オンラインシソーラスを用いる知的情報検索システム

酒井, 浩
九州工業大学情報科学センター

張, 裕民
武漢大学

有川, 節夫
九州大学大学院総合理工学研究科情報システム学専攻

https://doi.org/10.15017/17713

出版情報：九州大学大学院総合理工学報告. 10 (1), pp.59-66, 1988-06-30. 九州大学大学院総合理工学研究科
バージョン：
権利関係：

オンラインシソーラスを用いる知的情報検索システム

酒井　浩*・張　裕民**・有川節夫***

(昭和62年2月29日　受理)

An Intellectual Information Retrieval System Based upon Online Thesaurus

Hiroshi SAKAI, Yu-Min ZHANG and Setuo ARIKAWA

Theoretical and practical issues about an intellectual information system MIR-RF (Multi-stage Information Retrieval System Based upon Researcher Files), which was realized by Fortran in 1975, are discussed. One natural way to acquire knowledge is theoretically reformalized based on the logic program, in this case thesauri play an important role. According to the formalization, the representation and description of knowledge (or informations) are more simplified than those of former system, and a new MIR-RF system can easily be realized by Prolog.

1. はじめに

情報の検索に段階という動的概念を導入し知的な検索を可能にしようとする試みは有川，北川らにおいて定式化され，(研究者向け) 文献情報検索システムMIR-RF[1] が実現している．MIR-RF には，検索は各個人の知識と意図に反映してなされるべきであるという基本的な考えがある．システム利用者の検索意図は質問によってのみ表現されるので，質問を構成する適切なキーワードをどのようにし獲得し検索に利用するかということは MIR-RF の重要な問題であり，広く知識の獲得と関連した問題である．具体的には各個人がそれぞれのシソーラスを独自で作成管理し，それらを利用したキーワードの獲得を考える．

上記の特徴をもつシステムの記述言語は Fortran である．しかし，MIR-RF 作成時に Prolog に類似するプログラム言語があれば，自然にその言語が記述言語になったと考えられる．Prolog はそれ自身にデータベースの機能（特に検索の機能）を有するプログラム言語である．従って，本システムの記述には Fortran より Prolog の方が適していると考えられる．

*九州工業大学情報科学センター
**武漢大学
***情報システム学専攻

以上のような状況から，改めて Prolog 版 MIR-RF を作成した．検索システムは Prolog の得意とする単一化をそのまま利用でき，非常に簡単なプログラムで実現できた．本論文では，前半においてシステムの概略と検索に利用するシソーラスの知識表現，蓄積と管理等について論議する．さらに，その知識を利用する知的検索方式について論議する．後半において，厳密な定式化と実際の論理プログラムを示す．

2. 検索システムの概略と問題点

1975年 Fortran により実現された研究者向きの情報システムである MIR-RF は，文献情報の管理を目的とする情報システムである．システムは，主に知識の管理部，文献情報に対する自動索引部と索引の転置ファイルを用いる検索部により構成される．質問は，キーワードや著者名を指定することにより作成される．本論文の表題に示された知的検索は，キーワードを用いる主題の検索方式に対するものである．その方式は，一連の検索過程をシュミレートした厳密な理論にもとづく検索方式であり，検索過程で新たにキーワードを見い出し，それを用いてさらに次の検索を行うものである．

次に，検索システムの概略を示す．各文献情報は，表題，著者と出版社等の項目からなる文書情報である．

さらに，各文献情報には一意に自然数が対応しているとし，これを文献番号という．また，表題にはその文献の主題を表現するキーワードが含まれると考える．キーワードは，MIR-RF を構成する中の索引付けシステムにより自動的に表題から抽出される．さらに，検索の高速化のために，旧 MIR-RF ではキーワードから文献番号への対応関係を示す転置ファイルが自動的に作成される．検索は，このような文献情報の集合から目的に応じて一部の文献情報を取りだすことである．

このような検索システムの利用を考えるとき著者名や出版社名等の情報が明確であれば，質問の作成に問題はないが，主題についての検索の場合にはキーワードを抽出する検索付けシステムの特徴が間接的に検索結果に影響を与える．また，研究者の有するキーワードについての知識も検索結果に強く関係する．適切なキーワードがわからなければ，質問が書けないことにもなる．実際の検索は主題に関する検索が主であり，適切なキーワードを何らかの手段で獲得し，そのキーワードを質問の作成に利用することが望まれる．これは知識の獲得と関連する問題である．

MIR-RF では，この種の問題に対して次の 2 つ解決法を与えている．

1）研究者独自のシソーラス作成を支援する機能をシステムに付け，知識の蓄積が容易にできる環境の整備を行う．これは一種の知識ベース管理機能である．その知識ベースを以後研究者ファイル（Researcher File）とよぶ．

2）1つのキーワードから関連するキーワードを獲得し，それらを利用する検索方式を実現する．

研究者ファイルや知識の表現については 3 節，検索方式については 4 節以降に詳しく論議する．

3. MIR-RF の研究者ファイル

本節では，研究者ファイルにおける知識の表現や蓄積等について論議する．

3.1 知識の表現

現在 MIR-RF で対象とする知識は，キーワードが有する意味上の階層関係$\succcurlyeq$と同義関係$\doteqdot$である．

is_a 関係や part_of 関係等が階層関係$\succcurlyeq$の例である．異なる 2 つのキーワードに対する関係で，

キーワード 1 $\succcurlyeq$ キーワード 2

の記述はキーワード 1 がキーワード 2 の上位概念のキーワードであることを示すものとする．例として，

database $\succcurlyeq$ retrieval,
database $\succcurlyeq$ database management,
database $\succcurlyeq$ information storage,
retrieval $\succcurlyeq$ sequential pattern matching,
propositional logic $\succcurlyeq$ modal logic,
theorem prover $\succcurlyeq$ refutation $\succcurlyeq$ unification,

等が考えられる．

また，同義関係$\doteqdot$も異なる 2 つのキーワードに対する関係であり，

キーワード 1 $\doteqdot$ キーワード 2

の記述は，キーワード 1 とキーワード 2 が同義であることを示す．例は，

database management $\doteqdot$ DBMS,
retrieval $\doteqdot$ information retrieval $\doteqdot$ IR.

関係$\succcurlyeq$，$\doteqdot$は，いずれも推移法則が成立する．つまり，異なるキーワードw_1, w_2, w_3が，

$w_1 \succcurlyeq w_2$ かつ $w_2 \succcurlyeq w_3$ ならば，$w_1 \succcurlyeq w_3$,
$w_1 \doteqdot w_2$ かつ（$w_2 \doteqdot w_3$ または $w_3 \doteqdot w_2$）
ならば，$w_1 \doteqdot w_3$

となる．それぞれの推移関係を利用して，1 つのキーワードから新たなキーワードを見いだすことが，キーワード獲得の基本方式である．

次に，システム内における 2 つの関係$\succcurlyeq$と$\doteqdot$の表現形式を示す．旧システムにおいて木構造をした階層関係は，

［コード番号：キーワード］

というレコードの集合で表現された．ここで，コード番号は次下の条件を満足するように付加される．U を有限集合としその要素を記号とする．任意の記号を 1 個以上並べたものを記号列とよび，並べた個数を記号列の長さとし，$|\cdot|$とかく．記号列αとβに対して$\alpha\gamma=\beta$なる記号列γが存在するとき，αはβより上位であるという．任意の 2 つのキーワードを w_1, w_2 とし，そのコード番号をα_1，α_2とする．このときコード番号が満足する条件は，次の 3 条件である．

1）$w_1 \doteqdot w_2$ ならば，$\alpha_1 = \alpha_2$である．

2）$w_1 \succcurlyeq w_2$（$w_2 \succcurlyeq w_1$）ならば，α_1はα_2より（α_2はα_1より）上位である．

3）各記号列の長さは最小である．つまり，任意のコード番号αとβが$\alpha\gamma=\beta$ならば，αとβの間には（$|\gamma|-1$）個のコード番号があり，それぞれのコード番号には異なるキーワードが対応している．上記の例をこの形式で表現すると，

[1：database],
[11：retrieval],
[11：information retrieval],
[111：sequential pattern matching],
[12：database managemet],
[13：information storage],
となる.

さらに，旧システムにおいては，検索の高速化のために，
[キーワード：コード番号]
形式のレコードの集合をキーワードについてソートした転置ファイルが必要であった.

次に，論理プログラムによるシステム実現を前提として，関係$\succcurlyeq$と$\doteqdot$の表現を考える．そのために，関係$\succcurlyeq$と$\doteqdot$に対応する述語 r と eq を導入し，キーワードを各述語の引数に埋め込む．これにより，研究者ファイルを論理プログラムで扱える形式に変換できる．この表現形式ではコード番号が不要である．例えば，

r(theorem_prover, refutation).
r(refutation, unification).
eq(retrieval, information_retrieval).

と表現する．このとき，キーワードが2単語以上から構成されるならば，単語間に無名変数記号 '_' を補い，キーワードを1つの項（ターム）とする必要はある.

論理プログラムではルール形式の知識も自然に扱える．例えば，下位概念へのキーワードの推移関係
$w_1 \succcurlyeq w_2 \succcurlyeq w_3$ ならば，$w_1 \succcurlyeq w_3$,
を用いて w_1 から w_3 を見いだすための論理プログラムは述語 r* を用いて，

$r^*(X, Y)$: -$r(X, Y)$.
$r^*(X, Y)$: -$r(X, Z)$, $r^*(Z, Y)$.

と定義できる．また，同義語の推移関係
$w_1 \doteqdot w_2$ かつ $w_2 \doteqdot w_3$ ならば，$w_1 \doteqdot w_3$,
については，

$eq^*(X, X)$.
$eq^*(X, Y)$: -$eq(X, Y)$.
$eq^*(X, Y)$: -$eq(Y, X)$.
$eq^*(X, Y)$: -$eq(X, Z)$, $eq^*(Z, Y)$.
$eq^*(X, Y)$: -$eq(Z, X)$, $eq^*(Z, Y)$.

なる論理プログラムで実現できる．しかしこのままでは無限ループに入るので，同義語の推移関係を論理プログラムで記述するには assert や retract 等の述語を用いた別の制御が必要になる．つまり，過去の経過を unique という述語，

unique(X): -not(before(X)),
　　assert(before(X)).

によって残しながら推移させる必要がある．また，新たな検索を開始する際には，set という述語，

set: -retract(before(X)), fail.
set.

によって過去の経過の初期化をする必要がある.

3.2 知識の蓄積

MIR-RF では，研究者は個人の知識としてのシソーラスを利用者自ら作成する．つまり，自己の知識や見解等を検索に反映できる分だけ研究者にも次の2つの仕事が要求されることになる.

1）関心のある文献については，その文献情報のリストをシステムに追加する．これが，検索の対象となる文献情報の集合である.

2）研究者は，文献を熟読した際には重要なキーワードの集合を記録しておく．これを研究者メモとよび，メモの集合から研究者ファイルが構成される.

1）は，研究者が他の研究者による文献を不断に読むことを前提としている．2）は研究者ファイルが一度にできるのではなく，普段の研究動向の整理によって自然に形成されていくものであるという思想による.

3.3 知識の編集と操作

ここでは，研究者メモから研究者ファイルを作成する際の問題[2]を論議する．研究者メモ自身1つの研究者ファイルであるが，複数のメモファイルの統合を考える．研究者メモ M を次の3つ組

$M=<S, X, Y>$.
S はキーワードの集合,
$X=\{(w_1, w_2) \in S\times S \mid w_1 \succcurlyeq w_2\}$,
$Y=\{(w_1, w_2) \in S\times S \mid w_1 \doteqdot w_2\}$,

で定義する．M において，

$X \cap Y=\phi$, $X \cap Y^{-1}=\phi$ かつ $Y \cap Y^{-1}=\phi$

となる場合に研究者メモ M は無矛盾であるという．無矛盾であるときに，S の$\doteqdot$による同値類の集合は，関係$\succcurlyeq$についての半順序集合になる.

次に，研究者メモに対する演算∇を定義する．$M_1=<S_1, X_1, Y_1>$, $M_2=<S_2, X_2, Y_2>$を2つの研究者メモとするとき，演算∇を

$M_1 \nabla M_2=<S_1 \cup S_2, X_1 \cup X_2, Y_1 \cup Y_2>$,

のように定義し RM 和とよぶ，この演算∇により，研究者ファイルを作成する.

この際に，階層関係や同義関係において，矛盾したデータが存在することも考えられる．システムによるシソーラス作成の支援とはこの矛盾の検出である．旧システムでは，この▽演算による編集の過程において矛盾が生じた場合には警告が発せられ，研究者の記入のミスや見方の違いによって生じる矛盾の検出ができるようになっている．

論理プログラムにより新たに実現されたシステムにおいて，研究者ファイルの中に矛盾する階層関係

$r(w_1, w_2)$. と $r(w_2, w_1)$.

が存在する場合には，上記の r* を用いると，その推移関係が無限ループに入る．現在，このような矛盾の検出システムは，実現していないので，無矛盾性の維持には注意を要する．

4. 多段階検索の特徴

あるキーワードで検索を行った際に，検索結果から新たなキーワードを見いだし，それを用いて再度検索を行うことはよくある．多段階検索ではこのような一連の検索過程を考えている．1つのキーワード→文献→その文献のキーワードというような例に代表される知識の連鎖が多段階検索の基礎であり，知識の連鎖の回数が段階数に対応する．このような形式でキーワードの連鎖を見いだす方式のことを語連関（word association）とよぶ．

また，研究者ファイルが有するキーワードの連鎖関係を利用する検索方式のことを，研究者ファイルを用いる多段階検索とよぶ．

多段階検索では，次のように定義される再現率（recall ratio）の増加が期待できる．

$$\text{再現率}=\frac{\text{検索された必要な文献数}}{\text{必要である全文献数}}$$

しかし，次で定義される適合率（precision ratio）

$$\text{適合率}=\frac{\text{検索された必要な文献数}}{\text{検索された文献数}}$$

は減少すると考えられる．MIR-RF はより知的な検索システムであるが，通常の検索の評価からいうとその特徴は，検索もれを少なくすることにある．

5. 検索システムの数学的定式化

本節では，上記のシステムの数学的定式化を行う．

検索の対象になる文献情報の集合は，有限集合であり，各文献情報には1から有限集合の要素数までのいずれかの正整数が一意に対応しているものとする．文献情報に対応している正整数を文献番号とよび，その正整数の集合を D とする．以後，文献情報の意味で文献ともいうことにする．また，V を有限集合とし，その要素を単語とよぶ．2個以上かつ有限個の単語をブランク記号で連接した有限長のもの全体と V との和集合を V^+ とする．V^+ の要素をキーワードという．

〔定義1〕

任意の $n(\in D)$ に対して $K(n)$ が有限集合である写像：$D\rightarrow 2^{V^+}$を索引付けとよぶ．

各文献情報には有限個のキーワードが対応し，1つの索引付け K のもとで，文献 n の主題は $K(n)$ により特徴付けられていると考える．

〔定義2〕

次の2項関係 F, $F\subset D\times K(D)$

を索引付け K にもとづく文献集合 D 上の索引ファイルとよび，

$F^{-1}=\{(w, n)\mid (n, w)\in F\}$

をその転置ファイルとよぶ．

定義2の内容を論理プログラムで記述するには，関係 F に対応する述語 f を用意し，

(n, w) $(\in F)$ に対して，$f(n, w)$.

と事実（ファクト）として記述すればよい．転置ファイル F^{-1} に対する事実は不要である．

質問は，次の質問式によって記述される．

〔定義3〕

質問式は次のように構成される．

1）0，1は質問式である．

2）$q\in V^+$ なら，q は質問式である．

3）q_1, q_2 が質問式であれば，$(q_1 \vee q_2)$，$(q_1 \vee q_2)$，$(\sim q_1)$ は質問式である．

4）1），2），3）によって生成されるものだけが質問式である．質問式全体の集合を Q とかくことにする．

この質問式に対して，論理型検索を定義する．

〔定義4〕

索引付け K にもとづく D 上の論理型検索を以下の条件を満足する写像 L で与える．

$L: Q\rightarrow 2^D$.

条件は，

1）$L(0)=\phi$（空集合），$L(1)=D$（全体集合）．

2）$w\in V^+$ のとき，

$L(w)=F^{-1}(w)$ $(=\{(n\in D \mid w, n)\in F^{-1}\})$.

3) $q_1, q_2\in Q$ のとき,

$L((q_1 \vee q_2))=L(q_1)\cup L(q_2)$.

$L((q_1 \wedge q_2))=L(q_1)\cap L(q_2)$.

$L((\sim q_1))=D\setminus L(q_1)$.

定義4にもとづく実際の質問式の計算は転置ファイルから得られる文献集合の集合演算を行うことでできる.

次に,論理型検索の拡張として多段階検索を定義する.まず,索引付けにより自然に生じる語連関を利用する場合を考える.

〔定義5〕

1つの索引付け K にもとづく文献集合 D 上の索引ファイル F と,その転置ファイル F^{-1} に対して,

$F^{-1}\mathrm{o}F=\{(w_1, w_2) \mid (w_1, n)\in F^{-1},$

$(n, w_2)\in F$ なる n が存在する$\}$

と定義し,演算 o を $K(D)$ $(=\cup_{n\in D} K(n))$ 上の語連関係とよぶ.

〔定義6〕

語連関 o を用いる m 段階検索を次の条件を満足する写像 L^m で与える.

$L^m: Q\to 2^D$.

条件は,

1) $L^m(0)=\phi$ (空集合).

$L^m(1)=D$(全体集合).

2) $w\in V^+$ のとき,

$L^m(w)=(F^{-1}\mathrm{o}F)^{m-1}\mathrm{o}F^{-1}(w)$.

3) $q_1, q_2\in Q$ のとき,

$L^m((q_1 \vee q_2))=L^m(q_1)\cup L^m(q_2)$.

$L^m((q_1 \wedge q_2))=L^m(q_1)\cap L^m(q_2)$.

$L^m((\sim q_1))=D\setminus L^m(q_1)$.

また,単語が有する意味の階層性から推移関係を規定できる.

〔定義7〕

研究者ファイル RF(3.1 の知識の表現において記述している) に対して,

$RF^{-1}\mathrm{o}\downarrow RF=\{(w_1, w_2) \mid$

$[w_1: x]\in RF^{-1}, [z: w_2]\in RF,$

$xy=z$ なる長さ1の記号列 y がある.$\}$

と定義し, o↓ を研究者ファイルにおける下位概念への語連関とよぶ.上位概念への語連関は,

$RF^{-1}\mathrm{o}\uparrow RF=\{(w_2, w_1) \mid (w_1, w_2)\in RF^{-1}\mathrm{o}\downarrow RF\}$

で定義される.

研究者ファイルにおける語連関を利用する多段階検索の定義を次のように与える.

〔定義8〕

研究者ファイルを用いる下位概念への m 段階検索を次の条件を満足する写像 $M^m\downarrow$ で与える.

$M^m\downarrow: Q\to 2^D$.

条件は,

1) $M^m\downarrow(0)=\phi$ (空集合).

$M^m\downarrow(1)=D$(全体集合).

2) $w\in V^+$ のとき,

$M^m\downarrow(w)=(RF^{-1}\mathrm{o}\downarrow RF)^{m-1}\mathrm{o}F^{-1}(w)$.

3) $q_1, q_2\in Q$ のとき,

$M^m\downarrow((q_1 \vee q_2))=M^m\downarrow(q_1)\cup M^m\downarrow(q_2)$.

$M^m\downarrow((q_1 \wedge q_2))=M^m\downarrow(q_1)\cap M^m\downarrow(q_2)$.

$M^m\downarrow((\sim q_1))=D\setminus M^m\downarrow(q_1)$.

$M^m\uparrow$ については, o↓ を o↑ におきなおせば,他は定義8と同様である.

次に,定義7と8を論理プログラムの表現形式で考える.論理プログラムによる研究者ファイルの表現は,知識の表現のところで示したように,2引数の述語 r と eq により表現された.定義7の語連関に対応する論理プログラム $r^*(X, Y)$ (いくらかのステップで X から Y へ到達する) は,3.1において既に示している.また,ステップ数を r^* に組み込むこともできる.

$r^*(X, M, Y)$ (M ステップで X から Y へ到達する) という論理プログラムは,

$r^*(X, 0, Y)$: $-r(X, Y)$.

$r^*(X, M, Y)$: $-r(X, Z), M1$ *is* M-$1, r^*(Z, M1, Y)$.

と再帰的に記述できる.定義8において,質問式が,キーワード1個である場合の検索は,モデル論的には,推移関係により新たに見いだされたキーワードを用いた論理式

$\vee \quad f(X, w')$

新たなキーワードw'

を充足する文献 n を検索することを主張している.質問式が,キーワードの論理的結合である場合は,各キーワードに対して,上記の論理式を充足する文献を検索し,それらの集合演算を行うことで質問式の検索がなされる。具体的な論理プログラムは,6節で示す.

6. Prolog によるシステムの実現

本節では,Prolog による具体的プログラムを示しながら実現している機能の説明をする.ただ,研究者ファイル作成における支援等,まだ実現されていない

機能もある．なお，Prolog の表記は参考文献3）に従うものとする．

6.1　システムのデータ構造

1）文献情報ファイル（DF: Data File）

文献情報の項目は，図書の分類記号（主キーとする），書名，著者名，出版社名，発行年，ページの6項目とし，document という述語を用い，上記の順にしたがって以下のように与える．

```
document ('AS 921/1', [topics, in stochastic, process],
['ASH, R. B.' , 'GARDNER'] , ['ACADEMIC PRESS,
NEW YORK'], 1975, 321).
```

項目の値が一般に1つの項（term）でかけない書名，著者名，出版社名は，上記のようにリストで与える．変数は先頭がアルファベットの大文字である文字列であり，先頭が大文字で始まる項は，‘ ’で囲み変数と区別する．

2）索引ファイル（KF: Keyword File）

DF だけを用いる検索で理論上支障はないが，実行速度を考えて，書名，著者名，出版社名の3項目については，述語 g を用い以下のような索引を作る．

```
g(topics, 'AS921/1').
g('ASH, R. B.', 'AS921/1').
```

この述語 g は，5節で定義した2引数述語 f の引数の順序を入れ換えたものである．この KF は DF から自動的に作成される．（そのプログラムは，付録に示す．）その際，実際のデータ（INSPEC-C 1981年分）において，出現頻度が上位の128個に含まれる機能語51単語については，キーワードにしない．書名，著者名，出版社名それぞれ同じ述語を用いているが，混同するならば，別の述語を用いてもよい．

3）研究者ファイル（RF: Researcher File）

各研究者自身の知識，見解など独自の情報を集めたもので，本システムで対象としたのはキーワード間の意味の階層関係と同義関係からなるシソーラスである．上記したように，述語 r と eq により与える．

6.2　実現されたシステムの検索機能

本節では，6.1．において与えたデータを用いる標準的な検索方式や，より知的な検索方式についてプログラムを示しながら説明する．検索機能が以下のような簡単な記述で実現できるのは，Prolog 自体の検索機能がそのまま利用できるからである．

1）文献情報だけを用いる検索

document という述語は6個の引数をもつ，発行年，ページの2項目についての検索プログラムは，year, page という2つの述語を用いて，

```
year(X, N): -document(N,_,_,_, X,).
page(X, N): -document(N,_,_,_,_,X).
```

と記述できる．実際の質問では X にそれぞれの項目の値を与えると，そのときの N の値が質問の条件を充足する主キーの値になる．例えば，「出版年が1985年である文献」という質問は，

```
? - year(1985, N).
```

であり，Prolog インタプリタにより N に条件を満足する主キーの値が代入される．

書名，著者名と出版社名については，その情報をリストの形で与えており，質問の条件もリストで与える．検索は質問の条件リストが対応する document の引数の部分リストであるときに行う．プログラムは author, publish 等の述語を用いて，以下のようにかける．

```
author(X, N): -
    document(N,_, Au,_,_,_), sublist(X, Au).
publish(X, N): -
    document(N,_,_,_, Pu,_,_), sublist(X, Pu).
```

ここで，sublist という述語は第一引数が第二引数の部分リストであるときに真である述語で，

```
sublist([X|L], [X|M]): -prefix(L, M), !.
sublist(L, [_|M]): -sublist(L, M).
prefix([],_).
prefix([X|L], [X|M]): -prefix(L, M).
```

で表現される．例えば，「著者が Lipski である文献」という質問は，

```
? - author(['Lipski'], N).
```

となる．

述語 sublist は，リストの中での項の順序まで意味をもつ，その順序を考えない場合には，次の subset という述語を sublist の代わりに用いればよい．

```
subset([A|X],Y): -member(A, Y), subset(X, Y).
subset([], L).
member(X, [X|_]).
member(X, [_|Y]): -member(X, Y).
```

2）文献情報と索引情報を用いる検索

文献情報から不用語を除去して自動的に作られる検索情報（述語が g である事実）を用いると，title の表現ができ計算速度の面からも有益である．

書名の項目については，title, proce_title という2つの述語を用いて，

```
title([X|L], N): -proce_title(X, N), title(L, N).
title([], _).
proce_title(X, N): -stop_word(S), member(X, S), !.
proce_title(X, N): -g(X, N).
```

と表現できる．ここで，stop_word の引数は不用語である51単語からなるリストである．member は第一引数が第二引数の要素であるとき真となる述語である．

3）研究者ファイルと索引情報を用いる検索

本システムで対象とする研究者ファイルは，各キーワード間の上位，下位の概念に関する階層関係と同義関係であり，1つの語連関である．主に階層関係を利用する検索プログラムを示すが，同義関係を利用する検索プログラムも無限ループに対する制御を付けると，階層関係の場合と同様に記述できる．書名に関する検索について，階層関係を用いたプログラムを示す．

Ⅰ）下位概念を利用する多段階検索

これは，1つのキーワードより下位概念のキーワードを有するすべての文献の主キーを検索する方式である．

```
down 1(X, N): -title([X],N).
down 1(X, N): -r(X, Y), down 1(Y, N).
```

Ⅱ）下位概念を利用する m 段階検索

これは，1つのキーワードから階層関係 r を下位方向に最大 m 回用いて到達できるキーワードを有する文献の主キーを検索する方式である．

```
down 2(X, 0, N): -title([X], N).
down 2(X, M, N): -M>0, title(X, N).
down 2(X, M, N): -r(X, Y), L is M-1,
                  down 2(Y, L, N).
```

Ⅲ）下位概念を利用する m 段階目だけの検索

これは，1つのキーワードから階層関係 r を下位方向に m 回用いて到達できるキーワードを有する文献の主キーを検索する方式である．

```
down 3(X, 0, N): -title([X], N).
down 3(X, M, N):-r(X, Y), L is M-1,
                  down 3(Y, L, N).
```

以上3種の知的検索方式が簡単に記述できた．また，上記は下位概念についての記述であるが，上位概念についての記述は，適当な述語を用意し述語 r の引数の位置を入れ換えることにより同様に記述できる．例えば，上位概念を利用する多段階検索は，

```
up(X, N): -title([X], N).
up(X, N): -r(Y, X), up(Y, N).
```

とかける．

6.3. 文献，索引情報と論理式を用いる検索

上記の質問は，1個の素論理式による質問であったが，それぞれの素論理式と連言，選言と否定の論理記号により質問の拡張ができる．ただし，実際の入力の形式は積和形をした論理式に限られる．また，質問の否定と Prolog の否定とは多少意味が異なり，連言部の先頭が否定記号を有する素論理式であるときに，上記のプログラムは質問に意味とは異なる検索を行う．そこで，連言の先頭に否定がある場合は，すべての文献情報が満足する素論理式をその先頭に追加し意味の差異を除去している．以上のような注意をすれば，複合された条件も表現できる．例えば，

Ⅰ）「著者が Lipski で，1985年に出版された文献」という質問は，

```
? - author(['Lipski'], N), year(1985, N).
```

とかける．

Ⅱ）「1984年の文献で，null value([null, value] と記述）より下位概念のキーワードをもつ文献」については，

```
? - year(1984, N), down 1([null, value], N).
```

と表現できる．

Ⅲ）「1984年以外の年に出版された文献で，database より2段階までの下位概念のキーワードをもつ文献」という質問は，

```
? - not(year(1984, N)), down 2(database, 2, N).
```

と入力するが，

```
? - document(N,_,_,_,_), not(year(1984, N)),
    down 2(database, 2, N).
```

とシステム側で変換される．

検索方式に関連して以上4つの方式を示した．研究者ファイルを用いると，キーワードの上位，下位概念等の意味を扱う知的検索を行える．

Prolog による新システムの利点は，以下のとうりである．

1）Prolog の単一化機能をそのまま利用できるので，検索部のプログラムが簡単にかける．

2）転置ファイルが不用となり，データの追加，更新が簡略化される．

3）再帰的プログラムがかけるので，語連関や知識の連鎖等で記述ができるし，事実としての知識以外にもルールとしての知識まで拡張可能である．

自然言語に関する細かいキーワードのマッチングの

点には難点もあるが，それは Fortran による実現の場合にも生じることであり，総じて Prolog の方が本システム記述には適した言語と考えられる．現在，実現しているシステムの知的支援は，キーワードが有する概念の階層と同義関係からなるシソーラスのみであるが，今後は論理プログラムを１つの知識とした場合の知識の階層関係を利用するシステムの実現等も考えている．また，知識としての研究者ファイルの編集や管理を行うシステムの充実も必要である．

7. おわりに

Prolog 版 MIR-RF システムの実現に際して，知識の表現，管理と蓄積に関する問題点，またそれら知識を利用する多段階検索につて論議した．システムの記述に Prolog を用いることで，データの管理が簡素化され，プログラム自体も簡単に記述できた．さらに，知的な情報処理は，単にキーワードに関する知識から，ルールを含む知識まで容易に拡張できる．また，多段階検索の基礎であるキーワード獲得自体は，知識の獲得と関連する問題である．知識の獲得方式の論議がすすみ，階層や同義関係と同様にその方式をシステムに組み込めれば，文献情報を越えたより一般的な情報の検索や，知的な質問応答ができるようになると考えられる．

実現している MIR-RF システムは現在のところ単に文献情報を扱っているが，広くはこのように一般の知識処理の方式を定式化したものと考えられ，人工知能の分野にも深く関係する．

なお，このシステムは，DCL U-ステーション上の C - Prolog によって実現している．

参考文献

1) S. Arikawa and T. Kitagawa: Multistage Information Retrieval System Based upon Researcher Files, Proc. 2nd USA-Japan Comp. Conf., 149-153, (1975), and also in Res. Rept. Fund. Inform. Sci., Kyushu University, No. 51, (1975).

2) S. Takeya: Researcher File Manipulation in MIR-RF System, Res. Rept. Inform. Sci., Kyushu University, No. 55, (1975).

3) W. F. Clocksin and C. S. Mellish: Programming, in Prolog Springer-Verlag, (1981).

付　　録

文献情報ファイルから索引情報ファイルを自動作成するプログラム index は，現在のところ以下のとうりである．述語の下線のあるものは，C-Prolog の組み込み述語である．

```
index; -write('data file name'), read(Fname),
        write('index file name'), read(Oname),
        seeing(Input), see(Fname), tell(Oname),
        repeat,
        read(Term), process(Term),
        seen, see(Input), !.
process(end): -told, write('index end'), !.
process(X): -functor(X, F, N), check(N, X).
check(6, X): -arg(1, X, A), arg(2, X, A1),
              pro 2(A1, A), arg(3, X, A3),
              pro 2(A3, A), arg(4, X, A4),
              pro 2(A4, A), !, fail.
check(Y, X): -told, write('data error:'), nl,
              write(X), !.
pro 2([], F): - !.
pro 2([A | X], F): -pro(A, F), pro 2(X, F).
pro(A, F): -stop_word(S), member(A, S), !.
pro(A, F):-writeq(k(A, F):-write('.'), nl.
stop_word ([a, all, also, an, and, are, as, at, be, been,
            between, both, but, by, can, each, for,
            from, has, have, how, i, in, into, is, it, its,
            may, of, on, one, only, or, other, several,
            some, such, than, that, the, there, these,
            this, to, under, was, when, where, which,
            will, with]).
```