数理解析研究所講究録
666巻 1988年 145-157



# 相殺公理のための完全な推論体系

Jieh Hsiang (SUNY at Stony Brook), Michael Rusinowitch (CRIN), 坂井 公 (ICOT)

## 概要

頻繁に現れる公理のうちのあるものは，それを推論メカニズムの中に組み込むことによって自動定理証明が著しく効率的になる場合がある。本論文では，相殺公理のためのそのような推論メカニズムを提示し，その完全性を示す。

## 1. はじめに

自動定理証明に研究にとっては，効率のよい推論メカニズムを捜し出すということが最重要の課題である。普通，たとえば1階述語論理というように，極めて一般的な枠組みの上での研究が主流だが，頻繁に現れる公理のうちのあるものは，仮定として与えるよりも，推論メカニズムの中に組み込んだほうが，証明が効率的になる場合がある。等式を扱うための推論法則 paramodulation は，その典型例である。また，不等式，半順序，特殊な2項関係のための推論メカニズムについても研究されている[BIH 80], [SIN 73], [MaW 85]。

近年，相殺公理のための推論メカニズムに関心が高まってきている。Stickel は，Knuth-Bendixの完備化手続きの中に相殺技法を持ち込み，$x^3 = x$ を満足する環が可換であることの自動証明を効率的に行うことに成功した[Sti 84]。WosとMcCune は，相殺によってえられる有用な結論を見つけるために negative paramodulation と呼ぶ推論法則を提案した[WoS 86]。しかし，それらの方法は，相殺を ad hoc に扱っているだけであり，完全な推論メカニズムではない。従って，入力の中から相殺公理を除去するわけにはいかず，当然それは，冗長な推論結果を生み出す可能性がある。

本論文では，3章で resolution と paramodulation を基本推論法則とする推論体系の中に相殺公理のための一般的な推論法則を追加する。4章では term 間に導入された整列順序に基づいて，さらに制限の加わった（従ってさらに無駄の少ない）推論法則を導入し，5章でその推論体系の完全性を証明する。

## 2. 準備

### 2.1 記号，記法

いま，我々は，次の記号を与えられているものとする。

(1) 可算無限個の変数。x, y, z, x1, x2, ... などで表す。その集合を V と書く。

(2) 有限個の関数記号（演算子ということもある）。f, g, ... などで表す。

その集合 F と書く。

(3) 有限個の述語記号。p, q, p1, p2, ... などで表す。その集合を P と書く。P は，特別な述語記号 = （等号）を含む。

F, P, V は互いに共通な記号を持たないものとする。関数記号，述語記号は，通常の意味での階数（引数の個数，arity, rank ）を持ち，通常どうり，項（term）と素論理式（atomic formula, atom）が定義される。階数が0の関数記号を定数（constant）と呼び，a, b, c などで表すことがある。項を表すメタ記号として s, t, u, s1, s2,... などを使う。V と F から構成される項の全体を T(F,V) で，また，変数を含まない項（ground term, 定項と呼ぶ）の全体をT(F)で表す。素論理式は，述語記号と同じ p, q などで表すが，どちらを表しているかは，普通，文脈から明らかであろう。V, F, Pから構成される素論理式の全体を A(P,F,V) で，また変数を含まない素論理式（ground atom, 定素論理式と呼ぶ）の全体をA(P,F)で表す。

= は，2引数の（間置記法を用いる）述語記号であるが，引数は可換であるとする。つまり，s=t と t=s とは同じ素論理式と考え区別しない。= を述語記号に持つ素論理式を等式と呼ぶ。

c[t], p[t] などと書いて，t を部分項として持つ項や素論理式を表す。このとき，c[], p[] と書いて c[t], p[t] から t を除いた構造をそれぞれ表し，文脈と呼ぶ。従って，c[s] は c[t] の t を s で置き換えて得られる項を表す。c[s,t], p[t,s] などの記法も同じで s と t を部分項として持つ項や素論理式を表す。ただし，この場合 s と t が重なることはない，つまり，互いに他方の部分項になっているということはないものとする。

## 2.2 resolution, factoring, paramodulation

相殺公理について述べる前に，等式を含んだ反駁型定理証明において典型的な推論法則として知られる resolution, factoring, paramodulation について概観し，等式の反射公理 x=x を例として公理と推論法則の差について若干考察することにする。

次の3つの推論法則をそれぞれ resolution (Re), factoring (Fa), paramodulation (Pa) と呼ぶ。

$$\frac{p \vee C \qquad \neg q \vee D}{(C \vee D)\theta} \; p\theta = q\theta \qquad \text{(Re)}$$

$$\frac{p \vee q \vee D}{(p \vee D)\theta} \; p\theta = q\theta \qquad \text{(Fa)}$$

$$\frac{p[s] \vee C \qquad t=u \vee D}{(p[u] \vee C \vee D)\theta} \; s\theta = t\theta \qquad \text{(Pa)}$$

各推論の結果である$(C\vee D)\theta$, $(p\vee D)\theta$, $(p[u]\vee C\vee D)\theta$ をそれぞれresolvent, factor, paramodulant という。各推論法則の直観的意味は明らかであろう。この３つの推論法則は，次の意味で完全性を持つ。Cを任意の節集合とする。Cが充足不能である（従って完全性定理により矛盾する）ときかつそのときに限り，$C\cup\{x=x\}$ から始めて(Re), (Fa), (Pa) を繰返し適用することによって空節を導くことができる。

公理 x=x を用いる代りに次の推論法則を用いてもよい。

$$\frac{\neg s=t\vee C}{C\theta}\ s\theta=t\theta \qquad \text{(Ne)}$$

明らかに$C\theta$は x=x と $\neg s=t\vee C$ との間の resolvent でもある。従って 公理 x=x があれば，(Ne) は不要である。(Ne) から x=x は，一般には導けない。しかし，公理 x=xの代わりに (Ne) を用いた推論体系も次の意味で完全である。C を任意の節集合とする。C が充足不能であるときかつそのときに限り，C から始めて (Re), (Fa), (Pa), (Ne) を繰返し適用することによって空節を導くことができる。それでは，公理 x=x と (Ne) ではどちらを採用するのが得策だろうか。明らかに x=x は他の節との間に多数のparamodulantを生成する。それらは全て「無駄な」推論であり，そのような推論を行わない (Ne) のほうがメカニズムとしては優れているといえよう。

## 3. 相殺公理のための推論法則

### 3.1 単純な相殺公理

簡単のため，本論文では右相殺公理だけを扱うことにする。左相殺公理も，同様に（対称的に）扱えばよい。* が２引数の（中置記法を用いる）関数記号で，右相殺公理

$$\forall x,y,z\ (y*x=z*x\rightarrow y=z) \qquad \text{(A1)}$$

を満足するとしよう。通常の resolution と paramodulationによる推論体系においては，上の右相殺公理は，

$$\neg y*x=z*x\vee y=z \qquad \text{(C1)}$$

という節として表現できるが，これが生み出す paramodulant や resolventの数は非常に多い。(C1)を次の推論法則に置き換えることにより，「無駄な」paramodulant や resolvent の生成が制限される。

$$\frac{s=t\vee C}{(y=z\vee C)\theta}\ s\theta=(y*x)\theta,\ t\theta=(z*x)\theta \qquad \text{(Ca)}$$

(Ca)の結論である $(y=z\vee C)\theta$ は，明らかに(C1)と s=t∨C の間のresolventである。

(C1)から生成される resolvent や paramodulant は，他にもたくさんあるが，(A1)の代りに(Ca)を採用すれば，それらの多くを生成しないで済む。(Ca)による推論の実例を挙げる。

例

```
x*x=b*a                                   x=g(x)*a
-------  a／x                             --------  y*a／x
  a=b                                     y=g(y*a)
```

3.2　単位元を持つ場合の相殺

* は，右相殺公理を満足し，しかも左単位元 e を持つとしよう。

$$\forall x\ (e*x=x) \qquad \text{(A2)}$$

(A2) と相殺公理から

$$\forall x,y\ (y*x=x \rightarrow y=e) \qquad \text{(A3)}$$

が導かれる。環論における演算子＋は，0 を単位元に持つ。Stickelは，＋の左右相殺公理から導かれる

```
y+x=x            x+y=x
-----     と     -----
 y=0              y=0
```

とを推論法則に組み込むことによって環論における自動証明の効率が飛躍的に向上することを示した[Sti 84]。

(Ca)を修正して，次の推論法則を得る。

$$\frac{s=t \vee C}{(y=e \vee C)\theta} \quad s\theta=(y*x)\theta,\ t\theta=x\theta \qquad \text{(ICa)}$$

例

```
x*x=a
-----  a／x
 a=e
```

3.3　零元を例外とする相殺

例えば整域や体のようなより豊富な代数構造に関する理論の多くでは，相殺公理は零元を除いて成立する。今，＊が右相殺公理を満足するとし，0 を左零元とする。つまり，

$\forall x\ 0*x=0$ (A4)

である。このとき，零元を例外とする相殺公理は，

$\forall x,y,z\ (\neg x=0 \vee y*x=z*x \rightarrow y=z)$ (A5)

であるが，これに対応する推論法則は，やはり(Ca)を修正して簡単に得られる。

$$\frac{s=t \vee C}{(y=z \vee C \vee x=0)\theta}\quad s\theta=(y*x)\theta,\ t\theta=(z*x)\theta \qquad \text{(NCa)}$$

例

$$\frac{x*x=b*a}{a=b \vee a=0}\ a/x \qquad\qquad \frac{x=g(x)*a}{y=g(y*a) \vee a=0}\ y*a/x$$

## 4. 強単純化順序のもとでの推論法則

### 4.1 強単純化順序

T(F,V)上の順序≧は，次を満足するとき強単純化順序（strong simplification ordering）と呼ばれる。

(1) s≧t ならば，任意の代入 $\theta$ に対して $s\theta \geqq t\theta$ である。
(2) s≧t ならば，任意の文脈 c[] に対して c[s]≧c[t] である。
(3) t が s の真部分項ならば，s＞t である。
(4) T(F) は，≧によって整列される。

さらに，T(F,V)上の強単純化順序に基づいて，A(P,F,V)上の強単純化順序を，次を満足する順序として定義する。

(1) p≧q ならば，任意の代入 $\theta$ に対して $p\theta \geqq q\theta$ である。
(2) s≧t ならば，任意の文脈 p[] に対して p[s]≧p[t] である。
(3) p が等式でなく，s，t が p の部分項ならば p＞s=t である。
　s，t がともに u のまたは v の真部分項ならば，u=v＞s=t である。
(4) A(P,F) は，≧によって整列される。

条件(4)が欠けたものを単に単純化順序という。全ての単純化順序は（従って，強単純化順序も），整礎順序(well-founded ordering)であるという重要な性質を持ち，種種の停止性問題に寄与するところが大きい。このゆえに，単純化順序やその変形は，色々提案されている[KnB 70]，[Der 82]，[JLR 82]，[Pla 78]，[Pet 83]，[HsR 83]。これらの多くは，強単純化順序であるか，簡単な修正で強単純化順序になるものであり，たくさんの具体例を提供してくれているが，本論文では，順序に関してはこれ以上の深入りはしない。

## 4.2 強単純化順序のもとでの推論法則

前章で導入した推論体系に基づく自動定理証明戦略は，各推論法則を，強単純化順序を利用して定義した「有向」推論法則に置き換えることにより，完全性を損なわずに，著しく改善することができる。。今，ある強単純化順序≧が固定されているとする。直観的には，この強単純化順序を節内のリテラルを比較するのに用い，推論によって得られる結果が既に得られているものより複雑になることの無いように，推論を進めるのである。例えば，u が t よりも，強単純化順序に関して大きいとする。等式の左右の引数は，可換であるから，制限がつかない場合，t=u を用いる paramodulation には次の２方向が有りうる。

$$\frac{p[s]\vee C \quad t=u\vee D}{(p[u]\vee C\vee D)\theta}\ s\theta=t\theta \qquad \frac{p[s]\vee C \quad t=u\vee D}{(p[t]\vee C\vee D)\theta}\ s\theta=u\theta$$

しかし，前者は paramodulant が複雑になるので禁止してしまい，後者だけを推論法則として認めようというのが基本的なアイディアである。この推論法則は，paramodulation だけでなく，簡約(reduction)の考えが組み込まれていることに注意されたい。実際，p[s] もまた等式で C や D が無いものとすれば，後者の推論法則は，Knuth-Bendix の完備化手続き[KnB 70]における要対(critical pair)の生成に他ならない。

この強単純化順序に基づく推論法則を，以下に厳密に述べよう。印刷の都合上，s<t と書いて s≧t の否定を表すことにする。t>s とは異なることに注意されたい。ただし，強単純化順序はT(F)やA(P,F)上では全順序であるから，s, t が定項や定素論理式である場合 s<t と s>t は，同じ意味になる。

$$\frac{p\vee C \quad \neg q\vee D}{(C\vee D)\theta}\ (C\vee D)\theta<p\theta=q\theta \qquad \text{(ORe)}$$

$$\frac{p\vee q\vee D}{(p\vee D)\theta}\ D\theta<p\theta=q\theta \qquad \text{(OFa)}$$

$$\frac{p[s]\vee C \quad t=u\vee D \quad u\theta<t\theta=s\theta}{(p[u]\vee C\vee D)\theta}\ \begin{array}{l}C\theta<p[s]\theta\\ D\theta<(s=t)\theta\end{array} \qquad \text{(OPa)}$$

$$\frac{\neg s=t\vee C}{C\theta}\ s\theta=t\theta,\ C\theta<(s=t)\theta \qquad \text{(ONe)}$$

ここで節Cが素論理式pに対して，C<p であるというのは，Cに含まれる任意のリテラル q または ¬q に対して q<p ということである。(ORe) を有向resolution, (OPa)を有向paramodulationと呼ぶ。

例　　p, ¬p∨q, ¬q　から空節を導く resolution には,

$$\frac{\neg p \vee q \qquad \neg q}{\dfrac{p \qquad \neg p}{\square}\ C}\ A \qquad\qquad \frac{p \qquad \neg p \vee q}{\dfrac{q \qquad \neg q}{\square}\ D}\ B$$

空節　　空節

の２つがあるが, q>p のとき, 推論 B は有向resolutionではない。また, p(s), ¬p(t), s=tから空節を導くには,

$$\frac{\neg p(t) \quad s=t}{\dfrac{p(s) \qquad \neg p(s)}{\square}\ G}\ E \qquad\qquad \frac{p(s) \quad s=t}{\dfrac{p(t) \qquad \neg p(t)}{\square}\ H}\ F$$

空節　　空節

の２つがあるが t>s のとき, 推論 F は有向paramodulationではない。従ってどちらの場合も後者は, 強単純化順序に基づく推論の場合は証明と認められない。

このように強単純化順序に基づく推論法則を用いると resolvent や paramodulant が制限され, より見通しの良い証明戦略が得られる。

4.3　強単純化順序のもとでの相殺

強単純化順序のもとでは, 右相殺のための推論法則(Ca)は, 次のように修正される。

$$\frac{s=t \vee C}{(y=z \vee C)\theta}\ s\theta=(y*x)\theta,\ t\theta=(z*x)\theta,\ C\theta<(s=u)\theta \qquad (OCa)$$

しかし, 残念ながら(OCa)だけでは完全な推論体系は得られない。そのためには, 次の推論法則を追加しなければならない。

$$\frac{s1*t1=u1 \vee C1 \qquad s2*t2=u2 \vee C2}{(s1=s2 \vee C1 \vee C2)\theta}\ u1\theta=u2\theta,\ t1\theta=t2\theta,\ C1\theta<(s1*t1=u1)\theta,\ C2\theta<(s2*t2=u2)\theta \qquad (OCa')$$

(OCa') の順序に関する条件を取り去ったものは, 次のように (Pa) と (Ca) から派生する推論法則であるが, (OCa') を (OPa) と (OCa) から導くことはできない。

$$\frac{s1*t1=u1 \vee C1 \qquad s2*t2=u2 \vee C2}{(s1*t1=s2*t2 \vee C1 \vee C2)\theta}\ u1\theta=u2\theta \qquad (Pa)$$

$$\frac{}{(s1=s2 \vee C1 \vee C2)\theta} \quad t1\theta = t2\theta \qquad (Ca)$$

単位元を持つ場合の相殺や零元を例外とする相殺に関する強単純化順序に基づく推論法則も，(ICa) や (NCa) をもとにして同様に作れる。

T(F)上の強単純化順序をもとにして，A(P,F)上の強単純化順序を作ることは簡単である。そのとき，全ての等式 p がどんな非等式 q に対しても q>p となるように順序を定めることができる。興味深いことには，このような順序のもとでは，相殺公理の適用は，等式とその否定だけからなる節に対してだけ行われる。時にこのような際立った特長を持つほど，強単純化順序のもとでの推論は大きな制限を受けるのである。

さらに次の reduction (Rd) という推論法則（推論戦略？）を導入することにより，通常のresolution, paramodulation による推論体系に較べ，有向resolution,有向paramodulation を基本とする推論体系は，組み合わせ論的爆発の程度を劇的に軽減させることができる。

$$\frac{p[u] \vee C \quad s=t}{p[u] \text{ を } p[t\theta] \text{ で置き換える}} \quad s\theta = u \qquad (Rd)$$

## 5. 完全性

完全性は，transfinite semantic tree [HsR 86] の考え方を用いて示す。A(P,F) から {0,1} への写像Iを(Herbrand)解釈と呼ぶが，特に次を満足するものをC-解釈と呼ぶ。直観的には，I(p)＝1 は p が真，I(p)＝0 は pが偽であることを表す。

(e1)　I(s=s)＝1
(e2)　I(t=s)＝1 ならば I(p[t])＝I(p[s])
(c)　I(u*s=v*s)＝1 ならば I(u=v)＝1

Iを解釈とする。I(u=v)＝1 で u>v のとき，u を部分項として含む式p[u]はI-可約であるといわれ，I によってp[v]に簡約される。これをI-簡約と呼ぶが，>は整列順序であるからI-簡約は停止性を持つ。

さらに解釈IがCT-解釈であるということを次のように定義する。

(te1)　I(s=s)＝1
(te2)　p が q にI-簡約されるなら I(p)＝I(q)
(tc)　u, v, s, t がI-既約で u>v かつ I(v*s=t)＝1 ならば I(u*s=t)＝0

<補題> 全てのC-解釈は，CT-解釈である。また，逆も成り立つ。

<証明> IをCT-解釈とする。今，ある u, v, s に対して I(u*s=v*s)＝1 にもかかわらず，I(u=v)＝0 であったとしよう。(te1), (te2) より，u, v, s はI-既約で u>v と仮定し

ても一般性を失わない。さて v*s のI-簡約による既約形を t とする。明らかに I(u*s=t)＝1 である。この u, v, s, t は (tc) の仮定を満足する。よって I(u*s=t)＝0 であるが，これは矛盾である。逆にIをC-解釈としよう。u, v, s, t がI-既約，I(v*s=t)＝1なのに I(u*s=t)＝1 であったとしよう。(e2)より I(u*s=v*s)＝1，従って (c) よりI(u=v)＝1 であるが，これは(3-2)の u のI-既約性に反する。<証明終わり>

I をCT-解釈，p を定素論理式とし，$A_p$ を強単純化順序に関して p より小さい定素論理式の全体とする。I の $A_p$ への制限 $I_p$ を部分CT-解釈という。次に全ての可能な部分CT-解釈の超限帰納法による定義を与える。便宜上 p+1 と書いて強単純化順序に関して p の次の定素論理式を表すことにする。

(1) 空解釈は，部分CT-解釈である。

(2) $I_p$ が部分CT-解釈であるとする。このとき $I_p$ の拡張として得られる部分解釈 $I_{p+1}$ には，$I_{p+1}(p)＝0$ もしくは $I_{p+1}(p)＝1$ の２つがある。

(2-1) p＝(s=s) ならば，$I_{p+1}(p)＝1$ とする拡張だけが部分CT-解釈である。

(2-2) 定項 t, s, 文脈 q[] があって p ＝q[t], t＞s, $I_p(t=s)＝1$ のとき，$I_{p+1}(q)＝I_p(q)$ とする拡張だけが部分CT-解釈である。

(2-3) p＝(u*s=t), u, v, s, t は $I_p$ -既約，u＞v, $I_p(v*s=t)＝1$ ならば，$I_{p+1}(p)＝0$ とする拡張だけが部分CT-解釈である。

(2-4) (2-1), (2-2), (2-3) のいずれでもないとき，$I_{p+1}(p)＝0$ とする拡張，$I_{p+1}(p)＝1$ とする拡張のいずれも部分CT-解釈である。

(3) p が強単純化順序に関して limit element であるとする。互いに他の拡張となっている部分CT-解釈の列 $I_q$, $I_r$, $I_s$, ... があってq, r, s, ...の極限が p であるとき，$I_q$, $I_r$, $I_s$, ... の極限 $I_p$ は，部分CT-解釈である。

上の定義は，Peterson の E-interpertation [Pet 83]を修正したものであり，それが well-defined であることの証明は [Pet 83] Theorem 2 の議論を多少修正することで得られる。しかし，先に述べた部分CT-解釈の宣言的な定義と同等であることは，直観的にはほぼ明らかであろう。

部分CT-解釈の全体は，次のような１本の木の node の全体と１対１に対応付けられる。部分CT-解釈 $I_p$ に対応するnodeからは１本または２本の枝が伸びI(p)＝0 または I(p)＝1というラベルが付けられている。root から各nodeへ至る道に付けられたラベルの全体がそのnodeにおける部分CT-解釈を表している。node $I_p$ から延びる枝が２本ある場合枝I(p)＝1 は枝 I(p)＝0 の左に書くものとする。これは，次の点で通常の semantic tree [ChL 73] より拡張されている。

(1) 各node から延びる枝が２本とは限らず１本の場合もある。
(2) 木の高さがωとは限らず，もっと大きな順序数でありうる。

これをtransfinite CT-semantic tree と呼ぶ。

今，Sを定節（変数を含まない節）の集合とする。Sの要素

$$C = p1 \vee p2 \vee \ldots \vee pm \vee \neg q1 \vee \neg q2 \vee \ldots \vee \neg qn$$

があって，

$$I_p(p1) = I_p(p2) = \ldots = I_p(pm) = 0,$$
$$I_p(q1) = I_p(q2) = \ldots = I_p(qn) = 1$$

となるとき，$I_p$ はSに関する failure nodeと呼ばれる。また，C は，$I_p$によって偽となるという。$I_p$ がSに関する右極大non-failure nodeであるとは，$I_p$ は failure nodeでないが，transfinite CT-semantic tree 上で $I_p$ の右または下にあるどの nodeもSに関するfailure node であることをいう。

さて，準備が整ったので完全性の証明に進もう。まず次の２つの補題を示す。

〈補題〉 Sを定節の集合とし，$I_p$ がSに関するfailure node で p が強単純化順序に関してlimit element とする。このとき，$I_p$ の真部分解釈でSに関するfailure nodeであるものが存在する。

〈証明〉 Sの要素 $p1 \vee p2 \vee \ldots \vee pm \vee \neg q1 \vee \neg q2 \vee \ldots \vee \neg qn$ で

$$I_p(p1) = I_p(p2) = \ldots = I_p(pm) = 0,$$
$$I_p(q1) = I_p(q2) = \ldots = I_p(qn) = 1$$

となるものがある。p1, p2, ..., pm, q1, q2, ..., qnのうち，強単純化順序に関して最大のものをq としよう。このとき，$I_p$ の $A_{q+1}$ への制限は，明らかに $I_p$ の真部分解釈でfailure node である。〈証明終わり〉

〈補題〉 Sを定節の集合とする。Sから始めて (ORe), (OFa), (OPa), (ONe), (OCa), (OCa') を繰返し適用することによって得られる節の全体を $S^*$ とする。transfinite CT-semantic tree は，$S^*$ に関する右極大non-failure nodeを $I_p$ を持たない。

〈証明〉 $I_p$ の右または下にあるどの node も $S^*$ に関するfailure node である場合，$I_p$ もまたfailure node であることを証明する。$I_p$ から延びる枝によって分類する。

(1) p＝(s=s) の場合。$I_p$ から延びる枝は１本でラベルは，I(p)＝1 である。枝の先はfailure node だから，$S^*$ の要素でそこで偽となるものがある。(OFa) により，それが ¬s=s を含むとすれば，ただ１つと仮定してよい。つまり，¬s=s∨C という形をしていると考えてよい。C＜s=s で C は $I_p$ で偽となる。ところが，(ONe)によりC は $S^*$ の要素であるから $I_p$ は，failure node である。

(2) $p=q[t]$, $t>s$ なる t と s があって $I_p$ より上の枝でラベルが$I(t=s)=1$となっている場合。そのような t, s で t=s が強単純化順序に関して最小になるように選ぶ。このとき，枝 $I(t=s)=0$ がある。もしそうでなければ，t または s がさらに上の枝のラベルにより可約であり，それは s, t の選び方に矛盾することが簡単に示される。枝 $I(t=s)=0$ の先は，$I_p$ より左にあるから failure node である。(1)と同様，そこで偽となる $S^*$ の要素を $t=s\vee C$ としてよい。$C<t=s$ で C は $I(t=s)=0$ より上で（従って当然 $I_p$ で）偽となる。一方，$I_p$ から延びる枝は，$I(p)=I(q[s])$でその先は，failure nodeである。そこで偽となる $S^*$ の要素を $p'\vee D$ としてよい。$p'$ は枝のラベルが $I(p)=1$ か $I(p)=0$かに応じて $\neg p$ か p である。また，$D<p$で D は $I_p$で偽となる。このとき，(OPa)により $q[s]'\vee C\vee D$ ($q[s]'$ は $p'=p$か $p'=\neg p$かに応じて q[s]か $\neg q[s]$) は$S^*$ の要素である。$q[s]<p$ だから，枝$I(q[s])$は$I_p$ より上にある。よって$q[s]'\vee C\vee D$ は，$I_p$ で偽となるので，$I_p$は，failure node である。

(3) $p=(u*s=t)$, $u>v$ なる$I_p$-既約な u, v, s, t があって $I_p$ より上の枝のラベルで$I(v*s=t)=1$ である場合。$I_p$ から延びる枝は，$I(p)=0$ であり，枝の先は，failure node である。そこで偽となる $S^*$ の要素を$p\vee C$ としてよい。$C<p$ で C は $I_p$ で偽となる。さらに２つの場合に分ける。

(3-1) $t=v*s$ の場合。(OCa) より$u=v\vee C$ は $S^*$ の要素である。$p>u=v$ だから枝 $I(u=v)$ は $I_p$ より上にあるが，uの既約性よりそのラベルは，$I(u=v)=0$ である。よって$u=v\vee C$ は $I_p$ で偽になるので，$I_p$ は failure node である。

(3-2) $t=v*s$ でない場合。t の $I_p$-既約性から $v*s>t$ である。また，枝 $I(v*s=t)=0$ がある。もしそうでなければ，$v*s$ または t がさらに上の枝のラベルにより可約である。それは，v, s, t が既約であることに矛盾する。枝 $I(v*s=t)=0$ の先は，$I_p$ より右にあるのでfailure node である。そこで偽となる $S^*$ の要素を$v*s=t\vee D$ としてよい。$D<v*s$で，D は $I(v*s=t)=0$ より上で（従って当然 $I_p$ で）偽となる。(OCa')により $u=v\vee C\vee D$ は，$S^*$ の要素である。(3-1) と同様，これは$I_p$ で偽となるので，$I_p$ は failure node である。

(4) $I_p$ から延びる枝が２本の場合。同様に，$I(p)=1$ 側で偽となる $S^*$ の節を $\neg p\vee C$, $I(p)=0$ 側で偽となる $S^*$ の節を $p\vee D$ としてよい。$C\vee D<p$ であり，$C\vee D$ は$I_p$ で偽となる。ところが，(ORe) により $C\vee D$ は$S^*$ の要素であるから$I_p$ は，failure node である。<証明終わり>

相殺公理のための強単純化順序に基づく推論法則の完全性は，まず次の形の定節上の定理として述べられる。これを一般の節に持ち上げる(lift)のは，強単純化順序が，代入に関して安定であることから，明らかである。

<定理> Sを定節の集合とする。Sに等号公理と * の右相殺公理を追加したものが矛盾するときかつそのときに限り，Sから始めて (ORe), (OFa), (OPa), (ONe), (OCa), (OCa') を繰返し適用することによって空節を導くことができる。

<証明> $S^*$ が空節を含まない場合，transfinite induction によって，すべての $p\in A(P,F)$ に対して，自身は failure node でないが，その右の node はすべて failure node であるような部分CT-解釈 $I_p$ を作れる。

(1) $S^*$ 空節を含まないので，空解釈は failure node でない。またその右には，いかなる node も存在しない。

(2) $I_p$ が failure node でなく，その右の node はすべて failure node とする。補題より $I_p$ の拡張で failure node でないものがある。$I_p$ に２つの拡張があり，どちらもfailure nodeでなければ，$I_{p+1}(p)=0$と拡張する。$I_p$ のfailurenode でない拡張が１つだけなら，その拡張を行う。どちらの場合も，その拡張より右にあるnodeは，すべて failure node である。

(3) p が強単純化順序に関してlimit elementとする。互いに他の拡張となっている。non-failure node の列 $I_q$, $I_r$, $I_s$, ... があって，q, r, s, ...の極限が p であり，どのnode の右にもfailure nodeしかないものとする。補題より $I_q$, $I_r$, $I_s$, ... の極限 $I_p$ は，failure nodeではありえない。また，その右にfailure nodeをもたないことも明らかである。

このようにして得られる全ての $I_p$ の極限は，CT-解釈であり，$S^*$ を充足する。従って，先の補題によりC-解釈である。C-解釈の定義における(e1), (e2) は等号公理，(c) は * に関する右相殺法則に対応する条件であるから，この解釈は $S^*$（従ってS），等号公理，* に関する右相殺法則を充足する。Sに等号公理と * の右相殺公理を追加したものが矛盾するというのが定理の仮定であったから，Herbrandの定理よりこのようなことはありえない。ゆえに $S^*$ は空節を含む。<証明終わり>

前前章の強単純化順序を導入する前の推論体系は，前章の体系を部分系として含むから当然完全である。

## 6. むすび

本論文で相殺公理を推論法則として推論メカニズムの中に組み込み効率を上げることを提案した。また，その推論法則が反駁型定理証明において完全性を持つことを示した。この技法の主な利点は，冗長な結論が生成されるのを少なくすることによって，なるべく近道の証明を捜すことができるということである。また，強単純化順序を導入して，それに基づいた推論メカニズムにより，さらに改善が図れることを示した。

相殺推論法則の導入は，反駁型定理証明だけでなく，例えばKnuth-Bendix の手続きによって項書換えシステムの完備化を効率化するというような場合にも利用できる。試作したプロトタイプ（無向完備化手続き[Hsi 87], [Sak 87] に相殺推論法則を組み込んだもの。相殺推論法則を完全な１階述語論理の反駁型定理証明システムの中に組み込んだものは，まだインプリメントしていない）による実験では，群の公理の完備化，ブール環の可換性の証明の効率を上昇させることが明らかになっている。この原因は，例えば群の公理の完備化の場合 x+((-x)+y)→y から (-x)+(x+y)→y が直ちに得られることによる。

[参考文献]

[BIH 80] W.W.Bledsoe and L.M.Hines, "Variable Elimination and Chaining in a Resolution-based Prover for Inequalities," 5th CADE, 1980, 70-87.

[ChL 73] C.L.Chang and C.T.Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, 1973.

[Der 82] N.Dershowitz, "Ordering for Term Rewriting Systems," Theoretical Computer Science, 17, 3, 1982, 279-301

[HsR 87] J.Hsiang and M.Rusinowitch, "On Word Problems in Equational Theories," ICALP 87, 1987, 54-71.

[HsR 86] J.Hsiang and M.Rusinowitch, "A New Method for Establishing Refutational Completeness in Theorem Proving," 8th CADE, 1986, 141-152.

[JLR 82] J.P.Jouannaud, P.Lescanne and F.Reinig, "Recursive Decomposition Ordering," Conf. on Formal Description of Programming Concepts II, 1982, 331-346.

[KnB 70] D.E.Knuth and P.B.Bendix, "Simple Word Problems in Universal Algebras," Computational Algebra, J. Leach (ed.), Pergamon Press, 1970, 263-297.

[MaW 85] Z.Manna and R.Waldinger, "Special Relations in Automated Deduction," 12th ICALP, 1985.

[Pet 83] G.E.Peterson, "A Technique for Establishing Completeness Result in Theorem Proving with Equality," SIAM J. of Computing, 12, 1, 1983, 82-100.

[Pla 78] D.A.Plaited, "A Recursively Defined Ordering for Proving Termination of Term Rewriting Systems," UIUCDCS-R-78-943, Univ. of Illinois, 1978.

[Sak 87] 坂井 公, 「Knuth-Bendix の完備化手続きとその応用」, コンピュータソフトウェア, 4, 1, 1987, 2-22

[SlN 73] J.Slagle and K.Norton, "Experiments with an Automatic Theorem Prover having Partial Ordering Inference Rules," Comm. ACM, 1973, 682-688.

[Sti 84] M.Stickel, "A Case Study of Theorem Proving by Knuth-Bendix Method Discovering that x*x*x=x Implies Ring Comutativity," 7th CADE, 1984, 248-258.

{WoM 86] L.Wos, and W.McCune, "Negative Paramodulation," 8th CADE, 1986, 229-239.