データベース・システム 84−18
（１９９１．７．１８）

パターンマッチングを含むデータベース問合せについて

宝珍輝尚
ＮＴＴ情報通信処理研究所

本論文では、ラベル付き有向グラフで表現されたデータベースに対する問合せグラフとその問合せ処理方法を提案する。提案する問合せグラフは、要素ラベル列集合上での正規表現が可能なように問合せ言語G^+の考え方を発展させたものである。これにより、グラフ構造上のパターンに関する問合せが非手続き的に記述できる。問合せ処理では、問合せグラフを基本演算の組合せで処理する方法を採用する。本問合せ処理に対して、基本演算として提案したパスサーチ演算とトップノードを導入した格納構造が有効であることを示す。

A Study on Database Query Including Pattern-Matching

Teruhisa HOUCHIN
NTT Communications and Information Processing Laboratories
1-2356 Take, Yokosuka-Shi, Kanagawa 238-03, Japan

A query graph and a query processing method are suggested for a database expressed as a labeled directed graph. The proposed query graph may represent a regular expression over the set of the element label strings. This method is an enhancement of the query language G^+. This enables the query graph to describe the graph structural pattern of the required data non-procedurally. A query graph is decomposed and processed through a set of primitive operators. The proposed path search operators and the top-node graph data structure are efficient to this processing.

1．はじめに

先進的なＤＢＭＳ応用分野（ＣＡＤ，ＡＩ等）では、複数のデータ実体から構成される複合オブジェクトやデータ実体間の関連の記述といった関係モデルを超えるデータモデリング機能が必要とされ、この機能を持つ意味的なデータモデルの研究が行われてきた[7-10]。意味的なデータモデルを用いると、例えば、マルチメディアデータの内容，地点間の交通マップ，交換機間の接続等を比較的容易に表現できる。意味的なデータモデルでは意味をデータ実体間の関連で表現するため、意味的なデータモデルに基づくデータベースは、一種のグラフとみなすことができる。

このようなデータベースへの問合せに対しては、①グラフ構造に関する柔軟な検索条件の記述，ならびに，②非手続き的な問合せ記述といった要求がある。①は、例えば、地点Ａから電車かバスを乗り継いで行ける地点を求めるといった、ある関連パターンを持つノードを求めるのに不可欠である。また、②は問合せ手順を利用者が考えないことであり、一般利用者とのインタフェースとして不可欠である。

ここで、問合せ言語Ｇ*[1]では、問合せを問合せグラフ(query graph)と呼ぶ一種のグラフで表現し、問合せグラフの枝のラベルに、データを表現するグラフの枝のラベル列集合上の正規表現を指定可能としている。これにより①が可能である。

しかし、Ｇ*では基本的なパスサーチの組合せで問合せを記述するため、②を満足しない。特に、パス途中の点に対する検索条件を含む問合せは複数の問合せグラフを組み合わせてしか記述できない。

そこで本論文では、Ｇ*の考え方を発展させ、②をも満足する問合せグラフならびにその問合せ処理方式を提案する。提案する問合せグラフでは、要素ラベル列集合上での正規表現を可能とし、パス途中の点に対する条件をも一問合せグラフ中に記述できる。また、提案する問合せ処理は、パスサーチ演算を基本演算とし、これらの基本演算の組合せにより非手続き的な問合せグラフを処理するものである。

以下，２．で本研究の背景であるデータのグラフ表現，グラフのための格納構造とＧ*の問合せグラフについて述べる。３．で問合せグラフを提案し、４．で問合せ処理方法を提案する。５．で議論を行い、最後に６．でまとめと今後の課題を述べる。

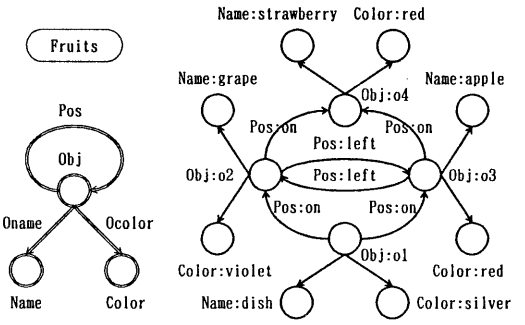
2．前提

2．1　データグラフ[6]

グラフでデータを表現するデータグラフを概説する。

データグラフは、データを表現するラベル付き有向グラフである。グラフ要素をデータベース内で一意に識別する一意識別子d_{ID}，同種のグラフ要素集合をデータグラフ内で一意に識別する要素名Ｎ，データｄによってラベルは３つ組(d_{ID},Ｎ,ｄ)で表される。点および枝にはこのラベルが付与される。本稿の図では、簡単化のため一意識別子d_{ID}を省略して示す。点集合Ｖと枝集合Ｅからなる順序組(Ｖ,Ｅ)が連結グラフである場合、グラフｇ(Ｖ,Ｅ)をデータ表現グラフという。また、データ表現グラフを成分にもつグラフをデータグラフという。データグラフの集合であるデータベース内でデータグラフを識別するために付与した名前をデータグラフ名という。同一データグラフ名のデータグラフの要素名のみをラベルとしたグラフを、該データグラフのスキーマグラフという。

スキーマグラフとデータグラフの例を図１に示す。これは、果物の静物写真の内容を表わすデータ表現グラフである。データグラフ名はFruitsである。本データグラフには、Obj，Name，Colorという点と Oname，Ocolor，Posという枝がある。枝Posは、隣接物体間の位置関係を表す。例えば、物体名appleの物体は、物体名grapeの左にあると表現されている。



(a) スキーマグラフ　　(b) データグラフ

図１　スキーマグラフとデータグラフの例

2．2　格納構造

グラフの計算機上での表現には、隣接行列や隣接点リストによる方法がある[14]。この中で、隣接点リストによる方法がＤＢＭＳの意識する格納構造によく使用されている[1,2,10,11]。本データ構造をＤＢＭＳの格納構造とすると、要素の探索は深さ優先探索や幅優先探索等にならざるを得ず、探索は巡航的となる。従って、巡航により要素に到着して初めて要素にアクセス可能となる。順方向のみ巡航可能な隣接点リストによるデータ構造の例を 図２(a)に示す。これは、図１のデータ表現グラフの一部を表わしたものである。

一方、著者はグラフのための格納構造として、グラフの点(枝)を直接指示するポインタ群で構成される点(トップノード)を設けたデータ構造を提案している[6]。点トップノードを持つ順方向隣接点リストによるグラフ表現例と枝トップノードによるグラフ表現例を、各々、図２(b),(c)に示す。top-n, top-eが各々点トップノード，枝トップノードである。

2．3　問合せグラフ

問合せにグラフを用いる研究[1-5,9]の中から、提案する問合せグラフの基盤であるG+[1]を概説する。

G+では問合せは問合せグラフ(query graph)とサマリグラフ(summary graph)によって記述される。問合せの結果、問合せグラフの検索パターンに合致するデータが、サマリグラフで表現された形で返却される。図１のデータグラフに対する同世代問合せの例を図３に示す。ここではPos:onの存在のみを仮定している。図中央の矢印の左が問合せグラフであり、右側がサマリグラフである。Q2ｏQ1は、Q1の結果にQ2を行うことを表す。問合せグラフ中の点線と実線は、各々、データグラフ中のパスと枝を表す。点線の枝のラベルには、データグラフの枝のラベル列集合上での正規表現が記述可能である。Q1は、値o1を持つ点から１以上の枝を経由して到達可能な点ｘと該点までのパス長を求める(count)という問合せである。Q2は、Q1の結果に対し、o1からのパス長(z)が等しい点ｘと点ｙを求める問合せである。問合せ結果は、Q2で得られた点ｘと点ｙをo1からのパス長をラベルとする枝で連結して返却される。

また、G+では サマリ演算子(summary operator)と呼ぶパスに関する２種の集約演算子:パス演算子(path operator)とセット演算子(set operator)を提案している。パス演算子はパスに沿った属性値の集約を計算し、セット演算子はパス集合に対して値の集約を計算する演算である。サマリ演算子には count,product,sum,min,maxがある。図４(a)に o1上の物体総数を求める問合せグラフを示す。枝上のコンマの左がパス演算子，右がセット演算子を表す。本問合せグラフは、パスに対して枝の数(count)を求め、パス集合に対してその値の和(+)をとることを表す。また、図４(b)にo1からo4への最短の物体列を求める問合せグラフを示す。これは、o1からo4へのパスの枝の数)を求め、パス集合中の最小のパスを選ぶことを表す。

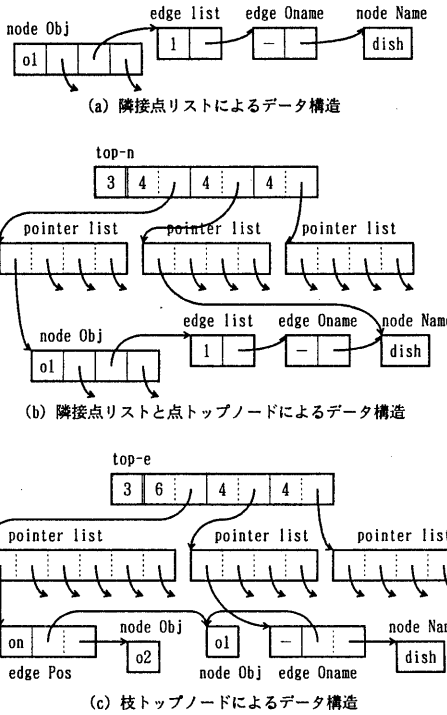


図２　グラフのためのデータ構造の例（図１のデータグラフの一部）

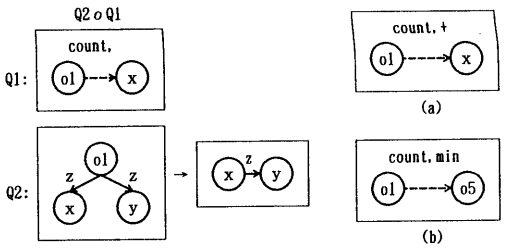


図３　G+の問合せ例　　図４　G+のサマリ演算子の例

3．問合せグラフ

非手続き的に問合せ記述が可能な問合せグラフを提案する。まず、問合せグラフに対する要求を述べ、次に、これらの要求への対処方法について述べる。これらの対処方法をまとめて問合せグラフを定義する。最後に問合せ例を示す。

3．1　要求

図1の果物の静物写真の内容を表わすデータグラフを考えてみよう。このようなデータグラフに対して、例えば、インテリアデザイナは盛りつけ方を研究するために、ある皿の上の3段以上の果物の上に赤いりんごがある写真を、皿とりんごとその位置関係を指定するのみで求めたいと思うであろう。これは、データを表現するグラフの構造に関するパターンを指定した問合せであり、本稿ではパターンマッチング問合せと呼ぶことにする。これらの問合せを記述するために必要な要求を次に示す。

［要求1］グラフ構造に関する柔軟な検索条件の記述.

これは、要求するデータを表現するグラフの構造的パターンを記述することである。パターンマッチング問合せを記述するためには、曖昧性のないパス指定ができるのみではなく、グラフ構造に関する柔軟な検索条件の記述が不可欠である。

［要求2］パスに関する検索条件の記述.

パス長に関する条件や最小コストのパスを求めたいという要求がある[1,13]。グラフ表現されたデータに対する検索にはパスに関する条件が記述できるべきである。

［要求3］値に関する検索条件の記述.

従来のデータベース問合せにもある記述能力で、明らかに不可欠である。

［要求4］利用者の求める形での問合せ結果の返却.

利用者インタフェースとして、所望のグラフ構造で問合せ結果が得られるべきである。このために、問合せは導出データベースの作成であるという考え方[3]を導入すべきである。

［要求5］問合せ手順を利用者が考えないこと.

処理順序を考えた複数の問合せグラフによる問合せは、慣れなければ困難である。利用者インタフェースとして、求めるものを記述するのみで問合せ可能とすべきである。

3．2　正規表現

G⁺で使用されている正規表現により要求1が満足される。しかし、パス途中の点の検索条件を含む問合せは 複数のguery graphとしてしか記述できず、G⁺の問合せグラフでは要求5は満足されない。

そこで、要求1，5を満足するために以下を導入する。

① 点ラベルを含んだパスラベル.

② データグラフの要素のラベル列集合上の正規表現.

さらに、点の値に関する条件を通常のように指定可能とすれば、要求3を満足することも同時に可能である。

3．2．1　パスとパスラベル

まず、パスについての定義を行う。

［定義1］データグラフG(V，E)において、$p=(x_1, x_2, \cdots, x_n)$をパスという。ここで、$x_i \in V$ならば$x_{i+1} \in E$，$x_i \in E$ならば$x_{i+1} \in V$である。要素$x_1$をパスの始要素，$x_n$をパスの終要素という。パスの始終要素がともに点のとき、該パスを完全パスと呼ぶ。パスの始要素または終要素が枝のとき、該パスを不完全パスと呼ぶ。不完全パスは、終不完全パス，始不完全パス，始終不完全パスの3種類に分類できる。始(終)不完全パスは、始(終)要素として枝を持ち、終(始)要素として点を持つパスである。始終不完全パスは、始終要素がともに枝であるパスである。パスの全ての点がdistinctのときpを単純パス(simple path)という[2]。また、パスを構成する枝の数をパス長という。　□

本定義は、文献[2]の定義を一般化したものである。完全パスが今までのパスに相当する。枝は、始終不完全パスの範疇に含まれる。

次に、パスラベルを定義する。

［定義2］パス$p=(x_1, x_2, \cdots, x_n)$において、$\lambda(p)=\lambda(x_1)\lambda(x_2)\cdots\lambda(x_n)$をパスラベルという。λは要素から組(要素名，データ)を取り出す関数である。　□

ここで定義したパスラベルは 文献[2]の定義と異なる。本定義によると、パスラベルは点のラベルを含む。これにより、点に関する検索条件が記述可能となる。

最後に、パスの等価性，同種性を定義しておく。

［定義3］2パスラベルλ(p)，λ(q)において、要素名が順序を含め同一で、かつ、データが順序を含め

等価であるとき、λ(p), λ(q)は等価 (λ(p)=λ(q)) であるという。また、2パスp, qにおいて、λ(p)=λ(q)のときp,qは等価である (p=q) という。2パスp,qが、各々を構成する要素に関らず以下の条件を満たすとき、pとqは同種であるという。

(1) pの始要素がqの始要素と同種で、かつ、pの終要素がqの終要素と同種のとき.

(2) p, qのいずれか一方が長さ0のパスのとき. □

3.2.2 εと括弧

正規表現のためには以下が必要である[14]。

(1) 長さ0のパスε

(2) 和集合P∪Qを表す(p+q),

(3) 連接集合PQを表す(pq),

(4) pの連接を0回以上繰り返してできるラベル列集合P*を表す(p*)

ここで、p,qはラベル列集合P,Qを表す正規表現.

連接pqはグラフの要素間の連結で表現できるので、連接以外を表現するために以下をグラフに加える。

(a) 長さ0のパスε

(b) 複数パス記述可能で、閉包フラグを持つ括弧

括弧内の複数パス記述により+を表現する。すなわち、括弧内の同種のパスp, qによって、p+qを表現する。従って、括弧内に2以上のパスが存在するときは該パスは同種でなければならない。また、閉包フラグにより括弧内のパスに*を適用するか否かを表現する。ここで、pp*(1以上の連接を繰り返してできるラベル列の集合PP*をあらわす)をp+と省略して表せるよう、閉包フラグに+の指定も可能とする。また、意味のない括弧を防ぐために、始括弧の右側は要素無しまたは終括弧以外とし、終括弧の左側は要素無しまたは始括弧以外とする。さらに、括弧の左右の両要素が点および枝であるとパスの定義を満たさないので、これも禁じる。終括弧も同様とする。

3.3 パス集約関数,パス述語と値域変数,一時変数

要求2のために、G+ではパス演算子とセット演算子を導入している。本稿では、G+と異なる考え方でパスに関する演算(パス集約演算子とパス述語)を導入する。

パス集約演算子は、G+のパス演算子にaverageを加え、点のラベルをパスラベルに含むための対処を施したものである。すなわち、パス集約演算子aggr(p,k)は、パスpの要素kの集約値をを計算するものである。ここで、aggrはcount,sum,ave,product,max,min, kは要素の種類で all,node,edgeのいずれかである。パス集約演算子は、算術演算子, 集約演算子, 値を返却する利用者定義演算子と共に、条件式や返却結果式中に記述可能とする。

パス述語max{x}は、パス集合Xからx_{max}を捜し、変数xをx_{max}に束縛し、真値を返却する述語である。ここで、パスに割り当てられた値の集合をX, 該値集合を値域とする変数をx, x_{max}=max(X)とする。min{x}も同様とする。パス集合Xが空のとき、パス述語は偽値を返却する。パス述語は、関係演算述語, 真偽値を返却する利用者定義述語と共に条件式中に記述可能とする。

さらに、値域変数と一時変数を導入する。値域変数は、検索条件式や返却結果を得る計算式を記述可能とするために不可欠である。本稿では、問合せグラフの要素ならびに括弧で囲まれたパスには(1以上の)値域変数を設定可能とする。一時変数も実際に問合せを記述する場合、良く使用される。値域変数と一時変数は、問合せグラフ内のどこからでも参照され得るので、問合せグラフ内で可視とする。

図4の問合せ例を、提案した演算, 変数を用いて図5に示す。点中および枝上に記述された変数(v,w,x,y)が値域変数で、その他の変数(z)は一時変数である。図5(a)では、パスを表す変数x中の点の数を求め、これを変数xの全てについて合計している。図5(b)では点の数が最小のパスを求めている。

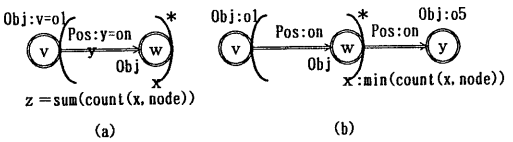


図5 パス集約演算子とパス述語の例

3.4 一時要素と結果式

G+では、サマリグラフを利用者に描かせて要求4を満足する。しかし、G+では問合せにあたり問合せグラフとサマリグラフの両グラフを描かなければならない。

提案する問合せグラフでは、検索パターンの指定ならびに問合せの返却結果の形を一問合せグラフで記述可能とし、さらに、返却結果として計算式を記述可能とする。このために以下の表記を可能とする。

(a) 一時要素名：問合せ対象のデータグラフに対するスキーマグラフの要素名以外の要素名

(b) 結果式：返却値を計算する式またはnon-return.

3．5 定義

以上の考え方を導入した点，枝，ε，括弧により問合せグラフを表現する。これらの要素の形式的な定義を付録に示す。概略的には、点,枝,終括弧に、値域変数,条件式,結果式を記述可能としている。以上の要素と一時変数により問合せグラフを定義する。

［定義４］６つ組(Ｖ,Ｅ,Ｚ,Ｉ,Ｔ,TV)を問合せグラフという。ここで、Ｖは点集合，Ｅは枝集合，Ｚはε集合，Ｉは始括弧集合，Ｔは終括弧集合，TVは一時変数集合．また、問合せグラフG_Qの要素名集合N_QがスキーマグラフG_Sの要素名集合N_Sに対して、

$$\exists_x \exists_y \ (x=y \wedge x \in N_Q \wedge y \in N_S)$$

なる関係があるとき、問合せグラフG_QはスキーマグラフG_Sに基づく問合せグラフであるという。 □

問合せグラフはパスから構成されるとも考えられる。そこで、問合せグラフを構成するパスを表現するグラフをパスグラフということとし、パスグラフの始点，終点は、パスのものと同じとする。また、後の定義のために、パスグラフG_Pの終括弧集合T_Pに終括弧btが存在する場合は、全該要素の閉包フラグがnullであるとき、G_Pを純連接パスグラフということとする。

3．6 問合せ例

問合せグラフのいくつかの例を示す。ここでは図１のデータグラフを問合せの対象とする。図６～図９では、点や枝の外側には要素名と条件式を、点の中，線上には要素に対する変数を示す。また、ハッチングされた点および太線の枝で利用者への返却要素を示す。

［例１］りんごの色を求めよ。

本問合せグラフを図６に示す。図で、名前がappleである物体の色を求めている。利用者に返却されるのは問合せグラフの全要素であることを示している。

［例２］皿の上方の赤い物体の名前を全部求めよ。

本問合せグラフを図７に示す。これは、名前がdishである物体を始点とし、Posの値がonである枝が１以上存在する物体で、色が赤の物体の名前を求める問合せグラフである。＋付きの括弧は、括弧内のパスが１回以上存在するという条件を示している。

［例３］皿からの距離がリンゴと等しく、リンゴと同じ色の物体を求めよ。

本問合せグラフを図８に示す。これは、名前がdishの物体を起点として名前がappleの物体と 値onなる枝Posに関して同じ長さのパスを持つ物体で、appleの色と同じ色を持つ物体の名前を求めるものである。このように点の値に関する条件が一問合せグラフ中に記述できる。問合せ結果は値appleを持つ点，Same Positionという一時定義枝と名前を値とする点によって構成されるグラフであり、導出ＤＢの適用例である。

［例４］平面的に輪を構成する物体を求めよ。

本問合せグラフを図９に示す。これは、値leftなる枝posをたどると自分に戻るような物体を求める問合せグラフである。

このように、提案した問合せグラフでは、問い合わせるパターンをそのまま記述でき、問合せ結果も共に記述できる。また、点に関する条件記述ならびに値に関する条件記述も可能である。

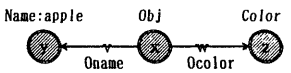


図６ 例１の問合せグラフ

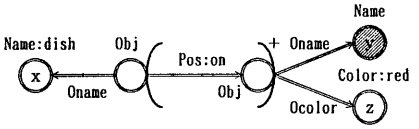


図７ 例２の問合せグラフ

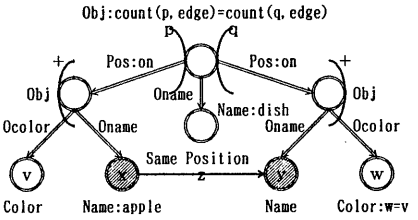


図８ 例３の問合せグラフ

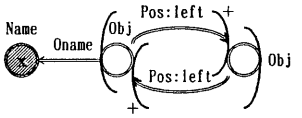


図９ 例４の問合せグラフ

4．問合せ処理

ここでは、２つの制限を設けて問合せ処理を議論する。１つは、問合せの対象をパターンマッチング問合せに限定することである。これは、問合せグラフの処理の中でパターンマッチング問合せが最も困難と考えられるためである。もう１つは、問合せグラフを一時スキーマを含まない問合せグラフに限定することである。これは、議論の簡単化のためである。

4．1　要求

問合せグラフの処理方法には２つのアプローチが考えられる。一つは、G⁺等[1,2]のように、パスサーチアルゴリズムのみであらゆる問合せを処理するアプローチである。もう一つは、関係データベースの問合せ処理のように複数の基本演算の組合せで処理を行うアプローチである。

前者のアプローチは、処理順序を考慮していないため、アクセスパスの最適化が困難である。本アプローチの主たる興味は要求データが存在するか否かである。そこで、ここでは後者のアプローチを採ることにする。本アプローチに対する要求を次に示す。

［要求６］パターンマッチング問合せに対する処理を基本演算の組合せで実現したい

［要求７］できる限り早く評価対象範囲を狭めたい

4．2　パスサーチ演算

要求６を満足するためには、基本演算が明確化されている必要があるが、パターンマッチング問合せについては何を基本演算とすべきかという議論は全くなされていない。そこで、まず、パターンマッチング問合せのための基本演算について議論し、次に、その基本演算の実現方法を述べる。

4．2．1　基本的な考え方

問合せグラフはグラフの構造としては一般的な有向グラフであり、単純パスサーチアルゴリズム[2]だけでは処理できない。なぜならば、問合せグラフに強連結成分が存在し得るからである。ここで、一般の有向グラフGに対して、Gの各強連結成分の全枝を短絡除去するとGにはサイクルは存在せずGは有向非サイクルグラフ(ＤＡＧ)になることが知られている[15]。そこで、問合せグラフの強連結成分を専門に処理する演算を導入し、該成分を処理させる方法を採るのは妥当と考えられる。

問合せグラフから強連結成分を除いた後のＤＡＧを、さらに入樹木，出樹木に分解して処理する方法も考えられる。しかし、これはグラフの分解，組み立てが複雑となるだけで、処理効果は少ない。

従って本稿では、問合せグラフの単純パスに対応するデータグラフを検索する演算(単純パスサーチ演算)と強連結成分に対応するパス群を検索する演算(回帰パスサーチ演算)により問合せグラフを処理することとする。

4．2．2　パスサーチ演算の実現法

(1) 単純パスサーチ演算

単純パスサーチ演算は文献[2]の単純パスサーチアルゴリズムを基本とする。文献[2]のアルゴリズムは、与えられた正規表現に対する決定性有限オートマトン(ＤＦＡ)をグラフの深さ優先探索に利用し、(ＤＦＡの)同一状態でグラフの同一点を訪れないように点へのラベリングを駆使するという方法である[2]。

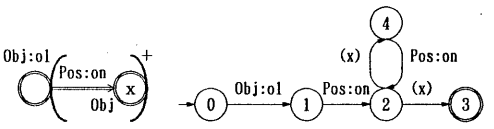
本稿での考慮点を次に示す。

(1) 点ラベルによるＤＦＡの状態遷移.
　本稿のパスラベルには点のラベルが含まれる。

(2) 変数の遷移図中のラベルへの記述.
　G⁺では複数のパス問合せも一パスサーチアルゴリズムで行うようにオートマトンを構築して処理を行うため、処理上複数のパス問合せ間の変数の授受はない。しかし、本稿のアプローチでは、基本演算の組合せで問合せ処理を行うため、基本演算間の変数の授受を考慮する必要がある。

変数は他の演算結果を表す。遷移図中で状態Ａから状態Ｂへの遷移ラベルに変数が記述された場合、状態ＡでのＤＦＡへの入力は該変数が表す要素のみに制限される。本入力は必ず状態Ａを状態Ｂに遷移させる。

パスグラフとそれに対する遷移図の例を図10に示す。図10(b)が状態遷移を表現する遷移図で、円で状態を表わしている。状態０が初期状態，二重円が終了状態を示している。ここで、例えば状態０から状態１への遷移は、要素名Objとデータolの組によって引き起こされる。遷移においてデータが記述されないときは、要素名のみ合致すれば遷移が起こることを示す。パスグラフ(図10(a))中の変数ｘが他の演算結果を表す変数なら



(a) 問合せグラフ　　(b) 遷移図

図10　問合せグラフと遷移図の例

ば、遷移図における遷移のラベルは図10(b)に示すように(x)となる。変数xが他の演算結果を表す変数でない場合は、遷移図のラベル(x)はObjとなる。従って、全Objが状態を遷移させることができる。

図1のデータグラフと図10(b)の遷移図をもとに今回提案するアルゴリズムの流れを概説する。まず、初期状態0においてObj:o1が入力されると状態1に遷移する。Obj:o1に対して状態番号である1が付加済みか検査し、未付加の場合は1を付加する。Obj:o2への枝PosとObj:o2が入力されると終了状態となり、Obj:o2は受理される。次に、状態2に戻りObj:o2の入力に対して状態4に遷移する。ここで、Obj:o4への枝PosとObj:o4が入力されると、Obj:o4は受理される。Obj:o4には出枝がないので、バックトラックしてObj:o1に戻り、Obj:o3への枝を調べて行く。この結果、パス集合{o1-o2, o1-o2-o4, o1-o3, o1-o3-o4}(-は枝Posを表す)が得られる。

(2) 回帰パス検索演算

次に、回帰パスサーチ演算の実現アルゴリズムａｐｓを示す。ａｐｓではｓｐｓ演算を使用して与えられる強連結問合せグラフ中のパスを評価する。なるべくｓｐｓ演算の使用回数を減らすために、強連結問合せグラフ中の最長パスを最初に評価すする。ここでは、このパスとして長さn-1のハミルトンパスを使用する(nは強連結問合せグラフの点の数)。長さmのハミルトンパスは、点aから点bに向かうパスp_{ab}をa行b列とする行列Vのm乗(V^m)によって求める[15]。ここで、点aから点bに向かうパスが複数ある場合、それらをp_1,…,p_kとすると$p_{ab}=p_1+\cdots+p_k$で表現する。ただし、計算の途中で行列の要素が有向花(diflower)となった場合は該有向花を抹消する。有向花とは、完全パスを表現する点の列$v_1\cdots v_k$において、$v_2=v_k$または$v_1=v_{k-1}$であるようなパスである。V^mの主対角以外の要素がハミルトンパスである。この最初に評価するパスを以降では巡航パスという。次に、巡航パスグラフ以外のパスグラフ集合PG中のパスグラフを脇パスグラフと呼び、制約として使用するため番号を付与しておく。以上の処理の後、巡航パスグラフの評価をｓｐｓ演算に依頼する。この結果得られたグラフに対する脇パスグラフの評価を、脇パスグラフ毎にｓｐｓ演算に依頼する。ここで、脇パスグラフの始終点として巡航パスグラフの評価で得られた要素を使用する。アルゴリズムの流れを以下に示す。

[アルゴリズムａｐｓの流れ]

```
入力：データグラフＰ，
      強連結成分を構成するパスグラフの集合PG
出力：データグラフＲ
手続き：
  長さn-1の ハミルトンパスを求め、
  巡航パスグラフpgMとする
  脇パスグラフに番号を付与(pg1,pg2,…とする)
  R 〈= sps(P,pgM)
  for each pgi∈{pg1, pg2, …}
      Q 〈= {}
      for each p∈P
          for each r∈R
              if r⊆p then
                  pgiの始終点 〈= r中の要素
                  S 〈= sps(p,pgi)
                  Q 〈= Q＋S
      R 〈= Q
```

4．3　問合せ処理方法

上記の考え方を取り込んだ問合せ処理方法について述べる。問合せ処理で主となる、問合せグラフの基本演算への分解方法と基本演算の評価方法を示す。

4．3．1　問合せグラフ分解

まず、問合せグラフから制約演算で処理できる部分を抽出する。すなわち、要素名に？を持たない純連接パスグラフを抽出してGRとする。GR以外の部分から強連結成分抽出してGHとする。強連結成分の抽出は、例えば文献[14]のアルゴリズムをそのまま使用できる。この後、順方向の巡航のみが可能なデータ構造の場合は、順方向のパスのみとなるようにパスグラフ集合GLを作成する。両方向に巡航可能なデータ構造の場合は、どの点も入出枝の合計が2以下となるようにパスグラフ集合GLを作成する。以上で、問合せグラフの分解が終了する。この結果、制約演算で処理できるパスグラフ集合GR，ａｐｓ演算で処理を行うグラフ集合GH，ｓｐｓ演算で処理を行うパスグラフ集合GLが得られる。

4．3．2　基本演算の評価

処理の流れを表わすために問合せ処理グラフを導入する。これは、関係データベースの問合せ処理で使用

されるものと同様である。問合せ処理グラフは、基本演算，要素集合，結果のデータグラフを点とし，点間のデータの流れを枝で表わすグラフである。ただし、要素集合を表わす点をソースノードに，結果のデータグラフを表わす点をシンクノードに持つ。なお、制約は整数値の大小判定や文字列の部分一致等を前提とし、最も処理コストの低いものとする。以下で基本演算と呼ぶのは、制約演算，ｓｐｓ演算，ａｐｓ演算である。また、枝の順方向の巡航のみを考慮する。両方向に巡航可能な場合も同様の議論が成り立つ。また、格納構造はトップノードを導入したデータ構造を前提とする。

問合せグラフを分解して得られた基本演算は、処理コストの低いものから順番に評価する。これは、格納構造にトップノードの導入を仮定したことによる。例２の問合せグラフの問合せ処理グラフとｓｐｓ演算のＤＦＡの遷移図を図11に示す。図では、処理コストの低い制約演算が最初に実行されることを示している。この評価順序は関係ＤＢへの問合せにおける基本演算の実行順序とうまく対応する。

５．議論

５．１　問合せグラフ

提案した問合せグラフはG⁺に比べ、パス途中の点に対する条件を問合せグラフ中に記述可能にでき、非手続き的な問合せ記述が可能である。

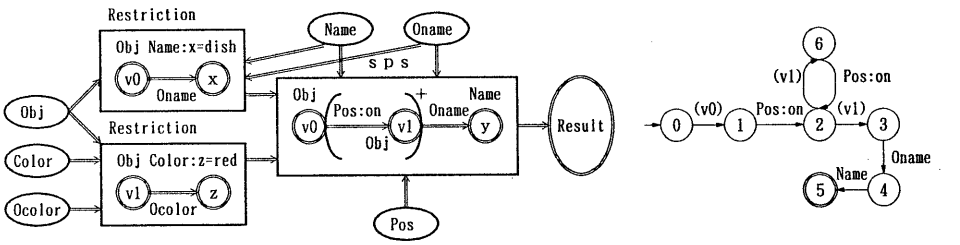
また、G⁺でもパスに関する演算を導入しているが、パス演算子にはaveがなく、セット演算子のcount,product，sumは値の計算で、従来の集約演算子で計算できる。さらに、セット演算子のmin，maxは値の計算ではなく、最小，最大の値を持つパスを求める演算である。本稿ではパス演算子，セット演算子という分け方をせず、パス演算子に相当するパス集約演算子とセット演算子のmin，maxに対応するパス述語を導入した。

さらに、G⁺では問合せにあたり問合せグラフとサマリグラフの両グラフを描くが、提案した問合せグラフでは、検索パターンの指定ならびに問合せの返却結果の形を一問合せグラフで記述できる。

５．２　問合せ処理

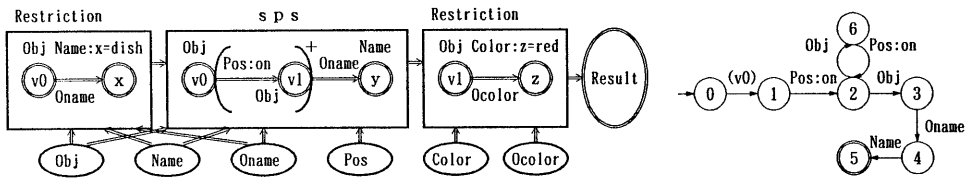
ここでは、４．３の問合せ処理と連接点リストによるデータ構造の場合の問合せ処理を比較する。ここで、データグラフ中のある一要素名を持つ要素のみが直接アクセス可能と仮定する。これ以外の要素がインデックス等により直接アクセス可能な場合は、該要素を含むように以下の議論を容易に拡張できる。

図11に対応する問合せ処理グラフとｓｐｓ演算の遷移図を図12に示す。制約演算は処理可能な要素に到達するまで実行できないため、例えば色がredという制約演算はｓｐｓ演算の後にしか処理できない。この効果



(a) 問い合わせ処理グラフ　　(b) 遷移図

図１１　問合せ処理グラフと遷移図の例（トップノードを持つデータ構造の場合）



(a) 問い合わせ処理グラフ　　(b) 遷移図

図１２　問合せ処理グラフと遷移図の例（隣接点リストのデータ構造の場合）

は遷移図の状態２から状態３，６への遷移に現われている。すなわち、図12では（変数v1は他の演算結果に対する変数ではないので）状態２における入力は全要素である。一方、図11では、本入力は変数v1で表される制約演算の結果のみである。従って、図11のｓｐｓ演算での評価範囲は図12の場合に比べ狭くできる可能性があり、効率の良い問合せ処理が実現できる。

また、この例の場合、v0からv1を経てｚまでを一パスサーチ演算で処理することも可能である。v1から複数の制約演算評価が必要な場合、一制約演算はｓｐｓ演算中で評価できる。しかし、他の制約演算はｓｐｓ演算の後でしか評価できない。

６．おわりに

本論文では、ラベル付き有向グラフで表現されたデータベースに対する問合せグラフとその問合せ処理方法を提案した。提案する問合せグラフでは、要素ラベル列集合上での正規表現を可能とし、非手続き的な問合せ記述を可能とした。この問合せ処理では、基本演算として提案したパスサーチ演算とトップノードを導入した格納構造が有効であることを示した。

筆者は、ＣＯＭＭＯＮと名付けた拡張可能ＤＢＭＳを試作中である。ＣＯＭＭＯＮでは、マルチデータモデルの実現を目指しており、本稿で述べたデータグラフを多種のデータモデルのプラットフォームとして使用する予定である。また、その格納構造は本稿で示したデータ構造を選択可能とする予定である。さらに、本稿で述べたパスサーチ演算を、グラフ表現されたデータに対する高水準な(非手続き的な)問合せを実現するために使用する予定である。

今後は、本稿で示した問合せグラフの表現能力評価，パスサーチ演算の性能評価，一時スキーマを含んだ一般の問合せグラフの処理方法ならびに問合せグラフからの最小数のパスグラフ抽出を含む最適化方式の検討が課題である。

謝辞　本研究を行うにあたり日頃から熱心に議論して頂く田中豪主幹研究員をはじめとするＮＴＴ情報通信処理研究所の皆様に感謝いたします。

文　献

(1) I. F. Curz et al:"G+: Recursive Queries Without Recursion",12th Expert Database Sys., pp.355-368, 1988.
(2) A. O. Mendelzon et al:"Finding Regular Simple Paths in Graph Databases", 15th VLDB, pp.185-193, 1989.
(3) A. M. Alashqur et al:"OQL:A query language for manipulating object-oriented databases", 15th VLDB, pp.433-442, 1989.
(4) M. Guo et al:"An Association Algebra For Processing Object-Oriented Databases", 7th Data Eng., pp.23-32, 1991.
(5) M. Gyssens et al:"Graph-Oriented Object Database Model", 9th ACM PODS, pp.417-424, 1990.
(6) 宝珍："拡張可能DBMS:COMMONの格納構造と基本演算，情処研資，82-6，1991.
(7) R. Hull and R. King:"Semantic database modeling: Survey, applications, and research issues", ACM Computing Survey, Vol.19, No.3, pp.201-260, 1987.
(8) D. W. Shipman:"The functional data model and the data language DAPLEX", ACM Trans. Database Sys., Vol.6, No.1,pp.140-173,1981.
(9) W. KIM:"A Model of Queries for Object-oriented Databases",15th VLDB,pp.423-432,1989.
(10) D. Jagannathan et al:" SIM: A database system based on the semantic data model", Proc. of ACM SIGMOD'88, pp.46-55, 1988.
(11) A. Chan, S. Danberg, S. Fox, W. K. Lin, A. Nori and D. Ries:"Storage and access structures to support a semantic data model",Proc. of 8th VLDB,pp.122-130, 1982.
(12) Mannino et al:Extensions to Query Languages for Graph Traversal Problems",IEEE Tr. Know. and Data Eng., Vol.2, No.3, 1990.
(13) J. Orenstain et al:"PROBE Spatial Data Modeling and Query Processing in an Image Database Application", IEEE Tr. Software Eng., Vol.14, No.5, 1988.
(14) Aho他：ｱﾙｺﾞﾘｽﾞﾑの設計と解析，ｻｲｴﾝｽ社，1977.
(15) 小野寺：ｸﾞﾗﾌ理論の基礎，森北出版，1969.

付録　問合せグラフ要素

①点：４つ組(n | ?,Y,p,r).　②枝：６つ組(n | ?,Y,p,v_i | bi | bt,v_t | bi | bt,r).　③長さ０のパス：組(v,e | bi | bt | φ) | (e,v | bi | bt) | (bi,v | e | bi | bt) | (bt,v | e | bi | bt | φ) | (φ,v | bi | φ).
④始括弧：組(v,EUIU{ε}) | (e,VUIU{ε}) | (bi | bt | ε | φ,VUEUIU{ε}).
⑤終括弧：6つ組(VUTU{ε},e,k,Y,p,r) | (EUTU{ε},v,k,Y,p,r) | (VUEUTU{ε},bi | bt | ε | φ,k,Y,p,r).
ここで、nは要素名または一時要素名，？はどの１要素名にも合致する要素名．Yは値域変数の集合．pは条件式．rは結果式．kは閉包フラグでnull,*有りまたは+有り．v_iは始点，v_tは終点，φはNULL,Vは点集合,Eは枝集合,Iは始括弧集合，Tは終括弧集合．ただし、始括弧biの第二要素とbiに対応する終括弧btの第一要素の数は等しい。また、始括弧と対応する終括弧内に複数のパスがあるとき、該パスは互いに同種である。さらに、btのkが*または+有りのとき 第一要素の集合の要素は１つであり、対応するbiとの間のパスは始点／終点不完全パスである。