AND木のBreadth-first Search Strategyを基礎とする論理型言語の並列分散処理系

吉田　幹、　山崎　進、　堂下　修司
（京都大学・工学部）

1. はじめに

論理型言語に関する研究は、第一階述語論理の式の部分クラスであるHorn節集合を計算の対象とする言語（Prologを代表とする）を中心に盛んに進められている。論理型言語の実行は与えられたプログラム（Horn節集合）に対する入力演繹で定められることが多く、また、その戦略も、探索空間であるAND/OR木をDepth-firstに調べる手法が大半である。このような言語の処理系の中核をなす演繹機構に焦点を置く研究は、現在まで各所で行われてきた。

本稿では、入力演繹を基本に、AND/OR木の探索戦略をBreadth-firstに行うことを前提に、そこから構成される演繹機構について述べる。論理型言語において、その処理が並列分散化されることは必然である。AND/OR木のAND木において、2つの基本となる探索戦略（Depth-first、Breadth-first Search Strategy）を比較すると、Horn節集合の演繹をAND木が均等に展開されて進行するものと解釈する上で、後者は自然であり、並列分散化にも向いていると考える。ここで構成、提示する演繹機構は、2つの特徴を持つ。第1には、Horn節に対応する複数のプロセスから構成されること、第2には、全体の演繹処理が、各プロセスのunifierの操作（受理、combination演算、送出）だけで進行することである。このようにHorn節集合という入力プログラムのSyntax上の特性（節、リテラルの配置の非順次性）が、Horn節対応の分散プロセス化で演繹に反映される点、及び、演繹における操作対象と操作が、unifierとそれに対するcombination演算という、それぞれ単一の概念であるという点で、この演繹機構は論理型言語の1つの処理系として興味深いと思われる。

2. Horn節集合の演繹機構
—AND木のBreadth-first Search Strategy

ここでは、AND木のBreadth-first Search Strategyに基づき、Horn節集合の演繹機構を並列分散処理するための準備を行う。

Horn節集合をS、ゴール節を$G=[-L_1^0, \cdots, -L_{n_0}^0]$とする。ここで、節は、リテラルのリストとして扱う。[,]の記法は構成要素のリストを示す。（尚、Horn節集合、及び、入力導出、入力演繹に関する諸定義は通常のものとほとんど同じになるので省略する。）

Horn節集合に対する演繹として、AND/OR木をDepth-firstに探索する入力演繹（IDdと名付ける）は、以下の定義にまとめることができる。

［定義1］--- IDdに関して

1° $C_0 = G$.

2° $C_k = [-L_1^k, \cdots, -L_{n_k}^k]$ とする。

$\exists B_{k+1} = [P_{k+1}, -Q_1^{k+1}, \cdots, -Q_{m_{k+1}}^{k+1}] \in S$,

$\exists$ mgu σ_{k+1}, $L_1^{k+1}\sigma_{k+1} = P_{k+1}\sigma_{k+1}$

のとき、

C_{k+1}は次のように与えられる。

a) $1 \leq i \leq n_k - 1$ について、

$$-L_{i+m_{k+1}}^{k+1} = -L_{i+1}^{k}\sigma_{k+1}$$

b) $1 \leq i \leq m_{k+1}$ について、

$$-L_i^{k+1} = -Q_i^{k+1}\sigma_{k+1}$$

c) $n_{k+1} = n_k + m_{k+1} - 1$

IDd は、帰納的に 1°, 2° から得られる 1つの演繹 $C_0, B_1, C_1, \cdots, B_n, C_n$ である。 ■

定義1 に対し、AND木を Breadth-First に探索する入力演繹（IDbと名付ける）は、次の定義で与えられる。

［定義2］ --- IDb に関して

1° $C_0 = G$.

2° $C_k = [-L_1^k, \cdots, -L_{n_k}^k]$ とする。

$^\exists B_{k+1} = [P_{k+1}, -Q_1^{k+1}, \cdots, -Q_{m_{k+1}}^{k+1}] \in S$,

$^\exists \mathrm{mgu}\ \sigma_{k+1}$, $L_1^{k+1}\sigma_{k+1} = P_{k+1}\sigma_{k+1}$

のとき。

C_{k+1} は次のように与えられる。

a) $1 \leq i \leq n_k - 1$ について、

$$-L_i^{k+1} = -L_{i+1}^k \sigma_{k+1}$$

b) $1 \leq i \leq m_{k+1}$ について、

$$-L_{i+n_k-1}^{k+1} = -Q_i^{k+1}\sigma_{k+1}$$

c) $n_{k+1} = n_k + m_{k+1} - 1$.

IDb は、帰納的に 1°, 2° から得られる 1つの演繹 $C_0, B_1, C_1, \cdots, B_n, C_n$ である。 ■

実際の多くの Prolog システムは、定義1 に見る演繹を用いている。形式上、2つの定義は項目 a), b) を除いて同一であり、両者とも入力演繹の定義の枠を越えない。定義1, 2 において探索順序が異なってくるのは、導出の結果、生ずる負リテラルのリストを中心節（C_k）のリストの先頭部か後尾部にもってくることに起因する。

AND木はターミナルノードの状態により次の4つの場合に分けることができる。

	ノード数	ターミナルノード	解
case1.	有限	すべてが □	有
case2.	〃	少くとも1つ □ でない	無.
case3.	無限	〃	〃
case4.	〃	すべてが □	〃

停止性について考察すると、case1, 2 については IDd, IDb の両者ともに保証され、case4 については、両者ともに停止性は保証されない。ところが、case3 については、IDb で停止性が保証されるのに対し、IDd では停止性は必ずしも保証されないことがわかる。

Horn節集合の演繹機構を並列分散化するため、次の定式化を行う。

［定義3］ --- IDD に関して

1° $D_0 = G\theta_0$, $\theta_0 = \epsilon$.

2° $D_k = [-L_1^k, \cdots, -L_{n_k}^k]\theta_k$ とする。

$1 \leq i \leq n_k$ について、

$^\exists B_i = ap(P_i, N_i) \in S$,

$^\exists \mathrm{mgu}\ \sigma_i$, $[L_i^k]\sigma_i = P_i\sigma_i$ のとき。

D_{k+1} は、次のように与えられる。

$$D_{k+1} = [-L_1^{k+1}, \cdots, -L_{n_{k+1}}^{k+1}]\theta_{k+1}$$

a) $[-L_1^{k+1}, \cdots, -L_{n_{k+1}}^{k+1}] = ap(N_1, \cdots, N_{n_k})$

b) $\theta_{k+1} = \theta_k + \sigma_1 + \cdots + \sigma_{n_k}$

帰納的に 1°, 2° から得られる 1つの演繹 $D_0, D_1, \cdots, D_n$ を IDD という。

ここで、P_i, N_i（$1 \leq i \leq n_k$）は、それぞれ正リテラル、負リテラルのリストである。 ■

S, G については、先の定義1, 2 と同様の定義を与える。表現 $ap(X_1, \cdots, X_m)$ は、X_i（$1 \leq i \leq m$）をリストとして、X_1 から X_m までリストを連結してできるリストを示すものとする。又、表現 $\varphi_1 + \cdots + \varphi_r$（$\varphi_i$ は unifier）は各 φ_i の combination(*) を示す。

(*) 定義[1] を、以下に示す。

$\varphi_1 = \{t_{11}/v_{11}, \cdots, t_{1n_1}/v_{1n_1}\}$, $\cdots$, $\varphi_r = \{t_{r1}/v_{r1}, \cdots, t_{rn_r}/v_{rn_r}\}$ とする（$r \geq 2$）。表現 $E_1 = (v_{11}, \cdots, v_{1n_1}, \cdots, v_{r1}, \cdots, v_{rn_r})$, $E_2 = (t_{11}, \cdots, t_{1n_1}, \cdots, t_{r1}, \cdots, t_{rn_r})$ について、$\mathrm{mgu}(E_1, E_2)$ が $\varphi_1, \cdots, \varphi_r$ の combination ■

ここで定義した演繹IDDは、AND木をBreadth-firstに探索する戦略で、木の深さ毎にその過程が進行するような演繹である。 D_nは、AND木において、深さnの全ノードのリストと対応する。 D_nの持つ情報は、入力演繹における中心節の持つ情報と同種で、この定義において、初めにプログラムとして与えられるHorn節集合内の負リテラル（置換はかかっていない）のリストと、それ全体にかかる置換から構成されるところに特徴がある。

[定義4] --- IDDcに関して

Horn節集合をS，ゴール節を $G = N_1^0$ とする。

1° $D_0 = G\theta_0$, $\theta_0 = \epsilon$,

2° $D_k = ap(N_1^k, \cdots, N_{n_k}^k)\,\theta_k$ とする。

$N_i^k = [-L_{i1}, \cdots, -L_{im_i}]$ に関し、

$1 \le j \le m_i$ について、

$\exists B_{ij} = ap(P_{ij}, N_{ij}) \in S$,

$\exists\, mgu\ \sigma_{ij}$,　$[L_{ij}]\sigma_{ij} = P_{ij}\sigma_{ij}$

のとき、$\rho_i = \sigma_{i1} + \cdots + \sigma_{im_i}$ とすると、

D_{k+1}は、次のように与えられる。

$D_{k+1} = ap(N_1^{k+1}, \cdots, N_{n_{k+1}}^{k+1})\,\theta_{k+1}$

a) $ap(N_1^{k+1}, \cdots, N_{n_{k+1}}^{k+1}) = ap(N_{11}, \cdots, N_{1m_1}, \cdots, N_{n_k1}, \cdots, N_{n_k m_{n_k}})$

b) $\theta_{k+1} = \theta_k + \rho_1 + \cdots + \rho_{n_k}$

帰納的に1°,2°から得られる1つの演繹 $D_0, D_1, \cdots, D_n$ をIDDcという。

ここで、各P_{ij}は、正リテラル1つからなるリスト、各N_{ij}, N_i^kは負リテラルのリストである。 ■

ここで定義した演繹IDDcは、先のIDDのD_nの持つ負リテラルのリスト情報の箇所を、正リテラルを除いた入力Horn節を連結した表現にまとめあげている所に特徴を有する。 この性質は、IDDで関連のなかった負リテラルを元来、それを含んでいた節でまとめて、節単位の一括処理を可能にするものである。

次に、下の例について、IDDcの演繹過程を示す。

[Horn節集合（プログラム），ゴール節(g)]

(a1)　[Add(0,v,v)]

(a2)　[Mult(0,u,0)]

(p1)　[Add(s(p),q,s(r)),
　　　-Add(p,q,r)]

(p2)　[Mult(s(x),y,w),
　　　-Mult(x,y,z), -Add(y,z,w)]

(g)　[-Mult(s(s(0)),s(s(0)),ans)]

（0は定数、sは関数記号、他の項はすべて変数である。以下Add, Multは略記）

[前処理]

• (g)について

-M(s(s(0)),s(s(0)),ans) と unification 可能節	unifier
(p2)	$\alpha = \{s(0)/x, s(s(0))/y, ans/w\}$

• (p1)について

-A(p,q,r)	unifier
(a1)	$\beta_1 = \{0/p, v/q, v/r\}$
(p1)	$\beta_2 = \{s(p)/p', q/q', s(r)/r'\}$

• (p2)について

-M(x,y,z), -A(y,z,w)	unifier*
(a2) , (a1)	γ_1
(a2) , (p1)	γ_2
(p2) , (a1)	γ_3
(p2) , (p1)	γ_4

* 得られる各unifierのcombinationがとられている。

$\gamma_1 = \{0/x, 0/y, 0/z, 0/w, 0/u, 0/v\}$

$\gamma_2 = \{0/x, s(p)/y, 0/z, s(r)/w, 0/q, y/u\}$

$\gamma_3 = \{s(x)/x', 0/y', w/z', w/w', w/v\}$

$\gamma_4 = \{s(x)/x', s(p)/y', q/z', s(r)/w',$
　　$s(p)/y, q/w\}$

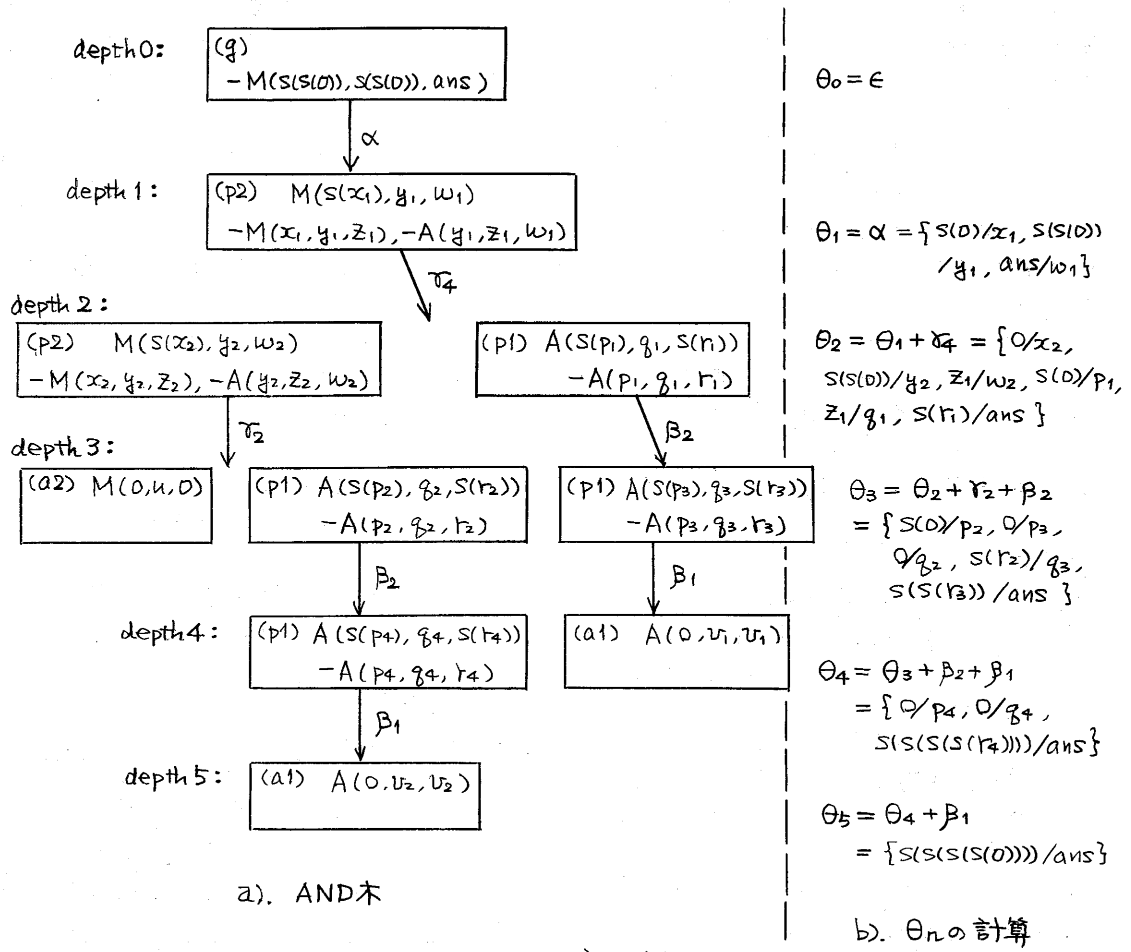


図1. プログラム例における IDDc の 演繹過程

IDDcの演繹は、前処理として、あらかじめ節の間でunificationを行っておき、その結果を本処理で用いることが可能となっている。定義4では、N_i^k を負リテラルのリストとして含む節に対する p_i が、その前処理で算出できる情報に当たる。前ページに、(g), (p1), (p2) の負リテラルを含む節について、その算出結果を示した。同じ節のunificationに際しては、注目している節の変数をrenamingした。図1は、IDDcの演繹過程を図示したものである。左側は、AND木に対応させた表現で、右側は、θ_nの計算の過程を左図のAND木のdepthと対応付けて示したものである。θ_5が、この計算の答である。

3. Unifierを媒介とする並列分散演繹機構

本章では、先に定義を与えた演繹に対し、さらに考察を進めていく。はじめに、演繹機構の定式化の上で基礎となるunificationに関する性質を与え、定義3, 4の演繹の正当性を立証する。次に、定義4の演繹IDDcに関

し、演繹機構としての特徴点を列記する。 最後に、並列分散演繹の記述として、テンプレートを用いた表現が、可能であることを例示する。

3.1 演繹機構の正当性

[命題1]

$mgu(A, B) = \theta$ のとき、

1. $mgu(A\rho, B) = (\theta+\rho)_{var(A\rho)\cup var(B)}$
   ここで、ρ：置換、
   $(\theta+\rho)_{var(A\rho)\cup var(B)}$ は $\theta+\rho$ で、$A\rho$ と B に含まれる変数に関する置換だけからなる

2. $A\rho(\theta+\rho)_{var(A\rho)\cup var(B)} = A\rho(\theta+\rho) = A(\theta+\rho)$
   $B(\theta+\rho)_{var(A\rho)\cup var(B)} = B(\theta+\rho)$

[証明]

2は定義から明らか。

1に関して、

A'：Aの変数のrenamingによって得られる表現。

ここで、A'の変数と$A\rho$、及びBの変数に共通部分はないものとする。

$mgu(A', B) = \theta'$ とする。

ρ'をρから $A'\rho' = A\rho$ となるよう構成する。 ρ'には$A\rho$中の変数に関する置換はないため、ρ'はA'と$A\rho$のmguとなる。 すなわち

$mgu(A', A\rho) = \rho'$

$(A', A\rho), (A', B)$ の mgsu[2] ωを考える。 $\omega = \rho'+\theta'$ である。

$$\omega = mgu(A', A\rho)\circ mgu(A'\rho', B\rho')$$
$$= \rho'\circ mgu(A\rho, B) \quad (\because B\rho' = B)$$

一方、

$A'\omega = A\rho\omega$ かつ $A'\omega = B\omega$ より

$A\rho\omega = B\omega$ すなわち ω は$(A\rho, B)$のunifierであることが言える。

従って、 $\omega = mgu(A\rho, B)\circ\delta, \exists\delta$

$\omega = \rho'\circ mgu(A\rho, B)$ において、ρ'中の置換の変数は、A'中の変数であり、$A\rho$、Bの変数と共通なものはないので、

$\omega_{var(A\rho)\cup var(B)} = mgu(A\rho, B)$ が言える。 ここで、明らかに、

$$(\theta'+\rho')_{var(A\rho)\cup var(B)} = (\theta+\rho)_{var(A\rho)\cup var(B)}$$

なので、1は成立する。 ■

[命題2]

$C_0 = [-L_1, \cdots, -L_m]\,\theta_0$ とする。

ここで、$-L_i$ $(1\le i\le n)$ は負リテラル、θ_0は、置換。

$1\le i\le n$ について、

$\exists B_i = [P_i, -N_{i1}, \cdots, -N_{in_i}] \in S$

$\exists mgu\ \sigma_i,\ L_i\sigma_i = P_i\sigma_i.$

かつ、m段のIDbが存在する時、

$$C_m = [-N_{11}, \cdots, -N_{1n_1}, \cdots, -N_{m1}, \cdots, -N_{mn_m}]\,\theta_0+\sigma_1+\cdots+\sigma_m$$

尚、P_iは正リテラル、$-N_{ij}$は負リテラル、SはHorn節集合である。

[証明]

$$C_k = [-L_{k+1}, \cdots, -L_m, -N_{11}, \cdots, -N_{1n_1}, \cdots, -N_{k1}, \cdots, -N_{kn_k}]\,\theta_k$$

とおける。

この段階で、次は $B_{k+1} = [P_{k+1}, -N_{(k+1)1}, \cdots, -N_{(k+1)n_{k+1}}]$ と導出を行うから、

$$C_{k+1} = [-L_{k+2}, \cdots, -L_m, -N_{11}, \cdots, -N_{1n_1}, \cdots, -N_{k+1\,1}, \cdots, -N_{(k+1)1}, \cdots, -N_{(k+1)n_{k+1}}]\,\theta_k\ mgu(L_{k+1}\theta_k, P_{k+1})$$

すなわち

$$\theta_{k+1} = \theta_k(\theta_k + \sigma_{k+1})_{var(L_{k+1}\theta_k)\cup var(P_{k+1})}$$
$$= \theta_k + \sigma_{k+1}$$

$(\alpha+\beta)+\gamma = \alpha+(\beta+\gamma) = \alpha+\beta+\gamma$ より

$\theta_m = \theta_0+\sigma_1+\cdots+\sigma_m$ となり、

$$C_m = [-N_{11}, \cdots, -N_{1n_1}, \cdots, -N_{m1}, \cdots, -N_{mn_m}]\,\theta_0+\sigma_1+\cdots+\sigma_m$$ ■

命題1は、unifierについて新規の性質を立証するもので、この命題を基本に、命題2が成立し、定義3で掲げたIDDの演繹機構を立証できる。

次に、定義4で定式化したIDDcであるが、この演繹機構は、定義3との比較から明らかであるが、形式的に、Dn を節で組織化したもので、演繹の手法としては、本質的な差異はない。IDDの立証は、IDDc の立証につながる。

3.2 演繹機構の特徴

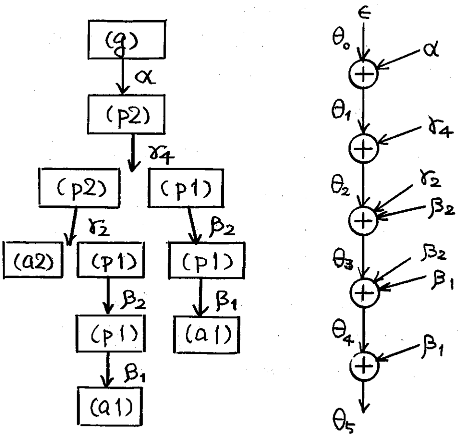


図2. AND木とunifier操作

上図2は、図1から得たもので、AND木とunifierの流れの対応を示したものである。⊕はcombination演算子を表わし、図の右側は、unifierに関するフローグラフとなっている。

図から、演繹IDDcは以下の特徴点を持つことがわかる。

a) 演繹は、AND木の深さの増加に対応して進行する。

b) AND木において、ノードは節、アークはunifierと対応する。

c) 演繹処理は、本質的には、unifierに対する操作（受理、combination演算、送出）であり、それはプログラムとして与えられたHorn節に対応するプロセスが行う。

又、この演繹機構には、２種類の並列分散性がある。１つは、⊕で記したcombination演算を除いて、ノードに分散したHorn節対応のプロセス内で独立に処理を進めることができること、もう１つは、combination演算を並列実行である。

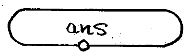
3.3 演繹機構のテンプレート表現

ここでは、IDDcの演繹処理をより具体化することを目的に置き、表現としてテンプレートを用い、明解に記述していく。例として、2章のものを再び用いる。

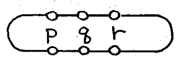
テンプレート表現は、Horn節（ANDノード）に対応するもの（CTと呼ぶことにする）とunifier（AND木のアーク）に当たるもの（UTと呼ぶ）に大別される。CT，UTの順で、以下に示す。

[CT]

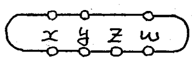
- (g) について



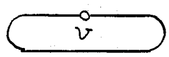
- (p1) について



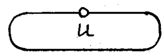
- (p2) について



- (a1) について

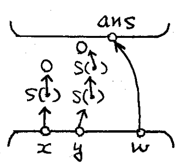


- (a2) について

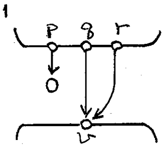


[UT]*

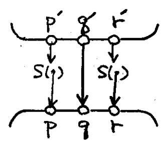
- α



- β1

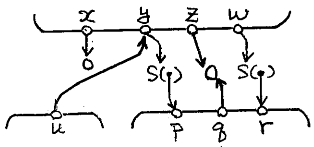


- β2

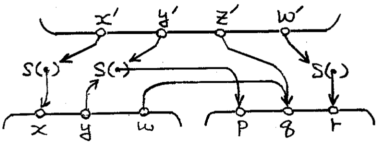


* 見やすくするため上下にCTの一部を示す。

• γ_2



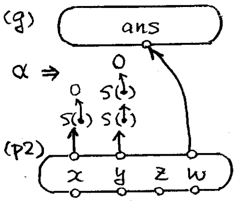
• γ_4



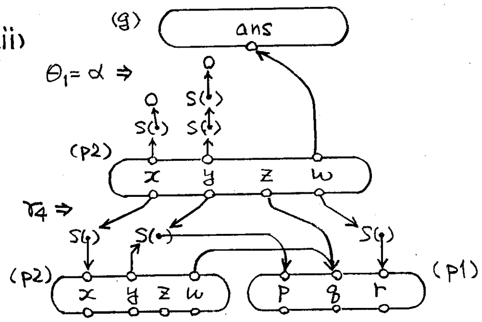
CT は、節内の変数（正リテラルは上部、負リテラルは下部）を列記した形、UT は unifier 情報をそのままグラフ表現した形をとる。

次に IDDc の演繹過程を追っていくことにする。

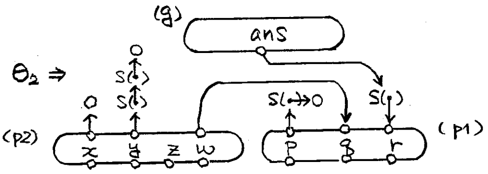
(i) IDDc 1段目は右のように得られる。



(ii)

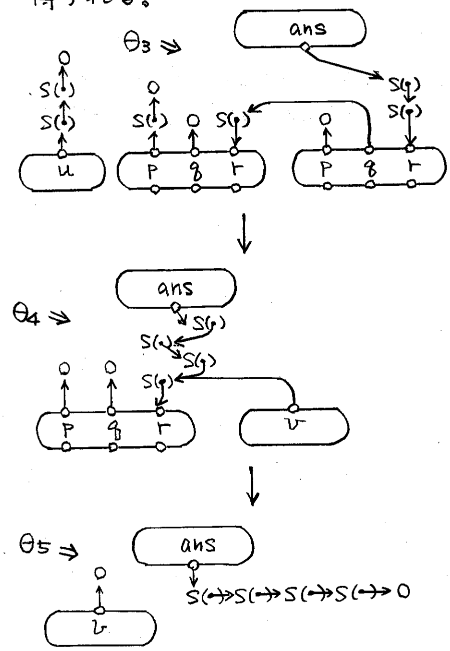


IDDc 2段目の処理は、上のように、(i) の結果に γ_4, (p2), (p1) に対応する CT, UT を連結することから始まる。



次に中央に位置する CT が取り去られる。これは、$\theta_1(=\alpha)$ と γ_4 から θ_2 を求める操作に当たる。CT を取り去る時は、上部と下部で、各変数について unification を行うが、これが、combination 演算（ここでは $\theta_1+\gamma_4$）に相当するものである。

(iii) 以下、$\theta_3 \to \theta_4 \to \theta_5$ となり、解が得られる。



4. おわりに

以上、基本的な Breadth-first の探索順序を有する入力演繹から始めて、AND木の深さ単位に演繹が進む IDD、さらに、Horn 節対応の分散化を加味した IDDc と演繹機構を定式化し、その正当性を立証した。最終的に得られた演繹機構 IDDc は、Horn 節対応に分散プロセスを生成することに加えて、処理が、unifier の操作だけで実現されるという特徴を持っている。又、演繹自体は、テンプレートを用いた表現により、明解に表現できる。この意味で、提示した演繹機構は、理論面、実際面の両面にわたり意義を持つもの

と考えている。

参考文献

[1] Chang, C.L. and Lee, C.T.L.:
Symbolic logic and mechanical theorem proving, Academic Press (1973).

[2] Malcolm, C.H. and Norman, R.:
Another generalization of resolution, JACM, Vol.25, No.3 (Jul. 1978), 341-351.