ノンホーン・マジックセット変換節数の削減手法とその評価

越村, 三幸
九州大学大学院システム情報工学研究科知能システム学専攻

長谷川, 隆三
九州大学大学院システム情報科学研究科知能システム学専攻

田中, 智浩
富士通株式会社

https://doi.org/10.15017/1523860

出版情報：九州大学大学院システム情報科学紀要. 2 (2), pp.247-252, 1997-09-26. 九州大学大学院システム情報科学研究科
バージョン：
権利関係：

九州大学大学院
システム情報科学研究科報告
第2巻 第2号 平成9年9月

Research Reports on Information Science and
Electrical Engineering of Kyushu University
Vol.2, No.2, September 1997

ノンホーン・マジックセット変換節数の削減手法とその評価

越村三幸* ・ 長谷川隆三* ・ 田中智浩**

Some Methods to Decrease the Number of Clauses Obtained by Non-Horn Magic Sets Transformation and Their Evaluation

Miyuki KOSHIMURA, Ryuzo HASEGAWA and Tomohiro TANAKA

(Received June 23, 1997)

Abstract: Non-Horn magic sets(NHM) method transforms a given clause set into a set of clauses simulating backward reasoning and those for controlling forward reasoning so as to prune the search space. To preserve the range-restricted condition, the transformation needs to attach an adornment to a predicate to show binding information of its arguments. However, introducing adornments causes combinatorial explosion of the number of transformed clauses because the number of adornments may increase exponentially. This paper presents four methods to decrease the number of transformed clauses: (1) obtaining necessary adornments by statical analysis, (2) extracting minimal adornments from necessary adornments, (3) calculating necessary adornments dynamically, and (4) transformation without adornments. These methods have been implemented on a UNIX workstation. We evaluated their effects by proving some problems in the TPTP problem library.

Keywords: Theorem proving, Bottom-up computation, Top-down computation, Magic sets, Adornment

1. 序　　論

上昇型証明と下降型証明を融合して，前向き推論型証明システムの効率を高めるノンホーン・マジックセット(NHM) [3]と呼ばれる手法が提案されている．これは，下降型証明によってゴールと関連するアトム(関連アトム)を計算し，関連アトムのみに上昇型証明を限定することにより，ゴールと無関連な推論を行う可能性のある前向き推論の欠点を補完するものである．

NHM法による証明では，入力節集合は下降型証明を模擬する節と上昇型証明を行う節に変換され，この変換節集合に対して前向き推論が適用される．変換の際に述語の引数の束縛状態を示す修飾子(adornment)が導入され，値域限定性が保たれる．値域限定性は，推論に単一化が不要になるといった実装効率上の恩恵をもたらす一方，修飾子の数は一般に指数関数的になるので，変換節数が組合せ的に増大するといった問題があった．本論文では，変換節数を削減する方法を提案する．

以降の節では，上昇型証明手法の一つであるモデル生成法とNHM法について触れた後，NHM法による節数増大の問題点を述べ，節数を削減する四つの方法を提案する．最後に，モデル生成型証明器MGTP [2]を用いた評価実験の結果を示し，提案手法の有効性を検証する．

平成9年6月23日受付

* 知能システム学専攻
** 知能システム学専攻修士課程 (現在，富士通株式会社)

2. モデル生成法と NHM 法

2.1 モデル生成法の原理

MGTPの節は，次のように含意形式で表現される：$A_1,\ldots,A_n \rightarrow B_1;\ldots;B_m$．但し，節の前件部の‘,’は連言を，後件部の‘;’は選言を表す．節は，$n=0$のとき正節と呼び，一方，$m=0$のとき負節と呼ぶ．それ以外の節$(m\neq 0, n\neq 0)$は混合節と呼ばれる．別に，$m\leq 1$なる節をホーン節，$m>1$なる節をノンホーン節と呼ぶ．また，節がアトム集合Mに基礎代入σのもとで違反しているとは，$M\models A_i\sigma \wedge M\not\models B_j\sigma$であることをいう．ここで，$M\models A_i\sigma$を満たす$\sigma$と$A_i$を見つけだす操作を連言照合と呼ぶ．

モデル生成法は，与えられた節集合に対するモデルを，空集合から始めて構成的に求める証明手法である．先ず，モデル候補の集合$\mathcal{M}$を$\{\emptyset\}$で初期化し，$\mathcal{M}$に対して，次の二つの規則の適用を繰り返す．

- モデル拡張規則：混合節もしくは正節$A_1,\ldots,A_n \rightarrow B_1;\ldots;B_m$が，あるモデル候補$M\in\mathcal{M}$に基礎代入$\sigma$のもとで違反している時，$M$の代わりに各$B_j\sigma$を$M$に加えてモデル候補を拡張する．
- モデル棄却規則：負節$A_1,\ldots,A_n \rightarrow false$が，あるモデル候補$M\in\mathcal{M}$に基礎代入$\sigma$のもとで違反している時，$M$を棄却する．

適用ができなくなったなったら手続きは終了し，$\mathcal{M}$の要素として節集合のモデルが構成される．$\mathcal{M}$が空なら，節集合は充足不能である．

なお本論文では，入力節集合は値域限定という制約を満たすものとする．値域限定とは，後件部に現れる変数は，全て前件部に現れるという制約である．これによりモデル候補の各要素は，基底であることが保証される．

2.2 ノンホーンマジックセット (NHM) 法

モデル生成法の基本は，モデル候補の下で違反節を検出し，モデル候補をその節を充足させるように拡張することである．しかしながら，違反節が複数存在する場合にどの節をモデル候補拡張に使用するかという選択基準がないため，負節(ゴール)とは無関連な節を選んでしまい，無駄なモデル候補拡張を行う可能性がある．この可能性を小さくする手法として，NHM法が提案されている[3]．

NHM法は，モデル生成法のような上昇型証明に下降型証明の利点を採り入れて，上記問題の解決を図る．すなわち，ゴールからの下降型証明によって，違反節のどれがゴールに関連しているかを判定する．これを実現するためにNHM法では，入力節を下降型証明を模擬するための節と上昇型証明を行う節に変換する．

変換法には，下降型証明を深さ優先探索で模擬するものと幅優先探索で模擬するものの二つあるが，ここでは，前者を行う深さ優先NHMを紹介する．

【定義1 深さ優先NHM変換】 節集合Sの節$A_1,\ldots,A_n \rightarrow B_1;\ldots;B_m$を次のように$n+1$個の節に変換することを，深さ優先NHM変換という．

$goal(B_1),\ldots,goal(B_m) \rightarrow goal(A_1),cont_{k,1}(V_k).$

$cont_{k,1}(V_k),A_1 \rightarrow goal(A_2),cont_{k,2}(V_k).$

$\vdots$

$cont_{k,(n-1)}(V_k),A_{n-1} \rightarrow goal(A_n),cont_{k,n}(V_k).$

$cont_{k,n}(V_k),A_n \rightarrow B_1;\ldots;B_m.$

ここで，kは変換前の元の節に付与された節番号，V_kは元の節に含まれるすべての変数の組である．

3. 修飾子の導入と変換節数の増大

入力節集合にNHM変換を施すと一般的には値域限定性が崩れる．例えば，$p(c,X,Y) \rightarrow false$にNHM変換を施すと$true \rightarrow goal(p(c,X,Y)),cont(X,Y)$なる節が得られるが，後件に現われる変数$X$と$Y$は値域限定を満たさない．この問題に対処するために，修飾子を導入して変換する方法が提案されている[3]．

これは，値域限定を満たさない変数を持つ引数を削除することによって変換節の値域限定性を保持するものである．先ほどの$goal(p(c,X,Y))$では，第2，3引数が削除され$goal(p(c))$となる．ただこのままでは，どの引数が削除されたのかわからないので，削除された引数をf(free)，そうではない引数をb(bound)とする．$goal(p(c))$では第1引数がb，第2，3引数がfなので，bffと表し，このbとfの列を修飾子と呼ぶ．そして，$goal(p(c))$の代わりに$goal(p^{bff}(c))$を用い，p^{bff}を修飾子つき述語と呼ぶ．bはその引数が基底となっていることを，fはそうではないことを表す．但し，関数項がbとなるのは，それに出現する全ての変数が基底(b)であるときに限る．

変換に修飾子を用いることにより，値域限定性は保持されるが，変換節数が一般的に組合せ的に増大する．これは，問題に現われる各述語記号に対して，その引数の数をnとすると，修飾子の数は2^n通りあることに起因する．変換節数の増大は，変換時間のみならず，証明時間の増大を引き起こすので，これを抑制する手段が必要となる．

4. 変換節数を減らす手法

本節では，前節で述べたNHM変換節数を減らす手法を四つ提案する．最初の二つは，変換に用いる修飾子を静的に絞り込むことによって，三つ目は変換時には修飾子を固定せず，証明時に修飾子の計算を行うことによって，四つ目の方法は修飾子を変換から排除することによって，変換節数を減らすことを狙う．

4.1 モード解析

あらゆる修飾子を用いて変換節を用意していたとしても，実際の証明では必ずしもそれら全てが必要となる分けではない．中には，証明がどのように行われたとしても，証明に現われ得ない修飾子も存在する．このような修飾子を前もって計算し，変換に用いなければ，変換節数を削減することができる．

このための手法としてNHM変換のためのモード解析[3]が提案されている．本節では，このモード解析をモデル生成法を用いて行う技法を示す．これは，入力節集合を変換してモード解析用節集合を作り，このモデルをモデル生成法によって計算する．証明に現われ得る修飾子は，モデル要素として求まる．

4.1.1 深さ優先NHM用モード解析手続き

入力: 節集合S.

出力: 修飾子つき述語の集合AD.

ステップ1: $AD := \{false^{\phi}\}$とおく．

ステップ2: Sの全ての節$C : A_1,\ldots,A_n \rightarrow B_1;\ldots;B_m$に対して以下の(1)から(3)の操作を行う．ここで，アトムAの述語記号をP_Aと表すことにし，$AD_j := \{P^{\alpha}_{B_j} | P^{\alpha}_{B_j} \in AD, \alpha$は修飾子$\}(1 \leq j \leq m)$とする．

(1) $\prod_{j=1}^{m} AD_j$の各要素$(P^{\alpha_1}_{B_1},\ldots,P^{\alpha_m}_{B_m})$について，$C$の複製$C'$を作り，各後件アトムの述語記号$P_{B_j}$を$P^{\alpha_j}_{B_j}$で置き換える．

(2) C'の前件部の各アトムA_iに対してA_1から順にA_nま

で以下の2-1)から2-5)の手順で修飾子β_iを求め，P_{A_i}を$P_{A_i}^{\beta_i}$で置き換えていく．

2-1) A_iが引数を持たなければ，$P_{A_i}^{\beta_i} := A_i^{\phi}$とする．

2-2) A_iの引数が基礎項ならば，その引数はbとする．

2-3) A_iの引数が変数の場合，その変数が後件部のある引数中でbと指定されているか，または，$A_1,\ldots,A_{i-1}$のいずれかに出現していれば，その引数をbとする．

2-4) A_iの引数が関数項の場合，その関数項に出現する各変数について，2-3) の処理を行い，全ての変数がbと指定されれば，その引数をbとする．

2-5) A_iの引数が上の2-2)～2-4)の条件のいずれも満たさないときは，当該引数をfとする．

(3) P_{A_n}まで置換が終了したら，$AD := AD \cup \{P_{A_1}^{\beta_1},\ldots,P_{A_n}^{\beta_n}\}$とする．

ステップ3: ステップ2の前後でADに変化がなければ，手続きは終了．そうでなければ，ステップ2を繰り返す．

修飾子は各述語記号に対して有限なので，本手続きは必ず停止し，ADに必要な修飾子が求まる．

4.1.2 モデル生成を利用したモード解析

モード解析手続きとモデル生成手続きの**Table 1**に示す対応に着目し，モデル生成を利用したモード解析手法を提案する．本手法では，入力節集合を変換し，そのモデル生成過程が，モード解析手続きを模擬するようにする．

Table 1 Correspondence between mode analysis and model generation

モード解析		モデル生成
集合 AD	↔	モデル候補
ステップ 1	↔	$\mathcal{M}$ の初期化
ステップ 2(1)	↔	連言照合
ステップ 2(3)	↔	モデル拡張
ステップ 3	↔	モデル発見の判定

先ず，ADの要素である修飾子つき述語のモデル候補要素としての表し方であるが，修飾子つき述語P^{α}を$P(\alpha')$と表記することにする．ここで，α'は修飾子α中のbを1，fを0に置き換え，1と0を‘,’(カンマ)で区切ったものである．例えば，p^{fb}は，$p(0,1)$と表記される．

ステップ2では，(1)で節Cの後件アトムに対応する修飾子つき述語の組合せ$\prod_{j=1}^{m} AD_j$を枚挙し，そのそれぞれについて，(2)でCの前件アトムの修飾子を計算し，(3)で(2)の結果得られた修飾子つき述語をADに加えている．したがって，大まかに言えば，節Cを$P'^{\alpha_1}_{B_1},\ldots,P'^{\alpha_m}_{B_m},\{Calc\} \rightarrow P'^{\beta_1}_{A_1},\ldots,P'^{\beta_n}_{A_n}$と変換すれば，ステップ2を模擬することができる．ここで，$Calc$が(2)の修飾子を計算する手続きに相当する．

変換が具体的にどのように行われるかを節$C_1 : p(f(X,Y),Z),q(X,f(Y,Z)) \rightarrow p(g(X,Z),Y);q(X,g(Z,Y))$を例に説明する．

変換節の前件$P'^{\alpha_1}_{B_1},\ldots,P'^{\alpha_m}_{B_m}$に相当するところは，$\{XZ/g(X,Z), YY/Y, XX/X, ZY/g(Z,Y)\}$と置き換えを行い，$p(XZ,YY),q(XX,ZY)$となる．この時，変数$X$が1すなわち基底となるのは$XZ$か$XX$が1となる場合であることに注意して，$X := XZ \vee XX$という制約を書き留めておくことにする．$Y$と$Z$に対しても，$Y := YY \vee ZY$，$Z := XZ \vee ZY$となる．

一方，$P'^{\beta_1}_{A_1},\ldots,P'^{\beta_n}_{A_n}$に相当するところは，$\{XY1/f(X,Y), Z1/Z, X1/X, XY1/f(X,Y)\}$と置き換えを行い，$p(XY1,Z1),q(X1,YZ1)$となる．この時，$XY1$が1となるのは，$X$と$Y$が共に1となる場合であるので，$XY1 := X \wedge Y$となる．$Z1$に対しては，$Z1 := Z$となる．

第二リテラル$q(X1,YZ1)$に出現する$X1$，$YZ1$に対しても，同様に$X1 := X$，$YZ1 := Y \wedge Z$となるが，これらの右辺に現れるX，Y，Zは，置き換え前の第一リテラルに出現しているので，$\{1/X, 1/Y, 1/Z\}$と置き換える．これは，深さ優先NHMでは，第二リテラルを証明する前に，第一リテラルの証明が終了しており，X，Y，Zの値が確定しているからである．

書き留めておいた制約をまとめて，$Calc$とすると変換節が得られる．以下では，幾つかの術語の定義を行い，この変換を形式的に定義する．

【定義 2 アトムの枠，源変数と合変数】 アトムAの基底でない各引数をAに現われない相異なる変数で置き換え，さらに，基底である引数を1で置き換えたアトムをAの**枠**という．このとき，元の引数位置に現われていた変数を置き換えた変数の**源変数**，源変数に対して置き換えた変数を**合変数**と呼ぶ．

【例 1】 アトム$p(f(a,b),g(X,Y))$の枠は$p(1,XY)$である．また，XYはXとYの合変数，XとYはXYの源変数である．

【定義 3 OR制約とAND制約】 変数の集合$\{X_1,\ldots,X_n\}(n \geq 1)$の変数$X$に対する**OR制約**とは，$X := X_1 \vee \ldots \vee X_n$のことをいう．また，**AND制約**とは，$X := X_1 \wedge \ldots \wedge X_n$のことをいう．

【定義 4 モード解析のための節変換】 節集合Sの正節以外の節$A_1,\ldots,A_n \rightarrow B_1;\ldots;B_m$を次のような節に変換することを，**モード解析のための節変換**と呼ぶ：$B'_1,\ldots,B'_m,\{Calc\} \rightarrow A'_1,\ldots,A'_n$．ここで，$B'_i$は$B_i$の，$A'_j$は$A_j$の枠である．また，$Calc$は，$B'_1,\ldots,B'_m$の各源変数に対する合変数のOR制約と，$A'_1,\ldots,A'_n$の合変数に対する源変数のAND制約の並びである．但し，A'_iの合変数のAND制約に現れる源変数の内，$A_1,\ldots,A_{i-1}$に現れる変数は，1で置き換える．

4.2 極小修飾子法

モード解析は，多くの問題に対して，変換節数削減効果があるものの，最悪の場合には，依然組合せ的に変換節数が増大する．そこで，さらに修飾子を絞り込む方法を提案する．

3引数述語$p(X,Y,Z)$の修飾子つき述語$p^{bbf}(X,Y)$と$p^{bff}(X)$を比較して見よう．元の述語で考えると$p^{bbf}(X,Y)$は$p(X,Y,_)$を$p^{bff}(X)$は$p(X,_,_)$を表している．実行時には，XとYは基底項に束縛されているので，後者は前者を包摂している．したがって，前者を排除し後者のみで推論を行っても完全性は損なわれない．この考えに基づく変換が極小修飾子のみを用いるNHM変換である．

これを3引数述語の8個の修飾子を例に説明する．8個の修飾子を上で述べた包摂関係で順序付けると**Fig. 1**のような束構造をなす．図の各節点は修飾子で枝が包摂関係を表し，枝の下の修飾子が上の修飾子を包摂している．

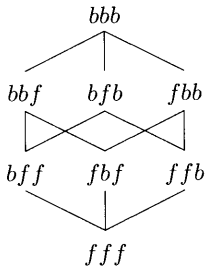

Fig.1　Lattice structure of adornments

この束構造で下の方の修飾子は，上の方の修飾子に比べて束縛が小さいので，修飾子間の順序関係($\leq$)もこれにならい，次のように定義する．

【定義 5 修飾子の順序関係($\leq$)】 ある述語の修飾子αとβに対して，$\alpha \leq \beta$は，αでbである部分は全てβでもbであることを表す．

【例 2】 $fbf \leq bbf$，$ffb \leq fbb$であるが，$bfb \leq fff$ではない．

この順序関係を用いて，修飾子の集合間に次のような順序関係を導入する．

【定義 6 修飾子集合の順序関係($\leq$)】 修飾子集合AとBに対して，$A \leq B$は，$\forall\beta \in B \exists\alpha \in A\ s.t.\ \alpha \leq \beta$であることを表す．

この関係を用いて，極小束縛部分集合と極小修飾子を次のように定義する．

【定義 7 修飾子集合の極小束縛部分集合，極小修飾子】 修飾子集合Aの極小束縛部分集合Bは，Aの部分集合Cの内，次の条件を満たす最小のものである：$C \leq A$．また，Bの要素をAの極小修飾子と呼ぶ．

【例 3】 $\{bbb, bbf, bfb, fbb, ffb\}$の極小束縛部分集合は$\{bbf, ffb\}$である．

モード解析で求めた述語の修飾子の内，極小修飾子のみを用いるNHM変換を**極小修飾子法**という．

4.3 修飾子を動的に計算する方法

モード解析は，証明に先だって証明に必要となる(可能性のある)修飾子の計算を行う．これに基づいて，証明に必要となる修飾子の組合せ分の変換節を静的に用意するため，節数が組合せ的に増大する．したがって，静的に用意するのをやめ，動的に修飾子を計算して生成すれば，変換節数を減らすことができる．

次のような節を考える．$r(X,Y) \rightarrow p(X,Y); q(X,Y)$．ここで，$p/2$，$q/2$に対する修飾子の集合が，それぞれ$\{bf, fb\}$であったとすると，修飾子つき節は，

$1: r^{bf}(X) \rightarrow p^{bf}(X); q^{bf}(X)$
$2: r^{bb}(X,Y) \rightarrow p^{bf}(X); q^{fb}(Y)$
$3: r^{bb}(X,Y) \rightarrow p^{fb}(Y); q^{bf}(X)$
$4: r^{fb}(Y) \rightarrow p^{fb}(Y); q^{fb}(Y)$

となり，これらの節に対して，NHM変換を施さなければならない．例えば節1に対する変換節は，

$1.1: goal(p^{bf}(X)), goal(q^{bf}(X)) \rightarrow goal(r^{bf}(X)), cont_{1,1}(X).$
$1.2: cont_{1,1}(X), r(X,Y) \rightarrow p(X,Y); q(X,Y).$

となり，変換節数は全体で8個となる．一方修飾子を動的に計算する方法では，

$goal(p^{\alpha}(X,Y)), goal(q^{\beta}(X,Y)), \{\gamma := f_r(\alpha,\beta)\} \rightarrow goal(r^{\gamma}(X,Y)), cont_1(X,Y).$
$cont_1(X,Y), r(X,Y) \rightarrow p(X,Y); q(X,Y).$

と変換節数は2個で済む．ここで，関数$f_r(,)$は，定義4の$Calc$に相当し，pとqの修飾子αとβから後件部のrの修飾子を(動的に)計算する関数である．

このように証明を進めながら，修飾子を計算する手法を**動的修飾法**と呼ぶ．

4.4 修飾子を用いない変換

修飾子を用いずにNHM変換を行えば，変換節数の増大は，定数倍に抑えることができる．この手法が比較的容易に行えるのは以下に述べる理由による．

修飾子を導入したのは，値域限定性を保持するためであった．値域限定性には次のように二つの利点がある．一つは，モデル候補の要素が基底であることが保証されるので単一化(unification)手続きが不要となり，照合(matching)手続きのみで証明が行え，推論速度の向上を図ることができること，もう一つは，ノンホーン節でのモデル候補拡張による証明の分岐で共有変数が生じないことである．

したがって，値域限定性が崩れた場合には，照合手続きに代えて単一化手続きを行い，さらに共有変数に対す

る処理を行わなければならない．前者は，実行効率の低下は避けられないものの原理的な困難さはない．一方，後者の共有変数の問題は，実装上大きな困難を伴う．共有変数がなければ，証明分岐後のそれぞれの分枝の証明を独立に行うことができるが，共有変数がある場合，各分枝の証明で行われる共有変数への値の代入の一貫性を管理する必要がでてくるからである．

ところで，NHM変換によって値域限定性に違反する可能性のある変数は，変換節の後件の*goal*述語に出現する変数のみである．*goal*述語は，ホーン節にしか現われないから，値域限定性を壊す可能性のある節はホーン節のみでノンホーン節は値域限定性を保持することになる．したがって，上記共有変数の問題は生ぜず，照合手続きを単一化手続きに代えるだけで，修飾子を導入しなくてもNHM変換節の証明を行うことができる．

以降，修飾子を用いないNHM変換を**無修飾子法**と呼ぶ．

5. 実 験 結 果

各種変換手法の評価を行うためにTPTP問題ライブラリ[1]のPUZ問題(全45 題)とSYN問題(全350題)を対象に，MGTPを用いて各手法の変換節削減効果と証明性能を比較した．計測には，Sun Ultra 1/140 (128MB)を用い，実行の制限時間は10分とした．

5.1 節数削減効果

Table 2に，各変換方式による変換節数の元の節数に対する増加率を示す．左から，2倍以下，2倍を超えて4倍以下，と順に2の冪乗で区切って，最右の列は128倍を超えた問題数を，手法毎に集計した．動的修飾子法と無修飾子法の変換節数は，同じになり，一番下の行に示してある．

全ての修飾子を用いる変換方式では，395題中241題について増加率が16倍を超えていることが分かる．モード解析によってこれが27題に減り，極小修飾子法を用いると16倍を超える問題はなくなる．さらに，動的修飾子法と無修飾子法では，8倍を超える問題もなくなる．

このように，モード解析は多くの問題に対して節数削減効果があることが分かる．極小修飾子法ではさらに節数の削減が達成され，ほとんど全ての問題に対して，節数の増加が数倍以内に抑えられる．動的修飾子法と無修飾子法でも同様の結果が得られている．

5.2 各方式の性能

Table 3は，PUZとSYNの全問題の内，各方式で証明できた問題数を示している．また括弧の中の数は，NHM効果のあった問題数を表している．ここでNHM効果があるとは，(1)無変換では解けず変換後に解けた，(2)証明木の枝数か節点数が無変換時の証明図に比べ減少した，のいずれかのことをいう．

Table 3 Number of Proved Problems

	PUZ	SYN
無変換	31	300
モード解析	28(10)	305(84)
極小修飾子	34(11)	309(71)
動的修飾子	28(10)	313(96)
無修飾子	33(12)	314(101)

証明できた問題数に着目すると，

- PUZ問題に対しては，極小修飾子と無修飾子では増加しているが，モード解析と動的修飾子では，逆に減少している．
- SYN問題に対しては，いずれの変換方式でも増加している．

また，NHM効果のあった問題数に着目すると，

- PUZ問題に対しては，いずれの方式にも差があまりみられない．
- SYN問題に対しては，無修飾子，動的修飾子，モード解析，極小修飾子の順に効果が減少している．

ことが分かる．

無修飾子では，下降型証明の模擬で，ゴールの引数情報を余すことなく伝搬していくのに対し，他の方法では，引数情報が失われやすい．修飾子つき述語によって，基底ではない引数が削除されるからである．ゴール情報の伝搬という点では，無修飾子が最も優れており，これが，無修飾子がNHM効果を最も示す理由であると考えられる．

Table 4は，典型的問題9題の変換節数と証明時間を示している．全ての変換方式において，

- 枝数が減少した問題は，SYN009-1，SYN349-1の2題，
- 節点数が減少した問題は，PUZ012-1，SYN009-1，SYN102-1の3題，増加した問題は，PUZ030-1の1題，
- 証明時間が減少した問題は，SYN009-1，SYN102-1，SYN312-1の3題，増加した問題は，PUZ010-1，PUZ025-1の2題，

となっている．また，PUZ030-1を除く8題では，いずれかの変換方式で，NHM効果がでている．

動的修飾子による証明では，動的に修飾子を計算し，無修飾子による証明では，照合手続きの代わりに単一化手続きを行う，オーバヘッドがそれぞれある．SYN102-1，SYN312-1では，いずれの変換方式でも証明木の大きさが同じであり，等しくNHM効果を得ていると考えることができる．したがって，証明時間差をそれらのオーバヘッドと見なすことができ，オーバヘッドは数割から数倍程度と見積もられる．PUZ010-1が，動的修飾子と無修飾子による証明で時間切れとなったのは，この

Table 2 The rate of increase in the number of transformed clauses

	2倍以下	2~4	4~8	8~16	16~32	32~64	64~128	128倍超
全	1	11	15	11	1	1	4	1
修飾子	14	41	32	29	211	8	8	7
モード	4	19	15	5	0	0	2	0
解析	103	58	76	88	15	4	5	1
極小	13	31	1	0	0	0	0	0
修飾子	202	142	5	1	0	0	0	0
動的修飾子	16	29	0	0	0	0	0	0
/無修飾子	81	265	4	0	0	0	0	0

上段：PUZ 問題 (45 題)
下段：SYN 問題 (350 題)

Table 4 Performance comparison

問題	無変換		モード解析		極小修飾子		動的修飾子		無修飾子	
PUZ010-1	128	218150	382	276300	372	270510	372	T.O.	372	T.O.
	216150(310772)		55325(627306)		55325(627071)		-		-	
PUZ012-1	18	130	86	100	47	180	32	50	32	50
	165(832)		7(92)		165(596)		7(84)		7(77)	
PUZ025-1	24	210	113	T.O.	58	310	58	T.O.	58	2050
	90(310)		-		78(969)		-		78(1033)	
PUZ030-1	43	30	193	220	98	100	98	2130	98	2090
	67(189)		67(471)		67(431)		462(2544)		426(2668)	
SYN009-1	7	1610	13	0	13	10	13	0	13	10
	19683(29526)		3(14)		3(14)		3(14)		3(14)	
SYN083-1	7	T.O.	46	T.O.	13	S.F.	13	1480	13	370
	-		-		-		1(689)		1(138)	
SYN102-1	71	4420	197	150	197	150	201	1980	201	2140
.007:007	1(3600)		1(198)		1(198)		1(198)		1(198)	
SYN312-1	8	T.O.	14	27800	14	27820	14	39350	14	39990
	-		1(9235)		1(9235)		1(9235)		1(9235)	
SYN349-1	10	1720	101	23400	28	2820	28	30580	28	180
	1152(2861)		768(13098)		768(7577)		768(12883)		8(99)	

上段左：節数
上段右：証明時間 (msec)
T.O.：時間切れ
S.F.：Segmentation Fault
下段：枝数 (節点数)

オーバヘッドが原因である．

一方，PUZ030-1が，モード解析と動的修飾子によって時間切れとなった原因は，証明が無限ループに陥ったためである．これは，証明戦略がエルブラン領域を不公平に探索しているためで，戦略を公平に行えば，制限時間内に証明が終了すると考えられる．

6. 結　　論

本論文では，組合せ的に増大するNHM変換節数を削減する四つの手法(モード解析法，極小修飾子法，動的修飾子法，無修飾子法)を提案し，その効果を実験的に確かめた．この結果，

- 各手法とも節数の削減効果がある．特にモード解析以外の手法では，全ての問題に対して節数の組合せ的増大を抑えることができる．
- 下降型証明においてゴール情報を最もよく伝搬する無修飾子法で，NHM 効果が最も顕著であり，以下，動的修飾子法，モード解析法，極小修飾子法と，伝搬の度合が小さくなるにつれて，NHM効果が薄れていく．
- 動的修飾子法と無修飾子法では，前処理がない分，動的なオーバヘッドが証明性能を劣化させる．

ことが分かった．

証明性能の点では，各手法に一長一短があるので，全ての問題を一つの手法で一律に解くのではなく，各問題を前処理によってある程度解析してから，解く手法を決定していくのが，現実的な道であると考える．例えば，モード解析によって節数が削減されない問題に対しては，極小修飾子法を適用する，といった方法である．

一方，多くの問題をNHM法で解くことによって，節数増大以外の問題点も明らかになった．一つは，証明に寄与しない負節があると，無駄な下降型証明を行ってしまう点，もう一つは，下降型証明が無限ループに陥ってしまうことがある点である．今後，これらの問題点の分析と解決を図っていく．

参 考 文 献

1) Suttner, C. and Sutcliffe, G : The TPTP Problem Library, Technical report AR-94-03, Institut für Informatik, Technische Universität München, Munich, Germany , 1994.
2) 長谷川 隆三，藤田 博：MGTP:並行論理型言語 KL1 によるモデル生成型定理証明系，情報処理学会論文誌，第 37 巻，第 1 号，pp.1-12，1996.
3) 長谷川 隆三，井上 克巳，太田 好彦，越村 三幸：上昇型定理証明の探索効率を高めるノンホーン・マジックセット，情報処理学会論文誌，第 38 巻，第 3 号，pp.453-461，1997.