

知識ベースとデータベースの統合利用によるグラフ問題の処理

高須 淳宏 大須賀 節雄

東京大学工学部境界領域研究施設

本論文は、知識処理を利用したモデル構築の方法、特にモデリングを計算機で支援するための基礎研究を目的とする。その一過程として様々な分野のモデルとして幅広く利用されているグラフを処理するシステムを開発した。このシステムは、データベース+知識ベース+推論エンジンよりなる知識処理システム上で構成されている。知識ベースと推論エンジンは、データ構造を持った論理系(多層論理)に基づいて作られたKAUSを利用する。またデータベースについては、構造を持ったデータを扱いやすくするため関係モデルを若干拡張する。次に、多層論理式によってグラフの様々な知識を表現し処理する。その結果、従来のグラフ処理システムのように与えられたグラフを解析することから、与えられた条件を満足するグラフを合成することまで柔軟にグラフを処理できることを示す。

"Integration of Knowledge Base and Data,
and Application to Graph Problem" (in Japanese)

by Atsuhiko TAKASU, Setsuo OHSUGA

(Institute of Interdisciplinary Research,

Faculty of Engineering, The University of Tokyo)

Modeling is the important part of the knowledge processing. Graph is used as model in various fields such as OR, communication, etc. So we construct graph processing system for the basis of computer aided modeling. This system is built on one of the typical knowledge information configuration that consist of data base, knowledge base and inference engine. we used knowledge base system KAUS. KAUS is based on Malti Layer Logic (MLL) that is extended first order logic and treats hierachical data structure explicitly, while, as data base, we extend relational model to treat structured data easily. Next, we expressed some of graph theory in MLL formula and point out that graph theory is expressed declaratively and treated flexibly.

1. はじめに

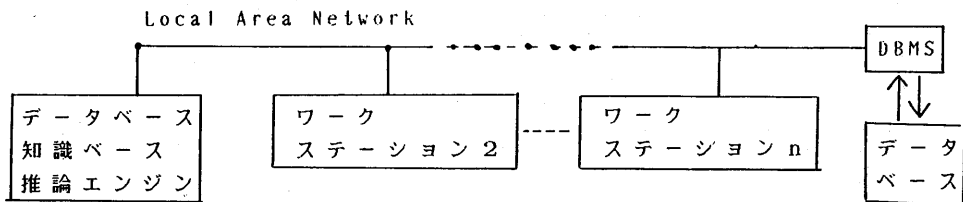
知識処理は、設計など人間の知的作業をサポートすることを目標としており、ここでは、多量の情報をたくさん人間が供用しながら作業が進められるような環境が要求される。そのためには、多量のデータや知識を合理的に扱う機構が必要である。このような環境に適応するため図1のようなシステム構成が提案されている([1])。多量のデータは、中央のデータベースで管理され、人間はここから必要なデータをワークステーションに移して作業を行う。また、ワークステーション上のシステムにはマン・マシン・インターフェイスをはじめとして、各種の機能が要求される。それらの機能の1つとしてモデルの構築及び処理の機能がある。モデリングは、人間の問題解決の標準的な方法であり、対象とする問題の領域に関係なく利用される方法である([2])。そこで、本研究では、モデル表現の1つの手段としてORや通信など様々な分野で幅広く利用されているグラフを取り上げ、知識ベースとデータベースを利用しグラフを処理するシステムを開発した。

2. 知識処理システム-KAUS

現在当研究室では、知識処理システムKAUSを開発している。知識処理においては、構造を明確に記述できること及び構造を含む表現の推論メカニズムを有することは重要である。KAUSは、第1階述語論理をベースとし階層的なデータ構造を中集合、集合-要素関係、構造-要素関係を使ってexplicitに記述できる多層論理にもとづいたシステムであり、図1のワークステーション上での核となるシステムをめざしている([3],[4],[5])。現在のところ、残念ながらデータ管理の部分が遅れている。そこで、本研究ではまずデータ管理のためのデータベースを考えた。データ管理は、KAUSが論理を基礎としたシステムであることも考えて、関係モデルが望ましいが、KAUSは、構造を持ったデータを扱うため従来の関係モデルではデータの記述が容易でない場合がある。そこで、関係モデルを拡張して表のなかに他の関係へのポインタを書き込むことにより構造を導入した現在はいくつかの構造化データベースが提案されているが([7])、この構造は多層論理のデータ構造に対応したもので、現在は直接データを与えているが、将来はホストのデータベースのデータをこの形に変換する処理系が必要になる。

多層論理の構造化されたデータは以下のようにデータベースで表現する。

- ・集合-要素の関係は、表とタブルの関係で表わす。
- ・構造-要素の関係は、タブルと属性の関係で表わす。
- ・構造-要素関係で要素が集合の場合、その集合を表わす関係へのポインタを使う。



ワークステーション

図1 知識処理システムの構成

1つはKAUS論理式をデータベース言語にコンパイルし、直接データベースのデータを使う。データベースは、構造化されており、多層論理のデータ構造と対応しているため、データベース言語へのコンパイルは、不可能ではない。

もう1つの方法は、データベースのデータを一旦KAUSのデータに変換する方法である。この方法では、データベースは、ホストのデータベースまたは、他のシステムとのデータ受け渡しの媒介の役割を果たす。

3 多層論理によるグラフの知識表現

グラフ理論は、古くから理論・応用の両面で研究されており、理論的にはかなり確立された学問分野である。また、その応用範囲は非常に広く、各種の問題のモデルとして利用できる。例えば、道路地図は、交差点を頂点、道路を辺とすグラフで表される。各辺に移動にかかる時間や距離、コストなどの情報を付加しグラフの最短経路を求めれば、最短時間経路や最低コスト経路、最短距離経路などがもとまる。さらに、道路地図に限らず、幾何的に対象をグラフで表わし、そのグラフの最短経路を求めることは、その問題の最適解を求めることと密接に関係しているため、グラフの最短経路の問題は、重要な問題である。また、グラフの頂点を、弧を通して流れる仮想的な物質の始点と終点として、物質の流れの回路網を表現する。さらに各弧に、流れの上限と単位流量あたりのコストを付加し、全体のコストを最小とするような流れを求める。この問題は、グラフを使って輸送問題を一般化したものとなっている。このように、グラフはさまざまな分野のモデルとして有効であることがわかる。

グラフは構造を持っており、グラフの諸定理はその構造に関する記述である。一方、これらの定理の適用に際しては、個々のデータが対象となるため、データの管理及び利用の方法が必要である。そこで、以下に、グラフの構造の記述方法グラフのデータを管理するためのデータベース、そして、グラフの定理の記述の例として単純連鎖を取り上げ、処理メカニズムとデータベースの利用法について述べる。

まずグラフの構造であるが、理論的にはグラフは、頂点の集合 (V)、辺の集合 (E)、結合写像 (E から $V \times V$ の中への写像) からなる。ただし結合写像は省略されることが多く、頂点の集合と辺の集合で表すのが普通である([8])。この構造は、多層論理のデータ構造 (集合-要素関係, 構造-要素関係) を使って、図2のように表わせる。ここで \ni は集合-要素関係を、 \triangleright は構造-要素関係を表す。

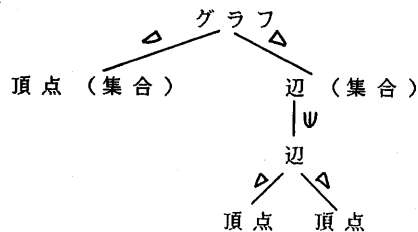


図2 抽象グラフの概念図

しかし、グラフを実際の問題に応用する場合は、辺や頂点に様々な情報を付加する必要が生じる。例えば、地図をグラフで表わすには、辺に距離、時間、コストなどの情報を付加しなければならないし、輸送問題では、各辺に流せる物質の

量の上限と単位流量あたりのコストを付加しなければならない。これらの例では、辺に対する付加情報しか必要でなかったが、場合によっては、頂点に情報を付加しなければならない。

そこで、グラフの定義を辺や頂点に情報を付加できるように拡張する。図3に拡張したグラフの構造の概念図を示す。

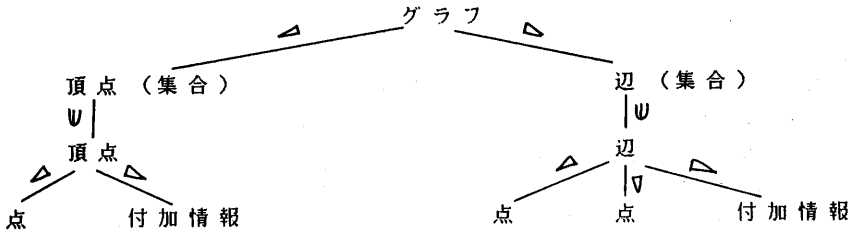


図3 グラフの概念図

続いてグラフのデータを管理するためのデータベースについて述べる。このデータベースは、グラフの構造を反映していることが望ましい。そこで、2節で示した構造を持ったデータベースを利用する。以下にその例を示す。まずグラフの集合を表す関係は、そのグラフの頂点の集合を表す関係へのポインタを示す 'vtx' と、辺の集合を表す関係へのポインタを示す 'arc' というアトリビュートよりなる。頂点の集合を表す関係は、点を示す 'vertex' と付加情報を示す任意個のアトリビュートよりなる。また辺の集合を表す関係は、始点を表す 'ip' と終点を表す 'tp' と付加情報を示す任意個のアトリビュートよりなる。図4には、グラフのデータベースの例を示す。

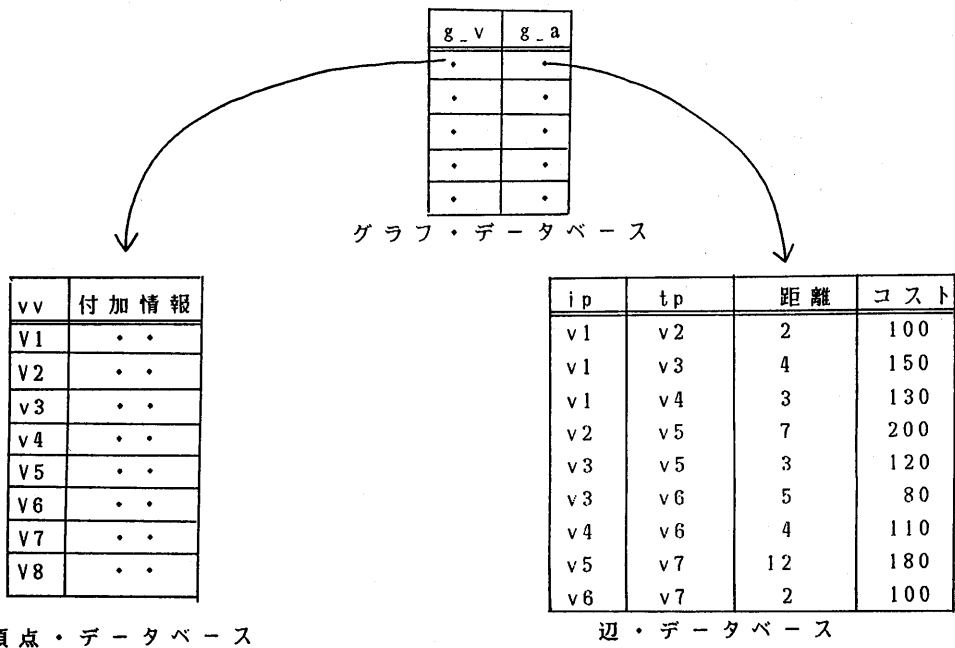


図4 グラフの構造化データベース

この構造に従って、最短経路、最長経路、最大流れなどグラフの様々な知識を多層論理式で表現できる。ここでは、これらの知識のうち単純連鎖の知識の例をあげる。以下に単純連鎖の定義と多層論理による知識を示す。

・単純連鎖の定義

任意のグラフ G ，任意の G の頂点 $I v$ ， $F v$ ，
 任意の G の頂点の順序集合 $P a t h$ について
 $P a t h$ が G において $I v$ から $F v$ への単純連鎖であるとは、
 ある整数 N ，ある整数の集合 D があって、
 集合 $P a t h$ の要素の数が N ， D が 1 から $N - 1$ までの整数の集合であり、
 D の任意の要素 N_2 にたいし、 G のある辺 $A r c$ があり、
 $P a t h$ の第 N_2 番目の要素が $A r c$ の始点であり、
 $P a t h$ の第 $N_2 + 1$ 番目の要素が $A r c$ の終点である。

・単純連鎖の知識

- [$\forall G / graph$] [$\forall I v, \forall F v / vertex$] (1)
- [$\forall Path / * vertex$] (2)
- ($path_of(G, Iv, Fv, Path) : -$ (3)
- [$\exists N / int$] [$\exists D / * int$] (4)
- ($number(Path, N, D),$ (5)
- $ele_of(Path, 1, Iv),$ (6)
- $ele_of(Path, N, Fv),$ (7)
- [$\forall N_2 / D$] [$\exists Arc / G : g_a$] [$\exists N_3 / int$] (8)
- ($N_3 is N_2 + 1,$ (9)
- $ele_of(Path, N_2, Arc : ip),$ (10)
- $ele_of(Path, N_3, Arc : tp)$ (11)
-)
-)
-)

多層論理による単純連鎖の定義式

ここで、 E / D はドメイン D の要素 E を、述語を区切る '，' は論理の AND を、' $V : i$ ' は V の構造要素 i を、 $*S$ は S の巾集合を示している。この例で、定義とその多層論理式を見比べればわかるように、K A U S では知識を再帰形を用いずに自然に記述できることを注意しておく。続いて推論メカニズムだが、基本的には PROLOG と同じように反ばく手続きになっているが、構造を含む表現の推論メカニズムを含んでいる ([3],[4],[5])。構造を表現するため論理式には変数のドメインを指定する冠頭部 ([限量詞, 変数/ドメイン])、構造要素を指定する ':', 巾集合を表す '*' がある。データベースへのアクセスは、冠頭部で行われる。この際、' V / D ' は関係 D のタプル V を、' $T : C$ ' はタプル T のアトリビュート C の値 (関係の場合もある) を、' $*S$ ' は関係 S の部分関係の集合を示す。例えば、上の論理式の 8 行目の [$\exists Arc / G : g_a$] では、グラフ G の辺の集合を示すデータベースへアクセスし、そのデータベースのタプルで、9, 10, 11 行目を満たすようなタプル $A R C$ を探す。このように、2 節のデータベースの構造化により、多層論理式で表現された知識を個々のデータに容易に適用できるようになる。

次に、従来のグラフの処理方法と今回の方法について簡単に考察してみる。こ

れまで、グラフは行列によって表現され、この表現にもとずいてプログラムが作られてきた。行列表現のおもな特徴は処理速度が早いことで、グラフをORの分野などで利用する場合は、莫大な量のデータを扱わなければならないので、この特徴は非常に重要となる。また、これまでグラフを計算機で扱う場合、グラフはほとんど行列で表現されてきたため、ソフトウェアの蓄積があることもメリットとなっている。反面、行列表現及びその解析ソフトウェアは、PERTのように処理が定まっているような場合には、威力を発揮するが、少しかわった解析をしようとした時にはプログラムを変更しなければならず、この変更が容易でない。1つには、データ表現に柔軟性がないように思われる。例えば、行列表現では辺や頂点に距離やコストなどの情報を任意個付加するのはむずかしい。しかし、こうした情報を構成的に表現できなければ、それに柔軟に対応するプログラムを作ることむずかしい。特に、モデルとしてグラフを使う場合は、目的とするグラフの解析や合成など柔軟な処理が要求される。

以下に、グラフを使ったいくつかの応用を示す。

4. PERTへの応用

まず始めに、最長経路の知識を使って作業計画に利用されるPERTへの応用を試みる。PERTでは、1つのプロジェクトをいくつかの部分作業に分割し、各作業の相互関係をグラフで表す。グラフの各辺は、部分作業に対応し、その作業の見積り時間の情報が付加される。

PERTで重要なものは、

プロジェクトを最短の時間で修了させるために重要な作業列

(クリティカル・パス)

と各作業について

最も早く作業を開始できる時間 --- 最早開始時刻 (作業開始からの最長経路)

最も早く作業を修了できる時間 --- 最早完了時刻 (作業開始からの最長経路)

全体のプロジェクトに影響を及ぼさない限界の開始時間

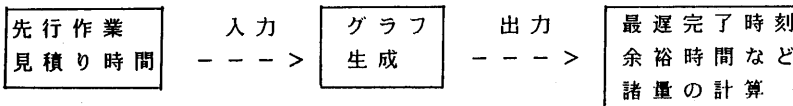
--- 最遅開始時刻 (作業修了からの最長経路)

全体のプロジェクトに影響を及ぼさない限界の修了時間

--- 最遅完了時刻 (作業修了からの最長経路)

全体のプロジェクトに影響を及ぼさない限界の遅れ --- スラック

などであり、括弧には、各量の計算につかわれるグラフの知識を示した。また、これらの値を計算するのに必要な情報は、各作業についてその先行作業と時間である。そこで、次のようなシステム構成でPERTの解析を行う。



PERT解析システム構成

5 最低コスト経路の問題

続いて、最短経路の知識を利用して、鉄道での最低コスト経路の問題への応用を試みた。

一般に問題を解決するためのいろいろな道をグラフで表わすことができれば、そのグラフの最短経路を求めることは、その問題の最適解を求めることに対応する(探索空間は普通グラフで表わされる)。このため最短経路の問題は、非常に応

用範囲がひろい。

ここでは、鉄道網をグラフで表わし、ある駅から他の駅への最低コスト経路を求める問題を考えた。ただ、鉄道会社が複数社あるため

- (1) 各会社ごとに1つのグラフで表し、全体として複数のグラフを扱うこと、
- (2) 各辺に直接コストの情報を付加できないこと

などの点が普通のグラフの最短経路問題と異なる。そのため

- ・グラフの各辺には、距離と会社の情報を付加し
- ・各会社ごとに、距離と料金の対応表を与えて

料金を計算する。

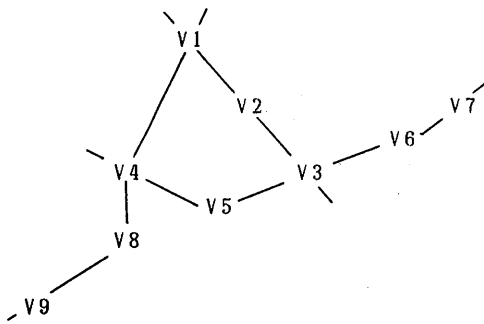
ここで問題なのは、グラフが膨大となり、解を求めるのが大変なためグラフを以下の方法で階層化する。

- (1) 線が交差する駅（乗り換え駅）を上レベルに持ち上げる

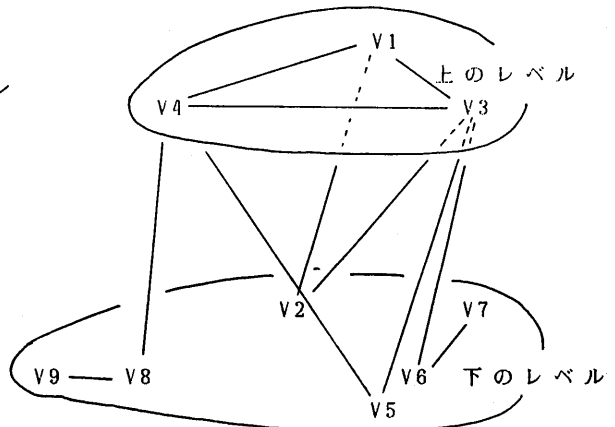
- (2) 上のレベルにある駅（乗り換え駅）間について

- ・2つの駅が繋がっていて、その間に乗り換え駅がない場合は、その2つの駅間に辺を作り距離などの情報を付加する。
- ・同一の会社の線で2通り以上につながっていれば辺に最短距離を付加する。
- ・異なった会社の線につながっている場合は、その2つの駅間に複数の辺を作りそれぞれ距離などの情報を付加する。

以下に階層化の例を示す。



平面的なグラフ



階層的なグラフ

質問時に乗る駅や降りる駅が下のレベルに含まれる場合も乗り換え駅と同じ方法で上のレベルに持ち上げ、上のレベルのグラフについて最低コスト経路を求める。こうすればグラフの頂点の数を最大、（乗り換え駅の数+2）まで減らすことができる。グラフの処理時間は、頂点の数あるいは、辺の数と相関関係があるため、このような階層化は有効である。データの階層化は多層論理の特徴の1つで処理効率の向上に役立つ。

6. おわりに

本論文では、まずはじめに構造を持ったデータを管理するためのデータベースを提案した。現在の関係モデルは、述語論理とむすびついて、その理論体系がもとめられつつある。そして、そのデータ構造はシンプルかつ強力である。しかし今後計算機の扱うデータが複雑になるにつれて、構造を持ったデータを *explicit* に扱えることが必要になってくるものと思われる。今回は、多層論理の

データ構造にあわせてデータベースを構造化する方法を提案したが、構造化の方法・構造化データベースの代数・論理との関係・問い合わせ言語・一貫性制約など考えなければならない問題がたくさんある。

本論文では、統いて多層論理式によってグラフの様々な知識を表現し処理した。その結果、宣言的に知識を表現することができ、またその処理は、グラフの解析問題から合成問題まで柔軟に処理できることが示せ、モデリングの枠組みができた。また、グラフはモデル記述の方法の1つであり、複数のモデル構築の方法を共用できることが望ましいが、その点、今回利用した知識処理システムKAUSは、巾集合を利用し、階層構造を持ったデータを処理するための推論メカニズムを有しているため、図5に示すように階層構造+グラフの処理が可能になり、より強力な記述力を与えられる。

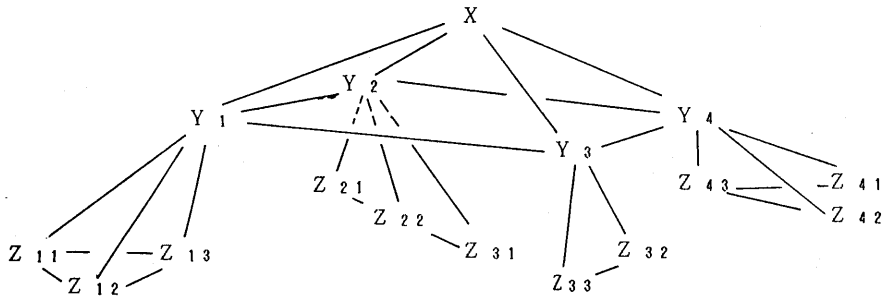


図5 階層構造とグラフ

最後にこのグラフ処理システムを使っていくつかの応用を試みたが、その結果多層論理の重要な概念であるデータの階層化によって処理効率を向上できる問題があることを示せた。

謝辞

いつも貴重な時間をさいて熱心に議論してくださる大須賀研究室の皆さんに感謝します。

参考文献

- [1]大須賀節雄：知識処理システムへの応用—新コンピュータ・システムへの道
コンピュータ・システム シンポジウム 1983 情報処理学会
- [2]大須賀節雄：知識ベース及びデータベースを用いたモデル構築技法
アドバンスド・データベース・システム予稿集, 1982
- [3]大須賀節雄：多層論理—モデル記述の為に述語論理
Proc. of The Logic Programing Conference '83
- [4]山内平行, 大須賀節雄：多層論理のモデル理論と証明論
知識工学と人工知能42-3(情報処理学会), 1985
- [5]阿久津他：KAUSコンパイラについて—知識ベースの構造に関する研究会
- [6]H.GALLAIRE, J.MINKER, J.M.NICOLAS:
Logic and Database:A Deductive Approach,
Computing Surveys Vol16, No2, June, 1984
- [7]P.Pistor, R.Traumueeller:A Database Language for Sets, List and Tables
- [8] R.G.Busacker, T.L.Saaty :
FINITE GRAPHS AND NETWORKS 1965 McGraw-Hill
- [9] V.Chachra, P.M.Chare, J.M.Moore :
Application of Graph Theory Algorithms 1979 North Holland