

B 1 「情報検索とSL-導出定理証明プログラム」

二村良彦 (日立中央研究所)

要 旨

データ・ベースに、明確な形式で含まれていない情報に関する問合わせに対しても、推論機能を用いることによって、応答することのできる、情報検索システムの新しい理論と構成法について述べる。本文で述べる推論機能は、情報検索システムとKowalski - KeuhnerのSL-resolution Theorem - Proverの特徴を、充分に利用して作られている。また、本文で述べる情報検索システムは、有意義な情報を自動的に作成し、データ・ベースに付加することができる。

1. 緒 言

導出原理^{注1)}を用いて情報検索(IR)を行なう場合には、データ・ベース(例えば、S)を第一階述語論理で表わし、問合わせを証明すべき定理(例えば、T)として表わす。定理Tの証明の途中で、特別な変数に対して代入された定数項が、応答を構成するために用いられる(Green[2]又はLuckham and Nilsson[3]参照)。従って、IRの問題も通常の定理証明と同様に、先ず証明を発見することにある。しかしIRにおける定理証明には、次の2つの特徴が有る。

(1) IRシステムは大きなデータ・ベースと証明に無関係な公理を多量に含む。

IRのためのデータ・ベースは、通常 10^5 項目以上の大きさである⁵⁾。そのうちの少数の公理だけが、問合わせT(定理)の証明に用いられる。Darlington⁴⁾はこの状況を次のように述べている。

$S = S(R) \cup S(I)$ とおく。ただし $S(R)$ は、Tの証明に使われる(即ち、Tの証明木(proof tree)に現われる)公理の集合、そして $S(I) = S - S(R)$ とする。

$S(I)$ を更に次のように細分する。

$$S(I) = S(I_1) \cup S(I_{21}) \cup S(I_{22})$$

ただし、 $S(I_1)$ の各要素は、 $S(R) \cup \bar{T}$ の要素と共通の述語を含まない。 $S(I_{21})$ の各要素と $S(R) \cup \bar{T}$ の要素の否定からは、導出形(resolvent)が導びけない。 $S(I_{22})$ の要素と $S(R) \cup \bar{T}$ の要素の否定とからは、導出形が導びける(即ち、 $S(I_{22}) = S(I) - \{ S(I_1) \cup S(I_{21}) \}$)。

$S(I_{22})$ の要素と $S(R) \cup \bar{T}$ の要素とから導出形が導びけ、その導出形と $S(I_1) \cup S(I_{21})$ の要素とから導出形が導びける場合がある。この場合に証明にとって無関係なクローズ(clause)が新たに生成される。問合わせTの証明において、 $S(I_{22})$ の影響をできるだけ小さくすることが、IRシステムにとって本質的に重要であるということを、Darlington⁴⁾は示した。

(2) IRシステムに対する問合わせは、比較的安定したデータ・ベースに対して行なわれる。

例えば、RAND Relational Data File⁶⁾は、70,000項目より成るデータ・ベースを持ち、サイバネティクスの分野に関する事項検索を行なうことができる。システムへの問合わせは、「誰がどんな論文を書いたか」

注1. 本論文では、導出原理とSL-導出原理に関する読者の知識を仮定している。特に“入力鎖(input chain)” “セル”, “分類木(classification tree)”および“許容制限(admissibility restriction)”のような用語に不慣れた読者は、文献[1]および[9]を参照されたい。

「誰がどの機関に所属しているか」等のようなものである。従って、データ・ベースの大部分は長期にわたって不変である。このような場合には、データ・ベースの構造と意味を充分に利用して、定理証明プログラムの探索戦略を最適化することができる。公理に対して最適化された証明手続は、その公理系における公理を証明する際に、最適化されていない証明手続よりも能率良く証明する。SlagleとNortonの定理証明プログラム⁷⁾は、半順序の公理に対する組込まれた推論規則を持っており、半順序に関する定理の証明に対して非常に強力である。その定理証明プログラムは、HoareのFINDプログラム⁸⁾に関する正しさを証明するための18個の補題を、全て証明した。Darlington⁴⁾は、IRにおいて集合 $S(I_{22})$ の影響をできるだけ小さくする方法については考察したが、公理の特徴を利用して証明手続を最適化する方法については考察していない。

最近の証明手続の中では、Kowalski and KuehnerのSL-導出⁹⁾が最もIRシステムに適しているように考えられる。その理由は、

- (1) $\bar{T}$ が支持集合 (support set) として指定されるならば、 $S(I_{22})$ の要素が $S(R) \cup \bar{T}$ の要素とリゾルブ (resolve) される機会は、先端クローズ (top clause) に対する入力導出 (input resolution) のときだけである。即ち、高々 $\#(\bar{T}) \times \#S(I_{22})$ 回である。
- (2) 証明木の枝分れ率を選出関数によって最少にすることができる。
- (3) データ・ベース S の大部分は、全ての問合わせ T に対して固定していると考えられるので、 S に基づいて有用な定理 (基本定理) を前以て生成しておき、それを分類木に加えておくことができる。基本定理の使用により、証明の発見が速くできる場合がある。

本論文では、(a)SL-導出を応用したIRシステムの構成法、(b)基本定理の生成法、(c)分類木の構成法、および(d)探索戦略における基本定理の利用法、について述べる。本文で特に新しい点は、基本定理の生成法と利用法に関するものである。

2. IRシステムのためのモデル

LevienとMaron⁵⁾は、relational date fileと呼ばれるIRシステムのためのモデルを提案した。そのシステムの目的は、大きな事項データの論理的分析を助けることにある。システムは、内包ファイル (intensional file, IF と略記する) と外延ファイル (extensional file, EF と略記する) の2つのファイルから構成される。IFは事項間の関係に関するデータを含み (このファイルは導出原理の用語を使うなら、“変数を含むクローズの集合”である)、EFは基本データ文 (事項) を含む (このファイルは“変数を含まないクローズの集合”である)。システムへの問合わせは、ファイルに明確な形式で含まれていない事項に関するものでも良い (図1参照)。

Extensional File

1. A. P. Smith is a system engineer. (or $S(\text{Smith})$)
2. A. P. Smith is affiliated with Acme Electric Corporation.
(or $A(\text{Smith}, \text{Acme})$)
3. A. P. Smith received the Ph.D. degree from UCLA.
(or $R(\text{Smith}, \text{Ph.D.}, \text{UCLA})$)
4. Ph.D. is a degree. (or $D(\text{Ph.D.})$)
- ⋮
- ⋮
- ⋮

Intensional File:

If one (x) received a degree (y) from some institution (z) then he graduated from that institution (z).

(or $\bar{D}(y) \bar{R}(x, y, z) \bar{G}(x, z)$)

•
•
•

Query

Is there any system engineer in Acme who graduated from UCLA?

(or $\bar{S}(x) \bar{A}(x, \text{Acme}) \bar{G}(x, \text{UCLA})$)

Figure 1: Example of the extensional file, intensional file and a query in the relational data file.

Levien 等は、ファイルから事項を引き出すために定理証明プログラムを使用していないが、彼等のモデルは、SL-導出定理証明プログラムに基づくIRシステムに対しても適用可能である (図2参照)。

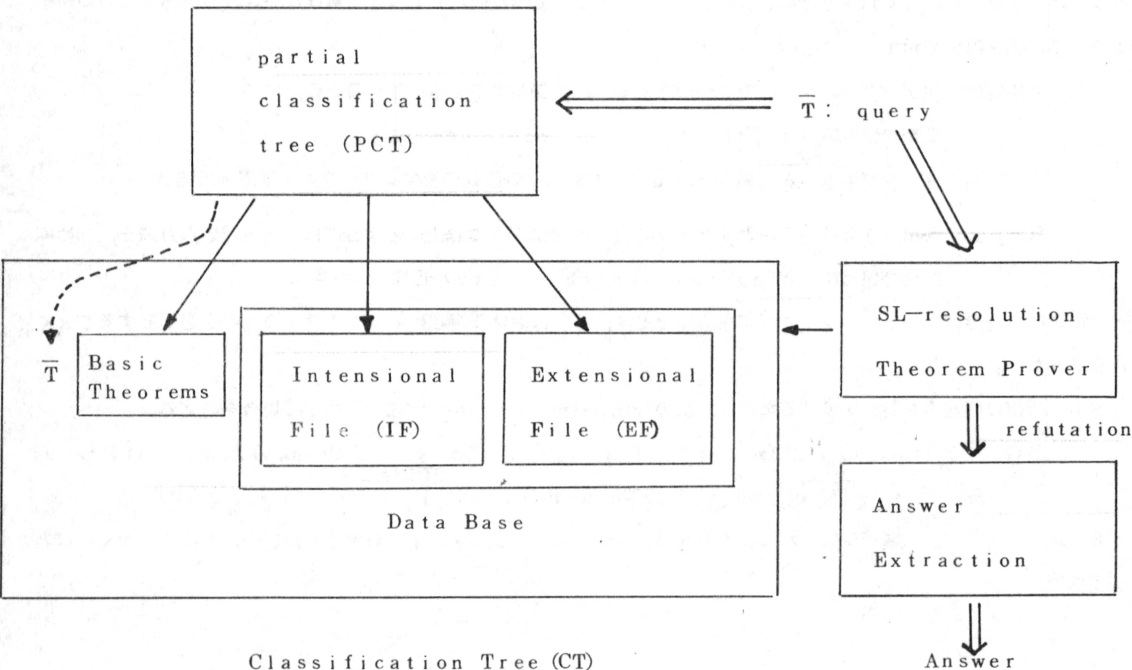


Figure 2: Model of IR system

Double arrows represent inputting data to processes or adding

date to other date. Single arrows and dotted arrows mean references to date.

$\bar{T}$ is added to PCT to construct CT. $\bar{T}$ will be removed from CT after an answer to $\bar{T}$ has been extracted. Basic theorems are constructed from IF (see Section 3).

図2において、PCT、IF、EFおよび基本定理は、システムに対して固定されている。Tは新しい関係および問合わせから成る。PCTは、次章で述べられる方法により自動的に作成できるのでシステムのユーザは、IFを定義することにより、自分にとって関連の深い分野に関する個人用IRシステムを、共通ファイルEFの上に構成することができる。

問合わせに対して応答を引き出す過程は、ANS述語²⁾又はトートロジーの再証明³⁾によって行なうことができる。

3. SL-導出定理証明プログラムのための探索戦略

3.1 定理証明プログラムの繊細さ

全ての問題に対して、常にうまく対処できるような探索戦略は存在しない。¹⁰⁾特に縦型探索戦略は、リテラルの入力順序に関して非常に繊細である。¹¹⁾運よくリテラルが都合の良い順序で入力されるならば、縦型定理証明プログラムは、すみやかに解に到達できる。しかしそうでないと、簡単な問題でもうまく解けない。SL-導出のこの繊細さを以下に例を用いて示す。

SL-導出定理証明プログラムは、次の3つの要素から成る探索戦略を用いるものと仮定する。

(S₁) 縦型探索戦略 (depth-first search strategy)

(S₂) リゾルブできる入力鎖 (input chain) の個数が最少のリテラルを選ぶ選出関数

(S₃) 選出されたリテラルとリゾルブできる入力鎖が2つ以上ある場合には、一番短い入力鎖を、初めに対角探索戦略 (diagonal search strategy)

S₂ における選出は、バックトラッキングを必要とせず、S₃ における選出はバックトラッキングを必要とすることに注意されたい。

SL-導出の問題点を明らかにするために、次の公理BGからいくつかの定理を証明する場合を考える。

$$BG = \{ P(x, x, E), P(x, E, x), P(E, x, x), \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(x, v, w) P(z, u, w), \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(z, u, w) P(x, v, w) \}$$

BGから17個の入力鎖が生成される。それをB₁, ..., B₁₇で表わす。IBG = { B₁, ..., B₁₇ }とおく。(表-1参照)。

Table 1: IBG

$$\text{IBG: } B_1 = P(x, x, E)$$

$$B_2 = P(x, E, x)$$

$$B_3 = P(E, x, x)$$

$$B_4 = \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(x, v, w) P(z, u, w)$$

$$B_5 = \bar{P}(y, u, v) \bar{P}(x, v, w) P(z, u, w) \bar{P}(x, y, z)$$

$$B_6 = \bar{P}(x, v, w) P(z, u, w) \bar{P}(x, y, z) \bar{P}(y, u, v)$$

$$B_7 = P(z, u, w) \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(x, v, w)$$

$$B_8 = \bar{P}(u, u, z) \bar{P}(u, z, w) P(z, u, w)$$

$$B_9 = \bar{P}(y, w, w) P(z, w, w) \bar{P}(y, y, z)$$

$$B_{10} = \bar{P}(x, v, z) \bar{P}(v, u, v) P(z, u, z)$$

$$B_{11} = \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(z, u, w) P(x, v, w)$$

$$B_{12} = \bar{P}(y, u, v) \bar{P}(z, u, w) P(x, v, w) \bar{P}(x, y, z)$$

$$B_{13} = \bar{P}(z, u, w) P(x, v, w) \bar{P}(x, y, z) \bar{P}(y, u, v)$$

$$B_{14} = P(x, v, w) \bar{P}(x, y, z) \bar{P}(y, u, v) \bar{P}(z, u, w)$$

$$B_{15} = \bar{P}(u, u, z) \bar{P}(z, u, w) P(u, z, w)$$

$$B_{16} = \bar{P}(z, u, w) P(x, w, w) \bar{P}(x, z, z)$$

$$B_{17} = \bar{P}(z, u, z) \bar{P}(u, u, v) P(z, v, z)$$

BGから $P(B, A, C) \vee \bar{P}(A, B, C)$ を、証明する。即ち、 $\text{ROB2} = \{P(A, B, C), \bar{P}(B, A, C)\} \cup \text{BG}$ の非充足性 (unsatisfiability) を証明する。

$B_8 = P(A, B, C)$ および $B_{10} = \bar{P}(B, A, C)$ をIBGに追加し、 B_{10} を先端クローズとして証明を開始する。 B_7 と B_{14} が先端クローズ B_{10} とリゾルブ可能である (この場合には、先端クローズは、1つのリテラルから成っているので、選出関数はそのリテラルを選ぶ)。クローズ B_7 と B_{14} は同じ長さなので、戦略 S_3 は、 B_7 と B_{14} に対して同じ優先順位を与える。この場合 S_3 にはどちらを選んだ方がより速く証明が得られるかわからない。それ故選択は運にまかせられる。ここで B_7 が最初に選ばれたと仮定す。(図3参照)すると選出関数は、 $\bar{P}(x, y, E)$

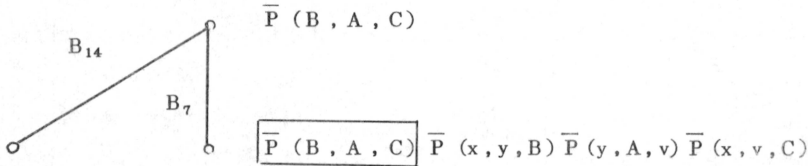


Figure 3: A node in the SL-resolution search tree.

又は $\bar{P}(y, A, v)$ のどちらかを選ぶなければならないが、それ等は、同じ個数 (4個) の入力鎖とリゾルブできる。従って選出関数もどちらを選んだ方が速く証明できるかわからない。このような場合には、トップダウン型SL-導出定理証明プログラムは、深くて多くのバックトラッキングをする探索におちいりやすい。

クローズの集合ROB2の非充足性は、横型のG-deductionを、用いては620秒で証明できたが、SL-導出では、生成されたクローズの個数が、制限を超えたために証明できなかったという報告がある。¹¹⁾

しかし、もし次の2つのクローズ B_{20} と B_{21} が前以てIBGに加えられていたならば、証明は容易にできたであろう。

(図4参照)

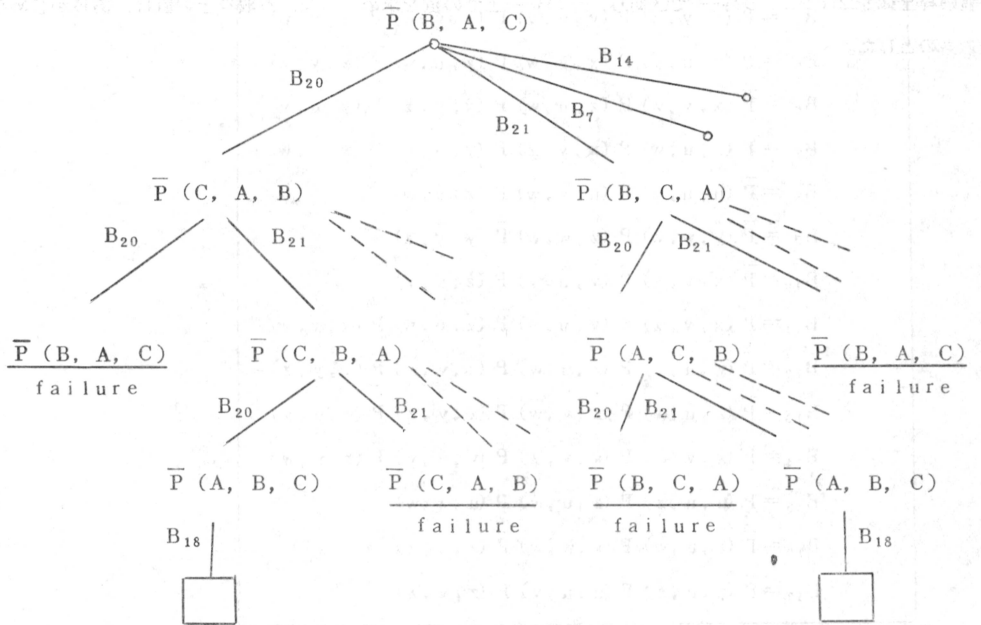


Figure 4: Search tree for SL-resolution.

Since B_{20} and B_{21} are shorter than B_7 and B_{14} , B_7 and B_{14} have never been tried before finding a first proof. The failures are caused by the violation of the admissibility restriction.

$$B_{20} = P(z, u, w) \bar{P}(w, u, z)$$

$$B_{21} = P(x, v, w) \bar{P}(x, w, v)$$

次に、与えられた公理から新しい入力鎖（例えば上の B_{20} , B_{21} ）を、生成する方法について述べる。この生成された入力鎖を基本定理と呼ぶ。

3.2 基本定理の生成法

補題 (lemma) を生成する方法とその利益は、Loveland¹²⁾ および Kowalski and Kuehner⁹⁾ によって研究された。彼等は1つの定理を証明する過程で補題を構成し、その過程の後の部分でその補題を使う。補題は証明しようとしている定理に対してだけ有効なので、定理の証明が終了すると捨てられる。

この節では、定理証明プログラムが使われるときには、いつでも利用することのできる「基本定理」を公理から自動的に導き、入力鎖として分類木に加える方法について述べる。

基本定理は、公理から導びくことのできる最も一般的と考えられるクローズである。しかし、与えられた定理又は定

理の与え方に対する証明手続の繊細さを増加させないためには、基本定理の選び方に充分注意する必要がある。

ここでは、生成される基本定理に対して次の4つの制限を設ける。

(1) 深さに関する制限 公理 (IC) に含まれる一番深くネストしたスコール関数よりも深くネストしたスコール関数を、含むクローズは基本定理としない。そのようなクローズは、一般性が乏しいと考えられるからである。

(2) レベルに関する制限 公理から一回だけの演導で導かれるクローズは、基本定理としない。そのようなクローズは、証明手続にとって有害無益であると考えられるからである。

(3) 長さに関する制限 基本定理の長さは1又は2に限る。長さ1の基本定理は常に有用であり、長さ2の基本定理は有用な場合が多いからである。

(4) 長さ2の基本定理に関する制限 長さ2の基本定理は、1つの入力鎖に対して高々1つしか生成しない。長さ2の基本定理は証明手続の能率を悪くする可能性があるからである。

基本定理を生成するアルゴリズムは、次の5ステップより成る。

ステップ1 L をリテラル、そして C をクローズとする LC 形の基本定理を生成したい場合には、入力鎖の中から先頭のリテラルが L 、又は L の alphabetic variant であるような全てのクローズを捜し、それ等の各クローズに対してステップ2を適用する。

ステップ2 $C_0 = L(x) L_1^0 \cdots L_n^0$ ($n \geq 1$) はステップ1で捜し出された入力鎖を、そして x は L に含まれる変数 (又は変数のベクトル) を表すものとする。このとき、 j を0に設定し、ステップ3を行なう。

ステップ3 $C_j = L(x) L_1^j L_2^j \cdots L_{n-j}^j$ とおく。

(1) C_j が深さに関する制限を超えているならば、何もしないで止まる。

(2) そうでなければ、全ての i ($1 \leq i \leq n-j$) に対して、 $D_i^j = \{ K \mid K \text{ は } IC \text{ 中の単位クローズで、かつ最も一般的な統合子 (most general unifier, mgu と略記する) } \sigma \text{ に対して、} L_i^j \sigma = \bar{K} \sigma \text{ かつ } L(x) \sigma = L(x) \text{ (即ち、} x \text{ の変数を定数と考えたときに } \bar{K} \text{ は、} L_i^j \text{ と mgu } \sigma \text{ によって統合できる)} \}$ とおく。

(a) ある i_0 ($1 \leq i_0 \leq n-j$) に対して $D_{i_0}^j \neq \emptyset$ ならばステップ4を行なう。

(b) そうでなければ、鎖の変換を終了し、

(i) $j \leq 1$ ならば何もしない (レベルに関する制限)。

(ii) $n-j \geq 2$ ならば何もしない (長さに関する制限)。

(iii) $n-j = 0$ ならば、 C_j を基本定理とする。

(iv) $n-j = 1$ かつクローズ LC に対して、この点に制御が最初に来た場合には、 C_j を基本定理とする。

でなければ、何もしない (長さ2の基本定理に関する制限)。

ステップ4 $1 \leq i \leq n-j$ および $0 \leq k(i)$ なる全ての i に対して、 D_i^j の要素を $D_{i_1}^j, D_{i_2}^j, \dots, D_{i_{k(i)}}^j$ とおく。

$\{ \sigma_{i_1}^{(j)}, \sigma_{i_2}^{(j)}, \dots, \sigma_{i_{k(i)}}^{(j)} \}$ は、次のような性質を持つ全ての mgu であるとする。

$$L(x) \sigma_{i_q}^{(j)} = L(x) \text{ かつ } \bar{L}_i^j \sigma_{i_q}^{(j)} = D_{i_q}^j \sigma_{i_q}^{(j)}, \quad (1 \leq q \leq k(i)).$$

D_j および E_j を次のように定義する。

$$D_j = \{ D_{i_m}^j \mid \# (\sigma_{i_m}^{(j)} - \{ D_{i_m}^j \text{ に含まれる変数に対する代入要素 } \}) \text{ が全ての } p, q \text{ (} 1 \leq p \leq n-j \text{ かつ}$$

$1 \leq q \leq k(p)$ に対する $\sigma_{pq}^{(j)}$ に対して最小である }。

(直感的には、 D_j に含まれる $D_{i_m}^j$ は IC における単位クローズであって、しかも「最も単純な $mg u \sigma_{i_m}^{(j)} 3$ 」又は「 C_j に対する影響が最小の $mg u \sigma_{i_m}^{(j)}$ 」によって $\overline{L_i^j}$ と統合できる。

$$E_j = \{ \sigma_{i_m}^{(j)} \mid D_{i_m}^{(j)} \mid D \}$$

E_j に含まれる各 $\sigma_{i_m}^{(j)}$ に対して、ステップ 5 を行なう。

ステップ 5 $C_{j+1} = (L(x) L_1^j \cdots L_{i-1}^j L_{i+1}^j \cdots L_{n-j}^j) \sigma_{i_m}^{(j)}$ とおく。

(1) $C_{j+1} = L(x)$ ならば、 $L(x)$ を基本定理とする。

(2) そうでなければ、(a) C_{j+1} の異なる場所に同じアトムが表われなければ、 j を $j+1$ に設定し、ステップ 3 を行なう。(b) そうでなければ、止まる (SL-導出の許容制限による)。

全てのサブプロセスが止まったときに、基本定理の生成プロセスは終了する。

3.3 基本定理の例

(1) 第 3.1 節の IBG からは、第 3.1 節の B_{20} と B_{21} だけが基本定理として生成され得る。

(2) 半順序に対する次の公理⁷⁾からは、基本定理は生成されない。

$$A_1 = \overline{P}(x, y) Q(x, y)$$

$$A_2 = \overline{P}(x, y) \overline{P}(y, z) P(x, z)$$

$$A_3 = \overline{P}(x, y) \overline{Q}(y, z) P(x, z)$$

$$A_4 = \overline{Q}(x, y) \overline{P}(y, z) P(x, z)$$

$$A_5 = \overline{Q}(x, y) \overline{Q}(y, z) Q(x, z)$$

$$A_6 = \overline{Q}(x, y) P(x, y) E(x, y)$$

$$A_7 = \overline{P}(x, x)$$

(3) 次の公理 $GROUP1^{10)}$ からは、2つの基本定理 T_3 , T_7 が生成される。

$GROUP1 = \{ \overline{P}(y, x, u) \overline{P}(x, v, z) \overline{P}(y, z, w) P(u, v, w), P(G(x, y), x, y), P(x, H(x, y), y) \}$ 。

$$T_3 = P(u, z, w) \overline{P}(G(v, w), G(z, v), u)$$

$$T_7 = P(u, z, w) \overline{P}(H(x, u), z, H(x, w))$$

この2つの基本定理は、例えば、 $P(K(x), x, K(x))$ の証明を、行なう際に証明手続の能率を多少損う。

3.4 基本定理の有効性

基本定理の生成に用いられた演算は、公理と導出形との間の導出だけであるので、入力鎖の集合に基本定理を追加しても、その集合の充足性 (satisfiability) に影響を与えない。従って、証明手続の妥当性 (soundness) を損うことはない。

基本定理を有効に利用するために、第 3.1 節の探索戦略 S_2 を次のように変更する：

(S_2') リゾルブできるもとの入力鎖 (即ち、基本定理の個数はかぞえない) の個数が最少のリテラルを選ぶ選出関数。

この変更と S_3 によって、新しい証明手続の探索空間は、第 3.1 節のものより大きくなる。従って証明手続は完全である。しかし、基本定理の追加により、探索木に図 5 のような節 (node) が発生する可能性がある。従って、基本定理の適用が不成功の場合には、基本定理を使用することによって証明手続は、通常の SL-導出よりも能率が悪くなる。

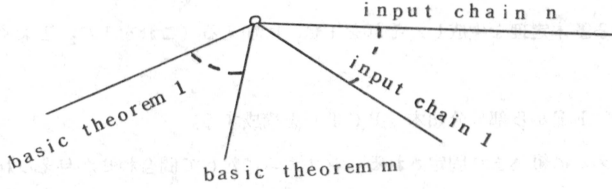


Figure 5: A node in the search tree using
basic theorems

4. IR システムに対する分類木の構成法

一般に IR システムは、大きなデータ・ベースを持っているが、その内包ファイル (IF) は外延ファイル (EF) と比べると、はるかに小さいと考えられる。前章で述べた基本定理生成法を、IR システムに応用する場合には、データ・ベース全体を使うよりは、IF だけに基づいて基本定理を生成する方が、基本定理生成のために、莫大な時間を必要としないで良い。この章では、IF だけに基づいて基本定理は生成されるものと考えられる。

分類木は、SL-derivation に現われるリテラル (例えば, L) の集合から、先頭のリテラルが、 $\bar{L}$ と統合可能な入力鎖の順序集合への関数である。

IF から生成された入力鎖と基本定理および証明すべき定理の否定から成る集合を、 IC とする。 $IC \cup EF$ 上のように順序 $\leq_f$ を定義する。

$\bar{L}_1 C_1, \bar{L}_2 C_2 \in IC \cup EF$ に対して $\bar{L}_1 C_1 \leq_f \bar{L}_2 C_2$ であるのは次の i, ii, 又は iii の 1 つが成立するときに限る。

- (i) $|L_1 C_1| < |L_2 C_2|$ ($|L_i C_i|$ はクローズ $L_i C_i$ の長さを表わす)。
- (ii) $|L_1 C_1| = |L_2 C_2|$ かつ $\bar{L}_1 C_1$ は基本定理ではない。
- (iii) $|L_1 C_1| = |L_2 C_2|$ かつ $\bar{L}_1 C_1$ と $\bar{L}_2 C_2$ が基本定理である。

この順序は、探索戦略 S_3 (第 3.1 節参照)、および基本定理は、入力クローズの中では最も意義が少ないというわれわれの考え方を表わしている。

分類木 f を次のように定義する。

$$f(L) = (\bar{L}_1 C_1, \bar{L}_2 C_2, \dots, \bar{L}_n C_n)$$

ただし、 $\bar{L}_i C_i \leq_f \bar{L}_{i+1} C_{i+1}$ ($i = 1, \dots, n-1$) かつ $\bar{L}_i C_i$ ($i = 1, \dots, n$) はその先頭のリテラル ($\bar{L}_i$) が、 $\bar{L}$ と統合可能な全てのクローズである (ただし $\bar{L}_i C_i \in IC \cup EF$)。

本文で提案される探索戦略は、次のように f に全面的に依存する。導出された鎖 (derived chain) の右端のセルを $K_1, \dots, K_m$ とする (ただし、 K_i はリテラル)。選出関数は、 $f(K_i)$ ($1 \leq i \leq m$) の中でその長さが最短のもの (例えば、 K_j) を選ぶ。その次に S_3 が、導出された鎖に対して $f(K_j)$ の要素を左から右に非決定性に適用する。従って、メモリ能率と探索能率の良い分類木の構成法は、探索戦略作成において本質的に重要である。

Kowalski と Kuehner⁹⁾ は、木表現による分類木の構成を提案した (これが分類木と呼ばれる所以である)。

IRシステムにおける分類木構成法の詳細は未定であるが、大ざっぱには次のように行なう。

ステップ1 内包ファイルから入力鎖の集合を作る（これを IC_1 とおく）。

ステップ2 IC_1 から基本定理を生成し、それを IC_1 に加える（これを IC_2 とおく、即ち、 $IC_2 = IC_1 \cup \{ \text{全ての基本定理} \}$ ）。

ステップ3 $IC_2 \cup EF$ から部分分類木（PCT）を構成する。

このPCTは、IRシステムに組込まれ固定される。システムに対して問合わせが与えられると、問合わせがPCTに追加され完全な分類木が作られる（図2参照）。応答が得られると問合わせは分類木から取除かれる。

5. 結 言

本文で述べたような、基本定理の生成法と使用法が本当に良いか否かは、実験によって確かめなければならないが、その実験はまだ行なわれていない。

システム作成上の最大の問題点は、分類木の構成法である。 QA^{13} 等で用いられている座標索引法（coordinate indexing）が分類木構成の際に利用できると思われる。しかし、現在のところ、著者は座標索引法によってハッシュされたリテラルと大きな外延ファイル（ 10^5 項目以上を含む）とをうまく統合する方法を知らない。

6. 謝 辞

本研究は、著者が1972年9月より1973年6月の間、ハーバード大学Department of Engineering and Applied Physics に留学中に行なったものである。研究を指導して下さい、ハーバード大学Cheatham教授およびWegbreit教授に謝意を表す。

7. 参考文献

- [1] Robinson, J. A., "A review of automatic theorem-proving." Proceedings of Symposia in Applied Mathematics 19, (1967), 1-18.
- [2] Green, C., "Theorem-proving by resolution as a basis for question-answering systems." Machine Intelligence 4, Meltzer, B. and Michie, D. (Eds), American Elsevier Publishing Co., Inc., New York (1969), 183-205.
- [3] Luckham, D., and Nilsson, N. J., "Extracting information from resolution proof trees." Artificial Intelligence 2 (1971), 27-54.
- [4] Darlington, J. L., "Theorem Proving and information retrieval." Machine Intelligence 4, Meltzer, B. and Michie, D. (Eds), American Elsevier Publishing Co., Inc., New York (1969), 173-181.
- [5] Levien, R. E., and Maron, M. E., "A computer system for inference execution and data retrieval." C. Ass. Comput Mach. 10 (1967), 715-721.
- [6] Kuhns, J. L., "Logical aspect of Question-Answering by computer." Software Engineering 2 (1971), Tou, J. T. (Ed), Academic Press.

- [7] Slagle, J. R. and Norton, L. M., "Experiments with an Automatic theorem-prover having partial ordering inference rules." to be appeared in C. Ass. Comput. Mach.
- [8] Hoare, C. A. R., "Proof of a Program: FIND." C. Ass. Comput. Mach. 14 (1971), 39-45.
- [9] Kowalski, R. and Kuehner, D., "Linear resolution with selection function." Artificial Intelligence 2 (1971), 227-260.
- [10] Reboh, R., et al., "Study of automatic theorem-proving programs." Artificial Intelligence Center, TN75, SRI, 1972.
- [11] Michie, D., et al., "G-deduction." Machine Intelligence 7, Meltzer, B. and Michie, D. (Eds), John Wiley & Sons, Inc., New York (1972), 141-165.
- [12] Loveland, D. W., "A simplified format for the model-elimination theorem-proving procedure." J. ACM 16 (July 1969), 349-363.
- [13] Rulifson, J. F., et al., "QA4: A procedural calculus for intuitive reasoning." Artificial Intelligence Center, TN73, SRI, 1972.

本 PDF ファイルは 1965 年発行の「第 6 回プログラミング・シンポジウム報告集」をスキャンし、項目ごとに整理して、情報処理学会電子図書館「情報学広場」に掲載するものです。

この出版物は情報処理学会への著作権譲渡がなされていませんが、情報処理学会公式 Web サイトの https://www.ipsj.or.jp/topics/Past_reports.html に下記「過去のプログラミング・シンポジウム報告集の利用許諾について」を掲載して、権利者の検索をおこないました。そのうえで同意をいただいたもの、お申し出のなかったものを掲載しています。

過去のプログラミング・シンポジウム報告集の利用許諾について

情報処理学会発行の出版物著作権は平成 12 年から情報処理学会著作権規程に従い、学会に帰属することになっています。

プログラミング・シンポジウムの報告集は、情報処理学会と設立の事情が異なるため、この改訂がシンポジウム内部で徹底しておらず、情報処理学会の他の出版物が情報学広場(=情報処理学会電子図書館)で公開されているにも拘らず、古い報告集には公開されていないものが少なからずありました。

プログラミング・シンポジウムは昭和 59 年に情報処理学会の一部門になりましたが、それ以前の報告集も含め、この度学会の他の出版物と同様の扱いにしたいと考えます。過去のすべての報告集の論文について、著作権者(論文を執筆された故人の相続人)を探し出して利用許諾に関する同意を頂くことは困難ですので、一定期間の権利者搜索の努力をしたうえで、著作権者が見つからない場合も論文を情報学広場に掲載させていただきたいと思います。その後、著作権者が発見され、情報学広場への掲載の継続に同意が得られなかった場合には、当該論文については、掲載を停止致します。

この措置にご意見のある方は、プログラミング・シンポジウムの辻尚史運営委員長(tsuji@math.s.chiba-u.ac.jp)までお申し出ください。

加えて、著作権者について情報をお持ちの方は事務局まで情報をお寄せくださいますようお願い申し上げます。

期間：2020 年 12 月 18 日～2021 年 3 月 19 日

掲載日：2020 年 12 月 18 日

プログラミング・シンポジウム委員会

情報処理学会著作権規程

<https://www.ipsj.or.jp/copyright/ronbun/copyright.html>