特　集　「定理証明，推論関係の新技術」

近年の定理自動証明技術
—システムコンペCASCとその周辺—

Recent Technologies of Automated Theorem Proving
— Around ATP System Completion CASC —

岩沼　宏治　山梨大学工学部コンピュータ・メディア工学科
Koji Iwanuma　Department of Computer Science and Media Engineering, Yamanashi University.
iwanuma@iw.media.yamanashi.ac.jp

Keywords: theorem proving, superposition, connection calculus, term indexing, strategy parallel.

1. はじめに

本論文では，一階論理の定理自動証明システムとそれを支える工学的技術の近年の進展，特に1990年代以降の進展に関して概説を行う．

定理の自動証明システムは，1960年代～70年代の前半まで盛んに研究されてきた．しかし本質的に極めて難しい問題であることから，その後研究は一時停滞期に入ったことはよく知られている．ここ20年ほど定理自動証明に関する邦文の専門書は出版されていないために，我が国ではその後の進展はあまりよく知られていないと思われる．70年末～80年代は主に，項書き換えシステム[外山 01]や一般化ユニフィケーション[Siekmann 89]などの等式証明の分野で理論的に大きな発展があった．また対話的に帰納法を用いる半自動証明システムも大きな発展を遂げている（[Bibel 98b] 第一部参照）．同時期に論理型プログラミングなどの関連分野も盛んに研究されている．このような背景のもとで，80年代後半からは工学的側面からの定理自動証明の研究が進んでいる．その結果，90年代の初めから先進的な定理自動証明システムが数多く開発され，多くの成果を生み出している．W. McCuneのEQPによるRobbins代数問題の解決[McCune 97]，ICOTで開発されたMGTPによる準群決定問題の解決[長谷川 01]などは数学分野における代表的な成果である．またハード設計・ソフトウェア工学などその他の分野でも成果が出ている[Bibel 98c]．

以上のような背景のもとに，定理自動証明の分野で最もメジャーな国際会議であるCADEの場において，定理自動証明システムコンペCASC（CADE automated theorem proving System Competition）が，1996～2000年まで毎年開催されてきた[Sutcliffe 97, Sutcliffe 00a, Sutcliffe 00b, Suttner 98, Suttner 99]．CASCの実施形態，ベンチマークテスト問題，およびシステム評価方法の妥当性などには種々の議論があるところではあるが，定理自動証明システムの開発，特に工学的な側面において競争を促進してきたことは事実と考えられる．CASCには一般の（等式入り）節集合部門，単位等式節部門，モデル生成部門などいくつかの部門があり，すべての部門において，参加システムのパフォーマンスは1996年の開催当初と比較して大幅な進歩が認められる．

本論文では，CASCで優秀な成績を納めてきた証明プログラムと工学的技術を中心にして，1990年代以降に発展してきた定理自動証明技術に関して概説する．

2. 定理証明システムコンペCASC

CASCにはいくつかの部門があり，システムの評価は部門ごとに行う．第一回から開催されている部門は，等式入りの一般の節集合を扱うMIX部門と，等式の単位節集合を扱うUEQ部門である．その後，部門が追加され，現在ではモデル生成を行うSAT部門，一般の一階式を扱うFOF部門，さらに2000年から特定の意味領域を扱うSEM部門の計5部門がある．MIX部門はさらに，ホーン節であるか否か，等式が出現するか否かなどで分類される五つの小さなカテゴリから構成されている．

コンペ参加者は登録後，主催者が用意するマシン（これまではすべてSUN Sparc）に事前に実行形式をインストールする．主催者側が本大会前に予備的な健全性テストを行い，それをパスしたシステムがコンペに本参加となる．システムの評価は部門ごとに行われ，問題ごとにある制限時間（具体的には300CPU秒）*1の中で証明を試み，時間内で証明できた問題の数の大小で評価を行う．ベンチマーク問題は，一階論理の問題ライブラリTPTP [Sutcliffe 01] から選出される．TPTPには3 000

*1 過去にメモリの使いすぎによるスラッシング現象を起こしたシステムがあり，それを防止・禁止する目的で，現在では実時間での時間制限も課せられてる．

表1 CASCでの成績上位証明プログラム

	MIX	UEQ	SAT	FOF
CASC-17 （参加総数 15システム）	E 0.6 E-SETHEO 2000csp Gandalf c-2.1	Waldmeister 600 [Waldmeister 799] SCOTT 5.0.0	GandalfSat 1.0 SCOTTSat 5.0.0 MACE 1.4b	VampireFOF 1.0 [SPASS 1.00TPTP] E-SETHEO 2000csp
CASC-16 （参加総数 16システム）	<E-SETHEO 99csp> Vampire 0.0 SPASS 1.00TPTP	Waldmeister 799 [Waldmeister 798] SCOTT	OtterMACE 437 SPASS 1.00TPTP [SPASS 0.95TPTP]	SPASS 1.00TPTP E-SETHEO-FLOTTER [SPASS 0.95TPTP]
CASC-15 （参加総数 14システム）	Gandalf c-1.1 SPASS 1.0.0a AI-SETHEO	Waldmeister 798 Otter-3.0.5 DISCOUNT/TSM	SPASS 1.0.0a MACE-1.3.2 Satchmo	SPASS 1.0.0a Otter-3.0.5 ***
CASC-14 （参加総数 18システム）	Gandalf SPASS 0.77 Otter-3.0.5	Waldmeister OTTER-3.0.5 Gandalf	SPASS 0.77 MACE-1.2 ***	SPASS 0.77 Otter-3.0.5 THINKER
CASC-13 （参加総数 18システム）	E-SETHEO Otter-Wos SETHEO	Otter 3.0.4z Waldmeister DISCOUNT/GL	— — —	— — —

注1: “[ ]”で囲まれたシステムは前年度の優勝プログラムである．CASC-16より，比較のために前年度の優勝プログラムが参加することになった．CASC-16，17の参加総数にも数えられていることに注意していただきたい．
注2: “***”は参加システムが勝ったことを示す．
注3: CASC-16のMIX部門のE-SETHEOは，大会当日には最も多くの問題の証明に成功し優勝とされたが，大会直後に実施される包括的な健全性テストで問題が見つかり，優勝を取り消された．ただし，のちに発見修正されたバグは極めてまれなケースのみに影響し，CASC-16の結果には影響がなかったことが確認されている [Sutcliffe 00a].

を越す問題が収録されているが，事前に複数の先進的自動証明システムを用いて実際にすべての証明を試みて，すべてのシステムで解けた問題と，全く解けなかった問題[*2]を取り除いておく．残りの問題の中から，適当な数のベンチマーク問題をコンペ当日にランダムに選出して利用する．

なお，過去に参加したすべてのシステムのソースコードは公開が義務付けられており，TPTPや実行結果その他のデータとともに以下の大会WWWサイトから習得できる．

http://www.cs.miami.edu/~tptp/CASC/

3. CASCへ参加してきた自動証明プログラム

CASCでの歴代の成績上位システムを表1に示す．各部門にはそれぞれ一位から三位までのシステムをあげた．紙面の都合上，昨年度開催されたCASC-17を中心に解説を加える．なお個々のシステム，またこれ以外の参加システム[*3]に関する資料は [Sutcliffe 97, Sutcliffe 00a, Sutcliffe 00b, Suttner 98, Suttner 99]，および前述のWWWサイトに詳細なポインタがあるので，参照していただきたい．

3·1 MIX 部 門

MIX部門では75題のベンチマーク問題が使用された．優勝と準優勝のEとE-SETHEOはどちらも57題の証明に成功しており，順位の差は証明に要した平均CPU時間の差である．3位のGandalf c-2.1は55題の証明に成功しており，上位との差はわずかである．Otterは多くの成果を輩出した定理自動証明プログラムであり，80年代末から90年代前半を代表するシステムである．実際，CASC-13と14では優秀な成績を残している．ただ現在では改良は終了しているようであり，CASCにはランドマークプログラムとして参加している．CASC-17のMIX部門では，Otterは8題を証明したに留まり，参加8システム中最下位となっている．これは近年の定理証明システムの進歩を如実に示す結果である．

Eの特徴として，等式証明法の一つであるsuperpositionをその基礎とする点があげられる．またシステムの効率的な実現のためには，完全弁別木（perfect discrimination tree）を項索引（term indexing）機構として用い，また複数の項を共有化して保持して部分的に並列書換えを実現している．superpositionと完全弁別木については，4章で簡単な解説を加える．次にE-SETHEO[*4]であるが，これはシステムEとSETHEOの統合システムである．SETHEOは連結タブロー（connection tableau）法を基礎としており，効率的な実行のために

*2 より正確には，充足不可能性が未解決な問題も取り除かれる．またCASCでは一階論理式にはすべて関数記号が出現する一階式のみがベンチマーク問題として使用される．これはエルブラン基底が可算無限となり，命題論理式とは本質的に異なる式をテスト問題とするためである．

*3 これまでのところ，日本からの参加はCASC-14のI-THOP [Iwanuma 97] のみである．ドイツ国からの参加が一番多く，次いで北欧と米国からの参加が多いようである．

*4 CASC-16と17のE-SETHEOと，CASC-13のE-SETHEOは別物である．後者はBrandのmodification法 [Brand 75] により等式計算を行うSETHEOシステムである．

PTTP（Prolog Technology Theorem Prover）技術[Stickel 92]を用いて，与えられた節集合を拡張WAM（Warren Abstract Machine）のコードへコンパイルを行う．また種々の補題技術を利用している．PTTP技術と補題技術についても4章で簡単に解説する．Gandalfは，一階論理，直感主義論理，タイプ理論等々に対する証明プログラム群の総称で，第3位となったGandalf-cは一階論理に対するもので，支持集合戦略，超導出，単位導出，順序付等号調整法，demodulationなどの複数の計算機構をもっている．Gandalf-cはCASC-14と15のMIX部門の優勝プログラムであるが，それはGandalf-cが初めて本格的に採用した時分割型並行多重戦略の利用に負うところが極めて大きい．この時分割並行戦略は，その後に大きな影響を与え，多くの定理証明プログラムに採用されている．これについても4章で略説を加える．

3·2 UEQ 部 門

UEQ部門は等式単位節の集合上で定理証明を行う部門である．この部門ではCASC-15以来，Waldmeisterが圧勝を続けている．CASC-17では新旧のWaldmeisterが30題中，30題と29題の証明にそれぞれ成功し，平均所要時間も40 CPU秒程度に収まっている．これに対して第3位のSCOTTは12題の証明に成功，平均所要時間も180 CPU秒程度となっており，Waldmeisterとの差はかなり大きい．ちなみに上述のEとE-SETHEOはともに8題の証明に成功し4位と5位，CASC-13同部門優勝のOtterは6題を解くに留まり，参加9システム中7位となっている．Waldmeisterが基礎とする証明法は，非失敗完備化（unfailing completion）[Bachmair 89]であり，その効率的な実現のために，完全弁別木の改良形を項索引機構として用いている．非失敗完備化については，本特集で別に解説される[外山 01]ので，ここでは深入りしない．ただUEQ部門では他にも非失敗完備化を用いる証明プログラムは多く，Waldmeisterの優秀さはその他の側面もかなり大きいことに注意していただきたい．3位のSCOTTはSemantically Constrained OTTerの略で，モデル発見器FINDERを組み込み，充足可能な部分節集合を探し（支持集合戦略と同様の原理で），Otterの導出順序の意味的誘導を行うものである．一般に充足可能な部分節集合の探索はコストが掛かるので，SCOTTの証明時間は大きめにでる傾向がある．

3·3 その他の部門

SAT部門で3位のMACEはOtterの開発者W. McCuneによる証明プログラムである．まず推論するモデルのサイズを推定し，与えられた節集合の基礎例で同じサイズのものを生成する．次にDavis-Putnam-Lovelandアルゴリズムを適用し，モデルの有無を決定するものである．FOF部門は任意の一階論理式の証明を競う部門であるが，1～3位までのプログラムはすべていったん節集合に変換してから証明を行っている．CASC-14のTHINKERは自然演繹法を直接適用しているが，参加3システム中3位に終わり，結果はあまり良くない．SPASSはsuperposition法を本格的に利用した最初の証明システムであり，過去に優秀な成績を納めている．

4. CASCでの証明戦略と工学的技術

CASCでは種々の証明戦略と技術が使用されているが，これらのうち，超導出，支持集合戦略，単位節導出などは文献[Chang 73]によく解説されている．本章ではその他の1990年以降に発展した技術について概説する．

4·1 等式証明：等号調整法とsuperposition

等式証明は1980年以降に大きく進展した分野の一つである．等式証明の手法としては等式調整法（paramodulation）がよく知られている[Chang 73, Hsiang 91]．文献[Chang 73]の記述が示すように，1970年代の前半においては，関数反射律公理と変数への等号調整の不必要性が証明できなかったが，1975年に[Brand 75]が不必要性の（間接的な）証明に成功している．

等式証明の本質は，複数の名前（項）をもつオブジェクト間の推論，すなわち人工知能のオントロジー問題と同じである．複数名称を効率的に扱うためには，対象オブジェクトに“正式な名前”という概念を導入して，統一的に扱うことが望ましい．よって名前（項）の間に順序を導入し，順序極小な項をオブジェクトの正規な名前と考えることは，極めて合理的である．以上の観点から，等号調整法は「項をより小さいものに書き換える」計算だけを行う順序付等式調整法（ordered paramodulation）[Hsiang 91]などへ改良されている．さらに，書き換える場所を“大きな項”だけに制限したbasic superposition [Bachmair 97]へ発展している．順序付等号調整法はOtterを初めとして多くの証明プログラムで使用されており，superposition法はSPASSで初めて本格的に使用され，Eが続いている．

以下にsuperposition法と順序付等号調整法の概要を示す．紙面の都合上，論理式はすべて基礎式であり．述語記号は等号“≈”だけをもつ体系を考える．また“≻”は基礎項と基礎リテラルの上の順序で，(1) 基礎リテラル上で全順序を成し，(2) 基礎項上で全順序かつ簡約順序（reduction ordering）[*5]を成し，(3) リテラル中の

*5 簡約順序≻とは，推移的かつ整礎（well-founded）で，任意の項s, t, uと代入θに対して「$s \succ t$ならば，必ず$u[s\theta] \succ u[t\theta]$」を満たす順序をいう．

$\succ$-最大の項を小さい項で置換したときに，必ず順序$\succ$上で小さいリテラルが得られる順序[*6]とする．

- Positive（ground）superposition

$$\frac{C \vee s \approx t \quad D \vee w[s] \approx v}{w[t] \approx v \vee C \vee D}$$

ただし，(i) $s \succ t$，(ii) $w \succ v$，(iii) $(s \approx t) \succ C$，(iv) $(w \approx t) \succ D$，(v) $(w \approx v) \succ (st)$ である場合．

- Negative（ground）superposition

$$\frac{C \vee s \approx t \quad D \vee w[s] \not\approx v}{w[t] \not\approx v \vee C \vee D}$$

ただし，(i) $s \succ t$，(ii) $w \succ v$，(iii) $(s \approx t) \succ C$，(iv) $(w \not\approx t) \succeq \max(D)$ である場合．

- Reflexivity resolution

$$\frac{s \not\approx s \vee C}{C}$$

ただし $(s \not\approx s) \succeq max(C)$ である場合．

- Ordered Factoring

$$\frac{C \vee s \approx t \vee s \approx t}{C \vee s \approx t}$$

ただし $(s \approx t) \succeq max(C)$ である場合．

- Equality Factoring

$$\frac{C \vee s \approx t \vee s \approx t'}{C \vee t \not\approx t' \vee s \approx t'}$$

ただし (i) $s \succ t$，(ii) $(s \approx t) \succ (s \approx t')$，かつ (iii) $(s \approx t) \succ C$ である場合．

一般の一階式に持ち上げる場合には，positive/negative superposition 規則に「s は変数ではない」などの制約が新たに加わる．

上記のすべての推論規則を用いる体系を superposition 法と呼び，equality factoring を排除した体系を strict superposition（SS）法と呼ぶ．Superposition 法と SS 法は，飽和型（saturation-based）証明法と呼ばれ，各節に推論規則をボトムアップ・網羅的に可能な限り適用していくシステムである．SS 法は通常の恒真式除去規則とは両立しない．

〔例 1〕 [Bachmair 97] 以下の節集合 S を考える．

$$\{a \approx b \vee a \approx c, b \approx c, a \not\approx b \vee a \not\approx c\}$$

S は明らかに充足不可能である．ここで項間の順序を $a \succ b \succ c$ とすれば，SS 法では S から恒真式 $b \not\approx b \vee a \not\approx c \vee a \approx c$ しか導出できない．よって，恒真式除去規則を適用すると，SS 法は不完全となる．

順序付等号調整法とは，SS 法において positive/negative superposition 規則の条件 (ii) および (v) を省略した体系である．書き換え可能な場所が増える代わりに，equality factoring が不用で，恒真式除去規則などが問題なく併用できる．なお実際の証明プログラムで使用されている順序は，計算の手間などを考慮して ad hoc なものを使うことも多い．その場合，完全性は必ずしも保証されないことに注意していただきたい．

4·2 連結タブロー：PTTP，補題と等式証明

連結タブロー法は本特集の [長谷川 01] を参照していただきたい．連結タブロー法において，拡張規則を適用する節点を閉じていない最左の節点に固定した体系をモデル消去（model elimination）法と呼ぶ．モデル消去法は Prolog に先祖導出（ancestor resolution）を導入した体系と極めて類似しており，節集合 S 上のモデル消去法での導出過程は，S から生成したある Prolog プログラム S_P に対する SLD 導出過程で模倣することが可能である．S_P は Prolog コンパイラを用いて，さらに WAM コードまたはマシンコードまで変換することができる．このコンパイルの過程でさまざまな最適化が行える．PTTP [Stickel 92] とは，以上のようにコンパイルによってモデル消去法を高速に実行する技術である．

PTTP 技術を利用した証明システムは，基本となる推論ステップは極めて高速である．しかし連結タブローやモデル消去法などのトップダウン型の定理証明法は，一般に同一ゴールの重複再計算が頻出する欠点をもつ．70 年代の段階でも，再計算の抑制のために各種の補題の利用が有用であることは認識されていた．しかし初期の実験 [Fleisig 74] では良い結果が得られず，その後しばらくの間，トップダウン型の定理証明法は表舞台から姿を消していた．近年，カット規則や補題の利用に関して多くの研究[Astrachan 92, Astrachan 97, Fuch 99, Iwanuma 97, Iwanuma 98, Iwanuma 99, Letz 94, Letz 98] が成され，再計算の抑制技術はかなり進んできた[*7]．このような背景のもとで，再び連結タブローとのトップダウン型証明法が復活している．

“成功した補題”の利用法は [長谷川 01] で解説されているので，本稿では局所的失敗補題（local failure caching）[Letz 94, Letz 98] と呼ばれる失敗記録の再利用に関して解説を行う．これはタブローにおける包摂（subsumption）枝刈り技法の一つの形態であり，オーバヘッドの少ない優れた手法である．連結タブローにおける包摂法一般に関する議論は [Baumgartner 97, Letz 94, Letz 98] を参照していただきたい．

局所的失敗補題の概念図を図 1 に示す．図 1 は節 $p(x) \vee q(x) \vee r(x)$ を含む節集合に対する節タブローであり，ゴール $q(x)$ が代入 θ_0 により反駁成功した状況で，続いてゴール $r(x)\theta_0$ 以下の反駁証明を試みている図である．$r(x)\theta_0$ の反駁証明にすべて失敗すれば，バックト

*6 許容的（admissible）と呼ばれる．厳密な定義は [Bachmair 97] を参照のこと．

*7 補題の効果については [Plaisted 94] を参照のこと．導出法や補題付タブローなどの各種の証明戦略の計算量（命題論理レベルで）の比較が成されている．

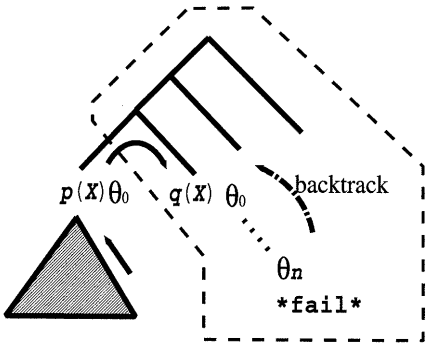

図1　局所的失敗補題

ラックして再度 $q(x)$ の別証明を試みる．成功すれば別の代入 δ_0 を生成し，ゴール $r(x)\delta_0$ の証明を試みる．このとき，もし代入 δ_0 が θ_0 の例になっていれば，$r(x)\delta_0$ の証明は必ず失敗する．よって，バックトラック時に θ_0 を“失敗補題”として記録しておけば，後に続くむだな失敗計算を抑制できる可能性がある．$q(x)$ と $r(x)$ の再計算がすべて失敗すれば，バックトラックしてさらに上位の節点の再計算に入るので，その時点で失敗補題 θ_0 を消去すればよい．残念ながら，失敗補題は局所的にしか働かない．対象をホーン節集合に限定すれば，成功と失敗双方の記憶を大域的に使用するキャッシング技術 [Astrachan 92] がすでに開発されている．

再計算の抑制技術は近年かなり進歩してきた．しかし連結タブロー法には長年にわたるもう一つの課題がある．等号計算の効率化である．

〔例2〕 以下の節集合 S は充足不可能である．

$$\{f(a,b) \approx a,\ a \approx b,\ f(x,x) \not\approx x\}$$

節 $f(x,x) \not\approx x$ を頂上節とした場合，等号調整は変数にしか適用できない．

例2[*8]より，連結タブローのようなトップダウン型証明では，変数への等号調整法を禁止すると完全性を維持できないことがわかる．さらに次の例より，順序制約とも両立できないこともわかる．

〔例3〕（例2の続き） 頂上節 $f(x,x) \not\approx x$ の変数への等号調整を許す．ただし順序制約として $f(a,b) \succ b \succ a$ を仮定[*9]し，順序付の等号調整を考える．この場合，頂上節から一回の順序付等号調整で生成される節は $f(a,a) \not\approx a$ だけである．節 $f(a,a) \not\approx a$ はこれ以上書き換えることができず，空節を導出できない．

タブローにおける効率的な等号計算は非常に難しい長年の課題である．現行 E-SETHEO に代表されるように，飽和型の順序付書換えシステム（例えば superposition）との融合は一つの有望な手法である．また変数への等号調整の禁止は，遅延評価機構の導入により部分的に対処できる [Synder 91]．Modification 法 [Brand 75] は，与えられた節集合を書き換えて等号計算を導出法で模倣する手法である．Modification 法は遅延評価と basic 法[*10]の性質を併せ持っており，連結タブロー法でしばしば使用されてきた．前述の SS 法により，modification 法への緩い（大域的な）順序制約[*11]の導入が最近可能になっている [Bachmair 98]．

4·3　時分割型並行多重戦略

時分割型並行多重戦略の有用性は，Gandalf が CASC-14 の MIX 部門で優秀したときに，この分野で改めて明確に認識されたと考えられる．

定理証明においては，適した証明戦略は問題ごとに異なる．また戦略を固定すると探索空間が一つ定まるが，解となる証明は通常一つとは限らず，探索空間に偏在している．そのため，どの証明戦略を利用するか，また探索空間をどのように探索するか，などで証明発見に要する計算時間は著しく異なってくる．このため多数のプロセッサを利用する並列定理自動証明の分野では，投入したマシンの台数効果以上の加速効果がしばしば観測され，超線形加速効果（super linear speed-up）と呼ばれてきた．また命題論理や組合せ問題の分野においては，ランダム化された再スタート型の逐次型探索が有効とされてきた．

これまで一階論理の定理証明においては，問題に適する証明戦略の推定は，問題の構文情報，または予備実行・抽象実行による意味的情報の収集に基づいて行われることが多かった．しかし，その推定には大きな限界があった．近年では時分割実行機構を利用して，複数戦略を並行実行する仕組みが利用されてきている．一つの戦略の中でパラメータを変えて並行探索する試みもある．これらの並行探索技術は，これまでいかなるやり方でも証明できなかった問題を証明可能にするものではない．基本的には，自動証明プログラムのパラメータ依存性などの挙動の不安定性を改善する技術であるが，安定化への貢献度合いは極めて大きい．

4·4　項索引機構

定理自動証明において項索引機構技術は，極めて重要な技術である．特に飽和型の自動証明においては，証明過程で極めて多数の節を生成するため，証明中盤以降の項の検索は極めて難しくなる．各節に出現する項や原子

*8 例2はもともと文献 [Synder 91] で，支持集合戦略における変数への等号調整の必要性を示す例として使用された．

*9 項 $f(a,b)$ が部分項 a と b より大きく，いわゆる subterm property を満たしており，単純化順序（simplification ordering）[Hsiang 91] を成している．

*10 等号調整により導入された項の更なる書換えを禁止する手法である．“閉包”を用いて簡潔に定義できる [Bachmair 97]．

*11 4·1節の superposition での順序制約は各推論規則に課せられた局所的なものであり，制約としては厳しいものである．

式の効率的な索引・検索機構は必須である．この重要性を明確に認識してつくられた最初の定理証明プログラムはOtterであった [McCune 92] と考えられる．その成功により，90年代以降に項索引の重要性が広く認識され，多くの自動証明プログラムで工夫されるようになった．

定理自動証明における操作対象は項である．関数記号や変数が入っているために，文書検索や関係データベースでの索引・検索に比べて，定理証明における索引・検索はかなり複雑となる．定理証明における代表的な項検索要求は以下の4種類である．

〔例4〕（**検索要求の例**） 項の集合 T と質問項 t が与えられたとき，

(1) t に単一化可能な項 $s_i \in T$ と最汎単一化代入 θ_i の組 $\langle x_i, \theta_i \rangle$ をすべて列挙する．

(2) t の一般項（generalization），すなわち $s_i\theta_i = t$ なる項 $s_i \in T$ と代入 θ_i の組 $\langle x_i, \theta_i \rangle$ をすべて列挙する．

(3) t の例（instance），すなわち $t\theta_i = s_i$ なる項 $s_i \in T$ と代入 θ_i の組 $\langle x_i, \theta_i \rangle$ をすべて列挙する．

(4) t の変種（variant）である項 $s_i \in T$ と，t と s_i の間の変数付替え代入 θ_i の組 $\langle x_i, \theta_i \rangle$ をすべて列挙する．

また証明過程において多くの節が生成・消去されることから，項の索引集合 index(T) への索引キーの追加と消去の高速性も強く望まれる．また通常，検索結果は一意ではなく，複数の項が出力され，関係データベースで言う2次検索となっている．以上からも定理自動証明における索引・検索の効率化はかなり難しいことがわかる．

項検索機構は属性ベース，集合ベース，木構造ベースの3種類に大きく分けられるが，属性ベースの索引機構はpre-filter[*12]として用いられ，正規の単一化やマッチング検査を行う候補の絞り込みに用いられる．現在多く用いられるものは後者二つである [Graf 95a]．Prologでよく使用される述語名と第一引数でのハッシュ機構は，集合ベースの検索機構に分類される．

現在，多くの証明プログラムで使用されているのは，[McCune 92] で導入された完全弁別木である．例を図2に示す．完全弁別木は，文字列の索引機構としてよく用いられるトライ（trie）を改良したものであり，弁別木の実線の各パスは（変数名が正規化された）項を表現している．一般に項は変数を含むために，変数名の違いをそのまま認めると部分構造の共有はほとんど起こらず，弁別木が巨大になる．そのため変数名を正規化し，初回の出現が項の左から数えて i 番目の変数を “$*i$” と付け替えている．

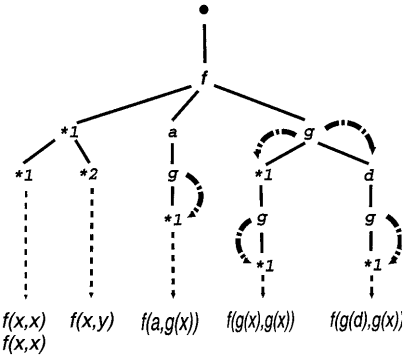

図2 ジャンプリストをもつ完全弁別木

検索は弁別木を巡回しながら行う．検索を高速化するためには，パス上に出現する各部分項の終端へ高速に移動できることが望ましい．そのため弁別木の各節点に対して（節点が変数と定数でない場合には），その節点を頂点とする部分項の終端の節点へのリンク（ジャンプリストと呼ぶ）を加えると有用である．図2では一点鎖線で示してある（図2の節点 f はすべての葉節点へリストをもつが，見やすさのために省略してある）．ただジャンプリストはメモリをかなり消費する．例えば，三つの項 $g^n(a)$, $g^n(b)$, $g^n(c)$ に対する弁別木のリンク数は $n+3$ であるが，ジャンプリストを使用すると，必要となるリンク総数は $3n+3$ 程度に跳ね上がる．

完全弁別木への新規項の挿入場所は一意に定まり，挿入は非常に高速である．しかし部分項の共有という点からはあまり性能が良くない．変数名を区別せずにすべて “$*$” へ書き換えれば，項の共有化は進み，弁別木を小さくできる．この形式を標準弁別木（standard discrimination tree）と呼ぶ．標準弁別木は，木を巡回検査する過程では変数値の検査ができないために，pre-filterとしてしか使えない．ほかにも高速で精密な技法がいくつも開発されており，[Graf 95a, Graf 95b] を参照していただきたい．この中で弁別木は通常の単一化やマッチングアルゴリズムとの親和性が極めて良く，現在でもよく使われている．Vampireのように，一つの証明プログラムの中に複数の索引機構を内蔵し，用途に応じて使い分けているものもある．

5. ま と め

本稿では1990年代以降の定理自動証明システムと技術の発展を，近年行われている証明システムのコンペCASCを軸として解説した．

CASCには問題がいくつかあることは事実である．例えばCASCに参加している自動証明プログラムはボトムアップ型あるいは飽和型と呼ばれるシステムが多い．これはベンチマークに使われる問題に数学的な問題が多いことにも深く関連している．一般に数学の問題は非常

*12 索引で引いたものすべてが所望の性質を正しくもつ索引機構はperfect-filterと呼ばれる．

にコンパクトに美しく記述されることが多いことから，証明過程において，証明目標に全く関係のない冗長な節が生成されることが少ない．また等式が問題記述の中心となることが多いために，順序付等式書換えが自然に導入できるボトムアップ型に有利である事情があると考えられる．しかしながら現実の諸問題を考えると，問題の記述は非常に長く，証明目標に対して全く関係のない記述も多数出現する冗長な式を扱う必要がある．これらの問題に対しては，ゴール指向性の強いトップダウン証明あるいはタブロー型の証明が有利であると考えられる．

応用に関する近年の傾向としては暗号プロトコルの安全性検証があげられる．また定理証明技術の最も自然かつ素直な発展としては，論理的帰結の発見枚挙問題 [Inoue 92, Iwanuma 00] がある．人工知能やハード設計・ソフトウェア工学への多くの重要な応用が考えられ，今後の発展が期待される．近年の定理自動証明技術の進展は [Bibel 98a, Bibel 98b, Bibel 98c, Robinson 01] に包括的にまとめられているので，興味ある方は参考にしていただきたい．

謝　辞

本研究で紹介した一部の研究は，通信・放送機構ならびに文部科学省科学研究費の補助を受けている．

◇ 参 考 文 献 ◇

[Astrachan 92] O. L. Astrachan and M. E. Stickel: Caching and lemmaizing in model elimination theorem provers, *Proc. of 11th Inter. Conf. on Automated Deduction (CADE-11), LNAI,* Vol. 607, pp. 778-782 (1992)

[Astrachan 97] O. L. Astrachan and D. W. Loveland: The use of lemmas in the model elimination procedure, *J. Automated Reasoning,* Vol. 19, pp. 117-141 (1997)

[Bachmair 89] L. Bachmair, N. Dershowitz and D. Plaisted: Completion without failure, in: H. Aït-Kaci and M. Nivat eds., *Resolution of Equations in Algebraic Structures,* Vol. 2, pp. 1-30, Academic Press (1989)

[Bachmair 97] L. Bachmair and H. Ganzinger: Strict Basic Superposition and Chaining, Max-Planck-Institut für Infomatik Technical Report, MPI-I-97-2-011, `http://www.mpi-sb.mpg.de` (1997). A short version: Strict Basic Superposition, *Proc. of CADE-15, LNAI,* Vol. 1421, pp. 160-174 (1998)

[Bachmair 98] L. Bachmair, H. Ganzinger and A. Voronkov: Elimination of Equality via Transformation with Ordering Constraints, *Proc. of 15th Inter. Conf. on Automated Deduction (CADE-15), LNAI,* Vol. 1421, pp. 175-190 (1998)

[Baumgartner 97] P. Baumgartner and S. Brüning: A disjunctive positive refinement of model elimination and its application to subsumption deletion, *J. Automated Reasoning,* Vol. 19, pp. 205-262 (1997)

[Bibel 98a] W. Bibel and H. Schmitt, eds.: *Automated Deduction - A basis for applications,* Vol. I: Foundations-Calculi and Methods, Kluwer (1998)

[Bibel 98b] W. Bibel and H. Schmitt, eds.: *Automated Deduction - A basis for applications,* Vol. II: Systems and Implementation Techniques, Kluwer (1998)

[Bibel 98c] W. Bibel and H. Schmitt, eds.: *Automated Deduction - A basis for applications,* Vol. III: Applications, Kluwer (1998)

[Brand 75] D. Brand: Proving theorems with the modification method, *SIAM J. Computing,* Vol. 4, No. 4, pp. 412-430 (1975)

[Chang 73] C-L. Chang and R. C-L. Lee: *Symbolic Logic and Mechanical Theorem Proving,* Academic Press (1973)
長尾，辻井，邦訳: コンピュータによる定理の証明, 日本コンピュータ協会 (1983)

[Fleisig 74] S. Fleisig, D. Loveland, A. K. Smiley III and D.L.Yarmush: An implementation of the model elimination proof procedure, *Journal of the ACM,* Vol. 21, No. 1, pp. 124-139 (1974)

[Fuchs 99] M. Fuchs: Lemma generation for model elimination by combining top-down and bottom-up inference, *Proc. of IJCAI-99,* pp. 4-9 (1999)

[Graf 95a] P. Graf: *Term Indexing, Lecture Notes in Artificial Intelligence,* Vol. 1053 (1995)

[Graf 95b] P. Graf: Substitution Indexing, *Proc. of 6th Inter. Conf. on Rewriting Techniques and Applications (RTA-95), LNCS,* Vol. 913, pp. 117-131 (1995)

[長谷川 01] 長谷川, 藤田, 越村: タブロー法とモデル生成型定理証明, 人工知能学会誌, Vol. 16, No. 5, pp. 661-667 (2001)

[Hsiang 91] J. Hsiang and M. Rusinowitch: Proving refutational completeness of theorem-proving strategies: the transfinite semantic tree method, *Journal of the ACM,* Vol. 38, pp. 559-587 (1991)

[Inoue 92] K. Inoue: Linear resolution for consequence finding, *Artificial Intelligence,* Vol. 56, pp. 301-353 (1992)

[Iwanuma 97] K. Iwanuma: Lemma matching for a PTTP-based Top-down Theorem Prover, *Proc. of 14th Inter. Conf. on Automated Deduction (CADE-14), LNAI,* Vol. 1249, pp. 146-160 (1997)

[Iwanuma 98] K. Iwanuma and K. Oota: Strong Contraction in Model Elimination Calculus: Implementation in a PTTP-Based Theorem Prover, *IEICE Transaction on Information and Systems,* Vol. E81-D, No. 5, pp. 464-471 (1998)

[Iwanuma 99] K. Iwanuma and K. Kishino: Lemma generalization and non-unit lemma matching for model elimination, *Proc. 5th Asian Computing Science Conference (ASIAN'99), LNAI,* Vol. 1742, pp. 163-176 (1999)

[Iwanuma 00] K. Iwanuma, K. Inoue and K. Satoh: Completeness of Pruning Methods for Consequence Finding Procedure SOL, *Proc. of Inter. Workshop on First-order Theorem Proving (FTP2000),* pp. 89-100, Scotland July (2000)

[Letz 94] R. Letz, C. Goller and K. Mayr: Controlled integration of the cut rule into connection tableau calculi, *J. Automated Reasoning,* Vol. 13, pp. 297-338 (1994)

[Letz 98] R. Letz: Clausal tableaux, in: W. Bibel and H. Schmitt, eds., *Automated Deduction. A basis for applications,* Vol. 1, pp. 39-68, Kluwer (1998)

[McCune 92] W. McCune: Experiments with Discrimination-Tree Indexing and path Indexing for Term Retrieval, *J. Automated Reasoning,* Vol. 9, pp. 147-167 (1992)

[McCune 97] W. McCune: Solution of the Robbins Problem, *J. Automated Reasoning,* Vol. 19, pp. 263-276 (1997)

[Plaisted 94] D. A. Plaisted: The Search Efficiency of Theorem Proving Strategies, *Proc. of 12th Inter. Conf. on Automated Deduction (CADE-12), LNAI,* Vol. 814, pp. 57-71 (1994)

[Robinson 01] A. Robinson and A. Voronkov, eds.: *Handbook of Automated Reasoning,* Vol. I, North Holland (2001)

[Siekmann 89] J. H. Siekmann: Unification Theory, *J. Symbolic Computation,* Vol. 7, No. 1, pp. 207-274 (1989)

[Stickel 92] M. E. Stickel: A prolog technology theorem prover: A new exposition and implementation in prolog, *Theoret. Comput. Sci.,* Vol. 104, pp. 109-128 (1992)

[Sutcliffe 97] G. Sutcliffe and C. Suttner, eds.: Special Issue: The CADE-13 Automated Theorem Proving System Competition, *J. Automated Reasoning,* Vol. 18, No. 2 (1997)

[Sutcliffe 00a] G. Sutcliffe: The CADE-16 ATP System Competition, *J. Automated Reasoning,* Vol. 24, pp. 371-396 (2000)

[Sutcliffe 00b] G. Sutcliffe: CADE-17 AT System Competition, http://www.cs.miami.edu/~tptp/CASC-17/ (2000)
[Sutcliffe 01] G. Sutcliffe and C. Suttner: The TPTP Problem Library for Automated Theorem Proving, http://www.cs.miami.edu/~tptp/index.html (2001)
[Suttner 98] C. Suttner and G. Sutcliffe: The CADE-14 ATP System Competition, *J. Automated Reasoning,* Vol. 21, No. 1, pp. 99-134 (1998)
[Suttner 99] C. Suttner and G. Sutcliffe: The CADE-15 ATP System Competition, *J. Automated Reasoning,* Vol. 23, pp. 1-23 (1999)
[Synder 91] W. Synder and C. Lynch: Goal Directed Strategies for Raramodulation, *Proc. of 4th Inter. Conf. on Rewriting Techniques and Applications* (*RTA-91*), *LNCS,* Vol. 448, pp. 150-161 (1991)
[外山 01] 外山: 完備化による等式証明, 人工知能学会誌, Vol. 16, No. 5, pp. 668-674 (2001)

2001年7月9日 受理

著者紹介

岩沼 宏治（正会員）

1983年東北大学通信工学科卒業．1985年同大学院修士課程修了．同年山形大学情報工学科助手．1992年より山梨大学コンピュータ・メディア工学科助教授．博士（工学）．これまで人工知能とソフトウェアの基礎，特に非単調論理と定理自動証明プログラムの研究開発に従事．近年は遺伝的プログラムや構造化テキスト文書の処理に興味をもつ．1987，89，90，91年人工知能学会全国大会優秀論文賞受賞．