

自然言語表現上の操作による 関係データベース質問の作成

天野浩文 上林弥彦

九州大学工学部情報工学科

専門家でない利用者が形式的な言語を直接用いることなくデータベースを利用できるようにするための自然言語インタフェースの研究が盛んに行われている。しかし、自然言語処理には未解決の問題も多い。そこで、本稿では、あらかじめ人間（たとえばデータベース管理者）が用意しておいた自然言語表現の断片を用いるような手法を提案する。利用者は、あらかじめ用意されている文字列の挿入や、画面上の文字列の削除といった単純な操作によって、自分の意図するデータの意味を表す自然言語風の表現を作成する。システムは、これを選択・射影・結合質問に対応させることができる。また、画面上での編集操作に関し、限量修飾語の挿入も許すようにすれば、集合の比較演算を含むような複雑な質問でも、画面上で簡単に作成できるようになる。

本稿では、上記のような擬自然言語表現と関係代数における選択・射影・結合質問との対応について述べるとともに、限量修飾語を含む擬自然言語表現に対応する関係データベース質問とについて検討する。

Formulation of Relational Database Queries by Operations on Natural Language Expressions

Hirofumi AMANO and Yahiko KAMBAYASHI

Department of Computer Science and Communication Engineering, Faculty of Engineering, Kyushu University
6-10-1, Hakozaki, Higashi, Fukuoka, 812 Japan

Many researchers are investigating natural language interfaces, which enable naive users to use databases without seeing a formal language. Natural language processing, however, has many problems to be solved. Therefore, this paper proposes an approach for editing queries, which utilizes fragments of natural language expressions prepared by a person such as a database administrator. A user may express his/her intention as a natural-language-like sentence by means of simple operations such as the insertion of prepared strings or the deletion of displayed strings. The system can translate the sentence into an SPJ query. Moreover, even a complex query which includes set comparisons can be formulated easily on the screen, if the insertion of quantifiers is permitted as an editorial operation.

This paper discusses the correspondence between a pseudonatural language expression and an SPJ query in relational algebra. It also investigates database queries expressed by a pseudonatural language expression which contains quantifiers.

1. まえがき

データベースシステムの急速な普及により、専門家でない利用者がデータベースを利用する機会はますます増大する傾向にある。この傾向は今後も続いていくものと考えられ、このような利用者でも容易にデータベースを利用できるような環境を整えることが重要な研究課題となっている。

一方、データベースに蓄えられた情報を利用するためには、自分の求めたい情報がどのようなものであるかをシステムに知らせるための質問を利用者自身が記述して、システムに処理できる形で入力する必要がある。現在、商用のシステムで採用されている質問言語はSQL[芝野8803]などの形式的な言語がほとんどである。しかし、非専門家がデータベースを利用する場合、形式的な言語による質問作成が負担となることも多い。

そこで、この問題点を解決するために、これまで多くの研究者がおもに

①形式的な質問言語を使いやすくする

②自然言語を質問言語とする

の2つの方向の研究を行ってきた。

前者の方向の成果の代表的な例はQBE[ZLOO 7505]である。QBEは覚えやすく使いやすい言語として知られている。利用者が自然言語で与えられた指示に従ってQBEで質問を作成する場合、構文上の誤りはわずか6%の質問にしか見られないという報告がある。しかし、そのようにして作成された質問のうち27%は構文上の誤りを含まないが利用者の意図通りでないという結果も同時に報告されている[THOMG7505]。すなわち、自分が必要とするものとは異なる情報を利用者が受け取る可能性が約3割はともあることになる。

後者の方向では、利用者の作成した自然言語質問を形式言語質問に変換する自然言語インタフェースが盛んに研究されてきた。しかし、自然言語処理にともなう解決の困難な種々の問題点のため、自然言語の利点を十分に生かすことのできる自然言語インタフェースはまだ得られておらず、システムの複雑さや、システムの個別化の困難さ、といった問題が残っている。

そこで、われわれは、2つの方向を組み合わせた手法、すなわち、自然言語の利点を取り入れながらその欠点を回避するような質問作成支援システム(擬自然言語インタフェース)の基礎研究を行ってきた[上林8511][KAMB 8609]。

利用者は、データの意味を表す自然言語表現(あらかじめ管理者が作成しておいたもの)にQBEに似た入力を行って選択・射影・結合質問を作成する。一方、システムは、質問に含まれる関係代数演算に対応する変換を上記の自然言語表現に加えることによって、利用者が作成しつつある質問の意味をより自然な形(擬自然言語表現と呼ぶ)で生成する。利用者は自分の作成しようとしている質問の意味を常時確認しながら作業することができるので、QBEにみられるような欠点を回避するのに役立つ。このような質問検証の機能の重要性については、著者らとは独立にLowdenらの指摘もある[LOWDD8608]。

ここで、あらかじめ用意しておく表現群に適切な構文的制限を設けておくことにより、質問の意味に対応する表現を生成するために必要な変換操作を単純で機械的なものに行うことができる。これによって自然言語処理の持つ困難さを回避し、小規模な計算機上でも実用に堪えるシステムが実現可能

である。また、質問に対応して出力される表現は、(もとの表現の制約からあくまで“自然言語風”のものになるが)特別の訓練を受けていなくても読めば理解できるような表現となっており、自然言語の利点の1つである可読性を利用することができる。

しかし、上述の方法はSQL言語などの形式的言語そのものによる入力、あるいはQBEに似た入力を前提としており、形式的言語を利用者の目から完全に隠すことはできない。

そこで、本稿では、このような擬自然言語を用いる手法を応用し、利用者が画面上で自然言語風の文字列(システムが選択枝として提供する)を切り貼りあるいは挿入/削除することによって質問を作成する手法を考える。すなわち、擬自然言語表現に対する編集操作によって、その表現に対応する質問の内部表現を変更する方式について検討する。したがって、利用者は次のどちらかの方法で質問を作成できるようにする。

(a) 自分の質問の意味としてシステムから出力された擬自然言語表現に編集を加える。

(b) システムが提供する選択枝を用いて最初から質問を編集する。

この方式では、自然言語質問を直接処理するかわりに、利用者自身に擬自然言語表現を作成させてそれを形式的な質問に変換する。また、表現の編集の際にシステムから提供される選択枝となる文字列は、質問検証のための書き換えの場合と同様に、あらかじめ人間が用意しておいたもの、あるいは、いくつかの決まった語句でよい。このため、従来の自然言語インタフェースよりも処理が容易である。

さらに、擬自然言語表現に対しいくつかの限量修飾語を導入することにより、SQLの入れ子質問にみられるような複雑な集合比較演算を含む質問を擬自然言語表現上の操作で容易に作成できる。

2. 基本的事項

関係データベースは関係の集まりであり、各関係は直観的に1枚の表と考えることができる。関係の各列は1つの属性に対応し、関係Rの列集合に対応する属性集合を関係スキーマと呼び、 $\underline{R}$ で表す。関係の行(値の並び)は組と呼ばれる。 $r[X]$ を組rの属性集合Xに対する値だけを集めたものとするとき、関係RのXへの射影 $R[X]$ は次のように定義される:

$$R[X] = \{r[X] \mid r \in R\}.$$

θ をスカラー比較演算子とすると、関係Rの θ 選択は $A(\in \underline{R})$ に対し次のように定義される:

$$R[A \theta 'c'] = \{r \mid r \in R \wedge (r[A] \theta 'c')\}.$$

関係RとSの θ 結合をTとすると、Tは次のように定義される:

$$R[A \theta B]S = \{t \mid t[R] \in R \wedge t[S] \in S \\ \wedge (t[A] \theta t[B])\}.$$

ただし、 $A \in \underline{R}$, $B \in \underline{S}$ で、 $\underline{T}$ は $\underline{R}$ と $\underline{S}$ の直和である。

複数の属性による選択や結合も同様に定義できる。

超グラフは、節点集合Nと超枝集合Eの対(N, E)で定義される。枝は、節点の空でない部分集合で表され、各枝の節点の和集合はNである。

超グラフの節点x, yが、同じ枝に含まれるか、xともyとも連結な節点zが存在するとき、x, yは連結であるとい

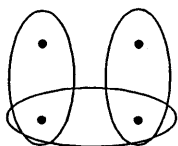


図1 Berge-非巡回超グラフ

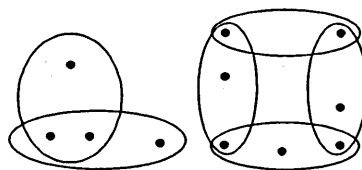


図2 Berge-巡回超グラフの例

う。すべての節点が連結であるとき、その超グラフは連結であるという。

超グラフにおける Berge-巡回 (Berge-cycle) は次の条件を満たす連鎖 $(S_1, x_1, S_2, x_2, \dots, S_m, x_m, S_{m+1})$ である。

- (1) $m \geq 2$
- (2) $x_1, \dots, x_m$ は相異なる節点
- (3) $S_i \neq S_{i+1} (1 \leq i \leq m)$, かつ, $S_1 = S_{m+1}$
- (4) x_i は S_i と S_{i+1} に含まれる $(1 \leq i \leq m)$

超グラフが Berge-巡回を持つとき、その超グラフは Berge-巡回 (Berge-cyclic) であるといい、そうでないとき、Berge-非巡回 (Berge-acyclic) であるという (図 1, 2)。

3. SPJ 質問とその意味を表す擬自然言語表現

本節では、SPJ 質問に対応する擬自然言語表現を得る方法の概要を述べる。ただし、最初に必要な関係をすべて結合し、その結果からある条件で選択し、必要な属性を射影する、という形式の質問だけを考える。結合の条件となった属性集合を J、選択の条件となった属性集合を S、射影される属性集合を P とする。

また、本稿では、次のようなデータベーススキーマを例として用いる。

```
SUPPLIER (S#, SNAME, STATUS)
PART      (P#, PNAME, COLOR)
SUPPLY    (S#, P#, QTY)
DEMAND    (PROJ#, P#, QTY)
```

これは、さまざまな部品とその供給者、およびそれを必要とするプロジェクトに関する情報を記録するためのデータベースである。

3.1 SPJ 質問から擬自然言語表現への変換

関係の属性の集合を、属性に対応する名詞句を含む自然言語の 1 文で記述できるような意味のあるまとまりに分け、これをオブジェクトと呼ぶ。これらの属性の名を陽に含む文章は、そのオブジェクト (属性集合) に対応する組の集合の意味を表すテンプレートであると考えることができる。

こうしてできる英文 (標準表現) は、出力文の材料となるもので、あらかじめ人間が作成しておく。また、質問によっては、これらから派生する副節 (関係詞節や前置詞句) も必要となるので、それも人間が与えておく。

たとえば、関係 SUPPLIER (S#, SNAME, STATUS) に対しては、次のような標準表現および副節を作成しておけばよい。

```
supplier of supplier-No. {S#}
: is called {SNAME}
: has a status {STATUS}
supplier called {SNAME}
```

```
: who has a supplier-No. {S#}
a status {STATUS}
: of supplier of supplier-No. {S#}
```

SPJ 質問の意味を表す英文は、上記の標準表現と副節に下記のような変換を施すことで得られる。

- ① 質問を表現するのに必要なオブジェクトの決定
 $\rightarrow$ SUP U J を被覆するオブジェクト集合を決定する。
- ② 結合に対応する変換
 $\rightarrow$ 土台の文に対し、結合される関係のオブジェクトに対応する表現を () 付きで埋め込む。
- ③ 選択に対応する変換
 $\rightarrow$ S に対応する語句の後に選択条件を示す修飾語句を埋め込む。
- ④ 射影に対応する変換
 $\rightarrow$ 出力属性集合 P を強調するため、表現中の P の要素の部分 [] でくくる。
- ⑤ 出力形式の最終的な調整
 $\rightarrow$ 自然な英文になるように、また、読みやすいように、書式を整える。

質問の意味を表すのに必要なオブジェクト集合を定めた後、結合条件にしたがって土台の文に副節を埋め込み、適切な部分に選択条件を表す表現を挿入して、射影属性を [] で強調すればよい。

先の SUPPLIER 以外に、部品の番号や名前を記録した PART と、供給者と部品の関連を表す SUPPLY があるとする。SUPPLY と SUPPLIER を S# で結合し、 $QTY \geq 100$ で選択して、{SNAME, P#, QTY} に射影する質問の意味を表す英文は、供給者番号と部品番号、数量を表す SUPPLY の文に、供給者名を表す副節 (標準表現から機械的に作れる) を埋め込むことで生成できる。

List [SNAME], [P#], [QTY] such that:

```
Supplier of supplier-No. {S#} (that is called
[SNAME]) supplies parts of part-No. [P#] in quantity
[QTY] (more than 100).
```

このようにして生成される擬自然言語表現は、標準表現と同様に、質問の解となる組の集合の意味を表すテンプレートとなっている。

段階④で求めたオブジェクト集合に対し、各属性を節点、各オブジェクトを超枝とする超グラフを考える。ただし、元の基底関係のスキーマにおいて 2 つのオブジェクトに共有される属性の部分、および、結合演算の条件に含まれる 2 つの属性のみを重ね合わせるものとする。同一名の属性であっても、結合に関与しないときには重ね合わせない。

上記の超グラフは、表現の骨格を表すと同時に質問の骨格 (関係の結合の構造) も間接的に表している。このような超

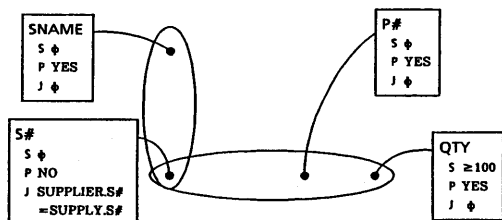


図3 オブジェクト超グラフの例

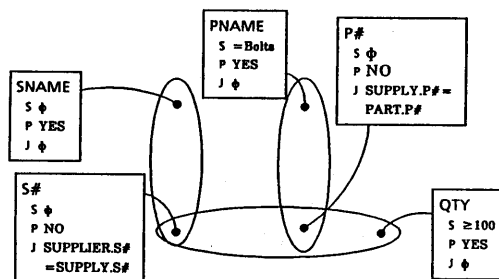


図4 変更を受けたオブジェクト超グラフ

グラフに、選択条件、結合条件、出力属性の情報を付加すれば、選択・射影・結合質問を表現できる。このような情報を付加した超グラフをオブジェクト超グラフと呼ぶことにする。上記の例に対するオブジェクト超グラフを図3に示す。

3.2 擬自然言語表現上でSPJ質問を編集する方法

利用者が、質問の意味と出力された文と自分の意図との相違に気付いた場合、何らかの形で質問を修正しなければならない。このとき、利用者の自由な入力を許すと、複雑な構文解析を行う必要が生じる。しかし、利用者がカーソルを適当な部分に合わせて、そこに挿入可能な副節の候補を表示させて選ぶ、あるいはその副節を削除する、という操作に限定しておけば、複雑な処理を必要としない。

そこで、利用者が画面上で出力文の任意の属性の部分にカーソルを合わせて、

- ① 選択の際の定数を書き換える
- ② 選択の際の比較条件を表す語句を書き換える
- ③ 選択の条件を新たに挿入する
- ④ 副節を新たに挿入する
- ⑤ 不要な副節を削除する
- ⑥ 不要な選択条件を削除する
- ⑦ 副節の前の比較を表す語句を書き換える
- ⑧ 属性名の部分を[]でくくる
- ⑨ []を解除する

の7種類の操作を組み合わせて行い、システムがそれを元の形式的な質問に反映させることを考える。

先に述べたように、出力文はオブジェクト超グラフで表せる。このオブジェクト超グラフから対応する質問を求めることは容易であり、上記の操作が表現に加えられたとき、対応するオブジェクト超グラフ上の操作を求めることができる。

①, ②, ③, ⑥, ⑦, ⑧, ⑨の操作は、対応するオブジェクト超グラフの超グラフとしての形状が変化せず、選択条件の定数あるいは比較演算子、結合条件の比較演算子が変化するだけの操作であるから、元のオブジェクト超グラフの上で、変更箇所に対応する部分を変更するだけでよい。

④, ⑤の操作は、質問に対応するオブジェクト超グラフの形状が変化する場合である。挿入された副節に対応する超枝を加えるか、削除された副節に対応する超枝を除去すればよい。

たとえば、前出の例について、利用者が、{P#}の後にPNAMEに関する副節を挿入し、PNAMEを定数'Bolts'で置換した場合の表現は、

Supplier of supplier-No. {S#} (that is called {SNAME})
supplies parts of part-No. {P#} (that is called

'Bolts') in quantity [QTY] (more than 100).

となる。対応する超グラフは、図4のようになる。これにより、質問はSUPPLIERとSUPPLYの結合とQTY ≥ 100による選択の他にPNAME='Bolts'による選択、および、P#によるPARTとの結合を行い、{SNAME, QTY}を求めるものとなっていることがわかる。

4. 限量修飾語を含む擬自然言語表現

簡単なSPJ質問の作成については、前節で述べた手法が有効である。SPJ質問は、関係データベース言語の標準となりつつあるSQLにおいては、単一のSELECT-FROM-WHERE節からなる質問(質問ブロック)にほぼ対応する[BRAD 82][LUK18612][芝野8803]。しかし、SQLの特色の1つに、質問ブロックの中に質問ブロックが埋め込まれた入れ子質問がある。入れ子質問は、関係除算などの集合演算を表すのによく用いられ、通常のデータベース利用でも無視することのできないクラスである。

SQLの入れ子質問を擬自然言語表現に変換する方法については、各レベルの質問ブロックを変換した後、それらの間の集合比較演算を表す文を挿入する単純な手法をすでに提案した[天野8603]。しかし、この方法で質問に対応する擬自然言語表現を生成しようとする、ブロック間参照がある場合、自然でわかりやすい表現を得るのが困難になることがある。これは、ブロック間参照のある入れ子質問を自然言語で表そうとすると、複数形や限量修飾語を用いなければ不自然になることが多いためである。

形式的言語による質問を自然言語で言い換える場合、いくつかの入れ子SQL質問には単純でわかりやすい表現が存在すると考えてもよさそうである。しかし、任意の入れ子質問にわかりやすい自然言語表現が対応するとは限らない。むしろ、数学的に記述できるというだけで、日常的な言葉で表しにくい質問が存在すると思うほうが自然である。

そこで、以下の議論では、任意の入れ子質問を扱うという目標から離れ、逆の方向、すなわち、扱いやすい形式の入れ子質問を擬自然言語表現で作成することを考える。

擬自然言語表現はQBEとよく似た性質を持っているが、"all"や他の集合に関する記述を含むQBE質問をSQL質問に変換するMcLeodのアルゴリズム[MCLE 7610]はここでは採用しない。このアルゴリズムは限量修飾語として"all"しか許しておらず、自然言語のさまざまな限量修飾語を導入するのに不十分だからである。

4.1 限量修飾語の導入

SPJ質問を記述するのに用いた擬自然言語に対し、限量

修飾語 ("all", "no", "only" など) を新たに導入し、たとえば、利用者が画面上で次のような表現を作成し、対応する S Q L 質問を得られるようにする。

Supplier of supplier-No. [S#] (who is called [SNAME]) supplies all parts {P#} (that are called "Bolts").

この擬自然言語表現は、前節までの方法で骨格となる表現を作成したのち、属性名 P# の前に限量修飾語 "all" を挿入することによって作成できる。ただし、属性に対応する名詞が単数形有的时候には、その複数形があらかじめ用意されているものとし、"all" の挿入後、それをを用いてシステムが書き直すものとする。

これに対し、システムは対応する入れ子 S Q L 質問を作成できなければならない。このため、本稿では、いくつかの限量修飾語について、それを含む擬自然言語表現が一般にどのような類型の入れ子質問に対応するかについて検討する。

この手法は、入れ子質問について自然な表現を生成できないという問題点[天野8603]の根本的な解決とはならないが、利用者が簡単な操作(限量修飾語の挿入)によって、ある類型の入れ子 S Q L 質問を作成できるという点で有意義である。

ただし、利用者が扱う表現がわかりやすく、かつ、確認のためにシステムが書き換えて出力する表現が簡単に生成できるよう、限量修飾語が埋め込まれる擬自然言語表現のオブジェクト超グラフは Berge-非巡回であるものとする。

4.2 限量修飾語を含む擬自然言語表現の解釈

[WODD 78], [HOBBS8701] によれば、限量修飾語を含む表現を厳密に解釈すると、その結果は次のような形に書き直せる。

For <quant> X such that (…X の定義域を指定する条件…).

(…そのような X が満たす制約…).

ただし、<quant> は限量修飾語を表す。以下、上記の形式を限量標準形と呼ぶことにする。

先に述べた限量修飾付き擬自然言語表現の S Q L による表現を求めるためには、まず、それを上記の限量標準形に当てはめる必要がある。しかし、限量修飾付き擬自然言語表現は単に限量修飾語が埋め込まれているだけであるから、定義域指定条件部と制約条件部との区別が明確でなく、いく通りかの解釈が成立しうる。このため、すべての可能な解釈をなんらかの方法で抽出し、それらを明確に区別できる形で書き換えて利用者に選択させる必要がある。

本稿ではデータベースを扱っているので、限量修飾語の付

いた属性(限量標準形では X に相当する)の値の集合はデータベースから取り出せるものでなくては意味がない。すなわち、定義域指定条件は、なんらかの関係データベース質問で書けるものでなければ意味がない。また、連結でない超グラフで定義域指定条件を記述しても、被限量属性を含む成分だけで考えても、結果は同じである。また、システムが表示する表現の自動生成が容易であるためには、定義域指定条件と制約条件部はオブジェクト超グラフであることが望ましい。したがって、定義域指定条件と制約条件部に関し、次の仮定をおく。

仮定1: 定義域指定条件は

- (1) 被限量属性のみであるか、または、
- (2) 連結なオブジェクト超グラフで表される。

仮定2: 制約条件は連結なオブジェクト超グラフで表される。

仮定3: 定義域指定条件部と制約指定条件部は被限量属性のみを共有する。

これらの仮定より、前述の例に対するオブジェクト超グラフの上では次の3通りの解釈が可能であることがわかる。

① For all parts,

supplier of supplier-No. [S#] supplies those parts, and they are called "Bolts".

「すべての部品を供給しており、しかも、それらの部品がすべて "Bolts" という名前であるような供給者」

② For all parts (that are called "Bolts"),
supplier of supplier-No. [S#] supplies those parts.

「"Bolts" という名前の部品をすべて供給している供給者」

③ For all parts (that are supplied by supplier of supplier-No. [S#]),
those parts are called "Bolts".

「供給している部品の名前がすべて "Bolts" であるような供給者」

ただし、①のような解釈はあまり自然と思われる解釈ではない。また、③のような解釈はもとの限量修飾付き擬自然言語表現の構文から許される解釈ではない。しかし、逆に、利用者が仮にそのような質問を作成したいと考えたときに、この手法でそれが作成できるようにするため、どちらも1つの解釈として採用することにする。

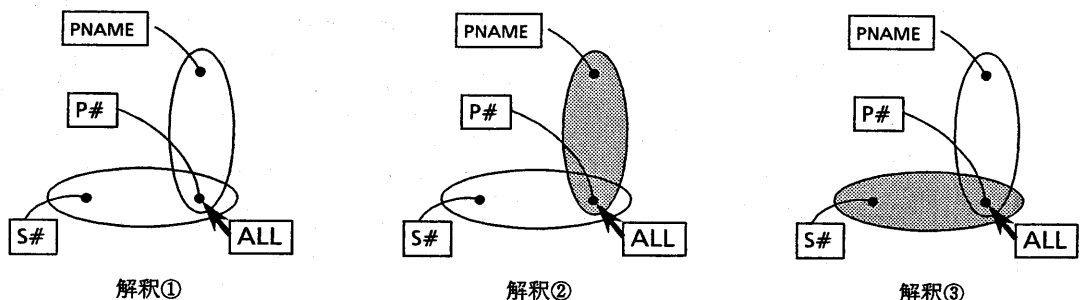


図5 定義域指定条件部の可能な範囲(各属性の付加的な情報は省略した)

これらの解釈に対応する定義域指定条件部がオブジェクト超グラフのどの部分に相当するかを図5に示す。

仮定より、一般に擬自然言語表現に限量修飾語が挿入されたとき、そのオブジェクト超グラフを定義域指定条件部と制約条件部とに分けるような分割のすべてを求める方法は、次のような簡単なものである。

- ① 限量付き擬自然言語表現のオブジェクト超グラフを、被限量属性のところで分割する。
- ② これらの連結成分（被限量属性のみを共有する）を2つのグループに分けるような分割を求める。ただし、すべての超枝が制約条件部に含まれる分割に対しては被限量属性を定義域指定条件部とみなす。また、すべての超枝が定義域指定条件部に含まれるような分割は除く。

さらに、これらを表す擬自然言語表現は、SPJ質問の意味を表すときと同様に、次のような形式で自動生成できる。

For <quant> {属性名} (…定義域指定条件部の擬自然言語表現…),
(…制約条件部の擬自然言語表現…).

ただし、制約条件部では <quant> のかわりに “those” を用いることにする。

4.3 限量修飾付き擬自然言語表現とSQL質問

4.2節の方法により、限量修飾付き擬自然言語表現の解釈が決定できたら、次にこれをSQL質問に変換する必要がある。ここで、定義域指定条件部と制約条件部とがもとの超グラフを二分するような形に定めたので、これらの解釈は、定義域指定条件部から定められるような被限量属性値の集合と、制約条件部から定められるような被限量属性値の集合との間の比較演算などによって表せることが多い。

定義域指定条件による被限量属性値の集合をA、制約条件による被限量属性値の集合をBとすると、いくつかの限量修飾語に対して次のような集合演算を用いることができる。

- (1) “all” → $A \subseteq B$
- (2) “only” → $A \supseteq B$
- (3) “no” → $(\langle \text{被限量属性} \rangle \notin A) \vee (\langle \text{被限量属性} \rangle \in B)$

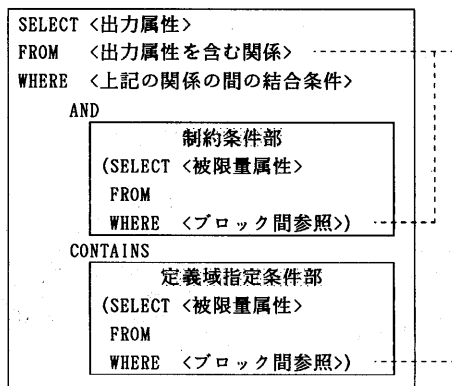


図6 “all”に対するSQL質問テンプレート

(4) “at least three” → $\text{COUNT}(A \wedge B) \geq 3$

したがって、限量修飾語の種類によって、定義域指定条件部と制約条件部が与えられたときの、対応するSQL質問のテンプレートを定めておけばよい。

たとえば、“all”の場合、「制約条件で束縛したときの被限量属性値の集合が、定義域指定条件で表される値の集合を含む」と考えることができるから、対応するSQL質問のテンプレートは図6のようになる。図のテンプレートの内側の質問ブロックがそれぞれの条件によって定められる値の集合を陽に求めている。出力属性を含む関係は外側の質問ブロックに置かれる。このとき、これらの関係を内側のブロックから参照することで、下位のブロックのFROM節から省略できる。

したがって、利用者が先の例のような限量修飾付き擬自然言語表現を作成し、システムから表示された選択枝のうち②を選択した場合、次のようなSQL質問が自動的に作成できる。

```
SELECT S#
FROM SUPPLY X
WHERE (SELECT P#
FROM SUPPLY
WHERE S# = X.S#)
CONTAINS
(SELECT P#
FROM PART
WHERE PNAME = "Bolts")
```

述語“CONTAINS”は現行のSQLではサポートされていないが、文献[LUKKL8612]には、“CONTAINS”を含む質問の“EXIST”による等価な表現が示されている。

“all”以外の限量修飾語に対するSQL質問テンプレートについては、付録を参照されたい。

5. 複数の限量修飾語を含む擬自然言語表現

4節で検討した限量修飾付き擬自然言語表現は、限量修飾語を1つだけ含むものである。しかし、擬自然言語表現において、限量修飾語が2つ以上現れる場合も検討する必要がある。たとえば、利用者が次のような表現を作成したいと思うのはそれほど特殊な状況ではない。

Supplier of supplier-No. [S#] (that has a status [STATUS]) supplies all parts (which are required for all projects).

ここで、[WOOD 78]によれば、1つの表現の中に限量修飾語が2個以上現れるときは、先の限量標準形の定義域指定条件部もしくは制約条件部の中に別の限量標準形が埋め込まれた形で考える必要がある。単一の限量修飾語の場合と同様に、上のような表現では、それらの限量修飾語の定義域指定条件部/制約条件部の区別は必ずしも明確ではない。

4節で述べたのと同様の理由から、上位の定義域指定条件部/制約条件部に含まれる限量標準形の定義域指定条件部/制約条件部も仮定1～3を満たすものと仮定する。すると、上記の表現に対しては次の3通りの解釈が成立しうる。

- ① For all parts such that:

For all projects,

those parts are required for those projects,
supplier of supplier-No. [S#] (that has a status [STATUS]) supplies these parts.

- ②For all parts,
For all projects,
supplier of supplier-No. [S#] (that has a status
[STATUS]) supplies these parts (that are
required for these projects).
- ③For all parts (that are supplied by supplier of
supplier-No. [S#] (that has a status [STATUS])),
For all projects,
these parts are required for these projects.

単一の限量修飾語の場合と同様に、②や③はもとの表現から自然に連想される解釈ではない。しかし、1つの選択枝としてそれぞれ採用する。

次に、これらに対応するSQL質問を求める必要がある。ここで、先に求めた“all”に対するテンプレートは、定義域指定条件によって定まる被限量属性値の集合を求める部分と、制約条件によって定まる被限量属性値の集合を求める部分とを、内側の2つの質問ブロックの形で陽に含んでいる。

上記の3つの解釈は、定義域指定条件部もしくは制約条件部のいずれか一方で入れ子になっているので、前述の質問テンプレートを用いて、入れ子になっているほうの部分にさらに質問ブロックを埋め込んでやればよい。たとえば、解釈①ではpartsの定義域指定条件部のところに限量標準形が埋め込まれているので、対応するSQL質問は内側の2つの質問ブロックのうち、下のほうで入れ子になる。

```
SELECT S#, STATUS
FROM SUPPLIER X
WHERE (SELECT P#
FROM SUPPLY
WHERE S# = X.S#)
CONTAINS
(SELECT P#
FROM DEMAND Y
WHERE (SELECT PROJ#
FROM DEMAND
WHERE P# = Y.P#)
CONTAINS
(SELECT PROJ#
FROM DEMAND))
```

このように、限量修飾語に対するSQL質問テンプレートが、定義域指定条件と制約条件とで定められる値の集合をそれぞれ陽に求める質問ブロックを含んでいれば、それぞれの条件の部分で限量標準形が入れ子になっていても、容易に対応するSQL質問を求めることができる。

6. むすび

本稿では、利用者が形式的な言語を用いることなくデータベースを利用できるようにするための一方法として、擬自然言語表現上の単純な操作で関係データベース質問を作成する手法について述べた。自己結合を含まないSPJ質問を作成する場合には、擬自然言語表現に対応する超グラフから、対応する質問を容易に求めることができる。本方式では自然言語による入力を直接に処理するわけではないが、利用者が自分の作成しつつある質問の意味を理解しながら作業できるという特色がある。

さらに、このような擬自然言語表現に限量修飾語を導入す

ると、集合の比較演算を含むような複雑な質問のいくつかが表現できるようになる。ただし、限量修飾付きの表現に対しては、限量修飾語の束縛範囲を考慮する必要がある。本稿では、限量修飾語が挿入された擬自然言語表現の解釈が明確でない場合に、可能な解釈のそれぞれを表す表現の生成法を求めた。また、いくつかの限量修飾語について、定義域指定条件と制約条件とが明らかになった場合、その擬自然言語表現に対応するSQL質問の形式を与えた。新しい限量修飾語を導入する場合には、それに対応する質問テンプレートを作成するだけでよい。この性質は種々のアプリケーションに対処できる利用者インタフェースにとり非常に重要である。

本稿で例示した質問テンプレートはどれも定義域指定条件部と制約条件部によって定められる被限量属性値の集合を、SQLの質問ブロックによってそれぞれ陽に求めている。したがって、複数の限量修飾語が用いられる場合でも、質問テンプレートを入れ子にするだけで対処できる。その他の限量修飾語については、対応するテンプレートが必ずしもこのような性質を持つとは限らない。しかし、質問テンプレートのそのような性質が必要になるのは、そのテンプレートの中に他のテンプレートが入れ子になる場合だけである。したがって、そのような限量修飾語が入れ子の最下位に位置する場合には問題がない。

謝辞

日頃より熱心に御助言・御討論くださる本学情報工学科今井浩助教授、古川哲也助手、ならびに、御協力くださる大学院生 Mohamed El-Sharkawi氏をはじめとする研究室の諸氏に深謝いたします。

なお、本研究は昭和63年度文部省科学研究費補助金特定研究「言語情報処理の高度化」のうちの「情報ドキュメンテーションのための言語の研究」の援助を受けている。

参考文献

- [BRAD 82]
Bradley, J.: File and Data Base Techniques, Holt, Reinhart & Wilston, 1982.
- [HOBBS8701]
Hobbs, J. R., Shieber, S. M.: An Algorithm for Generating Quantifier Scopings, *Comput. Linguist.*, Vol.13, No.1-2, pp.47-63, January 1987.
- [KAMB 8609]
Kambayashi, Y.: An Overview of Natural Language-Assisted Database UserInterface: ENLI, *Proc. IFIP 10th World Computer Congress*, pp.1055-1060, September 1986.
- [LOWDD8608]
Lowden, B. G. T., De Roeck, A. N.: The REMIT System for Paraohrasing Relational Query Expressions into Natural Language, *Proc. 12th Int. Conf. on Very Large Data Bases*, pp.365-371, August 1986.
- [LUKKL8612]
Luk, W. S., Kloster, S.: ELFS: English Language from SQL, *ACM Trans. Database Syst.*, Vol.11, No.4, pp.447-472, December 1986.

[MCLE 7610]

McLeod, D.: The Translation and Compatibility of SEQUEL and Query by Example, Proc. 2nd Int. Conf. on Software Engineering, pp.520-526, 1976.

[THOMG7505]

Thomas, J. C., Gould, J. D.: A Psychological Study of Query by Example, Proc. AFIPS National Computer Conf., pp.439-445, May 1975.

[WOOD 78]

Woods, W. A.: Semantics and Quantification in Natural Language Question Answering, Advances in Computers, Vol.17, pp.1-87, Academic Press, 1978.

[ZLOO 7505]

Zloof, M. M.: Query by Example, Proc. AFIPS National Computer Conf., pp.19-22, May 1975.

[天野8603]

天野, 上林: S Q L質問に対応する自然言語表現の生成, 情報処理学会第33回(昭和61年後期)全国大会, pp.837-838, 1986年3月.

[上林8511]

上林, 天野: 自然言語表現に対するデータベース操作, 情報処理学会自然言語処理研究会資料(85-NL-52), 1985年11月.

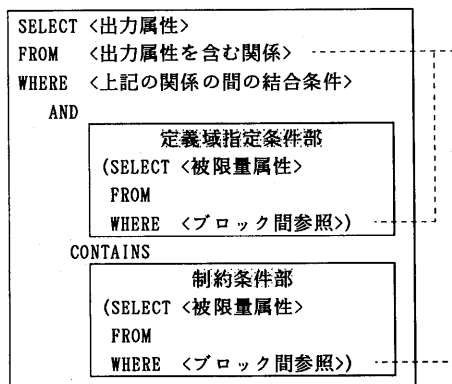
[芝野8803]

芝野: データベース言語 S Q L, 情報処理, Vol. 29, No. 3, pp.208-214, 1988年3月.

付録

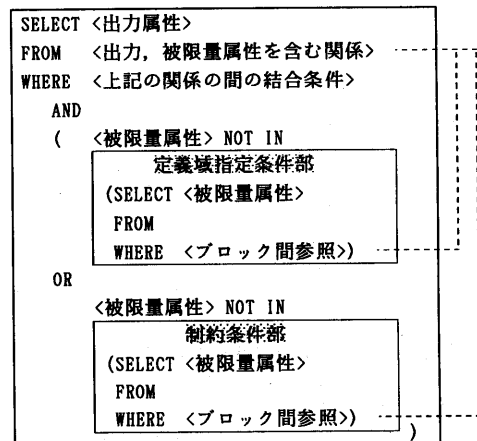
(1)"only"に対するテンプレート

定義域指定条件を満たす被限量属性値の集合の要素だけについて制約条件が満足されるから, 集合の包含関係が"all"の場合と反対になる.



(2)"no"に対するテンプレート

定義域指定条件を満たす値の集合のどれに対しても制約条件が成立しないから, 被限量属性の値が定義域に含まれないか, あるいは, 制約条件を満たさないかのどちらかである. ここでは, 2つの集合を陽に記述するため, 述語"IN"を用いている.



(3)"at least three"に対するテンプレート

定義域に含まれる値の集合の少なくとも3つの要素について制約条件が成立するという事は, 両方の条件を満たす値の集合が3個以上の要素を持つということである. 集合の要素数は集約関数"COUNT"で求められる.

