

推論関係型データベース管理システム Adbis†

松尾文碩** 二村祥一** 高木利久***

推論関係型データベース管理システム Adbis は、関係型データベースである外延的データおよび Horn 集合である内包的データの管理機能と Horn 集合による推論機能を有する。Adbis 開発の目的は、学術研究用共同利用計算機センターの利用環境下で研究者が実験・観測データをデータベースとして統合できるようにすることにあった。Adbis は、DMP と HSI の 2 個のプログラムモジュールから構成され、前者は外延的データを管理し、後者は内包的データの操作、主に導出原理に基づく推論を行う。両者は、それぞれ単独で使用することができる。Adbis は、オペレーティングシステム FACOM OS IV/F4 の上に実現し、九州大学大型計算機センターにおいて利用者の使用に供している。

1. まえがき

現在、実験科学の分野における観測データをデータベース化し、データ解析システムなどの応用プログラムを開発するのに適したデータベース管理システム (DBMS) は皆無といってよい。実験・観測データ処理の究極の目的は、知識や法則の帰納にあるので、学術研究用データベースシステムの応用プログラムは、帰納を支援するために複雑な検索・検証条件を扱う能力が要求される。また、DBMS は企業体などの管理運営用として発達してきたため、既存 DBMS はその論理能力を別にしても、共同利用計算機センターなどで計算機システム管理者以外の利用者がデータベースを管理するには適していない。さらに、ほとんどの DBMS は応用プログラムの開発以外にもデータベース形成のために大量の計算機資源を消費する煩雑な作業が必要であり、非職業的計算機従事者が手軽に使用できない。

データベース統合支援システム Adbis (A data base integration support) は、上述の問題を解消するために筆者らが開発した学術研究用 DBMS である。それは、Horn 集合 (Horn set⁸⁾) による推論機能をもつ関係型 (relational) システムであり、その設計には共同利用計算機センターの利用形態に適合するように配慮した。Adbis は、オペレーティングシステム

μ FACOM OS IV/F4¹⁷⁾ の上に実現し、すでに九州大学大型計算機センターにおいて利用者の使用に供している。本稿では、Adbis の設計とその実現について報告する。

2. Adbis の設計方針

2.1 拘束条件

Adbis は、九州大学大型計算機センターのような学術研究用共同利用計算機センターの利用者が実験・観測データをデータベース化するための実用 DBMS として開発を計画した。このことから Adbis の設計に対して次の拘束条件が生じた。

- i) 学術研究用応用プログラムの開発に適していること；とくに、複雑な検索・検証条件を扱うための支援機能が充実していること。
- ii) 職業的計算機従事者でない研究者が比較的手軽に使用できること；すなわち、システムの習得が容易で計算機資源節約型であること。
- iii) 共同利用計算機センターの利用形態に適合していること。

以上の条件を満たすために、Adbis は基本的に推論機能をもつ関係型データ構造のシステムとして設計し、効率面での洗練を図った。さらに、Adbis 利用者は他のソフトウェアを使うのと同じ環境でデータベースを管理し、応用プログラムを作成できるように配慮した。

2.2 推論関係型データベース管理システム

科学的観測データ処理の究極の目的は、知識・法則の帰納にあると考えられる。したがって、学術研究用データベースシステムの応用プログラムは、帰納を支援するために、仮説の検証や反例の検索など論理的深

† Adbis—An Inferential Relational Database Management System by FUMIHIRO MATSUO, SHOOICHI, FUTAMURA (Computer Center, Kyushu University) and TOSHIHISA TAKAGI (Department of Computer Science and Communication on Engineering, Faculty of Engineering, Kyushu University).

** 九州大学大型計算機センター

*** 九州大学工学部情報工学科

さの大きい処理を行う能力が要求される。このため、Adbis は第1階述語計算 (first-order predicate calculus) の導出原理 (resolution principle) に基づく推論機能を備えている。応用プログラム作成者は、検証や検索のためにこの機能を使うことができる。ただし、Adbis が扱う論理式は導出の効率^{9),11)}のために Horn 集合に限定している。Adbis は、完全に実用システムを目指しているため、非効率性は実用システムの価値を著しく損なうからである。しかし、Horn 集合は導出の手続き的解釈⁹⁾により完全なプログラム言語として使えることが保証されている^{14),15)}ので、プログラムで記述できる検証・検索条件であれば、原理的には Horn 集合によって指定できる。

Adbis において関係型データ構造の採用は必然的であった。DBMS が推論機能を備えようとするれば、関係型以外にデータベースと論理を結合できるデータ構造はないからである。この両者の結合は、**推論関係型システム (inferential relational system¹³⁾)**と呼ばれるシステムを形成する。このシステムでは、関係は関係型データベース内のものと論理式によって定義されたものとに分けられる。前者は、**外延的データ (extensional data)** または **基本関係 (base relation)** と呼ばれ、後者は**内包的データ (intensional data)** または **仮想関係 (virtual relation)** と呼ばれる。質問 (query) は、基本関係ならびに仮想関係の連言 (conjunction) で与えられるが、仮想関係を含む場合はシステムが下降型 (topdown) の推論によって基本関係に対する検索条件を作り出す。推論関係型システムは、推論能力以外にも次のような利点がある。

- 1) 仮想関係によって基本関係から別の基本関係を作る操作を大幅に減少できるか、あるいはまったくなくすることができる。
- 2) (1)によって基本関係の冗長が減少するため、ディスクスペースを節約できる。
- 3) データ保全 (data integrity) のための拘束条件を論理式で与えることにより、推論機構によって保全性の検査が可能である。

Adbis では、基本関係から別の基本関係を作る機能をもたず、それをすべて仮想関係定義によって補う。これは、検索効率上は不利であるが、2次記憶領域を節約でき、有害な冗長、更新異常、挿入異常、消去異常の発生を回避できる。

2.3 関係型データベース管理システム

推論関係型 DBMS は、学術研究用データベースシ

ステムに適していると考えられるが、わが国では数理論理学教育が普及していないため、Horn 集合反証方式のような簡単な推論体系に基づくシステムですらすべての研究者によって受け入れられるとは限らない。そこで、Adbis は、推論機能を切り離し、関係型 DBMS としても使用できるようにした。もちろん、この場合、機能低下は避けられない。とくに、データ操作言語がないので、その部分は応用プログラム側で補わねばならない。しかし、関係型データ構造として抽象化した直接アクセスデータの操作機能だけでも応用範囲は広い。関係型データベースは、集合論的關係に数学的基礎を置き、理解が容易であるため、研究者用データベースシステムに適しているからである。

Adbis では、データベース管理者が正規化 (normalization) をとくに意識しないで済むように工夫した。この概念は、関係型データベースを理解する上で大きな障害になっているからである。Adbis では、基本関係を、集合論的關係、**関数**、および**ベクトル関数**のいずれかとして定義する。関係 (以後、集合論的關係をたんに関係と呼び、関係、関数、ベクトル関数を総称するときには基本関係と呼ぶ) は、属性 (attribute) 間に関数従属 (functional dependency) がないか、あるいはあらかじめあることがわからないものをいう。関数は、一つの属性が他の属性に関数従属である場合、ベクトル関数は複数の属性が他の属性に関数従属である場合をいう。関係、関数、ベクトル関数は、研究者にとってなじみ深い概念であり、これらによって属性間の関連 (relationship) を表現することにより、基本関係を自然に Boyce-Codd の正規形へと導くことをねらった。さらに上位の正規形、たとえば第4、第5正規形⁵⁾については、基本関係が静的であることから、データベース管理者がこれらを意識する必要はほとんどない。

2.4 共同利用計算機システムとの適合性

既存の商用 DBMS の多くは、一つの計算機システムに一つしか設置できない構造に作られている。したがって、その DBMS を使ったデータベースシステムは、一つの計算機システムに1個しか存在しえない。このような DBMS を共同利用計算機センターなどでセンター利用者が使用する場合、センター運用者と多数のセンター利用者によって一つのデータベースシステムを構築、維持するのは、両者にとって非常に不便である。このため、事実上、利用者が使用できない DBMS が多い。この環境で望ましいのは、利用者が

他のセンター利用と同じように特別の処置をせずに各自のデータベースシステムを構築することができる DBMS である。

Adbis は、機密保護・データ保護機能のある程度犠牲にすることになるが、データベース用ディスクボリュームを集中管理せずに、データベースをデータ管理システム³⁾ 管理下の非 VSAM⁹⁾ ファイルとして任意のディスクボリューム上に作ることができるように設計した。また、Adbis 自体はプログラムライブラリの形式とした。したがって、Adbis 利用者は、普通のプログラムライブラリやファイルを使うのと同じように、各自のデータベースシステムの応用プログラムを作成し、データベースを管理することができる。

2.5 既存ソフトウェア資源の活用

学術研究用データベースシステムにおいては、数学ルーチン、統計ルーチン、図形処理ルーチンをはじめとしてこれまで蓄積されてきたソフトウェア資源の使用は不可欠である。応用プログラムに組み込み可能なこれらのプログラムは、Fortran サブルーチン形式のものが量質とも他のものに圧倒的に勝っている。また、Fortran は、システムプログラムの記述に適していないにもかかわらず、移植性への配慮やオブティマイザの性能の良さなどからシステムプログラムの開発に使われることが多い。そのため、Fortran の機能不足を補うためのアセンブラプログラムがいろいろな形で蓄積されている¹⁸⁾。さらに、わが国では研究者が使用するプログラム言語の 95% 以上は、Fortran であり、PL/I, COBOL の習熟者が少ないという事情がある。したがって、学術研究用応用プログラムは Fortran で書かれることが多いと思われるため、Adbis の基本インタフェースは、Fortran サブルーチン呼出し形式とした。さらに Horn 節からサブルーチン形式のプログラムを呼び出すことができる機構を設け、既存ソフトウェアの活用を図った。

3. Adbis の構成

図 1 に示したように、Adbis は二つのモジュール、DMP(Data Manipulation Primitives) と HSI(Horn Set Interpreter) から構成される。両モジュールと応用プログラムとのインタフェースは、DMP が 25 個、HSI が 6 個の Fortran サブルーチンである。

DMP は、関係型の外延的データを管理し、HSI は Horn 集合による内包的データを取り扱う。以後、外延的データベースをデータベースと呼ぶことにする。

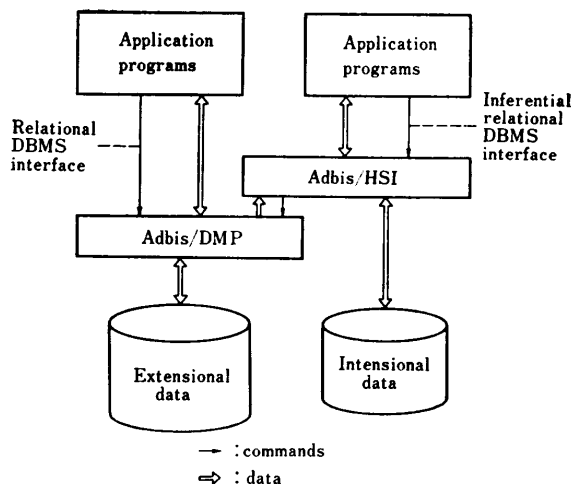


図 1 Adbis を使ったデータベースシステム
Fig. 1 The database system using Adbis.

DMP は、単独で使用でき、2.3 節で述べた関係型 DBMS の機能を与えるが、HSI は DMP と組み合わせて使用し、この結合が 2.2 節の推論関係型 DBMS を実現する。ただし、データベースを扱わないシステム作成のためには、HSI は単独で使用できる。

4. DMP=関係型 DBMS

外延的データ、すなわちデータベースの管理機能である DMP は、機能的にデータベースの創成・消去を行う部分とデータ操作を行う部分とに分けられる。データベースを創成するためには、他の DBMS 同様、最初にデータ定義言語によってデータベースを定義する。この言語を **DMP-DDL** (Data Definition Language) と呼ぶ。DMP のデータ操作は、データの蓄積、検索、更新、集合演算などを行うが、基本関係から新しい基本関係を作る機能をもたない。

なお、Adbis は Date⁵⁾ などが言うところの関係モデル (relational model) のシステムではなく、DMP は関係代数 (relational algebra¹⁶⁾) に基づく質問 (query) を処理するデータ操作機能を有しない。また、関係計算 (relational calculus¹⁶⁾) 型質問に関しても、DMP の機能は完全ではなく、その能力は大きくない。しかし、これは HSI によってかなりの部分を補うことができる。

4.1 データベースの構成とその創成

Adbis のデータベースは、基本関係の集合である。基本関係は、属性値の組 (tuple) の集合として表現され、これに対応した構造の物理的データを表形式 (tabular form) データと呼ぶ。基本関係は、関係、関

数、およびベクトル関数に分類されるが、表形式のデータ構造はいずれも同じである。関数では、関数値となる属性は組の最右端に位置し、ベクトル関数の場合は、値であるベクトルの要素となる属性群が組の右側を占める。DMP は、属性値から組を高速に検索するための索引を作る。これを**転置形 (inverted form)** データと呼ぶ。関係では、すべての属性について転置形データが作られるのに対し、関数とベクトル関数では、とくに指定しない限り関数値の属性についての転置形は作られない。すなわち、逆関数の存在を宣言したときだけ関数値についての転置形が作られる。DMP は、一つの基本関係ごとに表形式と転置形をいっしょにして一つのファイルを作成する。このファイルを**データファイル**と呼ぶ。データファイル群をデータベースとして統合するための情報は、**マスタファイル**に格納されている。マスタファイルは、DMP-DDL によるデータベース定義を記述したファイル (**データベース定義ファイル**) を TSS のエディタなどで作っておき、DMP のサブルーチン **DEFD\$D** (define dd) を呼び出すことによって作成される。生データからデータファイルを作るには、マスタファイル作成後、**CREA\$D** (create) を呼び出す。データファイルを消去するには **CLEA\$D** (clear) を使う。

DMP で許している属性値の型 (type) は、2 バイトならびに 4 バイト長の整数、4 バイトならびに 8 バイト長の実数、および任意の長さ k の固定長の文字列であり、DMP-DDL ではそれらは、それぞれ、I2, I4, R4, R8, C k で表される。

DMP の機密保護機能は、現在、データベース単位にパスワードが設定できるだけであり、商用 DBMS のようにきめ細くデータを保護する機能をもたない。パスワードは、データアクセスの権限の強さによって三つのレベルを設けている。リードパスワードは、検索だけが可能である。ライトパスワードでは、検索とレコードの追加・消去が許される。マスタパスワードでは、すべての操作が可能である。

図 2 に DMP-DDL によるデータベース定義例を示す。コマンド DATABASE, RELATION, FUNCTION, VECTFUNC, VARIABLE は、それぞれ、データベース、関係、関数、ベクトル関数、属性の定義を指示する。オペランド INVERSE は、逆関数の存在を宣言する。すなわち、転置形データの作成を指示する。DOMAIN と RANGE は、属性値の範囲を示す。SPACE は、使用するディスク領域の大きさ

```
* DATABASE DEFINITION OF CARDB
DATABASE NAME (CARDB)
ADMINISTRATOR (F9999)
EXPLANATION ('CAR-MANUFACTURER DATABASE')
MASTERPASSWORD (ALMIGHTY)
WRITEPASSWORD (UPDATE)
READPASSWORD (RETRIEVE)
RELATION NAME (MANUF CAR)
ALIAS (MC)
VARIABLE (MANUF, CAR)
SPACE (3000)
FUNCTION NAME (CARTYPE)
ALIAS (CT)
VARIABLE (CAR)
TYPE (C12)
RANGE ('SEDAN', 'COUPE', 'CONVERTIBLE', 'FORMULA')
INVERSE
VECTFUNC NAME (MANUFACT)
ALIAS (M)
VARIABLE (MANUF)
VECTOR (MANUNAME, SHARE, LOC)
INVERSE (LOC)
VARIABLE NAME (MANUF) TYPE (I4)
VARIABLE NAME (CAR) TYPE (I4)
VARIABLE NAME (MANUNAME) TYPE (C30)
VARIABLE NAME (SHARE) TYPE (R4)
VARIABLE NAME (LOC) TYPE (C15)
DOMAIN ('DETROIT', 'HAMOVER', 'MILWAUKEE', 'TOYOTA')
END
```

図 2 DMP-DDL によるデータベース定義

Fig. 2 A database definition written in DMP-DDL.

(単位キロバイト)を指示し、省略時は創成時に生データの大きさをもとに DMP が自動的に決める。ディスクに関しては、このほか機番やボリューム通番を指定することができる。

DMP は、データベース創成のために必須なサブルーチン以外に、創成を補助する 4 本のユーティリティプログラムを備えている。これらは、DMP-DDL によるデータベース定義の構文チェック；データベース創成用生データの保全性チェック；データベースから創成用形式データへの逆変換；データベース用ファイルの確保量および使用量の表示などを行う。

4.2 データ操作

表形式データのレコードは、基本関係の要素である。DMP は、レコードを識別するためのキーを作る。このキーと基本関係のキーとなる属性または属性の組を区別するために、前者を**レコードキー**と呼ぶ。もちろん、二つのキーの値は 1 対 1 対応しているが、レコードキーは利用者から見えない。転置形データにおける基本関係についての情報は、レコードキーである。属性についての条件を指定し、サブルーチン **FIND\$D** (find) を呼ぶと、転置形をもとにレコードキーの集合が求まる。この集合を**キー集合**と呼ぶ。検索したキー集合は、サブルーチン **SAVE\$D** (save) によって名前をつけて保存することができる。キー集合間は、サブルーチン **MEET\$D** (meet), **JOIN\$D** (join), **DIFF\$D** (difference) によって積集合、和集合、差集合を求める演算が可能である。キー集合からそれに対応する基本関係の要素の集合を求めるには、サブルーチン **EVAL\$D** (eval) を使う。この集合を

値集合と呼ぶ。値集合が EVAL\$D の用意した1次記憶バッファ領域より大きい場合は、値集合の先頭部分だけが1次記憶上に置かれ、残りは作業用2次記憶上に置かれる。残りを次々に1次記憶上に取り出すには、サブルーチン GETN\$D (get next) を使う。値集合は、サブルーチン SORT\$D (sort) によってソートが可能である。関数またはベクトル関数の関数値を求めたり、関係のキー値を指定してその要素があるかないかを調べるには、サブルーチン VALU\$D (value) を用いる。

このほか、データファイルにデータの追加、削除、更新を行うためのサブルーチン INSE\$D (insert), DELE\$D (delete), UPDA\$D (update) があるが、これらの機能は他のシステムとそれほど異なるものではない。

4.3 データファイル

DMP 実現のための核となるデータファイルの構造は、表形式および転置形データとも、索引部に B+木⁴⁾を採用した。1次記憶上のページバッファ領域は、LRU 方式によって管理し、ディスクの読み書きの仮想化を行っている¹⁾。ページサイズは、IBM3330 および 3350 型ディスクのトラックサイズに適合するように、3,000 バイトを採用した。

5. HSI+DMP=推論関係型 DBMS

DMPに HSI を組み合わせることによって、Adbis は 2.2 節で述べた推論関係型 DBMS として機能する。HSI の機能は、Horn 集合である内包的データの操作であり、そのおもなものは導出原理に基づく推論による反証 (refutation) である。

5.1 Horn 集合とその反証

Horn 集合は、たかだか1個の正リテラル (positive literal¹²⁾) しかもたない節 (clause¹²⁾) の集合であり、この節は Horn 節と呼ばれる。HSI における Horn 節の形式は、図3に示したように、正リテラルだけを使った Gentzen の Sequenz 形式⁷⁾をとり、Kowalski の流儀¹⁰⁾とは、右辺と左辺のリテラルの位置、矢印の方向が逆である。Horn 節における左辺と右辺のリテラルの数をそれぞれ m, n で表すと、 m, n の値の組み合わせによって Horn 節は4種類に分類される。それは、正節 (positive clause), 負節 (negative clause), 混合節 (mixed clause), 空節 (empty clause) であり、その m と n の値の組合せは、それぞれ、 $m=0$ かつ $n=1$, $m>0$ かつ $n=0$, $m>0$ かつ $n=1$,

```

DEFINE uncle;
parent(u,x),parent(u,v),parent(v,y),man(x)->uncle(x,y);
father(x,y)->parent(x,y);
mother(x,y)->parent(x,y);
father(x,y)->man(x);
->father('Kanmutennou','Kazuharashinnou');
:
->father('Tadamori','Riyomori');
->father('Tadamori','Tsunemori');
:
->father('Tadamori','Tadanori');
:
->father('Riyomori','Shigemori');
->father('Tsunemori','Atsumori');
->father('Tadamori','Tadayuki');
:
->mother('Tokiko','Tomomori');
->man('Atsumori');
:
END

```

図3 外部形式の Horn 集合

Fig. 3 A Horn set in the outer form of Adbis.

$m=0$ かつ $n=0$ である。HSI の主要な機能は、負節によって与えられた質問に対し、混合節と正節の集合を使って下降型の推論を行い空節を得ること、すなわち反証を行うことである。質問には、変数を含まない基礎節 (ground clause) による閉質問 (closed query) とそうでない開質問 (open query) がある⁶⁾。閉質問は、肯定または否定を答えとし、開質問は負節の左辺を充足不可能とする定数または定数の組の集合を答えとする。たとえば、いま図3の Horn 集合があるとき、閉質問

uncle ('Tadanori', 'Atsumori')—>

に対し、HSI は推論によって空節を導き、肯定の答えを返す。開質問

uncle ('Tadanori', x)—>

に対する答えは、{'Shigemori', ..., 'Atsumori', ...} となる。つまり、この場合、答えとなる定数は、 x にそれを代入した基礎節から空節が導かれるものである。HSI では、開質問の答えの個数 n を制限することができる。この場合は、HSI の反証方式によって得られた最初の n 個の定数または定数の組が答えになる。HSI における質問に対する反証の実行は、サブルーチン RESO\$H (resolve) を用いて行われる。

5.2 Horn 集合の形式と内包的データ

図3に示した Horn 節の形式を外部形式 (outer form) と呼ぶ。外部形式は、HSI とその外部との入出力に使われるだけで、この上で反証は行われない。反証時は、導出の高速化と記憶領域の節約のために、内部形式 (inner form) と呼ぶ形式を使用する。内部

形式の Horn 節は、基本的には2バイト長（一部、4バイト長）に符号化した述語記号、関数記号、変数および定数により節をポーランド式表記法によって表したものである。述語記号と関数記号の引数の個数は、それらの符号の中に埋め込まれている。

外部形式から内部形式への変換には、サブルーチン PARS\$H (parse) を使う。内部形式の Horn 集合は、LINK\$H (link and edit) によって複数の集合を結合編集して1個の集合にまとめることができる。内包的データとは、おもに内部形式の Horn 集合のことを指す。これは、SAVE\$H (save) によって2次記憶に保存することができ、REST\$H (restore) によって1次記憶に復元することができる。しかし、HSI は DMP のマスタファイルのような内包的データの管理情報を保持する機能をもたない。また、HSI には外部形式の Horn 集合に対する編集機能はなく、外部形式データの作成等には TSS のエディタなどを用いる。

5.3 データベースとプロシジャ

HSI は、DMP が管理する基本関係を変数をもたない正節、すなわち正基礎節の集合とみなす。関数とベクトル関数は、項 (term) としてではなく関係と同じように正節として扱われる。図3の Horn 集合は、大部分の節が正基礎節であり、この部分を DMP の関係として定義することにより、内包的データを小さくすることができる。

データベースでは、ある関係が定義されているとき、2次記憶量節約のためにその否定の関係を陽に定義することは少ない。つまり、データベースシステムでは、データの完全性を前提とした応用プログラムが作られることが多い。そこで、HSI では、たとえば—father ('Kiyomori', 'Atsumori') のような形式のリテラルの使用をデータベースに限り許すようにした。これが Horn 節中にリテラルとして使われたときは、関係 father において father ('Kiyomori', 'Atsumori') に対応したレコードがないとき、—>—father ('Kiyomori', 'Atsumori') という基礎節があたかも存在するかのよう扱う²⁾。

HSI では、リテラルによって実行形式のサブルーチンプログラムを呼び出すことができる。このプログラムを HSI では、プロシジャと呼び、データベース同様、正基礎節の集合とみなす。基本的演算をプロシジャで与えることによって、HSI の Horn 集合はプログラミング言語として機能する。また、Horn 集合をプログラミング言語とみたとき、Horn 集合で記述

するのが困難な手続きの表現をプロシジャによって補うことができる。

5.4 HSI の反証方式

導出原理による反証は、本質的にしらみつぶしの非決定的手順であり、これを求値機構として利用する場合、反証が違えば値が変わる可能性がある。閉質問の場合は、導出原理に基づく推論に当たってどの Horn 節を選択するかは、実行速度、すなわち効率に関係するだけであり、すべての答えを求める開質問の場合にも、選択の方式はほとんど依存しない。しかし、最初の n 個の答えを求める開質問の場合、とくに $n=1$ の場合には、Horn 節の選択制御の問題は結果と効率の両方に直接関係する。

HSI で採用した方式は、1段の先読み (look-ahead) によってシステムが自動的に選択する方式であり、unit preference strategy¹²⁾ に類似した選択を行う。これは、数論的関数の計算可能性と反証における後戻りとの関係についての理論的考察に基づいているが、詳細はここでは割愛する。

6. Adbis システムプログラム

Adbis は、オペレーティングシステム FACOM OS IV/F4 の上で動作する。Adbis のプログラムは、ほとんど Fortran 77 で記述され、一部アセンブラ言語で書かれている。DMP は、ユーティリティプログラムを含め、約7,000行の Fortran プログラムと若干のアセンブラプログラムから成る。HSI は、4,300行の Fortran プログラムである。実行時のプログラム領域の大きさは、DMP と HSI を同時に組み込んだとき、最大575キロバイトであるが、ほとんどの応用プログラムで使用する部分は、450キロバイト以下である。とくに、DMP の検索機能部分だけのときは、284キロバイトでよい。

7. む す び

Adbis の DMP は1981年6月に、また HSI は同年12月に九州大学大型計算機センターにおいて利用者に公開した。これによって、Adbis の開発計画は、ひとまず初期の目的を達成した。現在、筆者らは Adbis を使って XDT (CRYStallographic DaTa) と呼ぶデータベースシステムを開発している。XDT は、英国 Cambridge Crystallographic Data Centre が集積、配布を行っている有機および有機金属化合物の結晶構造データを扱うシステムである。この開発を

通して, Adbis の機能強化, 改善, 洗練を図る計画である。機能強化の一つとして, Tsuno と呼ぶ会話型データ操作言語 (DML) の開発に着手している。さらに, 非B木索引技法の開発, Horn 集合反証器の改良, 知識ベース管理機能の強化などを行う予定である。

謝辞 Adbis の開発に当たり, 激励と助言をいただいた九州大学大型計算機センター長加納省吾教授と九州大学工学部吉田将教授に謝意を表す。

参 考 文 献

- 1) Bayer, R. and McCreight, E.: Organization and Maintenance of Large Ordered Indexes, *Acta Inf.*, Vol. 1, No. 3, pp. 173-189(1972).
- 2) Clark, K. L.: Negation as Failure, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, pp. 293-322, Prentice-Hall, New York (1978).
- 3) Clark, W. A.: The Functional Structure of OS/360: Part III. Data Management, *IBM Syst. J.*, Vol. 5, No. 1, pp. 30-51(1966).
- 4) Comer, D.: The Ubiquitous B-tree, *Comput. Surv.*, Vol. 11, No. 2, pp. 121-137(1979).
- 5) Date, C. J.: *An Introduction to Database Systems*, Third ed., p. 574, Addison-Wesley, Reading, Mass. (1981).
- 6) Gallaire, H., Minker, J. and Nicolas, J. M.: An Overview and Introduction to Logic and Data Bases, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, pp. 3-30, Prentice-Hall, New York (1978).
- 7) Gentzen, G.: Untersuchungen über das logische Schliessen, *Math. Z.*, Vol. 39, pp. 176-210, 405-431(1935).
- 8) Henschen, L. and Wos L.: Unit Refutations and Horn Sets, *J. ACM*, Vol. 21, No. 4, pp. 590-605(1974).
- 9) Keehn, D. G. and Lacy, J. O.: VSAM Data Set Design Parameters, *IBM Syst. J.*, Vol. 13, No. 3, pp. 186-212(1974).
- 10) Kowalski, R.: Predicate Logic as Programming Language, *Information Processing 74*, North-Holland, Amsterdam, pp. 569-574(1974).
- 11) Kuehner, D.: Some Special Purpose Resolution Systems, in Meltzer, B. and Michie, D. (eds.), *Machine Intelligence 7*, Edinburgh Univ. Press, Edinburgh, pp. 117-128(1972).
- 12) Loveland, D. W.: *Automated Theorem Proving: A Logical Basis*, p. 405, North-Holland, Amsterdam (1978).
- 13) Minker, J.: Search Strategy and Selection Function for an Inferential Relational System, *ACM Trans. Database Syst.*, Vol. 3, No. 1, pp. 1-31(1978).
- 14) Šebelík, J. and Štěpánek, P.: Horn Clause Programs Suggested by Recursive Functions, *Logic Programming Work Shop*, pp. 348-359(1980).
- 15) Tärnlund, S.-Å.: Horn Clause Computability, *BIT*, Vol. 17, No. 2, pp. 215-226(1977).
- 16) Ullman, J. D.: *Principles of Database systems*, p. 379, Computer Science Press, Potomac, Maryland (1980).
- 17) 計算機マニュアル FACOM OS IV/F4 解説書, 富士通(株).
- 18) 計算機マニュアル FACOM OS IV/F4. TAC/LIB 解説書, 富士通(株).

(昭和57年7月16日受付)
(昭和57年10月4日採録)