



1対1対応を部分的に持つグラフ間の同形判定に関する一手法*

久保 登** 白川 功*** 尾崎 弘***

Abstract

Given two graphs, for which a partial one-to-one correspondence between vertices is specified, of practical use is the problem of whether or not there exists an isomorphism between them which is an extension of the partial one-to-one correspondence.

This article proposes an algorithm for this problem and then shows a part of the implemented results which reveal that the proposed algorithm can be efficiently applied to practical problems, such as the fault detection for an interconnection pattern of LSI chips.

1. ま え が き

二つのグラフが与えられたとき、それらが同形であるかどうかを判定する問題（同形判定問題）に対しては、多くの研究が行われてきたが^{1)~9)}、節点数が数百ないし千を越えるグラフに対しても適用できるような効率のよいアルゴリズムは見当たらないようである。

ところで、実際の同形判定問題の応用として取り扱うグラフでは、その判定を容易にするような何らかの情報があらかじめ与えられている場合が少なくないと考えられる。たとえば、論理設計で得られた回路図とそれを実現すべきLSIチップのレイアウトパターンとの間での結線構造の検査において、あらかじめいくつかの端子の間には1対1の対応が定められているという条件が付加されている。

本文では、このような条件下での同形判定問題を考察し、一つのアルゴリズムを提案するが、ランダム正規グラフについての実験結果から大規模なグラフに対しても十分適用できることが確認された。

2. 基本的考察

節点集合 V 、枝集合 E からなるグラフを $G=[V, E]$ で表わす。本文では、連結無向グラフを対象とするが、有向グラフや節点（または枝）にラベルの付いたグラフなどへの一般化は容易に行うことができる。以下では、二つのグラフを区別する必要のある場合には、 G^1, G^2 など上つき添字を用いる。

節点 u, v を両端点とする枝 e を非順序対 $e=\langle u, v \rangle$ で表わし、節点 v の隣接節点集合 $\Gamma(v)$ を次式で定義する。

$$\Gamma(v) \triangleq \{u | \langle u, v \rangle \in E\} \quad (1)$$

節点 v の線度 $d(v)$ を

$$d(v) \triangleq |\Gamma(v)| \quad (2)$$

で定義する。ただし、集合 A に対して $|A|$ はその元の個数を表わす。また、2節点 u, v 間の最短経路の長さを u, v 間の距離といい、 $d(u, v)$ で表わすことにする。

二つのグラフ $G^1=[V^1, E^1], G^2=[V^2, E^2]$ に関して全単射 $f: V^1 \rightarrow V^2$ で、次式

$$\langle u, v \rangle \in E^1 \Leftrightarrow \langle f(u), f(v) \rangle \in E^2 \quad (3)$$

を満たす f が存在するとき、 G^1 と G^2 は同形であるといい、 f を同形写像という。部分集合 $V_0^1 \subset V^1, V_0^2 \subset V^2$ の間の全単射 $f_0: V_0^1 \rightarrow V_0^2$ が与えられたとして、 G^1, G^2 間の同形写像 f で、上の f_0 の拡張であるような f が存在するとき、 G^1 と G^2 は初期対応 f_0 のもとで同形であるといい、このような同形写像も簡単に f で表わすことにする。

* An Algorithm for Graph Isomorphism under Partial One-to-one Correspondence by Noboru KUBO (Department of Electronic Engineering, Graduate School of Osaka University), Isao SHIRAKAWA, and Hiroshi OZAKI (Department of Electronic Engineering, Faculty of Engineering, Osaka University).

本研究の一部は文部省科学研究費補助金：総合研究A135017（昭和51年度、昭和52年度）「ネットワーク構造を持つシステムに関する基礎研究」の援助のもとに行われたものである。

** 大阪大学大学院工学研究科電子工学専攻

*** 大阪大学工学部電子工学科

本文では、二つのグラフ G^1, G^2 と初期対応 f_0 が与えられたとき、 G^1 と G^2 が f_0 のもとで同形かどうかの判定問題を取り扱う。

二つのグラフ G^1, G^2 における節点集合の分割を、順序系列 $\pi^1=(V_1^1, V_2^1, \dots, V_k^1), \pi^2=(V_1^2, V_2^2, \dots, V_l^2)$ で表わす。ただし、 $h=1, 2$ に対して

$$\left. \begin{aligned} V_i^h \subset V^h, \cup_i V_i^h = V^h \\ V_i^h \cap V_j^h = \phi, (i \neq j) \end{aligned} \right\} \quad (4)$$

である。分割系列における各要素 V_i^h を、 G^h の第 i 番目のセル、または単にセル i とよぶ。どのような節点 $u^1 \in V_1^1, u^2 \in V_j^2 (i \neq j)$ に対しても $f(u^1)=u^2$ とするような (初期対応 f_0 のもとでの) 同形写像 f が存在しないとき、この分割 π^1, π^2 を同形分割といひ、 $\pi^1 \underset{f_0}{\approx} \pi^2$ で表わす。

(性質) 同形分割 $\pi^1=(V_1^1, \dots, V_k^1) \underset{f_0}{\approx} \pi^2=(V_1^2, \dots, V_l^2)$ において、 $|V_i^1| \neq |V_j^2|$ なる G^1, G^2 のセル i が存在するならば、二つのグラフは初期対応 f_0 のもとで同形でない、

まず、初期対応を与えた場合と、そうでない場合について、同形判定問題の“複雑さ”を考察する。

(定義)¹⁰⁾ 言語 L_0 と L が、次の条件を満たすとき、 L は L_0 に帰着可能 (reducible) であるといひ、 $L \infty L_0$ と書く、

(条件) L_0 を認識する $T(m) (\geq m)$ の時間複雑度 (time complexity)¹⁰⁾ の決定性アルゴリズムが存在すれば、 $T(P_L(m))$ の時間複雑度で L を認識する決定性アルゴリズムが存在する。ただし、 m は入力系列の長さであり、 P_L は L に依存する多項式である。

次の2つの問題*を考察する。

(P1) 初期対応のない同形判定問題。

(P2) n_0 個の節点対間の初期対応のもとでの同形判定問題。ただし、 $n_0 = \lceil \varepsilon(n - n_0) \rceil^{**}$ 。

($n \triangleq |V^1| = |V^2|, \varepsilon$ は正定数)

[命題1] $P1 \infty P2, P2 \infty P1$

(証明は付録参照)

この命題は、ある程度の初期対応が与えられたとしても、同形判定問題の複雑さが本質的に軽減されないことを示している。しかし、複雑さというものは最悪の場合の手数に関する評価であることに注意すれば、この命題は、初期対応の情報を活用した同形判定アルゴリズムの実際の運用に当り、その効率が改善される

ことがないということを主張するものではない。

次にアルゴリズムの基本となる二つの命題を示すが、一般性を損うことなく、与えられた同形分割の系列の長さ k, l は相等しいと仮定する。

[命題2] $f_0(x^1)=x^2 (x^1 \in V^1, x^2 \in V^2)$ なる初期対応 f_0 のもとでの同形分割 $\pi^1=(V_1^1, V_2^1, \dots, V_k^1) \approx \pi^2=(V_1^2, V_2^2, \dots, V_k^2)$ が与えられたとき、節点 $x^1(x^2)$ から $G^1(G^2)$ のすべての節点 $u^1 \in V^1 (u^2 \in V^2)$ までの距離 $d(x^1, u^1)(d(x^2, u^2))$ によって、細分化して得られる分割系列を、 $\pi^1=(V_{01}^1, \dots, V_{0k}^1, \dots, V_{(n-1)1}^1, \dots, V_{(n-1)k}^1)(\pi^2=(V_{01}^2, \dots, V_{0k}^2, \dots, V_{(n-1)1}^2, \dots, V_{(n-1)k}^2))$ とする。ただし、 $h=1, 2$ に対して

$$\left. \begin{aligned} V_{ij}^h \triangleq \{u^h | d(x^h, u^h) = i, u^h \in V_j^h\} \\ (0 \leq i \leq n-1, 1 \leq j \leq k) \end{aligned} \right\} \quad (5)$$

このとき、 $\pi^1 \underset{f_0}{\approx} \pi^2$ が成り立つ。

(証明略)

[命題3] 初期対応 f_0 のもとでの同形分割 $\pi^1=(V_1^1, V_2^1, \dots, V_k^1) \approx \pi^2=(V_1^2, V_2^2, \dots, V_k^2)$ が与えられたとき、各節点 $v^h \in V_i^h (i=1, 2, \dots, k, h=1, 2)$ の隣接ベクトル $A(v^h)$ を次式で定義する。

$$A(v^h) \triangleq (a_1, a_2, \dots, a_k) \quad (6)$$

ここで

$$a_j \triangleq |\Gamma(v^h) \cap V_j^h| \quad (7)$$

である。このとき、ある節点 $w \in V^1$ に関する非空の

$$X_i^h \triangleq \{x^h | A(x^h) \neq A(w), x^h \in V_i^h\} \quad (8)$$

($h=1, 2$)

に対して

$$\left. \begin{aligned} V_i^{*h} \triangleq V_i^h - X_i^h \\ V_{k+1}^h \triangleq X_i^h \end{aligned} \right\} (h=1, 2) \quad (9)$$

と置いて、分割 $\pi^{*1} \triangleq (V_1^1, \dots, V_i^{*1}, \dots, V_k^1, V_{k+1}^1), \pi^{*2} \triangleq (V_1^2, \dots, V_i^{*2}, \dots, V_k^2, V_{k+1}^2)$ を定めれば、 $\pi^{*1} \underset{f_0}{\approx} \pi^{*2}$ が成り立つ。

(証明は Corneil⁹⁾参照)

3. 同形判定のアルゴリズム

以下に提案するアルゴリズムは、2つの段階に分けられる。第1段階では、初期対応として与えられた n_0 個の節点対に関して、命題2の距離分割を繰り返し適用し、さらにその後命題3の隣接ベクトル分割を行っている。第2段階では、このようにして得られた同形分割をもとにバックトラックを行い最終的に同形性を判定する。

以下の手続きは、同形判定されるべきグラフ G^1

* 以下の各問題は、自然な符号化により言語として表現できる。

** $\lceil x \rceil$ は、 x より小さくない最小の整数を表わす。

$= [V^1, E^1], G^2 = [V^2, E^2]$ と、それらの間の初期対応 $f_0: V^0 \rightarrow V^0$ (ただし, $n_0 \triangleq |V^0| = |V^0|$) に対して実行されるものとする。

[第1段階]

step 0: $\pi^1 \leftarrow (V_1^1 = V^1), \pi^2 \leftarrow (V_1^2 = V^2)$ とおく. $s \leftarrow 1, t \leftarrow 1$ として次から開始。

step 1 (命題2に基づいた距離分割):

(1-1) $f_0(x_i^1) = x_i^2$ なる節点对 x_i^1, x_i^2 に関する距離ベクトル DIS^A を求める。ここで、すべての $v^A \in V^A$ に対して $DIS^A(v^A) = d(x_i^A, v^A)$ である。

(1-2) 現時点での分割系列を $\pi^1 = (V_1^1, V_2^1, \dots, V_k^1), \pi^2 = (V_1^2, V_2^2, \dots, V_k^2)$ としたとき, $h = 1, 2$ に対して以下の操作を行う。

(i) 各 $j (j = 1, 2, \dots, k)$ について順次次の操作を行う: 各 $x \in V_j^A$ を $queue^* Q(DIS^A(x))$ に入れる。

(ii) 各 $i (i = 0, 1, \dots, n-1)$ について順次次の操作を行う: $Q(i)$ から節点 x を取り出し, $x \in V_j^A$ であれば, x を V_{i+1}^A の元とする。

ただし、このようにして得られた分割系列 π^1, π^2 において, $V_{i+1}^A = V_i^A = \emptyset$ であるようなセルは存在しないものとみなし, 系列から除去する。

(1-3) すべてのセルについて $|V_{i+1}^A| = |V_i^A|$ ならば次へ移り, そうでなければ同形でないとして判定して操作完了。

(1-4) $s \leftarrow s+1$ とし, $s \leq n_0$ であれば, (1-1) へもどり, そうでなければ step 2 へ行く。

step 2 (命題3に基づいた隣接ベクトル分割): 現時点での分割系列 $\pi^1 = (V_1^1, V_2^1, \dots, V_k^1), \pi^2 = (V_1^2, V_2^2, \dots, V_k^2)$ に対して, 式(8)を満たす $X_i^1, X_i^2 (X_i^1 \cup X_i^2 \neq \emptyset)$ が存在しなければ, 次の step へ行く。存在したとき, $|X_i^1| = |X_i^2|$ ならば, これらから π^{*1}, π^{*2} を求め $\pi^1 \leftarrow \pi^{*1}, \pi^2 \leftarrow \pi^{*2}$ として step 2 の操作を繰り返す。 $|X_i^1| \neq |X_i^2|$ ならば, 同形でないとして判定して操作完了。

[第2段階]

step 3: 現時点での分割系列 $\pi^1 = (V_1^1, V_2^1, \dots, V_k^1), \pi^2 = (V_1^2, V_2^2, \dots, V_k^2)$ において, $|V_i^1| = |V_i^2| = 1 (i = 1, 2, \dots, k)$ ならば, G^1 と G^2 は同形であると判定して操作完了, そうでなければ, 次へ行く。

step 4: $|V_{j_i}^1| = |V_{j_i}^2| \geq 2$ なる (最小個数の節点を含む) セル j_i から, 任意の節点对 $x_i^1 \in V_{j_i}^1, x_i^2 \in V_{j_i}^2$ を選び出し, $V_{j_i}^A \leftarrow V_{j_i}^A - \{x_i^A\} (h = 1, 2), \pi^1 \leftarrow (V_1^1, V_2^1, \dots, V_k^1, \{x_i^1\}), \pi^2 \leftarrow (V_1^2, V_2^2, \dots, V_k^2,$

$\{x_i^2\})$ として次へ行く。

step 5 (隣接ベクトル分割): 現時点での分割系列 π^1, π^2 に対して式(8)を満たす $X_i^1, X_i^2 (X_i^1 \cup X_i^2 \neq \emptyset)$ が存在しなければ, $t \leftarrow t+1$ として step 3 へもどる。存在したとき, $|X_i^1| = |X_i^2|$ であれば, これらから π^{*1}, π^{*2} を求め $\pi^1 \leftarrow \pi^{*1}, \pi^2 \leftarrow \pi^{*2}$ として step 5 の操作を繰り返す。 $|X_i^1| \neq |X_i^2|$ ならば, 次へ行く。

step 6: 現時点での分割系列 π^1, π^2 の各々を, それらを得るために実行した step 5 の直前の状態へもどし, かつ x_i^2 を同じセル j_i に属する別の節点として step 5 へ行く。このような節点が存在しなくなったとき, 次へ行く。

step 7: $t \leftarrow t-1$ とおく。 $t \leq 0$ ならば, 同形でないとして判定して操作完了。 $t > 0$ ならば, step 6 へもどる。

アルゴリズムの正しさは, 命題2および3より明らかであろう。時間複雑度およびスペース複雑度 (space complexity)¹⁰⁾ は, それぞれ $O(n!)$, $O(m+n)$ となることも容易に理解されよう。

$G^2 = G^1$ としたときの同形写像 f , すなわち自分から自分自身への同形写像を特に自己同形写像といい, ただ一つの自明な自己同形写像しか持ちえないグラフをリジッドグラフ (rigid graph)¹¹⁾ とよぶ。Erdős¹²⁾によれば, ランダムグラフは“ほとんど”がリジッドグラフであると予想される。しかしながら, 本文のアルゴリズムではリジッドグラフに対してもバックトラックが起こる場合がある。例えば, n_0 個の初期対応を与えた, Fig. 1 のグラフおよびそれに同形なグラフをアルゴリズムの入力とした場合を考える。step 1 の距離分割を実行した時, 節点集合は, 初期対応の節点を除けば, それぞれ7個の節点から成る3つのセルに分割される。そして, step 2 の隣接ベクトル分割では, どのセルもそれ以上細分化されないから, step 3 以降の第2段階に入る。ところで, Fig. 1 のグラフはリジッ

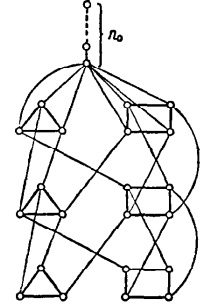


Fig. 1 A rigid graph.

* queue は first-in first-out 構造をもつリストである。

ドであるから、第2段階の step 4 および step 6 において、不適当な節点 x_i^2 を選んだとすれば、最悪の場合にはバックトラックを6回行うことになる。このように、リジッドグラフに対してもバックトラックが起こるのであるから、実行時間が多項式でおさえられないようなグラフのクラスも存在すると考えられる。

以上のように、本文のアルゴリズムは最悪の場合に対する理論的な保証が十分ではないが、実際の応用においては効率がよいと考えられる一つの根拠、すなわち初期対応の節点对の個数 n_0 が大規模なグラフに対しても比較的少なくてもよいと考えられる点について触れておくのは意味がある。

step 1 の距離分割によって得られる分割系列 π^1, π^2 において、各セルの大きさをほぼ1にするためには、初期対応の節点对の個数 n_0 をどの程度に設定すればよいかを線度 d (一定) のランダムグラフについて考える：いま、初期対応の一つの節点对に関して距離分割を適用したとする。このとき節点集合は、距離の違いによって大きさの比が、ほぼ $1 : d : d^2 : \dots : d^k$ (ただし、 k は $n \cong 1 + d + d^2 + \dots + d^k$ により定まる定数) の k 個の部分集合に分割されると考えられるが、ここでは簡単に、 $(1 + d + d^2 + \dots + d^{k-1}) : d^k \cong 1 : d$ の大きさの比で二つの集合に分割されるものとする。これより、距離分割を1回実行するごとに、分割系列の各セルについても、 $1 : d$ の比で二つのセルに分かれるものと考えることができる。初期対応の節点对がランダムに選ばれているとして、このような距離分割を n_0 回繰り返したとき、最大個数の節点をもつセルの大きさが1になるとすれば、

$$n \cdot \left(\frac{d}{1+d}\right)^{n_0} = 1 \tag{10}$$

が成り立つ。したがって、

$$n_0 \cong d \cdot \ln n \tag{11}$$

すなわち、大規模なグラフに対する n_0 も、初期対応の節点对がランダムに選ばれているならば、 n の対数に比例する程度で十分であろうと予想される。

4. 実験結果

アルゴリズムを FORTRAN でプログラムし、NE-AC 2200/700 で実行した。プログラムのステートメント数は、ランダムグラフの発生ルーチンなどを含め全体で 800 余りであり、使用したメモリは約 500kch

• どの節点の線度も相等しいグラフを正規グラフという。
 • $\lfloor x \rfloor$ は、 x を越えない最大の整数を表わす。

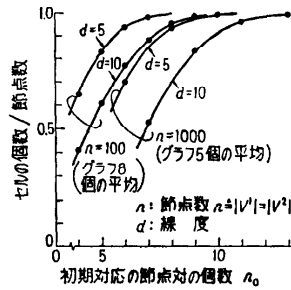


Fig. 2 The number of cells after distance partition.

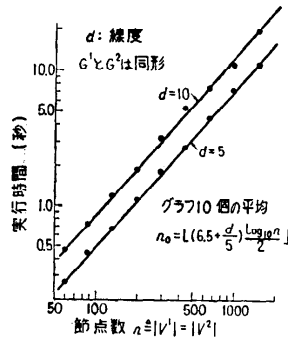


Fig. 3 Running time when G^1 and G^2 are isomorphic.

$\cong 63$ kW である。以下でその結果の一部について述べるが、入力したグラフはいずれも線度 $d=5, 10$ のランダムに発生させた正規グラフ* (ランダム正規グラフ) G^1 と、 G^1 の各節点名をランダムに置換して得られた同形グラフ G^2 である。また、初期対応の節点もランダムに選ぶことにした。

まず、初期対応の節点对の個数 n_0 をどの程度に設定すべきかを調べるために、節点数 $n=100, 1,000$ のグラフについて実験した (Fig. 2)。横軸は n_0 を表わし、縦軸は距離分割終了後の分割系列の長さ (セルの個数) とグラフの節点数 n との比を表わす。この図から、式 (11) の n_0 は大きく見積り過ぎであるが、 $\ln n$ に比例するという予想についてはほぼ確かめられたと考えられる。

以下の実験では n_0 を次式で与えることにした。

$$n_0 = \left\lfloor \left(6.5 + \frac{d}{5}\right) \frac{\log_{10} n}{2} \right\rfloor^{**} \tag{12}$$

次に、初期対応の節点对の個数 n_0 を式 (12) で与えた場合の実行時間を Fig. 3 に示す。図の各点はグラフ 10 個についての平均値を表わすが、ほとんどばらつきはない。また、いずれのグラフにおいても、一

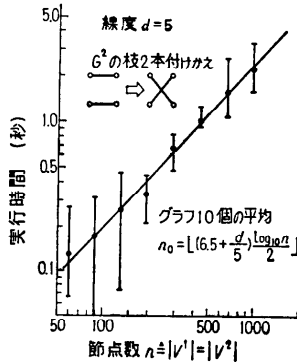


Fig. 4 Running time when G^1 and G^2 are not isomorphic.

度もバックトラックが起こらなただけでなく、距離分割終了後のセル数が $0.9n \sim n$ になることが確かめられた。これより、距離分割が実行時間の主要な部分を占めると予想されるが、実際 Fig. 3 ではほぼ $n^{1.2}$ に比例して増加している。

最後に、同形でないグラフを入力したときの実行時間を調べた (Fig. 4)。入力したグラフは上と同様に生成したものであるが、 G^2 の適当な2本の枝 $\langle u_1, v_1 \rangle$, $\langle u_2, v_2 \rangle$ を開放除去した後、枝 $\langle u_1, v_2 \rangle$, $\langle u_2, v_1 \rangle$ を付加している。これら同形ではないグラフについても、バックトラックは一度も起こらなかった。実行時間は大きくばらついているが、その最悪の場合でも同形グラフに対する実行時間を越えることはなかった。

5. むすび

本文で提案したアルゴリズムは、最悪の場合に対する理論的な保証が十分でなく、また特に新しい手法に基づくものでもないが、枝数の少ないグラフに対しては、比較的少数の初期対応を与えるならば、効率よく同形性の判定ができるところに特徴がある。

今後の問題としては、グラフが同形でないとき最小個数の枝をどのように付けかえれば同形となるか等グラフの診断がある程度可能なアルゴリズムの開発が残されている。

なお、LSI チップのレイアウトパターンにおける結線構造の検査に付随して生じる、1対1対応を部分的に持つグラフ間の同形判定問題の重要性に関しては、東京芝浦電気株式会社垂井忠明、吉田憲司の両氏のご示唆に負うところ大であり、ここに両氏のご討論に対して深謝する次第です。

参考文献

- 1) S. H. Unger: GIT-A Heuristic Program for Testing Pairs of Directed Line Graphs for Isomorphism, Commun. ACM, Vol. 7, No. 1, pp. 26~34 (1964).
- 2) J. Turner: Generalized Matrix Functions and the Graph Isomorphism Problem, SIAM J. Appl. Math., Vol. 16, No. 3, pp. 520~526 (1968).
- 3) D. G. Corneil & C. C. Gotlieb: An Efficient Algorithm for Graph Isomorphism, J. ACM, Vol. 17, No. 1, pp. 51~64 (1970).
- 4) D. G. Corneil: Graph Isomorphism, Tech. Rep. No. 18, Univ. of Toronto, Canada (1970).
- 5) A. T. Bertziss: A Backtrack Procedure for Isomorphism of Directed Graphs, J. ACM, Vol. 20, No. 3, pp. 365~377 (1973).
- 6) D. C. Schmidt & L. E. Druffel: A Fast Backtracking Algorithm to Test Directed Graphs for Isomorphism Using Distance Matrices, J. ACM, Vol. 23, No. 3, pp. 433~445 (1976).
- 7) 榎本, 米崎: グラフの同型判定のためのアルゴリズムとその性質, 信学技報, AL 76-49, pp. 93~102 (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, 信学会総合全国大会, 6-154 (1976).
- 9) J. E. Hopcroft & J. K. Wong: Linear Time Algorithm for Isomorphism of Planar Graphs, Proc. of 6th Ann. ACM Symp. on Theory of Computing, pp. 172~184 (1974).
- 10) A. V. Aho, J. E. Hopcroft & J. D. Ullman: The Design and Analysis of Computer Algorithms, Addison-Wesley (1974).
- 11) Z. Hedrin & A. Pultr: On Rigid Undirected Graphs, Canad. J. Math., Vol. 18, pp. 1237~1242 (1966).

付 録

命題 1 の証明

以下の証明では、 $P1$ の入力を二つのグラフ $G^{**} = [V^{**}, E^{**}] (h=1, 2)$ とし、 $P2$ の入力を $G^h = [V^h, E^h] (h=1, 2)$ および初期対応 $f_0: V_0^1 \rightarrow V_0^2 (n_0 \triangleq |V_0^1| = |V_0^2|)$ とする。

(i) $P1 \in P2$ の証明

G^{*1} から G^1 を次のように構成する。

$$n_0 \triangleq \lceil \varepsilon |V^{*1}| \rceil \tag{A.1}$$

$$V_0^1 \triangleq \{v_i | v_i \in V^{*1}, i=1, 2, \dots, n_0\} \tag{A.2}$$

とにおいて、

$$\begin{cases} V^1 \triangleq V^{*1} \cup V_0^1 & (A.3) \\ E^1 \triangleq E^{*1} \cup \{ \langle v_i, u \rangle \mid v_i \in V_0^1, u \in V^{*1} \} & (A.4) \end{cases}$$

同様に, G^{*2} から G^2 を構成し, 初期対応 f_0 を V_0^1 から V_0^2 への任意の全単射として定める.

このとき, G^1, G^2 を多項式時間で構成できることは明らかである. また, G^1 と G^2 が f_0 のもとで同形であるとき, かつそのときに限って G^{*1} と G^{*2} が同形であるのも明らかであろう. (証明終)

(ii) $P2 \propto P1$ の証明

以下では, G^1 から G^{*1} を構成する方法を示すが, G^{*2} の構成も同様に行う.

節点数 l 個の完全グラフを $C_l \triangleq [V_{c,l}, E_{c,l}]$ で表わし, C_l の特定の 1 節点を $u_{c,l}$ で表わすことにする. $V_0^1 = \{v_1, v_2, \dots, v_{n_0}\} \subset V^1$, $n \triangleq |V^1| = |V^2|$ としたとき, 次のように G^{*1} を構成する.

$$\begin{cases} V^{*1} \triangleq V^1 \cup \left(\bigcup_{n+1 \leq l \leq n+n_0} V_{c,l} \right) & (A.5) \\ E^{*1} \triangleq E^1 \cup \left(\bigcup_{n+1 \leq l \leq n+n_0} E_{c,l} \right) \\ \cup \{ \langle v_i, u_{c,l} \rangle \mid v_i \in V_0^1, l = n+i, i=1, 2, \dots, n_0 \} & (A.6) \end{cases}$$

ここで

$$|V^{*1}| = n + \sum_{l=n+1}^{n+n_0} l = 0(n^2) \quad (A.7)$$

$$|E^{*1}| \leq n^2 + \sum_{l=n+1}^{n+n_0} l^2 + \sum_{l=n+1}^{n+n_0} 1 = 0(n^3) \quad (A.8)$$

であることに注意すれば, 多項式時間で G^{*1} を構成できるのがわかる. また, G^{*1} と G^{*2} が同形のとき, かつそのときに限って G^1 と G^2 が f_0 のもとで同形であるのも明らかであろう. (証明終)

(昭和 52 年 10 月 14 日受付)

(昭和 53 年 3 月 1 日再受付)