データベース・システム 67－5
（１９８８．９．２２）

ネットワークデータベース設計支援システムの開発

古川哲也　　上林彌彦　　木實新一

九州大学工学部

データベースにおける問題として，意味制約の保持と質問処理の効率化がある．ネットワーク構造に基づくデータベースシステムは，リンクでデータ間の対応を表しており，スキーマ設計では，構造の表す制約と質問処理の容易さを同時に考慮しなければならない．本稿では，従属性制約の保持し質問を効率よく処理できるネットワークデータベースを設計するための支援システムの開発について述べる．

質問の処理効率は，質問に必要なデータの対応を容易に求めるための冗長なデータの付加によって向上する．このようなデータに対しては，従属性制約を保持を保持するための管理を要する．即ち，処理効率と冗長性の管理のトレードオフが重要な問題となる．本システムでは，質問処理とネットワーク構造の適合性のいくつかのクラスに基づき，この問題に対処している．

Development of a Network Database Design Support System

Tetsuya FURUKAWA, Yahiko KAMBAYASHI, and Shinichi KONOMI

Department of Computer Science and Communication Engineering, Kyushu University

6-10-1 Hakozaki, Higashi, Fukuoka 812, Japan

There are two problems of databases, preserving semantic constraints and query processing efficiency. In database systems based on the network structure, we should consider both of the dependency constraints expressed by the structures and query processing efficiency at the database design stage. In this paper we discuss development of a system to support the design of network databases which preserve dependencies and process queries efficiently.

Query processing efficiency is improved by adding redundant data to get the correspondence of data for the query easily. We have to manage these data to preserve dependencies. Thus the trade-off between query processing and management of redundancy becomes an important problem. The system deals with this problem using classes of suitability of network structures for query processing.

1．まえがき

データベースの大規模化にともない，処理速度の高速化がますます重要となっている．ネットワーク構造はリンクでデータ間の1対多の対応を表しており，データの対応を求める際には，関係の結合などの操作に比べ効率がよい．本稿では，ネットワーク構造を用いたデータベースのスキーマ設計を，質問処理を考慮して支援するシステムの開発について述べる．

データベースシステムは，利用者に対し速く応答し，正しいデータを供給しなければならない．正しいデータの供給のために，誤ったデータがデータベース内に蓄えられないように，データの持つ意味制約，一般には関数従属性や結合従属性などの従属性制約がデータベース設計時に構造に反映される．従属性制約を反映することにより，データの冗長性が削減され，データベース更新時の制約の保持も簡単になる．応答性については，質問の最適化などにより，処理の効率化がなされている．

ネットワーク構造の特徴は，データの対応をポインタリンクで表していることであり，それにより次の2つの効果がある．

(1) 同じ値を親で代表させ記憶効率がよくなる．
(2) データの対応を求めるには，リンクをたどるだけでよく，処理効率がよい．

しかし，ネットワーク構造は，構造自体に制約があり，質問処理効率も構造によって異なるため，構造の決定（スキーマの設計）時に従属性制約の保持と質問処理の効率化の両方を考慮しなければならない．従属性制約を用いた設計法についてはいくつかの結果が知られている[6][7][8][12]が，質問処理の効率を考えると，データベースがどのように使用されるかを解析し，代表的な質問は効率よく処理できるような構造とする必要がある．質問処理効率の向上は，主に冗長性の付加により達成されるが，冗長性が大きくなると更新時の制約保持の問題が生じる．即ち，検索処理の高速化と更新処理の簡単化や記憶効率のトレードオフが重要となる．このため，これまでのスキーマ設計は，実体関連モデルなどを用いて属性の対応関係から行われており，設計者の経験によるところが大きいものとなっている．

関係モデルに基づくデータベースの設計では，2つの問題，従属性制約の保持と質問処理の効率化は分離して扱われている．関係スキーム集合は結合従属性と一致するように設計され，関数従属性は関係のキーとして反映される．質問処理については，結合演算の処理が最もコストがかかり，そのための質問の最適化や効率の良い処理法などが提案されている．結合演算の高速化のために，索引構造を付加したり，結合される組をあらかじめリンクで結んでおくことも考えられる．索引やリンクを用いると，ネットワーク構造を用いたデータ表現となり，ネットワーク構造は物理レベルの効率の良い実現法でもある．

現在開発中のネットワークデータベース設計支援システムは，従属性制約を保持し，与えられた質問集合を効率よく処理できるネットワークスキーマを設計することを目的としている．その際，冗長性と処理効率のトレードオフの問題から，どの程度まで質問処理に適した構造にするのかを選べるようにしている[3]．本稿では，システムの持つ機能を概説し，システムの構成について述べる．

2．ネットワーク構造と質問処理の適合性

属性集合ABCDに対し，ABの値を指定して対応するCDの値を求める質問（Q(AB,CD)で表す）を考える．この質問の処理で最も効率がよいと思われるのは，図1(a)の有向グラフで表されるネットワーク構造，またはA,Bを入れ換えたものである．ここで，節点はレコード型を，有向枝は親子集合型を表す．レコード型R(X)は属性集合Xからなるレコードの集合であり，親レコード型がR_0,子レコード型がR_Mである親子集合型<R_0,R_M>は，R_0の各レコードとR_Mの任意個のレコードの対応（親子集合）の集合である．また下線はレコード型のキー属性である．ネットワーク構造を表す有向グラフをバックマン線図という．

この構造では，R_1でAの値によりレコードが唯一定まり，R_2でそのレコードに対応しBの値の指定を満たすレコードが定まる．それに対応するR_3のレコード集合のCDの値がこの質問の解となる．この構造での実現値を図1(b)，ABの値をa_1b_1とする．レコードa_1，a_1b_1がレコード型R_1,R_2でそ

れぞれ決定される．それに対応するR3のレコードはa1b1c1d1とa1b1c1d2があり，Qの解はそのCD部分に相当する{c1d1, c1d2}となる．

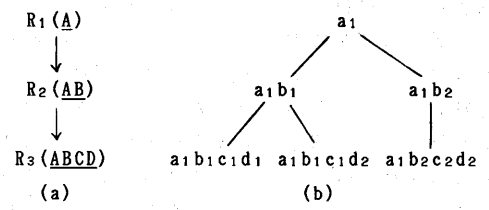
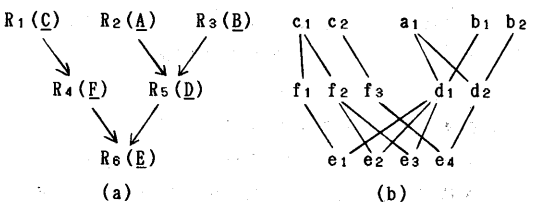


図1　Q(AB,CD)の処理効率のよいネットワーク構造

属性の全体集合をU＝ABCDEF，Uで満足される関数従属性の集合を{E→DF, F→C, D→AB}とする．Yが親，Xが子の属性集合である親子集合型は，子レコードが決まれば対応する親レコードが決まるので，関数従属性X→Yを表現していると見なすことができる．図1(a)の構造では，表現する関数従属性はAB→Aのような自明なもののみであるが，E→DF, D→ABをそれぞれ{E→D, E→F}, {D→A, D→B}とすると，図2(a)はすべての関数従属性を表現している．この構造では，各属性がレコード型のキーとなっており，最も冗長性が少ない．図2(b)に示す実現値はEFを除くと図1(b)と同じデータであり，これは図2(c)に示す関係で表すことができる．



A	B	C	D	E	F
a1	b1	c1	d1	e1	f1
a1	b1	c1	d1	e2	f2
a1	b1	c1	d1	e3	f2
a1	b2	c2	d2	e4	f3

(c)

図2　非冗長なネットワーク構造

しかし，この構造では質問Q(AB,CD)の処理効率は余りよくない．その理由は次の通りである．

(1) ABの値の指定を満たすものは，R2, R5, R3の経路で求めることができるが，図1(a)ではR1, R2で得られたのに対し検索するレコード型が多い．

(2) R4(F)やR6(E)は解には関係ないが，レコードの対応を求めるために検索する必要がある．

(3) R6からR4，R4からR1は子から親への検索であり，効率がよくない．

(4) R1(C)とR5(D)のレコードの対応はR6の存在により多対多となり，解が重複する．この例では，c1d1が2度求められ，これは図2(c)の関係の組(a1b1c1d1e1f1)と(a1b1c1d1e2f2)に対応している．

ここで対象としている質問とネットワーク構造の質問処理との適合性を次のように定義する．

［定義1］　バックマン線図Bの連結な部分グラフで表されるネットワーク構造をBの部分ネットワーク構造という．特に，属性集合Xに対してXと共通属性を持つレコード型とその間の経路のレコード型，親子集合型からなる部分ネットワーク構造をBXで表す．　□

［定義2］　バックマン線図Bで表されるネットワーク構造に含まれるレコード型をR1, R2, …, Rnとする．レコード集合t＝{r1, r2, …, rn}（riはRiのレコード(1≦i≦n)）でBに親子集合型<Ri, Rj>があれば，riはrjの親レコードのとき，tをBにおけるレコードの組と呼ぶ．また，Bに含まれるレコード型を構成する属性集合の和集合をUB＝{R.A | レコード型Rの属性A}とする．Bの関係はUB上の関係であり，Rel(B)で表す．Rel(B)の各組はBにおけるレコードのすべての組と1対1に対応し，対応する組とレコードの属性の値は等しい．　□

［定義3］　質問Qの解が，部分ネットワーク構造Bの関係Rel(B)に対する属性集合XSの選択及びXPへの射影演算で求められるとき，Qを単純質問Q(XS, XP)，BをQの質問スキーマBQという．　□

バックマン線図B，質問Qに対し，Bの冗長性によりBにおけるQの質問スキーマは一意ではない．質問Q(XS, XP)の処理効率が悪くなるのは，次の場合がある．これらは図2(b)での問題点に

対応している．

(1) X_Sの条件を満たすかどうかの検査が，その属性を含むレコード型が分散しているために遅くなり，解とはならないレコードの対応を処理中に求めなければならない．

(2) 巡航中にX_SX_P以外の属性集合からなるレコード型も巡航する．

(3) 親子集合型の方向とは逆方向の巡航を行う．

(4) 解が重複する．

(1),(2),(4) は，それぞれ選択，結合，射影演算に対する問題点である．

(1)はB_Q中でX_Sの属性を含むレコード型がそれのみで連結であればよく，(2)はX_SX_Pの属性を含むレコード型がそれのみで連結であればよい．従って(1),(2) に対するネットワーク構造が満たすべき条件は次のようになる．

［条件１］　選択属性連結条件：バックマン線図Bと質問$Q(X_S, X_P)$で，$B_Q(X_S)$中の各レコード型$R(X)$について，$X_S \cap X \neq \phi$．　□

［条件２］　質問属性連結条件：バックマン線図Bと質問$Q(X_S, X_P)$で，B_Q中のレコード型$R(X)$について，$X_SX_P \cap X \neq \phi$．　□

(3) については，質問スキーマの形態で特徴付けられる．

［条件３］　有向木条件：バックマン線図Bと質問$Q(X_S, X_P)$で，B_Qは，有向木となる．　□

(4) は，X_SX_Pの対応が一意であればよい．従って，それに対する条件は次のようになる．

［条件４］　非冗長解条件：バックマン線図Bと質問$Q(X_S, X_P)$で，$K(B_Q) = \cup K(R_i) - \cup(anc(R_i) - K(R_i))$　X_SX_P．ここで，R_iは子レコード型を持たないレコード型，$anc(R)$はRの祖先に含まれる属性集合である．　□

ネットワークスキーマBが質問Qについてこれらの条件を満たすとき，QはBでそれぞれ，選択属性連結質問，質問属性連結質問，非冗長解質問であるという．

このらの条件を満たすスキーマを図３に示す．

親子集合型$\langle R_1, R_7\rangle$はR_7のCの値を保持するために必要であるが，CEの対応は本来$R_1R_4R_6$の経路で表されており，$R_1R_7R_6$の経路でのCEの対応はそれに一致するよう更新の際の管理を必要とする．従って，冗長な構造を付加したときには，データの対応を表す構造とそれから得られる対応を表す構造を区別しておくことが重要である．データの対応を表す構造では，構造が従属性を満たしていれば更新は自由にできるが，冗長な構造では，対応を表す構造から得られる値を常に蓄えておかなければならず，更新時のコストが大きくなる．

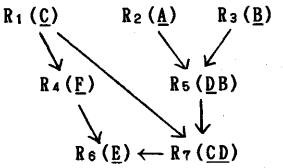


図３　条件１～４を満足する構造

３．オブジェクトを用いた設計[2]

意味的につながりのある属性集合をオブジェクトと呼ぶ．結合従属性の要素はオブジェクトであり，質問に関係する属性も意味的につながりがある．オブジェクトをスキーマに反映させることにより，従属性制約を保持し，質問処理効率の良いスキーマが設計できる．ネットワーク構造が表現するオブジェクトを次のように定義する．

［定義４］　バックマン線図Bで表現されるオブジェクトZは，属性集合Zを含むBの部分ネットワーク構造B'で，すべてのレコードの組のZの値の組に重複がないようなB'が存在する属性集合Zである．　□

Bの部分ネットワーク構造B'に含まれる属性集合$U_{B'}$も表現可能なオブジェクトであるが，データの対応を求める際にレコード型でキー以外の属性は無視することもできる．

［定理１］　ネットワーク構造Bで表現可能なオブジェクトは，Bの部分ネットワーク構造B'で，$\cup K(R_i) \subseteq Z \subseteq U_{B'}$（$R_i$はB'に含まれ，B'中に子レコード型を持たないレコード型）となるB'が

存在する属性集合Zである．そのような属性集合Zの集合をOBJ(B)で表す．　□

（証明）　オブジェクトの定義より，$Z \subseteq U_{B'}$である．$\cup K(R_i) \subseteq Z$であれば，子レコードが定まれば親レコードが定まるので，$\cup K(R_i) \rightarrow U_{B'}$が成り立つことから，$\cup K(R_i)$の値に重複はない，即ちZの値の組に重複はない．$\cup K(R_i) \not\subseteq Z$であればZの値の組に重複が存在し得る．（証明終り）

Algorithm 1：オブジェクト集合OBJを表現するネットワークスキーマの設計

(1) OBJを含み共通集合で閉じた最小の集合OBJ^{i+}を求める．
(2) OBJ^{i+}の各要素Xに対し，節点Xを作る．
(3) 節点$X_1, X_2 (X_1 \subseteq X_2)$で，$X_1 \subseteq X_3 \subseteq X_2$となる節点$X_3$が存在しなければ有向枝$\langle X_1, X_2 \rangle$を作る．
(4) 各節点Xをレコード型R(X)，各有向枝$\langle X_1, X_2 \rangle$を親子集合型$\langle R_1(X_1), R_2(X_2) \rangle$とする．
(5) 一般に，レコード型$R_i(X_i)$のキーはX_iであるが，関数従属性が与えられていればそれを用いてレコード型のキーを決定できる．キーが同じになるレコード型は併合する．
(6) 冗長な属性を除く．　□

図4は，オブジェクト集合を用いた設計の例である．括弧内の属性は(6)で削除される属性を表す．図2の構造と比較すると，A,Bのレコード型がなく，かわりにABのレコード型があることが異なっている．

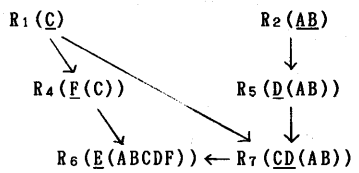


図4　オブジェクト集合を用いた設計

［定理2］　OBJ が結合従属性の要素をすべて含んでいれば，結果のネットワークスキーマでは，結合従属性を表す経路が存在する．　□

（証明）　結合従属性の要素は，関数従属性で分解できなければそれをキーとするレコード型が存在する．それらを共通集合で結ぶ経路は，関係の自然結合に対応しており，この部分で結合従属性を表している．（証明終り）

［定理3］　OBJが関数従属性集合Fに対し，$X \rightarrow Y (\in F)$のXの閉包X^+（$X \rightarrow Z$ となる最大のZ）を含めば，結果のネットワークスキーマは，Fを満足する．　□

（証明）　$X \rightarrow Y (\in F)$について，Xをキーとするレコード型は必ず存在し，X^+はXをキーとするレコード型の祖先に現れる．（証明終り）

質問$Q(X_S, X_P)$はRel(B)上，即ちネットワーク構造Bのすべてにわたって処理する必要はない．$Rel(B)[X_P X_S]$のみを考えればよく，$X_S X_P$がBで表現可能なオブジェクトであればよい．また，選択条件を満たすかどうかの検査が早い方がよく，X_Sも表現可能なオブジェクトであることが望ましい．

［定理4］　質問$Q(X_S, X_P)$に対し，ネットワークスキーマBがオブジェクト$X_S X_P$を表現していれば，QはBにおいて非冗長解質問である．また，適当な属性をレコード型に付加することにより，質問属性連結質問となる．同様にオブジェクトX_Sを表現していれば，適当な属性をレコード型に付加することにより選択属性連結質問となる．　□

（証明）　$X_S X_P$を表現する部分ネットワーク構造B'では，K(B')　$X_S X_P$であり，Qは非冗長解質問となる．また，B'で，$X_S X_P$と共通属性を持たないレコード型に祖先の$X_S X_P$との共通属性を加えることにより，質問属性連結質問となる．X_Sについても同様である．（証明終り）

図2(a)のスキーマは，オブジェクトBEを表現しているが，BEに対し属性連結ではない．R_5に属性Bを加えると連結にすることができる．

関係モデルではビュー設計はデータベース設計とは独立に考えられる．しかし，ネットワーク構造ではビューに対応する構造をスキーマに埋め込まなければならない．オブジェクトを用いることにより，ビューの最適な埋め込みが可能となり，従属性制約を反映していない部分，即ち管理を必要とする部分も明確になる．

4．質問処理効向上のためのスキーマ変換(3)

与えられたスキーマを，効率よい質問処理ができるように変換する．まず，オブジェクトの反映法を示し，次に条件１～４を満たすための変換法を示す．

ネットワーク構造Ｂで表現できないオブジェクトZを表現可能にするために，　Ｂにレコード型，親子集合型を加える．一般に，X＝Zとなるレコード型R(X)を加え，　Zを含むＢの表現可能なオブジェクトを与える部分ネットワーク構造Ｂ'で，　RがRel(Ｂ')[X]に一致するようにRを管理すればよい．しかし，　Ｂ'が大きくなるとRの管理のためのコストが大きくなる．Algorithm 2は，Zを表現するためにＢの部分構造を用い，部分構造が利用できないときのみにレコード型の付加を適用する．

Algorithm 2：オブジェクトZを表現可能にするためのネットワークスキーマＢの変換

(1) Zを含む属性集合で，　Ｂで表現可能な極小のオブジェクトとなるものZmを求める．
(2) Zに含まれる属性集合で，　Ｂで表現可能な極大のオブジェクトとなるものの集合Max(Z)を求める．
(3) Z'＝Zm-∪Max(Z)とする．
(4) Zmに含まれる属性集合で，Z'を被覆するＢで表現可能な少なくとも１つの∪Max(Z)の属性を含む極小のオブジェクトとなるものの集合{Zmi}を求める．
(5) Zmi∩Z＝Ziからなるレコード型Ri(Zi)をＢに加え，属性の包含関係などから親子集合型を加える．　□

Zmi をオブジェクトとする部分ネットワーク構造をＢmiとする．　RiはRel(Ｂmi)[Zi]に一致しなければならない．Max(Z)の要素をオブジェクトとする部分ネットワーク構造にRiを加えたものは，Zをオブジェクトとする．

Algoritm 3は，ネットワークスキーマＢを属性集合Z と共通属性を持つレコード型のみからなる部分ネットワーク構造Ｂz が存在するように変換するためのものである．条件１，２に対しては，ZをそれぞれXs，XsXpとし，　RをZと共通集合を持たないレコード型とすればよい．Algorithm 3をZと共通属性を持たないレコード型がなくなるまで繰り返すことにより，条件を満足するスキーマが得られる．　条件３に対しては，ZをXsXp，　RをＢz中で決めた有向木に含まれず親を持たないレコード型とすることにより，条件を満足するスキーマを得ることができる．

ここで，印付けされた構造は，質問処理のための冗長な構造であることを表す．

Algorithm 3：ネットワークスキーマＢ，　属性集合Z，　Ｂz中の指定されたレコード型Rに対し，次の変換を行う．

(1) Rの親レコード型，子レコード型がＢz中にそれぞれ１つずつ（Ro，Rmとする）のとき，　Ｂ，Ｂzに親子集合型<Ro，Rm>を加え，　ＢzからRを除く．　<Ro，Rm>を印付けする．　<Ro，R>，<R，Rm>，Rが印付けされていれば，　それをＢから除く．
(2) Rの親レコード型がＢz中にないとき，子レコード型を　R1，R2，…，Rn（必ず複数ある）とする．　Ｂ，　Ｂzにレコード型R'(X')
($X' = (\bigcup_{i=1}^{n} X_i \cap Z) \cup \bigcup_{i=1}^{n} K(R_i)$)，　親子集合型<R1，R'>，<R2，R'>，…，<Rn，R'>を加える．　ＢzからRを除く．Rが印付けされていればそれをＢから除く．
(3) その他のとき，　Rの適当な親レコード型をRo(Xo)とする．RにZ∩Xoを加える．　RoがＢz中で他に連結なレコード型を持たなければRoをＢzから除く．　□

Algorithm 4 は，条件３を満たすようにする変換で，　Z＝XsXpとして実行する．Ｂz中で(K(Ｂz)-Z)∩K(R)≠φとなり，　子レコード型を持たないレコード型が存在しなくなるまで繰り返すことにより，条件４を満足するスキーマを得ることができる．

Algorithm 4：ネットワークスキーマＢ，　属性集合Z，　Ｂz中で(K(Ｂz)-Z)∩K(R)≠φとなり，子レコード型を持たないレコード型R(X)に対し，K(Ｂz)がZの部分集合となるように変換する．

(1) Rの親レコード型をR1(X1)，R2(X2)，…，Rn(Xn)とする．Xi∩Z＝φのとき，Algorithm 3を用いてRiを除くか，　Zと共通集合を持つように変換する．

(2) レコード型R'(X')($X' = (X \cup \bigcup_{i=1}^{n} X_i) \cap Z$)，親子集合型<R',R>を作る．

(3) $K(R_i) \subseteq Z$のとき，親子集合型<R_i,R'>を作る．<R_i,R>のR_iとRのレコードの対応は<R_i,R'>，<R',R>によるとRの対応に一致するので，<R_i,R>をBから除く．

(4) $K(R_i) \not\subseteq Z$のとき，レコード型$R_i'(X_i')$($X_i' = K(R_i) \cap Z$)と親子集合型<R_i',R_i>,<R_i',R'>を作る．□

図3は，図2(a)の構造と質問Q(AB,CD)から，3つの条件を満たすように変換したものである．非冗長解条件を満たさなくてもよいとすると，R_7は付加されず，R_1,R_5の子レコード型はR6となりレコード型の数が減る．また，質問属性連結条件も満たさなくてよいのならば，R_6に属性Cは付加されず，親子集合型<R_1,R_6>も不要となり，データベースの更新は容易になる．

5．システムの構成とデータ表現

スキーマの設計は，次の2つのステップからなる．

Ⅰ：基本構造の設計

Ⅱ：構造の解析及び最適化

Ⅰの基本構造の設計には，Algorithm 1 を用いる．そのために，従属性集合および代表的な質問からオブジェクト集合を求めなければならない．すべての質問に対し，選択属性と質問属性のオブジェクトを反映すると，スキーマが複雑になる場合があるので，ここで使用する質問は，特に重要なものに限る．また，自然結合だけでなく，等結合や不等号結合を含むような質問に対しては，結合条件を満たす組をレコード型として加えることにより，単純質問となる．条件A≦Bでの結合は，レコード型R(AB)を加え，A≦Bを満たす組をRのレコードとすることにより，Rを通る経路でこの結合を行うことができる．従って，オブジェクトABを加えればよい．

Ⅰの結果のスキーマに対し，使用されなかった質問について，トレードオフとの兼ね合いから，条件1～4のうち指定されたものを満たすように変換する．また，冗長性が大きすぎ，スキーマが複雑になっている部分では，冗長性を除く．

ネットワークスキーマの設計ステップは次のようになる．

0：データベースに対する要求の解析

(1) データが満たす従属性を求める．

(2) データベースの使われ方から代表的な質問を求める．

Ⅰ：基本構造の設計

(3) 従属性制約から，オブジェクト集合を求める．

(4) 質問集合のうち，オブジェクトとして表現するものを決定する．

(5) (4)の質問のオブジェクト集合を求める．

(6) (3)と(5)のオブジェクト集合から，Algorithm 1 を用いてネットワークスキーマを設計する．

Ⅱ：構造の解析及び最適化

(7) Ⅰで使用されなかった質問に対し，条件1～4のうちどれを満たせばよいかを決定する．

(8) Algorithm 3,4を用いて，(7)の各質問を指定された条件を満たすように変換する．

(9) Algorithm 2 を用いて，さらに表現したいオブジェクトを表現可能となるように変換する．

(10) 冗長性が大きすぎる部分では，質問処理効率との兼ね合いを考慮して，冗長な構造を削除する．

(7)～(10)を繰り返す．

ステップⅡでは，各質問の選択属性や質問属性を用いるのではなく，部分質問に対して条件を満たすようにしても，処理効率には効果があり，冗長性の増加を抑えられる場合もある．例えば，質問属性がABCDE である質問を質問属性連結質問にするようにスキーマを変換するとあまりにも冗長性が大きくなる場合には，質問属性がABC,DEである2つの質問に対して質問属性連結質問となるように変換するような場合である．

設計支援システムは，(3),(5),(6),(8),(9) および(10)の冗長性の削除を行う．システムの利用者，即ち設計者は，代表的な質問の処理効率をあげるための冗長性をどの程度にしたらよいかということを，実際にスキーマを変換しながら決定することができる．

システムは，現在C言語を用いて開発中である．アルゴリズム中では，属性の包含関係の判定

や親レコード型，子レコード型を求める部分が多く，それを効率よく行うため，データをビットマップを用いて表現している．これにより，和集合，共通集合はビットごとの論理和，論理積で，包含関係は共通集合との排他的論理和で求めることができる．また，レコード型にはすべて番号を与え，それにより管理している．具体的には次のような構造体の配列を定義する．

```
struct node{
    int attributes;
    int key_attributes;
    int owner;
    int member;
} rec_type[number of record types]
```

例えば，番号３のレコード型の属性がCD，キー属性がC，親レコード型の番号が２，子レコード型の番号が４，５であるレコード型は次のようになる．ここで，0xは次に続く数が１６進数であることを表す．

```
rec_type[3].attributes = 0x3000
rec_type[3].key_attributes = 0x2000
rec_type[3].owner = 0x4000
rec_type[3].member = 0x1800
```

６．むすび

質問処理を考慮してネットワークスキーマ設計を支援するシステムの開発について述べた．処理効率は，適当な冗長性を付加することによって向上する．スキーマと質問処理との適合性の尺度として，４つの条件があり，スキーマはそれらの条件を満足するように設計される．　本システムでは，冗長性と処理効率のトレードオフは，システムの利用者が判断しており，各質問に対し，どの条件を満たすスキーマとするかは，利用者の指定によっている．現在，Algorithm 3,4 のプロトタイプを開発中である．これを用いて様々な場合を検査することにより，スキーマの冗長性の尺度を決め，トレードオフに対する決定も利用者を支援できるようにしたい．

参考文献

(1) Dayal,U. and Bernstein,P.A., "On the Updatability of Network View - Extending Relational View Theorey to the Network Model", Information Systems, Vol.7, No.1, pp.29-46, Jan. 1982.

(2) 古川，上林，"オブジェクト集合を用いたネットワークデータベースの設計"，情報処理学会研究報告，DB-60-7，昭和62年7月．

(3) 古川，上林，"質問処理効率化のためのネットワークデータベースのスキーマ変換"，電子情報通信学会論文誌，昭和63年（予定）．

(4) Kambayashi,Y., Furukawa,T., "Semantic Constraints Expressed by Network Model", Proc. Int. Conf. on Foundations of Data Organization, pp.201-206, May 1985.

(5) 川口，溝口，山口，角所，"データベースの論理設計を支援する知的インタビューシステム"，情報処理学会研究報告，AI-48-1，昭和61年9月．

(6) Kuck,S.M. and Sagiv,Y., "A Universal Relation Database System Implemented Via the Network Model", Proc. 1st ACM Symp. on PODS, pp.147-157, March 1982.

(7) Kuck,S.M., Sagiv,Y., "Designing Globally Consistent Network Schemes", Proc. ACM SIGMOD Int. Conf. on Management of Data, pp.185-195, May 1983.

(8) Lien,Y.E., "On the Equivalence of Database Models", J. ACM, Vol.29, No.2, pp.333-362, April 1982.

(9) 溝口，磯本，角所，"データベース論理設計支援エキスパートシステム"，ＩＣＯＴ研究論文，昭和59年．

(10) 滝沢，横塚，鈴木，"CODASYL データベースシステムに対する関係インタフェースシステム(LDP-V1.5)の設計と実現"，情報処理学会論文誌，第23巻，第6号，pp.665-675，昭和57年11月．

(11) Ullman,J.D., Principles of Database Systems, 2nd ed., Computer Science Press, 1982.

(12) Yannakakis,M., "Algorithms for Acyclic Database Schemes", Proc. 7th Int. Conf. VLDB, pp.82-94, Sept. 1981.