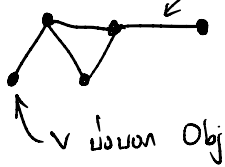
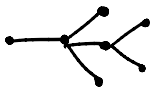
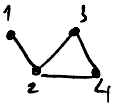
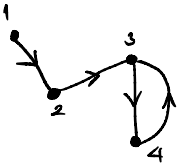
GRAPH

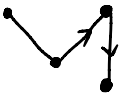
Vertex — Edge เช่น



,  , II : 

Undirected graph

- Directed graph : มีลักษณะของความสัมพันธ์ เช่น I : 

, 
  - I : $R = \{(1,2),(2,3),(3,4),(4,3)\}$
  - II : $R = \{\{1,2\},\{2,3\},\{3,4\},\{2,4\}\}$

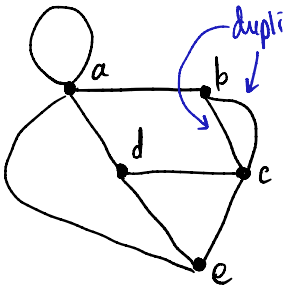
เช่น Social Graph : Social Relation , Six Degree of Freedom แสดง connection

Graphs & Trees

def $G = (V, E)$; $V = \{v_1, v_2, v_3, \ldots\}$ → set of vertices

$E = \{e_1, e_2, e_3, \ldots\}$ → set of edges ; $e_n = (v_i, v_j)$ หรือ $\{v_i, v_j\}$

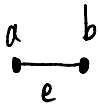
(Directed: Ordered pairs ; Undirected: mset)

eg. 

$V = \{a, b, c, d, e\}$

$E = \{\{a,a\},\{a,b\},\{b,c\},\{b,c\},\{c,d\},\{a,d\},\{c,e\},\{d,e\},\{a,e\}\}$

∴ G has 5 vertices , 9 edges

- terms 

  - e is incident with a, b
  - a is end point of e / a,b are end points of e
  - a is adjacent to b

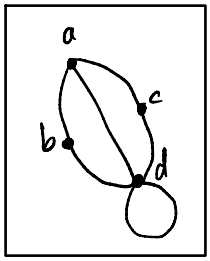
- Undirected Graph
  - Simple G. : No duplicate edges (unique) , No loops
  - Multigraph : No loops
  - Pseudograph : Any

- Directed Graph
  - Simple Directed G. : No duplicates ( , ), No loops
  - Directed Multigraph : Any , with all edges directed
  - Mixed G. : Any

Degree of a Vertex "Number of edges incident with v", $\deg(v)$

↓

* a loop has 2 degrees. → Number of times a vertex is an endpoint.



$\deg(a) = 3$
$\deg(b) = \deg(c) = 2$
$\deg(d) = 5$

Handshaking Theorem : Undirected graph $G(V,E)$ with e edges.

$$2e = \sum_{v \in V} \deg(v)$$

eg. 6 deg/v, 10 v : $e = \frac{6 \times 10}{2} = 30$ edges.

→ **Proof** In undirected graph, there must be an even # of V. with odd edges

Using proof by contradiction, there must be an odd # of v. with odd edges;

Let # of edges $c = 2n+1$, # of vertices $p = 2m+1$

The sum of degrees is

$$\sum_{i=1}^{p} (2c_i + 1) = 2\sum_{i=1}^{p} c_i + p \quad (2m+1)$$

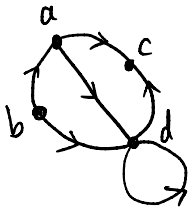
$$= \text{odd number } 2k+1$$

By HS Thm, The sum of degrees must be even; hence contradicts our assumption.

Therefore, the statement is correct. ∎

→ Directed G. Degree ┬ Out-degree ($\deg^+(v)$)
└ In-degree ($\deg^-(v)$)

e.g.

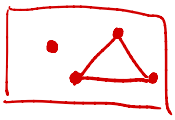


$\deg^+(a) = 2$, $\deg^-(a) = 1$
$\deg^+(b) = 2$, $\deg^-(b) = 0$
⋮

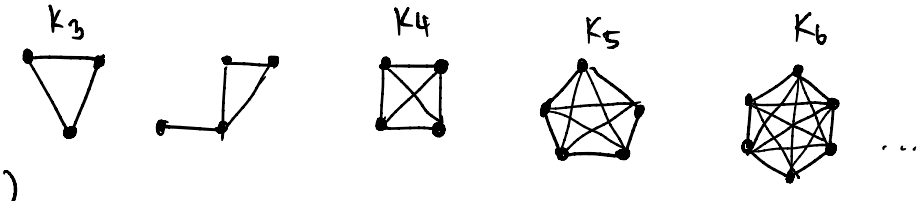
Handshaking Theorem (Directed Graph ver.)

$$\sum_{v \in V} \deg^-(v) = \sum_{v \in V} \deg^+(v) = |E|$$

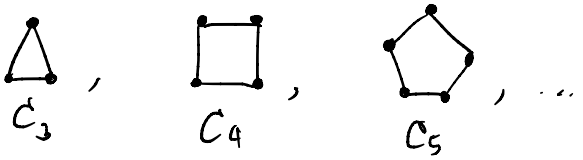
Special Simple Graphs

→ Incomplete Graphs :  1 is incomplete.

- Complete Graphs (K_n): For every $v \in V, |V| > 1$, v must have an edge.

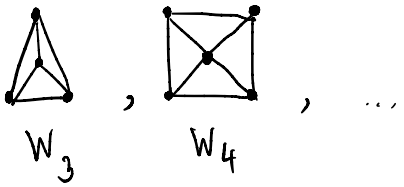


- Cycles (C_n)



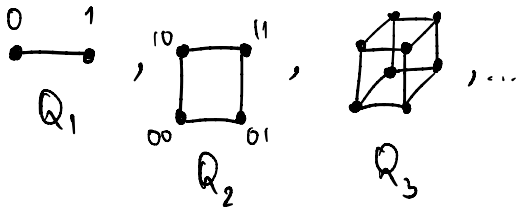
C_n: $|V| = n$, $|E| = n$

- Wheels (W_n)



W_n: $|V| = n+1$, $|E| = 2n$

- n-Dimensional Hypercube (Q_n)

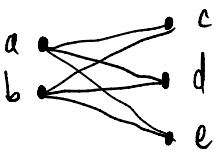


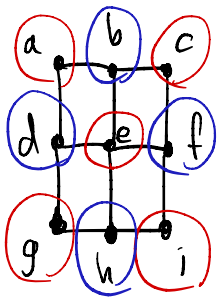
Q_n: $|V| = 2^n$, $|E| = \frac{n \cdot 2^n}{2}$

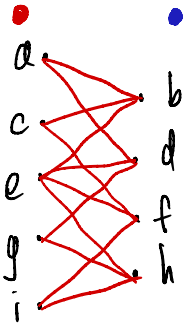
Example : Boolean Cube.

- Bipartite Graph : V can be partitioned to V_1, V_2 w/ no edges connects vertices within the same partition

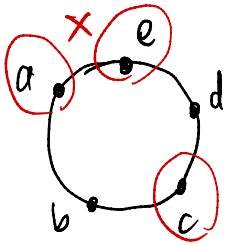
eg.



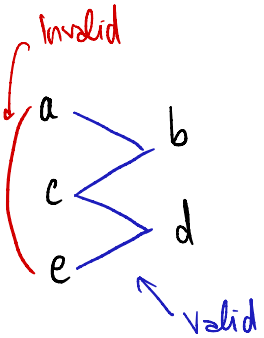




ไม่ Bipartite

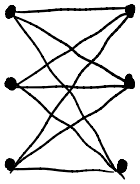


วัฏจักร C ไม่เป็น Bipartite

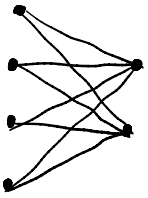


- Complete Bipartite Graph ($K_{m,n}$)

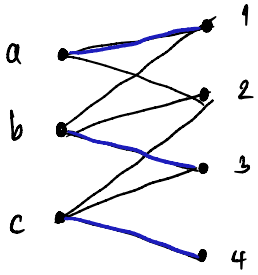
$K_{3,3}$



$K_{4,2}$



→ Complete Matching Problem

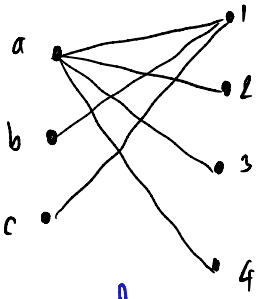


(a,1), (b,3), (c,4)

→ Hall's Marriage Theorem

ให้ $A = \{a,b\}$

$N(A) = \{1,2,3\}$

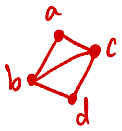


?

$A = \{b,c\}$

$N(A) = \{1\}$

" มี Complete Match ใน $G=(V,E)$, (V_1,V_2) ก็ต่อ $V_1 \to V_2 \iff |N(A)| \geq |A|$ $(\forall A \in V_1)$ "



$V = \{a,b,c,d,e\} : W \subseteq V$
$E = \{ab, ac, bc, bd, cd\} : F \subseteq E$
} Any combination.
$H = (W, F)$

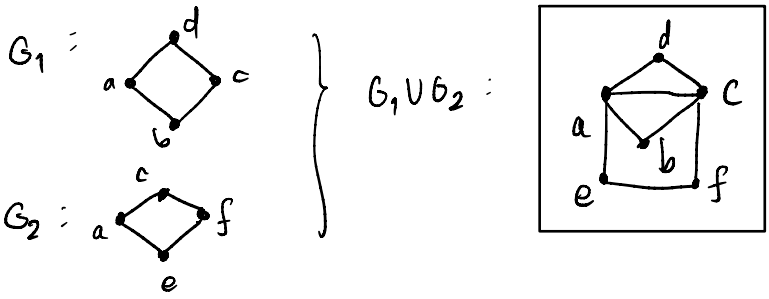
Subgraphs

Given $G = (V, E)$, there is a subgraph $H = (W, F)$; $W \subseteq V$, $F \subseteq E$
โดยที่ H ต้องเป็น graph ด้วย. *

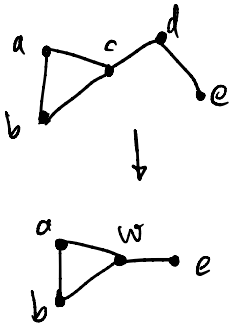
If $H \neq G$, then H is a proper subgraph.

Unions

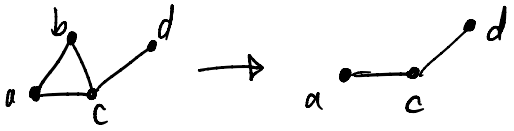
$G(V, E) = G_1(V_1, E_1) \cup G_2(V_2, E_2)$; $V = V_1 \cup V_2$, $E = E_1 \cup E_2$



Edge Contraction

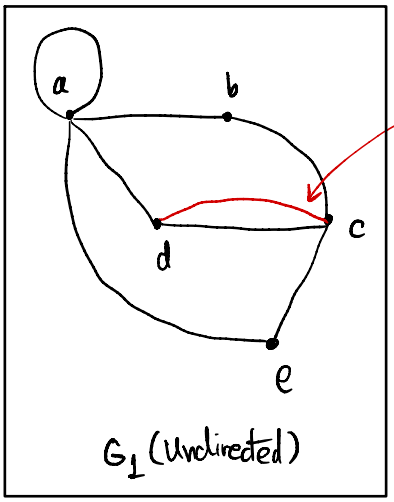


Vertex Removal



Graph Representation

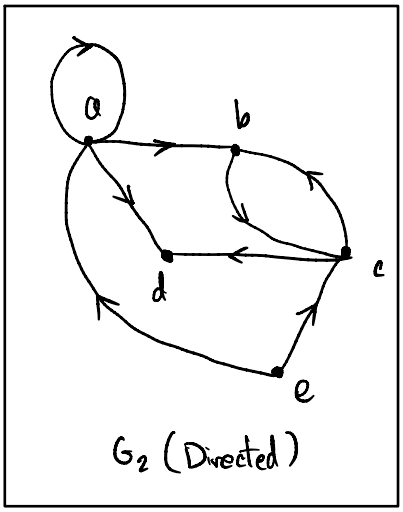
1. Adjacency Lists



G_1 (Undirected)

ข้อสังเกต : ใช้กับ multiple edges ไม่ได้

V	V_{adj}
a	a b d e
b	a c
c	b d e
d	a c
e	a c



G_2 (Directed)

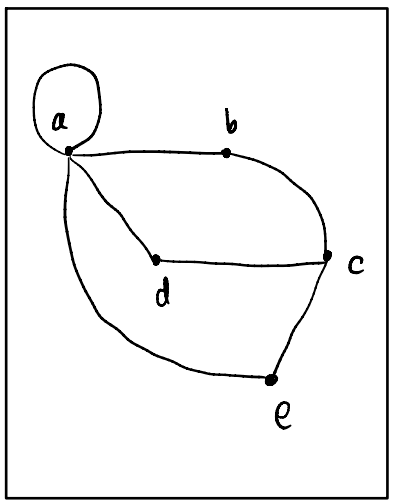
V_i	V_f (terminal)
a	a b d
b	c
c	b d
d	-
e	a c

2. Adjacency Matrices

$G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$: Let $A = [a_{ij}]$ จะได้ว่า $A_{n \times n}$

For undirected graph, a_{ij} = Number of edges corresponding to $\{v_i, v_j\}$

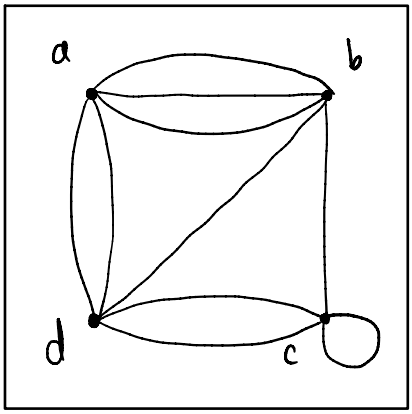
($a_{ij} = a_{ji}$)



ตัวอย่าง

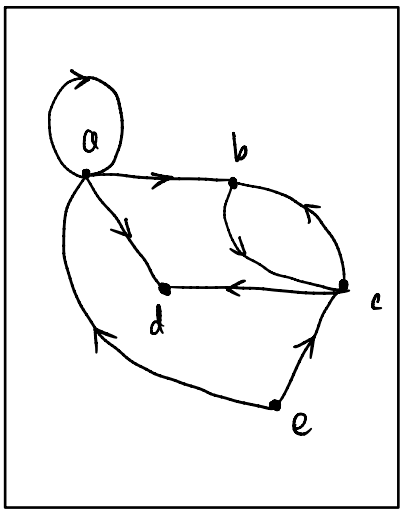
$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array} = A^T$$

(Symmetric สำหรับ undirected graphs)



$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix} \end{array}$$

For directed graphs, a_{ij} = Number of edges corresponding to $(\vec{v_i}, \vec{v_j})$



$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

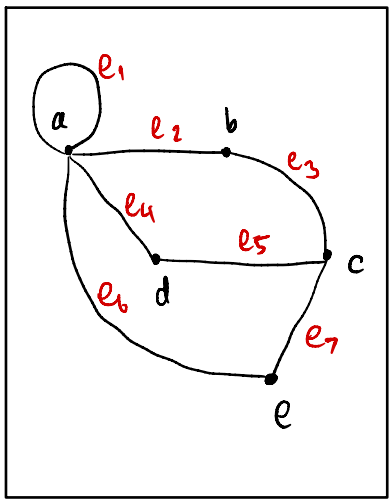
(b → a), (a → b)

3. Incidence Matrices

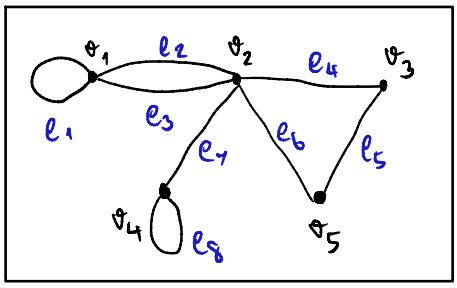
(unoriented) $M_{n \times m}$

$G = (V, E)$, $V = \{v_1, v_2, \ldots, v_n\}$, $E = \{e_1, e_2, \ldots, e_m\}$. Let $M = [m_{ij}]$

For undirected graphs, $m_{ij} = \begin{cases} 1 & \text{when } e_j \text{ is incident with } v_i \\ 0 & \text{elsewhere.} \end{cases}$



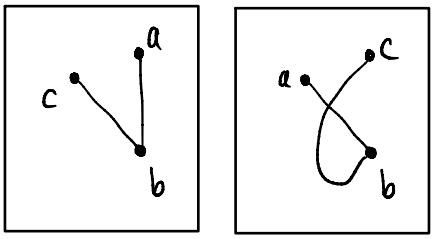
$$M = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$



$$M = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Isomorphism (สมสัณฐาน)

1.

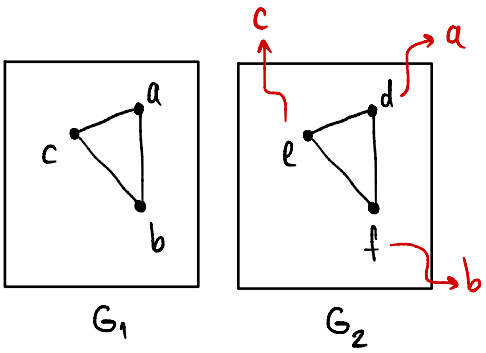


$G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$

ถ้าได้ G_1 และ G_2 เป็นกราฟอันเดียวกัน.

$$(G_1 = G_2) \leftrightarrow [(V_1 = V_2) \wedge (E_1 = E_2)]$$

2.



G_1 และ G_2 isomorphic กัน (Same shape, different names)

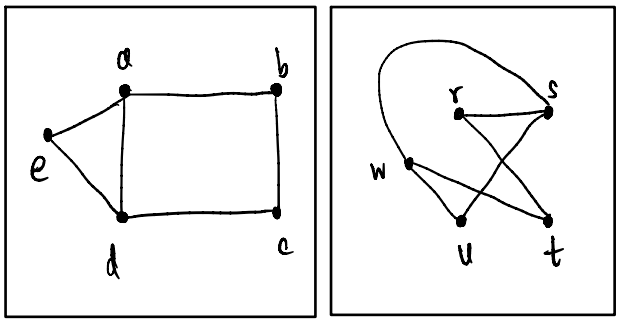
* เปลี่ยนชื่อตัวแปร 1-1 map (f_{1-1})

* Let $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ be simple graphs. G_1 and G_2 are isomorphic when there is a bijective function $f: V_1 \to V_2$ such that $v_1 \mapsto v_2$

$\forall_{a,b \in V_1}(a,b)$ a and b are adjacent in $G_1 \leftrightarrow$ f(a) and f(b) are adj. in G_2. "

* f is "isomorphism"

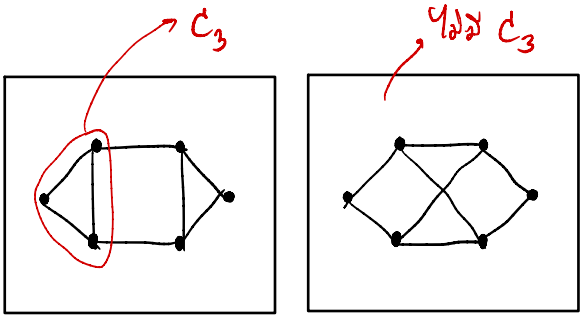
e.g.



Isomorphism f:

$s = f(a)$, $t = f(c)$, $u = f(e)$,

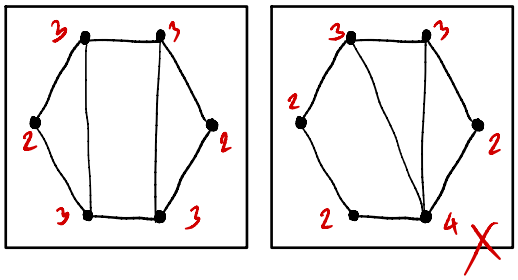
$r = f(b)$, $w = f(d)$



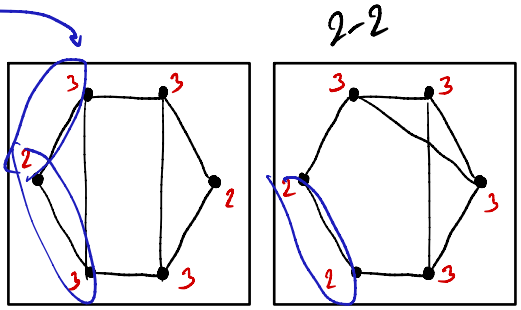
ไม่ isomorphic เพราะ ...

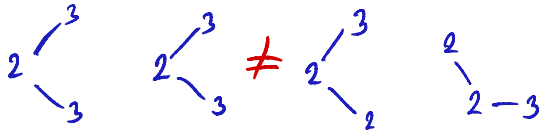
ใช้ Graph Invariants (สิ่งที่ไม่เปลี่ยน ถ้ากราฟ isomorphic กัน)

e.g. $|V|$, $|E|$, $\deg(v)$, Adjacency ที่เขียนแสดง: deg, ...



ไม่ isomorphic

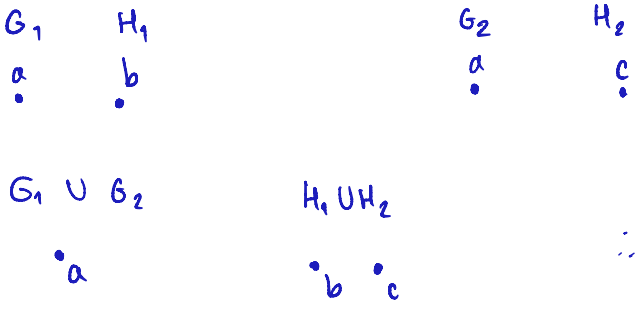




ไม่ isomorphic

e.g. G_1, H_1 : Isomorphic, G_2, H_2 : Isomorphic ; P/DP that $G_1 \cup G_2$ และ $H_1 \cup H_2$ are isomorphic

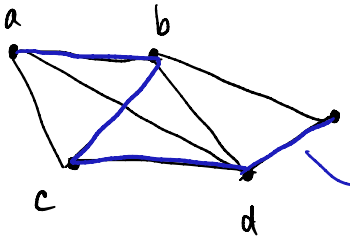
Counterexample :



$\therefore$ ไม่ Isomorphic.

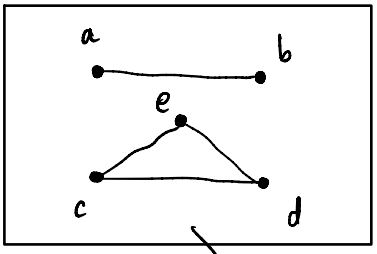
Graph Connectivity

- Path
- Length

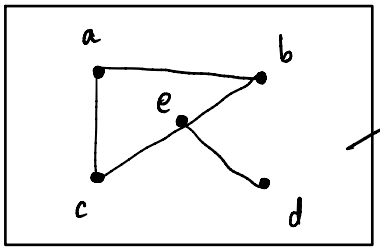


Path $(\{a,b\}, \{b,c\}, \{c,d\}, \{d,e\})$
หรือไม่มี multiple edges เขียนแค่
(a,b,c,d,e) ก็ได้

- **Simple Path** : edge ไม่ซ้ำกันทั้งหมด e.g. (a,b,c,b,c,d,e) เป็น non-simple
  (a,b,c,d,e) เป็น simple.
- **Circuit** : vertex เริ่ม = vertex จบ. eg. (b,c,d,b)
- **Connectedness** : A connected undirected graph $\leftrightarrow$ There is a path between every $(v_i, v_{j\neq i})$
  $(\exists(\text{path})\ \forall(v_i, v_{j\neq i}))$

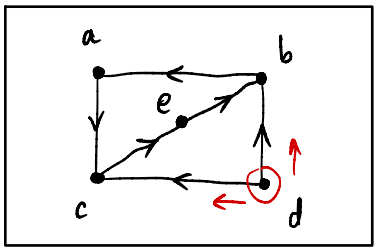


disconnected

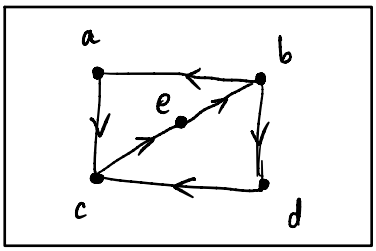


connected

- Directed graph is strongly connected $\leftrightarrow$ there is a path from v_i to $v_{j\neq i}$
- Directed graph is weakly connected $\leftrightarrow$ there is a path between v_i to $v_{j\neq i}$

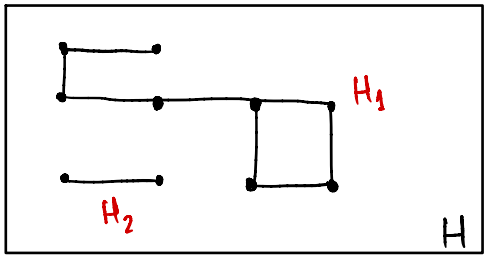


Weakly connected.



Strongly connected.

- **Connected component** : "Maximal connected subgraph of a graph" ("ขยายต่อไปไม่ได้แล้ว" (ชั้นๆ))



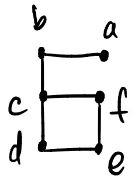
e.g. How many?

	a	b	c	d	e	f	g	h	i	j
a	X	1		1	1			1		1
b	1	X			1					1
c			X			1	1			
d	1			X	1					1
e	1	1		1	X			1		1
f			1			X	1			
g			1			1	X			
h	1				1			X		1
i									X	
j	1	1		1	1			1		X

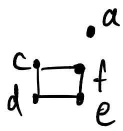
3 C.C.

- Cut Vertex, Cut Edge → เอาออกแล้วทำให้เกิด Connected component เพิ่ม.
  (Articulation point) (Bridge)

e.g. G:

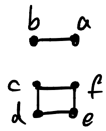


Cut b :



} cut vertices มี b, c.

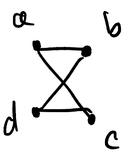
Cut {b,c}



} Cut Edges มี {b,c}, {b,a}.

- Path → นำมาตรวจสอบ Isomorphism ได้.

- Counting Paths :



Length 4 : (a,b,a,b,d), (a,b,a,c,d), ...
$a \to d$

↳ เขียน Adjacency matrix A. หา Length ยาว r : → A^r
ต้องการทราบ $v_i \to v_j$ ให้ดู $(a^r)_{ij}$ มีค่าเท่าไร เป็นคำตอบ.

(• Matrix Diagonalization : $A = PDP^{-1}$)

+ Diagonalization (opt.)

$$A=\begin{bmatrix} 2 & 0 & 0 \\ 1 & 2 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

λ: $\det(A-\lambda I)=0$

$$\begin{vmatrix} 2-\lambda & 0 & 0 \\ 1 & 2-\lambda & 1 \\ -1 & 0 & 1-\lambda \end{vmatrix} = 0$$

$$(2-\lambda)\begin{vmatrix} 2-\lambda & 1 \\ 0 & 1-\lambda \end{vmatrix} = 0$$

$$(2-\lambda)^2(1-\lambda)=0 \rightarrow \lambda_1=2,\ \lambda_2=1$$

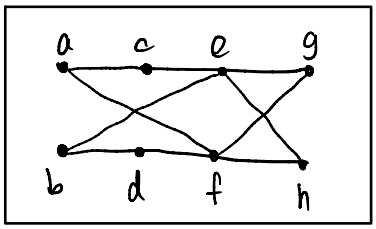
$m_1=2,\ m_2=1$ (multiplicity)

$\lambda_1=2$: $(A-\lambda I)\vec{v}=\vec{0}$

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ -1 & 0 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\therefore \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 \end{array}\right] = \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Euler Circuit คือ Simple circuit ที่ผ่านทุก edges ของ graph นั้น (edge ไม่ซ้ำ!)



มี Euler circuit เช่น

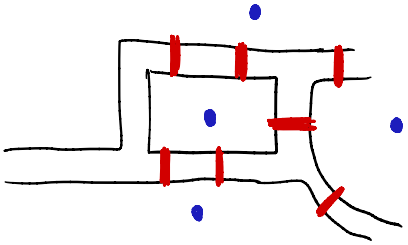
a c e g f h e b d f a

same circuits.

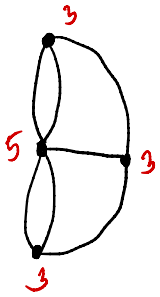
หรือ f h e b d f a c e g f

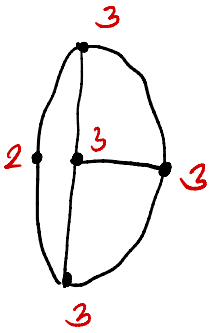
หรือ ...

Pb. Seven bridges of Königsberg



ข้ามครบทุกสะพานแบบไม่ซ้ำได้หรือไม่?

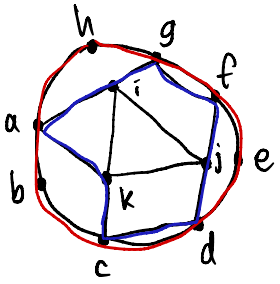




Thm. A connected multigraph with at least 2 vertices has an Euler Circuit
$\leftrightarrow$ each of its vertices has even degree.

(Both necessary and sufficient condition)

* Circuit ของสามารถสร้างจาก circuit ย่อย (subcircuit) ได้



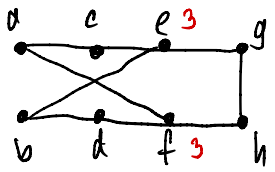
a b c d e f g h a

g i a k c d j f g

i k j i

จะได้ a b c d e f g i k j i a k c d j f g h a □

Euler Paths : Simple path containing every edges. (edge ไม่ซ้ำ!)



ebdfhgecaf

↳ conditions: connected multigraph has Euler path but not Euler circuit
↔ it has exactly 2 vertices with odd degree.
↳ ต้องเป็น เริ่ม, สิ้นสุด ของ Euler path.

Hamiltonian Path : Simple path ที่ผ่านครบทุก V เพียงครั้งเดียว, ไม่จำเป็นต้องผ่านทุก E

Hamiltonian Circuit : $v_1, v_2, \ldots, v_n, v_1$ เป็น H.Circuit ถ้า $v_1, \ldots, v_n$ เป็น H.Path.

↳ ไม่มี conditions แบบ necessary & sufficient

eg necessary : ถ้าเป็น HC ต้องไม่มี $\deg(v) < 2$ ใน graph.
(1) : Connected but no return.
(0) : Disconnected

e.g. necess : ต้องไม่มี Cut vertex / edge

e.g. $\deg(v) = 2$ ในกราฟ ต้องเป็นส่วนหนึ่งของ H. Circuit

□ Dirac's Theorem

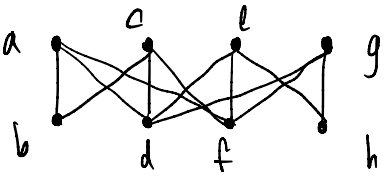
Sufficient : G เป็น H. Circuit if
G เป็น simple graph ที่มี $|V| \geq 3$ และ $\deg(v) \geq \frac{|V|}{2}$ (ทุก v)

□ Ore's Theorem

Sufficient G : simple graph $|V| \geq 3$ ที่ u, v ไม่ adjacent ถ้า
และ $\deg(u) + \deg(v) \geq |V|$ แล้ว G เป็น H. Circuit

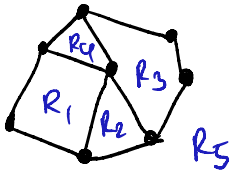
Planar Graphs : Graph ที่วาดบน plane แล้ว edge ไม่ตัดกันเลย.

Q_3 :



"Face"

Region



$F = 5$

$E = 11$

$V = 8$

Euler's formula : $|V| - |E| + |F| = 2$ ← Euler number

$k+1$ while k = # connected components

G: connected planar simple graph

- only if ($|V| \geq 3$, then $|E| \leq 3|V| - 6$) ($|E| \geq \frac{3|F|}{2}$)
- only if ($|V| \geq 3$ และ ไม่มี circuit length = 3, then $|E| \leq 2|V| - 4$)

↳ $K_{\geq 5}$ และ $K_{3,3}$ ไม่เป็น Planar Graph

* Planar graph มีคุณสมบัตินี้ แต่ ไม่ได้การันตีว่าถ้ามีคุณสมบัตินี้แล้ว planar *
(necessary condition)

$v - e + r = 2$

$v \geq 3$: $e \leq 3v - 6$

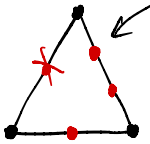
$e \leq 2v - 4$ (ไม่มี Δ)

Kuratowski's Theorem

A graph is nonplanar $\leftrightarrow$ it contains a subgraph homeomorphic to $K_{3,3}$ or K_5.

- Homeomorphism

"Smooth vertex"



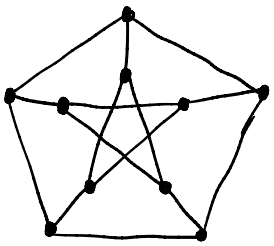
เพิ่ม/ลด vertex ที่ deg = 2 ไม่เปลี่ยนคุณสมบัติความเป็น planar

เพิ่ม เรียกว่า "elementary subdivision" ↔ inverse กัน

ลด เรียกว่า "smoothing" ~~edge contraction~~

∴ Homeomorphic คือการลด vertex deg=2 ออกแล้ว isomorphic กับ graph

e.g.



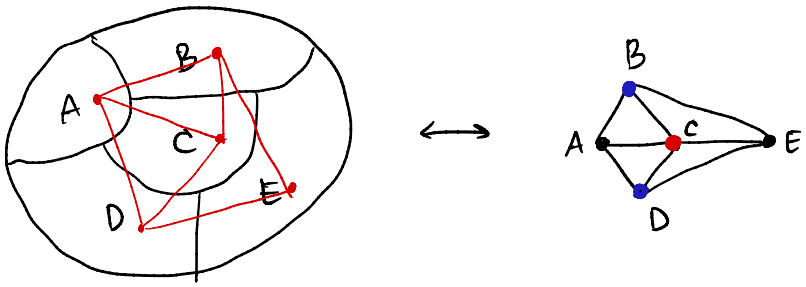
มี subgraph $K_{3,3}$

Graph Coloring

– Chromatic Number : จำนวนสีน้อยที่สุดที่ vertex ข้างเคียงกัน ไม่ใช้สีเดียวกัน

= 3

– Four Color Theorem : Simple Planar* Graph มี chromatic number ≤ 4 เสมอ



Chromatic# $K_n = n$ $\quad Q_n = 2$

$K_{m,n} = 2$

$C_n = 3$ if odd else 2

$W_n = 3$ if odd else 4

– 6-Color Theorem : Simple Planar Graph สามารถระบายได้ด้วย 6 สี.

6-color Proof

$S(n) \equiv$ "Planar Graph with n vertices can be colored using 6 colors."

Basis step : $S(6)$

Inductive step : $S(k) \rightarrow S(k+1)$

Lemma : G is a connected planar simple graph, then G has a vertex with deg ≤ 5

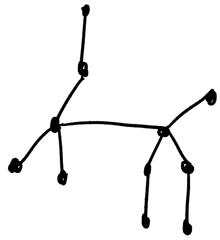
└ Contradiction proof : 모든 vertex 가 deg ≥ 6

(1): $2e \geq 6v$ → deg 합의 부등식 (Handshaking Thm.)

(2): $e \leq 3v-6$

→ $3v \leq e \leq 3v-6$: impossible

Trees
- Rooted Trees
- d-ary Trees (m-ary, k-ary)

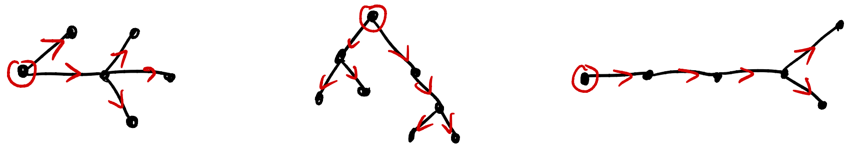


<u>Def.</u> Connected graph with no simple circuits

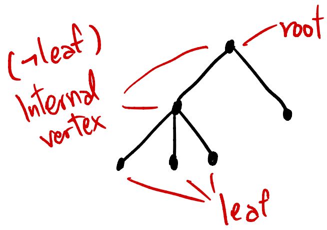
แต่ถ้าไม่จำเป็นต้อง connected → forest (หลาย trees)

<u>Thm.</u>
An undirected graph is a tree ↔ unique path between any two vertices

→ <u>Rooted Tree</u> → เป็น Directed Graph
- มี 1 vertex ที่เป็น "root" และ edge ทั้งหมดชี้ออกจาก root



  (เลือก vertex จาก undirected graph ใดมาเป็น root ก็ได้ ∵ ไม่มี circuit)
- root $deg^- = 0$
- โครงสร้าง Hierachy
  - parent, child, siblings
  - ancestor, descendant, successor



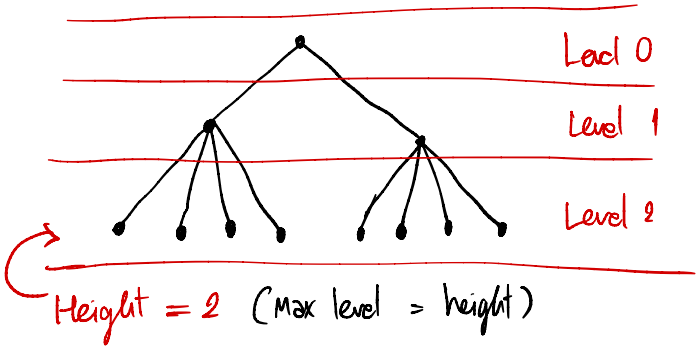
- Subtree : ส่วนหนึ่งจาก tree จะถือเป็น subtree (subtree เป็น tree)
- d-ary tree : <u>e.g.</u> binary tree (2-ary), quadtree (4-ary), octtree (8-ary)
  - <u>Full</u> d-ary tree : ทุก vertex ถ้ามี child ต้องมี d child

* <u>Tree</u> ที่มี n vertex มี n-1 edge เสมอ (Tree เป็น planar ใช้ euler's formula หา v, e | r=1)

* <u>Full d-ary tree</u> มี i internal vertices จะมี $n = di + 1$ vertices

(หา #leaf ได้ $n - i$

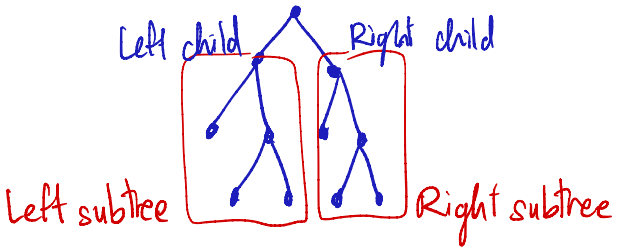
Tree Height (Depth) & Level



→ There are at most d^h leaves in d-ary tree of height h

→ d-ary tree is balanced if ALL leaves are at level h or h-1

Ordered Root Tree e.g. Binary Search Tree, Quadtree



e.g. Decision Tree, Prefix Code, Huffman Tree

Tree Traversal

- Pre order : อ่านตัวเองก่อน
- In order : อ่านลูกซ้าย, ตัวเอง, ลูกขวา
- Post order : อ่านตัวเองสุดท้าย

Spanning Tree

Def. Spanning tree of simple graph G is a subgraph of G that is a tree containing every vertex of G.

COMPUTATIONAL THEORY

Computation Structure

- Grammars → Sentence "Formal Language"
- Finite-state machines
- Turing Machine

Formal Language : ใช้ Definition ของ syntax : grammars

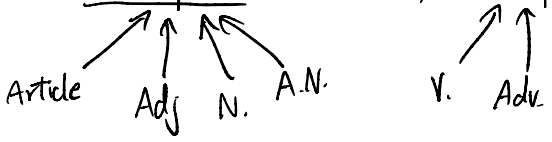
Finite sets

- Is it valid?
- How to make/generate a valid sentence?
- Provide models for both natural languages, programming languages

Grammars

G ของ L : ใช้ recursive generation

e.g. English (basic) : sentence ← Noun phrase followed by verb phrase.



Terminology

- Vocab (Alphabet) : $V = \{a, b, c, A, B, C, \ldots\}$
- Word, sentence : Any combinations from V.
- Empty (null) string : λ, String with no symbols. (String with zero length)
- V^* : A set of every combination of word.
- Language L over V : A subset of V^* $(L \subseteq V^*)$

Integer Power of a Language

Let $X = \{0, 1\}$

$X^0 = \{\lambda\}$, $X^1 = \{0, 1\}$, $X^2 = X \times X = \{00, 01, 10, 11\}$

$X^3 = X \times X^2 = \{0,1\} \times \{00, 01, 10, 11\} =$ ___.

$\therefore X^n$ = Bit strings of length n.

Kleene Star, Kleene Plus

$$X^* = \bigcup_{i \geq 0} X^i \quad \text{(รวม null string)} \qquad X^0 \cup X^1 \cup X^2 \cup X^3 \cup \ldots$$

$$X^+ = \bigcup_{i \geq 1} X^i = X^* - \{\lambda\} \quad \text{(ไม่รวม null string)}$$

e.g. $\Sigma = \{a, b\}$

$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$

$\Sigma^+ = \{a, b, aa, ab, \ldots\}$

Phrase-Structure Grammars

$G = (V, T, S, P)$

V : Symbols (elements)

$T \subseteq V$: Terminal elements.

$N = V - T$: Non-terminal elements.

$S \in N$: Start element.

P : Production $(w_0 \mapsto w_1)$ โดย w_0 ประกอบด้วยอย่างน้อย 1 $n \in N$

e.g. $V = \{a, b, A, B, S\}$,

$T = \{a, b\}$,

$S = S$,

$P = \{ S \mapsto Aba,$
$A \mapsto BB,$
$B \mapsto ab,$
$AB \mapsto b \}.$

▫ Derivation

$G = (V, T, S, P)$; $P = \{ Z_0 \to Z_1 \}$

Derivation

$w_0 = lz_0r$ ↓ $w_1 = lz_1r$ ↓ ⋮ ↓ $w_n = lz_nr$

} ใช้ 1 Production rule ใน 1 ครั้ง เรียกว่า "Directly derivable" : $(w_0 \Rightarrow w_1)$

: Derivable $(w_0 \Rightarrow^* w_n)$

e.g. $P: B \mapsto ab$, $A \mapsto abab$

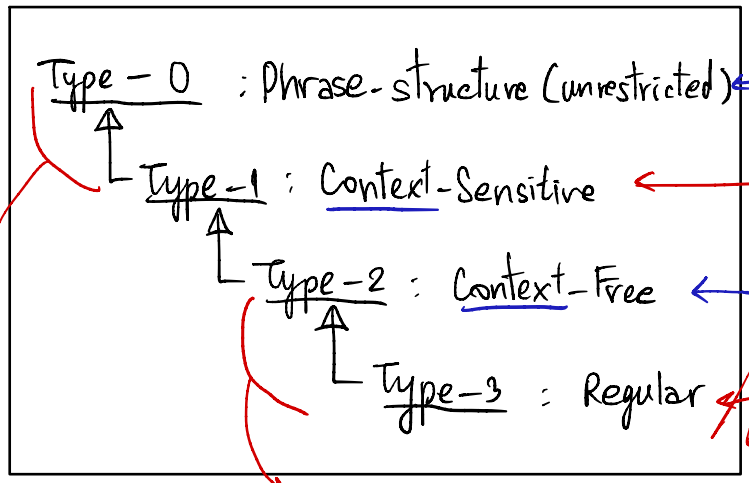
$\therefore ABa \Rightarrow Aaba$

$ABa \Rightarrow^* abababa$

↳ **Language** $L(G)$; $G = (V, T, S, P)$

$L(G) = \{ w : w \in T^*, S \Rightarrow^* w \}$

Types of Grammars (Chomsky hierachy)



"Context" = neighbor (element รอบข้าง)

Types of Grammar (formatted)

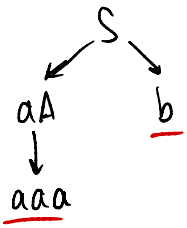
- **Type-0** : Unrestricted → Turing machine : $\gamma \to \alpha$
- [illegible] **Type-1** : Context-sensitive → Bounded turing machine : $\alpha A \beta \to \alpha \gamma \beta$ (α, β can't be empty at the same time)
  - or $S \to \lambda$
  - (เปลี่ยนได้ทีละตัว)
  - * α, β ตัวเดียวกัน
- [illegible] **Type-2** : Context-free → Push-down Automata : $A \to \alpha$
- [illegible] **Type-3** : Regular → DFA, NFA :
  - $A \to a$ or $A \to aB$ (Right-linear, Right-regular)
  - XOR
  - $A \to a$ or $A \to Ba$ (Left-linear, left-regular)

$\gamma \subseteq \Sigma^+$, α, β : Any $\subseteq \Sigma^*$, a : terminal, A, B : non-terminal

e.g. Regular : $S \to \lambda O$ → Non-terminal
$S \to wO$
$S \to T$
$S \to wT$

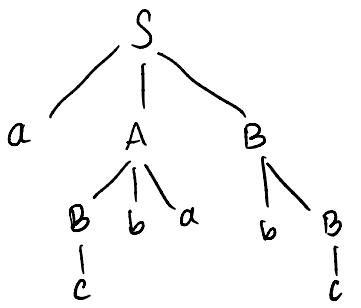
Context-free : $A \to w$

e.g. $G = (\{S, A, a, b\}, \{a, b\}, S, \{S \to aA, S \to b, A \to aa\})$



$\therefore L(G) = \{aaa, b\}$

e.g. $P = \{S \mapsto aAB, A \mapsto Bba, B \mapsto bB, B \mapsto c\}$



เป็นต้นไม้ของ word ที่ L สร้างได้ (acbabc)

□ Shorthand notation : $S \mapsto 11S$, $S \mapsto 0$ } $S \longmapsto 11S \mid 0$

Infinite language

e.g. $S \mapsto 11S \mid 0$

→ $S \Rightarrow 0$
$S \Rightarrow 11S \Rightarrow 110$
$S \Rightarrow 11S \Rightarrow 1111S \Rightarrow 11110$
⋮

} $L = \{(11)^*0\}$

e.g. $S \longmapsto aSb \mid \lambda$

$L = \{\lambda, ab, aabb, aaabbb, \ldots\}$

$L = \{a^n b^n \mid n \geq 0\}$

(Type-1)

Context-Sensitive

$L = \{ a^n b^n c^n \mid n \geq 1 \}$ เป็น CS แต่ไม่เป็น CF

↳ $P = \{ S \mapsto aSBC \mid aBC$

$CB \mapsto BC$

$aB \mapsto ab$

$bB \mapsto bb$

$bC \rightarrow bc$

$cC \rightarrow cc \}$

Parsing

Top-down : จาก $S \Rightarrow \ldots$

Bottom-up : จาก $\ldots \Rightarrow S$ (backward)

Regular Expressions

Let I be a set of input alphabet

ϕ เป็น Regex

$\{\lambda\}$ เป็น Regex

x เป็น Regex ; $\forall x \in I$

(AB), $(A+B)$, A^* เป็น Regex ถ้า A, B เป็น Regex

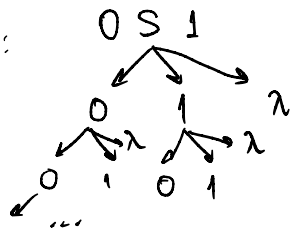
↓ ↓ ↓

Concat. Union Kleene star

10^* : 1, 10, 100, 1000, ...

$(10)^*$: λ, 10, 1010, 101010, ... $\neq (1+0)^*$: $\lambda, 0, 1, 00, 01, 10, 11, \ldots$

$0(0+1)^*1$: Any*(U*)



$= 01, 001, 011, 0001, 0011, 0101, 0111, \ldots$

$0+01$: 0, 01

$0(0+1)^* = 0 \cdot U^* = 0, 00, 01, 000, 001, 010, 011, \ldots$

$(0^*1)^*$: λ, 1, 01, 001, 0001, ...

11 0101 001001 00010001

$(00+01+10+11)^*$: even length

↳ $= ((0+1)(0+1))^*$

Finite-State Machines (FSM)

- Mealy*: State & Input → Output
- Moore: State → Output

} FSM: Has Output

- Finite-state Automata: No Output
  - Deterministic: Unique Transition
  - Non-deterministic: Can lead to >1 states

FSM: 1. Finite set of states: Starting state, input alphabet (Mealy only.)
2. Transition functions

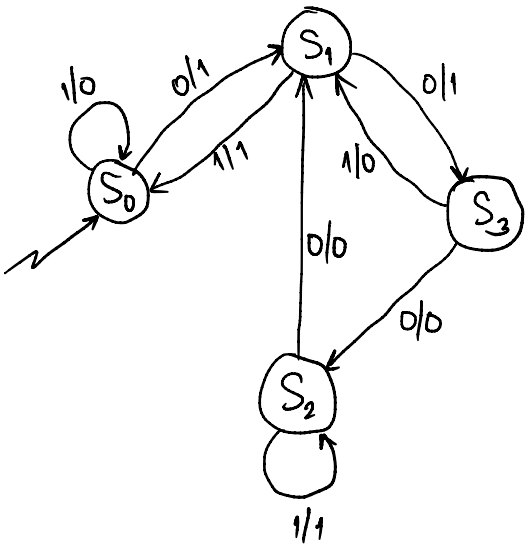
$M = (S, I, O, f, g, s_0)$

S: States
I: Input alphabet
O: Output alphabet
f: Transition function giving new state: $S^+ = f(S, I)$
g: Transition function giving an output: $O = g(S, I)$
s_0: Initial state (Default)

- Representation – STATE TABLE
  – STATE DIAGRAM

S	f		g	
	I=0	I=1	I=0	I=1
S_0	S_1	S_0	1	0
S_1	S_3	S_0	1	1
S_2	S_1	S_2	0	1
S_3	S_2	S_1	0	0

⇔



e.g. **Vending Machines**

$I = \{5, 10, 25, o, r\}$ — o: Orange button; r: Red button

$O = \{n, 5, 10, 15, 20, 25, OJ, AJ\}$ — n: not doing anything; OJ: orange juice; AJ: Apple juice

$S = \{S_0, S_1, S_2, S_3, S_4, S_5, S_6\}$; $S_0 = S_0$

0 5 10 15 20 25 30

State+ / Output

	State	I: 5	10	25	o	r
0	S_0	S_1/n	S_2/n	S_5/n	Do nothing. (S_k, n)	
5	S_1	S_2/n	S_3/n	S_6/n		
10	S_2	S_3/n	S_4/n	$S_6/5$		
15	S_3	S_4/n	S_5/n	$S_6/10$	S_3, n	S_3, n
20	S_4	S_5/n	S_6/n	$S_6/15$	S_4, n	S_4, n
25	S_5	S_6/n	$S_6/5$	$S_6/20$	S_5, n	S_5, n
30	S_6	$S_6/5$	$S_6/10$	$S_6/25$	S_6/OJ	S_6/AJ

e.g. **Binary Adder**

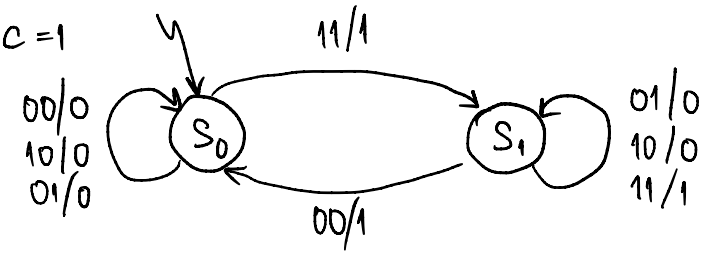
Input: $(x_n, \dots, x_0)_2$, $(y_n, \dots, y_0)_2$

Starting with $x_0 + y_0$, $x_1 + y_1$, ...

Ex $x_i + y_i + c_i$ — c_i: carry bit

S_0: $c = 0$

S_1: $c = 1$



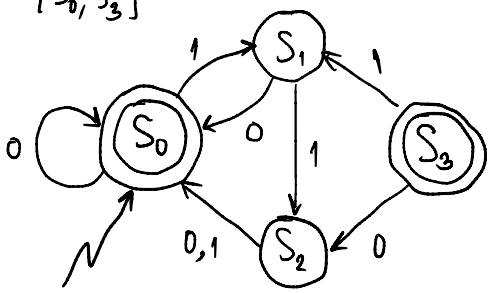
Finite State Automata : ไฟล์ output แต่ละ destinations (final states)
(Accepting states)

$M = (S, I, f, s_0, F)$; $F \subseteq S$ (F: Final/Accepting State)

e.g.

state	f, Input 0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_0
s_3	s_2	s_1

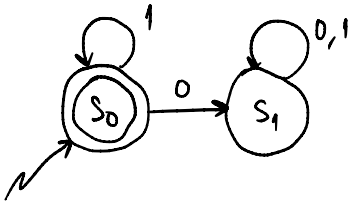
โดย $F = \{s_0, s_3\}$



Recognizing string using M : L(M)

ถ้า $L(M_1) \leftrightarrow L(M_2)$ แล้ว $M_1 \leftrightarrow M_2$

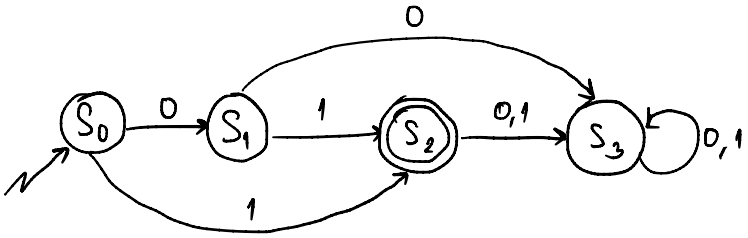
e.g.



λ ✓
0 ×
00 ×
01 ×
10 ×
11 ✓

1^* , $L(M_1) = \{1^n, n \in \mathbb{Z}_{\geq 0}\}$

e.g.

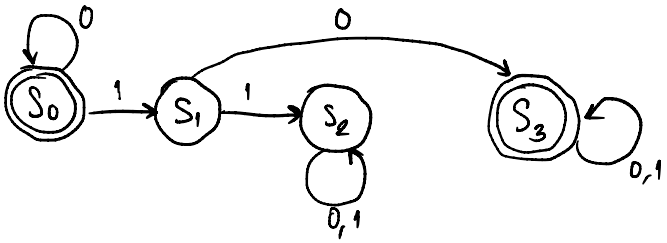


$s_0 \notin F$

~~λ~~ ×
1 ✓
01 ✓

$(1+01)$

e.g.



λ
0, 00, 000 (0*)
0*10(0+1)*

$0^* + 0^*10(0+1)^*$

$0^*(\lambda + 10(0+1)^*)$

Deterministic Finite-State Automata (DFA)

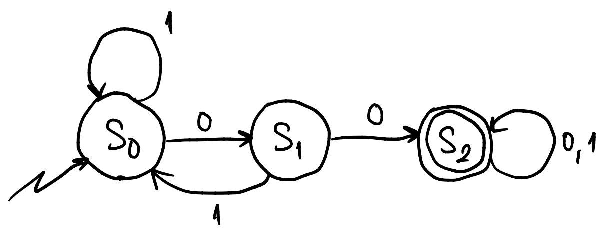
$M = (S, I, f, s_0, F)$

S: Finite set of states

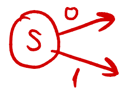
I: Finite input alphabets

$f: S \times I \rightarrow S$ (Mapping inputs to states yields next state)

e.g. สร้าง Deterministic FSA (DFA) ที่รับ set of bit strings containing two consecutive 0s.

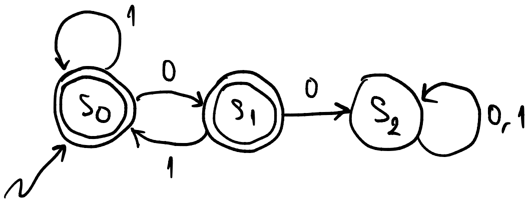


* ตอนสร้าง #ลูกศรออก state ต้องเท่ากับจำนวน Input เช่น $I = \{0, 1\}$

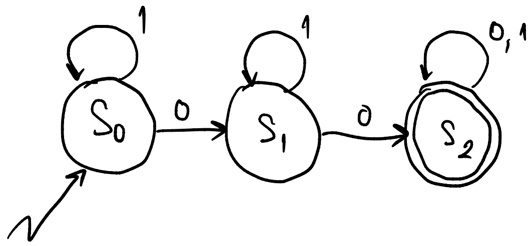


DETERMINISTIC

e.g. ตรงข้ามกับก่อนหน้า :

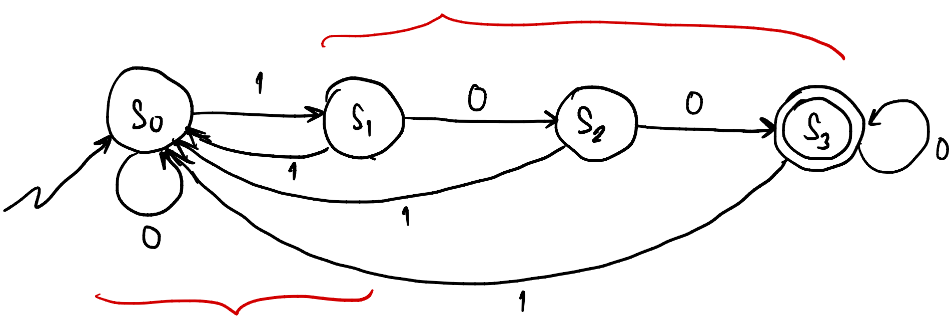


e.g. Set of bit strings containing at least two zeros.



e.g. Odd # of 1's AND end with at least two consec. 0's.

Check for last 2 0's



Parity check if odd.

Non-Deterministic Finite-State Automata (NFA) : สามารถแปลงเป็น DFA ได้เสมอ

$M = (S, I, f, S_0, F)$

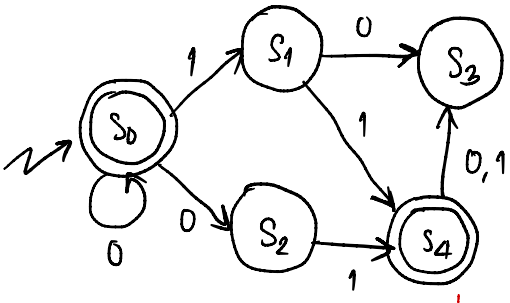
S: Finite set of states

I: Finite input alphabets

$f: S \times I \rightarrow P(S)$ Mapping inputs with states can yield any states at the same time. (including nothing)

e.g.

State	f(0)	f(1)
S_0	S_0, S_2	S_1
S_1	S_3	S_4
S_2	$\emptyset$	S_4
S_3	S_3	$\emptyset$
S_4	S_3	S_3



Reject immediately, ย้อนไปดู path อื่นที่ไม่ใช่ Final state. ("Sink state") → All

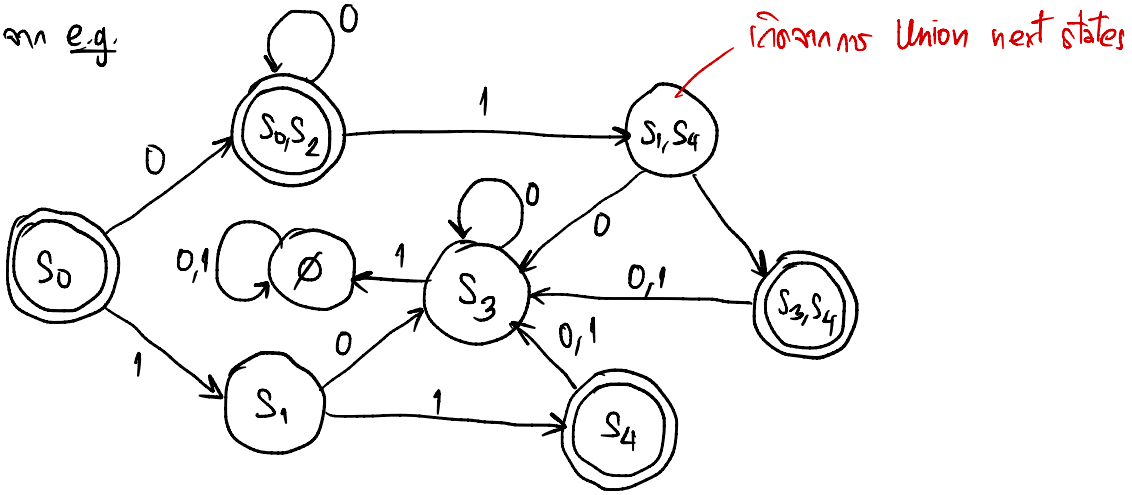
e.g. 01 : $S_0 \rightarrow S_0 \rightarrow S_2$ ✗

$S_0 \rightarrow S_2 \rightarrow S_4$ ✓

$\therefore 01 \in L(M)$

* NFA ต้องลองทุก possibility ด้วยการ backtracking เพื่อลองทุก path ไม่ลงเอยด้วย final state ถึงค่อย reject

NFA → DFA ตาม e.g.



Turing Machine

- Memory : Tape of Infinite length
- ใน Tape จะมีช่องให้ใส่ Alphabet (Read, Write ได้) "Square"
- Machine สามารถ เปลี่ยน Internal state, เปลี่ยน Symbol, Fwd, Back, Halt (Stop).

→ Turing's Thm. : → There are general-purpose Turing machine that can compute any other Turing Machine can do.

→ "Universal Turing Machine" ที่เป็น Machine ที่ทำงานได้หลายอย่าง.

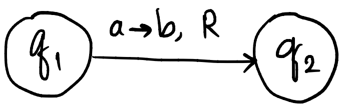
→ Church-Turing thesis (by Kleene) : whatever can be computed by mechanical procedure can be computed by a Turing Machine.

Def. $M = (S, I, f, s_0, B, T, F)$; $f : S \times I \to S \times I \times \{R, L\}$ (R = Right, L = Left)

B → Blank, T → Tape Alphabet

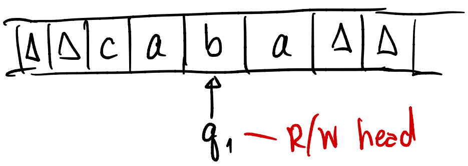
$M = (S, I, f, s_0, \Delta, T, F)$ — Δ: ใช้อะไรก็ได้ เช่น ␣, ◇, □, ...

e.g.



(Transition function)

e.g. Tape



เรียกว่าเป็น instantaneous description : caq_1ba

Moving (Transition) : $q_2xayb \succ xq_0ayb$: Move right, State $q_2 \to q_0$

$q_2xayb \succ xq_0ayb \succ xxq_1yb \succ xxyq_1b$

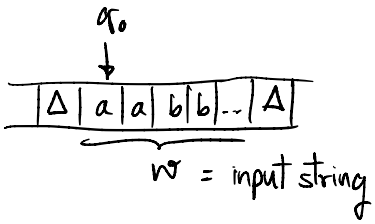
($a \to x$, $q_0 \to q_1$, MOV R) (MOV R)

Note สามารถเปลี่ยน Δ ได้ $\Delta \to \alpha$

Equiv. notation : $q_2xayb \succ^* xxyq_1b$

Initial configuration

Let q_0 be starting state : $q_0 w$

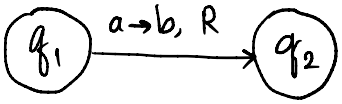


Accepted Language :

$$L(M) = \{ w : q_0 w \succ^* x_1 q_f x_2 \}$$

Standard Turing Machine : Deterministic, infinite length towards both direction, Tape is I/O.

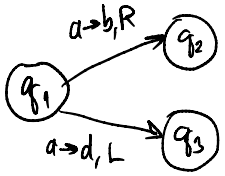
- ประกอบด้วย
  - Tape
  - Control Unit เป็น Automata
- Head
  1. Read
  2. Write
  3. Move
- States & Transitions



Notation : (q_1, a, q_2, b, R) — R = Mov Direction; q_1 = S., q_2 = N.S.

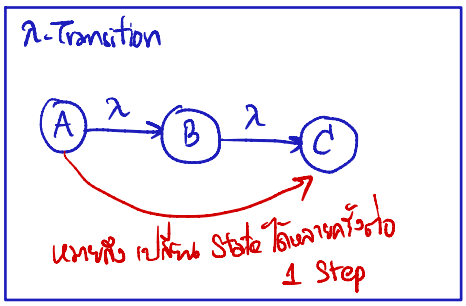
Determinism

↳ Deterministic TM :



ไม่ได้!

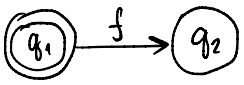
และ λ Transition ไม่ได้!



Halting

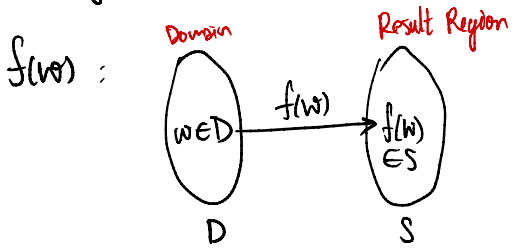
- Accept เมื่อ $q \in F$
- Reject เมื่อ $q \notin F$ หรือ อาจ Infinite loop
  - Halt & Reject เมื่อไม่มี transition ต่อไป.

** **Final States** ห้องไม่มี outbound transitions } เพื่อให้ไม่เกิด Infinite loop.
ต้องหยุดทันที.



ไม่ได้!

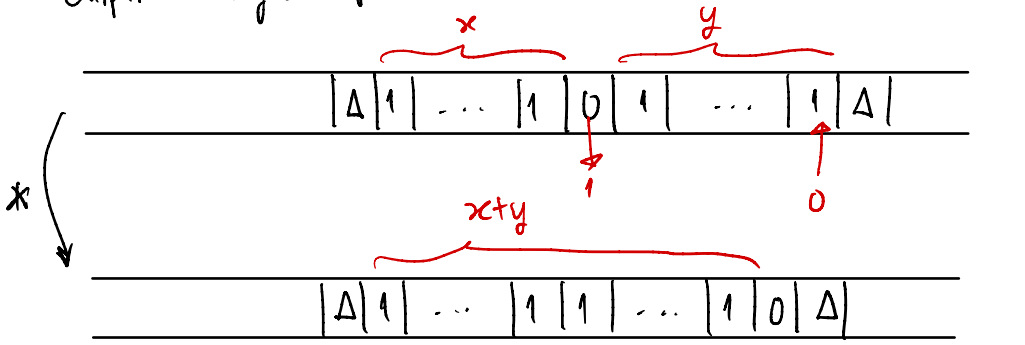
Computing Functions

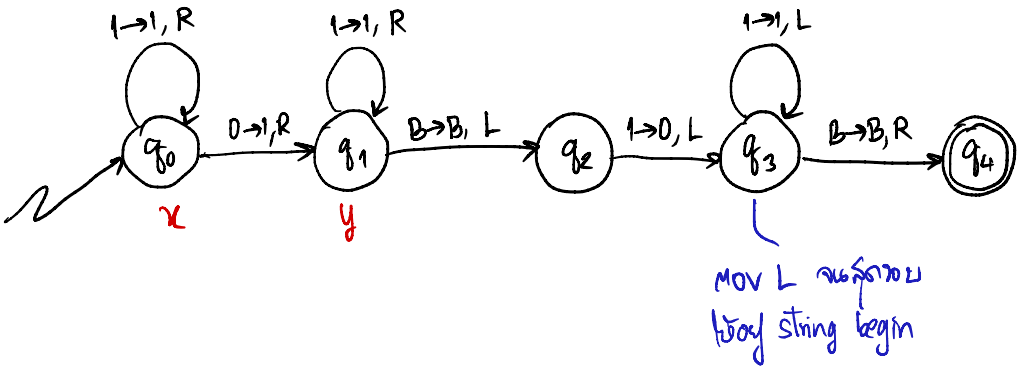


Def f is computable $\leftrightarrow$ TM M ที่ $q_0 w \succ^* q_f f(w)$

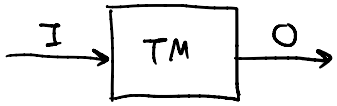
eg. $f(x,y) = x+y \quad ; \quad x, y \in \mathbb{Z}$

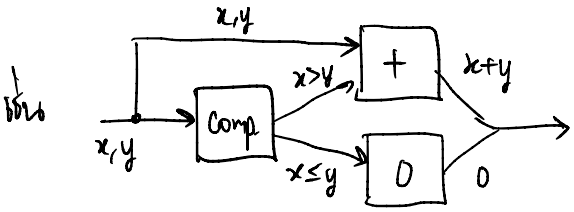
Input : $x0y$
Output : $xy0$
} แบบ Unary





Combining TMs





NUMBER THEORY

Outlines
- ✓ Theory of Divisibility
- ✓ Diophantine
- Congruence
- Prime Number
- Comp. Sys.
- Cryptography

$x = \lceil x \rceil - \{x\}^*$

$x = \lfloor x \rfloor + \{x\}$

→ **Floor Function**

*Def.

$\lfloor x \rfloor = n \; ; \quad x - 1 < \lfloor x \rfloor \leq x \; ; \quad x = n + \varepsilon \quad \{0 \leq \varepsilon < 1\}$

→ **Ceiling Function**

$\lceil x \rceil - 1 < x \leq \lceil x \rceil$

*Def.

$\lceil x \rceil = n \; ; \quad x \leq \lceil x \rceil < x + 1 \; ; \quad x = m - \delta \quad \{0 \leq \delta < 1\}$

Thm.

- $\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor$
- $\lceil x + y \rceil \leq \lceil x \rceil + \lceil y \rceil$
- $n = \lceil n \rceil = \lfloor n \rfloor$
- $\lceil x \rceil - \lfloor x \rfloor = 1$
- $\lfloor -x \rfloor = -\lceil x \rceil$
- $\lceil -x \rceil = -\lfloor x \rfloor$
- $\lfloor x \rfloor + n = \lfloor x + n \rfloor$

→ **Thm.** ให้ $F(x)$ เป็น continuous , $x < y \rightarrow F(x) \leq F(y) \; ; \; x, y \in D_f$

และ $F(x) \in \mathbb{Z} \rightarrow x \in \mathbb{Z}$ จะได้ว่า
- $\lfloor F(x) \rfloor = \lfloor F(\lfloor x \rfloor) \rfloor$
- $\lceil F(x) \rceil = \lceil F(\lceil x \rceil) \rceil$

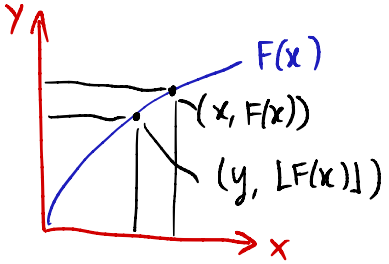
Pf.

$\lfloor x \rfloor \leq x$

$F(\lfloor x \rfloor) \leq F(x)$

$$\begin{cases} \lfloor F(\lfloor x \rfloor) \rfloor \leq \lfloor F(x) \rfloor \\ \lfloor F(x) \rfloor \leq \lfloor F(\lfloor x \rfloor) \rfloor \end{cases}$$

$\therefore \lfloor F(x) \rfloor = \lfloor F(\lfloor x \rfloor) \rfloor$ ∎



โดย $y \mapsto \lfloor F(x) \rfloor$

$y \mapsto F(y)$

$\therefore \lfloor F(x) \rfloor = F(y)$

$\therefore y \leq \lfloor x \rfloor$

$F(y) \leq F(\lfloor x \rfloor)$

$\lfloor F(x) \rfloor = \lfloor F(y) \rfloor \leq \lfloor F(\lfloor x \rfloor) \rfloor$

Divisibility $a|b$ เขียนว่า $b=ac$; $a,b,c \in \mathbb{Z}$

$a =$ "factor of b"

$b =$ "multiple of a"

Thm.
- $a|b \wedge a|c \rightarrow a|(b+c)$
- $a|b \rightarrow a|bc$
- $a|b \wedge b|c \rightarrow a|c$

Modularity (Def.) $m \bmod n = r$; $(m = np + r)$ e.g. $7 \bmod 4 = 3$, $-5 \bmod 3 = 1$

$= m - n\left\lfloor \frac{m}{n} \right\rfloor$

Theory of Divisibility (Def.)

Euclidean Algo.

$$\gcd(a, b) = \gcd(a \bmod b, b)$$

Prime : $p|ab \rightarrow (p|a \vee p|b)$

Thm. Every $n \in \mathbb{Z}$, $n > 1$ → n สามารถเขียนได้ในรูปผลคูณของ primes (uniquely)

Pf.
- n เป็น prime → จบ.
- n ไม่เป็น prime → $n = n_1 n_2$

Pf. $n = p_1 p_2 p_3 \cdots p_r = q_1 q_2 q_3 \cdots q_s$ → เป็นไปไม่ได้! (สมมติว่าเป็นไปได้)

$\forall p_i \in \text{Prime}, \forall q_i \in \text{Prime}$, ให้ $p_i \neq q_j$

Consider $p_1 > q_1$, $M = (p_1 - q_1) q_2 q_3 \cdots q_s$

จะได้ $p_1 \nmid M$ ($p_1 - q_1 < p_1$ และ $p_1 \nmid q_2 q_3 \cdots q_s$)

$M = p_1 q_2 q_3 \cdots q_s - q_1 q_2 \cdots q_s$

$M = p_1 q_2 q_3 \cdots q_s - p_1 p_2 \cdots p_r$

$M = p_1 (q_2 q_3 \cdots q_s - p_2 p_3 \cdots p_r)$

$\therefore p_1 | M$ ↯ → Assumption ผิด.

⊕ $n = a \cdot b$

$n = \sqrt{n} \cdot \sqrt{n}$

gcd , lcm : $\gcd(a,b) = 1 \leftrightarrow a, b$ are relatively prime.

$\gcd(a,b) \cdot \text{lcm}(a,b) = ab$

Extend : $\gcd(a,b) \cdot \text{lcm}(a,b) = ab$
$\gcd(b,c) \cdot \text{lcm}(b,c) = bc$
$\gcd(a,c) \cdot \text{lcm}(a,c) = ac$

$abc = \sqrt{\prod \gcd(x,y) \cdot \text{lcm}(x,y)}$

Factorization → Finding gcd using **Euclidean algorithm**

Given $a, b \in \mathbb{Z}$; $b > 0$, there exists unique $q, r \in \mathbb{Z}$ such that $a = bq + r$; $0 \le r < b$

Pf. By def. $a|b \rightarrow \frac{b}{a} \in \mathbb{Z}$; given $a, b, c, x, y \in \mathbb{Z}$ and $a|b$ and $a|c$, then $\frac{b}{a} \in \mathbb{Z}$ and $\frac{c}{a} \in \mathbb{Z}$.

Therefore, linear combination $\frac{b}{a} \cdot x + \frac{c}{a} \cdot y = \frac{bx+cy}{a}$ which is also an integer. So, $a|(bx+cy)$

From $a = bq + r$, let $m = \gcd(a,b)$, $n = \gcd(b,r)$.

We get $m|a$ and $m|b$, which $m|r$; $r = a - bq$.

As $m|b$ and $m|r$, $m \le \gcd(b,r) = n$. $(m \le n)$

Also $n|b$ and $n|r$, which $n|a$; $a = r + bq$.

As $n|a$ and $n|b$, $n \le \gcd(a,b) = m$. $(n \le m)$

$\therefore m = n \rightarrow \gcd(a,b) = \gcd(b,r)$ □

* $\gcd(b,r) = \gcd(b, a \bmod b)$ *

Pf. $a = bq + r$; $0 \le r < b$, show that $r = a \bmod b$.

By def. $a \bmod b = a - b\left\lfloor \frac{a}{b} \right\rfloor = r$.

As $a = bq + r$, $r = a - bq$. $\therefore q = \left\lfloor \frac{a}{b} \right\rfloor$.

$$\frac{a}{b} - 1 < \left\lfloor \frac{a}{b} \right\rfloor \le \frac{a}{b}$$

$$\frac{a-b}{b} < q \le \frac{a}{b}$$

$$a - b < bq \le a$$

$$a - b + r < bq + r \le a + r$$

$$a - b + r < a \le a + r$$

$$-b + r < 0 \le r$$

$\therefore r \ge 0$ and $r < b$

$\therefore 0 \le r < b$ □

Proof on linear combination, gcd. (Bézout's Identity)

Claim: $\gcd(a,b) = ax+by$; $a,b \in \mathbb{Z}_{>0}$, $x,y \in \mathbb{Z}$

$S = \{am+bn : m,n \in \mathbb{Z} \text{ and } am+bn > 0\}$

If $a,b \neq 0$, then $S \neq \emptyset$ and $S \subseteq \mathbb{Z}_{>0}$.

By well-ordering principle, $d \in S$ and let d be the least value.

$\therefore$ There exists $d = ax+by$; $x,y \in \mathbb{Z}$

As $a,d \in \mathbb{Z}$ and $d \in S$, $a = dq+r$; $0 \leq r < d$, $q,r \in \mathbb{Z}$ ← q is factor of a

$a = (ax+by)q + r$

$r = a - (ax+by)q$

$r = a - axq + byq$

$r = a(1-xq) + b(-yq)$

$\therefore$ r is a linear combination on a,b

while $r=0$ or $0<r<d$.

If $0<r<d$, this contradicts that d is the least value in S. ($r \in S$)

If $r=0$, $r \notin S$, $a = dq$; $d \mid a$.

In the same manner : $b = dp+s$; $0 \leq s \leq d$, $p,s \in \mathbb{Z}$

$\therefore d \mid b$.

From $c \mid a$ and $c \mid b \rightarrow c \mid ax+by \rightarrow c \mid d \rightarrow c \leq d$,

$d \in S$, $d > 0$; $0 < c \leq d$.

c is a common factor of a and b., d is maximum value of c.

Therefore $d = \gcd(a,b) = ax+by$.

□

Congruence

$a \equiv b \pmod m$ means $m \mid a-b$

means $a \bmod m = b \bmod m$

1. $a \equiv a \pmod m$
2. $a \equiv b \pmod m \rightarrow b \equiv a \pmod m$
3. $a \equiv b \pmod m \wedge b \equiv c \pmod m \rightarrow a \equiv c \pmod m$
4. $a \equiv b \pmod m \wedge c \equiv d \pmod m \rightarrow a \pm c \equiv b \pm d \pmod m$
5. $a \equiv b \pmod m \wedge c \equiv d \pmod m \rightarrow ac \equiv bd \pmod m$
6. $a \equiv b \pmod m, k \geq 1 \rightarrow a^k \equiv b^k \pmod m$
7. Polynomial degrees $\in \mathbb{Z}_{\geq 0}$, $a \equiv b \pmod m \rightarrow P(a) \equiv P(b) \pmod m$
8. $a \equiv b \pmod m \rightarrow \gcd(a,m) = \gcd(b,m)$ (Euclidean Algo.)
9. $a \equiv b \pmod m \wedge n \mid m \rightarrow a \equiv b \pmod n$
10. $\gcd(m,n) = 1$, $a \equiv b \pmod m \wedge a \equiv b \pmod n \rightarrow a \equiv b \pmod{mn}$
11. $ca \equiv cb \pmod m \wedge \gcd(c,m) = 1 \rightarrow a \equiv b \pmod m$
12. p is prime, $p \mid ab \rightarrow p \mid a$ or $p \mid b$
13. p is prime, $p \nmid a \rightarrow a^{p-1} \equiv 1 \pmod p$

$a \equiv b \pmod m \rightarrow a+k \equiv b+k \pmod m$

$a \equiv b \pmod m \rightarrow ak \equiv bk \pmod m$

$a \equiv b \pmod m \rightarrow ak \equiv bk \pmod{km}$

$ax \equiv b \pmod m$ has solutions iff $\gcd(a,m) \mid b$

Diophantine Eqn. (short.) : $ax+by=c$; $\gcd(a,b) \mid c$ **

(Bi-linear)

[$a,b,c \in \mathbb{Z} \rightarrow$ w.r. $x,y \in \mathbb{Z}$]

eg $195x+42y=12$ $\rightarrow \gcd(195,42)=3$

$195 = 42\times 4 + 27$

$42 = 27\times 1 + 15$

$27 = 15\times 1 + 12$

$15 = 12\times 1 + 3$

$12 = 3\times 4$

$\rightarrow 3 = 15 - 12\times 1$

$= 15 - (27-15\times 1)\times 1$

$= 2\cdot 15 - 1\cdot 27$

$= 2\cdot(42-27\cdot 1) - 1\cdot 27$

$= 2\cdot 42 - 3\cdot 27$

$= 2\cdot 42 - 3\cdot(195-42\cdot 4)$

$3 = 14\cdot 42 - 3\cdot 195$

$\rightarrow (195(-3) + 42(14))4 = 3\times 4$

$195(-12) + 42(56) = 12$

$\therefore (x,y) = (-12,56)$ → not closest to (0,0)

different signs.

$195(-12+42) + 42(56-195) = 12$

$195\left(-12+\frac{42n}{3}\right) + 42\left(56-\frac{195n}{3}\right) = 12$

gcd → 3

$\therefore (x,y) = (-12+14n,\ 56-65n)$

Choose $n=1$: $(x,y) = (2,-9)$ → closest!

$3x + 2y = 73$

mod 3 : $2y \equiv 1 \pmod 3$

$-y \equiv 1 \pmod 3$

$\therefore -y = 1 + 3n \rightarrow y = -1 - 3n$

$3x + 2(-1-3n) = 73$

$3x - 2 - 6n = 75$

$x - 2n = 25$

$x = 25 + 2n$

$(x, y) = (25+2n, -1-3n)$

$195x + 42y = 12$ $\quad \gcd(195, 42) = 3$, $3 \mid 12$ ✓

$65x + 14y = 4$

mod 14 : $9x \equiv 4 \pmod{14}$

$54x \equiv 24 \pmod{14}$

$-56x$

$-2x \equiv 24 \pmod{14}$

$x \equiv -12 \pmod{14}$

$x \equiv 2 \pmod{14}$

$\therefore x = 2 + 14n$

$65(2) + 65(14)n + 14y = 4$

$14y = 4 - 130 - 65(14)n$

$y = -9 - 65n$

$\therefore (x, y) = (2+14n, -9-65n)$

n=0 : $(2, -9)$ ✓

14×9

$14 \times 4 = 56$

$65 - 56 = 9$

$9 \times 6 = 54$

Ver. 2

$a = bq_0 + r_0$ $\longrightarrow$ $\frac{a}{b} = q_0 + \frac{r_0}{b}$

$b = r_0 q_1 + r_1$ $\frac{a}{b} = q_0 + \frac{1}{q_1 + \frac{r_1}{r_0}}$

$r_0 = r_1 q_2 + r_2$

$\vdots$

$r_n = r_{n+1} q_n$

$$\frac{a}{b} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\ddots + \cfrac{1}{q_n}}}}$$

ถ้า q_n $(n \to \infty)$: จำนวน $\mathbb{Q}'$

"Simple continued fraction" โดย $\frac{a}{b}$ เป็นจำนวน $\mathbb{Q}$ เมื่อ q_n finite และเป็น convergence ของ RHS.

eg. $q_0 = \lfloor \sqrt{2} \rfloor = 1$ (Use $a = b\left\lfloor \frac{a}{b} \right\rfloor + r$, $x = \lfloor x \rfloor + \{x\}$)

$\sqrt{2} = 1 + \frac{1}{x_1}$ $\longrightarrow x_1 = \frac{1}{\sqrt{2}-1} = \sqrt{2}+1$

$\longrightarrow \lfloor x_1 \rfloor = 2$

$$\sqrt{2} = 1 + \cfrac{1}{2 + \cfrac{1}{x_2 \cdots x_{n\to\infty}}}$$

$$a = \lfloor x_0 \rfloor + \cfrac{1}{\lfloor x_1 \rfloor + \cfrac{1}{\ddots}}$$

Let $C_0 = q_0 = P_0/Q_0$

$C_1 = q_0 + \frac{1}{q_1} = \frac{q_0 q_1 + 1}{q_1} = \frac{P_1}{Q_1}$

$C_2 = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2}} = q_0 + \frac{q_2}{q_1 q_2 + 1} = \frac{q_2(q_0 q_1 + 1) + q_0}{q_2 q_1 + q_0} = \frac{P_2}{Q_2}$

$\therefore C_k = \frac{P_k}{Q_k} = \frac{q_k P_{k-1} + P_{k-2}}{q_k Q_{k-1} + Q_{k-2}}$

$* \quad Q_i P_{i-1} - P_i Q_{i-1} = (-1)^i$

Residue Number System (RNS)