人 工 知 能 72－7
（1990．9．12）

# 論理プログラミング環境におけるEBLの有効性計算

山田誠二　辻三郎
yamada@keg.ce.osaka-u.ac.jp
大阪大学・基礎工学部

説明に基づく学習：EBLにおいて，効率化は学習の本質である．EBLでは，効率化の条件として，操作性規範が設定されるが，それにより必ずしも効率化が保証されていない．よって，これまでは学習後に試験例を大量に与える実験的手法で，効率化の検証が行われてきた．しかし，実験的手法は，計算コストが非常にかかり，実際に学習を行った後にEBLの有効性が初めてわかるという欠点があった．

本報告では，訓練例のみを領域理論だけで問題解決することによって，EBLが有効か否かを解析的に計算する手法を示し，いくつかの具体例による検証を行う．本手法では，実際に一般化・学習を行うことなく，PrologにおけるEBLの有効性を計算可能である．

# Computing the Utility of EBL in a Logic Programmming Environment

Seiji Yamada and Saburo Tsuji
Department of Control Engineering, Osaka University

In the last several years, a lot of research on EBL (Explanation-Based Learning) has been done, and some methods have been developed for coping with the problems in EBL. Since EBL does not always make problem solving more efficient, we need to determine whether we should use EBL by evaluating its utility. However, in most EBL systems, the utility is verified only by experimental results. Because the experiments in which a learned EBL system solves test examples cost a great deal, the evaluation of the utility is expensive. In this paper, we propose a framework for analyzing the EBL paradigm and a concrete method for computing the utility of EBL. As a result, the utility of EBL can be computed without the experiments in which a learned system actually solves the test example set.

## 1. はじめに

説明に基づく学習：EBL (Explanation-Based Learning) [Mitchell86][DeJong86] では，これまで解決できなかった問題が解決できるようになるという意味での，新しい知識は獲得されない．むしろ，学習前でも解決できた問題を，より効率よく解決できるようにすることにEBLの学習の本質があるといえる．類似に基づく一般化：SBL(Similarity-Based Learning)においては，学習前では解決不可能であった問題が，学習により解決可能になる．これは，効率の観点からみれば，「解決不可能＝効率最悪の状態」から「解決可能＝効率最悪でない状態」になることであり，その効率向上は原理的に保証されていると言える．これに対しEBLは，学習前が効率最悪の状態でないため，学習後の効率向上が原理的に保証されず，このことがEBLにおいて効率向上が最重要視される所以と考えられる．なお，EBLにより効率が向上する性質を，EBLの有効性と呼ぶことにする．

有効性の保証に関して，EBLでは入力の一つである操作性規範(operationality criterion)を満たす概念記述を学習するころとにより，その有効性が保証されるとしている．しかし，本報告中でも示すように，学習フェーズで用いられる操作性規範 [Mitchell86][DeJong86] が満たされても，効率向上が保証されない場合がある [Minton90][Mooney89][Yamada89]．直観的にもわかるように，操作性規範は，領域理論，問題解決フェーズ，訓練例などに依存している．よって，種々の研究者により，種々の分野で適切な操作性規範が，経験的・直観的に提案されているのが現状である．

与えらた操作性規範により，有効性が保証されない以上，なんらかの方法で，現入力(操作性規範も含む)でEBLにより，効率向上が実現できるか否かを判定することが重要になる．EBLの有効性の検証は，これまで，まずEBLシステムに訓練例を与え学習をさせ，次に学習後のEBLシステムに，大量の試験例を与え，問題解決の効率向上を実験的に調べることにより行われてきた．しかし，この手法では試験例を解決するための計算コストが大きし，実際にEBLを適用してみて初めてわかるという欠点がある．よって，本報告では，論理プログラミングで実現されたEBLにおいて，EBLの諸入力が与えられたとき，訓練例だけを問題解決するだけで，学習を行わずに，EBLの有効性を解析的に計算する手法を示す [山川90]．この方法により，実際にEBLを適用する前に，EBLシステムを用いるべきか否かの判断が可能である．

本報告の構成としては，まず，学習フェーズ，問題解決フェーズを包括した標準EBL手続き：SEPが設定され，この枠組みに基づいて，有限試験例系列及び無限試験例系列に対してのEBL有効性の計算法を示す．そして，最後に本手法をいくつかの具体例で検証する．

## 2. EBL実装言語としてのProlog

本研究では，最も一般的な論理プログラミング言語の一つであるPrologにより実装されたEBLを対象とする．実装言語として，Prologを選んだ理由を以下に示す．

1) 現在，EBLでは，証明木が，説明構造(explanation structure)として使われている．Prologを用いると説明構造の生成が容易であり，また，領域理論がホーン節で記述し易い．

2) 説明に基づく一般化：EBG (Explanation-Based Generalization)[Michell86][DeJong86] が，Prologの単一化(unification)を用いて簡単に実現できる．

3)1)2)の理由により，PrologがEBL実装のための言語として最も広く用いられている [Kedar-Cabelli 87][Hirsh87][Prieditis87]．

4)Prologの基本的手続きであるSLD導出が単純でわかりやすい．

## 3. 標準EBL手続き：SEP

EBL有効性を計算するには，具体的手続きの計算コストの算出が必要なので，EBLシステムにおける問題解決，一般化などの諸手続きの記述が必要となる．また，特殊な手続きに関して，有効性を計算しても一般性がないので，できるだけ標準的なEBLの処理手続きを設定し，それについて解析を進めて行く必要がある．標準的な手続きは，PrologによるEBLにおいては比較的容易に設定可能である．

ここで設定される標準EBL手続き：SEPは，学習フェーズ及び問題解決フェーズを統合したEBLシステム全体の手続きである．通常，EBLシステムでは，学習フェーズの記述に重点が置かれ，問題解決フェーズの記述が明確に与えられることは少ない．しかし，EBLの本質である「効率化」は，具体的な問題解決フェーズの記述(例えば，どのような探索戦略を用いているのか)に強く依存するため，EBLシステムの性能を考

える際，問題解決フェーズの記述が，必須となることが指摘されている[Keller87]．そのため，SEPは問題解決フェーズも包括したモデルとなっている．

### 3.1 SEPの入出力

(A)入力

1) 目標概念 GC：学習すべき概念を表わす述語を頭(head)に持つ確定節集合．その頭をGCHと呼ぶ．GCの本体には，少なくとも一つの操作不可能な述語を含む．

2) 領域理論 DT：訓練例が，なぜ目標概念の肯定例なのかを証明するための確定節集合．その本体(body)中に，カット，負のリテラルを含まないとする．

3) 例Eと目標Tの系列 TES：Eは，訓練例及び試験例からなる例を記述する事実節集合．例，目標概念，領域理論を用いて証明すべき目標がTで，目標概念の頭を例示化(instanciate)した事実節である．$TE_n=[T_n,E_n]$, $TES = [TE_1,..,TE_n] = [[T_1,E_1] ,..., [T_n ,E_n]]$ とする．

4) 操作性規範 OC：EBLにより効率が向上するために満たすべき条件．典型的には，説明構造のどの部分木で概念記述を生成すべきかの基準であり，その部分木の葉になるべき述語で表わされる．

以上は，従来のEBLの入力である．そして，以下が，学習フェーズ及び問題解決フェーズにおける各種手続きの記述である．

5) 説明構造生成手続き $GE(TE_i)$：目標 $T_i$ を例 $E_i$ と領域理論DTを用いて証明することにより，説明構造を生成する手続き．

6) 一般化手続き $G(ES_i)$：得られた説明構造 $ES_i$ を一般化し，(操作性規範を満たす)概念記述 $OCD_i$ を生成する手続き．

7) 知識管理手続き $M(OCD_i)$：得られた $OCD_i$ を現在の知識ベースに追加し，さらに必要な場合は，適当な知識ベースの更新を行なう手続き．

8) 問題解決手続き PS

- $PS(TE_i,DT)$：学習なし問題解決システムが，領域理論と例 $E_i$ だけで，目標 $T_i$ の証明を行なう手続き．
- $PS(TE_i ,OCD_{j-k})$：例系列 $\{TE_j,.., TE_k\}$ により，学習された概念記述 $\{OCD_j,..,OCD_k\}$ と例 $E_i$ だけで，目標 $T_i$ の証明を行なう手続き．

(B)出力

1) 操作可能な概念記述集合 OCDS：操作性規範を満たし，GHCを頭としてもつ確定節の系列．

### 3.2 SEP及びSNPアルゴリズム

以下に，SEPのアルゴリズムを示す．

```
<SEP アルゴリズム >
TES=[[T1,E1],...,[Tn,En]] とする.
procedure SEP(TES)
begin
  i ← 0 ;
  while not(i=n) do
    begin
      i ← i+1;
      if PS(TEi,OCDi-,i-1) fails then
        begin
          GE(TEi);
          G(ESi);
          M(OCDi);
        end
    end
end
```

同様に，標準学習なし手続き:SNPも定義する．SNPの入力は，SNPの入力から，$PS(TE_i,OCD_{j-k})$ を取り除いたものである．

```
<SNP アルゴリズム >
TES=[[T1,E1],...,[Tn,En]] とする.
procedure SNP(TES)
begin
  i ← 0 ;
  while (i=n) do
    begin
      i ← i+1;
      PS(Ei,DT);
    end
end
```

### 3.3 Prologによる手続きの特殊化

Prolog環境下という前提から，入力の一部は，以下のように設定されるのが妥当である．

4) 操作性規範 OC：事実節の述語を操作可能な述語とし，それ以外を操作不可能な述語とする．

5) $GE(TE_i)$ 及び $PS(TE_i,DT)$：$DT \cup GC \cup E_i \cup \{\leftarrow T_i\}$ で，$\leftarrow T_i$ を目標節としてSLD導出を行ない，説明構

造生成.

7) 知識管理手続き (M($OCD_i$) : 一般化手続きにより得られた $OCD_i$ を OCD の最後に追加.

8)PS($TE_i$, $OCD_{j-k}$) : {$OCD_j$,.., $OCD_k$} ∪ GC ∪ $E_i$ ∪ { ← $T_i$} で, ← $T_i$ を目標節として SLD 導出を行なう.

上記の M($OCD_i$) の設定に関しては, Prolog に由来するものではないが, 一般的に有効であることが報告されており [Shavlik88][Mooney89], 標準的と考える. SEP は, [Mooney89] の EGGS における手続きとほぼ同じである.

## 4. 用語説明

- EBL有効性 : SEP と SNP に, 同一の領域知識と例系列を与えときの, それぞれの全体の計算コストを LC, NLC とする. このとき, EBL の有効性 LE を, LE=(NLC − LC)/NLC と定義する. この式において, LE の値が, 1 に近いほど, 学習による効率向上が大きく, 反対に負であれば, 学習によって効率が低下することになる. よって, LE>0 が, 学習が有効である条件となる.
- SLD木 : SLD導出において, 導出により導かれた目標節に対応した節点で構成された木構造. 節点は選択基本式 (selected atom) で特徴づけられ, SLD導出はこの SLD 木を縦型探索する.
- 成功点 (success node): 空節を指示する節点.
- 失敗点 (failure node): その選択基本式と単一化可能な頭を持つ確定節がない節点.
- 路(path): SLD 木の根から葉までの節点と枝の系列. 葉が成功点のとき, 成功路といい, 葉が失敗点のとき, 失敗路という.

(以上の詳細は, [野口86] などを参照)

- SLD説明構造 : 例 TE の SLD木において, 最初に見つかる成功路. ただし, 節点の系列で表わされる. SLD 木中で, 操作性規範で示された操作可能な述語の節点を「操作可能節点」, それ以外を「操作不可能節点」という.
- SLD 経路 : 例 TE の SLD 木で, SLD 説明構造を見つけるまでに辿った節点の系列.

(以上の具体例は, Fig.2 を参照)

## 5. いくつかの仮定と計算コストの設定

### 5.1 例に関する仮定

解析的手法においては, 数学的扱いやすさ (tractability) のために, いくつかの拘束条件あるいは仮定を設ける必要がある. もちろん, それらは現実的に妥当なものでなければならない. 以下では, 本手法で用いられる仮定とその妥当性について述べていく.

通常, 学習システムのパフォーマンスの評価は, まず学習するための訓練例 (training example) が与えられ, 次にパフォーマンス向上を調べるために, 新たな学習がされない例, つまり訓練例で学習された概念の肯定例である試験例 (test example) が与えられる [Minton90]. 訓練例と試験例があらかじめ分類されて与えられているのが, 一般的である以上, 学習システムの評価において, 訓練例は, 事前にわかっていると仮定してよいと考えられる. よって, 以下の (A1) を仮定する.

(A1) 訓練例系列に関する仮定

訓練例系列 [$TE_1$,..,$TE_m$] は, 目標概念の肯定例 E と目標 T の系列であり, その要素についての一般化された SLD 説明構造は, どれも同一ではないとする. つまり, 学習後には, 同じ頭をもち本体の異なった m 個の概念記述 $OCD_1$,..,$OCD_m$ が得られる.

次に, 試験例について考える. 実際に試験例を与えることなく, EBL 有効性を計算するためには, 試験例の分布に関して何らかの仮定を立てることが必要になる. この試験例に関する仮定として最も妥当なものは何だろうか? 従来の学習システムでは, パフォーマンスの評価の際, 試験例をランダムに生成して与えている [Minton 90]. つまり, 学習された概念記述それぞれに対して, 試験例が偏りなく与えられている. よって, 「試験例は学習された概念記述それぞれに対し, 同じ個数づつ与えられる」と近似することは, 妥当であり, 以下の (A2) を仮定できる.

(A2) 試験例に関する仮定

試験例系列 [$TE_{m+1}$,...,$TE_n$] は, 訓練例系列による学習後, 学習器に与えられる目標概念の肯定例 E と目標 T の系列である. この系列は, 学習された OCD それぞれについての g 個の肯定例から構成される. よって, n=(g+1)m が成り立つ.

実際には, まず訓練例系列が, 次に試験例系列が学習器に与えられる. つまり, 例系列 TE 全体は [$TE_1$,..,$TE_m$,$TE_{m+1}$,..,$TE_n$] となり, 先頭要素から順に学習システムに与えられる.

さらに, EBL 有効性解析の初歩段階として, 扱い易さのために, 以下のことを仮定する.

(A3) 単一の目標概念

一つだけの目標概念の概念記述を学習すると仮定する. ただし, 概念記述は, 複数得られてもかまわない.

### 5.2 計算コストの設定

Prologの計算コストを設定する．Prologの計算コストの算出法には，正確で確立されてものはなく[近山90][梅村90]，通常，LIPS(Logical Inference Per Second)[Moto-oka81]が，Prologの計算速度の評価基準として，広く用いられている．そこで本論文では，LIPSの計算コストであるLI(Logical Inference)を用いることにする．LIは，推論の回数，正確にはSLD導出の回数のことであり，実際の処理系においては，述語の呼び出し回数と考えらる[梅村90]．

SLD木上で，ある成功路を見つけるまでのLIは，前述のSLD経路の要素数に等しい．よって，SLD説明構造を見つけるまでの，SLD経路がわかれば，その要素数から簡単に計算コストが算出できる．また，証明の成否によらず，コストは一定とする．

ただし，LIには，•バックトラックのコスト，•複雑な構造体の単一化のコスト，などが考慮されていない[梅村90][近山90]ため，厳密な計算コストは算出できない．特に，インデキシングしにくい節が大量にある場合や引数の数が多いとき，単純な節の場合のLIのコストとの差が，大きくなる可能性がある．しかし，EBLでは，生成される節(概念記述)の引数の数は増えることはなく，また不必要な節が大量に生成されることは，操作性規範により抑制される．よって，本研究では，大間かな計算コストとして，LIを用いることにした．

## 6. 有限試験例系列の有効性計算

本章では，訓練例系列のみを領域知識で証明した結果であるSLD木集合だけを用いて，実際に一般化•学習することなしに，有限試験例系列に対するEBLの有効性計算が可能であることを示す．

TES=[$TE_1$,..,$TE_m$,$TE_{m+1}$,..,$TE_n$]

[$TE_1$,..,$TE_m$]: 訓練例系列

[$TE_{m+1}$,..,$TE_n$]: 試験例系列(n=(g+1)m)

上記の例系列が与えられたとき，SEPとSNPそれぞれの計算コストLCとNLCおよびEBL有効性LEは，以下のようになる．

$$LC=\sum_{i=1}^{m} C(PS(TE_i,OCD_{1,..,i-1}) + GE(TE_i)+G(ES_i)+M(OCD_i)) + g\sum_{i=1}^{m} C(PS(TE_i,OCD_{1,..,m})) \quad (1)$$

$$= \sum_{i=1}^{m}\{(i-1)+\sum_{j=1}^{i-1} ON_j + N_i + G(ES_i)+M(OCD_i)\} + g\sum_{i=1}^{m}(i+\sum_{j=1}^{i} ON_j) \quad (2)$$

$$= \sum_{i=1}^{m}\{(i-1)+(m-i)ON_i+N_i+C(G(ES_i)+M(OCD_i)) + gi+g(m-i+1)ON_i\} \quad (3)$$

$$NLC=\sum_{i=1}^{n} C(PS(TE_i,DT)) = (g+1)\sum_{i=1}^{m} C(PS(E_i,DT)) = (g+1)\sum_{i=1}^{m} N_i \quad (4)$$

$$LE = 1-\frac{LC}{NLC} = 1 - \frac{\sum_{i=1}^{m} C(PS(TE_i,OCD_{1,..,i-1})+GE(TE_i)+G(ES_i)+M(OCD_i)))}{(g+1)\sum_{i=1}^{m} C(PS(TE_i,DT))} - \frac{g\sum_{i=1}^{m} C(PS(TE_i,OCD_{1,..,m}))}{(g+1)\sum_{i=1}^{m} C(PS(TE_i,DT))} \quad (6)$$

$$= 1 - \frac{\sum_{i=1}^{m}\{i-1+(m-i)ON_i+N_i+C(G(ES_i)+M(OCD_i))\}}{(g+1)\sum_{i=1}^{m} N_i} - \frac{\sum_{i=1}^{m}\{gi+g(m-i+1)ON_i\}}{(g+1)\sum_{i=1}^{m} N_i} \quad (7)$$

ただし，

•C(P)は，手続きPにかかる計算コストを返す関数である．

•$ON_i$は，PS($TE_i$,DT)のSLD説明構造中の操作可能節点数．これは，$OCD_i$の本体の述語数に等しい．

•$N_i$は，PS($TE_i$,DT)のSLD説明経路の全節点数．よって，C(PS($TE_i$,DT))=$N_i$．

各式を説明していく．前述の計算コストの定義より，SLD木上で手続きPが辿った節点数が計算コストに等しい．

•式(1): LCでは，訓練例$TE_1$~$TE_m$までは，それぞれの$TE_i$について，それまでに学習された$OCD_1$,..,$OCD_{i-1}$による証明: PS($TE_i$,$OCD_{1,..,i-1}$)を行うが，(A1)により必ず失敗し，GE($TE_i$),G($ES_i$),M($OCD_i$)により$OCD_i$を学習することをm回繰り返す．試験例$TE_{m+1}$~$TE_n$では，学習で得られた$OCD_1$~$OCD_m$で$TE_1$~$TE_m$を1回づつ証明すること:PS($TE_i$,$OCD_{1,..,m}$)をg回繰り返す．

•式(2): この式では，C(PS($TE_i$,$OCD_{1,..,i-1}$)), C(PS($TE_i$,DT)), C(PS($TE_i$,$OCD_{1,..,m}$))が展開されている．

a)C(PS($TE_i$,$OCD_{1,..,i-1}$))の展開: Fig.1は，$TE_i$を証明したときの知識ベースのSLD木を表わしている．GCHは，目標概念の頭である．$TE_i$を証明する際，$OCD_1$~$OCD_{i-1}$を用いるが，<A1>より，この証明はすべて失敗する．これにかかるコストが，式(1)のC(PS($TE_i$,$OCD_{1,..,i-1}$))である．これをFCとする．重要な点は，•一般化されたOCDが得

られないとFCは決定できないこと，•FCは例に依存すること，である．よって，FCを正確に算出するには，新たな制限を加えなければならない．筆者は，それらの制限は妥当なものではないと言う立場をとり，ここでは制限を加えず，FCに関しては，入力が未知な場合の標準的手法である，最悪の場合の解析(worst- case analysis)[Aho74]を行う．FCにおける最悪の場合とは，$OCD_1$~$OCD_{i-1}$の探索において，それぞれのOCDの本体中のすべての述語が調べられる場合，つまり最後の述語で失敗する場合である．よって，$C(PS(TE_i,OCD_{1\cdots i-1}))$は，$OCD_1$~$OCD_{i-1}$をすべて辿った場合の節点数となり，$(i-1)+\sum_{j=1}^{i-1}ON_j$と表わされる．ここで，$(i-1)$は$OCD_1$~$OCD_{i-1}$の頭の呼び出し回数で，$\sum_{j=1}^{i-1}ON_j$は本体の述語数の合計である．なお，最悪の場合の解析の妥当性については，9章で触れる．

b)$C(PS(TE_i,DT))$の展開：OCDによる証明が失敗すると，次はDTによる証明が行なわれ，この証明のコストが，$C(PS(TE_i,DT))$である．これは，前述の$N_i$の定義より，$C(PS(TE_i,DT))=N_i$と展開される．

c)$C(PS(TEi,OCD_{1\cdots m}))$の展開：これは，学習済みのOCDで試験例を証明するコストである．$OCD_i$の肯定例の証明では，$OCD_1$~$OCD_{i-1}$の証明を失敗した後に，$OCD_i$で成功する．この$OCD_1$~$OCD_{i-1}$での失敗分のコストも，a)のFCと同様に，決定できない．よって，同様の理由により，最悪の場合の解析を行う．その結果が，$i+\sum_{j=1}^{i}ON_j$である．i及び$\sum_{j=1}^{i}ON_j$は，それぞれ$OCD_1$~$OCD_i$の頭の呼び出し回数と本体の述語の呼び出し回数である．

•式(3)：式(2)のシグマを外した式．

NLC，LEについては，説明は割愛する．

LEより明らかなように，有限例系列に対する標準EBL手続きの有効性は，パラメータ，$N_i$,$ON_i$,$C(G(ES_i))$,$C(M(OCD_i))$により，実際に学習(一般化)することなく計算できる．$N_i$，$ON_i$は，領域理論による証明過程から，算出できる．しかし，$C(G(ES_i))$，$C(M(OCD_i))$は，現状ではコスト計算ができないので，LEを算出することはできない．次章で，現在計算可能なLEの特殊な場合について述べる．

## 7. 無限例系列の有効性計算

従来のEBL研究では，有限の試験例系列を対象とし，その例系列について学習による効率向上•一般性の維持を図るということをやってきた．([Keller87]において顕著)しかし，例えば，学習後，システムが十分に長い期間，使用されるなど，実際には，十分な量の試験例系列が与えられるとする方が妥当な場合も多い．この十分に大量の(つまり，無限大の)試験例系列が与えらえた場合のEBLの有効性は，これまで実験的にしか検証されていない．また，その検証は，学習後のEBLシステムに，大量の試験例系列を与え，実際に問題解決をさせた実験結果によるものである．当然のことながら，この検証には多量の計算コストがかかる．

ところが，前章で求めた有限例系列のEBL有効性LEを使うと，無限試験例系列に対する有効性を算出できる．試験例が十分に多いということは，TEnのnを無限大にすることに対応する．n=(g+1)mで，mは有限なので，$n\to\infty$は，$g\to\infty$とすることに等しい．よって，LEにおいて$g\to\infty$とすることにより，無限試験例系列のEBL有効性:ILEを求めることができる．

以下に，ILEの式を示す．

$$ILE=\lim_{g\to\infty} LE$$

$$=1-\frac{\sum_{i=1}^{m} C(PS(TE_i,OCD_{1\cdots m}))}{\sum_{i=1}^{m} C(PS(TE_i,DT))} \quad (8)$$

$$=1-\frac{\sum_{i=1}^{m}\{i+(m-i+1)ON_i\}}{\sum_{i=1}^{m} N_i} \quad (9)$$

こうして求められたILEは，パラメータとして，m,$ON_i$,$N_i$だけをもつ．mは，入力として与えられる．

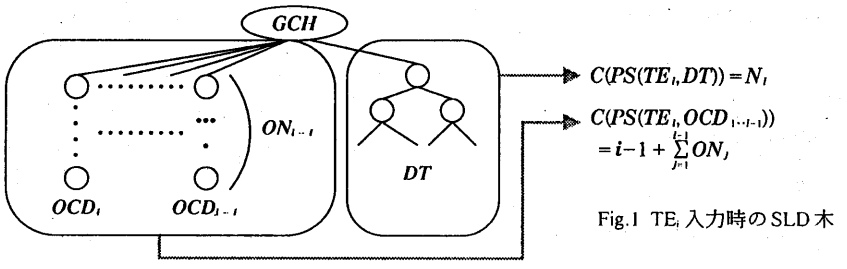



Fig.1 $TE_i$入力時のSLD木

また，$ON_i$は，領域理論を用いて訓練例系列のみを証明したSLD説明構造中の操作可能節点数であるし，$N_i$はその説明経路中の節点数である．よって，ILEのすべてのパラメータは，領域理論で訓練例のみを証明するだけで算出可能であり，それによりILEを計算できる．ここで重要なのは，ILEの計算において，一般化や操作可能な概念記述の生成などの学習フェーズの処理がまったく必要ないことである．

また，$M(OCD_i)$が変更されない限り，$\sum_{i=1}^{m} C(PS(TE_i, OCD_{1\sim m}))$, $\sum_{i=1}^{m} C(PS(TE_i, DT))$は不変なので，$GE(TE_i)$及び$G(ES_i)$をいくら改良したところで，ILEには影響がないことがわかる．

訓練例と領域知識からILEが算出でき，そのILEの値が正であれば，EBLは有効であり，使うべきであるという判断が可能である．しかし，ここで忘れてはならないのが，LE，ILEはともにEBLにとって最悪の場合の評価だったことである．このことは，例えLE，ILEが負の値をとっても，実際の実験ではEBLが有効である場合が生じることを意味している．よって，ILE，LEの扱い方としては，それが正の値をとれば，EBL有効性は必ず保証されるという十分条件として扱うべきである．

次章以降では，計算可能なILEをいくつかの具体例において，算出していく．

## 8. 実験

本章では，有効性計算の実験として，いくつかの入力に対し，ILEを計算し，実験値との比較を行う．

• 実験値の求め方

ILEの実験値を求めるために，実際にEBLシステムをインプリメントした．$G(ES_i)$として，Prolog-EBG [Kedar-Cabelli87]を用いた．訓練例による学習後，訓練例と同じ試験例をそれぞれのOCDについて5000個あたえて(つまり，g=5000)，それらをDTだけで証明した場合と$OCD_1$~$OCD_m$で証明した場合のCPUタイムを式(8)に代入して，ILEの実験値を求めた．

• 実験1 : SAFE-TO-STACK

まず最初の具体例として，[Mitchell86]のSAFE-TO-STACK (X,Y)で，ILEの計算を行なってみる．Fig.2に入力,SLD説明構造,SLD説明経路を示す．ここでは，訓練例が$TE_1$ひとつの場合と，$[TE_1,TE_2]$の2つが与えられる場合を考える．$TE_1$，$TE_2$のSLD説明構造は，図中の(A),(B)である．(A)では，対象の質量を体積と密度の積から，(B)では，デフォルト値から求めている．

• 訓練例系列 :$[T_1]$の場合 :(A)から，N1=11，ON1=7が求まる．よって，ILE=1 − (8/11)=0.27となり，EBL有効性は，保証される．

• 訓練例系列 :$[T_1,T_2]$の場合 : さらに，T2も与え，(B)より，N2=11，ON2=5となる．よって，ILE=1 − (22/22)=0となり，EBL有効性は保証されず，最悪の場合無効となる．

理論値と実験値を以下に示す．

| 訓練例 | 理論値 | 実験値 |
|---|---|---|
| [TE1] | 0. 27 | 0. 20 |
| [TE1, TE2] | 0 | 0. 04 |

• 実験2 : いくつかの木構造

Fig.3のような，$TE_1$~$TE_3$に対応して3つの異なった説明木が，生成される領域理論を与えた．それぞれの証明木は，$TE_1$では節点数27の一本の木になり，$TE_2$，$TE_3$では節点数31のbranching factorが2の木になる．葉と途中節点の数から，直観的に，$TE_1$ではEBLによりかなりの効率向上が，予想され，$TE_2$,$TE_3$では余り向上しないことが考えられる．ここでは，$[TE_1]$,$[TE_1,TE_2]$,$[TE_1,TE_2,TE_3]$の3種類の訓練例について調べた．その結果を以下に示す．

| 訓練例 | 理論値 | 実験値 |
|---|---|---|
| [TE1] | 0. 95 | 0. 92 |
| [TE1, TE2] | 0. 71 | 0. 66 |
| [TE1,TE2,TE3] | 0. 49 | 0. 59 |

ILEの理論値より，これらの訓練例においては，EBL有効性は保証される．しかし，その効率は訓練例が増えるに従い，徐々に低下する．この傾向が，理論値で確認されている．

• 実験3 : member

memberのような再帰ルールを含む場合，直観的にEBLを行うことは，効率低下につながると考えられる．Fig.4のような領域理論，訓練例で実験した．結果は，以下の通りである．

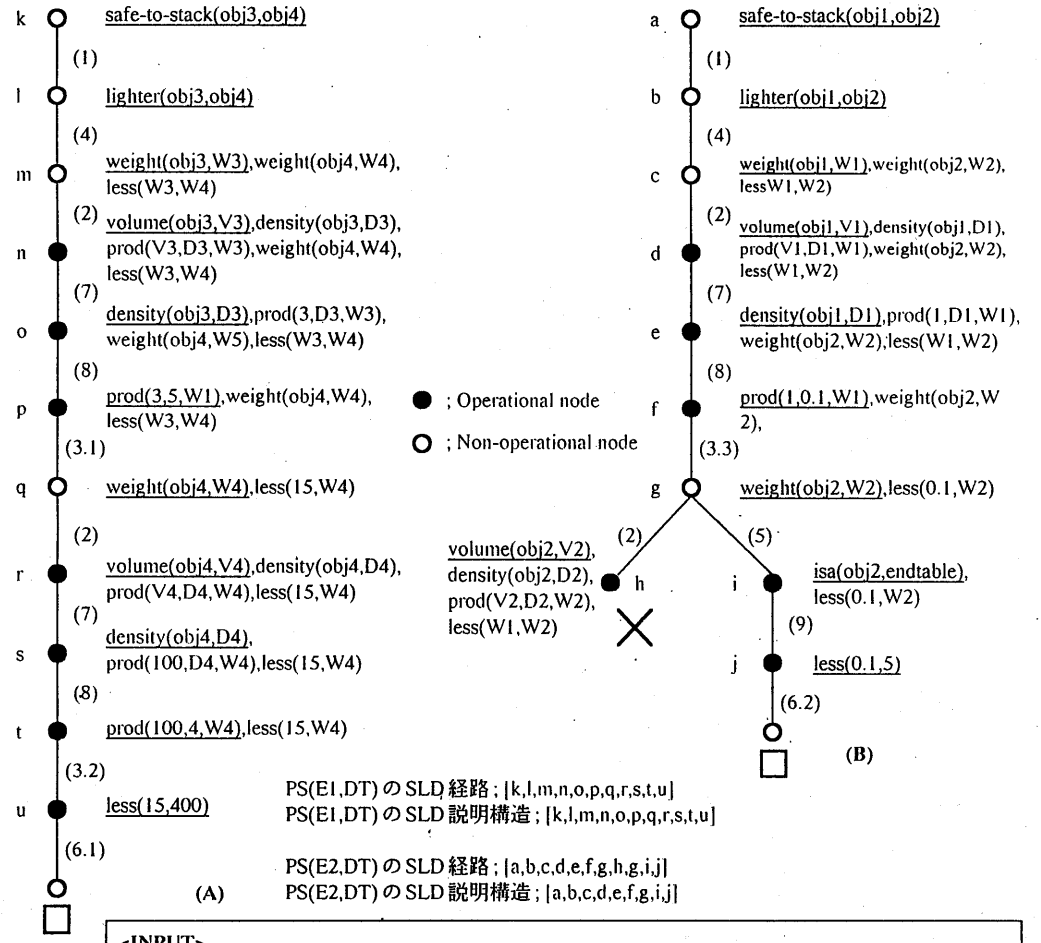
| 訓練例 | 理論値 | 実験値 |
|---|---|---|
| [TE1,TE2,TE3] | − 1.0 | − 0.43 |

memberでは理論値より，EBL有効性は保証されず，最悪の場合かなりの効率低下が生じる．実際，実験値では効率が低下している．

以上の例は，し意的なものである．どのような領域理論と訓練例で実験を行えばよいのかは難しい問題であり，今後の課題である．

## 9. 最悪ケース解析の妥当性

6章において，OCDによる証明がすべて失敗する手続きの計算コストFCの算出で，最悪の場合の解析を行った．まず，なぜこのFCが，正確に求められないの




PS(E1,DT) のSLD経路 ; [k,l,m,n,o,p,q,r,s,t,u]
PS(E1,DT) のSLD説明構造 ; [k,l,m,n,o,p,q,r,s,t,u]

PS(E2,DT) のSLD経路 ; [a,b,c,d,e,f,g,h,g,i,j]
PS(E2,DT) のSLD説明構造 ; [a,b,c,d,e,f,g,i,j]

```
<INPUT>
GC; safe-to-stack(X,Y): − lighter(X,Y). (1)
DT; weight(P1,W): − volume(P1,V1),density(P1,D1),prod(V1,D1,W). (2)
    prod(3,5,15). (3.1)   prod(100,4,400). (3.2)   prod(1,0.1,0,.1). (3.3)
    lighter(P1,P2): − weight(P1,W1),weight(P2,W2),less(W1,W2). (4)
    weight(P1,5): − isa(P1,endtable). (5)   less(15,400). (6.1)   less(0.1,5). (6.2)
OP; on(_,_),isa(_,_),color(_,_),volume(_,_),prod(_,_,_),less(_,_),density(_,_)
TE; T1=safe-to-stack(obj3,obj4), E1=[volume(obj3,3)(10),density(obj3,5)(11),volume(obj4,100)(12),
                                    density(obj4,4)(13)]
    T2=safe-to-stack(obj1,obj2), E2=[volume(obj1,1)(7),density(obj1,0.1)(8),isa(obj2,endtable)(9)],
<OUTPUT>
  T1 : safe-to-stack(X,Y): − volume(X,V1),density(X,D1),prod(V1,D1,W1),
                             volume(X,V2),density(X,D2),prod(V2,D2,W2),less(W1,W2).
  T2 : safe-to-stack(X,Y): − volume(X,V1),density(X,D1),prod(V1,D1,W1),isa(Y,endtable), less(W1,5).
```

Fig.2 SAFE-TO-STACK(X,Y)

かを説明する．考えられる理由は，以下の2つである．

1) 一般化されたOCDがないとFCは決定不能:FCは，あるOCDがその概念記述の否定例を証明しようとして失敗したときの計算コストである．よって，FCは，OCDの本体のリテラルのうち，否定例で真になるものと偽であるものを判別できなければ，計算できない．訓練例のSLD木からは，この判別はできない．また，EBGは，いくつかのバリエーションがあり [Mitchell87][DeJong87][Mooney86]，異なった手法を用いると同じ説明構造から，異なったOCDが生成される．よって，実際に一般化を行って，OCDを生成してみなければ，FCは正確にはわからない．

2) FCは，例に依存する:また，仮に，このFCが求まったとしても，それは訓練例に対するFCの値である．訓練例のFCを用いて，試験例のFCを計算するには，例えば，訓練例のFCと試験例のFCは同じであるというような仮定，つまりOCD$_i$の肯定例である試験例が，OCD$_i$以外のOCDの本体のリテラルをどの程度満たすかに関する仮定を設けねばならない．しかし，このような仮定は，(A1)~(A3)よりもはるかに拘束のきついものであり，望ましくない．

以上の理由で，本研究では，最悪の場合の解析を行い，EBL有効性を保証する条件を求めるのみにとどめた．ただし，あるOCDの肯定例において，別のOCDの本体中の必ず真になるリテラルは，それぞれのSLD木の共通部分を調べることでわかると考えられる．これにより，EBLの最良の場合の解析が，実現できる可能性があることを付け加えておく．また，確率論的な扱い [Greiner89] も考えられる．

```
GC, DT:
member(X, [X|_]).
member(X, [Y|Z]): - member(X, Z).

TE:
TE1 : [member(a, [a,b,c]), []].
TE2 : [member(b, [a,b,c]), []].
TE3 : [member(c, [a,b,c]), []].
```

Fig.4 memberの入力

## 10. まとめと今後の課題

本論文では，これまで実験的にしか検証されていなかったEBLの有効性を，解析的手法で調べる方法を示した．その結果，大量の試験例を与えなくても，訓練例の問題解決を行なうだけで，十分に多くの試験例を与えた場合のEBLの有効性を計算することができた．

以下に今後の課題を示す．

- LE/ILEにによる，操作性規範の動的選択
  操作性規範をLE/ILEのパラメータと考えることにより，訓練例，領域理論の変化に対応して最適な操作性規範を選択し，更新することが可能ではないかと考えている．
- 大規模な実験結果との比較．
- 説明構造生成，一般化及び知識管理手続きの定式化による有限試験例におけるEBL有効性の計算．
- 関数LE/ILEの解析．
- 個々のEBL研究のSEPから見た位置づけ．
- マクロオペレータによる一般化向上 [山田88] の理論付け．

```
GC : a: - b.
     a: - b1, b2.
     a: - b01, b02.

TE1 : [ a , [z]]
TE2 : [ a , [e1,e2,e3,e4,e5,e6,e7,
            e8,e9,e10,e11,e12,
            e13,e14,e15,e16]]
TE3 : [ a , [e01,e02,e03,e04,e05,
            e06,e07,e08,e09,
            e010,e011,e012,e013,
            e014,e015,e016]].

DT: b: - c.   m: - n.   b1: - c1,c2.    b01: - c01,c02.
    c: - d.   n: - o.   b2: - c3,c4.    b02: - c03,c04.
    d: - e.   o: - p.   c1: - d1,d2.    c01: - d01,d02.
    e: - f.   p: - q.   c2: - d3,d4.    c02: - d03,d04.
    f: - g.   q: - r.   c3: - d5,d6.    c03: - d05,d06.
    g: - h.   r: - s.   c4: - d7,d8.    c04: - d07,d08.
    h: - i.   s: - t.   d1: - e1,e2.    d01: - e01,e02.
    i: - j.   t: - u.   d2: - e3,e4.    d02: - e03,e04.
    j: - k.   u: - v.   d3: - e5,e6.    d03: - e05,e06.
    k: - l.   v: - w.   d4: - e7,e8.    d04: - e07,e08.
    l: - m.   w: - x.   d5: - e9,e10.   d05: - e09,e010.
              x: - y.   d6: - e11,e12.  d06: - e011,e012.
                        d7: - e13,e14.  d07: - e013,e014.
                        d8: - e15,e16.  d08: - e015,e016.
```

Fig.3 実験2の入力

＜謝辞＞

本研究におけるPrologの計算コストの評価法について，JUNET上で大変有益な議論をしていただいた近山氏(ICOT)，山崎氏(NTT)，中島氏(ETL)らの諸氏に，そして的確なコメントをいただいた岡夏樹氏(ICOT)に感謝します．また，WOL'90，ICOT KSA-WGでも，ありがたい助言をいただきました．最後に，日頃議論していただいている本学産業科学研究所安部憲広助教授と辻研FAIグループの皆さんに感謝する次第です．

なお，本研究は，平成元年度倉田奨励金の援助を受けて行われた．あわせて，感謝いたします．

＜参考文献＞

[Aho76] Aho,A.V., Hopcroft,J.E. and Ullman,J.D. : アルゴリズムの設計と解析(野崎・野下共訳)，サイエンス社(1976)

[近山90]近山氏(ICOT),山崎氏(NTT)，中島氏(ETL)，平田氏(ICOT)とのJUNET fj.lang.prologでの議論，5/14-6/7 (1990)

[DeJong86] DeJong,G. and Mooney,R.: Explanation-Based Learning: An Alternative View, Machine Learning, 1(2):145-176 (1986)

[Greiner89] Greiner,R. and Likuski,J.: Incorporating Redundant Learned Rules: A Preliminary Formal Analysis of EBL, IJCAI-89, pp.744-749 (1989)

[Hirsh87] Hirsh, H. : Explanation-Based Generalization in a Logic-Programming Environment, IJCAI-87, pp. 221-227 (1987)

[Kedar-Cabelli87] Kedar-Cabelli,S.T. and McCarty,L.T.: Explanation-based Gneralization as resolution theorem proving, Fourth International Machine Learning Workshop, pp.383-389 (1987)

[Keller87] Keller,R.M.: Defining Operationality for Explanation-Based Learning, AAAI-87, pp.482-487 (1987)

[Mitchell86] Mitchell,T.M. Keller,R.M. and Kedar-Cabelli, R. T. : Explanation-Based Generalization: A Unifying View, Machine Learning, Vol.1 No.1, pp.47-80 (1986)

[Minton90] Minton,S.: Quantitative results concerning the utility of explanation-based learning, Artificial Intelligence 42, pp.363-391 (1988)

[Mooney86] Mooney, R.J. and Bennet, S.W. : A Domain Independent Explanation-Based Generalizer, AAAi-86, pp.551-555 (1986)

[Mooney89] Mooney, R. J. : The Effect of Rule Use on the Utility of Explanation-Based Learning, IJCAI-89, pp.725-730 (1989)

[Moto-oka81] Moto-oka, T. : Challenge for Knowledge Information Processing Systems, Fifth Generation Computer Systems, pp.3-92 (1981)

[野口86]野口，滝沢：知識工学基礎論，オーム社(1986)

[Prieditis87] Prieditis, A. E. and Mostow. J. : PROLEARN: Towards A Prolog Interpreter that Learns, AAAI-87, pp.494-498 (1987)

[Shavlik88] Shavlik, J. W. : Generalizing the Structure of Explanations in Explanation-Based Learning, Ph.D. Thesis, Department of Compùter Science, University of Illinois, Urbana, IL, January (1988)

[梅村90]梅村，山崎：記号処理におけるベンチマーク，情報処理学会誌，Vol.31 No.3, pp.321-327 (1990)

[山田88]山田，安部，辻：直接解決可能性に基づく一般化：DSBG -操作可能なSOLVABLE概念の定義づけ-，人工知能学会誌，Vol.3 No.6 (1988)

[Yamada89] Yamada,S.: and Tsuji,S. Selective Learning of Macro-operators with Perfect Causality, IJCAI-89, pp.603-608 (1989)

[山田90]山田，辻：論理プログラミング環境におけるEBLの有効性計算，第4回人工知能学会全国大会論文集，pp.103-106 (1990)