计算机问题求解 – 论题1-8 - 集合及其运算

2015年11月5日

预习检查

$$x \in T,$$

$$x = s^2, \text{ for some } s \in S,$$

smaller space

$$x = 2n + 1, \text{ for some } n \in \mathbb{Z},$$

$$x \in S.$$

问题1:
两个集合“相等”究竟是什么意思?

关于集合相等

- 集合“完全由其包含的元素所确定”。
- 因此：两个集合相等，就是“两个集合所包含的元素**完全一样**”。
  - 完全一样如何以严格的数学方式表述?
- 与集合包含的关系
  - 对任意集合A, B, $A=B$ iff. $A \subseteq B$, 且 $B \subseteq A$

基本证明方式（1）

- **直接使用集合包含或相等定义**
  - 例1：$A \cup B = B \Rightarrow A \subseteq B$
  - 例2：$A \subseteq B \Rightarrow A \cap B = A$

例1，待证结论：$A \subseteq B$

即：对任何x, $x \in A \Rightarrow x \in B$

因此：证明过程如下：

对任何x, 假设$x \in A$

<填入适当内容>

$\therefore x \in B$

因此：$A \subseteq B$

因此：证明过程如下：

对任何x, 假设$x \in A$

由集合并定义：$x \in A \cup B$

由已知条件：$A \cup B = B$

$\therefore x \in B$

因此：$A \subseteq B$

基本证明方式（2）

- **利用运算定义作逻辑等值式推演**
  - **例：$A-(B\cup C) = (A-B) \cap (A-C)$**

$A-(B\cup C)=\{x|x\in A$, but $x\notin B\cup C\}$

$=\{x|\ x\in A$, but $(x\notin B$ and $x\notin C)\}$

$=\{x|(x\in A$, but $x\notin B)$ and $(x\in A$, but $x\notin C)\}$

$= (A-B) \cap (A-C)$

另一种等价的描述方式：

$$x\in A-(B\cup C) \Leftrightarrow (x\in A) \wedge (x\notin(B\cup C)) \Leftrightarrow x\in A \wedge x\notin B \wedge x\notin C$$

$$\Leftrightarrow (x\in A \wedge x\notin B) \wedge (x\in A \wedge x\notin C)$$

$$\Leftrightarrow (x\in(A-B)) \wedge (x\in(A-C))$$

$$\Leftrightarrow x\in((A-B) \cap (A-C))$$

基本证明方式（3）

- **利用已知恒等式或等**
  - 例：**A∩B=A ⇔ A-B=**
  - 例：**A∪(A∩B) = A**
  - 例：已知**A⊕B=A⊕C,** 证明**B=C**
  - 一个比较复杂的代入的例子：
    - 利用**A ∩B=A ⇔ A⊆B**证明：

((A∪B∪C) ∩ (A∪B))-((A∪(B-C)) ∩A)=B-A

A∩B=A⇒A-B=ϕ:

A-B=ϕ⇒A∩B=A:

A∩B=(A∩B)∪

=A

=

A∪(A∩B)

= A∩(E

B=Ø⊕B

=(A⊕A)⊕B

=A⊕(A⊕B)

=A⊕(A⊕C)

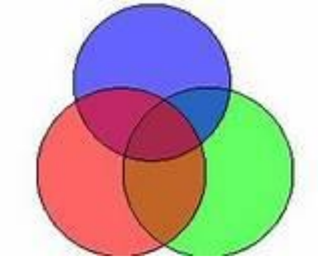
=C

(A∪B∪C) ∩ (A∪B)= (A∪B)

(A∪(B-C)) ∩A)=A, so 原式左边=(A∪B)-A=B-A

基本证明方式（4）

- **循环证明—系列逻辑等值式**
  - $A\cup B=B \Leftrightarrow A\subseteq B \Leftrightarrow A\cap B=A \Leftrightarrow A-B=\phi$
    (1) (2) (3) (4)
  - 对于上述等价命题序列，我们只需要证明：
    $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (4) \Rightarrow (1)$
  - 在以上例子的基础上，只要再证明：
    - $A-B=\phi \Rightarrow A\cup B=B$。
    - 注意：$A\cup B= (A\cup B) \cap E=(A\cup B) \cap(\sim B\cup B)$
      $=(A\cap\sim B)\cup B = B$



问题2：

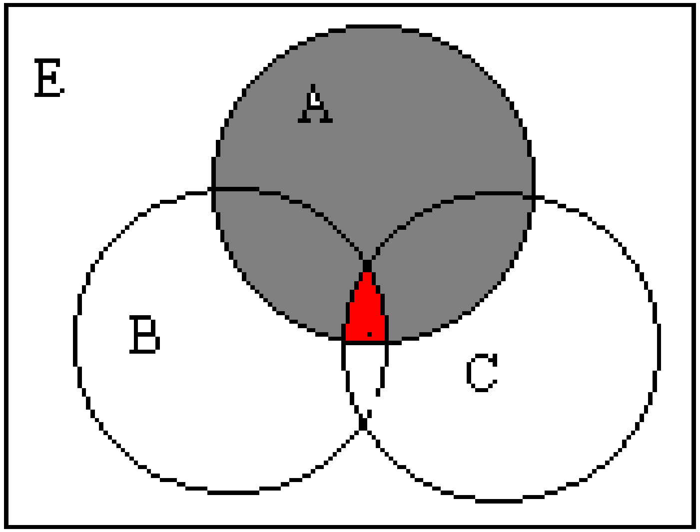
文氏图是否可以用于关于集合的数学证明？

文氏图与数学证明

- 文氏图不能代替数学证明，但可以帮助推测结论
- 例子：
  - (1) (A-B)∪(A-C)=A

充要条件：

$$A\cap B\cap C=\phi$$



证明从图形得到的猜想

- $(A-B)\cup(A-C)=A$ 当且仅当 $A\cap B\cap C=\phi$

  $\Rightarrow$ 假设$A\cap B\cap C\neq\phi$，即：存在$x\in A\cap B\cap C$,

  则$x\notin(A-B)$, $x\notin(A-C)$, $\therefore$ $x\notin(A-B)\cup(A-C)$, 但由已知：$(A-B)\cup(A-C)=A$，矛盾。$\therefore$ $A\cap B\cap C=\phi$

  $\Leftarrow$ 根据相对补运算定义，$A-B\subseteq A$, $A-C\subseteq A$; 假设 $(A-B)\cup(A-C)\neq A$，则$(A-B)\cup(A-C)$是A的真子集; 则存在$x\in A$,但$x\notin(A-B)\cup(A-C)$,即$x\notin(A-B)$ 且 $x\notin(A-C)$, 由相对补运算定义，$x\in A\cap B\cap C$, 与已知条件矛盾，$\therefore$ $(A-B)\cup(A-C)=A$

集合定律

Let X denote a set, and A, B, and C denote subsets of X. Then

1. $\emptyset \subseteq A$ *and* $A \subseteq A$.
2. $(A^c)^c = A$.
3. $A \cup \emptyset = A$.
4. $A \cap \emptyset = \emptyset$.
5. $A \cap A = A$.
6. $A \cup A = A$.
7. $A \cap B = B \cap A$. *(Commutative property)*
8. $A \cup B = B \cup A$. *(Commutative property)*
9. $(A \cup B) \cup C = A \cup (B \cup C)$. *(Associative property)*
10. $(A \cap B) \cap C = A \cap (B \cap C)$. *(Associative property)*
11. $A \cap B \subseteq A$.
12. $A \subseteq A \cup B$.
13. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$. *(Distributive property)*
14. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$. *(Distributive property)*
15. $X \setminus (A \cup B) = (X \setminus A) \cap (X \setminus B)$. *(DeMorgan's law)*
    (When X is the universe we also write $(A \cup B)^c = A^c \cap B^c$.)
16. $X \setminus (A \cap B) = (X \setminus A) \cup (X \setminus B)$. *(DeMorgan's law)*
    (When X is the universe we also write $(A \cap B)^c = A^c \cup B^c$.)
17. $A \setminus B = A \cap B^c$.

逻辑定律

(DeMorgan's laws) $\neg(P \vee Q) \leftrightarrow (\neg P \wedge \neg Q)$;
$\neg(P \wedge Q) \leftrightarrow (\neg P \vee \neg Q)$;

(Distributive property) $(P \wedge (Q \vee R)) \leftrightarrow ((P \wedge Q) \vee (P \wedge R))$;
$(P \vee (Q \wedge R)) \leftrightarrow ((P \vee Q) \wedge (P \vee R))$;

(Double negation) $\neg(\neg P) \leftrightarrow P$;

(Associative property) $(P \wedge (Q \wedge R)) \leftrightarrow ((P \wedge Q) \wedge R)$;
$(P \vee (Q \vee R)) \leftrightarrow ((P \vee Q) \vee R)$;

(Commutative property) $(P \wedge Q) \leftrightarrow (Q \wedge P)$;
$(P \vee Q) \leftrightarrow (Q \vee P)$.

问题3：

你是否觉得有关集合运算的性质都“似曾相识”，你想到过为什么吗？

逻辑表达式与集合表达式如何“对应”？

问题4：

如果一个问题的解可以通过枚举有限多个子集来得到，这个问题“难”吗？

一个例子

- 美国的总统大选采用一种非常特别的“选举人”制度。
- 每个州的选举人数是事先确定的。
- 问题：有“平局”的可能吗？

一个著名的“难”问题 – 背包问题

Input: A positive integer b, and positive integers $w_1, w_2, \ldots, w_n$ for some $n \in \mathbb{N} - \{0\}$.

Constraints: $\mathcal{M}(b, w_1, w_2, \ldots, w_n) = \{T \subseteq \{1, \ldots, n\} \mid \sum_{i \in T} w_i \leq b\}$, i.e., a feasible solution for the problem instance $b, w_1, w_2, \ldots, w_n$ is every set of objects whose common weight does not exceed b.

Costs: For each $T \in \mathcal{M}(b, w_1, w_2, \ldots, w_n)$,

$$cost(T, b, w_1, w_2, \ldots, w_n) = \sum_{i \in T} w_i.$$

Goal: *maximum*.

优化问题
简化版本

Input: A positive integer b, and $2n$ positive integers $w_1, w_2, \ldots, w_n$, c_1, $c_2, \ldots, c_n$ for some $n \in \mathbb{N} - \{0\}$.

Constraints:

$$\mathcal{M}(b, w_1, \ldots, w_n, c_1, \ldots, c_n) = \{T \subseteq \{1, \ldots, n\} \mid \sum_{i \in T} w_i \leq b\}.$$

Costs: For each $T \in \mathcal{M}(b, w_1, \ldots, w_n, c_1, \ldots, c_n)$,

$$cost(T, b, w_1, \ldots, w_n, c_1, \ldots, c_n) = \sum_{i \in T} c_i.$$

Goal: *maximum*.

优化问题
一般版本

另一个关于子集的问题：精确覆盖问题

- 问题的描述:
  - Given a set A and a finite set of subsets of A: $\{A_1, A_2, \ldots A_k\}$, a **exact cover** of A with respect to the A_i's is a set $S \subseteq \{A_1, A_2, \ldots A_k\}$, satisfying:
    - Any two sets in S are disjoint, and
    - $\cup S = A$
  - Mathematically, we call S a partition of A.
- 一个例子:
  - $A=\{a,b,c,d,e,f,g,h,i,j\}$; $A_1=\{a,c,d\}$, $A_2=\{a,b,e,f\}$, $A_3=\{b,f,g\}$, $A_4=\{d,h,i\}$, $A_5=\{a,h,j\}$, $A_6=\{e,h\}$, $A_7=\{c,i,j\}$, $A_8=\{i,j\}$
  - 解是： $\{A_1, A_3, A_6, A_8\}$

精确覆盖问题的矩阵表示

Let $|A|=n$, and there are m subsets for A_i's, we can represent the input of exact cover problem as a $m \times n$ matrix, with each row for a A_i .

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Solution:

Find a collection of rows of M: r_1, $r_2,\ldots r_k$, satisfying:

$r_i \wedge r_j = \mathbf{0}$ for $1 \le i,j \le k$, and

$r_1 \vee r_2 \vee \ldots \vee r_k = \mathbf{1}$

where $\mathbf{0}$=[0 0 0 0 0 0 0 0 0 0]

$\mathbf{1}$=[1 1 1 1 1 1 1 1 1 1]

and, $\wedge$ is boolean product, $\vee$ is boolean sum

Knuth's X Algorithm

- input: matrix *A*
- Initialization: label the rows of *A*;
- M=A; L={};
- (1) If there is a column of 0's in *M*, return "No solution"
- (2) Otherwise:
  - Choose the column *c* with the fewest 1's;
  - Choose a row *r* with a 1 in column *c*, $L=L\cup\{r\}$;
  - Eliminate any row r_i having the property: $r\wedge r_i\neq\mathbf{0}$;
  - Eliminate all columns in which *r* has a 1;
  - Eliminate row *r*;
  - If No row and column left, then output L, otherwise repeat (1) and (2) on resulted *M*;

$$M = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$M =$

1	0	1	1	0	0	[redacted]	0	0	0
[redacted]									
0	1	0	0	0	1	[redacted]	0	0	0
0	0	0	1	0	0	[redacted]	1	1	0
1	0	0	0	0	0	[redacted]	1	0	1
0	0	0	0	1	0	[redacted]	1	0	0
0	0	1	0	0	0	[redacted]	0	1	1
0	0	0	0	0	0	[redacted]	0	1	1

$M =$

1	0	1	1	0	0	[redacted]	0	0	0
[redacted]									
0	1	0	0	0	1	[redacted]	0	0	0
[redacted]									
0	0	0	0	1	0	[redacted]	1	0	0
0	0	1	0	0	0	[redacted]	0	1	1
0	0	0	0	0	0	[redacted]	0	1	1

$$M = \begin{pmatrix} \text{[redacted]} & 0 & 1 & 1 & 0 & 0 & \text{[redacted]} & 0 & 0 & 0 \\ \text{[redacted]} \\ \text{[redacted]} & 1 & 0 & 0 & 0 & 1 & \text{[redacted]} & 0 & 0 & 0 \\ \text{[redacted]} \\ \text{[redacted]} \\ \text{[redacted]} & 0 & 0 & 0 & 1 & 0 & \text{[redacted]} & 1 & 0 & 0 \\ \text{[redacted]} \\ \text{[redacted]} & 0 & 0 & 0 & 0 & 0 & \text{[redacted]} & 0 & 1 & 1 \end{pmatrix}$$

$M =$

1	0	1	1	0	0	0	0	0	0
[redacted]									
0	1	0	0	0	1	1	0	0	0
[redacted]									
0	0	0	0	1	0	0	1	0	0
[redacted]									
0	0	0	0	0	0	0	0	1	1

结果是: $\{A_1, A_3, A_6, A_8\}$

1	3	2
3	2	1
2	1	3

问题5:

你能否用集合模型描述“数独”（Soduku)问题?

简单一点，且考虑3×3的。

精确覆盖问题与“数独”

1	3	2
3	2	1
2	1	3

- Describe the problem in suitable form
  - 9 constraints for cells
    - Each cell contains exactly one number
    - e.g.,
      - $C1R1 = \{C1R1\#1, C1R1\#2, C1R1\#3\}$
      - $C1R2 = \{C1R2\#1, CR21\#2, C1R2\#3\}$
  - 9 constraints for numbers-columns
    - Each number exists in one column exactly once
    - e.g.,
      - $N1C1 = \{C1R1\#1, C1R2\#1, C1R3\#1\}$
      - $N2C1 = \{C1R1\#2, C1R2\#2, C1R3\#2\}$
  - 9 constraints for numbers-rows
    - Each number exists in one row exactly once
    - e.g.,
      - $N1R1 = \{C1R1\#1, C2R1\#1, C3R1\#1\}$
      - $N1R2 = \{C1R2\#1, C2R2\#1, C3R2\#1\}$

在C1R1
填入3

Construct the matrix of covering
27 columns for constraints
27 rows for “moves”

http://www.iroowe.com/sudoku-and-exact-cover%ef%bc%88%e6%95%b0%e7%8b%ac%e4%b8%8e%e5%ae%8c%e5%85%a8%e8%a6%86%e7%9b%96%ef%bc%89/

矩阵的构建

- 矩阵有27列，矩阵的行数依赖于数独的初始布局
- 前9列定义第一类约束(cell, Col-Row)
- 中间9列定义第二类约束(Num-Col)
- 后9列定义第三类约束(Num-Row)

初始为空

		CjRi									NkCi									NkRi								
		C1R1	C1R2	C1R3	C2R1	C2R2	C2R3	C3R1	C3R2	C3R3	N1C1	N1C2	N1C3	N2C1	N2C2	N2C3	N3C1	N3C2	N3C3	N1R1	N1R2	N1R3	N2R1	N2R2	N2R3	N3R1	N3R2	N3R3
填入1	C1R1#1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	C1R2#1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	C1R3#1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	C2R1#1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	C2R2#1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	C2R3#1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	C3R1#1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	C3R2#1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	C3R3#1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
填入2	C1R1#2	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0
	C1R2#2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	C1R3#2	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0
	C2R1#2	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
	C2R2#2	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
	C2R3#2	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
	C3R1#2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
	C3R2#2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
	C3R3#2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
填入3	C1R1#3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
	C1R2#3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0
	C1R3#3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
	C2R1#3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
	C2R2#3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
	C2R3#3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
	C3R1#3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
	C3R2#3	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
	C3R3#3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

填在(i,j) 中 — k 在第 i 行 — k 在第 i 列

课外作业

- problems: 6.7, 6.12, 6.14-18;
- problems: 7.1, 7.8-11;
- problems: 8.1, 8.4, 8.7, 8.8, 8.9, 8.11;
- problems: 9.2, 9.4, 9.12-14, 9.16
- project 27.3