知識工学と人工知能 47－8
（1986. 7. 11）

最良優先探索PROLOG

赤間 清

（北海道大学 文学部）

個々の知識に，相対的な確実度を付与することは，よりよい説明や仮説などを発見するシステムのための知識の体系をうまく表現する上で基本的である．それを可能にする有望な枠組みの1つに，確実度付き論理プログラムがある．本論文では，確実度付き論理プログラムの望ましいインタープリタを作成するために，最良優先探索アルゴリズムを与える．それは，標準的 Prolog のインタープリタが持つ特徴：1つの「質問」に対して複数の解があり，要請に応じて次の解を出し，各時点では，要請に応ずるための最小限の計算しかしない，を保存し，さらに，解は評価の高い順に出す，という特徴を満たす．そのアルゴリズムを基礎として，特定の形の確実度関数を持つBFSという拡張 Prolog システムが実現された．BFSは，さらに，無限計算を回避し，標準的 Prolog と類似の否定を扱うことができる．

Best First Search Prolog

Kiyoshi AKAMA

(Faculty of Letters, Hokkaido University, Sappro-shi, 060, Japan)

It is very important for many rule-based systems to specify certainties of rules and to compute certainties of inferred conclusions. One of the promising logical frameworks for it is logic program with uncertainties (LPU). This paper proposes a best first search (BFS) algorithm for LPU. It allows us to make useful interpreter for LPU that can give, to a given query, the current best solution in the remaining solutions (or tell us the failure of finding it) each time we ask the interpreter for another solution. Important is that each solution is found with the minimal computation required at each time. Using BFS algorithm we implemented an extended Prolog which has the BFS interpreter avoiding the infinite computations and dealing with the negation as failure similar to standard Prolog.

1．　はじめに

個々の知識に，相対的な確実度や信頼度などを付与することは，人間の多様な知識の体系を扱う上で基本的なものである．それを可能にする有望な枠組みの１つは，確実度付き論理プログラムである．それは，各ホーン節に確実度関数がつけ加えられた論理プログラムである[1]．確実度関数ｆは，ホーン節 $A \leftarrow B_1, B_2, \cdots, B_n$ と代入θを用いて，「$B_1\theta, B_2\theta, \cdots, B_n\theta$ならば$A\theta$」という推論をするとき，$B_1\theta, B_2\theta, \cdots, B_n\theta$の確実度 $b_1, b_2, \cdots, b_n$ から$A\theta$の確実度 a を計算するために用いられる．それを繰り返すことによって，証明可能なすべての原子論理式に対する確実度が計算される．従って確実度付き論理プログラムは，任意の原子論理式が証明可能か否かを示すだけでなく，どのような確実度で証明可能かも指定する．これは，確実度の違ういろいろな情報からなる知識を適切に扱うための新たな枠組みを示唆する．

確実度付き論理プログラムを扱うためには，確実度を計算できるように Prolog のインタープリタを拡張する必要がある．E. Shapiro はそのような インタープリタの概略を与えている[1]．それは，確実度のしきい値による探索打ち切りを行う方法を含んでいるものの，基本的には，先ず証明を構成し次にその確実度を計算する方法である．そのような方法には問題点がある．それは，確実度の高い証明も低い証明も区別しないで，証明の探索が行なわれることである．従って，ある原子論理式が証明できる最高の確実度を求めるには，全部の証明の探索が必要である．

これに対して本論文では，より高い確実度を与える証明をより優先して得るようなインタープリタを作るための理論およびその実現について述べる．そのインタープリタは，先ず最も確実度の高い解を与え，その後は要求に応じて，次に良い解を計算する．それは証明の探索を確実度関数の情報で制御することで達成される．各時点において，インタープリタの計算する量は必要最小限に抑えられている．我々はその探索制御を最良優先探索と呼ぶ．最良優先探索のインタープリタがあれば，確実度によって宣言的に記述された情報を用いて探索の優先順位を制御することが可能になる．

本研究は，帰納的学習システムの最良応答探索アルゴリズムの研究[2]から始った．帰納的学習システムの持つ知識は，システムの作成する仮説的な知識であり，各知識の確実度はさまざまである．従ってシステムは，より確実な知識を用いてなるべく良い応答を最小の時間で探索するアルゴリズムを持たねばならない．本論文の理論は，そのアルゴリズムの働く舞台を，学習システムのための知識表現から確実度付き論理プログラムに置き換えて得られた．

論理プログラムによる知識処理に確実度などを組み込む研究には，Fuzzy Prologの実現[3]などの試みがある．しかしそれらの研究では，最良優先探索は実現されていない．Fuzzy Prologの研究や本研究を，Prolog と 不確実性（fuzzyness あるいは uncertainty）との「調和点」を見出す研究として把えるとき，Fuzzy Prologの研究の選択した「調和点」は本研究の選択した「調和点」よりも，不確実性に近いところにあると言える．その結果それらのシステムでは，最良優先探索のコストが高くなり，その実現が困難である．我々はしかし，最良優先探索あるいはそれに準ずるものを備えることが，大局的に見て非常に重要であると考えている．大規模な知識の探索を確実度で制御することは，学習システムの応答探索や仮説の発見などを含む非常に多くの知識情報処理において欠くべからざる機能だと考えられるからである．

2．　確実度付き論理プログラム

2．1　確実度付き論理プログラム

確実度付き論理プログラムなどの定義をあたえる．ここの定義は E. Shapiro の定義[1]を改訂したものである．ホーン節は，Ａを原子論理式，Ｂをｎ個（$n \geq 0$）の原子論理式の論理積とするとき，$A \leftarrow B$の形の節（clause）である．確実度関数ｆは，Ｉを区間 $\{x \mid 0 < x \leq 1\}$ とするとき，I^n からＩへの関数である．確実度付き論理プログラム（logic programs with uncertainties）Ｐは，２項組$<A \leftarrow B, f>$の有限集合である．ここで，$A \leftarrow B$はホーン節，ｆは確実度関数であり，Ｂに出現する論理式の数とｆの引数の数は等しいものとする．

確実度関数は次のように使われる．ホーン節 $A \leftarrow B_1, B_2, \cdots, B_n$ が代入θの適用を受けて，

「もし$B_1\theta, B_2\theta, \cdots, B_n\theta$ならば$A\theta$である」

という推論に使われる場合を考える．ホーン節 $A \leftarrow B_1, B_2, \cdots, B_n$ に付随する確実度関数fは，$B_1\theta, B_2\theta, \cdots, B_n\theta$ に対する確実度 $b_1, b_2, \cdots, b_n$ から$A\theta$の確実度aを

$$a = f(b_1, b_2, \cdots, b_n)$$

によって算出する．ここで，特にn＝0の場合について注意しておく．そのとき，$I^n = \{\phi\}$ となるので，fはただ1つの値 $f(\phi)$ で特徴づけられ，

$$a = f(\phi)$$

となる．

確実度関数fには次の2条件（置換不変性，単調性）が要請されている．

(1) $f(p(X)) = f(X)$ … pは成分の任意の置換

(2) $X \leqq Y \rightarrow f(X) \leqq f(Y)$

ここで，I^nでの半順序$\leqq$は，I上の全順序$\leqq$により，

$X \leqq Y \leftrightarrow$ 全てのi（$1 \leqq i \leqq n$）に対して，（Xのi成分）$\leqq$（Yのi成分）

と定義されている．

Pにおける原子論理式Aの1つの証明pは，Pの節の有限回の適用からなる．それにそって，各節に付随する確実度関数を順次適用すれば，Aの確実度が求まる．それを，Aのpによる確実度と呼び，$c(A, p)$で表わす．原子論理式Aが確実度Cで証明可能であるとは，$C \leqq c(A, p)$となるAの証明pが存在することである．

2．2　確実度付き論理プログラムのインタープリタ

プログラムPに対する原子論理式Aの証明pは複数個存在しうる．そしてそれらの証明によって計算されるAの確実度は異なり得る．従って例えば，探索によってAの1つの証明p_1を得て，Aが確実度$c(A, p_1)$で証明可能と推論したとせよ．それはAの確実度に対する十分な情報になるとは限らない．実はp_2という証明が存在して，

$$c(A, p_1) < c(A, p_2)$$

を満足しているかも知れないからである．Aの証明p_2は，Aが確実度 $c(A, p_2)$ で証明可能であるという，より強い結果を与える．これは確実度が（0か）1しかなく，証明を得ることと最高の確実度を得ることが等価である標準的な論理プログラムの場合には顕在化しなかった特徴である．

E. Shapiroは原子論理式の確実度を求めるインタープリタについて述べている[1]．彼の方法は，基本的には，まず証明pを求めて，それに対して$c(A, p)$を計算する形式のものである．従って，相対的に重要性の低い証明の方が先に次々に求められてしまう場合があるという不都合を回避できない．

3．　インタープリタに要請される条件

Prologのインタープリタは次の特徴を持つ．

(1) 1つの「質問」に対して，複数の解を持ち得る．

(2) 要請があるときだけ次の解を出す．

(3) 次の解を要請された時点では，その要請に応ずるための最小限の計算しかしない．

このうち(3)は，探索の効率化にとって重要である．(3)は，あらかじめすべての解を求めてから，要請に応じて解を出力するのではないことを示す．もしそうであれば，要請される解の数に係わらずに全ての計算をすることになり，要求が途中で打ち切られる場合に無駄が生じる．

確実度付き論理プログラムのインタープリタは，標準的なPrologのインタープリタのこのような特徴をなるべく保存しながら自然に拡張して得られることが望ましい．確実度付き論理プログラムのインタープリタに是非追加すべき特徴を挙げる．

(4) 解は，評価の高い順に出す．

ここで解の評価とは，その解を産み出す証明によって計算される解の確実度を意味する．評価の高い解は，より重要な解である．また，より重要な解ほど要求される頻度が高い．(4)の条件を満たさない場合，最良の解だけを得るためにも全ての解を求める必要がある．(1)から(4)までを満たせば，最良の解から順次参照でき，既に参照した結果をもとにしてさらに次善の解を参照するか否か決定できる．また，確実度付き論理プログラムの中に確実度の極めて低い情報が大量に含まれていても，それらを

どの程度探索するかをうまく制御できる．
(5) 無限ループ，無限計算を回避する．
(6) 否定をうまく扱う．
これらは，知識処理のシステムとして満たすことが望ましい条件である．しかし，通常の Prolog のインタープリタの場合でも，(5)を扱っていないし，(6)は完全ではない．従って，これらをどの程度満たすべきかは計算コストとのかねあいで決める必要がある．

4． 証明木と部分証明木

確実度付き論理プログラムのインタープリタを明確に述べるために証明木と部分証明木を定義する．証明木は，証明の構造と確実度の計算を記述する．部分証明木は，直観的には，証明木のいくつかの頂点に対して，その頂点から下の部分木と，それに付随する情報を取り除いたものである．

4．1 確実度付き論理プログラムの証明木

確実度付き論理プログラムＰの証明木(proof tree)ＰＴとは，次の条件を満たす根付き木Ｔと写像ｇ，ｈ，ｖの４項組（Ｔ，ｇ，ｈ，ｖ）である．
(1) Ｔの頂点集合は，PAST，DONEの２つの集合に直和分割される．PASTは葉でない頂点の集合，DONEは葉の集合である．DONEはＴの根を含まない．
(2) ｇはＴの頂点に原子論理式あるいはTRUEを対応させる．特に，
ｇ（ｍ）＝TRUE ←→ ｍはDONEに属する
が成り立つ．
(3) ｈはPASTの頂点に，Ｐの元＜Ａ←Ｂ，ｆ＞と代入θのペアを対応させる．
(4) ｖはPASTから区間（０，１］への関数である．
(5) ｍをPASTの頂点とし，
ｈ（ｍ）＝（＜Ａ←Ｂ，ｆ＞，θ），
$B = B_1, B_2, \cdots, B_n$
とする．そのとき，
ｍの子はｎ個存在する，（それらを$m_1, m_2, \cdots, m_n$と書く）
$g(m) = A\theta$,
$g(m_i) = B_i\theta$,
$v(m) = f(v(m_1), v(m_2), \cdots, v(m_n))$
が成り立つ．
原子論理式Ａと証明木ＰＴ＝（Ｔ，ｇ，ｈ，ｖ）において，PASTに属するＴの根ｒに対して，
ｇ（ｒ）＝Ａ
が成り立つとき，ＰＴはＡの証明木であるという．そのとき（証明木ＰＴと，ＰＴの表わす証明ｐを同一視して）
ｃ（Ａ，ＰＴ）＝ｖ（ｒ）
が成り立つのは明らかである．
証明木をＰＴ＝（Ｔ，ｇ，ｈ，ｖ）とし，Ｔの葉でない頂点をｍとする．ｍを根とするＴの部分木をT_mとし，ｇ，ｈ，ｖのT_mへの制限をg_m，h_m，v_mとすれば，（T_m，g_m，h_m，v_m）はやはり証明木になる．

4．2 確実度付き論理プログラムの部分証明木

証明木を拡張して，部分証明木を定義する．確実度付き論理プログラムＰの部分証明木ＰＴは，次の条件を満たす根付き木Ｔと写像ｇ，ｈ，ｖの４項組（Ｔ，ｇ，ｈ，ｖ）である．
(1) Ｔの頂点集合は，PAST，FRONT，DONEの３つの集合に直和分割される．PASTは葉でない頂点の集合，FRONTとDONEは葉の集合であり，DONEはＴの根を含まない．
(2) ｇはＴの頂点に原子論理式あるいはTRUEを対応させる．特に，

$g(m) = \mathrm{TRUE} \leftrightarrow m$はDONEに属する

が成り立つ．

(3) hはPASTの頂点に，Pの元$<A \leftarrow B, f>$と代入θのペアを対応させる．

(4) vはPAST＋FRONTから区間（0，1］への関数であり，

$v(m) = 1$ … mはFRONTに属する頂点

が成り立つ．

(5) mをPASTの頂点とし，

$h(m) = (<A \leftarrow B, f>, \theta)$，

$B = B_1, B_2, \cdots, B_n$

とする．そのとき，

mの子はn個存在する，（それらを$m_1, m_2, \cdots, m_n$と書く）

$g(m) = A\theta$，

$g(m_i) = B_i\theta$，

$v(m) = f(v(m_1), v(m_2), \cdots, v(m_n))$

が成り立つ．

部分証明木の定義が証明木の定義と異なるのは，(1)と(4)である．証明木はFRONT＝φという条件を満たす部分証明木である．

部分証明木$PT = (T, g, h, v)$とそのPASTに属する頂点の1つmが与えられたとき，次のようにして新たな部分証明木$PT_m = (T_m, g_m, h_m, v_m)$を作ることができる．

(1) m以外のmの子孫の集合をEXCLUDEとする．

(2) PT_mのPASTを PAST − EXCLUDE とする．

(3) PT_mのFRONTを FRONT − EXCLUDE ＋ {m} とする．

(4) PT_mのDONEを DONE − EXCLUDE とする．

(5) TからEXCLUDEに属する頂点とそれに隣接する辺を取り除いて新しい根付き木T_mをつくる．

(6) 部分証明木の定義にあわせてg，hの定義域を制限しg_m，h_mを作る．

(7) 部分証明木の定義にあわせて，vの定義域や値を変更しv_mを作る．

この変更により新たな部分証明木は一意に定まる．この変形の直観的意味は，PTから，頂点mを根とする部分木とそれに付随する情報を取り除いてPT_mをつくることである．mの確実度の情報は失われるので，大きめに1と推定される．それに伴って，部分証明木の全ての頂点に対応する論理式の確実度も大きめの推定値に変る．このような変形を有限回（0を含む）施してPT_1からPT_2が得られるとき，PT_2はPT_1の部分証明木であるという．

PT_2がPT_1の部分証明木であり，v_1，v_2がPT_1，PT_2の各頂点の評価値を与える関数とするとき，確実度関数の単調性により，PT_2のPAST＋FRONTに属する任意のmに対して

$0 < v_1(m) \leqq v_2(m) \leqq 1$

が成り立つ．

5． 最良優先探索アルゴリズム

5．1 部分証明木の逐次的更新による証明発見

Prolog を含む多くの定理証明システムは，原理的には，部分証明木の逐次的更新によって完全な証明に至る方法を用いている．それは，部分証明木を1つの状態として，逐次的に状態遷移する過程とみなすことができる．注意すべきことは，ある原子論理式Aが与えられた場合，各部分証明木の根rに対応する原子論理式g（r）が終始Aになる訳ではないことである．g（r）の初期値はAであるが，部分証明木の逐次的更新の過程で，代入による限定をうけて次第に変化する．Aという「質問」から出発して最終的に到達される状態は，$\exists\theta$： $A' = A\theta$ を満たすある原子論理式 A' である．

質問Aに対する初期状態は，

$PT_0 = (T, g, h, v)$，

PAST＝φ，FRONT＝{r}，DONE＝φ，

$g(r) = A$，$v(r) = 1$

である．状態遷移は，現在の状態ＳのFRONTが空でないときだけ可能である．最初に，FRONTからあらかじめ定めたある計算規則（computation rule）ｃｒによって，１つの頂点ｍが選ばれる．次にプログラムＰから原子論理式ｇ（ｍ）とユニファイ可能な頭部を持つホーン節の１つＡ←Ｂが選ばれ，mgu（most general unifier）θが求められ，ｍがPASTに入った新しい部分証明木が次の状態となる．新しい証明木の写像ｇ，ｈは，ｍから下の部分だけでなく，木全体で変化する可能性がある．それはθによって各変数への束縛が追加されるからである．また，写像ｖは大部分の頂点において減少する．これは，ｖ（ｍ）が，１からｆ（１，１，…，１）に変化し，それが木全体に波及するためである．

このような状態遷移の連鎖は，各状態遷移においてどのホーン節を選ぶかによっていろいろなものがありうる．それは次の３つの場合に分けられる．(1) FRONT＝ϕ となって連鎖が終る．成功である．最終状態から証明木が得られる．(2) FRONT≒ϕ だが，計算規則ｃｒによって選ばれたｍに対応するｇ（ｍ）にユニファイするホーン節がＰに存在しないので連鎖が終る．失敗である．(3) 上のどちらにもならずに，連鎖が無限に続く．

プログラムＰと計算規則ｃｒを与えたとき，ある原子論理式Ａの作る初期状態PT_0から出発する全ての連鎖の集合は，PT_0 を根とする木Ｒを作る．ただしそれが上記の(3)の連鎖を持てば，無限木となる．木Ｒは，最良優先探索の基本となる探索空間である．

各状態ＰＴの根付き木Ｔの根ｒは常にPASTかFRONTの元なので，ｖ（ｒ）が存在する．ＰＴによって一意に定まるこのｖ（ｒ）をＰＴの評価値と呼び，ｖａｌ（ＰＴ）で表わす．ＰＴが証明木のとき，

ｖａｌ（ＰＴ）＝ｃ（ｇ（ｒ），ＰＴ）

である．

１つの状態遷移で，状態がＰＴ１からＰＴ２に遷移するとき，ＰＴ１はＰＴ２の部分証明木になっている．従って，連鎖に出現する任意の証明木に対する評価値は全ての遷移につれて単調に減少する．連鎖が証明木ＰＴで終結するとき，連鎖の途中に出現する部分証明木ＰＴ’の評価値ｖａｌ（ＰＴ’）は，（大きめの）近似値を与える．

５．２　最良優先探索アルゴリズム

木Ｒは，有限または無限の根付き木である．Ｒの各頂点ｍに対してｍの評価値が関数ｖａｌで与えられている．評価値はＲの根から葉の方向に行くにつれて単調に減少する．求めるべき解は，Ｒの中の証明木であり，定義によりそれらは木Ｒの葉全体の集合の部分集合をなしている．我々の問題は，Ｒの根から葉の方向にうまく効率的に探索して，解を評価の高い順にすべて求めることである．

木Ｒについて，次のような手続きを考える．

１：FRONT ＝｛Ｒの根｝，PAST ＝｛｝
２：FRONT ＝｛｝ならば，この手続きから出る．
３：FRONT からその元Ｓを１つ選ぶ．FRONT からＳを取り除き，かわりにＳの子をすべていれる．またＳを PAST に入れる．すなわち
　FRONT ＝FRONT －｛Ｓ｝＋［Ｓの子全部の集合］
　PAST ＝PAST ＋｛Ｓ｝
４：２へ行く．

この手続きの実行中のすべての時点で，次の性質が成り立つ．

「FRONT に含まれる頂点の評価の最大値をＭとする．木ＲのなかでＭより大きい評価値を持つ頂点は PAST にのみ存在しうる．」

従って，PASTに解がなく，FRONTの最大値を与えるものが解ならば，それは最大の評価値を持つ解である．これより，Ｍを確実に減少させて行く次のアルゴリズムが導かれる．

１：FRONT を初期状態S_0だけを含むリストとする．
２：FRONT が空であれば，結果は FAIL であり，この手続きから出る．
３：FRONT から評価が最大の要素Ｓを除く．
４：Ｓが解ならば，結果は EXIT である．情報Ｓを得てこの手続きから出る．
５：頂点Ｓの子頂点をすべてFRONTに追加する．
６：２へ行く．

この手続きは FAIL という結果がでるまで繰り返して呼び出す事ができる．ただし，２回目以降はステップ２から入る．m回目の呼び出しにおいて，その結果が EXIT のときは，ｖａｌの評価に関してm番目に良い解を与えられる．また，結果が FAIL のときは，求める解はもう存在しない．この探索アルゴリズムを最良優先探索アルゴリズムと呼ぶことにする．

６．　最良優先探索の実現

６．１　最良優先探索の実現

我々は，次の制限をおいて最良優先探索を行なう拡張PrologシステムＢＦＳを実現した．

(1)　確実度関数は，ｆ（b_1，b_2，…，b_n）＝ｋ×b_1×b_2×…×b_n の形である．

(2)　計算規則ｃｒは，標準Prologと同じく，左から右への深さ優先探索を与える計算規則である．

このとき，確実度関数ｆは定数ｋで特徴づけられる．また，部分証明木を定義そのままの形で状態として保持する必要はない．それにより，最良優先探索Ｐｒｏｌｏｇは大幅に簡単化される．例えば，

（原子論理式のAND結合，代入，評価値）

の形の３項組を状態として部分証明木のかわりに利用すると仮定する．そのとき初期状態は，「質問」を原子論理式Ｑとすれば，

（（Ｑ），空代入，１）

となる．また，現在の状態が，

（（Ｐ　Ｑ　Ｒ　…），Θ，Ｖ）

であるとき，Ｐ，Ｑ，Ｒ，…のなかから計算規則ｃｒに従ってＰを取り出し，Ｐの中からホーン節と確実度関数のペア＜Ａ←Ｂ，ｋ＞が選ばれ（Ｂ＝B_1，B_2，…，B_n），ＰとＡとのマッチングにより，mgu θ が得られたとする．そのとき，次の状態は，

（（$B_1\theta$　$B_2\theta$　…　$B_n\theta$　Ｑθ　Ｒθ　…），Θθ，ｋＶ）

となる．最終状態は，原子論理式のAND結合が（）となる状態で，

（（），Θ，Ｖ）

の形をしている．その最終状態に到達したとき，原子論理式 ＱΘ が確実度Ｖで証明されたことになる．以上の記述は説明のために簡単化したものであり，ＢＦＳの実際のシステムはもちろん，構造共有（structure sharing）などの技法を用いてメモリー並びに計算時間の節約がなされている．

ＢＦＳに組み込まれたもう１つの重要な機構は，しきい値（threshold）による枝刈りである．これによって，確実度があらかじめ設定されたしきい値ｔ以下になるような論理式の証明に至るような探索は回避される．これは，新しい状態を生成するとき，その評価値とｔを比較し，ｔより小さい状態の生成を中止することによって簡単に実現できる．

６．２　無限ループと無限計算

ＢＦＳでは無限ループの回避は，同じ部分証明木の中に祖先と同一の問題が出現した場合，その問題を無条件に失敗させることによって行なう．ただし同一性は，問題の等価性だけを満たすだけの緩いものではなく，変数を含めた完全一致という厳しい条件を用いて判定している．その理由を見るために，次の標準Prologによる述語定義を考える．

```
ex( [a] ).
ex( [a | X] )  :-  ex(X).
```

問題 ex(X)の第２の解を取りだそうとするとき，親問題 ex(X)から，Ｘ＝［ａ｜Ｙ］として，子問題 ex(Y)が作られる．ex(X)と ex(Y)は等価な問題であるが，これを同一として失敗させれば，第２の解は得られない．述語 ex は無限個の解

［a］，［a　a］，［a　a　a］，…

を持つ．この解をすべて取り出すには，この子問題を成功裏に解く必要がある．この場合，ex(X)とex(Y)は別の問題と判定することが必要である．

確実度の存在が無限計算回避の助けとなる場合がある．例えば，確実度付き論理プログラムＰに属するすべてのホーン節の確実度関数ｆに対して，

$$f(b_1, b_2, \cdots, b_n) \leqq d \times min(b_1, b_2, \cdots, b_n)$$

を満たすｄ（０＜ｄ＜１）が存在すると仮定する．そのとき，深さｉの証明木ＰＴに対して，

$$val(PT) \leqq d^i$$

が成り立つ．従って，ある程度大きい木はしきい値ｔ以下になり存在しないものとみなすことができる．部分証明木のなす木Ｒは，そのとき，事実上有限となる．ＢＦＳが採用する確実度関数は上の不等式を満足する．他に上の式を満たす確実度関数には，

$$f(b_1, b_2, \cdots, b_n) = k * min(b_1, b_2, \cdots, b_n)$$

の形を持つものなどがある．

６．３　否定の導入

否定（negation as failure）の組み込み述語は，最良優先探索に困難をもたらす．Ａの確実度が高いほどｎｏｔ（Ａ）の確実度は低くしなければならない．この非単調性が困難の原因である．最良優先探索は，原子論理式Ａに関して「確実度α以上では証明できない」という形の情報を扱うことが基本になっており，そのαの値を少しずつ下げていくプロセスにより得られた解が最良の解であることを保証する．しかしｎｏｔ（Ａ）に関して「確実度α以上では証明できない」という形の情報を得るには，Ａに対して「確実度α以下では証明できない」という形の情報を要求しなければならない．

困難の最も簡単な（安易な）解決は，ｎｏｔ（Ａ）が引き起こす探索の場合には，インタープリタへの要請(3)を満たすための逐次的な探索をやめ，その部分の探索を一括して行なうことである．例えばｎｏｔ（Ａ）が出現した場合，Ａに関する最良優先探索を（確実度にかかわらず）他に優先して行ない，それで得た確実度をｎｏｔ（Ａ）の確実度に（あらかじめ定めた方法で）変換する．このＡに関する最良優先探索の計算量を抑えるためには，適当なしきい値以上の部分だけを計算すればよい．ＢＦＳはこの方法を採用している．

７．　むすび

確実度付き論理プログラムのインタープリタを作成するために，最良優先探索のアルゴリズムが与えられた．それは，標準的 Prolog のインタープリタが持つ３つの特徴，(1)　１つの「質問」に対して複数の解を出し得る，(2)　要請に応じて次の解を出す，(3)　各時点では，要請に応ずるための最小限の計算しかしない，を保存し，さらに，(4)　解は，評価の高い順に出す，という特徴を持つ．そのアルゴリズムは，特定の形の確実度関数を仮定してＢＦＳという拡張Prologシステムとして実現された．ＢＦＳは，さらに，(5)　無限計算を回避し，(6)　標準的Prologと類似の否定を扱う．

最良優先探索アルゴリズムを用いれば，最良の解から順次参照し，参照結果をもとにしてさらに次善の解を参照するか否か決定できる．従って，よりよい説明や仮説の発見などに効果的に用いることができる．また，確実度付き論理プログラムの中に確実度の低い情報が大量に含まれていても，それらをどの程度探索するかをうまく制御できる．

今後の課題としては，効率的な最良優先探索システムの設計技法の発見，最良優先探索を知識表現や仮説生成などに応用する技術の開発などがある．

文献

[1] Shapiro,E:　"Logic Program with Uncertainties: A tool for Implementing Rule-Based Systems",　Proc. 8th IJCAI, p529-532 (1983)

[2] 赤間　清：　帰納的学習システムの最良応答探索，情報処理学会知識工学と人工知能研究会，41-12, p89-96 (1985)

[3] Sakakibara,Y:"On Semantics of Logic Programs with Uncertainties", 情報処理学会第３２回全国大会講演論文集，p1245-1246 (1986)

[4] Ishizuka,M and Kanai,N:"Prolog-ELF Incorporating Fuzzy Logic", New generation computing, 3 ,p479-486 (1985)