

データベース操作のための Horn 集合反証器†

松 尾 文 碩** 高 木 利 久***

データベースの操作を目的とした Horn 集合による推論機構である Horn 集合反証器について述べる。この反証器は、導出原理に基づく推論において前提となる Horn 節の選択を縦型探索方式で行うが、その選択は自動的である。選択方式は、数論的関数の計算可能性に関する理論的考察に基づいている。この反証器は、推論関係型データベース管理システム Adbis の構成要素として開発された。九州大学大型計算機センターにおいては、現在 Adbis を使った二つの科学用データベースシステムが公開されている。そこで反証器は、質問処理のための強力な道具として使われている。

1. はじめに

$A_1, \dots, A_m, B_1, \dots, B_n$ を原子論理式 (atomic formula), $\neg$ および $\vee$ をそれぞれ否定 (negation) と選言 (disjunction) を示す論理結合子とすれば、

$$(\neg A_1) \vee \dots \vee (\neg A_m) \vee B_1 \vee \dots \vee B_n \quad (1)$$

の形の論理式を節 (clause) と呼ぶ。本稿では、(1) の節を次の形式で表す。

$$A_1, \dots, A_m \rightarrow B_1, \dots, B_n \quad (2)$$

(2) において $n=0$ または 1 のとき、この節は Horn 節¹⁵⁾ または McKinsey 論理式^{17), 19)} と呼ばれる。Horn 集合 (Horn set¹²⁾) は、Horn 節の集合である。Kuehner¹⁵⁾ および Henschen と Wos¹²⁾ は、充足不可能 (unsatisfiable) な Horn 集合の反証 (refutation) が単純かつ効率的な戦略をもつ導出推論系 (resolution inference system) によって実現可能であることを示した。Horn 集合の能力を特徴づける別の側面は、その反証を手続き的に解釈することにより、Horn 集合がプログラミング言語として機能することである。Kowalski¹⁴⁾ によって提唱されたこの手続き的解釈の方法は、Hill¹³⁾ により LUSH と名づけられた。Horn 集合の手続き記述能力の完全性は、Hill¹³⁾, Andreka と Nemeti⁸⁾, Tärnlund²¹⁾, Šebelík と Štěpánek¹⁸⁾ および松尾¹⁾ によって示されている。LUSH に基づくプログラミング言語 PROLOG がマルセイユ大学、エジンバラ大学などで作られた。さらに、LUSH または PROLOG のデータベースシステムへの応用が実験的

に試みられている¹⁰⁾。

著者らは、Horn 集合をデータ操作言語 (Data Manipulation Language) とする関係型データベース管理システム Adbis²⁾ を開発し、実用に供している^{3), 4)}。Adbis の中核をなすプログラムモジュールは、Horn 集合に対する導出推論系であり、ここではこれを 'Horn 集合反証器' (Horn set refuter) と呼ぶことにする。本稿では、この反証器について述べる。

2. データベース操作のための Horn 集合反証器

$n=0$ または 1 の場合の (2) で表される Horn 節は、 $m=0$ かつ $n=1$ の場合を '正節' (positive clause); $m>0$ かつ $n=0$ の場合を '負節' (negative clause); $m>0$ かつ $n=1$ の場合を '混合節' (mixed clause); $m=0$ かつ $n=0$ の場合を '空節' (empty clause) と呼ぶ。また、 $n=1$ の場合、すなわち正節と混合節を '正規節' (regular clause) と呼び、正規節の集合を '正規集合' と呼ぶ。本稿では、原子論理式を 'アトム' (atom) と呼ぶことにする。

データベースの操作のために Horn 集合反証器 (以後、たんに反証器と呼ぶ) に要求される機能は、基本的には、正規集合 R と負節 Q が与えられたとき、導出原理 (resolution principle) に基づく推論によって $R \cup \{Q\}$ から空節を導くことである。反証器が用いる推論規則は、ただ 1 個である。これは、負節と正規節を前提 (premise) とし、負節を推論結果 (consequence) とするものであり、いま A, B をアトム; Γ, Δ, Λ をアトムの有限系列または空系列; η, σ を代入 (substitution¹⁶⁾); 表現 E への代入 σ の代入結果を $E\sigma$ で表すことにすれば、推論規則は次のように表すこ

† A Horn Set Refuter for Database Manipulation by FUMIHIRO MATSUO (Computer Center, Kyushu University) and TOSHIOHISA TAKAGI (Department of Computer Science and Communication 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

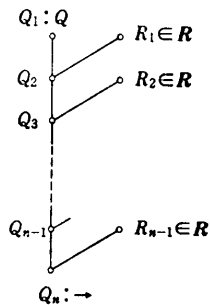


図 1 R による Q の反証
Fig. 1 Refutation of Q by R .

とができる。

[Horn 集合反証器の推論規則]

前提 1: $A, A, A \rightarrow$

前提 2: $\Gamma \rightarrow B$

結果: $\Delta\sigma, \Delta\sigma, \Gamma\eta\sigma \rightarrow$

ここで、 η は A と B に対する改名代入 (renaming substitution¹⁶⁾); σ は A と $B\eta$ の mgu (most general unifier¹⁶⁾) である。

正規集合 R による負節 Q の反証は、図 1 に示したような節の木構造によって表される。すべての i ($1 < i \leq n$) に対し、 Q_{i-1} と R_{i-1} はそれぞれ推論規則の前提 1 と前提 2 であり、 Q_i はその結果である。そして、 Q_1 は Q であり、 Q_n は空節であり、 $R_i \in R$ ($1 \leq i < n$)。 Q の左辺のアトムを系列を θ で表すと、 R による Q の反証が存在するとき、 $R \vdash \theta$ と書く。

このように、反証器は正規集合 R と負節 Q を与えると、下降型 (top-down) の input deduction¹⁶⁾ を実行する。 Q の左辺 θ が変数をもたないアトム系列であるとき、 Q を ‘閉質問’ (closed query) と呼び、変数を含むときは ‘開質問’ (open query) と呼ぶ。 θ が変数 $x_1, \dots, x_n$ を含んでいることを示すために $\theta[x_1, \dots, x_n]$ と書くことにする。 $x_1, \dots, x_n$ にそれぞれ定数 $c_1, \dots, c_n$ を代入した結果は $\theta_{x_1, \dots, x_n}[c_1, \dots, c_n]$ または $\theta[c_1, \dots, c_n]$ で表す。閉質問の場合、 $R \vdash \theta$ のとき反証器は肯定の答えを返す。反証を作ることには失敗したときには、否定の答えを返す⁹⁾。開質問の場合、 θ に含まれるすべての変数を $x_1, \dots, x_m$ とすると、反証器に要求される答えは、

$$\{(c_1, \dots, c_m) \mid R \vdash \theta_{x_1, \dots, x_m}[c_1, \dots, c_m]\}$$

である。反証器は、負節にそれらの定数を代入したあとに演繹を実行するのではなく、 $\theta[x_1, \dots, x_m] \rightarrow$ を図 1 の Q_1 とする反証において、 $x_1, \dots, x_m$ にそれぞれ代入される定数 $c_1, \dots, c_m$ の組を答えの一つの要素とする。したがって、反証器は異なる定数が代入される

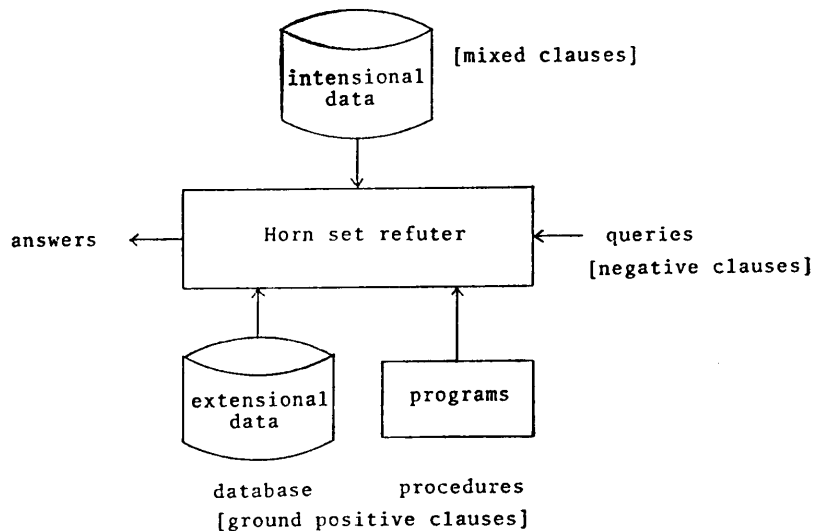


図 2 Adbis 内蔵の Horn 集合反証器
Fig. 2 The Horn set refuter used in Adbis.

反証をすべて網羅的に作り出さねばならない。また、答えとなる集合の大きさの上限 n をあらかじめ指定した開質問の処理を要求されることがある。この場合、答えの大きさが n を超えるときには、答えは反証器がつつぎに作り出す反証によって得られる最初の n 個の異なった定数の組である。したがって、答えは反証の作り方によって異なることになる。反証方式については 3 章で述べる。

反証器に与える正規集合 R に関して、Adbis ではこれを統一的に扱う方式をとっていない。変数を含まない正節、すなわち ‘正基礎節’ (positive ground clause) の集合は、関係型データベースとしてそれ以外の正規集合とは異なる管理が可能である。前者は、‘基本関係’ (base relation, basic relation) または ‘外延的データ’ (extensional data) と呼び、後者は ‘仮想関係’ (virtual relation) または ‘内包的データ’ (intensional data) と呼ぶ。また、サブルーチン形式のプログラムを ‘プロシジャ’ と呼び、正基礎節の集合とみなす。したがって、正規集合の形式として図 2 に示したように、外延的データ (正基礎節集合)、内包的データ (おもに混合節集合)、プロシジャ (正基礎節集合) の 3 種類がある。

推論規則の適用において、前提 1 のアトム A が変数を含み、前提 2 が外延的データに含まれる正基礎節である場合、推論規則として書かれた演算実行の代りに前提 2 となりうるすべての正基礎節を外延的データから抽出することが可能である。より正確に言えば、 $x_1, \dots, x_m$ を A に現れるすべての変数であるとし、そ

れぞれに定数 $c_1, \dots, c_m$ を代入したアトムを $A[c_1, \dots, c_m]$ と書くとき、データベース検索機能によって A から

$$C^A = \{(c_1, \dots, c_m) \mid R_E \vdash A[c_1, \dots, c_m]\}$$

が求まる。ここで、 R_E は外延的データである。閉質問に対する処理を効率的に行うためには、この機能は必須である。

データベースシステムでは、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まれであり、関係の完全性を前提とした処理が要求されることが多い。いま、 P を n 引数述語記号、定数 $c_1, \dots, c_n$ をその引数とするアトムを $P(c_1, \dots, c_n)$ で表す。外延的データにある P を述語記号とする正基礎節の集合を R とする。いま、関係 T が

$$T = \{(c_1, \dots, c_n) \mid \neg P(c_1, \dots, c_n) \in R\}$$

によって与えられているとき、あたかも集合

$$R' = \{\neg P(c'_1, \dots, c'_n) \mid (c'_1, \dots, c'_n) \notin T\}$$

(ここで、 $c'_1, \dots, c'_n$ は定数)

が存在するような処理が必要である。したがって、この場合、反証器には否定記号を伴ったアトムの処理が要求される。ここでは、これを「負リテラル」(negative literal) と呼ぶことにし、アトム的一种とみなす。いま、 P を述語記号とする負リテラルが推論規則の前提1の A として選択されたとしよう。 A から否定記号を除去したものを A' とすると、 A' と R に属するすべての節の右辺とが単一化可能(unifiable¹⁶⁾)でないとき推論が成立するものとし、

$$A, A' \rightarrow$$

が結果として導かれることにする。しかし、アトム A から C^A を求めたようなデータベース検索機能は、負リテラルに関しては R' が実在しないのであるから動作しない。

さて、プロシジャを正基礎節集合として反証器に与えるのは、二つの意味がある。一つは、基本演算(primitives)を反証器に与えることである。これは、反証器に計算機の機能を公理として与えることであるが、Adbis では基本演算を固定せず、任意のサブルーチン形式のプログラムが基本演算になりうるようにしているので、反証器はすべての計算機資源を使うことができる。第二は、Horn 集合で記述することが困難な手続きや効率が悪視される手続きはプロシジャとして与えることによって、プログラミング言語処理系としての Horn 集合反証推論系の弱点を補うことができ

ることである。

3. 反証方式

反証器の設計における最大の問題は、推論規則の適用において前提1の負節からアトム A として何を選ぶか、また正規集合からどの節を前提2として選ぶかを決定する方式にある。Adbis の反証器では、推論規則のある適用において選択肢が複数個あるとき、縦型(depth-first)探索方式の選択によって演繹を行う。この理由は、後述するように反証器を関数の求値機構として使用する場合、縦型探索方式の後戻り(backtrack)を行って最初に得られた反証が重要な意味をもつからである。問題によっては、横型(breadth-first)探索方式のほうが有効なことがあるが、ここでは横型方式については触れない。さて、縦型方式をとると、反証器の効率を決める最大の要因は後戻りの回数になるので、上記の正規節とアトムを選ぶ方式が反証器の性能を決めてしまう。この選択に関して次のような方式が考えられる。

- (a) 反証器が自動的に選択する。
- (b) 選択の情報は、すべて反証器の外部から与える。
- (c) (a)と(b)の中間型で、選択の情報は部分的に反証器に与えられ、外部からの指定のない部分は反証器が自動的に選択する。

反証器の負担は、(a)が大きく、(b)が小さい。しかし、(a)は自動選択方式が十分に効率的であるならば、使用上は最もすぐれている。そこで、Adbis の反証器では自動選択方式を採用した。この方式が効率的に問題になる場合は、外部から反証過程を制御できるようにする計画であったが、Adbis は1981年の公開以来、その機能を必要とする問題領域では使われていないため、実現を保留している。

Adbis の反証器の自動選択方式は、松尾¹⁾の Horn 集合反証による数論的関数の計算可能性と反証作成過程における後戻りの関係についての考察に基づいている。閉質問および通常の開質問の場合、反証作成方式は効率に関係するだけであるが、前節で述べたように開質問において答えの大きさ n を制限したとき、とくに $n=1$ 、すなわち関数の求値問題のようなときは、それは結果そのものに影響を及ぼす。したがって、自動選択方式については、まずそれが関数の求値問題とどう関連するかを問題にしなければならない。松尾の結果をこの反証器に適用すると次のようになる。

まず, m 引数の数論的関数 f が Horn 集合 (正規集合) R によって計算可能であるということは, 任意の自然数 $a_1, \dots, a_m, b$ に対し, $f(a_1, \dots, a_m) = b$ であるならば, 開質問 $F[y] \rightarrow$ が存在し, この答えが $\{b\}$ であるときであるとする. このとき, 推論に使う正規節は R に属するかプロシジャに属する正基礎節である. また, ここで使うプロシジャは, サクセッサ (successor) 関数とプレデッサ (predecessor) 関数だけである. すると, 次のことが成立する.

- (I) 原始帰納的関数のクラスを P , 部分帰納的関数のクラスを R , Horn 集合によって計算可能なクラスのうち後戻りのない決定性手順でつくられる反証が存在する関数のクラスを Q とすると,

$$P \subseteq Q \subseteq R,$$

ここで, $\subseteq$ は左辺が右辺の真部分集合 (proper subset) であることを示す.

- (II) $R-Q$ に属する関数 f に対しては, 上記の定義のように, すべての反証において y に代入される定数が正しい関数値であることを要求するとき, その意味で “計算可能” であるためには, 反証器の推論規則は 2 章で示した導出原理に基づくものだけでは不足であり, 負節の変数に定数を代入する推論規則が必要である. さらに, f を “計算する” ための正規集合 R には, この解の一意性を保障するためにいくつかの正規節を付加する必要がある, これが反証を長いものにする.

すなわち, 自動選択方式は, たとえそれが効率的に最適なものであっても正規集合作成者に隠されているならば, 作成者は解の一意性のための工夫を必要とすることがあり, しかもそのために反証作成の効率が低下するのである. したがって, 選択方式は反証作成の効率がよく, しかも人手で演繹過程を模擬できる程度には簡単でなければならない.

いま, 推論規則の前提 2 となる正規節を選ぶ基本的戦略として,

- (i) 正節を優先的に選ぶ;
- (ii) 正節が前提となりえないときは, その推論結果に対して再び推論規則を適用するとき (i) が可能であるものを優先する;
- (iii) (i) と (ii) の選択ができないときは, 適当に定めた優先順位に基づいて選択することにする.

このとき, 上記の Horn 集合によって計算可能であることの定義において, “この答えが $\{b\}$ である” の部分を “その答えの大きさ n を $n=1$ に制限した場合の

答えが $\{b\}$ である” に置き換えたものを新しい計算可能性の定義とする. つまり, すべての反証においてつねに y に b が代入されることを要求する代りに, 最初に得られた反証において y に代入された値を関数値とするように変更する. すると, $R-Q$ のクラスの関数は Horn 集合によって計算可能になる. すなわち, (II) で述べた代入規則は不要である. また, これらの関数のための正規集合が簡潔に書けるため, 反証作成の効率が向上する.

(i) ~ (iii) は, 本質的に単一優先戦略 (unit preference strategy¹⁶⁾) であり, (ii) は正節を前提として選ぶために 1 段の先読み (look-ahead) を行ったことに相当する. したがって, (i) ~ (iii) は効率的にもすぐれていると考えられるため, Adbis の反証器の自動選択方式はこの戦略に基づいたものを採用した.

4. Horn 集合の形式

Adbis の反証器は, 2 種類の形式の Horn 集合を取り扱う. その一つは, 外部形式 (outer form) と呼ばれ, 反証器とその外界との間の通信のために使用される. もう一つの形式は, 内部形式 (inner form) と呼ばれ, 反証器がその表現の上で演繹を行う.

4.1 外部形式

外部形式は, 文字列で表現され, Horn 節は基本的には (2) で示したような Gentzen の *Sequentz*¹¹⁾ 形式をとる. アトムと項 (term) は, 第 1 階言語 (first-order language) で通常使われている形式で表現される. すなわち, アトムは述語記号の後に $<$ で囲まれた項の並びを置いた形式をとる. 項と項はカンマで区切られる. 項は, 変数または定数である. 関数記号をもつ項については, 現在, Adbis の反証器ではこれを扱うことができない. したがって, 関数はすべて述語の形式で表現しなければならない. 述語記号と変数

```
[ positive clauses ]
->orbit('Callisto',1882104,0.0075)
>radius('Ganymede',2608)
mass('Europa',4.7E22)

[ mixed clauses ]
mass(x,y),radius(x,z),f(y,z,u)->density(x,u)
mass(x,y)radius(x,z)f(y,z,u)>density(x,u)
mass(x,y)radius(x,z)f(y,z,u)density(x,u)

[ negative clauses ]
density('Io',x)->
density('Callisto',x)>
density('Ganymede',x)
radius(x,y),greater(y,z),radius('Europa',z)->
radius(x,y)greater(y,z)radius('Io',z)
```

図 3 Horn 節の例

Fig. 3 Horn clauses.

```

DEFINE uncle,parent;
GLOBAL father,mother,male;
parent(u,x),parent(u,v),parent(v,y),vmale(x)->uncle(x,y);
father(x,y)->parent(x,y);
mother(x,y)->parent(x,y);
father(x,y)->vmale(x);
male(x)->vmale(x);
END

```

図 4 Horn 集合の例
Fig. 4 A Horn set.

は、英字で始まる 8 文字以内の英数字列で表される。定数は、数値定数と文字定数とがある。前者は FORTRAN の算術定数の形式をとり、後者は文字列を引用符 (') で囲んだ形で表される。図 3 に Horn 節の例を示す。図 3 に示したようにアトムとアトムを区切るカンマは省略することができる。矢印は “ $\rightarrow$ ” または “ $>$ ” で表されるが、これも省略可能である。こうすると、見掛け上、正節、混合節、負節が同じ形になることがあるが、正規節と負節とでは反証器と外界との通信の時期が異なるため、これらが混同されることはない。

正規節は、反証器が内包的データを入力する状態のとき、正規集合の要素として反証器に入力される。正規集合は、次の 4.2 節で述べる内部形式に “翻訳” されるため、正規集合には通常のコンパイラ言語によるプログラムと同様にコンパイルとリンクのための記述が必要である。正規集合の例を図 4 に示す。図 4 の最初の DEFINE は、この正規集合によって定義される仮想関係が負節または他の正規集合から参照できることを宣言する。この例では、負節または他の正規集合に対して、 $\rightarrow\text{uncle}(x,y)$ と $\rightarrow\text{parent}(x,y)$ の二つの正節が存在することを宣言したことになっている。つぎの行の GLOBAL は、この正規集合で参照しなければならない正節が外延的データにあるか他の正規集合によって定義された内包的データにあることを、それら正節の右辺のアトムの述語記号でもって宣言する。この例では、father, mother, male を右辺のアトムの述語記号とする正基礎節の集合が外延的データにあることを仮定している。図 4 の最後の行の END は、一つの “翻訳” 単位の終わりを指示するために使われる。

負節は、反証器が質問を受け付ける状態のとき、2 章で述べたように質問として反証器に入力される。反証器は、図 3 に示した形式の負節を、たとえそれが変数を含んだものであっても、閉質問とみなす。開質問であることを反証器に指示するためには、メタ述語記号 \$A を使う。たとえば、開質問

```

uncle ('Tadanori', x)→
$A(x, 'NEPHEW')

```

において、\$A の引数 x は負節に現れる変数 x に対して答えを要求することを示す。この例では変数の指定は一意的であるが、負節に現れる変数が複数のとき、\$A にはそれらのすべてか一部を任意の順序に並べることがで

きる。すなわち、\$A によって答えとする変数とそれらの出力の順序を指定することができる。\$A の最後の引数は、答えを保存するファイル名である。上述の例では、NEPHEW という名前のファイルに、たぶん定数の並び 'Shigemori', ..., 'Atsumori', ... が記録されるであろう。答えをファイルに入れる必要がないときは、この引数を省けばよい。

メタ述語記号は、このほか演繹の打ち切り条件を指定するための \$C、演繹過程のトレースの出力を要求するための \$T などがある。

4.2 内部形式

内部形式は、導出演算を計算量および領域量の両面で効率的に行うための形式であり、演算を実行するために外部形式の Horn 集合はいったん内部形式に変換される。内部形式の Horn 節は、基本的には 2 バイトで符号化した述語記号、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。述語記号の引数の個数は、その符号の中に埋め込まれている。導出演算時には、データベースおよびプロシジャ以外のすべての Horn 節は、この形式で表される。

5. データベースシステムにおける反証器

本稿の Horn 集合反証器は、推論関係型データベース管理システム Adbis の中核的構成要素として開発したものである。Adbis は、オペレーティングシステム FACOM OSIV/F4 のもとで動作するシステムであり、九州大学大型計算機センターで 1981 年に公開された。その後、Adbis を使ってデータベースシステムが構築されたのに伴い、Adbis にいくつかの機能が追加された。図 5 に Adbis を使ったデータベースシステムの構成を示す。図 5 に示した HSI (Horn Set Interpreter) に反証器が内蔵されている。DMP (Data Manipulation Primitives) は、外延的データのための関係型データベース管理システムである。Adbis は、当初 HSI と DMP の二つのモジュールから構成されていた²⁾が、その後 Tsuno および Tsuno-Kakushi と名づけた端末利用者とのインタフェースを与えるモジ

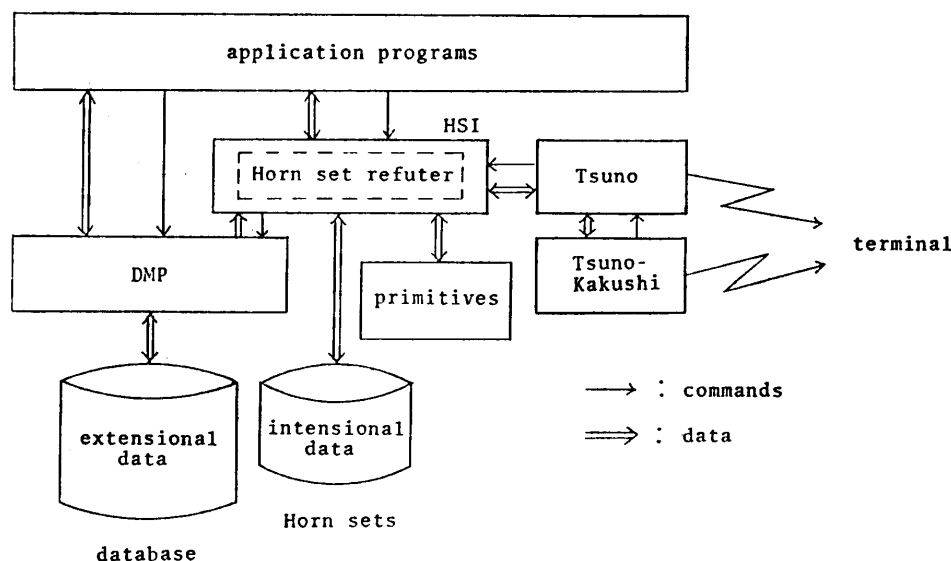


図 5 Adbis を使ったデータベースシステム
 Fig. 5 The database system using Adbis.

ルールが作られた。Tsuno は、別の見方をすれば、Horn 集合に基づく会話型データ操作言語である。その言語仕様は、核となる Horn 節の構文が 4.1 節で述べた形式をとるため、数理論理学の習熟者には簡潔であり、習熟者でなくてもいったん慣れてしまえば煩雑とは感じないようである。しかし、Tsuno はすべての利用者、とくにほかのデータベースシステムの検索に慣れた者にとって必ずしも使いやすいわけではない。そこでデータベースシステムで情報検索の機能の強い部分については、伝統的な情報検索システムのコマンドによって検索ができるようにした。これが Tsuno-Kakushi である。Tsuno および Tsuno-Kakushi の詳細については、ここでは割愛する。

現在、九州大学大型計算機センターでは Adbis を使った二つのデータベースシステムが利用者に公開されている。一つは、XDT (CRYStallographic DaTa system) であり、英国 Cambridge Crystallographic Data Centre⁷⁾ 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³⁾。XDT は 1982 年 11 月に公開された。もう一つのシステムは GENAS (GENe Analyzing System) であり、西独にある European Molecular Biology Laboratory の核酸塩基配列データ²³⁾を扱うものである⁴⁾。GENAS の公開は、1983 年 7 月であった。この二つのシステム構成は、図 5 に示したとおりである。応用プログラムについては、XDT は UNICS^{5), 6)} および PLUTO 78⁷⁾ を改造したものを、GENAS は Staden²⁰⁾ と Zuker²²⁾

のプログラムを改造したものを用いている。

これらのデータベースシステムにおいて複雑な処理は応用プログラムにゆだねられ、反証器はそれほど論理深度の大きい質問処理を行っていないこともあり、これまでこの反証器の方式についての問題点は顕在化していない。論理深度の大きい質問処理がそれほどない理由は、実験科学の分野では推論機構へのなじみがないことによる。データベースと推論機構を積極的に利用した研究方法は、今後の開拓に待つところが大きい。しかし、現状でも反証器の使用は効果がある。データベース定義を熟知した利用者であれば、Horn 集合を使ってデータベースのもつ情報を十分に活用した多様な質問を簡単に行えることなどは、反証器の効力を示すものである。また、2 章で述べたように任意のサブルーチン形式プログラムを反証器の基本演算とすることができることは、データベースシステムの構築において有用であった。すなわち、上述の PLUTO 78 のような既存プログラムはインタフェース部のわずかの改造だけで反証器の基本演算となり、Tsuno または Tsuno-Kakushi のコマンドによって呼び出すことができるようになった。

6. む す び

データベース管理システムに推論機構を付加することの利点の一つは、この機構がデータベースシステムに柔軟で強力な質問処理能力を与えることである。この意味では、本稿の Horn 集合反証器は学術研究用

データベース管理システム Adbis の推論機構として十分な役割を果たしていることが科学用データベースシステムの構築を通して実証された。しかし、これらデータベースシステムにおいて、反証器固有の推論能力はまだ十分に活用されているとはいえない。

この反証器は、推論に当たって前提となる Horn 節の選択を自動的に行うことに一つの特徴がある。選択方式は、数論的関数の計算可能性と反証作成過程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。この選択方式は、実用的にはまだ未解決の問題である。しかし、反証器が論理深度の大きい処理や非決定性の問題処理のために使われていないため、選択方式に対して問題提起を投げかける現象はこれまで発生していない。今後、データベースシステムにおいて推論機構を積極的に使った処理や知識工学的手法による処理を取り入れることによって、反証器の方式についての問題点を明らかにし、反証器をより洗練する必要がある。

参 考 文 献

- 1) 松尾文碩: Horn 集合の決定性反証によって計算可能な数論的関数, 情報処理学会論文誌, Vol. 25, No. 3, pp. 437-442 (1984).
- 2) 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, 情報処理学会論文誌, Vol. 24, No. 2, pp. 249-255 (1983).
- 3) 松尾文碩, 二村祥一, 高木利久, 河野重昭: Adbis による結晶構造データベースシステムの構築, 情報処理学会アドバンスト・データベース・システムシンポジウム論文集, pp. 39-48 (1982).
- 4) 松尾文碩, 榊 佳之, 久原 哲, 二村祥一, 高木利久, 篠原 武: 核酸塩基配列データベースシステム GENAS の構築, 情報処理学会第 27 回全国大会講演論文集, 3 K-8 (1983).
- 5) 桜井敏雄他: 結晶構造解析ユニバーサルプログラムシステム (第 4 報) UNICS II システム, 理化学研究所報告, Vol. 50, No. 3, pp. 75-91 (1974).
- 6) 桜井敏雄, 小林公子: 結晶構造解析ユニバーサルプログラムシステム (第 5 報) UNICS III システム, 理化学研究所報告, Vol. 55, No. 3, pp. 69-77 (1979).
- 7) Allen, F.H. et al.: The Cambridge Crystallographic Data Centre: Computer-Based Search, Retrieval, Analysis and Display of Information, *Acta Cryst.*, Vol. B-35, No. 10, pp. 2331-2339 (1979).
- 8) Andreka, H. and Nemeti, I.: The Generalized Completeness of Horn Predicate-Logic as a Programming Language, D.A.I. Res. Rep., No. 21, Dept. of Artificial Intelligence, Univ. of Edinburgh (1976).
- 9) Clark, K.L.: Negation as Failure, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, pp. 293-322, Plenum, New York (1978).
- 10) Futó, I., Parvas, F. and Szeredi, P.: The Application of PROLOG to the Development of QA and DBM Systems, in Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, pp. 347-376, Plenum, New York (1978).
- 11) Gentzen, G.: Untersuchungen über das Logische Schliessen, *Math. Z.*, Vol. 39, pp. 176-210, 405-431 (1935).
- 12) Henschen, L. and Wos, L.: Unit Refutations and Horn Sets, *J. ACM*, Vol. 21, No. 4, pp. 590-605 (1974).
- 13) Hill, R.: LUSH Resolution and Its Completeness, DCL memo 78, Dept. of Artificial Intelligence, Univ. of Edinburgh (1974).
- 14) Kowalski, R.: Predicate Logic as Programming Language, *Information Processing 74*, pp. 569-574, North-Holland, Amsterdam (1974).
- 15) Kuehner, D.: Some Special Purpose Resolution Systems, in Meltzer, B. and Michie, D. (eds.), *Machine Intelligence 7*, pp. 117-128, Edinburgh Univ. Press, Edinburgh (1972).
- 16) Loveland, D.W.: *Automated Theorem Proving: A Logical Basis*, p. 405, North-Holland, Amsterdam (1978).
- 17) McKinsey, J.C.C.: The Decision Problem for Some Classes of Sentences without Qualifiers, *J. Symbolic Logic*, Vol. 8, No. 3, pp. 61-76 (1943).
- 18) Šebelík, J. and Štěpánek, P.: Horn Clause Programs for Recursive Functions, in Clark, K.L. and Tärnlund, S.-Å. (eds.), *Logic Programming*, pp. 325-340, Academic Press, London (1982).
- 19) Shoenfield, J.R.: *Mathematical Logic*, p. 34, Addison-Wesley, Reading, Mass. (1968).
- 20) Staden, R.: Further Procedures for Sequence Analysis by Computer, *Nucleic Acids Res.*, Vol. 5, No. 3, pp. 1013-1015 (1978).
- 21) Tärnlund, S.-Å.: Horn Clause Computability, *BIT*, Vol. 17, No. 2, pp. 215-226 (1977).
- 22) Zuker, M. and Stiegler, P.: Optimal Computer Folding of Large RNA Sequences Using Thermodynamics and Auxiliary Information, *Nucleic Acids Res.*, Vol. 9, No. 1, pp. 133-148 (1981).
- 23) EMBL Nucleotide Sequence Data Library User Manual, European Molecular Biology Laboratory, Heidelberg (1982).

(昭和 58 年 8 月 15 日受付)

(昭和 58 年 11 月 15 日採録)