グラフ同型性判定問題
Graph Isomorphism

小畠教寛
北海道大学　大学院情報科学院

このスライドと実装例は，githubの
obatakyoukan/isomorphismに挙げています．

目次

- 問題説明
- 不変量
- *Certificate*
  - 探索空間の削減
  - 枝刈り方法

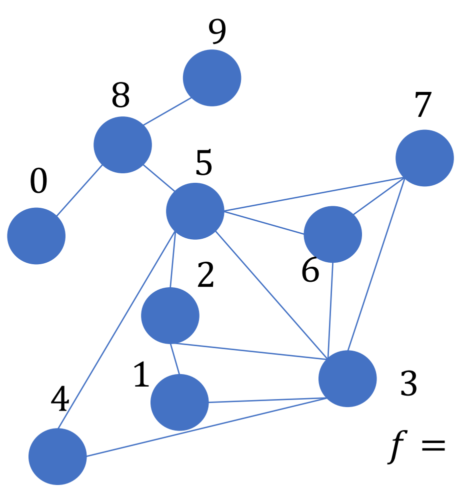
問題説明

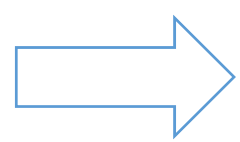
グラフ同型性

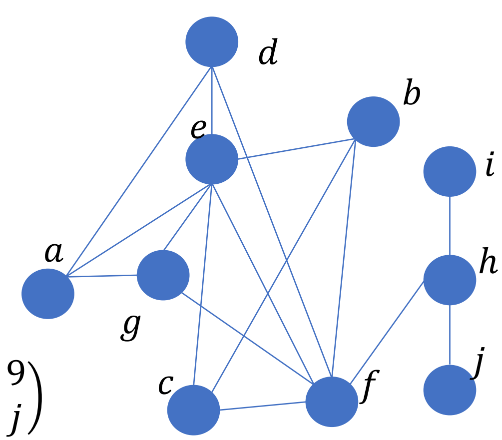
- 入力 : 無向グラフ $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$.
- 出力 : 以下の条件を満たす全単射 f.

$f\colon V_1 \to V_2$で，$\{f(x), f(y)\} \in E_2 \Leftrightarrow \{x, y\} \in E_1$.

このような f を**同型写像**(***isomorphism***) という.



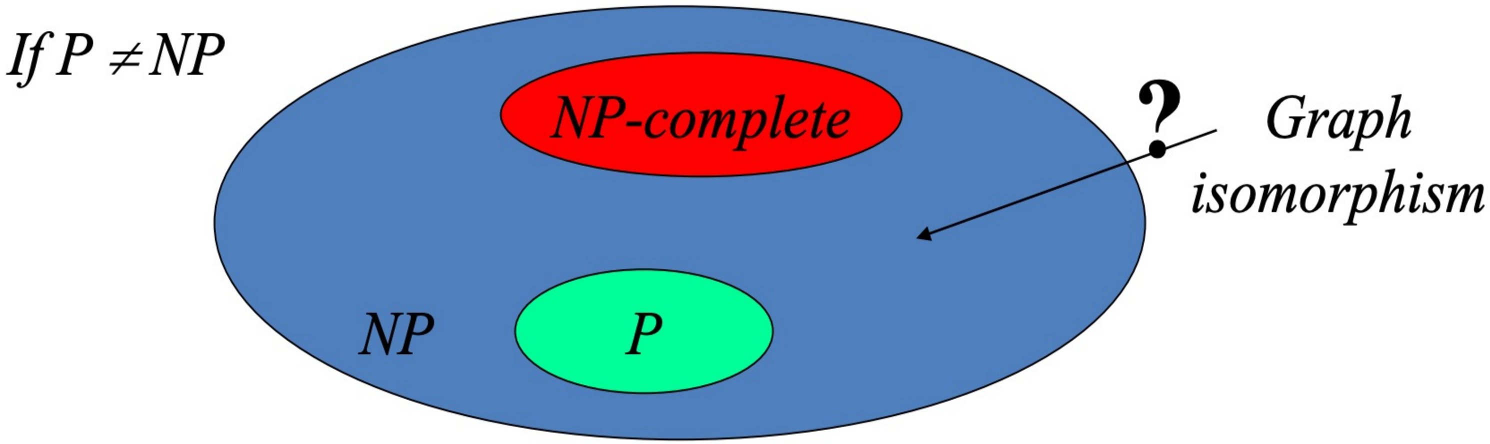




$$f = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ i & a & d & e & g & f & c & b & h & j \end{pmatrix}$$

グラフ同型性の計算クラス

グラフ同型問題は，NPに属するが,
$P \neq NP$ならば,
PとNP完全のどちらかに属するか未解明.



引用：北海道大学大学院 情報科学研究科 集中講義「大規模離散計算科学特論」
講義資料 「グラフ・部分グラフ同型判定の原理」客員教授 鷲尾隆（大阪大学産業科学研究所）
http://www-erato.ist.hokudai.ac.jp/lecture2013/docs/washio2.pdf

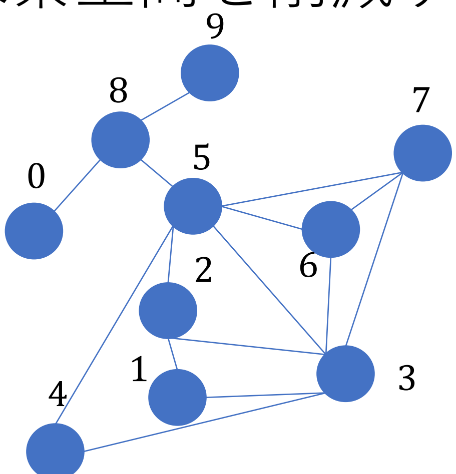
不変量

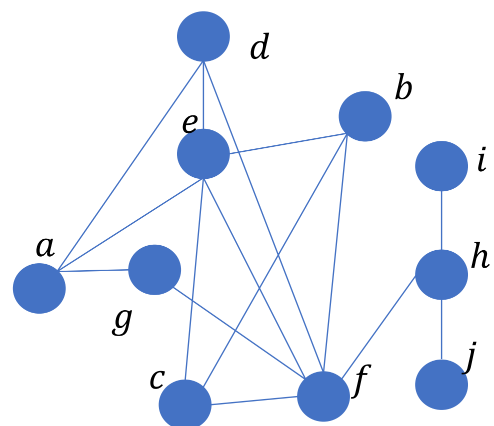
分割による探索空間の削減

集合 X の**分割(*partition*)**とは，X 全体を互いに重ならないブロックに分けることをいう．

このスライドでの分割は，頂点集合 V の分割とする．

頂点順序に依存しない特徴で，分割 X を与えることで，探索空間を削減する．





$X(G_1)$
$= \{0,9\},\{8\},\{2,4,6,7\},\{1\},\{3,5\}$

$X(G_2)$
$= \{i,j\},\{h\},\{b,c,d,g\},\{a\},\{e,f\}$

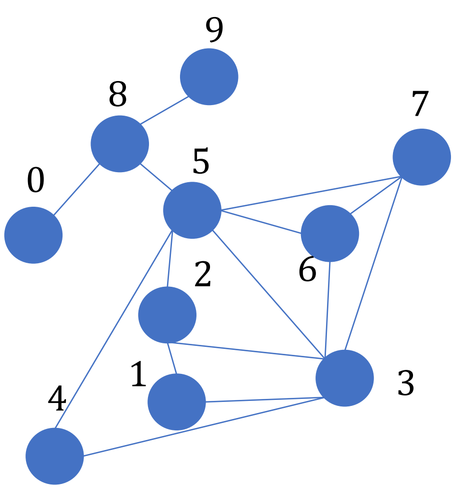
$10! = 3628800$ $(2!)(1!)(4!)(1!)(2!) = 96$

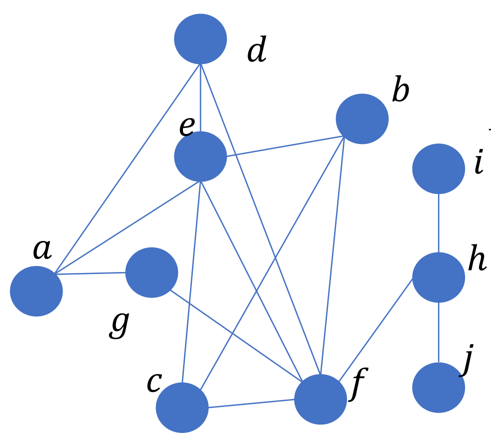
不変量とは

不変量(*invariant*)とは，以下の条件を満たすような関数Φである．

$\Phi(G_1) = \Phi(G_2)$ if G_1 and G_2 are *isomorphic* .

ex) ソートした次数の配列





$v \in V_1$:	0	1	2	3	4	5	6	7	8	9
$\deg(v)$:	1	3	3	6	3	6	3	3	3	1

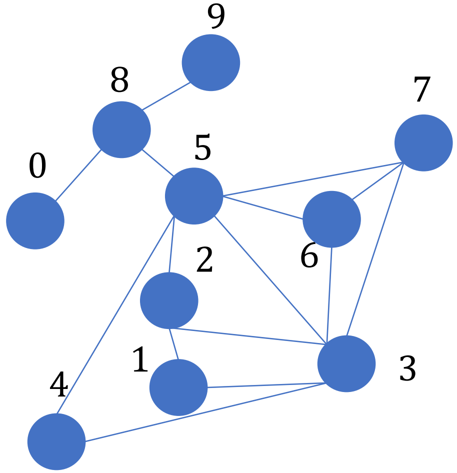
$v \in V_2$:	a	b	c	d	e	f	g	h	i	j
$\deg(v)$:	3	3	3	3	6	6	3	3	1	1

$\Phi(G_1) = [1\ 1\ 3\ 3\ 3\ 3\ 3\ 3\ 6\ 6]$

$\Phi(G_2) = [1\ 1\ 3\ 3\ 3\ 3\ 3\ 3\ 6\ 6]$

不変量による分割

n個の不変量のリストを $L = [D_1, D_2, \ldots, D_n]$とする．
このとき，分割 X では，
xとyが同じブロックに属する$\Leftrightarrow D_i(G,x) = D_i(G,y)$
を満たすようにする．



$v \in V_1$:	0	1	2	3	4	5	6	7	8	9
$\deg(v)$:	1	3	3	6	3	6	3	3	3	1

$\Downarrow$

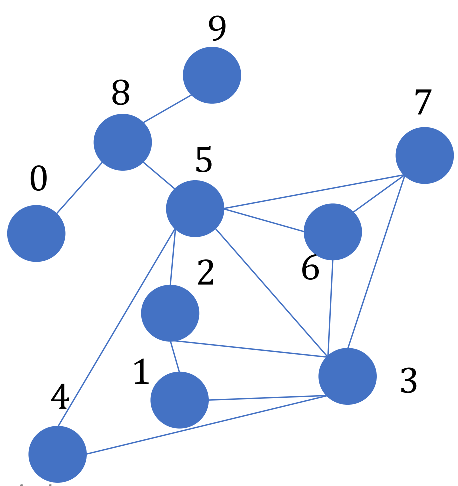
$X(G_1) = \{0,9\}, \{1,2,4,6,7,8\}, \{3,5\}$

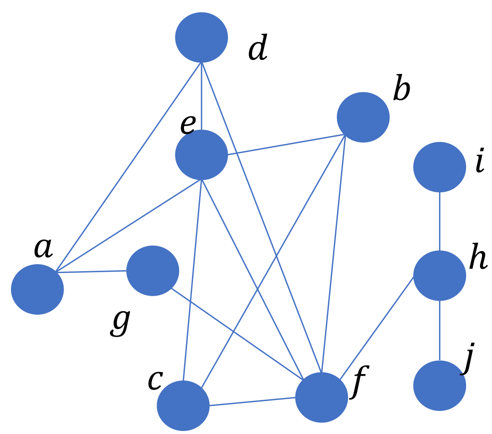
分割の例(1/4)

以下の関数で不変量を与え，同型性を判定する.

1. $D_1(G, x) = \deg(x)$
2. $D_2(G, x) = \left[\, d_j(x) : j = 1,2,\ldots, d_{n-1}(x)\right]$

ここで，$d_j(x) = |\{y : \deg(\,v\,) = j\,, v \in \Gamma(y)\}|$とする.





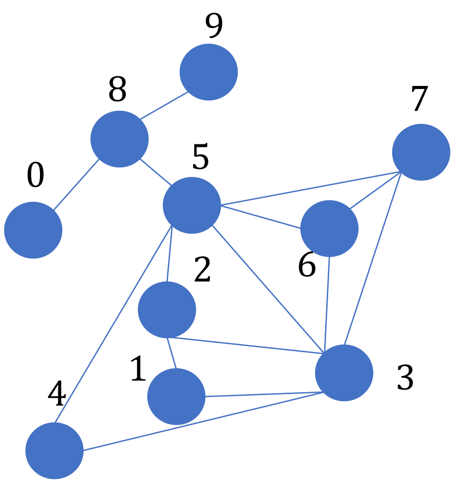
$X(G_1)$
$= \{0,9\}, \{8\}, \{2,4,6,7\}, \{1\}, \{3,5\}$

$X(G_2)$
$= \{i, j\}, \{h\}, \{b, c, d, g\}, \{a\}, \{e, f\}$

分割の例(2/4)

$v \in V_1$	: 0 1 2 3 4 5 6 7 8 9
$D_1(G_1, v)$:	1 3 3 6 3 6 3 3 3 1

$D_2(G_1, 0) = (0,0,1,0,0,0,0,0,0)$
$D_2(G_1, 1) = (0,0,2,0,0,1,0,0,0)$
$D_2(G_1, 2) = (0,0,1,0,0,2,0,0,0)$
$D_2(G_1, 3) = (0,0,5,0,0,1,0,0,0)$
$D_2(G_1, 4) = (0,0,1,0,0,2,0,0,0)$
$D_2(G_1, 5) = (0,0,5,0,0,1,0,0,0)$
$D_2(G_1, 6) = (0,0,1,0,0,2,0,0,0)$
$D_2(G_1, 7) = (0,0,1,0,0,2,0,0,0)$
$D_2(G_1, 8) = (2,0,0,0,0,1,0,0,0)$
$D_2(G_1, 9) = (0,0,1,0,0,0,0,0,0)$

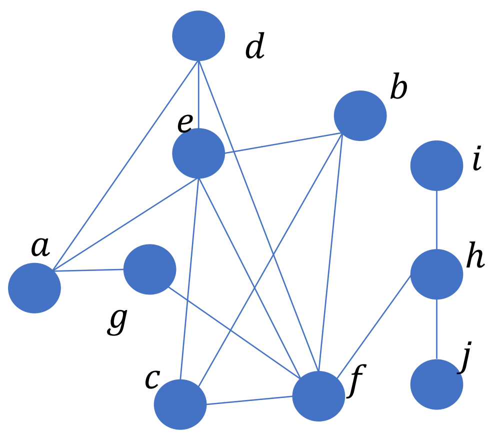


$X(G_1) = \{0,9\}, \{8\}, \{2,4,6,7\}, \{1\}, \{3,5\}$

分割の例(3/4)

$v \in V_2:$	a	b	c	d	e	f	g	h	i	j
$D_2(G_2, v):$	3	3	3	3	6	6	3	3	1	1

$$D_2(G_2, a) = (0,0,2,0,0,1,0,0,0)$$
$$D_2(G_2, b) = (0,0,1,0,0,2,0,0,0)$$
$$D_2(G_2, c) = (0,0,1,0,0,2,0,0,0)$$
$$D_2(G_2, d) = (0,0,1,0,0,2,0,0,0)$$
$$D_2(G_2, e) = (0,0,5,0,0,1,0,0,0)$$
$$D_2(G_2, f) = (0,0,5,0,0,1,0,0,0)$$
$$D_2(G_2, g) = (0,0,1,0,0,2,0,0,0)$$
$$D_2(G_2, h) = (2,0,0,0,0,1,0,0,0)$$
$$D_2(G_2, i) = (0,0,1,0,0,0,0,0,0)$$
$$D_2(G_2, j) = (0,0,1,0,0,0,0,0,0)$$

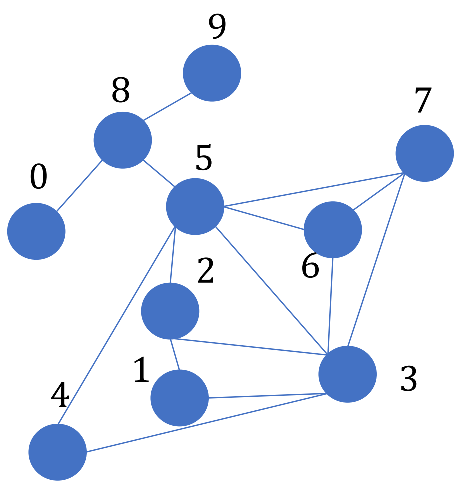


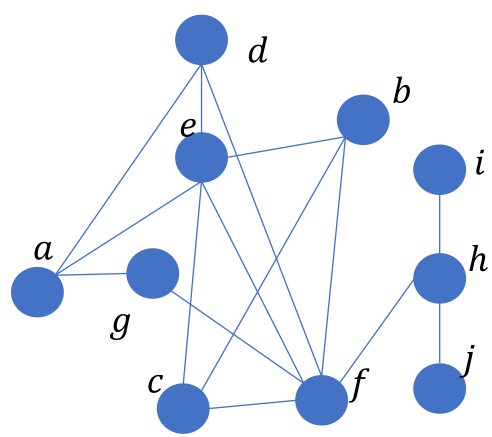
$$X(G_2) = \{i, j\}, \{h\}, \{b, c, d, g\}, \{a\}, \{e, f\}$$

分割の例(4/4)

$$\{0,9\} \leftrightarrow \{i,j\}$$
$$\{8\} \leftrightarrow \{h\}$$
$$\{2,4,6,7\} \leftrightarrow \{b,c,d,g\}$$
$$\{1\} \leftrightarrow \{a\}$$
$$\{3,5\} \leftrightarrow \{e,f\}$$

分割を与える不変量との対応関係から，同型な可能性があるのは，$(2!)(1!)(4!)(1!)(2!) = 96$通りのみで，これだけを確かめれば良い．





$X(G_1)$
$= \{0,9\},\{8\},\{2,4,6,7\},\{1\},\{3,5\}$

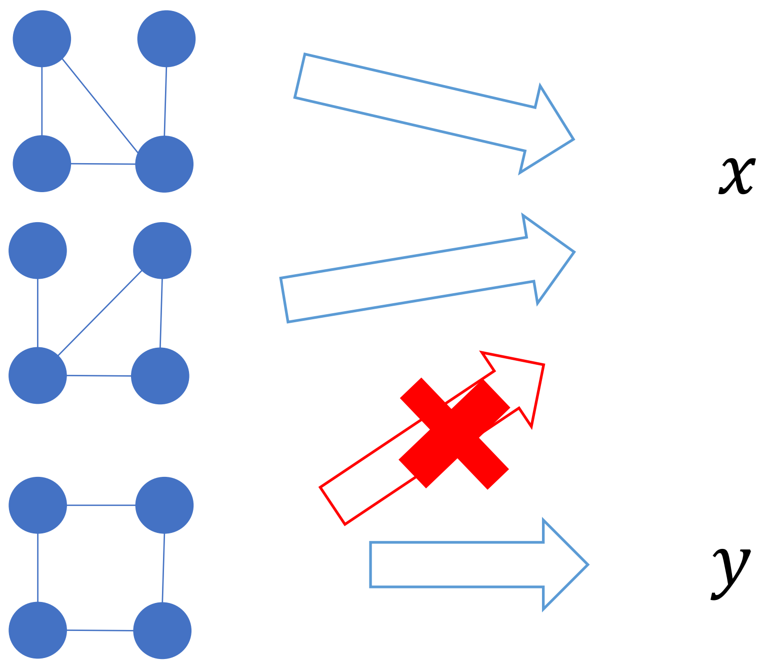
$X(G_2)$
$= \{i,j\},\{h\},\{b,c,d,g\},\{a\},\{e,f\}$

certificate

certificate

certificateとは，任意のグラフ G_1, G_2 に対して，以下の条件を満たす関数 Cert である．

$\mathrm{Cert}(G_1) = \mathrm{Cert}(G_2) \Leftrightarrow G_1$ and G_2 are *isomorphic*.



簡単な*certificate*の例

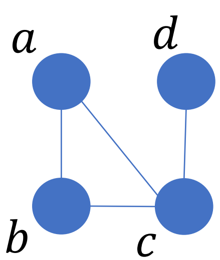
グラフを並べ替えることでできる順列 $\pi : V \to V$ に対して，行列 $A_\pi(G)$ は，以下のように定義する．

$$A_\pi(G)[u,v] = \begin{cases} 1 & \text{if } \{\pi(u), \pi(v) \in E\} \\ 0 & \text{otherwise} \end{cases}$$

このとき，$\mathrm{Num}_\pi(G)$ を $A_\pi(G)$ の対角成分より上の $n(n-1)/2$ 要素の書いたときに2進数で表れる数字．

$$\mathrm{Cert}(G) = \min\{\mathrm{Num}_\pi(G) : \pi \in \mathrm{Sym}(V)\}$$

ここで，$\mathrm{Sym}(X)$ は，X の全ての順列の集合とする．



$$\begin{array}{c c} & \begin{array}{cccc} d & a & b & c \end{array} \\ \begin{array}{c} d \\ a \\ b \\ c \end{array} & \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

$$\pi = \begin{pmatrix} a & b & c & d \\ 1 & 2 & 3 & 0 \end{pmatrix}$$

$001111_{(2)}$

15

certificate

探索空間の削減

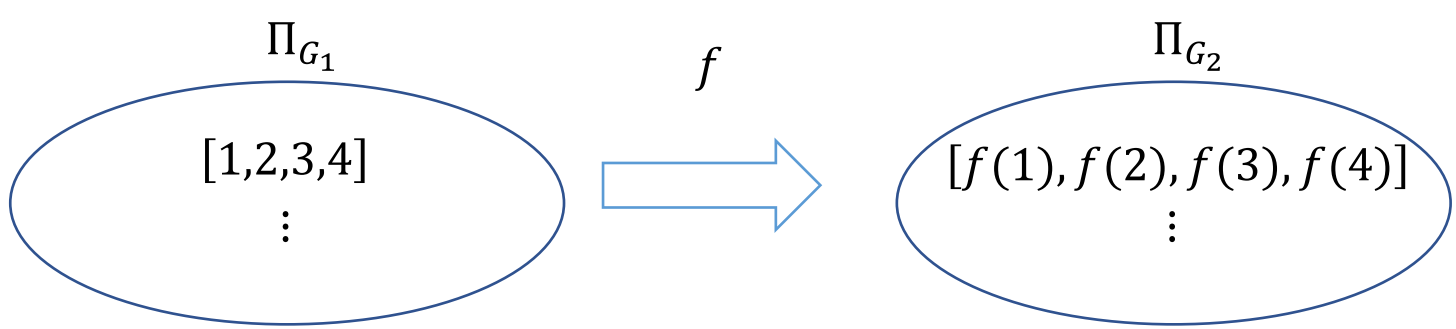
*certificate*の高速化アイデア

$$\mathrm{Cert}(G) = \min\{\mathrm{Num}_{\pi}(G) : \pi \in \mathrm{Sym}(V)\}$$

を計算するのに，$\mathrm{Sym}(V)$ は $|V|!$ 個の要素があり，膨大な時間が必要である．より効率的に計算するために，以下のように定義されたものを用いる．

$$\mathrm{Cert}(G) = \min\{\mathrm{Num}_{\pi}(G) : \pi \in \Pi_G\}$$

ここで，Π_G は，元々のVの順序に依存しないような，G の構造によって決められた順序の集合である．



Refine algorithmの概要

Refine algorithmとは，ある分割Aから，***refinement***である***equitable partition*** Bを得るアルゴリズムである.

このアルゴリズムは，以下の定理1の性質を持つ.

定理1

f を$G_1=(V_1,E_1)$から$G_2=(V_2,E_2)$への同型写像とすると，

G_1 と A_1をRefineの入力として，B_1 を得たとき，

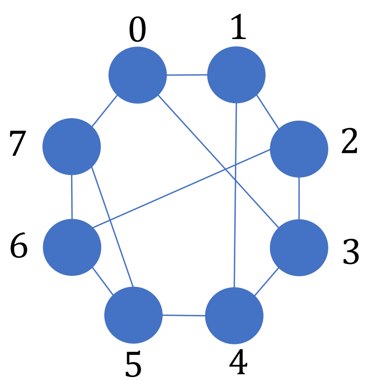
G_2 と $f(A_1)$ をRefineの入力としたとき，$f(B_1)$ を得る.

equitable partition

equitable partition B は，任意の i, j に対して，以下の条件を満たす分割である．

$$\forall u, v \in B[i] : |\Gamma(u) \cap B[j]| = |\Gamma(v) \cap B[j]|$$

ここで，$\Gamma(u) = \{x \in V : \{u, v\} \in E\}$である．



$B = [\{0\}, \{2,4\}, \{5,6\}, \{7\}, \{1,3\}]$

refinement

分割 B が分割 A の*refinement*であることは,

1. すべての $B[i]$ は，いくつかの $A[j]$に含まれる.
2. 任意の$u \in A[i_1], v \in A[j_1]$ $(i_1 \leq j_1)$に対して，$u \in B[i_2], v \in B[j_2]$ $(i_2 \leq j_2)$である.

を満たすということである.

ex) $B = [\{0\}, \{2,4\}, \{5,6\}, \{7\}, \{1,3\}]$は,
$A = [\{0\}, \{1,2,3,4,5,6,7\}]$の*refinement*である.
一方で，$B = [\{2,4\}, \{5,6\}, \{0\}, \{7\}, \{1,3\}]$は,
Aの*refinement*でない.

Refine algorithm

Refineは，以下の手順で行われる.

1. 分割 Aをコピーして分割 Bを作る.
2. 分割 SをBのブロックを逆順に並べたものにする.
3. While $S \neq \phi$ do
4. Sの末尾からブロックを取り出し，Tとする.
5. for each block $B[i]$ of B do
6. for each h, set $L[h] = \{v \in B[i]: D_T(v) = h\}$;
7. if Lが空でないものが2つ以上ある.
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順で Sの末尾に追加する.

ここで，$D_T(v) = |\Gamma(v) \cap T|$とする.

Refine algorithmの例(1/6)

$A=[\{0\},\{1,2,3,4,5,6,7\}]$でのRefineの実行結果.

手順1：

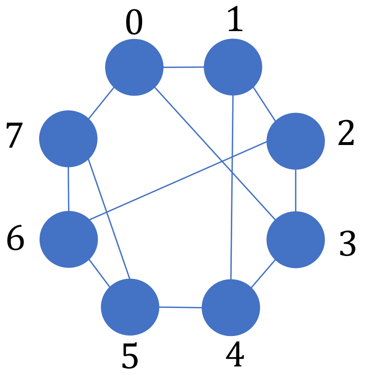
$$B=[\{0\},\{1,2,3,4,5,6,7\}]$$

手順2：

$$S=[\{1,2,3,4,5,6,7\},\{0\}]$$

Refineは，以下の手順で行われる.

1. 分割 Aをコピーして分割 Bを作る.
2. Let S be a list containing the blocks of B



Refine algorithmの例(2/6)

$$B=[\{0\},\{1,2,3,4,5,6,7\}], S=[\{1,2,3,4,5,6,7\},\{0\}]$$

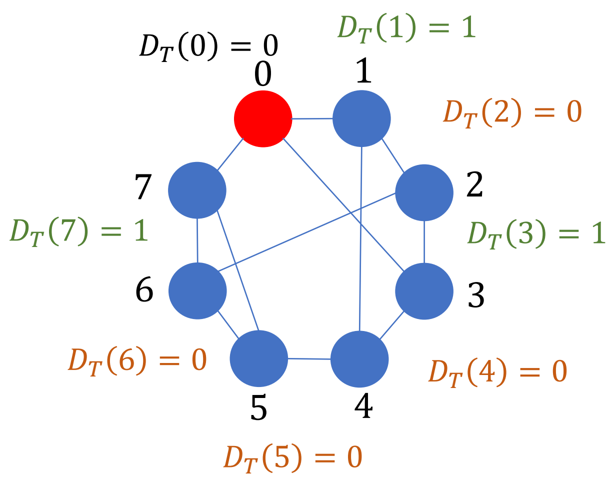
手順4：$T=\{0\}, S=[\{1,2,3,4,5,6,7\}]$

手順6：$L[0]=\{2,4,5,6\}, L[1]=\{1,3,7\}$

手順8：$B=[\{0\},\{2,4,5,6\},\{1,3,7\}]$,
$S=[\{1,2,3,4,5,6,7\},\{1,3,7\},\{2,4,5,6\}]$

3. While $S\neq\phi$ do
4. Sの末尾からブロックを取り出し，Tとする.
5. for each block $B[i]$ of B do
6. for each h, set $L[h]=\{v\in B[i]: D_T(v)=h\}$;
7. if Lが空でないものが2つ以上ある.
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順で Sの末尾に追加する.

ここで，$D_T(v)=|\Gamma(v)\cap T|$とする.



Refine algorithmの例(3/6)

$$B = [\{0\}, \{2,4,5,6\}, \{1,3,7\}],$$
$$S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}, \{2,4,5,6\}]$$

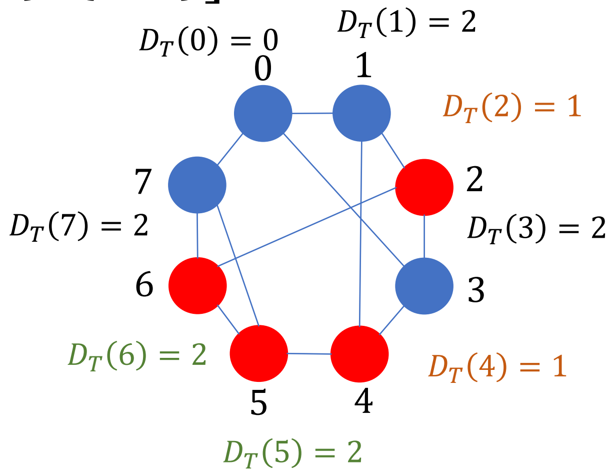
手順4：$T = \{2,4,5,6\}, S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}]$

手順6：$L[1] = \{2,4\}, L[2] = \{5,6\}$

手順8：$B = [\{0\}, \{2,4\}, \{5,6\}, \{1,3,7\}],$
$$S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}, \{5,6\}, \{2,4\}]$$

3. While $S \neq \phi$ do
4. Sの末尾からブロックを取り出し，Tとする．
5. for each block $B[i]$ of B do
6. for each h, set $L[h] = \{v \in B[i]: D_T(v) = h\}$;
7. if Lが空でないものが2つ以上ある．
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順で Sの末尾に追加する．

ここで，$D_T(v) = |\Gamma(v) \cap T|$とする．



Refine algorithmの例(4/6)

$$B = [\{0\}, \{2,4\}, \{5,6\}, \{1,3,7\}],$$
$$S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}, \{5,6\}, \{2,4\}]$$

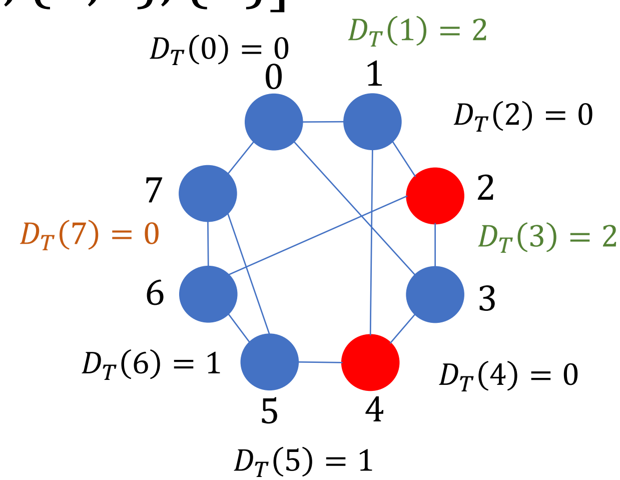
手順4：$T = \{2,4\}, S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}, \{5,6\}]$

手順6：$L[0] = \{7\}, L[2] = \{2,4\}$

手順8：$B = [\{0\}, \{2,4\}, \{5,6\}, \{7\}, \{1,3\}],$
$S = [\{1,2,3,4,5,6,7\}, \{1,3,7\}, \{5,6\}, \{1,3\}, \{7\}]$

3. While $S \neq \phi$ do
4. Sの末尾からブロックを取り出し，Tとする.
5. for each block $B[i]$ of B do
6. for each h, set $L[h] = \{v \in B[i]: D_T(v) = h\}$;
7. if Lが空でないものが2つ以上ある.
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順で Sの末尾に追加する.

ここで，$D_T(v) = |\Gamma(v) \cap T|$とする.



Refine algorithmの例(5/6)

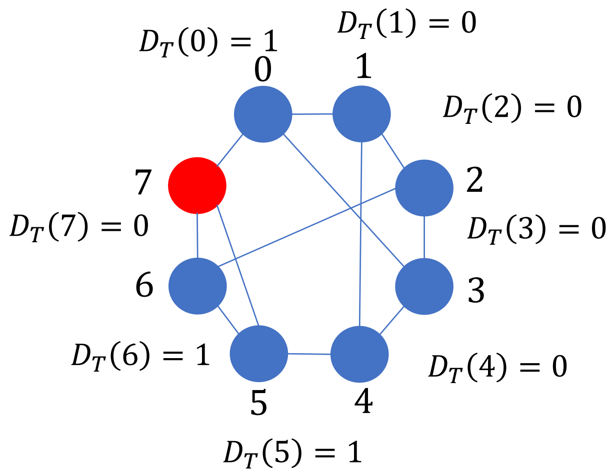
$$B = [\{0\},\{2,4\},\{5,6\},\{7\},\{1,3\}],$$
$$S = [\{1,2,3,4,5,6,7\},\{1,3,7\},\{5,6\},\{1,3\},\{7\}]$$

手順4：$T = \{7\}, S = [\{1,2,3,4,5,6,7\},\{1,3,7\},\{5,6\},\{1,3\}]$

手順7：すべての$B[i]$について，Lが空でないものが2つ以上ない.

3. While $S \neq \phi$ do
4. Sの末尾からブロックを取り出し，Tとする.
5. for each block $B[i]$ of B do
6. for each h, set $L[h] = \{v \in B[i]: D_T(v) = h\}$;
7. if Lが空でないものが2つ以上ある.
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順でSの末尾に追加する.

ここで，$D_T(v) = |\Gamma(v) \cap T|$とする.



Refine algorithmの例(6/6)

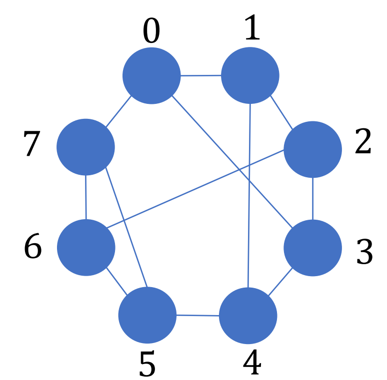
3から8行目を繰り返すと,

$$B = [\{0\}, \{2,4\}, \{5,6\}, \{7\}, \{1,3\}]$$

が得られる.

3. While $S \neq \phi$ do
4. Sの末尾からブロックを取り出し，Tとする.
5. for each block $B[i]$ of B do
6. for each h, set $L[h] = \{v \in B[i]: D_T(v) = h\}$;
7. if Lが空でないものが2つ以上ある.
8. $B[i]$の場所にLの空でないブロックを置き換え，逆順で Sの末尾に追加する.

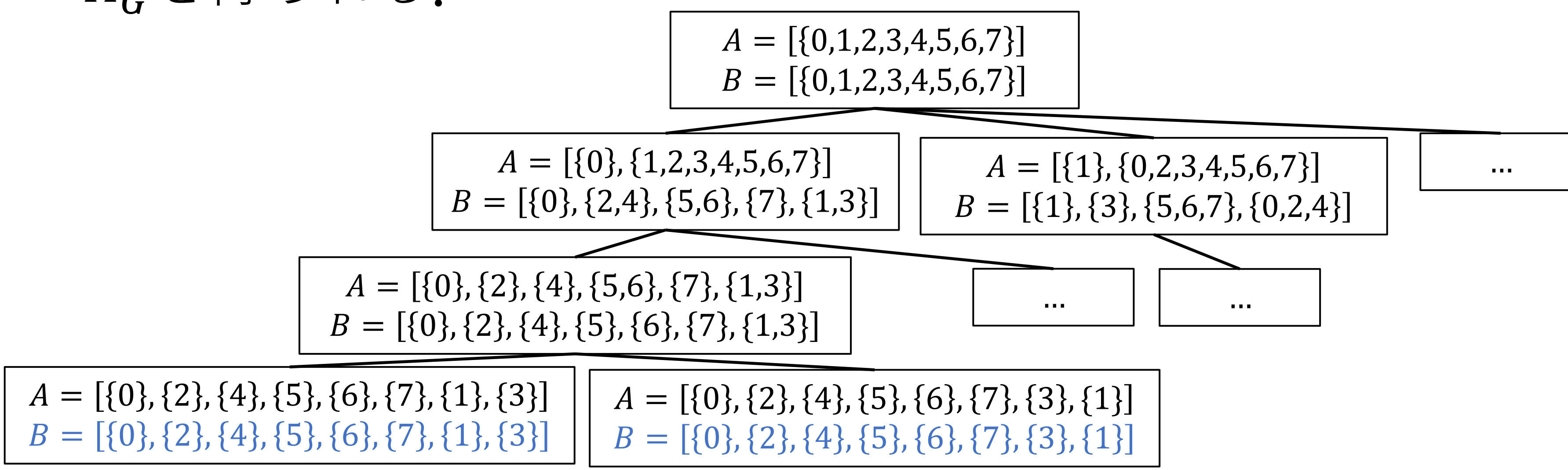
ここで，$D_T(v) = |\Gamma(v) \cap T|$とする.



RefineによるΠ_Gの作り方

各ブロックのサイズが 1になるまで，ブロックのサイズ$m(>1)$の最初のブロックを，サイズを 1と $m-1$に分割し，Refineアルゴリズムを適用する．

このようにすることで，頂点順序に関係ない順列の集合Π_Gを得られる．



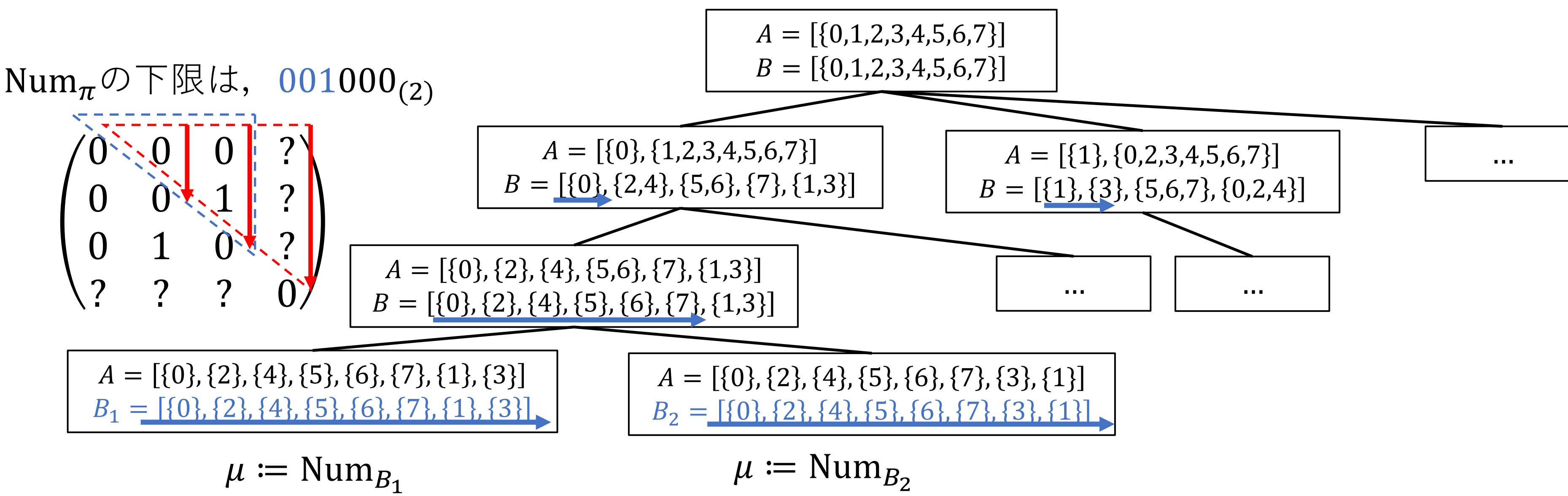
certificate

枝刈り方法

Certの下限による枝刈り

Num_πの最小値の候補をμとする．

先頭から既に決定している順序で，Num_πの下限が計算でき，μより大きくなると探索を打ち切る．



自己同型性

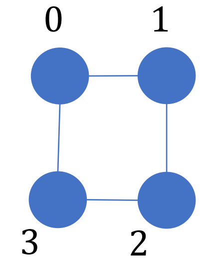
全単射 f が，**自己同型**(***automorphism***)であるとは,

$f: V \to V$で，$\{x, y\} \in E \Leftrightarrow \{f(x), f(y)\} \in E$

を満たすということである.

f, g が自己同型であるとき,

$f * g$ も自己同型である.

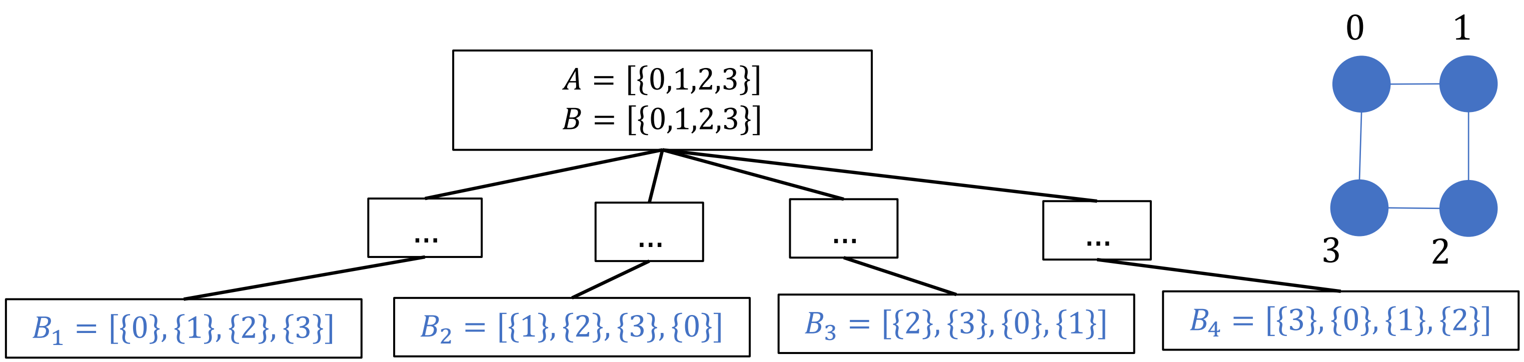


ex) $f = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{pmatrix}$が自己同型のとき,

$f * f = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 2 & 3 & 0 & 1 \end{pmatrix}$も自己同型である.

自己同型による枝刈り

探索中に見つけた自己同型な写像を用いて，自己同型群を探索しないようにする．



$\mathrm{Num}_{B_1} = \mathrm{Num}_{B_2}$ のとき， $f = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{pmatrix}$ は自己同型である．

まとめ

グラフ同型性判定問題を解くアルゴリズムに関する基本的なテクニックを紹介した.

- 不変量による頂点分割での探索空間削減
- $certificate$によるグラフ同型性判定
  - Refineアルゴリズムによる探索空間の削減
  - 下限の計算や自己同型性による枝刈り

Appendix

付録

定理1の証明(1/2)

定理1を証明するには,

$\forall v \in V_1, \forall T \subset V_1$について，$D_T(v) = D_{f(T)}\big(f(v)\big)$を示せばよい.

ここで，$D_T(v) = |\Gamma(v) \cap T|$とする.

これを示すには，$f(\Gamma(v) \cap T) = \Gamma\big(f(v)\big) \cap f(T)$を示せればよい.

fは同型写像なので，$(u, v) \in E_1 \Leftrightarrow \big(f(u), f(v)\big) \in E_2$であるので,

$f\big(\Gamma(v)\big) = \Gamma\big(f(v)\big)$である.

$\forall u \in \Gamma(v) \cap T$のとき，$u \in T$なので，$f(u) \in f(T)$である.

また，$u \in \Gamma(v)$なので，$f(u) \in f\big(\Gamma(v)\big) = \Gamma\big(f(v)\big)$である.

したがって，$f(u) \in \Gamma\big(f(v)\big) \cap f(T)$より,

$f(\Gamma(v) \cap T) \subset \Gamma\big(f(v)\big) \cap f(T)$である.

定理1の証明(2/2)

$\forall u \in \Gamma\big(f(v)\big) \cap f(T) = f\big(\Gamma(v)\big) \cap f(T)$のとき,

fは全単射であるので,

$u \in f\big(\Gamma(v)\big)$ならば，$f^{-1}u \in \Gamma(v)$であり,

$u \in f(T)$ならば，$f^{-1}u \in T$である.

したがって，$f^{-1}u \in \Gamma(v) \cap T$であり,

$u \in f(\Gamma(v) \cap T)$である.

よって，$\Gamma\big(f(v)\big) \cap f(T) \subset f(\Gamma(v) \cap T)$である.

以上より，$f(\Gamma(v) \cap T) = \Gamma\big(f(v)\big) \cap f(T)$である.