

化学構造データベースにおけるデータの表現とアクセスについて

樂玉琴 山田潤二 霜山友肖 黒沢和弘 信川茂久 大保信夫 北川博之 山口和紀 藤原謙
筑波大学

化合物情報は様々な分野において重要である。化学構造は本来3次元の情報であるが、それを2次的に表したものを化学グラフと呼んでいる。化学構造データベースを構築する際には、これらの3次元ないし2次元のデータを情報損失の少ないように記述し、またコンピュータで処理しやすく、そして、アプリケーションの要求が満たせるように表現する必要がある。本研究では、部分構造検索及びグラフの同型性判定処理の効率化などを考慮して、非定型的な化学構造の記述、表現について検討した。本アプローチでは、関係型データベースを拡張して、抽象データ型を扱うデータモデルを定式化し、化学グラフや化学式などのアプリケーションに特有なデータの表現を可能にした。また、この関係型データベースの上に関数型外部ビューを構築し、それによって関係型データベースにおける内部表現とは独立に、実世界のデータ構造に基づいた化学構造データの操作を可能にする方法を実現した。

DATA REPRESENTATION AND ACCESS
TO CHEMICAL STRUCTURE DATABASES

Yu Qin Luan Junji Yamada Tomoaki Simoyama Kazuhiro Kurosawa Sige-hisa Nobukawa
Nobuo Ohbo Hiroyuki Kitagawa Kazunori Yamaguchi Yuzuru Fujiwara
University of Tsukuba
Tennohdai 1-1-1, Tsukuba, Ibaraki, 305, Japan

Information about chemical compounds is very important in various fields. It is 3-dimensional in nature, and chemical graphs are of its 2-dimensional representation. In order to develop chemical structure databases, these 3-dimensional or 2-dimensional information has to be described and represented in 1-dimensional form. The representation scheme is very important because it determines whether the application specific operations can be supported effectively or not. In this study, chemical structures are represented in the extended relational data model which incorporates abstract data types to expand capabilities to accommodate nonnormal forms of chemical data. Functional view on top of the relational database is added to enable users to access database based on chemical view over the real world data structure without considering the underlying relational schema.

1 はじめに

化学構造情報は化合物の性質を決定する上で本質的な役割を果たして、化学、生物、薬学、農学、材料などさまざまな分野で必要なものである。現在既に8百万種を超える化合物が存在していて、また毎年30万種以上のスピードで増加している。このような莫大な数のデータを管理するために、化学構造データベースに対する要請が非常に高い。

化学構造データベースを構築する際に先ず問題になるのは、化学構造の表現である。つまり、本来3次元の構造をできるだけ情報損失の少ないように記述し、そして、記述された情報をできるだけコンピュータで処理しやすく、同時にアプリケーションの要求にも応じることができるよう表現しなければならない。これまで主として、線形表記法とトポロジカル表記法の2種類の表記法が提案されている。線形表記法は化学構造と表記の間に一对一の関係を目標としているため、特定の化合物だけを見つかる全構造検索に向いているが、ある部分構造が与えられ、それを含むすべての化合物を見つけ出す部分構造検索に対しては、結合表に代表されるトポロジカルな表記法が適している。

現在、CAS ONLINEという世界的に広く使われている化学構造データベースシステムがある。このシステムでは、部分構造検索を効率よく行なうために、比較的良好に出現する部分構造(スクリーンと呼んでいる)を予め抽出しておいて、化学構造に含まれるスクリーンに基づいて、それを2100ビット余りのビット列で表している。この方法は専門家によって選び出したスクリーンと一般ユーザーの考える部分構造とは必ずしも一致せず、またどのような部分構造が含まれているかが記述されているが、部分構造間の結合関係が十分に記述されていないため、検索漏れやノイズの問題が存在する。

一方、すべてのトポロジカルな構造情報を保持しながら、グラフ理論を応用して、部分構造を抽出しようとする試みが著者らによって行なわれ、BCT(Block Cutpoint Tree)分割法と呼ばれている[1]。BCTはグラフ理論におけるブロックと切断点をノードとする木構造で、化学グラフの大まかな形状を表している。この方法によって化学構造はブロック単位に分割され、各ブロックは結合表で表され、またブロック間の結合関係はBCTによって表される。この方法は深さ優先によるブロック探索アルゴリズムが存在しているため、機械的に分割しやすく、またグラフ

の環状部分がひとまとめに分割されるので、その部分を見つけ出すのに強力である。しかし反面、グラフの鎖状部分が全て頂点数が2のブロックにバラバラに分割されるため、その部分の特徴を利用した検索の処理効率が悪くなる。

CASの方法とBCT法とでは、部分構造に対する捉え方が異なっているが、いずれにしても化学構造に含まれる部分構造という限られた属性しか記述していない。化合物の属性としては他にもたくさんある。例えば化合物の中に含まれる"-OH"や"NH3"などの官能基の組み合わせの仕方も化合物の性質を決定する上で重要な役割を果たしている。これらが単独にも検索のキーとして使え、また構造によってはその長距離の相互関係までも記述する必要がある。BCT法ではすべてのトポロジカルな結合情報が表現されていて、そこからさまざまな属性が抽出され得るが、効率の問題がある。

関係型データモデルが実世界の対象に付随するさまざまな属性を記述するのに限界があるが、わかりやすさ、汎用性、データ独立性、共有性、整合性、障害回復などさまざまな機能を提供していて、従来の事務処理において、非常に大きな成果を収めてきて、多量なデータに対して、安定したデータ管理機能が提供できる。従って、関係型モデルは化学構造データベースを構築するのに一つの基本になり得るものである。

本研究はこのような背景と方針のもとで、化学構造データベースシステムを構築する際のデータの記述、表現及びアクセスなどの問題について検討し、それに基づいて、関係型DBMSを拡張したシステムの設計、開発を行った。

2 化学構造データの表現

化学構造は本来3次元の情報であるが、トポロジカルな結合関係のみ注目し、それを2次元的に表わしたものを化学グラフと呼んでいる。化学構造に関してはノードが確定原子とは限らないこと及び結合距離や結合角などの立体構造情報や互変異性体構造、イオン結合、芳香族などの結合情報の表現問題もあるが、第一にトポロジカルな構造情報を中心に総称表現までを含めた表現の問題を考える。

2次元の化学グラフをコンピュータで処理できるようにするには、それを何らかの形で一次的に表現する必要がある。結合表は化学構造を構成している水素以外のす

すべての構成原子について、番号、原子の種類、その原子と隣接している結合相手の原子番号、結合の種類などの情報を記述したものである。本研究では、結合相手情報の重複を除いて、自分自身より番号の大きい結合相手のみを記述した結合表を用いる。

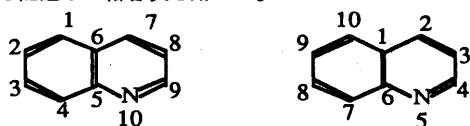


図1(a) 化学グラフ

((1 C ((2 2)(6 1)))	((1 C ((2 1)(10 1)))
(2 C ((3 1)))	(2 C ((3 2)))
(3 C ((4 2)))	(3 C ((4 1)))
(4 C ((5 1)))	(4 C ((5 2)))
(5 C ((6 2)(10 1)))	(5 N ((6 1)))
(6 C ((7 1)))	(6 C ((7 1)))
(7 C ((8 2)))	(7 C ((8 2)))
(8 C ((9 1)))	(8 C ((9 1)))
(9 C ((10 2)))	(9 C ((10 2)))
(10 N (()))	(10 C (()))

図1(b) 結合表

コンピュータによるグラフデータの処理において、グラフとグラフとの同型性判定が処理コストが大きい。例えば図1(a)の2つのグラフが同型であるかどうかは、図1(b)の2つの表を使って、番号付け換えに対応する組み合わせ量の計算をしないと判定できない。従って、頂点数の多いグラフに対して、結合表による単純な同型性判定は、長時間を要し、事実上不可能となる場合もある。従って、処理時間が短縮できるような表現法が要求される。また、表現の問題は、アプリケーション側からのアクセス要求と合わせて考える必要がある。化学構造データベースに対する最も汎用性の高い利用法は部分構造検索である。それを効率よく行うことを可能にすることが表現法に要求される。

上の2つの要求に対して、化学グラフを適切な部分グラフに分割して表現する方法が有効と考える。

2.1 化学グラフの分割

ここで述べる化学グラフの分割法は、基本的にBCT法を拡張したものである。BCT法はグラフ理論を応用して化学グラフをブロックに分割する方法である。

ブロックとは、どの頂点を除いても2つ以上のグラフに分割されない最大の部分グラフである。二重連結成分とも呼ばれている。本研究ではBCT分割法の欠点をなくすため、鎖状部分をひとまとめに処理できる分割単位を導入し、スーパーブロック分割法と呼ぶ分割法を提案する。

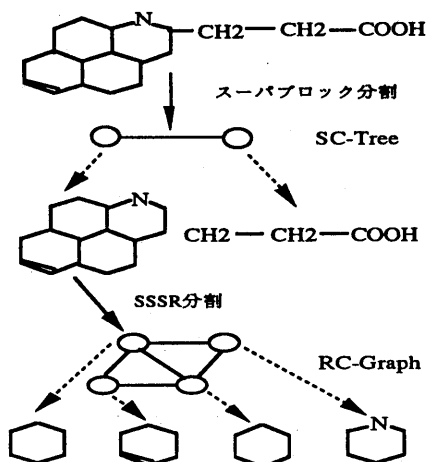


図2 スーパーブロック分割とSSSR分割

スーパーブロック分割法とは、図2に示したように化学グラフを環状スーパーブロック(頂点数が3以上のブロック及びスピロ環など)と、鎖状スーパーブロック(もとのグラフから環状スーパーブロックに含まれる頂点を全て除いた後に残る連結成分)に分割する方法である。

環状スーパーブロックについては、更にSSSR (Smallest Set of Smallest Ring)法によって、それを構成している最小リングの集合に分割する。これによって、最小リング単位での部分構造検索も効率よくできる。図2において、SC-Tree (Super block Connection Tree) はスーパーブロック間の結合関係を表したもので、RC-Graph (Ring-Connection-Graph) は最小リング間の結合関係を表したものである。

上記の分割を図3に示されたDAG(Directed-Acyclic-Graph)で簡潔に、しかも情報の損失なしに表現できる。

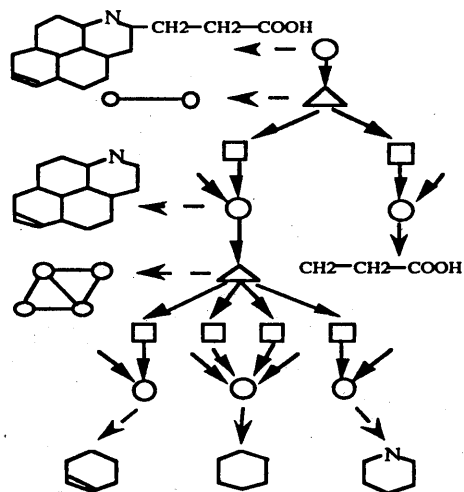


図3 分割表現モデル

2.2 化学構造データベースの関係型スキーマ

化学グラフを分割することによって、スーパーブロック及び化学的意味の大きい最小リングを用いた部分構造検索が効率よく行え、また同型性判定の対象となるグラフの頂点数も減り、処理時間の短縮が期待できる。同型性判定に関しては更に、頂点の数や化学式などの属性を用いて適合する化合物を先に絞っておくことによって、判定の対象となるグラフの数を減らすことが有効である。図3の分割表現モデル及びこのような考えに基づいて、化学構造データを次のスキーマで関係型データベースに格納する。

- R1: COMPOUND (C#, name, vertices, edges, rings, formula, chem-graph, func-group)
 R2: SUBSTRUCTURE (S#, vertices, edges, rings, formula, chem-graph, type, func-group)
 R3: CONNECTION (CG#, nodes, edges, graph, type, edge-maps)

R4: MAPPING (M#, vertex-maps)

R5: COMP-CONNEX (C#, CG#)

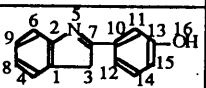
R6: SUB-CONNEX (S#, CG#)

R7: CONNEX-MAP (CG#, M#)

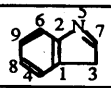
R8: MAP-SUB (M#, S#)

R1、R2、R3、R4は図3のDAGにおけるノードに対応していて、それぞれ化学グラフ、部分グラフ、部分グラフ間の結合、部分グラフと化学グラフ間の頂点の対応を表している。R5からR8までは図3のDAGにおけるアークに対応している。なお異なる化学構造における同一部分構造は共有される形となっている。大文字で書かれた識別子と化合物の名前以外の属性はすべて化学構造の結合表(chem-graph属性)に基づいて計算された特徴量で、適合する化合物を絞り出すのに使われる。Func-group属性は化学構造あるいは部分構造に含まれる官能基のリストを一定の順序に従って2進数で表したものである。下線のついた属性は抽象データ型(後述)として定義されている。図4に化学構造データベースの一インスタンスを示す。

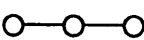
R1: COMPOUND

C#	NAME	VERTICES	EDGES	RINGS	FORMULA	CHEM-GRAPH	FUNC-GROUP
C1	16	18	3	C ₁₄ H ₁₁ ON		110
.....

R2: SUBSTRUCTURES

S#	VERTICES	EDGES	RINGS	FORMULA	CHEM-GRAPH	TYPE	FUNC-GROUP
S1	9	10	2	C ₈ H ₆ N		CYCLIC-SB	10
.....

R3: CONNECTION

CG#	NODES	EDGES	GRAPH	TYPE	EDGE-MAPS
CG1	3	2		SC-TREE	(((S1 . 7)(S2 . 6))(S2 . 2)(S3 . 1)))
.....

R4: MAPPING

M#	VERTEX-MAPS
M1	((C1 . 7)(S1 . 7))
M2	(((C1 . 10)(S2 . 6)) ((C1 . 13)(S2 . 2)))
.....

R5: COMP-CONNEX

C#	CG#
C1	CG1
.....

R7: CONNEX-MAP

CG#	M#
CG1	M1
.....

R6: SUB-CONNEX

S#	CG#
S1	CG2
.....

R8: MAP-SUB

M#	S#
M1	S1
.....

図4 化学構造データベースインスタンス

2.3 抽象データ型(ADT)の導入

関係型データベースにおいて、属性値のデータタイプとしては整数、実数、文字列、そして時間や日付など限られたタイプしか許されていない。従来の事務処理において、属性値をそのまま出したり、あるいはその平均値や最大値などを求めたりする集合操作をサポートするだけで良かったが、化学構造データベースの場合、2.2節のスキーマで下線のついた属性に対しては、化学のアプリケーションに特有な操作を施したいことがある。また、関係型データベースでは、レコードの長さなどに物理的な制限があり、レコードで表現できる属性が限られてしまい、化合物のすべての属性が表現しきれない。

これらの問題も含め、化学グラフや化学式などのデータ、化学グラフにおいて極めて重要な特許情報や反応形式の記述によく用いられる総称表現及び特殊結合様式などをADTとして表現することが有効である。ADTは、データとその上の操作をひとまとめにカプセル化したもので、必要であれば、いつでも操作関数を加えることができるなどの利点もっている。従って、ADTを用いて化学グラフなどの新しいタイプのデータを表わすことによって、アプリケーションに特有な操作が行える。また、ユーザはアプリケーションの要求に応じて、必要な関数を加えていけば、求めたい属性値がデータベースの中に格納されていなくても、あたかもそれが存在するかのように操作できる。次にADT定義の一例を示す。

```
def-ADT(  
  name = FUNC-GROUP,  
  include-COOH  
  FUNC-GROUP → boolean,  
  what-func-groups  
  FUNC-GROUP  
  → set-of CHEMICAL-FORMULA,  
  .....  
  filename = "/usr1/ran/adt/FUNC-GROUP")
```

ここで、"include-COOH"はある特定の化学構造あるいは部分構造の中に"-COOH"という官能基が含まれているかどうかをチェックする関数で、"what-func-groups"は含まれている官能基の種類をそれらの化学式の集合の形で返す関数である。"Filename"はこれらのADT関数の定義本体を格納するファイルの名前である。

2.4 ADT-QUELによる問い合わせ

関係型データベースにおいて、非手続き的な問い合わ

せ言語によるアクセスがサポートされている。ADTを導入した上記関係型データベースに対しては、ADT関数を使つての問い合わせをサポートしなければならない。ADT-QUELはそのために開発されたもので、通常のQUELのretrieve節とwhere節にADT関数及びそれらの関数合成の使用を可能にしたものである[4]。次に官能基間の相互関係を規定した検索の例を一つ示す。

例: 官能基"-NH₂"と"-COOH"を含む化合物の名前及び化学構造を出力せよ。ただし、2つの官能基の間に任意の部分構造が一つ結合されているものに限る。

```
range of t1 is COMPOUND  
range of t2 is CONNECTION  
range of t3 is COMP-CONN  
retrieve (t1.NAME, (@display t1:CHEM-GRAPH))  
where (include-NH2 t1.FUNC-GROUP)  
and (include-COOH t1.FUNC-GROUP)  
and t1.C# = t3.C# and t3.CG# = t2.CG#  
and (distance t2.GRAPH H2N CHO2) = 2
```

ここで、"display"は"CHEMICAL-GRAPH"ADT上で定義された関数である。この場合、WHERE節の条件を満たす化合物は複数ありうるので、その各々を引数として、"display"関数を適用させなければならない。"@"は"apply-to-all"という関数合成を表している。"Distance"は与えられたSC-Treeにおける与えられた2つの部分構造間の距離を計算するADT関数である。

3 関数型外部ビューを通してのアクセス

3.1 関数型外部ビューの導入

2.2節のスキーマでわかるように、関係型データベーススキーマは複数の関係によって構成され、また複数の関係間の関連は記述されていない。その結果、実世界の一つのオブジェクト(例えばある化合物)は複数の関係に存在している複数のタプル、いわゆる複合オブジェクト(Complex Object)というものによって表現せざるを得ない。つまり本来一つのもは、データベースの中では複数のものとして表現され、しかもそれらの間の関連は複数の関係のジョインという形で表現されている。このような表現法では、ユーザが化学構造を本来の一つのものとして認識しにくい上、それにアクセスするとき、関係型スキーマに基づいて、関係と関係のジョインを多く取るような冗長な問い合わせ文を書かねばならない。

DAPLEX[2]に代表される関数型データ言語は、文字、数値以外の非定型データを管理の対象とするいわゆる非

ビジネスアプリケーションの論理インタフェースとして適していると考えられる[3]。そこで、本システムでは化学構造のような複合オブジェクトが一つのものとして認識、操作できるように、2章で述べた拡張関係型データベースの上に、ユーザの認識に基づいた関数型外部ビューをかぶせることを提案する。このビューを通して、ユーザは関係型内部スキーマを全く意識する必要がなく、問い合わせを発することができる。

3.2 エンティティタイプの定義

関数型データモデルにおける基本概念はエンティティとその上で定義される関数である。エンティティはオブジェクトを表し、関数はその属性を与える。エンティティタイプは"def-entity-type"というLISPのマクロによって定義される。次に、化合物エンティティタイプの定義を示す。

```
(def-entity-type COMPOUND)
  ((ID char)
   (FG-BIT FUNC-GROUP)
   (SB-SEG SC-TREE)
   (COMPONENTS set-of SUBSTRUCTURE)
   .....))
```

ここで、"ID"、"FG-BIT"などはアクセサ関数と呼ばれ、エンティティの属性に対するアクセス手段を提供し、必要に応じて定義できる。アクセサ関数の値域となるものは、基本的に(a)基本データタイプ、(b)ADT、(c)エンティティタイプの3種類がある。"set-of"は多値関数を表している。

"ID"はシステムが用意する関数で、エンティティから、そのデータベース内における識別子へのマッピングである。"FG-BIT"は化合物の中に含まれる官能基のビット列を返す関数である。"SB-SEG"と"COMPONENTS"はそれぞれ化学構造の分割で得られたSC-Tree(エンティティ)および分割された部分構造(エンティティの集合)をすべて返す関数である。

外部ビューは複数のエンティティタイプによって構成される。図5に化学構造データベースの外部ビューを示す。図5において、丸は基本データタイプ、四角はADTs、楕円はエンティティタイプを表している。点線はエンティティタイプ間の階層を表していて、矢印のついた実線はアクセサ関数、二重矢印は多値関数を表している。

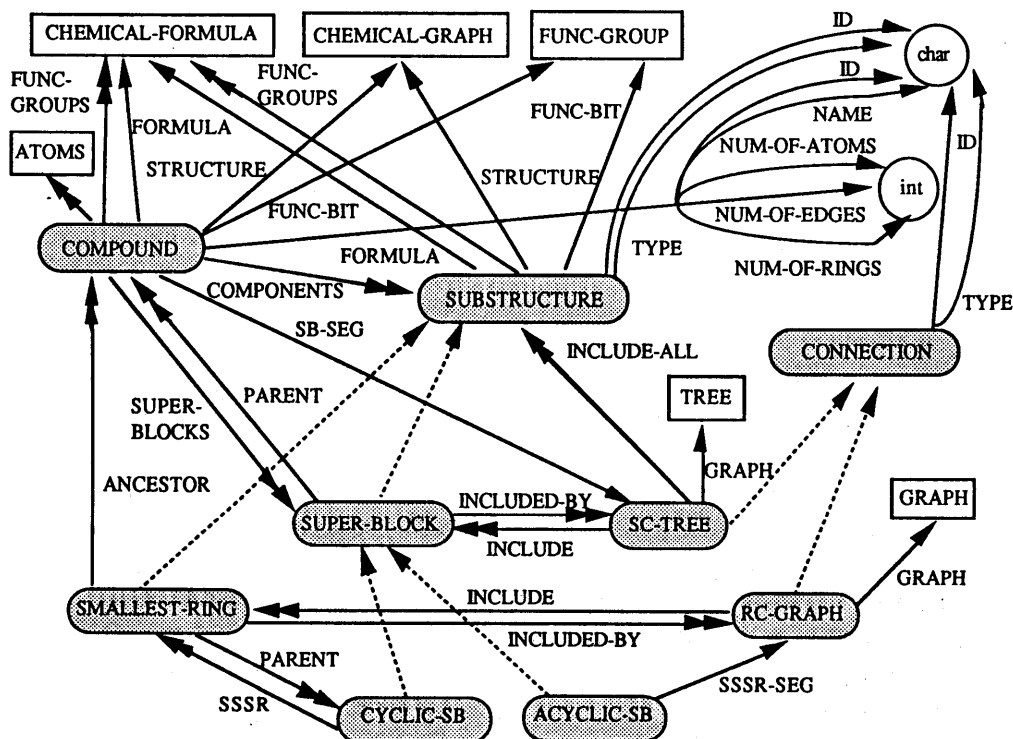


図5 関数型外部ビュー

3.3 アクセサ関数の定義

アクセサ関数の定義本体は外部ビューと内部スキーマ間の橋渡しであり、ADT-QUEL文によって与えられる。ここで"SB-SEG"の定義を例として示す。

```
(def-accessor-func SB-SEG(i)
  (range of t is COMP-CONNEX
    retrieve (cvt t.CG#)
    where t.C# = ID(i)))
```

上の定義において、iはエンティティ(この場合ある化合物)を表す。この関数は外部ビューにおけるエンティティ間の関係を定義しているが、これは実際データベース内に格納されているそれぞれのエンティティの識別子間の関係を辿って初めて求まるので、片方のエンティティに対して"ID"関数を適用して先ずデータベースの中に入り、そして求められたもう一方のエンティティの内部識別子に対して、"ID"の逆関数"cv"を使って、エンティティの世界へ逆変換する。

3.4 外部ビューを通しての問い合わせ

エンティティタイプ及びその上のアクセサ関数が定義された後、ユーザは内部スキーマ及びアクセサ関数の定義本体などを意識する必要がなく、外部ビューのみに基づいて、本研究で開発された関数型問い合わせ言語を使って問い合わせを発する。次に問い合わせの一例を示す。例1: -NH₂を含み、SC-treeがO-O-Oである化合物の構造を示せ。

```
(for-each (i COMPOUND)
  (where (and
    (include-NH2 (FG-BIT i))
    (match (tree-constant) (GRAPH (SB-SEG i))))))
  (@display (STRUCTURE i)))
```

ここで、"include-NH₂"は(FG-BIT i)で求めた官能基ビット列の中に"-NH₂"が含まれていれば"true"、でなければ"false"を返す関数である。"Tree-constant"は"TREE"ADT上で定義され、与えられたSC-Tree O-O-Oの内部表現(木構造)を表す定数関数である。GRAPHはエンティティタイプSC-TREE上で定義されたアクセサ関数で、(SB-SEG i)で求められたSC-Treeの木構造を返す。"Match"は"TREE"ADT上で定義され、引数として与えられた2つの木構造が同型

かどうかを判定する関数である。"STRUCTURE"は化合物の内部構造を返す関数で、where節で示された2つの条件を同時に満たす化合物に対してのみ、その2次元構造が表示される。

4 化学グラフデータベースシステムの構築

4.1 システム構成

本システムは図6に示されているように、G-BASEという商用関係型DBMSの上で構築されている。G-BASEはDM-LISPと呼ぶLISPインタフェースを提供している。LISPのS式で表現された問い合わせはLRTという関数群によってDM-LISPのフォームに変換され、そのフォームを評価することによってデータベースアクセスが行われる。

ADT-QUELインタプリタと呼ぶモジュールは、ADT-QUELの形式で表現された問い合わせをLRTが受け付けるLISPのS式に変換する。なお、LRTはADT関数を受け付けるように拡張された[4]。

ENTITY HANDLERと呼ぶモジュールは、3章で述べた外部ビューを通してのアクセスを可能にするために開発したもので、エンティティタイプの定義を司るデータ

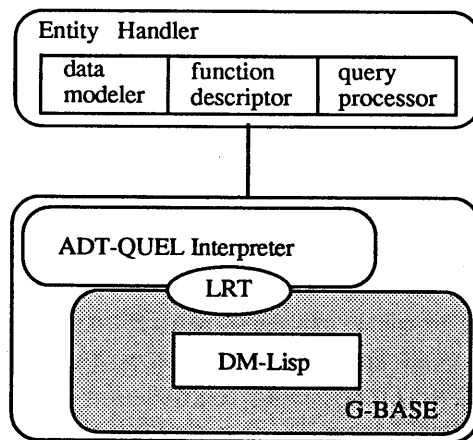


図6 システム構成

モデラとアクセサ関数の定義を管理する関数ディスクリプタ及び外部ビューを通しての問い合わせをADT-QUELの形式に変換する問い合わせプロセッサという3つのサブモジュールから構成されている。

4.2 問い合わせの変換

外部ビューで発せられた問い合わせは、問い合わせプロセッサによって、アクセサ関数の定義などを参照しながら、一定のルールに基づいて、ADT-QUELの形式に変

換され、更にADT-QUELインタプリタを通してG-BASEに渡され、処理される。

例えば、3.4節で示した例1は次のようなADT-QUELに変換される。

```
range of t1 is COMPOUND
range of t2 is CONNECTION
range of t3 is COMP-CONNEC
retrieve (@display t1.CHEM-GRAPH)
where (include-NH2 t1.FUNC-GROUP)
and (match (tree-constant) t2.GRAPH)
and t2.CG = t3.CG
and t3.C = t1.C
```

4.3 問い合わせ例

本システムでは、サブブロックと最小リングという限定された部分構造による検索のみならず、関数を必要に応じて定義することによって、様々な属性による検索ができ、また例1や次の例で示すようなより包括的な検索もサポートできる。

例2: 頂点数が6で、鎖状でない部分構造をもつ化合物の構造を示せ。

```
(for-each ((i COMPOUND)(j SUBSTRUCTURE))
  (where (and (= 6 (NUM-OF-ATOMS j))
              (not (= (TYPE j) 'ACYCLIC-SB))
              (member j (COMPONENTS i))))
    (@display (STRUCTURE i)))
```

この問い合わせは、頂点数が6の部分構造を単環の形で含んでいても、縮合環の一部として含んでいても、適合する化合物として検索される。関係型データベースに対して直接問い合わせるとき、この2つの場合に分けて考えないといけない。また、関係型スキーマを参照し、どの関係とどの関係をジョインしたら、目的の属性が求まるかを辿らないといけないから、次のような非常に複雑な問い合わせ文を書かざるを得ない。

```
range of t1 is COMPOUND
range of t2 is SUBSTRUCTURE
range of t3 is COMP-CONNEC
range of t4 is CONNEC-MAP
range of t5 is MAP-SUB
```

```
range of t6 is SUB-CONNEC
range of t7 is CONNEC-MAP
range of t8 is MAP-SUB
retrieve (@display t1.CHEM-GRAPH)
where t2.VERTICES = 6
and t2.TYPE != "ACYCLIC-SB"
and ((t2.S = t5.S and t5.M = t4.M
      and t4.CG = t3.CG and t3.C = t1.C)
     or
      (t2.S = t5.S and t5.M = t4.M and t4.CG = t6.CG
      and t6.S = t8.S and t8.M = t7.M
      and t7.CG = t3.CG and t3.C = t1.C))
```

5 結論及び将来課題

本研究では化学構造データベースを構築する際に、3次元の化学構造の記述、表現問題を検討し、それに基づいて、関係型データベースを拡張するためにADTの概念を導入した。それにより、化学構造データをより柔軟な形で表現され、アプリケーション側からの多様なアクセス要求がサポートできるようになり、総称表現を含み、化学構造におけるトポロジカルな結合関係以外の構造情報の表現問題についての見通しが得られた。また関数型ビューを構築し、それによって、関係型データベースにおけるデータの表現と実世界のデータに対するユーザの認識との間のギャップを埋めることができた。

今後、同型性判定処理の効率化及び問い合わせの最適化などの課題に取り組む予定である。

参考文献

- [1] Fujiwara, Y., Nakayama, T., A Graph Data Base for Storage of Chemical Structures Organized by the Block-Cutpoint Tree Technique, *Analytica Chimica Acta*, 133, 1981, pp.647-656.
- [2] Shipman, D., The Functional Data Model and the Data Language DAPLEX, *ACM Trans. Database Syst.*, Vol.6, No.1, 1981, pp.140-173.
- [3] Luan, Y.Q., Ohbo, N., Kitagawa, H. and Fujiwara, Y., *Functional Approach to Chemical Structure Databases*, DASFAA, Seoul, Apr.1989.
- [4] Jiang, S.J., Kitagawa, H., Ohbo, N., Suzuki, I., and Fujiwara, Y., *Abstract Data Types in Graphics Database*, Proceedings of IFIP TC-2 Working Conference on Visual Database Systems, Tokyo, Apr.1989.