

## 対話型データ検索・処理システム SCLAMS†

相 沢 輝 昭\*\* 畑 田 のぶ子\*\*

定型の少量のデータをオンライン対話形式で迅速に検索することを主目的とした小規模データベース・システム SCLAMS を試作した。これはファイルよりも小さい S.L. (Scaled Lattice) と呼ぶ一種の多次元の表をデータのモデルとし、処理の基本単位としたところに特徴がある。システムは3モードから成り、蓄積モードでは与えられたファイル中のデータを S.L. の形に分解してシステムのデータベースに取り込む。検索モードでは必要なデータを S.L. 単位で引き出す。操作モードでは、検索モードで引き出された S.L. に対して、抽出条件を指定して更に細部のデータを抽出したり、数値データに対する簡単な計算処理を施したり等の各種操作を行う。これらは、そのために設計した非手続き型のデータベース照会言語 SML を通して行われる。

S.L. データ・モデルに基づく照会言語 SML はその一部に比較的容易に自然言語を混成させることができる。その1つの試みとして SML にカナ書き日本語を混成させる実験についても報告する。

### 1. ま え が き

本格的な商用データベース・システムが各方面で実現になりつつあるが、それらは基本的に、ある程度の規模にまとまったデータを前提としているように思われること、また、導入から稼動に至るまでに綿密な諸検討が必要とされていることなどのため、コンピュータ室周辺に散在する一切のデータをもれなく1つのデータベース・システムに吸収することは、なおしばらくはむづかしいように思われる。

SCLAMS (SCaled LAttice Manipulation System) は、こういった種々の理由で本格的データベース・システムに収容しきれなかったデータに対する簡易型データ検索・処理システムと言えるものである。ユーザから見たその特徴は、

(1) すべての処理は文字表示装置を介したオンライン対話型である。

(2) 既存の単発のファイル中の必要なデータを手軽にシステムに取り込み、その上で、それらデータの多角的検索ならびに各種計算処理等を可能にしている。

(3) 特に定型の少量のデータの検索が手軽かつ迅速。したがって、質問結果に応じて次の行動を決める試行錯誤的な使い方に向いている。

これら特徴を効果的に実現するために、われわれはデータ・モデルにまでさかのぼって検討し、ファイルよりも小さい Scaled Lattice (以下 S.L. と略記) と称する一種の多次元の表を処理の基本単位とした。その結果、SCLAMS はシステム構成上さらに次の特徴

を持つことになった。

(4) データの検索は、まず可能性のある S.L. を大まかに引き出し、ついでその中から目的のデータを取り出すという二段構えであり、前者についてはキーワード流の検索手法によりシステムに蓄積されている S.L. に不案内のユーザにも比較的楽に利用できる。

(5) S.L. はごく自然に数字でいう写像とも見せる。それを利用してデータベース照会言語を非手続き的人工言語として設計したが、さらにその一部に日本語を混成させ、人工言語と自然言語の長所をともに生かすことを試みた。

### 2. Scaled Lattice データ・モデル

論理的にまとまっていて、しかもできるだけ単純なデータの集りの基本単位として、われわれは図1に示すように、一定の対象(たとえば人口)を表わすデータを、いくつかの観点(たとえば年次別、県別、性別)に着目して分類整理したデータ群を取り上げた。

個々のデータは、観点に対応するものさし(scale)

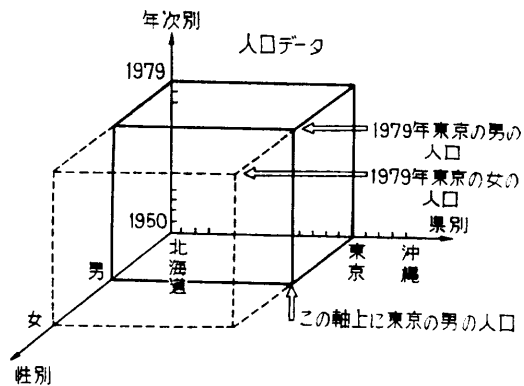


図1 Scaled Lattice のイメージ

Fig. 1 Graphical image of Scaled Lattice.

† SCLAMS: An Interactive Data Retrieval and Manipulation System by TERUAKI AIZAWA and NOBUKO HATADA (NHK Technical Research Laboratories).

\*\* NHK 総合技術研究所

の目盛り（たとえば北海道，青森，…）の交叉する各格子点 (lattice point) に配置されている形で，あるまとまりを持っているわけで，そのような感じをこめてこれらデータ群を以下 S. L. (Scaled Lattice) と呼ぶ。また各観点をスケールと呼ぶ。

直観的には S. L. は多次元の表と考えられる。これは使用上いくつかの利点を持っている。それを図1の例に即して説明しよう。

(1) 記憶容量および検索時間の節約

図1のデータを通常のファイル形式で蓄積するとすれば下図のようになる。

年次別	県別	性別	人口
1950	北海道	男	2,169
1950	北海道	女	2,126
1950	青森	男	636

S. L. データに特徴的なことは年次別，県別，性別スケールの各要素のすべての組合せが出現することである。

このような場合，まず各スケールの要素を一定の順序に並べて1回だけ記憶する。次に人口データを，各スケールの要素を添字とする3次元配列の形式に並べる。こうすることによって，まず記憶容量が節約できる。

さらにたとえば“1970年の東京の男”の人口データを検索するには，“1970年”，“東京”，“男”がそれぞれ対応するスケールの何番目の要素かを知るだけで事は足りる。同様に“女”の人口データだけをすべて検索することも容易である。これらは検索時間の短縮につながる。

(2) 表形式による結果表示が容易

S. L. データを人間に対して視覚的に表示する基本的な方法は，S. L. が実質的に多次元の表であることに着目して，特定の2つのスケールを2次元断面としてそれを通常の表形式で表示するというものであろう。実際に使用してみてこの方法がことのほか便利であることが分かった。1つには人間にとって表形式が想像以上に見やすく理解しやすいこと。今1つには，特定の2つのスケールを2次元断面として選択することは残りのスケールの要素を特定のものに固定すること，つまり一種の制約条件による検索を行っていることに等しい。言い換えれば，その種の検索が表示スケールの選択と

いう形できわめて手軽に実行できるわけで，これも S. L. モデルの利点の1つである。

逆に欠点としていわゆる空値 (null value) の多発があげられる。S. L. ではスケールの要素を組み合わせるまず格子という枠組を作り，その格子点にデータを配していくわけであるが，現実にはある格子点に対するデータが未収集であったり，その格子点に対応するスケールの要素の組合せが論理的に不可能であったりすることが起り得る。

このような場合，それらの格子点にも人工的な空値を与えてやる必要がある。これに対して，通常のファイル形式ではそれに該当するレコードを単純に省略すればよい。ごく大ざっぱな計算によれば<sup>1)</sup>， $n$ 個のスケールを持つ S. L. に対して，空値の割合  $k$  が  $n/(n+1)$  以上になると通常のファイル形式にくらべて却って記憶容量が多くなる。この  $n$  と  $k$  の数値例は次の通りであり，幸いなことに想像以上に大きい。

$n$	2	3	4	5	6
$k$	0.667	0.750	0.800	0.833	0.857

S. L. はまた自然に写像とも見なせる。それによって SCLAMS のデータベース照会言語が設計され，またいわゆる関係モデルとの関連性もはっきりする。これについては5. で述べる。

3. システムの主要構成

SCLAMS の主要構成と基本的なデータの流りは図2に示す通りである。

SCLAMS のデータベースは論理的には S. L. の集合である。主要機能として，外部データをデータベースに取り込む蓄積モード，データベースから所要のデータを引き出す検索モードと操作モードがある。

(1) 蓄積モード

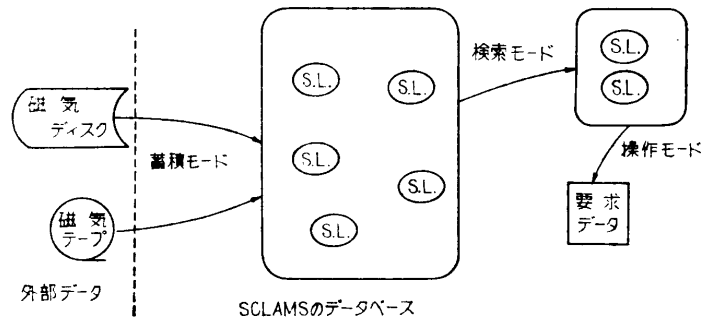


図2 SCLAMS の主要構成  
Fig. 2 Outline of SCLAMS.

計算機ファイルの形になっている外部データから必要なデータをデータベースに取り込むためのモード。具体的には、ファイルのレコード中の必要部分に対してのみ、その位置、データ型等の記述を与えることにより、その部分のデータを S.L. の形式に変換して取り込む。

なるべく広範な形式のファイルに対応できるようにしたため、このモードはシステム全体の大きな部分を占めることになった。

(2) 検索モード

S.L. の集合である SCLAMS のデータベースから必要なデータを S.L. 単位で引き出すためのモード。引き出すべき S.L. の指定法に 2 種類ある。

a) 蓄積モードへの入力である外部データ・ファイルごとに、そのファイルのレコード記述文(蓄積モードでユーザが与える)を手がかりとする。

b) S.L. は一定の対象(たとえば人口)のデータの集りであり、それを表わす用語(たとえば「ジッコウ」)が一種のキーワードとして蓄積モードでユーザにより付与される。蓄積されたすべての S.L. に関するそれら用語の KWIC 索引表(蓄積モードで自動的に作られる)を用い、キーワードによる文献検索の手法に習う。

特に方法 b) はシステムに蓄積されている S.L. を前もって知らないユーザにとって便利であろう。このようにキーワードによる文献検索流の手法が自然に応用できることも S.L. モデルの利点の 1 つと考えられる。

(3) 操作モード

検索モードで取り出された S.L. に対して、

- a) その内容を任意の 2 次元断面で見る、
- b) 抽出条件を指定してさらに細部のデータを引き出す、

c) 数値データに対する簡単な計算処理(集計、平均、分散、最大、最小など)を施す、等の各種操作を行うためのモード。

実際の使用に当っては a) で事足りることが多いので、それについてはライトペン等による簡単な画面操作ですませられるようにし、その他については、そのために設計した非手続き型データベース照会言語 SML (Scaled lattice Manipulation Language) によることにした。前者については 4.2 で、後者については

5. で説明する。

4. データの蓄積と検索

ここでは SCLAMS の蓄積モードと検索モードの全体、および操作モードのごく一部を、ユーザの使用手順に即した形で例を用いて説明する。操作モードの主要部であるデータベース照会言語 SML を用いた各種のデータ操作については 5. で述べる。

4.1 データの蓄積

図 3 のようなファイルを考える。これは電気メーカの支店別・製品別売上状況ファイルである。

一般にファイルのレコードは図 3 の下部に示すように 3 つの部分から成り立っていると考える。すなわち、1) レコードを一意的に識別するキーの部分、2) キーに対する繰返しのないデータを記述したフィールド形式の部分、3) 繰返しのあるデータを記述したテーブル形式の部分である。これらの各部分から S.L. に関するデータが次のように取り込まれる。

- キーの部分 ⇒ スケール
- フィールド形式の部分 ⇒ S.L. データ
- テーブル形式の部分 ⇒ スケールと S.L. データ

図 3 のファイルのデータの場合、以下の 3 つの S.L. として取り込むのが最も自然と考えられる。

- SL1: 販売員数 (支店別, 製品別)
- SL2: 年間目標売上高 (支店別, 製品別)
- SL3: 売上高 (支店別, 製品別, 月別)

これらとその出所との関連は次の通りである。

- キーの部分 ⇒ 支店別スケール, 製品別スケール
- フィールド形式の部分 ⇒ SL1, SL2
- テーブル形式の部分 ⇒ 月別スケール, SL3

最後のテーブル形式の部分からのデータの取り込み

支店	製品	販売員数	年間目標 売上高	月別売上高												
				1	2	3	4	5	6	7	8	9	10	11	12	
東京	テレビ															
東京	ラジオ															
東京	クーラー															
大阪	テレビ															
大阪	ラジオ															

図 3 ファイルの基本形式

Fig. 3 Fundamental form of a file.

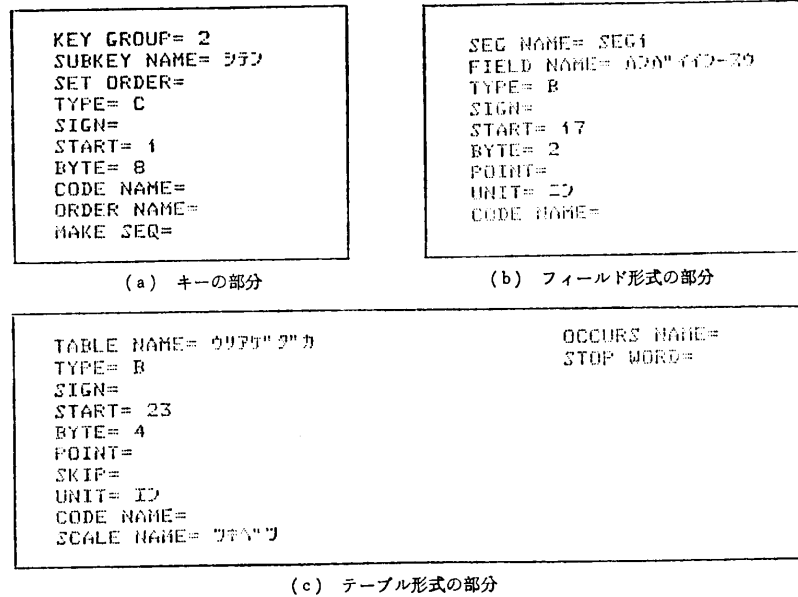


図 4 ファイルのレコードの記述法

Fig. 4 An example of description of file records.

は SCLAMS 独特の部分で、レコード内の繰返し部分を新たな次元と見てそのスケールを作成すると共に、1 次元ふえた形の S.L. データが取り込まれる。

このような取り込みをするのに必要なパラメータの与え方、つまりファイルのレコードの記述法を図 4 に示す。いわゆるキーワード方式で、ユーザは等号の右辺に必要なパラメータを指定すればよい。無指定のものについてはシステムが適当な値を補う。パラメータはキー、フィールド形式、テーブル形式の各部分に分けて与える。図 4 (a) (b) における主要パラメータはデータの位置とタイプである。

蓄積モードはこのほかに次のような機能を持っている。

(1) 大容量ファイル等に対して、特定のフィールドの値が特定の条件を満たすレコードのみを抽出した新しい小容量ファイルを作成し、データの取り込み等の後続操作をすべてその小容量ファイルに対して行う。

(2) キーの部分のないファイル、あるいは、あってもそれから作ったスケールの組合せでは S.L. データに空値が多発する恐れのあるファイルに対して、レコード番号を要素とする人工的なスケールを発生させ、それを唯一のスケールとする 1 次元の S.L. としデータを取り込む。

(3) 取り込みに際して、データの表現形式を（たとえば数値コードから文字列に）変換する。

蓄積モードで行われる今 1 つの重要なことに、後の検索モードで使われる KWIC 索引表の作成がある。これは S.L. のデータの対象を表わす用語（具体的には、たとえば上記のフィールド形式の部分から取り込まれる S.L. の場合は、「ハンバイインースウ」「ネンカンモクヒョウウリアゲダカ」等のフィールド名) に関する KWIC 索引表である。作成に当たっては、助詞のほかに「数、別、密度、増加率、…」等の計 79 語を見出しの対象としないいわゆるストップワードとした。

#### 4.2 データの検索

最初に、検索モードによる S.L. 単位の検索を述べる。これには前述のように次のいずれかの方法による。

(1) ファイルごとの S.L. のリストを利用する

図 3 のファイルを前節の方法でシステムに取り込んだ場合を例に取ろう。取り込みの際につけたファイルの名前（“FILE 1” とする）を入力すると、このファイルから取り込まれた S.L. のリストが図 5 のように表示される。これらのうちの必要なものに対応する ‘?’ マークをライトペンで指示すれば、それらがデータベースから作業域に検索されて来る。

この方法は当該ファイルの内容に詳しい専用ユーザ向きと言えよう。

(2) KWIC 索引表を利用する

たとえば人口に関するデータが欲しい場合 “ジンコウ” と入力すると、KWIC 索引の中の “ジンコウ” の

```

FILE NAME=FILE1
KEY=(1)000 (2)000
***"1"ノ"カカ"ヒヨウモノヲシヨウサイ***
*(3)KEY="1"ノ"カカ"FILE1
? *SEG1
? *SEG1
? *SEG1
? *SEG1

```

図 5 FILE1 (図3のファイル) に関連する S.L. のリスト

Fig. 5 List of S.L.'s related to FILE 1 (in Fig. 3).

```

***"1"ノ"キ-ワ-ト"ノ"カカ"ヒヨウモノヲシヨウサイ(200ノイ)***
(アタノ"キ-ワ-ト"ノ"カカ"ヒヨウモノヲシヨウサイ)
*****
? 1 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 2 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 3 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 4 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 5 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 6 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 7 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 8 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 9 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 10 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 11 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 12 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 13 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 14 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。
? 15 *0100 リイ"イ"ヒヨウ"カカ" 5"1000 カ"。

```

図 6 KWIC 索引表の例

Fig. 6 An example of KWIC index table.

```

F1 : コロヒヨウ"イ"ヒヨウ"カカ"
S1 : "1"ノ"カカ" 6/ 1:1947 1950
S2 : "1"ノ"カカ" 2/ 1:100 100
S3 : "1"ノ"カカ" 47/ 1:100 100
F2 : "1"ノ"カカ"ヒヨウ"カカ"
S4 : "1"ノ"カカ" 22/ 1:1952 1953
S5 : "1"ノ"カカ" 2/ 1:100 100
S6 : "1"ノ"カカ" 47/ 1:100 100
LIST

```

図 7 最終的に検索された S.L. のリスト

Fig. 7 List of retrieved S.L.'s.

```

SCLAMS MODE=M PH=3 DISPLAY OF RESULTS (VTYPE=300,LISTNO=01/02) 00.04.03/14:49
コロヒヨウ"イ"ヒヨウ"カカ"
| "1"ノ"カカ" 6)
- "1"ノ"カカ" 2)
"1"ノ"カカ" 47) 1 "1"ノ"カカ"
1947 1,934 1,919
1950 2,169 2,126
1955 2,429 2,344
1960 2,545 2,494
1965 2,583 2,589
1970 2,554 2,631
1=NEXT 2=BACK 3=ANSN 4=SHL 7=TATE 8=YOKO 9=DANMEN 10=RESET 11=MODE 12=INIT

```

図 8 北海道の人口

Fig. 8 Display of the population data in Hokkaido.

近辺が図6のように表示される。この中から欲しい S.L. を(むしろ広範囲に)ライトペンで指定すると、それら S.L. のみがスケールと共に表示される。この段階で必要なら更に S.L. を絞ることができる。最終的に欲しい S.L. をやはりライトペンで指定すると、それらがデータベースから作業域に検索されて来る。この方法は、システム内の蓄積データについてあま

り詳しくないユーザや、一般に蓄積済の S.L. が多くなった場合などに有効であろう。

以上(1),(2)のいずれかの方法で検索されて来た S.L. は、そのリストが図7のように表示される。図中、F1,F2,S1,...,S4 は S.L. やスケールに付けられた略号である。ここまでの検索モードの役目である。

以下は操作モードとなり、これら S.L. の内容を見

るには図7の最下行の“LIST”の次に、見たいS.L.の略号(およびセミコロン)を入力してやればよい。たとえば“F1;”と入力すればF1(=コクセイチョウサジンコウ)に関するデータが図8のように表示される。また“S3;”と入力すればS3(=トドウフケンベツ)スケールの要素が表示される。

S.L.の表示断面の切り換えや、画面スペースの関係から断面の一部分しか表示されていない場合の別の部分の表示等は、ライトペンまたはファンクション・キーのワンタッチ操作で実行できる。

## 5. 照会言語 SML によるデータの操作

### 5.1 写像としての S.L.

SCLAMSの通常の使用に当っては4.で述べたことでほぼ事足りるが、もう少しきめ細かにS.L.データを取り出したり、更にはデータに対する簡単な計算を施したりしたくることがある。そのために用意したのがデータベース照会言語SML(Scaled lattice Manipulation Language)である。実は4.2で特定のS.L.の内容を表示するのに用いた

```
LIST F1;
```

という書き方は、SMLの最も簡単な使用例であった。このSMLは、S.L.を写像あるいは関数と見て“集合と写像”を記述する数学流の記法をほぼそのまま転用したものである。

一般に、スケール $S_1, \dots, S_n$ を持つS.L.をFとする。このときFは自然に次のような写像と考えられる。

$$F: S_1 \times \dots \times S_n \rightarrow A,$$

ここでAはFのデータを表現するに足る十分な要素を持った集合である。

このように考えたとき、S.L.は、関係データベースで言ういわゆる第3正規形にある関係を関数従属性ごとに分解したもの、と見ることができよう<sup>2)</sup>。ただし重要な差異があつて、それは第3正規形における関数従属性に対応する関数の定義域(すなわち主キーの集合)が一般にある直積集合の部分集合であるのに対して、写像としてのS.L.の定義域は常に直積集合そのものであることである。定義域を直積集合に取ることの得失はまさに2.で述べたことである。

S.L.モデルと一見似ているデータ・モデルにCO-DASYL開発委員会言語構造グループの提唱したいわゆる情報代数における属性空間モデルがある<sup>3)</sup>。しかしそれは定義域と直域を区別しないで属性空間

$$S_1 \times \dots \times S_n \times A$$

を作り、データFはその部分集合と考えるという点で、S.L.モデルとは異なっている。

### 5.2 SML の使用例とシンタックスの概略

いま以下の2つのS.L.が検索されているとして、SMLの代表的な使用例を示そう。

$$F1: S1 \times S2 \times S3 \rightarrow \text{人口値の集合},$$

$$F2: S1 \times S2 \rightarrow \text{テレビ契約数の集合},$$

ここで

$$S1 = \text{年次別} = \{1950, 1951, \dots, 1979\},$$

$$S2 = \text{県別} = \{\text{ホッカイド}, \dots, \text{オキナワ}\},$$

$$S3 = \text{性別} = \{\text{オトコ}, \text{オンナ}\}.$$

【例1】1975年の東京の男の人口(S.L.の個々のデータの抽出)。

```
LIST A;
```

$$A = F1(1975, \text{トウキョウ}, \text{オトコ});$$

全県のデータを出したいときは $A = F1(1975, \text{, オトコ})$ 、また全データを出したいときは $A = F1$ と書けばよい。

【例2】1975年の男の人口が百万人以上の県(S.L.のデータに応じてスケールの要素を抽出する)。

```
LIST KEN;
```

$$\text{KEN} = \langle X: F1(1975, X, \text{オトコ}) \rangle = 100 \text{ マン};$$

【例3】1960年代の大阪の女性人口の平均値(簡単な計算処理)。

```
LIST A;
```

$$A = \text{MEAN}(B);$$

$$B = F1(\text{NEN}, \text{オオサカ}, \text{オンナ});$$

$$\text{NEN} = \langle 1960, 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969 \rangle;$$

1960年代の年次を表わす集合NENを簡単に

$$\text{NEN} = S1.11-20;$$

と指定することもできる。これは年次別スケールS1の11~20番目の要素の意。簡単な計算処理として可能なものは今のところ次の6種で、順に、個数、最大値、最小値、合計、平均値、分散を表わす。

COUNT, MAX, MIN, SUM, MEAN, VAR.

【例4】1975年の女の人口が百万人以上の県の、1970年のテレビ契約数(2つのS.L.の結合処理)。

```
LIST A;
```

$$A = F2(1970, \text{KEN});$$

$$\text{KEN} = \langle X: F1(1975, X, \text{オンナ}) \rangle = 100 \text{ マン};$$

[例5] 1975年の人口が百万人以上、かつ同年のテレビ契約数が20万以上の県(集合演算).

LIST A;

A=B & C;

B=<X: F 1 (1975, X, オンナ)  
>=100 マン>;

C=<Y: F 2 (1975, Y)>=20 マン>;

集合演算として可能なものは & (交わり), | (結び), - (差) であるが, 処理の便宜上, 演算対象を同一の S.L. の値域の部分集合, もしくは同一のスケールの部分集合に制限している.

上例からも分かるように SML によるリクエスト文の記述の一般形は図9の形をしている. ここで  $a_1, \dots, a_m$  は  $b_1, \dots, b_n$  のうちのどれかであり, また  $b_1, \dots, b_n$  の並べ方は任意である.

Expression の型は基本的に次の6種に分類できる.

- 1) 数字または文字リテラル
- 2) 集合関数式 例: COUNT( $x$ )
- 3) S.L. 式 例: F( $x_1, \dots, x_n$ )
- 4) 集合演算式 例:  $x$  &  $y$
- 5) 集合の定義式

要素を列挙するものと, 条件によるものがある. 前者の列としては

<3, 5, 7, 4>, <トウキョウ, ナゴヤ>

等. 後者の一般形は

< $x_i: q_1, \dots, q_k(F(x_1, \dots, x_i, \dots, x_r)\theta y)$ >

である. ここで  $q_i$  は論理限定記号で, E'X#B あるいは A'Y#C の形をしている. これらはそれぞれ  $\exists X \in B$  あるいは  $\forall Y \in C$  の意味を持つ,  $\theta$  は比較詞で, >, =, <=等のいずれか. なお  $k=0$  の場合も許す. その場合 F の外側のカッコは省略できる (5.2 の諸例参照).

- 6) スケールまたはその要素の略式表現

例: S または S.i または S.i-j

これらをそれぞれ S の全要素, S の第 i 要素, S の第 i~第 j 要素を表わす.

### 5.3 SML への日本語の混成

LIST  $a_1, \dots, a_m$ ;

$b_1 = \text{expressin } 1$ ;

$b_2 = \text{expressin } 2$ ;

⋮

$b_n = \text{expressin } n$ ;

図9 SML による記述の一般形

Fig. 9 General form of SML expression.

本章でこれまでに示したように SML はそれなりに強力かつ柔軟な表現能力を持っているが, その独特の記号的性格がなじみにくいことも確かであろう. そこで SML の一部に日本語を混成させることを試みた. 自然言語によるデータベースへのアクセスの試みは英語によるものがいくつかあり<sup>4)</sup>, 日本語ではヤチマタ<sup>5)</sup>の例がある. これらの試みに対して, われわれの方法の特徴は, 照会言語全体を日本語化するのではなく, SML の変数の定義文のみを日本語化したこと, 更にその日本語表現中に SML の変数を埋め込めるようにして, いわば自然言語と人工言語の混成型としたことである.

例として5.2にあげた S.L. に対する次のようなリクエストを考える.

[例6] 1975年の男の人口が, 1965年の東京の女の人口以上の県とその個数.

このリクエストは従来の SML により次のように表現されよう (これを仮に I 型と呼ぶ).

LIST A, B;

A=<X: F 1 (1975, X, オトコ)>=C>;

B=COUNT (A);

C=F 1 (1965, トウキョウ, オンナ);

これに対して今回の試みは次のような II 型表現をも許すようにすることである.

LIST A, B;

A='1975' ノ 'オトコ' ノ ジンコウガ C イジョ  
ウノケン;

B=A ノ コスウ;

C=1965 ノ トウキョウ ノ オンナ ノ  
ジンコウ;

このような II 型表現をも処理できるようにするために, II 型表現をそれに対応する I 型表現に変換する一種の翻訳ルーチンを作成し, SCLAMS に追加した. その詳細は文献<sup>6), 7)</sup>に譲るが, 主な特徴は次の2点である.

- (1) 日本語の辞書と文法がコンパクト

辞書は約100語程度, 文法規則の数も適用条件付きの文脈自由型プロダクションで15程度である. これらがこのように小さくなったのは, 日本語表現を SML の等号の右辺にのみ許したことによって, 可能な表現の範囲が自ら小さくなったことが1つの理由である. もう1つの理由は, 辞書を大きくする可能性のある「1975」, 「トウキョウ」, 「オトコ」等のスケールの要素を辞書から除き, '印で囲んだり文脈から判断させ

るようにしたことである。

(2) カナ書き日本語の分かち書きの仕方がほぼ自由原則として常識的な分かち書きならどのようなものでもよいようにした。ただし、辞書から除いた上述のスケールの要素がまぎらわしい場合に限って、その前後をスペースまたは▽印で囲む。例6のII型表現の場合、変数Aの右辺はほぼベタ書き、またCの右辺は完全な分かち書きである。このことは基本的に left-to-right parsing によって可能になっている。

## 6. 実働化と使用実験

SCLAMS の試作システムは IBM の S/370 モデル 138 に対して実働化した。対話用端末には IBM 3278 文字表示装置および IBM 3287 プリンタを用いた。

プログラムは PL/I (F コンパイラ) とアセンブラで書き、そのステップ数は PL/I 部分が蓄積モードと検索モードで各約2千、操作モードで約3千である。後者には混成された日本語を SML に翻訳するルーチン約5百ステップも含む(これには辞書や文法規則も含まれている)。アセンブラは主として文字表示装置とのインタフェース、蓄積モードでの汎用ファイル・ハンドラに用い、そのステップ数はそれぞれ約2千と4千である。

プログラムが予想外に大きくなったのは考えられる機能を可能な限り盛り込もうとしたため、実際の使用に当っては2.の(2)で述べた基本機能(特定の2次元で表を見ること)ですむことが圧倒的に多い。SML をこれだけの機能に限定し、またデータ取り込みに用いるファイルのフォーマットを標準化すれば、かなり有効なデータ検索システムがマイクロ・コンピュータでも実現できると思われる。

SCLAMS は現在 512 k バイトのメモリ領域で動いており、そのうち約半分は作業域である。処理の高速化のために操作モードで用いる S.L. の全データは一旦この作業域に移してから使用している。それを外部メモリに回せば必要な主メモリ容量は大幅に減ることになる。

これまでに3~5次元の S.L. を2百以上取り込んだ。各 S.L. の持つデータ数は2~5千(空値データの割合は0.5~0.8)で、取り込みに要する時間はほぼ60ms/1データ(CPU時間)、したがって1つの S.L. はほぼ2~5分で取り込める。当初目標にした操作モードでの処理の高速化は一応満足されており、上記程度の S.L. に対する各種応答時間はほぼ瞬時で

ある。また SML に日本語を混成させた場合、その翻訳に要する時間はおおむね1秒以内(経過時間)である。

## 7. むすび

Scaled Lattice と呼ぶ一種の多次元の表を処理の基本単位とした簡易型データ検索・処理システム SCLAMS を試作したのでその概要を報告した。

データの検索や処理に主眼をおいたため、システムに取り込まれたデータの更新機能は今回の試作システムには加えなかった(したがって、現在は、その必要の生じた時にはもとのファイルを何らかの方法で更新した上で改めて取り込みをやり直すことになる)。

Scaled Lattice データ・モデルとそれに基づくデータベース照会言語 SML の効用の1つは比較的容易に自然言語を混成させられることであろう。データベース照会言語の今後のあり方として、SCLAMS でわれわれが試みたような、人工言語と自然言語との混成型も有望と思われる。

試作システムはかなり大規模の実データに対して試作中であり、将来のレベルアップに対する有用なデータも得られつつある。

終りに、本研究に当って終始適切なご助言ご支援を頂いた鈴木前総合技術研究所次長ならびに町田情報処理研究部長、また実働化に当って種々ご助言を頂いた沓沢主任研究員に感謝する。

## 参 考 文 献

- 1) 相沢(編): SCLAMS 概説, NHK 総技研(1980).
- 2) 植村: データベースシステムの基礎, オーム社(1979).
- 3) 星(訳): 情報代数, bit 臨時増刊(1978-10).
- 4) たとえば Hendrix, G. 他: Developing a natural language interface to complex data, ACM Trans. on Database Systems, Vol. 3, No. 2, pp. 105-147 (1978).
- 5) 藤崎他: データベース照会システム「ヤチマタ」と名詞句データ模型, 情報処理, 20 巻1号, pp. 77-84 (1979).
- 6) 畑田他: 日本語を混成させたデータベース照会言語とその使用実験, 情報処理学会第21回全国大会, pp. 1017-1018 (1980).
- 7) Aizawa, T. et al.: Using a natural-artificial hybrid language for database access, Proc. of the 8th Int'l Conf. on Computational Linguistics, pp. 543-549 (1980).

(昭和55年10月27日受付)

(昭和56年1月22日採録)