データベース・システム 69－1
（1989. 1. 19）

関係データベースにおける集合制約質問

岩井原瑞穂　　上林彌彦
九州大学工学部

全ての要素が互いに似ているとか，近くにある，独立である，などの性質を満たす集合を求めることができる，集合制約質問について考察する．質問は集合の性質を表す制約として与えられ，答えは制約を満たす，基底関係の部分集合となる．制約は要素の間で定まる相対的な条件であり，個々の要素が満たすべき条件ではない．制約を記述するために，組集合上で定義された，反射的かつ対称的な二項関係を用いる．解は任意の２要素が二項関係を満たす集合となる．また，集合制約質問の部分クラスである，ＦＤ型集合制約質問を定義し，その処理法について検討する．組合せの問題は，最大の要素数の解が最適であることが多い．ＦＤ型の質問の最大の解を求める問題について，条件のＦＤが２つまでの多項式アルゴリズムを示す．あわせて，ＦＤが３つ以上の場合はＮＰ完全となることを示す．

Set Constraint Queries in Relational Databases

Mizuho IWAIHARA and Yahiko KAMBAYASHI

Department of Computer Science and Communication Engineering, Kyushu University

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

We propose Set·Constraint Queries, that can generate sets which satisfy conditions like all elements are similar, located near, independent, etc. The queries are given as "constraints", and the solutions are subsets of database relations which satisfy the constraints. The constraints are relative conditions between elements, not the conditions each element must satisfy independently. To denote the constraints, we use reflexive and symmetric binary relations defined on tuple sets, and the solutions are tuple sets where any two tuples satisfy the binary relations. We also define a subclass of Set Constraint Query whose constraints are FDs, and consider some methods to process it. Since maximum cardinality solutions are often optimum and necessary, we present polynomial algorithms which generate maximum solutions of FD Constraint Queries in cases the number of the given FDs is at most two, and show that the problem is NP-complete when the constraints consist of more than two FDs.

1. まえがき

データベースに対する質問の答えは，条件を満たす組またはオブジェクトの集合であることが多い．これに対して，答えが条件を満たす集合，または組合せの集合であるような質問が考えられる．集合の個々の要素が条件を満たすだけではなく，集合全体として条件を満たすような質問である．すなわち，集合の要素それぞれについてでは意味はなく，集合としてまとめられたとき，初めて意味を持つような集合を求めることになる．このような条件を満たす集合を求める質問を，本稿では集合制約質問と呼ぶ．

例えば，次のような質問である．

【例１．１】 次のような組み合せを求める．

- 各人の時間帯が重ならない作業の割り当て
- 互いの距離がｒ以下の部品の集合
- 年齢が５才離れている人のペア □

データベース理論において，集合が陽に構成要素として扱われる例として，オブジェクト指向データベースや，非正規関係などが挙げられる[13]．これらにおいては，集合を用いて１対多の対応を容易に表せることにより，関係データベースが不得手とするような，複雑なデータ関連を自然に表すことができる．このような場合，集合の構造はスキーマとして静的に固定されて扱われることが多く，集合制約質問のように，動的に集合を生成する機能はあまり考慮されていない．また，個々の要素がある性質を満足するような集合を考えるのが普通である．

関係データベースの利用者インタフェースとして，非正規関係を用いることにより，冗長度が小さく分かりやすい表現が得られるが，正規形から非正規形への変換，すなわち集合を生成する操作はNEST/UNNEST[7]など，基本的に同じ値を持つ組を１つにまとめる操作のみで，組合せの条件を記述するには，表現力が不足する．

論理データベースにおいては，再帰質問[3]を用いて推移的閉包や，親子関係での同世代の集合などを求めることが出来る．これらの質問は，最小不動点が存在することにより，答えは１つの集合となるが，組合せを求める問題では，答えは複数の集合となるのが普通である．集合制約質問では，推移的閉包を求めることもできる．

集合を構成要素として陽に扱う重要な分野として，論理データベースに集合型を導入する研究がある[2][9][10]．これらの文献では，述語論理の強力な記述力を用いた，大変表現力の高いデータベース言語が提案されている．本稿で考察する質問も，これらの言語で表すことができる．これらの言語の処理系を実現する場合に，集合を扱うときに生じる特有の問題がある．例えば，答えの集合の数が莫大なものになる場合や，答えを計算するときの手間が大きい場合である．このような困難な質問も言語で記述できることは問題があり，除外されるように制限することが望ましいが，そのためには容易に求められる質問のクラスが明確にされねばならない．この点はまだあまり議論されていない．本稿では，集合制約質問で現実的な時間で求められる質問のクラスについても考察している．

集合制約質問の機能を備えるためには，集合全体を制約するような，条件を明確に記述できる手段が必要となる．本稿では，反射的・対称的である二項関係関係を与えられた条件とし，この二項関係の数学的性質より定義される集合を，質問の答えとする方法をとる．

例１．１においては，二項関係γ_rを用いて，答えＳは次のように表される．

$$t_1\ \gamma_r\ t_2 \quad \text{iff} \quad \| t_1[\text{座標}] - t_2[\text{座標}] \| \leq r$$

$$\forall t_1, t_2 \in S \quad ; \quad t_1 \gamma_r t_2$$

二項関係を質問に利用した例としては，利用者が定義した二項関係を比較演算子として用いる方法[8]などがあるが，集合を定義するために利用した例は，著者の調査した範囲では見あたらない．

以下２節では基本的事項を説明し，３節では集合制約質問の定義及び質問の分類を行う．そしてその部分クラスである，ＦＤ型集合制約質問について述べる．４節ではその処理法を検討する．質問の条件のＦＤが１つ及び２つの場合の多項式アルゴリズムを述べ，３つの場合はＮＰ完全になることを示す．

2. 基本的事項

まず関係データベースについて説明する．

属性集合$X = \{A_1, A_2, ..., A_n\}$の各属性$A_i$は，定義域$D_i$を持つ．属性集合Ｘからなる関係Ｒ(Ｘ)は，$t : \{A_1, A_2, ..., A_n\} \rightarrow D_1 \times D_2 \times \cdots \times D_n$で表される写像ｔの有限集合であり，写像により各$A_i$は$D_i$の要素に写されるものとする．Ｘを関係スキーム，ｔをＲの組と呼ぶ．関係Ｒは列を属性，行を組とした，表の形で表現できる．本稿では，属性をＡ，Ｂ， Ｃ，･･･で表し，属性集合をＸ，Ｙ，･･･で表す．属性集合の連接は，その和集合を表す．

次に，二項関係及びその数学的性質について述べる．二項関係とは，集合Ａと集合Ｂの直積Ａ×Ｂの部分集合$R\gamma$（$R\gamma \subseteq A\times B$）のことである．$(a_i, b_j)\in R\gamma$ということを表わすために，$a_i\gamma b_j$という記法を用いる．

二項関係αが定義されている直積集合の２つの項が，同一の集合Ａである場合，すなわち$R\alpha \subseteq A\times A$のとき，$\alpha$はＡ上の関係であると言う．このとき，種々の重要な性質を生じる．

集合Ａ上の二項関係α，及びＡの任意の要素a,b,cについて，

- $a\alpha a$ のとき αは反射的(reflexive)，
- $a\alpha b$ ならば $b\alpha a$ のとき αは対称的(symmetric)，
- $a\alpha b$，$b\alpha c$ ならば $a\alpha c$ のとき αは推移的(transitive)という．

集合Ａ上の二項関係αについて，

- αが反射的，対称的であるとき，αは同調関係(compatible)であるという．
- αが反射的，対称的，推移的であるとき，αは同値関係(equivalence)であるという．

同調関係と同値関係は集合を分類する性質がある．$a\alpha b$のとき，節点aと節点bに枝があるとして，同調関係αを無向グラフを用いて表すことができる．

Ａ上の同調関係γが与えられたとき，Ａの部分集合Ｃであって，Ｃの任意の元a,bについて，$a\gamma b$ が成り立つものは，γによる同調類(compatibility class) と呼ばれる．さらにどのような同調類にも真に含まれることのない同調類は極大同調類と呼ばれる．同調類は，対応する無向グラフ上では，節点の部分集合で，集合内の任意の２要素の間に枝がある，つまり完全グラフをなす部分に対応する．完全な部分グラフのことをクリークともいう．

３．　集合制約質問

３．１　集合制約質問の定義

同調関係により定まる同調類は，同調関係による制約を受けた集合であるとみなすことができる．その集合は同調関係の定義を反映した性質を持つ．これを用いて，任意の要素が互いに似ているとか，同じである，近くにある，互いに独立である，交換可能である，などの集合を求めることができる．関係の組を１つの実体とし，組の属性値を実体の性質とみなし，質問で得たい組集合の特徴を，属性値で定義された２組間の同調関係で表すことにより，解の組集合を定めることができる．

【定義３．１】

関係Ｒ（Ｘ）の組集合を定義域とする，反射的かつ対称的な二項関係γが与えられたとき，γをＲ（Ｘ）上の集合制約質問と呼ぶ．質問の解は次の条件を満たす集合Ｓである．

$$\forall t_1, t_2 \in S \subset R(X);\quad t_1\gamma t_2 \qquad \square$$

定義３．１では，Ｓが解ならばその部分集合も解となり，解の数は指数関数的になる．そのため，次の方法が考えられる．

① 極大集合を解とする．

しかしn個の点からなるグラフの 極大クリークの最大数は，$3^{n/3}$であるため， 全ての解を出力することは適当でない．

② 極大集合の解を１つ出力する．

これは，組を解に１つづつ加えてゆくことにより，極大にでき，多項式時間で可能である．しかし出力される解は組の入力の順序により異なったものとなる．

③ 最大集合の解を１つ出力する．

組合せを求める問題では，要素数最大の組合せが最適であることがよくある．しかし最大クリーク問題はＮＰ完全であり[11]，任意の質問に対する処理法を求めることは難しい．

３．２　これまでの質問言語との対応

論理型データベース言語で，集合が扱えるように拡張されているものがあるが，このうちでＬＤＬ１[2]とＥＬＰＳ[9][10]は，表現力が等しい言語であることが知られている[10]． 集合制約質問は，ＥＬＰＳの節を用いて，次のように表せる．

$$P(X):-(\forall x_1 \in X)(\forall x_2 \in X)\,C(x_1, x_2)$$

述語Ｃは同調関係を表し，また組x_1，x_2から属性値への関数記号が含まれていることとする．さらに否定を用いて極大，最大を表すことができるが， ３．１節で述べた①②③の問題点も存在することになる．また集合の階層の深さを任意に許すと，解であることを判定するのに少なくとも指数時間を必要とする質問も書ける[5]．多項式時間で求められる質問しか書けないように，言語を制約することが重要となる．

推移的閉包を求めることのできる，不動点質

問と呼ばれる質問のクラスがある[3]. これは,要素を集合に加えて単調に要素数を増加させてゆき，要素が増えなくなる不動点が，最小不動点となる条件を必要とする．集合制約質問では,推移的な二項関係を定義することにより，推移的閉包を求められる．しかし推移的でない場合,極大集合が不動点となるが，極大集合同士の共通集合は極大とはならないため，最小不動点は存在しないことになる.

NEST/UNNEST[7]による非正規関係代数は，推移的閉包を表せないことが知られている[12].また後で述べるように，集合制約質問で，集合の深さが1のNESTを表すことができる.

次に計算複雑度の観点から検討する．質問の計算複雑度は，ある組が質問の解集合に含まれるか判定する問題の複雑度であるが，２つの種類がある．１つはデータベースサイズの関数であるデータ複雑度と，もう１つは質問の記述サイズの関数である記述複雑度である．以下本稿では，前者について考える.

関係論理／代数や，不動点質問はＰ，つまり多項式のクラスである[3]. 集合制約質問は最大集合を求める場合は，ＮＰのクラスに含まれることになる．中間の計算に，深さ１の集合変数を用いるクラスである$CALC_{0,1}$[5]は，第二階言語のクラスと一致する．その部分クラスである，集合変数が存在記号のみで束縛されているクラスの$CALC_{0,\exists_1}$は，述語変数が同じく存在記号で束縛されていた第二階言語のクラスと一致する．しかも$CALC_{0,\exists_1}$はＮＰのクラスと一致する．集合制約質問を中間の計算に用いる，解ががフラットな言語は，$CALC_{0,\exists_1}$のクラスに含まれるといえる.

３．３　質問の分類

集合制約質問は，集合の定義に二項関係を用いているため，その定義の方法により，以下のような特徴づけができる.

①　基本論理式

述語論理において，真偽が与えられる最小単位の述語である．例えば２組の距離がｒ以下であるという，例１．１で定義した論理式γ_rなどである．他にも，２つの組のある属性の値が等しいという論理式等がある.

②　再帰的定義を含む場合

互いに連結している部品である，というようなことを表せる.

③ ∃，∀などの限量子で修飾されている場合

④　ブール論理式

基本論理式がAND・OR・NOTで結合された，すなわちブール論理式で定義されたクラスがある.これはかなり単純なクラスと考えられる.

ブール論理式で定義された質問の例を示す.

【例３．１】

質問Q_3：図３．１に示す関係講義(科目,時限)は行われている全ての講義の科目と時限を表す．同じ科目の講義がいくつかの異なる時限で行なわれており、また同じ時限でいくつかの講義が行なわれているとする。

ここである学生が自分の時間割をつくることを考える。学生は同じ科目を２つ以上受講しないし,また同じ時限で２つ以上の科目は受講できない。そしてできるだけ多くの講義を受講できる時間割をつくりたい.

科目	時限
英語	火２
英語	火２
物理	月２
物理	水１
数学	火２
数学	月２
数学	水１
化学	木１
化学	金２

図４．１

可能な講義の組み合せからなる講義の集合は次の条件を満たす。

①　各講義の科目はそれぞれ異なる。

②　同じ時限の講義は存在しない。

③　集合の要素数は最大である。

①と②より，関係・講義上の同調関係γ_cを定義する.

$$t_1 \ \gamma_c \ t_2 \quad \text{iff} \quad t_1 = t_2 \ \vee \ (t_1[\text{科目}] \neq t_2[\text{科目}] \ \wedge \ t_1[\text{時限}] \neq t_2[\text{時限}])$$

質問の答はγ_cの要素数最大の同調類である．□

質問Q_3は３．１節の③で述べた最大の集合を１つ求める質問である.

従来の関係データベースシステムでは，このような質問は応用プログラムによって，手続き的に記述されていた.

３．４　従属性を用いた質問の変換

従属性はデータベース内の事実として，与えられている条件である．ＦＤ，ＭＶＤなどは,二項関係で表されることによって，集合制約質問と同じレベルで取り扱うことができる．意味制約により，質問の簡単化を行うことを検討する．例えば，関係Ｒ(ＸＹＺ)がＦＤ：Ｘ→Ｙは,

以下のように表せる．

$$\forall t_1, t_2 \in R \;;\; t_1[X]=t_2[X] \Rightarrow t_1[Y]=t_2[Y]$$ □

"⇒"の部分を置き換えることにより，ＦＤは属性の等式による述語と，そのブール論理式で表せることがわかる[4].

【例３．２】　質問Q_3の条件は，以下に示すＦＤの集合に書き換えることができる．

F_1 : 科目→時限

F_2 : 時限→科目 □

質問処理の効率化に，意味制約は以下のように用いることができる．

(1)　条件の削減

意味制約はデータベース内での公理であるとして，質問の定義の中で意味制約と同じ節を，trueに置き換えることができる．この削減により，条件が恒真となったとき，答えは対象としている関係の組全体からなる集合１つである．

(2)　矛盾

質問の定義と意味制約を合わせた論理式が，矛盾を生じる場合，質問の答えは空集合となる．

(3)　singleton

質問の定義と意味制約により，答えである集合すべての要素数が，１つしかないことが証明できる場合がある．つまり，

$t_1 \gamma t_2 \Rightarrow t_1=t_2$　が導けるときである．この場合，対象としている関係に含まれる１つの組が，答えである要素数１の集合となる．

例として，関係Ｒ(ＸＹ)(Ｘ∩Ｙ＝ϕ)上で，次の二項関係γの質問を考える．

$t_1 \gamma t_2$ iff $t_1[X]=t_2[X]$

これは関係Ｒ(ＸＹ)にNEST(Ｙ)または，GROUP-BY(Ｘ)を行うことと等しい．またＦＤ：ϕ→Ｘを満たす関係を求めることとも等価である．ここでさらにＲ(ＸＹ)上で，ＦＤ：Ｘ→Ｙが成立していると仮定する．すると解はすべて要素数１となる．

３．５　ＦＤ型集合制約質問

ここでは，集合制約質問のある部分クラスについて考察する．

【定義３．２】 Ｒ(Ｘ)の組t_1, t_2について，属性Ａ⊂Ｘについての基本論理式$t_1[A]=t_2[A]$を正リテラルと呼び，単に$\underline{A}$で表す．正リテラルの否定を負リテラルと呼ぶ．正または負リテラルのAND・OR・NOTの結合で定義されたＲ上の二項関係をブール型二項関係と呼ぶ． □

ブール型二項関係は必ずしも同調関係ではない．同調関係であるためには，主乗法標準形に変換したとき，各和項に少なくとも１つ正リテラルを含まねばならない．

ブール型二項関係の部分クラスを定義する．

【定義３．３】　ブール型二項関係γで表される集合制約質問について，γを主乗法標準形にしたとき，各和項に１つ正リテラルを含むものを，ＦＤ型集合制約質問と呼ぶ． □

ＦＤ型同調関係の各和項

$\underline{A_i} \vee \neg \underline{B_{i1}} \vee \neg \underline{B_{i2}} \vee \cdots \vee \neg \underline{B_{ik(i)}}$は，

ＦＤ：$B_{i1}B_{i2}\cdots B_{ik(i)} \rightarrow A_i$を表す．

ＦＤは同調関係であるが，同値関係ではない．なぜならば，ＦＤ：Ｘ→Ｙにおいて，

$t_1[X] \neq t_2[X]$，$t_2[X] \neq t_3[X]$，$t_1[X]=t_3[X]$が成り立つとすると，$t_1 \gamma t_2$，$t_2 \gamma t_3$であるが，$t_1[Y] \neq t_3[Y]$のとき，$t_1 \gamma t_3$は成立しないからである．

ブール型二項関係質問は，論理関数の場合と同じ手法で簡単化できる[4].

４．　ＦＤ型集合制約質問の処理法

本節では，ＦＤ型質問の解を求めるアルゴリズムについて考察する．条件のＦＤ集合は，極小被覆に変換されているとする．ＦＤ集合の要素数により，場合分けをする．

４．１　ＦＤが１つの場合

まず３．１節の①に対応する極大の解を求めるアルゴリズムを示す．

【アルゴリズム１】

入力 : 関係Ｒ(組数m)，Ｒ上のＦＤ　F_1

ステップ１： 関係ＲをＦＤの左辺が等しい組の集合$S_1, \ldots, S_p$に分ける．

ステップ２． 各S_iについてS_iをＦＤの右辺が等しい組の集合$S_{i1}, \ldots, S_{iq(i)}$に分ける．

ステップ３： $S = S_{1r(1)} \cup S_{2r(2)} \cup \cdots \cup S_{pr(p)}$　$(1 \leqq r(i) \leqq q(i))$とする．

Ｓは極大集合である． □

【例４．１】　３節で示した質問Q_3について，アルゴリズム１を適用する．関係・講義及びF_1：科目→時限を入力した場合のステップ２の結果を図４．１に示す．$u_1, .., u_4$から１つづつ組を選べば，F_1を満たす集合が得られる．F_2を入力した場合を図４．２に示す． □

アルゴリズム１のステップ２までは，ｍの多項式時間で求められる．しかしステップ３では，３．１節で述べたような，解の個数がｍの指数関数に比例するときがあり，多項式時間で求められない場合がある．そのような例を示す．

$m=n^2$とし，ステップ１において，各S_iの要素数がｎ，ｐ＝ｎとなったとする．さらに，ステップ２において，$q(i)=n$，各$S_{iq(i)}$は要素数が１つになったとする．このとき，全ての極大かつ最大の解の個数はn^nとなる．

この例のように，ＦＤ型集合制約質問においても，全ての極大の解の個数は莫大なものとなる．３．１節の② ③の，極大または最大の解を１つ求めるのが適当である．

１つのＦＤの質問の最大の解を１つ求めるには，アルゴリズム１のステップ２に続いて，各ｉについて，$S_{i1},...,S_{iq(i)}$のうちで最大の要素数のものを求め，それを$S_{it(i)}$とすれば，$S=\cup_{i=1,...,p} S_{it(i)}$ を解とすればよい．

４．２　　ＦＤが２つのときの最大の解

ＦＤが複数になると，１つの場合のように，単純なアルゴリズムを設計することは難しい．以下において，次の条件を満たすＦＤについて考察する．

【条件４．１】　Ｒ(Ｘ)上のＦＤ型集合制約質問の条件のＦＤ：Ｙ→Ｚについて，Ｙ∪Ｚ＝Ｘとなる．□

条件４．１を満たすＦＤの左辺は，Ｒ(Ｘ)のキーとなる．このため，解はＦＤの左辺の値が皆異なる組からなる集合である．これにより，質問Q_3のような，ある属性の値が皆違う，といった集合を表せる．

条件４．１を満たす２つ以上のＦＤからなる質問において，最大の解を求める問題は，以下のようにして最大独立点集合問題に帰着できる．

質問のＦＤの左辺の値が等しい２つの組は，同時に解の集合に含まれない．またその逆も成り立つので，各ＦＤについて，左辺の値の等しい組に対応する点を，超枝で覆った超グラフを求めれば，解はその超グラフの独立点集合となる．ここでは，超枝は普通のグラフのクリークに対応し，超枝からは１つしか点がとれないことを表す．

任意のグラフに対して最大独立点集合を求める問題はＮＰ完全である[11]が，制限したグラフでは多項式時間のアルゴリズムが存在する場合もある．

条件４．１を満たす２つのＦＤによる質問の最大の解を求めるアルゴリズムを示す．そのために，次の性質を用いる．

【補題４．１】

ある超グラフＨの独立点集合Ｐは，次の条件を満たす超グラフＨ’が存在するならば最大である．Ｈ’はＨの全ての点を覆う．その各超枝は，Ｐの点に１対１に対応し，互いに素であり，Ｈのどれかの超枝の部分集合となる．

証明　Ｈの１つ超枝からは，１つしか独立点が取れないので，補題の条件を満たす超グラフがあれば，その超グラフの超枝の数よりも多く独立点が取れないからである．□

１つの独立点をとる度に，その点を含む超枝をグラフから除いて行き，グラフが空になったとする．そのとき独立点と除いた超枝に１対１の対応がつけば，その独立点集合は最大であることが補題４．１から分かる．

以下のアルゴリズム２では，まず２つのＦＤに対応する，２色の超枝からなる超グラフを作る．そのとき，同色の超枝は互いに素となる．１つの超枝にしか含まれない点を独立点として取ってゆき，全ての点が２つの超枝に含まれるようになれば，グラフを２部グラフに変換し，最大マッチングを求めるアルゴリズムを適用する．

【アルゴリズム２】

入力： 関係Ｒ(Ｘ)及び条件４．１を満たすＦＤ　F_1，F_2

出力：F_1，F_2を満たすＲ(Ｘ)に含まれる最大の組集合

ステップ１： Ｒ(Ｘ)を，F_1の左辺の値が等しい集合$S^1_1,...,S^1_r$に分け，同様にF_2について$S^2_1,...,S^2_s$に分ける．すべてのS^1_i，S^2_i について，それを超枝として持つ超グラフＨをつくる．

ステップ２： 他の超枝に含まれる超枝があれば，それを除く．

ステップ３： １つの超枝にしか含まれない点 ｔ があれば，それを独立点とし，ｔ を含む超枝をＨから除く．

ステップ４： 適用できなくなるまでステップ２，３を繰り返す．Ｈが空になれば停止する．

ステップ５： Ｈの組は全てどれか２つの超枝に含まれている．ある２つの超枝の共通部分に点が２つ以上含まれているならば，点を除いて共通部分が１つの点になるようにする．

ステップ６： Ｈの任意の点ｔに対して，F_1の超

枝およびF_2の超枝がただ1つ存在して，それらの超枝の共通部分が tとなる．この性質より，HのF_1の超枝の集合に点集合U，及びF_2の超枝の集合に点集合Vが それぞれ1対1に対応し，超枝の共通部分の点の集合に1対1に対応する枝集合Eからなる，2部グラフB＝(U，V，E)を作る．

ステップ7： Bの最大マッチングMを求めるアルゴリズム[11]を適用する．得られたマッチングに対応するHの点を，独立点とする．□

【定理4．1】 アルゴリズム2は正当であり，R(X)の組数mの多項式時間で停止する．

証明 ステップ4までで得られた独立点は，除かれた超枝と対応づけられる．残りの独立点に，それを含む超枝が対応付けられればよい．

ステップ7で求めるBの最大マッチングMは，Hの独立点集合に対応し，しかも最大である．なぜならば，Bの点，枝はそれぞれHの超枝，点に対応しているため，Bの最小点被覆Cは，Hの超枝による最小被覆である．2部グラフはCとMの数は同じで，Mの両端の点のうちどちらかがCの点となる性質がある．これより，Mに対応する独立点を含む，補題4．1の条件を満たす超グラフが存在する．

ステップ1は$O(m\log m)$である．ステップ2の超枝の数は$O(m)$であり，ステップ2－3は$O(m^3)$でできる． ステップ6の2部グラフの枝と点の総数は$O(m)$であり，ステップ7のアルゴリズムは$O(m^2)$ のものが知られている[11]．□

【例4．2】 質問Q_3に対して，アルゴリズム2を適用する．ステップ1の結果は 図4．3の超グラフとなる．$u_1,...,u_4$は互いに素であり，$v_1,...,v_5$も同様である．

ステップ2でv_4，v_5が除かれる．ステップ3でt_8，t_9のうちどちらか，ここではt_8を独立点とする．u_4が除かれる．

ステップ6の2部グラフは図4．4となる．この最大マッチングを求めると，例えば枝 (u_1, v_1)，(u_2, v_3)，(u_3, v_2)がマッチングとなる．この枝にはそれぞれ組t_1, t_4, t_5が対応し，これらが超グラフの独立点となる．図4．5が質問Q_3の解であり，F_1，F_2を満たしている．□

4．3 FDが3つ以上のときの最大の解

ここでは，FDが3つの場合の集合制約質問の最大の解を求める問題が，NP完全であることを証明する．これより，3つ以上のFDの場合は，大変困難になると考えられる．

3つのFDの問題はNPに含まれる．NP完全であることを示すために，浅野[1]の示した方法を応用する．

文献[1]では，平面上の様々な幾何図形によって構成される交差グラフの最大独立点集合を求める問題が，NP完全であることを示している．その証明法の概略は，以下の通りである．

まず，NP完全である平面3充足問題を，制限された平面グラフの最大独立点問題に帰着する．平面3充足問題とは，一般の3充足問題において，論理和項とそれに含まれる変数を結ぶ2部グラフが，平面グラフである場合に制限した問題である．

ここでいう制限された平面グラフとは，偶数個の点からなる循環グラフの集合を 図4．6に示すような3つの点からなる部分グラフで結合した形である．循環グラフの1つの点に2つ以上の部分グラフが結合することはない．

	科目	時限	
t1	英語	火2	u1
t2	英語	火2	
t3	物理	月2	u2
t4	物理	水1	
t5	数学	火2	u3
t6	数学	月2	
t7	数学	水1	
t8	化学	木1	u4
t9	化学	金2	

図4．1 F_1の適用

	科目	時限	
t1	英語	月2	v1
t6	数学	月2	
t3	物理	月2	
t2	英語	火2	v2
t5	数学	火2	
t4	物理	水1	v3
t7	数学	水1	
t8	化学	木1	v4
t9	化学	金2	

図4．2 F_2の適用

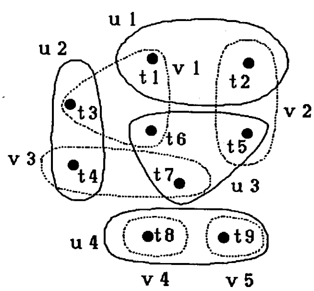

図4．3 超グラフH

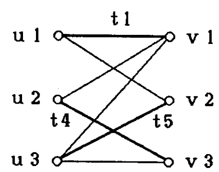

図4．4 2部グラフB

	科目	時限
t1	英語	月2
t4	物理	水1
t5	数学	火2
t8	化学	木1

図4．5 最大の解

以下，この平面グラフをG_1とし，任意のG_1の最大独立点問題が，３つのＦＤの質問の最大の解を求める問題に帰着できることを示す．

平面グラフG_1の各点に１対１で対応する組を持つ，３つの属性からなる関係Ｒ(ＡＢＣ)をつくる．そして，G_1の最大独立点集合が，Ｒの次の３つのＦＤからなる質問の最大の解になるようにする．

F_1: $A \to BC$，F_2: $B \to CA$，

F_3: $C \to AB$

各組の値は以下のようにして定める．G_1の各点を，３つのＦＤに対応する色を持つ，３色の超枝の集合で覆う(図４．７)．G_1の循環グラフの偶数個の点は，F_1とF_2の超枝で交互に覆う．３角形の部分グラフはF_1の超枝で覆い，３角形と循環グラフを結ぶ枝の２点はF_3の超枝で覆う．

次にG_1の全ての点について，３色のＦＤのうち，どれかの色の超枝に含まれていなければ，１つの点からなるその色の超枝で，その点を覆う．以上の操作により，G_1の全ての点は，３つの色の異なる超枝で覆われ，同色の超枝は互いに素となる．

各色の超枝の集合について，順に番号をつける．各点に対応する組の属性の値に，ＡはF_1，

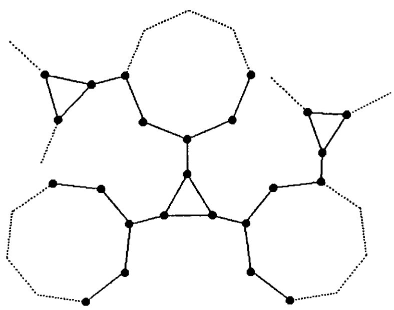

図４．６　平面グラフのＮＰ完全問題

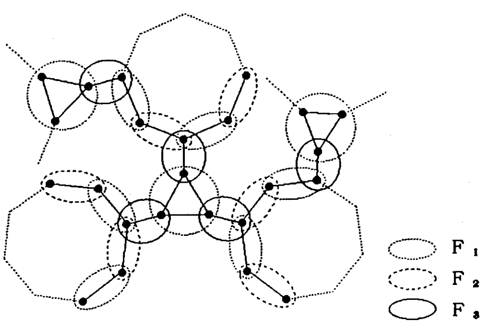

図４．７　集合制約質問への帰着

ＢはF_2，ＣはF_3のそれぞれの超枝の番号を割り当てる．以上により，G_1の任意の独立な２点は，Ｒの３つのＦＤによる二項関係を満たす２組となる．この変換は多項式時間でできることは容易に分かる．以上により次の定理を得る．

【定理４．２】　３つの異なるＦＤによる集合制約質問は，ＮＰ完全である．　□

5.　まとめ

集合制約質問とその記述法について考察し，部分クラスであるＦＤ型集合制約質問の処理法について検討した．多数ある解のうちの１つを求める場合について考えたが，できるだけ利用者の要求に沿う組合せを求めるためには，対話的に条件を加えて，解を絞り込んで行く方法が考えられる．

参考文献

[1] Asano,T., "Difficulty of the Maximum Independent Set Probrem on Intersection Graphs of Geometric Objects", 電子情報通信学会技術研究報告COMP88-19.

[2] Beeri,C., Naqvi,S., Ramakrishnan,R., et al, "Sets and Negation in a Logic Database Language(LDL1)", ACM PODS 1987,pp.21-37..

[3] Chandra,A.K., "Theory of Database Queries", ACM PODS 1988, pp.1-9.

[4] Delobel,C., Casey,R.G., "Decomposition of a Data Base and the Theory of Boolean Functions", IBM J.Res.Develop. 17, pp.374-386, 1973.

[5] Hull,R., Su,J., "On the Expressive Power of Database Queries with Intermidate Types", ACM PODS 1988,pp.39-51.

[6] 岩井原，上林，"二項関係によるデータベースの集合化質問"，昭和63年度電気関係学会九州支部連合大会論文集,pp.930.

[7] Jaeschke,G., Schek,H-J."Remarks on the algerbra of non-first-normal-form relations", ACM PODS, pp.124-138,1982.

[8] Kambayashi,Y., Tanaka,K., Yajima,S., "A Relational Data Language with Simplified Binary Relation Handling Capability", VLDB, pp.338-350, 1977.

[9] Kuper,G.M., "Logic Programming With Sets", ACM PODS 1987, pp.11-20.

[10] Kuper,G.M., "On the Expressive Power of Logic Programming Languages with Sets", ACM PODS 1988, pp.11-14.

[11] Papadimitriou,C.H., Steiglitz,K., "Combinational Optimization", Prentice-Hall, 1982.

[12] Paredaens,J., "Possibilities and Limitations of Using Flat Operators in Nested Algebra Expressions", ACM PODS 1988, pp.29-38.

[13] Scholl,M.H. & Schek,H.J. (Eds.), "Proc. Int. Workshop on Theory and Applications of Nested Relations and Complex Objects", Darmstadt, 1987.