forall x(x \in A \rightarrow x \in B)$
真子集 (proper subsets) $A \subset B$　当且仅当$\forall x(x \in A \rightarrow x \in B) \wedge \exists x(x \in B \wedge x \notin A)$
集合可以作为另一个集合的成员
例子：$A = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$,　$B = \{x \mid x$是集合 a,b 的子集$\}$ **注意到，这两个集合是相等的！且** $a \notin A$,　$\{a\} \in A$

3.1.1　集合的大小

令 S 为集合，若 S 中有 **n 个不同的元素**，且 n 为非负整数，则 S 为有限集。n 为 S 的基数 (cardinality), 记为 $\mid S \mid$
幂集 (power set), S 的幂集是 S 所有子集的集合，记为 $\mathcal{P}(A)$。若 $\mid S \mid = n$, $\mid \mathcal{P}(S) \mid = 2^n$

3.1.2　笛卡尔积 Cartesian Products

有序 n 元组 (ordered n-tuple) $(a_1, a_2, \ldots, a_n)$ 是以 a_1 为第一个元素...，是有序聚集！
有序 n 元组是相等的当且仅当每一对对应的元素相等，即 $a_i = b_i$
有序二元组称为**序偶** (ordered pair):(a,b), (c,d)
笛卡尔积: $A \times B = \{(a, b) \mid a \in A \wedge b \in B\}$. 需要注意的是：若 A，B 不为空集，又或 A,B 不等，则 $B \times A \neq A \times B$
而且，$(A \times B) \times C \neq A \times B \times C$
用 $A^n = A \times A \times, \ldots, \times A$ 表示集合 A 与自身的笛卡尔积。
笛卡尔积 $A \times B$ 的一个子集 R 称为是从集合 A 到集合 B 的一个**关系 relation**,R 的元素是序偶，(第一个元素属于 A，第二个元素属于 B)。

真值集 truth set

将集合理论与谓词逻辑结合，给定谓词 P，论域 D，定义 P 的真值集为 D 中使 P(x) 为真的元素 x 组成的集合，记为 $\{x \in D \mid P(x)\}$

练习题：

$\mid A \mid = m, \mid B \mid = n, then \mid A \times B \mid = mn$

$\mid A \mid = m, \mid B \mid = n, \mid C \mid = p, then \mid A \times B \times C \mid = mnp$

$\mid A \mid = m, and n is a integral number, then \mid A^n \mid = m^n$

求有限集的所有子集的方法

令 $S = \{a_1, a_2, \cdots, a_n\}$, 把 S 的每个子集用**长度为 n 的位串**表示，其中第 i 位为 1 当且仅当 $a_i \in S$。为了产生 S 的所有子集，列出所有 2^n 个长度为 n 的位串，再按照对应位置写出子集即可！(即利用二进制编码实现)

3.2 集合运算

差集:$A - B = A \cap \overline{B}$

对称差 (symmetric difference)—$\oplus$, $A \oplus B$: 属于 A 或属于 B 但不同时属于 A 与 B 的元素组成的集合。对称差常用公式：

对称差定义 $A \oplus B = (A \cup B) - (A \cap B)$

对称差定义 $A \oplus B = (A - B) \cup (B - A)$

交换律 $A \oplus B = B \oplus A$

结合律 $A \oplus (B \oplus C) = (A \oplus B) \oplus C$　可由定义证得

- $(A \oplus B) \oplus B = A$
- $A \oplus A = \varnothing$
- $A \oplus \varnothing = A$
- $A \oplus U = \overline{A}$
- $A \oplus \overline{A} = U$
- 当 $A \oplus B = C$ 时，$A \oplus C = B$

模糊集 fuzzy sets 全集 U 中每个元素在模糊集合 S 中都有一个隶属度，即在 [0,1] 范围内的实数。模糊集合 S 的表示法是列出元素及其隶属度（0 除外）。

模糊集的补集 元素在 $\overline{S}$ 中的隶属度等于 1 减去在 S 中的隶属度

模糊集的并集 元素在两个集合 S，T 中的隶属度最大值

模糊集的交集 元素在两个集合 S，T 中的隶属度最小值

3.3 函数 functions

也称：映射 mapping, 或变换 transformation. 对于 $A \to B$ 的函数，A 中元素在 B 中只能有一个指派。

3.3.1 一对一函数 one-to-one 或单射函数 injection

$a, b \in A, \quad \forall a \forall b (f(a) = f(b) \to a = b)$

3.3.2 映上函数 onto 或满射函数 surjection

即对 B 中每个元素 b 都是定义域 A 中某个元素的像！

3.3.3 一一对应或双射函数 bijeciton

即，既是单射，又是满射！

假设 $f: A \to B$。

要证明 f 是单射的：证明对于任意 x，$y \in A$，如果 $f(x) = f(y)$，则 $x = y$。

要证明 f 不是单射的：找到特定的 x，$y \in A$，使得 $x \neq y$ 且 $f(x) = f(y)$。

要证明 f 是满射的：考虑任意元素 $y \in B$，并找到一个元素 $x \in A$ 使得 $f(x) = y$。

要证明 f 不是满射的：找到一个特定的 $y \in B$，使得对于任意 $x \in A$ 有 $f(x) \neq y$。

3.3.4 反函数

f 为 A 到 B 的一一对应。f^{-1}：它指派给 b 的元素是 A 中使得 f(a)=b 唯一的元素 a。即有：当 $f(a) = b$ 时 $f^{-1}(b) = a$。

3.3.5 函数组成

$(f \circ g)(a) = f(g(a))$，注意：构造函数与它的反函数不论以什么次序合成, 得到的都是恒等函数！

3.3.6 函数的图

f 的图为：序偶集合 $\{(a,b) \mid a \in A \quad and \quad f(a) = b\}$

（n 为整数，x 为实数）

(1a) $\lfloor x \rfloor = n$ 当且仅当 $n \leqslant x < n+1$
(1b) $\lceil x \rceil = n$ 当且仅当 $n-1 < x \leqslant n$
(1c) $\lfloor x \rfloor = n$ 当且仅当 $x-1 < n \leqslant x$
(1d) $\lceil x \rceil = n$ 当且仅当 $x \leqslant n < x+1$
(2) $x-1 < \lfloor x \rfloor \leqslant x \leqslant \lceil x \rceil < x+1$
(3a) $\lfloor -x \rfloor = -\lceil x \rceil$
(3b) $\lceil -x \rceil = -\lfloor x \rfloor$
(4a) $\lfloor x+n \rfloor = \lfloor x \rfloor + n$
(4b) $\lceil x+n \rceil = \lceil x \rceil + n$

3.4 序列与求和

序列 sequence 是一个从整数集的一个子集到一个集合 S 的**函数**！用记号 a_n 表示**整数 n 的像**。a_n 是序列的一个项 term。

几何级数是这样的序列: $a, ar, , ar^2, \ldots, ar^n, \ldots$. 可知，几何级数是指数函数 $f(x) = ar^x$ 的离散的对应体。

递推关系

为序列的项找到一个显式公式，称为**闭公式 closed formula**, 这时边说解决了带有初始条件的递推关系！

求和 Summations$\sum$

求和式**下标平移** $\sum_{j=1}^{5} j^2 = \sum_{k=0}^{4}(k+1)^2$, $\sum_{s\in\{0,2,4\}} s$ 表示将集合中的元素全部相加!

3.5 集合的基数 Cardinality

一个有限集合 (finite sets) 的基数定义为该集合的元素个数。现扩展到无限 (infinite sets) 集合。

集合 A 与集合 B 有相同的基数，当且仅当存在从 A 到 B 的一个**一一对应**。写成 $|A| = |B|$。

若存在一个从 A 到 B 的**一对一函数**，则 $|A| \leqslant |B|$。进一步，若 A,B 的基数不同，则 $|A| < |B|$

3.5.1 可数集 countable sets

把无线集分为两组，一组是与自然数集有相同的基数，另一组是具有不同基数。

正整数集, 整数集，有理数集是可数无限集。**实数集**是不可数的!

- 可数的: 集合或是有限集，或与自然数集有相同的基数。记为 $|S| = \aleph_0$, 称作阿里夫零。
- 不可数的: 集合不可数

无限集是可数的当且仅当可以把集合中的元素排列成序列(下标是正整数)。这是因为从正整数集合到集合 S 的一一对应关系 f 可以用序列 a_1，a_2，…，a_n，…表示，其中 $a_1 = f(1)$，$a_2 = f(2)$，…，$a_n = f(n)$，…。

Theorem 2 *若 A,B 是可数集合，则 $A \cup B$ 也是可数集合*

Theorem 3 *若集合 A,B 存在 $|A| \leqslant |B|$ 和 $|B| \leqslant |A|$, 那么 $|A| = |B|$*

3.6 矩阵 matrix

矩阵简便记法 $\boldsymbol{A} = [a_{ij}]$, 表示 A 是其第 (i,j) 元素为 a_{ij} 的矩阵。

0-1 矩阵

所有元素为 0 或 1。两个 0-1 矩阵 A,B 的并：$\boldsymbol{A} \vee \boldsymbol{B}$；A,B 的交:$\boldsymbol{A} \wedge \boldsymbol{B}$, 均

为对应元素位置的布尔运算！

两个 0-1 矩阵的**布尔积** (Boolean product)：$\boldsymbol{A} \odot \boldsymbol{B}$, 类似矩阵的普通乘积，但是要**用 $\vee$ 代替加法，用 $\wedge$ 代替乘法**。

0-1 矩阵 A 的 r 次布尔幂记作:$\boldsymbol{A}^{[r]}$, 且 $\underbrace{\boldsymbol{A}^{[r]} = \boldsymbol{A} \odot \boldsymbol{A} \odot \boldsymbol{A} \odot \cdots \odot \boldsymbol{A}}_{r个\boldsymbol{A}}$。

另外定义 $A^{[0]}$为I_n。

4 算法 Algorithm

4.1 algorithm

定义：算法是进行一项计算或解决一个问题的准确指令的有限序列。

4.2 函数的增长 The Growth of Functions