离散数学

Discrete Mathematics

邱一航

F2003802

5200309101S5

Lecturer: Qinxiang Cao
(曹钦翔, SJTU-JHC)

PROPOSITIONAL LOGIC 命题逻辑

- Proposition, Truth Value
- Logic Connectives

p, q stands for two propositions. — propositional variables (命题变元)

- $\neg p$ "Not p". 非
- $p \wedge q$ "p And q" 与
- $p \vee q$ "p Or q" 或

} Logic Connectives 逻辑连接词

Def. J is a truth assignment. (真值指派) ← J could be viewed as a mapping

if $J: \Sigma \longrightarrow \{T, F\}$. $(\Sigma = \{p, q, r, \cdots\})$

Def. $[\![\phi]\!]_J$ (denotation) is recursively defined:

(1) $[\![p]\!]_J = J(p)$ (atom) 原子命题

(2) $[\![\phi \wedge \psi]\!]_J = [\![\wedge]\!]([\![\phi]\!]_J, [\![\psi]\!]_J)$

(3) $[\![\phi \vee \psi]\!]_J = [\![\vee]\!]([\![\phi]\!]_J, [\![\psi]\!]_J)$

(4) $[\![\neg \phi]\!]_J = [\![\neg]\!]([\![\phi]\!]_J)$

理解为语义

Def. ϕ is a tautology (永真式 / 重言式)

iff $[\![\phi]\!]_J = T$ for any J.

Def. ϕ is a contradiction (矛盾式)

iff $[\![\phi]\!]_J = F$ for any J.

Def. ϕ is a contingency iff it's not a tautology or a contradiction.

Def. ϕ is satisfiable iff there exists a J such that

$[\![\phi]\!]_J = T$.

It's obvious that a proposition is either a satisfiable proposition or a contradiction.

Proposition { satisfiable proposition { tautology / contingency } / contradiction }

- Satisfiability

Def. $J \vDash \phi$ is recursively defined:

(Next Page)

一个真值指派

"$J \vDash \phi$": J 满足 ϕ

一个命题，类似一种性质

$\vDash$: 满足

Def. $\mathcal{J} \models \phi$ is recursively defined:

(1) $\mathcal{J} \models \phi$ iff $\mathcal{J}(p) = T$ (atom)

(2) $\mathcal{J} \models \phi \wedge \psi$ iff $\mathcal{J} \models \phi$ and $\mathcal{J} \models \psi$

object language 对象语言 — meta language 元语言

(3) $\mathcal{J} \models \phi \vee \psi$ iff $\mathcal{J} \models \phi$ or $\mathcal{J} \models \psi$

(4) $\mathcal{J} \models \neg\phi$ iff $\mathcal{J} \not\models \phi$

In fact,

$$\mathcal{J} \models \phi \iff [\![\phi]\!]_{\mathcal{J}} = T$$
$$\mathcal{J} \not\models \phi \iff [\![\phi]\!]_{\mathcal{J}} = F$$

- Relationships between propositions

Def. Φ is a set of propositions.
ψ is a proposition.

$\Phi \models \psi$ means for any $\mathcal{J}$, if $[\![\phi]\!]_{\mathcal{J}} = T$ for any $\phi \in \Phi$, then $[\![\psi]\!]_{\mathcal{J}} = T$

Notations: $\phi \models \psi$ means $\{\phi\} \models \psi$

$\phi_1, \phi_2, \phi_3 \cdots \phi_n \models \psi$ means $\{\phi_1, \phi_2, \cdots \phi_n\} \models \psi$

$\Phi, \phi_1, \phi_2, \cdots \phi_n \models \psi$ means $\Phi \cup \{\phi_1, \phi_2, \cdots, \phi_n\} \models \psi$

Example. $p \vee q, \neg p \vee r \models q \vee r$

$p, q \models p \wedge q$

$p \models p \vee q$

$\neg p \models p \wedge q$

$\Phi \models \psi$ reads "ψ is a consequence of Φ." or "Φ entails ψ".

* You could say Φ is stronger than ψ. 语义后承

Def. $\phi \equiv \psi$ iff for any $\mathcal{J}$, $[\![\phi]\!]_{\mathcal{J}} = [\![\psi]\!]_{\mathcal{J}}$

(logically equivalent)
逻辑等价

[DETAILS] An atom proposition is definitely a contingency.

When $\phi \equiv \psi$, ϕ and ψ doesn't have to relate to each other.

E.g. $(\neg p) \vee p \equiv (\neg q) \vee q$.

So in fact all tautologies is logically equivalent to each other.
(if their included propositional variables are given truth value under the same truth assignment $\mathcal{I}$)

[ADDITION] ABOUT "$\models$"

(1) $\mathcal{I} \models \phi$ ← a proposition "满足"
↑
Truth assignment

(2) $\Phi \models \psi$ ← a propositions "推出"
↑
A set of propositions

(3) $\mathcal{I} \models \Phi$ means under the truth assignment $\mathcal{I}$, all propositions in Φ is true.
($[\![\phi]\!]_{\mathcal{I}} = T$, for any $\phi \in \Phi$)

- Laws of Logic Connectives

$\phi \vee \neg\phi$ is a tautology.
$\phi \wedge \neg\phi$ is a contradiction.
$\phi, \psi \models \phi \wedge \psi$ ($\wedge$-Introduction)
$\phi \wedge \psi \models \phi$ ($\wedge$-Elimination)
$\phi \models \phi \vee \psi$ ($\vee$-Introduction)
if $\Phi, \phi_1 \models \psi$, $\Phi, \phi_2 \models \psi$. then $\Phi, \phi_1 \vee \phi_2 \models \psi$

$\wedge$ "and" conjunction 合取
$\vee$ "or" disjunction 析取

Thm. Contrapositive (逆否律)
if $\Phi, \neg\phi_1 \models \phi_2$, then $\Phi, \neg\phi_2 \models \phi_1$

PROOF Given $\mathcal{I}$. Assume $\mathcal{I} \models \Phi$, $\mathcal{I} \models \neg\phi_2$.
We need to prove that $\mathcal{I} \models \phi_1$
Proof by contradiction. Assume that $\mathcal{I} \not\models \phi_1$, which means $\mathcal{I} \models \neg\phi_1$.
Thus, $\mathcal{I} \models \phi_2$, which is a contradiction under assumption that $\mathcal{I} \models \neg\phi_2$.

Qed

Thm. $\neg(\neg\phi) \equiv \phi$ (Double Negation)
$\phi \vee \phi \equiv \phi$, $\phi \wedge \phi \equiv \phi$ (Idempotent Laws, 幂律)
$\phi \vee \psi \equiv \psi \vee \phi$, $\phi \wedge \psi \equiv \psi \wedge \phi$ (Commutative Laws, 交换律)
$(\phi \vee \psi) \vee \chi \equiv \phi \vee (\psi \vee \chi)$, $(\phi \wedge \psi) \wedge \chi \equiv \phi \wedge (\psi \wedge \chi)$ (Associative Laws, 结合律)
$\phi \vee (\psi \wedge \chi) \equiv (\phi \vee \psi) \wedge (\phi \vee \chi)$
$\phi \wedge (\psi \vee \chi) \equiv (\phi \wedge \psi) \vee (\phi \wedge \chi)$
(Distributive Laws, 分配律)

* $\phi \vee (\phi \wedge \psi) \equiv \phi$
* $\phi \wedge (\phi \vee \psi) \equiv \phi$
(Absorption Law, 吸收律)

$\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi$
$\neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$
} (De Morgan's Law, 德摩根定律)

If we need to use these laws to prove two propositions are logically equivalent, we obviously need these theorems:

Thm. Transitivity (传递性)

If $\phi_1 \equiv \phi_2$, $\phi_2 \equiv \phi_3$, then $\phi_1 \equiv \phi_3$.

Thm.

- Relationships Between Tautology, Contradiction, Satisfiability and Consequence

ϕ is a tautology if and only if $\neg\phi$ is a contradiction.
contradiction tautology.

ϕ is satisfiable if and only if ϕ is not a contradiction.
a tautology $\neg\phi$ is not satisfiable

Thm. $\phi_1, \phi_2, \cdots \phi_n \vDash \psi$ iff $\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_n \wedge \neg\psi$ is not satisfiable.
$\{\} \vDash \phi$ iff ϕ is a tautology.
$\phi \equiv \psi$ iff $\phi \vDash \psi$ and $\psi \vDash \phi$.

Thm. If $\phi \equiv \psi$, then
ϕ is a tautology iff ψ is a tautology.
ϕ is a contradiction iff ψ is a contradiction.

Thm. If $\phi \vDash \psi$ and $\psi \vDash \chi$, then $\phi \vDash \chi$
(≡) (≡)

- Normal Form (范式)

Def. Disjunctive Normal Form, or DNF

(1) Literal (文字) $p, q, \cdots, \neg p, \neg q, \cdots$

(2) conjunctive clauses (合取子句): the conjunction of literals.
$\neg p \wedge q$, $p \wedge \neg q \wedge r$

(3) DNF: the disjunction of conjunctive clauses.

Specially, for $p \wedge q \wedge r$, different essays have different definitions.
We make it a rule only in this note that only one literal is a conjunctive clause and only one conjunctive clause is a DNF.

Thm. For any given proposition ϕ, there exists a ψ in DNF such that (s.t.) $\phi \equiv \psi$.

Proof In a truth value table, for any line where ϕ is True, we construct a conjunctive clause whose truth value is true. Then we get a feasible ψ, which is the disjunction of these clauses.

Now we prove $\phi \equiv \psi$, namely

$$\phi \equiv \bigvee_{[\![\phi]\!]_J = T} \left(\bigwedge_{J(p_i)=T} p_i \wedge \bigwedge_{J(p_i)=F} \neg p_i \right) := \psi$$

$$[\![\phi]\!]_{J_0} \equiv \left[\!\!\left[\bigvee_{[\![\phi]\!]_J = T} \left(\bigwedge_{J(p_i)=T} p_i \wedge \bigwedge_{J(p_i)=F} \neg p_i \right) \right]\!\!\right]_{J_0}$$

当 $[\![\phi]\!]_{J_0} = T$, 显然成立

当 $[\![\phi]\!]_{J_0} = F$. 取 $[\![\phi]\!]_{J_1} = T$. J_0 与 J_1 中必有n个命题变元不同.

对每个合取子句,

对比 J_0 与 J_1

必有 literal 的值为 F.

故每个合取子句都是 F.

* 实际上每个子句只有唯一一种情况是 T

其余情况全为 F.

Proof' We can build DNF(ϕ) recursively:

(1) DNF(p) = p

(2) DNF($\phi \vee \psi$) = DNF(ϕ) $\vee$ DNF(ψ)

(3) DNF($\phi \wedge \psi$)

DNF(ϕ) = $C_1 \vee C_2 \vee \cdots \vee C_n$

DNF(ψ) = $C_1' \vee C_2' \vee \cdots \vee C_m'$

$$\begin{aligned} \text{DNF}(\phi \wedge \psi) = {} & (C_1 \wedge C_1') \vee (C_1 \wedge C_2') \vee (C_1 \wedge C_3') \vee \cdots \vee (C_1 \wedge C_m') \\ & \vee (C_2 \wedge C_1') \vee (C_2 \wedge C_2') \vee \cdots \vee (C_2 \wedge C_m') \\ & \cdots \\ & \vee (C_n \wedge C_1') \vee (C_n \wedge C_2') \vee \cdots \vee (C_n \wedge C_m') \end{aligned}$$

$$= \bigvee_{i=1}^{n} \bigvee_{j=1}^{m} (C_i \wedge C_j')$$

(4) DNF($\neg \phi$)

DNF(ϕ) = $C_1 \vee C_2 \cdots \vee C_n$

$\because \neg\phi \equiv \neg(C_1 \vee C_2 \vee \cdots \vee C_n) \equiv \neg C_1 \wedge \neg C_2 \wedge \cdots \wedge \neg C_n$

By De Morgan's law, $\neg C_1, \neg C_2, \cdots \neg C_n$ can be ... clauses, the conjunction of which is ...

$\neg C_1, \neg C_2, \cdots \neg C_n$ can be converted into ~~disjunctive~~ disjunctive clauses.

for example: if $C_i = \bigwedge_{j=1}^{n_i} C_{ij}$.

then $\neg\phi \equiv \bigwedge_{i=1}^{n} \neg C_i = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{n_i} \neg C_{ij}$

Then we could use 3) to turn it into a DNF.

Def. Conjunctive Normal Form

(1) Disjunctive Clause

(2) CNF: The conjunction of disjunctive clauses. $(p \vee q) \wedge (\neg p \vee \neg q)$

Thm. For every ϕ, there exists ψ in CNF. such that $\phi \equiv \psi$

Proof ϕ. $\neg\phi \equiv \psi'$. (ψ' in DNF)

then $\phi \equiv \neg\neg\phi \equiv \neg\psi'$. Using De Morgan's Law, $\neg\psi'$ can be converted into a CNF.

Proof' 真值表中值为 ~~T~~ True 的行，排除掉即可 In other words, we use false lines in the truth table and construct χ in DNF ~~that~~ such that $\chi \equiv \neg\phi$, then we get ψ in CNF s.t. $\psi \equiv \neg\chi$ (De Morgan's Law.)

- Functional Completeness (功能完备)

Def. A collection of logical operators S is functionally complete if for any ~~variable~~ set of propositional variables Σ and any f, which is a mapping from Σ's truth assignments to truth values, there exists a compound proposition ϕ that involves only the logical operators in S s.t. $[\![\phi]\!]_{\mathcal{I}} = f(\mathcal{I})$ for any $\mathcal{I}$.

Thm. $\{\neg, \wedge, \vee\}$ are functionally complete, and so are $\{\neg, \wedge\}, \{\neg, \vee\}$. (Using De Morgan's Law)

实质蕴含 (imply)

p	q	$p \to q$	$p \oplus q$	$p \leftrightarrow q$ (iff (if and only if))
T	T	T	F	T
T	F	F	T	F
F	T	T	T	F
F	F	T	F	T

$(p \to q) \equiv ((\neg p) \vee q)$

So $\{\neg, \to\}$ is functionally complete.

~~Functionally~~

$\{\wedge, \vee\}$ is not functionally complete.

You can't find how to express $\neg p$.

Thm. $\{\neg, \oplus\}$ is not functionally complete.

Proof. $\psi \oplus \phi \equiv \phi \oplus \psi$. $(\phi \oplus \psi) \oplus \chi \equiv \phi \oplus (\psi \oplus \chi)$

* ϕ, ψ, χ 有奇数个为真

奇数 → T $\xrightarrow{\oplus T}$ F (偶数)

偶数 → F $\xrightarrow{\oplus T}$ T (奇数)

$\neg(\phi \oplus \psi) \equiv (\neg\phi) \oplus \psi$ (相当于改变奇偶性)

So we could convert any propositions in this case into forms of:

$c_1 \oplus c_2 \oplus c_3 \oplus \cdots \oplus c_n$, $c_i \in \{p, \neg p\}$. ⟶ 只能说明有奇数T/偶数F

And obviously it's impossible to express $p \vee q$.

Thm. if $p_1 \oplus p_2 \oplus \cdots \oplus p_n \equiv \phi$, ϕ is in DNF, then ϕ has at least 2^{n-1} conjunctive clauses

Proof Any conjunctive clause stands for one or more lines where ϕ is true.

Now we prove in this case, it can represent only 1 line.

if p_i does not appear in a clause C.

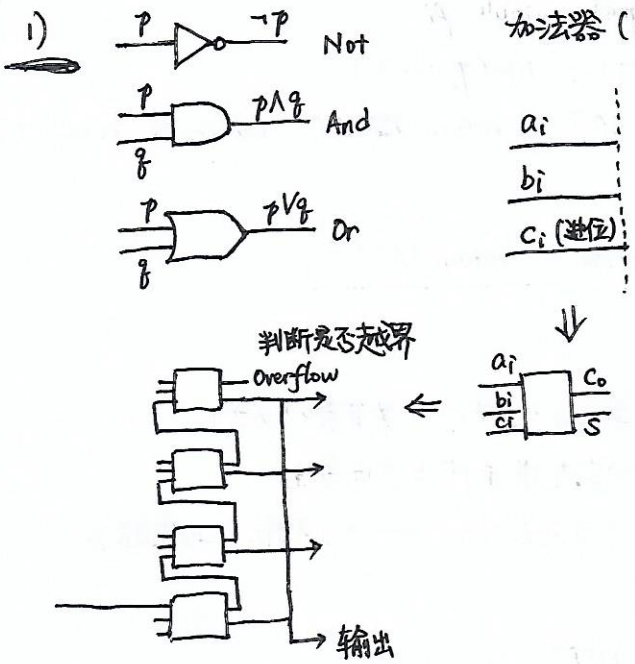
then $[\![C]\!]_{J(p_i \mapsto T)} = T$, $[\![C]\!]_{J(p_i \mapsto F)} = T$, which is impossible.

QED

直观上也易理解：间隔行为 True 和 False, $\Rightarrow \frac{1}{2} \times 2^n = 2^{n-1}$

- Application

1)



2) Sudoku

- p_{ijk} : the number k is in the position (i,j).

$\bigwedge_{ij} \bigvee_{k} p_{ijk}$ 每格一个数 $\quad \bigwedge_{ij} \bigwedge_{k_1 \neq k_2} \neg(p_{ijk_1} \wedge p_{ijk_2})$ 只有1个数 $\quad \cdots\cdots$

可写为一个命题表示数独情形 (ϕ)

$J \vDash \phi$: 有解

$J \nvDash \phi$: 无解

3) SAT Solver : whether a proposition is satisfiable.

Observations:

(1) It's easy to determine whether a DNF is SAT.

<2> hard generic proposition

<3> analyze CNF , but hard to determine.

<4> Converting a proposition into a DNF or CNF is time-consuming.

Though, we can try to find a CNF s.t. they are both SAT or UNSAT, which is much easier.
then, we analyze whether the CNF is SAT or UNSAT.

e.g. $(p \wedge q) \vee \neg(q \wedge r)$

$\Rightarrow (p \wedge q \leftrightarrow p_1) \wedge (q \wedge r \leftrightarrow p_2) \wedge (\neg p_2 \leftrightarrow p_3) \wedge (p_1 \vee p_2 \leftrightarrow p_4) \wedge p_4$

[保持3可满足性]

每个小短句都只有3个变元，2个逻辑连接词，可列真值表得到 CNF.

$\Rightarrow \cdots\cdots$ (此时长度一般为 $4n+1$; 第一步若有 n 个连接词，第一步最多 $4n$ 个子句.
第二步最多 $n \times 4 = 4n$，大部分情况可缩减)

总共 $8n$ 行 $\leftrightarrow$ 一半真一半假 $\rightsquigarrow$ $4n$ 个子句

① Brute Force

1) $\mathcal{J}$ is a total assignment, $[\![\phi]\!]_{\mathcal{J}} = F$. return UNSAT

2) $\mathcal{J}$ is a total assignment. $[\![\phi]\!]_{\mathcal{J}} = T$. return SAT

3) find an unassigned variable p_i.

$BF(p_i \mapsto T)$; $BF(p_i \mapsto F)$

if one is SAT return SAT otherwise return UNSAT

② BF^+ (剪枝优化)

if $\mathcal{J}$ causes a confliction. return UNSAT.

③ DPLL

UnitPro($\mathcal{J}$): 推导部分必然赋值 (若要整体为T)
Unit Propagation 由部分真值指派推导其它变元

若某一析取子句只有1个变元未指派 $\longrightarrow$ 可推其真值指派

- ~~若无单个变元~~
- 若冲突, return UNSAT
- 若无, 枚举下一变元

④ CDCL (Conflict Drive Clause Learning)

Using DPLL, but when we find a conflict
we add a new clause involving the picks which lead to it by unit propogation.

e.g. $p_1 \mapsto T \Rightarrow p_7 \mapsto F$

$p_2 \mapsto T$, $p_3 \mapsto T \Rightarrow p_8 \mapsto T$ → Conflict

$p_4 \mapsto T$, $p_5 \mapsto T$, $p_6 \mapsto T \Rightarrow p_9 \mapsto F$, $p_{10} \mapsto T$

then we add $\neg p_1 \vee \neg p_3 \vee \neg p_6$.

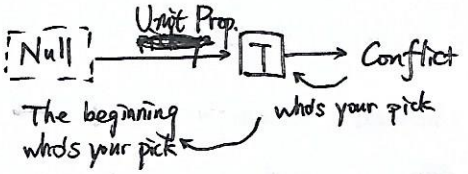
Then we get back to "$p_3 \mapsto T$" since we can find out that $p_6 \mapsto F$.

- Let $\mathcal{J}$ be empty assignment at the beginning

CDCL($\mathcal{J}$)

- set $\mathcal{J}$ to UnitPro($\mathcal{J}$)
- if $\mathcal{J}$ causes a conflict on ϕ
  - × Let D be the result of ConflictCauseGen()
  - × D is empty ⟶ return UNSAT

e.g. $p \wedge \neg p$



get "null" (empty)

  - × D isn't empty:

    add D to CNF

    remove assignments after the 2nd last pick in D from $\mathcal{J}$

    and you could figure out what the true last pick could be.

- Otherwise, if $\mathcal{J}'$ is a total assignment, return SAT
- Otherwise,
  - × pick a propositional variable p which is not assigned by $\mathcal{J}'$
  - × pick a value $t \in \{T, F\}$
  - × set $\mathcal{J}$ to $(\mathcal{J}' \cup \{p \mapsto t\})$
  - × go to step 2 (CDCL($\mathcal{J}$))

FIRST ORDER LOGIC 一阶逻辑

- Predicate logic (谓词逻辑)

  $P(x,y,z,w)$ means $x+y=z+w$. Then $P(c_3,c_5,c_6,c_2)$ means $3+5=6+2$

  ($x,y,z,\dots$: variables; P: Predicate Symbol)

  (Using c_i to represent for i; $c_1, c_2, c_3, \dots$: constants)

- Quantifier logic (量词逻辑)

  $\forall$: Universal Quantifier

  $\exists$: Existential Quantifier

- Quantifiers with Restricted Domain (论域)

  $\forall (x: P(x)),\ Q(x) \rightleftharpoons \forall x\ (\text{if } P(x) \text{ then } Q(x)) \rightleftharpoons \forall x\ (P(x) \to Q(x))$

  $\exists (x: P(x)),\ Q(x) \rightleftharpoons \exists x\ (P(x) \wedge Q(x))$

  ↑ Informal Expression

x	P(x)	Q(x)	if P(x) then Q(x)
a	T	T	T
b	T	F	F
c	F	?	T
d	F	?	T
		T/F	

Uncertain

$\forall x$ if $P(x)$ then $Q(x)$ ⟶ We use symbol "$\to$" to express it.

T
T
T
T

p	q	$p \to q$
T	T	T
T	F	F
F	T	T
F	F	T

给"$\to$"理解上：加上 for all

Thm. $\phi \vee \psi \to \chi \equiv (\phi \to \chi) \wedge (\psi \to \chi)$

$\phi \wedge \psi \to \chi \equiv \phi \to (\psi \to \chi)$ ②

Thm. $\phi \to \psi \equiv \neg\phi \vee \psi$

$\neg\phi \equiv \phi \to F$

Thm. $\phi \to (\psi \to \phi)$ is a tautology.

$(\phi \to \psi \to \chi) \to (\phi \to \psi) \to (\phi \to \chi)$ is a tautology.

ϕ 和 ψ 能 imply χ (由②)

☆ imply 默认右结合

即 $\phi \to \psi \to \chi$ 表示 $\phi \to (\psi \to \chi)$

Thm. $\Phi, \psi \vDash \chi$ if and only if $\Phi \vDash \psi \to \chi$

$\phi \to \psi, \phi \vDash \psi$.

$\forall (x: \{\})\ x<x$ True ($F \to F$)

$\exists (x: \{\})\ x \neq x$ False ($F \wedge T$)

($A \neq \{\}$)

$\forall (x: A)\ (0=1)$ False

$\exists (x: A)\ (0=0)$ True

- Function

  $f(t)$

- Serious Definition of first order logic (FOL) / Syntax of FOL

  Def. Term (项) t: variables (x, y, z), constants $(c_1, c_2, c_3, \cdots)$ and functions ($f(x), g(y,z), h(h(w))\cdots$)

  Proposition φ: $P(t)$, $Q(t_1, t_2)$, $\cdots$

  $\phi_1 \wedge \psi$, $\neg\phi_2$, $\cdots$

  $\forall x \varphi_1$ $\exists x \varphi_2$

  Def. A syntax (句法) of FOL (Language): S is a set containing constants, function symbols and predicates.

  e.g. $\{0, 1, +, \overset{2}{-}, \overset{1}{-}, \times, =, <\}$ ⟶ 可用于说明整数 in 一阶逻辑符号集

  ($-^2$: minus, $-^1$: negative)

  $\forall x \forall y.\ (x+y = y+x)$

  * variables can stand for anything, not included.

  Def. S-structure has 4 parts =: $\mathcal{A}$:

  surface (表层结构)

  (1) a domain (A)

  (2) $\mathcal{A}(c) \in A$ — contains objects (对象) $\mathcal{A}(c)$: $S_{(c)} \mapsto A$

  (3) $\mathcal{A}(f) \in A \to A$ when f only contains one variables (一元函数)

  $\mathcal{A}(f)$: $S_{(f)} \mapsto (A \to A)$

  $\mathcal{A}(g) \in A \times A \to A$

  $\mathcal{A}(g)$: $S_{(g)} \mapsto (A \times A \to A)$ 二元

  $\mathcal{A}(p)$: $S_{(p)} \mapsto (A \to \{T, F\})$ (4) $\mathcal{A}(p) \in A \to \{T, F\}$ 一元谓词 (谓词)

  $\mathcal{A}(Q)$: $S_{(Q)} \mapsto (A \times A \to \{T, F\})$ $\mathcal{A}(Q) \in A \times A \to \{T, F\}$ 二元 谓词

  e.g. $A = \mathbb{Z}$. $\mathcal{A}(\text{"0"}) = 0$ (符号 数学对象)

  $\mathcal{A}(\text{"+"}, 1, 2) = 3$ } 函数

  (function, objects)

  $\mathcal{A}(\text{"="}, 2, 2) = T$

  $\mathcal{A}(\text{"="}, 2, 3) = F$ } 谓词.

  ..... ↘ Objects picked from the domain

  Def. S-interpretation $(\mathcal{A}, \beta) =: J$

  $\begin{cases} \mathcal{A} \text{ is a } S\text{-structure} \\ \beta(x) \in A \end{cases}$

  $\beta: \{x, y, z, \cdots\} \mapsto A$

  ($\beta(x)$: a variable.)

  ↘ Sometimes we use J to represent S-interpretation. In this case, $J(c) = \mathcal{A}(c)$, $J(f) = \mathcal{A}(f)$, $J(p) = \mathcal{A}(p)$, $J(x) = \beta(x)$.

  Def. Denotation of terms ($[\![t]\!]_J$)

  $[\![x]\!]_J = J(x)$, $[\![c]\!]_J = J(c)$, $[\![f(t_1, t_2, \cdots t_n)]\!]_J = J(f)\left([\![t_1]\!]_J, [\![t_2]\!]_J, \cdots [\![t_n]\!]_J\right)$

  (x 在 J 上的解释)

  Truth of propositions:

  $[\![P(t_1, t_2, \cdots t_n)]\!]_J = J(p)\left([\![t_1]\!]_J, [\![t_2]\!]_J, \cdots [\![t_n]\!]_J\right)$

$[\![\varphi \wedge \psi]\!]_J = [\![\wedge]\!]_0([\![\varphi]\!]_J, [\![\psi]\!]_J)$. $[\![\varphi \vee \psi]\!]_J = [\![\vee]\!]([\![\varphi]\!]_J, [\![\psi]\!]_J)$

$[\![\neg \varphi]\!]_J = [\![\neg]\!]([\![\varphi]\!]_J)$

$[\![\exists x \varphi]\!]_J = T$ iff exists $a \in A$, $[\![\varphi]\!]_{J[x \mapsto a]} = T$

将x解释为a

(把a赋值给x)

$[\![\forall x \varphi]\!]_J = T$ iff for any $a \in A$, $[\![\varphi]\!]_{J[x \mapsto a]} = T$

NOTICE

(1) 逻辑符号 [redacted] $\{\wedge, \vee, \neg, \otimes, \downarrow, \rightarrow, \leftrightarrow$ / $\forall, \exists$

非逻辑符号 $\{P, Q, \cdots$ / $c_1, c_2, c_3, \cdots$ / $f, g, \cdots$ / x, y, z

(2) 项 Term ~ First Order Language

Object ~ Interpretation 对象

FOL Propositional Logic

Def. SATISFIABILITY: $J \models \varphi$ ~ $J \models \varphi$

CONSEQUENCE: $\Phi \models \psi$ ~ $\Phi \models \psi$

LOGICAL [redacted] EQUIVALENCE: $\varphi \equiv \psi$ ~ $\varphi \equiv \psi$

φ is valid (永真) ~ tautology

φ SAT ~ φ SAT

Thm. $\{P(t) \models \exists x (P(x))$ / $\forall x (P(x)) \models P(t)$ $\rightarrow$ $\{$(1) $\psi_{[x \mapsto t]} \models \exists x \psi$ / (2) $\forall x \varphi \models \varphi_{[x \mapsto t]}$

Thm. If $\Phi \models \psi(x)$, x does not freely appear in Φ, then $\Phi \models \forall x \psi(x)$.

e.g. $\forall y (P(y) \wedge Q(y)) \models P(x)$, then $\forall y (P(y) \wedge Q(y)) \models \forall x (P(x))$

Thm. If $\Phi, \psi(x) \models \chi$, x does not freely appear in Φ or χ, In fact, $\forall x (P(x) \wedge Q(x)) \models P(x)$ 也可导出后者.

then $\Phi, \exists x \psi \models \chi$

{ Binding Appear : 存在对x的约定 ($\exists, \forall, \cdots$) $\exists x \varphi(x)$

Freely Appear : 不存在对x的约定 [redacted] $\psi(x)$, $\forall y (\psi(x))$

NOTICE

$x \mapsto t$: 一旦冲突, 把原来的变量重命名

e.g. $\varphi = \exists y (x = y + y + 1)$

$\varphi_{[x \mapsto y]} = \exists u (y = u + u + 1)$

Otherwise, $\exists y (y = y + y + 1)$ ✗

Thm. $\neg \forall x \phi \equiv \exists x \neg \phi$

$\neg \exists x \phi \equiv \forall x \neg \phi$

$\forall x (\phi \wedge \psi) \equiv (\forall x \phi) \wedge (\forall x \psi) \equiv \forall x \phi \wedge \forall x \psi$

$\forall x \phi \vee \forall x \psi \models \forall x (\phi \vee \psi)$

$\exists x (\phi \vee \psi) \equiv \exists x \phi \vee \exists x \psi$

$\exists x (\phi \wedge \psi) \models \exists x \phi \wedge \exists x \psi$

$\forall x \forall y \phi \equiv \forall y \forall x \phi$

$\exists x \exists y \psi \equiv \exists y \exists x \psi$

$\exists x \forall y \chi \models \forall y \exists x \chi$

NOTES

变元 (→ x)

$[\![R(x,y)]\!]_{\mathcal{J}[x \mapsto a]} = T$ (√)

$[\![R(a,a+1)]\!]_{\mathcal{J}} = T$ (×)

论域对象不能出现!

$\mathcal{J}_{[x \mapsto a,\ y \mapsto x+1]}$ (×)

要用论域对象赋值!

$[\![\ ]\!]_{\mathcal{J}[x \mapsto a,\ y \mapsto b]} = T$ (√) $\Leftrightarrow$ $\mathcal{J}(R, a, b) = T$ (√)

$[\![\forall x \phi]\!]_{\mathcal{J}} = T$ iff for any $a \in A$. $[\![\phi]\!]_{\mathcal{J}[x \mapsto a]} = T$.

变元符号 (→ x)　　对象 (→ a)

PROOF SYSTEM 推理系统

Def. $\Phi \vdash \psi$ (Φ derives ψ) iff it can be established by the following proof rules in finite steps: "推导出"

- $\phi_{[x \mapsto t]} \vdash \exists x \phi$ ← A rule (proof rule)
- $\forall x \phi \vdash \phi_{[x \mapsto t]}$
- If $\Phi \vdash \psi(x)$ and x does not freely occur in Φ, then $\Phi \vdash \forall x \psi$
- If $\Phi, \psi(x) \vdash \chi$ and x does not freely occur in Φ or χ, then $\Phi, \exists x \psi \vdash \chi$
- $\phi, \psi \vdash \phi \wedge \psi$
- $\phi \wedge \psi \vdash \phi$, $\phi \wedge \psi \vdash \psi$
- $\phi \vdash \phi \vee \psi$, $\psi \vdash \phi \vee \psi$
- If $\Phi, \phi \vdash \chi$, $\Phi, \psi \vdash \chi$, then $\Phi, \phi \vee \psi \vdash \chi$
- If $\Phi, \psi \vdash \chi$ and $\Phi, \neg\psi \vdash \chi$, then $\Phi \vdash \chi$
- If $\Phi, \neg\psi \vdash \chi$ and $\Phi, \neg\psi \vdash \neg\chi$, then $\Phi \vdash \psi$ (反证)
- If $\phi \in \Phi$, then $\Phi \vdash \phi$
- If $\Phi \subseteq \Psi$ and $\Phi \vdash \phi$, then $\Psi \vdash \phi$

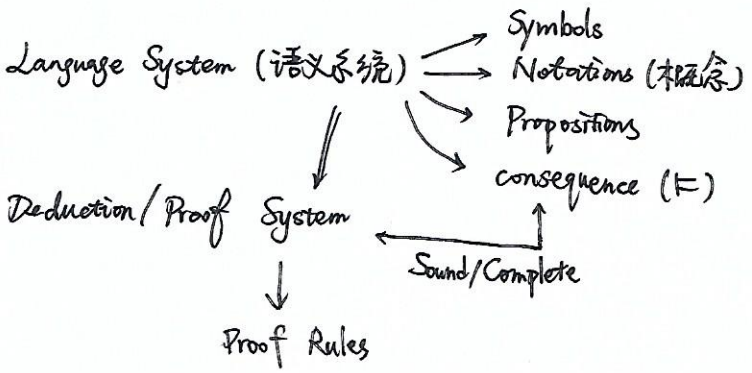
- Soundness & Completeness

(所有推理都满足 $\Phi \vdash \psi$)

Def. A first order logic ("$\vdash$") is sound if $\Phi \vdash \psi$ implies $\Phi \models \psi$.
可靠性

Def. is complete if $\Phi \models \psi$ implies $\Phi \vdash \psi$.
完全性 (所有 $\Phi \vdash \psi$ 都有对应的推导关系)

Proof of Soundness. All these rules remained true when you replace "$\vdash$" with "$\models$".



SET THEORY 集合论

- Axiomatic Set Theory's Relationship with Logics

  A certain set of symbols (FOL) ⟶ its system of proof rules ⟶ AST

- Naive Set Theory

  Def. Membership $a \in A$ (a is a member of the set A)

  Def. Subset $A \subseteq B$ iff $\forall x\ (x \in A \longrightarrow x \in B)$

  Proper Subset "$A \subset B$" "$A \subsetneq B$" We use "$A \subseteq B$ and $A \neq B$" to express it here.

  Def. Unions $A \cup B$

  $$\forall x \left(x \in A \cup B \leftrightarrow x \in A \vee x \in B \right)$$

  Intersection $A \cap B$

  $$\forall x\ (x \in A \cap B \leftrightarrow x \in A \wedge x \in B)$$

  Def. ~~Set-theoretic~~

  Set-theoretic Difference $A \setminus B$

  $$\forall x\ (x \in A \setminus B \leftrightarrow x \in A \wedge \neg (x \in B))$$

  $\not\leftrightarrow \neg x \in B$ is also acceptable.

  Def. Empty set $\emptyset$ $\quad \forall x\ \neg x \in \emptyset$

  Def. Enumerating Finite Objects

  We use "$\{a_1\}$", "$\{a_1, a_2, \cdots a_3\}$" to represent a enumerating finite set.

  $$\forall x\ (x \in \{a_1, a_2, \cdots, a_n\} \longleftrightarrow x = a_1 \vee x = a_2 \vee \cdots \vee x = a_n)$$

  So $\{1,2,2\}$ is acceptable, but $\{1,2,2\} = \{1,2\}$ and has only two members.

  Describing a Property

  $$\forall x\ (x \in \{a \mid P(a)\} \longleftrightarrow P(x))$$

  Def. Ordered Pairs (a,b) $\quad (1,2) \neq (2,1)$

  Def. Cartesian Product (笛卡尔积, 直积)

  $$A \times B = \{(a,b) \mid a \in A, b \in B\} = \{x \mid \exists a \exists b\ (x = (a,b) \wedge a \in A \wedge b \in B)\}$$

  左结合性. $A \times B \times C = (A \times B) \times C$

  Def. Power Set (幂集)

  $\mathcal{P}(A)$ $\quad \forall x\ (x \in \mathcal{P}(A) \leftrightarrow x \subseteq A)$

  $\mathcal{P}(\emptyset) = \{\emptyset\}$. $\quad \mathcal{P}(\{a,b\}) = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$

  Def. $A = B \leftrightarrow \forall x\ (x \in A \leftrightarrow x \in B)$ $\quad \emptyset = \{x \mid x \geq 0 \wedge x < -8\}$

- Axiomatic Set Theory —— Why it Appears

Problems of Naive Set Theory — Russel's Paradox

- Let $A=\{a \mid a \notin a\}$. Is A a member of A?
- ~~If $A \in A$, then $A \notin A$; if $A \notin A$, then $A \in A$.~~
- For any $x \in A \leftrightarrow x \notin x$. Let $x=A$, we get $A \in A \leftrightarrow A \notin A$. Contradiction!

Solution The expression like $\{a \mid P(a)\}$ is prohibited and replaced by $\{a \in A \mid P(a)\}$.

Q: Does there exist a U such that $\forall x,\ x \in U$?

Assume $\exists U.\ \forall x.\ x \in U$. We can construct $\{a \in U \mid a \notin a\} =: A$

Then we get Russel's Paradox. ($A \in A \leftrightarrow A \notin A$)

So $\nexists U$ such that $\forall x.\ x \in U$.

- ZF Set Theory

"$\Phi \vdash \psi$" ~ Imagination / Ideal Axiomatic ST

Axioms Theromes

Problems: <1> Generic properties about equality are missing: $x=y \wedge x \in z \rightarrow y \in z$

<2> $(x=y) \leftrightarrow (\forall u\ u \in x \leftrightarrow u \in y)$ —— Can we quantify over sets only?

Can we add "$\forall x \forall y$"? What about natural numbers?

<3> $\{x_1, x_2, \dots, x_n\}$

<4> Describing a property: $\{a \in A \mid P(a)\}$ cannot be a function symbol.

Solutions: <1> Generic properties about equality:

- $t=t$
- $t_1=t_2 \wedge \phi_{[x \mapsto t_1]} \rightarrow \phi_{[x \mapsto t_2]}$

<3> Enumeration Notation

- New function symbol $\{x\}$
- $\forall x \forall y\ (x \in \{y\} \leftrightarrow x=y)$
- Write $\{x_1, \dots x_n\}$ as $\{x_1\} \cup \dots \cup \{x_n\}$

NOTES:

$\{x\}$ [redacted] $=: f(x)$

[redacted]

$x \in u =: R(x,u)$

$\forall x \forall y\ (R(x,f(x)) \leftrightarrow x=y)$

Axiom Schema Of Separation (分离公理)

$\forall u \forall x \exists y \forall z\ (z \in y \leftrightarrow z \in x \wedge \phi(x,u))$

ϕ is a proposition in which x does not freely occur. and y

} More than 1 axiom. (for any ϕ, there exist such an axiom)

It doesn't give the definition of $\{a \in A \mid P(a)\}$, but it means such sets exist, which is already enough for us to go on. ($\{z \in x \mid \phi(z,u)\} =: y$)

Now we only have <2> unsolved. Mathematicians decide to express all things by sets.

Axiom Schema of Extensionality (外延公理)

$\forall x \forall y \, (x=y \leftrightarrow \forall z \, (z \in x \leftrightarrow z \in y))$

Everything is a set.

Representing Natural Numbers as Sets. (The process called "Encoding")

$0 := \{\}$

$1 := \{0\}$

$2 := \{0,1\}$

……

$n+1 := n \cup \{n\} = \{0,1,2,\cdots,n\}$

(不严谨符号)

e.g. $n < m := n \in m$ (可视为集合)

$n \leq m := n \subseteq m$

Is $\mathbb{N}$ a set under this encoding?

Axiom of Infinity (无穷公理)

$\exists x \; \text{Inductive}(x)$

Def. $\text{Inductive}(t) := \left(\emptyset \in x \wedge \forall y \, (y \in x \longrightarrow y \cup \{y\} \in x) \right)_{[x \mapsto t]}$

归纳集

Is $\mathbb{N}$ a set?

Def. $\mathbb{N}$ can be encoded as $\text{Inductive}(t) \wedge \forall x \, (\text{Inductive}(x) \longrightarrow t \subseteq x)$

(Let t be $\mathbb{N}$)

认为$\mathbb{N}$是最小归纳集.

e.g. Existence of $\mathbb{N}$: $\exists x \, (\text{Inductive}(x) \wedge \forall y \, (\text{Inductive}(y) \longrightarrow x \subseteq y))$

Uniqueness of $\mathbb{N}$: $\forall x \forall y \Big((\text{Inductive}(x) \wedge \forall z \, (\text{Inductive}(z) \longrightarrow x \subseteq z)) \wedge (\text{Inductive}(y) \wedge \forall z \, (\text{Inductive}(z) \longrightarrow y \subseteq z)) \longrightarrow x = y \Big)$

Representing Ordered Pairs as Sets. (Encoding Ordered Pairs)

Def. (t_1, t_2) can be encoded as $\{\{t_1\}, \{t_1, t_2\}\}$

$(x,y) = (x',y') \longrightarrow x = x' \wedge y = y'$

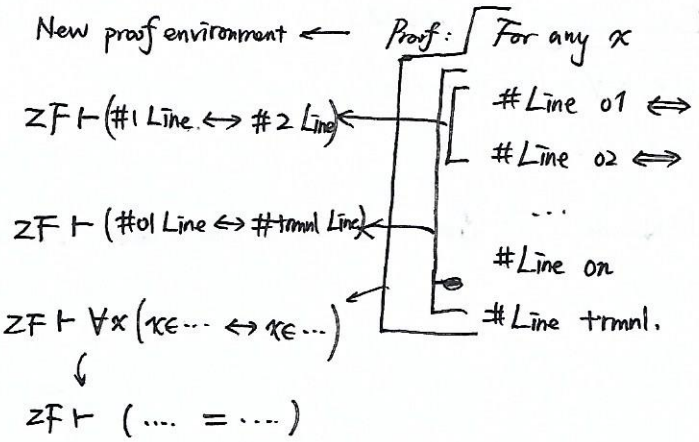
① $x \neq y$ $\quad x=x', y=y'$

② $x = y$ $\quad \{\{t\}\} \longrightarrow x = x' = y = y'$.

Thm. If ϕ is a proposition and $ZF \vdash \phi$, we call ϕ a mathematical theorem of ZF.

Here ZF represents the set of axioms in ZF.

- Mathematical Proof



RELATIONS 关系

- Binary Relations

We use ordered pairs to describe relationships, using the set of ordered pairs to define relations.

e.g. $(3.7, \pi) \longrightarrow 3.7 > \pi$

Use $\{(x,y) \mid (x,y) \in \mathbb{R}\times\mathbb{R},\ x>y\}$ to define ">" relation.

e.g. Congruence (同余)

x and y are congruent modulo 7 $\longrightarrow$ $\{(x,y) \in \mathbb{R}\times\mathbb{R} \mid 7 \mid (x-y)\}$

* Congruence 是表示在一定运算下仍保持某种 relation 性质

Def. $R \subseteq \mathbb{A}\times\mathbb{B}$ is called a *relation* from A to B.

Specially, $R \subseteq \mathbb{A}\times\mathbb{A}$ is called a relation on A.

* $\emptyset$ is also a relation on $\mathbb{A}$

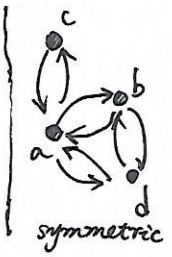
Note. aRb means $(a,b) \in R$.

- Properties of Binary Relations.

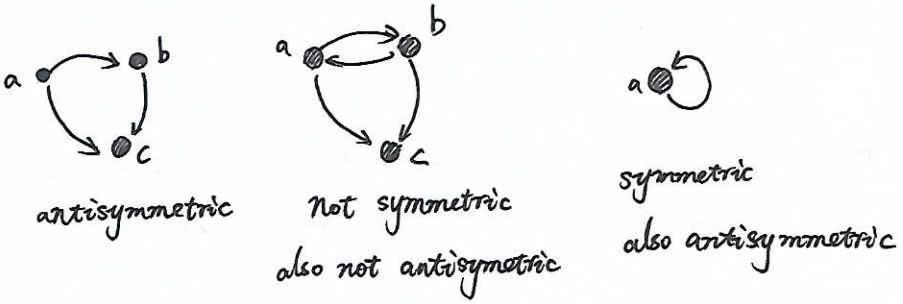
Def. Given $R \in \mathbb{A}\times\mathbb{A}$

(1) R is *reflexive* on $\mathbb{A}$ iff $\forall a.\, a\in\mathbb{A}.\ aRa$.
自反

(2) R is *symmetric* on $\mathbb{A}$ iff $\forall a,b,\ a,b\in\mathbb{A},\ aRb \longrightarrow bRa$.
对称



(3) R is *antisymmetric* on $\mathbb{A}$ iff $\forall a,b.\ a,b\in\mathbb{A}\ (aRb \wedge bRa \longrightarrow a=b)$.



(4) $\mathcal{R}$ is transitive 传递性 iff $\forall a \forall b \forall c \, (a\mathcal{R}b \wedge b\mathcal{R}c \longrightarrow a\mathcal{R}c)$

e.g. "$<$" is antisymmetric and transitive.

e.g. $\{(x,y) \in \mathbb{R}\times\mathbb{R} \mid x=y\}$: Identity Relation (等同关系)

reflexive, symmetric, antisymmetric and transitive

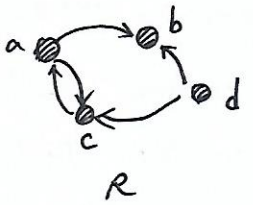
- Operators

e.g. $R_1 \cap R_2$. $R_1 \cup R_2$. $\bar{R} := \mathbb{A}\times\mathbb{B} \setminus R$.

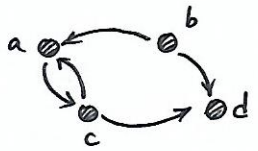
↑ 需要在上下文中明确关系对应的范围.

Def. Inverse of $\mathcal{R}$ ($\mathcal{R}^{-1}$) 逆

$\mathcal{R}^{-1} := \{(a,b) \mid b\mathcal{R}a\}$



$\mathcal{R}$

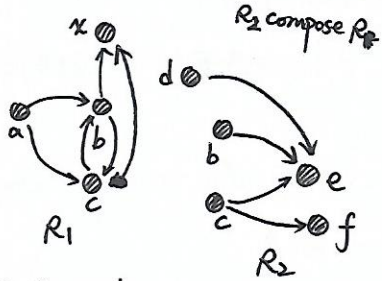


$\mathcal{R}^{-1}$

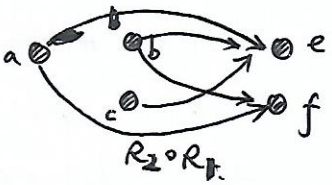
Def. Composition : If $R_1 \subseteq \mathbb{A}\times\mathbb{B}$, $R_2 \subseteq \mathbb{B}\times\mathbb{C}$.

then $R_2 \circ R_1 = \{(a,c) \mid \exists b \, (a R_1 b \wedge b R_2 c)\}$

R_2 compose R_1 —— The sequence is just like the composition of multiple functions.



R_1 R_2



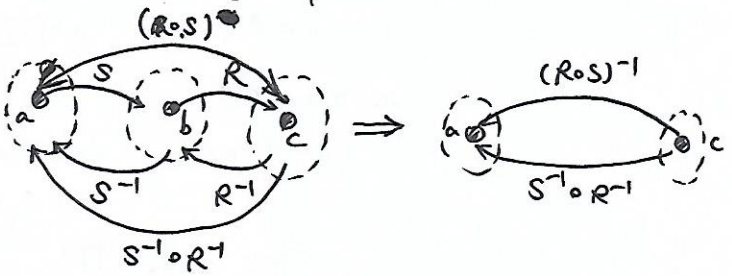
$R_2 \circ R_1$

* In fact, $R \circ S$ and $S \circ R$ is also acceptable, two different symbol systems.

First we go through a path in R_1 and then we go through a path in R_2.

Thm. $(\mathcal{R}^{-1})^{-1} = \mathcal{R}$.

$(R \circ S)^{-1} = S^{-1} \circ R^{-1}$



- Equivalence Relations

Def. $\mathcal{R} \subseteq \mathbb{A}\times\mathbb{A}$ is an equivalent relation on A iff it is reflexive, symmetric and transitive.

e.g. Congruence: $\{(a,b) \mid 7 \mid a-b\}$ $\in \mathbb{R}\times\mathbb{R}$

$\{(n,m) \mid (n<0 \wedge m<0) \vee (n=0 \wedge m=0) \vee (n>0 \wedge m>0)\}$

To understand it emotionally: the two parts of the relation can be replaced with those with similar properties. "换着用"

Q: Is a relation, symmetric and transitive, also a reflexive relation?

A: Nope. e.g. $\mathbb{A}=\{1,2,3\}$. $\mathcal{R}=\{(1,2),(3,1),(1,1),(3,3)\}$

not reflexive

e.g. $\mathbb{A}=\mathbb{N}$. $\mathcal{R}=\emptyset$.

- Equivalence Class (等价类)

Def. $\mathcal{R}$ is an equivalence relation on $\mathbb{A}$, $a\in\mathbb{A}$.

$[a]_{\mathcal{R}} :== \{x\in\mathbb{A} \mid a\mathcal{R}x\}$ is an equivalence class.

↓

representative (代表元)

Def. $\mathbb{A}$ partition $P\subseteq\mathcal{P}(\mathbb{A})$ of $\mathbb{A}$ satisfies:

(1) $\forall B\ (B\in P \longrightarrow B\neq\emptyset)$, i.e. $\emptyset\notin P$

(2) $\forall B\,\forall B'\ (B\in P\wedge B'\in P\wedge B\neq B' \longrightarrow B\cap B'=\emptyset)$

(3) $\bigcup P=\mathbb{A}$.

P is named a partition (分割/划分) of $\mathbb{A}$.

"A中每个元素最多、最少被划分一次"

Lemma. If P is a partition of $\mathbb{A}$, $a\in\mathbb{A}$. then there exists exact one $B\in P$ s.t. $a\in B$.

Thm. If $\mathcal{R}$ is an equivalence relation on $\mathbb{A}$.

then $\{[a]_{\mathcal{R}} \mid a\in\mathbb{A}\}$ is a partition of $\mathbb{A}$.

[Proof]. Since $[a]_{\mathcal{R}}\subseteq\mathbb{A}$. it's obvious that $\{[a]_{\mathcal{R}} \mid a\in\mathbb{A}\}\subseteq\mathcal{P}(\mathbb{A})$

1. $\forall a\in\mathbb{A}$. $a\in[a]_{\mathcal{R}}$ (since $a\mathcal{R}a$). Thus, $[a]_{\mathcal{R}}\neq\emptyset$ $(\forall a\in\mathbb{A})$

2. $\forall a,b\in\mathbb{A}$.

① if $a\mathcal{R}b$. $a\mathcal{R}c\wedge a\mathcal{R}b\Rightarrow a\mathcal{R}c\wedge b\mathcal{R}a\Rightarrow b\mathcal{R}c$

$a\mathcal{R}b\wedge b\mathcal{R}c\Rightarrow a\mathcal{R}c$

Thus, $[a]_{\mathcal{R}}=[b]_{\mathcal{R}}$

② if $\neg a\mathcal{R}b$. Then $[a]_{\mathcal{R}}\cap[b]_{\mathcal{R}}=\emptyset$.

(Prove it by contradiction.

If not, exists $c\in\mathbb{A}$. $a\mathcal{R}c\wedge b\mathcal{R}c\Longrightarrow a\mathcal{R}c\wedge c\mathcal{R}b\Longrightarrow a\mathcal{R}b$.

Contradiction.)

3. $\bigcup\{[a]_{\mathcal{R}} \mid a\in\mathbb{A}\}=\mathbb{A}$.

$\subseteq$: Plain to see. Left $\subseteq\mathcal{P}(\mathbb{A})$.

$\supseteq$: i.e. To prove $\forall a\in\mathbb{A}.\ \exists B\in\{[a]_{\mathcal{R}} \mid a\in\mathbb{A}\},\ a\in B$.

(Obvious. Pick $B=[a]_{\mathcal{R}}$)

Thm. If P is a partition of $\mathbb{A}$. then $R=\{(a,b)\in\mathbb{A}\times\mathbb{A} \mid \exists B(B\in P \wedge a\in B \wedge b\in B)\}$ is an equivalence relation, and $P=\{[a]_R \mid a\in\mathbb{A}\}$.

[Proof]. 1. R is reflexive, i.e. $\forall a\in\mathbb{A}.\ \exists B\in P$ s.t. $a\in B$. From Lemma it's clear to see.

2. R is symmetric.

$$aRb \Leftrightarrow \exists B\ (B\in P\wedge a\in B\wedge b\in B) \Leftrightarrow \exists B\ (B\in P\wedge b\in B\wedge a\in B) \Leftrightarrow bRa$$

3. R is transitive.

Suppose aRb, bRc.

Thus, $\exists B_1, B_2 (B_1\in P \wedge B_2\in P \wedge a\in B_1 \wedge b\in B_1 \wedge b\in B_2 \wedge c\in B_2)$

~~$\Leftrightarrow \exists B_1, B_2 (B_1\in P \wedge B_2\in P \wedge B_1=B_2 \wedge a\in B_1 \wedge b\in B_1 \wedge c\in B_2)$~~

Thus, $B_1=B_2$, so aRc.

4. $\forall a\in\mathbb{A}.\ [a]_R\in P$. By definition, $[a]_R=\{b \mid aRb\}$

$$=\{b \mid \exists B\ (B\in P\wedge a\in B\wedge b\in B)\}$$

Since P is a partition of $\mathbb{A}$, exists one $B\in P$ has $a\in B$.

Assume it is B_0.

Thus, $[a]_R=\{b \mid b\in B_0\}=B_0\in P$.

If exists a $B\in P$. $B\neq[a]_R$ $(\forall a\in\mathbb{A})$. $\because B\neq\emptyset$.

Thus, $\exists b\in B$. Then $B=[a]_R$. Contradiction!

- Transitive Closures and Reflexive Transitive Closures
  傳遞閉包 自反傳遞閉包

Def. $R\subseteq\mathbb{A}\times\mathbb{A}$. Find a R' s.t.

(1) $R\subseteq R'$

(2) R' is transitive.

(3) $\forall T\subseteq\mathbb{A}\times\mathbb{A}$. if $R\subseteq T$ and T is transitive, $R'\subseteq T$. ("the smallest")

} [First Definition of Transitive Closure]

Then we call R' R's transitive closure.

Def. Suppose $R\subseteq\mathbb{A}\times\mathbb{A}$.

Let $R^1=R$. $R^2=R\circ R$, $\cdots$ $R^{n+1}=R^n\circ R$. $R^+=\bigcup_{n=1}^{\infty} R^n$ is R's transitive closure.

Uniqueness is obvious.

We need to prove the existence of R', i.e. to prove $R'=R^+$.

Lemma. $R^n\circ R^m=R^{n+m}$. $(n,m=1,2,3,\cdots)$

[Proof] If $m=1$. $R^n\circ R^m=R^n\circ R=R^{n+1}=R^{n+m}$

If $m=k+1$. Assume $R^n\circ R^k=R^{n+k}$ holds.

$$R^n\circ R^m=R^n\circ R^{k+1}=R^n\circ(R^k\circ R)=(R^n\circ R^k)\circ R$$
$$=R^{n+k}\circ R=R^{n+k+1}=R^{n+(k+1)}=R^{n+m}$$

What is the proof like in Atomatic Set Theory?

R^n: $(a,b)\in R^n \iff \text{Powerof}R = \{(n,a,b)\in \mathbb{N}\times\mathbb{A}\times\mathbb{A} \mid \forall a\forall b\,(n=1 \to aRb) \wedge$

$\forall a\forall b\forall m\,(n=m\cup\{m\} \to \exists c\,((m,c,b)\in \text{Powerof}R \wedge aRc))\}$

$n+m$: $+ \subseteq \mathbb{N}\times\mathbb{N}\times\mathbb{N}$

$(n,m,r)\in + \longrightarrow (n, m\cup\{m\}, r\cup\{r\})\in +$.

$n+0=n$

$n+(m\cup\{m\}) = (n+m)\cup\{n+m\}$.

$$\bigcup_{n=1}^{\infty} R^n = \left\{(a,b)\in\mathbb{A}\times\mathbb{A} \;\middle|\; \exists n\in\mathbb{N},\ (n,a,b)\in \text{Powerof}R\right\}$$

Lemma. $R_1\subseteq R_1'$, $R_2\subseteq R_2'$ implies $R_1\circ R_2 \subseteq R_1'\circ R_2'$

[Proof] If $(a,c)\in R_1\circ R_2$, then $\exists b$ $(a,b)\in R_2$, $(b,c)\in R_1$. Thus, $(a,b)\in R_2'$, $(b,c)\in R_1'$

$(a,c)\in R_1'\circ R_2'$.

Lemma. If $R_n\subseteq T$ (for any n), then $\bigcup_{n=1}^{\infty} R_n \subseteq T$.

[Proof] If $(a,b)\in \bigcup_{n=1}^{\infty} R_n$, then $\exists n$, aR_nb. Thus, aTb.

Now we prove the existence of transitive closure, i.e. to prove $R^+=R^{tc}$.

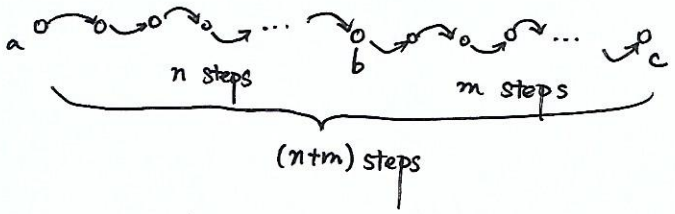
[Proof] Lemma 1. $R\subseteq R^+$

Proof: $R=R^1\subseteq \bigcup_{n=1}^{\infty} R^n = R^+$. Qed.

Lemma 2. R^+ is transitive.

Proof. If aR^+b, bR^+c, there exists n,m s.t. aR^nb, bR^mc.

Thus, $aR^{m+n}c$. So aR^+c. Qed.



Lemma 3. If $T, R\subseteq\mathbb{A}\times\mathbb{A}$. $R\subseteq T$. T is transitive. then $R^+\subseteq T$.

Proof. 1. $R^1 = R\subseteq T$

2. IH. (Inductive Hypothesis) If $R^n\subseteq T$

$R^{n+1} = R^n\circ R \subseteq T\circ T\subseteq T$ (since T is transitive)

Thus, for any n. $R^n\subseteq T$. Thus, $R^+\subseteq T$. Qed.

Def. R' is R's reflexive transitive closure iff

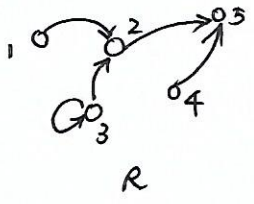
[First Def.]
(1) $R\subseteq R'$
(2) R' is reflexive and transitive.
(3) $\forall T\subseteq\mathbb{A}\times\mathbb{A}$. if $R\subseteq T$ and T is reflexive and transitive then $R'\subseteq T$.

[Second Def.]

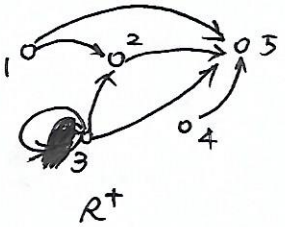
Suppose $R \subseteq \mathbb{A} \times \mathbb{A}$

Let $R^0 = \{(a,a) \mid a \in \mathbb{A}\}$. $R^1 = R$. $R^2 = R \circ R$. $R^{n+1} = R^n \circ R$

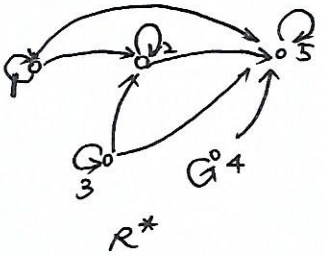
$R^* = \bigcup_{n=0}^{\infty} R^n$ is R's reflexive and transitive closure.



R



R^+



R^*

FUNCTIONS 函数

- Functions

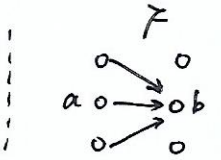
Def. F is a function from A into B if $F \subseteq A \times B$ and $\forall a \in A\ \exists b \in B\ (aFb)$,
$\forall a \forall b \forall b'\ (aFb \wedge aFb' \to b = b')$

Notation. $F: A \to B$ represents F is a function from A into B.

If $x \in A$. $F(x)$ represents the only y s.t. xFy.

If aFb. b is called the 像 image of a.

a is called a preimage of b. 一个 原像



If $F: A \to B$. A is called domain 定义域, B is called codomain 陪域.

$\{F(a) \mid a \in A\}$ is called range / image 值域.

Thm. If $f, g: A \to B$, $f = g$ iff. $\forall x \in A (f(x) = g(x))$.

[Proof]. "$\Rightarrow$": obvious

"$\Leftarrow$": Suppose $\forall x \in A\ (f(x) = g(x))$.

To prove $\forall a \in A\ \forall b \in B.\ (a,b) \in f \leftrightarrow (a,b) \in g$.

afb iff. $b = f(a)$ iff. $b = g(a)$ iff. agb. Qed.

Thm. If $f: B \to C$, $g: A \to B$,

then $f(g(a)) = (f \circ g)(a)$ $(\forall a \in A)$, $f \circ g$ is a function from A into C.

[Proof] 1. $\forall a \in A.\ (\exists c \in C\ (a(f \circ g)c))$

For any $a \in A$,

By $g: A \to B$. $\exists b \in B.\ agb$.

By $f: B \to C$. $\exists c \in C.\ bfc$

Thus, $a(f \circ g)c$

2. $\forall a\ \forall c\ \forall c'.\ (a(f \circ g)c \wedge a(f \circ g)c' \to c = c')$

Given such a, c, c'.

We know exists b, b'. s.t. $agb \wedge agb' \wedge bfc \wedge b'fc'$.

By $g: A \to B$, $b = b'$

By $f: B \to C$, $c = c'$.

3. $\forall a$. $g(a)$ is the b s.t. agb

$f(g(a))$ is the c s.t. bfc.

$(f \circ g)(a)$ is the c'. s.t. $a(f \circ g)c'$

Thus, $a(f \circ g)c$.

Thus, $c = c'$, i.e. $(f \circ g)(a) = f(g(a))$.

Qed.

Thm. If $f: A \to B$, then $\{(x,y) \in A \times A \mid F(x) = F(y)\}$ is an equivalence relation.

- Special Classes of Functions

Def. If $f: A \to B$, we say

1) f is one-to-one / an injection (单射) if $\forall a, a' \in A \ (f(a)=f(a') \to a=a')$

2) f is onto / a surjection (满射) if $\forall b \in B \ \exists a \in A. \ (f(a)=b)$.

3) f is a bijection (双射, 一一对应) if f is both an injection and a surjection.

Thm. If $f: B \to C$ and $g: A \to B$.

1) $f \cdot g$ is an injection, $f \circ g$ is an injection

2) surjection surjection

3) f and $f \circ g$ is an injection. g is an injection.

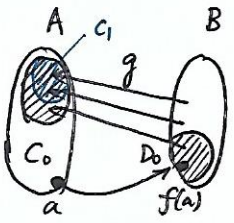
* Partial Functions are sometimes represented as $f: A \longrightarrow B$.

Thm. If $f: A_1 \longrightarrow A_2$, $g: B_1 \longrightarrow B_2$ are injections,

$\{((a_1,b_1),(f(a_1),g(b_1))) \in (A_1 \times B_1) \times (A_2 \times B_2) \mid a_1 \in A_1, b_1 \in B_1\}$ is an injection from $A_1 \times B_1$ into $A_2 \times B_2$.

Thm. (Berstain's Theorem)

If $f: A \to B$, $g: B \to A$ are injections,
then exists a bijection between A and B.



[Proof] Let $C_0 = \{a \in A \mid \forall b \in B, a \neq g(b)\}$.

$D_0 = \{f(a) \in B \mid a \in C_0\}$

$C_1 = \{g(b) \in A \mid b \in D_0\} = \{a \in A \setminus C_0 \mid \forall b \in B \setminus D_0, a \neq g(b)\}$.

$D_1 = \{f(a) \in B \mid a \in C_1\}$

...

$C_{n+1} = \{g(b) \in A \mid b \in D_n\} = \{a \in A \setminus \bigcup_{i=0}^{n} C_i \mid \forall b \in B \setminus \bigcup_{i=0}^{n} D_i, a \neq g(b)\}$.

$D_{n+1} = \{f(a) \in B \mid a \in C_{n+1}\}$

Then, we have an bijection from $\bigcup_{n=0}^{\infty} C_n$ into $\bigcup_{n=0}^{\infty} D_n$, i.e. f.

Now we prove that g is an bijection from $A \setminus \bigcup_{n=0}^{\infty} C_n$ into $B \setminus \bigcup_{n=0}^{\infty} D_n$.

1. Suppose $b \in B \setminus \bigcup_{n=0}^{\infty} D_n$. $g(b) \in A$. $g(b) \notin C_n$

If $g(b) \in C_n$. { if $n \neq 0$. Contradiction!
if $n = k+1$. $g(b) = g(b')$ $b' \in D_n$ $(\exists b')$

Since g is injective, $b = b'$. $b \in D_n$. Contradiction!

e.g. $A = B = \mathbb{N}$.

$f(x) = x$ $C_0 = \{0\}$ $C_1 = \{1\}$ $C_2 = \{2\}$ ...
$g(x) = x+1$ $D_0 = \{0\}$ $D_1 = \{1\}$ ...

In this case. $A = \bigcup_{n=0}^{\infty} C_n$. $B = \bigcup_{n=0}^{\infty} D_n$

Nevertheless.
$f(x) = x+1$ $C_0 = \{0\}$ $C_2 = \{2\}$ $C_3 = \{4\}$ ...
$g(x) = x+1$ $D_0 = \{1\}$ $D_2 = \{3\}$ ...

In this case $A \setminus \bigcup_{n=0}^{\infty} C_n \neq \emptyset$. $B \setminus \bigcup_{n=0}^{\infty} D_n \neq \emptyset$.

Thus, $b \in B \setminus \bigcup_{n=0}^{\infty} D_n$, $g(b) \in A \setminus \bigcup_{n=0}^{\infty} C_n$.

2. Suppose $a \in A \setminus \bigcup_{n=0}^{\infty} C_n$.

a) no $b \in B$. $a = g(b)$ Then $a \in C_0$. Contradiction!

b) exists $b \in B$. $a = g(b)$.

If $b \in D_n$, $a = g(b)$. Then $a \in C_{n+1}$. Contradiction!

Thus, $b \in B \setminus \bigcup_{n=0}^{\infty} D_n$.

Qed.

- Equinumerosity (等勢)

Def. $A \approx B$ if exists a $f: A \to B$ is a bijection.

Notice. Let $A = \{a = 2n \mid n \in \mathbb{N}\}$. $A \subseteq \mathbb{N}$ → "A is smaller than $\mathbb{N}$"

But $A \approx \mathbb{N}$.

Let $B = \{b = 3n \mid n \in \mathbb{N}\}$. A and B are not related. can't be compared.

But $A \approx B$.

$\lim_{n\to\infty} \frac{|\{x \in A \mid x \le n\}|}{|\{x \in B \mid x \le n\}|} \Rightarrow$ "A is larger than B".

$[0,1] \approx (0,1)$

[P1.] We can construct a bijection: $0 \to \frac{1}{3}$ $\frac{1}{3} \to \frac{1}{9}$ ... $\frac{1}{3^n} \to \frac{1}{3^{n+1}}$, ...

$1 \to \frac{2}{3}$ $\frac{2}{3} \to \frac{8}{9}$ ... $1 - \frac{1}{3^n} \to 1 - \frac{1}{3^{n+1}}$, ...

The rest remain the same $(f(x) = x)$.

[P2.] Use Bernstein's Theorem.

$g(x) = x$, $g: (0,1) \to [0,1]$

$f(x): [0,1] \to [\frac{1}{3}, \frac{2}{3}]$. i.e. $f: [0,1] \to (0,1)$

.....

NOTICE: You have to prove a relation is a function if it's not plain to see that.

e.g. $f(n,m) = 2^n(2m+1)$ is a function, which is obvious.

$F([a]_{R_1}, [b]_{R_2}) = [(a,b)]_R$. ← You need to prove it's a function.

Thm. $\mathcal{P}(A) \approx \underbrace{(A \to \{0,1\})}_{\text{a set of functions}} =: 2^A$

Proof. Let $F(X)(a) = \begin{cases} 0 & \text{if } a \notin X \\ 1 & \text{if } a \in X \end{cases}$ $(X \subseteq A)$

It's plain to see that it's a function.

If $F(X) = F(Y) \Rightarrow \forall a \ F(X)(a) = F(Y)(a) \Rightarrow \forall a, a \in X \leftrightarrow a \in Y \Rightarrow X = Y$

Thm. ~~$\approx$ is an equivalence relation,~~

1) $A \approx A$
2) if $A \approx B$ then $B \approx A$
3) if $A \approx B$, $B \approx C$ then $A \approx C$.

Qst. Is "$\approx$" an equivalence relation?

Nope!

Remember Russell's Paradox?

Thus, "$\approx$" is not a binary relation on a set.

(because the set consisting of all sets does not exist)

Thm. If $A_1 \approx A_2$, $B_1 \approx B_2$ then $A_1 \times B_1 \approx A_2 \times B_2$.

If $A_1 \approx A_2$, $B_1 \approx B_2$ then $(A_1 \to B_1) \approx (A_2 \to B_2)$.

[2. Proof] Suppose $f: A_1 \to A_2$ is a bijection. (because $A_1 \approx A_2$)

$g: B_1 \to B_2$ is a bijection. (because $B_1 \approx B_2$)

Lemma. f^{-1} is also a function. f^{-1} is also a bijection.

.....

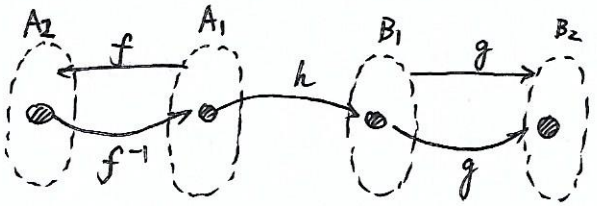
Qed.

We construct $H: (A_1 \to B_1) \to (A_2 \to B_2)$ s.t.

$$H(h)(x) = g(h(f^{-1}(x))), \text{ where } h: A_1 \to B_1$$

i.e. $H(h) = g \circ h \circ f^{-1}$, where $h: A_1 \to B_1$

It's plain to see that H is a function.



How we construct H.

1. We prove H is a ~~bijection~~ injection

If $H(h_1) = H(h_2) \Rightarrow \forall x \left(g \circ h_1 \circ f^{-1}(x) = g \circ h_2 \circ f^{-1}(x)\right)$

$\Rightarrow \forall x \left(h_1 \circ f^{-1}(x) = h_2 \circ f^{-1}(x)\right)$ (because g is a bijection)

$\Rightarrow \forall y \left(h_1(y) = h_2(y)\right)$ (because f^{-1} is a surjection)

$\Rightarrow h_1 = h_2$

2. We prove H is a ~~surjection~~ sur~~jection~~. (obvious)

Thm. $(A \times B \to C) \approx (A \to (B \to C))$

[Proof] $f: A \times B \to C$

$H(f)(a)(b) = f(a,b)$

$H(f)$: $B \to C$; $A \to B \to C$

$H(f) = g$ s.t. $g(a)(b) = f(a,b)$ (function name)

g: $B \to C$.

- Countable Sets

Def. A is countable iff. $A \approx \mathbb{N}$.

e.g. $\mathbb{N}$. $\mathbb{Z}$. $\{x \mid x \text{ is even}\}$. $\mathbb{N}\times\mathbb{N} =: \mathbb{N}^2$. $\mathbb{Q}$ ($\frac{p}{q}$, q 由非0再排 p) (p,q) 对角线排.

(0, 1, -1, 2, -2, ...)

Differ. $\mathcal{R}^2$ ($\mathcal{R}$ is a relation)

↑ composition of $\mathcal{R}$

$\mathbb{N}^3 = \mathbb{N}^2 \times \mathbb{N} \approx \mathbb{N}\times\mathbb{N} = \mathbb{N}^2 \approx \mathbb{N}$.

$\mathbb{N}^n$ $(n \in \mathbb{N})$.

The set of finite sequences of $\mathbb{N}$ is countable.

[Proof.] For a decent length: $\{1,3,5\}$ length=3.

we can find a bijection between sequences of length and $\mathbb{N}^{\text{length}}$ such.

i.e. $FS \approx \mathbb{N}^{\text{length}}$.

$\Rightarrow$ (length, $FS(x)$) — countable.

sequence

$FS: \{x \mid \text{length}(x) = \text{length}\} \xrightarrow{\text{countable}} \mathbb{N}^{\text{length}}$

$\mathbb{N}^{\text{length}}$ ↓ countable

$\mathbb{N}$.

Thm. if A is a set then $\mathcal{P}(A) \not\approx A$.

Corollary. $\mathcal{P}(\mathbb{N})$ is uncountable.

[Proof] Prove it by Contradiction / Proof by Contradiction.

Suppose $F: A \to \mathcal{P}(A)$ is a bijection.

Let $B = \{a \in A \mid a \notin F(a)\} \subseteq A$

There exists exact one $b \in A$ s.t. $F(b) = B$.

$b \in F(b) \Leftrightarrow b \in B \Leftrightarrow b \notin F(b)$. Contradiction!

Thus, there does not exist such F.

Thus, $\mathcal{P}(A) \not\approx A$.

Thm. $\mathbb{R} \approx 2^{\mathbb{N}} \approx \mathcal{P}(\mathbb{N})$

[Proof]. $f: 2^{\mathbb{N}} \to \mathbb{R}$

十进制下有 $\cdots\overline{9999999\cdots} = \cdots\overline{10000000\cdots}$ 的问题!

Bernstein's Theorem

bijection

0, 1, 2, 3, ⋯ , f(0) f(1) f(2) f(3) ⋯ → 0. $\overline{f(0)\, f(1)\, f(2)\, f(3)\cdots}$ injection.

0/1 0/1 不用二进制转十进制 原因?

$g: \mathbb{R} \to 2^{\mathbb{N}}$

0 ~ 9 "." occur only 1 time

"0000" "1001" "1111"

$a_x \cdots a_1 . b_1 \cdots b_x \cdots \to$ 0, 1, 2, 3, ⋯ 4π ⋯ ⋯

4-digit binary codes

No need to consider the problem of "$0.\dot{9} = 1.0$".

injection.

Thm. $(\mathbb{N}\to\mathbb{N})\approx 2^{\mathbb{N}}$

[Proof.] $2^{\mathbb{N}} \preccurlyeq \mathbb{N}^{\mathbb{N}} \preccurlyeq \mathbb{R}^{\mathbb{N}} \approx (2^{\mathbb{N}})^{\mathbb{N}} \approx 2^{\mathbb{N}\times\mathbb{N}} \approx 2^{\mathbb{N}}$

$(\mathbb{N}\to\mathbb{N})$

$\therefore \mathbb{N}^{\mathbb{N}} \approx 2^{\mathbb{N}}$

Thm. $\mathbb{R}^{\mathbb{R}} \approx 2^{\mathbb{R}} \approx 2^{2^{\mathbb{N}}}$.

Thm. The set of continuous functions from $\mathbb{R}$ to $\mathbb{R}$ is equinumerous to $\mathbb{R}$.

[Thought] Dedekind 切割定理. 只要知道有理数点. 就可以

单射: 常数函数 (连续) (无理数处用有理数逼近即可)

$\mathbb{R} \preccurlyeq A \preccurlyeq \mathbb{R}^{\mathbb{Q}} \approx (2^{\mathbb{N}})^{\mathbb{N}} \approx 2^{\mathbb{N}\times\mathbb{N}} \approx 2^{\mathbb{N}} \approx \mathbb{R}$

↓ Continuous functions set

the set of

Prove that all monotonically increasingly functions from $\mathbb{R}$ into $\mathbb{R}$ is equinumerous to $\mathbb{R}$.

[Proof] 1. Disconnect points are countable.

$\lim_{x\to x_0^-} f(x) > \lim_{x\to x_0^+} f(x)$. $\exists q\in\mathbb{Q}$. $\lim_{x\to x_0^-} f(x) < q < \lim_{x\to x_0^+} f(x)$ Thus, countable disconnect points $\approx \mathbb{Q}$.

2. Mono Func $\longrightarrow \mathbb{R}^{\mathbb{Q}}\times\mathbb{R}^{\mathbb{N}}$ (injection)

their function values

f is not continous on x_0. iff. $\lim_{x\to x_0^+} f(x) \neq \lim_{x\to x_0^-} f(x)$.

由 $\mathbb{R}^{\mathbb{Q}}$ 我们找到了间断点位置 (无理)

由 $\mathbb{R}^{\mathbb{N}}$ 我们确定无理间断点的值

We can also construct a surjection

$H: \mathbb{R}^{\mathbb{Q}}\times\mathbb{R}^{\mathbb{N}} \longrightarrow$ Mono Func.

e.g. 3个间断点.

$\mathbb{R}^{\mathbb{N}}$ { 1~□ 2~□ 3~□ 4 5 ~ 随意 ⋮

Let $f(x) = \begin{cases} f(a) + \frac{x-a}{b-a}[f(b)-f(a)] \\ f_0(x) \end{cases}$ $(x\in(a,b).\ x\notin\mathbb{R}^{\mathbb{N}}$ (↑ ↑ $\mathbb{Q}$ $\mathbb{Q}$) $(f_0(x)\in\mathbb{R}^{\mathbb{N}})$

$R = \{(x,y) \mid x-y\in\mathbb{Q}\}$ $P = \{[x]_R \mid x\in\mathbb{R}\}$.

$P\to\mathbb{R}$: ~~$f(a)\ a\to[x]_R=a \Rightarrow f(a)=x$~~ $f([a]_R) = a$.

$\mathbb{R}\to P$: Define g^*: $2^{\mathbb{N}}\to\mathbb{R}$. s.t. $\forall l_1, l_2.\ l_1\neq l_2.\ g^*(l_2)-g^*(l_1)\notin\mathbb{Q}$

↑ 为证明是单射

$2^n \le g(l,n) = 2^n + \sum_{i=0}^{n-1} l(i)\cdot 2^i < 2^{n+1}$ ← l 的第 i 位. 0/1

到第 n 位小数之后

e.g. $l = 001010\cdots$

$g(l,1) = (1)_2 = 1$

$g(l,1) = (10)_2 = 2$

$g(l,2) = (100)_2 = 4$

$g(l,3) = (1100)_2 = 12$

$g(l,4) = (10100)_2 = 20$

Fact: $\forall l_1\neq l_2.\ \exists n.$ $\forall n'>n.$

$g(l_1,n') \neq g(l_2,n)$.

$2^n \le g(l,n) < 2^{n+1}$

$g^*(l) = \sum_{n=0}^{\infty} 10^{-g(l,n)}$

这样保证 $f(l_1)-f(l_2)$ 由1消掉

$f(l) = \sum_{n=0}^{\infty} 10^{-g(l,n)}$

$f(l_1)-f(l_2)+\frac{1}{9} \to 111111\cdots$

↓ 0 2

(处理 退一)

前位 有0/2

一定没有循环节

$(2^n\le g(l,n)<2^{n+1})$

Back to Axiomatic Set Theory

- Other Axioms in ZF(C)

Def. Axiom Scheme of Replacement 替换

For any set A and any "mapping" F. $\{F(a) \mid a \in A\}$ exists.

$X \longmapsto X \times \mathbb{N}$

X: any set

a mapping

which is not a function,

the latter is foremost a relation. (Since the set of all sets does not exist)

Formally, $\left(\forall x\, \forall y_1\, \forall y_2 \left(\phi_{[y \mapsto y_1]} \wedge \phi_{[y \mapsto y_2]} \longrightarrow y_1 = y_2\right)\right)$

(ϕ: any propositions)

$\longrightarrow \forall v\, \exists w\, \forall y \left(y \in w \leftrightarrow \exists x \left(x \in v \wedge \phi_{[\quad]}\right)\right)$

y.

where x, y may freely occur in ϕ

y_1, y_2, v, w does not freely occur in ϕ.

Def. Axiom of Regularity 正则

$\forall x \left(x \notin x\right)$

Def. (ZFC) Axiom of Choice 选择

Need to state it if you want to use Axiom of Choice

For any set S, if $\emptyset \notin S$ then exists $f: S \longrightarrow US$ s.t. $f(x) \in X$ (for any $x \in S$)

e.g. $S = \{\{0\}, \{1, 2\}, \{3, 4, 5\}\}$

f

$US = \{0, 1, 2, 3, 4, 5\}$

e.g. $R = \{(x, y) \mid x - y \in \mathbb{Q}\}$. $P = \{[x]_R \mid x \in R\}$

$F: P \longrightarrow UP$. Thus, we need Axiom of Choice.

- Constructing Certain Objects in ZF(C)

$\mathbb{N} = \{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}, \cdots\cdots\}$

0 1 2

$\mathbb{Z} \approx \mathbb{N}$. of course, we can use a bijection to encode $\mathbb{Z}$.

$\mathbb{Z} \approx \mathbb{N} \times \{0, 1\} \setminus \{(0, 0)\}$

$+$ $-$ 去掉"正零"表述

What to do with "$\mathbb{N} \subseteq \mathbb{Z}$"?

$\mathbb{Z} \approx \mathbb{N} \cup \underbrace{(\mathbb{N} \setminus \{0\}) \times \{1\}}_{\text{negative integers}}$. ⟶ To simplify, we use the former one.

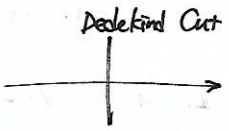
($\mathbb{Z} := \mathbb{N} \times \{0,1\} \setminus \{(0,0)\}$)

$\mathbb{Q} \approx \{(n,m) \mid n \in \mathbb{Z},\ m \in \mathbb{N}^* \setminus \{0\},\ \underline{(n,m)=1}\}$

n, m are co-prime to each other

Def. $(X, Y) \in \mathcal{P}(\mathbb{Q}) \times \mathcal{P}(\mathbb{Q})$ is a Dedekind Cut of rational numbers if

1. $X \neq \emptyset$, $Y \neq \emptyset$
2. $X \cup Y = \mathbb{Q}$
3. $\forall x \in X\ \forall y \in Y\ (x < y)$
4. Y does not have a least element. i.e. $\forall y \in Y\ \exists y' \in Y.\ (y' < y)$



* If the cut $\in \mathbb{Q}$, the cut $\in X$.

$\mathbb{R} := (X, Y)_{\text{Dedekind Cuts}}$

~~Thm. Dedekind Cuts ≅ $\mathbb{R}$.~~

{ injection ✓
surjection $\underline{\sup X \text{ exists}}$. $\underline{x = \sup X \in \mathbb{R}}$.

Def' Suppose $l: \mathbb{N} \longrightarrow \mathbb{Q}$ "a infinite sequence of rational numbers"

We call l a Cauchy Sequence. $\forall \varepsilon > 0,\ \exists N \in \mathbb{N}.\ \forall n, m \geq N.$

$|l(n) - l(m)| < \varepsilon$.

[Thought]. Notice that multiple C.S. might have the same limitation.

Use "Identification Class".

[Def]. Suppose l_1, l_2 are two Cauchy Sequences of rational numbers.

They are "equivalent" iff $\forall \varepsilon > 0.\ \exists N.\ \forall n > N.\ |l_1(n) - l_2(n)| < \varepsilon$.

- reflexive ✓
- symmetric ✓ (abs)
- transitive ✓ $< \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$.

} ⟶ equivalent relation $=: \mathcal{R}$.

$\mathbb{R} := \{[a]_{\mathcal{R}} \mid a \text{ is a Cauchy Sequence of rational numbers}\}$.

GRAPH THEORY 图论

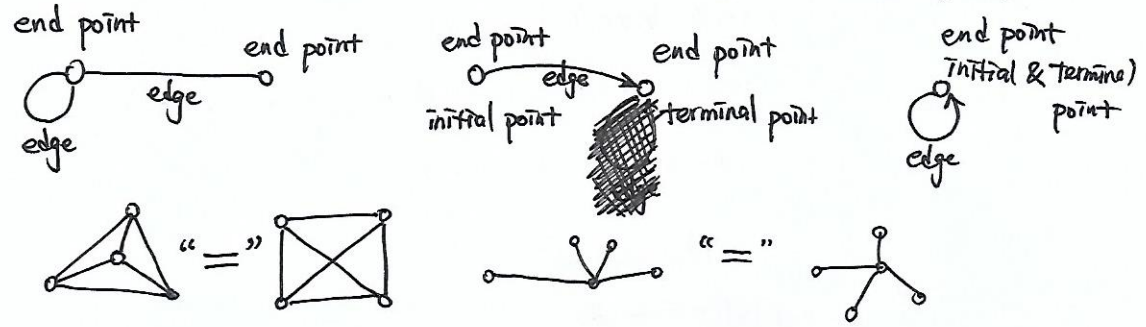
- Graph

Def. A Graph $G=(V,E)$. in which

$\begin{cases} V \text{ is a set of vertices (nodes)} \\ E \text{ is a set of edges.} \end{cases}$

Infact, G includes other information : $\begin{cases} f: E \to V & \text{initial nodes} \\ g: E \to V & \text{terminal nodes} \end{cases}$

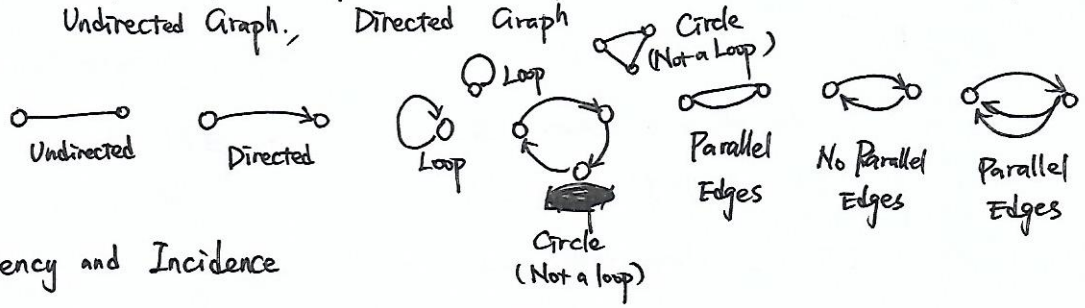
1. $G:=(V, E, fst, snd)$ 2. $G:=(V, E, \mathcal{E})$ $\mathcal{E}: E \to \mathcal{P}(V)$



- Undirected / Directed Graph, Parallel Edges, Loops,

Def. Simple Graph, Simple Directed Graph —— No Loop, No Parallel Edge

Undirected Graph, Directed Graph

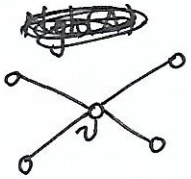


- Adjacency and Incidence

e is incident with u, v.

u, v are adjacent / neighbors

- Neighborhood



$Ngh(v)$ is the set of all neighborhoods of v. "Neighborhood"

$Ngh(A) := \bigcup_{v \in A} Ngh(v)$ for $A \subseteq V$.

- Degree

Def. If $G=(V, E)$

$Deg(v) :=$ "the number" of edges incident with v. $(v \in V)$

Specially : if there's a loop, it contributes 2 to the degree.

$$Deg(v) := \left| \{e \in E \mid fst(e)=v\} \right| + \left| \{e \in E \mid snd(e)=v\} \right| = \sum 1 + \sum 1$$

the first endpoint — the second endpoint — $fst(e)=v$ — $snd(e)=v$

Thm. If $G=(V,E)$ is a finite undirected graph.

then $2|E| = \sum_{v\in V} Deg(v)$. (握手定理)

[Proof] $\sum_v Deg(v) = \sum_v \left(\sum_{fst(e)=v} 1 + \sum_{snd(e)=v} 1 \right) = \sum_v \sum_{fst(e)=v} 1 + \sum_v \sum_{snd(e)=v} 1$

$= \sum_e \sum_{fst(e)=v} 1 + \sum_e \sum_{snd(e)=v} 1 = \sum_e 1 + \sum_e 1 = |E| + |E| = 2|E|$

(only work for finiteness) Qed. □

Def. If $G=(V,E)$ is a directed graph.

In-degree: $Deg^-(v) = |\{e\in E \mid e \text{ is from } \overset{\text{some}}{u} \text{ to } v\}|$ $\quad \sum_v Deg^-(v) = |E|$

Out-degree: $Deg^+(v) = |\{e\in E \mid e \text{ is from } v \text{ to } \overset{\text{some}}{u}\}|$ $\quad \sum_v Deg^+(v) = |E|$

$Deg(v) = Deg^-(v) + Deg^+(v)$ is definitely an even number.

- Special Graphs

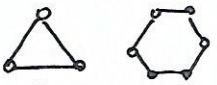
Complete Graph K_n: a complete graph with n vertices

↳ is firstly a simple graph (undirected).

$$|E_{K_n}| = \frac{n(n-1)}{2}$$

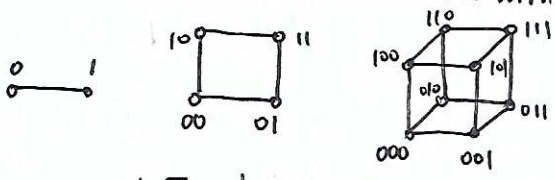
Cycles C_n: a cycle (must be a simple graph)

$$|E_{C_n}| = n$$



n-Cube: relations between n-bit binary codes.

u and v are connected with an edge iff. only one bit is different.



$$|E_{Q_n}| = n\,2^{n-1}$$

Bipartite Graphs (二分图)

$G=(V,E)$ is a bipartite graph if exists a partition $\{V_1, V_2\}$ on V ($V_1\neq\emptyset$, $V_2\neq\emptyset$)

s.t. every edge connects a vertex from V_1 and a vertex from V_2.

显然不能有三角形

Complete Bipartite Graphs (完备二分图)

$G.\ \{V_1, V_2\}$. for any $u \in V_1$, $v \in V_2$. exists $\{u,v\}$ in E.

$|V_1|=m.\ |V_2|=n \longrightarrow K_{m,n}$ (Notation)

Matching (匹配)

$M \subseteq E$. so that ~~no edges in M~~

no two edges in M share/are incident with the same vertex.

A *maximal matching* is a matching with the largest number of edges

$\emptyset$ is a matching.

A complete matching (完备匹配)

G — B.g. Bipartition $= \{V_1, V_2\}$.

A complete matching from V_1 to V_2 is a matching M s.t. every vertex in V_1 is incident with an edge in M.

Thm. Hall's theorem

If $G = (V, E)$ is a bipartite graph with a bipartition (V_1, V_2)

G has a complete matching from V_1 to V_2 iff.

for any $A \subseteq V_1$, $|A| \leq |Ngh(A)|$

[Proof] $\Rightarrow$ is Obvious

$\Leftarrow$: Proof by induction.

Base Step. 1. if $|V_1|=0$. Obvious.

Base Step 2. if $|V_1|=1$. When $|A| \leq |Ngh(A)|$ $(A \subseteq V_1)$.

i.e. $|Ngh(v_1)| \geq 1$. exists matching ~~.

Induction Step. Case 1. For any $A \subseteq V_1$. $A \neq \emptyset$. $A \neq V_1$.

$|A| < |Ngh(A)|$ holds.

Pick $u \in V_1$, $v \in Ngh(u)$.

$|V_1'| = n-1$ ← $V_1 \setminus \{u\}$ and $Ngh(V_1 \setminus \{u\})$ satisfies the requirement.

We already know there exists a complete matching $\mathcal{E}$.

Then $\mathcal{E}^* = \mathcal{E} \cup \{u,v\}$ is a complete matching from V_1 to V_2.

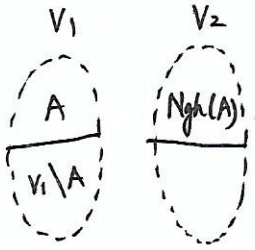
Case 2. $\exists A \subseteq V_1$. $A \neq \emptyset$. $A \neq V_1$. $|A| = |Ngh(A)|$

Divide V_1 into $\{A, V_1 \setminus A\}$.

We know: $\forall B \subseteq V_1 \setminus A$. $Ngh(B) \subseteq Ngh(A \cup B)$

Thus, $|Ngh(B)| \geq |A \cup B| = |A| + |B|$

a new graph ~~$Ngh_{V_1 \setminus A}(B) = Ngh(\quad) \setminus Ngh(A)$~~



$$|Ngh_{newgraph}(B)| = |Ngh(A \cup B) \setminus Ngh(A)|$$
$$= |Ngh(A \cup B)| - |Ngh(A)|$$
$$\geq |A \cup B| - |A| = |A| + |B| - |A| = |B|.$$

The requirements are still satisfied.

Thus, exists a complete matching $\mathcal{E}_1$ from A to $Ngh(A)$.

and ——— $\mathcal{E}_2$ $V_1 \setminus A$ to $Ngh(V_1 \setminus A)$.

$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$ is a complete matching from V_1 to V_2.

- Paths, Circuits 路径, 回路

  Def. A path of length n from u to v : $e_1, e_2, \cdots e_n$ such that $(u = x_0), x_1, x_2, \cdots (x_n = v)$ ~~are endpoints~~ where e_i is from x_{i-1} to x_i (for each i).

  Def. In $G = (V, E)$, a path of positive length $e_1, e_2, \cdots e_n$ from u to u is called a circuit.

  Def. Simple path / circuit: Path / Circuit not covering the same edge for two times.

  ~~Specially, in simple graph, simple path never visit the same~~

  Specially, in simple graph, we use only vertices to express the path/circuit.

- Connectivity 连通性, Strong-Connectivity, Reachability 可达性

  Def. In undirected graph $G=(V, E)$. u, v are connected if there exists a path from u to v. $(u, v \in V)$

  Def. An undirected graph $G=(V, E)$ is connected if for any $u, v \in V$, u and v are connected.

  Def. $G = (V, E)$. $G' = (V', E')$.

  G' is a subgraph of G iff. $V' \subseteq V$, $E' \subseteq E$. If $G \neq G'$, G' is a proper subgraph.

  ↑

  * Default. every edge has its endpoints.

Def. Induced Subgraph

$G=(V,E)$. $V' \subseteq V$. $E' = \{e \in E \mid e \text{ is incident with } u \text{ and } v \text{ and } u,v \in V'\}$.

We call $G'=(V', E')$ { the subgraph induced by V'. / an induced subgraph of V' in G.

Def. Connected Components 连通块 / 连通分支

$G=(V, E)$ is an undirected graph.

A connected component (of G) is a connected subgraph that is not a proper subgraph of another connected subgraph of G.

Thm. In undirected graph $G=(V, E)$, the connectivity relation is an equivalence relation.

Thm. If G is nonempty then G's connected components are induced subgraph of equivalent classes of the connectivity relation in G.

Also another definition of Connectivity Component

[Proof] 1. If $G'=(V', E')$ is G's connected component.

1a) Since G is nonempty, V' is also nonempty.

Suppose $v \in V'$. We are to prove $V' = [v]_{con(G)}$, i.e. $V' \subseteq [v]_{con(G)}$, $V' \supseteq [v]_{con(G)}$

V'相同

$\supseteq$: a path $v-x-y-\cdots-u$ $(\forall u \in [v]_{con(G)})$ if $u \notin V'$.

We add the path into the connected component, get a new subgraph G_1 of G.

G_1 is also a connected subgraph but it's not a subgraph of G'.

$\subseteq$: $\forall u \in V'$. u is connected to v in G'

Thus, u is connected to v in G.

i.e. $u \in [v]_{con(G)}$.

In summary, $V' = [v]_{con(G)}$

E'相同

1b) For any $e \in E$ and e is from u to v, if $u, v \in V'$, then $e \in E'$.

(If not, $(V', E' \cup \{e\})$ is a "larger" connected subgraph. Contradiction!)

2. If V' is an equivalent class of $con(G)$. $G'=(V', E')$ is a subgraph of G induced by V'.

2a) We prove G' is (a) connected (subgraph). ← obvious.

For any $u, v \in V'$, we know u is connected to v in G.

Suppose the path from u to v is $e_1, e_2, \cdots e_n$, passing $x_1, x_2, \cdots x_{n-1}$.

Obviously $x_i \in [u]_{con(G)} = V'$. (Every x_i have a path leading to u).

Thus, $e_i \in E'$.

Thus, u is connected to v in G'.

2b) G' is the "max" connected subgraph.

one more vertex → not connected

one more edge → $\nexists$ G' is the induced subgraph. Contradiction!

Def. Strongly-Connected Directed Graph 强连通图

Strongly-connected: exist a path from u to v and v to u. (in directed graph)

a path from

A strongly-connected component of $G=(V, E)$ (directed) is a strongly-connected subgraph that is not a proper subgraph of another strongly-connected subgraph of G.

Def. Reachability —— exists a path from u to v.

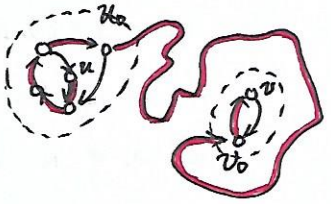
Mutual Reachability —— exists paths from u to v and v to u. * symmetric

In undirected graph. reachability = connectivity = mutual reachability.

Thm. In directed graph $G=(V, E)$, the mutual reachability is an equivalence relation.

Thm. If G is nonempty then G's strongly-connected components are induced subgraph of equivalent classes of the mutual reachability relation in G.

Thm. If $u_0, v_0 \in V$, v_0 is reachable from u_0, then for $u \in [u_0]_{MRch(G)}$ and $v \in [v_0]_{MRch(G)}$, v is reachable from u. (Obvious)



Thm. If $e_1, e_2, \cdots e_n$ is a circuit in directed graph G, passing through $x_1, x_2, \cdots x_n$.

Then $e_1, e_2, \cdots e_n, x_1, x_2, \cdots x_n$ belongs to the same strongly-connected component.

[Proof]. $x_0 \in [x]_{MRch(G)}$. $e_0 \in E' \longrightarrow V$

Induced Subgraph!

- Connectivity with Vertices and Edges Removal or Added.

① Add an edge in an undirected graph.

Thm. Suppose $G=(V, E)$ and $G'=(V, E \cup \{e_0\})$. are directed graphs. $e_0 \notin E$. e_0 is from u to v.

(1) If u connects to v in G, i.e. $[u]_{conn(G)} = [v]_{conn(G)}$

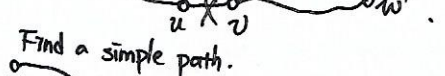
then for any $w \in V$. $[w]_{conn(G)} = [w]_{conn(G')}$.

(2) If u is not connected to v in G. then $[u]_{conn(G')} = [v]_{conn(G')} = [u]_{conn(G)} \cup [v]_{conn(G)}$

[Proof]. (1) $G \to G'$. Obvious. (左大表命题右大表) $[w]_{conn(G)} \subseteq [w]_{conn(G')}$

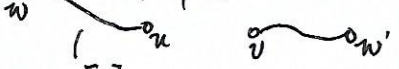
$G' \to G$.

F1.



connected. Qed.

F2. Find a simple path.



$w \in [u]_{conn(G)}$. $w' \in [v]_{conn(G)} \Rightarrow w, w' \in [u]_{conn(G)} = [v]_{conn(G)}$

(2) $[u]_{conn(G')} \supseteq [u]_{conn(G)} \cup [v]_{conn(G)}$.

$w \in [u]_{conn(G')}$. $w \to u$.

$x \in [v]_{conn(G')}$ $x \to v \to u$. Obvious

2 $\xrightarrow{\text{merge}}$ 1.

$[u]_{conn(G')} \subseteq [u]_{conn(G)} \cup [v]_{conn(G)}$

simple path. $(x \in [u]_{conn(G')})$ $x \to u$

① 有 e_0 : $x \to v \to u$. $x \in [v]_{conn(G)}$

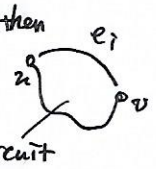
② 无 e_0 : $x \to u$ $x \in [u]_{conn(G)}$

② Remove an Edge e_i in an undirected graph

1) e_i is included in a circuit. Nothing changed

2) e_i is not included ~. $|conn(G)|+1$.

If u and v are connected after e_i removed. then



Def. Cut Edge / Bridge

Suppose $G=(V, E)$ is an undirected graph.

A cut edge (or bridge) is an edge $e \in E$. s.t. the removal of the edge e results in more connected components.

② Add an edge in a directed graph.

Thm. Suppose $G=(V, E)$. $G'=(V, E \cup \{e_0\})$ are directed graphs, $e_0 \notin E$.

e_0 is from u to v.

(1) $u \in [v]_{MRch(G)}$. for any $w \in V$. $[w]_{MRch(G')} = [w]_{MRch(G)}$.

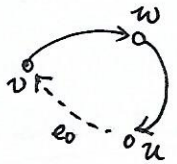
(2) u is reachable from v in G but v is not reachable from u in G.

$\geq 2 \xrightarrow{\text{merge}} 1$. then $[u]_{MRch(G')} = [v]_{MRch(G')} = \{w \mid u$ is reachable from w in G and w is reachable from v. in $G.\}$

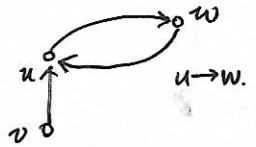
e.g.

[Proof] 1.) u is now mutually reachable from v. $[u]_{MRch(G')} = [v]_{MRch(G')}$.

2) $\{\} \subseteq [u]_{MRch(G')}$



3) $[u]_{MRch(G')} \subseteq \{\}$



(1) e_0 not included $v \to w$ ✓

(2)

$v \to w$ ✓

(3) u is not reachable from v in G.

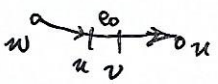
Nothing happen to $[w]_{MRch(G')}$.

$w \to u$.

(1) e_0 not included. $w \to u$ ✓

(2) e_0 included

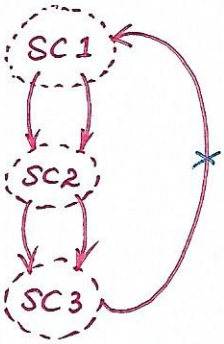
$w \to u$ ✓.



④ Remove an edge in a directed graph.

[Prop.] If removing an edge and there are k strongly-connected components, there are $(k-1)$ other edges that will cause the number of strongly-connected components increase. (FALSE !!)

Counter-example:



→ rise to 3.

SC1, SC2, SC3 are three complete directed graphs themselves.

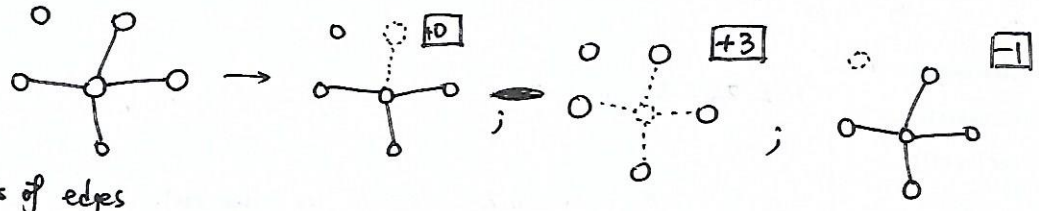
↓

Remove any edge won't influence any vertex's reachability.

⑤ Remove a vertex in an undirected graph

Def. Cut Vertices

$G=(V,E)$ is an undirected graph, a cut vertex (articulation vertex) is a vertex the removal of which would result in more connected components.



⑥ Remove lots of edges

Def. In an undirected graph $G=(V,E)$. $E'\subseteq E$ is called an edge cut if $G-E'$ is disconnected.

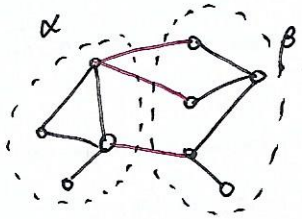
For disconnected graph. $\forall x \in P(E)$ is an edge cut. All graphs have its edge cut. (All edges)

Def. In an undirected graph $G=(V,E)$. $V'\subseteq V$ is called a vertex cut if $G-V'$ is disconnected.

~ A complete graph does not have a vertex cut.

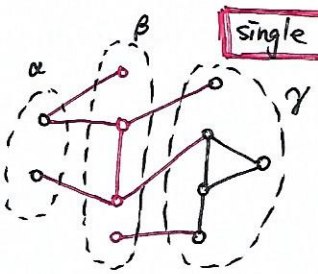
Edge Connectivity $\lambda(G)$ —— $\min|E'|$. (E' is an edge cut of G)

Vertex Connectivity $\kappa(G)$ —— $\min|V'|$. (V' is a vertex cut of G or $G-V'$ is a single vertex)



Edge Cut.

α & β: disconnected

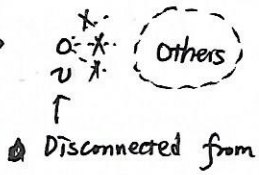


Vertex Cut

α & γ: disconnected.

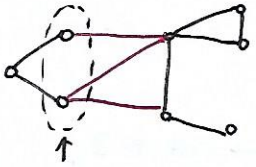
Thm. If $G=(V,E)$ is an undirected graph and $|V|\geq 2$.

$$\kappa(G) \leq \lambda(G) \leq \min_{v\in V} \deg(v)$$

[Understanding] 1. $\lambda(G) \le \min_{v \in V} \deg(v)$. Obvious → 

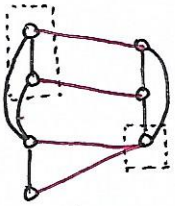
Disconnected from others

2. $\kappa(G) \le \lambda(G)$.

CASE 1. 

Delete

Not all vertices left/right are ~~connected~~ incident with edge cuts.

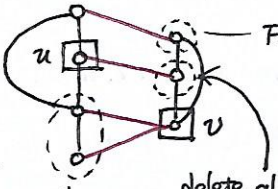
CASE 2. 

Delete all left vertices → Only right left. Conn.

Same when deleting all right vertices.

↓

Pick two vertices which are not adjacent to (one left, one right) each other.

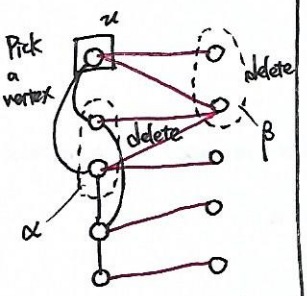


For those still connected by an edge in edge cuts, randomly pick one.

delete all vertex adjacent to u in e.c.

delete/cut all vertex adjacent to v in edge cuts

CASE 2 METHOD 2.



Each vertex in ~~cut~~ vertices cuts is correspondent to an edge in edge cut.

$\therefore \kappa \le \lambda$.

In the remaining graph, u is disconnected from all other vertices.

α: one vertex ~ one edge in edge cut

β: one vertex ~ one edge in edge cut

(no same edge used in α)

$\Rightarrow \kappa \le \lambda$

CASE 2'. If n vertices are all adjacent to all m vertices in right part.
(left) (right)

We have mn edges in the edge cut.

We pick a vertex, cut all others. $(m+n-1)$ vertices in vertex cut.

$mn \ge m+n-1 \iff (m-1)(n-1) \ge 0$. $(\because |V| \ge 2)$

Thus, $\kappa \le \lambda$. → $\kappa(G) \le \lambda(G)$

- Tree

Def. Tree is a connected ~~graph~~ (nonempty) graph without any circuits. (undirected)

We make it a rule that tree has at least 1 vertex.

Thm. If adding an edge to a tree, the new graph will have a simple circuit.

[Proof] For any $u, v \in V$, there exists a path from u to v in T.

Let $G = T \cup \{e_0\}$.

↓

Definitely having a simple path

The simple path and e_0 construct a simple circuit in G.

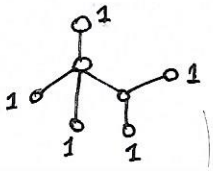
Thm. If removing an edge from a tree, the new graph is disconnected.

[Proof] By Contradiction.

If still connected. simple path + e → circuit.

Def. Leaves

A vertex is a *leaf* is a tree if its degree is 1.



Thm. Every tree (if $|V| \geq 2$) has at least one leaf.

[Proof] By Contradiction.

If not. pick a vertex. $\deg() \geq 2 \xrightarrow{1}$ $\deg() \geq 2 \xrightarrow[1]{\text{another edge}}$ $\deg() \geq 2 \xrightarrow{\text{another edge}}$ $\deg() \geq 2$

.....

Finite Graph. Back to a vertex visited. Circuit. Contradiction.

Thm. Every tree (if $|V| \geq 2$) has two leaves.

[Proof]. Pick the (leaf.) → ○ → ○ → .... → ○ STOP. Otherwise, Infinite.

deg()=1

↑

Another leaf.

Thm. A tree with n vertices has $(n-1)$ edges.

BASE STEP. $n=1$. Obvious.

INDUCTION STEP. cut a leaf → still a tree (still connected and w/o circuits). ✓

[Propo]. If add an edge and then has a circuit. (w/o circuits before). it's a tree.

[Proof] By Contradiction. If not connected. → won't have a circuit if new edge connects two. conn. comp.

[Propo]. If cut any edge and then disconnected. (connected before) it's a tree.

[Proof] By Contradiction. If has a circuit → cut an edge. still connected.

Thm. All edges in a tree are cut edges.

In fact, those are all equivalent definition of a tree.

- G is connected and w/o simple circuits
- G is connected and $|E| = |V| - 1$.
- G is connected and every edge in G is a cut edge.
- G contains no simple circuits and $|E| = |V| - 1$.
- G contains no simple circuits, adding any edge to G generates a simple circuit.
- There is a unique simple path between any two of G's vertices.

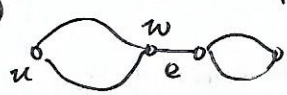
[Pf.] ⇒: Obviously. there exists at least one path between any two of G's vertices.

Assume there are two vertices u_1, v_1. between which there are more than one path.

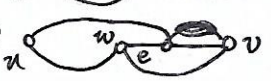
Consider the case where the total length $\mathcal{L}(u_2, v_2)$ is minimum. u_2, v_2. i.e. $\mathcal{L}(u_2, v_2) = \text{Min}$

Then the two paths construct a circuit.

Now prove it is a simple circuit. Assume there exists a shared edge e_0.

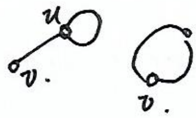
1) 

Therefore, $\mathcal{L}(u \to w) < \mathcal{L}(u \to v)$. Contradiction!

2) 

$\mathcal{L}(u \to w) < \mathcal{L}(u \to v)$ ——— !

⇐: Obviously G is connected.

If exists a circuit,  ⇒ ∃ two simple paths between u and v.

Qed.

* Also can be used when there is only one vertex. ("any two", not "any two different")

- Rooted Trees

Def. A rooted tree is a tree with a designated vertex (as the root).

We adopt the definition that a rooted tree is an undirected graph.

Def. Level 层数

$G=(V,E)$ is a rooted tree with the root r.

the level of a vertex v is the length of the unique simple path from r to v.

(Some definitions make level = length + 1)

the height of a tree is the maximal level of its vertices

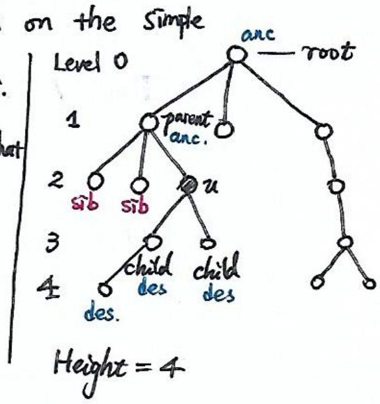
the parent of a vertex $v \neq r$ is the unique vertex u s.t. $\{u,v\} \in E$ and this edge appears in the unique simple path from r to v.

Also, v is called a child of u.

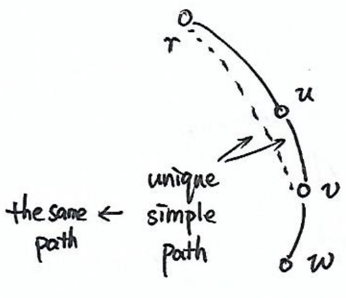
Vertices u, v are siblings if they have the same parent.

the ancestors of a vertex $v \neq r$ is the vertices on the simple path from r to v, excluding v but including r.

the descendants of a vertex v are all vertices that have v as one of their ancestors.



Thm. In a rooted tree G, if u is one ancestor of v and v is an ancestor of w, then u is an ancestor of w.



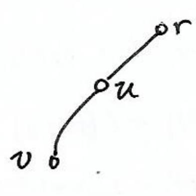
Also, need to prove $w \neq u$ (obvious) → Defn.

Thm. If G is a rooted tree, $Anc := \{(u,v) \mid u \text{ is an ancestor of } v \text{ in } G\}$.

$Par := \{(u,v) \mid u \text{ is the parent of } v \text{ in } G\}$.

Anc is the transitive closure of Par.

[Proof] 1. If u is an ancestor of v. i.e. $(u,v) \in Anc$.



Suppose the path from v to r is: $v, e_1, x_1, \cdots e_k, u, e_{k+1}, \cdots e_n, r$ (with $v = x_0$, $u = x_k$, $r = x_n$)

Thus, x_i is the parent of x_{i-1} in G.

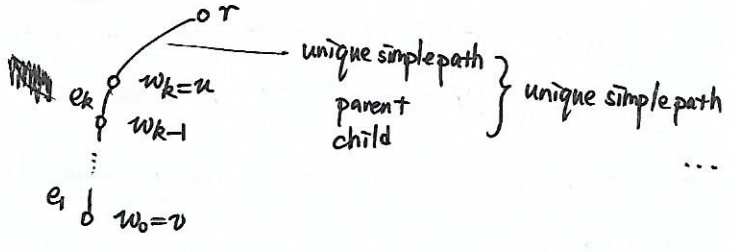
$(\forall i \in \{1, 2, \cdots, n\})$

Therefore, $(u,v) \in Par^k$.

2. If $(u,v) \in Par^+$.

Since $Par^+ = \bigcup_{k=1}^{\infty} Par^k$, exists $k \in \mathbb{N}^+$, $w_0, w_1, \dots w_k$.

$v = w_0$, $u = w_k$. w_{i+1} is the parent of w_i $(i \in \{0,1,2,\dots k-1\})$



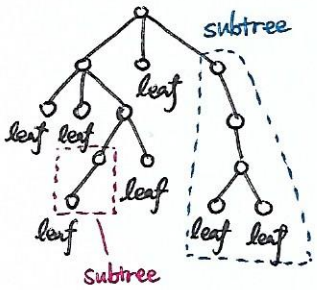
Thus, $(u,v) \in Anc$.

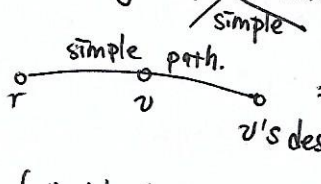
Thus. $Anc = Par^+$.

Def. a vertex is a *leaf* if it has no children. Otherwise, it is an *internal vertex*.

Def. a subtree is the subgraph induced by the vertices including a vertex and all it descendants.

[Pf.] Obviously w/o circuits.





→ connected.

(* Need to prove all vertices in the simple path from v to v's des. is either v or v's des.)

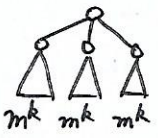
Def. If G is a rooted tree, we call it

a m-ary tree if all its internal vertices have at most m children.

a binary tree if it is 2-ary.

Thm. An m-ary tree of h height has at most m^h leaves.

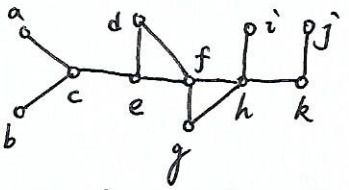
[Pf.] Induction



At most m children

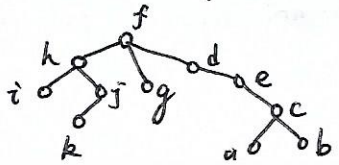
→ At most m^{k+1}. $(m \cdot m^k)$

- Depth-First Search (DFS) —— A way to spin a tree
  生成 樹



1) Start from a vertex
2) Continuously augment a simple path until no edge is augmentable. Also, never get to a visited vertex.
3) Trace back. Until the deepest vertex with an available unvisited edge to an unvisited a vertex.

DFS:



Def. Spanning Tree 生成树

Suppose $G=(V,E)$ is an undirected graph.

A spanning tree of G is a subgraph of that contains all vertices of G.

Thm. All connected graph have their spinning trees.

[Pf.] If there's a circuit, cut an edge. → Still connected

Continue until $|E^*|=|V|-1$. → A tree.

Notation. In DFS, the kth move is from u to v through edge $\{u,v\}$.

Forward / Backward Step.

Step 2 Step 3.

Thm. Properties of DFS.

Suppose a DFS process with s steps whose kth step is from v_{k-1} to v_k through e_k.

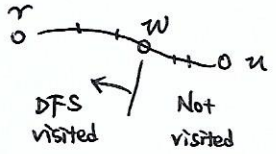
(1) If exist $e\in E$, $w\in V$ s.t. $w\notin\{v_0,v_1,\cdots v_k\}$, e connects v_k and w.
then $(k+1)$th step is forward.

(2) If kth step is forward, $v_k\notin\{v_0,v_1,\cdots v_{k-1}\}$

(3) If kth step is backward, $v_{k-1}=v_i$, the i-th step is forward.
then $v_k=v_{i-1}$. (从哪来回哪去) ——— 回溯

(4) All edges in DFS's forward steps forms a spanning tree of G.

[Pf.] 1. All vertices are visited in the DFS process.



So we take a backward step when there are still edges augmentable.

Contradiction.

2. We prove after a vertex visited, the DFS's forward-step edges form a tree.

BASE. o root is a tree.

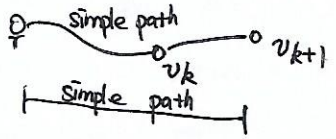
INDUCTION. 1) connected. add an edge connecting new vertex with a conn. graph.

2) no circuit. The deg(current new edge)=1.

Not forming any circuits.

Q.ed.

(5) Forward step: $v_k\to v_{k+1}$. v_k is the parent of v_{k+1}.



(6) Every vertex (except root) is the terminal vertex of exactly one forward step. → F.S. only goes to unvisited vertices

initial vertex of exactly one backward step.

Lemma. Suppose $G=(V,E)$ is a connected undirected graph and a DFS process of G contains s steps while the k-th step is from v_{k-1} to v_k through e_k.

For every $k \leq s$, the set of "visited vertices" $\{v_0, v_1, \cdots v_k\}$ can be divided into three categories:

- the starting vertex
- vertices which appear as exactly one forward step's termination and as no backwards step's starting point in the first k-th steps.

  → Enter the subtree of it and Haven't came out yet
- vertices which appear as exactly one forward step's termination and as exactly one backwards step's starting point in the first k-th steps.

  → Enter the subtree and Have already came out.

And one simple path from v_0 to v_k satisfies:

- It contains only forward edges: $e_{n_1}, e_{n_2}, \cdots e_{n_l}$. where $1 = n_1 < n_2 < \cdots < n_l < k$
- It passes through $v_0 = v_{n_1-1}, v_{n_1} = v_{n_2-1}, \cdots v_{n_{l-1}} = v_{n_l-1}, v_{n_l} = v_k$.
- $\{v_0, v_{n_1}, \cdots v_{n_l}\}$ contains only the first two categories.

[Proof] INDUCTION.

k-th step. All holds.

(1) forward step → Cata. II + $\{v^*\}$ (new vertex). simple path + e_{new}. → holds.

(2) backwards step → v_k : CATA.II CATA.III. simple path − e_{k+1} → holds.

Qed.

Corollary. 1. Every vertex appears as the starting point of exactly one backwards step.

2. Forward step: $(n-1)$ (↑ $(n-1)$ terminations). Backwards step: $(n-1)$ (↑ $(n-1)$ starting points)

3. If $v_i = y = u$. then v_k is a descendant of u. $(i<k<j)$

[pf.] i, j: u is CATA.II.

k: $u \in$ CATA.II → u is included in the simple path from r to v_k.

→ descendants.

Qed.

Thm. A back edge is always from a vertex to one of its ancestors, one of its descendants or itself.

* a back edge is an edge in G which is not included in the DFS spanning tree of G. (DFS 트리)

[Proof] ~~By contradiction.~~

Suppose there exists an edge between u and v. and u has an ealier backwards step.

(0) $u = v$.

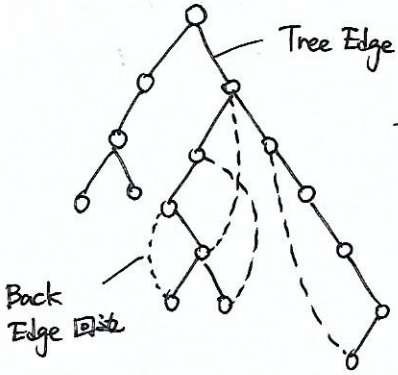
(1) At k-th step. $v \in$ CATA.II. (k-th) $\rightarrow$ v is included in the simple path from r to u.

$\rightarrow$ v is u's ancestor.

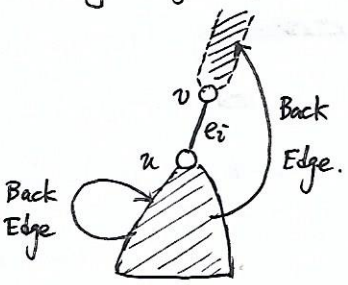
(2) At k-th step. $v \in$ CATA.III Impossible. (since u has an earlier b.s.)

(3) $v = r$. Obvious v is u's ancestor.

(4) v: still unvisited. $\longrightarrow$ Forward Step (k-th). Contradiction Impossible.

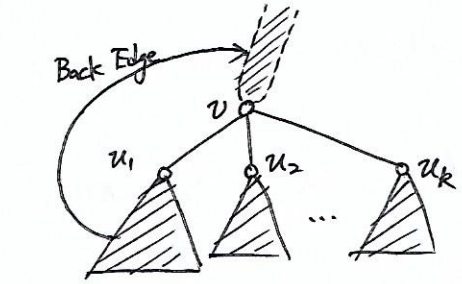


$\rightarrow$ A easier way to find a cut edge.



$\rightarrow$ Then e_i is not a cut edge.
Othermise, e_i is a cut edge.

$\rightarrow$ A easier way to find a cut vertex.



(One is enough for this subtree)

$\rightarrow$ If all k subtrees has (at least) one back edge from a vertex in it to v or v's ancestors

$\Rightarrow$ not a cut vertex

If one of its subtrees has no back edges between it and v and v's ancestors. $\longrightarrow$ a cut vertex

- Breadth-First Search.

  Too easy to take notes.

  Queue.

- Minimum spanning tree

Def. A weighted undirected graph. is .....

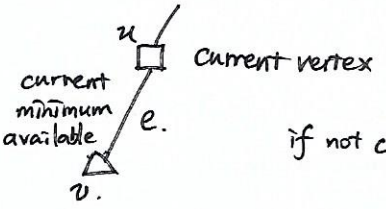
Minimum Spanning Tree

Prim's Algorithm

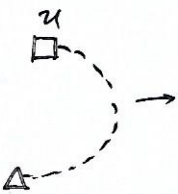
1. Pick an initial vertex $=: (V_0, E_0) = T_0$
2. Choose the $e_i \in E \setminus E_{i-1}$, which has the least weight and connects a $v \in V \setminus V_{i-1}$ with a $u \in V_{i-1}$. (which obviously not forming a circuit).

   Until: $V_n = V$. i.e. $n = |V| - 1$.
3. T_n is the minimum spanning tree. $(n = |V| - 1)$

[Proof]. T_n is with the minimum weight.



if not choose e,



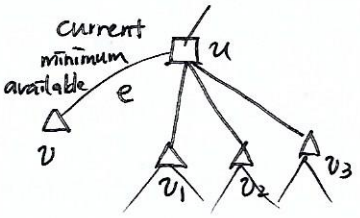
Another spanning tree

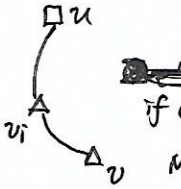
if $e' = e$ → MST, too.

Difference: an edge e'.

→ $e' < e$

→ Contradiction.





Difference: an edge e'.

→ $e' < e$ → Contradiction.

if $e' = e$ → MST, too.

[Prim's Correction]

It is suffice to prove: For any n, there exists G's minimum spanning tree T, s.t. T_n is T's subgraph.

BASE STEP. $n = 0$. Obvious

INDUCTION HYPOTHESIS: exists minimum spanning tree T s.t. T_k is T's subgraph.

To prove: exists minimum. spanning tree T_1 s.t. T_{k+1} is T_1's subgraph.

CASE 1. $e_{k+1} \in T$. Obvious.

CASE 2. $e_{k+1} \notin T$. Suppose e_{k+1} is from u to v.

find the $(k+1)$th least weighted edge in T. Suppose it is e^*.

$w_{e^*} = w_{e_{k+1}}$. (Otherwise, we won't pick e_{k+1} in the first place.

$w_{e^*} < w_{e_{k+1}}$ ↗

$w_{e^*} > w_{e_{k+1}}$ → Not a M.S.T.

replace e^* with an e_{k+1}

No circuit → "minimum available"

↓

Contradiction.

Delete e^*. Add e_{k+1}.

Still a MST. → T^*

And T_{k+1} is T^*'s subgraph.

Kruskal's Algorithm

1. Initialization $T_0 = \emptyset =: \{V_0, E_0\}$. ($V_0 = V = V_1 = \cdots = V_n$. $E_0 = \emptyset$)
2. Every time, choose the least weighed edge which does not form a circuit, add it to T_i. Suppose the edge is e_{i+1}. $T_{i+1} = T_i \cup \{e_{i+1}\}$.
   - Until $n = |V| - 1$.
3. T_n is a MST of G.

[Proof] [Kruskal's Correctness]

T_n is for sure a tree. ($n-1$ edges, no circuit $\rightarrow$ a tree)

BASE STEP. T_0. is a subgraph of T.

INDUCTION HYPOTHESIS. T_k is T's subgraph.

CASE. 1. T_{k+1} is T's subgraph. Obvious.

CASE 2. T_{k+1} is not. $T = (V, E)$

Suppose e_{k+1} is from u to v.

A simple path p of T connects u and v.

At least an e in p is not included in T_{k+1}. Otherwise, circuit.

Consider $T^* = (V, E^*)$. $E^* = E \setminus \{e\} \cup \{e_{k+1}\}$. ($|e| = |e_{k+1}|$)

T^* is a MST. T_{k+1} is T^*'s subgraph.

- Huffman Coding

Letters appearing more should receive shorter encodings $\rightarrow$ Less storage space

e.g.				
	"f"	00	0	90%
	"b"	10	10	5%
	"a"	01	110	3%
	"$"	11	111	2%
Storage Space		"2"	"1.15"	

Prefix Codes: an encoding of letters (a map from alphabet/letters to 0,1 strings)

any

For two different letters a, b. $f(a)$ and $f(b)$ are not prefix of each other.

Decoding:

0) $S = x_i$

1) Is S a complete code?

   N $S + \{x_{i+1}\}$; $i \rightarrow i+1$

   Y 2)

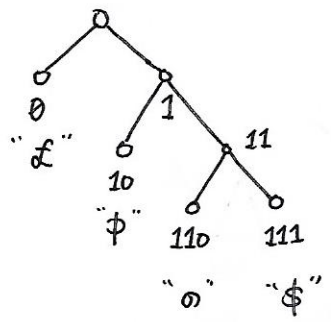
2) Decode. $\rightarrow$ 0)

Using prefix codes.

Can prove the process.

and the uniqueness of decoding solution.

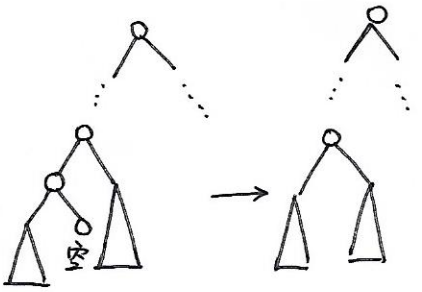
Decoding Process ⟷



A prefix code ⟷ A Tree

A Leaf corresponds a letter

or is empty (ϕ)



* A leaf is empty → can find a way to decrease average coding length. ↑

Huffman Coding Algorithm (哈夫曼编码)

letter c_i ~ frequency w_i.

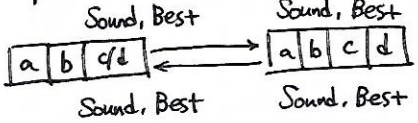
1) Construct a forest consisting of individual trees with only one vertex without edges.
2) Sort the trees from maximum weight to minimum weight
3) Choose the two trees with the least two weights: T_1, T_2.
4) Merge. Create a fresh root, make its left subtree T_1, right subtree T_2.
   name left branch "0", right branch "1".

The end of a step. Consider the merged trees a vertex, i.e. the fresh root.
Back to 1).

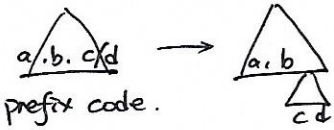
[e.g. The next page].

[Proof] Need to prove "Merge" does not destroy "the best solution".

i.e. Sound, Best ⟷ Sound, Best



→: Obvious:

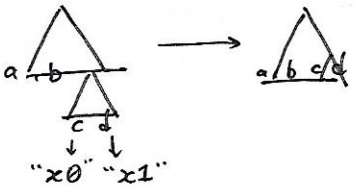


a is not "c/d"'s prefix
→ a is not "c"s/"d"'s prefix
"c/d" is not "a"'s prefix
→ "c"/"d" is not "a"'s prefix

Total code length $\mathcal{L}$ — Best

constant $\mathcal{L} + 1 \cdot w(c/d)$ — Best, still

←:



a is not b's prefix.
b is not a's prefix.

if a is "c/d"'s prefix → a is c's prefix. ✗
if "c/d" is a's prefix.

a → "x0y" ("y" can be empty)
"x1y"

then c is a's prefix or d is a's prefix. ✗

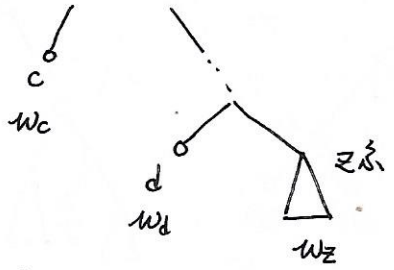
1) If in the Best Solution of "$\overset{a\ b}{\underset{c\ d}{\triangle}}$". $l_c = l_d$.

"c": 1110
"d": 1010 → 1111
"交": 1111 → 1010

} → c/d 111.

$\mathcal{L}$ stays the same.

2) If $l_c \neq l_d$. Let $l_c < l_d$.



Obviously $w_c < w_d$ (they're the least two)

If we switch "c" with "z-series"

$\mathcal{L}_{org} = \mathcal{L}_{other} + l_d w_d + l_c w_c + l_z w_z$

$\mathcal{L}_{now} = \mathcal{L}_{other} + l_d w_d + l_c w_z + l_z w_c$

$l_c < l_z$. $w_z > w_c \Rightarrow \mathcal{L}_{org} > \mathcal{L}_{now}$.

Contradiction.

$\Rightarrow$ Best holds.

~~Euler Circuit & Euler Path~~

[Formal Proof]

(其实前面 proof 已经基本上很正式了)

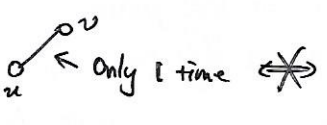
- Euler Circuit, Euler Path

Def. An Euler Circuit is a simple circuit containing every edge of G.
An Euler Path is a simple path containing every edge of G.
(G can be directed or undirected.)

* Connectivity between u and v

But!

Euler Circuit/Path



← Can be used "twice" (every direction one time)

Thm. In an undirected graph. G,

(Any vertex can be both starting point and termination)

G has an Eular Circuit $\iff$ Degrees of all vertices is even.

G has an Eular Path $\iff$ Exactly two vertices have odd degrees.

Also, the two vertices are the starting point and termination of the path respectively.

$\Leftarrow$:

[Proof] By Induction. ($\Rightarrow$: Obvious) (Prove all graphs ..... have E.C./E.P.).

BASE STEP. One edge: 

Two edges:  .

INDUCTION STEP. G has at least one simple circuit.

Find a "longest" simple circuit (move on to another vertex through an unvisited edge until degree of starting point "drops" to 0. in the graph with visited edges removed.)

- If it's a. E.C. ✓
- If it is not a Euler Circuit.

By Induction Hypothesis, the subgraph of G generated by removing the circuit must have an Euler circuit from u to u. in every connected components.

Connect the circuits and we get an Euler Circuit.

Euler Path $\longrightarrow$ If odd-degree vertices are u, v.

Add an edge: $\{uv\}$. $\longrightarrow$ Have an Euler Circuit.

Remove $\{uv\} \longrightarrow$ Get a Euler Path.

- Hamilton Circuit, Hamilton Path

Def. A *Hamilton Path* is a simple path: $v_0, v_1, \cdots v_n$ s.t. $v_i \neq v_j$ $(i \neq j)$ and $V = \{v_1, \cdots v_n\}$.

A *Hamilton Circuit* is a simple circuit: $v_0, v_1, \cdots v_n, v_0$ s.t. $v_i \neq v_j$ $(i \neq j)$ and $v_0, v_1, \cdots v_n$ is a Hamilton Path.

[Prop.] Graph with denser edges is more likely to have a Hamilton Path/Circuit.

Thm. Ore's Theorem

$G = (V, E)$ is a simple graph with $|V| \geq 3$.

If for any $u \neq v \in V$. either an edge $e \in E$ connects u and v or $\deg(u) + \deg(v) \geq |V|$.

G has a Hamilton Circuit.

[Proof] By Contradiction. Suppose G has no Hamilton Circuits.

(1) Construct a simple graph $H \supseteq G$ by adding edges to G while ensuring that H is Hamilton-Circuit-free.

Then H is the "maximal Hamilton-Circuit-free" simple graph with $|V| = |V_H|$.

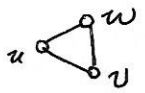
(2) Pick u and v which are distinct and has no edge connecting them.

Then Add $\{uv\}$ into H and we'll get a Hamilton Circuit. (Def. of H).

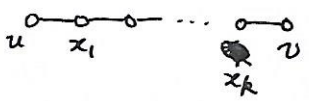
Therefore, exists a Hamilton Path from u to v.

(3) Since in G those u, v not connected by any edges. $\deg(u)+\deg(v) \geq |V|$.

in H. $\deg_H(u)+\deg_H(v) \geq |V|$.

CASE 01. $|V|=3$.  Obvious. (If $\deg(u)+\deg(v)=3$. Not a simple graph)

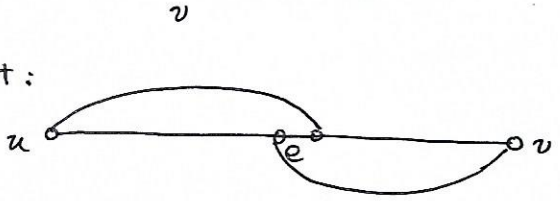
CASE 02. $|V| \geq 4$

 Hamilton Path. $k=|V|-2$.

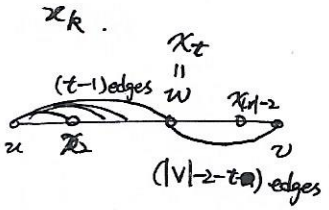
Since $\deg(u)+\deg(v)=|V|$. Other $|V|-2$ edges.

A simple path. u has no other edge incident with u and x_1.

If not:



Then have $\leq |V|-3$ edges. Contradiction.



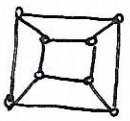
Total: $|V|-3$ edges.

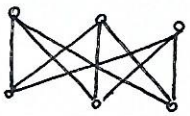
- Planar Graph (A not-that-rigorous glimpse of planar graphs)

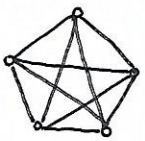
Def. $G=(V,E)$ is an undirected graph.

G is a *planar graph* if it can be drawn in the plane without any edge crossing, where an edge crossing is an intersection point of two edges other than their endpoints.

e.g. K_4 is a planar graph.  because 

Q_3 is a planar graph.  because 

$K_{3,3}$ is not a planar graph. 

K_5 is not a planar graph. 

Def. Region: A planar graph split the plane into regions through its edges and vertices (according to a specific drawing.) A not-that-rigorous definition.

Thm. Euler's Formula

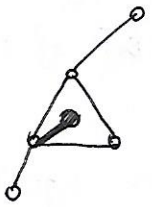
$r = |E| - |V| + 2$ if G is a connected planar graph. (r is the number of regions)

[Proof] ~~BASE STEP~~. (A not rigorous one)

By Induction on $|E|$.

BASE STEP. $|V|=1$. $|E|=0$. $r=1$. Obvious.

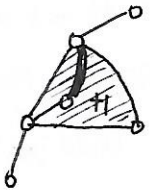
INDUCTION STEP. (1) A new vertex added. (2) No new vertex added.



$|V| = |V_{org}| + 1$

$|E| = |E_{org}| + 1$

$r = r_{org}$



$|V| = |V_{org}|$

$|E| = |E_{org}| + 1$

$r = r_{org} + 1$

In the new graph, Euler's formula still holds.

Note. In a disconnected graph : k connected components

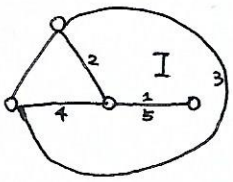
$$r = |E| - |V| + k + 1$$

Def. *Degree of Regions*

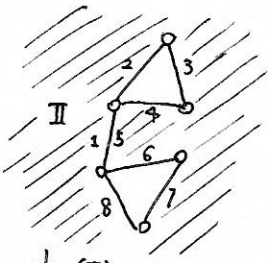
The degree of a region is the number of edges on its boundaries.

(If one's both sides is the same region, the edge need to be counted twice.)

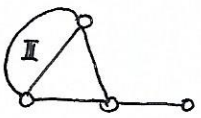
e.g.



$\deg(I) = 5$.



$\deg(II) = 8$.



$\deg(III) = 2$

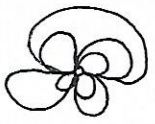
* 对比 I 和 III.

Thm. $\sum_{\text{region } R} \deg(R) = 2|E|$. (an analogy to $\sum \deg(v) = 2|E|$.)

Corollary. If $G = (V, E)$ is a simple connected planar graph. $|V| \geq 3$.

then $|E| \leq 3|V| - 6$.

[Pf.] Each region in G has a degree at least 3.



← If not simple

($\deg(R) = 1$. × ∞ Loop

$\deg(R) = 2$ ∞ Not simple.)

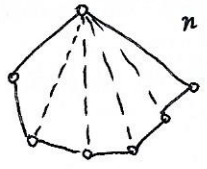
$$2|E| = \sum_R \deg(R) \geq \sum_R 3 = 3r$$

By Euler's Formula, $r = |E| - |V| + 2$

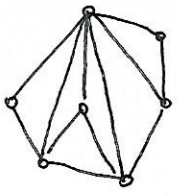
Therefore, $\frac{2}{3}|E| \geq |E| - |V| + 2 \Rightarrow \frac{1}{3}|E| \leq |V| - 2$.

Qed.

[Understanding]

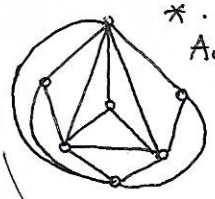


$n + (n-3) \sim \underline{2n}$



Add a vertex

+3 edges $\sim \underline{3n}$.



*. Add a vertex outside

Corollary. Any connected simple planar graph has at least 1 vertex with degree less than 6.

Corollary. K_5 is not a planar graph. [Pf.] $|E|=10. > 3|V|-6=9.$

[Pf.] $\sum_{\text{region } R} \deg(R) = 2|E| \leq 6|V| - 12.$ Thus, ... Qed.

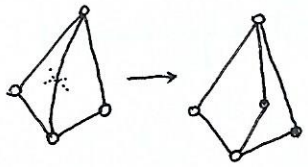
Corollary. In a connected simple planar graph $G=(V,E)$. if every simple circuit has length at least 4. then $|E| \leq 2|V|-4$.

[Pf.] $2\cdot|E| = \sum_{\text{region } R} \deg(R) \geq 4r.$ (A region ~ A circuit)

$\frac{|E|}{2} \geq r = |E|-|V|+2.$ Thus, $\frac{|E|}{2} \leq |V|-2$ Qed.

Corollary. $K_{3,3}$ is not a planar graph.

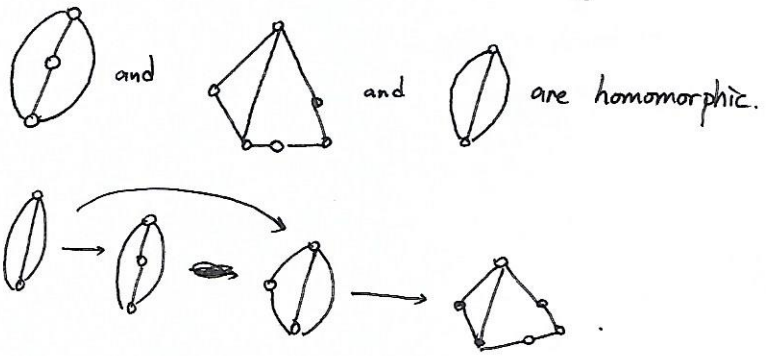
Def. *Elementary Subdivision*



remove an edge "$u-v$"

add a vertex "w" add two edges "$u-w$" "$v-w$"

Def. G_1 and G_2 are *homomorphic* iff there exists G, both G_1 and G_2 can be obtained by a sequence of Elementary Subdivision from G. (E.S can be empty)



and and are homomorphic.

Thm. Kuratowski's Theorem

A simple graph is nonplanar iff. it has a subgraph homomorphic to $K_{3,3}$ or K_5.

[A glimpse into Pf.]

A Planar graph's subgraph is a planar graph.

A graph homomorphic to a planar graph is a planar graph.

How to construct a way making it homomorphic to $K_{3,3}$ or K_5. No need to know now.

[Application]. K_6 is not a planar graph. (K_5 — subgraph !)

Algorithms:

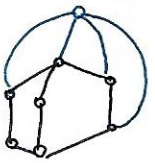
- Hopcroft - Tarjan *Algorithm*.

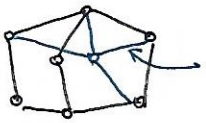
  反正不记得多个 Tarjan 算法

  Complexity: $O(n)$.

- Another Easier Algorithm (But Not-that-efficient)

Easy to draw a circuit in a plane.



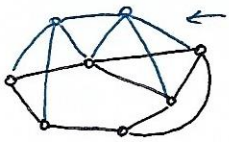


Added vertex & edges (as a whole thing) must be in the same region.
(Consider the vertex's region. Otherwise, not a planar representation)

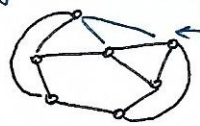
Def. Suppose $G=(V,E)$ is an undirected graph. $G'=(V',E')$ is a subgraph of G
If $e_1, e_2 \in E \setminus E'$ shares at least one endpoint $v \in V \setminus V'$, then e_1 and e_2 are edge connected w.r.t (with respect to) G'.

关于

Edge-connected components: an equivalent class of reflexive transitive closure of edge connection.

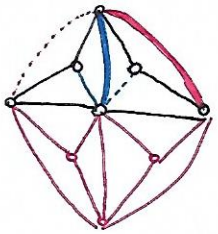


← Blue: Edge-connected component.



not edge-connected !

endpoints in V.
included in an edge-connected component
≥
on boundaries of exactly one region.



← 6 edge-connected components

Then: an edge-connected component must be in the exactly one region (now).

Algorithm. Suppose $G=(V,E)$ is a simple graph with no cut edge.

STEP 1. Pick a simple circuit. embed it into a plane. Let the result $P_{i_0}=(V_{P_{i_0}}, E_{P_{i_0}})$

STEP 2. For every region R of P_i and for every edge-connected component B separated by P_i. test whether edges in B are only incident with $V \setminus V_{P_i}$ and R's boundary vertices.

- If no. ~~B fail to be embedded~~. It's impossible to embed B.
  return "Nonplanar".
- If B can only be embed into exactly one region. → STEP 3 (B,R)
- If all B ~~has~~ have more than 1 choices. Pick an arbitary B.
  Pick a region R where B can be embeded → STEP 3 (B,R)

STEP 3. (B,R) Pick a simple path of B connecting two vertices of R.
embed the edges and vertices in the path into P_i. Get a P_{i+1}.

STEP 4. $i \to i+1$. Back to STEP 2.

No "backwards" → Why?

两种方案都有得选时，才会出现要随机的情况. → 否则，唯一方案必先选. 不会错过解.

每个 e.c.c. → 完全对称.

选的方案多的时候 → 少的那个最终会只有唯一方案

COMBINATORICS 组合数学 组合数学

1. Number of permutations of $\{1,2,\cdots,n\}$: $n! = n\cdot(n-1)\cdot\cdots\cdot 2\cdot 1$.
排列

2. Number of choices of choosing m objects from $\{1,2,\cdots,n\}$

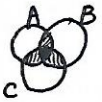
$(1<m<n)$ $C_n^m = \frac{n!}{m!(n-m)!} = \binom{n}{m}$.

$(m=0 \text{ or } n)$ $\left\{\begin{array}{l} 1 \\ 0 \end{array}\right.$ $\binom{m>n \text{ or}}{m<0}$

$(a+b)^n = \sum_{i=0}^{n}\binom{i}{n} a^i b^{n-i}$

$\sum_{i=0}^{n} C_n^i = 2^n$. $\sum_{i=0}^{n}(-1)^i C_n^i = 0$.

3. Inclusion-exclusion 容斥原理



$n=2$: $|A\cup B| = |A| + |B| - |A\cap B|$

$n=3$: $|A\cup B\cup C| = |A|+|B|+|C| - |A\cap B| - |B\cap C| - |C\cap A| + |A\cap B\cap C|$

交集元素个数比并集元素个数好算.

Thm. If $A_1, A_2, \cdots A_n$ are infinite sets, then

$$\left|\bigcup_{i=1}^{n} A_i\right| = \sum_{\ell=1}^{n} \sum_{1\le k_1<k_2<\cdots<k_\ell\le n} (-1)^{\ell+1} \left|\bigcap_{j=1}^{\ell} A_{k_j}\right|.$$

[To understand]

How many sets you're going to apply "union".

i.e. $|A_1\cup A_2\cup\cdots\cup A_n| = |A_1|+\cdots+|A_n|$
$-|A_1\cap A_2| - \cdots - |A_1\cap A_n| - |A_2\cap A_3| - \cdots$
$-|A_{n-1}\cap A_n| + |A_1\cap A_2\cap A_3| + \cdots + |A_{n-2}\cap A_{n-1}\cap A_n|$
$+\cdots+(-1)^{n+1}|A_1\cap A_2\cap\cdots\cap A_{n+1}|$

[Proof] LHS = Left Hand Side $= \sum_{u\in\bigcup_{i=1}^{n} A_i} 1$.

$$RHS = \sum_{u\in\bigcup_{i=1}^{n} A_i} \sum_{\ell=1}^{n} \sum_{1\le k_1<\cdots<k_\ell\le n} (-1)^{\ell+1} f_{(k_1,\cdots k_\ell)}(u)$$

$$f_{(k_1,\cdots k_\ell)}(u) = \begin{cases} 0, & \text{if } u\notin \bigcap_{j=1}^{\ell} A_{k_j} \\ 1, & \text{if } u\in \bigcap_{j=1}^{\ell} A_{k_j}. \end{cases}$$

Suffice to prove $1 = \sum_{\ell=1}^{n}\sum_{1\le k_1<\cdots<k_\ell\le n} f_{(k_1,\cdots k_\ell)}(u) =: rhs$ (for any $u\in\bigcup_{i=1}^{n} A_i$)

Suppose u appears in exactly m "A_i"s, i.e. $A_{b_1}, A_{b_2}, \cdots A_{b_m}$. $(b_1<b_2<\cdots<b_m \le n)$

$f(k_1,\cdots k_\ell)(u)=1$ iff. $\{b_1, b_2, \cdots b_m\}\supseteq\{k_1,\cdots k_\ell\}$.

Thus, $rhs = \sum_{\ell=1}^{n}(-1)^{\ell+1} C_m^\ell = \sum_{\ell=1}^{m}(-1)^{\ell+1} C_m^\ell = 1 - \sum_{\ell=0}^{m}(-1)^\ell C_m^\ell = 1$. Qed.

4. Disrangement 错排问题

How many bijections $f: \{1,2,\cdots,n\}\to\{1,2,\cdots,n\}$ satisfies $\forall m.\ f(m)\neq m$.

[Solution] Consider the remaining part. $A = \{f \mid \exists m.\ f(m)=m\} = \bigcup_{i=1}^{n}\{f \mid f(i)=i\}$
$\{f \mid f(i)=i\} = A_i$

$|A| = n(n-1)! - C_n^2(n-2)! + \cdots + (-1)^{n+1} C_n^n (n-n)!$

(since for any ℓ. $1\le\ell\le n$. $\left|\bigcap_{j=1}^{\ell} A_{k_j}\right| = C_n^\ell (n-\ell)!$)

Better if you write the formal form of I-E.

Thus, there are $\left(n! - \sum_{i=1}^{n} (-1)^{i+1} C_n^i (n-i)!\right)$ bijections.

5. How many onto functions (surjections) $f: \{1,2,\cdots,m\} \to \{1,2,\cdots n\}$?

[Solution]. Consider the remaining part: $A = \{f \mid \exists i \in \{1,2,\cdots n\}\ \forall j \in \{1,2,\cdots,m\}\ f(j) \neq i\}$

$$= \bigcup_{i=1}^{n} \underbrace{\{f \mid \forall j \in \{1,2,\cdots m\}.\ f(j) \neq i\}}_{\text{!!}\ A_i}$$

Since for any l, $1 \le l \le n$.

$$\left|\bigcap_{i=1}^{l} A_{k_i}\right| = (n-l)^m$$

Thus, there are $\left(n^m - |A| = n^m - \sum_{l=1}^{n} C_n^l (n-l)^m (-1)^{l+1}\right)$ surjections.

$$= \sum_{l=0}^{n} C_n^l (n-l)^m (-1)^l$$

When $m<n$? $=0$.

6. How many numbers from $\{0,1,\cdots m-1\}$ are relatively prime to m? (written as $\varphi(m)$)

Suppose $m = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_n^{\alpha_n}$. where p_i are distinct prime numbers, $\alpha_1, \cdots \alpha_n$ are positive.

[Solution] Consider the remaining part $A = \{k \mid 0 \le k \le m-1,\ k \in \mathbb{N},\ (k,m) > 1.\}$

$$= \bigcup_{i=1}^{n} \left\{k \,\middle|\, \begin{array}{l} 0 \le k \le m-1,\ k \in \mathbb{N}, \\ p_i \mid (k,m) \end{array}\right\}. = \bigcup_{i=1}^{n} \underbrace{\left\{k \,\middle|\, \begin{array}{l} 0 \le k \le m-1,\ k \in \mathbb{N} \\ p_i \mid k \end{array}\right\}}_{\text{!!}\ A_i}$$

Since for any l, $1 \le l \le n$.

$$\left|\bigcap_{i=1}^{l} A_{k_i}\right| = \left\lfloor \frac{m-1}{p_{k_1} p_{k_2} \cdots p_{k_l}} \right\rfloor + 1 = \frac{m}{p_{k_1} p_{k_2} \cdots p_{k_l}}$$

Thus, $|A| = \sum_{l=1}^{n} (-1)^{l+1} \sum_{1 \le k_1 < \cdots k_l \le n} \frac{m}{\prod_{j=1}^{l} p_{k_j}}$. # of numbers is $m - \sum_{l=1}^{n} (-1)^{l+1} \sum_{1 \le b_1 < \cdots < b_l \le n} \frac{m}{\prod_{j=1}^{l} p_{b_j}}$

[Make $m = \sum_{\text{Pick 0 bs}} \frac{m}{\prod_{j=1}^{l} p_{b_j}}$:= "1"

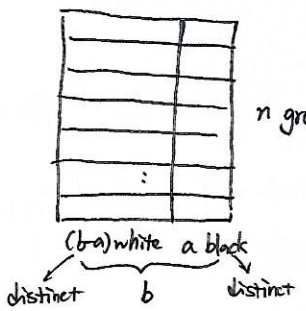
$$= m\left(1 - \frac{1}{p_1} - \frac{1}{p_2} - \cdots - \frac{1}{p_n} + \frac{1}{p_1 p_2} + \cdots\right) = m \cdot \prod_{i=1}^{n} \left(1 - \frac{1}{p_i}\right).$$

$$= m \sum_{l=0}^{n} (-1)^l \sum_{1 \le b_1 < \cdots < b_l \le n} \frac{1}{\prod_{j=1}^{l} p_{b_j}}$$

(Since $1 - \frac{1}{p_1} - \frac{1}{p_2} + \frac{1}{p_1 p_2} = (1 - \frac{1}{p_1})(1 - \frac{1}{p_2})$, ...)

[Note]. $(0,0)$. → 不讨论. $(0,m) = m$. $(\forall m)$. $(m,m) = m$.

7. An explanation / A proof of Binomial Expansion (二项式展开)



Pick one ball from each group. All white: $(b-a)^n$.

Consider the remaining part. → At least one black.

Thus. $(b-a)^n = b^n - \sum_{l=1}^{n} C_n^l a^l b^{n-l} (-1)^{l+1} = \sum_{l=0}^{n} (-1)^l a^l b^{n-l}$