

2019-1 Review

魏恒峰

hfwei@nju.edu.cn

2019 年 12 月 26 日



Are You Ready ?

1. “算一算” (Let us Calculate!)

(1) 某公司要从赵、钱、孙、李、吴 5 名员工中选派某些人出国考察。由于某些不可描述的原因, 选派要求如下:

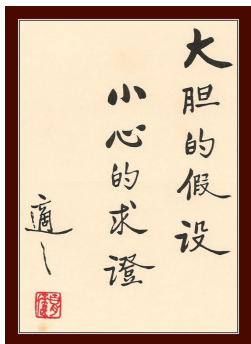
- | | |
|-------------------|--|
| (1) 若赵去, 钱也去; | (1) $Z \rightarrow Q$; |
| (2) 李、吴两人中必有一人去; | (2) $L \vee W$; |
| (3) 钱、孙两人中去且仅去一人; | (3) $(Q \wedge \neg S) \vee (S \wedge \neg Q)$; |
| (4) 孙、李两人同去或同不去; | (4) $(S \wedge L) \vee (\neg S \wedge \neg L)$; |
| (5) 若吴去, 则赵、钱也去; | (5) $W \rightarrow Z \wedge Q$; |
| (6) 只有孙去, 赵才会去。 | (6) $Z \rightarrow S$ 。 |

请使用形式化推理的方法帮该公司判断应选哪些人出国考察。

Z, Q, S, L, W

$$\begin{aligned} & (1) \wedge (2) \wedge (3) \wedge (4) \wedge (5) \wedge (6) \\ & = \dots \\ & = \text{ONE PAGE HERE} \dots \\ & = \neg Z \wedge \neg Q \wedge S \wedge L \wedge \neg W \end{aligned}$$





1. “算一算” (Let us Calculate!)

(2) 给定如下“前提”，请判断“结论”是否有效，并说明理由。
前提如下：

(1) 每个人或者喜欢美剧，或者喜欢韩剧（可以同时喜欢二者）；
 $\forall x : A(x) \vee K(x)$

(2) 任何人如果他喜欢抗日神剧，他就不喜欢美剧；
 $\forall x : J(x) \rightarrow \neg A(x)$

(3) 有的人不喜欢韩剧。
 $\exists x : \neg K(x)$

结论：有的人不喜欢抗日神剧。 $\exists x : \neg J(x)$

x : Human $Q : H(x)?$

$A(x), \quad K(x), \quad J(x)$

2. 常用证明方法

证明: 从 $\{1, 2, 3, \dots, 3n\}$ ($n \in \mathbb{Z}^+$) 中任选 $n + 1$ 个数, 则总存在两个数, 它们的差不超过 2。

Proof by the pigeonhole principle:

$$\{1, 2, 3\}, \quad \{4, 5, 6\}, \quad \dots, \quad \{3n - 2, 3n - 1, 3n\}$$



Proof by contradiction:

$$1, 4, 7, \dots, 3n + 1$$



3. 集合的势 (Cardinality)

A 是由所有半径为有理数、圆心在 x 轴 (实数轴) 上的圆组成的集合。
请问 A 的势是什么, 并给出证明。

$$|\mathbb{R}| \leq |\mathbb{Q} \times \mathbb{R}| \leq |\mathbb{R} \times \mathbb{R}| = |\mathbb{R}|$$

Definition ($|A| \leq |B|$)

$|A| \leq |B|$ if there exists an *one-to-one* function f from A into B .

$Q : \text{Is “}\leq\text{” a partial order?}$

Theorem (Cantor-(Dedekind)-Schröder-Bernstein (1887))

$$|X| \leq |Y| \wedge |Y| \leq |X| \implies |X| = |Y|$$

$$\exists f : A \xrightarrow{1-1} B \wedge g : B \xrightarrow{1-1} A \implies \exists h : A \xleftrightarrow[\text{onto}]{1-1} B$$

By Julius König (1906).

Suppose that A and B are disjoint.

$$A \uplus B$$

$$a \in A : \cdots \rightarrow f^{-1}(g^{-1}(a)) \rightarrow g^{-1}(a) \rightarrow a \rightarrow f(a) \rightarrow g(f(a)) \rightarrow \cdots$$

$$(i) \quad \cdots \leadsto \cdots$$

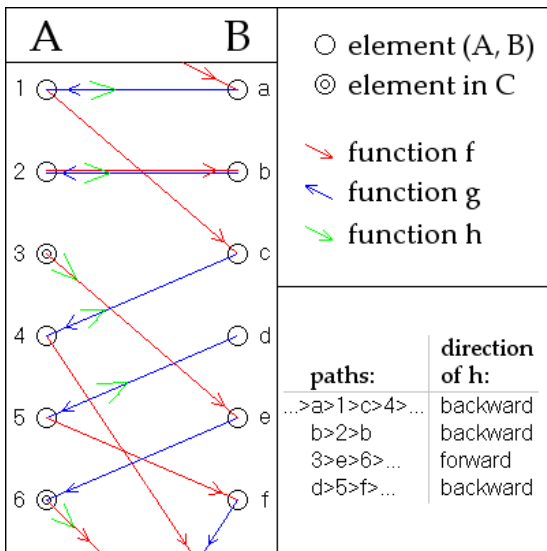
$$(ii) \quad a \in A \leadsto \cdots$$

$$(iii) \quad b \in B \leadsto \cdots$$

$$(iv) \quad a \in A \leadsto a \in A$$

Partition of $A \uplus B$





4. 关系与序 (Order)

一个自反 (reflexive) 且传递 (transitive) 的二元关系 $R \subseteq X \times X$ 称为 X 上的拟序 (preorder/quasiorder)。

令 $\leq \subseteq X \times X$ 为拟序, 请证明:

(1) 如果定义 X 上的关系 \sim 为

$$x \sim y \triangleq x \leq y \wedge y \leq x,$$

则 \sim 是 X 上的等价关系 (equivalence relation)。

reflexive + symmetric + transitive

4. 关系与序 (Order)

一个自反 (reflexive) 且传递 (transitive) 的二元关系 $R \subseteq X \times X$ 称为 X 上的拟序 (preorder/quasiorder)。

令 $\leq \subseteq X \times X$ 为拟序, 请证明:

(2) 如果定义商集 (quotient set) X/\sim 上的关系 \preceq 为

$$[x]_{\sim} \preceq [y]_{\sim} \triangleq x \leq y,$$

则 \preceq 是偏序关系 (partial order)。

reflexive + antisymmetric + transitive

Well-definedness!!!

Well-definedness: Independence of Representative

$$[x_1] = [x_2] \wedge [y_1] = [y_2]$$



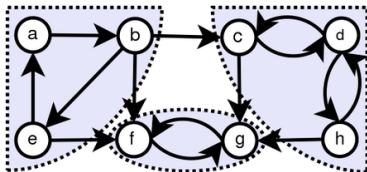
$$[x_1] \preceq [y_1] \iff [x_2] \preceq [y_2]$$

$$[a]_n + [b]_n = [a + b]_n$$

$$[a]_n \times [b]_n = [ab]_n$$



\preceq : ??? Reachability relationship in a directed graph



$\sim, [x]_{\sim}$: Strongly Connected Component (SCC)

\preceq : Reachability relationship in a condensed directed acyclic graph

7. 算法设计与正确性证明

在“掼蛋”游戏中, 5 张大小连续的扑克牌构成一个顺子 (如 $A\ 2\ 3\ 4\ 5$ 和 $10\ J\ Q\ K\ A$ 都是顺子; 不考虑花色)。

任给 13 张从小到大的牌 (允许不同花色重复, 如 $A\ 3\ 3\ 4\ 5\ 7\ 8\ 9\ 10\ J\ J\ Q\ K$):

- (1) 请设计算法, 找到所有的顺子。
- (2) 请使用“循环不变式” (loop invariants) 证明你设计的算法的正确性。
- (3) 数学归纳法的正确性也是需要证明的。请证明第一数学归纳法的正确性。(不允许使用第二数学归纳法证明。)

7. 算法设计与正确性证明

(1) 任给 13 张从小到大的牌, 请设计算法, 找到所有的顺子。

Preprocessing:

$A\ 3\ 3\ 4\ 5\ 7\ 8\ 9\ 10\ J\ J\ Q\ K$

$A\ 3\ 4\ 5\ 7\ 8\ 9\ 10\ J\ Q\ K$

$A\ 3\ 4\ 5\ 7\ 8\ 9\ 10\ J\ Q\ K\ A$

“Sliding Window” Algorithm:

```
i ← 1 // starting index of the sliding window
cnt ← 1
while (i ≤ n - 4) // n: # of cards
    while (cnt ≠ 5)
        if (P[i + cnt] == P[i + cnt - 1] + 1)
            cnt++
        else // fail: skip cnt
            i ← i + cnt
            cnt ← 1
            break
    if (cnt == 5) // succeed: slid by one
        print P[i] ... P[i + cnt - 1]
        i++
        cnt ← 4 // only need to check the new card
```



HAPPINESS IS



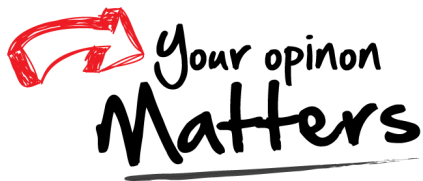
...an exam going well
after a night of studying.

HAP3622
facebook.com/itsthehappyguy

(c) lastlemon.com 2014



Thank
You!



Office 302

Mailbox: H016

hfwei@nju.edu.cn