



## 分割アルゴリズムに基づく同型グラフの検索について\*

榎本 肇\*\* 片山卓也\*\* 米崎直樹\*\*

### Abstract

Structured data are represented by simple graphs, so it is necessary to decide the equivalence of them in order to retrieve them. We introduce a weak isomorphism of graphs called "tree-isomorphism" and its basic properties are discussed.

The tree-isomorphism is defined by an algorithm based on an automorphism partition of a set of vertices. It is shown that the time and space complexities of the algorithm are proportional to a polynomial order of vertices constructing a graph. Moreover, we apply the partitioning algorithm to a backtrack procedure for graph isomorphism, and related file organizations are also considered.

### 1. ま え が き

内容検索や事実検索のためのデータベースでは、データ間の関係を表現するために構造を用い、データの検索はその構造に沿って行われるが、データの持つ構造自身をキーとするような検索は行われない。一方図形、ネットワーク、結線回路図、化学構造式といった構造自身の意味を持つデータにより構成されるデータベースにおいては、その構造自身を問題とする述語に対する検索が重要となる。特に、指定された構造と同型な構造をもつデータを高速に検索できるようにデータベースを構成しておくことは不可欠である。

本稿では、構造がグラフとして表現される場合を対象として以上の問題について考察する。

グラフの同型性を  $O(N^k)$  ( $k$  は、グラフの点の数  $N$  によらない定数) 時間で判定するアルゴリズムは、未だ知られておらず、また NP-complete な問題であるかも分かっていない。

点集合の分割を基にしたアルゴリズムとして、Corniel, Gotlieb の方法<sup>1)</sup>が効率的なものとして知られているが、この方法では、多角形で  $O(N^4)$ ,  $h$ -強正則

なグラフ<sup>2)</sup>については  $O(N^{5+h})$  ( $h$  の上限は  $N$ ) の時間を要し、用いられる conjecture の反例となるグラフに対してはトリビアルな方法をとらねばならない。また、同型判定のためのヒューリスティックなバックトラック・アルゴリズムにおいても、バックトラック木の大きさを制限するために、点集合の分割が有効に用いられている<sup>3), 5), 7)-9)</sup>。

グラフが木である場合には、 $O(N)$  の簡単な方法<sup>1), 3)</sup>が知られており、平面グラフに関しても  $O(N)$  の方法<sup>11)</sup>が知られているが、特殊な符号化によって部分グラフの構造を整数符号で表わすので、一般のグラフに対してこの方法を直接拡張することはできない。

これまでに与えられたいずれの方法も、2つのグラフの同型判定を高速に行うことだけを目的にしたものであり、大量のグラフの中から同型なグラフを求めるには、個々のグラフに対して同型判定を行う必要があるので必ずしもむいていない。

本稿では、まず木同型分割と呼ばれる点集合の分割を基に、 $N$  の多項式時間内で計算されるような、木同型と呼ばれる弱い同型性の概念を提唱し、その性質を明らかにする。また種々のグラフのクラスにおける同型性との関係について論じ、そのアルゴリズムの複雑さについても考察する。

さらに、以上の結果を基に、このような分割を用いた同型なグラフの効率的検索を行うための、ファイル

\* On the retrieval of isomorphic graphs based on a partition algorithm by Hajime ENOMOTO, Takuya KATAYAMA and Naoki YONEZAKI (Department of Computer Science, Faculty of Engineering, Tokyo Institute of Technology)

\*\* 東京工業大学工学部情報工学科

の構成を与える。

### 2. 基礎的定義

ここで取り扱うグラフは、ループおよび多重辺をもたないグラフとする。まず、ラベル無し、無向グラフについて考える。ラベル付きあるいは有向グラフの場合への拡張は容易である<sup>10)</sup>。以下特に定義されないグラフ理論的用語は文献<sup>14)</sup>で用いられるのと同様に普通の意味で用いられる。

**定義 1** グラフ  $G(V, E)$  は、点の集合  $V$  と辺の集合  $E$  からなり、 $E$  は  $V \times V$  の部分集合である。 $(x, y) \in E$  ならば点  $x, y$  は隣接している。特にグラフ  $G$  の点集合、辺集合をそれぞれ、 $V(G), E(G)$  で表わす。

**定義 2**  $V(G), V(H)$  の間に隣接性を保存する全単射  $\phi$  が存在するとき、 $G, H$  は同型であるといい、 $G \cong H$  と書く。特に  $\phi$  が  $V(G)$  から  $V(G)$  への写像であるとき、そのような  $\phi$  を自己同型写像という。

**定義 3** グラフ  $V(G, E)$  において、 $x, y \in V(G)$  なる2点を結ぶ経路 (path) が存在すれば、その中で最短経路の長さを点  $x, y$  の距離といい、 $d(x, y)$  で表わす。

**定義 4** グラフ  $G$  において、点  $x$  から距離  $i$  にある点の集合  $R_i(x)$  を次のように定義する。

$$R_i(x) \triangleq \{y \mid d(x, y) = i, x, y \in V(G)\}$$

**定義 5**  $V(G)$  の  $k$  個のセルへの分割を  $P_{V(G)}^k$  と書き、各セルに含まれる点の集合を  $V'(G)^i (1 \leq i \leq k)$  で表わす。また、特に区別の必要の無い場合は、セル自身も  $V'(G)^i$  で表わす。

**定義 6**  $P_{V(G)}^k$  が次の条件を満たすとき、 $V(G)$  の自己同型分割という。  $1 \leq i, j \leq k, i \neq j$  なるすべての  $i, j$  について、 $x \in V'(G)^i$  かつ  $y \in V'(G)^j$  のとき、 $\phi(x) = y$  なる自己同型写像  $\phi$  が存在しない。

**定義 7** 自己同型分割  $P_{V(G)}^k$  が次の条件を満たすとき  $V(G)$  の完全自己同型分割という。  $1 \leq i \leq k$  なるすべての  $i$  について  $x, y \in V'(G)^i$  ならば、 $\phi(x) = y$  なる自己同型写像  $\phi$  が少なくとも1つ存在する。

**定義 8**  $x \in V(G)$  なるすべての  $x$  について、 $\Sigma = V(G) \cup \{ @, \textcircled{ } \}$  上の列、 $t(x)$  を次のように定義する。

$$t(x) \triangleq \alpha(R_1(x)) \cdot \prod_{2 \leq i \leq d} \{ @ \cdot \alpha(f(R_i(x))) \}$$

ここで、

- i)  $d$  は  $d \leq m(x)$  を満たす定数。ただし、 $G_*$  を  $x$  の連結成分とすると、 $m(x) = \bigcup_{1 \leq i \leq j} R_i(x) =$

$V(G_*) - \{x\}$  なる最小の  $j$ 。

- ii)  $\cdot$  は接続を表わす。
- iii)  $@, \textcircled{ }$  は  $V(G)$  に表われない特殊なシンボル。
- iv)  $\alpha(E)$  は集合  $E$  の要素を適当な順序で接続したものを表わす。
- v)  $f(R_i(x)) \triangleq \{y \cdot \alpha(p(x, y)) \cdot \textcircled{ } \mid y \in R_i(x)\}$  ただし、各  $y \in R_i(x)$  について、 $p(x, y) \triangleq \{z \mid (z, y) \in E(G) \wedge d(x, z) = i - 1\}$  とする。

(例-1) **Fig. 1** のグラフ  $G_1$  の点  $a, b$  について、列  $t(a), t(b)$  はそれぞれ次のようになる。

$$t(a) = bhg @ cb @ fg @ dc f @ ec f @$$

$$t(b) = cah @ ec @ dc @ gah @ fedg @$$

**定義 9**  $N$  を自然数の集合とすると、写像  $h: V(G) \rightarrow N$  を次のように定義する。

$$P_{V(G)}^k \text{ において、} x \in V'(G) \text{ ならば、} h(x) = j.$$

**定義 10** (辞書式順序)  $N$  を自然数の集合とすると、 $\Sigma' = N \cup \{ @, \textcircled{ } \}$  にアルファベット順序 ( $\leq$ ) を次のように定義する。

$$\forall i \in N \text{ について、} i \leq @, i \leq \textcircled{ } \text{ また } @ \leq \textcircled{ }.$$

$$i, j \in N \text{ について、} i \leq j \text{ ならば } i \leq j.$$

このとき、 $(\Sigma')^*$  に辞書式順序を次のように定める、 $\alpha, \beta \in (\Sigma')^*, \alpha = a_1 \cdot a_2 \cdots a_p, \beta = b_1 \cdot b_2 \cdots b_q$  について、 $\alpha \leq \beta$  であるのは、

- i)  $a_1 \leq b_1$
- ii)  $j \geq 2$  かつ  $a_j < b_j$  ならば、 $(\forall i, 1 \leq i < j)$  について、 $a_i = b_i$  なる  $j$  が存在する。
- iii)  $p \leq q$  かつ  $(\forall i, 1 \leq i \leq p)$  について、 $a_i = b_i$  のいずれかの場合である。

**定義 11**  $x \in V(G)$  なるすべての  $x$  について、 $\Sigma' = N \cup \{ @, \textcircled{ } \}$  上の列  $\bar{t}(x)$  を、次のように定義する。

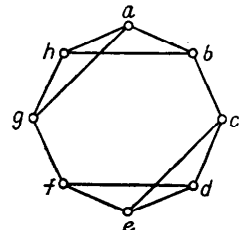
$$\bar{t}(x) \triangleq \alpha'(\{h(v) \mid v \in R_1(x)\}) \cdot \prod_{2 \leq i \leq d} \{ @ \cdot \alpha'(f(R_i(x))) \}$$

ここで、

- i)  $\alpha'(E)$  は集合  $E$  の要素を辞書式順序 ( $\leq$ ) に従って接続したものを表わす。
- ii)  $f(R_i(x)) \triangleq \{h(y) \cdot \alpha'(\{h(z) \mid z \in p(x, y)\}) \cdot \textcircled{ } \mid y \in R_i(x)\}$ 。
- iii) その他の定義は  $t(x)$  の場合と同じ。

### 3. 木同型分割とその性質

ここでは、木同型の概念の基礎となる木同型分



**Fig. 1** Graph  $G_1$

割を、グラフ  $G$  が与えられたとき  $V(G)$  を細分するアルゴリズム  $A$  により定義し、その計算の複雑さについて考察する。

また、木同型分割の種々の性質について論ずる。

### 3.1 木同型分割アルゴリズム

**定義 12** 木同型分割は、次のアルゴリズム  $A$  によって得られる分割  $P_{V(G)}$ \* である。

(アルゴリズム  $A$ )

**Step 1:**  $\forall x \in V(G)$  について  $t(x)$  を構成する。

**Step 2:**  $k=1$  とし、 $V^1(G)=V(G)$  とする。

**Step 3:**  $\forall x \in V(G)$  について  $\bar{i}(x)$  を求める。

**Step 4:**  $\forall x, y \in V(G)$  について  $\bar{i}(x), \bar{i}(y)$  が異なれば、 $x, y$  が異なるセルに入るように  $P_{V(G)}$ \* を細分する。その結果のセルの個数を  $k'$  とする。

**Step 5:**  $k' > k$  ならば  $k=k'$  として次へ。

$k'=k$  ならば終了。

**Step 6:**  $\bar{i}(x)$  の辞書式順序 ( $\leq$ ) にしたがって、分割されたセルを順序づけし、Step 3 へ。

(例-2) グラフ  $G_1$  に対する木同型分割の例を以下に示す。対応するステップをカッコをつけて示す。

(2)  $k=1, V^1(G_1)=V(G_1)=\{a, b, c, d, e, f, g, h\}$

(3)  $\bar{i}(a)=\bar{i}(d)=\bar{i}(e)=\bar{i}(h)$   
 $=111 @ 11 @ 11 @ @ 111 @ 111 @$   
 $\bar{i}(b)=\bar{i}(c)=\bar{i}(f)=\bar{i}(g)$   
 $=111 @ 11 @ 11 @ 111 @ @ 1111 @$

(4), (5)  $k'=2 > k, k=2$  とする。

(6)  $V^1(G_1)=\{b, c, f, g\}, V^2(G_1)=\{a, d, e, h\}$

(3)  $\bar{i}(a)=\bar{i}(d)=\bar{i}(e)=\bar{i}(h)$   
 $=112 @ 11 @ 11 @ @ 211 @ 211 @$   
 $\bar{i}(b)=\bar{i}(c)=\bar{i}(f)=\bar{i}(g)$   
 $=122 @ 122 @ 21 @ 21 @ @ 1122 @$

(4), (5)  $k'=k=2$ 。 終了。

### 3.2 アルゴリズム $A$ の手数および記憶領域

アルゴリズム  $A$  の計算の手数および必要な記憶領域が  $N$  の多項式のオーダーであることを示す。

**定理 1** 最悪の場合、アルゴリズム  $A$  は  $O(N^3)$  の記憶領域を必要とする。ただし、 $\|V(G)\|^* = N$ 。

**証明**  $\|R_i(x)\| = a_i$  とすると  $t(x)$  の長さ  $L$  は次式を満足する。

$$L \leq a_1 + \sum_{i=1}^{m(x)-1} (a_i + 2) \cdot a_{i+1} + m(x) - 1$$

$$= 2 \sum_{i=1}^{m(x)} a_i + \sum_{i=1}^{m(x)-1} a_i \cdot a_{i+1} + m(x) - 1 - a_1$$

ここで、 $\sum_{i=1}^{m(x)} a_i = N - 1, m(x) \leq N - 1$  である。

ところで、 $\sum_{i=1}^{m(x)-1} a_i \cdot a_{i+1} \leq N^2/4$  となるから、

$$L \leq 3N - 5 + N^2/4 = O(N^2).$$

したがって、 $\forall x \in V(G)$  に対する  $t(x)$  に必要な記憶領域は、

$$S = \sum_{x \in V(G)} |t(x)|^{**} \leq N \times L = O(N^3)$$

となる。 $\bar{i}(x)$  および  $\{h(v) | v \in R_i(x)\}, f(R_i(x))$  ( $2 \leq i \leq d$ ) のソートに必要な記憶領域は  $O(S)$  であり<sup>13)</sup>、また  $V^i(G)$  ( $1 \leq i \leq k$ ) の記憶領域は  $O(N)$  であるから、アルゴリズム  $A$  に必要な記憶領域は  $O(N^3)$  を越えない。ところで、Fig. 2 のような完全2部グラフ (complete bipartite graph) を考えれば、 $\forall x \in V(G_2)$  について、

$$|t(x)| = N/2 + (N/2 + 2)(N/2 - 1) + 1 = O(N^2)$$

となるから、アルゴリズム  $A$  は最悪の場合  $O(N^3)$  の記憶領域を必要とする。 (証明終)

**定理 2** 最悪の場合でも、アルゴリズム  $A$  は  $O(N^3)$

以下の時間しか必要としない。

**証明** i) Step 1 の手数

グラフ  $G$  が、各点に対して、それと隣接する点をリストの形で与えたもので表現されているとすると、 $t(x)$  を求めるのに、 $x$  からリストをたどって行き、すでに通過した点にフラッグを立てる方法をとれば、 $O(N^2)$  時間以内で  $t(x)$  は求まる。 $\forall x \in V(G)$  について  $t(x)$  を求めるのは、 $O(N^3)$  時間以内で求められる。

ii) Step 3 の手数

$\sum_{x \in V(G)} |t(x)| \leq O(N^3)$  であるから  $O(N^3)$  時間。

iii) Step 4, 6 の手数

異なる長さの列の辞書式ソートは、列の長さの和を  $l_{total}$ 、表われるシンボルの種類を  $m$  とすると、 $O(l_{total} + m)$  時間で行われる<sup>13)</sup>。ここでは、 $m = O(N)$  となるから、 $\bar{i}(x)$  中の  $f(R_i(x))$  のソートは  $O(N^2)$  時間で行え、またセルの順序付けについては、

$$l_{total} = \sum_{x \in V(G)} |t(x)| = O(N^3)$$

であるから  $O(N^3)$  時間で行える。ところで細分は高々  $N$  回しか行われないから全体で、ア

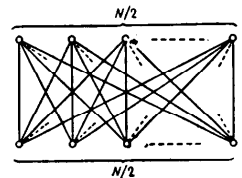


Fig. 2 Complete bipartite graph  $G_2 = KCN/2, N/2$

\* |E| で集合  $E$  の要素数を表わす。

\*\* |t| で列  $t$  の長さを表わす。

ルゴリズムは  $O(N^4)$  以下の時間しか必要としない。

(証明終)

3.3 木同型分割の性質

定理 3 木同型分割は自己同型分割である。

証明 分割の手続き Step 4 の実行回数  $i$  に関する、次のような帰納法で証明される。木同型分割の結果が  $P_{V(G)^k}$  であるとし、 $P_{V(G)^k}$  に至るまでに、アルゴリズム  $A$  の Step 4 が  $l$  回実行されたとする。  $0 \leq i < l \leq k$  について、

i)  $i=0$  のとき、すなわち分割が行われる前には、 $V(G)$  は 1 つのセルに入っている ( $k=1$  である)。から、明らかに自己同型分割である。

ii)  $i=m$  のとき、すなわち Step 4 が  $m$  回実行されたとき、その分割が自己同型分割になっているとすれば、 $m+1$  回目の分割でも、その結果は自己同型分割になっていることを示す。  $m+1$  回目の分割は、 $m$  回目の分割の細分になっており、セルが細分されるのは次の場合である。

(a)  $\bar{i}(x), \bar{i}(y)$  の @, ⊗ の位置に一致しないものがある場合。

ソートの一意性により、 $x, y$  の間には  $G$  の自己同型写像は存在しない。

(b)  $\bar{i}(x), \bar{i}(y)$  の @, ⊗ の対応はあるが、セル名の対応がない場合。

セル名が一致しないのは、 $m$  回目の細分の結果、 $\bar{i}(x), \bar{i}(y)$  中で同位置にあるシンボルに対する点で異なるセルに入っている点が存在したため、このときは帰納法の仮定により、これらの点の間には自己同型対応はない。ソートの一意性より、 $\phi(x)=y$  となる自己同型写像  $\phi$  が存在するとすると、これらの点を対応づけるものでなければならないから、 $x, y$  の間には  $G$  の自己同型対応は存在しない。  $m+1$  回目の細分では、(a), (b) の場合  $x, y$  は異なるセルに入るから、その結果も自己同型分割となっている。 i), ii) より、木同型分割は、自己同型分割である。(証明終)

定理 4  $\bar{i}(x), \bar{i}(x)$  において  $\forall x \in V(G)$  について  $d=m(x)$  とし、 $p(x, y)$  リストを無視したときの、アルゴリズム  $A$  による分割は、必ずしも完全自己同型分割にはならない。

証明 そのような例を Fig. 3 に示す。

(a)  $p(x, y)$  リストを無視すると、

$$V^1(G_3) = \{a, c, e, g\}$$

$$\forall x \in V^1(G_3) \text{ について}$$

\* 集合  $E$  に対して  $h(E)$  で  $\{h(e) | e \in E\}$  を表わす。

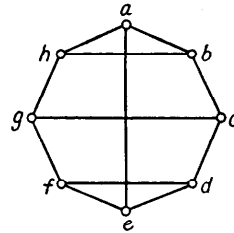


Fig. 3 Graph  $G_3$ : Example of Theorem 4

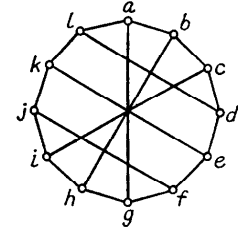


Fig. 4 Graph  $G_4$ : Example of Theorem 5

$$h(R_1(x))^* = \{1, 2, 2\}$$

$$h(R_2(x)) = \{1, 1, 2, 2\}$$

$$V^2(G_3) = \{b, d, f, h\}$$

$\forall y \in V^2(G_3)$  について

$$h(R_1(y)) = \{1, 1, 2\}$$

$$h(R_2(y)) = \{1, 1, 2\}$$

$$h(R_3(y)) = \{2\}$$

(b)  $p(x, y)$  リストを無視しない場合

この場合 Fig. 3 の例では完全自己同型分割になっている。

$$V^1(G_3) = \{c, g\}$$

$\forall x \in V^1(G_3)$  について

$$h(R_1(x)) = \{1, 2, 2\}$$

$$f(R_2(x)) = \{2 \cdot 1 \cdot 2 \cdot \textcircled{c},$$

$$2 \cdot 1 \cdot 2 \cdot \textcircled{g}, 3 \cdot 2 \cdot \textcircled{c}, 3 \cdot 2 \cdot \textcircled{g}\}$$

$$V^2(G_3) = \{b, d, f, h\}$$

$\forall y \in V^2(G_3)$  について

$$h(R_1(y)) = \{1, 2, 3\}$$

$$f(R_2(y)) = \{1 \cdot 1 \cdot 2 \cdot \textcircled{b}, 2 \cdot 1 \cdot \textcircled{d}, 3 \cdot 3 \cdot \textcircled{f}\}$$

$$f(R_3(y)) = \{2 \cdot 1 \cdot 2 \cdot 3 \cdot \textcircled{h}\}$$

$$V^3(G_3) = \{a, e\}$$

$\forall z \in V^3(G_3)$  について

$$h(R_1(z)) = \{2, 2, 3\}$$

$$f(R_2(z)) = \{1 \cdot 2 \cdot \textcircled{a},$$

$$1 \cdot 2 \cdot \textcircled{e}, 2 \cdot 3 \cdot \textcircled{a}, 2 \cdot 3 \cdot \textcircled{e}\}$$

(証明終)

定理 5 距離を制限したときの分割、すなわち  $\bar{i}(x), \bar{i}(x)$  において、 $\forall x \in V(G)$  について  $d < m(x)$  としたときの木同型分割は必ずしも完全自己同型分割にはならない。

証明 そのような例を Fig. 4 に示す。

(a) 距離=3 に対するリストを無視した場合

$$(d=2)$$

$$V^1(G_4) = \{b, e, h, k\}$$

$$V^2(G_4) = \{a, c, d, f, g, i, j, l\}$$

(b) 距離=3 (最大距離) まで考慮した場合  
( $d=3$ )

$$\begin{aligned} V^1(G_i) &= \{e, k\} \\ V^2(G_i) &= \{b, h\} \\ V^3(G_i) &= \{d, f, j, l\} \\ V^4(G_i) &= \{a, c, g, i\} \end{aligned}$$

距離=2 までの分割では、このグラフの完全自己同型分割になっていないが、最大距離=3 までの分割では、完全自己同型分割になっている。(証明終)

定理 4, 5 より,  $t(x)[d=m(x)]$  に含まれる情報は、完全自己同型分割を求めるためには、すべて必要であることがわかる。

#### 4. 木同型性

本章では、木同型分割をもとに木同型を定義し、その種々の性質について論ずる。

**定義 13** グラフ  $G$  に対して、列  $Q(G)$  を次のように定義する。いま  $\forall x \in V(G)$  について  $d=m(x)$  とした場合の木同型分割を  $P_{V(G)}^*$  とする。  $1 \leq i \leq k$  について、  $\|V^i(G)\| = n_i, x \in V^i(G)$  ならば  $t(i) = \overline{t(x)}$  とすると、  $Q(G) \triangleq \{n_i \cdot \overline{t(i)}\}_{1 \leq i \leq k}$

**定義 14** (木同型性)

グラフ  $G, H$  は次の条件が満たされるとき木同型であるといい、  $G \stackrel{!}{\cong} H$  と書く。

$$G \stackrel{!}{\cong} H \Leftrightarrow Q(G) = Q(H)$$

##### 4.1 木同型判定の手数および記憶領域

**定理 6** 任意のグラフのクラスについて木同型判定の手数と記憶領域のオーダーは、木同型分割のそれに等しい。

証明 “ $Q(G)=Q(H)?$ ” の判定には、  $O(|Q(G)| + |Q(H)|)$  の手数と記憶領域しか必要としない。また、木同型分割の記憶領域は、定理 1 と同様に  $O(|Q(G)| + |Q(H)|)$  であり、また手数は、定理 2 と同様に  $O(|Q(G)| + |Q(H)|)$  より小さくない。ところで、木同型判定の手数は、木同型分割の手数と、“ $Q(G)=Q(H)?$ ” の手数の和であるから題意が成立する。(証明終)

##### 4.2 木同型の性質

**補題 1**  $G \stackrel{!}{\cong} H \Rightarrow G \stackrel{!}{\cong} H$

証明  $G$  に対する木同型分割が、一意に定まることより明らか。(証明終)

**定理 7** 木同型分割が完全自己同型分割を与えるならば、次式が成立する。

$$G \stackrel{!}{\cong} H \Leftrightarrow G \stackrel{!}{\cong} H.$$

証明 グラフ  $G, H$  の木同型分割をそれぞれ、  $P_{V(G)}^*, P_{V(H)}^*$  とする。

補題 1 より条件が成立するなら、  $G \stackrel{!}{\cong} H \Rightarrow G \stackrel{!}{\cong} H$  を示せばよい。いま、一般性を問わずに  $V(G) \cap V(H) = \emptyset$  と仮定し、  $G \cup H$  で  $(V(G) \cup V(H), E(G) \cup E(H))$  を表わすとする。ここで  $G \stackrel{!}{\cong} H$  とすると定義 14 より、  $Q(G)=Q(H)$  であり、  $k_1=k_2$  で各セルに付属するリストもそれぞれ等しく、また  $G, H$  が非連結であることより、  $G \cup H$  の木同型分割  $P_{V(G \cup H)}^*$  は、

$$P_{V(G \cup H)}^* = \{V^1(G) \cup V^1(H), V^2(G) \cup V^2(H), \dots, V^{k_1}(G) \cup V^{k_1}(H)\}$$

となる。ところで条件より  $P_{V(G \cup H)}^*$  は完全自己同型分割となり、セル  $V^i(G \cup H) = V^i(G) \cup V^i(H)$  ( $1 \leq i \leq k$ ) 中の任意の 2 点間には、同型写像  $\phi$  が少なくとも 1 つ存在する。  $x \in V^i(G), y \in V^i(H)$  について、  $\phi(x)=y$  を考えれば、これは  $G, H$  間の同型写像になっている。したがって条件が成立するなら、  $G \stackrel{!}{\cong} H \Rightarrow G \stackrel{!}{\cong} H$ 。(証明終)

#### 5. 特定のグラフのクラスに対する木同型性

本章では、特定のグラフのクラスに対する木同型判定アルゴリズムの計算の複雑さ、及び同型性との関連について述べる。

##### 5.1 tree (cycle なしグラフ)

**定理 8**  $G, H$  が tree であれば、その(木)同型判定に必要な手数と記憶領域は、それぞれ高々  $O(N^3), O(N^2)$  である。

証明 定理 6 より、木同型分割の複雑さを調べればよい。  $G$  が tree であれば、  $(\forall i, 1 \leq i \leq m(x)) (\forall y \in R_i(x)) [\|p(x, y)\| = 1]$  であるから、  $\forall x \in V(G)$  について  $|t(x)| = O(N)$ 。したがって、必要な記憶領域のオーダーは

$$\sum_{x \in V(G)} |t(x)| = O(N^2).$$

また細分は高々  $N$  回しか起こらないから、判定の手数の上限は、  $O(N^3)$  となる。(証明終)

**定理 9**  $G$  が tree であれば、  $G \stackrel{!}{\cong} H \Leftrightarrow G \cong H$ ,

証明  $\forall x \in V(G)$  について  $d=m(x)$  とした場合の  $G$  の木同型分割  $P_{V(G)}^*$  の適当なセル  $V^i(G)$  について、  $x \in V^i(G)$  を根とする tree  $T$  を、  $Q(G)$  より次のように構成する。

**Step 1:**  $x$  を根とし、  $R_0(x)=x, j=1$  とする。

**Step 2:**  $j > m(x)$  ならば終了。それ以外ならば次へ。  $R_{j-1}(x)$  の各点  $z$  に、  $\{h(y) | y \in R_1(z)\} -$

$\{h(A(z)^*)\}$ の点を接続し、 $h(y)$ でラベルづけする。

**Step 3:**  $j=j+1$  として、Step 2へ。

ところで、 $Q(G)$ より  $\|V(G)\|, \|E(G)\|$ , また  $G$ が連結グラフであるかを決定でき、 $G$ が tree であるための必要十分条件は、 $\|V(G)\| - \|E(G)\| = 1$  でありかつ連結グラフであることであるから、tree  $G$  から得られる  $Q(G)$  は tree 以外のグラフからは得ることができない。また、 $\forall x, y \in V^i(G)$  について、 $V^i(G)$  は木同型分割のセルであるから ( $\forall j, 1 \leq j \leq m(x) = m(y)$ )  $[\{h(z) | z \in R_j(x)\} = \{h(z) | z \in R_j(y)\}]$  となり、 $Q(G)$  は tree 以外のグラフからしか得られないので、 $R_j(x)$  中の点の間には隣接関係は存在しない。したがって、上記の構成方法によれば、 $\forall x \in V^i(G)$  を根とする tree は一意に定まり、全て同型となることがわかる。よって  $G$  が tree であれば、 $\forall i (1 \leq i \leq k) \forall x, y \in V^i(G)$  について、点  $x, y$  の間には自己同型写像が存在し、 $V(G)$  の木同型分割は完全自己同型分割となる。ところで  $G$  が tree であり  $Q(G) = Q(H)$  であれば、前述のように  $H$  は tree であり、 $H$  の木同型分割は完全自己同型分割となる。よって定理 7 より題意が成立する。(証明終)

### 5.2 完全グラフ

**定理 10**  $G, H$  が完全グラフであれば、(木)同型判定に必要な手数と記憶領域は、ともに  $O(N^2)$  である。

証明 完全グラフは、すべての点が距離 1 で到達可能であるから、 $\forall x \in V(G)$  について、 $|t(x)| = O(N)$ 、また分割は起こらないので、手数、記憶領域とも、 $\sum_{x \in V(G)} |t(x)| = O(N^2)$  となる。(証明終)

**定理 11**  $G$  が完全グラフであれば、 $G \stackrel{!}{=} H \Leftrightarrow G \cong H$ 。

証明 完全グラフ  $G$  から得られる  $Q(G)$  からは、ただ 1 つのグラフしか再構成できないことを示せばよい。 $G$  の木同型分割は、 $V(G)$  そのものであり、すべての点を含んでいる。また、 $Q(G)$  より、各点が隣接している点の数が  $N-1$  であることがわかるので、この条件を満足するグラフは  $N$  点の完全グラフのみである。したがって、 $N$  点の完全グラフ  $G$  より得られる  $Q(G)$  は、 $N$  点の完全グラフ以外からは得られない。(証明終)

### 5.3 多角形 (simple polygon)

このグラフのクラスは Corneil の方法<sup>9)</sup>では、同型

判定に  $O(N^4)$  時間を要し、このグラフのクラスは、2-強正則な推移部分グラフ<sup>9), 12)</sup>を含んでいないグラフに対して  $N$  の最も悪い依存度を持つことが予想されている。

**定理 12**  $G, H$  が多角形であれば、(木)同型判定に必要な手数と記憶領域はともに  $O(N^2)$  である。

証明  $\forall x \in V(G)$  について、 $|t(x)| = O(N)$  となり、すべての点の間に自己同型写像が存在するので分割は起こらず、定理 10 の場合と同様に、手数、記憶領域ともに  $O(N^2)$  となる。(証明終)

**定理 13**  $G$  が多角形であれば、 $G \stackrel{!}{=} H \Leftrightarrow G \cong H$ 。

証明  $Q(G)$  からは、 $G$  の各点の次数、 $\|E(G)\|$ 、連結グラフであるか否かが決定できる。 $G$  が多角形であれば、各点の次数は 2 であり、 $\|E(G)\|$  は  $N$  に等しく、連結グラフである。このようなグラフは、 $N$  多角形以外に存在しないことは簡単に示せる。(証明終)

### 5.4 完全 2 部グラフ (complete bipartite graph)

**定理 14**  $G, H$  が完全 2 部グラフであれば、(木)同型判定に必要な手数と記憶領域は、高々  $O(N^3)$  である。

証明 完全 2 部グラフにおいて、隣接関係のない点どうしの集合をそれぞれ、 $V_1, V_2$  とする。

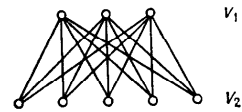


Fig. 5 Complete bipartite graph  $G_5 = K(3, 5)$

$V(G) = V_1 \cup V_2$ . Fig.

5 に  $K(3, 5)$  の完全 2 部グラフを例として示す。

今、 $\|V_1\| = a$  とすると、 $V(G)$  の分割のセルは高々 2 個であるから、

$$\sum_{x \in V(G)} |t(x)| = \{N - a + 1 + (a - 1)(N - a + 2)\} \cdot a + \{a + 1 + (N - a - 1)(a + 2)\} (N - a) = O(N^3)$$

また  $\|V_1\| = \|V_2\|$  ならば、細分は行われず  $\|V_1\| \times \|V_2\|$  ならば、分割は一度しか行われぬ。したがって、判定の手数の上限も  $O(N^3)$  となる。(証明終)

とここで、2 部グラフについては、次の定理が成立することが知られている。

**定理 15** グラフが 2 部グラフであるための必要十分条件は、どのサイクルの長さも偶数であることである。(証明略)

**補題 2**  $G$  が 2 部グラフかどうかは、 $Q(G)$  より決定することができる。

証明  $G$  のすべてのサイクルが偶数であるための必

\* tree  $T$  において  $R_{j-1}(x)$  中で  $z$  が隣接している点を  $A(z)$  で表す。ただし  $R_{-1}(x) = \emptyset$  とする。

要十分条件は、 $\forall x \in V(G)$ ,  $\forall i(1 \leq i \leq m(x))$  について、 $R_i(x)$  中の点の間に隣接関係が存在しないことである。距離  $i$  の点から  $i+1$  の点へ至る辺の数、また距離  $i-1$  の点から  $i$  の点へ至る辺の数、および  $R_i(x)$  中の点の次数は  $Q(G)$  より決定されるから、 $R_i(x)$  中の点の間の隣接関係が存在するか否かは、 $Q(G)$  より決定できる。以上と定理 15 より補題が成立する。

(証明終)

**定理 16**  $G$  が完全 2 部グラフであれば、

$$G \cong H \iff G \cong H$$

証明  $G$  が完全 2 部グラフ  $K(m, n)$  であったとすると、その木同型分割は、 $m \neq n$  ならば  $\|V(G)\| = m, \|V^2(G)\| = n$  なる 2 つのセルとなり、 $m = n$  ならば、セルは  $V(G)$  自身のみとなる。ところで、補題 2 より  $Q(G)$  から  $G$  が 2 部グラフであるかは決定され、また  $\|E(G)\|$  が決められるので、“ $\|E(G)\| = m \cdot n$ ?” を調べれば、完全 2 部グラフであるかがわかる。

したがって、 $G = K(m, n)$  から得られる  $Q(G)$  からは、 $K(m, n)$  しか再構成されない。(証明終)

### 5.5 unicycle グラフ

これは、サイクルをちょうど 1 つ持つグラフで、tree 以外のグラフで Ulam の conjecture が成立することが証明されている唯一のグラフのクラスである。

**定理 17**  $G, H$  が unicycle グラフであれば、その(木)同型判定に必要な手数と記憶領域は、それぞれ  $O(N^3), O(N^2)$  である。

証明  $G$  がサイクルを 1 つしか持たなければ、 $(\forall i, 1 \leq i \leq m(x))(\forall y \in R_i(x))\|p(x, y)\| \leq 2$  であるから、 $\forall x \in V(G)$  について  $|t(x)| = O(N)$ 。

また、細分は高々  $N$  回しか起こらない。(証明終)

**定理 18**  $G$  が unicycle グラフであれば、

$$G \cong H \iff G \cong H$$

証明  $G$  が連結グラフであれば、点の数と辺の数が等しいときのみ unicycle グラフである。 $Q(G)$  より、 $G$  の連結性、 $\|V(G)\|, \|E(G)\|$  は決定されるから、 $G$  が unicycle グラフであるかは決定できる。unicycle グラフであれば、サイクル上の点集合は次のように決定できる。

i) サイクルの長さが奇数のとき。

この場合、どの点から見てもある等距離にある点の間に、唯一隣接しているものが存在する。 $Q(G)$  より  $R_i(x)$  中の点の間に、隣接関係が存在するかどうかを決定できる。そのような隣接する等距離の点が、最小

距離上にあるものがサイクル上の点であるから、サイクル上の点と、サイクル上にない点は必ず異なるセルに入り、そのようなセル名を  $Q(G)$  より決定できる。

ii) サイクルの長さが偶数のとき。

この場合、 $\forall x \in V(G)$  について、 $t(x)$  中に  $\|p(x, y)\| = 2$  なる  $y$  が必ず 1 つ表われる。 $y \in R_i(x)$  とすると、サイクル上の点  $x$  に対する  $i$  の値は、サイクル上にない点  $x'$  に対する  $i$  の値より必ず小さい。したがって、最小の  $i$  を持つ  $x$  を含むセルが、サイクル上の点を含むセルであり、サイクル上にない点は、サイクル上の点と同一セル中に含まれない。

i), ii) より、 $Q(G)$  からサイクル上の点と、サイクル上にない点が判定できると、定理 9、定理 13 より、tree, サイクルはそれぞれ一意に決定でき、その隣接位置は、 $Q(G)$  からわかるので unicycle グラフを唯一、再構成できる。(証明終)

### 5.6 その他のグラフのクラス

以上いくつかのグラフのクラスに対して木同型が同型となることを示した。もちろんリジッドなグラフ、すなわち、木同型分割の各セルがただ 1 つの点しか含まないようなグラフのクラスについても木同型は同型となり、またほとんどのグラフに対して木同型が同型となることが実験的に予想されている\*。しかし特殊な例として  $k$ -強正則なグラフ ( $k \geq 2$ )<sup>4), 12)</sup> のクラス中の一部のグラフに対しては、木同型分割が完全自己同型分割とならないものが存在する。例えば、26 個の点よりなる 2-強正則グラフ<sup>12)</sup> がそのような例となる。

## 6. バックトラック・アルゴリズムへの応用

一般のグラフに対しては木同型  $\equiv$  同型は成立しないので、厳密な同型性を調べるにはバックトラック・アルゴリズムが必要となる。この場合、同型対応の検索木をヒューリスティックに小くするかが問題となる。 $G \cong H \Rightarrow G \cong H$  であり、またほとんどのグラフに対して木同型が同型となることが予想されるので、木同型分割の結果は非常によいヒューリスティックな基準になりうる。

木同型分割を用いて、 $G_1, G_2$  の同型対応を調べるバックトラック・アルゴリズムを次に示す。

### (アルゴリズム B)

**Step 1:** 初期分割として、 $V(G_1), V(G_2)$  の木同型分割を行う。

\* 16 bit, 32 kword のマシン上で、 $N \leq 25$  のランダムなグラフに対する実験による。

**Step 2:**  $Q(G_1) \neq Q(G_2)$  であれば,  $G_1, G_2$  は同型でない。  $Q(G_1) = Q(G_2)$  ならば  $i=1$  として次へ。

**Step 3:**  $Pv(G_1)^i, Pv(G_2)^i$  の対応する複数個の点を含むセルから点を1つずつ選び, それぞれ  $x_1^i, x_2^i$  とする。  $x_1^i, x_2^i$  をそれぞれその点のみを含む独立なセルに入れ木同型分割を行い, セルを細分する。

**Step 4:** その結果,  $Q(G_1) = Q(G_2)$  ならば,  $x_1^i, x_2^i$  を同型対応づけ,  $i=i+1$  として Step 7 へ。  $Q(G_1) \neq Q(G_2)$  ならば次へ。

**Step 5:**  $x_2^i$  を選び直すために Step 3 へ。 対応するセルに候補となる  $x_2^i$  がこれ以上存在しない場合は,  $i=i-1$  として次へ。

**Step 6:**  $i=0$  ならば,  $G_1, G_2$  は同型でない。  $i>0$  ならば Step 5 へ。

**Step 7:** すべてのセルが1つの点よりなる場合は,  $G_1, G_2$  は同型。 そうでなければ Step 3 へ。

Schmidt, Druffel<sup>9)</sup> のバックトラック・アルゴリズムにおける分割に用いられる情報は,  $i(x)$  にすべて含まれるので, 木同型分割は, 9)における分割の細分を与えることになる。 したがって探索木の高さを小さくすることができる。 また探索木の枝分れにおいて, セルの要素数の小さいセルから点を選ぶ戦略をとれば, 探索木の幅を小さくすることができる。

計算時間の上限は,  $N$  の指数関数となるが, ほとんどのグラフに対しては, バックトラックを必要としないことが実験により確かめられた。

## 7. 同型グラフの検索

本章では, グラフとして表現される構造を持つデータが大量に存在する場合, その中から同型なグラフを効率的に検索するためのファイル構成法, およびその検索について述べる。

全サーチによる方法は非現実的であるから, 適当なインデックスを導入する必要があるが, グラフの標準形\*に対する線形表現\*\*をそのキーとする方法も, その変換の手数を考えれば非現実的である。

前章まで述べてきたように, ほとんどのグラフに対して,  $d=m(x)$  まで考慮した木同型分割は完全自己同型分割となっていることが予想されている。 さらに定義 13 において  $d < m(x)$  とした場合も同型なグラ

フは必ず同じ  $Q(G)$  の値を持つ。 またその値は,  $d=m(x)$  に対する  $Q(G)$  の値を得る過程で得ることができ, ほとんどの場合  $O(N^3)$  以下の時間で求めることができることが実験的に確かめられた。 そこで, このような  $Q(G)$  グラフ  $G$  に対する同型グラフの検索のためのキーとするインデックス・シーケンシャルなファイル構成をとることにする。 またもちろん,  $d=m(x)$  について  $Q(G)=Q(H)$  であっても,  $G \cong H$  となることがあるので,  $Q(G)=Q(H)$  を  $G, H$  に対するハッシュ値と見なし,  $G, H$  に関するデータレコードは順に並べておくことにする。 もちろん, あふれが起こる場合は, あふれ領域の処理が必要となる。

**Fig. 6** (次頁参照) にそのファイル構成を示す。

このファイル構成では, 多段のインデックス・エリアを持ち, 簡単のために, 各レコードは必ず1トラックにおさまるものとしている。 レコードはキーの辞書式順序に従って順に並んでいる。 またレコードの間にすき間はなく, あふれはオーバフロー・エリアで処理する。 インデックス・エリアには, シリンダー番号, トラック番号が書かれており, その順序はキーの辞書式順序に従っている。 またレコードもその順序に従っているので, インデックス・エリアにはそのサブ・インデックス・エリアのすべてのレコードを記録しておく必要はない。

グラフ  $G$  と同型なグラフを検索するには,  $Q(G)$  を  $\forall d(1 \leq d \leq M(G)^*)$  について求め, マスター・インデックスを読み出す。 そして2分法により,  $M(G)$  と同じ数のキーを見つける。 もし同じものが見つからない場合は, その直前, 直後のキーを見つけ, そのポインタの指す間のサブ・インデックスのレコードを主記憶に読み込む。 再び2分法により,  $d=1$  についての  $Q(G)$  と同じリストまたは, 直前, 直後のリストを見つけ, そのポインタが示すサブ・インデックスを読み込み, 同様の操作を距離  $M(G)$  まで行う。 このようにして, 目的のデータ・レコードが得られたら, アルゴリズム  $B$  を用いて同型判定を行えばよい。 この場合, 木同型分割はすでに得られているので, Step 3 より実行すればよい。  $\forall x \in V(G)$  について  $d=m(x)$  とした  $Q(G)$  と同じリストを持つグラフが複数個存在するときには, リストをたどって順にあふれ領域のデータについて調べることになるが, 5. で述べたようにこのような例は, 非常に少ないことが実験的に予想される。 以上のように木同型分割をもとにしたファイル構成をしておけば, 同型グラフの検索においてサーチす

\* 与えられたグラフ  $G$  と同型で, 次のような点の順序づけのなされたグラフ  $G_c$  をグラフの標準形という。 すなわち,  $G, H$  が同型であれば, またそのときに限り,  $G_c, H_c$  の同じ順序のついた点の間に同型対応がつく。

\*\* 点の順序づけに従って  $G$  の隣接行列を並べかえその各列を順に一列に連接したものの。



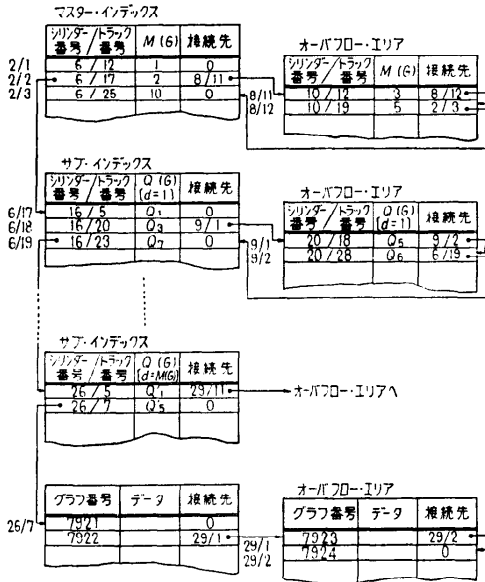


Fig. 6 File organization for a retrieval of isomorphic graphs

べきデータの量を制限することができる。

8. むすび

グラフとして表現される構造を持つデータが大量に存在するとき、指定された構造と同様な構造を効率的に検索する方法について述べた。まず点の数の多項式時間で計算される木同型分割を基に、木同型と呼ばれる弱い同型性を定義し、いくつかの特徴的なグラフのクラスに対して木同型が同型となることを示した。また同型性を調べる木同型分割を用いるバックトラック・アルゴリズムを与えたが、探索木を非常に小さくできるという点で効果的であることが実験的に確かめられた。以上の準備の下に、同型構造の検索に対するファイル構造を与えたが、ここでも木同型分割の結果が効果的に用いられることが確かめられた。

なお、本研究を行うにあたり、日ごろ熱心に御討論頂いた本学榎本・片山研究室の諸氏に感謝します。

参考文献

1) Snolenski, Y. A.: A method for linear recording of graphs. USSR Comput. Math. and Math. of Phys. 2. pp. 396~397 (1963)

2) Unger, S.H.: GIT-A heuristic program for testing pairs of directed line graphs for isomorphism. C. ACM. 7. 1. pp. 26~34 (1964)

3) Busacker, R., and Saaty, T.: Finite Graphs and Networks—An introduction with Applications. p. 196. McGraw-Hill New York. (1965)

4) Corneil, D. G. and Gotlieb, C. C.: An efficient algorithm for graph isomorphism. J. ACM. Vol. 17, No. 1, pp. 51~64 (1970)

5) Bertiss, A. T.: A Backtrack procedure for isomorphism of directed graphs. J. ACM. Vol. 20, No. 3. pp. 365~377 (1973)

6) Yang, C. C.: Structural preserving morphism of finite automata and an application to graph isomorphism. IEEE Trans. on Comput. C-24, 11, pp. 1133~1139 (1975)

7) Ullman, J.R.: An algorithm for Subgraph isomorphism. J. ACM. Vol. 23, No. 1, pp. 31~42 (1976)

8) 田村浩一郎: グラフの同型アルゴリズムの一方 法, 昭和 51 年度, 電子通信学会総会全国大会. 6-154 (1976)

9) Schmidt, D.C. and Druffel, L. E.: A Fast Backtracking Algorithm to test directed graphs for isomorphism using distance matrices. J. ACM Vol. 23, No. 3, pp. 433~445 (1976)

10) Hopcroft, J.E., and Tarjan, R.E.: Isomorphism of planar graphs. In Complexity of Computer Computations. R. E. Miller and J. W. Thatcher, Eds., p. 143. Plenum Press. New York (1972)

11) Hopcroft, J. E., and J. K. Wong: Linear time algorithm for isomorphism of planer graphs. PROC OF 6th ANNUAL ACM SYMPOSIUM ON THEORY OF COMPUTING, pp. 172~184

12) Bose, R. C.: Strongly regular grophs, partial geometries, and partially balanced designs. Pacific J. Math. 13, pp. 389~420 (1963)

13) AHO. Hopcroft. Ullman: The design and Annalysis of computer algorithms. Addison-Wesley (1974)

14) F. Harary: Graph theory. Addison-Wesley (1969)

15) 榎本 肇, 片山卓也, 米崎直樹, 宮地利雄: 構造検索のためのアルゴリズムとそのファイル構造, 電子通信学会パターン認識と学習研究会資料 PRL 75-58 (1975)

16) 榎本 肇, 米崎直樹: グラフの同型判定のためのアルゴリズムとその性質. 電子通信学会オートマトンと言語研究会資料, AL 76-49 (1976)

(昭和 51 年 11 月 29 日受付)  
(昭和 52 年 4 月 28 日再受付)

$$*M(G) = \max_{x \in V(G)} \{m(x)\}$$